Sparkflows Documentation

Release 0.0.1

Sparkflows

Jan 27, 2021

Contents

1	Architecture & Deployment 1.1 Architecture & Deployment	3 3
2	Installation 2.1 Installation	7 7
3	Configuration 3.1 Configuration	35 35
4	Authentication 4.1 Authentication	75 75
5	Security 5.1 Security	89 89
6	Operating Fire Insights 6.1 Operating Guide	97 97
7	Quick Start Guide 7.1 Quickstart Guide	107 107
8	User Guide	117 117
9	Web App User Guide 9.1 Analytical Apps User Guide	147 147
10	Data Science 10.1 Machine Learning User Guide	159 159
11	Time Series 11.1 Time Series Analysis	183 183
12	Tutorials 12.1 Tutorials	193 193
13		393

14	Frequently Asked Questions 14.1 FAQ	401 401
15	Administration 15.1 Administration Guide	409 409
16	Databricks Integration 16.1 Databricks Guide	411 411
17	AWS Integration 17.1 AWS Guide AWS Guide	435 435
18	AZURE Integration 18.1 AZURE Guide	469 469
19	Load Balancer Integration 19.1 Load Balancer	479 479
20	Superset 20.1 Superset	485 485
21	Python 21.1 Python Integration	491 491
22	Performance 22.1 Performance Tuning	501 501
23	Developer Guide 23.1 Developer Guide	503 503
24	Processors 24.1 Processors	515 515
25	Release Notes 25.1 Release Notes	707 707
26	REST API Authentication 26.1 REST API Authentication	719 719
27	REST API's using Python 27.1 REST API Examples using Python	727 727
28	REST API's using Java 28.1 REST API Examples using Java	733 733
29	REST API's using curl 29.1 REST API Examples using curl	735 735
30	Third Party Acknowledgements 30.1 Third Party Acknowledgements	773 773
31	Indices and tables	779

Fire Insights makes it incredibly fast and easy to do Self-Serve Data Preparation and Advanced Analytics.

With the power of Fire Insights at your hands, seamlessly find value from your data and scale to Petabytes of data.

Install on the cloud, on-premise or even on your laptop. Fire Insights seamlessly integrates with the most complex of Enterprise Environments.

CHAPTER 1

Architecture & Deployment

1.1 Architecture & Deployment

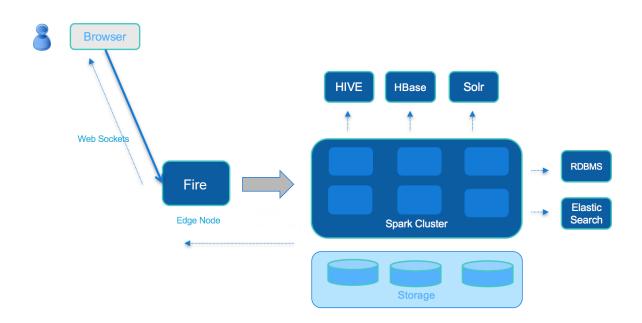
Sparkflows can be installed in one of two ways:

- On a standalone machine. In this case all the processing would happen within the single process.
 - This can be used to run Sparkflows on your laptop/desktop.
- On the Edge node of a Hadoop/Spark Cluster.
 - In this case, the jobs for processing would be submitted to the Hadoop/Spark Cluster.

1.1.1 Fire Architecture

Fire consists of three core components:

- Web Browser for defining end-to-end workflows for building data products and applications
 - Users interact with the web based drag and drop user interface for creating Datasets and Workflows
 - Workflows leverage the exhaustive set of functional and operational nodes such as Data Profiling, Data Cleaning, ETL, NLP, OCR, Machine Learning etc. displayed in the user interface.
- · Web Server running on an Edge node in a Apache Spark Cluster
 - For running the workflows, they are submitted to the web server. The web server submits the workflow to the Apache Spark cluster as a spark job using spark-submit. The results of the workflow execution are streamed back and displayed in the Browser.
 - Web Server provides a host of other features likes interactive execution, schema inference and propagation, user permissions and roles, LDP integration etc.
- · Apache Spark cluster on which the workflows are executed as Spark jobs
 - Workflows are saved in a JSON string.
 - Workflows can also be submitted on the spark cluster through spark-submit via a command line interface



1.1.2 Fire Deployment Options

Fire Insights can easily be deployed:

- On an Apache Hadoop/Apache Spark Cluster or
- On a standalone machine

Deployment on an Apache Hadoop/Apache Spark Cluster

The clusters could be based on the Apache Hadoop distribution from Cloudera, Hortonworks, MapR or any other Hadoop Cluster distributors.

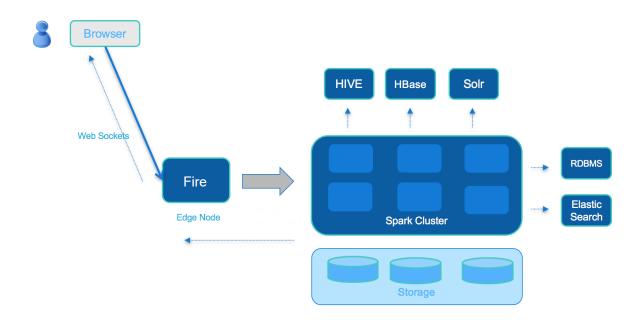
The cluster can be on-premise or on the cloud.

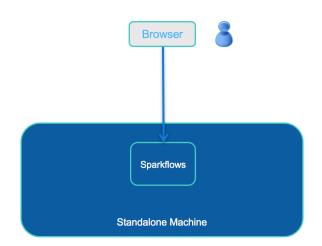
Deployment on a Standalone Machine

In this mode, Fire is installed on a mac/windows/linux machine. All the executions happen on that machine, in the web server.

This mode can be used for:

- Designing Workflows to be finally deployed on a larger Apache Spark Cluster
- For analyzing smaller sets of data





CHAPTER 2

Installation

2.1 Installation

2.1.1 Installer for laptop/desktop

You can download and use the Installer for installing/upgrading Fire Insights on your laptop or desktop. This is not recommended to use on the server, where you need better control over the Installation process.

Prerequisites

Java 8 can be downloaded and installed from here : https://www.oracle.com/java/technologies/javase-jdk8-downloads. html

You may have to set JAVA_HOME after the installation.

Download

Download the installer from : https://www.sparkflows.io/download

Execute

Execute the installer with : java -jar sparkflows-installer-1.0.jar Default port for sparkflows is : 8080 You can also change the port number while installing or starting the server. **When you finish**

[•] JDK 1.8

- Browse to http://<system-ip>:port
- Login with below credentials :
- Username : admin
- Password : admin

2.1.2 Linux/Mac OS Installation Prerequisites

Below are the Prerequisites for installing Fire Insights on a mac or linux machine:

```
JDK 1.8+ installed.
java and jar have to be in the PATH
8 GB+ of RAM.
Python 3.6+ (when running Python and PySpark, otherwise not needed)
```

If Fire would be connected to an Apache Spark Cluster:

```
Spark 2.X is needed on the clusterFire has to be installed on an Edge node of the Spark Cluster
```

If using Python and PySpark (not needed for the core features of Fire Insights)

- Python 3.X can be set up with the Python virtual environment and activated

Downloading and Installing Java 8

Java 8 can be downloaded and installed from here : https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html

You may have to set JAVA_HOME after the installation.

There are various ways for Installing Java 8 on Linux. Some are listed below.

Using Linux RPM Package

- Download the Linux x64 RPM Package
- yum localinstall jdk-8u202-linux-x64.rpm (this has to be run as the root user)

Update .bash_profile to add the below:

```
export JAVA_HOME=/usr/java/jdk1.8.0_202-amd64/
export PATH=$PATH:$JAVA_HOME/bin
```

Download OpenJDK

- https://openjdk.java.net/install/
- Install OpenJDK on Ubuntu

https://docs.datastax.com/en/jdk-install/doc/jdk-install/installOpenJdkDeb.html

2.1.3 Linux/Mac OS Installation Steps

Fire can run independently on any machine, since we package Apache Spark along with or it can be connected to a Spark cluster.

If Sparkflows Fire needs to be connected to a Spark Cluster, install it on an edge node of the cluster. The edge node has the hadoop binaries and spark configs.

Quick Installation Steps of Fire with H2 DB

- Download the fire tgz file from:
 - https://www.sparkflows.io/download OR
 - https://www.sparkflows.io/archives
- Unpack it:

```
tar xvf fire-x.y.z.tgz
```

• Create H2 DB:

```
cd <fire install_dir>
./create-h2-db.sh
```

• Launch Fire Server:

```
cd <fire install_dir>
./run-fire-server.sh start
```

• Open your web browser and navigate to:

```
<machine_name>:8080
```

• Login with:

admin/admin **or** test/test

Detailed Installation Steps

- Glossary
 - <install_dir>: location where you unzipped fire tgz file. For example this can be your home directory.
 - <machine_name> : hostname where your installed Fire
 - # : used for comments and documentation
- Download the fire tgz file from:
 - https://www.sparkflows.io/download OR
 - https://www.sparkflows.io/archives
- Unzip it:

tar xvf fire-x.y.z.tgz

• Set up H2 or MySQL DB

Fire can be configured to run with H2 db or MySQL. H2 is very easy to set up with Fire. For production deployments MySQL is recommended.

- ../database/h2-db
- ../database/mysql-db
- Launch Fire:

```
cd <fire install_dir>
./run-fire.sh start
```

• Launch Fire Server:

```
cd <fire install_dir>
./run-fire-server.sh start
```

• Test by opening your web browser and going to:

```
localhost:8080
OR
<machine_name>:8080
```

• Login with:

```
username: admin and password: admin.
```

Note: Two user accounts come preconfigured with Fire.

- admin/admin
- test/test

You may change these usernames and passwords in Fire under the menu Administration/Users

Stopping Fire

Stop Fire with the below:

./run-fire.sh stop

Stopping the Fire Server

Stop the Fire Server with the below:

./run-fire-server.sh stop

Connecting to Apache Spark Cluster

Now that you have Fire installed, you may want to connect it to your Apache Spark Cluster.

• Connecting to Apache Spark Cluster

2.1.4 Windows Installation Prerequisites

Below are the Prerequisites for installing Fire Insights on a windows machine:

```
JDK 1.8 installed.
java and jar have to be in the PATH
8+ GB of RAM on the machine.
Python 3.6+ (when running Python and PySpark, otherwise not needed)
```

Check JDK 1.8 is installed

· Check the JDK version installed on your machine:

```
Open the command window
Type the following command to check your java version : java -version
```

• If JDK 1.8 is not installed, follow the JDK installation steps mentioned below.

Install JDK 1.8

- Download JDK 1.8 for windows using the link below:
 - https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- · Install java by double clicking on the downloaded exe file
- After installation, make sure that java 1.8 is in the path:

```
Open a new command window
Type the following command to check your java version : java -version
```

Note: If you have multiple versions of Java installed on you system, you can update the PATH using the steps outlined in either of the links below:

- https://javatutorial.net/set-java-home-windows-10
- https://www.java.com/en/download/help/path.xml

Note: With the above steps, you would have Fire Insights running locally on your laptop. It would not be able to submit jobs to a Spark cluster. For that, Fire Insights has to be installed on the edge node of the cluster.

winutils.exe

winutils.exe is needed for running Apache Spark/Hadoop on windows machines. Follow the below steps to setup winutils.exe on your machine:

- Download winutils.exe **from https:**//github.com/steveloughran/winutils

winutils.exe can be directly downloaded from link below:

- https://github.com/steveloughran/winutils/blob/master/hadoop-2.7.1/bin/winutils.exe
- Create hadoop folder in Your System : C: \hadoop.

↑ 📙 → This PC → Windows (C:) → had	loop	ບ V Search h	adoop 🔎
Name	Date modified	Туре	Size
	This folder is empty.		

• Create bin folder in hadoop directory : C: \hadoop \bin.

↑ 📜 → This PC → Windows (C:) → hadoop		・ ひ Search hadoop	م ر
Name	Date modified	Туре	Size
📕 bin	04-02-2020 14:35	File folder	

• Copy the downloaded winutils.exe to the bin directory : C:\hadoop\bin\winutils.exe.

	> This PC > Windows (C:) > hadoop > bin		✓ ບ Search bin	ېر
	Name	Date modified	Туре	Size
ь.	winutils	03-02-2020 13:17	Application	107 KB
Ŀ				

- Add a new Environment Variable.
 - HADOOP_HOME = C:\hadoop.
- Update the System Environment Variable PATH by adding : %HADOOP_HOME%\bin.
- · Guide to setting Environment Variables on Windows

https://www.architectryan.com/2018/08/31/how-to-change-environment-variables-on-windows-10/

Troubleshooting

Running into an exception when saving files

org.apache.spark.SparkException: Job aborted due to stage failure: Task 1 in stage 33.0 failed 1 times, most recent failure: Lost task 1.0 in stage 33.0 (TID 131, localhost): java.io.IOException: (null) entry in command string: null chmod 0644

If you run into an exception like above, then there is problem with the setup of winutils.exe.

2.1.5 Windows Installation Steps

Fire Insights can be installed to run independenly on Windows.

Installation Steps of Fire Insights with H2 DB

- Download the fire tgz file from:
 - https://www.sparkflows.io/download OR

New System Variable		×
Variable name:	HADOOP_HOME	
Variable value:	C:\hadoop	
Browse Directory	. Browse File	OK Cancel
	(50) L :	
%HADOOP_HON	/IE%\bin	
		OK Cancel

- https://www.sparkflows.io/archives
- Unpack the downloaded tgz file. Below are some tools which can be used for it:

```
WinRar : https://www.rarlab.com/download.htm
WinZip : https://www.winzip.com
7-Zip : https://www.7-zip.org/download.html
```

• Create H2 DB:

```
cd <fire install_dir>
.\create-h2-db.bat
```

• Launch Fire Server:

```
cd <fire install_dir>
.\run-fire-server.bat start
```

• Open your web browser and navigate to:

<machine_name>:8080

• Login with:

admin/admin **or** test/test

Note: Two user accounts come preconfigured with Fire Insights.

- admin/admin
- test/test

You may change these usernames and passwords in Fire under the menu Administration/Users

Stopping the Fire Server

Stop the Fire Server with the below:

```
.\run-fire-server.bat stop
```

Stopping Fire Helper Processes

Stop Fire helper processes with the below:

```
.\run-fire.bat stop
```

2.1.6 Python Installation on Linux - Redhat/CentOS

Python is only needed if you need to use Python and the PySpark engine in Fire Insights. Python modules in Fire Insights use Python 3.7+.

Check if Python 3.7+ is Installed

Use the below commands:

```
python --version
python3.7 --version
```

Install Python 3.7 (if not installed)

Some References for Installing Python:

• CentOS : https://tecadmin.net/install-python-3-7-on-centos/

Prerequisites

Python installation requires the GCC compiler to be available on the machine. Use the following command to install the prerequisites for installing Python.

yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel

Download and extract the downloaded package

· Download python from below Link

- https://www.python.org/downloads/
- https://www.python.org/ftp/python/3.7.0/Python-3.7.0.tgz

Download and untar:

```
wget https://www.python.org/ftp/python/3.7.0/Python-3.7.0.tgz
tar xzf Python-3.7.0.tgz
```

Compile Python source code

Compile the Python source code on your system using altinstall:

```
cd Python-3.7.0
./configure --enable-optimizations
make altinstall
python3.7 --version
```

[sparkflows@python-test ~]\$ python3.7 --version Python 3.7.0

Create Python virtual environment & Activate it

Create Python virtual environment & Activate it:

```
python3.7 -m venv venv
source venv/bin/activate
python --version
```



(venv) [sparkflows@python-test ~]\$ python --version
Python 3.7.0

Upgrade pip version

Upgrade pip version with 20.0 or above:

pip install pip --upgrade

Install dependency for fbprophet package (CentOS 7)

Run below command with sudo privilege

• Install development tool:

yum install -y xz-devel

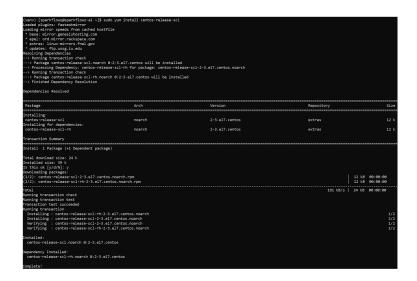
• Install the CentOS SCL release file:

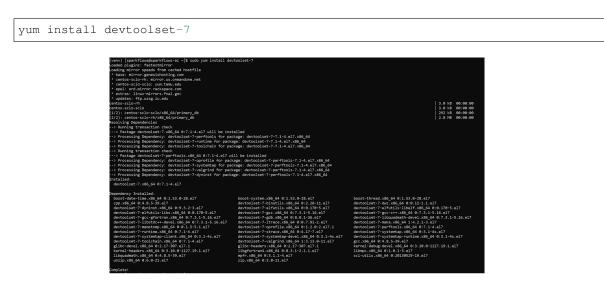
yum install centos-release-scl

• Install Developer Toolset version 7:



Loaded plugins: fastestmirror Loading mirror speeds from cac * base: mirror.genesishosting * eptl: ord.mirror.rackspace. * extras: linux-mirrors.fnal. * updates: ftp.ussg.iu.edu Resolving Dependencies > Running transaction check	ai ~]\$ sudo yum install ~y xz-devel hed hostfile .com gov h:5.2.2-1.el7 will be installed Alon			
Package	Arch	Version	Repository	Size
Installing: xz-devel	×86_64	5.2.2-1.el7	base	46 k
Transaction Summary Install 1 Package				
Total download size: 46 k Installed size: 165 k Downloading packages: xz-devel-5.2.2-1.el7.x86_64.rp	ve			46 kB 00:00:00
Running transaction check Running transaction test Transaction test succeeded Running transaction Installing : xz-devel-5.2.2-				1/1
Verifying : xz-devel-5.2.2- Installed: xz-devel.x86_64 0:5.2.2-1.el Complete!				1/1





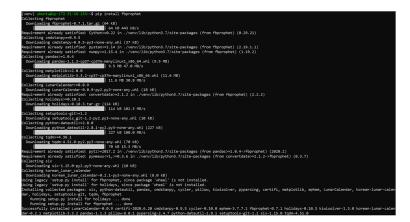
• launch a new shell instance using the Software Collection scl tool & Check GCC version:

```
scl enable devtoolset-7 bash
gcc --version``
```

(venv) [sparkflows@sparkflows-ai ~]\$ scl enable devtoolset-7 bash [sparkflows@sparkflows-ai ~]\$ gcc --version gcc (GCC) 7.3.1 20180303 (Red Hat 7.3.1-5) Copyright (C) 2017 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

• Install fbprophet package:

pip install fbprophet



• Check pip list:

pip list

Reference

	thon-test ~]\$ pip list
Package	Version
	2020.6.20
cmdstanpy	0.9.5
convertdate	2.2.2
cycler	0.10.0
Cython	0.29.21
ephem	3.7.7.1
fbprophet	0.7.1
holidays	0.10.3
kiwisolver	1.2.0
korean-lunar-calendar	0.2.1
LunarCalendar	0.0.9
matplotlib	3.3.2
numpy	1.19.2
pandas	1.1.3
Pillow	7.2.0
pip	20.2.3
PyMeeus	0.3.7
pyparsing	2.4.7
pystan	2.17.1.0
python-dateutil	2.8.1
pytz	2020.1
setuptools	39.0.1
setuptools-git	1.2
six	1.15.0
tqdm	4.50.0

Links

• https://linuxize.com/post/how-to-install-gcc-compiler-on-centos-7/

Install Other Packages

Install the required packages:

```
cd fire-x.y.x/dist/fire
pip install -r requirements.txt
```

requirements.txt file is available in the installation directory of fire insights:

fire-x.y.x/dist/fire/requirements.txt

Reference

Links

- https://docs.aws.amazon.com/cli/latest/userguide/install-linux-python.html
- https://aws.amazon.com/premiumsupport/knowledge-center/ec2-linux-python3-boto3/
- https://blog.teststation.org/centos/python/2016/05/11/installing-python-virtualenv-centos-7/

Delete a venv

To delete a virtual environment, follow below steps:

```
source venv/bin/activate
pip freeze > requirements.txt
pip uninstall -r requirements.txt -y
deactivate
rm -r venv/
```

Installing pip & wheel

- yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
- yum install python-pip
- yum install python-wheel

Add below in .bash_profile

- export PYSPARK_PYTHON=/usr/bin/python3
- export PYSPARK_DRIVER_PYTHON=/usr/bin/python3

For Ubuntu

• Ubuntu : https://docs.python-guide.org/starting/install3/linux/

2.1.7 Python Installation on MacOS

Python is only needed if you need to use Python and the PySpark engine in Fire Insights. Python modules in Fire Insights use Python 3.6+.

Check if Python is Installed

- python -version
- python3 -version

Install Python 3 (if not already there)

- One way to install Python 3 on macOS is by installing Anaconda https://docs.anaconda.com/anaconda/ install/mac-os/
- Use brew install python3

Add below in .bash_profile

- alias python='python3'
- export PYSPARK_PYTHON=/usr/bin/python3
- export PYSPARK_DRIVER_PYTHON=/usr/bin/python3
- Sometimes a soft link to Pythons's executables is broken for some reason. sudo ln -s /usr/bin/python3.x /usr/bin/python

Install Other Packages

Install the required python packages for Fire Insights:

• pip install -r requirements.txt

requirements.txt file is available in the installation directory of Fire Insights.

• fire-x.y.x/dist/fire/requirements.txt

2.1.8 Python Installation on Windows

Python is only needed if you need to use Python and the PySpark engine in Fire Insights. Python modules in Fire Insights use Python 3.6+.

Below are steps for installing Anaconda.

- Download Anaconda from the below Link
 - https://www.anaconda.com/products/individual
 - https://www.anaconda.com/products/individual#Downloads

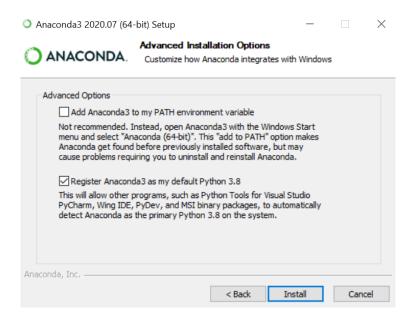
Once the download completes, run the .exe installer

Click Next to confirm the installation

	Anaconda3 2020.07 (64-)	bit) Setup		_		\times
	O ANACONDA.	(64-bit) S Setup will guide 2020.07 (64-bit It is recommend before starting	you through the t). ded that you do Setup. This will n files without h	nda3 202 he installation of ose all other app I make it possible having to reboot	Anaconda3 lications to update	
Agree to the Licens				Next >	Cance	
	Anaconda3 2020.07 (64- ANACONDA. Press Page Down to see the	License Agreeme Please review the 2020.07 (64-bit).	license terms b	— Defore installing /	Anaconda3	×
	End User License Agreemee Copyright 2015-2020, Ana All rights reserved under th This End User License Agre and Anaconda, Inc. ("Anac (which was formerly known If you accept the terms of t agreement to install Anacor	nt - Anaconda Indivi 	====== nse: your use of Ar pution). I Agree to coni	naconda Individu	al Edition	~
	Anaconda, Inc. ————		< Back	I Agree	Cano	cel

Advanced Installation Options screen

It is recommended to not check "Add Anaconda to my PATH environment variable"



Open the Anaconda Prompt from the Windows start menu

At the Anaconda prompt, check the conda --version



Reference Link

• https://problemsolvingwithpython.com/01-Orientation/01.03-Installing-Anaconda-on-Windows/

Create virtual environment using conda

Run below command to Create virtual environment using conda.

• conda create --name venv python=3.7

Activate Virtual environment and Check list of python package

Run Below command to activate and check list of python package available by default.

- conda activate venv
- python --version
- pip list

Install Other Dependent Packages

Install the other required packages:

```
(base) C:\Users>conda create --name venv python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done
 => WARNING: A newer version of conda exists. <==
 current version: 4.8.3
 latest version: 4.8.5
Please update conda by running
    $ conda update -n base -c defaults conda
## Package Plan ##
  environment location: C:\Users\NMBR\anaconda3\envs\venv
 added / updated specs:
    - python=3.7
The following NEW packages will be INSTALLED:
 ca-certificates pkgs/main/win-64::ca-certificates-2020.7.22-0
                       pkgs/main/win-64::certifi-2020.6.20-py37_0
pkgs/main/win-64::openssl-1.1.1h-he774522_0
 certifi
 openssl
                       pkgs/main/win-64::pip-20.2.3-py37_0
pkgs/main/win-64::python-3.7.9-h60c2a47_0
 pip
 python
  setuptools
                       pkgs/main/win-64::setuptools-50.3.0-py37_0
  sqlite
                        pkgs/main/win-64::sqlite-3.33.0-h2a8f88b_0
                        pkgs/main/win-64::vc-14.1-h0510ff6_4
  vc
  vs2015_runtime
                        pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
                       pkgs/main/noarch::wheel-0.35.1-py_0
pkgs/main/win-64::wincertstore-0.2-py37_0
  wheel
  wincertstore
  zlib
                        pkgs/main/win-64::zlib-1.2.11-h62dcd97_4
Proceed ([y]/n)? y
 reparing transaction: done
Verifying transaction: done
Executing transaction: done
 To activate this environment, use
      $ conda activate venv
  To deactivate an active environment, use
      $ conda deactivate
```

(base) C:\Use	ers>conda activate venv
(venv) C:\Use Python 3.7.9	ers>pythonversion
(venv) C:\Use	ers>pip list
Package	Version
certifi	2020.6.20
pip	20.2.3
setuptools	50.3.0.post20201005
wheel	0.35.1
wincertstore	0.2

• pip install -r requirements.txt

(very) C:\Users\WBM\fire\fire\3.1.0_spark_2.4\dist\fire>pip install -r requirements.txt Processing c:\users\WBM\fire\fire.3.1.0_spark_2.4\dist\fire>pip install -r requirements.txt Collecting assers\rbstword.7.1
Uiing cached astor=0.7.1.py2.py3=none=any.whl (27 kB) Processing c: \u00fcvasshirts(viasshirts)=hypdra1loca1lphp1cach=hyhels172\bla5Pid6fPB86c7dfffe46ce2be4852f38ebd84596c77c81b9\docopt=0.6.2-py2.py3=none=any.whl
Collecting generals.3.3 Using cached gest-0.3.3-py2.py3-none-any.whl (9.7 k8) Collecting groupsion_2.04.3
Using cached grpcio=1.24.3-cp37-cp37m-win_amd64.whl (1.6 MB) Collecting https://www.lan.amd64.whl (1.6 MB)
Using cached h5py-2.10.0-cp37-cp37n-win_amd64.whl (2.5 M8) Collecting jedi=0.13.3
Using Cached jedi-0.13.3-py2.py3-none-any.whl (178 k8) Collecting Kerus==2.3.1
Using cached Keras-2.3.1-py2.py3-none-any.wh1 (377 k8) Collecting Markdownw3.0.1
Using cached Markdown-3.0.1-py2.py3-none-any.wh1 (80 kB) Collecting mock=2.0.0 Using cached mock=2.0.0-py2.py3-none-any.wh1 (56 kB)
Collecting numpy=m1.17.3 Using cached numpy=1.27.3 Using cached numpy=1.27.3-cp37-cp37m-xin and64.uhl (12.7 M8)
Collecting pandas=1.1.2 Using cached pandas-1.1.2-cp37-cp37m-win_and64.whl (9.4 MB)
Collecting parson#0.3.4 Using cached parso=0.3.4-py2.py3-none-any.uhl (93 k8) Collecting bers=5.1.3
Collecting porms.1.3 Using cached pbr-5.1.3 py2.py3-none-any.whl (107 kB) Collecting prdarimami.7.1
Using cached pmdarima-1.7.1-cp37-cp37m-win_amd64.whl (607 kB) Collecting prompt-toolkit==2.0.9
Using cached prompt_toolkit-2.0.9-py3-none-any.whl (337 kB) Collecting protobuf==3.9.2
USing cached protobul-3.9.2-cp37rcp37m-win_amd64.whl (1.0 MB) Collecting ptpython=2.0.4 USing cached ptpython=2.0.4-py3-none-amy.whl (47 kB)
Collecting pydjine0.18.7 Using cached pydjine0.18.7 Using cached pydjine0.18.7 pyd.pyd;none-any.uhl (197 k8)
bollecting pyssite(8.5.0)=0.4.6 Using cached pyssite(8.5.0)=20,073-none-any.whi (77.40)
Collecting cauthlib>=3.0.0 Using cached cauthlib>∃.1.0-py2.py3-none-any.whl (147 kB)
Installing collected packages: six, absl-py, astor, docopt, gast, gropcio, numpy, h5py, parso, jedi, keras-preprocessing, scipy, pyyanl, keras-applications, Keras, Markdow pbr, mock, pytz, python-dateutil, pandas, setuptools, jobils, urilis3, Cython, scikit-learn, patsy, statsmodels, pndarima, wcwidth, prompt-toolkit, protobur/, tygemets, p thom. pwdi, pysark, ternocion, urast, astrumanse, not-sinsur, insorbeard-laulin-mit, Merkeuz, avaarl, avaarl-andulas, carbetools, raa, acoula-auth, aauthilb, idma, ch
Jet, requests, requests-oauthlib, google-auth-oauthlib, tensorboard, tensorflow-estimator, google-pasta, tensorflow, tornado, futures, jmespath, docutils, botocore, boto3 cixisolver, pyparsing, pillow, cycler, matplotlib, scorecardpy, typing-extensions, JPype1, JayDeBekpi, category-encoders, future, tabulate, colorama, h2o
Attempting uninstall: setuptools Found existing installation: setuptools 50.3.0.post20201005 Uninstalling setuptools 50.3.0.post20201005:
Uninstalling Setuptools 96.30.post2020005 Successfully uninstalled setuptools-50.3.0.post20201005 Successfully installed (ython-0.29.17 Pipuai-1.0.2 ZwQ6B4Api-1.2.1 Keras-2.3.1 Markdown-3.0.1 Pygments-2.3.1 Werkzeug-0.14.1 abs1-py-0.7.1 astor-0.7.1 asturparse-1.6.3 b
d-1.0 b obscore-1.0 0 cachetools-4.1.: category-encoders-2.2.2 chandet-3.0 4 colorame-0.4.3 cyclar-0.10.0 docopt-0.6.2 docutls-0.16 future-0.18.2 futures-2.2.0 gast-0. B coogle-auth-1.2.2. google-auth-auth-10-6.1. google-pasta-0.2.0 gprice1.2.4.3 Nor3-3.0.1.2 h5pr-2.10.0 idine-2.10 jedi-0.13.3 (mespath-0.7.1 jobilb-0.14.1 kerns-applica D coogle-auth-1.2.2.1 google-auth-auth-10-6.1. google-pasta-0.2.0 gprice1.2.4.3 Nor3-3.0.1.2 h5pr-2.10.0 idine-2.10.0 idine-2.10 jedi-0.13.3 (mespath-0.7.1 jobilb-0.14.1 kerns-applica D coople-auth-1.2.2.1 google-auth-auth-1.2.1 google-auth-0.14.3 Nor3-3.0.12 h5pr-2.10.0 idine-2.10.0 idine-2.10 jobilb-0.14.1 kerns-applica D coople-auth-1.2.2 jobilb-0.14.3 (mespath-0.14.3 kerns-1.0 k
moil@ keras-preprocessing=1.1.2 kiusiouv=-1.2.0 matplotlb=3.2. mock-3.0 μmpy-1.17.3 outhlb=3.10 opt-sins=3.3 g pands=1.1.2 parso=0.3.4 paty=0.5.1 pbr=1.0 plion=7.2.0 pdiminum_1.1.1 parso=0.3.4 paty=0.5.1 pbr=1.0 plion=7.2.0 pdiminum_1.1.2 parso=0.3.4 paty=0.5.1 pbr=1.0 plion=7.2.0 pdiminum_1.1.2 parso=0.3.4 paty=0.5.1 pbr=1.0 plion=7.2.0 pdiminum_1.2.0 pdiminu
 tabulate-0.8.7 tensorboard-2.3.0 tensorboard-plugin-wit-1.7.0 tensorflow-2.3.0 tensorflow-estimator-2.3.0 termcolor-1.1.0 tornado-6.0.1 typing-extensions-3.7.4.3 url 0.1.35.10 veroith.0.1 7. unent-1.1.2 veroith.0.1

Install dependency for fbprophet package (Windows 10)

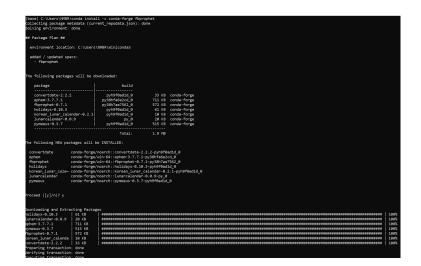
Install pystan:

• conda install pystan -c conda-forge

Install fbprophet:

• conda install -c conda-forge fbprophet

	tadata (curi	III pystam -c conda-forge modata-json): done	
WARNING: A newer current version: 4.3 latest version: 4.8	8.3	conda exists. <==	
Please update conda b	y running		
\$ conda update -n	base -c de	faults conda	
## Package Plan ##			
environment location	n: C:\Users	\MMBR\miniconda3	
added / updated spe - pystan			
m2w64-pkg-config-0.2	469 KB		100%
m2w64-headers-git-5.	5.6 MB	***************************************	100%
m2w64-isl-0.16.1	655 KB		100%
scipy-1.5.0	12.0 MB		100%
libnetcdf-4.7.4	601 KB		100%
m2w64-binutils-2.25.	44.3 MB	***************************************	100%
m2w64-libwinpthread-	30 KB		100%
pillow-7.2.0	736 KB	***************************************	100%
m2w64-windows-defaul	3 KB		100%
intel-openmp-2019.4 msys2-conda-epoch-20	1.4 MB 2 KB		100%
msys2-conda-epoch-20 m2w64-gcc-libs-5.3.0	2 KB 518 KB		100%
m2w64-gcc-ada-5.3.0	33.5 MB		100%
libssh2-1.9.0	218 KB		1005
m2w64-winothreads-gi	45 KB		100%
libcurl-7.71.1	278 KB		100%
tornado-6.0.4	646 KB		100%
cycler-0.10.0	9 KB		100%
bzip2-1.0.8	148 KB		100%
xarray-0.16.1	529 KB		100%
m2w64-libiconv-1.14	1.5 MB		100%
kiwisolver-1.2.0	62 KB		100%
Preparing transaction			
Verifying transaction			
Executing transaction	: cone		



Check the version of fbprophet Installed:

• pip list

(base) C:\Users\NMBR>p	ip list
Package	Version
arviz	0.10.0
certifi	2020.6.20
cffi	1.14.0
cftime	1.2.1
chardet	3.0.4
conda	4.8.5
conda-package-handling	1.7.0
convertdate	2.2.2
cryptography	2.9.2
cycler	0.10.0
Cython	0.29.21
ephem	3.7.7.1
fbprophet	0.7.1
holidays	0.10.3
idna	2.9
kiwisolver	1.2.0
korean-lunar-calendar	0.2.1
LunarCalendar	0.0.9
matplotlib	3.3.2

Once the above steps have completed successfully, run the below command to ensure everything was setup correctly.

• python ./dist/__main__.py

Using TensorFlow backend. Starting the PySpark Server Starting PySpark Server on port : 8085 PySpark Server is listening on port : 8085

Enable PySpark Engine in Fire Insights

Login to Fire Insights application and go to configurations and set app.enablePySparkEngine to true and save the changes. Now you can start using PySpark engine in Fire Insights.

Removing Conda virtual Environment

- conda deactivate
- conda env remove --name name of virtual environment
- Delete those package from exact location.

	INFER HADOOP CLUSTER CONFIG			Q рузр
PP DATABERCKS AIRFLOW AWS SPARK HDFS HADOOP YARN HIVE KERBEROS LDAP ALERT PLUGHIS MODULE UI-SETTINGS				
AME	TITLE	VALUE		DESCRIPTION
app.firePySparkHostPort	http://host.port of Pyspark Fire Processes to connect to. default: http://localhost.8085	http://localhost/8086	ß	http://host.port.separated.by.comma.of.the Pyspark Fire Processes to connect to
app.enablePySparkEngine	Enable/Disable pyspark engine	true	8	Enable/Disable pyspark engine.

2.1.9 Python Installation on Ubuntu

Python is only needed if you need to use Python and the PySpark engine in Fire Insights. Python modules in Fire Insights use Python 3.7+.

Check if Python 3.7+ is Installed

Use the below commands:

python --version python3.7 --version

Install Python 3.7 (if not installed)

Some References for Installing Python:

• Ubuntu : https://linuxize.com/post/how-to-install-python-3-7-on-ubuntu-18-04/

Prerequisites

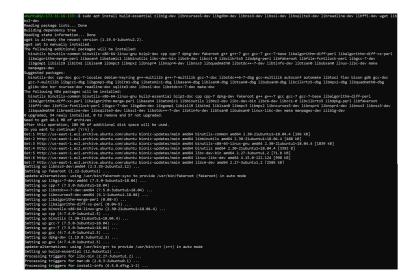
update the packages list and install the packages necessary to build Python source:

sudo apt update

<pre>ubuntWip:p172-31-16-133-4 sudo apt update Hit:1 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB] Get:3 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB] Get:4 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [74.6 kB] Get:5 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe man64 Packages [8570 kB] Get:6 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe InRelease [74.6 kB] Get:6 http://Us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe InRelease [74.6 kB] Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe InRelease [151 kB] Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/undates/main ma64 Packages [151 kB] Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [188 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [24.6 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/nestricted amd64 Packages [14 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/nestricted Translation-en [24.6 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/nestricted Translation-en [598 kB] Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [598 eB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/minitverse Translation-en [598 eB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/minitverse Translation-en [598 eB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min amd64 Packages [16.0 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [476 B] Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [476 B] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/mineres Translation-en [476 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-ba</pre>	
<pre>Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [87. KB] Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe and64 Packages [8570 kB] Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe ranslation-en [4941 kB] Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe ranslation-en [4941 kB] Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe ranslation-en [4941 kB] Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe ranslation-en [4941 kB] Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/undtates/mainticates/ Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/undtates/ Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/undtates/main ranslation-en [368 kB] Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/main ranslation-en [367 kB] Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/main ranslation-en [367 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/restricted amd64 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/restricted ranslation-en [24.6 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/universe amd64 Packages [31.9 kB] Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/universe amd64 Packages [19.6 kB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/universe amd64 Packages [10.8 kB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-undtates/universe ranslation-en [476 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min amd64 Packages [10.8 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.8 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.8 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.8 kB] Get:</pre>	
<pre>Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6, [6] Get:4 http://secast-1.ec2.archive.ubuntu.com/ubuntu bionic/universe and64 Packages [8570 kB] Get:5 http://security.ubuntu.com/ubuntu bionic/security/InRelease [88.7 kB] Get:6 http://security.ubuntu.com/ubuntu bionic/security/main and64 Packages [1399 kB] Get:7 http://security.ubuntu.com/ubuntu bionic/universe and64 Packages [1398 kB] Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse ranslation-en [4941 kB] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [148 kB] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [567 kB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [57 kB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [57 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/nestricted Translation-en [57 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/nestricted Translation-en [54 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse and64 Packages [168 kB] Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse and64 Packages [18.9 kB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse and64 Packages [18.9 kB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [18.9 kB] Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [18.9 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe and64 Packages [18.9 kB] Get</pre>	
<pre>Get:4 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe and64 Packages [8576 kB] Get:5 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [4941 kB] Get:7 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [4941 kB] Get:8 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [4941 kB] Get:8 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [4941 kB] Get:8 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [496 kB] Get:8 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [496 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-undates/main Translation-en [367 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-undates/main Translation-en [367 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-undates/netricted and64 Packages [184 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-undates/universe and64 Packages [184 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe and64 Packages [194 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe and64 Packages [194 kB] Get:19 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe and64 Packages [19. kB] Get:19 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [6980 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [474 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 B] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 B] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 kB] Get:23 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 kB] Get:24 http://us-east-1.ec2.archive.uburtu.com/uburtu bioni</pre>	
<pre>Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-security/main and64 Packages [1399 k8] Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 k8] Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [151 k8] Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 k8] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [175 k8] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main and64 Packages [175 k8] Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [367 k8] Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [46 k8] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [567 k8] Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [564 k8] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [598 k8] Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [598 k8] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [18.9 k8] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 k8] Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [4764 k8] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [4768 k8] Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [4768 k8] Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/minerse amd64 Packages [18.9 k8] Get:22 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [276 k8] Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [24 k8] Get:26 http://security.ubuntu.com/ubuntu bionic-security/restrict</pre>	
<pre>Get:6 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic/universe Translation-en [4941 k8] Get:7 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic/multiverse Translation-en [4941 k8] Get:8 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic/multiverse Translation-en [4984 k8] Get:8 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic/multiverse Translation-en [4984 k8] Get:8 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic/multiverse Translation-en [4984 k8] Get:8 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/msin Translation-en [496 k8] Get:18 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/msin Translation-en [496 k8] Get:18 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted amd64 Packages [48 k8] Get:18 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted Translation-en [494 k8] Get:18 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe amd64 Packages [184 k8] Get:16 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe amd64 Packages [180 k8] Get:16 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse amd64 Packages [180 k8] Get:17 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [496 k8] Get:18 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [496 k8] Get:21 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [496 k8] Get:22 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [496 k8] Get:21 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [496 k8] Get:21 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [24 k8] Get:22 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [24 k8] Get:23 http://us-exst-1.ec2.archive.uburtu.com/uburtu bionic-security/miverse Translation-en [24 k8] Get:24 http://us-exst-1.ec2.archive.uburtu</pre>	
<pre>Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-multiverse and64 Packages [15] kB] Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [15] kB] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [15] kB] Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse and64 Packages [175] kB] Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main and64 Packages [175] kB] Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted and64 Packages [184 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted and64 Packages [184 kB] Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe and64 Packages [184 kB] Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe and64 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe and64 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [354 kB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse and64 Packages [18.0 kB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main and64 Packages [18.0 kB] Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/min Translation-en [4764 B] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/miverse and64 Packages [18.0 kB] Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/miverse and64 Packages [18.0 kB] Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/miverse and64 Packages [18.0 kB] Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [276 kB] Get:25 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [276 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [24 kB] Get:27 http://security.ubuntu.com/ubuntu bionic-se</pre>	
<pre>Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB] Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [175 kB] Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [175 kB] Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main renslation-en [367 kB] Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [184 kB] Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted renslation-en [354 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [186 kB] Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [187 kB] Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [18.0 kB] Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [18.0 kB] Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [18.0 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.0 kB] Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.0 kB] Get:22 http://security.ubuntu.com/ubuntu bionic-security/miverse amd64 Packages [18.0 kB] Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntubionic-security/universe amd64 Packages [18.0 kB] Get:26</pre>	Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
<pre>Get:9 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic/multiverse Translation-en [168 kB] Get:10 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/main am6de Packages [1725 kB] Get:11 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted am6de Packages [184 kB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted Translation-en [24.6 kB] Get:31 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted Translation-en [24.6 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/nestricted Translation-en [24.6 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse Translation-en [588 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse Translation-en [588 kB] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse Translation-en [588 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/min am6de Packages [18.0 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/min am6de Packages [18.0 kB] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/min am6de Packages [18.0 kB] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [4764 B] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/miverse Translation-en [476 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/main Translation-en [27.1 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse am6de Packages [18.7 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse am6de Pa</pre>	Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1399 kB]
<pre>Get:10 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/main am64 Packages [1725 kB] Get:11 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/mstricted am644 Packages [184 kB] Get:12 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted Translation-en [24.6 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/unverse am644 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/unverse am644 Packages [184 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe am644 Packages [181 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe am644 Packages [18.1 kB] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse am644 Packages [18.1 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main am644 Packages [18.1 kB] Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main am644 Packages [18.3 kB] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [4764 B] Get:24 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [24.5 kB] Get:25 http://security.uburtu.com/uburtu bionic-security/restricted Translation-en [24.1 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/restricted Translation-en [24.1 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe am644 Packages [18.9 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [24.4 B] Get:27 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [2964 B] Fetched 21.6 MB in 5s (4413 kB/s) Readin</pre>	Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
<pre>Get:11 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/main Translation-en [267 k8] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted mad64 Packages [184 k8] Get:31 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/nestricted restricted mad64 Packages [1681 k8] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/mulverse amd64 Packages [1861 k8] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/mulverse mad64 Packages [187 k8] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/mulverse Translation-en [598 k8] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/mulverse Translation-en [698 b] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [698 b] Get:36 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 b] Get:26 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/min Translation-en [476 k8] Get:27 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/minerse Translation-en [476 k8] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 k8] Get:22 http://us-east-1.uec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 k8] Get:23 http://security.uburtu.com/uburtu bionic-security/mastricted amd64 Packages [180 s8] Get:24 http://security.uburtu.com/uburtu bionic-security/miverse amd64 Packages [187 s8] Get:25 http://security.uburtu.com/uburtu bionic-security/miverse amd64 Packages [187 s8] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse Translation-en [24 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/miverse Translation-en</pre>	Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
<pre>Get:12 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted mran64 Packages [164 kB] Get:13 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/restricted Translation-en [24 ck kB] Get:14 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe amd64 Packages [1681 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [594 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [596 kB] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [598 kB] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main amd64 Packages [18.0 kB] Get:19 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main ranslation-en [4764 B] Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4768 B] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4768 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [4588 B] Get:22 http://security.uburtu.com/uburtu bionic-security/main Translation-en [27.1 kB] Get:23 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [166 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/rese Translation-en [27.1 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [24.1 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [2964 B] Fetched 21.6 MB in 5s (4413 kB/s) Reading mackage 11sts Done Building dependency tree Reading state information Done</pre>	Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1725 kB]
<pre>Get:13 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updatss/nestricted Translation-en [24.6 k8] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updatss/universe amd64 Packages [168] k8] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updatss/universe amd64 Packages [168] k8] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updatss/universe amd64 Packages [16.8 k8] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updatss/universe amd64 Packages [16.8 k8] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [5980 B] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [10.8 k8] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [4764 B] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [10.3 k8] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [10.3 k8] Get:23 http://us-east-1.uburtu.com/uburtu bionic-security/main Translation-en [276 k8] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [10.9 k8] Get:25 http://security.uburtu.com/uburtu bionic-security/miverse amd64 Packages [10.9 k8] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse amd64 Packages [12.9 k8] Get:26 http://security.uburtu.com/uburtu bionic-security/multiverse amd64 Packages [12.9 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [24.8 k8] Fetched 21.6 H8 in 5s (4413 k8/s) Reading package list Done</pre>	Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [367 kB]
<pre>Get:14 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe maßed Packages [1681 k8] Get:15 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [354 k8] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe amd64 Packages [31.9 k8] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse amd64 Packages [18.0 k8] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse amd64 Packages [18.0 k8] Get:19 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main amd64 Packages [18.0 k8] Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18.0 k8] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18.0 k8] Get:24 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [476 k8] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted Translation-en [276 k8] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted Translation-en [27.1 k8] Get:26 http://security.uburtu.com/uburtu bionic-security/restricted Translation-en [21.1 k8] Get:26 http://security.uburtu.com/uburtu bionic-security/nestricted Translation-en [24.1 k8] Get:27 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [1879 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [296 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/main Translation-en [296 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse amd64 Packages [12.9 k8] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 k8 in 5s (4413 k8/s) Reading ackage lists Done Building dependency tree Reading state information Done</pre>	Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [184 kB]
<pre>Get:15 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [554 kB] Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/universe Translation-en [6980 B] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main am664 Packages [10.0 kB] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [6980 B] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe arG64 Packages [10.3 kB] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe arG64 Packages [10.3 kB] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe arG64 Packages [10.5 kB] Get:23 http://security.uburtu.com/uburtu bionic-security/main Translation-en [276 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/mesrited am664 Packages [10.6 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/mesra mG64 Packages [10.7 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse am64 Packages [10.7 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse am64 Packages [10.7 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/universe am64 Packages [12.9 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse am64 Packages [12.9 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 MB in 5s (4413 kB/s) Reading package lists Done</pre>	Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [24.6 kB]
<pre>Get:16 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse amG46 Packages [31.9 kB] Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse Translation-en [6980 B] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main ramG46 Packages [10.0 kB] Get:28 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/min Translation-en [4764 B] Get:23 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [4588 B] Get:23 http://security.uburtu.com/uburtu bionic-security/min Translation-en [276 kB] Get:23 http://security.uburtu.com/uburtu bionic-security/miverse amG4 Packages [166 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/miverse amG4 Packages [167 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/mitiverse amG4 Packages [167 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse amG4 Packages [127 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 kB in 5s (4413 kB/s) Reading ackage lists Done Building dependency tree Reading tack information Done</pre>	
<pre>Get:17 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-updates/multiverse Translation-en [6980 B] Get:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main amd64 Packages [18.0 kB] Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe ard64 Packages [18.0 kB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe ard64 Packages [18.0 kB] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe ard64 Packages [18.0 kB] Get:22 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe ard64 Packages [18.0 kB] Get:22 http://security.uburtu.com/uburtu bionic-security/restricted and64 Packages [18.6 kB] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted and64 Packages [18.7 kB] Get:25 http://security.uburtu.com/uburtu bionic-security/universe and64 Packages [18.7 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [24 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/unitverse Translation-en [2964 B] Fetched 21.6 PB in 75 (4413 kB/s) Reading apendancy tree Building dependency tree Reading state information Done</pre>	Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [354 kB]
<pre>Set:18 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main amd64 Packages [18. kB] Get:29 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main man64 Packages [18. kB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18. 3 kB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18. 3 kB] Get:21 http://security.uburtu.com/uburtu bionic-security/nestricted amd64 Packages [166 kB] Get:23 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [166 kB] Get:24 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [167 kB] Get:25 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [167 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [12.9 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/multiverse amd64 Packages [12.9 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 NB in 5s (A413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [31.9 kB]
<pre>Get:19 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/main Translation-en [4764 B] Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18.3 kB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe amd64 Packages [18.3 kB] Get:22 http://security.uburtu.com/uburtu bionic-security/main Translation-en [276 kB] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [166 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [1679 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [1879 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe amd64 Packages [1279 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [296 kB] Fetched 21.6 MB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6980 B]
<pre>Get:20 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe armd64 Packages [18'3 KB] Get:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translation-en [4588 B] Get:23 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [16's KB] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [16's KB] Get:25 http://security.uburtu.com/uburtu bionic-security/restricted amd64 Packages [16's KB] Get:26 http://security.uburtu.com/uburtu bionic-security/niverse amd64 Packages [16's KB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [22.1 kB] Get:27 http://security.uburtu.com/uburtu bionic-security/universe Translation-en [241 kB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 KB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [10.0 kB]
<pre>Set:21 http://us-east-1.ec2.archive.uburtu.com/uburtu bionic-backports/universe Translationen [4588 8] Get:22 http://security.uburtu.com/uburtu bionic-security/main Translationen [276 kB] Get:23 http://security.uburtu.com/uburtu bionic-security/restricted md64 Packages [166 kB] Get:24 http://security.uburtu.com/uburtu bionic-security/restricted Translationen [27.1 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/restricted Translationen [27.4 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe and64 Packages [1679 kB] Get:26 http://security.uburtu.com/uburtu bionic-security/universe and64 Packages [127, kB] Get:27 http://security.uburtu.com/uburtu bionic-security/multiverse and64 Packages [12, skB] Get:28 http://security.uburtu.com/uburtu bionic-security/multiverse Translationen [2964 B] Fetched 21.6 kB in 5s (4413 kB/s) Reading nackage lists Done Building dependency tree Reading state information Done</pre>	Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]
<pre>Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [276 kB] Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [166 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe mand64 Packages [1679 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [221. kB] Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [241. kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [241. kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [241. kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [241. kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 NB in 5s (4433 kB/s) Reading package listsDone Building dependency tree Reading state information Done</pre>	Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.3 kB]
<pre>Set:23 http://security.ubuntu.com/ubuntu bionic-security/restricted am64 Packages [166 kB] Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [22.1 kB] Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe am64 Packages [1279 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [241 kB] Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse am64 Packages [12.9 kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 kB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B]
<pre>Get:24 http://security.ubuntu.com/ubuntu bionic-security/nestricted Translation=n [22.1 kB] Get:25 http://security.ubuntu.com/ubuntu bionic-security/niverse amd64 Packages [1879 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 HB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	
<pre>Set:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1079 kB] Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [241 kB] Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB] Fetched 21.6 kB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [166 kB]
<pre>Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation=en [241 kB] Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse and64 Packages [12.9 kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation=en [2964 B] Fetched 21.6 VB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done</pre>	Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [22.1 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB] Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 MB in 55 (4431 kB/s) Reading package lists Done Building dependency tree Reading state information Done	Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1079 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B] Fetched 21.6 MB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done	Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [241 kB]
Fetched 21.6 MB in 5s (4413 kB/s) Reading package lists Done Building dependency tree Reading state information Done	Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [12.9 kB]
Reading package listsDone Building dependency tree Reading state informationDone	Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2964 B]
Building dependency tree Reading state information Done	Fetched 21.6 MB in 5s (4413 kB/s)
Reading state information Done	Reading package lists Done
	Building dependency tree
57 packages can be upgraded. Run 'apt listupgradable' to see them.	Reading state information Done
	57 packages can be upgraded. Run 'apt listupgradable' to see them.

• Install needed dependency:

sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3dev libssl-dev libsqlite3-dev libreadline-dev libffi-dev wget libbz2-dev``



Download and extract the downloaded package

- · Download python from below Link
 - https://www.python.org/downloads/
 - https://www.python.org/ftp/python/3.7.0/Python-3.7.0.tgz

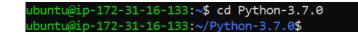
Download and untar:

```
wget https://www.python.org/ftp/python/3.7.0/Python-3.7.0.tgz
tar xzf Python-3.7.0.tgz
```



Next, navigate to the Python source directory and run the configure script which will perform a number of checks to make sure all of the dependencies on your system are present:

```
cd Python-3.7.0
```



• Build & compile:

```
./configure --enable-optimizations
```

• Install the Python binaries by running the following command:



make altinstall

Note: Do not use the standard make install as it will overwrite the default system python3 binary.

Verify it by typing:

python3.7 --version

Create Python virtual environment & Activate it

Create Python virtual environment & Activate it:

```
python3.7 -m venv venv
source venv/bin/activate
python --version
```

Upgrade pip version

Upgrade pip version with 20.0 or above:

pip install pip --upgrade

untu@ip-172-33-16-133:+/Pvthon-3.7.0\$ sudo make altinstall	
<pre>c -pthread -c -kho-unused-result -Wsign-compare -ONDEBUG -g -fwrapy -03 -Wall -std#c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia</pre>	lizer
Werror=implicit-function-declaration -II./Include -DPv BUILD CORE -o Programs/python.o./Programs/python.c	
c -pthread -c -Who-unused-result -Wsign-compare -DNDEBUG -g -fwrapy -03 -Wall -std=c99 -Wextra -Who-unused-result -Who-unused-parameter -Who-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/acceler.c	
c -pthread -c -Nho-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Nall -std=c99 -Wextra -Who-unused-result -Who-unused-parameter -Who-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/grammar1.o Parser/grammar1.c	
c -pthread -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/listnode.c	
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -03 -Nall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/node.c	
c pthread -c +ho-unused-result +ksign-compare -DNDEBUG g -fmrapv -03 -Wall - study-09 -Wextra +kho-unused-result -kho-unused-parameter -kho-missing-field-initia Werromirabilis-function-declaration -l. i./i/Locude -DPV BUILD COEP - D Parse/carser.c Parse/carser.c	Lizer
werrormampicit-runction-declaration -11./include -νν_pullU_Cuk -0 varser/parser.0 rarser/parser.0 c opthread -c -Nho-unused-result-Noisin-compane -DMDEMS or finanzy -finanzy -03 -Nall -stdrkc99 -Westra -Nho-unused-result-Nho-unused-parameter -Nho-missing-field-initia	
C "ptriesd °C *#mo°unused"ebult *#sign"compare 'umudou g *twiapy 'uo 'Hail -starty' *#extra *#mo°unused*tebult *#mo°unused*parameter '#mo*missing*tield*initia Merrom:#mplicit.function=declaration -II./Include -DPS 001LD CORE -o Parsen/bitset.c Parsen/bitset.c	11zer
werson-subjilitersonstionswerserenden -i signiserende o <u>r son son son son son son son son son son</u>	14000
c pureau Commonitation declaration - Collaria - Contractor - Marine - Scorest - Marine - Mar	TTTE.
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapy -03 -Nall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPv BUILD CORE -o Parser/firstsets.o Parser/firstsets.c	
c -pthread -c -Nho-unused-result -Wsign-compare -DNDEBUG -g -fwrapy -03 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy BUILD CORE -o Parser/grammar.o Parser/grammar.c	
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -03 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/pgen.c	
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Nall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/myreadline.o Parser/myreadline.c	
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Nall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Parser/parsetok.c Parser/parsetok.c	
c -pthread -c -Nno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -03 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initia	lizer
Werror=implicit=function-declaration -II./Include - OPy_BUID_CORE -o Parser/tokenizer.o Parser/tokenizer.c c.sthread - «Nor-unused-result +Wein-compare DOBBWG = franzo -O3 +Weil 1 - std=c9 - Werra +Nor-unused-result +Nor-unused-parameter -Nor-missing-field-initia	
c -pthread -c -Nno-unused-result -Nsign-compare -DNDEBUG -g -firspy -03 -Nall -stdreg9 -Naxtra -Nno-unused-result -Nno-unused-parameter -Nno-missing-field-initia Merrorispiisif-function-declaration - 11./T.Neude - DPV BUILD CORE -0 Objects/abstract.c Objects/abstract.c	Lizer
werrorampicatriuntioniaecianation ". "./include "urgoulu_cue o uogectaaostrast.o uogectaaostrast.o o phtnead - o Nho-unused-result-Nsign-compane - DNDEBUG of -fwrapy - 03 -Nall - stdres9 -Nextra-a Nho-unused-result-Nho-unused-parameter -Nho-missing-field-initia	
c primar c anonications i assistemente ano so se ani accessante ano desta ano ante accestate ano ante a solo secondaria ante ante ante a solo secondaria ante ante ante ante ante ante ante ant	11260
meron amplatar fonction outprover and the strand of the st	lizon
werrowinglicitfunction-declaration -1././include -DPV BUID CORE -0 Objects/boolobject.c	
<pre>c -pthread -c -hno-unused-result -Wsign-compare -DDDEBUG -g -fwrapv -03 -Hall -stdc299 -Waxta - who-unused-result -Wno-unused-parameter -Wno-missing-field-initia</pre>	lizer
Werror*implicit-function-declaration -II./Include -DPy_BUILD_CORE -o Objects/bytes_methods.c Objects/bytes_methods.c	
/usr/local/lib/python3.7/lib-dynload/ sysconfigdata m linux x86 64-linux-gnu.py	
-r /usr/local/lib/python3.7/lib-dynload/_pycache	
eating directory /usr/local/share/man/man1	
sr/bin/install -c -m 644 ./Nisc/python.man \	
/usr/local/share/man/man1/python3.7.1	
test "xupgrade" != "xno" ; then \	
case upgrade in \	
upgrade) ensurepip="altinstallupgrade" ;; \	
install ") ensurepip="altinstall" ;; \	
esac; \	
./python -E -m ensurepip \	
<pre>\$ensurepiproot=/ ; \</pre>	
e directory '/home/buhrtu/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions an of that directory. If executing pip with sudo, you may want sudo's -H flag.	p-oun
of that surjectory, if executing pip with suce, you have want suce 5 -n fing. e directory '/home/uburu/.cache/pip' of its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner	-
e directory /noneydomco/cache/pip or its parent directory is not owned by the current oser and caching wheels has been disabled, theck the permissions and owner directory. If executing the bit sold, you may want sudo's + flag.	
usrettory, ar executing pap matrixed with source of the so	
leting sturols	
lecting pip	
stalling collected packages: setuptools, pip	
ccessfully installed pip-10.0.1 setuptools-39.0.1	

ubuntu@ip-172-31-16-133:~\$ python3.7 --version Python 3.7.0

ubuntu@ip-172-31-16-133:~\$ python3.7 -m venv venv ubuntu@ip-172-31-16-133:~\$ source venv/bin/activate (venv) ubuntu@ip-172-31-16-133:~\$ python --version Python 3.7.0

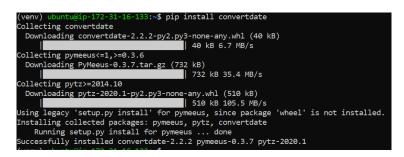


Install dependency for fbprophet package (Ubuntu 18.04)

• pystan dependency:

• convertdate dependency:

pip	install	convertdate
-----	---------	-------------



• fbprophet dependency:

pip install fbprophet



• Check pip list:

pip list

Install Other Packages

Install the required packages:

31-16-133: ~\$ pip list
Version
2020.6.20
0.9.5
2.2.2
0.10.0
0.29.21
3.7.7.1
0.7.1
0.10.3
1.3.0
0.2.1
0.0.9
3.3.2
1.19.2
1.1.3
8.0.1
20.2.4
0.3.7
2.4.7
2.19.1.1
2.8.1
2020.1
39.0.1
1.2
1.15.0
4.51.0

```
cd fire-x.y.x/dist/fire
pip install -r requirements.txt
```

requirements.txt file is available in the installation directory of fire insights:

```
fire-x.y.x/dist/fire/requirements.txt
```

Delete a venv

To delete a virtual environment, follow below steps:

```
source venv/bin/activate
pip freeze > requirements.txt
pip uninstall -r requirements.txt -y
deactivate
rm -r venv/
```

2.1.10 Running Diagnostics

Linux

Fire Insights needs jdk 1.8 to be available

• java -version

java version "1.8.0_101"

Mac OS

Fire Insights needs jdk 1.8 to be available

• java -version

java version "1.8.0_101"

Windows

Fire Insights needs jdk 1.8 to be available

• java -version

java version "1.8.0_101"

CHAPTER 3

Configuration

3.1 Configuration

3.1.1 Database Setup

Fire stores metadata in a Relational Database. These include:

- Applications
- Dataset Definitions
- Workflows
- Users
- Groups
- Roles
- Dashboards

Below are the details for setting up a database for Fire:

H2 Database

Fire can be setup up to easily run with H2 database. Fire runs H2 in embedded mode. The H2 database is used for storing the metadata of the DataSets, Workflows, Dashboards, Users, Groups, Roles etc.

More details of the H2 database can be found here : http://www.h2database.com/html/main.html

If you are want to run multiple instances of Fire for high availability, configure Fire to run with MySQL.

Creating/Upgrading the H2 database

Execute the following steps on your CLI:

• Mac/Linux:

```
cd <install_dir>/fire-x.y.z
./create-h2-db.sh
```

• Windows:

```
cd <install_dir>\fire-x.y.z
.\create-h2-db.bat
```

If you would like to use different values for the db, username, password, update them in <install_dir>/ fire-1.4.0/conf/db.properties:

```
spring.datasource.url = jdbc:h2:file:~/firedb
spring.datasource.username = fire
spring.datasource.password = fire
spring.datasource.driverClassName = org.h2.Driver
```

Note: firedb is created in the users home directory and the name is firedb.mv.db

Recreating H2 database

If you need to recreate the H2 database, follow the steps below to create a new empty H2 DB:

```
Stop the running Fire server
Move the existing firedb files to another temp location on your disk
Recreate the H2 DB using the steps in the above section for creating a brand new_
→empty H2 DB for Fire
```

MySQL Database

Fire can easily be setup up to run with MySQL

More details of the MySQL database can be found here : https://www.mysql.com/

Install MySQL

- Install MySQL on a machine.
- It might be easier to install it on the same machine you are installing Fire on.

Create the DB for Fire in MySQL

- Create the database for Fire in MySQL
- Let us call it firedb:

create database firedb;

Create the User for Fire in MySQL and grant it Permissions

Create the User for Fire in MySQL:

```
CREATE user 'fire'@'%' IDENTIFIED BY 'fire';
GRANT ALL PRIVILEGES ON firedb.* TO 'fire'@'%' WITH GRANT OPTION;
```

- In CREATE user, the user we are creating is fire who is allowed to access the database from anywhere % and his password is fire.
- Next, this user has been granted all permissions. This, of course can be further restricted based on your use case.

Configure Fire to connect to MySQL

• Copy db.properties.mysql file into the conf directory as db.properties:

```
cd fire-x.y.z
cp conf.orig/db.properties.mysql conf/db.properties
```

• Update the following fields in conf/db.properties based on the values you used in creating the DB for fire. The below assumes that the database name you created for Fire is firedb. It also assumes that MySQL has been installed on the same machine as Fire:

```
# Connection url for the database "firedb"
spring.datasource.url=jdbc:mysql://localhost:3306/firedb
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.jpa.database=MYSQL
# Username and password
spring.datasource.username=fire
spring.datasource.password=fire
```

Install the MySQL Connector Jar file

- Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html
- Extract the JDBC driver JAR file from the downloaded file. For example:

```
tar zxvf mysql-connector-java-8.0.11.tar.gz
```

- just copy the path location for `JDBC driver JAR file
- copy the mysql JDBC driver JAR file to the fire-server-lib directory of fire-x.y.z:

```
cd fire-x.y.z
cp /pathlocation of jdbc jar file/mysql-connector-java.jar fire-server-lib
```

Create the Tables for Fire in MySQL

• Create the tables for Fire in MySQL by executing the create-mysql-db.sh script:

```
cd fire-x.y.z
./create-mysql-db.sh
```

Troubleshooting

MySQL has a problem where one of the default users in the user table is '' @ localhost, which winds up denying all localhost users later in the table. If you are accessing mysql from localhost, assuming Fire and MySQL have been installed on the same machine, then you need to delete this entry in mysql.user table:

Here is a link on stackoverflow that talks about this:

http://stackoverflow.com/questions/1412339/cannot-log-in-with-created-user-in-mysql

Microsoft SQL Server Database

Fire can easily be setup up to run with Microsoft SQL Server.

More details of the Microsoft SQL Server database can be found here : https://www.microsoft.com/en-us/sql-server/ default.aspx

Install Microsoft SQL Server

- Install Microsoft SQL Server on a machine.
- It might be easier to install it on the same machine you are installing Fire on.

Create the DB for Fire in Microsoft SQL Server

- Create the database for Fire in Microsoft SQL Server
- Let us call it firedb:

CREATE DATABASE firedb;

Create the User for Fire in Microsoft SQL Server and grant it Permissions

Create the User for Fire in Microsoft SQL Server and give it Permissions.

Configure Fire to connect to Microsoft SQL Server

• Copy db.properties.sqlserver file into the conf directory as db.properties:

```
cd fire-x.y.z
cp conf.orig/db.properties.sqlserver conf/db.properties
```

• Update the following fields in conf/db.properties based on the values you used in creating the DB for fire. The below assumes that the database name you created for Fire is firedb. It also assumes that Microsoft SQL Server has been installed on the same machine as Fire:

```
# Connection url for the database "firedb"
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=firedb
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
spring.jpa.database=SQLSERVER
# Username and password
spring.datasource.username=fire
spring.datasource.password=fire
spring.jpa.hibernate.dialect=org.hibernate.dialect.SQLServer2008Dialect
```

Install the Microsoft SQL Server Connector Jar file

- Download the Microsoft SQL Server JDBC driver from https://www.microsoft.com/en-us/download/details. aspx?id=11774
- Untar the file sqljdbc_6.0.8112.200_enu.tar.gz
- You will get JDBC jar file on untaring sqljdbc42.jar
- Copy the Microsoft SQL Server JDBC driver JAR file to the fire-server-lib directory of fire-x.y.z

Create the Tables for Fire in Microsoft SQL Server

- Tables in Microsoft SQL Server can be created by using the DDL script : db/sqlserver/fire-schema. sqlserver.sql
- They can also be created by executing the create-sqlserver-db.sh script:

```
cd fire-x.y.z
./create-sqlserver-db.sh
```

Aurora MySQL Database

Fire can easily be setup up to run with Aurora MySQL

More details of the Aurora MySQL database can be found here : https://aws.amazon.com/rds/aurora/

Create Aurora MySQL database on AWS

- Login to AWS.
- Create Aurora MySQL Database which is accessible from machine where Fire is running.

Create the DB for Fire in Aurora MySQL

- Create the database for Fire in Aurora MySQL
- Let us call it firedb:

```
create database firedb;
```

Create the User for Fire in Aurora MySQL and grant it Permissions

Create the User for Fire in MySQL:

```
CREATE user 'fire'@'%' IDENTIFIED BY 'fire';
GRANT ALL PRIVILEGES ON firedb.* TO 'fire'@'%' WITH GRANT OPTION;
```

- In CREATE user, the user we are creating is fire who is allowed to access the database from anywhere % and his password is fire.
- Next, this user has been granted all permissions. This, of course can be further restricted based on your use case.

Configure Fire to connect to Aurora MySQL

• Copy db.properties.mysql file into the conf directory as db.properties:

```
cd fire-x.y.z
cp conf.orig/db.properties.mysql conf/db.properties
```

• Update the following fields in conf/db.properties based on the values you used in creating the DB for fire. The below assumes that the database name you created for Fire is firedb. It also assumes that MySQL has been installed on the same machine as Fire:

```
# Connection url for the database "firedb"
spring.datasource.url=jdbc:mysql://Endpoint:3306/firedb
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.jpa.database=MYSQL
# Username and password
spring.datasource.username=fire
spring.datasource.password=fire
```

Install the MySQL Connector Jar file

- Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html
- Extract the JDBC driver JAR file from the downloaded file. For example:

tar zxvf mysql-connector-java-8.0.11.tar.gz

- just copy the path location for `JDBC driver JAR file
- copy the mysql JDBC driver JAR file to the fire-server-lib directory of fire-x.y.z:

```
cd fire-x.y.z
cp /pathlocation_of_jdbc_jar_file/mysql-connector-java.jar fire-server-lib
```

Create the Tables for Fire in Aurora

• Create the tables for Fire in MySQL by executing the create-mysql-db.sh script:

```
cd fire-x.y.z
./create-mysql-db.sh
```

Troubleshooting

MySQL has a problem where one of the default users in the user table is '' @ localhost, which winds up denying all localhost users later in the table. If you are accessing mysql from localhost, assuming Fire and MySQL have been installed on the same machine, then you need to delete this entry in mysql.user table:

Here is a link on stackoverflow that talks about this:

http://stackoverflow.com/questions/1412339/cannot-log-in-with-created-user-in-mysql

3.1.2 Connecting to Apache Spark Cluster

Overview

Fire can be configured to submit the spark jobs to run on an Apache Spark Cluster.

- Install Fire on an edge node of your Apache Spark Cluster.
 - The edge node has the hadoop/hive/spark configuration files set up.
 - Make sure that you are already able to run your spark jobs from this node using spark-submit.
- Update the below configurations under the menu, "Administration/Configuration"

Note: In order for Fire to connect to the Apache Spark Cluster, it needs to be installed as a user which can impersonate other users. More details are below in the page. For the rest of the documentation on this page, we assume that it has been installed as the user sparkflows.

Fire User Setup

The user with which Fire is running has to be a proxy user in HDFS. That way it can impersonate the logged in user.

Below are the steps for setting the Fire user to be a proxy user on HDFS.

Update core-site.xml of Hadoop to allow Fire user to impersonate

https://www.cloudera.com/documentation/enterprise/5-8-x/topics/admin_hdfs_proxy_users.html

- In your core-site.xml file for Hadoop, allow sparkflows user to impersonate other users. Without impersonation enabled for this user, your Sparkflows application users trying to run jobs against a hadoop cluster would not be able to do so.
- Also, allow the appropriate groups that the sparkflows users will be able to impersonate belong to.
- In the example below, user sparkflows is allowed to impersonate users from hosts host1 and host2. The users being impersonated belong to the groups hive, hfs, hadoop, spark. Your permissions are likely going to be different and more restrictive.

Below is an example:

```
<property>
        <name>hadoop.proxyuser.sparkflows.hosts</name>
        <value>host1,host2</value>
        </property>
        <name>hadoop.proxyuser.sparkflows.groups</name>
        <value>hive,hfs,hadoop,spark</value>
        </property>
```

Cloudera Manager

If you are using Cloudera Manager, you can set the above settings for impersonation in HDFS/Configuration.

cloudera ⁻ man	ager Clusters		harts - Adminis	tration +	Search (Hotke	y; /) Support - admin -
HDFS (Clus	ter 1)					June 16, 2017, 2:37 AM UTC
Status Instances	Configuration 🔀	Commands Charts Library Cache St	tatistics Audits	NameNode Web UI ₽ Quick Links ▼		Actions -
Configuration					Switch to the	classic layout Role Groups
Filters		core				
♥ STATUS						Show All Descriptions
C Error	0	Shared Hadoop Group Name	HDFS (Service-Wide)		0
A Warning	0		hadoop			
Edited	1					
Non-default	1	Cluster-wide Advanced Configuration	HDFS (Service-Wide	a C		View as XML
Has Overrides	0	Snippet (Safety Valve) for core-site.xml			Î	VIEW as AML
✓ SCOPE			Name	hadoop.proxyuser.sparkflows.hosts		
HDFS (Service-Wide)	4		Value	*		
Balancer	0		Description	Description		
DataNode	0			Final		
Gateway	0				1	
HttpFS	0		Name	hadoop.proxyuser.sparkflows.groups		
JournalNode	0		Value	*		
NFS Gateway	0		Description	Description		
NameNode	0			Final		
SecondaryNameNode	0					
Failover Controller	0		+			

Ambari

If you are using Ambari, you can set the above settings for impersonation in ${\tt HDFS/Configuration}$ under Custom core-site

· HDFS	Summary Heatmaps	Configs Quick Links -		Service Actions -
YARNMapReduce2	Group Default (8)	Manage Config Groups	p.proxyuser	₿ ▼
 Tez Hive Pig 	about a day age	ternal V1 admin about a day ago HDP-2.6		
ZooKeeperAmbari InfraAmbari Metrics		uthored on Thu, Jun 22, 2017 00:19		Discard Save
 Ranger SmartSense Spark2 	Custom core-site			
 Spark2 Zeppelin Notebook 	hadoop.proxyuser.HTTP. groups	users	₽ •	0
Cerberos Actions ▼	hadoop.proxyuser.HTTP. hosts	*	●	5 e C
	hadoop.proxyuser.livy. groups	•		•
	hadoop.proxyuser.livy. hosts	•		0
	hadoop.proxyuser.yarn. groups	•		•

Infer Hadoop Configs

Infer Hadoop Configs button under Administration/Configuration automatically infers some of the configurations of the cluster from the hadoop config files on the edge node to help with the process. Use it to get the initial set of configurations.

🕑 Fire I	nsights 셉 DATA BROWSERS ~	📰 DATASETS 🛛 🛔 WORKFLOW 👻	e workflow executions 🗸 🖵 dashboard 🏼 🖄 admi		
Cont	figurations				
SAVE CO	ONFIGURATIONS INFER HADOOP O				
ID	NAME	TITLE	VALUE1		
1	app.runOnCluster	Run on Spark Cluster	false		
3	app.impersonateUsers	Impersonate Users	true		
4	app.postMessageURL	Fire ui postback URL	http://localhost:8080/messageFromSparkJob		
5	app.sparkSubmitJar	Spark Submit Jar File	/user/centos/fire-1.4.2/fire-lib/fire-core-1.4.2-jar-with-depende		
6	app.nodesDir	Nodes Directory	nodes		

Fire Configurations for connecting to an Apache Spark Cluster

Below are the configuration details for connecting Fire to an Apache Spark Cluster.

Pa-	Value	Description
rame-		
ter		
app.runO	nGluester	Indicate to run on the spark cluster. By default it is set to false
app.post1	Alesspagetor Albert 1998	Indicate the URL on fire server which receives messages from the spark jobs
	messageFromSparkJob	running on the cluster. Set localhost to the machine name on which Fire is
		running. Replace 8080 with the port number on which Fire is running.
app.spark	S/ulser/t/en/tos/fire-	fire-lib directory of the Sparkflows install contains the fire core jar used in
	2.1.0/fire-lib/fire-	submitting the workflows to the Spark cluster. Set it correctly to be the abso-
	core-2.1.0-jar-with-	lute path of the fire core jar.
	dependencies.jar	
hdfs.nam	enholdfe:1//Racalhost:8020	Update the hdfs namenode URI. Set localhost to the machine on which the
		namenode is running.
hdfs.nam	eniderURI	Set it to file:// when the files are on the local filesystem. This can be the case
		when HDFS is not there.
hdfs.nam	emoalpitfsRN/	Set it to maprfs:/// for mapr.
hive.JDB	Cjd DB hURL://localhost:	Update the hive JDBC DB URL if you would be accessing HIVE from Spark-
	10000	flows. This is the URL of the HiveServer 2 server.
spark.sql	- HIVEContext	Set it to either HIVEContext or SQLContext based on whether you want
context		to use HIVEContext or SQLContext in your job. Use HIVEContext if you
		would be accessing the HIVE tables.
spark.ma	stæarn	Set it to yarn for connecting to a spark cluster running YARN
spark.ma	st ep ark://spark_master_hos	tn Set eitptorthe spark master URL when connecting to a spark cluster running in
		standalone mode. Port is normally 7077.
spark.spa	rkspark-submit	Spark Submit command for submitting the Spark jobs to the cluster. It can be
submit		spark2-submit for Spark2 CDH clusters. Make sure to provide the full path
		or spark-submit should be in the path.

Create New Users in Fire

Fire allows creating multiple users. Create the users in Fire under Administration/Users who would be building and running workflows.

These users have to exist on HDFS. So ensure that these users Home Directory are created on HDFS $% \left(\mathcal{A}^{\prime}\right) =\left(\mathcal{A}^{\prime}\right) \left(\mathcal{A}^{\prime}\right$

Also create the home directory for the users on HDFS. The example code below creates the home directory for the user xyz onto HDFS. It also changes the permission of the directory.

- su hdfs
- hadoop fs -mkdir /user/xyz
- hadoop fs -chown xyz:hadoop /user/xyz

Setting up PySpark

If running with PySpark the following might need to be added to point PYSPARK to the right version of python on the cluster machines. Below is an example where python is at /home/ec2-user/venv/bin/python

It is also important that all the users are able to execute the python executable.

spark-env.sh:

```
export PYSPARK_PYTHON=/home/ec2-user/venv/bin/python
export PYSPARK_DRIVER_PYTHON=/home/ec2-user/venv/bin/python
```

spark-defaults.conf:

```
spark.yarn.appMasterEnv.PYSPARK_PYTHON=/home/ec2-user/venv/bin/python
spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON=/home/ec2-user/venv/bin/python
```

3.1.3 Customizing Fire Installation

Below are the details of Configuring Fire for various requirements:

Configuring Max Upload File Size

Fire allows users to upload files into HDFS through their Browser.

The settings which controls it is in conf/application.properties:

```
# max file size
multipart.maxFileSize: 10Mb
multipart.maxRequestSize: 10Mb
```

Increasing Memory of Fire Server

By default, when the Fire web server is started with run-fire-server.sh, it is given 1.5 GB of memory.

Below is from run-fire-server.sh:

```
nohup ${JAVA} -server -Djava.ext.dirs=./user-lib/ -Xmx1548m -Xms1356m -

→XX:+CMSClassUnloadingEnabled -XX:PermSize=512m -XX:MaxPermSize=512m -jar ./app/fire-

→ui-1.3.0.war --spring.config.name=application,db,log4j --spring.config.

→location=file:./conf/ &
```

- In order to increase the amount of memory for the Fire web server, increase the value of -Xmx based on the amount of memory available on your server.
- For example, you could raise it to 5 GB or 10 GB or more up to 25 GB.
 - -Xmx5g
 - -Xmx10g
 - -Xmx25g
- The increased memory size, if available, allows Fire to handle more requests and return results faster. Of course, when connected to an Apache Spark cluster, the full jobs are submitted to the Spark cluster through spark-submit, allowing it to be very scalable and not dependent on the Fire web server.
- The interactive execution of the workflows in the workflow editor, is run within Fire on a small subset of the data. These interactive executions would benefit from increased memory.

3.1.4 Configuring HTTPS for Fire Server

You can choose to run the Fire Server either on http or https connection.

The ports for http and https are configured in the file conf/application.properties:

```
http.port=8080
https.port=8443
```

HTTP

http://hostname:8080/login

HTTPS

https://hostname:8443/login

keystore.jks

Fire Server comes with a pre-configured keystore in the conf folder of the install.

- · conf/keystore.jks
- · conf/keystore.properties : Stores the keystore password

Generating New Keystore

You can use the following command for generating a new keystore:

keytool -genkeypair -alias sparkflows -keyalg RSA -validity 365 -keystore keystore.jks

You will be prompted with the following questions and enter something similar to the SAMPLE answers:

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
 [Unknown]: John Smith
What is the name of your organizational unit?
 [Unknown]: BigData
What is the name of your organization?
 [Unknown]: MyOrg
What is the name of your City or Locality?
 [Unknown]: San Francisco
What is the name of your State or Province?
 [Unknown]: California
What is the two-letter country code for this unit?
 [Unknown]: CA
Is CN=John Smith, OU=BigData, O=MyOrg, L=San Francisco, ST=California, C=CA correct?
  [no]: yes
Enter key password for <sparkflows>
        (RETURN if same as keystore password): Press the return key or Type and note,
\rightarrow down the password
```

Copy the keystore into the Fire installation directory

- Copy the generated keystore.jks file into the conf folder of your installation.
- Update keystore.properties with the new password.

Note: When the keystore is updated, the password also has to be updated in case it changes.

The Fire web server would also have to be restarted for the changes to take effect.

Use keytool commands

Listing entries in Keystore

List entries in keystore:

keytool -list -keystore keystore.jks

Importing a Certificate to an existing Keystore

Importing a Certificate to an existing Keystore:

3.1.5 HTTPS : Importing Self-Signed Certificates

Fire Insights comes with a self-signed certificate. It is contained in conf/keystore.jks.

When using the self-signed certificate, the Browser will complain as it has not been issued by a Certificate Authority.

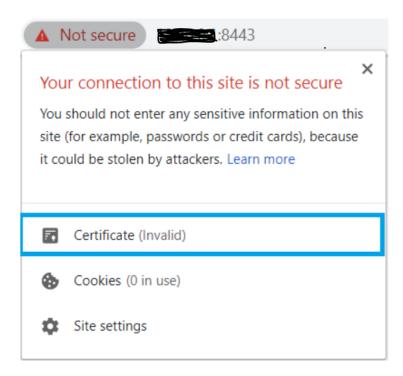
This warning message can be supressed by importing the self-signed certificate into the Browser inside Trusted Root Certification Authorities.

Below are the steps for importing self-signed certificate into your Browser.

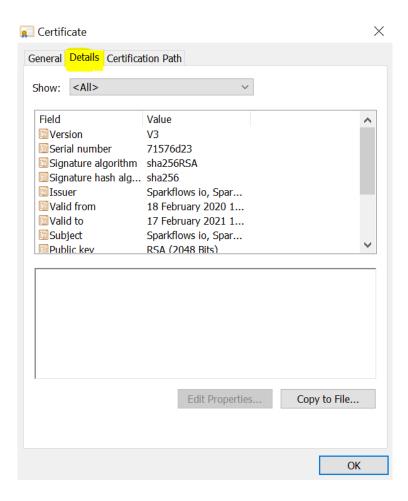
Export the certificate to your machine

- Got to URL for the HTTPS port.
 - https://privateip:8443/login
- Click on Not secure option.
- Click on Certificate.
- View Certificate.
- Click on Details option to see detail information of certificate.
- Click on copy to file option to copy certificate to local machine.
- Select below option and press Next.
- Select the Name & file location of certificate.

C	A No	t secure		8443/login	
					A
					Your connection is not private
					Attackers might be trying to steal your information from 137.117.83.79 (for example, passwords, messages, or credit cards). <u>Learn more</u>
					NET:::RR_CERT_AUTHORITY_INVALID
					Help improve Chrome security by sending <u>UBLs of some pages you visit. limited system</u> information.and some page content to Google. <u>Privacy</u> policy
					Advanced Back to safety
	A	No	t sec	ure	8443/login



TI	Details Certif	te Information rtificate is not trusted. To enable trust, ficate in the Trusted Root Certification
_		Sparkflows io
	-	Sparkflows io 18-02-2020 to 17-02-2021
		Issuer Statement



属 Certifi	cate								\times
General	Details	Certifica	tion Pat	h					
Chave	<all></all>								
Show:	<aii></aii>								
Sign	al numbe nature alg nature ha er d from d to ject	er gorithm Ish alg	sha256 Sparkfi 18 Feb 17 Feb Sparkfi	5RSA 5 lows io, pruary 2 pruary 2	Spar 020 1 021 1 Spar s)				~
				Edit	Propertie	25	Cop	y to F	
									OK
Expo	ort File For	kport Wiza mat can be expo		variety of	file formats	i.			×
	Select the f	ormat you v	vant to us	e:					
	_	, encoded bi							
	-	e-64 encode							
	<mark>•</mark> Cryp	otographic N	lessage Sy	ntax Stan	dard - PKC	S #7 Certif	icates (.P7	B)	
		Include all o	ertificates	in the cer	tification pa	ith if possil	ble		
		sonal Inform							
		Include all o					ble		
		Delete the p			ort is succe	ssful			
		Export all ex							
		Enable certi							
	⊖ Micr	osoft Seriali	ized Certifi	icate Store	(.SST)				
							Next		Cancel

~	F Certificate Export Wizard
	File to Export Specify the name of the file you want to export
	File name:

Cancel	Next

• After upadating the details Success msg will apear.

←	F Certificate Export Wizard	
	Completing the Certificate Export Wizard	
	You have successfully completed the Certificate Export wizard.	
	You have specified the following settings: File Name C:\Users Desktop\sparkflows.p7b Export Keys No Include all certificates in the certification path No File Format Cryptographic Message Syntax Standard -	s not private st(Certificate Export Wizard × dit ALI The export was successful. ity co
	Finish Cancel	

Next we need to add the exported certificate to the Browser.

Add Certificate to Browser

- Using Google chrome
- Go to below location after opening Google Chrome.
 - Settings -> Advanced -> Privacy and Security-> Manage Certificates
- Click on Manage Certificate icon.
- Click on import.

← -	← → ♂ Chrome chrome://settings/privacy					
Sett	ings	Q Search settings				
÷	You and Google	Site Settings Control what information websites can use and what content they can show you	•			
Ê	Autofill Appearance	Allow Chrome sign-in By turning this off, you can sign in to Google sites like Gmail without signing in to Chrome	-			
Q	Search engine	Send a "Do Not Track" request with your browsing traffic				
	Default browser	Allow sites to check if you have payment methods saved	-			
ل Adva	On startup	Preload pages for faster browsing and searching Uses cookies to remember your preferences, even if you don't visit those pages	-			
0	Privacy and security	Manage certificates Manage HTTPS/SSL certificates and settings	Ľ			
\oplus	Languages					
<u>+</u>	Downloads	Languages				

Sett	Settings		Search settings	
*	You and Google		Warn you if passwords are exposed in a data breach	-
Ê	Autofill		Help improve Chrome security	
0	Privacy and security		To detect dangerous apps and sites, Chrome sends URLs of some pages you visit, limited system information, and some page content to Google	
۲	Appearance		Send a "Do Not Track" request with your browsing traffic	
Q	Search engine			
	Default browser		Allow sites to check if you have payment methods saved	-
U	On startup		Preload pages for faster browsing and searching Uses cookies to remember your preferences, even if you don't visit those pages	
Adva	nced 🔺		Manage certificates	12
۲	Languages		Manage HTTPS/SSL certificates and settings	

Certificates					×
Intended purp	ose: <all></all>				\sim
Personal Oth	ner People Interme	diate Certificatio	n Authorities	Trusted Root Cert	ification Auth
Issued To	Issued By	Expir	Friendly N		
Import	Export	Remove			Advanced
Certificate int	ended purposes				
					View
					Close

• Select certificate from local system, Authorities option and press yes to save it.

use Trusted Root Certification

←	ᡒ Certi	ficate Import Wizard					
		cate Store rtificate stores are system areas where certificates are kept.					
		ndows can automatically select a certificate store, or you can specify a location for e certificate. Automatically select the certificate store based on the type of certificate					
		Place all certificates in the following store					
		Certificate store:					
	Personal Browse						
		Select Certificate Store					
		Select the certificate store you want to use.					
		 Personal Trusted Root Certification Authorities Enterprise Trust Intermediate Certification Authorities Trusted Publishers Untrusted Certificates Third-Party Root Certification Authorities 					
		Show physical stores Next Cancel					
1	b Do	OK Cancel guages					

- Once the above process complete, close the browser and start again and try to login with above URL, It should work without any warnings.
- · Help Url: https://peacocksoftware.com/blog/make-chrome-auto-accept-your-self-signed-certificate

3.1.6 Running on Another Port

There are 2 processes involved when running Fire.

- fire server
- fire

User's Browser talks with fire server, and in turn fire server talks with fire.

Both fire server and fire processes can be configured to listen on different ports.

Running Fire Server on Another Port

By default the fire server runs on the following ports:

- 8080 (http)
- 8443 (https)

Below are the steps for running fire server on a different port.

- Navigate to the conf folder under Sparkflows install directory
- Open application.properties file:

Security Warning



You are about to install a certificate from a certification authority (CA) claiming to represent:

Unknown

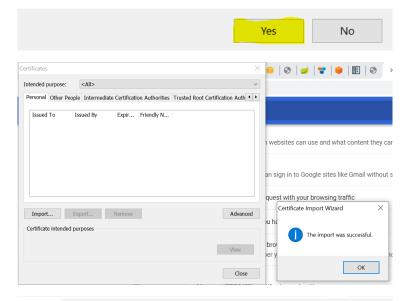
Windows cannot validate that the certificate is actually from "Unknown". You should confirm its origin by contacting "Unknown". The following number will assist you in this process:

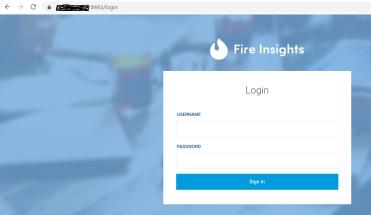
Thumbprint (sha1): 3925098D 23F7B407 3DE02A85 D522B594 1EA68669

Warning:

If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?





- Configure http and https port numbers: Default 8080 for http and 8443 for https
- http.port=8080
- https.port=8443
- In the Fire UI, under Administration/Configuration update the below property with the right port number.
 - app.postMessageURL
- Restart Fire Server using one of the commands below depending on the environment (Unix/Linux or Windows) run-fire-server.sh start run-fire-server.bat

Running Fire on Another Port

Fire by default runs on port 8081.

In order to run Fire on a different port:

- · Navigate to the conf folder under Sparkflows install directory
 - Open application.properties file:
 - Configure the http port
 - fire.http.port=8081
- Restart Fire using ./run-fire.sh start

3.1.7 YARN Configurations

Fire can submit jobs to a YARN cluster. It can submit the spark jobs to run on YARN in either client or cluster mode.

Client Mode

For configuring to run in client mode, set the following parameter under Administration/Configuration:

```
spark.deploy-mode : client
```

In this mode, the spark driver runs on the same machine on which Fire is running. The workflow json file is written out to the directory **/tmp/fire/workflows** on the machine on which Fire is running.

Cluster Mode

For configuring to run in cluster mode, set the following parameter under Administration/Configuration:

```
spark.deploy-mode : cluster
```

In this mode, the spark driver runs on the spark cluster. The workflow json file is written out onto HDFS in the directory **.fireStaging** under the users HOME directory on HDFS.

The spark job reads the workflow json file from HDFS.

Impersonation

• Normally app.impersonateUsers is set to true so that the jobs are run as the logged in user.

```
Note: The logged in user into Fire should exist on HDFS
```

3.1.8 Configuring HTTPS for Fire

Fire server can listen on HTTPS. Fire Server comes with a pre-configured keystore.

Below are the steps for configuring Fire with your keystore and certificates.

Generate a Keystore

You can use the following command for generating the Keystore:

You will be prompted with the following questions and enter something similar to the SAMPLE answers:

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]: John Smith
What is the name of your organizational unit?
 [Unknown]: BigData
What is the name of your organization?
 [Unknown]: MyOrg
What is the name of your City or Locality?
 [Unknown]: San Francisco
What is the name of your State or Province?
 [Unknown]: California
What is the two-letter country code for this unit?
  [Unknown]: CA
Is CN=John Smith, OU=BigData, O=MyOrg, L=San Francisco, ST=California, C=CA correct?
 [no]: yes
Enter key password for <sparkflows>
       (RETURN if same as keystore password): Press the return key or Type and note_
\rightarrow down the password
```

Copy the keystore into the Fire installation directory

Copy the generated keystore.jks file into the conf folder of your installation.

Update the keystore password

Update keystore.properties to udpdate the password of the new keystore.jks file:

```
keystore.password=123456
```

Port Number

Fire by default listens on port 8443 for https.

This is configured in the file conf/application.properties:

```
#Configure http and https port numbers : Default 8080 for http and 8443 for https
http.port=8080
https.port=8443
```

Finally restart the Fire Server

Restart the Fire server for the changes to take effect:

./run-fire-server.sh stop
./run-fire-server.sh start

3.1.9 Configuring Kerberos

Fire runs with a kerberized Spark cluster.

Steps for configuring Kerberos on Fire

- · Generate a keytab for Fire
- Place it in .../fire-x.y.z/conf directory:

```
While this is the recommended location, the keytab file can be placed in any \rightarrow another directory too.
```

• Make sure only the user running fire application has access to the keytab. For example:

-r----- 1 fire staff 436 Jun 29 16:06 hive.keytab

· Go to Administration/Configuration and update the following configurations to enable Kerberos for Fire

Configuration	Example Value	Details
kerberos.enabled	true	Set it to true to enable Kerberos for Fire
kerberos.keytab	/user/ec2-user/fire.keytab	Absolute path of the keytab generated for
		Fire
kerberos.principal	fire@EXAMPLE.COM	Kerberos Principal of the keytab of Fire
ker-	EXAMPLE.COM	Kerberos Realm
beros.KERBEROS_REALM		
kerberos.KERBEROS_KDC	hostname.example.com	KDC Server
kerberos.hiveServer2Principal	hive/hive2_host@EXAMPLE.COM	HIVE Server2 Principal

Steps for generating the keytab for Fire

Below are the steps for generating the keytab file. We have chosen fire as the principal name. But you can have it as any user you are running Fire with.

• Start kadmin.local and add the new principal fire@EXAMPLE.COM:

\$ kadmin.local kadmin.local: addprinc -randkey fire@EXAMPLE.COM WARNING: no policy specified for fire@EXAMPLE.COM; defaulting to no policy Principal "fire@EXAMPLE.COM" created.

• Create fire keytab file:

```
kadmin.local: xst -norandkey -k fire.keytab fire@EXAMPLE.COM
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type aes256-cts-hmac-
⇔sha1-96 added to keytab
WRFILE: fire.keytab.
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type aes128-cts-hmac-
\hookrightarrow shal-96 added to keytab
WRFILE: fire.keytab.
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type des3-cbc-shal_
→added to keytab
                     WRFILE:fire.keytab.
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type arcfour-hmac_
→added to keytab WRFILE:fire.keytab.
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type des-hmac-shal_
→added to keytab WRFILE:fire.keytab.
Entry for principal fire@EXAMPLE.COM with kvno 1, encryption type des-cbc-md5_
→added to keytab WRFILE:fire.keytab.
```

• Exit kadmin.local:

```
kadmin.local: exit
```

Verifying that the keytab file was correctly created

Below are the steps for verifying the keytab file.

• Ensure that the keytab file was created and it has the right permissions:

\$ ls -l fire.keytab -rw----- 1 root root 382 Jul 24 17:55 fire.keytab

• Further verify the contents of keytab file. A normal keytab file depending on your krb5.conf settings, looks like this:

(continued from previous page)

```
1 07/24/16 17:55:07 fire@EXAMPLE.COM (aes256-cts-hmac-shal-96)
1 07/24/16 17:55:08 fire@EXAMPLE.COM (aes128-cts-hmac-shal-96)
1 07/24/16 17:55:08 fire@EXAMPLE.COM (des3-cbc-shal)
1 07/24/16 17:55:08 fire@EXAMPLE.COM (arcfour-hmac)
1 07/24/16 17:55:08 fire@EXAMPLE.COM (des-hmac-shal)
1 07/24/16 17:55:08 fire@EXAMPLE.COM (des-cbc-md5)
```

3.1.10 Configuring Pipelines

Fire uses Apache Airflow for executing Pipelines. Hence Airflow has to be installed on the same machine as Fire. Below are the configurations needed in Fire for Airflow.

(Configurations								
	SAVE CONFIGURAT	INFER HADOOP CLUSTER CONFIG		٩					
A	PP SPARK H	IDFS HADOOP YARN HIVE	KERBEROS LDAP ALERT AIRFLOW						
	NAME TITLE		VALUE	DESCRIPTION					
	airflow.home Airflow HOME		/home/sparkflows/airflow	Airflow HOME Directory					
	airflow.endpointURL Airflow Endpoint URL		http://localhost:8090	G Airflow Endpoint URL					
	airflow.enabled	Enable/Disable airflow	true	C Enable/Disable airflow feature					
	SAVE CONFIGURATIONS								

Airflow Installation

It explain the steps involved in installing Airflow on Centos and RHEL. Detailed Airflow Install Instructions is at:

https://airflow.apache.org/installation.html

- · Login to machine
- Before installing airflow update installed package:
- yum -y update
- Install python-pip and any required packages:
- sudo yum install epel-release
- sudo yum install python-pip
- Check the version of pip that is installed and if reqd upgrade:
- pip -V
- pip install –upgrade setuptools
- Note that for 1.10 you now need to preface install commands or export this env var:
- export SLUGIFY_USES_TEXT_UNIDECODE=yes

- Install gcc , gcc-c++ and dependencies for python 2.7
- sudo yum -y install gcc gcc-c++ kernel-devel
- sudo yum -y install python-devel libxslt-devel libffi-devel openssl-devel
- Airflow needs a home, ~/airflow is the default
- export AIRFLOW_HOME=~/airflow
- Install from pypi using pip
- pip install apache-airflow
- To check airflow version
- · airflow version

[sparkflows@airflowmachine ~1\$ airflow version [2019-01-10 07:45:54,808] {__init__.py:51} INFO - Using executor SequentialExecutor

- Generate a Fernet key for Airflow(optional)
- python -c "from cryptography.fernet import Fernet; print(Fernet.generate_key().decode())"
- fgrc0MPUG1n3Q352Fp705A-bysNHX6EFRr7nYFTmXXA=
- update in airflow.cfa
- fernet key: fgrc0MPUG1n3Q352Fp705A-bysNHX6EFRr7nYFTmXXA=
- Initialize the Airflow database
- airflow initdb
- Start the web server, its default port is 8080, If any other application is running on 8080, we can update other port for airflow
- airflow webserver -p 8090

[sparkf lows@airf lownachine ~15 airf low webserwer -p 8090 [2819-61-18 07:52:31,003] <initpy:51)< td=""> [INFO - Using executor SequentialExecutor [INFO - Using executor SequentialExecutor</initpy:51)<>	
<pre>12019-01-10 07:52:31,585] (models.py:271) INFO - Filling up the DagBag from /home/sparkflows/airflow/dags 12019-01-10 07:52:31,595] (models.py:380) ERROR - Failed to import: /home/sparkflows/airflow/dags Iraceback (most recent call last): File "/usr/lih/python2.7/site-packages/airflow/models.py", line 377, in process_file n = imp.load_source(mod_name, filepath) File "/home/sparkflows/airflow/dags/lest-pieline_3.py", line 9 Test-pieline_3.dag = DAG('Iest-pieline_3', description= '', schedule_interval=None, catchup=False, default_args=args) SyntaxError: can't assign to operator Running the Cunicorn Server with: Morkers: 4 sync Insecut: 128 Logfiles:</pre>	
<pre>[2819-61-16 87:52:32 +06061 r355901 [INFO] Starting gunitering 19 6 [2819-61-16 87:52:32 +06061 r355901 [INFO] Listeng gunitering 19 6 [2819-61-16 87:52:32 +06061 r355901 [INFO] Listeng uorker vith pid: 35600 [2819-61-16 87:52:32 +06061 r355001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +06061 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +06061 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32 +060601 r356001 [INFO] Booting worker vith pid: 35600 [2819-61-16 87:52:32,8801 [.initpy:51] INFO - Using executor SequentialExecutor [2819-61-16 87:52:32,8801 [.initpy:51] INFO - Using executor SequentialExecutor [2819-61-16 87:52:32,9801 [.initpy:51] INFO - Filling up the DagBag from /home/sparkflow/dags/lest-pieline_3.py [raceback freenet coall last]: I main_load_source(nod name, filepapt) File "/home/sparkflows/airflow/dags/lest-pieline_3.py", line 377, in process_file I main_load_source(nod name, filepapt) File "/home/sparkflows/airflow/dags/lest-pieline_3.py", line 37, schedule_interval=None, catchup=False, default_args=args) SyntaxError: can't assign to operator [2819-61-16 87:52:33,127] (models.py:71) INFO - Filling up the DagBag from /home/sparkflows/airflow/dags [2819-61-16 87:52:33,127] (models.py:721) INFO - Filling up the DagBag from /home/sparkflows/airflow/dags [2819-61-16 87:52:33,2201 [.info - py:721] INFO - Filling up the DagBag from /home/sparkflows/airflow/dags [2819-61-16 87:52:33,2201 [.info - py:721] INFO - Filling u</pre>	

• Start the scheduler

[sparkflows@airflowmachine ~]\$ airflow scheduler [2019-01-10 07:55:57,016] {initpy:51} INFO - Using executor	r SequentialExe	cutor				
2019-01-10 07:55:57,231 (jobs.py:583) ERROR - Cannot use more than 1 thread when using sqlite. Setting max_threads to 1 2019-01-10 07:55:57,233 (jobs.py:1545) INFO - Processing files using up to 1 processes at a time 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Processing files using up to 1 processes at a time 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Processing file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Processing file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Processing file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each file at most - 1 times at flow/dags 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each files in /home/sparkflows/airflow/dags 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each files in /home/sparkflows/airflow/dags 2019-01-10 07:55:57,233 (jobs.py:1540) INFO - Process each files in /home/sparkflows/airflow/dags						
DAG File Processing Stats						
File Path	PID Runtime	Last Runtime	Last Run			
/home/sparkflows/airflow/dags/Testpipelineworkflow.py	36859 0.90s					
/hone/sparkflous/airflow/dags/pipeline1csu_455.py /hone/sparkflous/airflow/dags/vpPipeline_99.py /hone/sparkflous/airflow/dags/SQL_PIPELINE_TEST_33.py /hone/sparkflous/airflow/dags/Telco_churn_pipeline_66.py						

- airflow scheduler
- Login in browser
- http://x.y.z.w:8090

XAirflow DAGs Data Profiling - Browse + Admin + Docs + About + 2019-01-10 07:35:02 UTC										
DA	DAGs									
Search:										
	0	DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links		
ø	On	Cars_Testcsv_pipeline_129	None	sparkflows		2019-01-10 06:53 🚯	$\bigcirc \bigcirc \bigcirc \bigcirc \bigcirc$	◎ ♥ 兼 由 翰 朱 重 ゲ 重 ♡ ⊗		
ø	On	Housingpipelineworkflow_459	None	sparkflows	00000000	2019-01-08 11:22 0	$\bigcirc \bigcirc \bigcirc$	⊙ ₱ # di liù 木 트 ⁄ 		
ø	On	SQL_CSV_PIPELINE_65	None	sparkflows	0000000	2019-01-09 11:38 0	$\bigcirc \bigcirc \bigcirc$	⊙ ♥ 兼 di 🛍 木 ☱ ☱ 🖓 😣		
ø	On	SQL_PIPELINE_TEST_33	None	sparkflows	00000000	2019-01-09 11:34 🕄	3	⊙ ♥ 兼 di 🛍 木 ☱ ☱ 🖓 😣		
Ø	On	SQL_Pipeline_482	None	sparkflows	00000000	2019-01-08 20:42 🕄		⊙ ♥ 業 di 🛍 🛧 ☱ ☱ 🛇 😣		
Ø	On	TELCO_CSV_PIPELINE_67	None	sparkflows	0000000	2019-01-10 05:36 🚯	3	। • • • • • • • • • • • • • • • • • • •		
Ø	On	Telco_churn_pipeline_66	None	sparkflows	0000000	2019-01-09 11:41 🕄	$\bigcirc \bigcirc \bigcirc$	। • • • • • • • • • • • • • • • • • • •		
ø	On	TestRashmi 0		sparkflows				≎ ⊗		
ø	On	TestRashmi_290	None	sparkflows	30000000	2019-01-08 08:56 3	$\bigcirc \bigcirc \bigcirc$	○ ♥ 業山論 木圭 ゲ ≣ ♡ ⊗		
ø	On	Testpipelineworkflow	None	sparkflows	30000000	2019-01-04 12:36 🚯	2	○ ♥ 業山論未至 / Ⅲ 2 ⊗		
Ø	On	Testpipelineworkflow_99		sparkflows				€ ⊗		
Ø	On	Years_Telco_pipeline_97	None	sparkflows	0000000	2019-01-10 06:49 🚯		◎ ♥ 兼 山 翰 朱 重 ゲ 重 ♡ ⊗		
ø	On	Years_cars_pipeline_68	None	sparkflows	00000000	2019-01-10 05:51 🚯		⊙ ♥ 兼 di 🛍 木 ☱ ☱ 📿 😣		
Ø	on	example_bash_operator	0 0 * * *	airflow	0000000	2019-01-03 00:00	5	 ○ ● ※ 山 翰 木 三 ヶ 三 ○ ⊗ 		
	_									

3.1.11 Different Default Values on Startup

Overview

Fire has a number of properties under Administration / Configuration. When initially installed they have certain default values. Administrators can log into Fire through their Browser and update the Properties.

However, there might be cases where you want Fire to come up with different default values for the Configurations when installed. This enables more automation and the Administrator does not have to go in and manually change the

default values.

Steps

Below are the Steps to override the default Configuration values:

- Update the file conf/configuration.properties with the new key/value pairs
- Now the default values are populated with the values provided in configuration.properties.

Fire comes with an empty conf/configuration.properties file. You can put in your values into it.

Remove properties from conf/configuration.properties

Fire will continue to take the final values from conf/configuration.properties for any property which is there in the file.

If you would like Fire not to use any of the properties from conf/configuration.properties, but take it from the database, then remove or comment out those properties in conf/configuration.properties.

Saving the new values into the DB

When the configuration values are saved, they get updated in the database.

Even if they are removed from configuration.properties, they would have been saved in the database.

3.1.12 Configuring LDAP/OAuth Authentication

Fire Insights supports various types of authencations:

- Database Authentication
- LDAP Authentication
- OAuth Authentication

Database Authentication

Fire Insights can authenticate the user against its own database.

User's password are stored encrypted.

This is the default authentication mechanism of Fire Insights. Users created in Fire are stored in the database.

LDAP Authentication

Fire Insights can be configured to authenticate the user against LDAP. Users have to be added to Fire, before they can log into Fire and start using it.

The following configurations have to be set appropriately.

configuration/authentication/../_assets/installation/ldap

LDAP Parameters

Name of Param-	Description	Example	
eter			
ldap.Order	Order in which to authenticate the user. Possible values are		
	DB, LDAP_DB, DB_LDAP.		
ldap.URL	The URL of the LDAP server. The URL must	ldap://localhost:10389	
	be prefixed with ldap:// or ldaps://. The URL		
	can optionally specify a custom port, for example:		
	ldaps://ldap_server.example.com:1636.		
ldap.Base	The distinguished name to use as a search base for	dc=sparkflows,dc=com	
	finding users and groups. This should be similar to		
	'dc=sparkflows,dc=com'.		
ldap.UserDn	Distinguished name of the user to bind as. This is used to	uid=john,ou=development,dc=	sparkflows,dc=co
	connect to LDAP/AD for searching user and group informa-		
	tion. This may be left blank if the LDAP server supports		
	anonymous binds.		
ldap.Password	The password of the bind user.	xyz	
ldap.UserSearchBas		ou=development	
ldap.UserSearchFilt	efThe base filter for searching for users. For Active Directory,	For Active Directory : (ob-	
	this is typically '(objectClass=user)'.	jectClass=user) Other Exam-	
		ple : (uid={0})	
	as Broup Search Base	ou=groups	
Idap.GroupSearchFilterroup Search Filter		For Active Directory : (ob-	
		jectClass=group) Other Ex-	
		ample : (member={0})	

Table 1: LDAP Parameters

Note

For ldap.UserSearchFilter we can use strings like (uid={USERNAME}) In this case {USERNAME} would be replaced by the real username of the user when searching in LDAP during Add User.

LDAP Certificate

If ldaps is being used, the ldap certificate needs to be imported into cacerts.

For Reference : https://docs.oracle.com/cd/E19509-01/820-3399/ggfrj/index.html

Importing a user from LDAP into Sparkflows

Once LDAP is enabled in Sparkflows, users can be imported into Sparkflows from LDAP.

- Go to Administration/User
- Click on Add/Sync User
- Enter the username and click on Search
- User details are fetched from LDAP
- Click on Add User to create the user in Sparkflows

User Login

Once LDAP is enabled in Sparkflows, all the authentication for login in Sparkflows are done against LDAP.

Search Order

Sparkflows would search in LDAP and then in its DB. Search order is determined by the parameter ldap.Order.

If it is set to LDAP_DB, it would first search for the User in LDAP and then in its own DB. This allows having the admin user in the Sparkflows DB if needed, so that all users are not locked out of the system in case LDAP goes down or ends up with invalid Configurations.

Reference

Below are some great links for reference:

• Active Directory Search Filter Syntax : https://msdn.microsoft.com/en-us/library/aa746475(v=vs.85).aspx

What if I get locked out

ldap.Order determines the order in which Sparkflows tries to log in the user. In case you are locked out of Sparkflows and are not able to log in, you can do the following:

• Add the below line to conf/configuration.properties:

ldap.Order=DB

• Then restart the fire server. Now you should be able to log in with your admin account.

Once things are back to normal, you can remove the line you added to configuration.properties and restart the fire server.

Notes

• Search strings are not case sensitive

OAuth Authentication

Fire Insights supports OAuth Authentication.

Create Users in Fire

First create the user in Fire under Administration/Users.

Log into Fire with the admin user in order to be able to create the New Users.

Configuring OAuth

In order the configure OAuth in Fire Insights, add the OAuth configuration parameters to conf/application. properties.

Below is an example of configuring OAuth in Fire with Okta.

```
# Okta settings
oauth.client.clientId: 0oadvfdsfsdA7Y68356
oauth.client.clientSecret: YSWFdZf9kfdsfsdfsdfsdfsdnI0SVrswOJpHl
oauth.client.accessTokenUri: https://xyz.okta.com/oauth2/default/v1/token
oauth.client.userAuthorizationUri: https://xyz.okta.com/oauth2/default/v1/authorize
oauth.client.clientAuthenticationScheme: form
oauth.client.scope: openid profile email
oauth.resource.userInfoUri: https://xyz.okta.com/oauth2/default/v1/userinfo
```

Fire OAuth URL

In order to log in the user into Fire using OAuth, use the following URL:

• http://machine_name:port/login/oauth

This URL will take the user to the OAuth login page. After the user logs in there, the user is redirected back to Fire and is logged in.

If the user is already logged in, going to the above URL, automatically brings up the Fire page for the user.

3.1.13 HDInsight Integration

Fire Insights runs seamlessly on Azure HDInsight.

Fire can be installed on the master or edge nodes of the cluster.

HDInsights and Ports

https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-port-settings-for-services

Linux-based HDInsight clusters only expose three ports publicly on the internet; 22, 23, and 443. These ports are used to securely access the cluster using SSH and services exposed over the secure HTTPS protocol.

Internally, HDInsight is implemented by several Azure Virtual Machines (the nodes within the cluster) running on an Azure Virtual Network. From within the virtual network, you can access ports not exposed over the internet. For example, if you connect to one of the head nodes using SSH, from the head node you can then directly access services running on the cluster nodes.

To join additional machines to the virtual network, you must create the virtual network first, and then specify it when creating your HDInsight cluster. For more information, see Extend HDInsight capabilities by using an Azure Virtual Network

https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-extend-hadoop-virtual-network

Port Configuration

Fire Insights by default listens on ports 8080 and 8443.

On HDInsight, port 8080 generally is already in use. So configure Fire Insights to listen on another port, say 8090.

Edit conf/application.properties:

```
#Configure http and https port numbers : Default 8080 for http and 8443 for https
http.port=8090
https.port=8443
```

Open the Port for access

Now the port 8090 needs to be opened to be accessed by the users using their Browser.

https://stackoverflow.com/questions/45239566/accessing-http-on-custom-port-in-azure-hdinsight-cluster

Add proxy user

Fire needs to impersonate the logged in user.

In Ambari for the HDInsight cluster, add the Fire user in HDFS to be the proxy user.

Suppose Fire is installed as the user fire. Add the below to HDFS/Configuration in Ambari:

```
hadoop.proxyuser.fire.groups=*
hadoop.proxyuser.fire.hosts=*
```

Connecting Fire Insights to the HDInsight Cluster

In Fire Insights, under Administration/Configuration, configure the following for it to be able to connect to the HDInsight cluster.

- hdfs.namenodeURI=wasb://
- app.runOnCluster=true
- app.postMessageURL=
- app.sparkSubmitJar=

Clicking on Infer Hadoop Configuration would correctly infer these. Hit Save after that.

3.1.14 MapR Integration

This document describes details when installing Fire Insights on a MapR cluster.

Download Fire Insights

• Download MapR specific binary from : https://www.sparkflows.io/archives

Turn off Impersonation

• In Administration / Configuration of Sparkflows:

Update http port

• Set http port` to be different in `conf/application.properties if there are other processes using the specified ports

Fire User

• Fire has to be installed as a user which can submit jobs to the MapR cluster. Say we installed Fire as user mapr:

```
Create a mapr user {\tt in} sparkflows {\tt and} log {\tt in} {\tt as} that user Start using Sparkflows
```

3.1.15 Upgrading Fire

Stop Fire if it is running

Stop Fire with the below command from the directory in which it is installed:

run-fire-server.sh stop

Download the new fire tgz file

Download Fire tgz file from:

```
https://www.sparkflows.io/download OR
https://www.sparkflows.io/archives
```

Unpack it

Unpack the tgz file with below on unix/linux:

tar xvf fire-x.y.z.tgz

Upgrade the H2 or MySQL database

- If you have updated the conf/db.properties file, copy it from your old location to the new directory
- Backup your existing H2 db files. By default they are in your home directory as firedb.mv.db
- If you are using MySQL, backup the fire database in MySQL.

• Execute the following commands on the Command Line to upgrade the Fire database:

```
cd <install_dir>/fire-x.y.z
./create-h2-db.sh OR ./create-mysql-db.sh
```

the above command creates or updates the existing db if one already exists

Restart Fire

Restart the Fire Server:

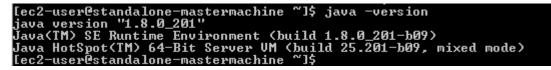
run-fire-server.sh start

3.1.16 Running Apache Spark Standalone

Fire can be run on Spark Standalone cluster. In this case, Hadoop does not need to be installed.

Installing Spark Standalone

- Install Java
 - wget –no-cookies –no-check-certificate –header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "https://download.oracle.com/otn-pub/java/jdk/8u201-b09/ 42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.rpm"
 - yum localinstall jdk-8u201-linux-x64.rpm
 - Java -version



Install Scala

- Install Scala
 - wget http://www.scala-lang.org/files/archive/scala-2.10.1.tgz
 - tar xvf scala-2.10.1.tgz
 - sudo mv scala-2.10.1 /usr/lib
 - sudo ln -s /usr/lib/scala-2.10.1 /usr/lib/scala
 - export PATH=\$PATH:/usr/lib/scala/bin (we can add in .bash_profile)
 - scala -version

```
[ec2-user@standalone-mastermachine ~]$ scala -version
Scala code runner version 2.10.1 -- Copyright 2002-2013, LAMP/EPFL
[ec2-user@standalone-mastermachine ~]$
```

Install Apache Spark

- Download Spark
 - wget http://d3kbcqa49mib13.cloudfront.net/spark-2.1.0-bin-hadoop2.7.tgz
- Extract, create a new directory under the /usr/local called spark and copy the extracted connect into it
 - tar xf spark-2.1.0-bin-hadoop2.7.tgz
 - mkdir /usr/local/spark
 - cp -r spark-2.1.0-bin-hadoop2.7/* /usr/local/spark
- Setup some Environment variables before you start spark-shell (in .bash_profile)
 - export SPARK_EXAMPLES_JAR=/usr/local/spark/examples/jars/spark-examples_2.11-2.0.0.jar
 - PATH=\$PATH:\$HOME/bin:/usr/local/spark/bin
- Start you Scala Shell and run Spark
 - Go to sparkflows home directory
 - cd /usr/local/spark/bin
 - ./spark-shell

[root@standalone=mastermachine bin]# ./spark-shell Using Spark's default log4j profile: org/apache/spark/log4j=defaults.properties Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel). 19/02/11 07:14:14 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform using builtin-java classes where applicable 19/02/11 07:14:14 WARN Utils: Service 'SparkU' could not bind on port 4040. Attempting port 4041.
19/02/11 07:14:18 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException Spark context Web UI available at http://do.0.2.59:4041
Spark context available as 'sc' (master = local[*], app id = local-1549869255043). Spark session available as 'spark'. Melcome to
Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_201) Type in expressions to have them evaluated. Type :help for more information.
scala>

- Start a standalone master server by executing:
 - ./sbin/start-master.sh (from spark home directory)
- Once started, the master will print out a spark://HOST:PORT URL
- You can also find this URL on the master's web UI,
 - http://Master_host_ip:8080/ by default

Setup Spark Slave(Worker) Node

- Go to SPARK_HOME/conf/ directory.
- Edit the file spark-env.sh Set SPARK_MASTER_HOST
 - If spark-env.sh is not present, spark-env.sh.template would be present. Make a copy of sparkenv.sh.template with name spark-env.sh and add/edit the field SPARK_MASTER_HOST. Part of the file with SPARK_MASTER_HOST
 - cp ./conf/spark-env.sh.template ./conf/spark-env.sh
- Add a line in spark-env.sh :
 - SPARK_MASTER_HOST='MASTER_HOST_IP'

Spark Master at spark://standalone-mastermachine.utubfjsv433eppdemeo0qw3myh.bx.internal.cloudapp.net:7077

URL: spark://standalone-mastermachine.u REST URL: spark://standalone-mastermac Allrew Workers: 1 Cores in use: 16 Total 2 Used Memory in use: 61 9 GB Total 2 0 GB Us Applications: 0 Running, 3 Completed Drivers: 2 Running, 2 Completed Status: ALIVE Workers	hine.utubfjsv										
Worker Id				Address	State	Cores			Memory		
worker-20190208095216-10.0.2.59-44876			10.0.2.59:44876	ALIVE	16 (2 U	16 (2 Used) 61.9 GB (2.0 GB Used)					
Running Applications		0								0 1-1-	Printle
Application ID	Name	Cores	Memory pe	er Node	Submitted Time	e		User		State	Duration
Running Drivers											
Submission ID	Sub	mitted Time		Worker		State	Cores	Memory	M	ain Class	
driver-20190211051401-0003	(kill) Mor	Feb 11 05:14:01 UTC 2	019	worker-20190208095216-10.0.2.59-	worker-20190208095216-10.0.2.59-44876		G 1 1024.0 M		org.apache.spark.examples.SparkPi		examples.SparkPi
driver-20190211050948-0002	(kill) Mor	1 Feb 11 05:09:48 UTC 2	019	worker-20190208095216-10.0.2.59-	44876	6 RUNNING 1		1024.0 M	B on	g.apache.spark.	examples.SparkPi
Completed Applications		Namo	Correc	Momony nor Nodo	Submitte			lie		State	Duration

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20190208124403-0002	Spark shell	16	1024.0 MB	2019/02/08 12:44:03	root	FINISHED	1.6 h
app-20190208122832-0001	Spark shell	16	1024.0 MB	2019/02/08 12:28:32	root	FINISHED	35 s
app-20190208114249-0000	Spark shell	16	1024.0 MB	2019/02/08 11:42:49	root	FINISHED	37 min

Start spark as slave

- Goto SPARK_HOME/sbin and execute the following command.
 - ./start-slave.sh spark://MASTER_HOST_IP:7077

Installing Fire

Install Fire on the master node.

- Download Fire Jar from website
 - wget https://s3.amazonaws.com/sparkflows-release/fire/rel-x.y.z/2/fire-x.y.z.tgz
 - tar xvf fire-x.y.z.tgz
- Go to below directory:
 - cd fire-x.y.z
 - Update the port of Fire-ui & Fire to 8090 & 8082 as default port 8080 & 8081 is used by standalone spark, we can chose any other also.
 - From fire-x.y.z directory, we need to go conf/application.properties and update the port No.

```
#Configure http and https port numbers : Default 8080 for http and 8443 for https
http.port=8090
https.port=8443
# Configure http ports for fire
fire.http.ports=8082
spring.jackson.deserialization.FAIL_ON_UNKNOWN_PROPERTIES=false
```

- Create database & run fire & fire-ui server
 - ./create-h2-db.sh
 - ./run-fire.sh start

- ./run-fire-server.sh start

Configuring Fire

Below are the configuration for Fire to submit the jobs to the Spark Standalone Cluster.

- Once The server fire & fire-ui start
 - Login to http://Machine_ip:8090/#/dashboard
 - With password admin/admin.
 - Upload default applications.
 - Create a user ec2-user.
 - Login with ec2-user

configurations in spark

The following configurations have to be set appropriately

- · Go to administration section and open Spark configuration there we need to add Below details in specific setup like below
 - spark.master: spark://Master_host_ip:7077
 - spark.deploy-mode: client
 - spark.sql-context: SQLContext
 - After above updates save the configurations.

PP AIRFLOW	SPARK HDFS HADOOP YA	RN HIVE KERBEROS LDAP ALERT	
NAME	TITLE	VALUE	DESCRIPTION
spark.master	Spark Master	spark://Master_host_ip:7077	If spark-submit is used, should it run locally or on the cluster. Possible values : local/yarn/spark/mesos More details at : https://spark.apache.org/docs/latest/submitting- applications.html#master-urls
spark.deploy-mode	Spark deploy mode	client	Whether to deploy the spark driver on the worker nodes (cluster) of locally as an external client (client) (default: client). Possible values client / cluster
spark.historyServerURL	Spark History Server URL	http://localhost:18089	Spark History Server URL
spark.executor- memory	Spark Executor Memory		Spark Executor Memory size to be used in spark-submit. Not used if it is empty. eg: 1G,
spark.num-executors	Number of Executors (Not used)		Not Used - Enable dynamic allocation instead on the spark cluster - spark.dynamicAllocation.enabled
spark.executor-cores	Number of Spark Executor Cores	1	Number of Spark Executor Cores to be used in spark-submit. Not used if its value is less than or equal to 0
spark.sql-context	SQLContext or HiveContext	SQLContext	Whether to use SQLContext or HiveContext
spark.spark-submit	spark-submit command to use	spark-submit	Use spark2-submit with spark2, depending on your setup

Now go to application and try to run any workflows

ïre Insights 🛛 DATA BROWSERS 👻 🔿 APPLICATIONS 👻 🧿 SCHEDULED 👻		E PROCESSORS -	≅ • o\$ • ≜ •
total_day_calls			
total_day_charge			
total_eve_minutes	╶┼╌┼╌┼╌┞╼╏╼╸		
total_eve_calls total_eve_charge	╶╌╌╌╌╌╴╴		
total_eve_cnarge total_night_minutes			
total_night_calls			
total_night_charge			
total_intl_minutes			
total_intl_calls			
total_intl_charge			
number_customer_service_calls			
Output Schema		 	
Executing Node fire.nodes.doc.NodeDocLarge : 14 Feb 11, 2019 7:03:10 AM			
Executing house methodes.upc.housebockdigs , 14 Feb 11, 2013 7.33.10 Avi			
Successfully finished executing the workflow			

3.1.17 Running Fire as a Service

Fire Insights can be configured to run as a service. This way when the machine reboots, Fire Insights would be automatically restarted.

Below are the steps for configuring Fire Insights as a service.

CHAPTER 4

Authentication

4.1 Authentication

Fire Insights supports various types of authencations:

- Database Authentication
- LDAP Authentication
- OAuth Authentication

4.1.1 Database Authentication

Fire Insights can authenticate the user against its own database.

User's password are stored encrypted.

This is the default authentication mechanism of Fire Insights. Users created in Fire are stored in the database.

4.1.2 LDAP Authentication

Fire Insights can be configured to authenticate the user against LDAP. Users have to be added to Fire, before they can log into Fire and start using it.

The following configurations have to be set appropriately.

23	ldəp.Order	LDAP Order	08		LDAP Order. Possible values : DB / DB ,LDAP. For any change to be reflected required restart of the Sparkflows server.
24	Idap.URL	LDAP URL	Idap.//loca/host.10389	œ	LDAP URL
25	idap.Base	LDAP Base	dc+example.dc+com	8	LDAP Base
26	Idap.UserDn	LDAP UserDn	uid+john,ou=bindusers,dc=example,dc=com	œ	LDAP UserDn
27	Idap.Password	LDAP Password	johnspassword	œ	LDAP Password
28	idap.UserSearchBase	LDAP UserSearchBase	oursparkflow	ß	LDAP UserSearchBase
29	Idap.UserSearchFilter	LDAP UserSearchFilter	0uid+(00)	8	LDAP UserSearchFilter
30	Idap.GroupSearchBase	LDAP GroupSearchBase	ourgroups	œ	LDAP GroupSearchBase
31	Idap.GroupSearchFilter	LDAP GroupSearchFilter	member+(0)	8	LDAP GroupSearthFilter

LDAP Parameters

	Table 1. LDAF Farameters		
Name of Param-	Description	Example	
eter			
ldap.Order	Order in which to authenticate the user. Possible values are		
	DB, LDAP_DB, DB_LDAP.		
ldap.URL	The URL of the LDAP server. The URL must	ldap://localhost:10389	
	be prefixed with ldap:// or ldaps://. The URL		
	can optionally specify a custom port, for example:		
	ldaps://ldap_server.example.com:1636.		
ldap.Base	The distinguished name to use as a search base for	dc=sparkflows,dc=com	
	finding users and groups. This should be similar to		
	'dc=sparkflows,dc=com'.		
ldap.UserDn	Distinguished name of the user to bind as. This is used to	uid=john,ou=development,dc=	sparkflows,dc=co
	connect to LDAP/AD for searching user and group informa-		
	tion. This may be left blank if the LDAP server supports		
	anonymous binds.		
ldap.Password	The password of the bind user.	xyz	
ldap.UserSearchBas	seUser Search Base	ou=development	
ldap.UserSearchFilt	efThe base filter for searching for users. For Active Directory,	For Active Directory : (ob-	
	this is typically '(objectClass=user)'.	jectClass=user) Other Exam-	
		ple : (uid={0})	
	as Group Search Base	ou=groups	
ldap.GroupSearchF	Iteroup Search Filter	For Active Directory : (ob-	
		jectClass=group) Other Ex-	
		ample : (member={0})	

Table 1: LDAP Parameters

Note

For ldap.UserSearchFilter we can use strings like (uid={USERNAME}) In this case {USERNAME} would be replaced by the real username of the user when searching in LDAP during Add User.

LDAP Certificate

If ldaps is being used, the ldap certificate needs to be imported into cacerts.

For Reference : https://docs.oracle.com/cd/E19509-01/820-3399/ggfrj/index.html

Importing a user from LDAP into Sparkflows

Once LDAP is enabled in Sparkflows, users can be imported into Sparkflows from LDAP.

- Go to Administration/User
- Click on Add/Sync User
- Enter the username and click on Search
- User details are fetched from LDAP
- Click on Add User to create the user in Sparkflows

User Login

Once LDAP is enabled in Sparkflows, all the authentication for login in Sparkflows are done against LDAP.

Search Order

Sparkflows would search in LDAP and then in its DB. Search order is determined by the parameter ldap.Order.

If it is set to LDAP_DB, it would first search for the User in LDAP and then in its own DB. This allows having the admin user in the Sparkflows DB if needed, so that all users are not locked out of the system in case LDAP goes down or ends up with invalid Configurations.

Reference

Below are some great links for reference:

• Active Directory Search Filter Syntax : https://msdn.microsoft.com/en-us/library/aa746475(v=vs.85).aspx

What if I get locked out

ldap.Order determines the order in which Sparkflows tries to log in the user. In case you are locked out of Sparkflows and are not able to log in, you can do the following:

• Add the below line to conf/configuration.properties:

ldap.Order=DB

• Then restart the fire server. Now you should be able to log in with your admin account.

Once things are back to normal, you can remove the line you added to configuration.properties and restart the fire server.

Notes

• Search strings are not case sensitive

4.1.3 OAuth Authentication

Fire Insights supports OAuth Authentication.

Create Users in Fire

First create the user in Fire under Administration/Users.

Log into Fire with the admin user in order to be able to create the New Users.

Configuring OAuth

In order the configure OAuth in Fire Insights, add the OAuth configuration parameters to conf/application. properties.

Below is an example of configuring OAuth in Fire with Okta.

```
# Okta settings
oauth.client.clientId: 0oadvfdsfsdA7Y68356
oauth.client.clientSecret: YSWFdZf9kfdsfsdfsdfsdfsdfsdnI0SVrswOJpHl
oauth.client.accessTokenUri: https://xyz.okta.com/oauth2/default/v1/token
oauth.client.userAuthorizationUri: https://xyz.okta.com/oauth2/default/v1/authorize
oauth.client.clientAuthenticationScheme: form
oauth.client.scope: openid profile email
oauth.resource.userInfoUri: https://xyz.okta.com/oauth2/default/v1/userinfo
```

Fire OAuth URL

In order to log in the user into Fire using OAuth, use the following URL:

• http://machine_name:port/login/oauth

This URL will take the user to the OAuth login page. After the user logs in there, the user is redirected back to Fire and is logged in.

If the user is already logged in, going to the above URL, automatically brings up the Fire page for the user.

4.1.4 SSO

Single sign-on (SSO) enables you to authenticate your users using your organization's identity provider. If your identity provider supports the SAML 2.0 protocol, you can use Fire Insights SSO to integrate with your identity provider.

Below are the steps for setting up & configuring OneLogin with Fire Insights.

SAML OneLogin setup

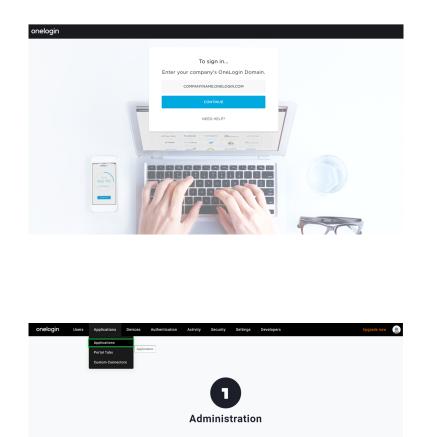
Below are steps to setup SAML 2.0 OneLogin

- 1. Create an account at one Login
- 2. SignIn into oneLogin
- 3. Go to administrator >> Click on applications menu >>
- 4. Add an app
- 5. Select an application:

```
Search application 'SAML Test Connector'
Select the application SAML Test Connector (Advanced).
```

Vinit

Add App



nelogin Users Applications	Devices Authentication	Activity Security	Settings Developer	5	Upgrade now 🧕 Vinit
ind Applications					
Q. SAML Test Connector					
SAML Test Connector (Advanced) OneLogin, Inc.				SAML2.0	
SAML Test Connector (SP Shibboleth) OneLogin, Inc.				SAML2.0	

Activity Security

onelogin

Applications

6. Input an application name and save it.

onelogin User	Applications	Devices Authenticatio	n Activity	Security	Settings	Developers	Upgrade now	
App Listing / Add SAML Test Co	nnector (Advanc	ced)					Cancel	Save
Configuration	Portal Display Na Sparkflo Visible in p	ws						
		ar loon		are Icon	e icon at least 51 irrent .PNG or .SVI			

7. Configure the newly created app and add below information:



onelogin Users	Applications Devices Authentication Activity Security Settings Developers	Upgrade r
pplications / SAML Test Connec	tor (Advanced)	More Actions
info		
Configuration	Auchence (TrittlyD)	
Parameters	https://iocalhost.8443/sparkflow/saml/metadata	
Rules	Recipient	
550	https://tocahoot.8443/samUSS0	
Access	ADS (Consumer) URI, Validator*	
Users	A second s	
Privileges	() "Required.	
Setup	AC8 (Consumer) URL*	
	https://localhost.8443/sami/SSO	
	① *Required	
	Single Legout URL	
	https://ocalhost.8443	
	Login URI,	
	https://ocalhoot.8443	
	(i) Only required if you select Service Provider as the SAML Initiative.	

Fire Insights SAML oneLogin Configuration

Fire Insights can be Configured with SAML 2.0 OneLogin as below.

Go to folder conf/sso.saml.properties file:

Add below information from newly created application in oneLogin:

1. Enable sso in sparkflows:

sparkflows.sp.sso.enable=true

2. Create user locally in application if user doesn't exist in Fire Insights, otherwise app will show page 'User not found':

sparkflows.sp.auto.user.create=true

3. Metadata url of identity provider.

4. Identifier of the SP entity (must be a URI) Audience URI

onelogin user	Applications Devices Arthenetication Archity Security Settings Developers	Upgrade now 🥘 🔌
Appleations / SAML Test Connect	tor (Advanced)	More Actions + 11
		Vendor Homepage
infa		Paupply artiflament mappings
Configuration	Autorice (EdityE)	SAML Metadata
Patameters	https://iocalhost.1443/sparkflow/sami/instradata	Debde
N/ei	Request	
550	Migs:/licelited.8443/seet/350	
Access	BCR (Carename) UPL Valuation*	
Starts.		
Prinleges	(i) Headed	
Setup	act Gauged W11	

saml2.sp.entityid=https://localhost:8443/sparkflow/saml/metadata

5. Identifier of the IdP entity (must be a URI)

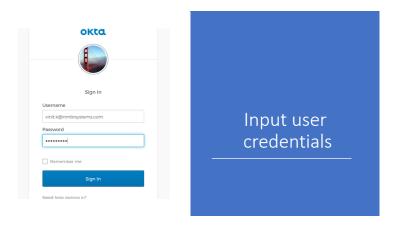
SAML Test Connec	or (Advanced)	
into		
Configuration	Audionice (Entity/E)	
Parameters	https://tocalhost.84.80/spaikflow/saminmetadata	
Rules	Recplant	
\$90	https://ocalhott.0442/sam0/550	
Access	ALS (Denume) US: Valuator*	
Users		
Privileges	Stephend	
Setup	ALS (Comumer) LRL*	
	https://ocalhoet3048/stami/550	
	(a) "Required	

6. Algorithm that the toolkit will use on signing process.

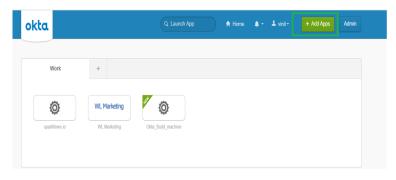
SAML Test Conne	ctor (Advanced)	More Actions
	548L2.0	
info	X.504 Certificate	
Configuration	Standard Strength Certificate (2018-bit)	
Parametera	Charge View Details	
Dutan	SAM, Signature Algorithm	
550	301 *	
	THEFT	
Access	https://app.com/apr//app.com/app///app.com/app///app.com/app//app//app.com/app//app.com/app.com/app//app.com/app//app.com/app//app.com/app//app.com/app//app.com/app//app.com/app//app.com/app//app.com/app//app.com/app.com/app//app.com/app//app.com/app.	
Users		
Privileges	SAML 2.0 Engening (HTTP) https://stanfilows-descenispin.com/stantil/https:com/stantil/https:com/stantil/Stifian-d7d1-d102-add5-d8/stifee	
Setup	under Schneuzons eine vonnehle rouszante mund unde bonzante zuzulten, et al. etzes einer seine Te-	
	SLO Endpoint (HTTP)	
	https://spaddows-des.com/agis.com/trast/sam8/http-redirect/sis/1208217	
	Login Hint	
	Enable login hirs	

saml2.security.signature_algorithm=http://www.w3.org/2001/04/xmldsig-more#rsa-sha1

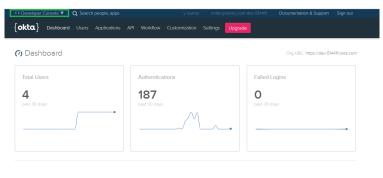
Note: Ma		ector (Advanced)	As the Standard University of Standard Univer	ame or your ip		
	infa Configuration	SAAK.2.0 8.507 CentRoste 5 Sanded Strength CentRoste (2048-bit)				
	Parameters Pales	Change View Octals SAMI, Signature Algorithm SHM-1 *				
SAML of		hover UR. https://pp.onelogit.com/saml/metadata/95d15a143514163-a3254 5446.2.1 (https://pp.onelogit.com/saml/metadata/95d15a143514163-a3254	BidzeeBhild B			
	Phyloges Detup	tespic, reparkflower des anviogie, conzesser vanzez, tespi-pozzana di 5d tu 56.4. Despore (x177) Tespic, reparkflower des anviogie, conzesser vanzez, tespi-rederect valur 133		F	Below are steps to se	tup SAML 2.0 okta
					Create an account at	-
			Enter you	r sign-in URL		
			mycompany	• okta.com		
				Next		



- 2. SignIn into okta
- 3. After login go to home and Click on Admin



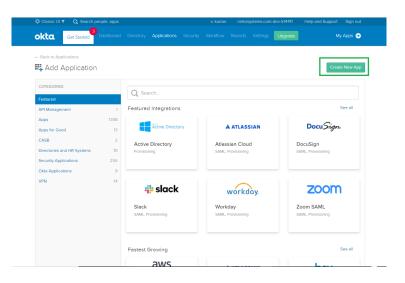
4. Click on Developer Console

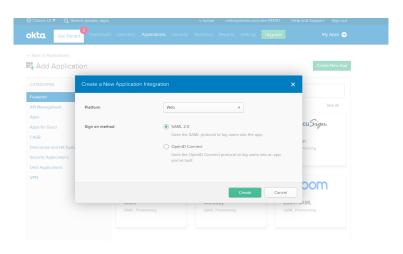


- 5. Add an app:
- 6. Create New App:
- 7. Select SAML 2.0
- 8. Input app name and click next:
- 9. Configure the newly created app and add below information
- 10. Click finish

Note: Make sure to change localhost to your domain name or your ip

Classic UI V Q Search people, apps				ystems.com			
Get Started Directory Applications	Security V	/orkflow	Reports	Settings	Upgrade	My Apps	¢
🕫 Dashboard							
Status				۲	Shortcuts	Applications	
	People				III Assi	gn Applications	
	Search Peo	ople			Add	People vate People	
No notifications to view!	Applications				💄 Dea	ctivate People	
	Search Apr				C Reset Passwords		
						IthInsight	
					Reports		
					Okta Usa	sge	
Usage - Last 30 Days				0		ion Usage	
						us Activity Assignments	
						sword Health	
					SMS Usa MFA Usa		
					System L		
						apable Apps	





General Settings				
App name	FIRE-INSIGHT			
App logo (optional) 🔞	Ø			
			Browse	
	Upload Logo			
	Requirements			
	 Must be PN0 Less than 1W 			
	For Best Results	, use a PNG image with		
	 Minimum 42 Landscape o Transparent 			
App visibility	Do not displa	y application icon to users		
	Do not displa	y application icon in the Okta Mobile app		

https://localhost:8443/saml/SSO
https://localhost:8443/saml/SSO
https://localhost:8443/saml/SSO
https://localhost:8443/sparkflow/saml/metadata
Unspecified
Signed
Signed
RSA_SHA256
SHA256
Unencrypted
Disabled
PasswordProtectedTransport
Yes
http://www.okta.com/\${org.externalKey}

General Sett	Ings	Configure SAML		3 Feedback	
General Setti		Conligure same		Peedback	
3 Help Okta Support under	stand how you confi	gured this application			
				Why are you asking	me this?
				This form provides Ok	
Are you a customer or partner	· · ·	Okta customer adding an internal app		background informatie Thank you for your he	
	🔘 l'm a s	oftware vendor. I'd like to integrate my app w	vith Okta	Thank you for your ne	ip-we appreciate i
The optional question	s helow assist Okta Sun	port in understanding your app integration			
The optional question	s below assist Okta Sup	port in understanding your app integration.			
The optional question		port in understanding your app integration.			

Fire Insights SAML Okta Configuration

Fire Insights can be configured with SAML 2.0 Okta as below.

Go to folder conf/okta.saml. properties file:

Add below information from newly created application in oneLogin:

1. Enable sso in sparkflows:

sparkflows.sp.sso.enable=true

2. Create user locally in application if user doesn't exist in Fire Insights, otherwise app will show page 'User not found':

```
sparkflows.

→sp.auto.user.create=true
```

3. Copy Okta config info

```
# Identifier of the SP entity_

→ (must be a URI) Audience URI

saml2.sp.entityid=https:/

→/localhost:8443/

→ sparkflow/saml/metadata

# Algorithm that the toolkit_

→ will use on signing process

saml2.security.

→ signature_algorithm=http:/

→/www.w3.org/2001/

→ 04/xmldsig-more#rsa-sha256
```

4. Right click on identity provider metadata and select Copy link address

SAML Settings	E
GENERAL	
Single Sign On URL	https://localhost:8443/saml/SSO
Recipient URL	https://localhost:8443/saml/SSO
Destination URL	https://iocalhost:8443/sami/SSO saml2.sp.entity
Audience Restriction	https://localhost:8443/sparkflow/saml/metadata
Default Relay State	
Name ID Format	Unspecified
Response	Signed
Assertion Signature	Signed saml2.security.signature_algorithm
Signature Algorithm	RSA_SHA256
Digest Algorithm	SHA256
Assertion Encryption	Unencrypted
SAML Single Logout	Disabled

5. Capture Issuer url

Note: Make sure to change localhost to your domain name or your ip

\leftarrow Back to Applications	
FIRE-INSIGHT	
Active View Logs	
General Sign On Mobile Import Assignments	
Settings Edi	it
on methods require additional configuration in the 3rd party application. Application username is determined by the user profile mapping. Configure profile mapping SAML 2.0	
Default Relay State	
SAML 2.0 is not configured until you complete the setup instructions. View Setup Instructions	
Identity Provider metadata s available if this application supports cynamic configuration.	

Settings Edit	
SIGN ON METHODS The sign-on method determines how a user signs into and manages their credentials for an application. Some sign- on methods require additional configuration in the 3rd party application. Application username is determined by the user profile mapping. Configure profile mapping	
SAML 2.0 Default Relay State	
SAML 2.0 is not configured until you complete the setup instructions. View Setup Instructions Identity Provider metadata is available if this application supports dynamic configuration.	
on methods require additional configuration in the 3rd party application. Application username is determined by the user profile mapping. Configure profile mapping SAML 2.0 Default Relay State SAML 2.0 is not configured until you complete the setup instructions. View Setup Instructions	

okt	a	
	Ho	w to Configure SAML 2.0 for FIRE-INSIGHT Application
	The	following is needed to configure FIRE-INSIGHT
	0	Identity Provider Single Sign-On URL:
		https://dev-514411.okta.com/app/dev-514411_fireinsight_1/exkds69HBx1g2HeHZ357/sso/san1
	0	Identity Provider Issuer:
		http://www.okta.com/exkMz8GRBxtjg2HeHZ357

CHAPTER 5

Security

5.1 Security

5.1.1 User Group Role Permission

Fire Insights supports Users, Groups, Roles, Permissions. A User can belong to multiple groups and have multiple roles.

Each role can have multiple permissions.

Groups

There can be multiple groups in Fire Insights.

GROUPS LIS	т				
				Search	Q ADD GROUP
ID.	NAME	USERS	FIRST NAME	LAST NAME	ACTION
1	DEFAULT	admin test	admin test	admin Test	/ 8
33	GROUP-A	USER-A-1 USER-A-2	USER-A-1 USER-A-2		× 8
34	GROUP-B	USER-8-1 USER-8-2	USER-8-1 USER-8-2		/ 8

Users

Fire Insights supports multiple users. Each user can belong to multiple groups, and also have multiple Roles.

USER	LIST									
						Sec	arch	Q, 400 USER	DIPORTAL USER	• • •
D	FIRST NAME	LAST NAME	USER NAME	E-MAL	NOLE	OROUP	LAST LOGIN	SUPER USER	ACTIVE/INACTIVE	ACTION
1	admin	odmin	admin		ADMIN	DEFAULT	2020/12/29 23:31:17 PDT	true	0	•/
2	test	test	test		ANALYST	DEFAULT	2020/07/14 03:16:30 PDT	false		۰ ا

USER DETAIL				
ld:	2	Roles:		ADMIN
First Name:	fest		ų.	ANALYST
Last Name:	hest .		~	DEFAULT
User Name:	1051			GROUP-A GROUP-B
Email ID:				ACTIVE
			2	SUPERUSER STATUS
				SAVE CINCEL

Permissions

Fire Insights supports the following Permissions. Permissions are associated with Roles.

Title	Description
users.manage	create, modify & disable user
groups.manage	Create, modify & delete the group
roles.manage	Create, modify & delete the roles
projects.manage	Create, modify & delete the projects
configurations.manage	modify diifferent configurations
datasets.view	view dataset in specified project
datasets.modify	modify datasets in specified project
workflows.view	view workflows in specified project
workflows.modify	modify workflows in specified project
workflows.execute	execute workflow in specified project
apps.modify	modify analytics application
apps.execute	execute analytics application
apps.view	view analytics application

Nome	ANALYST		
Permissions	users.monoge	groups.manage	roles.manage projects.manage
	configurations.manage	datasets.view	datasets.modify
	workflows.view	workflows.modify	workflows.axecute
	apps.modify	opps.execute	2 oppsview
	SAVE BACK		

Roles

A user can have multiple Roles. The actions which a user can do depends on the Roles they belong to.

USER ROLES LIST				
			Search	Q ADD FOLE
D.	ROLE	EDIT	DELETE	
1	ADMIN	EDIT	DELETE	
2	ANALYST	EDIT	DELETE	

5.1.2 Sharing Projects

A project can be shared with multiple Groups. A Project is visible only to those users who belong to the groups with whom it has been shared with.

Below, the Project is shared with the DEFAULT group.

The following permissions can be given to a group during sharing of the project.

All users belonging to the group get the associated permissions on the Project.

		Owner: admin	admin		SHARE
	GROUP NAME	WORKFLOW PERMISSION	DATASET PERMISSION	REPORT PERMISSION	ACTIONS
	DEFAULT	READ WRITE EXECUTE	READ WRITE	READ WRITE EXECUTE	/ 8
SHARE PROJECT - DATAS	SCIENCEAPPLICATION				GROUPS: [DEFAULT V] Z Ad
SHARE PROJECT - DATAS			Z wate	Decute	GROUPS: DEFAULT V Z Ad
			🖸 Write 🖬 Write	Decre	GROUPS : DEFAULT V Z AS
Workflow	Read			🖬 Decire	GROUPS: [DEFAULT V 4 Ad

5.1.3 Databricks Security

Users in Fire Insights access Databricks via Databricks Tokens.

Whenever users interact with Databricks in Fire Insights, they have the access which is assigned to the token in Databricks.

Below diagrams show the integration of Fire Insights with Databricks.



Viewing DB/Tables

In Fire Insights users can view the databases and tables. They are accessed via JDBC from Databricks cluster using the token.

The same applies if users chose to execute a query to view a few records from the table.

Executing Workflows

When users execute workflows in Fire Insights, they are submitted to the Databricks cluster view the REST API using the Databricks token. These jobs post back messages to Fire Insights. They use a token generated specifically for the job to post back the messages.

Databricks Connections

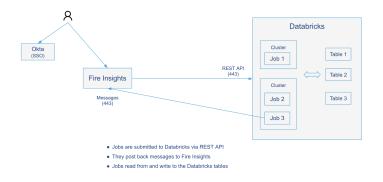
The Databricks cluster details and token are specified in a Connection. The user uses the connections when talking to Databricks.

Connections can be at the global level or at the Project level. Global level connections are created by the admin user. Project level connections are created by the Project users.

Fire Insights would also support defining Group level connections.

5.1.4 Admin user

Fire Insights support variety of permissions for Roles. Each user can be assigned one more more Roles.



DATABRICKS DB					Connection Databrida_cluster Type-databrida	8
SCHEMAS	o	* SQL Editor			Limit to 100 rows 💌 🕴	H
Search	۹	1 SELECT * FRO	H cars_csv LINIT 10			
▼ effectual arg_inflation bila_stanta_comple_ buildingconserts_bi- comp_cov cleaned_arg_clopens_ cleaned_arg_clopens_ cleaned_arg_clopens_ cleaned_arg_clopens_ cleaned_arg_clopens_ cleaned_arg_clopens_ cleaned_comp_	yrogi _wih_in _wih_in					
price2 price_elosticity product_csv		el .	62	a	ç4	
CHEMA INFO			0.0	2.3	3.0	
Schema: default		2	1.0	3.0	2.0	
ablec coracav		3	0.0	13	1.0	
Columns		-4	0.0	4.1	5.0	
		5	0.0	3.1	6.0	

Add Connection

CONNECTION TYPE @*	Databricks ~
CONNECTION NAME @ *	Databricks Connection
TOKEN 🚱*	
TITLE @*	Datbricks
	Datbricks
URL 🕢*	jdbc:spark://dbc-d5b87134-8f82.cloud.databricks.com:443/default;transportMode=http; //
	TEST CONNECTION SAVE CANCEL

Permissions supported by Fire Insights

Below are the permissions supported by Fire Insights.

Nome	ADM	N						
Permissions	4	users.manage	<i>~</i>	groups.manage	4	roles.manage	~	projects.manage
	1	configurations.manage	1	connections manage	~	datasets.view	~	datasets.modify
	\sim	worldlows.view	~	workflows.modify	~	workflows.execute		
	~	apps.modify	~	apps.execute	~	appsview		
	SAVE	EMCK						

Title	Description
users.manage	create, modify & disable user
groups.manage	Create, modify & delete the group
roles.manage	Create, modify & delete the roles
projects.manage	Create, modify & delete the projects
configurations.manage	modify diiferent configurations
connections.manage	add & modify diifferent connections
datasets.view	view dataset in specified project
datasets.modify	modify datasets in specified project
workflows.view	view workflows in specified project
workflows.modify	modify workflows in specified project
workflows.execute	execute workflow in specified project
apps.modify	modify analytics application
apps.execute	execute analytics application
apps.view	view analytics application

Permissions for Admin User

In Fire Insights generally the below permissions are associated with Admin features

- users.manage
- groups.manage
- roles.manage
- configurations.manage

An admin user in Fire Insights is one who has users.manage permission.

Admin User Rights

The Admin user gets the following rights.

Operating Fire Insights

In Fire Insights an admin user can do the following administration tasks:

- Configure Fire Insights
- Run Diagnostics
- Manage Users, Groups, Permissions
- Load Sample Projects

- View Server Logs
- Cleanup Data

Projects/Data etc

As regards to Projects, the Admin user can do the following:

- View all the Projects
- View the executions of all the workflows
- View the executions of all the Analytical Apps
- Onboarding Analytics Apps for a Customer
 - Who creates that Project which will hold the Analytics App? Admin user. Now the admin user becomes the owner of that Project and be able to see everything.
 - Who shares that project with the Group of the Customer? Admin user.

Deleting Users/Groups

In Fire Insights, users and groups cannot be deleted. Users can be made inactive.

Superuser

A user in Fire Insights can be marked to be a super-user. A super-user has all the same rights as the admin user.

Details on the Admin user rights

Diagnostics

The admin user can view detailed informations about Machine environments.

DIAGNOSTICS			FUN DUAGNOSTICS
DIAGNOSTICS INFO			
GENERAL			
	JAN HEDON: FTHOM HEDON: FORT MURLED IN COMPACTATION MESAGE FROM SPARIOS MURLEO OF pay sendialization MURLEO of pay sendial	14.0,20 Parten 22 Parten 23 Parten 24 Parten 25 Parten 2	
POSTBACK URL			
	POSTEACE URL : MACHINE IP ADDRESSES : MACHINE HOSTNAME :	https://15.222.164.26.6443/messageFromSparkjab 25.222.264.28 build-machine	

Usage Statistics

The admin user can view Total Users, Groups, Roles, Projects, Workflows & Workflows Executions

USAGE STATISTICS	
1.mT	COUNT
Total Users :	6
Total Groups :	3
Total Projects :	27
Total Workflows :	213
Total Workflows Executions :	9

Runtime Statistics

The admin user can view Total Logged In Users, Total Fire Spark Processes, Total Fire Pyspark Processes & Total Running Jobs

Total Logged in User	Total Fire Spark Pro	1000	Total Fire Pyspark Process	Total Running Jobs
1	0		0	4
ogged In Users Information				

Sample Projects

The admin user can RELOAD SAMPLE PROJECTS, as by default Fire Insights comes with sample projects containing different types of workflows & datasets

Fin	e Insight	s Q] data -		Reload Sample Projec	ts	· · · ·	0	- B PROCESSORS -	o;- ⊕- ≣-
PRO	OJECTS			Reloading of sample projects will	delete all the changes in existing	sample projects if there is any.		ATE IMPORT EXPORT	RELOAD SAMPLE PROJECTS
0	ıD	USER NAME	NAME	SAMPLE DATA PATH :	cation where the sample data is			AST UPDATED	ACTIONS
•	129	admin	ANOMALY	 SAMPLE DATA PATH Is the to Leave it empty to accept the Some examples of SAMPLE I 	default location.	odolid.		021/01/22 13:23:46 IST	18
٥	130	admin	Analytics		running on a hadoop cluster and	you upload the sample data to the	/tmp HDPS location so that	021/01/22 13:23:46 15T	/8
۰	131	admin	Connector	 s3a://bucket_name/s 	ampledata : în cose you are runn	ning on AWS and the sample data h	as been uploaded to \$3	021/01/22 13:23:46 IST	/8
	132	admin	CreditCore				OK CANCEL	021/01/22 13:23:46 IST	/8
	133	odmin	Datalingin				or owner	021/01/22 13:23:46 IST	18
	13-4	admin	DataPrepar	ation DataPreparat	ion Application	Yes	021/01/22 13:23:46 IST	2021/01/22 13:23:46 IST	/8

Global Connections

The admin user can Add Connections which everyone can use and also connections at the Group Level.

Server Logs

The admin user can view Fire Server Logs, Fire Logs, Fire Exception Logs & Fire Pyspark Logs

Cleanup Data

The admin user can Delete old workflow executions for cleaning the DB which is Older than Last 7 days, Older than Last 30 days, Older than Last 90 days & Delete All Executions

	NS						Add Connection For All Add Connection For All
D	USER ID	USER NAME	CONNECTION NAME	URL	TYPE	STATUS	DATE ORATED
SERVER LOG	ŝ				Type your text	hare	Q REFRE
SERVER LOG	\$				Type your had	hare	Q BURE
SERVER LOG	iS Fire Loga – Fire Exception L	oga Fire Pyspork.Loga			Type your fast	here	Q BUTBE
Fire Server Logs	Fire Logs Fire Exception L				Type your text	here	Q BOTHE
Fire Server Logs 21/01/22 11:05:29 19 2021-01-22 11:05:29	Fire Logs Fire Exception L FO UserURI: Logged In user : USEI 1269 INFO 1837 [atp/00990233-	R-B-1 -211) fireui services.ProjectServiceIm			Type your helt	hare	Q REFE
Fire Server Logs 21/01/22 II:05:29 II 2021-01-22 II:05:29 2021-01-22 II:05:29 21/01/22 II:05:29 II	Fire Logs Fire Exception L IFO UserUR: Logged In user 1 USEI 1269 INFO 1837 [qtp108962313 1269 INFO 1837	R-B-1 -211] Fireui.services.ProjectServiceIm -211] Fireui.aop.LoggingAspect : Exec -B-1	cution time of ProjectController.getAllc		Tiple your helt	hana	Q area
Fire Server Logs 21/01/22 11:05:29 II 2021-01-22 11:05:29 2021-01-22 11:05:29 II 2021-01-22 11:05:29 II 2021-01-22 11:05:29 II	Fire Logs Fire Exception L 4FO User/UR: Logged in user : USEI 1289 INFO 1837	R-B-1 -211) fireui.services.ProjectServiceIm -213) fireui.services.gingAspect : Exec -8-1 -223) fireui.sop.LoggingAspect : Exe -8-1		latasets :: 1 ms	Type your helt	hm	Q BUR
Fire Server Logs 20/0/22 10.05/29 II 2021-01-22 11:06/29 2021-01-22 11:06/29 2021-01-22 11:06/29 II 2021-01-22 11:06/29 20/01/22 11:06/29 2021-01-22 11:06/29 2021-01-22 11:06/29	Fire Loga Fire Exception L FO Used/H: Logadd In user: USE USE L26 (BFO 1057 (BE0068233) EV Deb068233 FO Used/H: Logadd In user: USE USE (BFO 1057 (BE0068233) FO Used/H: Logadd In user: USE USE (BFO 1057 (BE0068233) FO Used/H: Logadd In user: USE USE (BFO 1057 (BE0068233) FO Used/H: Logadd In user: USE USE (BFO 1057 (BE0068233) FO Used/H: Logadd In user: USE USE (BFO 1057 (BE0068233) FO Used/H: Logadd In user: USE USE (BE006823) FO Used/H: Logadd In user: USE USE (BFO 1057	8-8-1 2011 finalisanicas.ProjectSanicalm 2011 finalicap.LoggingAspect 1 Exec -8-1 2021 finalicap.LoggingAspect 1 Exe -8-2321 finalicap.LoggingAspect 1 Exe -8-1 2021 finalicap.LoggingAspect 1 Exe -8-1 2021 finalicantrollars.UsanAminCo	sation time of ProjectController.getAlls cution time of DataSetController.getD cution time of WorkflowController.get ntroller : Current user : USER-B-1	latasets :: 1 ms WorkFlows :: 2 ms	Type year heat	Nrs	Q RIVE
Fire Server Logs 20/0/22 th:05:29 if 2021-01-22 th:05:29 2021-01-22 th:05:29 2021-01-22 th:05:29 if 2021-01-22 th:05:29 if	Fire Loga Fire Exception L FO Usechil: Logach In user: USE USE USE TO Exception L USE USE TO EXC = DepodeB233. Expediate State EO Usechil: Logach In user: USE Expediate State	e.B1 2011 (Heau Services: Project Service! Hr 2011 (Heau Service): Exect 8-B-1 2021 (Hreau LoggingAspect : Exec 8-B-1 2021 (Hreau CopyLoggingAspect : Exe 8-B-1 2011 (Hreau Controllers: UserAdminicO- 2011 (Hreau Controllers: UserAdminicO- 2015 (Heau Controlle	ution time of ProjectController.getAllc cution time of DataSetController.getD cution time of WorkflowController.get troller : Current user : USER-B-1 cution time of UserAdminController.ge	latasets :: 1 ms WorkFlows :: 2 ms #CurrentUser :: 1 ms	Type year hed	hars_	•
Fire Server Logs	Pre Log Pre Exception L 101 User101 Economic regioner 201 Us	P.B1 201) finesi.sen-ices.ProjectSen-iceIm 201) finesi.sep.loggingAspect : Exec 4-21 2021 finui.op.loggingAspect : Exe 3-21 2021 finui.op.loggingAspect : Exe 8-8-1 2011 finui.op.loggingAspect : Exe 8-8-1	ution time of ProjectController.getAll. cution time of DataSetController.get cution time of WorkflowController.get ntroller : Current user : USER-B-I cution time of WarkflowExecutionsCo cution time of WarkflowExecutionsCo	latasets :: 1 ms WorkFlows :: 2 ms #CurrentUser :: 1 ms	Typer your heat	hre	Q Refe

CLEANUP DATA

Workflow Executions		
Total number of Workflow Executions: 9 Total number of Workflow Executions Result: 157		
Delete old workflow executions for cleaning the DB.	DELETE WORKFLOW EXECUT	IONS -
	Older than 7 days Older than 30 days Older than 90 days Delete All Executions	

CHAPTER 6

Operating Fire Insights

6.1 Operating Guide

6.1.1 Logs in Fire Insights

In Fire Insights there are 2 processes which run:

- fire server
- fire engine

Logs for Fire Web Server

The logs for Fire Web Server go into fireserver.log. The logging level is determined by the properties file conf/log4j.properties.

Example log4j.properties

How to change the various logging levels

Logs for Fire Engine

The logs for Fire Engine go into fire.log.

6.1.2 Installing JDBC Drivers for Workflows

Fire has JDBC Processors for reading from JDBC sources or writing to JDBC sinks.

In order to connect to a JDBC source like Oracle/DB2 etc. the JDBC driver needs to be installed in Fire.

Below are the steps for installing the JDBC driver into Fire:

- Download the JDBC jar file
- Copy it into 'fire-user-lib' directory under the Fire installation
- Restart fire

Download the JDBC jar file

Download the JDBC jar file for the Database you are looking to connect to.

Copy it into fire-user-lib

Under the Fire installation directory, there is fire-user-lib directory. Copy the downloaded JDBC jar file into it.

Stop Fire Processes

Stop the running Fire processes with ./run-fire.sh stop They will be restarted automatically.

Running Workflows depending on the jars added

When running workflows which depend on the jar file, select the checkbox for that jar file in the Workflow Execution Page.

Downloading the JDBC jar files

MySQL

- MySQL connector can be downloaded from : https://dev.mysql.com/downloads/connector/j/
- After downloading untar it with : tar xvf mysql-connector-java-5.1.46.tar.gz
- After untaring the jdbc jar file is available in the directory
- Use the jar file (mysql-connector-java-5.1.46.jar) for installation in Fire

PostgreSQL

• PostgresSQL JDBC drivers can be downloaded from : https://jdbc.postgresql.org/download.html

Oracle

• Oracle JDBC drivers can be downloaded from : https://www.oracle.com/technetwork/database/features/jdbc/ jdbc-drivers-12c-download-1958347.html

JDBC Drivers

When using the JDBC processors, the following can be used for the JDBC Driver. Below are the JDBC URL's for some databases:

- MySQL : com.mysql.jdbc.Driver
- PostgreSQL : org.postgresql.Driver
- Oracle : oracle.jdbc.driver.OracleDriver

Example JDBC URL

Below are some example JDBC URL for reading from Relational sources when using the JDBC Processors:

- MySQL : jdbc:mysql://localhost:3306/mydb
- PostgreSQL : jdbc:postgresql://localhost:5432/mydb

6.1.3 Installing JDBC Drivers for Interactive Dashboard

Interactive Dashboard work with JDBC sources. The appropriate JDBC jars have to be installed.

Below are the steps for installing the JDBC driver for Interactive Dashboards:

- Download the JDBC jar file
- Copy it into 'fire-server-lib' directory under the Fire installation
- Restart fire-server

Download the JDBC jar file

Download the JDBC jar file for the Database you are looking to connect to.

Copy it into fire-server-lib

Under the Fire installation directory, there is fire-server-lib directory. Copy the downloaded JDBC jar file into it.

Restart Fire Server

Restart Fire with ./run-fire-server.sh restart Fire does not need to be restarted.

Downloading MySQL Connector

- MySQL connector can be downloaded from : https://dev.mysql.com/downloads/connector/j/
- After downloading untar it with : tar xvf mysql-connector-java-5.1.46.tar.gz
- After untaring the jdbc jar file is available in the directory
- Use the jar file (mysql-connector-java-5.1.46.jar) for installation in Fire

6.1.4 Running Tesseract in Fire

In order to run Tesseract, perform the below installation steps:

Download & Install the Tesseract Language Data files

- Download and Install the tesseract language data files on each of the worker nodes of the cluster
- Install them in the same directory on each of the worker nodes
 - git clone https://github.com/tesseract-ocr/tessdata.git
- Make sure that the tessdata directory is accessible to all the users.

Set TESSDATA_PREFIX as an Environment Variable and restart the Sparkflows server

- Point the environment variable TESSDATA_PREFIX to the tessdata directory
 - export TESSDATA_PREFIX=/home/centos/tessdata
- · Restart the sparkflows server
- If the above is not done correctly, then the Sparkflows server would exit when any OCR node is run

Include TESSDATA_PREFIX in spark configs when submitting the job

Include the following in spark configs when running workflows containing the OCR node:

- --conf spark.executorEnv.TESSDATA_PREFIX=/home/centos/tessdata
- where the tesseract language data files are in /home/centos/tessdata directory on each of the worker nodes

Error if TESSDATA_PREFIX is not set correctly

If TESSDATA_PREFIX is not set, the spark program would run into the error below.

- Error opening data file /Users/saudet/projects/bytedeco/javacpp-presets/tesseract/cppbuild/macosx-x86_64/share/tessdata/eng.traineddata
- Please make sure the TESSDATA_PREFIX environment variable is set to the parent directory of your "tessdata" directory.
- · Failed loading language 'eng'
- Tesseract couldn't load any languages!

6.1.5 Running Apache OpenNLP Model Jars in Fire Insights

When running locally

- Create a directory called opennlp-models-1.5 on the local file system
- Download the Apache OpenNLP model jar from : http://opennlp.sourceforge.net/models-1.5/
 - eg: wget http://opennlp.sourceforge.net/models-1.5/en-ner-person.bin
- Copy the Apache OpenNLP model jar into the opennlp-models-1.5 directory created

When running on a Spark cluster

- Copy the model file onto HDFS into a directory called opennlp-models-1.5
- For example /user/centos/opennlp-models-1.5/en-ner-person.bin
- The model file should be accessible by all the users who would use it

OpenNLPNameFinder 😯

Schema :					
Column Name		lines	lines		
Column Type		string			
Model :	/user/centos/opennlp-mo	dels-1.5/en-ner-person.bin			
Input Text Column :	lines : string		\$		
Output Column :	ner				
				ОК	Cancel

6.1.6 Installing/Using OpenNLP model jars

When running locally

- Create a directory called opennlp-models-1.5 on the local file system
- Download the OpenNLP model jar from : http://opennlp.sourceforge.net/models-1.5/
 - eg: wget http://opennlp.sourceforge.net/models-1.5/en-ner-person.bin
- Copy the OpenNLP model jar into the opennlp-models-1.5 directory created

When running on a Spark cluster

- Copy the model file onto HDFS into a directory called opennlp-models-1.5
- For example /user/centos/opennlp-models-1.5/en-ner-person.bin
- The model file should be accessible by all the users who would use it

Using OpenNLP model jars

• Specify the path of the jar file in the dialog box of the Open NLP nodes in the workflow

• For example for the OpenNLPNameFinder node the path can be : /user/centos/opennlp-models-1.5/en-ner-person.bin

OpenNLPNameFinder 😧					
Schema :					
Column Name	lines				
Column Type	string				
Model : 😡 Input Text Column :	/user/centos/opennlp-models-1.5/en-ner-person.bin				
Output Column :	ner				
	OK Cancel				

6.1.7 Using Juypter

Jupyter is extensively used by Data Scientists.

Overview

Fire can be used to easily create a downsampled dataset. Fire provides a sample processor for it.

Once the dataset size has been reduced, Data Scientists can model with it in Jupyter.

Once the modeling process is complete, the algorithm can be run on the full data in Fire.

6.1.8 Maintenance Tasks

Cleaning H2 DB

Fire Insights by default uses the H2 embedded database.

It is important to keep the size of the database in control. All the Fire Insights tables are relatively small except those which store the result of workflow execution.

Cleaning Old Workflow Executions

It is important to regularly delete the old workflow executions in order to keep the size of the H2 DB in control.

• Go to the Administration/Cleanup Data

• Click on Delete old Workflow Executions in order to delete the old workflow executions.

Compact H2 DB File

If the H2 DB file size grows too large (> 3GB), then follow the steps below for compacting it.

By default H2 DB file is in the home folder of the user running Fire Insights. It is named as firedb.mv.db

- Store Fire Insights
- Make a copy of firedb.mv.db file to be safe
- · Use the commands below for compacting it

java -cp ~/fire-3.1.0/db/h2/h2-1.4.199.jar org.h2.tools.Shell URL: jdbc:h2:./firedb Driver : org.h2.Driver User : fire Password : fire

SHUTDOWN COMPACT

Deleting old files

Regularly delete the following folders:

- /tmp/fire/workflowlogs
- /tmp/fire/workflows

6.1.9 Installing MySQL

This document captures the details for installing MySQL on Centos7

Steps for installing MySQL on Centos7

Check your hostname

To check your hostname run:

hostname hostname -f

• Update your system

Run below command to update your system:

sudo yum update

• Install wget if its not on your system

You will need wget to complete this guide. It can be installed as follows:

sudo yum install wget

Install MySQL

MySQL must be installed from the community repository.

· Download and add the repository

Download and add the repository, then update:

```
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm
sudo yum update
```

· Install MySQL as usual and start the service

Install MySQL as usual and start the service. During installation, you will be asked if you want to accept the results from the .rpm file's GPG verification. If no error or mismatch occurs, enter y:

sudo yum install mysql-server
sudo systemctl start mysqld

Harden MySQL Server

Harden security Concern

Run the mysql_secure_installation script to address several security concerns in a default MySQL installation:

sudo mysql_secure_installation

• To check Existing password generated

To check Existing password generated:

```
sudo grep 'temporary password' /var/log/mysqld.log
```

• You can also create new password while installing too.

Using MySQL

The standard tool for interacting with MySQL is the mysql client which installs with the mysql-server package. The MySQL client is used through a terminal

• Root Login

To log in to MySQL as the root user:

```
mysql -u root -p
```

• When prompted, enter the root password you assigned when the mysql_secure_installation script was run

You'll then be presented with a welcome header and the MySQL prompt as shown below:

mysql>

To Provide access from remote pcs

Inorder to Access MySQL from Remote PC, run below command:

```
CREATE USER 'root'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

NOTES * The Port on which MySQL Running ie 3306, should be accessible from target machine.

Create a New MySQL User and Database

In the example below, testdb is the name of the database, testuser is the user, and password is the user's password:

```
create database testdb;
create user 'testuser'@'localhost' identified by 'password';
grant all on testdb.* to 'testuser' identified by 'password';
```

Create a Sample Table

Log back in as testuser

Login with testuser:

```
mysql -u testuser -p
```

• Create a sample table

Create a sample table called customers. This creates a table with a customer ID field of the type INT for integer (auto-incremented for new records, used as the primary key), as well as two fields for storing the customer's name:

Reset the MySQL Root Password

If you forget your root MySQL password, it can be reset.

• Stop the current MySQL server instance

Stop the current MySQL server instance, then restart it with an option to not ask for a password:

```
sudo systemctl stop mysqld
sudo mysqld_safe --skip-grant-tables &
```

Reconnect to the MySQL server

Reconnect to the MySQL server with the MySQL root account:

mysql -u root

• Use the following commands to reset root's password

Use the following commands to reset root's password. Replace password with a strong password:

```
use mysql;
update user SET PASSWORD=PASSWORD("password") WHERE USER='root';
flush privileges;
exit
```

• Restart MySQL

Then restart MySQL:

sudo systemctl start mysqld

MySQL JDBC Driver

Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html

Extract the JDBC driver JAR file from the downloaded file. For example:

tar zxvf mysql-connector-java-8.0.11.tar.gz

mysql-connector-java.jar

CHAPTER 7

Quick Start Guide

7.1 Quickstart Guide

The quickstart gets you started with Fire Insights.

Let's get started!

7.1.1 Step 1: Create Project

Before you can start creating a workflow, you will need to create a 'Project'. Project is a bucket where all your artifacts such as datasets, workflows, dashboards etc. related to a project would reside. Projects are equivalent to workspaces in IDEs.

From the landing page of Fire Insights, click on "Create Application" to create a new application.



Specify name and description, and click on "Create/Update" button. The new application is created and it is now ready to use.

7.1.2 Step 2 : Upload Data Files

Every workflow needs data to work on. As a next step, you will upload a CSV file that you want to process in your workflow.

Crea	ite Application	1			×
APPL	ICATION NAME	Nev	v application		
DESC		test			
				CREATE/UPDATE	CANCEL
Applicat	ions				
ID	USER NAME	NAME	DESCRIPTION	DATE CREATED LAST UPDATED	ACTIONS
33	sparkflows	Test_app	Test	06/26/2019 at 9:15:16AM 06/26/2019 a	t 9:15:16AM 🛛 💉 🏦

If you have your data in CSV file, click on "Data Browsers" and select "HDFS". Your home directory will be displayed. Initially, it will be empty as you have not uploaded any file.

HC	DFS										
_	Home / user / sportflows Q Search HOFS Home / user / sportflows Reveal # Revea										
	NAME		PATH	SIZE	¢	LAST UPDATED	OWNER	GROUP	PERMISSIONS	ACTIONS	
•	Airline.cov	8	/user/sparkflows/Airline.csv	278 bytes		05/19/2019 at 4:50PM	sparkflows	sparkflows	rw-rr	t 🕹 🗠 🔍	
	🗅 data	œ	/user/sparkflows/data			06/07/2019 at 11:25AM	sparkflows	sparkflows	rwxr-xr-x	<u>ن</u> ۹	

Click on "Upload File" button. Choose one or more CSV files that you want to upload.

After selecting the files, click "Upload All".

In order to use CSV files in workflow, Fire Insights requires that you wrap them in datasets. In the next step, you will create datasets from the files you have just uploaded.

7.1.3 Step 3 : Create Dataset

Before any data can be used in a workflow, it needs to be wrapped in a dataset. If you uploaded CSV files in the previous step, in this step you will wrap them in a dataset.

The steps involved in creating a dataset are:

- Open the Application where you want to create dataset
- Click on "Datasets" tab
- Click on "Create" and choose "Datasets"
- Select your dataset type and enter the fields in the dialog
- Update the schema of the dataset
- Click "Save"

When you open your application, all existing datasets specific to the application are displayed in the Datsets tab.

Click on "Create" and choose "Dataset" from the dropdown.

In the pop-up choose "CSV" and then click "OK".

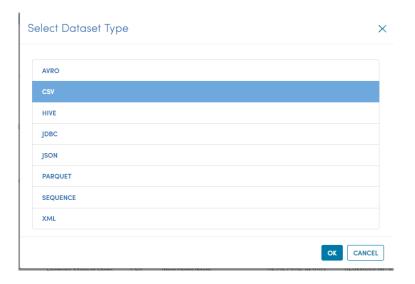
Fill in the required fields as below.

- Name : Name of the new dataset
- Description : Description of the new dataset

PROGRE	SS STATUS	ACTIONS	
ИВ		1	Ŵ
1			P

I	нс	DFS											
	Home / user / sporkflows Q Search HDPS												
		NAME \$		SIZE \$	LAST UPDATED	OWNER	GROUP	PERMISSIONS	ACTIONS				
		Airline.csv	/user/sparkflows/Airline.csv	278 bytes	05/19/2019 at 4:50PM	sparkflows	sparkflows	rw-rr	t 🕹 🖄 🔍				
	0	Clickthru.cov	/user/sparkflows/Clickthru.csv	2989 bytes	07/17/2019 at 4:46PM	sparkflows	sparkflows	rw-rr	t ± @ &				

Datasets	Workf	ows Dashboard	Interacti	ive Dashboard Workfl	low Exec	utions Schedules	Share	Credentials Con	nections						
Datase	əts					Q S	arch Dataset			CREA	(TE -	ſ	EXPOR	T DAT	TASE
- ID	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACT	ION				
97	spar	kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	1	۲	đ	8		&
1	admin	Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn		06/24/2019 at 10:31	PERMISSION_MODIFY		۲	ф	8		&
7	admin	Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans		09/16/2018 at 5:50P	PERMISSION_MODIFY		۲	6	8		&
6	admin	NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY	1	۲	6	8		æ
٦. c	admin.			IntRol Train	C91/	(term (data (intrail		07/12/2016 at 7/22444			~		2		0



- Has Header Row : Indicate whether the dataset has a header row specifying the name of the columns or not
- Delimiter : Indicates the delimiter to be used between the fields in the data
- Path : Path for the location of the file or directory containing the data files for the dataset

DataSet Details - CSV				O HELP
NAME: 😡 *	New housing dataset	CATEGORY: @	csv	
DELIMITER: O		HAS HEADER ROW: O	* TRUE O FALSE	ADD READ OPTIONS
DESCRIPTION: 0				
PATH: @*	/tmp/data/housing.csv			BROWSE
SAMPLE DATA: 0				UPDATE SAMPLE DATA/SCHEMA
SCHEMA: Ø				
NAME O	DATA TYPE 😡	FORMAT O	ML TYPE 😡	

Now click on "Update dataset/schema" to update the schema of the dataset. Sample data for the dataset will be displayed followed by the schema.

In the example below, a dataset is created from a housing.csv file. It is a comma separated file with a header row specifying the names of the various columns.

	<pre>price","lotsize","bedrooms","bathrms","stories","driveway","rec</pre>
	000,5850,3,1,2,"yes","no","yes","no","no",1,"no"
	500,4000,2,1,1,"yes","no","no","no","no",0,"no"
"3",4	500,3060,3,1,1,"yes","no","no","no","no",0,"no"
"4",6	500,6650,3,1,2,"yes","yes","no","no","no",0,"no"
"5",6	000,6360,2,1,1,"yes","no","no","no","no",0,"no"
"6",6	000,4160,3,1,1,"yes","yes","yes","no","yes",0,"no"
"7",6	000,3880,3,2,2,"yes","no","yes","no","no",2,"no"
"8",6	000,4160,3,1,3,"yes","no","no","no","no",0,"no"
"9",8	800,4800,3,1,1,"ves","ves","no","no",0,"no"
"10",	8500,5500,3,2,4,"yes","yes","no","no","yes",1,"no"
	0000,7200,3,2,1,"yes","no","yes","no","yes",3,"no"

If the data file did not have a header row, Fire Insights will give standard column names of "C0, C1" etc.

You can update the column names in the schema based on your data.

d NTEGER)	price (INTEGER)	Iotsize (INTEGER)	bedrooms (INTEGER)	bathrms (INTEGER)	stories (INTEGER)	driveway (STRING)	recroom (STRING)	fullbase (STRING)	gashw (STRING)	airco (STRING)	garagepl (INTEGER)	prefarea (STRING)
	38500	4000	2	1	1	yes	no	no	no	no	0	no
	49500	3060	3	1	1	yes	no	no	no	no	0	no
	60500	6650	3	1	2	yes	yes	no	no	no	0	no
	61000	6360	2	1	1	yes	no	no	no	no	0	no
	66000	4160	3	1	1	yes	yes	yes	no	yes	0	no
	66000	3880	3	2	2	yes	no	yes	no	no	2	no
	69000	4160	3	1	3	yes	no	no	no	no	0	no
	83800	4800	3	1	1	yes	yes	yes	no	no	0	no
	88500	5500	3	2	4	yes	yes	no	no	yes	1	no
	90000	7200	3	2	1	yes	no	yes	no	yes	3	no
	30500	3000	2	1	1	no	no	no	no	no	0	no
CHEMA: 🔞												
IME 😡			D	ATA TYPE O		FORMA	0			ML TYPE	0	
d				INTEGER		• form				NUM	cours	

Now click "Save' to save the new dataset and you are ready to use it in your workflows.

7.1.4 Step 4 : Create Workflow

After you have created the datasets, you can start building workflows to process them.

A typical workflow takes one or more dataset, cleans them and joins them, and creates an enriched dataset. After the enriched dataset is created, you can add additional processors to build machine learning models.

At a high level, creating a workflow involves the following steps:

- Open the Application where you want to create your workflow
- Click "Workflows" tab
- Create empty workflow
- Add processors
- Save workflow

Application

Open the application where you want to create your new workflow.

Applications							CREATE
ID	USER NAME	NAME	DESCRIPTION	DATE CREATED	LAST UPDATED	ACTIONS	
33	sparkflows	Test_app	Test	06/26/2019 at 9:15:16AM	06/26/2019 at 9:15:16AM	e 🗇	A
161	sparkflows	FICO		07/17/2019 at 3:33:51AM	07/17/2019 at 3:33:51AM	e 🗇	
193	sparkflows	New application	test	07/17/2019 at 3:55:55PM	07/17/2019 at 3:55:55PM	D 🗊	
1	admin	Analytics	Analytics Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
2	admin	Connectors	Connectors Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
3	admin	DataPreparation	DataPreparation Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
4	admin	Demo	Demo Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	D 🗊	
5	admin	ETL	ETL Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	D 🗊	
6	admin	FileFormats	FileFormats Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
7	admin	Financials	Financials Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
8	admin	Languages	Languages Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
9	admin	MachineLearning	MachineLearning Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
10	admin	MarketingAnalytics	MarketingAnalytics Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	

Workflows Tab

Click "Workflows" tab to view the list of workflows already in the application. The workflow list will be empty if no workflows have been created earlier.

Create Empty Workflow

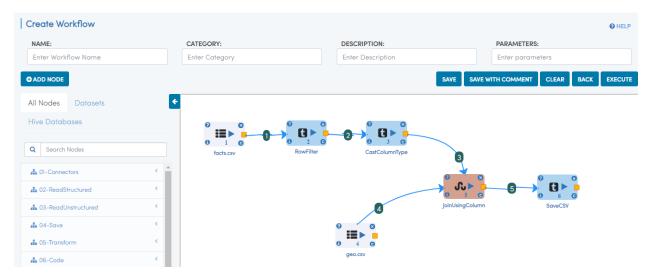
Click "Create" button and choose the type of workflow you want to create. In the "Create Workflow" page, enter a name, category and description of the workflow. Category is used to group various workflows. For instance, if you have several workflows for customer reports, you can group them by specifying "Customer Reports" category.

Click "Save" to save the empty workflow.

Application – ETL								
Datasets Workflows Dash	board Interac	live Dashboard	Workflow Executions	Schedules Share Credentials	Connections			
Workflow List		Q Search		CR	EATE - EXPORT WORKFLOWS	COPY W	VORKFLOWS	
						WORKFL	OW WIZARD	
Filter By Category		USER NAME	WORKFLOW NAME	TYPE CATEGORY	LAST UPDATED	ACTION		
■ Show All Categories	897	sparkflows	ETL Customer	scala ETL	07/18/2019 at 1:17PM	۰ 🇨 ا	> 🕨 🕼	₼ ₼
-	45	admin	Complex Customer ETL	scala ETL	07/17/2019 at 12:03AM	۰ 🌶 ا	 Image: Image: Image:	1
>-	47	admin	Multi Join	scala -	06/24/2019 at 10:31AM	۰ 🇨 ا	> > 🕼	1
> ETL	46	admin	Simple Customer ETL	scala ETL	06/24/2019 at 10:31AM	•	Image:	b

Add Processors

After you have saved the empty workflow, you can start adding processors to process the datasets that you had defined earlier. Click on the processors on the left hand side pane. This will make the processor appear on the workflow canvas. Add other processors, configure and connect them as needed. Two processors can be connected by clicking on the yellow box in the first processor and dragging it to the second processor.



Save Workflow

Once you are satisfied with your workflow, save the workflow by clicking on 'Save' button. Each time the workflow is saved, a new version of workflow is created.

7.1.5 Step 5 : Execute Workflow

After you have created a workflow, it is time to execute it and view the results.

Executing a workflow involves the following steps:

- Go to Application page where you want to execute the workflow
- Click "Workflows" tab
- Click on the play button

- Specify parameter(if any)
- Click on Execute

Application page

Open the application where you have created the workflow to be executed.

Application	S						CREATE
ID	USER NAME	NAME	DESCRIPTION	DATE CREATED	LAST UPDATED	ACTIONS	
33	sparkflows	Test_opp	Test	06/26/2019 at 9:15:16AM	06/26/2019 at 9:15:16AM	e 🗇	A
161	sparkflows	FICO		07/17/2019 at 3:33:51AM	07/17/2019 at 3:33:51AM	e 🗇	
193	sparkflows	New application	test	07/17/2019 at 3:55:55PM	07/17/2019 at 3:55:55PM	e 🖉	
1	admin	Analytics	Analytics Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
2	admin	Connectors	Connectors Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	D 🗇	
3	admin	DataPreparation	DataPreparation Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🖉	
4	admin	Demo	Demo Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
5	admin	ETL	ETL Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	_
6	admin	FileFormats	FileFormats Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
7	admin	Financials	Financials Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
8	admin	Languages	Languages Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
9	admin	MachineLearning	MachineLearning Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🗇	
10	admin	MarketingAnalytics	MarketingAnalytics Application	06/24/2019 at 10:31:27AM	06/24/2019 at 10:31:27AM	e 🖉	

Workflows

Click "Workflows" tab to view the list of workflows in the application.

Application – ETL					
Datasets Workflows Dashboo	ard Interactive Dashboard	Workflow Executions	Schedules Share Credentials	Connections	
Workflow List	Q Search		CREA		
					WORKFLOW WIZARD
Filter By Category	ID USER NAME	WORKFLOW NAME	TYPE CATEGORY	LAST UPDATED	ACTION
Show All Categories	897 sparkflows	ETL Customer	scala ETL	07/18/2019 at 1:17PM	• / > > • • • •
	45 admin	Complex Customer ETL	scala ETL	07/17/2019 at 12:03AM	• / • • • d d d d
>-	47 admin	Multi Join	scala -	06/24/2019 at 10:31AM	• / • • • d d d i
> ETL	46 admin	Simple Customer ETL	scala ETL	06/24/2019 at 10:31AM	• / • • • 4 4

Click on the Play Button

Against each workflow there are a list of icons under "Actions" column for performing various actions on a specific workflow.

Click "Play" icon under "actions" column to execute the workflow.

Execute workflow page

Specify any paramters for your workflow.

٩						SWORKFLOW EXECUTIONS -				
T	Execute Wo	rkflow								
B/	CK EDIT									
	Workflow Id: 🛛		32						EXECUTE	
	Workflow Name: 🕻)	ETL Custo	omer						
	Spark-submit-con	f: 0								
			eg:exe	cutor-memory 2gni	um-executors 5e	executor-cores 2driver-memory	2g			
	Choose jar files: 6									
	Workflow F	tesults								
	Workflow									
	1		32							
		DESCRIPTION:	ETL Workflow							
		NODES:	Nodes							
			ReadCSV							
			ReadCSV							
			ReadCSV							
			Reducav							

Execute Workflow

Once you have specified the parameters, click on "Execute" button. The results of execution are streamed back into your browser.

7.1.6 Step 6 : Create Dashboard

Dashboards allow you to display the output of multiple workflows in one place.

The steps involved in creating a dashboard are:

- Go to Dashboard tab
- Click on Create New Dashboard
- Drag and drop selected Nodes from the workflows into the Dashboard canvas
- Save the Dashboard

Dashboards

Selecting Dashboard tab will take to Dashboard page.

Applic	ation – Marke	etingAnalytics								
Datase	ts Workflows	Dashboard In	teractive Dashboard	Workflow Executions	Schedule	s Share	Credentials	Connections		
Das	nboards				Q Sec	arch Dashboard			CREA	TE NEW DASHBOARD
ID	DASHBOARD NAME		CATEGORY	DESCRIPTION		LAST UPDATED		VIEW	EDIT	DELETE

Create Dashboard

WORKHOW LIST	DASHBOARD NAME:	Enter Dashboard Name	CATEGORY:	Enter Category	DESCRIPTION:	Enter Description	• HELP
Drag and drop nodes on the dashboard panel	SAVE						
Q Search Workflow							_
🗞 WF: Telco Churn Data Analysis							
StringIndexe VectorAssen							
(Split) RandomFor							
Predict BinaryClass							
(Summary) (CSV							
Correlation Documenta							
GraphValue							
𝗞 WF: Bike Sharing Analysis							
DatasetStru) (Extract							
							+++++

This would open up the Dashboard Designer Page.

Name Dashboard

Give a name to your dashboard. You can also add a description for the new dashboard.

Build Dashboard

On the left hand side of the Dashboard Designer, the list of workflows would show up. With each workflow, the nodes inside the workflow would be displayed.

Nodes inside the workflow can be dragged and dropped onto the dashboard to make them part of the dashboard.

In the dashboard below we have added two nodes to the dashboard.

WF: Telco Churn Data Analysis StringIndex VectorAsser Split Predict BinaryClass Summary CSV Correlation Data Analysis Name: Correlation between two series of doto. Type: transform CSV Correlation Documente GraphValue	Workflow List Drag and drop nodes on the dashboard panel	DASHBOARD NAME: Bike Sharing & Churn SAVE VIEW	CATEGORY: Enter Category	DESCRIPTION: Enter Description	@ HELI
Split RandomFor colculates the correlation Rentals Image: Correlation Predict BinaryClass between two series of data. Image: Correlation Type: transform Correlation Documenta Image: Correlation Image: Correlation Image: Correlation Correlation Documenta Image: Correlation Image: Correlation Image: Correlation GraphValue Image: Correlation Image: Correlation Image: Correlation Image: Correlation	Q Search Workflow & WF: Telco Churn Data Analysis		rkflow : Bike Sharing		^
Summary CSV Correlation Documenta GraphValue Image: Construction of the co	Split RandomFor	calculates the correlation	itals		
	Summary CSV	- dato.			
	GraphValue WF: Bike Sharing Analysis				

Save Dashboard

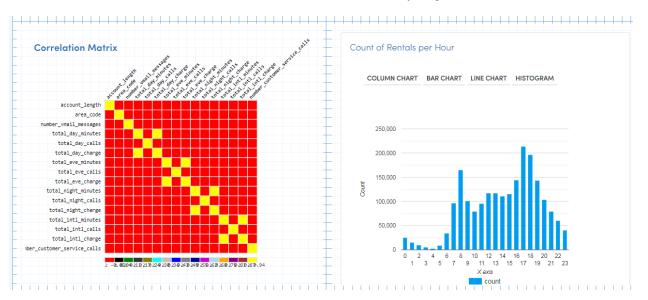
Finally save the dashboard. In order to view the dashboard, click on the 'View' button.

View Dashboard

Click on the 'View' button to view the dashboard.

The dashboard shows the content from the latest execution of the workflow.

If the workflow has never been executed, the dashboard would not show anything.



CHAPTER 8

User Guide

8.1 User Guide

8.1.1 Datasets

Fire Insights allows you to define your DataSets. These DataSets are then used in Workflows as data sources. DataSet sources can be local file system when running in local mode, or HDFS & HIVE when running on a Spark cluster.

Schema

• DataSets have Schema defined for them. This allows Fire Insights to read and create a DataFrame out of it. The DataFrame is then used for transforms, machine learning etc.

File formats

- Sparkflows supports various File formats and is able to infer the schema. These include CSV/TSV, Parquet, Avro, JSON, XML files.
- Sparkflows also supports creating datasets from HIVE tables. This is not necessary as in the Workflows HIVE Processors can be directly connected to specific HIVE tables (instead of creating a Dataset in Fire for them).

Dataset Listing Page

When you open any application, all existing Datasets specific to the application are displayed in the Datasets tab.

Application	- Analytics													
Datasets Wa	orkflows Dashboard	Interaction	ve Dashboard Workfi	ow Execut	ions Schedules	Share	Credentials Con	nections						
Datasets					Q Sec	rch Dataset			CREA	(TE *	E	XPOR	T DAT	ASETS
	ER NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACT	ION				
97 spc	ar kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	1	۲	đ	B (ð (80 18
1 odi	min Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn		06/24/2019 at 10:31	PERMISSION_MODIFY	1	۲	ф	8	ð (86 18
🗌 7 odi	min Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans		09/16/2018 at 5:50P	PERMISSION_MODIFY		۲	ு	8	ð (86 🖻
6 ad	min NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY	1	۲	đ	8	ð (8 0
□ 6 mi	ante Colford Toole		Intibul Toole	cou	Noon felata finten?		07/02/2016		ھ	•	DA I	car I	<u>.</u>	e e

Creating New Datasets

You can define a New Dataset by clicking on the Create Dataset button in the Dataset page. It will bring up the dialog box below. Select the format of the file for which the new Dataset is being created.

AVRO	
csv	
HIVE	
JDBC	
JSON	
PARQUET	
SEQUENCE	
XML	

Entering Field Details

Below are the details of the fields in the Create Dataset page:

- NAME : Name of the New Dataset we are creating.
- **DESCRIPTION** : Description of the New Dataset.
- HAS HEADER ROW : This is used for CSV/TSV files. It indicates whether the dataset has a header row specifying the name of the columns or not.
- **DELIMITER** : Delimiter field is also used for CSV/TSV files. It indicates the delimiter to be used between the fields in the data.
- **PATH** : It defines the location of the file or directory containing the data files for the Dataset. It can either point to a single file, or to a directory containing a set of files. All the files have to have the same schema.

DataSet Details - CSV				© HELP SAVE BACK
NAME: @*	New Csv dataset	CATEGORY: 0		
DELIMITER: Ø		HAS HEADER ROW: Ø	○ TRUE	ADD READ OPTIONS
DESCRIPTION: 0				
PATH: 😡	/tmp/data/housing.csv			BROWSE
SAMPLE DATA: 😧				UPDATE SAMPLE DATA/SCHEMA

Updating the Schema of the Dataset

You can update the Schema of the Dataset by clicking on Update Sample Data/Schema. It would display sample data for the dataset followed by the Schema inferred by Fire Insights.

In this example, the data file did not have a header row. So Fire gave it standard column names of C0, C1 etc. You can update the column names in the schema based on your data.

CO (STRING)	C1 (STRING)	C2 (STRING)	C3 (STRING)	C4 (STRING)	C5 (STRING)	C6 (STRING)	C7 (STRING)	C8 (STRING)	C9 (STRING)	C10 (STRING)	C11 (STRING)	C12 (STRING)		
1	42000	5850	3	1	2	yes	no	yes	no	no	1	no		
2	38500	4000	2	1	1	yes	no	no	no	no	0	no		
3	49500	3060	3	1	1	yes	no	no	no	no	0	no		
4	60500	6650	3	1	2	yes	yes	no	no	no	0	no		
5	61000	6360	2	1	1	yes	no	no	no	no	0	no		
6	66000	4160	3	1	1	yes	yes	yes	no	yes	0	no		
7	66000	3880	3	2	2	yes	no	yes	no	no	2	no		
8	69000	4160	3	1	3	yes	no	no	no	no	0	no		
9	83800	4800	3	1	1	yes	yes	yes	no	no	0	no		
10	88500	5500	3	2	4	yes	yes	no	no	yes	1	no		
n	90000	7200	3	2	1	yes	no	yes	no	yes	3	no		
SCHEMA: O														
NAME 😡			D/	TA TYPE O		FORMA	FORMAT 😧				ML TYPE O			
CO				STRING			format			TEXT				
CI				STRING			format			TEXT				

Saving the New Dataset

Click on the Save button to save the New Dataset created.

8.1.2 Workflows

Creating Workflows

Fire Insights enables users to define end-to-end workflows for data pipelining leveraging pre-packaged nodes for common ETL and Machine Learning models. Workflows are then saved and executed to produce results. Sparkflows provides a a very intuitive and user friendly editor to achieve the same.

Define New Workflow

Click on 'Create New Workflow' for creating a New Workflow, It supports two engines - spark & pyspark. It will open the Workflow Editor where the workflow can be created.

← → C ▲ Not secure 137.117.8	83.79:8080/#	/createWorkflow/scala			er Q 🕁
Fire Insights 🛛 🕲 DATA BROWSERS	 O APPL 	ICATIONS - O SCHEDULED -	SWORKFLOW EXECUTIONS - 2 ADMINISTRATION	N ▼ III PROCESSORS ▼	≡ • o; •
Create Workflow					O HELP
NAME:		CATEGORY:	DESCRIPTION:	PARAMETERS:	
Enter Workflow Name		Enter Category	Enter Description	Enter parameters	
ADD NODE				SAVE SAVE WITH COMMENT CLEAR	R BACK EXECUTE
All Nodes Datasets Hive Data	ibases 🗲				
Q Search Nodes					
A 01-Connectors	< ^				
A 02-ReadStructured	<				
4 03-ReadUnstructured	<				
di 04-Save	<				
👍 05-Transform	<				
dh 06-Code	<				
A 07-JoinUnion	<				
A 08-Visualization	<				
4 09-MachineLearning	<				

Adding New Nodes to the Workflow

• Workflows editor has a list of Nodes menu on the LHS. Clicking on any of the Nodes creates it in the workspace.

Creating Edges

- Nodes can be connected by edges.
- Click on the orange box and drag to the next node to create an edge between them.

Deleting Edges

• Edges can be deleted by double clicking on them.

Saving Workflow

- Give the workflow a name.
- Click on the Save button to create the new workflow.

View Workflows

You can view the workflows by going to the Workflows Page inside specific applications.

Datasets Workflows Da		Interac	tive Dashbor	ard Workflow Executions Sch	hedules	Share	Credentials Connec	tions	
Vorkflow List			Q Sec	arch			CREATE -	EXPORT WORKFLOWS	COPY WORKFLOWS
									WORKFLOW WIZARD
Filter By Category		ID	USER NA	WORKFLOW NAME	TYPE	CATEGORY	LAST UPDATED	ACTION	
Show All Categories	^	588	90	Neeroj - Predict Join & analyse Mo	scala		07/06/2019 at 10:39AM		B B B C
	-11	550	90	Neeroj_Data Contextualization	scola		07/05/2019 at 7:06PM	@ / Þ Þ	B B B 1 0 C
>-	-11	513	90	Contextualizedata- Model Workflo	scala		07/04/2019 at 5:36PM	• / Þ Þ	0 0 0 0 0 0
> Apache Log		481	ge	02 - Data Contextualization	scala		07/04/2019 at 3:03PM	• / > >	
> Clickstream		9	odmin	JetrailDataAnalysis	scola	Distribution	07/04/2019 at 11:25AM	• / > >	0 0 0 0 C
> Distribution	-11	10	odmin	NYC Taxi Average Speed	scala	Taxi Speed	07/04/2019 at 11:24AM	• / > >	
Distribution	-11	11	odmin	Farmers Markets on Geo Map	scolo	Geo	07/04/2019 of 11:22AM	@ / > >	B B B 0 C
Flights Delay		13	odmin	Transaction Data Analytics	scala	Transaction	07/04/2019 at 11:20AM	• / > >	
> Geo		417	90	1 - Team5PnGFullWorkFlow	scala	hackthon	07/02/2019 at 5:06PM	• / > >	
 Contractor 	-11	385	00		scolo		07/01/2019 at 12:34PM		

Executing Workflows

Fire Workflows can be executed in the following ways:

- Interactively within the User Interface
- · Submitting the workflows using spark-submit through the command line
- · Scheduling for execution with your scheduler of choice

Interactively within the User Interface

Workflows can be executed from the browser by going into the Execute page of the workflow.

Execute Workflow			
BACK EDIT			
WORKFLOW ID: O	13	EXECUTE	DEBUG
WORKFLOW NAME: 0	Transaction Data Analytics		
SPARK-SUBMIT-CONF: 0			
	eg:executor-memory 2gexecutor-cores 2driver-memory 2g		
PROGRAM PARAMETERS: 0			
CHOOSE JAR FILES: Ø	MYSQL-CONNECTOR-JAVA-8.0.11 JAR		
EMAIL ON SUCCESS:			
EMAIL ON FAILURE:			
Workflow Result			
Workflow			
DESCRIPTION:	Plots number of Transactions by Brand		
NODES:	NODES		
	DatasetStructured		
	PrintNRows		
	GraphGroupByColumn		
	Notes		
	GraphGroupByColuma		

Executing Workflows with spark-submit

Workflows are saved as text files in JSON format. Workflows can be submitted to be run on the cluster with sparksubmit:

```
spark-submit --class fire.execute.WorkflowExecuteFromFile --master yarn --

deploy-mode client --executor-memory 1G --num-executors 1 --executor-cores_

in fire-core-1.4.2-jar-with-dependencies.jar --postback-url http://

d<machine>:8080/messageFromSparkJob --job-id 1 --workflow-file _

in kmeans.wf
```

In the above:

For providing extra variables to the workflow, the following parameters can be added to spark-submit:

--var name1=value1 --var name2=value2 --var name3=value3

In the workflow, these variables can be used with \$name1 \$name2 Specific nodes make use of the variables by substituting \$name with the value provided for the name.

For running the workflow in debug mode, add the following parameters:

--debug true

Workflow JSON

In Sparkflows, workflows are saved as JSON Strings.

The View JSON Workflow page of the Workflow displays the JSON representations of the workflow.

Scheduling Workflow execution with Scheduler of choice

Since Fire workflows can be submitted with spark-submit, you can use your scheduler of choice for scheduling the execution of the workflows.

· Click on Schedule Button of Workflow we want to schedule

View JSON Workflow

Back

Analysis Flow Json

Analysis Flow Fire Json

```
1
                   ٤
 2
     "fire.workflowengine.Workflow",
 3
     {
 4
       "nodes": [
 5
         "java.util.ArrayList",
 6
         ſ
 7
           t
             "fire.nodes.dataset.NodeDatasetStructured",
 8
 9
             {
               "id": 1,
10
               "name": "DatasetStructured",
11
               "path": "data/bike_sharing_sample_dataset.csv",
12
13
               "datasetType": "CSV",
14
               "separator": ",",
15
               "filterLinesContaining": "season",
16
               "header": true,
               "schema": [
17
18
                  "fire.workflowengine.FireSchema",
19
                 {
                   "columnNames": [
20
21
                     "datetime",
                     "season",
22
23
                     "holiday",
                     "workingday".
24
```

Application - Analytics			
Datasets Workflows Dashbor	ard Interactive Dashboard Workflow Executions Schedules	Share Credentials Connections	
Workflow List	Q Search	CREATE * EXPORT WORKFLOWS	COPY WORKFLOWS WORKFLOW WIZARD
Filter By Category	D USER NAME WORKFLOW NAME	TYPE CATEGORY LAST UPDATED	ACTION
E Show All Categories	A admin DistributionGraphs	scala Distribution 06/24/2019 at 10:31AM	● / ▶ ▶ ◙ ◙ ☞ ㅎ 0 ∩ ₽
a choir for congressor	3 admin Clickstream Data Analysis	scala Clickstream 06/24/2019 at 10:31AM	· • / • • • • • • • • • • • • • •
> -	2 admin ApacheAccessLog	scala Apache Log 06/24/2019 at 10:31AM	• / • • • • • • • • • • • •
> Apache Log	1 admin Analyze Flights Delay	scala Flights Delay 06/24/2019 at 10:31AM	• / • • • • • • • • • • • •
> Clickstream			

• Click on Tab Schedule New Job for Workflow

w	orkflows Sche	duled							
sci	EDULE NEW JOB FOR 1	WORKFLOW BACK							
				WORKFLOWS SC	HEDULED FOR DIS	TRIBUTIONGRAPHS			
ID	USERNAME	WORKFLOW ID	WORKFLOW NAME	LAST UPDATED	SCHEDULED AT	SPARK SUBMIT OPTIONS	PROGRAM PARAMETERS	LIB JARS	ACTION

• Update the scheduled timing & email notifications after success & failure of workflow as per our requirments.

Schedule Job		
SPARK SUBMIT OPTIONS:		
Optional		
eg:executor-memory 2g	executor-cores 2driver-memory 2g	li
PROGRAM PARAMETERS:		
CHOOSE LIBJARS:	🗏 mysql-connector-java-8.0.11.jar	
EMAIL ON SUCCESS:		
EMAIL ON FAILURE:		
◎ minute		
O HOURLY		
O DAILY		
○ WEEKLY		
MONTHLY		
	CANCEL	OK

• Click on OK to save the changes.

wo	orkflows Sch	eduled							
SCH	EDULE NEW JOB FOR	WORKFLOW BACH		WORKFLOWS SCHEDU	iled for <u>distribu</u>	TIONGRAPHS			
ID	USERNAME	WORKFLOW ID	WORKFLOW NAME	LAST UPDATED	SCHEDULED AT	SPARK SUBMIT OPTIONS	PROGRAM PARAMETERS	LIB JARS	ACTION
97	sparkflows	4	DistributionGraphs	07/16/2019 at 10:14AM	Minute At:09				/8

Debugging Workflows

Many times it is helpful to be able to debug the workflows. One easy way is to check the debug checkbox in the UI when executing the workflow.

Running in debug mode does a few things:

- Performs a count() after executing each Processor. This makes it easier to track errors. It takes out Sparkflows lazy execution from the picture.
- Displays the number of records processed at each stage.
- Display more information, for each SQL etc. which are being executed.

Passing Parameters to Workflows

Fire Insights runs the spark jobs with spark-submit. It takes in the workflow JSON as a parameter. There are multiple ways to pass extra parameters to the workflow. If the same parameter is specified multiple times, the order of precendence in which they are applied is as shown below:

- Through Program Parameters passed during Workflow Execution
- By specifying the parameters in the Workflow Editor
- Through a Parameter Processor in the workflow
- A Node creating a variable during execution time

Through Program Parameters in Fire during Workflow Execution

Key/Value pairs can be passed to Fire during Workflow Execution. An example of it is --var doctor=1 These Key/Value pairs would override any Key/Value pair passed through the Parameter Processor in the workflow.

Below is a screenshot:

Execute Workflow	
BACK	
	1505 EXECUTE
WORKFLOW NAME: 🕖	TESTING HIVE PARAMETERS
SPARK-SUBMIT-CONF: 0	
	EG:EXECUTOR-MEMORY 2GNUM-EXECUTORS 5EXECUTOR-CORES 2DRIVER-MEMORY 2G
PROGRAM PARAMETERS: 🕖	var doctor=1

By specifying the parameters in the Workflow Editor

Parameters can be specified in the Workflow Editor. They can be specified in the following format:

They can be passed with --var name1=value1 --var name2=value2

Through a Parameter Processor in the Workflow

A Parameter Processor can be added to the workflow. It allows passing key/value pairs to the workflow.

A Processor creating a variable during execution time

A Processor can also create a parameter during run time. A Processor creates a new variable and puts it into the JobContext.

jobContext.nodeGeneratedParameters.put(variable, ""+count);

This parameter can then be later used by another Processor.

For example NodeCount puts the count of records into a variable in the Job Context.

NodeAssert uses this variable when evaluating expressions.

Parameters C NodeParameters			8
OUTPUT STORAGE LEVEL :	DEFAULT		\$
KEY VALUE ARRAY : Ø			
		PARAMETER VALUES 🕖	
low		15	•
high		78	•
output		transactions	•
			OK CANCEL

Through --var parameters with spark-submit

Fire Insights workflow can also be directly executed on the cluster with spark-submit.

In this case, extra parameters can be passed with --var:

```
spark-submit --class fire.execute.WorkflowExecuteFromFile --master yarn --

→deploy-mode client fire-core-3.1.0-jar-with-dependencies.jar --postback-url_

→http://<machine>:8080 --job-id 1 --workflow-file kmeans.wf --var_

→name1=value1 --var name2=value2
```

In the workflow, these parameters can be used with \$name1 \$name2

Specific nodes make use of the parameters by substituting \$name with the value provided for the name.

An example would be : --var id=3

When specifying the expression in the RowFilter Node we can use : id > \$id

In the above **\$id** would be replaced with **3**.

Specifying --var parameters for all in Sparkflows User Interface

Sparkflows also allows specifying the **-var** parameters to be passed to all the jobs submitted through Sparkflows. Below is the screen under Administration/Configuration.

In the above, **app.vars** parameter allows specifying a space separated list of name=value pairs.

Each of these are passed to the jobs submitted by Sparkflows with --var name=value

Workflow Execution Results

The results of Workflow Execution are streamed into the Browser as they are executed and displayed in rich Format. A workflow may run for a very long time.

The results of past executions can also be viewed in the Workflow Executions page.

8.1.3 Visualizations

Visualization Processors

There are a number of Nodes/Processors in Fire which produce rich visualizations.

Conf	figurations			
SAVE CC	INFER HADO	OP CLUSTER CONFIG		Search:
ID	NAME	TITLE	VALUE1	DESCRIPTION
1	app.runOnCluster	Run on Spark Cluster	false	Whether to submit the Spark Jobs to the Spark Cluster or Run Locally
3	app.impersonateUsers	Impersonate Users	true 🕝	Whether to impersonate the logged in user, or run everything as the user Fire is running with
4	app.postMessageURL	Fire ui postback URL	http://localhost:8080/messageFromSparkJob	Fire ui postback URL for the messages of executing jobs
5	app.sparkSubmitJar	Spark Submit Jar File	/user/centos/fire-1.4.2/fire-lib/fire-core-1.4.2-jar-with- dependencies.jar	Absolute path of the Fire jar file to be used for submitting the spark jobs
6	app.nodesDir	Nodes Directory	nodes 🕢	Directory which contains the nodes json
7	app.allowExecutionInEditor	Allow Execution in Workflow Editor	true 🕝	Whether to allow execution in the Workflow Editor or not
35	app.vars	Extra variables to be passed	0	Extra variables to be passed to spark-submit with var. Format : name1=value1 name2=value2 name3=value3

Workflow Executions

									kenth:
ы	Analysisflow_ID	USER ID	Name	Category	Description	Start Time	Last Updated	View Models	View Result
١.,	31	1	Housing	Machine Learning	Housing Price Prediction	07012016 at 11:51:11AM	07010916 at 11.51:144M	View.Model	View Execution Result
2	52	1	Patropa	Recommendations	Movie Lans Recommendations	0101/2016 at 11:51/26AM	01010016 at 11.51.344M	View Model	Ves Executor Result
а.	8	1	BikeSharing, Analysis	Machine Learning	Exectaining Analysis	00/01/2016 at 11:51/42AM	01010016 at 11 51 504M	View.Model	View Execution Result
4	21	1	Housing	Machine Learning	Housing Price Prediction	07/01/2016 at 12:00:41PM	07/01/0016 at 12:09.429M	View Model	View Execution Result
5	40	1	kmeans	Machine Learning	#D/ears	01010016 at 417079M	01/01/0016 at 417/08PM	View Model	View Execution Result
6	5	1	BikeSharing, Analysis	Machine Learning	BiteSharing Analysis	provoes at 5.10 permit	01010016 at 51015PM	View.Model	Vex Executor, Result

()

These Processors can be added to any workflow and are applied to the data.

Visualization Processors include:

- Graph Values
- Geo
- Group by Column
- Weekday Distribution
- Monthly Distribution
- Yearly Distribution
- Heatmaps
- Tables

Batch Dashboards

Fire allows you to create Dashboards.

Processors in Fire can output data in Tables, Charts, Maps and Simple Strings. Dashboards allow combining the output of various processors into one User Interface.

For example we might want to output a chart of number of bike rentals per hour, another by per day and another map displaying the total number of bike rentals per city for the day. Dashboards can combine all these into one view.

Creating Dashboards

- For creating Dashboards, drag and drop the required processors from the workflows into the Dashboard Canvas.
- When the corresponding workflows are run, the output is stored by Fire into the relational store. These get displayed into the dashboard.

Editing Dashboards

Editing Dashboards is like creating dashboards, except that you click the edit button to edit the corresponding Dashboard.

Edit Dashboard Vorkflows	DASHEGARD NAME: Bike Shoring Dr	ashboa CATEGORY: Enter Cotegory	DESCRIPTION: Enter Description SAVE VIEW	0 HE
Q bioj	Workflow : Bike Sharing Analysis	Workflow : Bike Sharing Analysis	Workflow : Blike Sharing Analysis	
e Bike Sharing Analysis	Name: Graph Count of	Name: Correlation	Name: Summary	
DatasetStructured Extract Hour from Time	Rentals	Type: transform	Type: transform	
Cast Count to Double Assemble Features for	Type: transform			
VectorIndexer Split 80-20				
GBTRegression Predict				
RegressionEvaluator Count of Rentals per H				
Graph Count of Rentals Workflow Notes				
Correlation Summary				

Viewing Dashboards

Once a Dashboard has been created you can view it, by clicking on the View button.

board: Bike Sharing Dashboard				Grid Component Size Co	ntrols: E
ount of Rentals per Hour	Correlation between various Columns	Bike Shar	ing Dataset S	ummary	
		summory	season	holiday	wor
COLUMN CHART BAR CHART LINE CHART	11/1/1/10/1	count	10656	10886	105
		mean	2.5056139996325556	0.02856880396839978	0.6
		min	1.0	0.0	0.0
254,000	worklegday	25_percentile	2.0	0.0	0.0
200.000	wather and an an an an an an an an	50_percentile	3.0	0.0	1.0
20000	homidity	75_percentile	4.0	0.0	1.0
150,000	caseal registered	max	4.0	1.0	1.0
100,000	court in the second	stdev	1.116174309344325	0.16559885062470958	0.4
	Lang and an	variance	1.2458450888402808	0.027755177029474297	0.21
51,000	atang				
	windepeed				

Streaming Dashboards

- Fire allows you to create Streaming Workflows.
- Streaming workflows have a mini-batch duration say 30 seconds.
- In this case, the output in the Dashboards get updated every 30 seconds as new data come in.

Interactive Dashboard

Fire allows you to create interactive Dashboard.

Fire allows us to create New Dataset using JDBC data type from MYSQL DB & use datasets in creating charts & dashboard.

Creating I-Dashboard

• For creating I-Dashboard, Create JDBC datasets if not available.

You can define a New Dataset by clicking on the Create Dataset button in the Dataset page.

It will bring up the dialog box below. Select the format of the file for which the new Dataset is being created.

Select Dataset Type	×
AVRO	
CSV	
HIVE	
ЈДВС	
JSON	
PARQUET	
SEQUENCE	
XML	
	OK CANCEL

Entering Field Details

Below are the details of the fields in the Create Dataset page:

- NAME : Name of the New Dataset we are creating.
- **DESCRIPTION** : Description of the New Dataset.
- **CATEGORY** : category of the New Dataset.
- **JDBC DRIVER** : Enter JDBC DRIVER.
- JDBC URL : Enter JDBC URL for MYSQL DB.
- USER : username for MYSQL DB.
- **PASSWORD** : password for MYSQL DB.
- **DB** : Database for MYSQL DB.
- **TABLE** : Table for MYSQL.

DataSet Details - JDBC			O HELP SAVE BACK
NAME: @*	New JDBC Dataset	JDBC URL: @*	jdbc:mysql://server_ip:3306
CATEGORY: O	jdbc	USER: @*	fire
DESCRIPTION: 0	laan details	DB: 😶*	testdb
JDBC DRIVER: 0*	com.mysql.jdbc.Driver	PASSWORD: @*	
		TABLE: @*	fire

Updating the Schema of the Dataset

You can update the Schema of the Dataset by clicking on Update Sample Data/Schema. It would display sample data for the dataset followed by the Schema inferred by Fire Insights.

You can update the column names in the schema based on your data.

SAMPLE DATA: 0

Loan_ID (STRING)	Gender (STRING)	Married (STRING)	Dependents (STRING)	Education (STRING)	Self_Employed (STRING)	ApplicantIncome (INTEGER)	Coo (NTE		LoanAmount (INTEGER)	Loan_Amount_Term (STRING)		redit_History TRING)	Property_Area (STRING)	Loan_Statu (STRING)
.P001003	Male	Yes	1	Graduate	No	4583	1508		128	360	1		Rurol	N
.P001005	Male	Yes	0	Graduate	Yes	3000	٥		66	360	1		Urban	Y
P001006	Male	Yes	0	Not Graduate	No	2583	2358	3	120	360	1		Urban	Y
P001008	Male	No	0	Graduate	No	6000	0		141	360	1		Urban	Y
					DATA TYPE 😶			FORMAT @				ML TYPE 😧		
NAME														
					DATA TYPE 🛛			FORMAT @				ML TYPE 😧		
Loan_ID					STRING			FORMAT @				MLTYPE O		
Loon_ID														
					STRING		* *	format				TEXT		
Loon_ID Gender					STRING		* *	format				TEXT		
Loon_ID Gender Married	ents				STRING STRING STRING		•	format format				TEXT TEXT TEXT		

Saving the New Dataset

Click on the Save button to save the New Dataset created.

Interactive Dashboard

Click on Interactive Dashboard tab in the same application where you have created JDBC Dataset.

Applic	ation - D	ASHBOAR	D APPLICATION							DASHBOARD APPLICATION
Datasets	Workflows	Dashboard	Interactive Dashboard	Executions	Schedules	Share	Credentials	Connections		
Intera Charts	ctive Das	hboards								
Choo	ise a JDBC data	sset ¢	EATE NEW						Q Search Dashboo	rd

Click on chart tab & select Choose a JDBC dataset, there you will find all JDBC datasets created under your application.

Ap	plication – D	ASHBOA	RD APPLICATION									DASHBOAR	D APPLICATION
atas	ets Workflows	Dashboard	Interactive Dashboard	Executions	Schedules	Share	Credentials	Connections					
Inte	eractive Das	shboards											
Cha	rts Dashboard												
	Choose a JDBC date	laset ¢	REATE NEW						C	2 Search	Dashboard		
		laset ¢	REATE NEW						C	2 Search	Dashboard		
	Choose a JDBC date Choose a JDBC date	laset ¢	REATE NEW DESCRIPTION						C	A Search	Dashboard	EDIT	DELETE

Select any JDBC dataset for which you want to create chart & select CREATE NEW

It will take you to new page, as below

NAME iDashboard 1		DESCRIPTION	SAVE BAC
e COUNT	BAR CHART *	Y	
A Loan_ID	Filters	X And Drop to create group of bars	
A Gender			
A Married		Group results to see chart	
A Dependents			
A Education			
A Self_Employed			
ApplicantIncome			
CoopplicantIncome			
LoanAmount			
A Loon_Amount_Term			
A Credit_History			
A Property_Area			
A Loan_Status			

Select the chart type, you want to see chart

Selected Bar chart & updated column for x & y axis and add some filter

Add NAME, DESCRIPTION & save it

Once you save it, the chart will appear in chart list page

Similarly you can create different chart using specified chart type

Now using existing chart, you can create new dashboard

Select Dashboard tab & Click on CREATE DASHBOARD

it will take us to New Dashboard page

Using drag & drop you need to add chart in canvas, Add NAME, DESCRIPTION & SAVE it.

Once the Dashboard got saved successfully, it will show in dashboard list page from where you can view, edit & delete it.

Exporting Visuals

Fire Insights enables you to export the output, dashboards and visuals in various ways.

		COUNT	BAR CHART -				
				У			
				x	And Drop to create group of bors		
					Group results to see chart		
			Geo Chart				
			and a second bill bill				
		Loan_Amount_Term					
		Property_Area					
	And						
		AME Loan chart Analysis		DESCRIPTION Loan or	nalysis of Education with their income & filter wi		SAVE BACK
		COUNT	010 01007 -				
				y • Applicantincome * • Co	oppicantinc*		
				× Education ×			
				2000000	Applicantincome Coopplicant	tincome	
		Credit_History					
		Loan_Status					
				200000			
				0.1	Graduate	Not Graduate	
		Application - DAS	HBOARD APPLICATIO	N			DASHBOARD APPLICATION *
		100 M 10			Phone Condentials Con 1		
				ru Executions Schedules	enure Credentials Connections		
		nteractive Dashb	bards				
		harts Dashboard					
		Choose o IDBC dotores	CREATE NEW			Q Search Dashbourd	
		annes a jour adraset.				 Second Control data 	
		CHARTNESS	DESCRIPTION				SDIT DOLT
				ion with their income & filter with Marrie	ed status using bar chart		
					and an and the second s		
					ter with price using bar chart		
		-					
		Application - DAS		N			DASHROADD APRICATION
ntercetive Dashboards							and a resolution *
				rd Executions Schedules	Share Credentials Connections		
CECCECCEC C C C C C C C C C C C C C C C		Interactive Dashb	oards				
CECCECCEC C C C C C C C C C C C C C C C		Charts Dashboard					
Mk OSCIENTION Mk OSCIENTION Mk OSCIENTION Not ML Not Constant	NM Decembra NM Isometrial NM Isometrial Isometrial Fare Fare	CREATE DASHBOARD				Q Search Dashboard	
shi Lice don't Description Des	Ale de contrat de la contrat de contrat						
shi Lice don't Description Des	sort L Lea chor L Can	NAME		DESCRIPTION		SAVE 📥 EXF	PORT BACK OHELP
Luch dram 1 Composed 1 Compo	4 us no nom 4 de sou har han 5 te sou har han 6 te sou har har han 6 te sou har har han 6 te sou						-
la Consultation the Consultation the Consultation the Supervised States the Consultation the Supervised States the Consultation the	d 2 cm, the d 2 c	Loan chart					
LG co or vinh R burg C co or vinh R co	d G do do rank d E do ran	Churn_chart					
Li Buakong di Gooderalm A berg Mi Sanderalm Manog and Con charl Analysis Con charl An	Le Leadors de la de la d	Geo chart with					
er di Caso dari dali la bergi dali funzione de la bergi da stationara le me tere Mili Can and Houseg Daveband di BECEPTION. Can and Houseg Dava Analysis Pri Stationara Pri Pri Pri Pri Pri Pri Pri Pri	the de do dart hills de do dart hills de do dart hills te do da	Housing chart					
La large and basis portational dem here Mar Lone and taxing Daraktood dem here Mar Lone and taxing Daraktood dem here Mar Lone and taxing Daraktood dem here Dong and Dara Analysis The Lone and taxing Daraktood dem here Dong and Dara Analysis The Lone and taxing Daraktood dem here Dong and Dara Analysis The Lone and taxing Daraktood dem here Dong and Dara Analysis The Lone and taxing Daraktood dem here Dong and Dara Analysis The Lone and taxing Daraktood dem here Dong and Daraktood dem here	de à long and Drag a diptiloard item here Ten dans Conc., chort Conc., chort Conc.	hart					
Log and Drag a distilicand term term	Long of Dags & dathbord lies here Mix Long of Charlysis Cong charl Hardysis Cong charl Hardysis Co						
Met Lean and Houring Darkkowst The Construction Construc	Met (control hours) Darboard (control hours) Darb Arshys) (control hours) (control hours) Darb Arshys) (control hours) (co						
rh Loon charl Analysis T and that the Bandrog T and the Bandrog T and that the Bandrog T	off Loca charl Andryjs Andrew Andrew Construction Cons			Drog c	and Drop a dashboard item here		
rh Loon charl Analysis T and that the Bandrog T and the Bandrog T and that the Bandrog T	rhf Loca charl Analysis A charl with Barger The ch						
rh Loon charl Analysis T * X1 * X Bashabring charl And and and Bashabring Catation and Catation and Catationand Catation and Catation and Catation and Catatio	rh too chort Analysis T x x x x too chort Analysis T x x x x too chort Analysis T x x x x too chort Analysis T count of the too chort Analysis T co						
rh Loon charl Analysis T * X1 * X Bashabring charl And and and Bashabring Catation and Catation and Catationand Catation and Catation and Catation and Catatio	rh too chort Analysis T x x x x too chort Analysis T x x x x too chort Analysis T x x x x too chort Analysis T count of the too chort Analysis T co	ME Loan and Housing D	ashboard	DESCRIPTION Loop on	nd Housing Data Analysis	SAVE A DYD	ORT BACK OHELP
Lion both Change data Banderson Borg data Banderson Borg data Banderson Court, chert Court,	Lion both Change data Basedors Borg of the Borg of th						
M ⁿ Chu-goarts Min Sharah sharah Bharah sharah sharah Bharah	here details de la subset de la	Loan chart	an chan Analysis	т- х	1* x Divestigning chan	X1* ¥	
Cac dat with with with with with with with wit	Cac data with with with with with with with with	slysis Churn short		rre Coappleantincome	500000 weather temp temp	midity windspeed	Search.
a long and	a long and	Geo chart with 1800	600		49000		
a long and	a long and	Housing chart 1400	.000		30000		YES (394)
a long and	a brag man and a brag	art 600	1000		20000		APPLY GLOBAL
borg	bing bing	Geo chart with 400	1000		50000		price D
		along	0	Not Graduate			
		He				T* X1* X	100237 116598
							APPLY GLOBAL
		14		bed	montes and contracts and stories		
		12 10 8	_				
المرابع المراجع المراجع المراجع المراجع المراجع المراجع المراجع	المرابع المرابع المرابع المرابع						A4. (124)
Chum_chort T- x1- x 0 0.000 0 cmm 8 cmm	Chum_chart T+ x1+ X 0 00% 0 07%	6-4-					
		4 2 0					AR (92)
		4- 2- 0- 10 ⁵		2 ⁵ 10000 ° 10000 ° 10000 ° 10000 ° 1	1998 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999		AR (92)
		4- 2- 0- 105		9 CROST, CANES, CANES, CROST, C	מאזכון ^מ משפון ^מ משנון ^מ משנון ^מ משנון ^מ משנון ^מ משנון ^מ משנון ו		□ AR (92) 2 AZ (89) □ CA (52) □ CO (96)

Арр	lication - DASHBOAR	D APPLICATION								DASHBOARD APPLICATION
Datasets	s Workflows Dashboard	Interactive Dashboard	Executions	Schedules	Share	Credentials	Connections			
Inter	ractive Dashboards									
Charte	Dashboard									
C	REATE DASHBOARD								Q Search Dashboar	i.
ID	DASHBOARD NAME		DESCRIPTION				USER NAME	LAST UPDATED		ACTIONS
33	Dashboard- I		chart visualizati	on			sparkflows	2020+07+15 12:36:24 151		•/8
36	Dashboard-II		geo chart visual	ization			sparkflows	2020-07-15 13:13:29 IST		•/8
										•/A

Exporting dashboard

Since Fire Insights is Browser based end to end, its easy to export the pages as PDF files.

- Go to dashboard under your application where you have created batch dashboard
- On clicking on view option, able to visualize etc. added in that dashboard, there you will have Export option, Click on that.

Correlation Matrix	Summar	y				Row Valu	∋s		
	summary	request_id	sales_id	sf_id	accos	REQUEST_ID	SALES_ID	SF_ID	ACCOUNT_NUM
Salar State Salar Sala	count count	2489	2489	2489	2489	IntegerType	IntegerType	IntegerType	IntegerType
adard gits of a coord good good at a contra	and the second second	207.497	1061.111	159499.894	159501	266	909	159625	159627
SALES ID	min	0.0	554.0	159400.0	15940;	317	1243	159625	159627
SF. ID	25_percentile	183.0	909.0	159420.0	159423	281	1285	159625	159627
COUNT_NUMBER	50_percentile	224.0	1051.0	159445.0	159442	204	1198	159625	159627
UNIQUE_ID	75_percentile	252.0	1215.0	159605.0	15960;	288	988	159625	159627
USERVAILID	max	331.0	1627.0	159625.0	159622	251	1232	159625	159627
ROLE_ID	stdev	69.33	205.244	90.514	90.514	281	1040	159625	159627
ASIRILITY_ID	variance	4806.638	42125.157	8192.724	8192.7	19-4	1286	159625	159627
DEAL_ID						184	907	159625	159627

It will Export the whole batch dashboard in pdf format on local machine.



Exporting output

Once the workflow successfully completed, the output result can be exported.

• Go to application page where you created workflow & successfully executed.

Clicking on Executions tab the latest workflow execution will show in list page.

On action icon you can see view result, it will take to next page.

On opening above link, able to view result of specific workflow submitted & have Export option through which you can export result in local machine in pdf format & view that.

Appli	ication – An	alytics									Analytics *
Datasets	Workflows	Dashboard	Interactive Dashboard	Executions Scho	edules Share	Credentials	Connections				
Work	flow Execu	tions						Search by name			Q Ø HELP
								REFRESH DELE	TE STOP KILL	DELETE OLD WORKFLOW	EXECUTIONS *
Status:	All Running	Stopped	Completed Failed S	itarting Stop	Kiled			REFRESH DELE	TE STOP KILL	DELETE OLD WORKFLOW	EXECUTIONS *
Status:	All Running WORKFLOW ID	Stopped	Completed Foiled S	itarting Stop	Kiled	STATUS	ACTION	CATEGORY	TE STOP KILL DESCRIPTION	DELETE OLD WORKFLOW	EXECUTIONS *
						STATUS	ACTION			START TIME	LAST UPC
ID	WORKFLOW ID	USER NAME	NAME	WEW	D Lat *		ACTION	CATEGORY	DESCRIPTION	START TIME 80cf 2020-07-30 18:08:5	LAST UPC 87 IST 2020-07
1D 141	WORKFLOW ID	USER NAME sporkflows	NAME Transaction Data Analytics		0 Lad * 0 Lad *	Completed	ACTION	CATEGORY Transaction	DESCRIPTION Plats number of Trans	start TIME soct 2020-07-30 18:08:5 d m 2020-07-30 17:05:3	1 LAST UPC 57 IST 2020-03 19 IST 2020-03

Applic	cation – An	alytics									Analytics *
Datasets	Workflows	Dashboard	Interactive Dashboard	Executions Schedu	les Share	Credentials	Connections				
Workf	low Execu	tions						Search by name		۹	O HELP
								REFRESH DEL	ETE STOP KILL DE	LETE OLD WORKFLOW EXECUT	IONS -
Status:	All Running	Stopped	Completed Failed	Starting Stop P	iled						
ID .	WORKFLOW ID	USER NAME	NAME	View Result		STATUS	ACTION	CATEGORY	DESCRIPTION		
								Childown	DESCRIPTION	START TIME	LAST UPD
141	14	sparkflows	Transaction Data Analytics	6 🗮 🛦 📼	<u>ad</u> *	Completed		Transaction	Plots number of Transact	2020-07-30 18:08:57 IST	2020-07
141140	14 326	sparkflows sparkflows	Transaction Data Analytics Telco Churn Prediction - D.	6 <mark>=</mark> 4 =							

low Valu										
id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType
6246	205	7	707	1078778070	12564	2012-03-02	12.0	oz	1	7.59
56246	205	63	6319	107654575	17876	2012-03-02	64.0	oz	1	1.59
96246	205	97	9753	1022027929	0	2012-03-02	1.0	CT	1	5.99

The counts by column: brand

50 -----

ow Val	ues									
8	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
ntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType
6246	205	7	707	1078778070	12564	2012-03-02	12.0	OZ	1	7.59
6246	205	63	6319	107654575	17876	2012+03+02	64.0	OZ	1	1.59
6246	205	97	9753	1022027929	0	2012-03-02	1.0	ст	1	5.99
he cou	column: brand nts by colu		d					eceived Files		0

8.1.4 Scheduling

Fire allows you to schedule workflows by time to be executed.

Scheduling Workflows

Fire allows you to schedule workflows to be run at regular intervals.

Scheduling New Workflow

The workflows page displays the list of various workflows.

Under Action column, there is an icon to schedule any given workflow.

Clicking on the icon takes you to a page for creating new schedules for the workflow. Clicking on Schedule New Job for Workflow opens the dialog for creating a new schedule.

Schedule Job	
SPARK SUBMIT OPTIONS: Spark Submit Optior	
HOURLY DAILY WEEKLY MONTHLY CUSTOM CRON PATTERN	
MINUTE:	Cancel

Viewing Workflows Scheduled

Scheduled/By Time page displays the various workflows scheduled.

No	rkflows Sch	neduled							
	USERNAME	WORKFLOW ID	WORKFLOW NAME	LAST UPDATED	SCHEDULED AT	SPARK SUBMIT OPTIONS	PROGRAM PARAMETERS	LIBJARS	ACTION
93	ge	n	Farmers Markets on Geo Map	07/28/2019 at 6:23PM	Hourly At:08				/8
34	ge	3	Clickstream Data Analysis	07/28/2019 at 6:23PM	Daily At 02:05				18

Editing a Scheduled Workflow

You can edit a schedule by clicking on the edit icon, updating the new values and saving it.

Viewing Results of Workflow Executions

When workflows are scheduled, they are executed by Fire at the specified schedule.

The results of the execution of the workflows can be viewed in the Workflow Executions Page. This allows us to view the results of past execution, logs of the run etc.

Workf	low Exec	ution	s											
DELETE OL	D WORKFLOW	EXECUT	IONS									Search:		
EXECUTION	WORKFLOW ID	USER ID	NAME	CATEGORY	DESCRIPTION	START TIME	LAST UPDATED	DURATION	STATUS	VIEW EXECUTIONS	VIEW RESULTS	VIEW FULL RESULTS	STOP EXECUTION	LOC
979	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/15/2017 at 11:03:00PM		00-25180263:00	COMPLETED	Weav Executions	View Results	View Full Results		Log
947	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 11:03:00PM		00-25178823:00	COMPLETED	Wew Executions	View Results	View Full Results		Log
946	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 11:01:00PM		00>25178822:00	COMPLETED	Wew Executions	View Results	View Full Results		Log
945	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 10:01:00PM		00>25178761:00	COMPLETED	Wew Executions	View Results	View Full Results		Log
944	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 9:01:00PM		00>25178701:00	COMPLETED	Wew Executions	View Results	View Full Results		Log
943	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 8:01:00PM		00>25178641:00	COMPLETED	Wew Executions	View Results	View Full Results		Log
914	21	3	Dedup Customers	ETL	Finding the similar users in two different data sets.	11/14/2017 of 7:01:00PM		0025178581.00	COMPLETED	Wew Executions	View Results	View Full Results		Log

Deleting a Scheduled Workflow

Go to the Scheduled/By Time page. It would display the list of scheduled workflows.

Click on the delete icon next to any schedule workflow to delete the schedule.

Notifications & Alerts

Users in general like to be alerted when a job completes or fails, specially in Big Data where Jobs can run for hours together.

Email Notifications/Alerts when Executing Workflows

When executing the workflows, you can specify email addresses for receiving emails when the workflow fails or succeeds.

٩	Fire Insights 🛛 🗠 DATA BROWSERS 👻	⊙ APPLICATIONS 🔹 ⊙ SCHEDULED 🔹 😫 WORKFLOW EXECUTIONS 👻 🖄 ADMINISTRATION 👻 III PROCESSORS 🕤		
	BACK EDIT			
	Workflow Id: 0	27 Execut	E Debug	
	Workflow Name: 🛛	Dedup Customers		
	Spark-submit-conf: 0	eg:executor-memory 2gnum-executors 5executor-cores 2driver-memory 2g		
	Program Parameters: 🕑			
	Choose jar files: 🛛			
	Email on success:			
	Email on failure:			

Email Notifications/Alerts when Scheduling Workflows

When scheduling the workflows, you can specify email addresses for receiving emails when the workflow fails or succeeds.

Schedule Job

SPARK SUBMIT OPTIONS:

Spark Submit Options			
eg:executor-memory	gnum-executors 5executor-cores	s 2driver-memory 2g	/i
PROGRAM PARAMETERS:			
CHOOSE LIBJARS:			
EMAIL ON SUCCESS:			
EMAIL ON FAILURE:			
O DAILY			
• WEEKLY			
		CANCEL	ОК

SMTP Configurations

Administrator has to set up the SMTP configurations under Administration/Configuration

-				
SAVE CONFIGURATIONS	INFER HADOOP CLUSTER CONFIG			٩
PP SPARK HDFS	HADOOP YARN HIVE KERBEROS			
NAME	TITLE	VALUE		DESCRIPTION
alert.mailFrom	Mail From		ß	Email address from which to send email alerts
alert.mailSmtpHost	Mail SMTP Host		ß	SMTP Host to be used for sending emails
alert.mailSmtpPort	Mail SMTP Port		ß	SMTP Port to be used for sending emails
	Mail SMTP Auth User		ß	SMTP Auth Username for sending emails
alert.mailSmtpAuthUser				

Triggering Workflows by Event

Workflow Executions can be triggered by sending an event to a Kafka Topic. Fire can be configured to poll for events from those topics.

Use Case

The kind of use cases this can handle are:

- A job loads data into HIVE
- Now the job wants to trigger another workflow
- It pushes an event to a Kafka Topic to trigger the workflow

Event Format

Events which are pushed to Kafka are string with the fields separated by | (pipe).

Below is the format of the event.

TypelValuelSpark Submit ConfigslExtra Jar FileslProgram ParameterslEmails on SuccesslEmails on Failure

- Type : Type determines the kind of data in the Value column
 - 0 : workflow id
 - 1 : workflow name
 - 2 : workflow uuid
- Value : This defines the value. Values are based on the Type used:
 - ID of the workflow

- Name of the workflow
- UUID of the workflow
- Spark Submit Configs: Extra Spark Submit configurations to be applied when running the Spark Job.
- Extra Jar files : Extra jar files to use in spark-submit.
- Program parameters : Extra program parameters if any.

- Program Parameters are passed to the workflow. Example : --var key1=value1.

- Email on Success : email addresses to send Success email on Job Completion.
- Email on Failure : email addresses to send Failure email on Job Failure.

Example Events

• 0151 | | |success@sparkflows.iolfailure@sparkflows.io

In the above example:

- 0 : Trigger by workflow id
- 5 : Workflow id to trigger
- success@sparkflows.io : Email address to send regarding success of the workflow
- failure@sparkflows.io : Email address to send regarding failure of the workflow

Configuring Fire to listen for Events from the Kafka Topic

Fire has to be configured to listen for Events from the Kafka Topic. Each user can configure their own. The Jobs would be fired as a user who configured it.

8.1.5 Export / Import of Applications

Fire enables you to export your Applications and download them to your computer.

It then also enables you to import your Applications back into any instance of Fire.

This is useful when you need to move/copy your Application from one environment to another.

Exporting Applications

Fire allows you to export Applications and download them to your computer. Below are the steps for exporting Applications in Fire.

Go to the Applications Page

Арр	olicati	ons			I Summary			IMPOR	π	EXPORT CREAT
					III List					
	ID	USER NAME	NAME	DESCRIPTION		DATE CREATED		LAST UPDATED	ACT	IONS
	17	sparkflows	Test-opp	TEST		10/11/2019 at 4:	28:22PM	10/11/2019 at 4:28:22PM	ø	1
	33	sparkflows	DataQuality			10/19/2019 at 10	:05:42	10/19/2019 at 10:05:42	ø	<u>ڨ</u>
	34	sparkflows	Banking			10/19/2019 at 10	:13:53P	10/19/2019 at 10:13:53P	1	1
	36	sparkflows	RR_Test			10/25/2019 at 1	:23:18A	10/25/2019 at 12:23:18A		÷.
	38	sparkflows	Telco_Enterprise_Feas	Telco_Enterprise_F	easibility_UseCa	10/31/2019 at 4	54:00P	10/31/2019 at 4:54:00P		÷.
	39	sparkflows	SCHOOL	Testing		11/12/2019 at 10	29:23PM	11/18/2019 at 3:30:23PM	ø	Û
	41	sparkflows	Treatment	Test		11/15/2019 at 1:0	6:53PM	11/15/2019 at 1:06:53PM	1	1
	43	sparkflows	testapp			11/16/2019 at 11:	50:11AM	11/16/2019 at 11:50:11AM		÷.
	65	sparkflows	Banking	Banking Application	1	12/06/2019 at 1	56:49PM	12/06/2019 at 1:56:49PM	ø	÷
	67	sparkflows	Machine Learning App			12/06/2019 at 2	11:26PM	12/06/2019 at 2:11:26PM	ø	Û
-										~

Select the Applications you want to export

- Select the Applications you would like to export.
- Then click on the Export button.

🔶 Fire Insig	ghta 🖓 DATA BROWSERS 👻 III APP	LICATIONS - © SCHEDULED - ¢	WORKFLOW EXECUTIONS - 🖉 MODELS	* III PROCESSORS *		0 * = • • • • •
Applicati	ions			Q Search		IMPORT EXPORT CREATE
• •	USER NAME	NAME	DESCRIPTION	DATE CREATED	LAST UPDATED	ACTIONS
36	sparkflows	RR_Test		10/24/2019 of 11:53:18AM	10/24/2019 of 11:53:18AM	/8
38	sparkflows	Telco_Enterprise_FectsIbility_UseC	Telco_Enterprise_Feasibility_UseCase1	10/31/2019 of 4:24:00AM	10/31/2019 of 4:24:00AM	/8
39	sparkflows	SCHOOL	Testing	11/12/2019 of 8:59:23AM	11/11/2019 at 2:00:234M	/8
41	sparkflows	Treatment	Test	11/14/2019 of 11:38:53PM	11/14/2019 at 11:36:53PM	× 8
43	sparkflows	testopp		11/15/2019 of 10:20:11PM	11/15/2019 at 10:20:11PM	/8
65	sparkflows	Banking	Banking Application	12/05/2019 of 12:26:49AM	12/06/2019 at 12:26:49AM	× 8
	admin	Analytics	Analytics Application	10/11/2019 of 3:54:34AM	10/11/2019 at 3.54:34AM	/8
2	admin	Connectors	Connectors Application	10/11/2019 of 3:54:344M	10/11/2019 of 3.54.344M	× 8
🗹 a	admin	CreditCordFroudDetection	CreditCardFraudDetection Application	10/11/2019 at 3:54:34AM	10/11/2019 at 3:54:34AM	/8
4	admin	DataPreparation	DataPreparation Application	10/11/2019 of 3:54:344M	10/11/2019 of 3.54.344M	/8
	admin	Demo	Demo Application	10/11/2019 at 3:54:34AM	10/11/2019 at 3:54:34AM	/8
□ •	admin	ETL	ETL Application	10/11/2019 of 3:54:344M	10/11/2019 of 3.54.344M	× 8
D 7	admin	FileFormats	FileFormats Application	10/11/2019 at 3:54:34AM	10/11/2019 at 3:54:34AM	/8
•	admin	Financials	Financials Application	10/11/2019 of 3.54:344M	10/11/2019 of 3.54.344M	/8
□ ⁹	admin	Languages	Languages Application	10/11/2019 at 3:54:34444	10/11/2019 at 3:54:34AM	/8
10	admin	MachineLearning	MachineLearning Application	10/11/2019 of 3.54:344M	10/11/2019 of 3.54.344M	×8
• •	admin	MarketingAnalytics	MarketingAnalytics Application	10/T/2019 of 3:54:35AM	10/11/2019 of 3:54:35AM	/8

• In the dialog box which comes up, select whether you want to export workflows or datasets or both.

✓ ◎ SCHEDULED ✓ ♦ 1	Export Application Components	×	
	WORKFLOWS DATASETS		
	EXPORT	CANCEL 3:18AM	LAST UPDATED 10/24/2019 at 11:53:18AM
terprise_Feasibility_UseC	Telco_Enterprise_Feasibility_UseCase1	10/31/2019 at 4:24:00AM	10/31/2019 at 4:24:00AM
	Testing	11/12/2019 at 8:59:23AM	11/18/2019 at 2:00:23AM
t	Test	11/14/2019 at 11:36:53PM	11/14/2019 at 11:36:53PM
		11/15/2019 at 10:20:11PM	11/15/2019 at 10:20:11PM
	Banking Application	12/06/2019 at 12:26:49AM	12/06/2019 at 12:26:49AM

• Fire will now export the selected applications and download them to your computer.

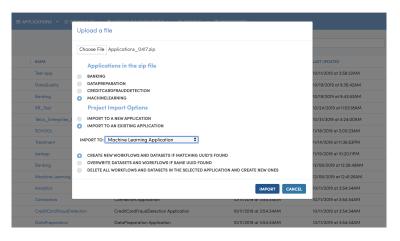
Importing Applications

Fire allows you to import Applications. Below are the steps for importing Applications in Fire.

Export Information!						
TOTAL PROJECTS	TOTAL WORKFLOWS	TOTAL DATASETS				
5	90	30				
		ОК				

Go to the Applications Page

- Click on the Import button.
- Choose the zip file from your computer to Import from. You would have previously downloaded this zip file from Fire during the export process.
- Select the name of the Application which you would like to import from the zip file. Fire would display all the available Applications in your zip file.



Select the Options for importing the Application

There are two options when importing Applications:

- Import to a New Application
 - In this case, the selected Application would be imported as a new Application in Fire Insights.
- Import to an Existing Application

When importing to an existing Application, there are 3 possible methods to choose from:

- Create new workflows and datasets when matching UUID's found.
- Overwrite datasets and workflows if matching UUID found.

• Delete all workflows and datasets in the selected Application and create the imported workflows and datasets as new ones.

On Success

On successful import of the Application into Fire Insights, the success dialog is displayed along with the details of the import.

💿 SCHEDULED 👻 🍪 🕅	Import Inforr	mation!			×	
	COMPONENTS	TOTAL	SUCCESS	FAILURE		
	PROJECTS	1	1	0		LAST U
	TWORKFLOWS	48	48	0	:22AM	10/11/
	DATASETS	22	22	0	5:42AM	10/19,
					3:53AM	10/19.
				_	3:18AM	10/24
orise_Feasibility_UseC	т			ОК	4:00AM	10/31/
	Testing			11/12/2019 at	8:59:23AM	11/18/
	Test			11/14/2019 at	11:36:53PM	11/14/
				11/15/2019 at	10:20:11PM	11/15/

8.1.6 Data profiling

Fire Insights allows you to clean the datasets using dataset profile. Below are the steps for Data Profiling in Fire.

Go to the Applications Page

Go to application page where you need to create dataset or already have existing.

select dataset tab.

	opplico	ution -	Analytics	hourd Executions Set	ork los	Share Credentials	Connection			Analytics *
	atase					Q	Search Datas			GREATE * EXPORT DATASETS
Disp	olay List E	ly Type:	ALL CSV SEQUENCE AV	RO PARQUET HIVE	JSON	XML JDBC				
	ID	USER NA.	NAME CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACTION
	1	admin	Telco Chum Prediction	Lobeled Telco Chum Predicti	CSV	/tmp/data/chum_pr		2020-07-14 15:50:00 I	PERMISSION_MODIFY	/ • & 2 2 4 1
	7	admin	Transaction Dataset	Retail Transactions Datset	CSV	/tmp/data/transactio		2018-09-16 17:50:18 IST	PERMISSION_MODIFY	/ • & @ / @ =
	6	admin	NYC Trip Data	NYC Trip Data	CSV	/tmp/data/trips-subs		2016-07-16 18:25:29 IST	PERMISSION_MODIFY	/ • 🗞 🖉 🖉 🗎
	5	admin	JetRail Train	JetRoil Train	CSV	/tmp/data/jetrail/Trai		2016-07-13 07:22:57 1	PERMISSION_MODIFY	/ • & 2 2 1 1
	2	admin	Farmers Market	Formers Market State Wise	CSV	/tmp/data/formers		2016-06-25 22:13:21 IST	PERMISSION_MODIFY	/ • & 0 0 0 0 0
	4	odmin	General Payment D.,	General Payment Dataset	CSV	/tmp/data/Sample		2016-05-09 18:23:56 1	PERMISSION_MODIFY	/ • & 8 8 7 1 1
	3	odmin	Flights Delay	Flights Delay Dataset	CSV	/imp/doto/flights_do		2016-05-28 01:36:021	PERMISSION_MODIFY	/ • & @ / @ =

Select a dataset & under action icon choose Dataset profile.

Once you Click on Dataset profile, it will take us to next page.

Click on RUN DATA PROFILING option

Once you click on above option, will get notifications about process is getting started.

Once the execution process completed, after refresh the status will updated to green, if its completed and check its execution result in RHS

A	pplic	ation -	Analytics							Analyti
Date	asets	Workflow	s Dashboard Interactive Dash	board Executions Sch	edules	Share Credentials	Connection			
D	atas	ets				٩	Search Datas	et		CREATE * EXPORT DATAS
Disp	olay List	By Type:	ALL CSV SEQUENCE AV	RO PARQUET HIVE	JSON	хмг јовс				
3	ID	USER NA.	NAME CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	AC Dataset Profile
1	1	odmin	Telco Chum Prediction	Lobeled Telco Chum Predicti	CSV	/tmp/data/churn_pr		2020-07-14 15:50:00 L.	PERMISSION_MODIFY	/ @ 🖧 🕢 🖓 🔒
	7	admin	Transaction Dataset	Retail Transactions Datset	CSV	/hmp/data/transactio		2018-09-16 17:50:18 IST	PERMISSION_MODIFY	/ • & @ @ @ @
	6	admin	NYC Trip Data	NYC Trip Data	CSV	/tmp/data/trips-subs		2016-07-16 18:25:29 IST	PERMISSION_MODIFY	/ • & @ @ @ @
	5	odmin	JetRoil Troin	JetRoil Train	CSV	/tmp/data/jetrail/Trai		2016-07-13 07:22:57 L.	PERMISSION_MODIFY	/ • & @ 6 @ 8
	2	odmin	Formers Market	Formers Market State Wise	CSV	/tmp/data/formers		2016-06-25 22:13:21 IST	PERMISSION_MODIFY	/ • & @ 6 @ 8
	4	odmin	General Payment D	General Payment Dataset	CSV	/tmp/data/Sample		2016-06-09 18:23:56 1	PERMISSION_MODIFY	/ • & @ @ @ @
	3	odmin	Flights Delay	Flights Deloy Dotoset	CSV	/tmp/data/flights_da		2016-05-28 01:36:02 1	PERMISSION_MODIFY	1 • & 0 6 6 6 6

Dataset Profile - Telco Churn Prediction	
BACK	DATA CLEANING WORKFLOW *
Dataset Profile Sample Data	
RUN DATA PROTEING	

🕹 Fire Insights 🕘 data browsers - III applications - O scheduler - O edecutions - 🗞 models - 🛢 data quality - 🛢 processor	S * Data Profiling Execution X
Dataset Profile - Telco Churn Prediction	VIEV Running.
BACK	DATA CLEANING WORKFLOW *
Dataset Profile Sample Data	
RUN DATA PROFILING EDIT DATA PROFILING REFRESH Execution Result	
Executions	
2020-07-30 17/05/39 IST Sketus : RUNNING	

Dataset Profile – Telco Churn Predi	ction												v	EW ALL C	MTASET P	ROFILES	WORKFLO	W EXECUT
BACK															D	ATA CLEA	NING W	ORKELOW
Dataset Profile Sample Data																		
RUN DATA PROFILING EDIT DATA PROFILING REF	ESH		Exe	cutio	n Re	sult												
Executions	Correlation Table	Correlation /	Matrix	Corre	elation 1	lable	Correl	ation Mc	atrix									
2020-07-30 17:05:39 IST	Correlation Table																	
	Correlation	Table																
	account_length		1.00	-0.02	-0.01	-0.00	0.03	-0.00	-0.01	0.01	-0.01	0.00	-0.01	0.00	0.00	0.01	0.00	-0.00
	area_code		-0.02	1.00	-0.00	-0.02	-0.02	-0.02	0.01	-0.01	0.01	0.00	0.01	0.00	-0.00	-0.01	-0.00	0.02
	number_vmail_mess	roges	-0.01	-0.00	1.00	0.00	0.00	0.00	0.02	-0.00	0.02	0.01	0.00	0.01	0.00	0.00	0.00	-0.01
	total_day_minutes		-0.00	-0.02	0.00	1.00	0.00	1.00	-0.01	0.01	-0.01	0.01	0.00	0.01	-0.02	-0.00	-0.02	0.00
	total_day_calls		0.03	-0.02	0.00	0.00	1.00	0.00	-0.00	0.00	-0.00	0.00	-0.01	0.00	0.01	0.01	0.01	-0.01
	total_day_charge		-0.00	-0.02	0.00	1.00	0.00	1.00	-0.01	0.01	-0.01	0.01	0.00	0.01	-0.02	-0.00	-0.02	0.00
	total_eve_minutes		-0.01	0.01	0.02	-0.01	-0.00	-0.01	1.00	0.00	1.00	-0.02	0.01	-0.02	0.00	0.01	0.00	-0.01
	total_eve_calls		0.01	-0.01	-0.00	0.01	0.00	0.01	0.00	1.00	0.00	0.00	-0.01	0.00	-0.01	0.01	-0.01	0.01
	total_eve_charge		-0.01	0.01	0.02	-0.01	-0.00	-0.01	1.00	0.00	1.00	-0.02	0.01	-0.02	0.00	0.01	0.00	-0.01

Dataset Profile - Telco Churn Prediction		WEW ALL DATASET PROFILES WORKFLOW EXECUTION DATA CLEANING WORKFLOW *
Encoderational Contractions (Entering) Encodings 2000/07/07/03/07/ Intering 2000/07/07/03/07/ Intering 2000/07/07/07/ 2000/07/07/07/ 2000/07/07/ 2000/07/ 2000/07/ 2000/07/ 2000/07/ 2000/ 20	Execution Result Overland Market Correlation Matrix	
	tratal_inti_rates	

8.1.7 Pipeline

Fire supports Pipelines. Pipelines allow running workflows in a defined order.

Pipeline List

The Pipeline tab displays the list of Pipelines for the current Application.

Application Dashboard - Analytics		
Workflows Datasets Dashboard Workflow Executi	ns Scheduled Share App Credentials Pipeline Pipeline Executions	
Pipleline List	Q Search Pipelines	CREATE PIPELINE
Filter By Category	DID NAME DESCRIPTION CATEGORY LAST UPDATED A	CTION
I≣ Show All Categories	1 Example Pipeline 01/07/2019 of 9:37PM	▶ ✔ ▶ @

Creating a Pipeline

Each Application now allows creating Pipelines.

Below is an example Pipeline with 3 Workflows.

pipeline Name: Example Pipeline	Category: Enter Category	Description: Enter Description	Parameters:
			SAVE CLEAR BACK EXECUTE
Q Search Nodes	←		
쇼 Workflow_1			
	Clickstream Data	Formers Markets on Geo Map	Graphs
	0 1	0 2	

Executing a Pipeline

Pipelines can be executed like workflows. When a Pipeline is executed, its execution is submitted to Airflow.

The Pipeline tab displays the list of Pipelines for the current Application.

Application Dashboard - Analytics		
Workflows Datasets Dashboard Workflow Execut	ons Scheduled Share App Credentials Pipeline Executions	
Pipleline List	Q Search Pipelines	CREATE PIPELINE
Filter By Category	DID NAME DESCRIPTION CATEGORY LAST UPDATED ACTION	
題 Show All Categories	🗌 1 Example Pipeline 01/07/2019 at 9:37PM 💿 🇨 🕨 🗃	

Clicking on the Execute Action icon opens the Pipeline Execute Page.

Execute Pipeline			
Pipeline Id: 0		1	EXECUTE
Pipeline Name: 0		Example Pipeline	
Email on success:			
Email on failure:			
Pipeline Result			
Pipeline Details			
Pipeline Id:	1		
Description:			
Pipeline Nodes:	PIPELINE NODES		
	Clickstream Data Analysis		
	Farmers Markets on Geo Map		
	DistributionGraphs		

Pipeline Execution

Once a Pipeline is fired, its details are visible in Pipeline Executions.

Pipe	line Task List						Q Search by Tas	kName
ID	TASK NAME	VIEW	CATEGORY	DESCRIPTION	START TIME	LAST UPDATED	DURATION(SEC)	STATUS
193	Clickstream Data Analysis	€ ≡ ∴ ≡ *	Analytics	Fired By Pipeline - Airfl	01/13/2019 at 6:08:15PM	01/13/2019 at 6:08:15PM	0	STARTING
194	Farmers Markets on Geo	€ ≡ ≞ ≡ *	Analytics	Fired By Pipeline - Airfl	01/13/2019 at 6:08:15PM	01/13/2019 at 6:08:15PM	0	STARTING
195	DistributionGraphs	¢≡∴≡*	Analytics	Fired By Pipeline - Airfl	01/13/2019 at 6:08:15PM	01/13/2019 at 6:08:15PM	0	STARTING

8.1.8 OCR with Tesseract

In order to run Tesseract, the below Installation steps have to be performed.

Download & Install the Tesseract Language Data files

• Download and Install the tesseract language data files for Version 3.X on each of the worker nodes of the cluster:

```
https://github.com/tesseract-ocr/tessdata/releases
wget https://github.com/tesseract-ocr/tessdata/archive/3.04.00.tar.gz
```

• Install them in the same directory on each of the worker nodes:

```
git clone https://github.com/tesseract-ocr/tessdata.git
```

Include TESSDATA_PREFIX in spark configs when submitting the job

• Include the following in spark submit configs when running workflows containing the OCR node:

--conf spark.executorEnv.TESSDATA_PREFIX=/home/ec2-user/tessdata

• Where the tesseract language data files are in /home/ec2-user/tessdata directory on each of the worker nodes

Error if TESSDATA_PREFIX is not set correctly

If TESSDATA_PREFIX is not set, the spark program would run into the error below:

```
Error opening data file /Users/saudet/projects/bytedeco/javacpp-presets/tesseract/

→cppbuild/macosx-x86_64/share/tessdata/eng.traineddata

Please make sure the TESSDATA_PREFIX environment variable is set to the parent_

→directory of your "tessdata" directory.

Failed loading language 'eng'

Tesseract couldn't load any languages!
```

The above error would be in the Job logs. If yarn is being used it would be in the yarn logs:

yarn logs -applicationId job_application_id

When the job is being executed, Fire displays the job_application_id in the browser.

CHAPTER 9

Web App User Guide

9.1 Analytical Apps User Guide

9.1.1 Creating Analytics App

Fire Insights enables you to create Analytics Apps.

Below is the process for creating a new Analytics App.

- Go to APPLICATIONS / ANALYTICS APPS
- Click on "Create Analytics App"
- Add mandatory fields i.e. "Name", "select notebook"
- Click on add stage button to add different stages
- Click "Save" Or "Publish"

Go to Analytics Apps

When you go to ANALYTICS APPS under APPLICATIONS all existing analytics app are displayed. Where you can EDIT, VIEW and DELETE existing analytics app.

Fire Insights 🛛 🖓 DATA BROWSERS	 III APPLICATIONS 	- O SCHEDULER - O EXECUTION	- 💩 MODELS - 🛢	DATA QUALITY - 🛛 ADMINE	STRATION - PROCESSORS -		≪- ⊖- ≡- 4
NALYTICS APPLICATIONS							CREATE ANALYTICS APP
TPA Propensity Model It is used to predict the behaviour of your customer or prospect base.	* 8	Balanced Score Card	88	Upload File	88	Sales Analytics	8 0
5010	-	publish	-	5040	-	50V0	-
D&BNotebook test	60			Edir View Delete			
publish							

Click on Create Analytics App

Fill in the required fields as below.

- Name : Name of the new analytics app
- category : Category of the new analytics app
- Description : Description of the new analytics app
- Execution Type: : Select execution type i.e notebook and select notebook from the available notebook list

IAME*		CATEGORY:		BPTION:	
XECUTION TYPE*	~	NoteBookIntegration	~	PUBLISH	SAVE
TAGES:					
Uplood File 🔹 🖋	Addressable Market	@ / O			

"Save" or "Publish" the analytics app before adding stages.

9.1.2 Adding Stages

Click on "Add stages" button to add a new stage. Select stage type and enter the stage name.

		Add stages		×		
DIT ANALYTICS APPLICATION		STAGE TYPE: 0	Run	~		
NAME*		STAGE NAME: 0	Run		DESCRIPTION:	
		1				
EXECUTION TYPE*				OK CANCEL		
	~	NoteBookIntegration		~	PUBLISH	SAVE
STADES:						
Uplood File 💿 🥒 🔘	Addressable Marks		•			

- You can rearrange the stages by dragging and dropping.
- You can EDIT, VIEW and REMOVE stages.

Examples for adding various Stages

1 : Upload Stage

- In upload stage we will first add column component and divide in two columns
- In first column add file component to choose files to upload to databricks. In this component in File tab in "STORAGE" select "Base64"

ALANCED SCORE CARD		O SCHEDULER - O DECUTIONS - & D	NODELS - 🖀 DATA QUALITY -		• PROCESSONS •	C+ O+ E-
lacoh Huli() GNIAA, UNOUT E COLIMINE E RADIET E RADIET E RADIET E Date D Val.	2	Drag and Drag a form component			Drug and Drug a form component	
ire Insights 2) DATA BROW						
		O SCHEDULER - O EXECUTIONS - 💩 A	MODELS + 🖀 DATA QUALITY +	O ADMINISTRATION	• Recessors •	
LANCED SCORE CARD worth field() contract. 27 Into Albumar 28 Into Albumar 28 Into Albumar 29 Contract.		C ROHERAN - C CECUTION - C)	See		B INSCISION Drig and Drug a from component	C+ 0+ E+

• In other column we will add one textfield to add "DESTINATION PATH" where the browse file should get uploaded. Set its property name to destinationPath.

Fire Insights 🛛 🖓 DATA BROWSER	S - III APPLICATIONS - O SCHEDULER - G	EXECUTIONS - 💩 MODELS - 🛢 DATA QUALITY -	O ADMINISTRATION - PROCESSORS -	¢;- 0- Ⅲ- ≜-
BALANCED SCORE CARD :	UPLOAD FILE			DONE SAVE BACK
Search field(s)	UPLOAD		DESTINATION PATH	
GENERAL	File Name	Size		
>_ TEXT FIELD	Drop f	les to attach, or browse		
A TEXTAREA				
C CHECKBOX				
III SELECT				
© маю ∎ rez				
E BUTTON				
LAYOUT				

- Add upload button and set action to event. Set the button event name to upload.
- Also add next button to go to next stage and perform actions depending upon event. Set the event name as next for the next button.

Click on "DONE" or "SAVE" to save added components for that stage

2 : Parameters Stage

• In parameters stage we can add select, text-field, select boxes, buttons etc components

For example:

- First we will add column component and divide it in two columns
- Then, lets add select boxes example in first column by adding select boxes component. In this component in Data tab add all possible values you want to add.

🌢 Fire Insights 👘	Button Component	-		6	Hep ^X ≜ d+ 0+ ■+ ▲+
BALANCED SCC	Display API Conditional Logic Layout	Preview			SAVE BACK
	LABEL Ø	UPLOAD			
Search field(ii)	Upload Action @				
GENERAL	Event × *	SINE CHICE	REMOVE		
> TEXT FIELD <>	□ SHOW VALIDATIONS @				
А ТЕХТАНКА	BUTTON EVENT ()				
D SELECT BORES	THEME ()				
O RADIO	Secondary × •				
E PLE E SUTTON	sate 0 Medium × •				
LAYOUT	□ BLOCK BUTTON ⊕				
	LEFTICON @				
	Enter Icon classes				
	SHORTCUT ()				
					*
Fire Insights 6) DATA BROWSERS - IIII APPLICATIONS - 🔿 SCHEDULER - 🏵 EXECUTIONS - 🗞 MODELS -			35065 -	ot- ⊕- ≣- ≛-
-					
BALANCED SCC	RE CARD : UPLOAD FILE				DONE SAVE BACK
Search field(s)					
GENERAL	UPLOAD		DESTINATION PATH		
> TEXT RELD	File Name	Size	STORE -		
A TEXTAREA	Drop files to attoch, or browse				
III SELECT					
O MORO					
LAYOUT					
DATODI					
🌢 Fire Insights 👘	Button Component	-			× · · · · · ·
BALANCED SCO	Display API Conditional Logic Layout	Preview			
DALANCED SCC	LABEL 0 -	NEXT			SAVE BACK
Search field(ii)	Nest				
GENERAL	ACTION®	-			
> TEXT FIELD	Event × •	SAUE CANCEL	REMOVE		
A TEXT AREA	BUTTON EVENT @				
	next				
BRLECT BORRS BELLECT O RADIO	THEME () Secondary × •				
li na	\$2E 0				
LAYOUT	Medium X ¥				
	LEFT ICON ®				
	Enter icon classes				
	NONTICON Ø				
	Enter icon classes				
	······································				
🕑 Fire Insights 🛛 d) data browsers - III Applications - O scheduler - O executions - 🔊 models -	🛢 DATA QUALITY 🕒		35045 -	¢+ ⊖+ ≣+ ≜+
BALANCED SCC	RE CARD : UPLOAD FILE				DONE SAVE BACK
Search field(s)	UPLOAD		DESTINATION PATH		
GENERAL	File Name	Size			
> TEXT FIELD <pre> HTML ELEMENT </pre>	Drop files to attach, or browse		URIOND		
A TEXTAREA					
SELECT BORES					
III SELECT © MORD					
na na					
LAYOUT					
Fire Insights 6	2) DATA BROWSERS • III APPLICATIONS • ⊘ SCHEDULER • @ EXECUTIONS • S MODELS •	DATA QUALITY -	O ADMINISTRATION - PROCI	35045 -	¢: e- ≡- ≛-
BALANCED SCC	RE CARD : ADDRESSABLE MARKET				DONE SAVE BACK
Search field(s)	SELECT INDUSTRIES				
GENERAL				Drog and Drop a form component	
> TEXT FIELD <2> HTML ELEMENT	MANUFACTURING TRANSPORATION BEWICES				
А ТЕХТАНКА					
B SELECT BORES	E				
■ 9814CT © 84060					
E PLE BUTTON					
LAYOUT					

• Then, lets add select example in the second column by adding select component. In this component in Data tab add all possible values you want to add.

Search field(s)			
GENERAL	SELECT INDUSTRIES	REGION	•
> TEXT FIELD	CONSTRUCTION		
W HTML ELEMENT	TRANSPORATION		
A TEXTABLE			
CHECKBOX			
BELICT BORES			
HE SELECT			
0 MORO			
LAYOUT			

- Now, lets add column component in the bottom and divide into two columns for adding back and next button.
- Add back button in first column to go to back stage and perform actions depending upon event, where we will add event name as back.

🌢 Fire Insights	Button Component		d ⁰ Hep [×]
BALANCED SCO	Display API Conditional Logic Layour	Preview	SAVE BACK
	LABEL @*	MACK	
	Back	_	
	ACTION @		•
GENERAL	Event x +	MUE CANCEL REMOVE	
> TEXT PIELD	SHOW VALIDATIONS @		
45 HTML ELEMENT			
A TEXTAREA	BUTTON EVENT ()		
	back		
SELECT BORES	THEME (0		
© MOD	Primary × •		
B ns	202 Q		1 No. 10
B BUTTON	Medum × •		
LAYOUT	BLOCK BUTTON		
	LEFT ICON (0		
	Enter icon classes		
	RISHT ICON Ø		
	Enter Icon classes		
	SHORTCUT @		
	· · ·		

• Add next button in second column to go to next stage and perform actions depending upon event. Set its event name as next. We can also add CUSTOM CSS CLASS like float-right, float-left etc

Fire Insights	Button Component	6 ⁷ He0 [×] [±] 6 ⁴ = - ≜-
BALANCED SCC	Digity API Conditional Logic Layout	Preview axx axx
-		B [™] Hep [×]
	Ener ton stores Sen (CoO) Ener ton stores Sentor 0 Sentor 10	

Click on "DONE" or "SAVE" to save the added components for that stage.

3 : Run Stage

- In run stage we will execute the notebook with all parameters added in the App.
- Let's first add title in page if needed with "html element" component like below.

Fire Insights	2) DATA BROWSERS		O SCHEDULER -	© EXECUTIONS -	& MODELS -	DATA QUALITY -		PROCESSORS +	o(+ ⊕+ ⊞+ ∆+
BALANCED S	CORE CARD : ADI	DRESSABLE MAR	KET						DORE SHE BACK
Search field(s) Contant. 1. Too Factor 0. The BLANKA Conceptor 1. Too Factor 1. Too Factor		SELECT INDUSTRIES INTURAL RESOUR CONSTRUCTIONS TRAINFORMING SERVICES	25				REGION		
Fire Insights	HTML Element C4	omponent							g0mg× 4 4 0 ≣ 4.
BALANCED S	LABEL @ • HTML	ionditional Logic Layo	e.			Preview RUN			SAVE BACK
CENERAL > TEXT PRED V HTML ELEMENT A TEXT AREA	page-title mb-2					MR CH	CEL REMOVE		
	ATTRIBUTES @ ATTRIBUTE + ADD ANOTHER		WLUE		0				
LAYOUT	CONTENT @								
	CUSTOM CSS CLA	55 Q							
	Custom CSS Cite								•

- Now, lets add column component in the bottom and divide it into two columns for adding the back and run buttons.
- Add back button in first column to go to back stage and perform actions depending upon event. Set its event name as back.
- Add next button in second column to go to next stage and perform actions depending upon event. Set its add event name as execute. We can also can add CUSTOM CSS CLASS like float-right, float-left etc

🌢 Fire Insights 🛛	Button Component		e ^p +wo [×]	- ⁰⁶ -	0- E	8- A-
BALANCED SCO	Doplay API Conditional Logic Layout	Preview				
	LABEL @*		8.8			
	Run ACTION @					
GENERAL	Event × •	SARE CANCEL BEMONE				
> TIDO PILD	C SHOW VALIDATIONS @					
А ТОТАНА	BUTTON EVENT @					
D SELECT BOXES	THEME ()					
O RADIO	Primary X 💌					
B FLE BUTTON	52E Ø					
LAYOUT						
	LEFT ICON (0)					
	Enter icon classes					
	RIGHT ICON ()					
	SHORTCUT @					
	· · ·			•		

Click on "DONE" or "SAVE" to save added components for that stage

9.1.3 Integrating with Databricks Notebook

The Web App in Fire Insights can trigger a Notebook in Databricks.

Fire Insights passes 2 parameters to the Notebook:

• postback-url

ALANCED SCORE CA				I - 🗟 DATA QUALITY -		CC+ ⊕- ■-
	I #	IUN				
GENERAL		MOX .				8.8
> TEXT FIELD						_
ITTAL ELEMENT						
A TOTARA						
Снескох						
SELECT DOKES						
E RELICT						
0 MARO						
na 🗎						
BUTTON						
LAYOUT						

• job-id

Add wheel file to your Databricks Notebook

Add the wheel file to your Databricks Notebook. This is to enable using the Fire Insights API's for sending data to it.

Outputing details to Fire Insights

The Databricks Notebook can output text, tables and charts to be dispalyed in Fire Insights.

Below are the examples for it.

Create a RestWorkflowContext Object

First create a RestWorkflowContext for communicating with Fire Insights Server

```
jobId = dbutils.widgets.get("job-id")
webserverURL = dbutils.widgets.get("postback-url")
print(webserverURL)
print(jobId)
from fire_notebook.output.workflowcontext import RestWorkflowContext
restworkflowcontext = RestWorkflowContext(webserverURL, jobId)
```

Outputing Text

Below is how to output text to Fire Insights

```
restworkflowcontext.outStr(9, "Test String")
```

Outputing PySpark Dataframe as Table

The below code outputs the contents of PySpark Dataframe to Fire Insights as a table

from pyspark.sql.types import *
schema = StructType([StructField("c1", DoubleType()))\

(continues on next page)

(continued from previous page)

```
,StructField("c2", IntegerType())])
test_list = [[0.0, 2], [1.0, 4], [2.0, 8], [3.0, 16], [4.0, 32], [5.0, 64], [6.0, 
→128]]
df = spark.createDataFrame(test_list,schema=schema)
restworkflowcontext.outDataFrame(9, "PySpark Dataframe", df)
```

Outputing Pandas Dataframe as Table

The below code outputs the contents of Pandas Dataframe to Fire Insights as a table

```
# list of strings
lst = ['Geeks', 'For', 'Geeks', 'is',
                          'portal', 'for', 'Geeks']
# Calling DataFrame constructor on list
df = pd.DataFrame(lst, columns=['name'])
print(df)
restworkflowcontext.outPandasDataframe(9, "Names", df)
```

Outputing CHART

Output the chart in fire by selecting x & y column and Different type of chartType: COLUMNCHART, BARCHART & LINECHART

from pyspark.sql.types import *

schema = StructType([StructField("c1", DoubleType()) ,StructField("c2", IntegerType())])

test_list = [[0.0, 2], [1.0, 4], [2.0, 8], [3.0, 16], [4.0, 32], [5.0, 64], [6.0, 128]]

df = spark.createDataFrame(test_list,schema=schema)

restworkflowcontext.outDataframeChart(title= "Example Chart", x_column = "c1", y_columns = ["c2"], chart_type = "LINECHART", df = df, numRowsToDisplay = 10)

Outputing HTML

Below is how to output html to Fire Insights

```
htmlstr1 = "<h3>You can view HTML code in notebooks.</h3>"
restworkflowcontext.outHTML(9, title="Example HTML", text = htmlstr1)
```

Outputing Plotly

Below is how to output plotly to Fire Insights

```
import plotly.graph_objs as go
import plotly
```

(continues on next page)

(continued from previous page)

9.1.4 Running Analytics App

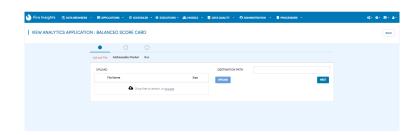
Once the Analytics App has been created, they can be executed.

Below are the steps for executing an Analytics App.

Click on Analytics App Name

ANALYTICS APPLICATIONS							
							CREATE ANALYTICS APP
TPA Propensity Model It is used to predict the behaviour of your customer or prospect base.	¢0	Balanced Score Card	#8 80	Upload File	#8 80	Sales Analytics	#0 00
10/10	-	5010	-	1010	-	9299	-
D&BNotebook test	#0 00						
publish	-						

Go through the various Stages



Examples of the various Stage Pages

1: Upload

- Browse files you want to upload to databricks.
- Add destination path of dbfs where you want to upload choose file.
- If added path is not there in dbfs then it will first create the folder in dbfs and then upload the file.
- Then, click on upload button to upload to DBFS and see the csv file data in tabular format.

Fire Insights 2 DATA BROWSERS	- III APPLICATIONS - O SCHEDULER - O EXECUTIONS -	& MODELS - 🛢 DATA QUALITY - 😁 ADMINISTRATION - 🛢 PROCESSORS -	¢- e- ≡- ≜-
VIEW ANALYTICS APPLICATIO	N : BALANCED SCORE CARD		BACK
	Upload File Addressable Morket Run		
	UPLOAD File Nome		
	X cont.cov	S2# GRCMD NOT 2/020 Bytes	

IEW ANALYTICS APPLICA	TION : BALANCED S	C					840
		n	0.0	2.3	3.0		
		22	1.0	3.0	2.0222		
		33	0.0	u	1.0		
	Uplood File Address	4	0.0	41	6.0		
	UPLOAD	6	0.0	3.1	6.0		
		6	1.0	2.1	2.0		
	File Nome	4	0.0	41	6.0		EXT
	× cons.csv	6	0.0	3.1	6.0		
		6	1.0	2.1	2.0		
		4	0.0	40	5.0		
		6	0.0	2.1	6.0		
		6	1.0	2.1	2.0		
		4	0.0	40	6.0		
		5 C	0.0	3.1	6.0		
		66	1.0	2.1	2.0		
						ox	
		_					

AME IS DIRECT	DRY	DIZ14	92	ACTION
fears.cov. foise		/hest/cons.cov	20	0

- You can browse dbfs and check if the file uploaded successfully.
- Click on "NEXT" button to go to next stage.

2 : Parameters

- Select the parameters of your interest
- If you click on "BACK" or "NEXT" button the selected value will remain as it is and you can change it if needed

VIEW ANALYTICS APPLICATION : BALANCED SCORE CARD	🔥 Fire Insights 🛛 🕲 DATA BROWSERS	- III APPLICATIONS - © SCHEDULER - © EXECUTION	- 💩 models - 🖀 data quality - 😝 a	OMINISTRATION - PROCESSORS -	¢+ ⊕+ ≡+ ≜+
Upie of Tre Addressed Motor Tere RELCT Notifies RELCT Notifi	VIEW ANALYTICS APPLICATION	DN : BALANCED SCORE CARD			BACK
S VIDAU HEORES ORIGINAL OF CONTRACTORS ORIGIN					
		NATURAL RESOURCES CONSTRUCTION MANUFACTURING TRANSPORATION SERVICES	RESION		

• Click on "NEXT" button to move to next page

3 : Run

- In this stage you will execute the Analytics App with the added parameters in the earlier stages.
- You can click on back button and change the value and run Analytics App again.
- Click on "RUN" button to execute the app and view the results.

C C C C C C C C C C C C C C C C C C C	• •• =• A•
Cipical Fix Addressible Mohae Exc RUN acc	BACK
RUN	
Genotion stands and Resource Reserved (r/ww.j/s227/writer.j/s26758)	

CHAPTER 10

Data Science

10.1 Machine Learning User Guide

10.1.1 Feature Generation

Feature generation is the process of creating new features from one or multiple existing features, potentially for using in statistical analysis. This process adds new information to be accessible during the model construction and therefore hopefully result in more accurate model.

eration. These merado.	
Title	Description
DateToAge	Convert Date to Age
CaseWhen	Based on the value, convert it to another value
Scala	Write Scala code in Spark for generating new Features
SQL	Write SQL code for generating new features
StopWOrdRemover	Removes Stop Words
Tokenizer	Tokenizes a string into Tokens
OneHotEncoder	Applies one hot encoding
TF/IDF	Finds the TF and IDF
IndexString	Converts a column containg String to numeric values

Table 1: Fire Insights provides a number of processors for Feature Generation. These include:

10.1.2 Feature Selection

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, in the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for several reasons:

- simplification of models to make them easier to interpret by researchers/users
- shorter training times

- to avoid the curse of dimensionality
- enhanced generalization by reducing overfitting (formally, reduction of variance)
- https://en.wikipedia.org/wiki/Feature_selection

Apache Spark has the following Feature Selectors. Fire Insights provides them as Processors to be easily used in the workflows:

Feature Selection Processors in Fire Insights

Title	Description
VectorSlicer	VectorSlicer is a transformer that takes a feature vector and outputs a new feature
	vector with a sub-array of the original features. It is useful for extracting fea-
	tures from a vector column. VectorSlicer accepts a vector column with specified
	indices, then outputs a new vector column whose values are selected via those
	indices.
RFormula	RFormula selects columns specified by an R model formula. RFormula produces
	a vector column of features and a double or string column of label. Like when
	formulas are used in R for linear regression, string input columns will be one-hot
	encoded, and numeric columns will be cast to doubles. If the label column is of
	type string, it will be first transformed to double with StringIndexer. If the label
	column does not exist in the DataFrame, the output label column will be created
	from the specified response variable in the formula.
ChiSqSelector	ChiSqSelector stands for Chi-Squared feature selection. It operates on labeled
	data with categorical features. ChiSqSelector uses the Chi-Squared test of inde-
	pendence to decide which features to choose. It supports five selection methods:
	numTopFeatures, percentile, fpr, fdr, fwe

Table 2: Apache Spark based Feature Selection Processors in Fire Insights

More details regarding the Feature Selectors in Spark can be found at:

https://spark.apache.org/docs/2.2.0/ml-features.html#feature-selectors

- VectorSlicer
- RFormula
- ChiSqSelector

VectorSlicer

VectorSlicer is a transformer that takes a feature vector and outputs a new feature vector with a sub-array of the original features. It is useful for extracting features from a vector column. VectorSlicer accepts a vector column with specified indices, then outputs a new vector column whose values are selected via those indices. There are two types of indices,

Integer indices that represent the indices into the vector, setIndices().

String indices that represent the names of features into the vector, setNames(). This requires the vector column to have an AttributeGroup since the implementation matches on the name field of an Attribute.

Specification by integer and string are both acceptable. Moreover, you can use integer index and string name simultaneously. At least one feature must be selected. Duplicate features are not allowed, so there can be no overlap between selected indices and names. Note that if names of features are selected, an exception will be thrown if empty input attributes are encountered.

RFormula

RFormula selects columns specified by an R model formula. Currently Spark supports a limited subset of the R operators, including '~', '.', ':', '+', and '-'. The basic operators are:

- ~ separate target and terms
- – concat terms, "+ 0" means removing intercept
- – remove a term, "- 1" means removing intercept
- : interaction (multiplication for numeric values, or binarized categorical values)
- . all columns except target

Suppose a and b are double columns, we use the following simple examples to illustrate the effect of RFormula:

- $y \sim a + b$ means model $y \sim w0 + w1 * a + w2 * b$ where w0 is the intercept and w1, w2 are coefficients.
- $y \sim a + b + a:b 1$ means model $y \sim w1 * a + w2 * b + w3 * a * b$ where w1, w2, w3 are coefficients.

RFormula produces a vector column of features and a double or string column of label. Like when formulas are used in R for linear regression, string input columns will be one-hot encoded, and numeric columns will be cast to doubles. If the label column is of type string, it will be first transformed to double with StringIndexer. If the label column does not exist in the DataFrame, the output label column will be created from the specified response variable in the formula.

ChiSqSelector

ChiSqSelector stands for Chi-Squared feature selection. It operates on labeled data with categorical features. ChiSqSelector uses the Chi-Squared test of independence to decide which features to choose. It supports five selection methods: numTopFeatures, percentile, fpr, fdr, fwe. * numTopFeatures chooses a fixed number of top features according to a chi-squared test. This is akin to yielding the features with the most predictive power. * percentile is similar to numTopFeatures but chooses a fraction of all features instead of a fixed number. * fpr chooses all features whose p-values are below a threshold, thus controlling the false positive rate of selection. * fdr uses the Benjamini-Hochberg procedure to choose all features whose false discovery rate is below a threshold. * fwe chooses all features whose p-values are below a threshold. The threshold is scaled by 1/numFeatures, thus controlling the family-wise error rate of selection. By default, the selection method is numTopFeatures, with the default number of top features set to 50. The user can choose a selection method using setSelectorType.

10.1.3 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

• https://en.wikipedia.org/wiki/Cluster_analysis

Clustering Processors in Fire Insights

Title	Description
Gaussian Mixture	A Gaussian Mixture Model represents a composite dis-
	tribution whereby points are drawn from one of k
	Gaussian sub-distributions, each with its own probabil-
	ity. The spark.ml implementation uses the expectation-
	maximization algorithm to induce the maximum-
	likelihood model given a set of samples.
KMeans	k-means is one of the most commonly used clustering
	algorithms that clusters the data points into a predefined
	number of clusters. The MLlib implementation includes
	a parallelized variant of the k-means++ method called
	kmeansll.
LDA	LDA is implemented as an Estimator that supports both
	EMLDAOptimizer and OnlineLDAOptimizer, and gen-
	erates a LDAModel as the base model.

Table 3: Apache Spark ba	sed Clustering Processors	in Fire Insights
real real real real real real real real	8	

Table 4: H2O based Clustering Processors in Fire Insights

Title	Description
KMeans	K-Means falls in the general category of clustering al-
	gorithms. Clustering is a form of unsupervised learning
	that tries to find structures in the data without using any
	labels or target values. Clustering partitions a set of ob-
	servations into separate groupings such that an observa-
	tion in a given group is more similar to another obser-
	vation in the same group than to another observation in
	a different group.

Clustering Algorithms in Apache Spark

Apache Spark provides a number of Algorithms for Clustering.

https://spark.apache.org/docs/latest/ml-clustering.html

- K-means
- Latent Dirichlet allocation (LDA)
- Bisecting k-means
- Gaussian Mixture Model (GMM)
- Power iteration clustering (PIC)
- Streaming k-means

K-means

https://spark.apache.org/docs/latest/ml-clustering.html#k-means

k-means is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters. The MLlib implementation includes a parallelized variant of the k-means++ method called kmeans||. The implementation in spark.mllib has the following parameters:

k is the number of desired clusters. Note that it is possible for fewer than k clusters to be returned, for example, if there are fewer than k distinct points to cluster. - maxIterations is the maximum number of iterations to run. - initializationMode specifies either random initialization or initialization via k-meansll. - runs This param has no effect since Spark 2.0.0. - initializationSteps determines the number of steps in the k-meansll algorithm. - epsilon determines the distance threshold within which we consider k-means to have converged. - initialModel is an optional set of cluster centers used for initialization. If this parameter is supplied, only one run is performed.

Latent Dirichlet allocation (LDA)

https://spark.apache.org/docs/latest/ml-clustering.html#latent-dirichlet-allocation-lda

LDA is implemented as an Estimator that supports both EMLDAOptimizer and OnlineLDAOptimizer, and generates a LDAModel as the base model. Expert users may cast a LDAModel generated by EMLDAOptimizer to a DistributedLDAModel if needed.

Latent Dirichlet allocation (LDA) is a topic model which infers topics from a collection of text documents. LDA can be thought of as a clustering algorithm as follows:

- Topics correspond to cluster centers, and documents correspond to examples (rows) in a dataset.
- Topics and documents both exist in a feature space, where feature vectors are vectors of word counts (bag of words).
- Rather than estimating a clustering using a traditional distance, LDA uses a function based on a statistical model of how text documents are generated.

LDA supports different inference algorithms via setOptimizer function. EMLDAOptimizer learns clustering using expectation-maximization on the likelihood function and yields comprehensive results, while OnlineLDAOptimizer uses iterative mini-batch sampling for online variational inference and is generally memory friendly.

LDA takes in a collection of documents as vectors of word counts and the following parameters (set using the builder pattern):

- k: Number of topics (i.e., cluster centers)
- optimizer: Optimizer to use for learning the LDA model, either EMLDAOptimizer or OnlineLDAOptimizer
- docConcentration: Dirichlet parameter for prior over documents' distributions over topics. Larger values encourage smoother inferred distributions.
- topicConcentration: Dirichlet parameter for prior over topics' distributions over terms (words). Larger values encourage smoother inferred distributions.
- maxIterations: Limit on the number of iterations.
- checkpointInterval: If using checkpointing (set in the Spark configuration), this parameter specifies the frequency with which checkpoints will be created. If maxIterations is large, using checkpointing can help reduce shuffle file sizes on disk and help with failure recovery.

All of spark.mllib's LDA models support:

- describeTopics: Returns topics as arrays of most important terms and term weights
- topicsMatrix: Returns a vocabSize by k matrix where each column is a topic

Bisecting k-means

Bisecting K-means can often be much faster than regular K-means, but it will generally produce a different clustering.

Bisecting k-means is a kind of hierarchical clustering. Hierarchical clustering is one of the most commonly used method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

- Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Bisecting k-means algorithm is a kind of divisive algorithms. The implementation in MLlib has the following parameters:

- k: the desired number of leaf clusters (default: 4). The actual number could be smaller if there are no divisible leaf clusters.
- maxIterations: the max number of k-means iterations to split clusters (default: 20)
- minDivisibleClusterSize: the minimum number of points (if >= 1.0) or the minimum proportion of points (if < 1.0) of a divisible cluster (default: 1)
- seed: a random seed (default: hash value of the class name)

Gaussian mixture

A Gaussian Mixture Model represents a composite distribution whereby points are drawn from one of k Gaussian sub-distributions, each with its own probability. The spark.mllib implementation uses the expectation-maximization algorithm to induce the maximum-likelihood model given a set of samples. The implementation has the following parameters:

- k is the number of desired clusters.
- convergenceTol is the maximum change in log-likelihood at which we consider convergence achieved.
- maxIterations is the maximum number of iterations to perform without reaching convergence.
- initialModel is an optional starting point from which to start the EM algorithm. If this parameter is omitted, a random starting point will be constructed from the data.

Power iteration clustering (PIC)

Power iteration clustering (PIC) is a scalable and efficient algorithm for clustering vertices of a graph given pairwise similarities as edge properties, described in Lin and Cohen, Power Iteration Clustering. It computes a pseudoeigenvector of the normalized affinity matrix of the graph via power iteration and uses it to cluster vertices. spark.mllib includes an implementation of PIC using GraphX as its backend. It takes an RDD of (srcId, dstId, similarity) tuples and outputs a model with the clustering assignments. The similarities must be nonnegative. PIC assumes that the similarity measure is symmetric. A pair (srcId, dstId) regardless of the ordering should appear at most once in the input data. If a pair is missing from input, their similarity is treated as zero. spark.mllib's PIC implementation takes the following (hyper-)parameters:

- k: number of clusters
- maxIterations: maximum number of power iterations

• initializationMode: initialization model. This can be either "random", which is the default, to use a random vector as vertex properties, or "degree" to use normalized sum similarities.

Streaming k-means

When data arrive in a stream, we may want to estimate clusters dynamically, updating them as new data arrive. spark.mllib provides support for streaming k-means clustering, with parameters to control the decay (or "forgetfulness") of the estimates. The algorithm uses a generalization of the mini-batch k-means update rule. For each batch of data, we assign all points to their nearest cluster, compute new cluster centers, then update each cluster

10.1.4 Regression

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common form of regression analysis is linear regression, in which a researcher finds the line (or a more complex linear function) that most closely fits the data according to a specific mathematical criterion.

• https://en.wikipedia.org/wiki/Regression_analysis

Apache Spark

Title	Description
Linear regression	LinearRegression analysis is a set of statistical pro-
	cesses for estimating the relationships between a depen-
	dent variable and one or more independent variables.
Generalized linear regression	Contrasted with linear regression where the output is
	assumed to follow a Gaussian distribution, generalized
	linear models (GLMs) are specifications of linear mod-
	els where the response variable Yi follows some distri-
	bution from the exponential family of distributions
Decision tree regression	Decision trees and their ensembles are popular methods
	for the machine learning tasks of classification and re-
	gression. Decision trees are widely used since they are
	easy to interpret, handle categorical features, extend to
	the multiclass classification setting, do not require fea-
	ture scaling, and are able to capture non-linearities and
	feature interactions.
Random forest regression	Random forests are ensembles of decision trees. Ran-
	dom forests combine many decision trees in order to re-
	duce the risk of overfitting.
Gradient-boosted tree regression	Gradient-Boosted Trees (GBTs) are ensembles of deci-
	sion trees. GBTs iteratively train decision trees in order
	to minimize a loss function.
Survival regression	Survival Analysis is a set of statistical tools, which ad-
	dresses questions such as 'how long would it be, before
	a particular event occurs'; in other words we can also
	call it as a 'time to event' analysis.
Isotonic regression	Isotonic regression is the technique of fitting a free-
	form line to a sequence of observations under the fol-
	lowing constraints: the fitted free-form line has to be
	non-decreasing everywhere, and it has to lie as close to
	the observations as possible.

Table 5: Apache Spark based Regression Processors in Fire Insights

Regression Algorithms in Apache Spark

https://spark.apache.org/docs/latest/ml-classification-regression.html#regression

- Linear regression
- Decision tree regression
- Random Forest regression
- Gradient-boosted tree regression
- Survival regression
- Isotonic regression

Scikit Learn

Title	Description
Ridge regression	Ridge regression addresses some of the problems of Or-
	dinary Least Squares by imposing a penalty on the size
	of the coefficients. The ridge coefficients minimize a
	penalized residual sum of squares
Lasso regression	The Lasso is a linear model that estimates sparse coeffi-
	cients. It is useful in some contexts due to its tendency
	to prefer solutions with fewer non-zero coefficients, ef-
	fectively reducing the number of features upon which
	the given solution is dependent.
Gradient Boosting regression	GB builds an additive model in a forward stage-wise
	fashion; it allows for the optimization of arbitrary differ-
	entiable loss functions. In each stage a regression tree is
	fit on the negative gradient of the given loss function
Random forest regression	A random forest is a meta estimator that fits a number of
	classifying decision trees on various sub-samples of the
	dataset and uses averaging to improve the predictive ac-
	curacy and control over-fitting. The sub-sample size is
	always the same as the original input sample size but the
	samples are drawn with replacement if bootstrap=True
	(default).

Table 6: Scikit Learn based Regression Processors in Fire Insights

Regression Algorithms in Scikit Learn

https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification

- Ridge regression
- Lasso regression
- Gradient Boosting regression
- Random Forest regression

Linear Regression

The interface for working with linear regression models and model summaries is similar to the logistic regression case.

When fitting LinearRegressionModel without intercept on dataset with constant nonzero column by "l-bfgs" solver, Spark MLlib outputs zero coefficients for constant nonzero columns. This behavior is the same as R glmnet but different from LIBSVM.

Generalized linear regression

Contrasted with linear regression where the output is assumed to follow a Gaussian distribution, generalized linear models (GLMs) are specifications of linear models where the response variable Yi follows some distribution from the exponential family of distributions.

Spark's GeneralizedLinearRegression interface allows for flexible specification of GLMs which can be used for various types of prediction problems including linear regression, Poisson regression, logistic regression, and others.

Decision tree regression

Decision trees are a popular family of classification and regression methods.

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Random Forest Regression

Random forests are a popular family of classification and regression methods.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Gradient - boosted Tree Regression

Gradient-boosted trees (GBTs) are a popular regression method using ensembles of decision trees.

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Survival Regression

In spark.ml, we implement the Accelerated failure time (AFT) model which is a parametric survival regression model for censored data. It describes a model for the log of survival time, so it's often called a log-linear model for survival analysis. Different from a Proportional hazards model designed for the same purpose, the AFT model is easier to parallelize because each instance contributes to the objective function independently.

Isotonic Regression

Isotonic regression or monotonic regression is the technique of fitting a free-form line to a sequence of observations under the following constraints: the fitted free-form line has to be non-decreasing (or non-increasing) everywhere, and it has to lie as close to the observations as possible.

Isotonic regression has applications in statistical inference. For example, one might use it to fit an isotonic curve to the means of some set of experimental results when an increase in those means according to some particular ordering is expected. A benefit of isotonic regression is that it is not constrained by any functional form, such as the linearity imposed by linear regression, as long as the function is monotonic increasing.

Another application is nonmetric multidimensional scaling, where a low-dimensional embedding for data points is sought such that order of distances between points in the embedding matches order of dissimilarity between points. Isotonic regression is used iteratively to fit ideal distances to preserve relative dissimilarity order.

Software for computing isotone (monotonic) regression has been developed for the R statistical package, the Stata statistical package and the Python programming language

10.1.5 Classification

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (subpopulations) a new observation belongs.

 https://en.wikipedia.org/w/index.php?search=Clssification+in+machine+learning&title=Special%3ASearch& go=Go&ns0=1

Apache Spark MLlib

Title	Description
Logistic Regression	Logistic regression is a popular method to predict a cat-
	egorical response. It is a special case of Generalized
	Linear models that predicts the probability of the out-
	comes.
Decision tree classifier	Decision trees and their ensembles are popular methods
	for the machine learning tasks of classification and re-
	gression. Decision trees are widely used since they are
	easy to interpret, handle categorical features, extend to
	the multiclass classification setting, do not require fea-
	ture scaling, and are able to capture non-linearities and
	feature interactions.
Random forest classifier	Random forests are ensembles of decision trees. Ran-
	dom forests combine many decision trees in order to re-
	duce the risk of overfitting.
Gradient-boosted tree classifier	Gradient-Boosted Trees (GBTs) are ensembles of deci-
	sion trees. GBTs iteratively train decision trees in order
	to minimize a loss function.
Multilayer perceptron classifier	Multilayer perceptron classifier (MLPC) is a classifier
	based on the feedforward artificial neural network.
Naive Bayes	Naive Bayes classifiers are a family of simple proba-
	bilistic, multiclass classifiers based on applying Bayes'
	theorem with strong (naive) independence assumptions
	between every pair of features.

Table 7: Apache Spark based Classification Processors in Fire Insights

Classification Algorithms in Spark MLlib

https://spark.apache.org/docs/latest/ml-classification-regression.html#classification

- Logistic Regression
- Decision tree classifier
- Random forest classifier
- Gradient-boosted tree classifier
- Multilayer perceptron classifier
- Linear Support Vector Machine
- One-vs-Rest classifier
- Naive Bayes

Scikit Learn

	=
Title	Description
Logistic Regression Classifier	In the multiclass case, the training algorithm uses the
	one-vs-rest (OvR) scheme if the 'multi_class' option
	is set to 'ovr', and uses the cross-entropy loss if the
	'multi_class' option is set to 'multinomial'.
Gradient Boosting classifier	GB builds an additive model in a forward stage-wise
	fashion; it allows for the optimization of arbitrary differ-
	entiable loss functions. In each stage n_classes _ regres-
	sion trees are fit on the negative gradient of the binomial
	or multinomial deviance loss function.
Random forest classifier	A random forest is a meta estimator that fits a number
	of decision tree classifiers on various sub-samples of the
	dataset and uses averaging to improve the predictive ac-
	curacy and control over-fitting. The sub-sample size is
	always the same as the original input sample size but the
	samples are drawn with replacement if bootstrap=True
	(default).

 Table 8: Scikit Learn based Classification Processors in Fire Insights

Classification Algorithms in Scikit Learn

https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

- Logistic Regression
- Gradient-boosting classifier
- Random Forest classifier

Fire Insights provides processors for the above Algorithms.

Logistic Regression

Logistic regression is a popular method to predict a categorical response. It is a special case of Generalized Linear models that predicts the probability of the outcomes. In spark.ml logistic regression can be used to predict a binary outcome by using binomial logistic regression, or it can be used to predict a multiclass outcome by using multinomial logistic regression. Use the family parameter to select between these two algorithms, or leave it unset and Spark will infer the correct variant.

Multinomial logistic regression can be used for binary classification by setting the family param to "multinomial". It will produce two sets of coefficients and two intercepts.

When fitting LogisticRegressionModel without intercept on dataset with constant nonzero column, Spark MLlib outputs zero coefficients for constant nonzero columns. This behavior is the same as R glmnet but different from LIB-SVM.

Decision tree classifier

Decision tree learning is one of the predictive modeling approaches used in statistics, data mining and machine learning. It uses a decision tree to go from observations about an item to conclusions about the item's target value. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

Random forest classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

Gradient-boosted tree classifier

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed simultaneously with the more general functional gradient boosting perspective. It later introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

Multilayer perceptron classifier

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to refer to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Naive Bayes

In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models.

It remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

10.1.6 Prediction

Prediction is to identify data points purely on the description of another related data value. It is not necessarily related to future events but the used variables are unknown. Prediction derives the relationship between a thing you know and a thing you need to predict for future reference.

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.

The word "prediction" can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you're using machine learning to determine the next best action in a marketing campaign. Other times, though, the "prediction" has to do with, for example, whether or not a transaction that already occurred was fraudulent. In that case, the transaction already happened, but you're making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.

What is Prediction?

- Predicting the identity of one thing based purely on the description of another, related thing
- Not necessarily future events, just unknowns
- Based on the relationship between a thing that you can know and a thing you need to predict

Why are Predictions Important?

Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things – customer churn likelihood, possible fraudulent activity, and more. These provide the business with insights that result in tangible business value. For example, if a model predicts a customer is likely to churn, the business can target them with specific communications and outreach that will prevent the loss of that customer.

Predictor => Predicted

- When building a predictive model, you have data covering both
- When using one, you have data describing the predictor and you want it to tell you the predicted value

Usual Examples

- Predicting levels of sales that will result from a price change or advert.
- Predicting whether or not it will rain based on current humidity
- Predicting the colour of a pottery glaze based on a mixture of base pigments
- Predicting how far up the charts a single will go
- Predicting how much revenue a book of debt will bring

Techniques

Most prediction techniques are based on mathematical models:

- · Simple statistical models such as regression
- Non-linear statistics such as power series
- Neural networks, RBFs, etc
- All based on fitting a curve through the data, that is, finding a relationship from the predictors to the predicted

10.1.7 Model Evaluation

Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.

Evaluation Processors in Fire Insights

Title	Description
NodeRegressionEvaluator	Evaluator for regression, which expects two input
	columns: prediction and label. Regression analysis is
	used when predicting a continuous output variable from
	a number of independent variables.
NodeBinaryClassificationEvaluator	Evaluator for binary classification, which expects two
	input columns: rawPrediction and label. Binary classi-
	fiers are used to separate the elements of a given dataset
	into one of two possible groups (e.g. fraud or not fraud)
	and is a special case of multiclass classification.
NodeMulticlassClassificationEvaluator	Evaluator for multiclass classification, which expects
	two input columns: score and label. A multiclass clas-
	sification describes a classification problem where there
	are M>2 possible labels for each data point (the case
	where M=2 is the binary classification problem)

Table 9: Apache Spark based Evaluation Processors in Fire Insights

• https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f

Machine learning continues to be an increasingly integral component of our lives, whether we're applying the techniques to research or business problems. Machine learning models ought to be able to give accurate predictions in order to create real value for a given organization.

While training a model is a key step, how the model generalizes on unseen data is an equally important aspect that should be considered in every machine learning pipeline. We need to know whether it actually works and, consequently, if we can trust its predictions. Could the model be merely memorizing the data it is fed with, and therefore unable to make good predictions on future samples, or samples that it hasn't seen before?

In this article, we explain the techniques used in evaluating how well a machine learning model generalizes to new, previously unseen data. We'll also illustrate how common model evaluation metrics are implemented for classification and regression problems using Python.

Model Evaluation Techniques

The above issues can be handled by evaluating the performance of a machine learning model, which is an integral component of any data science project.

Methods for evaluating a model's performance are divided into 2 categories: namely, holdout and Cross-validation. Both methods use a test set (i.e data not seen by the model) to evaluate model performance. It's not recommended to use the data we used to build the model to evaluate it. This is because our model will simply remember the whole training set, and will therefore always predict the correct label for any point in the training set. This is known as overfitting.

Holdout

The purpose of holdout evaluation is to test a model on different data than it was trained on. This provides an unbiased estimate of learning performance.

In this method, the dataset is randomly divided into three subsets:

1)Training set is a subset of the dataset used to build predictive models.

2)Validation set is a subset of the dataset used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning a model's parameters and selecting the best performing model. Not all modeling algorithms need a validation set.

3)Test set, or unseen data, is a subset of the dataset used to assess the likely future performance of a model. If a model fits to the training set much better than it fits the test set, overfitting is probably the cause.

The holdout approach is useful because of its speed, simplicity, and flexibility. However, this technique is often associated with high variability since differences in the training and test dataset can result in meaningful differences in the estimate of accuracy.

Cross-Validation

Cross-validation is a technique that involves partitioning the original observation dataset into a training set, used to train the model, and an independent set used to evaluate the analysis.

The most common cross-validation technique is k-fold cross-validation, where the original dataset is partitioned into k equal size subsamples, called folds. The k is a user-specified number, usually with 5 or 10 as its preferred value. This is repeated k times, such that each time, one of the k subsets is used as the test set/validation set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get the total effectiveness of our model.

For instance, when performing five-fold cross-validation, the data is first partitioned into 5 parts of (approximately) equal size. A sequence of models is trained. The first model is trained using the first fold as the test set, and the remaining folds are used as the training set. This is repeated for each of these 5 splits of the data and the estimation of accuracy is averaged over all 5 trials to get the total effectiveness of our model. As can be seen, every data point gets to be in a test set exactly once and gets to be in a training set k-1 times. This significantly reduces bias, as we're using most of the data for fitting, and it also significantly reduces variance, as most of the data is also being used in the test set. Interchanging the training and test sets also adds to the effectiveness of this method.

• https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Model Evaluation in Fire Insights

10.1.8 Model Persistence

Save / Load Model allows you to save your model to files and load them later in order to make predictions.

Fire Insights allows you to save the ML Model created. The ML Models can be loaded in the same or other workflows to be used for scoring. The ML Models can also be downloaded from HDFS Browse Page.

The ML models can be saved into the following locations:

- HDFS : when Fire Insights is connected to a Hadoop Cluster
- S3 : when Fire is configured and connected to AWS.
- Local Machine FileSystem : when Fire is running in local mode

In order to save onto S3, the model path can be provided as s3://models/priceprediction

Persisting SparkML Models

Spark ML Models

Spark ML models are saved into a directory with multiple files in it. Fire Insights has processors for saving and loading the Spark ML models.

Save Model processor

NodeModelSave processor, saves the given Apache Spark ML model at the given location.

label	message		id	words	features	
double	string		double	array	vectorudt	
SE LEVEL :		DEFAULT modelsavepath		•	BROWSE HDFS VIEW FILE	
PUT:		* TRUE O FALSE				
-	double E LEVEL : O	double string	double string LLVEL 0 DEFAULT modeltarrepath	double string double tLIVEL.0 DEFAULT modelsovepath	double shing double array turve.e	double string double array vectorudt

ML Save Workflow



Load Model processor

10 ML Model Loadz	NodeModelLood				< ×
OUTPUT STORAGE LEVEL : 0		DEFAULT	Ŧ		
PATH :		modelsavepath		BROWSE HDFS VIEW FILE	
MODEL TYPE : 0		PIPELINE	Ŧ		
					OK CANCEL

ML Load Workflow

🔥 Fire Insights 🛛 DATA BROWSERS - 🖽 APPLICATIO	O SCHEDULED - 🕫 EXECUTIONS - 20 MODELS - 20 DATA QUALITY - 20 AGMINISTRATION - 111 PROCESSORS - 9- 11- 12- 12- 12-
Edit Workflow	© HELP
NAME: Read Model - Type of fault in steel plates	CATEGORY: Model Penistence 0
O ADD NODE	SAVE SAVE WITH COMMENT CLEAR BACK EXECUTE WORK/LOW HISTORY COMMARE CHANGE UUD BEAUTEY
All Nodes Datasets Hive Databases	•
Q Search Nodes	
A 01-10	
A 02-Parse	processed_fouh.cs Weburkasenbler Dringindeser
A 03-Prepare	
A 04-DataValidation	Predict PrefMices
▲ 05-FeatureEngineering	
A D6-Code	ML Model Load
A 06-Filter	
A 07-joinUnion	
4 08-Group	
A 09-DataProfiling	
A 10-Visuelization	
▲ 11-ML-SparkML	
₫ 12-ML-H2O	
A 13-ML-AWSSogemaker	

Persisting H2O Models

H2O Models

H2O Models can be saved in binary format or in MOJO format. Fire Insights has processors for saving and reading them back.

Save H2o Model processor

H2OModelSave Processor saves the H2O model at the specified path in the binary format.

NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
түре	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer
FORMAT	dd/MM/yyyy HH:mm											
OUTPUT STOR	AGE LEVEL : 🖗			DEFAULT					٠			
ATH • : 😡				/tmp/h2o-kmean	s-model					BROWSE HDFS	VIEW FILE	

Load H2o Model processor

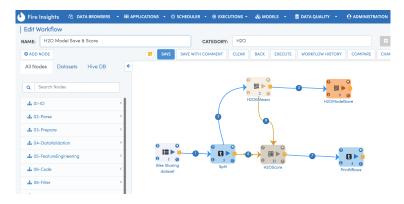
H2OModelLoad Processor loads the H2O model in binary format from the specified path.

NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer
FORMAT	dd/MM/yyyy HH:mm											
UTPUT STOR	AGE LEVEL : 😡			DEFAULT					٠			
(TH * : 🛛				/tmp/h2o-kmean	s-model					BROWSE HDFS	VIEW FILE	

More details of saving and loading the H2O Models is available here:

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/save-and-load-model.html

Save and Load H2O Workflow



Below is a workflow, which saves the generated H2O model on the file system.

Below is a workflow, which load back the saved model and used in batch scoreing.



Persisting Scikit Learn Models

Scikit-Learn models are persisted with pickle. Fire Insights has processors for saving and loading the pickle files. More details of the pickle format is available here:

https://scikit-learn.org/stable/modules/model_persistence.html

10.1.9 Model Serving

Fire Insights allows you to save your models. These models can be saved to:

- HDFS : when running on a Hadoop Cluster
- S3 : when running on AWS
- ADLS : when running on Azure
- Local file system : when running on your laptop or independent machine

Once these models are saved, they can be served in various ways.

Scoring with Workflows

Fire Insights enables you to build workflows. Workflows provide for reading data, transforming them and also creating machine learning models. Fire Insights supports a number of ML frameworks including Scikit Learn, H2O, Spark ML, Keras etc.

Models built with the workflows can be saved onto the File System. The models can then be scored with another workflow.

Data Preparation and Scoring Environments

The workflows built with Fire Insights can run on a variety of environments. These include:

- Standalone machine
- AWS EMR
- Azure HDInsights
- Databricks
- Cloudera

In any of these environments, Fire Insights does not need to be installed for model scoring. When running on Standalone machine, scoring can be performed with running java/python using the supplied jar/wheel files and the workflow json.

When running on clusters, scoring can be performed with spark-submit using the supplied jar/wheel files and the workflow json.

Workflow Patterns for Scoring Models

There are a few patterns by which Fire Insights enables Data Preparation/Feature Engineering and Model Scoring.

- One workflow for Data Preparation/Feature Engineering, another for Model Training and the third for Model Scoring
- One workflow for Data Preparation/Feature Engineering plus Model Training. And another workflow for Data Preparation/Feature Engineering plus Model Scoring.

Using 3 Workflows

In this pattern, one workflow is built to read in the input datasets, perform Data Preparation and also Feature Engineering. This workflow prepares the input datasets to be used for Training and also Scoring and saves it to the File System.

The second workflow reads in the prepared data, builds the model and then save it to the File System.

The third workflow also reads in the prepared data, reads in the ML model and then scores the input data. The result of scoring can be saved to the File System, Relational Database, Cassandra, MongoDB, HIVE etc.

Using 2 workflows

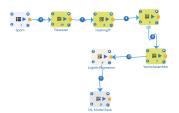
In this pattern, one workflow is built to read in the input datasets, perform Data Preparation/Feature Engineering and then finally build the ML Model.

For the second workflow, the first workflow is cloned with one click, and the model nodes are removed from the workflow. They are replaced with nodes which read in the model and then score the datasets.

Serving Spark MLlib Models

Fire Insights creates Apache Spark MLlib models. These models get saved as files on the File System.

NoveModelSave saves the Spark ML models as files. It uses the Spark interfaces to save the model.



Once the SparkML model is saved, they can be loaded and used in scoring. Fire Insights enables saving both Spark ML models and pipelines.

Batch Model Scoring:

By using NodeModelLoad & selecting the particular type of model to be loaded, the model would be loaded in the workflow and it can be used for scoring the input data.



Online Scoring with Kafka and Spark Streaming:

Scalable messaging platform like Kafka to send newly acquired data to a long running Spark Streaming process. The Spark process can then make a new prediction based on the new data.

Serving H2O Models

H2O allows you to persist the models you have built to either a Plain Old Java Object (POJO) or a Model ObJect, Optimized (MOJO).

Fire Insights has the following processors for persisting the H2O Models.

- H2OMojoSave
- H2OModelSave

Once the H2O model is saved, they can be used for serving.

H2O-generated MOJO and POJO models are intended to be easily embeddable in any Java environment. The only compilation and runtime dependency for a generated model is the h2o-genmodel.jar file produced as the build output of these packages.

We can use our H2OModelLoad or H2OMojoLoad to make a batch prediction, real-time prediction using Spark Streaming, Kafka or Storm. Or you can expose your model as a REST API.

https://h2o-release.s3.amazonaws.com/h2o/rel-ueno/2/docs-website/h2o-docs/pojo-quick-start.html

Serving H2O MOJO models

The below page on the H2O website gives details on serving a MOJO model.

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/productionizing.html#step-2-compile-and-run-the-mojo

Serving H2O POJO models

The details for serving a POJO models is described in this page.

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/productionizing.html#building-a-pojo

```
import java.io.*;
   import hex.genmodel.easy.RowData;
2
   import hex.genmodel.easy.EasyPredictModelWrapper;
3
   import hex.genmodel.easy.prediction.*;
4
5
   public class main {
6
   private static String modelClassName = "gbm_pojo_test";
7
    public static void main(String[] args) throws Exception {
      hex.genmodel.GenModel rawModel;
10
      rawModel = (hex.genmodel.GenModel) Class.forName(modelClassName).newInstance();
11
      EasyPredictModelWrapper model = new EasyPredictModelWrapper(rawModel);
12
13
      RowData row = new RowData();
14
      row.put("Year", "1987");
15
      row.put("Month", "10");
16
      row.put("DayofMonth", "14");
17
      row.put("DayOfWeek", "3");
18
      row.put("CRSDepTime", "730");
19
      row.put("UniqueCarrier", "PS");
20
      row.put("Origin", "SAN");
21
22
      row.put("Dest", "SFO");
23
      BinomialModelPrediction p = model.predictBinomial(row);
24
      System.out.println("Label (aka prediction) is flight departure delayed: " + p.
25
   \rightarrow label);
      System.out.print("Class probabilities: ");
26
      for (int i = 0; i < p.classProbabilities.length; i++) {</pre>
27
        if (i > 0) {
28
          System.out.print(",");
29
        }
30
        System.out.print(p.classProbabilities[i]);
31
      }
32
      System.out.println("");
33
```

(continues on next page)

(continued from previous page)

34 35

}

}

Useful links:

https://medium.com/spikelab/building-a-machine-learning-application-using-h2o-ai-67ce3681df9c

Serving AWS SageMaker models

When the SageMaker models are built in Fire Insights, SageMaker automatically provides a REST endpoint for online scoring of the models.

The details for it are available here:

- https://aws.amazon.com/blogs/machine-learning/creating-a-machine-learning-powered-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-with-amazon-api-gateway-mapped-rest-api-gateway-gatew
- https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aw

Serving Scikit Learn Models

Fire Insights provides the following processors for persisting the Scikit Learn models as pickle files:

• SaveAsPickle

Once the Scikit Learn model is saved, they can be used for serving.

The details for Scikit Learn Model Persistence is available here:

• https://scikit-learn.org/stable/modules/model_persistence.html

Serving Tensorflow Models

Fire Insights provides the following processors for persisting the Tensorflow models:

- NodeSaveKerasModel
- NodeLoadKerasModel

Integration with MLflow

Fire Insights integrates deeploy with Apache MLflow. Fire Insights can be configured to output the models to MLflow.

CHAPTER 11

Time Series

11.1 Time Series Analysis

Time series analysis is a statistical technique that deals with time series data, or trend analysis. Time series data means that data is in a series of particular time periods or intervals.

https://www.statisticssolutions.com/time-series-analysis/

Fire Insights provides a number of features for Time Series Analysis.

11.1.1 Time Series Feature Engineering

Fire Insights provides a number of Processors for Feature Engineering of Time Series Data. These include:

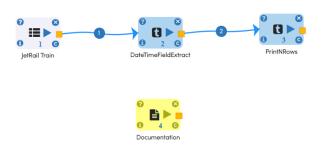
Features	Description
DateTimeFieldExtract	Extracts year, month, day of month, hour, minute, second and week of year from times-
	tamp/date columns
Days to holiday	Days remaining for next holiday
Days from holiday	Days passed after holiday
Time-segmentation	Divide data in morning, afternoon, evening, night to get more idea about time based pattern
MovingWindowingFu	nctators lates the moving values using the given function
WindowingAnalytics	Implements window functions is mainly through the operators rolling and expanding
Exponential Mov-	The Exponential Moving Average (EMA) assigns a greater weight to the most recent price
ing Average (EMA)	observations. While it assigns lesser weight to past data, it is based on a recursive formula
	that includes in its calculation all the past data in our price series.

Table 1: Update New features where needed

DateTimeFieldExtract

Below is the sample workflows which contains DateTimeFieldExtract processor in Fire Insights.

It reads the JetRail Train dataset & use DateTimeFieldExtract processor which create New DataFrame by extracting Date & Time field and print the result.



DateTimeFieldExtract processor Configuration:

NAME	Datetime		Count	
TYPE	fimestomp		integer	
FORMAT	dd-MM-yyyy HHimm			
UTPUT STOP	RAGE LEVEL : Ø	DEFAULT	~	
OLUMN : 🛛		Datetime : timestamp	~	
TRACT YEA	LR : 😡	true	~	
TRACT MO	NTH : O	true	~	
TRACT DAY	OF MONTH : Ø	frue	~	
TRACT HOU	UR : 🖗	true	¥	
TRACT MIN	IUTE : Ø	true	*	
TRACT SEC	COND : 0	true	~	
TRACT WEE	EKOFYEAR : O	folse	×	

Output result of DateTimeFieldExtract processor:

ecuting Node fire.nodes.et	LNodeDateTimeField	Extract : 2 : Sep 24, 2020	4:41:09 AM				
Input Schema							
inpor oction to							
Row Values							
ow Values							
Datetime	Count	Datetime_year	Datetime_month	Datetime_dayofmonth	Datetime_hour	Datetime_minute	Datetime_second
TimestampType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType
2012-08-25 00:00:00.0	8	2012	8	25	0	0	0
2012-08-25 01:00:00.0	2	2012	8	25	1	0	0
2012-08-25 02:00:00.0	6	2012	8	25	2	0	0
2012-08-25 03:00:00.0	2	2012	8	25	3	0	0
2012-08-25 04:00:00.0	2	2012	8	25	4	0	0
	2	2012	8	25	5	0	0
2012-08-25 05:00:00.0							

MovingWindowingFunctions

Below is the sample workflows which contains MovingWindowingFunctions processor in Fire Insights.

It reads the ticker dataset, concatenate the input column, casting specified column to new data type, use Moving-WindowingFunctions processor which calculates the moving value of selected function of input column and print the result.

MovingWindowingFunctions processor Configuration:

Output result of MovingWindowingFunctions processor:



3 Moving	WindowFunctions	NodeMovingWindowFunction	16					1	×
NAME	year	month	date	symbol	temp	price	date_new		4
TYPE	integer	integer	integer	string	integer	double	integer		
FORMAT									
UTPUT STORA	GE LEVEL : Ø		DEFAULT		~				
WINDOW START : O			-1					1	
INDOW END : 4	0		1						
ARTITION COL	UMN NAME : 😡		symbol : string 👻						
RDER COLUM	N NAME : O		date_new : integer			~			
ARIABLES LIST	: 0								1
NPUT COLUMNS	0		FUNCTIONS @						
price		✓ ovg					~ •		
price		Ý	min				~ •		
							or	CANC	- 01
								Chine	

ecuting Not	de fire.nodes.et	H.NodeMoving	MindowFunct	ions:8:Sep 2	4, 2020 4:07:35	AM							
Input Schem	-												
inpui scher													
Row Values													
Row Values													
year	month	date	symbol	temp	price	date_new	mean_price	min_price	max_price	stddev_price	variance_price	skewness_price	kurtosis_price
IntegerType	IntegerType	IntegerType	StringType	IntegerType	DoubleType	IntegerType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType
2015	8	14	ADN	76297	186.58	2015814	186.58	186.58	186.58	0.0	0.0	0.0	0.0
2015	8	12	к	55449	68.91	2015812	68.91	68.91	66.91	0.0	0.0	0.0	0.0
2015	8	13	к	33654	65.91	2015813	65.91	68.91	66.91	0.0	0.0	0.0	0.0
2015	8	18	BLK	19198	328.8	2015818	328.8	328.8	328.8	0.0	0.0	0.0	0.0
2015	9	3	FLIR	85729	28.38	201593	29.375	28.38	30.37	1.407	1.98	0.0	-2.0
	8	18	FUR	45667	30.37	2015818	29.375	28.38	30.37	1.407	1.98	0.0	-2.0
2015		20	MAT	384263	22.76	2015820	22.76	22.76	22.76	0.0	0.0	0.0	0.0

11.1.2 Time Series Visualizations

Fire Insights provides a number of Processors for the visualization of the time series data.

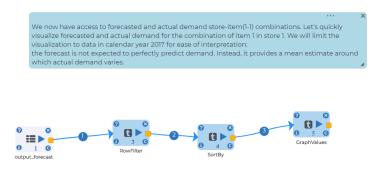
Charts	Description
Line	Perfect for series of data points to form a continuous line. Example - Represent Daily sales
	data.
Bar	Bar charts are a fundamental visualization for comparing values between groups of data.
	Best way to represent Categorical data.
Scatter	Scatter plots are used to observe relationships between variables.
Histogram	Histograms are a type of graph that shows the distribution of a dataset. They graph the
	percentage or the number of instances of different categories.
Pie	Illustrate the percentage breakdown of a small number of data points, then they can be very
	effective.

Table 2:	Undate	New	features	where	needed
1u010 2.	Opulle	11011	reatures	where	neeucu

Charts : LineChart

Perfect for series of data points to form a continuous line. Example - Represent Daily sales data

Below is the sample workflows which contains Time Series data and visualize using line chart in Fire Insights.



Configurations for visualization processors in Fire Insight: * Set number of columns want to represent on y axis with respect to x axis * Set chart type based on data type

NAME	forecast_date	sales_pred_mean		sales_pred_lower	date	store	item	sales	
IVPE	date	double		double	double	integer	integer	integer	
ORMAT									
TLE :			Graph						
LABEL :									
			year						
LABEL :			sales						
AX VALUES	TO DISPLAY		10000						
HART TYPE									
MART TIPE			Line Chart			~			
STREAMIN	IG?: 0		false			~			
COLUMN :			forecast_date : d			~			
			lorecasi_bate. o	ole					
COLUMNS		Available			Selec			•	•
		sales_pred_lowe date : double	r : double			es_pred_mean : doubl as : integer			

Output result of Visualization processor:

Charts : BarChart



Charts : Scatter



11.1.3 Time Series Modeling

Fire Insights provides a number of Processors for Time Series Modeling. These include:

Models	Description
Prophet	Prophet is a procedure for predicting time series data based on an additive or multiplicative
	model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holi-
	day effects. It is best for time series that have strong seasonal effects and several seasons of
	historical data. Prophet is robust model to missing data and shifts in the trend, and able to
	handles outliers. For more: https://facebook.github.io/prophet/
Arima	ARIMA is a model which is used for predicting future trends on a time series data.
	It is model that form of regression analysis. For more: https://en.wikipedia.org/wiki/
	Autoregressive_integrated_moving_average
XGBoost	XGBoost is gradient boosting algorithm. It is also known as 'regularized boosting' tech-
	nique - seeks a goot bias-variant trade-off to reduce overfitting allows cross-validation at
	each iteration of the boosting process and thus it is easy to get the exact optimum num-
	ber of boosting iterations in a single run. For more: https://docs.h2o.ai/h2o/latest-stable/
	h2o-docs/data-science/xgboost.html#limitations
LSTM	LSTM is special kind of recurrent neural network that is capable of learning long term
	dependencies in data. This is achieved because the recurring module of the model has
	a combination of four layers interacting with each other. This is a great benefit in time
	series forecasting, where classical linear methods can be difficult to adapt to multivariate
	or multiple input forecasting problems. For more: https://www.tensorflow.org/tutorials/
	structured_data/time_series

Table 3: Update New features where needed

Prophet

Below is the sample workflows which contains Prophet processor in Fire Insights.

Equation - y(t)=g(t)+s(t)+h(t)+t,

where:

- Trend g(t): models non-periodic changes
- Seasonality s(t): represents periodic changes
- Holidays component h(t): contributes information about holidays and events

It reads the AirPassengers dataset & use Prophet processor which forecasting of univariate time series data and print the result.



Prophet processor Configuration:

AME	Month		Passengers		
YPE	string		integer		
DRMAT					
TPUT STORA	AGE LEVEL : O	DEFAULT		~	
DS COLUMN : O		Month : string		¥	
:0		Passengers : intege	r	~	
SROWTH: 0		linear		~	
RLY SEASO!	NALITY:0	auto		~	
WEEKLY SEASONALITY: 0		auto		~	
DAILY SEASONALITY : 0		auto		~	
SONALITY N	MODE: 0	additive		~	
NTERVAL WIDTH : 0		0.95			

Output result of Prophet processor:

Executing Node Prophet	4 : 2020-10-14 08:02:51					
Input Schema						
Prophet						
Prophet						
ds	trend	trend_lower	trend_upper	yhat	yhat_lower	yhat_upper
TimestampType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType
1960-12-02 00:00:00	477.9876242764134	477.9876242764134	477.9876242764134	477.9876242764134	386.616797959175	568.0876649635197
1960-12-03 00:00:00	478.0818661809157	478.0818661809157	478.0818661809157	478.0818661809157	386.49584866086366	563.735761691239
1960-12-04 00:00:00	478.1761080854181	478.1761080854181	478.1761080854181	478.1761080854181	387.8146535048279	568.5364524533405
1960-12-05 00:00:00	478.27034998992036	478.27034998992036	478.27034998992036	478.27034998992036	387.1029238667477	573.7147279632921
1960-12-06 00:00:00	478.36459189442263	478.36459189442263	478.36459189442263	478.36459189442263	391.4535753677844	570.7778473695247
1960-12-07 00:00:00	478.45883379892496	478.45883379892496	478.45883379892496	478.45883379892496	390.9153770778575	562.878389360325
1960-12-08 00:00:00	478.55307570342734	478.5530353942389	478.5530821573438	478.55307570342734	388.94753523098456	566.1174101135832
1960-12-09 00:00:00	478.6473176079296	478.64703669160457	478.6473365080639	478.6473176079296	393.8028360196598	567.4649991588647

ARIMA

Below is the sample workflows which contains ARIMA processor in Fire Insights.

- AR (Autoregression): A changing variable that regresses on its own lagged/prior values.
- I (Integrated): Differencing of raw observations to allow for the time series to become stationary
- MA (Moving average): Dependency between an observation and a residual error from a moving average model

In terms of y, the general forecasting equation is:

 $\hat{\mathbf{y}}\mathbf{t} = \mu + 1 \mathbf{y}\mathbf{t}\cdot\mathbf{1} + \dots + \mathbf{p} \mathbf{y}\mathbf{t}\cdot\mathbf{p} - \theta \mathbf{1}\mathbf{e}\mathbf{t}\cdot\mathbf{1} + \dots + \theta \mathbf{q}\mathbf{e}\mathbf{t}\cdot\mathbf{q},$

where: * $\mu \rightarrow \text{constant}$

- 1 yt-1 +... + p yt-p \rightarrow AR terms (lagged values of y)
- θ 1et-1 -..... θ qet-q \rightarrow MA terms (lagged errors)

It reads the AirPassengers dataset & use ARIMA processor which Forecast the airline passengers count, generate a new column with unique index/value for each row in dataset and print the result.

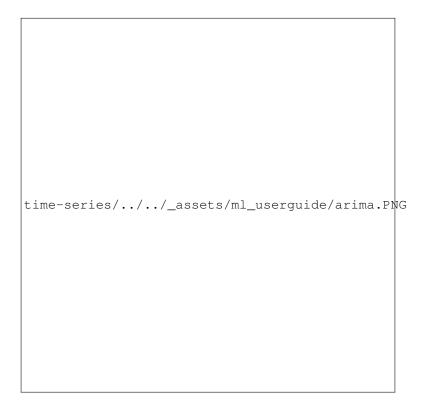
ARIMA processor Configuration:

Output result of ARIMA processor:

H2OXGBoost

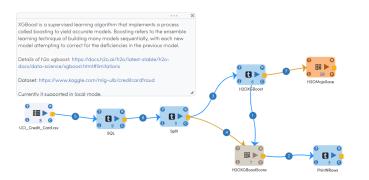
Below is the sample workflows which contains H2OXGBoost processor in Fire Insights.

It reads the UCI_Credit_Card dataset & use H2OXGBoost processor supervised learning algorithm that implements a process called boosting to yield accurate models and save the model in s3 location.



2 ARIMAIZ ROMANDARIA				1
NAME	Month		Passengers	1
TYPE	string		integer	
FORMAT				
UTPUT STOR	AGE LEVEL : Ø	DEFAULT		×
Y: 0		Passengers : integ	er .	v
SEASONAL : 0		true		~
EPWISE : O		true		~
ACE : 0		true		~
JPPRESS WA	RNINCS : O	true		~
ERROR ACTION : O		ignore		
CORING: 0		mse		
ORECAST: 0		15		

name						
Summary:						
		548	IMAX Resu	lts		
Dep. Vari				Observations		144
Model:				Likelihood		-674.913
Date: Time:	ble	d, 14 Oct 2	020 AIC :44 BIC			1365.825
Sample:		60:13	0 HDI			1309.528
sample:			0 nga 144			13/5.45/
Covariano	Type:		opg			
	, ,					
	coef	std err	2	P> z	[0.025	0.975]
ar.11	-0.5582	0.117	-4.782		-0.787	-0.329
ar.L2	0.4935	0.113	4.375		0.272	0.715
ar.L3	0.1238	0.128	0.970		-0.126	0.374
ar.14	-0.5213	0.085	-6.136		-0.688	-0.355
ma.11	0.9059	0.094	9.657		0.723	1.091
ma.L2 ma.L3	-0.5590 -0.7385	0.145	-3.866		-0.842 -0.952	-0.276 -0.525
signa2	724 1724	85.616	8 458		556.369	*0.525 891.976
						091.976
Liung-Box				Janque-Bena		14.59
Prob(Q):				Prob(JB):		0.00
	fasticity (H):			Skew:		0.74
	two-sided):		0.00	Kurtosis:		3.52



H2OXGBoost processor Configuration:

ame	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT	١.,
YPE	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	
ORMAT																				
SPONS	E COLUMI	N :					label :	string							~					
07000		S CATEGORIO																		
RESPU	VOE COLIN	5 CALEGORI	AL:0				true								~					
ATURE	COLUMNS	÷:0			Available									Sele	cted					ï
	TURE COLUMNS - O				BILL_AN BILL_AN BILL_AN BILL_AN BILL_AN BILL_AN PAY_AN PAY_AN PAY_AN PAY_AN PAY_AN	11 : integ 172 : integ 173 : integ 173 : integ 175 : integ 171 : integ 173 : integ 173 : integ 173 : integ 175 : integ 176 : integ 176 : integ 176 : integ	991 991 991 997 997 997 997 997 991 991	onth : inte	ger				* *	AI PI PI PI PI PI PI PI	MIT_BAL: integer DUCATION: in ARRIAGE: integer NY_0: integer NY_2: integer NY_4: integer NY_5: integer NY_6: integer	teger iger				

H2OMojoSave processor Configuration:

10 H2	ОМојс	Savez	• Node	H2OMejoServe																×
NAME	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMTI	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMTS	BILL_AMT6	PAY_AMT	•
TYPE	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	in
FORMAT	ORMAT																			
OUTPUT S	OUTPUT STORAGE LEVEL : O								DEFAULT											
PATH *: 0 s3a://fire-sample-data/data/output/ml-h2a											BROWS	E HDFS BI	XOWSE S3	VIEW FILE						

On successful submission of the job, the model get saved to specified locations, you can just view the model at specified location.

facoing an i	imal Hashina														
Log Pile Loop	fan														
eostlas-m	NOM 194844134	em.log													
Apoche Spor	k job Loundheo	1													
3.1.0. (park)	2.4/top/world	(Lawer, Society So	Ldegar sakto da o 200200112200 Jaarlef Inar 2010	percent from -	-postback-art)	Ing" Alexan Magazineze	enstas Rechtla 54-119-115-15	alloanatallean 0	litastavia ancorava, con	r false rom continense	hflas-file / mroegerkool	hanayland source 	Mina- t signaturtext -	-joi-14 80040	66-296-4475
Row Tolues															
Row Volues ID	LIMIT_BAL		EDUCATION			PAY_0	PKY_2	PAY_3	PWY_4	PAY_5	EXY_6		DILL_AMTS		
	integer/spe				Mol.	-			-	-	-	-	Integerition	integerType	integer/ype
1	NOBOR	2	2 maharitra	2	14	D.	a subscribe	d	0	D.	a subtraine	28239	11017	12020	1611
10	20000	1	3	1	35	a	-2	a	a	4	d.	a	0	0	a
12	280000	2	1	1		4	d	d	4	4	2	122011	21670	1986	487
12	200000	2	1	2	51	-1	-1	-1	-1	-1	2	12261	2670	9986	657
16	10000	1	2	1	30	1	2	2	0	D	2	45823	87189	65/01	08782
14	70000	1	2	2	30	1	2	2	0	0	2	65803	67369	65701	68762
ur	20000	1	1	1	34	D		2	3	2	2	15375	16212	17528	1832.0
σ	20000	1	1	2	34	0		2	2	2	2	92318	16110	17426	18358
•															
Ovput Sch	una														
Executing No.	de firecrodes.h	2s.Node+20A	AginBowe : 10 : De	+ 14, 2029 12/H	06794										
D Input Scherr	10														
R20Migo5c	210														
Saring 192	Bolo to file	n altar Artica	septembra (b	talastast.lali	No. PETMAL	VID-Auto-Ba	AC MARTINESS	MEA.							
Cupul Sch	ana -														
	de fen soder d	Lo Bulletone	NAME OF COLUMN	terms an elizated											
Executing No.															

#S3 Buckets s3a://fire-sample-data/data/output/ml-h	to/							
UPLOAD FILE CREATE DIRECTORY				Q Sec	ch			
NAME	ратн	SIZE	IS DIREC	TORY	OWNER	GROUP	LAST UPDATED	ACTION
573b94ef-5317-4a3d-8486-bff97423c963	data/output/ml-h2o/573b94ef-5317-4a3d-8486-bf97423c963/		true					

CHAPTER 12

Tutorials

12.1 Tutorials

12.1.1 Reading - Writing Data

Creating Dataset for CSV Files

When working with data in Fire Insights, the first step is to create a dataset that you plan to process subsequently. Dataset is a wrapper around your data which makes it easy to handle it in Sparkflows workbench.

When datasets are created, Fire Insights automatically infers the schema using Spark-CSV library from Databricks.

Datasets List

When you open any application, all existing Datasets specific to the application are displayed in the Datasets tab.

Ap	plice	ation	- Analytics														
Dat	asets	Wor	kflows Dashboar	d Interact	ive Dashboard Workf	low Exec	utions Schedule	is Share	Credentials Cor	nections							
Da	tase	əts					٩	Search Dataset			CREA	TE -	1	EXPO	RT DA	TASE	TS
	ю	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACT	ION					
	97	spor	kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	1	۲	ф	Ø	B	80	Û
	1	adm	in Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn.		06/24/2019 at 10:31	PERMISSION_MODIFY	1	۲	Ф	Ø	6	80	8
	7	adm	in Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans_		09/16/2018 at 5:50P	PERMISSION_MODIFY	ø	۲	6	ø	.	&	•
	6	adm	in NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY	1	۲	ф	Ø	B	&	Û
	6		a Internet Trade		IntDail Tesis	000	Anna Adata Anteni		07/12/2016		ھ	•	EDA.	es.		æ	~

Dataset Creation

Choose type of Dataset to Create

Navigate to the "Datasets" tab in your application. Click on the "Create" button and choose "Dataset". In the pop-up choose "CSV" and then click "OK".

elect Dataset Type	>
AVRO	
Csv	
HIVE	
JDBC	
JSON	
PARQUET	
SEQUENCE	
XML	

Dataset Details

Clicking "OK" will take you to Dataset Details page where you can enter information about your dataset. In the screenshot below, we create a dataset from a housing.csv file. It is a comma separated file with a header row specifying the names of the various columns.

File Contents	
"id", "price", "lotsize", "bedrooms", "bathrms", "stories", "driveway" "1",42000,5850,3,1,2, "yes", "no", "yes", "no", "no", 1, "no" "2",38500,4000,2,1,1, "yes", "no", "no", "no", "no", 0, "no" "3",49500,3060,3,1,1, "yes", "no", "no", "no", "no", 0, "no" "4",66500,66550,3,1,2, "yes", "yes", "no", "no", "no", 0, "no" "5",61000,6360,2,1,1, "yes", "yes", "no", "no", "no", 0, "no" "6",66000,4160,3,1,1, "yes", "no", "no", "no", "no", 0, "no" "7",66000,3880,3,2,2, "yes", "no", "no", "no", "no", 0, "no" "8",69000,4160,3,1,3, "yes", "no", "no", "no", "no", 0, "no" "8",69000,4160,3,1,3, "yes", "no", "no", "no", "no", 0, "no" "9",83800,4800,3,1,1, "yes", "yes", "no", "no", "no", 0, "no" "10",88500,5500,3,2,4, "yes", "yes", "no", "no", "yes", 1, "no" "11",9000,7200,3,2,1, "yes", "no", "no", "no", "yes", 3, "no"	',"recro
	ок

For the housing.csv file, we will fill in the required fields as below.

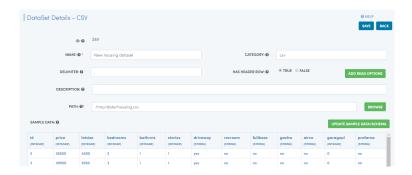
We specified a name for the dataset we are creating. 'Header' is set to true indicating that the file has a header row, field delimiter is comma and we also specified the path to the file.

Update Sample data/schema

Once we have specified the above, we hit the 'Update Sample data/schema' button. This brings up the sample data, infers the schema and displays it. We can change the column names and also the data types. Format column is used for specifying the format for date/time fields.

DataSet Details – CSV				HELP SAVE BACK
NAME: 🛛 *	New housing dataset	CATEGORY: 0	CSV	
DELIMITER: 0		HAS HEADER ROW: @	* TRUE O FALSE	ADD READ OPTIONS
DESCRIPTION: 0				
PATH: 🔂*	/tmp/data/housing.csv			BROWSE
SAMPLE DATA: 🖗				UPDATE SAMPLE DATA/SCHEMA
SCHEMA: O				
NAME O	DATA TYPE 😡	FORMAT O	ML TYPE O	

d INTEGER)	price (INTEGER)	Iotsize (INTEGER)	bedrooms (INTEGER)	bathrms (INTEGER)	stories (INTEGER)	driveway (STRING)	recroom (STRING)	fullbase (STRING)	gashw (STRING)	airco (STRING)	garagepl (INTEGER)	prefarea (STRING)
1	38500	4000	2	1	1	yes	no	no	no	no	0	no
	49500	3060	3	1	1	yes	no	no	no	no	0	no
	60500	6650	3	1	2	yes	yes	no	no	no	0	no
5	61000	6360	2	1	1	yes	no	no	no	no	0	no
5	66000	4160	3	1	1	yes	yes	yes	no	yes	0	no
	66000	3880	3	2	2	yes	no	yes	no	no	2	no
1	69000	4160	3	1	3	yes	no	no	no	no	0	no
)	83800	4800	3	1	1	yes	yes	yes	no	no	0	no
0	88500	5500	3	2	4	yes	yes	no	no	yes	1	no
1	90000	7200	3	2	1	yes	no	yes	no	yes	3	no
2	30500	3000	2	1	1	no	no	no	no	no	0	no
SCHEMA: 😡												
AME 🖸			D	ATA TYPE O		FORMA	10			ML TYPE	0	



Save the Dataset

Clicking the 'Save' button creates the new dataset. The dataset is now ready for use in any workflow within the specific application.

Application – Analytics		
Datasets Workflows Dashboard Interactive Dashboard	Workflow Executions Schedules Share Credentials Connections	
Datasets	Q Search Dataset	CREATE - EXPORT DATASETS
D ID USER NAME CATEGO DESCRIPTION	TYPE PATH DELIMIT LAST UPDATED PERMISSION	ACTION
289 ge New housing do csv	CSV /tmp/data/housi , 07/15/2019 at 6:32PM PERMISSION_MODIFY	/ • 🛛 🖉 👉 🗞 🕯
97 spor kidneycdr JDBC CDR details	JDBC com.mysql.jdbc 07/01/2019 of 3:11PM PERMISSION_MODIFY	/ • • • • • • •

Creating Dataset for AVRO Files

When working with data in Fire Insights, the first step is to create a dataset that you plan to process subsequently. Dataset is a wrapper around your data which makes it easy to handle it in Sparkflows workbench.

When datasets are created, Fire Insights automatically infers the schema using Spark-Avro library.

Datasets

A	oplico	ation - A	Analytics														
D	atasets	Workflo	ws Dashboard	Interact	ive Dashboard Workf	low Execu	utions Schedules	Share	Credentials Con	nections							
D	atase	ets					Q Se	arch Dataset			CRE	ATE -		EXPO	RT DA	TASE	rs
	ю	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	AC	TION					
	97	spor	kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	ø	۲	6	Ø	•	&	Û
	1	admin	Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn		06/24/2019 at 10:31	PERMISSION_MODIFY	ø	۲	đ	Ø	B	&	Û
	7	admin	Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans		09/16/2018 at 5:50P	PERMISSION_MODIFY	ø	۲	đ	Ø	•	&	1
	6	admin	NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY	1	۲	đ	Ø	B	&	Û
	6	autoria.	Internal Toxics		Int Dail Train	CON	Anna Adata Gateal		07/02/20016 7-222444		۵	-	DA	-			~

Dataset Creation

Navigate to the "Datasets" tab in your application where you want to create a new dataset. Click on the "Create" button and choose "Dataset". In the pop-up choose "AVRO" and then click "OK".

Select Dataset Type

	^
AVRO	
CSV	
HIVE	
JDBC	
JSON	
PARQUET	
SEQUENCE	
XML	
	OK CANCEL

Clicking "OK" will take you to Dataset Details page where you can enter information about your dataset. In the screenshot below, we create a dataset from a sample.avro file.

DataSet Details - AVRO	>			OHELP SAVE BACK
HAME O	Sample avro	ENTIN G ²	Aseniparkfovs/data/avot/part-r-00	BIOVEL
CATEGORY 0	ava			
DESCRIPTION 0	hel			
SAMPLE DATA Ø	NE DATA DOLLAR			

We specified a name, category, description & path of avro file for the dataset we are creating.

Once we have specified the above, we hit the

'Update Sample data/schema' button. This brings up the sample data, infers the schema and displays it. We can change the column names and also the data types. Format column is used for specifying the format for date/time fields.

	c2	c3		c4	
(UBLE)	(DOUBLE)	(DOUBLE)		(000818)	
	1.0	3.0		2.0	
	0.0	ш		1.0	
1	0.0	4.1		5.0	
	0.0	3.1		6.0	
	1.0	2.1		2.0	
	0.0	2.3		3.0	
	1.0	3.0		2.0	
	0.0	1.1		1.0	
	0.0	4.1		5.0	
	0.0	3.1		6.0	
	1.0	2.1		2.0	
HEMA: 0					
AE 😡	DATA TYPE 😡	FORMAT 0		ML TYPE Ø	
	DOUBLE	• format		NUMERIC	
	DOUBLE	format		NUMERIC	Ø HELP SAVE B
1	DOUBLE	•) format	PATH: Q * /us	NUMERIC er/sparkflows/data/avrol/part-r-00r	SAVE B
ı DataSet Details - AVRO	Double	• format	PATH: O * /us		SAVE B
) DataSet Details – AVRC NAME 9 * CATEGORY 9	DOUBLE Sample avro	format	PATH: Q * /us		SAVE B
i iataSet Details - AVRC	DOUBLE Sample avro	format	PATH: Q* /up		SAVE
nataSet Details - AVRC NAME 0* CATEGOM 0 DESCRIPTION 0	DOUBLE Sample avro	format	Path: Q * /us		SAVE B
Intra Set Details - AVRO NAME 9* CATEGORY 0 DESCRIPTION 0 SAMPLE DATA 0 UPDATE SAM	Sample avra	format	PATH: O * /ub		SAVE B
n name o * category: o description: o sample data: o update sam st st st	Bomple ono Gomple ono extra characterization test classification color color color color color color	c3 (c/sma)	PATH: Q * /us	er/sparkflows/data/avrat/part-r-00r	SAVE B
) DataSet Details - AVRO NAME 0 · CATEGORY 0 DESCRIPTION: 0 SAMPLE DATA 0 CIPONTE SAM ct ct cs	Sample avro evero fest composition composi	c) (00004) 10	PATH: Q * Zub	er/spariflows/data/avrol/part-t-00i ed. courses 2.0	SAVE B
n NAME O CATEGORY: O DESCRIPTION: O SAMPLE DATA: O UPDATE SAM et comments 28 20	Boundary Could Le Sample avro Could Le avro Could Le test Could Le test Could Le to avro Could Le avro Could Le to avro Could Le to avro Could Le to avro Could Le	c) (c)(Hall) (c)	ΡΑΤΗ Φ ^ο Λυ	er/sport/flows/data/ovrol/port-r-001 c4 (0098L) 20 10	SAVE B
) DataSet Details - AVR(NAME 0 * CATEGOM 0 DESCRIPTION 0	Sample avro evero fest composition composi	c) (00004) 10	PATH O" Lag	er/spariflows/data/avrol/part-t-00i ed. courses 2.0	

Clicking the 'Save' button saves the new avro dataset. The avro dataset is now ready for use in any workflow within the specific application.

Application -	,	Dashboard Executio	ns Sc	thedules Share Credent	ials Connections		ANALYTIC	:s •
Datasets Display List By Type	ALL CSV SEQUENCE	AVRO PARQUET	HIVE	Q Search Dataset			CREATE - EXPORT DATAS	ets
D ID USER	NAME CATEGO	DESCRIPTION	TYPE	PATH DEUMIT	LAST UPDATED	PERMISSION	ACTION	
737 spor	Sample avro avro	test	AVRO	/user/sparkflows	01/06/2020 at 9:38	PERMISSION_MODIFY	/ • 0 0 0 0 8	1
667 admir	Telco Churn Predicti	Labeled Telco Churn P	CSV	/tmp/data/chur ,	01/05/2020 at 10:56	PERMISSION_MODIFY	100000	0
673 admir	Transaction Dataset	Retail Transactions Da	CSV	/tmp/data/trans ,	09/16/2018 of 5:50P	PERMISSION_MODIFY	/ • 6 6 6 8	
672 admir	NYC Trip Data	NYC Trip Data	CSV	/tmp/data/trips ,	07/16/2016 at 6:25PM	PERMISSION_MODIFY	1 • 6 7 6 8	5 🗎

Creating Dataset for JSON Files

When working with data in Fire Insights, the first step is to create a dataset that you plan to process subsequently. Dataset is a wrapper around your data which makes it easy to handle it in Sparkflows workbench.

When datasets are created, Fire Insights automatically infers the schema using Spark-Json library.

Datasets

Datasets	Workfle	ows Dashboard	Interacti	ve Dashboard Workfl	ow Execu	utions Schedules	Share	Credentials Con	nections					
Datase	əts					Q Se	arch Dataset			CREATE		EXPO	ORT DA	ATASE
	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACTION				
97	spar	kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	10	đ	ø	B	&
1	admin	Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn		06/24/2019 at 10:31	PERMISSION_MODIFY	10	6	ø	6	&
7	admin	Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans		09/16/2018 at 5:50P	PERMISSION_MODIFY	10	6	B		&
6	admin	NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY	10	6	ø	•	&
— .	admin			latDail Train	000	Anno (data Gateai)		07/12/2016 7.22444			DA			

Dataset Creation

Navigate to the "Datasets" tab in your application where you want to create a new dataset. Click on the "Create" button and choose "Dataset". In the pop-up choose "JSON" and then click "OK".

Select Dataset Type	×
AVRO	
CSV	
HIVE	
Јрвс	
JSON	
PARQUET	
SEQUENCE	
XML	
	OK CANCEL

Clicking "OK" will take you to Dataset Details page where you can enter information about your dataset. In the screenshot below, we create a dataset from a customer.json file.

DataSet Details - JSON				O HELP SAVE BACK
NAME: 😡 *	Custmoer sample json	PATH: 😡*	/user/sparkflows/data/customer.json	BROWSE
CATEGORY: @	json			
DESCRIPTION: Ø	test			
SAMPLE DATA: O UPDATE SAME	PLE DATA/SCHEMA			

We specified a name, category, description & path of json file for the dataset we are creating.

Once we have specified the above, we hit the 'Update Sample data/schema' button. This brings up the sample data, infers the schema and displays it. We can change the column names and also the data types. Format column is used for specifying the format for date/time fields.

Clicking the 'Save' button saves the new json dataset. The json dataset is now ready for use in any workflow within the specific application.

Creating Dataset for Parquet Files

Fire insights supports reading from several file formats including Parquet files. Parquet files have schema embedded in them. Fire Insights is able to extract schema of Parquet files automatically.

address struct)					first_name (STRING)		last_name (STRING)		
Brighton,MI,4 B Blue Ridge Blvd,48	1116]				Josephine		Darokjy		
Bridgeport, NJ, 8 W Cerritos Ave #54	1,08014]				Art		Chemel		
SCHEMA: Ø									
IAME O		DATA TYPE O		FORMAT O			ML TYPE 😡		
address		STRUCT	•	StructField(addre	ss,StructType(St	ructField(city,Stri	TEXT		
first_name		STRING	٣	format			TEXT		
last_name		STRING		format			TEXT		
								O HELP SAVE BAC	
DataSet Details - JSO1								SAVE BA	
NAME: 😡 *	Custmoer sample js	son			PATH: 😶	/user/sparkflows	/data/customer.json		
		ion			PATH: @*	/user/sparkflows	/data/customer.json	SAVE BA	
NAME: 😡 *	Custmoer sample js	ion			PATH: ₽*	/user/sparkflows	/data/customer.json	SAVE BA	
NAME: 0 * CATEGORY: 0 DESCRIPTION: 0	Custmoer sample js	ion			PATH: ⊕ *	/user/sparkflows	rdata/customer.json	SAVE BA	
NAME: 0 * CATEGORY: 0 DESCRIPTION: 0	Custmoer sample ja json test	ion			PATH: 0 *	/user/sparkflows	/data/customer.json	SAVE BA	
NAME: 0 · CATEGORY: 0 DESCRIPTION: 0 SAMPLE DATA: 0 UPDATESAM address	Custmoer sample js jsan fest jete Data/schema	101			first_name	/user/sparkflows	last_name	SAVE BA	
NAME 0 * CATEGORY 0 DESCRIPTION: 0 SAMPLE DATA: 0 UPDATE SAM UDDATE SAM	Custmoer somple ja json test PLE DATA/SCHEMA	ion			first_name (3TRING)	/user/sporkflows	last_name (STRNG)	SAVE BA	

Datasets

The existing datasets are displayed in the DataSets page of specific application.

Dataset Creation

Navigate to the "Datasets" tab in your application where you want to create a new dataset. Click on the "Create" button and choose "Dataset". We now create a dataset for people.parquet. It is a parquet file.

In the 'Create DataSet' page fill in the required fields as below.

Specify the name of the dataset you are creating.

After specifying name and path, click the 'Update Sample data schema' button. This brings up the sample data, extracts the schema and displays it. Below we see that there are 2 fields : age and name. Age is of type integer and name is of type string.

Clicking the 'Save' button creates the new DataSet for us.

Now you are ready to use the dataset in your workflows.

Creating Dataset from MySQL Table

When working with data in Fire Insights, the first step is to create a dataset that you plan to process subsequently. Dataset is a wrapper around your data which makes it easy to handle it in Sparkflows workbench.

Datasets	Workf	lows Dashboo	ird Inte	ractive Dashboard	Executio	ons Schedules	s Share	Credentials C	onnections		
Dataset	s					٩	Search Data	set		CREATE *	EXPORT DATAS
Display Lis	t By Type	ALL CSV	SEQUE	NCE AVRO PAR	QUET	HIVE JSON	XML	JDBC			
ID	USER	NAME	CATEGO	DESCRIPTION	TYPE	PATH	DEUMIT	LAST UPDATED	PERMISSION	ACTION	
738	spar	Custmoer samp	json	test	JSON	/user/sparkflows		01/06/2020 at 10:49	PERMISSION_MODIFY	/ 👁 🙆	8 🖻 💩
737	spor	Sample avro	avro	test	AVRO	/user/sparkflows		01/06/2020 at 9:38	PERMISSION_MODIFY	/ 👁 🙆	C 🕞 💩
667	admin	Telco Churn Pre		Labeled Telco Churn P	CSV	/tmp/data/chur		01/05/2020 at 10:56	PERMISSION_MODIFY	/ 👁 🙆	12 🕞 💩 1
673	admin	Transaction Dat		Retail Transactions Da	CSV	/tmp/data/trans		09/16/2018 at 5:50P	PERMISSION_MODIFY	/ 👁 🙆	8 🖻 💩

Applicat	ion - Ar	alytics													
Datasets	Workflow	s Dashboard	Interacti	ve Dashboard Workfl	ow Execu	tions Schedules	Share	Credentials Con	nections						
Dataset	S					Q So	arch Dataset			CREA	TE •	EXP	ORT D.	ATASE	TS
0	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH	DELIMITER	LAST UPDATED	PERMISSION	ACT	ON				
97	spor	kidneycdr	JDBC	CDR details	JDBC	com.mysql.jdbc.D		07/01/2019 at 3:11PM	PERMISSION_MODIFY	ø		8	•	&	Û
	admin	Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/churn		06/24/2019 at 10:31	PERMISSION_MODIFY	ø	۱	6	6	&	Û
7	odmin	Transaction Dat		Retail Transactions Dat	CSV	/tmp/data/trans_		09/16/2018 at 5:50P	PERMISSION_MODIFY	ø	۱	6		&	1
6	admin	NYC Trip Data		NYC Trip Data	CSV	/tmp/data/trips		07/16/2016 at 6:25PM	PERMISSION_MODIFY		۱	6 2	•	&	Û
• ••	and so in	Internal Trade		IntiBall Tests	000	Anna Adata Gateall		07/12/2016		۵	~		-	e.	~

elect Dataset Type		
AVRO		
CSV		
HIVE		
JDBC		
JSON		
PARQUET		
SEQUENCE		
XML		

DataSet Details - PARC	QUET		O HELP SAVE BACK
NAME: 😡 *	New people parquet dataset	CATEGORY: Ø	
DESCRIPTION: 0			
PATH: 😶	/user/sparkflows/data/people.parquet		BROWSE
SAMPLE DATA:			UPDATE SAMPLE DATA/SCHEMA
SCHEMA: 0			

ataSet Details – PARG	QUET					HELP SAVE BAC
NAME: 0 *	New people parquet dataset		CATEGORY:	0		
DESCRIPTION: 0						
PATH: 🚱	/user/sparkflows/data/people.parquet					BROWSE
SAMPLE DATA: 😡					UPDATE SAMPL	LE DATA/SCHEM
age INTEGER)			name (STRING)			
10			Andy			
9			Justin			
SCHEMA: Ø						
KAME 😡	DATA TYPE 😡	F	ORMAT 😧		ML TYPE O	
age	INTEGER		format		NUMERIC	

Application - Analytics								
Datasets Workflows Dashboard Interacti	e Dashboard Workflow Executions	Schedules Share Cr	redentials Connection	ons				
Datasets		Q Search Dataset		CRI	ATE *	EXPORT	DATASE	TS
D USER NAME	CATEGO DESCRIPTION TYPE	PATH DELIMIT	LAST UPDATED	PERMISSION	ACTION			
321 spar New people parquet dataset	PAR	/user/sparkflows	07/15/2019 at 8:32PM	PERMISSION_MODIFY	10	0	8 🖻	80

When datasets are created, Fire Insights automatically infers the schema of the dataset.

Datasets

When you open any application, all existing datasets specific to the application are displayed in the Datasets tab.

Applicat	ion - A	Analytics														
Datasets Datasets	Workflo	ws Dashboard	Interacti	ve Dashboard Workfl	ow Execu		Q Search	Share	Credentials Con	nections	CRE	NTE -		XPORT	DATAS	FTS
	USER	NAME	CATEGORY	DESCRIPTION	TYPE	PATH			LAST UPDATED	PERMISSION		ION				کنن
97	spor	kidneycdr	JDBC	CDR details	JDBC	com.mysql.j		Lonarda	07/01/2019 at 3:11PM	PERMISSION_MODIFY			6	80	* &	Û
	admin	Telco Churn Pre		Labeled Telco Churn Pr	CSV	/tmp/data/e			06/24/2019 at 10:31	PERMISSION_MODIFY			_	2		8
□ 7 □ 6	admin admin	Transaction Dat_		Retail Transactions Dat NYC Trip Data	CSV	/tmp/data/t			09/16/2018 at 5:50P 07/16/2016 at 6:25PM	PERMISSION_MODIFY		-		8 0 8 0		
- -	a dan in	IniDail Teals		IntBall Tools	cou	Anna Islania I	intenti		07/12/2016		م		-			~

Dataset Creation

Navigate to the "Datasets" tab in your application where you want to create a new dataset. Click on the "Create" button and choose "Dataset". In the pop-up choose "JDBC" and then click "OK".

Select Dataset Type	×
AVRO	
CSV	
HIVE	
јовс	
JSON	
PARQUET	
SEQUENCE	
XML	
ОК	CANCEL

Specify the name of the dataset you are creating and other required parameters such as JDBC DRIVER, JDBC URL, USER, PASSWORD, DB, & TABLE etc.

Once you have filled in required information, hit 'Update Sample data/schema' button. This brings up sample data, infers the schema and displays it. You can change column names and data types as needed. Format column is used for specifying the format of date/time fields.

DataSet Details - JDBC			O HELP SAVE BACK
NAME: 🛛 *	New Mysql dataset	CATEGORY: 0	
JDBC DRIVER: @ *	com.mysql.jdbc.Driver	JDBC URL: @ *	jdbc:mysqt://13.82.42.223:3306
USER: 🛛 *	root	PASSWORD: @ *	
DB: 🛛 *	userdb	TABLE: 😡 *	Loandata
DESCRIPTION: 0			
SAMPLE DATA: 😡			UPDATE SAMPLE DATA/SCHEMA

DataSe	t Detail	s – JDB	С								HELP SAVE BAC	ск
	N	IAME: 🛛 *	New Mysq	l dataset			CA	TEGORY: 😡				
	JDBC DF	IVER: 🛛 *	com.mysql	.jdbc.Driver			lal	BC URL: 😧 *	jdbc:mysql://13.82.42.2	23:3306		
		USER: 🛛 *	root				PASS	WORD: 0*				
		DB: 🛛 *	userdb					TABLE: 🛛 *	Loandata			
	DESCR	IPTION: 0										
SAMPLE D	ATA: 😡									UPDATE SAME	LE DATA/SCHEMA	A
Loan_ID (STRING)	Gender (STRING)	Married (STRING)	Dependents (STRING)	Education (STRING)	Self_Employed (STRING)	ApplicantIncome (INTEGER)	CoapplicantIncome (INTEGER)	LoanAmount (INTEGER)	Loan_Amount_Term (INTEGER)	Credit_History (INTEGER)	Property_Area	2
LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1	Urbon	
LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1	Urbon	
LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360	null	Urbon	

Clicking the 'Save' button creates the new dataset that can be used in any workflow or Interactive dashboard within the specific application.

Application - Analytics					
Datasets Workflows Dashboard	Interactive Dashboard Workflow Executio	ns Schedules Share Cred	ientials Connections		
Datasets		Q Search Dataset		CREATE *	EXPORT DATASETS
D ID USER NAME	CATEGO DESCRIPTION	TYPE PATH DELIN	MIT LAST UPDATED	PERMISSION	ACTION
353 spar New Mysql dataset		JDBC com.mysql.jdbc	07/16/2019 at 9:02A	PERMISSION_MODIFY	/ • @ 6 6

Reading from RDBMS in Workflow

Fire has JDBC Processors for reading from JDBC sources or writing to JDBC sinks.

In order to connect to a JDBC source like MySQL/Oracle/DB2 etc. the JDBC driver needs to be installed in Fire Insights.

Use the steps here for installing the corresponding JDBC driver for your RDBMS:

• http://docs.sparkflows.io/en/latest/operating/installing-jdbc-drivers.html

Workflow for reading from MySQL

Below is a workflow which reads data from MySQL and saves to a CSV file. It reads in the data from the $dm_product$ table in MySQL and saves it to a CSV file.



JDBC Processor Configuration

Below are the configuration details of the JDBC Processor. It uses the provided user for reading from the MySQL database. On clicking on *Refresh Schema*, Fire gets the schema of the table in MySQL and populates the entries.

ReadJDBC 🕄			2
URL :	jdbc:mysql://localhost:3306/L2		
USER : 🚱	root		
PASSWORD :	••••••		SHOW PASSWORD
DB TABLE :	dm_product		
DRIVER :	com.mysql.jdbc.Driver		
SCHEMA COLUMNS : O REFRESH SC			
COLUMN NAMES OF THE TABLE @	COLUMN TYPES OF THE TABLE 😧	COLUMN FO	RMATS 🕑
product_id	INTEGER	\$ format	•
product_name	STRING	\$ format	•
product_description	STRING	♦ format	OK CANCEL

Results of reading from MySQL table

The below screenshot displays some of the records read from the MySQL table by Fire.

ROW VALUES	odes.dataset.NodeDatasetJDBC Apr 29, 2018 1:4	
product_id	product_name	product_description
IntegerType	StringType	StringType
1	Husky Rope 50	Rope
2	Husky Rope 60	Rope
3	Husky Rope 100	Rope
4	Husky Rope 200	Rope
5	Granite Climbing Helmet	Safety
6	Husky Harness	Safety
7	Husky Harness Extreme	Safety
8	Granite Signal Mirror	Safety
9	Granite Carabiner	Climbing Accessories
10	Granite Belay	Climbing Accessories

Specifying a sub-query

In the configuration of the JDBC node, for db_table anything that is valid in a FROM clause of a SQL query can be used. For example, instead of a full table we could also use a subquery in parentheses.

More details are available on the Spark Guide : https://spark.apache.org/docs/1.6.0/sql-programming-guide.html# jdbc-to-other-databases

JDBC 🕄	
URL :	jdbc:mysql://localhost/employees?user=root&password=sparkflows
DB Table :	(select first_name from employees.employees) emp
Driver :	com.mysql.jdbc.Driver

Above we have specified a subquery which selects only the 'first_name' from the employees table.

DBC	
Executing	g Node fire.nodes.dataset.NodeDatasetJDBC
B Row	Values
first_na	ime
StringTy	/pe
Georgi	
Bezalel	
Parto	
Chirstia	n
Kyoichi	
Anneke	
Tzvetan	
Saniya	
Sumant	
Duangk	aew

JDBC Drivers

Below are the JDBC URL's for some databases:

- MySQL : com.mysql.jdbc.Driver
- PostgreSQL : org.postgresql.Driver
- Oracle : oracle.jdbc.driver.OracleDriver

Example JDBC URL

Below are some examples of JDBC URL for reading from Relational sources:

- MySQL : jdbc:mysql://localhost:3306/mydb
- PostgreSQL : jdbc:postgresql://localhost:5432/mydb

Read PDF File

This workflow reads in PDF file from the given location. It then parses its content and creates DataFrame then prints the results.

Workflow

Below is the workflow that shows:

- How to read in PDF file from the given location and create the DataFrame from it
- Prints the result

	•••	×	
	This workflow reads in a PDF File and parses its content into String		
		4	
Ⅲ ► ⊙ ⊡	0		i 🕨
)F		Print	NRows

Reading And Parsing PDF File

DatasetPDF processor uses the passed location to download PDF file, parse its content into string and create the DataFrame.

Processor Configuration

1 PDF	NodeDatasetPDF			2	×
OUTPUT STORA	AGE LEVEL : 😧	DEFAULT	~		
PATH : 😧		/tmp/data/pdf-sample.pdf		BROWSE HDFS	
				VIEW FILE	
FILE NAME : 🛛		filename			
FILE CONTENT :	0	contents			
				OK CANCE	L

Processor Output

Prints the Results

It prints the result onto the screen.

Reading and Writing from ElasticSearch

Elastic Search is often used for indexing, searching and analyzing datasets. Fire Insights makes it easy to read data from Elastic Search, clean it and transform it as needed.

PDF		2	×
Row Values			
filename	contents		
StringType	StringType		
pdf- sample.pdf	Adobe Acrobat PDF Files Adobe® Fortable Document Format (PDF) is a universal file format that preserves all of the fortil, formatting, colours and graphics of any source document, regardless of the application and platform used to create 1. Adobe PDF is an ideal format for electricine, document distribution at a lovecome the problems commonly exolutient with electricine. The sharing - Anyone, anywhere coores PDF file. Buy unreals in the resAdobe Acrobat Reader, and and the file formats sometimet court open files because they don't have the applications used to create the document. PDF files of ways point courted on any printing which. PDF files of ways (palyor associal) contends to a contend the document is - PDF files of ways point courted) van any printing which. PDF files of ways (palyor associal) contends to applications used to create the document is - PDF files of ways point courted) van any printing which. PDF files of ways (palyor associal) contends the point of the share the applications used to create the document is - PDF files of ways point courted) van any printing which is - PDF files of ways and the varies of files. Advisor and document is - PDF files of ways point courted) van applications and varies in compatibilities. The files Acrobat Readob is advisor on the varies of the iso and graphics and ways and ways and warraw of applications and palyor on the VDF files of ways and warraw of the share warden from their source files and dominates and page on at the file of advisors on the VDF files of warden and the source files and dominates of the advisors on the VDF files on warden files on the files of warraw of the source files and dominates on the other of the other advisors of the source files and the other files of the other beautified and the other files and dominates on the other advisors of the other beautified on the tother files and bominates on the other advisors of the other beautified on the tother advisors of the other beautified on the tother files and bominates on	D	
			-
		6	ж

Elasticsearch-hadoop provides native integration between Elasticsearch and Apache Spark. In the example below we will first load data from HDFS into Elastic Search and then read it back into Apache Spark from Elastic Search.

If your data is already in Elastic Search, skip to "Workflow for Reading data from Elastic Search".

Loading data into Elastic Search

Create a new empty workflow. Drag and drop the source dataset from which you want to load data into Elastic Search. If you don't have a dataset for the source data, create one.

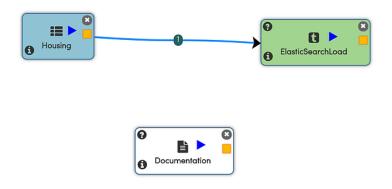
Once the source processor is on the workflow canvas, drag and drop "SaveElasticSearch" processor in the workflow. Configure your Elastic Search processor in the dialog box shown below.

SCHEMA:												
Column Name	id	price	lotsize	bedrooms	bothms	stories	driveway	recroom	fulbose	goshw	airco	garas
Column Type	integer	double	integer	integer	integer	integer	string	string	string	string	string	integ
NDEX NA	ME:0		sp	arkflows/hou	sing							
ELASTIC S	EARCH H	ost : 🛛	los	alhast								
ELASTIC 8	EARCH P	ORT : O	92	00								
ES.INDEX	AUTO.CRI	EATE : O	In	0								
	WAN.ON	LY : 🛛	th	0								
IS.NODE												
	INGEST.C	NLY : O	fok	ie.								

After configuring "SaveElasticSearch" processor, connect your data source processor to Elastic Search processor.

The example workflow below reads a Housing dataset which is in CSV format from HDFS. The 'SaveElasticSearch' takes in the incoming data and loads it into the Elastic Search Index 'sparkflows/housing'.

Note: Documentation processor is just for documentation purposes.



Workflow Execution

When the example workflow above is executed, it reads in the dataset from HDFS and saves it into Elastic Search.

Workflow	Results
Executing N	ode fire.nodes.dataset.NodeDatasetStructured : Oct 14, 2017 4:25:12 PM
Output S	Schema
Executing N	ode fire.nodes.elasticsearch.NodeElasticSearchLoad : Oct 14, 2017 4:25:12 PM
Input Sc	hema
Executing N	ode fire.nodes.doc.NodeDocLarge : Oct 14, 2017 4:25:19 PM
Successfully	finished executing the workflow

Reading data from Elastic Search

Reading data from Elastic Search is easy. Drag and drop 'ReadElasticSearch' process into your workflow and configure it. The screenshot below shows the dialog box for the Elastic Search Read processor.

In the dialog above, 'Refresh Schema' button infers the schema of the index. Thus it is able to pass down the output schema to the next processor making it easy to build workflows.

The SQL field specifies the SQL to be used for reading from Elastic Search. It allows you to limit the columns of interest, and apply where clauses etc.

The Elastic Search processor understands the SQL and translates it into the appropriate QueryDSL. The connector pushes down the operations directly to the source, where the data is efficiently filtered out so that only the required data is streamed back to Spark. This significantly increases the query performance and minimizes the CPU, memory and I/O operations on both Spark and Elastic Search clusters.

The example workflow below reads the data from the sparkflows/housing index in Elastic Search and prints out the first few lines.

ElasticSearchRead	0					
NDEX NAME : O	sporkflows/housing					
LASTIC SEARCH HOST : 🛛	locahost					
LASTIC SEARCH PORT : O	9200					
SPARK TEMPORARY TABLE FOR READING FROM ES : 0	esindex					
SQL FOR READING FROM ELASTIC SEARCH : 0	SELECT * FROM esindex					
Schema Columna : 0 REFREE						
	O COLUMN TYPES OF THE TABLE	O COLUMN FORMATS O				
OLUMN NAMES OF THE TABLE						
	STRING	format				
OLUMN NAMES OF THE TABLE airco bathms	STRING LONG	format format				



Workflow Execution

When the example workflow above is executed, it reads in the index from Elastic Search and displays the first few lines.

Output Sche	no											
ecuting Node (fre.nodes.util.NodePri	ntFirstNRows : Oct 14, 201	7 4.39.25 PM									
Input Schem	0											
Row Volues												
NRCO	BATHRMS	BEDROOMS	DRIVEWAY	FULLBASE	GARAGEPL	GASHW	10	LOTSIZE	PREFAREA	PRICE	RECROOM	STORIES
UringType	LongType	LongType	StringType	StringType	LongType	StringType	LongType	LongType	StringType	RoatType	String5ype	LongType
10	1	2	yes	no	0	no	2	4000	no	38500.0	no	1
	1	2	y#1	no	0	no	5	6360	no	61000.0	no	1
10	1	3	341	no	0	no	8	4160	10	69000.0	no	3
10	1	3	341	no	0	no	13	1700	10	27000.0	no	2
10	1	3	no	no	0	no	14	2680	10	36000.0	no	1
10	2	2	no	y#5	1	no	20	3986	no	45000.0	yes	1
°0	1	2	345	no	0	no	25	4950	no	42000.0	no	1
10	1	2	yes	no	0	no	26	3000	no	42300.0	no	2
195	1	2	yes	y#5	0	no	28	4960	no	44000.0	no	1
0	1	2	945	10	0	10	34	4500	no	\$1500.0	00	

Processing multiple files

This workflow reads in multiple files available in specific directory. It then filters and calculates number of bedrooms with specific prices and then prints the results.

Workflow

Below is the workflow. It does the following:

- Reads multiple csv files available in specific directory.
- Filters it to calculate number of bedrooms with specific prices.
- Prints the results.



Reading CSV files

It reads multiple CSV files available in specific directory using ReadCSV processor.

Processor Configuration

Processor Output

PUT STORAGE I	.EVEL : O			DEFAULT					٣				
••••				/user/spar	kflows/housing	/				BROWSE HD	FS		
					VIEW FILE								
RATOR : 😡													
NER : O				true					Ŧ				
P MALFORMED	:0			false					,				
EMA COLUMNS	:0 REFRESH	SCHEMA O		10.00									
UMN NAMES F		OCHEMOX 0	•	LUMN TYPES FOR	THE CSV D		co	LUMN FORMAT	S FOR THE CSV	ด			
										-			
	de fire.nodes.c	dataset.NodeD	latasetCSV : 1.	Aug 7, 2019 4:5	6:22 AM							OK CAN	
xecuting Nor Row Values		dataset.NodeD	latasetCSV : 1 .	Aug 7, 2019 4:5	6:22 AM								
xecuting Nor Row Values	5					driveway	recroom	fulbase	aashw	airca	gargaepl	,	
Row Values	s price	latsize IntegerType	tatasetCSV : 1 . bedrooms IntegerType	Aug 7, 2019 4:5	6:22 AM stories IntegerType	driveway StringType	recroom StringType	fullbase StringType	gashw StringType	airco StringType	garagepl		
Row Values	s price	lotsize	bedrooms	bathrms	stories							prefarea	
Row Values Row Values IntegerType	s price IntegerType	lotsize IntegerType	bedrooms IntegerType	bathrms IntegerType	stories IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	prefarea StringType	
xecuting Nor Row Values Row Values IntegerType 1 2	Price IntegerType 42000	lotsize IntegerType 5850	bedrooms IntegerType 3	bothrms IntegerType 1	stories IntegerType 2	StringType yes	StringType no	StringType yes	StringType no	StringType no	IntegerType	prefarea StringType no	
Row Values Row Values Id IntegerType 1 2 3	Price IntegerType 42000 38500	IntegerType 5850 4000	bedrooms IntegerType 3 2	bothrms IntegerType 1 1	stories IntegerType 2 1	StringType yes yes	StringType no no	StringType yes no	StringType no no	StringType no no	IntegerType 1 0	prefarea StringType no no	
-	price IntegerType 42000 38500 49500	IntegerType 5850 4000 3060	bedrooms IntegerType 3 2 3	bothrms IntegerType 1 1 1	stories IntegerType 2 1 1	StringType yes yes yes	StringType no no no	StringType yes no no	StringType no no no	StringType no no no	IntegerType 1 0 0	prefarea StringType no no no no	

Filter its data

It then filters to calculate number of bedrooms with specific prices using SQL processor.

Processor Configuration

OLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
OLUMN TYPE	integer	integer	integer	integer	integer	integer	string	string	string	string	string	integer	string
OLUMN FORMAT													
					emp_table								
L : 😧													
		rom fire te	mp_table WH	IERE price BETW	EN 30000 and	100000							
1 select bedroc													2

Processor Output

Print the results

It will print the results with the output required after filter aggregation.

Processor Configuration

Executing Node fire.nodes.etl.NodeSQL :	· · · · · · · · · · · · · · · · · · ·		
Input Schema			
Row Values			
Row Values			
bedrooms		price	
IntegerType		IntegerType	
3		42000	
2		38500	
3		49500	
3		60500	
PrintNRows @ @NedePrintFireNR	993		,
	DEFAULT	٣	
TPUT STORAGE LEVEL : 🖗	DEPAGET		
TPUT STORAGE LEVEL : O	Row Values		

Processor Output

rintNRows			$\langle \rangle$
Executing Node fire.nodes.util.NodePrintl	firstNRows : 3 Aug 7, 2019 5:17:33 AM		
Input Schema			
Row Values			
Row Values			
		price	
bedrooms		price IntegerType	
bedrooms			
bedrooms IntegerType		IntegerType	
bedrooms IntegerType 3		IntegerType 42000	

Saving Data to HIVE

As par of your data pipeline or workflow, you might want to save data to HIVE after it has been read from a data source, cleaned and transformed. After data is saved in HIVE it can be read from another workflow or accessed through BI tools such as Tableau.

Cluster vs Standalone Mode

In your workflow, drag and drop a "SaveAsHIVETable" processor. Configure the processor to save your data into HIVE as a table which can be read later.

Note: Fire Insights can run in cluster mode or in the standalone mode. These settings are in Administration/Configuration.When connecting to HIVE, Sparkflows must be running in cluster mode on an edge node of a Hadoop cluster. HIVE settings have to be correctly set under Administration/Configuration-> app.runOnCluster.

The example workflow below, contains "SaveAsHIVETable" processor. It reads Housing dataset and saves it into the HIVE 'housing_table'.

When the example workflow is executed, data is written into HIVE table 'housing_table'.

ategory:				iption: -					Save	Clear	Back E	Execute
						Prope	erty	Value	}			
						HIVE '	Table	housing_	table			
	:					Overw Outpu						
•	_				_							
(Structured	8	0→		3 3				
								stable				
•			Dataset	Structured								
•			Dataset	Structured								
•			Dataset	Structured								
			Dataset	Structured								
•			Dataset	Structured								
 4 4 4 Characterized Node f 	ine nodes save Node	SaveAsTable	Dataset	Structured								
 Control of the second se	ine nodes save Node	Garan Aa Table Bohaize	Dataset	battyres	stories	chreway	recrosm	futbase	Özepan	airco	gungepi	prefarea
Checking Node 1	: : : :				stories integoritype	delvenuary SorrigType		Autilianse Stringligze	gaahw SoringType	airco StringTjse	garappi integritype	prefarea StregType
Creating Node 1 Executing Node 1 IntegerType	a price DoubleType	lotsize	bedrooms	batterns			recroom					
Creating Node 1 Executing Node 1 IntegerType	a price DoubleType	lotsize	bedrooms	batterns			recroom		StringType			
Checking Node 1	a price DoubleType	lotsize	bedrooms	batterns			recroom					

The 'housing_table' gets created with the schema of the Housing Dataset.

le - housing_table	
0_	
Name	Туре
id	int
price	double
lotsize	int
bedrooms	int
bathms	int
stories	int
driveway	string
recroam	string
fulbase	string
gashw	string
airco	string
garagepl	int
	string

Writing to Parquet Files

Fire Insights enables you to write your Dataframe to Parquet Files.

Workflow for writing to Parquet file

Below is a workflow example which reads in transaction data. It then writes it out to Parquet files.

	This workflow reads in the Trans Dataset. It then saves it in Parqu		
1 🎞 🕨	0		
Transaction Dataset		SaveParquet	
	··· ×	· • • • ;	×
This node creates Data Transaction dataset	frame by reading	Saves the Dataframe in the specified location in Parquet Format.	

DatasetStructured Processor

Node DatasetStructured creates a Dataframe of your dataset named Transaction Dataset by reading data from HDFS, HIVE etc. which had been defined earlier in Fire by using the Dataset feature.

As a user you have to select the Dataset of your interest as shown below.

1 DatasetStructured2	NodeDatasetStructured Details			2 ×
OUTPUT STORAGE LEVEL : O		DEFAULT Transaction Dataset	۲ ۲	
		Transaction Caraser	*	OK CANCEL

SaveParquet Processor

SaveParquet processor saves the incoming DataFrame into the specified path in Parquet Format. When running on Hadoop, Parquet files gets saved into HDFS.

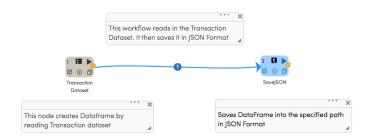
The DataFrame might be written as multiple part files in the specified folder, depending on the size and partition of the DataFrame.

NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double
FORMAT											
UTPUT STOR/	GE LEVEL : 😡				DEFAULT					v	
(TH • : 🛛					parquetdir					BROWSE HDPS	NEW FILE
VE MODE : C					Overwrite					~	
RTITION COL	UMN NAMES	.0		Available					Selected		•
				id : integer chain : integer dept : integer category : inte company : inte brand : integer date : string productsize : productsize : productsize : purchaseana	iger eger r double ure : string rtity : integer				→ ←		

Writing to JSON Files

Fire Insights enables you to write your DataFrame to JSON Files.

Workflow for writing to JSON file



Reading From Dataset

Node TransactionDataset creates DataFrame of your dataset named 'Transaction Dataset' by reading data from HDFS, HIVE etc. which had been defined earlier in Fire by using the Dataset feature. As a user you just have to select the Dataset of your interest and configure the details as shown below.

1 DatasetStructured?	urbured Details	
OUTPUT STORAGE LEVEL : 0	DEFAULT	~
DATASET *: 0	Transaction Dataset	~
		OK CANCEL

SaveJSON Processor Configuration

Node SaveJSON saves DataFrame into the specified path in JSON Format. When running on Hadoop, JSON files gets saved into HDFS.

2 SavejSC	0N2 0	interreption										×
NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount	•
TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double	
FORMAT												
OUTPUT STORA	GE LEVEL : O				DEFAULT					~		1
PATH • : 🛛					json					BROWSE HOPS	ICW FILE	
SAVE MODE : 0					Append					~		1
PARTITION COL	JMN NAMES : 🛛			Available id : integer					Selected		*	٠
				 a meger chain : Integer dept : Integer category : Integ company : Integ brand : Integer date : string productrize : do productrize : do productrizes until purchasequanti purchasequanti 	er uble e : string ty : integer			1	<u>ب</u>			
											OK CA	WCEL

Reading and Writing from MongoDB

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need. Here we are loading data from HDFS and Saving it into MongoDB.

Workflow for Loading data into MongoDB

The below workflow reads in the Sample Dataset which is in CSV format from HDFS.

It then saves the data into MongoDB.

The below diagram shows the dialog box for the SaveMongoDB Processor.



Workflow Execution

When we execute the Workflow, it reads in the dataset from HDFS and loads it into MongoDB.

Workflow Result
Executing Nade fire.nodes.dataset.NodeDatasetCSV : 1 May 10, 2019 7:23:23 AM
Output Schema
Executing Node fire.nodes.mongodb.NodeSaveMongoDB : 2 May 10, 2019 7:23:25 AM
Input Schema
Output Schema
Successfully finished executing the workflow

Workflow for Reading data from MongoDB

The below workflow reads Data in MongoDB.It then prints the data.

The below diagram shows the dialog box for the ReadMongoDB Processor.

In the above dialog, the 'Refresh Schema' button infers the schema of the collections. Thus it is able to pass down the output schema to the next Processor making it easy for us to build the workflow.

Workflow Execution

When we execute the Workflow, it reads in the Sample collection from MongoDB and displays the first few lines.

We see that the Sample data records we wrote to MongoDB in the first workflow is read back now.

12.1.2 Data Exploration



UTPUT STORAGE LEVEL : 🕑	DEFAULT		•
AONGODB URI : 😧	mongodb://localhost:27017/		
NONGODB DATABASE : O	test		
NONGODB TABLE : O	sampledatacollection		
CHEMA COLUMNS : O REFRESH SCHEMA			
COLUMN NAMES OF THE TABLE Ø	COLUMN TYPES OF THE TABLE O		COLUMN FORMATS Ø
_id	STRUCT	۲	StructField(_id,StructType(StructField(oid,StringTyp
d11	STRING		format

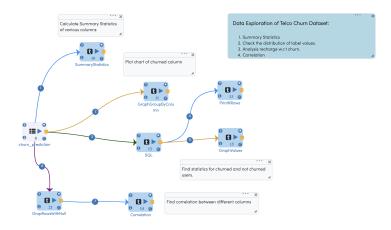
Output Schema			
Executing Node fire.nodes.util.NodePrintFirstNRows : 2 May 10,	2019 7:36:56 AM		
D Input Schema			
Row Values			
_ID	om	DT2	ID
StructType(StructField(oid,StringType,true))	StringType	StringType	IntegerType
[5cd526f79oo493ebf5e99eef]	2003-07-25	2007-03-11	1
[5cd526179aa493eb15e99e10]	2000-05-01	2013-12-26	2
[5cd526f79aa493ebf5e99ef1]	2006-07-21	2000-08-31	3
[5cd526f79oo493ebf5e99ef2]	2012-05-18	2001-05-12	4
[5cd526f79aa493ebf5e99ef3]	1999-10-20	2015-08-05	5
[5cd526f79oo493ebf5e99ef4]	2007-08-31	2012-03-05	6
	0000 01.15	2002-07-26	7
[Scd526f79oo493ebf5e99ef5]	2012-01-15		
[5cd526f79oo493ebf5e99ef5] [5cd526f79oo493ebf5e99ef6]	2007-06-14	2012-12-18	8
		2012-12-18 2015-07-10	8

Telco Churn Data Exploration

Data Profiling is extremely helpful in understanding the data. Fire Insights provides a number of processors for users to profile their data.

Workflow for Data Profiling

Below is a workflow which profiles the Telco Churn Dataset.



Input Telco Churn Data

The input dataset looks like below:

state	account_length	area_code	phone_number	intl_plan	voice_mail_plan	number_vmail_messages	today_day_minutes	today_day_calls	today_day_change	total_eve_minute
StringType	DoubleType	DoubleType	StringType	StringType	StringType	DoubleType	DoubleType	DoubleType	DoubleType	DoubleType
KS	128.0	415.0	382-4657	no	yes	25.0	265.1	110.0	45.07	197.4
ОН	107.0	415.0	371-7191	no	yes	26.0	161.6	123.0	27.47	195.5
NJ	137.0	415.0	358-1921	no	no	0.0	243.4	114.0	41.38	121.2
ОН	84.0	408.0	375-9999	yes	no	0.0	299.4	71.0	50.9	61.9
ОК	75.0	415.0	330-6626	yes	no	0.0	166.7	113.0	28.34	148.3
AL	118.0	510.0	391-8027	yes	no	0.0	223.4	98.0	37.98	220.6
MA	121.0	510.0	355-9993	no	yes	24.0	218.2	88.0	37.09	348.5
мо	147.0	415.0	329-9001	yes	no	0.0	157.0	79.0	26.69	103.1
LA	117.0	408.0	335-4719	no	no	0.0	184.5	97.0	31.37	351.6
WV	141.0	415.0	330-8173	yes	yes	37.0	258.6	84.0	43.96	222.0
IN	65.0	415.0	329-6603	no	no	0.0	129.1	137.0	21.95	228.5

Workflow Execution Result

When the above workflow is executed, it produces the below results. The good thing about Fire Insights is that the Data Profiling runs in a distributed fashion. So, whatever the number of records in the input dataset, it scales seamlessly.

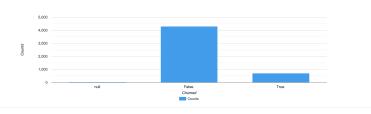
Summary Statistics

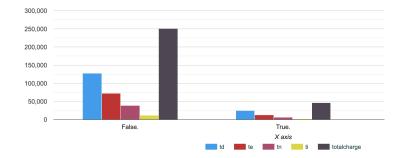
Counts by Churned Column

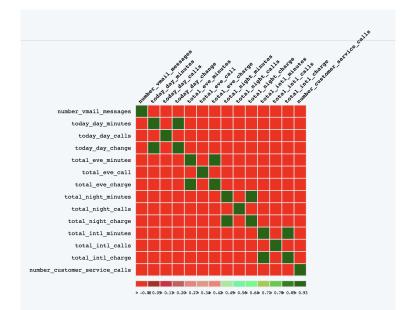
Graph of counts of various attributes for Churned and Not Churned customers

Summary									
summary	number_vmail_messages	today_day_minutes	today_day_calls	today_day_change	total_eve_minutes	total_eve_call	total_eve_charge	total_night_minutes	total_night_call
count	5000	5000	5000	5000	5000	5000	5000	5000	5000
mean	7.755	180.289	100.029	30.65	200.637	100.191	17.054	200.392	99.919
min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25_percentile	0.0	143.7	87.0	24.43	166.3	87.0	14.14	166.9	87.0
50_percentile	0.0	180.1	100.0	30.62	201.0	100.0	17.09	200.4	100.0
75_percentile	17.0	216.2	113.0	36.75	234.1	114.0	19.9	234.7	113.0
max	52.0	351.5	165.0	59.76	363.7	170.0	30.91	395.0	175.0
stdev	13.546	53.895	19.831	9.162	50.551	19.826	4.297	50.528	19.959
variance	183.505	2904.639	393.276	83.944	2555.435	393.09	18.463	2553.057	398.349









Correlation Matrix

12.1.3 Machine Learning

Telco Churn Prediction

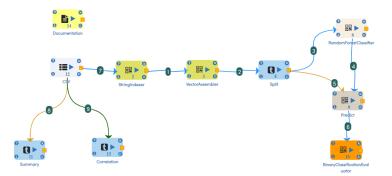
Fire Insights enable us to create a Random Forest Model to predict churn and evaluate the results.

The dataset is artificial Churn Data based on claims, similar to real world. It is taken from the following location.

- https://www.sgi.com/tech/mlc/db/
- https://www.sgi.com/tech/mlc/db/churn.all
- https://www.sgi.com/tech/mlc/db/churn.name

KS, 128, 415, 382-4657, no, yes, 25, 265.1, 110, 45.07, 197.4, 99, 16.78, 244.7, 91, 11.01, 10, 3, 2.7, OH, 107, 415, 371-7191, no, yes, 26, 161.6, 123, 27.47, 195.5, 103, 16.62, 254.4, 103, 11.45, 13.7, 3, NJ, 137, 415, 358-1921, no, no, 0, 243.4, 114, 41.38, 121.2, 110, 10.3, 16.62, 254.4, 103, 11.45, 13.7, 3, OH, 84, 408, 375-9999, yes, no, 0, 299.4, 71, 50.9, 61.9, 88, 5.26, 196.9, 89, 8.86, 6.6, 7, 1.78, 2, F OK, 75, 415, 330-6226, yes, no, 0, 166.7, 113, 28.34, 148.3, 122.2, 12.61, 186.9, 121, 8.41, 10.1, 3, 2.7, AL, 118, 510, 391-8027, yes, no, 0, 223.4, 98, 37.09, 248.5, 108, 29.62, 212.6, 118, 9.57, 7.5, 7, 2.0 NO, 147, 415, 329-9031, yes, no, 0, 157, 79, 26.69, 103.1, 94, 8.76, 211.8, 96, 9.53, 7.1, 6, 1.92, 0, LA, 117, 408, 335-4719, no, no, 0, 184.5, 97, 31.37, 351.6, 80, 29.89, 215.8, 90, 9.71, 8.7, 4, 235, 1 N, 65, 415, 329-6603, no, no, 0, 129.1, 137, 21.95, 228.5, 83, 19.42, 208.6, 111, 9.4, 12.7, 6, 3.43, 117, 408, 363-4107, no, no, 0, 128.4, 97, 31.91, 314.4, 145, 139, 196, 94.8, 29, 1.1, 2.7, 6, 3.43, 117, 455, 136-926, no, no, 0, 129.1, 137, 21.95, 228.5, 83, 19.42, 208.8, 111, 9.4, 12.7, 6, 3.43, 114, 64, 136, 366-1107, no, no, 0, 128.8, 96, 21.9, 104.9, 71, 8.92, 141.1, 128, 6.35, 11.2, 2, 3.02, 1 A, 168, 408, 363-1107, no, no, 0, 128.6, 96, 21.9, 104.9, 71, 8.92, 141.1, 128, 6.35, 11.2, 2, 3.02, 1 A, 153, 565, 212.3, 56, 317.8, 97, 27.01, 160.6, 128, 7.23, 5.4, 91, 466, 108, 546, 369-468, 300, no, 0, 9, 126.7, 70, 20.52, 307.2, 76, 26.11, 203, 99, 9.14, 13.1, 6, 3.54, 4, 91, 361, 366-2292, no, no, 0, 332.9, 67, 56.59, 317.8, 97, 27.01, 160.6, 128, 7.23, 5.4, 9.146, 10, 55, 408, 366-3100, no, no, 0, 332.9, 67, 56.59, 317.8, 97, 27.01, 160.6, 128, 7.23, 5.4, 9.146, 10, 85, 408, 386-480, no, no, 0, 132.9, 165, 512.3, 51, 128, 4.3, 73, 21.9, 143, 333-92.80, 90, 90, 23.88, 803, 75, 4.02, 13.8, 4, 3.73, 142, 134, 145, 337-2782, no, no, 0, 190.7, 114, 32.42, 218.2, 112, 18.55, 129.6, 121, 538, 814, 3, 2.151, 144, 15, 336-1298, no, no, 0, 190.7, 114, 32.42, 218.2, 112, 18.55, 129.6, 121, 538, 814

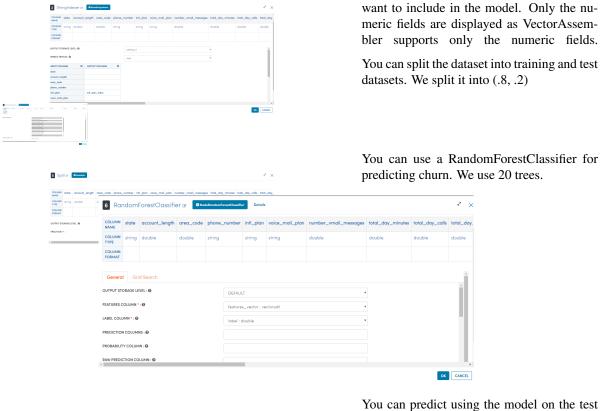
Below is the workflow you can use for creating the model for Churn Prediction.



The workflow performs the following steps:

- Reads in the dataset from a tab separated file
- Applies StringIndexer on the field "intl_plan"
- Applies VectorAssembler on the fields we want to model on
- Splits the dataset into (.8, .2)
- Performs Random Forest Classification

- Performs prediction using the model generated on the remaining 20% dataset
- Finally evaluates the prediction results



You can predict using the model on the test dataset.

In the VectorAssembler, select the fields you

COLUMN NAME	state	account_length	area_code	phone_number	intl_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_day
COLUMN	string	double	double	string	string	string	double	double	double	double
COLUMN FORMAT										
COLUMN NAME	state	account_length	area_code	phone_number	intl_plan	voice_mail_plan	number_vmail_messages	total_day_minutes	total_day_calls	total_da
COLUMN TYPE	string	double	double	string	string	string	double	double	double	double
COLUMN										

You can evaluate the quality of our results.

Next, You can execute the workflow.

From the evaluator You get the following results:

Bike Rental Prediction

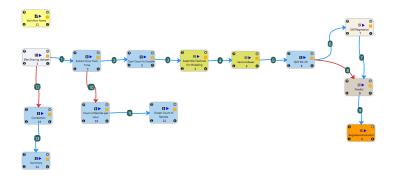
This workflow reads in a dataset. It then Predicts the number of bikes to be rented in any given hour.

BinaryClassificationEvaluator		
Accuracy is 0.9284253578732107		
Confusion Matrix		
Confusion Matrix		
	Predicted_Label	Count
Confusion Matrix	Predicted_Lobel	Count 54
Confusion Matrix Target_Label		
Confusion Matrix Target_Label 1.0	1.0	54

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Extracts hour from time using datatype timestamp.
- Calculates Count to datatype double.
- Assembles features for modelling.
- Calculates vectorindexer.
- Splits it.
- GBTRegression.
- Prediction.
- RegressionEvaluator.
- Correlation with columns.
- Summary analysis.
- Calculate count for rental per hour.
- Analyse using Graph.



Reading from Dataset

It reads sample Dataset file.

DatasetStructured 2 o Details NodeDatasetStructured		ې ۱
OUTPUT STORAGE LEVEL : O	DEFAULT Bike Sharing dataset	
		OK CANCEL

Processor Output

Row Values											
latetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
TimestampType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType	IntegerType	DoubleType	IntegerType	IntegerType	IntegerType
2011-01-01 00:00:00.0	1	0	0	1	9.84	14.395	81	0.0	3	13	16
2011-01-01 01:00:00.0	1	0	0	1	9.02	13.635	80	0.0	8	32	40
0.00:00:50 10-10-105	1	0	0	1	9.02	13.635	80	0.0	5	27	32
2011-01-01 03:00:00.0	1	0	0	1	9.84	14.395	75	0.0	3	10	13
2011-01-01 04:00:00.0	1	0	0	1	9.84	14.395	75	0.0	0	1	1
2011-01-01 05:00:00.0	1	0	0	2	9.84	12.88	75	6.0032	0	1	1
2011-01-01 06:00:00.0	1	0	0	1	9.02	13.635	80	0.0	2	0	2
2011-01-01 07:00:00.0	1	0	0	1	8.2	12.88	86	0.0	1	2	3
2011-01-01 07:00:00:0	1	0	0	1	8.2	12.88	86	0.0	1	2	3
2011-01-01 06:00:00.0	1	0	0	1	9.84	14.395	75	0.0	1	7	8
2011-01-01 09:00:00.0	1	0	0	1	13.12	17.425	76	0.0	8	б	14

Extract hour from time using datatype timestamp

It Extracts hour from time using datatype timestamp using DateTimeFieldExtract Node.

Processor Configuration

CHEMA :												
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer
COLUMN FORMAT	dd/MM/yyyy HH:mm											
XTRACT YEAR : 🕹			da tru	tetime : timestam	Þ				•			
DUTPUT STORAGE LEV	ΈL: Θ		DE	FAULT					٣			
XTRACT MONTH : @												
			tru	0					•			
XTRACT DAY OF MOR	пн: 0		tru	e					*			
EXTRACT HOUR : O			tru	ie.								
EXTRACT MINUTE : O			fa	sa					*			
EXTRACT SECOND : 0			fa									

Processor Output

Input Schema													
Row Values													
datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	cesual	registored	count	datetime_year	
TimestompType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType	IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	
2011-01-01 00:00:00.0	1	0	o	1	9.84	14.395	81	0.0	3	13	16	2011	
2011-01-01 01:00:00.0	1	0	0	1	9.02	13.635	80	0.0	8	32	40	2011	
2011-01-01 02:00:00.0	1	0	c	1	9.02	13.635	80	0.0	5	27	32	2011	
2011-01-01 03:00:00.0	1	0	0	1	9.84	14.395	75	0.0	3	10	13	2011	
2011-01-01 04:00:00.0	1	0	o	1	9.84	14.395	75	0.0	0	1	1	2011	
2011-01-01 05:00:00.0	1	0	0	2	9.84	12.88	75	6.0032	0	1	1	2011	
2011-01-01 06:00:00.0	1	0	0	1	9.02	13.635	80	0.0	2	0	2	2011	
2011-01-01 07:00:00.0	1	0	0	1	8.2	12.88	86	0.0	1	2	3	2011	
2011-01-01 08:00:00.0	1	0	0	1	9.84	14.395	75	0.0	1	7	8	2011	
2011-01-01 09:00:00.0	1	0	o	1	13.12	17.425	76	0.0	8	6	14	2011	

Calculate Count to datatype double

It Calculates cast the Count field to datatype double using CastColumnType Node.

Processor Configuration

SCHEMA :															
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetime
COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer	integer	integer	integer
COLUMN FORMAT	dd/MM/yyyy HH:mm														
DUTPUT STO	ORAGE LEVEL : 😡				DE	FAULT									
NEW DATA 1	TYPE: 0				sec hol wo we ten alte hun wir cos cos dot dot dot	letime : fim ison : integ iday : integ kingday : i athar : inte tp : double mp : double mp : double mp : double indity : inte dispeed : do ual : intege jatered : intege letime_yec letime_doy UBLE	er spr integer gør le toger toger tager t tager t	or .							
	USTING COLS : O														

Processor Output

Assemble features for modelling

It Assembles features columns into a feature vector using VectorAssembler Node.

			in part of the	15, 2018 3:29:1	C MIN							
Input Schema												
Row Volues												
datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	datetime_year	datetime_mont
TimestampType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType	IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType
2011-01-01 00:00:00.0	1	σ	0	1	9.84	14.395	81	0.0	3	13	2011	1
2011-01-01 01:00:00.0	1	0	0	1	9.02	13.635	80	0.0	8	32	2011	1
2011-01-01 02:00:00.0	1	0	0	1	9.02	13.635	80	0.0	5	27	2011	1
2011-01-01 03:00:00.0	1	0	0	1	9.84	14.395	75	0.0	3	10	2011	1
2011-01-01 04:00:08.0	1	0	0	1	9.84	14.395	75	0.0	0	1	2011	1
2011-01-01 05:00:00.0	1	0	0	2	9.84	12.88	75	6.0032	0	1	2011	1
2011-01-01 06:00:00.0	1	0	0	1	9.02	13.635	80	0.0	2	0	2011	1
2011-01-01 07:00:00.0	1	0	0	1	8.2	12.88	86	0.0	1	2	2011	1
2011-01-01 08:00:00.0	1	0	0	1	9.84	14.395	75	0.0	1	7	2011	1
2011-01-01 09:00:00.0	1	0	0	1	13.12	17.425	76	0.0	8	6	2011	1

	le Features orAssembler	101 2100	ennig ta	•											4
CHEMA :															
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetir
COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	double	integer	integer	intege
	dd/MM/yyyy HH:mm														
IPUT COLU					sea holi wor we cos reg dat dat dat dat terr ote win	FAULT son : integ day : integ kingday : ather : inte nidity : integ istered : in etime_yec etime_yec etime_hou etime_hou p : double mp : double dspeed : c n1 : double	ier ger ger tager in ar : integer nth : integer vofmonth ir : integer le louble	er integer				•			
UTPUT CO	LUMN : O				feat	ure_vecto	e								

Processor Output

Calculate vectorindexer

It identifies categorical features and index them using vectorindexer Node.

Processor Configuration

Processor Output

Split it

It will split our dataset into seperate training and test sets using split Node.

OK

xecuting Node fi	re.nodes.ml.Ne	odeVectorAsse	mbler : 4 Nov	15, 2018 3:33:1	5 AM							
Row Values												
datetimo	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	cesual	registered	datetime_year	datotimo_mon
TimestompType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType	IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType
2011-01-01 00:00:00.0	1	0	c	1	9.84	14.395	81	0.0	3	13	2011	1
2011-01-01 01:00:00.0	1	0	0	1	9.02	13.635	80	0.0	8	32	2011	1
2011-01-01 02:00:00.0	1	0	c	1	9.02	13.635	80	0.0	5	27	2011	1
2011-01-01 03:00:00.0	1	0	c.	1	9.84	14.395	75	0.0	з	10	2011	1
2011-01-01 04:00:00.0	1	0	c	1	9.84	14.395	75	0.0	0	1	2011	1
2011-01-01 05:00:00.0	1	0	0	2	9.84	12.88	75	6.0032	0	1	2011	1
2011-01-01 06:00:00.0	1	0	c	1	9.02	13.635	80	0.0	2	c	2011	1
2011-01-01 07:00:00.0	1	0	c	1	8.2	12.88	86	0.0	1	2	2011	1
2011-01-01 08:00:00.0	1	0	0	1	9.84	14.395	75	0.0	1	7	2011	1
2011-01-01 09:00:00.0	1	0	0	1	13.12	17.425	76	0.0	8	6	2011	1

CHEMA :															
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetin
COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	double	integer	integer	integer
COLUMN FORMAT	dd/MM/yyyy HH:mm														
DUTPUT ST	DRAGE LEVEL : 😧				DE	AULT						•			
NPUT COLL	JMN : 😡				fea	ture_vecto	or : vector	udt							
DUTPUT CO	LUMN : 😡				feat	ure_vecto	r_index								
MUMIXAN	CATEGORIES : 0				31										

stecoling Hode II	re.nodes.ml.N	odel/ectorInde	xer : 5 Nov 15,	2018 3:36:09 /	M							
Input Schema												
Row Values												
datetime	Rection	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	datetime_year	datetime_mai
TimestampType			IntegerType	IntegerType	DoubleType	DoubleType			IntegerType		IntegerType	IntegerType
2011-01-01 00:00:00.0	1	0	0	1	9.84	14.395	81	0.0	3	13	2011	1
2011-01-01 01:00:00.0	1	o	0	1	9.02	13.635	80	0.0	8	32	2011	1
2011-01-01 02:00:00.0	1	0	0	1	9.02	13.635	80	0.0	5	27	2011	1
2011-01-01 03:00:00.0	1	0	0	1	9.84	14.395	75	0.0	3	10	2011	1
2011-01-01 04:00:00.0	1	0	0	1	9.84	14.395	75	0.0	0	1	2011	1
2011-01-01 05:00:00.0	1	0	0	2	9.84	12.88	75	6.0032	0	1	2011	1
2011-01-01 06:00:00.0	1	0	0	1	9.02	13.635	80	0.0	2	0	2011	1
2011-01-01 07:00:00.0	1	0	0	1	8.2	12.88	86	0.0	1	2	2011	1
2011-01-01 08:00:00.0	1	0	0	1	9.84	14.395	75	0.0	1	7	2011	1
2011-01-01 09:00:00.0	1	0	0	1	13.12	17.425	76	0.0	8	6	2011	1

tetime														
	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetim
nestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	double	integer	integer	integer
/MM/yyyy I:mm														
GE LEVEL : O				DE	FAULT						•			
	MM/yyyy mm	MM/yyyy mm	MM/yyyy mm	MM/yyyy mm	MM/yyyy mm		MW/yyyy LELEVEL:0 DEFAULT			MM/yyyy LELEVEL: 0 DEFAULT		MM/yyyy Interest • DEFAULT •		MM/yyyy ELEVEL: 0

Processor Output

	ire.nodes.ml.N	odeSplit : 6 No	rv 15, 2018 3:38	:37 AM								
Input Schema												
Row Values												
datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	datetime_year	datetime_mon
TimestampType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType	IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType
2011-01-01 07:00:00.0	1	0	0	1	8.2	12.88	86	0.0	1	2	2011	1
2011-01-01	1	0	0	1	13.12	17.425	76	0.0	8	6	2011	1

GBTRegression

It validates held out test sets in order to know about high confidence using GBTRegression Node.

ОК

Processor Configuration

Processor Output

Prediction

It will make prediction on future data using Prediction Node.

Processor Configuration

Processor Output

RegressionEvaluator

It validates held out test sets in order to know about high confidence using RegressionEvaluator Node.

HEMA															
COLUMN IAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	date
OLUMN YPE	limestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	double	inleger	integer	integ
	dd/MM/yyyy HH:mm														
ıb G	rid Search														
TPUT STO	DRAGE LEVEL : 🕹				DE	AULT									
ATURES C	OLUMN : O				fea	ture_vects	or_index :	vectorudt				,			
BEL COLU	IMN : O				cou	int : double	,								
EDICTION	COLUMNS : O														
PURITY : (9				var	iance						•			
SS FUNCT	10N : Ø				squ	ared									
IX BINS : (Ð				32										
AX DEPTH	0				5										
X ITERAT	IONS : O				6										
N INFORM	ATION GAIN : 🥹				0.0										
N INSTAN	CES PER NODE : 0				1										
IBSAMPLII	NG RATE : \varTheta				1.0										
ED :															
EP SIZE : C					0.1										
CHE NODI					fals	e						•			
	T INTERVAL : O				10										
X MEMOR	RY : 😡				256										

GBTRegression
GBTRegression
Connot exercit fire,notes.nl.holdstiftegression hole in inoritize failur

 Predict 7
 0
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1

Predict		1
Predict		
Cannot execute fire.modes.ml.NodePredict Node in Workflow Editor		
		OK

		sionEvaluato ressionEvaluato		Detail	s											9
	SCHEMA :															
	COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetime
	COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	double	integer	integer	integer
	COLUMN FORMAT	dd/MM/yyyy HH:mm														
	OUTPUT ST	ORAGE LEVEL : 😡				DE	FAULT						٠			
	LABEL COL	UMN : 🛛				co	unt : doubl	-					٠			
	PREDICTIO	N COLUMN : O				pr	ediction : d	ouble					•			
	METRIC NA	ME : 😡				m	se					_	٠			
Processor Output	Regress	ionEvaluator													ΟK	CANCEL
																ок

Correlation with columns

It will analyse correlation between various columns using Correlation Node.

Processor Configuration

CHEMA :												
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
COLUMN TYPE	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer
COLUMN FORMAT	dd/MM/yyyy HH:mm											
NPUT COLUMN FOR C	ORRELATION : O		sec hol wo we hur cos reg cou terr ote	relation between son : integer rkingday : integer rkingday : integer ather : integer ual : integer istered : integer mp : double mp : double dapeed : double								

Processor Output

Summary analysis

It visualizes our data to get sense of whether the features are meaningful using Summary Node.

Correlation													
Cannot execute fire.m	odes.ml.NodeCorrelation Nod	e in Workflo	/ Editor										
													•
Summary 🛛 🛛	Details												14
SCHEMA :													
COLUMN NAME d	atetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	
COLUMN TYPE	mestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer	
COLUMN FORMAT d	d/MM/yyyy HH:mm												
OUTPUT STORAGE LEVEL :			-	FAULT									
TTLE :													
TITLE :			Bike	Sharing Dataset	Summary								
COLUMN NAMES :			holi wor wec hum cos regi cou sem ater	son : integer day : integer sther : integer ather : integer ual : integer istered : integer nt : integer istered : integer mp : double dispeed : double									

Processor Output

Summary	,
Summary	
Carnot execute fire.rodes.ml.NodeSummary Node in Noriflov Editor	
	OK

Calculate count for rental per hour

It calculates count for rental per hour using query with SQL Node.

Processor Configuration

Processor Output

Analyse using Graph

It will analyse graph with bike rental counts and hours of the day using GraphValue Node.

Processor Configuration

Processor Output

CHEMA :															
COLUMN NAME	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	datetime_year	datetime_month	datetir
COLUMN	timestamp	integer	integer	integer	integer	double	double	integer	double	integer	integer	integer	integer	integer	intege
	dd/MM/yyyy HH:mm														
UTPUT STO	ORAGE LEVEL : 😡				DE	FAULT						•			
EMP TABLE	E : 😜				fire,	_temp_tat	ble								
QL : 0	E: 0 ect datetime_hour	, sum(cou	nt) as cou	nt from fire_t		group by d		our order b	/ datetime_ho	ur.					
	ect datetine_hour	r, sua(cou RESH SCHE	-	-	emp_table (group by d		our onder b	(datetime_ho	r					
QL:0	ect datetine_hour		-	-	emp_table (group by d	stetime_b	our order b	_		JUMN FORMA	TS 🛛			
QL:0	olumns : O IIII		-	-	emp_table ;	proup by d	stetime_b	our onder b	_		LUMN FORMA	TS 🖗			

RegressionEvaluator

Input Schema		
Row Values		
datetime_hour	count	
IntegerType	LongType	
σ	16	
1	40	
2	32	
3	13	
4	1	
5	1	
6	2	
7	3	
8	8	
9	14	

Graph Count of Rentals @ NodeGraphValues					n
SCHEMA :					
COLUMN NAME	date	time_hour		count	
COLUMN TYPE	integ	jer		long	
COLUMN FORMAT					
OUTPUT STORAGE LEVEL : O	DEFAULT				
TITLE :	Count of Rentals	per Hour			
X LABEL :	X axis				
Y LABEL :	Y oxis				
CHART TYPE :	Line Chart		٠		
IS STREAMING? : O	false		٠		
X COLUMN :	datetime_hour	: integer	•		
Y COLUMNS :	datetime_hour: count : long	: integor	_		
			*		

gressionEvaluator	4
Executing Node fire nodes graph NodeGraphVolues : 11 Nov 15, 2016 3:48:39 AM	
) input Schemo	
Count of Rentals per Hour	
Count of Rentals per Hour cause cause and an ouer Intercent Intercent Intercent	
40 20 0 1 2 3 4 6 6 7 8 9	
	ОК

Farmers Market Prediction

It demonstrate to predict "the number of farmer's markets in a given zip code" based on the income and taxes paid in a given area using the past data. It seems plausible that areas with higher income have more farmer's markets simply because there is more of a market for those goods. Of course there are many potential holes in this idea, but that's part of the desire to test it.

DataBricks has published a clean approach to build this use case. It feature a Python notebook that demonstrates how to create ML Pipeline to preprocess a dataset, train a Machine Learning model, and make predictions.

Using Fire Insights visual designer, you can try to execute this approach visually and declaratively. This note speaks to that.

As the DataBricks link highlights:

- The first of the two datasets that you can work is the Farmers Markets Directory and Geographic Data. This dataset contains information on the longitude and latitude, state, address, name, and zip code of Farmers Markets in the United States. The raw data is published by the Department of Agriculture. The version on the data that is found in Databricks (and is used in this tutorial) was updated by the Department of Agriculture on
- Dec 01, 2015.
- The second you can work is the SOI Tax Stats Individual Income Tax Statistics ZIP Code Data (SOI). This study provides detailed tabulations of individual income tax return data at the state and ZIP code level and is provided by the IRS. This repository only has a sample of the data: 2013 and includes "AGI". The ZIP Code data show selected income and tax items classified by State, ZIP Code, and size of adjusted gross income. Data are

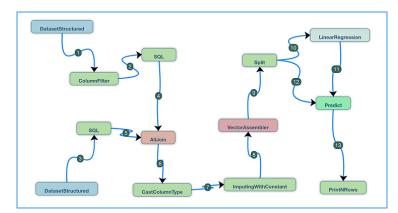
based on individual income tax returns filed with the IRS and are available for Tax Years 1998, 2001, 2004 through 2013. The data include items, such as:

- Number of returns, which approximates the number of households
- Number of personal exemptions, which approximates the population
- Adjusted gross income
- Wages and salaries
- Dividends before exclusion
- Interest received

Below is an overview of the workflow. You can create using the Fire Insights Visual Designer.

This workflow was simply created via the drag and drop capabilities of the Fire Insightss Designer UI. This ability to construct this data processing pipeline (or any DAG - Distributed Acyclic Graph, for that matter) in a WYSIWYG Plug-and-Play manner is a key innovation to continue our community's collective march to on-demand-instant-analytics. Benefits include:

- It opens up the power of ETL and ML (such pre-packaged functionality is available as a catalog of "Nodes") to a wider audience of analysts and semi-technical resources.
- The actual execution can either be local (testing) or can be submitted to a Apache Spark cluster.
- You can see during the adoption that a single workbench improves collaborative iteration across data engineers, data scientists and analysts, which in turn accelerates time-to-market.
- As one might observe, the visual approach doubles up as workflow documentation and hence contributes to solving the data-lineage problem.



This workflow consists of the following steps:

- Using the DatasetStructured Nodes: Read in the data from 2 different datasets Farmers_Markets and Income Tax Return Data per Zip Code (both comma separated files:
- Instead of a CSV, one can easily read it from a data-lake or a Persistence Store (HDFS/RDBMS/NoSQL).
- Using the ColumnFilter node: Filter out the following columns from the Income Tax Return dataset and pass it to a SQL query node, so we can do further computation.
- State
- Zipcode
- MARS1 Single Returns
- MARS2 Joint Returns
- NUMDEP Number of Dependents
- A02650 Tota Income Amount
- A00300 Taxable Interest Amount
- A00900
- A01000

- Using the SQL Node: Execute the following SQL to get the various aggregates from the filtered data from the Income Tax Return dataset
- select zipcode, sum(MARS1) as single_returns, sum(MARS2) as joint_returns, sum(NUMDEP) as numdep, sum(A02650) as total_income_amount, sum(A00300) as taxable_interest_amount from fire_temp_table group by zipcode
- Using another SQL Node: Extract certain columns from the Farmers_Market dataset using the below SQL query:
- select cast(zip as int) as zip, count(*) as count from fire_temp_table group by zip
- Using the AllJoin node Join the two filtered datasets using the following query:
- select a.zipcode , a.single_returns, a.joint_returns, a.numdep, a.total_income_amount, a.taxable_interest_amount, b.count, b.zip from fire_temp_table1 a LEFT OUTER JOIN fire_temp_table2 b ON(a.zipcode=b.zip)
- Using the CastColumnType Node change the column type of the count column from Long to Double
- Using the ImputingWithConstant node, fill the blanks across all columns with constants.
- Using the VectorAssembler node, concatenate columns single_returns, joint_returns, numdep, total_income_amount, taxable_interest_amount into a feature vector feature_vector
- Using Split node: Split the dataset into (.7, .3)
- 70% rows are used for training and 30% are used for prediction
- The model is evaluated based on how it predicts on the remaining 30%.
- Using the LinearRegression Node Perform LinearRegression:
- This is a Spark MLLib provided algorithm that Sparkflows exposes to you as a plug-andplay "node". LinearRegression from SparkML.
- Using Predict Node: Perform prediction using the model generated on the remaining 30% dataset
- Finally evaluate the result using the PrintNRows node.

First Dataset

Column Filter

SQL

Second Dataset

SQL

AllJoin - Join the two datasets

CastColumnType

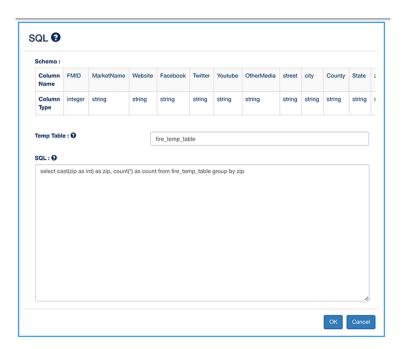
Workflow	
Flow Id:	20
Description:	Farmer markets prediction by Zip code leveraging tax data.
Nodes:	Nodes
	DatasetStructured
	ColumnFilter
	SQL
	DatasetStructured
	SQL
	AllJoin
	CastColumnType
	ImputingWithConstant
	VectorAssembler
	Split
	LinearRegression
	Predict
	PrintNRows

DatasetStructured		
Dataset :	IRS_2013_AGI	\$
		OK Cancel

Schema :												
Column Name	STATEFIPS	STATE	zipcode	agi_stub	N1	MARS1	MARS2	MARS4	PREP	N2	NUMDEP	'
Column Type	integer	string	integer	integer	double	double	double	double	double	double	double	•
				A00100 : N02650 : A02650 : N00200 :	ble double double double ble couble double double double double							

Schema :									
Column Name	STATE	zipcode	MARS1	MARS2	NUMDEP	A02650	A00300	A00900	A01000
Column Type	string	integer	double	double	double	double	double	double	double
Temp Table : 😡			fire_temp_ta	ble					
SQL : 0									
select zipcode, s								um(A02650	0) as
								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as
select zipcode, s								um(A02650	0) as

DatasetStructured		
Dataset :	farmer_market	\$
		OK Cancel



Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount
Column Type	integer	double	double	double	double	double
emp Table1	:0	Г	fire_temp_table1			
emp Table2	:: 0	[fire_temp_table2	2		
QL : 0						
		a_returns, a.joint_re e1 a LEFT OUTER 、			ne_amount, a.taxable_inte zipcode=b.zip)	rest_amount, b.count,

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	long	integ
			joint_r numd total_i	e_interest_ : long	uble			

ImputingWithConstant

Name	Schema :									
Type Columns Constants Zipcode Single_returns joint_returns numdep total_income_amount taxable_interest_amount count	Column Name	zipcode	single_re	turns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
zipcode single_returns joint_returns numdep total_income_amount taxable_interest_amount count				double	double	double	double	double	inte	
zipcode single_returns joint_returns numdep total_income_amount taxable_interest_amount count				Colu	imns	0	Constants	9		
joint_returns numdep total_income_amount taxable_interest_amount count				zipco	ode	-				
numdep total_income_amount taxable_interest_amount count				single_returns		[
total_income_amount taxable_interest_amount count				joint_	_returns		[
taxable_interest_amount count				nume	dep		ļ			
count							ļ			
				_		ount	ļ			
zip					t		l			
				zip			l			

VectorAssembler

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	double	inte
			joint_r numd total_i taxabl	_returns : do eturns : dou ep : double income_am le_interest_ : double	uble			

Split

LinearRegression

Predict

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	double	inte
Fraction 1								
			.7					

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	double	inte
Features C	Column : C	•	feature	_vector : ve	ectorudt	¢		
Label Colu	ımn : 😡		count :	double		\$		
Prediction	Columns	: O						
Fit Interce	pt : 😡		true			\$		
Maximum	Iterations	: O	10					
Regulariza	tion Parar	m : 🖸	0.0					
ElasticNet	Param : 🕻	•	0.5					
Solver : 😡			auto			\$		

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	double	inte

Print N Rows

Schema :								
Column Name	zipcode	single_returns	joint_returns	numdep	total_income_amount	taxable_interest_amount	count	zip
Column Type	integer	double	double	double	double	double	double	inte
Num Row	s to Print :		10					

Next you can execute the workflow and it come up with predictions for number of farmers markets in a zip code.

ripcode	single_returns	joint,returns	numdep	total_income_amount	taxable_interest_amount	aip	count	feature_vector	prediction
integer7;pe	DoubleType	Double7ype	DoubleType	DoubleType	DoubleType	Integer7ype	Double7ype	org apache spark milib linalg Vector/JDT0171b0bce	DoubleType
30631	270.0	180.0	490.0	19845.0	126.0	0	0.0	[270.0,180.0,490.0,19845.0,126.0]	0.30575029684995314
32631	90.0	120.0	80.0	16322-0	85.0	0	0.0	[90.0,120.0,80.0,16322.0,85.0]	0.30624634920807126
34431	1300.0	1520.0	1490.0	128482.0	1252.0	0	0.0	[1300.0,1520.0,1490.0,128482.0,1252.0]	0.30490547793436895
81431	50.0	60.0	50.0	5397.0	4.0	0	0.0	[50.0.60.0.50.0.5397.0.6.0]	0.3061853327404323
81431	50.0	60.0	50.0	5397.0	4.0	0	0.0	[50.0.60.0.50.0.5397.0.8.0]	0.3061853327404323
85031	3800.0	2910.0	19160.0	301310.0	390.0	0	0.0	[3608.0.2913.0.19160.0.301310.0.398.0]	0.2826031322521401
85001	3800.0	2910.0	19160.0	301310.0	390.0	٥	0.0	[3608.0.2913.0.19160.0.301310.0.398.0]	0.2826031322521431
95231	640.0	670.0	1790.0	62440.0	95.0	0	0.0	3540-0.870-0.1790-0.82440-0.95-0	0.30377863716850784
95631	1003.0	1420.0	1360.0	172263.0	1065.0	95631	1.0	[1060.0,1420.0,1300.0,172283.0,1065.0]	0.3052569158562556
6232	770.0	720.0	880.0	129932.0	457.0	0	0.0	1772 0 730 0 880 0 129932 0 457 51	0.30565296005413617

Clustering Houses

This workflow reads in a dataset. It then performs KMeans Clustering on the Housing Dataset.

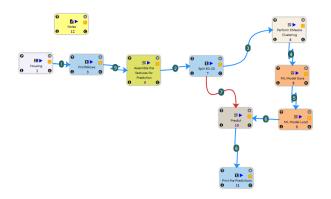
Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Prints the results.
- Assembles the features for predictions.
- Splits it.
- Perform KMeans Clustering.
- ML Model save.
- ML Model Load.
- Prediction.
- Print the prediction results.

Reading from Dataset

It reads sample Dataset file.



DatasetStructured z o Details NodeDatasetStructured			2 1
OUTPUT STORAGE LEVEL : 0	DEFAULT	Ŧ	
DATASET : 😧	Housing	*	
			OK CANCEL

Processor Output

Autoring Not	le fire.nodes.do	ataset.NodeDa	tasetStructured	: 1 Nov 15, 2018	11:24:38 PM							
Row Values												
id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType
	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no
3	49500.0	3060	3	1	1	yes	no	no	no	no	0	no
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no
5	61000.0	6360	2	1	1	yes	no	no	no	no	0	no
6	66000.0	4160	3	1	1	yes	yes	yes	no	yes	0	no
7	66000.0	3880	3	2	2	yes	no	yes	no	no	2	no
8	69000.0	4160	3	1	3	yes	no	no	no	no	0	no

Prints the results

It prints the sample dataset file results.

Processor Configuration

Processor Output

Assemble the features for predictions

It assembles the features for predictions using VectorAssembler Node.

HEMA :													
OLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
OLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string
OLUMN FORMAT													
JTPUT STORAGE LEVEL	DEF						•						

Input Scher	ma											
Row Values												
d	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	preforea
ntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType
	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no
2												

OLUMN NAME	id	price	lotsize	bedroom	is bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
DUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string
OLUMN FORMAT													
TPUT STORAGE LEV	EL:0				DEFAULT id : integer lotsize : integer bedrooms : integer bothrms : integer	q				•			
					stories : integer garagept : integer price : double					ų.			

Processor Output

Executing Not	de fire.nodes.r	nl.NodeVector	Assembler : 5	Nov 15, 2018 11	:28:10 PM								
Input Scher	na												
Row Values													
id	price	lotsize	bedrooms	bothrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType	org.apach
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no	[5850.0,3.0
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no	[4000.0,2.0
3	49500.0	3060	3	1	1	yes	no	no	no	no	0	no	[3060.0,3.0
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no	[6650.0,3.0
5	61000.0	6360	2	1	1	yes	no	no	no	no	0	no	[6360.0,2.0
9													

Split it

It splits features of prediction using Split Node.

Processor Configuration

COLUMN FORMAT COLUMN	odeSplit														
COLUMN FORMAT COLUMN	HEMA :														
	COLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features
	COLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	vectorudt
JTPUT STORAGE LEVEL : 0 DECAMET	COLUMN FORMAT														
	COLUMN FORMAT		double	integer	integer	DEFAULT	integer	string	string	string	string	string	integer	string	vecto
	RACTION 1 * :					.8									

Processor Output

Split 80-20	
Executing Node fire.nodes.ml.NodeSplit : 7 Nov 15, 2018 11:33:45 PM	
Input Schema	
	ok

Perform KMeans Clustering

It performs KMeans Clustering on the Housing Dataset using KMeans Node.

Processor Configuration

Processor Output

HEMA :															
DLUMN NAME	id	price	otsize	bedrooms	bathrm	is stories	driveway	recroom	fullbase	gashw	airco g	ragepl	prefarea	features	3
DLUMN TYPE	integer	double	nteger	integer	integer	integer	string	string	string	string	string in	eger	string	vectoru	dt
DLUMN FORMAT															
TPUT STORAGE	LEVEL : O				DEFAULT						۲				
TURES COLUMN	*:0				features	: vectorudt									
0					3										
X ITERATIONS : 0	Ð				20										
EDICTION COLU	MN : O														
iD :															
LERENCE : O					0.0001										
TMODE : O					k-mean										
TSTEPS : O	ans Clusterir	ng			5									OK	CANCEL
rform KMe	de fire.nodes.i	-	ans : 6 No	v 15, 2018 1	5									OK	CANCEL
rform KMe	de fire.nodes. ma	-	ans : 6 No	v 15, 2018 1	5									OX	CANCEL
rform KMe Executing No	de fire.nodes. ma	-	tans : 6 No bedroen		5		driveway	recroom	fullbose	goshw	airco	garagep	4 pref		CANCEL
rform KMee Executing No Input Scher	de fire.nodes.r ma	nl. NodeKMe	bedroom	ns bati	5				fullbase StringType					area f	eatures
rform KMee Executing Not Input Scher Row Values	de fire.nodes.r ma	nl. NodeKMe	bedroom	ns bati	5 1:36:36 PM	stories								area fi ngType c	eatures org.opache
rform KMee Executing Nor Input Scher Row Values Id IntegerType	de fire.nodes.n ma price DoubleType	IntegerTyp	e Integer	ns batt Type Inte	1:36:36 PM	stories IntegerType	StringType	StringType	StringType	StringTyp	e StringTy	be Integer	Type Strin	area fi ngType c	
rform KMee Executing Nor Input Scher Row Values Id IntegerType 6	price DoubleType 6600.0	IntegerTyp 4160	bedroon Integer 3	ns bott Type Inte 1	1:36:36 PM	stories IntegerType	StringType yes	StringType yes	StringType yes	StringTyp	e StringTy yes	oe Integer	Type Strin	area \$ ngType c [eatures xrg.apache

ML Model save

It will save ML Model with given path using ModelSave Node.

Processor Configuration

CHEMA :														
COLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features
COLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	vectorudt
COLUMN FORMAT														
UTPUT STORAGE LEV ATH TO SAVE THE ML					DEFAULT	oth					•			
VERWRITE OUTPUT :					TRUE O FALS									

Processor Output

ML Model Load

It will Load ML Model with given path using ModelSave Node.

Processor Configuration

	de fire.nodes.r	nl.NodeMode	ISave : 8 Nov	15, 2018 11:37	59 PM									
Input Scher	ma													
Row Values														
id	price	lotsize	bedrooms	bothrms	stories	d	Iriveway	recroom	fullbase	gashw	airco	garagepl	preforea	features
IntegerType	DoubleType	IntegerType	IntegerType	IntegerTyp	e Intege	rType S	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType	org.apache
1	42000.0	5850	3	1	2	у	es	no	yes	no	no	1	no	[5850.0,3.0
1	42000.0	5850	3	1	2	у	es	no	yes	no	no	1	no	(5850.0,3.0
3	49500.0	3060	3	1	1	У	es	no	no	no	no	0	no	(3060.0,3.0
4	60500.0	6650	3	1	2	У	es	yes	no	no	no	0	no	(6650.0,3.0
5	61000.0	6360	2	1	1	У	res	no	no	no	no	0	no	[6360.0,2.0,
5	61000.0	6360	2	1	1		es	no 	no 	no	no	0	no 	16260.0.0
5 C L Model Los deModelLoad KEMA : DLUMN NAME	61000 0	6360	2	,	,					••	**		**	16360.0.0.0
E Model Los deModelLoad	ad <table-cell></table-cell>	price I	otsize bec	irooms bo	, thrms	stories	drivewa	y recroo	n fullbas	e gashw	airco	garagepl	prefarea	
E Model Los deModelLoad IEMA : DLUMN NAME	ad 2 id integer	price I	2	irooms bo	, thrms					••	airco	garagepl	prefarea	(C) CO O O O O
E Model Load deModelLoad rema : blumn name blumn type	ad 2 id integer	price I	otsize bec	irooms bo	, thrms	stories	drivewa	y recroo	n fullbas	e gashw	airco	garagepl	prefarea	features
E Model Load deModelLoad rema : blumn name blumn type	ad Z id integer	price I	otsize bec	, irooms bc ger int	, thrms	stories	drivewa	y recroo	n fullbas	e gashw	airco	garagepl	prefarea	features
E Model Load deModelLoad IEMA : DLUMN NAME DLUMN TYPE DLUMN TYPE TPUT STORAGE L	ad Z id integer	price I double i	otsize bec	rooms bc ger int	thrms eger	stories	drivewa	y recroo	n fullbas	e gashw	airco string	garagepl	prefarea	features

Processor Output

Executing Not	de fire.nodes.r	nl.NodeModel	Load : 9 Nov 1	5, 2018 11:39:0	I PM								
Input Scher													
Input Scher	na												
Row Values													
id	price	lotsize	bedrooms	bothrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	preforea	features
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType	org.apach
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no	[5850.0,3.
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no	[4000.0,2.
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no	[4000.0,2.0
3	49500.0	3060	3	1	1	yes	no	no	no	no	0	no	[3060.0,3.0
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no	[6650.0,3.0
4													

Prediction

It predicts features updated using Predict Node.

Processor Configuration

Processor Output

Print the prediction results

It Print the prediction results.

Processor Configuration

lotsize integer	bedrooms integer	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features
integer	integer										
	unegei	integer	integer	string	string	string	string	string	integer	string	vectorudt
lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features
integer	integer	integer	integer	string	string	string	string	string	integer	string	vectorudt
							, , , , , , , , , , , , , , , , , , , ,	, , ,	, , ,		

Row Values	5												
id	price	lotsize	bedrooms	bothrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	preforea	features
ntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType	org.apache.sp
5	61000.0	6360	2	1	1	yes	no	no	no	no	0	no	[6360.0,2.0,1.0
											_		

Print the Pred NodePrintFirstNR		7 0													i
SCHEMA :															
COLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	features	prediction
COLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	vectorudt	double
COLUMN FORMAT															
OUTPUT STORAGE LE	/EL : 😡				DEFAU	ιτ						¥			
TITLE :					Row Vo	lues									
NUM ROWS TO PRINT	•				10										
															OK CANCEL

Processor Output

Print the Predictions	2
Executing Node Fire.nodes.util.NodePrintFirstNRows : 11 Nov 15, 2016 11:40:51 PM	
Inpul Schema	
	ĸ

TFIDF

This workflow reads in a dataset. It then Tokenizes and then performs TF/IDF on text content.

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Tokenizes message column.
- Performs TF.
- Performs IDF.
- Prints the results.





Reading from Dataset

It reads sample Dataset file.

Processor Configuration

DatasetStructured @ O Details NodeDatasetStructured			
DUTPUT STORAGE LEVEL : O	DEFAULT	٠	
DATASET : O	Spam	¥	

acuting Node fire.nodes.datase	et.NodeDatasetStructured : 1 Nov 15, 2018 4:57:59 AM	
Row Values		
label	message	id
DoubleType	StringType	DoubleType
1.0	this is a spam	2.0
D.O	i am going to work	1.0
0.0	this is not a spam	3.0
1.0	this is a spam	2.0
0.0	i am going to work	1.0
D.0	this is not a spam	3.0
.0	this is a spam	2.0
0.0	i am going to work	1.0

Processor Output

Tokenizes message column

It Tokenizes message column generated by sample dataset file using Tokenizer Node.

Processor Configuration

Tokenizer 7 0 NodeTokenizer					
SCHEMA :					
COLUMN NAME	label	message		id	
COLUMN TYPE	double	string		double	
COLUMN FORMAT					
DUTPUT STORAGE LEVEL : \varTheta	DEFAULT		•		
NPUT COLUMN : O	message : string		٣		
DUTPUT COLUMN : 🛛	words				

Processor Output

xecuting Node fire.r	odes.ml.NodeTokenizer : 2 Nov 15, 2018 -	4:58:45 AM		
Input Schema				
Row Values				
label	message	id	words	
DoubleType	StringType	DoubleType	ArrayType(StringType,true)	
1.0	this is a spam	2.0	WrappedArray(this, is, a, spam)	
0.0	i am going to work	1.0	WrappedArray(i, am, going, to, work)	
0.0	this is not a spam	3.0	WrappedArray(this, is, not, a, spam)	
1.0	this is a spam	2.0	WrappedArray(this, is, a, spam)	
1.0	i am going to work	1.0	WrappedArray(i, am, going, to, work)	
0.0	r an going to nork			

Perform TF

It performs TF on text column using HashingTF Node.

CHEMA :					
COLUMN NAME	label		message	id	words
COLUMN TYPE	doub	sle	string	double	array
COLUMN FORMAT					
DUTPUT STORAGE LEVEL : O		DEFAULT		•	
NPUT COLUMN : O		words : array	*		
DUTPUT COLUMN : O		rawFeatures			

Processor Output

ecuting Node	fire.nodes.ml.NodeHashin	gTF : 4 Nov 15, 2018	4:59:38 AM	
Input Schem	2			
Row Values				
Row Values				
label	message	id	words	rawFeatures
DoubleType	StringType	DoubleType	ArrayType(StringType,true)	org.apache.spark.ml.linalg.VectorUDT@3bfc3ba7
1.0	this is a spam	2.0	WrappedArray(this, is, a, spam)	(1000,[170,281,373,473],[1.0,1.0,1.0,1.0])
0.0	i am going to work	1.0	WrappedArray(i, am, going, to, work)	(1000,[173,329,388,493,527],[1.0,1.0,1.0,1.0,1.0])
0.0	this is not a spam	3.0	WrappedArray(this, is, not, a, spam)	(1000,[18,170,281,373,473],[1.0,1.0,1.0,1.0,1.0])
1.0	this is a spam	2.0	WrappedArray(this, is, a, spam)	(1000,[170,281,373,473],[1.0,1.0,1.0,1.0])
	i am going to work	1.0	WrappedArray(i, am, going, to, work)	(1000,[173,329,388,493,527],[1.0,1.0,1.0,1.0,1.0])
0.0				

Perform IDF

It performs IDF on text column using IDF Node.

Processor Configuration

CHEMA :						
COLUMN NAME	label		message	id	words	rawFeatures
COLUMN TYPE	double		string	double	array	vectorudt
COLUMN FORMAT						
UTPUT STORAGE LEVEL : O IPUT COLUMN : O UTPUT COLUMN : O			AULT Features : vectorudt		v v	
IINDOCFREQ : 0		0				

Processor Output

Prints the results

It will print the result after performing TF/IDF on text content.

Input Scher	ma									
Row Values										
label	message	id	words		rawFeatures			features		
DoubleType	StringType	DoubleType	ArrayType(String	Type,true)	org.apache.s	spark.ml.linalg.Vector	UDT@3bfc3ba7	org.apache.spark.	ml.linalg.VectorU	DT@3bfc3ba7
1.0	this is a spam	2.0	WrappedArray(spam)	his, is, a,	(1000,[170,28	1,373,473],[1.0,1.0,1.0,1	.0])	(1000,[170,281,373, [0.318453731118534		5346,0.3184537311185346,0
0.0	i am going to work	1.0	WrappedArray(going, to, work)	, am,	(1000,[173,32	9,388,493,527],[1.0,1.0),1.0,1.0,1.0])	(1000,[173,329,388 [1.011600911678479		4799,1.0116009116784799,1.0
0.0	this is not	3.0	WrappedArray(his is not	(1000 /18 170	281,373,473],[1.0,1.0,1.	010100	(1000,[18,170,281,3]	73.4731.	
	a spam		a, spam)		(1000,[10,170,	,201,373,473],[1:0,1:0,1:	0,00,00])			5346,0.3184537311185346,0
1.0	a spam this is a	2.0				1.373 4731 (1.0.1 0.1 0.1			9,0.318453731118	5346,0.3184537311185346,0
1.0 intNRows I dePrintFirstN IEMA :	this is a		a, spam)	his is a				[1.011600911678479	9,0.318453731118	5346,0.3184537311185346,0
IntNRows dePrintFirstN IEMA : DLUMN NAME	this is a	la	a, spam) WraonedArraw bel	his. is. a.		id	.on	[1.0160091678479 0000 1170 281 373	9,0.318453731118	features
intNRows dePrintFirstN IEMA :	this is a	la	a, spam) WrappedArrow	his is a		1 373 4731 // 01 01 01 0		(1.011600911678479 (1000.1170.281.373	9,0.318453731118	
INTNROWS dePrintFirstN IEMA : DLUMN NAME DLUMN TYPE	this is a	la	a, spam) WraonedArraw bel	his. is. a.	0000 070 28	id	.on	[1.0160091678479 0000 1170 281 373	9,0.318453731118	features
INTN ROWS I dePrintFirstN IEMA : DLUMN NAME DLUMN TYPE DLUMN FORMAT	this is a	la	a, spam) WraonedArraw bel	message string	0000 070 28	id	.on	[1.0160091678479 0000 1170 281 373	9,0.318453731118	features

Processor Output

Executing Nor	de fire.nodes	.util.NodePrint	FirstNRows : 3 Nov 15, 2018 5	:01:19 AM	
Input Scher	na				
Row Values					
label	message	id	words	rowFeatures	features
DoubleType	StringType	DoubleType	ArrayType(StringType,true)	org.apache.spark.ml.linalg.VectorUDT@3bfc3ba7	org.apache.spark.ml.linalg.VectorUDT@3bfc3ba7
1.0	this is a spam	2.0	WrappedArray(this, is, a, spam)	(1000,[170,281,373,473],[1.0,1.0,1.0,1.0,1.0])	(1000,[170,281,373,473], [0.3184537311185346,0.3184537311185346,0.3184537311185346,0.31
0.0	i am going to work	1.0	WrappedArray(i, am, going, to, work)	(1000,[173,329,388,493,527],[1.0,1.0,1.0,1.0,1.0])	(1000,[173,329,388,493,527], [1.016009116784799,1.0116009116784799,1.0116009116784799,1.0116
0.0	this is not	3.0	WrappedArray(this, is, not,	(1000,[18,170,281,373,473],[1.0,1.0,1.0,1.0,1.0])	(1000,[18,170,281,373,473], [1.0116009116784799.0.3184537311185346.0.3184537311185346.0.31

Earthquake Prediction

Objective

As the motivation behind earthquake prediction is to empower crisis measures to decrease demise and devastation, inability to give notice of a significant earthquake that happens, or possibly a satisfactory assessment of the hazard, can bring about legitimate risk, or even political cleansing.

Dataset

Dataset contains 2 columns as below:

- Acoustic_data Acoustic wave reading
- Time_to_failure Time remaining before the next earthquake

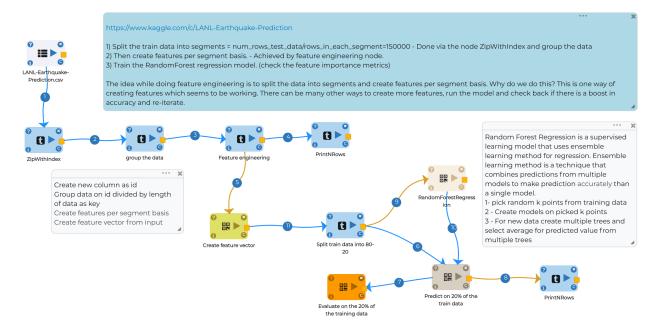
acoustic_data	time_to_failure
IntegerType	DoubleType
12	1.469
6	1.469
8	1.469
5	1.469
8	1.469
8	1.469
9	1.469
7	1.469
-5	1.469
3	1.469
5	1.469
2	1.469
2	1.469
3	1.469

Random Forest Regression Workflow for Earthquake Prediction

Random Forest Regression model belongs to family of bagging regression. It is a supervised learning model that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple models to make prediction more accurately than a single model.

Features of Random Forest -

- · Aggregates many decision trees
- · Prevents overfitting



Prepare data for modeling

Follow workflow arrow

- ZipWithIndex- Creates new feature column from dataframe index as ID
- Group data- Creates new feature column as key obtained by ID divided by length of data
- Feature Engineering- Groups by data on key to create all statistical measures (min, max, mean, quartiles etc) as new feature
- Feature Vector Merge multiple columns to form vector

Data modeling

• Before we create Random Forest Regression model, split data (80:20) into train and test for performance evaluation.

Input Schema id acoustic_data time_to_failure id IntegerType DoubleType LongType

Row Values

Row Values			
acoustic_data	time_to_failure	id	key
IntegerType	DoubleType	LongType	DoubleType
12	1.469	0	0.0
6	1.469	1	0.0
8	1.469	2	0.0
5	1.469	3	0.0
8	1.469	4	0.0
8	1.469	5	0.0
9	1.469	6	0.0
7	1.469	7	0.0
-5	1.469	8	0.0
3	1.469	9	0.0

Input Schema

input benefita			
acoustic_data	time_to_failure	id	key
IntegerType	DoubleType	LongType	DoubleType

Row Values

Row Values

segment	max_a	min_a	avg_a	std_a	var_a	p_50	p_25	p_75	time_to_failure_label
DoubleType	IntegerType	IntegerType	DoubleType						
0.0	14	-5	5.28	3.344	11.185	5.0	3.0	7.75	1.469
1.0	13	-4	5.88	3.612	13.047	6.0	4.0	8.75	1.469

INPUT COLUMNS : Ø	Available			Selected	1	•
	time_to_failure_	Jabel : double	> <	segment : double max_a : integer avg_a: double std_a: double p_50: double p_50: double p_75: double p_75: double		
OUTPUT COLUMN *: 🛛		feature_column				

Random Forest Regression

• Sets feature vector corresponding to label(time_to_failure_label).	•	Sets feature vector	corresponding to label	(time_to_failure_label).
--	---	---------------------	------------------------	--------------------------

- Sets number of features for each split node of tree.
- For regression the measure of impurity is variant.
- In random forest, the impurity decrease from each feature can be averaged across trees to determine the final importance of the variable.
- The maxBins signifies the maximum number of bins used for splitting the features, where the suggested value is 100 to get better results.
- The maxDepth is the maximum depth of the tree (for example, depth 0 means one leaf node, depth 1 means one internal node plus two leaf nodes).
- Information gain is calculated by comparing the entropy of the dataset before and after a transformation.

FEATURES COLUMN : 0	feature_column : vectorudt	~
LABEL COLUMN : 😧	time_to_failure_label : double	~
PREDICTION COLUMN : 0		
FEATURE SUBSET STRATEGY : 🚱	auto	~
IMPURITY : 😡	variance	~
MAX BINS : 🛛	32	
MAX DEPTH : 🚱	5	
MIN INFORMATION GAIN : 😡	0.0	
MIN INSTANCES PER NODE : 🚱	1	
NUM TREES : 🚱	20	
SUBSAMPLING RATE : 0	1.0	
SEED : 🛛		
CACHE NODE IDS : 😧	falso	

CHECKPOINT INTERVAL : 😧

MAX MEMORY : 😧

10	false	ie		
	10			
256	256			

Model evaluation

• Multiple ways to evaluate regression model such as R square, Root mean square error(rmse), mean square error(mse)

OUTPUT STORAGE LEVEL : 😡	DEFAULT	~
LABEL COLUMN :	time_to_failure_label : double	~
PREDICTION COLUMN : 🛛	prediction : double	~
METRIC NAME : 0	rmse	~

12.1.4 Analytics

Analyze Flights Delays

This workflow reads in a dataset. It then analyzes flights delay with sample datasets and prints the results.

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Prints the sample datasets results.
- Column to be cast for new datatype double.
- Column to be cast for new datatype string.
- Updates the column name of datatype string.
- Prints the result of data updating after stringindexer Node.
- Executes the SQL queries with the given conditions.
- Prints the results.

Reading from Dataset

It reads Dataset files.

Processor Configuration

lodeDatasetStructured			
UTPUT STORAGE LEVEL : Ø	DEFAULT	•	
ATASET : O	Flights Delay	×	

Processor Output

Print the sample datasets results

It prints the sample datasets results.

CHEMA :												
COLUMN NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_D
COLUMN TYPE	integer	integer	string	string	integer	integer	string	integer	string	integer	integer	integer
COLUMN FORMAT												
DUTPUT STO TTLE :	ORAGE LEVEL : \varTheta			DEFAU Row Vo								
IUM ROWS	TO PRINT : 0			10								

CHEMA :												
COLUMN NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_D
COLUMN	integer	integer	string	string	integer	integer	string	integer	string	integer	integer	integer
COLUMN FORMAT												
UTPUT ST	ORAGE LEVEL : 😧			DEFAU	LT				·			
ITLE :				Row Vo	lues							
UM ROWS	TO PRINT : 0			10								

Processor Output

Input Schema												
Row Values												
DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CR5_DEP_TIME	DEP_TIME	DEP_DELAY_NEW	CRS,
IntegerType	IntegerType	StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	IntegerType	Inte
1	3	AA	N338AA	1	12478	JFK	12892	LAX	900	914	14	1225
1	3	AA	N339AA	2	12892	LAX	12478	JFK	930	1132	122	1800
1	3	AA	N335AA	3	12478	JFK	12892	LAX	1200	1157	0	1510
1	3	AA	N367AA	5	11296	DFW	12173	HNL	1305	1307	2	1745
1	3	AA	N364AA	6	13830	OGG	11298	DFW	1755	1753	0	500
1	3	AA	N364AA	7	11298	DFW	13830	066	1200	1205	5	1625
	3	AA	N372AA	8	12173	HNL	11298	DFW	1800	1839	39	520
1	3	AA	N3KBAA	9	12892	LAX	13303	MIA	2155	2211	16	535
	3	AA	N328AA	10	12892	LAX	12478	JFK	2115	2122	7	525
1	3	AA	NSDHAA	14	13830	OGG	12892	LAX	2313	2306	0	620

Column to be cast for new datatype double

It casts for new datatype double using castcolumn type Node.

Processor Configuration

COLUMN												
NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_
COLUMN TYPE	integer	integer	string	string	integer	integer	string	integer	string	integer	integer	intege
COLUMN												
NUTPUT ST	ORAGE LEVEL : O			DEFAUL	T							
				FL_NUM ORIGIN ORIGIN DEST_A DEST_S CRS_DI	RPORT_ID	: integer		_				
				DEP_DI CRS_AF ARR_TE ARR_DE CRS_EL	CLAY_NEW : IN TIME : IN ME : Integer CAY_NEW : APSED_TIM CE : Integer	leger Integer						
IEW DATA 1	TYPE : O			DEP_DI CRS_AF ARR_TE ARR_DE CRS_EL	W TIME IN ME : integer (LAY_NEW) APSED_TIM (2E : integer	leger Integer						

Processor Output

Input Schemo												
Row Values												
DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	DEP_TIME	DEP_DELAY_NEW	ARR_TIME	ARR_
IntegerType	IntegerType	StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	IntegerType	Integ
1	3	AA	N338AA	1	12478	JFK	12892	LAX	914	14	1238	13
1	3	AA	N339AA	2	12892	LAX	12478	JFK	1132	122	1951	m
1	3	AA	N335AA	3	12478	JFK	12892	LAX	1157	0	1523	13
1	3	AA	N367AA	5	11298	DFW	12173	HNL	1307	2	1746	1
1	3	AA	N364AA	6	13830	OGG	11298	DFW	1753	0	452	0
1	3	AA	N364AA	7	11298	DFW	13830	000	1205	5	1630	5
	3	AA	N372AA	8	12173	HNL	11298	DFW	1839	39	620	60
1	3	AA	N3KBAA	9	12892	LAX	13303	MIA	2211	16	552	17
1	3	AA	N328AA	10	12892	LAX	12478	JFK	2122	7	523	0
1	3	AA	NSDHAA	14	13830	OGG	12892	LAX	2306	0	617	0

Column to be cast for new datatype string

It casts for new datatype string using castcolumn type Node.

Processor Configuration

Processor Output

Updates the column name of datatype string

It updates the column name of datatype string using stringindexer type Node.

SCHEMA :												
COLUMN NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_0
COLUMN TYPE	integer	integer	string	string	integer	integer	string	integer	string	double	integer	intege
COLUMN FORMAT												
UTPUT STO	DRAGE LEVEL : 0			DEFAUL	т							
				CARRIE TAIL_NU FL_NUM ORIGIN ORIGIN DEST_/ DEST_/ DEP_TI DEP_TI CHS_AI ARR_TI	NRPORT_ID	ID : integer : integer louble : Integer ioubla						
NEW DATA T	YPE: O			STRING				•				
	ISTING COLS : O											

Executing No	ode fire.node	s.etl.NodeCas	tColumnType : 4 Nov	/ 14, 2018 1:12	39 AM							
Input Sche	ima											
Row Value	15											
CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	DEP_TIME	DEP_DELAY_NEW	ARR_TIME	ARR_DELAY_NEW	DISTANCE	CRS_DE
StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	Double
AA	N338AA	1	12478	JFK	12892	LAX	914	14	1238	13	2475	900.0
AA	N339AA	2	12892	LAX	12478	JFK	1132	122	1951	m	2475	930.0
AA	N335AA	3	12478	JFK	12892	LAX	1157	0	1523	13	2475	1200.0
AA	N367AA	5	11296	DFW	12173	HNL	1307	2	1746	1	3784	1305.0
AA	N364AA	6	13830	OGG	11298	DFW	1753	0	452	0	3711	1755.0
AA	N364AA	7	11298	DFW	13830	066	1205	5	1630	5	3711	1200.0
AA	N372AA	8	12173	HNU	11295	DFW	1839	39	620	60	3784	1600.0
AA	N3KBAA	9	12892	LAX	13303	MIA	2211	16	552	17	2342	2155.0
AA	N328AA	10	12892	LAX	12478	JFK	2122	7	523	0	2475	2115.0
AA	NSDHAA	14	13830	OGG	12892	LAX	2306	0	617	0	2486	2313.0

ок

	ngIndexer											
CHEMA :												
COLUMN NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_
COLUMN	string	string	string	string	integer	integer	string	integer	string	double	integer	intege
COLUMN FORMAT												
OUTPUT ST	ORAGE LEVEL : 0			DEFAUL								
				DEFAUL								
IANDLE INV	VALID : 😝			skip				,				
INPUT COL	LUMNS O	OUTPUT COLUMNS	Θ									
DAY_OF_I	MONTH	DAY_OF_MONTH_IN	DEX									
		DAY_OF_MONTH_IN DAY_OF_WEEK_IND										
DAY_OF_												
DAY_OF_	WEEK	DAY_OF_WEEK_IND										
DAY_OF_ CARRIER	WEEK	DAY_OF_WEEK_IND										
DAY_OF_I CARRIER TAIL_NUM FL_NUM	WEEK	DAY_OF_WEEK_IND	EX									
DAY_OF_I CARRIER TAIL_NUM FL_NUM ORIGIN_A	A A	DAY_OF_WEEK_IND CARRIER_INDEX	EX									
DAY_OF_ CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN	WEEK	DAY_OF_WEEK_IND CARRIER_INDEX	EX I_INDEX									
DAY_OF_ CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR	WEEK	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_I DAY_OF_I CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CRS_DEP,	A A A A A A A A A A A A A A A A A A A	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_I CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CR5_DEP,	A A A A A A A A A A A A A A A A A A A	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_' CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CR5_DEP, DEP_TIME	A A AND AND AND AND AND AND AND AND AND	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_' CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CRS_DEP, DEP_TIME DEP_DELA	WEEK A A A A A POORT_ID	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_' CARRIER TAIL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CR5_DEP_ DEP_TIME DEP_DELA CR5_ARR,	A A A A A A A A A A A A A A A A A A A	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_' CARRIER TAL_NUM FL_NUM ORIGIN_A ORIGIN DEST_AIR DEST CRS_DEP_ TIME DEP_DELA CRS_ARR, ARR_TIME	A A A A A A A A A A A A A A A A A A A	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									
DAY_OF_' CARRIER TAI_NUM FL_NUM ORIGIN_A ORIGIN_A DEST_AIR DEST_CRS_DEP_ DEP_TIME DEP_DEL/ CRS_ARR_ ARR_TIME ARR_DEL/	A A A A A A A A A A A A A A A A A A A	DAY_OF_WEEK_IND CARRIER_INDEX ORIGIN_AIRPORT_IE	EX I_INDEX									

Processor Output

tringIndexer	
Executing Node fire.nodes.ml.NodeStringIndexer : 5 Nov 14, 2018 1:25:33 AM	
Input Schema	
Stringindexer encodes a string column of labels to a column of label indices.	
day_of_month_category	day_of_month_conegoryIndex
1	0.0
StringIndexer encodes a string column of labels to a column of label indices.	
day_of_week_category	day_of_week_categoryindex
3	0.0
StringIndexer encodes a string column of labels to a column of label indices. conter_cetagory	carrier_cotegoryIndex
AA	0.0
StringIndexer encodes a string column of labels to a column of label indices.	
origin_airport_id_cotegory	origin_airport_id_categoryindex
12892	0.0
12892 12478	0.0
12892 12478 13830	0.0 1.9 2.0
12892 12478	0.0

StringIndexer encodes a string column of labels to a column of label indices.

Prints the Results

It prints the result of data updating after stringindexer Node.

BCHEMA :												
COLUMN NAME	DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_D
COLUMN TYPE	string	string	string	string	integer	integer	string	integer	string	double	integer	integer
COLUMN FORMAT												
DUTPUT ST	ORAGE LEVEL : 🖗			DEFAU								
NUM ROWS	TO PRINT : 0			Row Vo	lues							

Processor Output

												_
Input Sch	ema											
Row Value	16											
CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	DEP_TIME	DEP_DELAY_NEW	ARR_TIME	ARR_DELAY_NEW	DISTANCE	CRS_D
StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	Doub
AA	N338AA	1	12478	JFK	12892	LAX	914	14	1238	13	2475	900.0
AA	N339AA	2	12892	LAX	12478	JFK	1132	122	1951	111	2475	930.0
AA	N335AA	3	12478	JFK	12892	LAX	1157	0	1523	13	2475	1200.
AA	N367AA	5	11298	DFW	12173	HNL	1307	2	1746	1	3784	1305.0
AA	N364AA	6	13830	OGG	11295	DFW	1753	0	452	0	3711	1755.0
АА	N364AA	7	11296	DFW	13830	OGG	1205	5	1630	5	3711	1200.
AA	N372AA	8	12173	HNL	11295	DFW	1839	39	620	60	3784	1800.0
AA	N3KBAA	9	12892	LAX	13303	MIA	2211	16	552	17	2342	2155.0
AA	N328AA	10	12892	LAX	12478	JFK	2122	7	523	0	2475	2115.0
AA	NSDHAA	14	13830	OGG	12892	LAX	2306	0	617	0	2486	2313.0

Executes the SQL queries

It executes the SQL queries with the given conditions.

Processor Configuration

Processor Output

Prints the Results

It prints the results after satisfied condition by sql queries.

Processor Configuration

												7
	Y_OF_MONT	H DAY_OF_	WEEK CARRIER	TAIL_NUM	FL_NUM ORIGIN	_AIRPORT_I	D ORIGIN	DEST_AIRPORT_I	D DEST	CRS_DEP_TIME	DEP_TIME	DEP_DE
DLUMN strin	ng	string	string	string	integer integer		string	integer	string	double	integer	integer
DLUMN												
TPUT STORAG	E LEVEL : 😡			DEFAULT					•			
AP TABLE : O				fire_temp	_table							
i select fi	ire_temp_table	*,case she	n fire_temp_table.Of	P_DELAY_NEW >	40 then 1.0 else	0.0 END ms lø	bel from fire_	temp_table				
IEMA COLUM		SH SCHEMA	0									
PUT COLUMN	NAMES O		c	NUTPUT COLUM	N TYPES 😡			OUTPUT COLUMN FO	RMATS 😜			
		is.etl.NodeSQI	L : 7 Nov 14, 2018 1:3	5:10 AM							OK	
Input Sch												
Row Valu												
CARRIER StringType	TAIL_NUM StringType	FL_NUM	ORIGIN_AIRPORT_ID	StringType	DEST_AIRPORT_ID		DEP_TIME IntegerType	DEP_DELAY_NEW	ARR_TIME	ARR_DELAY_NEW	IntegerType	CRS_DI
AA	N338AA	1	12478	JFK	12892	LAX	914	14	1235	13	2475	900.0
AA	N339AA	2	12892	LAX	12478	JFK	1132	122	1951	m	2475	930.0
AA	N335AA	3	12478	JFK	12892	LAX	1157	0	1523	13	2475	1200.0
AA	N367AA N364AA	5	11298	DFW OGG	12173	HNL DFW	1307	2	1746 452	1	3784	1305.0
AA AA	N364AA	7	11298	DFW	13830	000	1205	5	1630	5	3711	
AA				DFW HNL				5 39	1630 620	5 60	3711 3784	1200.0
аа ла аа ла	N364AA N372AA N3KBAA	7 8 9	11298 12173 12892	HNI. LAX	13830 11298 13303	OGG DFW MIA	1205 1839 2211	39 16	620 552	60 17	3784 2342	1200.0 1800.0 2155.0
аа АА АА	N364AA N372AA N3KBAA N328AA	7 8 9 10	11298 12173	HNL	13830 11298 13303 12478	OCG DFW	1205 1839	39 16 7	620	60 17 0	3784	1200.0 1800.0 2155.0 2115.0
AA AA AA AA AA	N364AA N372AA N3KBAA	7 8 9	11298 12173 12892 12892	HNL LAX LAX	13830 11298 13303	OGG DFW MIA JFK	1205 1839 2211 2122	39 16	620 552 523	60 17	3784 2342 2475	1200.0 1800.0 2155.0 2115.0
AA AA AA AA AA AA AA MA AA EXTRACT	N364AA N377AA N3KBAA N328AA N5DHAA N5DHAA	7 8 9 10	11298 12173 12892 12892	HNL LAX LAX	13530 11298 13303 12478 12892	OGG DFW MIA JFK	1205 1839 2211 2122	39 16 7	620 552 523	60 17 0	3784 2342 2475	1200.0 1800.0 2155.0 2115.0
AA AA AA AA AA AA AA AA AA AA AA AA AA	NBS64AA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIZCHAA ELEVEL® ELEVEL®	7 8 9 10 14	11298 12173 12892 12892	HRL LAX DGG DEFAU Row Vo 10	13530 11294 13303 12652 12652 LT	OGG DFW MIA JFK	1205 1839 2211 2122	39 16 7	620 552 523 617	60 17 0	3784 2342 2475	1200.0 1800.0 2155.0 2115.0 2313.0
AA AA AA AA AA AA AA AA AA AA AA AA AA	NBS64AA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIJZZAA NIZCHAA ELEVEL® ELEVEL®	7 8 9 10 14	1292 1275 12852 12830 13830	HBL LAX LAX CGG BEFAU Bow Vo 14, 2016 1:37	13530 11294 13303 12652 12652 LT	OCG DPW MBA JPK LAX	1205 1839 2211 2122	39 16 7	620 552 523 617	60 17 0	3784 2144 2475 2486 2486 2486 2486 2486 2486 2486 2486	1200.0 1800.0 2155.0 2155.0 2313.0 2313.0 0 8 6 8 6
AA AA AA AA AA AA AA AA AA AA		2 3 9 10 14 14 14 14 14 14 14 14 14 14	1292 1073 12852 12852 13830 139300 13900 10000 10000 10000 10000000000	HBL LAX LAX OGG DEFAU Row Vo 10 10 10 10 10 10 10 10 10 10 10 10 10	13530 1294 13303 13262 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 128555 12855 12855 12855 12855 12855 12855 12855 12855 12	OCG DRV MBA JRC LAX LAX CAX CAX CAX CAX CAX CAX CAX CAX CAX C	1206 10.15 2211 2222 2206 2206 2006 2006 2006 2007	39 10 7 0 0 0 0 0 0 0 0 0 0 0 0 0	620 552 523 617	60 17 0 0 0 - - - - - - - - - - - - -	3784 2344 2445 2445 2446 2445 2445	1200.0 1800.0 2155.0
AA AA AA AA AA AA AA AA AA IntNROWS EP/IntFirst Resource of PI Interference of PI Interferenc	85640A H3520AA H3220AA H3280A1 H3280A1 H3280A1 H3280A2 H3280A1 H3280A1 H3280A1	2 3 9 10 14 14 14 14 10 10 10 10 10 10 10 10 10 10	1298 1073 12992 1992 1992 1992 1992 1992 1992 19	нац LAX LAX CAX CGG CGG CGG CGG CGG CGG CGG CG	13930 1298 13303 12478 12692 12692 12692 45 AM 655 AM 12692 12692 12692	OCC DPW MBA LAX LAX DFK LAX DFK LAX DFK DFK DFK DFK DFK DFK DFK DFK DFK DFK	1205 (А.3) 2211 (222 22306 2022 2006 2022 2006 2022 2006 2022 2006 2022 2006 2022 2006 2022 2006 2020 200 200 200 200 200 200 200 200 200 200 200 200 200 200 20	39 16 7 0 0 0 0 0 0 0 0 0 0 0 0 0	620 592 523 617 617 617 617 617 617 617 617 617 617	60 17 0 0 0 0 0 0 0 0 0 0 0 0 0	3784 2144 2475 2486 2486 2486 2486 2486 2486 2486 2485	1200.0 1800.0 2155.0
AA AA AA AA AA AA AA AA AA AA AA IntNROWS EE EE EE EE EE EE EE EE EE EE EE EE EE		2 3 9 10 14 14 14 14 14 14 14 14 14 14	1292 1073 12852 12852 13830 139300 13900 10000 10000 10000 10000000000	482 442 443 444 445 445 445 445 445 445	13530 1294 13303 13262 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 12855 128555 12855 12855 12855 12855 12855 12855 12855 12855 12	OCG DRV MBA JRC LAX LAX CAX CAX CAX CAX CAX CAX CAX CAX CAX C	1206 10.15 2211 2222 2206 2206 2006 2006 2006 2007	39 10 7 0 0 0 0 0 0 0 0 0 0 0 0 0	620 552 523 617	60 · · · · · · · · · · · · · · · · · · ·	3784 2344 2445 2445 2446 2445 2445	1200.0 1800.0 2155.0
AA AA AA AA AA AA AA AA AA IntNRows E: E: IntNRows But stoed AA AA AA AA	NS640A NS640A NI27AA NS840A NI27AA NS840A NI27AA NS640A	2 8 9 10 14 14 14 14 14 14 14 14 14 14	1299 10/73 1299 10/73 1299 10/93 10/9 10/93 10/93 10/93 10/9 10/93 10/93 10/9 10/93 10/9 10/93 10/9 10/93 10/9 10/93 10/9 10/9 10/9 10/9 10/9 10/9 10/9 10/9	нац LAX LAX CAX CGG CGG CGG CGG CGG CGG CGG CG	13930 1294 13303 12478 1300 12478 1370 12478 1370 12478 1370 1370 1370 12478 1370 1 7 7 100 100 100 100 100 100 100 100 10	OCG DFW MIA JFC LAX LAX EXT EXT EXT EXT EXT EXT EXT EXT EXT EX	1205 1619 1211 122 122 122 122 122 122	39 10 7 0 0 0 0 0 0 0 0 0 0 0 0 0	600 592 523 617 617 617 617 617 617 617 617 617 617	60 17 0 0 0 0 0 0 0 0 0 0 0 0 0	3784 2142 2475 2486 2495 2486 2497 2497 2497 2475	1200.0 1800.0 2155.0 2155.0 2150.0
AA AA AA AA AA AA AA AA AA AA FUTSTORAG EFI TOTORAG Security Notes and the AA AA AA AA AA AA AA AA AA AA AA AA	Kontechnic Kontechnich Kontechnich Kontechnic Kontechnic Kontech	2 3 3 10 14 14 14 14 14 14 15 15 11 2 1 2 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1	1298 1073 12892 12892 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830	на, LAX LAX LAX CGG CDFFAL Row Yo To To CDFFAL Row Yo To To CDFFAL Row Yo To To CDFFAL Row Yo To CDFFAL	13500 11264 13303 12652 1275 12753 127555 127555 127555 127555 127555 127555 127555 127555 127555 1	OCC DPW MIA LAX LAX EXT EXT EXT EXT EXT EXT EXT EXT EXT EX	1205 10.13 2211 2205 2205 2205 2005	39 10 7 0 7 7 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0	600 992 523 617 617 617 617 617 617 617 617 617 617	60 17 0 0 0 0 0 0 0 0 0 0 0 0 0	3784 2142 2475 2486 2475 2486 2475 1492 2475 2475 2475 2475 2475 2475 2475 247	2203.0 1600.0 2155.0 215.0
AA AA AA AA AA AA AA AA AA AA BANNOWS EE E BANNOWS DEF BANNOWS EE BANNOWS DEF BANNO DEF BANNO	ISSIAN ISSIAN	2 3 9 14 14 14 14 14 14 14 14 14 14	1229 1073 12892 12892 12992 12	нац LAX LAX LAX CGG CDEFAU Row Yo Yo Yo Yo Yo Yo ShringTypa JYX LAX DFW CGG DFW	13930 1298 13303 12478 13262 124788 12478 12478 12478 12478 12478 12478 12478 12478 12	осс ргу ма јк цх цх ма	1205 (A3 221) (22 22 22 22 2 2 2 2 2 2 2 2 2 2 2 2 2	39 10 7 0 7 7 0 0 0 0 0 10 10 10 10 10 10	600 592 523 617 617 617 617 617 617 617 617 617 617	60 10 10 10 10 10 10 10 10 10 1	3784 2142 2475 2486 2 4 7 2486 2 4 7 5 2475 2475 2475 2475 2475 2475 2	1200.0 1600.0 2155.0 215.0
AA AA AA AA AA AA AA AA AA AA FUTSTORAG EFI TOTORAG Security Notes and the AA AA AA AA AA AA AA AA AA AA AA AA	Kontechnic Kontechnich Kontechnich Kontechnic Kontechnic Kontech	2 3 3 10 14 14 14 14 14 14 15 15 11 2 1 2 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1	1298 1073 12892 12892 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830 13830	на, LAX LAX LAX CGG CDFFAL Row Yo To To CDFFAL Row Yo To To CDFFAL Row Yo To To CDFFAL Row Yo To CDFFAL	13500 11264 13303 12652 1275 12753 127555 127555 127555 127555 127555 127555 127555 127555 127555 1	OCC DPW MIA LAX LAX EXT EXT EXT EXT EXT EXT EXT EXT EXT EX	1205 10.13 2211 2205 2205 2205 2005	39 10 7 0 7 7 0 0 0 0 0 10 10 10 10 10 10	600 992 523 617 617 617 617 617 617 617 617 617 617	60 17 0 0 0 0 0 0 0 0 0 0 0 0 0	3784 2142 2475 2486 2475 2486 2475 1492 2475 2475 2475 2475 2475 2475 2475 247	1200.0 1800.0 2150.0
AA AA AA AA AA AA AA AA AA IntNRows E E E Rows to P IntNRows E E Rows to P IntNRows Build Cases Build Cases Build Cases Build Cases Build Cases Cases Build Cases	H5564A H277A H278A H288A H280A H280HA	2 3 9 10 14 14 14 14 14 14 14 14 14 14	1298 1973 19892 19892 19830 19940 199500 19950 19950 19950 199500 19950 19950 19950 19950 19950 19950	на ЦАХ ЦАХ ЦАХ ОСС ВГЛ С ОССАН С С С С С С С С С С С С С	13500 1298 13303 13303 13503 12692 13502 13502 13502 13502 13502 13502 13502 13502 1005	OCC DPV MIA LAX LAX LAX SHORE	1206 IA35 IA35 I231 I232 I230 I2 I I2 I I2 I I2 I I2 I I2 I I2 I I2	39 10 7 7 7 7 7 7 7 7 7 7 7 7 7	600 592 523 617 617 617 617 617 617 617 617 617 617	ARE_DELVA_MENT Image:Type Top	3784 2142 2475 2486 2 475 2486 2 475 2475 2475 2475 2475 2475 2475 247	1200.0 1600.0 2150.0 215.0

Processor Output

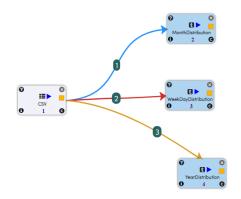
Distribution Graphs

Workflow

This workflow reads a CSV file. It then plots graphs of distribution of data by Week, Month & Year.

Below is the workflow. It does the following:

- Reads a CSV file.
- Distribution of data by Week.
- Distribution of data by Month.
- Distribution of data by Year.



Reading CSV file

Processor Configuration

Processor Output

Distribution of data by Week

It reads CSV files.

It plots graphs of distribution of data by Week using GraphWeekDistribution Node.

DUTPUT STORAGE LEVEL : O		DEFAULT		•		
ATH * : 🕹		data/YearSample.csv			BROWSE HDFS	
					VIEW FILE	
PARATOR : \varTheta						
EADER : 0		true		*		
ROP MALFORMED : 0		folse		•		
CHEMA COLUMNS : O REFRESH SCHEMA	•					
DLUMN NAMES FOR THE CEV O	COLUM	N TYPES FOR THE CSV O		COLUMN FORMATS FOR THE CSV		
d	INTE	GER	٣	format		•
in	STRI	NG	٠	format		•
12	STRI	NG		format		•

ecuting Node fire.nodes.dataset.No	odeDatasetCSV : 1 Nov 14, 2018 5:21:16 AM	
Row Values		
	dit	di2
ntegerType	StringType	StringType
	2003-07-25	2007-03-11
	2000-06-01	2013-12-26
	2006-07-21	2000-08-31
	2012-05-18	2001-05-12
	1999-10-20	2015-08-05
	2007-08-31	2012-03-05
	2012-01-15	2002-07-26
	2007-06-14	2012-12-18
	2005-10-03	2015-07-10
D	2000-12-26	2003-05-20

CHEMA :					
COLUMN NAME		id	dtl	dt2	
COLUMN TYPE		integer	string	string	
COLUMN FORMAT					
DUTPUT STORAGE LEVEL : O	DEFAULT		٣		
TITLE :	Week Day	Distribution			
HART TYPE : O	Line Char	t	•		
r Columns : O	dt1 : string dt2 : string				

OK

Processor Output

/eekDay[Distribution
Executing	Node fire.nodes.graph.NodeGraphWeekDayOlstribution : 3 Nov 14, 2018 5:27:36 AM
🔘 Input S	chemo
Week Day	y Distribution
	ay Distribution
	4
4	
	¹⁰ Wednesday Tuesday Friday Thursday Monday Saturday Saturday ■ d1 ■ d2
	٥

Distribution of data by Month

It plots graphs of distribution of data by month using GraphMonthDistribution Node.

Processor Configuration

CHEMA :				
COLUMN NAME		id	dt1	dt2
COLUMN TYPE		integer	string	string
COLUMN FORMAT				
DUTPUT STORAGE LEVEL :	DEFAULT Month Dis	tribution	•	
HART TYPE : O	Line Char	1	•	
COLUMNS : 0	dt1 : string dt2 : string		-	

NonthDistribution								
Executing Node fire.r	nodes.graph.Node0	GraphMonthDistribution	: 2 Nov 14, 2018	5:28:43 AM				
Input Schema								
Month Distribution								
Month Distribution	IN	IRT HISTOGRAM						
2	AR CHART UNE CHA	HISTOGRAM						
8 1 -								
0	2 3 4	5 6 7	8 9 #2	10 11	12			
								•

Distribution of data by Year

It plots graphs of distribution of data by year using GraphYearDistribution Node.

Processor Configuration

YearDistribution 7 0 NodeGraphYearDistribution					4
SCHEMA :					
COLUMN NAME		id	dt1	dt2	
COLUMN TYPE		integer	string	string	
COLUMN FORMAT					
OUTPUT STORAGE LEVEL : O	DEFAULT		٣		
TITLE :	Year Distr	ibution			
CHART TYPE : 0	Line Char	1	•		
Y COLUMNS : O	dt1 : string dt2 : string				
				ОКС	ANCEL

Processor Output

YearDistri	bution																		1
Executio	g Node fire	.nodes.gr	aph.Noc	leGraph	YearDistr	ibution :	4 Nov	14, 2018	5:29:32	AM									
Input !	Schema																		
Year Dist	ribution																		
Year Di	STRIBUTION	BAR CHART	LINE	CHART	HISTOGR	м													
	2																		
56	1	h		h	ł	h													
	0 2003	2007	2006	2012	2005	2000 t1	1999 dt2	2015	2013	2001	2002	-							
																			ок

Farmers Markets On Geo Maps

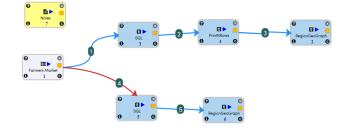
This workflow reads in a dataset. It then plots number of Farmers Market by City and by State on a Graph.

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Executes SQL Query for state count.
- Prints the result after executing query for state counts.
- Plots Graph for farmers with state counts.

- Executes SQL Query for city counts.
- Plots Graph for farmers with city counts.



Reading from Dataset

It reads sample Dataset files.

Processor Configuration

DatasetStructured 2 O Details NodeDatasetStructured			1
OUTPUT STORAGE LEVEL :	DEFAULT	٠	
DATASET : 😧	Farmers Market	٠	
			OK CANCEL

Row Values						
FMID	MarketName	Website	Facebook	Twitter	Youtube	01
IntegerType	StringType	StringType	StringType	StringType	StringType	St
1011871	Stearns Homestead Farmers' Market	http://Stearnshomestead.com				
1011878	100 Mile Market	http://www.pfcmorkets.com	https://www.facebook.com/100MileMorket/?fref=ts			ht
1009364	106 S. Main Street Farmers Market	http://thetownofsixmile.wordpress.com/				
1010691	10th Steet Community Formers Market					н ty
1002454	112st Modison Avenue					
1011100	12 South Farmers Market	http://www.12southformersmarket.com	12_South_Formers_Market	@t2southfrmsmkt		6
1009845	125th Street Fresh Connect Formers' Market	http://www.i25thStreelFormersMarket.com	https://www.facebook.com/125thStreetFarmersMorket	https://twitter.com/FormMarket125th		In
1005586	12th & Brandywine Urban Farm Market		https://www.facebook.com/pages/12th-Brandywine- Urban-Farm-Community-Garden/253769448091860			h

Execute SQL Query

It Executes SQL Query for state count from the SQL node.

Processor Configuration

NodeSQL	0															3
SCHEMA :																
COLUMN NAME	FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia	street	city	County	State	zip	Season1Date	Season1Time	Season2Date	Season2
COLUMN TYPE	integer	string	string	string	string	string	string	string	string	string	string	string	string	string	string	string
COLUMN FORMAT																
OUTPUT ST	ORAGE LEV	EL : 0				DEFAULT										
TEMP TABL	E : 😡					fire_temp_	table									
QL:0																
	ce state,		TTON TIPE	terp_table :	100p 07 3											
SCHEMA CO	DLUMINS : 6	REFRESH SCHE														
	DLUMINS : 6	REFRESH SCHE				COLUMN TY				out	PUT COLU	IMN FOR	AATE O			
SCHEMA CO	DLUMINS : 6	REFRESH SCHE				COLUMN TY					PUT COLU	IMN FORF	AATS O			1
SCHEMA CO	DLUMINS : 6	REFRESH SCHE			OUTPUT	COLUMN TYP				• fo		IMN FOR	AATS Ø		•	
Schema Co NUTPUT COL Stole	DLUMINS : 6	REFRESH SCHE				COLUMN TYP				• fo	rmat	UMN FORP	MATE Ø			

Processor Output

count
LongType
L. C.
1
1
1
1
1
1
3

Prints the Results

It prints the results after executing query for state counts by SQL Node.

CHEMA :				
COLUMN NAME		State	count	
COLUMN TYPE		string	long	
COLUMN FORMAT				
DUTPUT STORAGE LEVEL : 😡	DEFAULT		•	
mue :	Row Values			
UM ROWS TO PRINT : 0	100			

Processor Output

xecuting Node fire.nodes.util.NodePrintFirstNRo	ws : 4 Nov 14, 2018 4:23:19 AM		
Input Schema			
Row Values			
State		count	
StringType		LongType	
Dhio		1	
District of Columbia		1	
Delaware		1	
Missouri		1	
Michigon		1	
Tennessee		1	
South Carolina		1	
New York		3	

Analyze using Graph

It plots Graph for farmers with state counts using RegionGeoGraph Processor.

Processor Configuration

SCHEMA :				
COLUMN NAME		State	count	
COLUMN TYPE		string	long	
COLUMN FORMAT				
DUTPUT STORAGE LEVEL : O	DEFAULT		•	
ITLE:	State Map			
COLUMN 1:	State : string		*	
COLUMN 2 :	count : long		*	
DISPLAY MODE :	markers		*	
RESOLUTION :	provinces		*	
REGION : O	US		•	

Processor Output

xecuring Node Hra.nodes.graph NodeGraphRegionGeo : 2 Nov I4, 2018 4:22:12 AM	
Input schemo	
tate Map	
iate Mop	
Row Values	
State	count
StringType	LongType
Ohin	1
	1
District of Columbia	

Execute SQL Query

It executes SQL Query for City count from the SQL node.

Processor Configuration

CHEMA :																
COLUMN	FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia	street	city	County	State	zip	Season1Date	Season1Time	Season2Date	Season2
COLUMN	integer	string	string	string	string	string	string	string	string	string	string	string	string	string	string	string
COLUMN																
NUTPUT ST	ORAGE LEV	EL : 🛛				DEFAULT							٠			
MP TABL	E:0					fire temp t	lable									
L: 0		count(*) count 4	ros firo_t	onp_table gr												
OL: 0 1 sele	oct city,		_		oup by ci	ty limit 10				our	PUT COLU		4475 Đ			
ICHEMA CO	et city,		_		oup by ci	ty lisit 10					PUT COLU	JMN FOR	MATS @			
1 sold	oct city,		_		OUTPUT	ty list 10 COLUMN TYP				• for		MAN FORM	MATE @			

Input Schema	
Row Values	
city	count
StringType	LongType
Lamar	1
Six Mile	1
Nashville	1
Washington	1
Wilmington	1
Kalamazoo	1
Parma	1
New York	3

Analyze using Graph

It plots Graph for farmers with City counts using RegionGeoGRaph Node.

Processor Configuration

SCHEMA :				
COLUMN NAME		city	count	
COLUMN TYPE		string	long	
COLUMN FORMAT				
DUTPUT STORAGE LEVEL : O	DEFAULT		*	
TITLE :	City Map			
COLUMN 1:	city:string		•	
COLUMN 2 :	count : long		•	
DISPLAY MODE :	markers		•	
RESOLUTION :	provinces		•	
REGION : O	US			

Processor Output

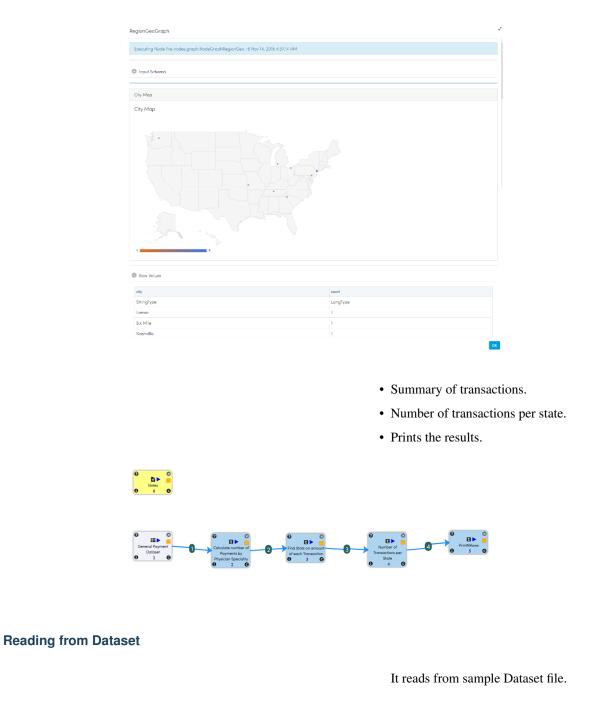
General Payment Data Analysis

Workflow

This workflow reads in a dataset. It then performs detailed analytics on general payment dataset.

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Calculates count transactions by speciality.



DatasetStructured 2 O Details NodeDatasetStructured			
OUTPUT STORAGE LEVEL : @	DEFAULT	×	
DATASET : 😡	General Payment Dataset	•	
			OK CANCEL

Row Values								
Covered_Recipient_Type	Teaching_Hospital_ID	Teaching_Hospital_Name	Physician_Prefile_ID	Physician_First_Name	Physician_Middle_Name	Physician_Lost_Name	Physician_Name_Suffix	Recipient_Prin
StringType	StringType	StringType	IntegerType	StringType	StringType	StringType	StringType	StringType
Covered Recipient Physician			2	LINDSEY	R	GROBER		75 N COUN
Covered Recipient Physician			2	LINDSEY		GROBER		75 N Countr
Covered Recipient Physician			з	JUSTIN		RACKLEY		1001 SAM P
Covered Recipient Physician			4	LISA	L	HAMAKER		633 W GERI
Covered Recipient Physician			4	LISA	L	HAMAKER		633 W GER

Calculate count transactions by speciality

It will calculate count transactions by speciality using BarChartCal Node.

Processor Configuration

CHEMA :							
COLUMN NAME	Covered_Recipient_Type	Teaching_Hospital_ID	Teaching_Hospital_Name	Physician_Profile_ID	Physician_First_Name	Physician_Middle_Name	Physician_Last_Nar
COLUMN	string	string	string	integer	string	string	string
COLUMN FORMAT							
UTPUT ST	ORAGE LEVEL : 😡		DEFAULT		•		
COLUMNNAME *: 0			Physician_Specialty : string				

xeculing Node fire.nodes.ml.NodeBarChartCal : 2 Nov 15, 2018 1:32:19 AM	
Input Schema	
Distribution of categorical values of physician_specially	count
Allopathic & Osteopathic Physicians/ Internal Medicine/ Endocrinology, Diabetes & Metabolism	4
Allopathic & Osteopathic Physicians/ Emergency Medicine	2
Allopathic & Osteopathic Physicians/ Anesthesiology	1
Allopathic & Osteopathic Physicians/ Internal Medicine	1
Allopathic & Osteopathic Physicians/ Family Medicine	2

Summary of transactions

It finds stats on amount of each transaction using Summary Node.

Processor Configuration



Processor Output

ind Stats on amount of ea	Transaction						
Executing Node fire.nodes.ml	Fxeculing Node Fire nodes ml.NodeSummary : 3 Nov 15, 2018 133:32 AM						
Input Schema							
Summary							
summary	total_amount_of_payment_usdellars						
count	10						
mean	33.384						
min	10.24						
25%	11.37						
50%	12.6						
75%	14.67						
max	19.49						
stdev	43.85205080723136						
variance	1923.0023600000002						

Number of transaction per state

It finds number of transactions per state using SQL Node.

Processor Configuration

Processor Output

Prints the results

It will print the result of output getting from SQL Node.

Number NodeSQL	of Transactions per 9	State 🛛 😡					4
SCHEMA :							
COLUMN NAME	Covered_Recipient_Type	Teaching_Hospital_ID	Teaching_Hospital_Name	Physician_Profile_ID	Physician_First_Name	Physician_Middle_Name	Physician_Last_Nar
COLUMN TYPE	string	string	string	integer	string	string	string
COLUMN FORMAT							
OUTPUT STO	RAGE LEVEL : O		DEFAULT				
TEMP TABLE	Ð		fire_tamp_table				
SQL: 🛛							
SCHEMA CO	LUMINS : Ø REFRESH SCHEMA						
DUTPUT COL	UMN NAMES Ø	OUT	PUT COLUMN TYPES 🛛	ou	TPUT COLUMN FORMATS O		
Recipient_	State	s	TRING	•	ormat		•
count			ONG	• fe	ormat		•
							•

Number of Transactions per State	/
Executing Node fire.nodes.etl.NodeSQL : 4 Nov 15, 2018 1:37:53 AM	
Input Schema	
Row Values	
Recipient_State	count
StringType	LongType
VA	1
IL .	2
PA	4
NY	2
тх	1
	οκ

CHEMA :		
COLUMN NAME	Recipient_State	count
COLUMN TYPE	string	long
COLUMN FORMAT		
UTPUT STORAGE LEVEL : O	DEFAULT Row Volues	v
UM ROWS TO PRINT : 0	10	

Processor Output

xecuting Node fire.nodes.util.NodePrintFirstNRows : 5 Nov 15, 201	1:38:55 AM	
Input Schema		
Row Values		
Recipient_State	count	
Recipient_State StringType	count LongType	
StringType	LongType	
StringType VA	LongType 1	
StringType VA IL	LongType 1 2	

Jetrail Data Analysis

Workflow

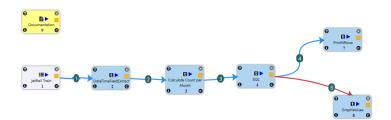
This workflow reads in a dataset. It then calculates the monthly trend in JetRail Dataset and annalyses using graph.

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- Extracts date time field.
- Calculates count per month.
- Executes query for months.
- Print the results.
- Graphical analysis.

Reading from Dataset

It reads from sample Dataset file.



DatasetStructured @ O Details NodeDatasetStructured			2
OUTPUT STORAGE LEVEL : 🖗	DEFAULT	*	
DATASET : O	JetRail Train	٣	
			OK CANCEL

Processor Output

Row Values	
latetime	Count
imestompType	IntegerType
012-08-25 00:00:00.0	8
012-08-25 0E00:00.0	2
012-08-25 02:00:00.0	6
012-08-25 03:00:00.0	2
012-08-25 04:00:00.0	2
012-08-25 05:00:00.0	2
012-08-25 06:00:00.0	2
012-08-25 07:00:00.0	2
012-08-25 08:00:00.0	6
012-08-25 09:00:00.0	2

Extract date time field

It extracts year and month field from date time field of timestamp using date time field extract Node.

Processor Configuration

Processor Output

Calculate count per month

It calculates count per month using query by SQL Node.

deDateTimeFieldExtract			2
HEMA :			
DLUMN NAME	Datetime	Count	
OLUMN TYPE	timestamp	integer	
OLUMN FORMAT	dd-MM-yyyy HHmm		
TPUT STORAGE LEVEL : 0			
TPUT STORAGE LEVEL : O	DEFAULT	•	
LUMN : 😡	Datetime : timestamp		
FRACT YEAR : Ø	true	*	
TRACT MONTH : O	true	*	
TRACT DAY OF MONTH : 0	folse	*	
TRACT HOUR : 0	false	*	
TRACT MINUTE : O	false	*	
TRACT SECOND : O	false	*	

excuting Node fire.nodes.etl.NodeDateTimeF	ieldExtract : 2 Nov 14, 2018 11:49:02 PM		
) Input Schema			
Row Values			
Datetime	Count	Datetime_year	Dotetime_month
TimestompType	IntegerType	IntegerType	IntegerType
2012-08-25 00:00:00.0	8	2012	8
2012-08-25 01:00:00.0	2	2012	8
2012-08-25 02:00:00.0	6	2012	8
2012-08-25 03:00:00.0	2	2012	8
2012-08-25 04:00:00.0	2	2012	8
2012-08-25 05:00:00.0	2	2012	8
2012-08-25 06:00:00.0	2	2012	8
2012-08-25 07:00:00.0	2	2012	8
2012-08-25 08:00:00.0	6	2012	8
2012-08-25 09:00:00.0	2	2012	8

ок

ICHEMA :									
COLUMN NAME	Datetime		Count	Dateti	Datetime_year	Date	Datetime_month		
COLUMN TYPE	timestamp		integer inte		nteger	inte	3er		
COLUMN FORMAT	dd-MM-yyyy HH:mm								
DUTPUT STORAGE LEVEL : O		DEFAULT							
EMP TABLE : O		fire_temp_table							
tire_temp_table									
1 soloct cast(Datotime_ye	ar as string) as Datating year , c m fire_temp_table group by Datetin			nonth ,					
i solot castibattine ye 2 swa(Gount) as Count fr				nonth ,					
I soloct cast(batche ye 2 swat(Guart) as Count fr Count free	n firz_tesp_table group by Ostetia I SCHEMA				UTFUT COLUMN FORMAT	30			
E ESCET CASESDATETHOL YE 2 SAR(COUNT) IN COUNT FO COMMA COLUMNS IN ETTAGE UTFUT COLUMN NAME O	n fire_teng_table group by belefit	e_yeer, Oxtetime_rooth		0	UTFUT COLUMN FORMAT	50			•
2 son(Count) as Count $\hat{T}r$	I KCHIMA 0 orr	vr Column TryPs Ø		•		30		_	•

Processor Output

xecuting Node fire.nodes.etl.NodeSQ	L : 3 Nov 15, 2018 12:03:54 AM		
Input Schema			
Row Values			
Datetime_year	Datetime_month	Count	
StringType	StringType	LongType	
2012	8	34	

Execute query for months

It executes query for grouping and selecting required fields, calculates sum of counts by SQL Node.

Processor Configuration

Processor Output

Prints the Results

It prints the results after executing SQL Query

align center width 60%

Graphical analysis

It will graphically represent month with count using GraphValue Node.

Processor Configuration

Processor Output

NYC Taxidata Analysis

This workflow reads in a sample dataset. It then analyses average speed of taxis at each hour with sample data and prints the results.

Below is the workflow. It does the following:

- Reads data from a dataset.
- Extracts hour from pickup time.
- Calculates the speed per hour.

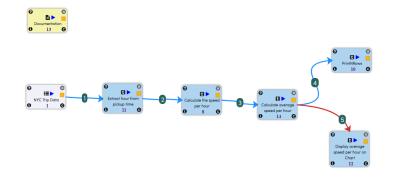
Chapter 12. Tutorials

Workflow

GraphValues 22 NodeGraphValues				8
SCHEMA :				
COLUMN NAME		year_month	Count	
COLUMN TYPE		double	double	
COLUMN FORMAT				
OUTPUT STORAGE LEVEL : 0	DEFAULT			
TITLE:	Graph			
XLABEL	year_month	1		
Y LABEL :	count			
CHART TYPE:	Line Chart			
IS STREAMING7 : O	false			
X COLUMN :				
Y COLUMNS :	year_mont			
			OK C	INCEL

GraphValues	,
Executing Node fire.nodes.ç	graph.NodeGraphValues : 8 Nov 15, 2018 12:12:26 AM
Input Schema	
Graph	
Graph социми снаят Вая сная	YT UNE CHURT HETTOGRAM
34	Cont
	or a state of the

- Calculates the average speed per hour.
- Prints the results.
- Displays average speed per hour on chart.



Reading from Dataset

It reads sample Dataset files.

Processor Configuration

Read in NYC Trip Data 2 0 Details NodeDatasetStructured			2
OUTPUT STORAGE LEVEL : O	DEFAULT	×	
DATASET : 🖗	NYC Trip Data	×	
		ОК	CANCEL

Row Values							
medallion	hack_license	vendor_id	rate_code	store_and_fwd_flag	pickup_datetime	dropoff_datetime	possenger_count
StringType	StringType	StringType	IntegerType	StringType	TimestampType	TimestampType	IntegerType
740BD5BE61840BE4FE3905CC3EBE3E7E	E48B185060FB0FF49BE6DA43E69E624B	CMT	1	N	2013-01-10 12:44:00.0	2013-01-10 12:53:00.0	1
EA05309C30E375695E44C96108ACB10E	IDI0D8AC5B07D8086I76I365A05A8AE2	CMT	1	N	2013-02-10 19:44:00.0	2013-02-10 20:03:00.0	1
EA05309C30F375595F44C96108ACB10F	1D10D8AC5B07D80861761365A05A9AE2	CMT	1	N	2013-02-10 22:01:00.0	2013-02-10 22:09:00.0	1
B36D7AB5B422EA1A0588EFD1D8155EF3	669F420B42A0739A5D1058184AD227ED	CMT	1	N	2013-03-10 12:14:00.0	2013-03-10 12:26:00.0	1
28172009F5513B25F4091C0929C4515D	ABE06BCFEBF3F5F7339CA2170EECIDEE	CMT	1	N	2013-02-10	2013-02-10	1
90C0771FE30F703616B3A31414470E18	8D466661814D45936D897907D4B4C602	CMT	1	N	2013-03-10 18:35:00.0	2013-03-10 18:37:00.0	1
37E1F5BD15A35A1652745C88604FC19E	4606384EA9F41F9BC76AA34B44FB5AAB	CMT	1	N	2013-03-10 13:32:00.0	2013-03-10 14:00:00.0	1
56EC8E3AA6218867A1341249F26531F3	DD72E3EE33F0CEDB5F068218010CC62A	CMT	1	N	2013-04-10 12:09:00.0	2013-04-10 12:26:00.0	1
4E67DD6E62DAD3CC9F8C84132F96AD2D	B169EAD638D515A2DAD3CC8B268A3FEF	CMT	2	N	2013-02-10 17:07:00:0	2013-02-10 18:04:00.0	1
6DFB708A2122478FI36B3227E18930B5	8992ECFE72103DD0DC2165D9E3CF8DFC	CMT	1	N	2013-02-10 23:17:00.0	2013-02-10 23:34:00.0	1

Extract hour from pickup time

It extracts hour from pickup time using datetimefieldextract Node.

Processor Configuration

CHEMA :											
COLUMN NAME	medallion	hack_license	vendor_id	rate_code	store_and_fwd_flag	pickup_datetime	dropoff_datetime	passenger_count	trip_time_in_secs	trip_distance	pic
COLUMN	string	string	string	integer	string	timestamp	timestamp	integer	integer	double	dou
COLUMN						dd/MM/yyyy HH:mm	dd/MM/yyyy HHomm				
	ONTH : O	0			pickup_datetime : time folse folse	anange		•			
EXTRACT H					true			•			
XTRACT SE	ICOND : O				false						

Processor Output

Input Schema							
Row Values							
medallion	hock_license	vendor_id	rate_code	store_and_fwd_flag	pickup_datetime	dropoff_datetime	passenger_count
StringType	StringType	StringType	IntegerType	StringType	TimestampType	TimestampType	IntegerType
740BD50E61840BE4FE3905CC3EBE3E7E	E46B185060F80FF49BE6DA43E69E624B	CMT	1	N	2013-01-10 12:44:00.0	2013-01-10 12:53:00.0	1
EA05309C30F375595F44C96108ACB10F	1D10D8AC5B07D80861761365A05A9AE2	CMT	1	N	2013-02-10 19:44:00.0	2013-02-10 20:03:00.0	1
EA05309C30F375695F44C96108ACB10F	1D10D8AC5B07D80861761365A05A9AE2	CMT	1	N	2013-02-10 22:01:00.0	2013-02-10 22:09:00.0	1
B36D7AB5B422EAIA0588EEDID8I55EE3	669F420B42A0739A5D1058184AD227FD	CMT	1	N	2013-03-10 12:14:00.0	2013-03-10 12:26:00.0	1
281/2009F5513B25F4091C0929C4515D	ABED8BCFEBF3F5F7339CA2170EEC1DEE	CMT	1	N	2013-02-10 00:33:00.0	2013-02-10 01:06:00.0	1
90C0771FE30F703616B3A31414470E18	8D4666661814D45936D897907D4B4C602	CMT	1	N	2013-03-10 18:35:00.0	2013-03-10 18:37:00.0	1
37E1F5BD15A35A1652745C88604FC19E	4606384EA9F4IF98C76AA34B44F85AA8	CMT	1	N	2013-03-10 13:32:00.0	2013-03-10 14:00:00.0	1
56EC8E3AA6218867A1341249F26531F3	DD/2E3EE33F0CED86F068218010CC62A	CMT	1	N	2013-04-10 12:09:00.0	2013-04-10 12:26:00.0	1
4E67DD5E52DAD3CC9F8C84132F96AD2D	BI69EAD638D5I5A2DAD3CC8B268A3FEF	CMT	2	N	2013-02-10 17:07:00.0	2013-02-10 18:04:00.0	1
6DFB708A2122478F136B3227E18930B5	8992ECFE72103DD0DC2165D9E3CF8DFC	CMT	1	N	2013-02-10 23:17:00.0	2013-02-10 23:34:00.0	1

Calculate the speed per hour

It calculates the speed per hour using SQL Node.

CHEMA :											
COLUMN	medallion	hack_license	vendor_id	rate_code	store_and_fwd_flag	pickup_datetime	dropoff_datetime	passenger_count	trip_time_in_secs	trip_distance	pic
	string	string	string	integer	string	timestamp	timestamp	integer	integer	double	dou
COLUMN						dd/MM/yyyy HH:mm	dd/MM/yyyy HH:mm				
UTPUT STC	ORAGE LEVEL :	0			DEFAULT			•			
EMP TABLE	E : 😡				temptable						
1 selec					ur, rip_distance = 0.0) then	0.0 else (temptable	.trip_distance*00*00)	/temptable.trip_time	e_in_secs end		
2 3 case 4 as sp	when (tempt peed from te	able.trip_time_ sptabla	in_secs = 0 of			0.0 else (temptable	.trip_distance*00*00)	/teeptable.trip_tim	s_in_secs end		
1 selec 2 3 case 4 as sp	when (tempt peed from te	oble.trip_time_ eptable	in_secs = 0 of	r temptable.t		0.0 cise (temptable	.trip_distance*00*00) OUTPUT COLUMM		t_la_secs and		
1 selec 2 3 case 4 as sp CHEMA CO	sites (terpt paed from te	oble.trip_time_ eptable	in_secs = 0 of	ourr	rip_distance = 0.0) then	0.0 else (temptable			t_la_secs and		
1 selec 2 case 4 as sp CHEMA CO ITPUT COL medallion	sites (terpt paed from te	eble.trip_time_ eptable REFRESH SCHEAM	in_secs = 0 of	ourr	rip_distance = 0.0) then UT COLUMN TYPES	0.0 else (temptable	OUTPUT COLUMN		be out at		

Processor Output

Executing Node fire.nodes.efl.NodeSQL : 8 Nov 14, 2018 2:28:58 AM		
Input Schemo		
Row Values		
medallion	pickup_datetime_hour	speed
StringType	IntegerType	DoubleType
740BD5BE6/840BE4FE3905CC3EDE3E7E	12	8.059701492537313
EA05309C30F375595F44C96108ACB10F	19	16.119402985074625
EA05309C30F375695F44C96108ACB10F	22	16.5
B36D7AB5B422EAIA0588EFDID8155EF3	12	4.702467343976778
28172009F5513B25F4091C0929C4515D	0	27.1975-49770290966
90C0771FE30F703616B3A31414470F16	18	18.26086956521739
37E1F5BD16A35A1652745C88604FC19E	13	4.321725691476591
56EC8E3AA6218867A1341249F26531F3	12	16.20689655172414
4E67DD5E52DAD3CC9F8C84132F96AD2D	17	18.38342573579603
	23	15.78646329637941

Calculate the average speed per hour

It calculates the average speed per hour using GroupBy Node.

OK

Processor Configuration

CHEMA :			
COLUMN NAME	medallion	pickup_datetime_hour	speed
COLUMN TYPE	string	integer	double
COLUMN FORMAT			
179/1 510AAC LEVEL • 0 DOUTING COLUMNS • 0		OEFAURT madallan: string pickag_detertimg_tour : integer space1 : double	
		AGGREGATE OPERATION TO USE O	
speed		• mg	• •
NING CLAUSE I O			
HERE CLAUSE : 0			
pickup_datatime_hour is not null OR speed is not n	a -		

xecuting Node fire.nodes.etl.NedeGroupBy : 14 Nov 14,	018 2:3419 AM	
Input Schema		
Row Values		
pickup_datetime_hour	avg_speed	
integerType	DoubleType	
2	9.656356129412743	
22	16.5	
13	4.321728691476591	
19	16.119402985074625	
17	18.38342573679603	
23	15.78646329837941	
D	27.197549770290966	
8	18.26086956521739	

Prints the results

It will print the result with the output of GroupBy Node.

Processor Configuration

PrintNRows 2 0 NodePrintFirstNRows		
SCHEMA :		
COLUMN NAME	pickup_datetime_hour	avg_speed
COLUMN TYPE	integer	double
COLUMN FORMAT		
OUTPUT STORAGE LEVEL : 😡	DEFAULT	•
TITLE :	Row Volues	
NUM ROWS TO PRINT : O	24	

Processor Output

Executing Node fire.nodes.util.NodePrintFirstNRows : 10	v 14, 2018 2:43:05 AM
Input Schemo	
Row Values	
pickup_datetime_hour	avg_speed
IntegerType	DoubleType
12	9.656355129412743
22	16.5
13	4.321728691476591
19	16.119402985074625
17	18.38342573679603
23	15.78646329837941
0	27.197549770290966
18	18.26086956521739

Analyze using Chart Graph

It displays average speed per hour on chart using Graphvalue Node.

Processor Configuration

Processor Output

Transaction Data Analytics

This workflow reads in a dataset. It then prints the results from the sample dataset and analyses using graphs.

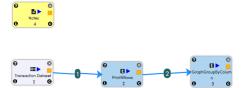
Display average speed per hour on Chart NodeGraphValues		
SCHEMA :		
COLUMN NAME	pickup_datetime_hour	avg_speed
COLUMN TYPE	integer	double
COLUMN FORMAT		
OUTPUT STORAGE LEVEL : O	DEFAULT	•
TITLE :	Graph	
X LABEL :	Hour	
Y LABEL :	Avg Speed	
CHART TYPE :	Line Chart	•
IS STREAMING? : O	false	Ŧ
X COLUMN :	pickup_datetime_hour : integer	
Y COLUMNS :	pickup_datefime_hour : integer ovg_speed : double	1
	nall-theer or norm	

Display average speed per hour on Chart
Executing Node fire.nodes.graph.NodeGraphValues ; 12 Nov 14, 2018 2:48:04 AM
Input Schema
Graph
Graph Colume Caret are Caret are Caret are Coret
0 12 22 13 19 17 23 0 18 arg_speed

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset.
- It then prints the results from the sample dataset.
- Analysing using graphs.



Reading from Dataset

It reads Dataset File.

Processor Configuration

DatasetStructured 2 O Details NodeDatasetStructured			1
OUTPUT STORAGE LEVEL : O	DEFAULT	٣	
DATASET : O	Transaction Dataset	•	
			OK CANCEL

Processor Output

Row Values										
d	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType
86246	205	7	707	1078778070	12564	2012-03-02	12.0	OZ	1	7.59
86246	205	63	6319	107654575	17876	2012-03-02	64.0	OZ	1	1.59
86246	205	97	9753	1022027929	0	2012-03-02	1.0	ст	1	5.99
86246	205	25	2509	107996777	31373	2012-03-02	16.0	OZ	1	1.99
86246	205	55	5555	107684070	32094	2012-03-02	16.0	OZ	2	10.38
86246	205	97	9753	1021015020	0	2012-03-02	1.0	СТ	1	7.8
86246	205	99	9909	104538848	15343	2012-03-02	16.0	OZ	1	2.49
86246	205	59	5907	102900020	2012	2012-03-02	16.0	OZ	1	1.39
86246	205	9	921	101128414	9209	2012-03-02	4.0	OZ	2	1.5
86246	205	73	7344	1068142161	20285	2012-03-02	8.0	CT	1	5.79

Prints the sample Dataset Results

It prints sample Dataset Results.

Processor Configuration

HEMA :											
COLUMN NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
COLUMN TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double
COLUMN FORMAT											
DUTPUT STORAGE LEVE	L:0				DEFAULT Row Values					•	

Processor Output

Input Schem	na									
Row Values										
d	chain	dept	category	company	brond	date	productsize	productmeasure	purchasequantity	purchaseamount
ntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType
36246	205	7	707	1078778070	12564	2012-03-02	12.0	OZ	1	7.59
36246	205	63	6319	107654575	17876	2012-03-02	64.0	oz	1	1.59
	205	97	9753	1022027929	0	2012-03-02	1.0	СТ	1	5.99

Analysing using Graph

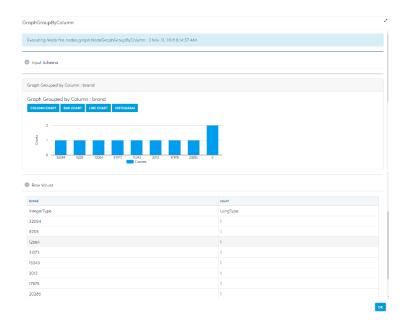
It helps to analyse using graph with Graph grouped by column brand and count.

Processor Configuration

COLUMN NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
COLUMN TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double
COLUMN FORMAT											
					X axis						
CLABEL :					C. MARK						
Y LABEL :					Y curis						
					Y axis brand : intege					•	

Processor Output

12.1.5 Data Preparation



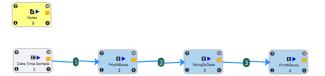
Convert To Timestamps

This example converts to timestamp from the input sample dataset using string to date Node.

Workflow

Below is the workflow. It does the following:

- Reads data from a sample dataset file.
- Prints sample dataset result.
- Converts sample string to timestamp.
- Prints the expected result.



Reading from Dataset

It reads sample Dataset File.

Processor Configuration

DatasetStructured @ O Details NodeDatasetStructured			2
OUTPUT STORAGE LEVEL : O	DEFAULT	•	
DATASET : 0	Date Time Sample	•	
			OK CANCEL

Processor Output

ecuting Node fire.nodes.	dataset.NodeDatasetStructured : 1 Nov 12, 2018 11:19:26 PM		
Row Values			
d	trans_dt	val	
ntegerType	StringType	DoubleType	
331800486	2012-03-15 01:34:46	2.8599978961939436E18	
1331857433	2012-03-15 17:23:53	2.7814041951551519E18	
1331856300	2012-03-15 17:05:00	2.7814041951551519E18	

Prints the sample Dataset Results

It prints the results of the sample dataset available.

ОК

Processor Configuration

CHEMA :				
COLUMN NAME	id	trans_dt	val	
COLUMN TYPE	integer	string	double	
COLUMN FORMAT				
UTPUT STORAGE LEVEL : O	DEFAULT		Ŧ	
TLE :	Row Volues			
JM ROWS TO PRINT : 0	3			

Processor Output

Convert To Timestamps

It converts To Timestamps using stringtodate Node.

Processor Configuration

Input Schema			
Row Values			
4	trons_dt	val	
ntegerType	StringType	DoubleType	
331800486	2012-03-15 01:34:46	2.8599978961939436E18	
331857433	2012-03-15 17:23:53	2.7814041951551519E18	
331856300	2012-03-15 17:05:00	2.7814041951551519E18	

CHEMA :				
COLUMN NAME	id	trans_dt		val
COLUMN TYPE	integer	string		double
COLUMN FORMAT				
DUTPUT STORAGE LEVEL : 1	DEFAULT		•	
IPUT COLUMN NAME : O	trans_dt:string			
IPUT COLUMN FORMAT : 🛛	yyyy-MM-dd HH:mm:ss			
UTPUT COLUMN NAME : Ø	trans_timestamp			
UTPUT COLUMN TYPE : O				

			Input Schema
			Row Values
trans_timestamp	val	trans_dt	id
TimestampType	DoubleType	StringType	IntegerType
2012-03-15 01:34:46.0	2.8599978961939436E18	2012-03-15 01:34:46	1331800486
2012-03-15 17:23:53.0	2.7814041951551519E18	2012-03-15 17:23:53	331857433
2012-03-15 17:05:00.0	2.7814041951551519E18	2012-03-15 17:05:00	331856300
2012-03-15 01:34:46.0 2012-03-15 17:23:53.0	2.8599978961939436E18 2.7814041951551519E18	2012-03-15 01:34:46 2012-03-15 17:23:53	1331856300

Prints the Results

It prints the results after converting to Timestamps.

Processor Configuration

CHEMA :						
COLUMN NAME	id		trans_dt	val	trans_timestamp	
COLUMN TYPE	integer		string	double	timestamp	
COLUMN FORMAT						
DUTPUT STORAGE LEVEL : Ø		DEFAUL	т		Y	
ITLE :		Row Val	105			
IUM ROWS TO PRINT : 0		10				

Processor Output

Input Schema			
Row Values			
d	trans_dt	val	trans_timestamp
ntegerType	StringType	DoubleType	TimestampType
331800486	2012-03-15 01:34:46	2.8599978961939436E18	2012-03-15 01:34:46.0
331857433	2012-03-15 17:23:53	2.7814041951551519E18	2012-03-15 17:23:53.0
	2012-03-15 17:05:00	2.7814041951551519E18	2012-03-15 17:05:00.0

Data Validation

Workflow

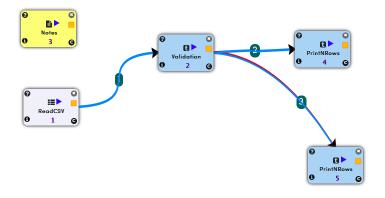
This example performs different kinds of data validation on input dataset like valid/invalid email,valid/invalid date,null/not null check etc.

Below is the workflow. It does the following:

- Reads data from a CSV file.
- Performs specific validation on specific columns.

Reading from CSV File

It reads data from a CSV file.



Processor Configuration

DUTPUT STORAGE LEVEL : O	DEFAULT	¢
ATH * : O	data/validation.csv	BROWSE HDFS VIEW FILE
EPARATOR : 🛛		VIEW FILE
EADER : O	false	¢
ROP MALFORMED : O	false	¢
CHEMA COLUMNS : O REFRESH SCHEMA	0	
KEYRESH SUHEHA	COLUMN TYPES FOR THE CSY O	COLUMN FORMATS FOR THE CSV Ø
COLUMN NAMES FOR THE CSV O	-	COLUMN FORMATS FOR THE CSY @
REFACES SCHOOL	COLUMN TYPES FOR THE CSY O	
COLUMN NAMES FOR THE CEV O	COLUMN TYPES FOR THE CSY O	¢ format
DUJMN NAMES FOR THE CEV O n 12 13	COLUMN TYPES FOR THE CEV O	¢ format format
COLUMN HAMES FOR THE CEV O	COLUMN TYPES FOR THE CEY O STRING INTEGER INTEGER	format format

Processor Output

Performing Validation

It performs different validation on different columns.

OK CANCEL

Processor Configuration

Processor Output

Multi-Validation Workflow

This workflow performs multiple validations on each incoming record

xecuting Node fir	e.nodes.dataset.NodeData	asetCSV : 1 Nov 15, 2018 12:0	4:27 PM		
Row Values					
fi	f2	ß	f4	email	dt
StringType	IntegerType	IntegerType	IntegerType	StringType	StringType
1	2	3	4	aa@bb.com	2018-05-05
6	7	8	9	bb@bb.com	2018-05-05
3	7	8	9	cc@bb.com	2018-05-05
9	7	8	9	cc@	2018-05-05
10	7	8	9	try@abc.com	2018/12/23
11	9	8	7	try@nc.com	23-12-2018

Validation @ O						
SCHEMA :						
COLUMN NAME	fl	f2	f3	f4	email	dt
COLUMN TYPE	string	integer	integer	integer	string	string
COLUMN FORMAT						
DUTPUT STORAGE LEVEL : O	E	DEFAULT		\$		
DESCRIPTION : O						
VARIABLES LIST :						
VARIABLES LIST:						

emoil \$ IS_VALID_EMAIL_ADDRESS \$ dt \$ IS_VALID_DATE_FORMAT \$ yyyy-MM-dd \$	
dt 💠 IS_VALID_DATE_FORMAT 💠 yyyy-MM-dd	
fi \$ VALUE_GREATER_THAN \$ 5	
fi	

OK CANCEL

xecuning Node In	re.nodes.etl.NodeValidation	: 2 Nov 15, 2018 12:06:12 PM	1		
Input Schema					
Row Values					
fi	f2	f3	f4	email	dt
StringType	IntegerType	IntegerType	IntegerType	StringType	StringType
6	7	8	9	bb@bb.com	2018-05-05

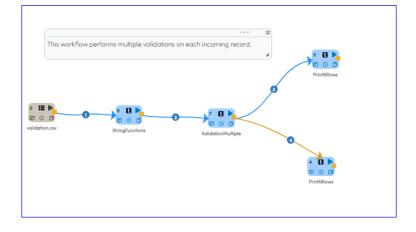
Validations

- Records which pass validation are output into the first edge
- Records which fail validation are output into the seconds edge
- Ensures that field is greater than or equal to specified string value
- Ensures that field is less than or equal to specified string value
- Ensures that field matches given datePattern
- Ensures that the email is valid
- Ensures field length is greater than or equal to specified length

Workflow

Below is the workflow. It does the following:

- Reads data from a CSV file.
- Performs specific validation on specific columns.



Reading from CSV File

 ${\tt DatasetCSV}$ processor reads data from a CSV file.

Processor Configuration

UTPUT STORAGE LEVEL : 🛛	DEFAULT		~			
ATH * : 0	/tmp/data/validation.csv			BROWSE HDFS VIEW FILE		
PARATOR : 🛛						
EADER : 🛛	false		~			
ROP MALFORMED : O	false		~			
CHEMA COLUMNS : 0 REFRESH SCHEMA	•					
COLUMN NAMES FOR THE CSV O	COLUMN TYPES FOR THE CSV O		COLUMN FORMATS FOR THE CSV	0		
fl	INTEGER	~	format		•	
f2	INTEGER	~	format		•	
f3	INTEGER	~	format		•	
			format			

Executing Node fit	re.nodes.dataset.NodeDat	asetCSV : 6 : Dec 24, 2020	6:09:25 PM				
Row Values							
ow Values							
fl	f2	f3	f4	email	date	name	
IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	
1	2	3	4	aa@bb.com	2018-05-05	Robert ryl	
6	7	8	9	bb@bb.com	2018-05-05	Daniell Hammack	
3	7	8	9	cc@bb.com	2018-05-05	Versie Hillebrand	
9	7	8	9	000	2018-05-05	Markita Hansen	
10	7	8	9	try@abc.com	2018/12/23	Kary Hendrixson	
11	9	8	7	try@nc.com	23-12-2018	Niesha Huffines	

String Functions

StringFunctions processor performs specified operation on the selected column (i.e. trim function for column 'name' in this case)

Processor Configuration

	n	f2	f3	f4	email	date	name	
TYPE	integer	integer	integer	integer	string	string	string	
FORMAT								
UTPUT STOR	AGE LEVEL :		DEFAULT		~			
PUT COLUMI	N NAME : 😡	Available			Selected		•	•
		email : string date : string			name : stri	19		
		dule : siring						
				•				
				÷				
				• •				
				÷				
				÷				
				÷				
				 €				
				 €				
RING FUNCT	10N : Đ		trim	• •				

Processor Output

	ien oorenen nooren nigi o	nctions : 8 : Dec 24, 2020 6	s:10:28 PM				
Input Schema							
							_
Row Values							
ow Values							
fl	f2	f3	f4	email	date	name	
IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	
1	2	3	4	aa@bb.com	2018-05-05	Robert ryl	
6	7	8	9	bb@bb.com	2018-05-05	Daniell Hammack	
3	7	8	9	cc@bb.com	2018-05-05	Versie Hillebrand	
9	7	8	9	cc@	2018-05-05	Markita Hansen	
	7	8	9	try@abc.com	2018/12/23	Kary Hendrixson	
10							

Performing Validation

ValidationMultiple processor performs different validation on different columns.

Processor Configuration

NAME	n	2	8		54	enol		date	nome	
196	irreger	integer	integer		integer	string		string	string	
TAMR										
CRIPTION : 6			DEFAU					×		
IABLES UST :		WALLER O	60.0 CONDITION O	Function O	VALUES O		CONDITION @ FUNC	:Tox Ø	VALUES O	
ABLES LIST : LUMINS Ø	FUNCTION O		CONDITION @	Function @	walues @		contintion @ Runs	tion e		
ABLES LIST : SUMNS O emoil		VALUES O	condition @		~					
HBLES UST : cumes e email		VALUED 0		•] [~		· ·	~		

Processor Output

	ltiple							
Input Schema								
ValidationMulti	ple							
TotalRecords	: 6 ,PercentOf8	SadRecords: 33.	.33, PercentOfG	oodRecords: 66.	.67			
Row Values								
Row Values								
	f2	f3	f4	email	date	name	validation_result	validation_result_reason
Row Values	f2 IntegerType	f3 IntegerType	f4 IntegerType	email StringType	date StringType	name StringType	validation_result	validation_result_reason StringType
Row Values							_	
Row Values fl IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	
Row Values fl IntegerType 1	IntegerType 2	IntegerType 3	IntegerType	StringType aa@bb.com	StringType 2018-05-05	StringType Robert ryl	IntegerType	

Prints the Valid Records

Processor Output

Prints the Invalid Records

Processor Output

Decision / JSON Parser / SortBy / Empty Dataset

Fire provides the following processors:

- JSON Parser Processor
- Decision Processor
- SortBy Processor
- Empty Dataset Processor

https://www.sparkflows.io/single-post/2018/09/05/New-Processors-Decision-JSON-Parser-SortBy-

Executin	g Node fire.	nodes.util.N	odePrintFirst	NRows : 9 : D	ec 24, 2020	6:14:43 PM				
Input Sch	ema									
Row Valu	55									
low Value	15									
fl	f2		f3	f4		email	date	name	validation_result	validation_result_reason
IntegerT	ipe Int	egerType	IntegerTyp	pe Integ	erType	StringType	StringType	StringType	IntegerType	StringType
1	2		3	4		aa@bb.com	2018-05-05	Robert ryl	1	
6	7		8	9		bb@bb.com	2018-05-05	Daniell Hammack	1	
3	7		8	9		cc@bb.com	2018-05-05	Versie Hillebrand	1	
11	9		8	7		try@nc.com	23-12-2018	Niesha Huffines	1	
										o
NRows										
becuting Node	fire.nodes.util.N	odePrintFirstNRow	m : 4 : Dec 24, 202	10 6:15:29 PM						
put Schema										
ow Values										
w Volues	12	ß	64	email	date	name	validation_result	validation_result_reaso	n	
	IntegerType	integer7ype	Integerfype	StringType	StringType	StringType	Integer?ype	StringType		
w Volues 11 ntegerfype		8	9	008	2018-05-05	Markita Hansen	0	cc@ is not a valid email add	915,,,	
1	7									

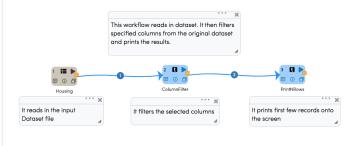
Column Filter

Workflow

This workflow reads in a dataset. It then filters specified columns from the original dataset and prints the results.

Below is the workflow. It does the following:

- Reads data from a dataset.
- It then filters specified columns from the original dataset.
- Prints the results.



Reading from Dataset

It reads in the input Dataset File.

Processor Configuration

DatasetStructured ProductiveConsectiveCount Detail			2 ×
OUTPUT STORAGE LEVEL : O	DEFAULT	v	
DATASET • : 🛛	Housing	~	
			OK CANCEL

Processor Output

tasetStruct	ured											1
Executing Nod	ie fire.nodes.data	set.NodeDatasetS	Rructured : 1 : Jan 5	, 2021 8:23:26 AM								
Row Values												
ow Values id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no
3	49500.0	3060	3	1	1	yes	no	no	no	no	0	no
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no

Column Filter

It filters the selected columns.

Processor Configuration

NAME id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	
TYPE intege	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	
ORMAT													
JTPUT STORAGE LEVEL	Ð			DEFAULT					*				
LUMNS : 😡			id : integer					Sele	cted			+	,
			price : double lottice : integer bathrms : integer driveway : string fulbase : string garagep! : integer prefarea : string				÷	rei ga air	ries : integer croom : string isthw : string rco : string				

olumnFilter					1
Executing Node fire.nodes.etl	NodeColumnFilter : 2 : Jan 5, 2021 8:24:07 AM				
Input Schema					
Row Values					
Row Values bedrooms	stories	recroom	gashw	airco	
IntegerType	IntegerType	StringType	StringType	StringType	
3	2	no	no	no	

Prints the Results

Drop Columns

Workflow

It prints the first few records onto the screen.

This workflow reads in a dataset. It then drops some columns from the original dataset and prints the results.

Below is the workflow. It does the following:

- Reads data from a dataset.
- It then drops some columns from the original dataset.
- Prints the results.



Reading from Dataset

It reads Dataset File.

Processor Configuration

	setStruct	ured										
TPUT STO	RAGE LEVE	L:0		DEFA	ULT				•			
TASET : 😧				Hous	ing				Ŧ			
tasetStructu	ured										ОК	CANCE
	le fire.nodes.do	ataset.NodeDa	tasetStructured	l : 1 Nov 12, 2016	8 6:42:38 AM							
	e fire.nodes.do	ataset.NodeDa	tasetStructured	l : 1 Nov 12, 2018 bathrms	s 6:42:38 AM	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
Row Values						driveway StringType	recroom StringType	fullbose StringType	gashw StringType	eirce StringType	gorogept IntegerType	preforeo StringType
Row Values id	price	lotsize	bedrooms	bathrms	stories				-			
Row Values Id IntegerType	price DoubleType	lotsize IntegerType	bedrooms IntegerType	bathrms IntegerType	stories IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType
Row Values Id IntegerType 1 2	price DoubleType 42000.0	lotsize IntegerType 5850	bedrooms IntegerType 3	bathrms IntegerType 1	stories IntegerType 2	StringType yes	StringType no	StringType yes	StringType no	StringType no	IntegerType	StringType
Row Values IntegerType 1 2 3	price DoubleType 42000.0 38500.0	Iotsize IntegerType 5850 4000	bedrooms IntegerType 3 2	bathms IntegerType 1 1	stories IntegerType 2 1	StringType yes yes	StringType no no	StringType yes no	StringType no no	StringType no no	IntegerType 1 0	StringType no no
Row Values id IntegerType 1 2 3 4	price DoubleType 42000.0 38500.0 49500.0	Iotsize IntegerType 5850 4000 3060	bedrooms IntegerType 3 2 3 3	bothrms IntegerType 1 1 1	stories IntegerType 2 1 1	StringType yes yes yes	StringType no no no	StringType yes no no	StringType no no no	StringType no no no	IntegerType 1 0 0	StringType no no no
Row Values	price DoubleType 4200D.0 38500.0 49500.0 60500.0	lotsize IntegerType 5850 4000 3060 6650	bedrooms IntegerType 3 2 3 3 3	bathrms IntegerType 1 1 1 1 1	stories IntegerType 2 1 1 2 2	StringType yes yes yes yes	StringType no no no yes	StringType yes no no no	StringType no no no no	StringType no no no no	IntegerType 1 0 0	StringType no no no no
Row Values id IntegerType 1 2 3 4 5	price DoubleType 42000.0 38500.0 49500.0 60500.0 61000.0	Iotsize IntegerType 5850 4000 3060 6650 6360	bedrooms IntegerType 3 2 3 3 3 2	bathrms IntegerType 1 1 1 1 1 1	stories IntegerType 2 1 1 2 1 1 2 1	StringType yes yes yes yes yes	StringType no no no yes no	StringType yes no no no no	StringType no no no no no	StringType no no no no no	IntegerType 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	StringType no no no no no

Drop Columns

It drops the columns whichever we want.

Processor Configuration

SCHEMA :													
COLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
COLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string
COLUMN FORMAT													
	il : O			DEF						•			
OUTPUT STORAGE LEV	EL : •			id : in price lotsia bodh stori drive	AULT tleger t double te tinteger trast integer trast integer tray tring to may the transformed transf					•			

Processor Output

Prints the Results

It prints the results after dropping the columns.

Processor Configuration

ecuting Node fire node	s.etl.NodeDropColumns : 2 Nov 12, 2018	6:43:37 AM			
scaling hode methode					
Input Schema					
Row Values					
id	price	driveway	gashw	garagepl	
IntegerType	DoubleType	StringType	StringType	IntegerType	
1	42000.0	yes	no	1	
2	38500.0	yes	no	0	
3	49500.0	yes	no	0	
	60500.0	yes	no	0	
4		ves	no	0	
4	61000.0				

IodePrintFirstNRow	5													
CHEMA :														
COLUMN NAME	id	price	lotsize	bedroon	ns	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
COLUMN TYPE	integer	double	integer	integer		integer	integer	string	string	string	string	string	integer	string
COLUMN FORMAT														
UTPUT STORAGE LEVEL	: 0				DEFAU Row Vi						•			

ecuting Node fire.nodes	.util.NodePrintFirstNRows : 3 Nov 12, 20	18 6:44:22 AM			
Input Schema					
Row Values					
i .	price	driveway	gashw	garagepl	
ntegerType	DoubleType	StringType	StringType	IntegerType	
	42000.0	yes	no	1	
	38500.0	yes	no	0	
	49500.0	yes	no	0	

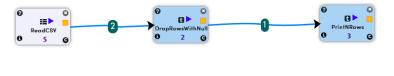
Drop Rows With Null

This example drops/filters the rows containing any null values from the input dataset.

Workflow

Below is the workflow. It does the following:

- Reads data from a CSV file.
- Drops rows having any null values in any of the columns.





Reading from CSV File

It reads data from a CSV file.

Processor Configuration

DUTPUT STORAGE LEVEL : O	DEFAULT		\$	
PATH * : 0	/user/ec2-user/data	a/data-with-nulls.csv	BROWSE HD	FS
			VIEW FILE	-
EPARATOR : 0	,			
IEADER : O	true		\$	
ROP MALFORMED : 0	false		\$	
CHEMA COLUMNS : O REFRESH SCHEMA	0			
REFREDIT SCHENOX	0 COLUMN TYPES FOR THE CO	SY Ø COLUA	AN FORMATS FOR THE CSY O	
REFREDIT SCHENOX		SV O COLUM COLUM		
NEWEN GUILDIN	COLUMN TYPES FOR THE C		at	•
COLUMN NAMES FOR THE CSY O	COLUMN TYPES FOR THE CO	\$ form	at	
COLUMN NAMES FOR THE CSY O	COLUMN TYPES FOR THE CO DOUBLE STRING	 form form 	at at	
COLUMN NAMES FOR THE CSY O id name gender	COLUMN TYPES FOR THE CO DOUBLE STRING	form form form form	ot at at	

Processor Output

	nodes.dataset.NodeDatase	LCCU - E No. 12 2018 11/16	16 4 4 4							
acuting Node fire.r	iodes.ddidsei.ivodeDdidse	IC3V : 5 NOV 13, 2016 11:10.	ID AIM							
Row Values										
id	name	gender	senior_citizen	resident	Family					
DoubleType	StringType	StringType	StringType	StringType	StringType					
1.0	ABC	F	Y	Y	N					
2.0	DEF	м	N	N						
3.0	GHR	м	Y	Y	Y					
4.0	JKL	F	N	Y	N					
5.0	RIT	м	Y	Y						
NaN		F	Y	Y	N					
5.0	PQR		Y		Y					

Dropping rows with null

It drops the rows which contain any null value.

Processor Configuration

CHEMA :						
COLUMN NAME	id	name	gender	senior_citizen	resident	Family
COLUMN TYPE	double	string	string	string	string	string
COLUMN FORMAT						
DUTPUT STORAGE LEVEL : O		DEFAULT			\$	

Family
StringType
N
Y

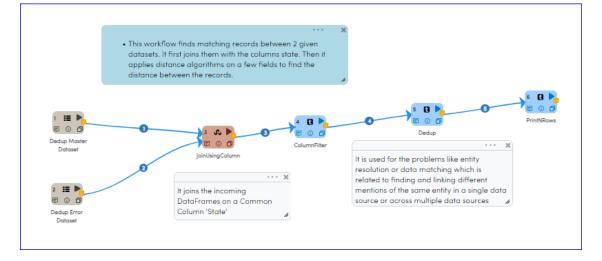
Dedup Customers

Data deduplication refers to a technique for eliminating redundant data in a data set. In the process of deduplication, extra copies of the same data are deleted, leaving only one copy to be stored.

Workflow

Below is the workflow. This workflow does the following:

- Finds matching records between 2 given datasets. It first joins them with the column "State".
- Then it applies distance algorithms on a few fields to find the distance between the records.



Input Datasets

There are 2 input datasets in this case "Dedup Master Dataset" & "Dedup Error Dataset" as shown below,

Dataset 1:

Dataset 2:

Join input DataFrames

JoinUsingColumn processor joins the incoming DataFrames on a join column

asetStructu	ired										2
Executing Node	fire.nodes.dataset.No	deDatasetStrue	ctured : 1 : Jan 5, 2	021 10:29:40 AM							
Row Values											
first_name	last_name	gender	birth_date	ethnicity	SSN	med_number	state	city	address	zip	id
StringType	StringType	StringType	StringType	StringType	IntegerType	DoubleType	StringType	StringType	StringType	IntegerType	IntegerType
Nathan	Cordova	female	11/09/1946	Asian	255383175	6.358029309E9	Nevada	Falanolzace	899 Casjole Grove	68085	848
Luke	Quick	female	5/28/1952	Asian	125087187	6.427424655E9	Pennsylvania	Gitopupriwa	499 Pihtowi Center	86488	157
Jodi	Baldino	female	1/22/1971	Asian	516395786	5.981030723E9	Georgia	Gupcowekuro	196 Pipafof Way	65928	980
Guled	Shen	male	4/15/1948	Pacific Islander	143355093	7.285462698E9	New Jersey	Fawwelcoja	682 Purima Junction	14597	225

atasetStructured											2
Executing Node fire.noc	des.dataset.NodeDatasetS	tructured : 2 : Jo	in 5, 2021 10:30:01	AM							
Row Values											
error_first_name	error_last_name	gender	birth_date	ethnicity	SSN	med_number	state	city	address	zip	error_id
StringType	StringType	StringType	StringType	StringType	IntegerType	DoubleType	StringType	StringType	StringType	IntegerType	IntegerTyp
Martinealyse	Nguyen	male	10/16/1943	Pacific Islander	833949858	1.036415183E9	Rhode Island	Gubaganuziw	802 Vepcat Circle	70969	462
Dillon	Ramirez	female	02/01/1965	Pacific Islander	174725823	5.124618654E9	Kansas	Pakepucawar	718 Fohgut Highway	14316	365
Rebecca	Manzanares	male	09/08/1900	Pacific Islander	637701044	1.248122191E9	Delaware	Ticgazsile	248 Ogugtez River	30854	797
Roegan	Mcneelv	NA	09/05/1960	Pacific	411405118	7 30208256759	Georgia	Mowuzeboouwi	29A	61440	117

"State". ColumnFilter processor filters the columns to get the required DataFrame as shown below:

Co	olumnFilter					2.5
	first_name	last_name	id	error_first_name	error_last_name	error_id
	StringType	StringType	IntegerType	StringType	StringType	IntegerType
	Nathan	Cordova	848	Nathan	Cordova	848
	Nathan	Cordova	848	jesse	Martinez	411
	Luke	Quick	157	eMrecdes	hCnarro	965
	Jodi	Baldino	980	Raegan	Mcneely	117
	Guled	Shen	225	Kaitlyn	Young	878
	Guled	Shen	225	Kiana	Cyaagding	575
	Guled	Shen	225	Lishrka	Reyna	141
	Guled	Shen	225	jądy	Plenty Wolf	556
	Viarlenny	Picazzo Banuelos	598	Lsis	Rodriguez	806
	Viarlenny	Picazzo Banuelos	598	Tilane	Thompson	976
	Viarlenny	Picazzo Banuelos	598	Andrew	Lattimer	865
						ОК

Data Deduplication

can be used:

Dedup is used for the problems like entity resolution or data matching. Entity resolution or data matching is the problem of finding and linking different mentions of the same entity in a single data source or across multiple data sources. Here Levenshtein Algorithm is used for data Deduplication. There are more options for Algorithms that

- Full matching: Full matching makes use of all individuals in the data by forming a series of matched sets in which each set has either 1 treated individual and multiple comparison individuals or 1 comparison individual and multiple treated individuals
- Levenshtein: It counts the number of edits (insertions, deletions, or substitutions) needed to convert one string to the other.
- Jaro-Winkler: The Jaro–Winkler distance is a string metric measuring an edit distance between two sequences. Jaro-Winkler are suited for comparing smaller strings like words and names.
- Jaccard (3 gram) : This takes consecutive words and group them as a single object. A 3-gram is a consecutive set of 3 words. Used for emails or small documents.
- Longest Common Subsequence : If a set of sequences are given, the longest com-

mon subsequence problem is to find a common subsequence of all the sequences that is of maximal length used in revision control systems, such as SVN and Git, for reconciling multiple changes made to a revisioncontrolled collection of files.

- Date Difference: Calculates the number of days between two dates.
- Notional Distance

NAME	first_name		last_name		id		error_first_name		error_last_nar	me	error_id	
TYPE	string		string		integer		string		string		integer	
FORMAT												
ONFIDENCE SCO					DEFAULT					~		
HS VARIABLES	θ	RHS VARIABLES	θ	ALGORITHM	TO USE 0	WEIG	SHTS 6	OUTPUT COLUMN	0			
						WEIGH	01110	OUTPOT COLUMN	0			
īrst_name		error_first_name	• •	levenshtein	*			levenshteinScoreFirs				
		error_first_name error_last_name		levenshtein levenshtein	~	0.5			tName			
ast_name						0.5 0.5		levenshteinScoreFirs	tName			
ast_name			· ·		~	0.5		levenshteinScoreFirs	tName			
īrst_name ast_name d error_first_name error_last_name			~		~	0.5	-	levenshteinScoreFirs	tName			

Dedup Processor Configuration

Dedup Processor Output

first_name	last_name	id	error_first_name	error_last_name	error_id	levenshteinScoreFirstName	levenshteinScorelastName	confidenceScore
StringType	StringType	IntegerType	StringType	StringType	IntegerType	DoubleType	DoubleType	DoubleType
Nathan	Cordova	848	Nathan	Cordova	848	1.0	1.0	1.0
Nathan	Cordova	848	Jesse	Martinez	411	0.0	0.1	0.05
Luke	Quick	157	eMrecdes	hCnarro	965	0.1	0.0	0.05
Jodi	Baldino	980	Raegan	Mcneely	117	0.0	0.0	0.0
Guled	Shen	225	Kaitlyn	Young	878	0.1	0.2	0.15
Guled	Shen	225	Kiana	Cyaagdng	575	0.0	0.1	0.05
Guled	Shen	225	Lishrka	Reyna	141	0.0	0.2	0.1
Guled	Shen	225	jqdy	Plenty Wolf	556	0.0	0.2	0.1
Viarlenny	Picazzo Banuelos	598	Lsis	Rodriguez	806	0.0	0.2	0.1
Viarlenny	Picazzo Banuelos	598	Tilane	Thompson	976	0.3	0.1	0.2
Viarlenny	Picazzo Banuelos	598	Andrew	Lattimer	865	0.2	0.1	0.15

Prints the Results

It prints the first few records onto the screen.

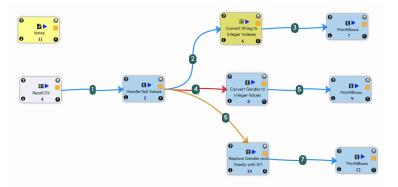
Handling Null Values

Workflow

This example removes null values from the input dataset.

Below is the workflow. It does the following:

- Reads data from a CSV file.
- Replaces null values in certain columns with constant values.
- Converts certain columns to 0/1 based on their value. It does it in 3 different ways.
- Using StringIndexer Processor
- Using CaseWhen Processor
- Using FindAndReplaceUsingRegex Processor



Reading from CSV File

Processor Configuration

Processor Output

Replacing null values

It reads in the CSV file data-with-nulls.csv.

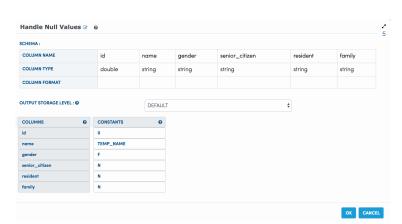
It replaces null values in certain columns with user defined constant values.

OUTPUT STORAGE LEVEL : 🛛	DEFAULT		\$	
ATH *: 0	/user/ec2-user/data/data-with-	-nulls.	csv	BROWSE HDFS
				VIEW FILE
EPARATOR : 0				
IEADER : 😧	true		\$	
ROP MALFORMED : 0	false		\$	
CHEMA COLUMNS : O REFRESH SCHE	MA 0			
OLUMN NAMES FOR THE CSV O	COLUMN TYPES FOR THE CSV O		COLUMN FORMATS FOR THE	CSV 🛛
id	DOUBLE	\$	format	•
name	STRING	\$	format	•
gender	STRING	\$	format	•
		\$	format	•

ReadCSV

Executing Node fire.nodes.dataset.NodeDatasetCSV : 4 Nov 11, 2018 1:16:51 PM

id	name	gender	senior_citizen	resident	family
DoubleType	StringType	StringType	StringType	StringType	StringType
1.0	ABC	F	Y	Y	N
2.0	DEF	м	N	N	
3.0	GHR	м	Y	Y	Y
4.0	JKL	F	N	Y	N
5.0	RIT	м	Y	Y	
NaN		F	Y	Y	N
6.0	PQR		Y		Y
NaN	ORT		Y	Y	N



ОК

Processor Configuration

Processor Output

id	name	gender	senior_citizen	resident	family
DoubleType	StringType	StringType	StringType	StringType	StringType
1.0	ABC	F	Y	Y	Ν
2.0	DEF	м	Ν	N	Ν
3.0	GHR	м	Y	Y	Y
4.0	JKL	F	Ν	Y	Ν
5.0	RIT	м	Y	Y	Ν
0.0	TEMP_NAME	F	Y	Y	Ν
6.0	PQR	F	Y	N	Y
0.0	ORT	F	Y	Y	Ν

Converting to 0/1 using StringIndexer

It converts strings like Y/N to 0/1 for the specified columns using the StringIndexer Processor.

Processor Configuration

Convert String to Integ	ger Indexes 🕜	0					
SCHEMA :							
COLUMN NAME	id	name	gender	senior_citizen	n	esident	family
COLUMN TYPE	double	string	string	string	s	tring	string
COLUMN FORMAT							
DUTPUT STORAGE LEVEL : 😡		DEFAULT			\$		
ANDLE INVALID : 🛛		skip			\$		
INPUT COLUMNS	OUTPUT COLUMNS	Θ					
id							
name							
gender	gender_index						
senior_citizen	senior_citizen_inde	x					
resident	resident_index						
family	family_index						

Processor Output

Converting to 0/1 using CaseWhen

It converts strings like Y/N to 0/1 for the specified columns using the CaseWhen Processor.

Processor Configuration

id	name	gender	senior_citizen	resident	family	gender_index	senior_citizen_index	resident_index	family_index
DoubleType	StringType	StringType	StringType	StringType	StringType	DoubleType	DoubleType	DoubleType	DoubleType
1.0	ABC	F	Y	Y	N	0.0	0.0	0.0	0.0
2.0	DEF	м	N	N	N	1.0	1.0	1.0	0.0
3.0	GHR	м	Y	Y	Y	1.0	0.0	0.0	1.0
4.0	JKL	F	N	Y	N	0.0	1.0	0.0	0.0
5.0	RIT	м	Y	Y	N	1.0	0.0	0.0	0.0
0.0	TEMP_NAME	F	Y	Y	N	0.0	0.0	0.0	0.0
6.0	PQR	F	Y	N	Y	0.0	0.0	1.0	1.0
0.0	ORT	F	Y	Y	N	0.0	0.0	0.0	0.0

Convert Gender to Inte	eger Values	Ø				
SCHEMA :						
COLUMN NAME	id	name	gender	senior_citizen	resident	family
COLUMN TYPE	double	string	string	string	string	string
COLUMN FORMAT						
OUTPUT STORAGE LEVEL : 😡		DEFAULT			\$	
OUTPUT COLUMN NAME : 🕖		gender_	new			
KEY VALUE ARRAY : 🛛 🕤						
VHEN CONDITION @			VALUE 😡			
gender == 'F'			1			•
gender == 'M'			0			•
ELSE : 🛛		0				
						OK CANCE

Processor Output

id	name	gender	senior_citizen	resident	family	gender_new
DoubleType	StringType	StringType	StringType	StringType	StringType	IntegerType
1.0	ABC	F	Y	Y	Ν	1
2.0	DEF	м	Ν	Ν	Ν	0
3.0	GHR	м	Y	Y	Y	0
4.0	JKL	F	N	Y	Ν	1
5.0	RIT	м	Y	Y	Ν	0
0.0	TEMP_NAME	F	Y	Y	Ν	1
6.0	PQR	F	Y	Ν	Y	1
0.0	ORT	F	Y	Y	N	1

Converting to 0/1 using FindAndReplaceUsingRegex

It converts strings like Y/N to 0/1 for the specified columns using the FindAn-dReplaceUsingRegex Processor.

Processor Configuration

SCHEMA :							
COLUMN NAME	ic	I	name	gender	senior_citizen	resident	family
COLUMN TYPE	d	ouble	string	string	string	string	string
COLUMN FORMAT							
OUTPUT STORAGE LEVEL :	0		DEFAULT			\$	
VARIABLES LIST :							
NPUT COLUMNS @		FIND 😧			REPLACE @		
gender	\$	Μ			0		•
gender	\$	F			1		•
family	\$	Ν			0		•
family	\$	Y			1		
							OK CANCE
id	name		senior_	citizen	resident	gender	OK CANCE family
id DoubleType	name StringTy	лре	senior_ String		resident StringType	gender StringType	
		гре				-	family
DoubleType	StringTy	/pe	String		StringType	StringType	family StringType
DoubleType	StringTy ABC	/pe	String Y		StringType Y	StringType	family StringType 0

Remove Duplicate Rows

0.0

6.0

0.0

TEMP_NAME

PQR

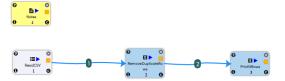
ORT

This workflow reads CSV file. It then removes duplicate rows from the original CSV file and prints the results.

Workflow

Below is the workflow. It does the following:

- Reads data from a CSV file.
- It then removes duplicate rows from the original CSV file.
- Prints the results.



Reading from CSV file

It reads CSV file.

Processor Configuration

DUTPUT STORAGE LEVEL : O	DEFAULT		
	DEMOCI	•	
ин*: 0	data/duplicate.csv	BROWSEHDFS	
		VIEW FILE	
PARATOR : O			
IEADER : O	true	•	
ROP MALFORMED : 0	folse	•	
CHEMA COLUMNS : O REFRESH SCHEMA	2		
DLUMN NAMES FOR THE CSV O	COLUMN TYPES FOR THE CSV 🕹	COLUMN FORMATS FOR THE CSV @	
:1	INTEGER	format	
	INTEGER	• format	
2			

Processor Output

ecutina Node fire.nodes.dataset.No	deDatasetCSV : 1 Nov 13, 2018 1:55:36 AM		
Row Values			
1	c2	c3	
ntegerType	IntegerType	IntegerType	
	2	3	
1	5	6	
	2	3	
,	8	9	
1	5	6	
l i i i i i i i i i i i i i i i i i i i	6	9	
	3	5	

Remove Duplicate Rows

It removes Duplicate Rows available.

Processor Configuration

Processor Output

Prints the Results

It prints the results after Removing Duplicate Rows.

UMN NAME	c1	c2	c3	
DLUMN TYPE	integer	integer	integer	
LUMN FORMAT				
PUT STORAGE LEVEL : O	DEFAULT		•	
ER : 0				
ER : O	first		•	
UMNS : O	c1 : integer		·	
	c2 : integer c3 : integer			
noveDuplicateRows				OK C
noveDuplicateRows xecuting Node fire nodes.etl.NodeRemoveDuy Input Schema	plicateRows : 2 Nov 13, 2018 2:03:58 AM			OK C
xecuting Node fire.nodes.etl.NodeRemoveDup	plicateRows : 2 Nov 13, 2018 2:03:58 AM			oc d
xecuting Node fire.nodes.ett.NodeRemoveDup	plicateRows : 2 Nov 13, 2018 2:03:58 AM	a		OK C
xecuting Node five.nodes.ett.NodeRemoveDu Input Schema Row Yolues			ger/pe	ox o
xecuting Node fire nodes.etl.NodeRemoveDu Input Schema Row Values	a		perfype	ot c
xecuting Node fire nodes.atl.NodeRemoveDu Input Schema Row Yolues at IntegerType	<2 IntegerType	Inte	geriype	ok c

Processor Configuration

CHEMA :						
COLUMN NAME		c1	c2		c3	
COLUMN TYPE		integer	integer		integer	
COLUMN FORMAT						
UTPUT STORAGE LEVEL : O ITLE : UM ROWS TO PRINT : O	DEF/ Row	AULT Values		v		

Processor Output

Removing Special Characters

Workflow

This workflow reads in a dataset. It then removes the special characters from columns of the original dataset and prints the results.

Below is the workflow. It does the following:

- It reads the CSV and creates a DataFrame.
- It find and replaces the special characters with empty space in the columns

PrintNRows			2
Executing Node fire.nodes.util.No	dePrintFirstNRows : 3 Nov 13, 2018 2:08:55 AM		
Input Schema			
Row Values			
cl	c2	0	
IntegerType	IntegerType	IntegerType	
1	2	3	
4	5	6	
7	8	9	
Row Values			
			OK
			-
		Create	new DataFrame containing the rows
			tisfy the given condition (i.e. removes
		the rov	vs with empty space)
		• Print t	ne specified number of records in the
		DataFr	ame after execution of workflow
		Duturi	une uner execution of worknow
	This workflow reads in a dataset.	14 4h	
	special characters from columns		
	dataset and prints the results.		
		4 0	3 8 .
bad_data_string.	To Remove Any	RowFilter -	PrintNRows
csv	Special character In data	Remove the rows with empty space	
	X	~	
	It finds and replaces the special characters with empty	Creates new DataFrame containing the required rows	
	space in the columns		

Reading from Dataset

DatasetCSV processor reads in the input Dataset file and creates DataFrame.

Processor Configuration

JTPUT STORAGE LEVEL : O	DEFAULT		~	
TH * : 🖌	/tmp/data/bad_data_string.csv		BROWSE HDFS	VIEW FILE
PARATOR : 0				
ADER : 😡	true		~	
ROP MALFORMED : 0	folse		~	
CHEMA COLUMNS : O REFRESH SCHEMA O				
COLUMN NAMES FOR THE CSV O	COLUMN TYPES FOR THE CSV O	COLUMN FORMATS FO	R THE CSV O	
c0_str	STRING	▼ format		•
	STRING	✓ format		

adCSV			· · · · ·
Executing Node fire.nodes.dataset.Node	eDatasetCSV : 1 : Jan 5, 2021 8:43:04 AM		
ReadCSV			
Input Path: hdfs://sparkflows-c	demo.c.sparkflows-168107.internal:8020/tmp/data/bad_data_	string.csv	
Row Values			
ow Values			
c0_str	c1_str	c2_date	
StringType	StringType	StringType	
abc def	abc	04/25/2013	
abc def	abc	04/25/2013	

To Remove Any Special character in data

FindAndReplaceUsingRegex processor find and replaces the special characters with empty space in the columns

Processor Configuration

To Rem	ove Any Special character In	data 🖉 🔍 NodeField	AndReplaceUsingRegex		
NAME	c0_str		cl_str	c2_date	
TYPE	string		string	string	
FORMAT					
ND :		\W+			
EPLACE : 0					
					OK CANO

Processor Output

To Remove Any Special character I	n data		2 ×
Executing Node fire.nodes.etLNodeFindAndRe	placeUsingRegex : 2 : Jan 5, 2021 8:44:22 AM		
Input Schema			
Row Values			
Row Values			
c1_str	c2_date	c0_str	
StringType	StringType	StringType	
abc	04/25/2013	abcdef	
obc	04/25/2013	123abc	
			OK

RowFilter - Remove the rows with empty space

RowFilter processor creates new DataFrame containing the rows that satisfy

the condition provided (For example : Removes the rows with empty spaces as shown below)

Processor Configuration

	c0_str	c1_str	c2_date	
TYPE	string	string	string	
FORMAT				
UTPUT STORA	GE LEVEL : O	DEFAULT	v	
ONDITIONAL E	XPRESSION * : 0			
1 c0_str				

Processor Output

RowFilter - Remove the rows w	vith empty space		~ >
Executing Node fire.nodes.etl.NodeRow	Filter : 4 : Jan 5, 2021 8:44:43 AM		
Input Schema			
Raw Values			
Row Values			
c1_str	c2_date	c0_str	
StringType	StringType	StringType	
abc	04/25/2013	abcdef	
abc	04/25/2013	123abc	
			OK

Prints the Results

Rename Columns

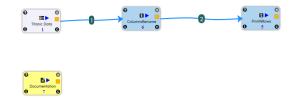
Workflow

It prints the first few records onto the screen.

This workflow reads in a dataset. It then renames columns from the original dataset and prints the results.

Below is the workflow. It does the following:

- Reads data from a dataset.
- It then renames columns from the original dataset.
- Prints the results.



Reading from Dataset

It reads Dataset file.

Processor Configuration

NodeDatasetStructured			
DUTPUT STORAGE LEVEL : 0	DEFAULT	•	
DATASET : 😡	Titanic Data	*	

Processor Output

	re metriodes.c	iataset.NodeL	latasetStructured : 1 Nov 13, 2018 2:3	IU:49 AM							
Row Yolues											
Possengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		s
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	с
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	s
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		s
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q

Rename Columns

It renames columns we want.

Processor Configuration

Processor Output

Prints the Results

It prints the results after Renaming Columns.

COLUMN NAME	Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
OLUMIN TYPE	integer	integer	integer	string	string	string	integer	integer	string	double	string	string
COLUMN FORMAT												
JTPUT STORAGE LEVEL : O	-			AULT					۲			
CURRENT COLUMN NAMES	COLUMINS NE PId	W NAME 0										
iunvived	WhetherSurvi	ved										
Poless												
fame	FullName											
5ex												
Age												
SibSp Parch												
Farch												
are												
Fare Cobin												

			Rename : 6 Nov 13, 2018 2:37:39 A/								
Input Scher	na										
Row Values											
٩d	WhetherSurvived	Pclass	FullName	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType
	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 211/1	7.25		s
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	с
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
t	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	s
5	0	а	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
5	0	3	Moran, Mr. James	mole		0	0	330677	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	mole	54	0	0	17463	51.8625	E46	s
3	0	3	Palsson, Master: Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		s
0	1	2	Nosser, Mrs. Nicholos (Adele Achem)	female	14	1	0	237736	30.0708		с

HEMA :												
COLUMN NAME	Pld	WhetherSurvived	Pclass	FullName	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
COLUMN TYPE	integer	integer	integer	string	string	string	integer	integer	string	double	string	string
COLUMN FORMAT												
UTPUT STORAGE LEVEL	.:0		DEFAU	LT					•			
TLE :			Row Vo	Row Values								
UM ROWS TO PRINT : 6			3									
			3									

Processor Output

ecuning no.	de fire.nodes.util	.NodePrintFirs	tNRows : 5 Nov 13, 2018 2:45:29 A	M							
Input Scher											
inpui scher	na										
Row Values											
Row values											
ы	WhetherSurvived	Pelass	FullNome	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emborked
tegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType
	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		s
	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	С
	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		s

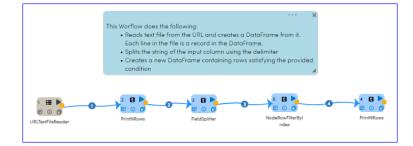
REST - CSV Reader & Parse

This workflow reads in a dataset from URL. It then parses the dataset and prints the results.

Workflow

Below is the workflow. It does the following:

- Reads data from the URL and creates a DataFrame
- Prints few records
- Splits the string of the input column using the delimiter
- Creates a new DataFrame containing rows satisfying the provided condition
- Prints the result



Reading from URL

DatasetURLTextFileReader processor uses the passed URL to download the data and create the DataFrame.

Processor Configuration

1 URLTextFileReader	NodeDatasetUrtTextFileRead	dor	2.5
OUTPUT STORAGE LEVEL : O		DEFAULT http://winterolympicsmedols.com/medols.cov]
			OK CANCEL

Processor Output

LTextFileReader	· · · · · · · · · · · · · · · · · · ·
ow Values	
line	
StringType	
Year,City,Sport,Discipline,NOC,Event,Event gender,Medal	
1924,Chamonix,Skating,Figure skating,AUT,individual,M,Silver	
1924,Chamonix,Skating,Figure skating,AUT,individual,W,Gold	
1924,Chamonix,Skating,Figure skating,AUT,pairs,X,Gold	
1924,Chamonix,Bobsleigh,Bobsleigh,BEL,four-man,M,Bronze	
1924,Chamonix,Ice Hockey,Ice Hockey,CAN,ice hockey,M,Gold	
1924, Chamonix, Biathlon, Biathlon, FIN, military patrol, M, Silver	
1924,Chamonix,Skating,Figure skating,FIN, pairs,X,Silver	
1924,Chamonix,Skating,Speed skating,FIN,10000m,M,Gold	
1924,Chamonik,Skating,Speed skating,FIN,10000m,M,Silver	
1924 Champaix Skating Speed skating FIN 1500m M.Gold	

Prints the Records

It prints the first few records onto the screen.

Parsing the DataFrame

FieldSplitter processor parses and creates new DataFrame by splitting the string of the input column using the delimiter as shown below:

Processor Configuration

3 FieldSpl	NodeFieldSplitter			1
NAME	line			
TYPE	string			
FORMAT				
OUTPUT STORA	AGE LEVEL : O	DEFAULT	~	
NPUT COLUMN	1:0	line : string	~	
UTPUT COLUN	MNS · D			

Processor Output

ow Values								
line	Year	City	Sport	Discipline	NOC	Event	Event gender	Medal
StringType	StringType	StringType	StringType	StringType	StringType	StringType	StringType	StringType
Year, City, Sport, Discipline, NOC, Event, Event gender, Medal	Year	City	Sport	Discipline	NOC	Event	Event gender	Medal
1924.Chamonix.Skating.Figure skating.AUT.individuaUM.Silver	1924	Chamonix	Skating	Figure skating	AUT	individual	м	Silver
1924, Chamonix, Skating, Figure skating, AUT, Individual (W, Gold	192.4	Chamanix	Skating	Figure skating	AUT	individual	w	Gold
1924,Chamonix,Skating,Figure skating,AUT,pairs,X,Gald	1924	Chamonix	Skoting	Figure skoting	AUT	poirs	x	Gold
1924, Chamonis, Bobsleigh, Bobsleigh, SEL, four-man, M, Bronze	122.4	Chamonix	Bobsleigh	Bobsleigh	BEL	four-man	м	Bronze
1924, Chamoniu Joe Hockey Joe Hockey, CANJoe hockey, M.Gold	192.4	Chamanix	Ice Hockey	Ice Hockey	CAN	ice hockey	м	Gold
1924, Chamonis, Blathlon, Blathlon, FIN, military potrol, M, Silver	122.4	Chamonix	Biathlon	Biothion	FIN	military patrol	м	Silver
1924, Chamonix, Skating, Figure skating, FIN, pairs, X, Silver	192.4	Chamanix	Skating	Figure skating	FN	pairs	х	Silver
1924.Chamonix.Skating.Speed skating.FIN.10000m.M.Gold	192.4	Chamonix	Skoting	Speed skoting	FIN	10000m	м	Gold
1924,Chamonix,Skating,Speed skating,FIN,10000m,M,Silver	122.4	Chamonix	Skating	Speed skating	FN .	10000m	м	Silver

Row Filter by Index

RowFilterByIndex processor creates a new DataFrame containing required rows as shown below:

Processor Configuration

Processor Output

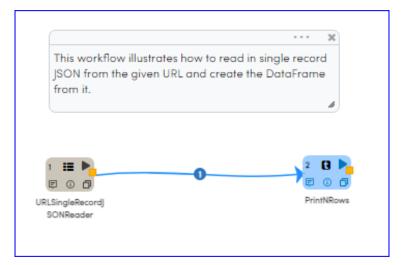
	line	Year	City	Sport	Discipline	NO	c	Event	Event gender		Medal	1.1
TYPE	string	string	string	string	string	strie	ng	string	string		string	
FORMAT												
TPUT STOR	AGE LEVEL : O			DEFAULT					~			
DES: 0												
1.0												
DEXESRANGE	E:0											
3												
											ox	CNIC
											ax	CANC
											CX	
odeRowF	FilterByIndex										CX	сию 2
		abruchterfluinder - 5 - De	< 91 2020 IP 001-98 AM								CK	
		aBowFilterByIndex : 5 : De	c 31, 2020 11:08:18 AM								CK	
Executing	Node fire.nodes.ett.Node	sTowFiterSylndes : 5 : De	c 31, 2020 11:08:18 AM								CC	
Executing	Node fire.nodes.ett.Node	aBowTiterSylndes : 5 : De	c 31, 2020 11 08:18 AM								X	
Executing Input Sche	Node fire nodes et Node	aBox773erSylnder : 5 : De	c 31, 2000 11:08:18 AM									
Executing	Node fire nodes et Node	aRov/FilterSylinder ; S : De	e 31, 2020 1108/88 AM									
Executing Input Sche	Node Hreunodes et Node	aBourFilerSylnder (5 : De	e 31, 2020 1108/18 AM									
Executing Input Sche Row Voluer	Node Hreunodes et Node	eRov/PiterSylinder S : De	e 31, 2020 1108/18 AM	Yeer	City	Sport	Discipline	NOC	Event	Event gender	Medal	
Executing Input Sche Row Volues	Node Treunodecet Mode	=Buy/TheOylode() 5 De	e 31, 2020 1108 H AM	Year Stringtype		Sport Strigtype	Discipline	NOC	Event	Event gender Stringtige		
Executing Input Sche Row Volues Row Volues Line StringType	Node Treunodecet Mode		c 31 2000 HOR H AM		StringType						Medal	
Executing I Input Sche I Row Volues Line StringType	Node Tre-nodec at Node		c 31, 2020 1108 III AM	StringType	StringType	StringType	StringType	StringType	StringType	StringType	<u>Medal</u> Stragiyae	

Prints the Results

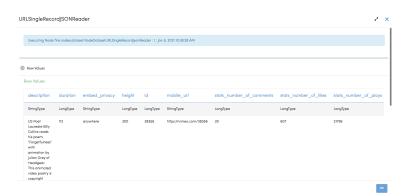
It prints the result onto the screen.

REST Read And Parse JSON	
	This workflow reads in single record JSON from the given URL. It then parses the dataset and prints the results.
Workflow	
	Below is the workflow that shows:
	• How to read in single record JSON from the given URL and create the DataFrame from it
	• Prints the result
Reading from URL And Parsing	
	DatasetURLSingleRecordJSONReader processor uses the passed URL to download single record JSON, parse the dataset and create the DataFrame.
Processor Configuration	

Chapter 12. Tutorials



UTPUT STORAGE LEVEL : O	DEFAULT		~
RL:0	http://vimeo.com/api/v2/video/	3356.json	
HEMA COLUMNS : O REFRESH SCHEMA O			
COLUMN NAMES O	COLUMIN TYPES O	COLUMN FORMATS O	
description	STRING	✓ format	•
duration	LONG	✓ format	•
embed_privacy	STRING	✓ format	•
height	LONG	← format	•
id	LONG	✓ format	•
and the set	OTDING.	and the second	



Processor Output

Prints the Results

It prints the result onto the screen.

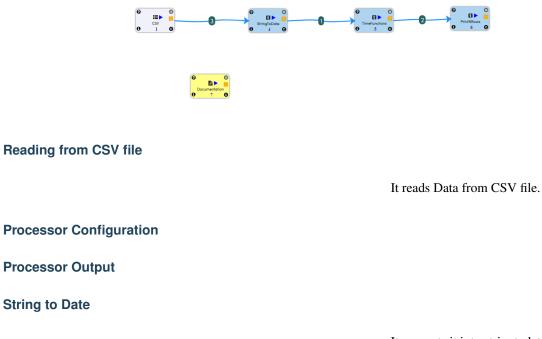
String To Date Timefunctions

Workflow

This workflow reads a CSV file. It then converts it into stringtodate and then to time-functions and prints the results.

Below is the workflow. It does the following:

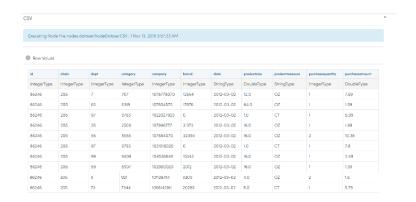
- Reads a CSV file.
- It then converts it into stringtodate using stringtodate Node.
- Convert it into timefunctions using timefunctions Node.
- Prints the results.



It converts it into stringtodate using stringtodate Node.

OK

CSV 2 0 NodeDatasetCSV						
DUTPUT STORAGE LEVEL : 😜	DEFAULT					
WITH *: O	data/transaction.csv	data/transaction.csv				
EPARATOR : 0			VIEW FILE			
HEADER : O	frue					
DROP MALFORMED : 0	folse					
OLUMN NAMES FOR THE CSV 🖗	COLUMN TYPES FOR THE CSV O	COLUMN FORMATS FO	R THE CSV O			
id	INTEGER	* format	•			
chain	INTEGER	GER • format				
dept	INTEGER	* format	•			
calegory	INTEGER	• format	•			
company	INTEGER	• format	•			
brand	INTEGER	• format	•			
date	STRING	* format	•			
productsize	DOUBLE	• format	•			
producines		• format	•			
productmeasure	81HING	 format 				
	STRING INTEGER	format				



StringToDate 🛛	0										
CHEMA :											
COLUMN NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount
COLUMN TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double
COLUMN FORMAT											
NPUT COLUMN NAME :	TPUT STORAGE LEVEL : O							*			
IPUT COLUMN FORMAT	UUMN FORMAT: O yyyy-MM-dd										
UTPUT COLUMN TYPE					date_type TIMESTAMP					•	

Processor Output

xecuting Not	de fire.nodes.e	tl.NodeString1	oDate : 4 Nov	13, 2018 4:02:5	57 AM						
Input Scher											
Input scher	na										
Row Values											
а	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount	date_type
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType	TimestampType
86246	205	7	707	1078778070	12564	2012-03- 02	12.0	OZ	1	7.59	2012-03-02 00:00:00.0
86246	205	63	6319	107654575	17876	2012-03- 02	64.0	OZ	1	1.59	2012-03-02 00:00:00.0
86246	205	97	9753	1022027929	0	2012-03- 02	1.0	CT	1	5.99	2012-03-02 00:00:00.0
86246	205	25	2509	107996777	31373	2012-03- 02	16.0	OZ	1	1.99	2012-03-02 00:00:00.0
86246	205	55	5555	107684070	32094	2012-03- 02	16.0	OZ	2	10.38	2012-03-02 00:00:00.0
86246	205	97	9753	1021015020	0	2012-03- 02	1.0	CT	1	7.8	2012-03-02 00:00:00.0
86246	205	99	9909	104538848	15343	2012-03- 02	16.0	oz	1	2.49	2012-03-02 00:00:00.0
86246	205	59	5907	102900020	2012	2012-03- 02	16.0	07	1	1.39	2012-03-02 00:00:00.0
86246	205	9	921	101128-414	9209	2012-03- 02	4.0	OZ	2	1.5	2012-03-02 00:00:00.0
86246	205	73	7344	1068142161	20285	2012-03- 02	8.0	CT	1	5.79	2012-03-02 00:00:00.0

Time Functions

It converts it into timefunctions using time-functions Node.

Processor Configuration

CHEMA :												
COLUMN NAME	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount	date_type
COLUMN TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double	timestamp
COLUMN FORMAT												
UTPUT STORAGE LEI						pe : timest				×		
ME FUNCTIONS : 0					YEAR N	NONTH® DA	YOFMONT	HI DAYOFYEAR	WEEKOFYEAR DAYOF	VEEK		

Processor Output

Prints the Results

It prints the results after using string to date timefunctions.

Input Scher	ma											
Row Values	1											
Id	chain	dept	category	company	brend	date	productsize	productmeasure	purchasequantity	purchaseamount	date_type	date_type
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType	TimestampType	IntegerTy
86246	205	7	707	1078778070	12564	2012-03- 02	12.0	oz	1	7.59	2012+03-02 00:00:00.0	2012
86246	205	63	6319	107654575	17876	2012-03- 02	64.0	oz	1	1.59	2012-03-02 00:00:00.0	2012
86246	205	97	9753	1022027929	0	2012-03- 02	1.0	СТ	1	5.99	2012-03-02 00:00:00.0	2012
86246	205	25	2509	107996777	31373	2012-03- 02	16.0	oz	1	1.99	2012-03-02 00:00:00.0	2012
86246	205	55	5555	107684070	32094	2012-03- 02	16.0	oz	2	10.38	2012-03-02 00:00:00.0	2012
86246	205	97	9753	1021015020	0	2012-03- 02	1.0	ст	1	7.5	2012+03-02 00:00:00.0	2012
86246	205	99	9909	104538848	15343	2012-03- 02	16.0	oz	1	2.49	2012-03-02 00:00:00.0	2012
86246	205	59	5807	102900020	2012	2012-03- 02	16.0	OZ	1	1.39	2012-03-02 00:00:00.0	2012
86246	205	9	921	101128414	9209	2012-03- 02	4.0	oz	2	1.5	2012-03-02 00:00:00.0	2012
86246	205	73	7344	1068142161	20285	2012-03- 02	8.0	СТ	1	5.79	2012-03-02 00:00:00.0	2012

CHEMA :														
COLUMN	id	chain	dept	category	company	brand	date	productsize	productmeasure	purchasequantity	purchaseamount	date_type	date_type_year	date_
COLUMN TYPE	integer	integer	integer	integer	integer	integer	string	double	string	integer	double	timestamp	integer	intege
OLUMN														
UTPUT ST	ORAGE LEV	EL : O				DEF	AULT				٠			
TLE :						Row	Values							
JM ROWS	TO PRINT :	0				10								

Processor Output

Date-Time Field Extract

Workflow

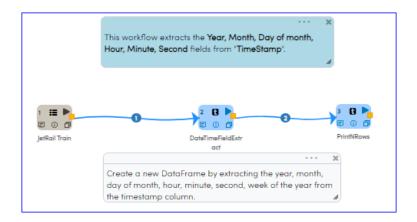
Below is the workflow. It does the following:

- Reads data from a dataset.
- It creates a new DataFrame by extracting Year, Month, Day of month, Hour, Minute, Second fields from "TimeStamp"
- Prints the results.

Reading from Dataset

It reads in the input Dataset File.

ixecuting Not	de fire.nodes.u	itil.NodePrintF	irstNRows : 6 M	iov 13, 2018 4:	5:17 AM							
Input Scher	πα											
Row Values												
ia -	chain	dept	colegory	company	brand	date	productaize	productmeasure	purchasequantity	purchaseemount	date_type	date_type_
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	DoubleType	StringType	IntegerType	DoubleType	TimestampType	IntegerTy
86246	205	7	707	1078778070	12564	2012-03- 02	12.0	OZ	1	7.59	2012-03-02 00:00:00.0	2012
86246	205	63	6319	107654575	17876	2012-03- 02	64.0	oz	1	1.69	2012-03-02 00:00:00.0	2012
86246	205	97	9753	1022027929	0	2012-03- 02	1.0	CT	1	5.99	2012-03-02 00:00:00.0	2012
86246	205	25	2509	107996777	31373	2012-03- 02	16.0	OZ	1	1.99	2012-03-02 00:00:00.0	2012
86246	205	55	5555	107684070	32094	2012-03- 02	16.0	oz	2	10.38	2012-03-02 00:00:00.0	2012
86246	205	97	9753	1021015020	0	2012-03- 02	1.0	СТ	1	7.8	2012-03-02 00:00:00.0	2012
86246	205	99	9909	104538848	15343	2012-03- 02	16.0	oz	1	2.49	2012-03-02 00:00:00.0	2012
86246	205	59	5907	102900020	2012	2012-03- 02	16.0	07	1	1.39	2012-03-02 00:00:00.0	2012
86246	205	9	921	101128414	9209	2012-03- 02	4.0	oz	2	1.5	2012-03-02 00:00:00.0	2012
86246	205	73	7344	1068142161	20285	2012-03- 02	8.0	СТ	1	5.79	2012-03-02 00:00:00.0	2012





Processor Output

latasetStructured		2.3
Executing Node fire nodes dataset. NodeDatasetStructured : 1 ; Jan 5, 2021 10:20:08 AM		
Row Values		
Datetime	Count	
TimestampType	IntegerType	
2012-08-25 00:00:00.0	8	
2012-08-25 01:00:00.0	2	
2012-08-25 02:00:00.0	6	
2012-08-25 03:00:00.0	2	

Date-Time Field Extract

It creates a new DataFrame by extracting the year, month, day of month, hour, minute, second, week of the year from the timestamp column.

Processor Configuration

me amp A-yyyy Hitzmm O	DEFAU(T Dotefring : filmestomp	Count Integer	
V-yyyy HHomm			
		~	
Ð		~	
	Datetime : timestamp		
		~	
	true	~	
	true	~	
: 0	frue	~	
	true	~	
	true	~	
	true	~	
)	frue	~	
	-	but tou tou tou tou tou tou tou	true v true v true v true v

Processor Output

Do	teTimeFieldE	xtract							2 ×
	Executing Node fi	ire.nodes.etl.Nod	leDateTimeFieldExtract	2 : Jan 5, 2021 10:20:53 Al	и				
0	Input Schema								
0	Row Values								
	low Values								
	Datetime TimestampType	Count	Datetime_year	Datetime_month	Datetime_dayofmonth	Datetime_hour	Datetime_minute	Datetime_second	Datetime_weekofyea
	2012-08-25 00:00:00.0	8	2012	8	25	0	0	0	34
	2012-08-25 01:00:00.0	2	2012	8	25	1	0	0	34
	2012-08-25	A	2012	A	95	2	0	n	74 CX

Prints the Results

Concat Columns Workflow

Reading from HDFS File

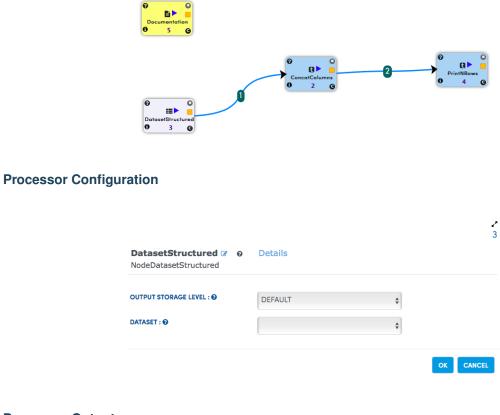
It prints the first few records onto the screen.

This example concats columns in the input dataset with the specified separator.

Below is the workflow. It does the following:

- Reads data from file present on HDFS.
- Concats the specified columns with specified separator.

It reads data from a file present on HDFS.



Processor Output

		Januagen House	Juluseisiluciure	30 : 5 NOV 14,	2018 12:14:20	PM					
Row Values											
Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringTyp
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	с
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		s

Concating columns

It concats the specified columns in cofiguration with the specified separator.

Processor Configuration

We need to provide all the desired columns to be concatenated without any separator or space, like NameSexAge etc. Columns

would get concatenated in same order defined in configuration like Name then Sex then Age.

HEMA :												
COLUMN NAME	Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
OLUMN TYPE	integer	integer	integer	string	string	string	integer	integer	string	double	string	string
COLUMN FORMAT												
JUJMNS∶€			Surviv Pclass Name Sex : s Age : : Age : : SibSp Parch Ticket Fare : Cabin	rgerid : integer : integer : string tring tring : integer : integer : string double : string rked : string	-							
NCATENATED COLUM	N NAME : O		Name	SexAge								

Processor Output

rld	Survived	Pelass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	NameSexAge
pe	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType	StringType
	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S	Braund, Mr. Owen Harris male 22
	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	с	Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 3
	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S	Heikkinen, Miss. Laina female 26
	1	1	Futrelle, Mrs. Jacques Heath	female	35	1	0	113803	53.1	C123	S	Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35

Joining Multiple Datasets

Fire Insights allows you to quickly do complex data preparation and ETL on Big Data.

Fire Insights has a number of features for enabling it including:

- Reading data from multiple sources
- Cleaning data
- oins, GroupBy, Cube, SQL etc. to transform data
- Writing results to various sinks

Fire Insights also gives you detailed control over your Spark jobs with Repartition, Coalesce, Cache etc.

Overview

Datasets

Workflow

Data Parsing and Cleaning

In this example, we start with 5 datasets, read them in & understand their schema in the process, perform data cleaning and then apply appropriate aggregations and joins.

The cleansed and tranformed datasets are written to HDFS as CSV files. These dataset can as well we written as Parquet, Avro, JSON, XML files or to HIVE/Relational tables as needed.

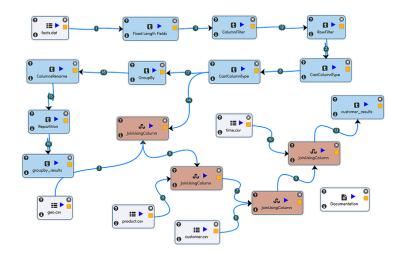
- facts.dat : Contains fixed length records of products sold to customers
- geo.csv : Contains mapping of geo ids to geo names
- product.csv : Contains mapping of product ids to product names
- customer.csv : Contains mapping of customer ids to customer names
- time.csv : Contains mapping of various time interval ids to corresponding names

The workflow achieves the following tasks:

- Parses the facts data and performs various cleanup operations on it.
- Performs groupby with aggregations operations and saves it to a file.
- Joins the fact data with various dimensions to create a large table and saves it to a CSV file.

The workflow is shown below:

While the various dimension data is available as CSV files, the fact data is in fixed field size format.



Each record has a fixed number of characters. In each record each field consists of fixed number of characters. The steps for data parsing and cleaning are as follows:

- Read in the fixed length record
- Filter out invalid records
- Cast some columns to numeric values

Group By and Aggregates

Joins with various Dimension Data

Time Function

The fact data is then joined with various dimension data. These include:

The data is then aggregated and counted and averages are calculated. It is then saved as

- Geo
- Product
- Customer

CSV file.

• Time

The final dataset is saved as CSV file.

There are many instances when you want to do time-series analysis. Fire Insights provides Date-Time features with TimeFunctions operator.

Creating additional features from the timestamp column helps you to know more about the data and run modeling algorithms on them. Fire Insights has NodeTimeFunctions for creating these time series features.

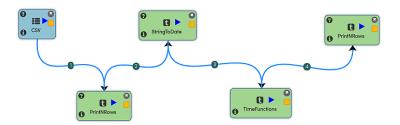
Dataset

Let us take a Transaction Dataset which is in CSV format on HDFS. The dataset has a "DATE" column.

ID	CHAIN	DEPT	CATEGORY	COMPANY	BRAND	DATE	PRODUCTSIZE	PRODUCT
IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	IntegerType	StringTyp
86246	205	7	707	1078778070	12564	2012-03- 02	12	oz
86246	205	63	6319	107654575	17876	2012-03- 02	64	oz
86246	205	97	9753	1022027929	0	2012-03- 02	1	СТ
86246	205	25	2509	107996777	31373	2012-03- 02	16	oz
86246	205	55	5555	107684070	32094	2012-03- 02	16	oz
86246	205	97	9753	1021015020	0	2012-03- 02	1	СТ

Workflow for applying TimeFunctions

In the example workflow below, additional date time features are being created from the date column.



In the above workflow:

- The 'CSV' processor reads in the CSV data from HDFS.
- The 'StringToDate' processor converts the column Date, which is in string format to 'timestamp'.
- The 'TimeFunctions' processor takes in the timestamp column and then applies various timefunctions to it to generate additional output columns.

The diagram below shows the dialog box for the TimeFunctions processor. Timestamp column was selected as input, and various time functions were applied to it.

CHEMA :										
Column Name	id	chain	dept	categ	ory company	brand	date	productsize	productmeasure	purchasequa
Column Type	integer	integer	integer	intege	er integer	integer	string	integer	string	integer
IMESTAN		IN NAME	:0		date_type : tim	estamp			\$	
IME FUN	CTIONS	VEA								
IME FUN	CTIONS	MO NO	ΝΤΗ	ч						
	CTIONS	✓ MO ✓ DAY		н						
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY 	NTH OFMONT							
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY ✓ WEE ✓ DAY 	NTH OFMONT OFYEAR EKOFYEAR OFWEEK							
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY ✓ OAY ✓ WEE ✓ DAY ✓ QUA 	NTH OFMONT OFYEAR KOFYEAR OFWEEK ARTER							
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY ✓ WEE ✓ DAY ✓ QUA ✓ QUA ○ HOU 	NTH OFMONT OFYEAR EKOFYEAR OFWEEK ARTER JR							
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY ✓ WEE ✓ DAY ✓ WEE ✓ DAY ✓ QUA → HOL → MIN 	NTH OFMONT OFYEAR KOFYEAR OFWEEK ARTER JR							
	CTIONS	 ✓ MOI ✓ DAY ✓ DAY ✓ WEE ✓ DAY ✓ QUA ✓ QUA ○ HOU 	NTH OFMONT OFYEAR KOFYEAR OFWEEK ARTER JR JR UUTE OND							

Workflow Execution

When the example workflow is executed, additional columns are produced for the various time functions that were selected.

DATE_TYPE	DATE_TYPE_YEAR	DATE_TYPE_MONTH	DATE_TYPE_DAYOFMONTH	DATE_TYPE_DAYOFYEAR	DATE_TYPE_WEEKOFYEAR	DATE_TYPE_DAYOFWEEK	DATE_TYPE_QUARTER	DATE_TYPE
TimestampType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	StringType
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00.0	2012	3	2	62	9	6	1	Spring
2012-03-02 00:00:00:0	2012	3	2	62	9	6	1	Spring
2012-03-02	2012	3	2	62	9	6	1	Spring

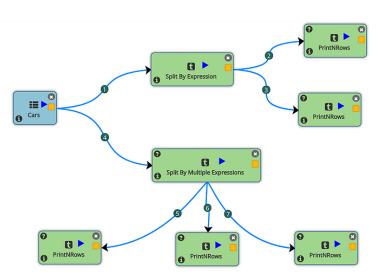
Split Dataset By Expression

Fire Insights allows you to split incoming dataframes. Based on your needs, use the processors described below:

- 'SplitByExpression': This processor splits the incoming dataset based on an expression. Rows satisfying the expression go into one dataframe and the rest go into another dataframe.
- 'SplitByMultipleExpressions': This processor splits the incoming dataset into multiple

dataframes based on up to five conditional expressions. The output of each expression is routed to a separate output path.

• 'Split': This processor splits the incoming dataframe into two based on the percentage specified for the split. Split processor is especially useful in machine learning workflows.



In the example workflow above, 'Split By Multiple Expressions' processor splits the incoming dataframe into three output dataframes. The three conditions are on column c1 - "c1<3", "c1>=3 and c1<5", and "c1>=5". As mentioned earlier, 'Split-ByMultipleExpressions' can split incoming dataframe into up to five dataframes.

Output

String Functions

For the example workflow, the three output dataframes are shown below:

String functions are useful to tranform strings in your dataframe. The "StringFunction" processors allows you to apply common string operations such as 'trim', 'upper', 'lower', 'lefttrim', 'righttrim' etc. to strings.

Workflow

chema :				
Column Name	¢1	¢2	3	c4
Column Type	double	double	double	double
Conditional	e1<3			
Expression 1 to split the Data on				
:•				
Conditional				
	e1 >= 3 and c1 < 6			
Expression 2 to split the Data on	e1 >= 3 and e1 < 5			
Expression 2 to	e1 >= 3 and c1 < 6			
Expression 2 to split the Data on : •				
Expression 2 to split the Data on : • Conditional Expression 3 to split the Data on	e1>= 3 and c1 < 6			
Expression 2 to split the Data on : • Conditional Expression 3 to				
Expression 2 to split the Data on : • Conditional Expression 3 to split the Data on : •				
Expression 2 to split the Data on :•• Conditional Expression 3 to split the Data on :•• Conditional Expression 4 to				
Expression 2 to split the Data on :•• Conditional Expression 3 to split the Data on :•• Conditional				
Expression 2 to split the Data on :•• Conditional Expression 3 to split the Data on :•• Conditional Expression 4 to split the Data on				

c1	c2	c3	c4
DoubleType	DoubleType	DoubleType	DoubleType
1.0	0.0	2.3	3.0
2.0	1.0	3.0	2.0
1.0	0.0	2.3	3.0
2.0	1.0	3.0	2.0
1.0	0.0	2.3	3.0
2.0	1.0	3.0	2.0

c1	c2	c3	c4
DoubleType	DoubleType	DoubleType	DoubleType
3.0	0.0	1.1	1.0
4.0	0.0	4.1	5.0
3.0	0.0	1.1	1.0
4.0	0.0	4.1	5.0
3.0	0.0	1.1	1.0
4.0	0.0	4.1	5.0

c1	c2	c3	c4
DoubleType	DoubleType	DoubleType	DoubleType
5.0	0.0	3.1	6.0
6.0	1.0	2.1	2.0
5.0	0.0	3.1	6.0
6.0	1.0	2.1	2.0
5.0	0.0	3.1	6.0
6.0	1.0	2.1	2.0

In the example below, different string functions are applied to input dataset.

The example workflow below, read data from HDFS/Hive and applies different string functions on different columns of the dataset.

Read data from HDFS

Processor Configuration

DatasetStructured Contemporate Contemporate Contemporate Contemporate Contemporation Contemporat	Ø	Details		
OUTPUT STORAGE LEVEL : 😧		DEFAULT	*	
DATASET : 😧		Housing	\$	

Processor Output

Apply string functions

The 'StringFunctionMultiple' processor below, converts contents of 'driveway' column to upper case and trims contents of 'gashw' column.

Workflow

xecuting No	de fire.nodes.o	dataset.Nodel	DatasetStructu	red : 1 Nov 12,	2018 1:16:59 P/	N						
												_
Row Values												
id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	pref
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	Strin
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no
2	38500.0	4000	2	1	1	yes	no	no	no	no	0	no
3	49500.0	3060	3	1	1	yes	no	no	no	no	0	no
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no
5	61000.0	6360	2	1	1	yes	no	no	no	no	0	no
6	66000.0	4160	3	1	1	yes	yes	yes	no	yes	0	no
7	66000.0	3880	3	2	2	yes	no	yes	no	no	2	no

COLUMN NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
COLUMN TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string
COLUMN FORMAT													
OUTPUT STORAGE LEVE	EL :			DEF	AULT					\$			
DESCRIPTION : 0													
VARIABLES LIST :													
		FUI					REPLAC	E EXISTING C	OLS 🛛				
driveway		¢ up	oper				\$					\$	•
gashw		\$ tri	m				\$					*	•
													OK CANCE

Processor Output

Data Preparation-1

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

Workflow

Below is the workflow. It does the following:

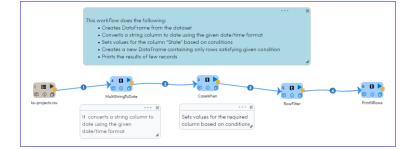
- Reads data from the dataset
- converts a string column to date using the given date/time format

2

oom	fullbase	gashw	airco	garagepl	prefarea	driveway_upper	gashw_trim
ngType	StringType	StringType	StringType	IntegerType	StringType	StringType	StringType
	yes	no	no	1	no	YES	no
	no	no	no	0	no	YES	no
	no	no	no	0	no	YES	no
	no	no	no	0	no	YES	no
	no	no	no	0	no	YES	no
	yes	no	yes	0	no	YES	no
	yes	no	no	2	no	YES	no
	no	no	no	0	no	YES	no
	yes	no	no	0	no	YES	no
	no	no	yes	1	no	YES	no

StringFunctionsMultiple

- Sets values for the column "State" based on conditions
- Creates a new DataFrame containing only rows satisfying given condition
- Prints the results of few records



Reading from Dataset

It reads in the input Dataset File.

Processor Configuration

Processor Output

Convert String to Date

MultiStringToDate converts a string column to date using the given date/time format.

PUT STORAGE I	LEVEL : O			DEFAULT					~				
•:0				'tmp/data/ks-pr	ojects.csv					BROWSE HDFS	NEW FILE		
RATOR : O													
DER : O				true					~				
P MALFORMED				true					~				
EMA COLUMNS	REFRESH DO												
LUMN NAMES FO	R THE CSV Ø		COLUMN	TYPES FOR THE C	5V O		COLL	IMN FORMATS FOR	THE CSV 🛛				
D			INTEG	SER			♥ for	mat				•	
name			STRIP	łG			• for	mat				•	
												,	
adCSV ReadCSV Input Path: Row Values	hdfs://sparkfl	ows-demo.c.spa	rkflaws-168107.in	ternal:8020/	tnp/data/ks-	projects.csv							•
ReadCSV Input Path: Row Values												usd	
ReadCSV Input Path: Row Values ow Values	name	category	main_category	currency	deadline	goal	launched	pledged	state	bockers	country	usd pledged	
RwodCSV Input Path: Row Values ow Values ID IntegerType	name StringType	category StringType	main_calegory StringType	currency StringType	deadline StringType	goal DoubleType	StringType	DoubleType	StringType	IntegerType	StringType	usd pledged DoubleType	
ReadCSV Input Path: Row Values ow Values	name	category	main_category	currency	deadline	goal DoubleType 1000.0						usd pledged	

(AME	ID	name	category	main_category	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd_pledged_real	usd_goal_real	1
TYPE	integer	string	string	string	string	string	double	string	double	string	integer	string	double	double	double	
ORMAT																
ABLES LIST	: •			INPUT COLUMN	FORMATS O			OUTPU	T COLUMN NA	MES O			NEW DATA	YPES 0		
0.000																
deadline				v yyyy-MM-d	4			dear	sline_date				DATE		~ •	

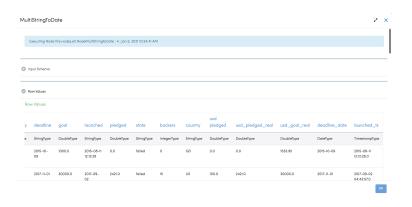
Processor Output

Settings values for required Column

CaseWhen sets values for the required column based on conditions as shown in example below:

Processor Configuration

Processor Output



	ID	name	category	main_category	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd_pledged_real	u 1
TYPE	integer	string	string	string	string	string	double	string	double	string	integer	string	double	double	doul
FORMAT															
															×
OUTPUT ST	ORAGE LI	EVEL : 🛛			DEFA	ULT						~			
OUTPUT CO	DLUMN N	AME : 😡			failed	_index									
EY VALUE	ARRAY : 6	•]												
WHEN CO							VALUE 0								
state =	= "failed"						0							•	
state =	= 'cancele	:d'					1							•	
LSE : 0					2										

Row Value														
	65													
ategory	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd_pledged_real	usd_goal_real	deadline_date	lounched_ts	failed_index
	StringType	StringType	DoubleType	StringType	Double7ype	String7ype	IntegerType	StringType	Double?ype	DoubleType	Double7ype	DateType	Timestamp7ype	Integer/Type
	GBP	2015-10- 09	1000.0	2015-08-11 12:12:28	0.0	folied	0	68	0.0	0.0	1533.95	2015-10-09	2015-08-11 12:12:28.0	0
10	USD	2017-11-01	30000.0	2017-09-02 04:43:57	2421.0	foiled	15	us	100.0	2421.0	30000.0	2017-11-01	2017-09-02 04:43:57.0	0
10	USD	2013-02- 26	45000.0	2013-01-12 00:20:50	220.0	foiled	3	US	220.0	220.0	45000.0	2013-02-26	2013-01-12 00:20:50.0	0
	USD	2012-04- 16	5000.0	2012-03-17 03:24:11	1.0	foiled		US	1.0	1.0	5000.0	2012-04-16	2012-03-17 03:24:11.0	0
10	USD	2015-08- 29	19500.0	2015-07- 04 08:35:03	1283.0	conceiled	14	US	1283.0	1283.0	19500.0	2015-08-29	2015-07-04 08:35:03.0	

Creating DataFrame with required rows

RowFilter creates a new DataFrame containing only rows required.

Processor Configuration



Processor Output

Row Value	15													
alegory	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd_pledged_real	usd_goal_real	deadline_date	launched_ts	failed_index
	String7ype	StringType	DoubleType	StringType	Double7ype	StringType	IntegerType	StringType	Double?ype	DoubleType	Double7ype	DoteType	TimestompType	integer7ype
10	USD	2017-11-01	30000.0	2017-09-02 04:43:57	2421.0	folled	15	US	100.0	2421.0	30050.0	2017-11-01	2017-09-02 04:43:57.0	0
10	USD	2013-02- 26	45000.0	2013-01-12 00:20:50	220.0	foiled	3	US	220.0	220.0	45000.0	2013-02-26	2013-01-12 00:20:50.0	0
	USD	2012-04- 16	5000.0	2012-03-17 03:24:11	1.0	falled		us	1.0	1.0	5000.0	2012-04-16	2012-03-17 03:24:11.0	o
10	USD	2015-08- 29	19500.0	2015-07- 04 08:35:03	1283.0	conceled	14	US	1283.0	1283.0	19600.0	2015-08-29	2015-07-04 08:35:03.0	
	USD	2016-04-	50000.0	2016-02-26	52375.0	successful	224	US	52375.0	62375.0	60000.0	2016-0.4-01	2016-02-26	2

Prints the Results

It prints the first few records onto the screen.

Data Cleaning

This workflow cleans the input data. It does the following:

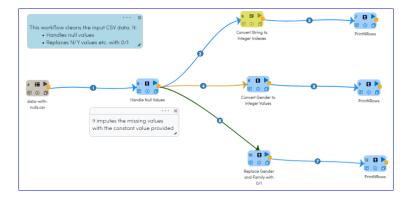
- Handles null values
- Replaces N/Y values etc. with 0/1

Below is the workflow. It does the following:

• Reads data from a dataset

Workflow

- Handles the null values by imputing the missing values with the constant value provided in the specified columns
- Convert Strings to Integer Indexes
- Convert Gender to Integer Values
- Replace Gender and Family with 0/1



Reading from Dataset

DatasetCSV reads in the input Dataset file and creates DataFrame from it.

Processor Output

Row Volues					
Row Values					
id	name	gender	senior_citizen	resident	family
DoubleType	StringType	StringType	StringType	StringType	StringType
1.0	ABC	1	Y	Y	N
2.0	DEF	м	N	N	
3.0	GHR	м	Y	Y	Y
4.0	JRL	F	N	Y	N
5.0	RIT	м	Y	Y	
0.0		F	Y	Y	N
6.0	PQR		Y		Y
0.0	ORT		Y	Y	N

Handling Null Values

ReplaceMissingValueWithConstant processor handles the null values by imputing the missing values with the constant value provided in the specified columns.

Processor Configuration

AME	id .	name	gender		senior_citizen	resident	family	
YPE	double	string	string		string	string	string	
ORMAT								
IPUT STORA	IGE LEVEL : 0		DEFAULT			~		
CUMINS Ø				CONSTANTS 0				
id			~	0			•	
nome			~	TEMP_NAN	1E		•	
gender			*	F			•	
senior_citiz	201		~	N			•	
resident			~	N			•	
family			~	N			•	

Processor Output

Row Values						
Row Values						
id	name	gender	senior_cilizen	resident	family	
DoubleType	StringEype	StringType	StringType	StringType	StringType	
1.0	ABC	r	Y	Ŷ	N	
2.0	DEF	м	N	Ν	N	
1.0	CHR	м	Y	Ŷ	Y	
4.0	J0.	F	N	Y	Ν	
5.0	RIT	м	Y	Ŷ	N	
0.0	TEMP_NAME	F	Y	Y	Ν	
6.0	PQR	F	Y	N	Y	
0.0	ORT	ę.	Y	Y	N	

Convert Strings to Integer Indexes

StringIndexer processor encodes a string type column to a column of label indices.

Processor Configuration

NAME	id .	nome	gender		senior_citizen	resident	family	
TYPE	double	string	string		string	string	string	
FORMAT								
TPUT STORAG	E LEVEL : O		DEFAULT			~		
NDLE INVALID	.0		skip		~			
RIABLES LIST :	0							
NPUT COLUMINS	0			OUTPUT COLUM	N5 0			
gender			*	gender_ind	84		•	
senior_citize	n		~	senior_citizen_index				
resident			~	resident_inc	lax		•	
family			~	♥ family_index				

Processor Output

Convert Gender to Integer Values

CaseWhen processor sets values for the variables based on conditions, as shown below:

nvert String to Integer Indexes		
StringIndeser encodes a string column of labels to a column of label indices.		
tringIndexer encodes a string column of labels to a column of label in	ndices.	
gender_category	gender_categoryIndex	
F	0.0	
M	1.0	
StringIndexer encodes a string column of labels to a column of label indices. tringIndexer encodes a string column of labels to a column of label in	ndices.	
	ndos. serior_citzen_cotegor/index.	
iringIndexer encodes a string column of labels to a column of label i		
tringIndexer encodes a string column of labels to a column of label is senior_citizen_category	senior_cilizen_calegoryIndex	
ringindeser encodes a string column of labels to a column of label is senior_citizen_cotegory Y	senior_citizen_cotegoryIndex	

*
resident_categoryIndex
0.0
10
family_calegoryIndex
0.0
10

w Values									
а	name	gender	senior_citizen	resident	family	gender_index	senior_citizen_index	resident_index	family_index
Double7ype	StringType	StringType	StringType	StringType	StringType	DoubleType	DoubleType	DoubleType	DoubleType
.0	ABC	7	Y	Y	Ν	0.0	0.0	0.0	0.0
2.0	DEF	м	Ν	Ν	Ν	1.0	1.0	1.0	0.0
1.0	GHR	м	Y	Y	Y	1.0	0.0	0.0	1.0
1.0	JR.	F	Ν	Y	Ν	0.0	1.0	0.0	0.0
5.0	RIT	м	Y	Y	Ν	1.0	0.0	0.0	0.0
0.0	TEMP_NAME	F	Y	Y	Ν	0.0	0.0	0.0	0.0
6.0	PQR	7	Y	Ν	Y	0.0	0.0	1.0	1.0
1.0	ORT	F	Y	Y	N	0.0	0.0	0.0	0.0

8 Convert	Gender to Integer Values	NodeCoultNam					~
NAME	ы	nome	gender	senior_citizen	resident	family	1
TYPE	double	string	string	string	string	string	
FORMAT							
DUTPUT STORAG	SE LEVEL : O		DEFAULT		~		
DUTPUT COLUM	N NAME : O		gender_new				
EY VALUE ARRAY	f:00						
WHEN CONDITIO	N O		W	LUE O			
gender == P						•	
gender == 74				0		•	
ilse : O			0				
						ox o	CANCE

Processor Output

low Values							
w Values							
id	name	gender	senior_citizen	resident	family	gender_new	
DoubleType	StringType	StringType	StringType	StringType	StringType	Integer7ype	
1.0	ABC	e e	Y	Y	N		
2.0	DEF	м	N	N	N	0	
3.0	GHR	м	Y	Y	Y	٥	
4.0	JKL	F	N	Y	N		
1.0	RIT	м	Y	Y	N	٥	
0.0	TEMP_NAME	F	Y	Y	N		
1.0	PQR	r.	Y	N	Y		
0.0	ORT	F	Y	Y	N		

Replace Gender and Family with 0/1

FindAndReplaceUsingRegexMultiple processor sets values for the variables based on conditions, as shown below:

Processor Configuration

NAME	id .	nome		gender	senior_citizen		resident	family	1
TYPE	double	string		dring	string		string	string	
FORMAT									
UTPUT STORAG				DEFAULT			*		
INPUT COLUMINS	9		FIND \varTheta			REPLACE O			
gender		~	м			0		•	
gender		~	F					۰	
family		*	N			0		۰	
			Y					•	

Processor Output

Prints the Results

It prints the first few records onto the screen.

Row Values						
w Values						
id	name	senior_citizen	resident	gender	family	
DoubleType	StringType	StringType	StringType	StringType	StringType	
1.0	ABC	Y	Y		0	
2.0	DEF	N	N	0	0	
3.0	GHR	Y	Y	0		
4.0	JRL	N	Y		0	
5.0	RIT	Y	Y	0	0	
0.0	TEMP_NAME	Y	Y		0	
6.0	PQR	Y	N			
0.0	ORT	Y	Y		0	

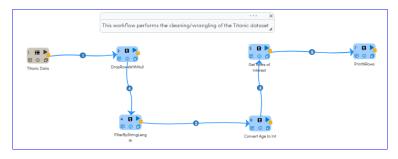
Titanic Data Cleaning/Wrangling

This workflow shows how to wrangle the Titanic Dataset with Sparkflows.

Workflow

This workflow performs the following steps:

- Reads the Titanic dataset
- Drops Rows containing Null values
- Filters the Rows for whom Age has not been specified
- Changes the data type of the Age column to integer
- Filters rows for persons of age > 30 and who are female



Reading Titanic dataset

DatasetStructured processor creates a Dataframe of your dataset named Titanic Data by reading data from HDFS, HIVE etc. which had been defined earlier in Fire by using the Dataset feature.

Executing Node f	ire.nodes.datasi	et.NodeDataset5	itructured : 1 : Jan 5, 2021 9:58:36 AM								
ow Values											
w Values											
assengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
ntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType
	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		s
	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.283	C85	с
	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		s
i.	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	s

Processor Output

Dropping the rows with null values

DropRowsWithNull processor drops the rows with null values.

Processor Configuration

tutorials/data-engineering/../../_assets/tutorials/data-engineering/titani

Processor Output

Executing Node f	ire.nodes.etl.No	deDropRowsWith	Wull : 2 : jan 5, 2021 9:58:19 AM								
Input Schema											
Row Values											
Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType
2	1	1	Curnings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.283	C85	с
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	s
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.863	E46	s

Filter by string length

FilterByStringLength processor filters the rows within the provided string length

Processor Configuration



Processor Output

w Values											
Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
integerType	IntegerType	integer7ype	StringType	StringType	StringType	IntegerType	Integer7ype	String?ype	DoubleType	String7ype	StringType
2			Curnings, Mrs. John Bradley (Florence Briggs Thayer)	femole	38		0	PC 17599	71.283	C85	с
4			Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35		0	113803	53.1	C123	s
7	0		McCarthy, Mr. Timothy J	male	54	0	0	17463	51.863	E-05	5
		3	Sandstrom, Miss. Marguerite Rut	femole	4			PP 9549	16.7	G6	8
12			Bonnel, Miss. Elizabeth	female	5.0	0	0	113783	26.55	C103	5
22		2	Beesley, Mr. Lowrence	mole	34	0	0	248598	13.0	D56	8
24			Sloper, Mr. William Thompson	mole	28	0	0	113786	35.5	A6	s
20	٥		Fortune, Mr. Charles Alexander	male	19	3	2	19950	263.0	C23 C25 C27	5
53			Harper, Mrs. Henry Sleeper (Myna Haxturi)	femole	43		0	PC 17572	76.729	D33	с
55	0		Ostby, Mr. Engelhart Comelius	mole	65	0		113509	61.979	830	с

Convert Age to Integer

CastColumnType processor performs conversion of Age to integer type.

6 Conver	t Age to Int🛛	NodeCastCa	olumnType									ŕ	. ×
NAME	Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	•
TYPE	integer	integer	integer	string	string	string	integer	integer	string	double	string	string	
FORMAT													
		Sur Poli Sex Sib Par Ticl Far Cal	rvived : integer css : integer me : string c: string Sp : integer ket : string re : double bin : string barked : strir	er				ð ¢	Age : strir	1g			
NEW DATA TYPE	E:0			INTEGER					~				
REPLACE EXIST	ING COLS : 0			frue					~				*
												ОК СА	NCEL

Processor Output

nvert Age to	Int										~
Executing Node f	fire.nodes.etl.Noc	deCastColumnTy	pe : 6 : Jon 5, 2021 9:59:29 AM								
nput Schema											
Row Values											
ow values											
Passengerid	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age
	Survived IntegerType	Pclass IntegerType	Name StringType	Sex StringType	SibSp IntegerType	Parch IntegerType	Ticket StringType	Fore DoubleType	Cabin StringType	Embarked StringType	Age
Passengerid											
Passengerld	IntegerType	IntegerType	StringType Cumings, Mrs. John Bradley (Florence	StringType	IntegerType	IntegerType	StringType	DoubleType	StringType	StringType	IntegerType

Get Rows of Interest

RowFilter processor filters the data based on provided conditions as shown below:

Processor Configuration

Processor Output

Prints the results

It prints the first few records onto the screen.

NAME	Passenger	ld S	urvived	Pclass	Name	e Sex	Age	SibSp	Parch	h Tic	ket Fi	are	Cabin	Embarked	
TYPE	integer	in	nteger	integer	r string	string	integer	intege	r integ	er str	ing d	ouble	string	string	
FORMAT															
UTPUT STOR	AGE LEVEL : 0				DEFAU	LT					~				
ONDITIONAL	EXPRESSION * :	0													
														ок	
	r est x.rodes.etJNodeRowFit	er : 5 : Dec 24, 2	2020 7.25:59 PM											OK C	
Executing Node for		er : 5 : Dec 24, 3	2020 7.26.59 PM											OKC	ANCEI
Executing Node fir		er : 5 : Dec 24, 3	2020 7.35.59 PM											OKC	
Executing Node fir Input Schema Row Values		er : 5 : Dec 24, 2	2020 7:25 59 PM											OK C	
t Rows of Inte Executing Node Its Input Scheme Row Yolues Passengerid	n nodes eff. Node Row Fil		2020 7:35:59 PM				Sex	SibSp	Parch	Ticket	Fare	Cabin	e Emba		
Executing Node for Input Schema Row Values ow Values	a nodes et NodeRowFit	class						SibSp	Parch	Ticket	Fore Double7ppe			rhed Age	,
Executing Node Br Input Schema Row Values ow Values Passengerid	a nodes et NodeRowFit	closs tegerīype	Name StringType		Fiorence Briggs		StringType							feed Age	
Executing Node Br Input Schema Row Values ow Values Passengerid IntegerType	s nodes at NodeBowFilt Survived P IntegerSyste Int	class segerifyse	Name StringType	john Bradley (hayer)	StringType female	integerType	irreger7ype	StringType	Double7ype	StringT	iype StringTy	rked Age pe integrity	,
Executing Node for Input Schema Row Values ow Values Passengenid IwegerType 2	Nodec.atl.NodeBowFile Survived Pr IntegerSype Int I 1 1	class regerīype	Nome StringType Currings, Mrs. J	john Bradley (cques Heath ('hayer)	StringType female female	integerType	integerType 0	StringType PC 17589	Double7ype 71.283	C85	iype StringTy C	field Age pe irregory 38	,

Data Wrangling

Data wrangling is the process of gathering, selecting, and transforming data to answer an analytical question. Also known as data cleaning or "munging". This workflow reads in a dataset. It then wrangles the dataset based on provided conditions and prints the results.

Workflow

Below is the workflow. It does the following:

- Reads data from a dataset
- It then create new DataFrame based on the rules provided
- Prints the results



Reading from Dataset

DatasetStructured processor creates a Dataframe of your dataset by reading data from HDFS, HIVE etc. which had been defined earlier in Fire by using the Dataset feature.

Processor Output

low Values										
w Values										
id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType
	42000.0	5850	3	1	2	yes	no	yes	no	no
2	38500.0	4000	2	1	1	yes	no	no	no	no
3	49500.0	3060	3	1	1	yes	no	no	no	no
4	60500.0	6650	3	1	2	yes	yes	no	no	no
5	61000.0	6360	2	1	1	yes	no	no	no	no
6	66000.0	4160	3	1	1	yes	yes	yes	no	yes
7	66000.0	3880	3	2	2	yes	no	yes	no	no
8	69000.0	4160	3	1	3	yes	no	no	no	no
9	83800.0	4800	3	1	1	yes	yes	yes	no	no

Data Wrangling

DataWrangling processor creates new DataFrame after applying the provided rules

Processor Configuration

			bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	1
	eger double	integer	integer	integer	integer	string	string	string	string	string	integer	string	
FORMAT													
JLES : substring col:driveway drop col:id; rename col:recroom t set col:gashw_upper	to recreationroo					 dro del del del set set del 	ame colicit to pp colicit,c2; ete row:(exp ete row:(dat colicit value: colicit value: rive value:AV rive value:SU	ression); did == 'P100 eAge >= 90) UPPER(c2); LOWER(c2); ERAGE(Score	e) as:'avgS				

Processor Output

Executing	iode fire.nodes.etl.No	SeDatoWrangling : 2 :	Dec 24, 2020 8:013	20 PM									
D Input Schen	0												
Row Values													
Row Values													
price	lotsize	bedrooms	bathrms	stories	driveway	recreationroom	fullbase	gashw	airco	garagepl	prefarea	substring_column	
price DoubleTyp		bedrooms IntegerType	bothrms IntegerType	stories IntegerType	driveway StringType	recreationroom StringType	fullbase StringType	gashw StringType	airco StringType	garagepl IntegerType	prefarea StringType	substring_column	
DoubleTyp	integerīype	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType	StringType	StringType	
DoubleTyp 42000.0	integerType	IntegerType 3	IntegerType 1	IntegerType	StringType yes	StringType no	StringType y#s	StringType	StringType no	IntegerType	StringType	StringType ye	
DoubleTyp 42000.0 38600.0	e Integer7ype 5850 4000	integerType 3 2	IntegerType	IntegerType 2 1	StringType yes	StringType no no	StringType yes	StringType no no	StringType no no	IntegerType I	StringType no no	StringType ye ye	

Prints the Results

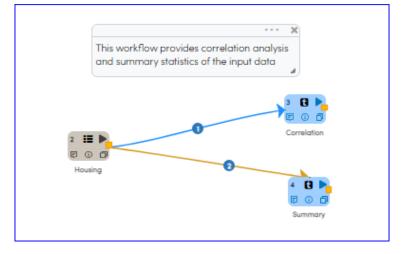
Profiling-Correlation

It prints the first few records onto the screen.

This workflow reads in a dataset. It then creates the correlation analysis and summary statistics.

Workflow

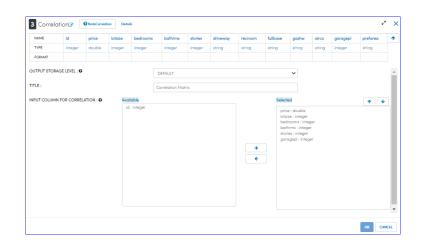
- Below is the workflow. It does the following:
- Reads data from a dataset.
- Perform correlation analysis of the required columns
- Provide summary statistics of the dataset



Performing Correlation analysis

Correlation processor performs correlation analysis on the selected columns as shown below:

Processor Configuration



Processor Output - Correlation matrix

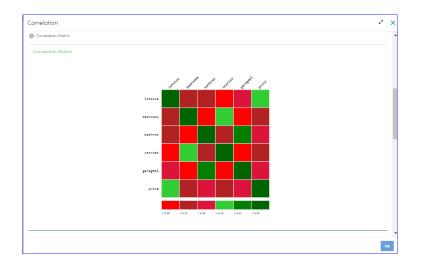
) Input Schema						
Correlation Table						
Correlation Table						
	lotsize	bedrooms	bathrms	stories	garagepl	price
lotsize	1.00	0.23	0.30	0.21	0.42	0.65
bedrooms	0.23	1.00	0.08	0.55	0.20	0.29
bathrms	0.30	0.08	1.00	0.23	0.81	0.52
stories	0.21	0.55	0.23	1.00	0.12	0.27
garagepl	0.42	0.20	0.81	0.12	1.00	0.48
price	0.65	0.29	0.52	0.27	0.48	1.00

Processor Output - Correlation Matrix Heat Map

Processor Output - Sample Rows of Input Dataset

Summary Statistics

Summary processor provides summary statistics of the input dataset.



Row Values											
ow Values											
id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1
2	38500.0	4000	2	1	1	yes	no	no	no	no	0
3	49500.0	3060	3	1	1	yes	no	no	no	no	0
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0
5	61000.0	6360	2	1	1	yes	no	no	no	no	0
6	66000.0	4160	3	1	1	yes	yes	yes	no	yes	0
7	66000.0	3880	3	2	2	yes	no	yes	no	no	2
8	69000.0	4160	3	1	3	yes	no	no	no	no	0
9	83800.0	4800	3	1	1	yes	yes	yes	no	no	0
10	88500.0	5500	3	2	4	yes	yes	no	no	yes	1

Summary statistics provides useful information about sample data. eg: measures of spread.

It provides a table with number of non-null entries (count), mean, standard deviation, and minimum and maximum value for each numerical column.

Processor Configuration

	id	price	lotsize	bedroom	a bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	
TYPE	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	
FORMAT														
UTPUT STORA	GE LEVEL : 0				DEFAULT					~				
TLE :					Summary									
DLUMN NAME	S : 0		Ave	ilable					Select	ted			^	
			ie.	d : integer					pric	e : double				
										ize : integer frooms : inte				
										hrms : integ				
										ries : integer ragept : integ				
								→	90	ugepi . inieş	per .			
								+						

Processor Output: Summary Statistics

nmary						
Summary						
ummary						
summary	lotsize	bedrooms	bathrms	stories	garagepl	price
count	20	20	20	20	20	20
mean	4296.05	2.65	1.2	1.6	0.45	52722.5
min	1700.0	1.0	1.0	1.0	0.0	27000.0
25_percentile	3185.0	2.0	1.0	1.0	0.0	37900.0
50_percentile	3986.0	3.0	1.0	1.0	0.0	45000.0
75_percentile	5200.0	3.0	1.0	2.0	1.0	66000.0
max	7200.0	4.0	2.0	4.0	3.0	90000.0
stdev	1431.639	0.671	0.41	0.883	0.826	19226.725
variance	2049589.945	0.45	0.168	0.779	0.682	3.69666967105E8

Processor Output: Sample Rows of Input Dataset

Change Data Capture

There are many times when we need to Change Data Capture.

Below is one way to do CDC with Fire.

Row Values											
ow Values											
id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl
IntegerType	DoubleType	IntegerType	IntegerType	IntegerType	IntegerType	StringType	StringType	StringType	StringType	StringType	IntegerType
1	42000.0	5850	3	1	2	yes	no	yes	no	no	1
2	38500.0	4000	2	1	1	yes	no	no	no	no	0
3	49500.0	3060	3	1	1	yes	no	no	no	no	0
4	60500.0	6650	3	1	2	yes	yes	no	no	no	0
5	61000.0	6360	2	1	1	yes	no	no	no	no	0
6	66000.0	4160	3	1	1	yes	yes	yes	no	yes	0
7	66000.0	3880	3	2	2	yes	no	yes	no	no	2
8	69000.0	4160	3	1	3	yes	no	no	no	no	0
9	83800.0	4800	3	1	1	yes	yes	yes	no	no	0
10	88500.0	5500	3	2	4	yes	yes	no	no	yes	1
	00000.0	720.0	2	2				1.000		1.000	9

Overview

We have streaming events coming in. The events can be updates to the existing records. In the final table, we need to publish only the latest record.

Design

We keep a staging table. This table would have all the records coming in. We do dedup at the end of the day and publish it to the final table.

Let us say that we are getting real time events of orders. As we get these events we append it to the staging table. If there are updates to an order, say an order got cancelled, we will have multiple records for that order in the staging table.

There is a final published order table where there are no duplicates. It gets updated once a day.

We join the final order table with the staging table. In doing so we get multiple order entries. We take the one with the latest timestamp and drop the others. Then for a given order we have only one record in the final table. We rewrite the final orders table with the newly calculated records.

12.1.6 Data Quality

Data Quality

Data quality is an important aspect whenever we ingest data. Incomplete or wrong data can lead to more false predictions by a machine learning algorithm, we may also lose opportunities to monetize our data because of the data issues and business can lose their confidence on the data.

In sparkflows, user can create the workflow using Summary, Correlation etc nodes to get more details about the dataset.

Sample Dataset: http://eforexcel. com/wp/downloads-16-sample-csv-files-data-sets-for-testing/ Example:

Workflow

Below is the workflow to do Data Profile.

- Reads data from a sample dataset.
- Summary of the numeric fields.
- Correlation of the fields in dataset
- Verfiy the quality of data in sparkflows *Data Quality* tab.

SampleData

Summary

Correlation

Data Quality Page

Summary Results

Correlation Results

Details	OATASET SUHMARY									
ATA PROFILING		COUNT	MEAN	MIN	0,000000	N.,70000711	19,7080934	AM	steev	INDATES
Debard Survivory	Age_in_Ys	1008	40.38	21.04	30.91	40.73	48.05	55.96	321	122.30
Deload Survivory Convestion	Solery	1000	00088.53	40043.00	62060.00	19085.00	189971.00	199543.00	4579.54	204/082207.52
	Nvight_in_Kpi	1008	58.47	45.00	48.00	67.00	68.00	80.00	13.90	193.32
NTA QUALITY	Age in Company (Neon)	1000	10.09	0.00	3.0	7.8	8.44	37.57	6.57	73.43

12.1.7 Code

SQL Examples in Fire

Fire provides a SQL processer in which SQL can be written.

Example 1

select bedrooms, avg(lotsize)
→as avg_lotsize from fire _
→temp_table group by bedrooms

Example 2

s	select fire_
	\rightarrow temp_table.*, case when_
	⇔fire_temp_table.DEP_DELAY_NEW_
	↔> 40 then 1.0 else 0.0 END
·	→as label from fire_temp_table

Scala Examples in Fire

Fire provides a Scala processer in which Scala code can be written.

Below are a few code examples in Scala.

Calculate count of houses by bathrooms

For each bedroom type, find the house with the lowest price

(continues on next page)

(continued from previous page)

<pre>val outDF = lowestPriceDF.</pre>
<pre></pre>
<pre>outDF.registerTempTable("outDF")</pre>

Jar File Execution Example in Fire

Let's take a scenario where through CI/CD pipeline, the application jar file is built successfully and pushed into the S3 bucket.

Below are steps to execute the jar file:

Step 1: Copy jar file from s3 path to /tmp directory.

aws s3 cp s3://bucket-name/ ⇔example-application.jar /tmp

Step 2: Execute jar file from /tmp directory.

java -cp /tmp/example-→application.jar MainClass

In the fire, both steps can be run with UnixShellCommands Node.

1 UnixShellCommands @ 🛛 🛚 🗤	aShellCommand	 ×
OUTPUT STORAGE LEVEL : @ SHELL COMMAND : @	DEFAULT	\$
1 aws s3 cp s3://bucket-name/example-ap 2 java -cp /tmp/example-application.ja		۲ ^۵
		OK CANCEL

12.1.8 NLP

Name Finder

Fire provides NameFinder Processor to easily detect named entities and numbers in text. It takes in a column name in the input DataFrame containing text. It then detects the entities and stores them into a new column.

To be able to detect entities the Name Finder needs a model. The model is dependent on the language and entity type it was trained for.

https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html#tools.namefind.recognition.cmdline

The OpenNLP project offers a number of pre-trained name finder models which are trained on various freely available corpora. They can be downloaded at the OpenNLP download page.

Steps for installing the OpenNLP models in Fire are covered here : http://docs.sparkflows.io/en/latest/operating/ installing-opennlp.html

Workflow

http://opennlp.sourceforge.net/models-1.5/

Below is a workflow which uses the NameFinder Processor.



It consists of 3 Processors:

- TextFiles It reads in the input text file and creates a row from each line of text.
- OpenNLPNameFinder It extracts the entities from each line of text.
- PrintNRows It prints the first 10 rows of the result.

Textfiles

It reads in the input files from the directory data/ner-person. It places each line in the column 'line'.

Processor Configuration

Read in ner-person text file $\ensuremath{\mathscr{C}}$ $\ensuremath{\Theta}$		
OUTPUT STORAGE LEVEL : 0	DEFAULT	
PATH : O	data/ner-person	BROWSE HDFS
OUTPUT COLUMN NAME : 0	lines	VIEW FILE
	lines	
		OK CANCEL

Processor Output

lines
StringType
Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 .
Mr . Vinken is chairman of Elsevier N.V. , the Dutch publishing group .
Rudolph Agnew , 55 years old and former chairman of Consolidated Gold Fields PLC , was named
a director of this British industrial conglomerate .

OpenNLPNameFinder

It extracts entities from the text in the input column 'line' and stores them in the output column 'ner'. When running on the Hadoop Cluster, the model file has to be on HDFS and users have to have access to it.

	lines	
	string	
DEFAULT	\$	
opennlp-models-1.5/en-ner-person.bin		
lines : string	\$	
ner		
	opennlp-models-1.5/en-ner-person.bin lines : string	DEFAULT ¢ opennip-models-1.5/en-ner-person.bin lines : string ¢

PrintNRows

It prints the first 10 rows from the result.

CHEMA :				
COLUMN NAME		lines	ner	
COLUMN TYPE		string	string	
COLUMN FORMAT				
UTPUT STORAGE LEVEL : 😧	DEFAULT		\$	
ITLE :	Row Values			
UM ROWS TO PRINT : 0	10			

12.1.9 Streaming

Streaming Analytics Bike Sharing Dataset

Streaming Analytics with Apache Kafka and Apache Spark Streaming.

At Fire we are obsessed with powering our users to build amazing data analytics applications in < 30 mins.

Below we build a Streaming Analytics work-flow and dashboard. It-

- Reads bike sharing data from Kafka
- Parses the incoming data
- Finds the number of rentals on an hourly basis
- Displays the results visually in a graph.

The dataset contains bike rental info from 2011 and 2012 in the Capital bikeshare system, plus additional relevant information.

This dataset is from Fanaee-T and Gama (2013) and is hosted by the UCI Machine Learning Repository. It consists of 10877 rows (can be found in /data directory of the Fire installation). Each record is count of rentals grouped by a given hour in the past and environmental factors at that time (sea-

son, holiday, temperature, wind-speed etc.)

Start Kafka and create Topic 'bike-sharing'

- The quick start guide of Kafka is at : https: //kafka.apache.org/quickstart
- The steps for Kafka are:

DataSet

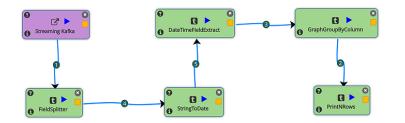
- Download Kafka
- Start zookeeper and Kafka server. You can also use an existing instance of Zookeeper/Kafka
- bin/zookeeper-server-start.sh config/zookeeper.properties
- bin/kafka-server-start.sh config/server.properties
- Create the topic 'bike-sharing'
- bin/kafka-topics.sh –create –zookeeper localhost:2181 –replication-factor 1 –partitions 1 –topic bike-sharing

Send the data file 'bike_sharing_noheader.csv' to the Kafka Topic

- bike_sharing_noheader.csv is in the data directory of the Fire Install
- cat bike_sharing_noheader.csv | bin/kafkaconsole-producer.sh –broker-list localhost:9092 –topic bike-sharing

Workflow

Below is a workflow for Streaming Analytics of the Bike Sharing dataset.



It consists of 6 Nodes:

- StreamingKafka It reads in streaming data from the Kafka topic bike-sharing.
- FieldSplitter It splits each line in fields.
- StringToDate Converts the datetime column into Timestamp type.
- DateTimeFieldExtract : Extracts year, month, day, hour from the datetime column.
- GraphGroupByColumn Groups the data on the hour column, sums it up and display it in a Graph.

• PrintNRows : Prints the first 10 records in a table.

Streaming Kafka

It reads in streaming data from Kafka and creates a dataframe with one column containing the lines.

Streaming K	afka 🚱	
Batch Duration in Seconds : 😡	30	
Kafka Brokers : 😧	localhost:9092	
Consumer Group :	21	
Kafka Topics : 😡	bike-sharing	
Number of Threads : 😡	1	
		OK Cancel

FieldSplitter

It splits each line on the separator - comma - and outputs a new DataFrame with the columns defined.

FieldSplitter	0			
Schema :				
Column Name			line	
Column Type			string	
Input Column : $oldsymbol{\Theta}$ Column Names : $oldsymbol{\Theta}$ Separator : $oldsymbol{\Theta}$	datetime,season,holia	line : string fay,workingday,weather,temp,ate		•
				OK Cancel

StringToDate

It converts the datetime column into new column of type 'Timestamp'.

chema :												
Column Name	line	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	reg
Column Type	string	string	string	string	string	string	string	string	string	string	string	stri
Input C Format	olumn : 🕑		MM/yyyy F		latetime : strin	g				\$		
Output Name :	Column Ø	date	etime_dt									
Output Type: O	Column	TIM	ESTAMP	\$								

DateTimeFieldExtract

It extracts the year, month, day of month and hour from the datetime_dt column.

chema :												
Column Name	line	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	regi
Column Type	string	string	string	string	string	string	string	string	string	string	string	strir
	-											
Columr	n : 🛛			d	latetime_dt : tir	mestamp				\$		
					latetime_dt : tii	mestamp				¢		
Columr Extract		true	!	¢	latetime_dt : tii	mestamp				*		
Extract Extract				•	latetime_dt : tir	mestamp				\$		
Extract	Year: O				latetime_dt : tir	mestamp				\$		
Extract Extract O Extract	Year: 🕢 Month: Day of	true	!	\$	latetime_dt : tii	mestamp				\$		
Extract Extract Ø	Year: 🕢 Month: Day of		!	•	latetime_dt : tii	mestamp				\$		
Extract Extract O Extract Month:	Year: 🕢 Month: Day of	true	•	\$	latetime_dt : tii	mestamp				\$		

GraphGroupByColumn

Aggregates the data on the hour column, and displays it in a Graph.

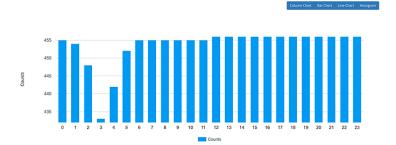
Executing the workflow

When the workflow is executed, Fire submits a spark streaming job to the Spark cluster. The spark streaming job keeps running and

Schema :												
Column Name	line	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	regis
Column Type	string	string	string	string	string	string	string	string	string	string	string	strin
Title : X Label Y Label		Gra X a Y a	xis	ed by Colu	mn							
Group E	By Colur	nn :		d	latetime_dt_ho	ur : integer				\$		
Chart T	ype :			C	olumn Chart					\$		

processing the incoming from Kafka. Below are some of the output produced by the job.



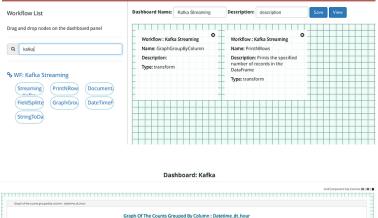


datetime,dt,hour IntegerType	count
IntegerType	LongType
0	455
i	454
2	448
3	433
4	442
5	452
6	455
7	455
8	455
9	455

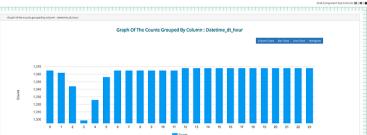
Streaming Dashboard

Since we are still very much under 30 minutes, we also go ahead and create a Dashboard for the workflow. Since we have set the mini-batch duration to be 30 seconds, the Dashboard would update itself every 30 seconds.

Below is the Dashboard editor. Select the nodes whose output you want displayed and



drag and drop them onto the canvas.



12.1.10 OCR

OCR with Tesseract

https://www.sparkflows.io/single-post/OCR-with-Tesseract-in-Sparkflows

12.1.11 REST API

Python - Infer Spark Cluster Configurations

Below is an example Python program for inferring the Apache Spark cluster configurations using the REST API.

It would infer the cluster configurations with latest changes and save the new results.

#!/usr/bin/python
import requests
import json
<pre>token_url = "http:/ →/localhost:8080/oauth/token"</pre>
<pre>infer_configuration_api_</pre>

(continued from previous page)

```
save_configuration_api_

url = "http://localhost:8080/
→api/v1/configurations"
#Step A - resource owner
→ supplies credential #Resource.
→owner (enduser) credentials
#input your own username
RO_user = 'admin'
#input your own password
RO_password = 'admin'
#client
↔ (application) credentials
client_id = 'sparkflows'
client_secret = 'secret'
#step B,
\hookrightarrow C - single call with resource
↔owner credentials in the
→body and client credentials
\leftrightarrowas the basic auth header
⇔will return#access_token
data = {'grant_type
↔': 'password', 'username': RO_
→user, 'password': RO_password}
access_
→token_response = requests.
→post(token_url, data=data,
→ verify=False, allow_
→redirects=False, auth=(client_
→id, client_secret))
print (access_
→token_response.headers)
print (access_
→token_response.text)
tokens = json.loads(access_
→token_response.text)
print ( "access token:_

+ tokens['access_token'])

# Step-
→ now use the access_token
→to call infer configuration
⇔api and its save api.
api_call_headers
→= { 'Aut (continues on next page) earer
```

(continued from previous page)

```
print( api_call_headers)
#infer the hadoop configuration
infer_configuration_
→api_response = requests.
→get(infer_configuration_
→api_url, headers=api_
print("_
→infer configuration response_
→: "+ infer_configuration_
→api_response.text)
#save the hadoop configuration
save_configuration_
→api_response = requests.
→post(save_configuration_
→api_url, infer_configuration_
→api_response, headers=api_
```

```
print(" configuration after_

→save : "+save_configuration_

→api_response.text)
```

12.1.12 Time Series

Stock Forecasting

Objective

Dataset

Stock forecasting helps production units to get an idea about raw material, pricing of goods, improvement in supply, chain man-

Dataset contains 4 columns as follows:-

agement and proper control of sales.

- Date Date when product was sold
- · Store Store id from where product got sold
- Item Item id
- Sales Quantity of product sold
- Predict future sales of items at particular store

Prophet Time Series Modelling Workflow on Multivariate Data

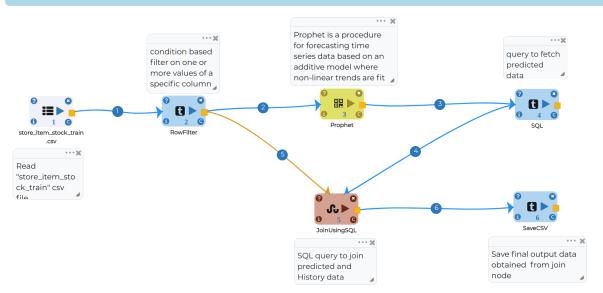
Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends fit with yearly, weekly, daily, seasonality and holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data

and shifts in the trend, and typically handles outliers well.

Safety Stock Calculations for Inventory Management

Periodically, we need to order product to replenish our inventory. When we do this, we have in mind a future period for which we are attempting to address demand along with an estimate the demand in that period.

When actual demand exceeds our forecasts, we run the risk of a stockout (out of stock) situation with its associated potential loss of sales and reduced customer satisfaction. To avoid this, we often include additional units of stock, above the forecasted demand, in our replenishment orders. The amount of this *safety stock* depends on our estimates of variability in the demand for this upcoming period and the percentage of time we are willing to risk an out of stock situation.



Node 1 - ReadCSV

- Reads the given CSV file : store_item_stock_train.csv
- Below are the first 10 rows of data
- Columns contain data as datetype, store and item which are categorical variables and sales which is a continuous variable.

UTPUT STORAGE LEVEL : 0	DEFAU				
	DEFAU		Ÿ		
ATH*:0	/tmp/da	ta/store_itern_stock_train.csv		BROWSE HDPS VIEW FILE	
EPARATOR : 0					
IEADER : 0	true				
			~		
DROP MALFORMED : 0	false		~		
CHEMA COLUMNS: O REFRESH SCHEMA O					
COLUMN NAMES FOR THE CSV O	COLUMN TYPE	IS FOR THE CSV 0	COLUMN FORMATS FOR THE CSV O		
date	DATE	~	format		•
store	INTEGER	v	format		•
item	INTEGER	~	format		•
banes	INTEGER	Ť	format		•
sales	INTEGER	~	format		OK CANC
	INTEGR		format	sales	
ato	INTEGR		kem	sakes 13	
ate 013-01-01 013-01-02	INTEGR	store 1 1	Kem	13	
1800 015 01 01 015 01 02 015 01 03	NTEGR	store 1	isem 1	13 11 14	
NR 013 07.01 013 07.01 013 07.01 013 07.04 013 07.04	INTEGR	store	iem 1	13 11 14 13	
ate 015-01-01 015-01-02 013-01-04 013-01-04 015-01-06	INTEGR	store	Rem	13 11 14 13 10	
ate 013-040 013-042 013-043 013-044 013-013-05 013-016	INTEGR	store	tem	13 11 14 13 10 12	
MAR 615-61-01 015-01-62 015-01-63 015-01-64 015-01-65 015-01-65 015-01-67	INTEER	500 1 1 1 1 1 1 1 1 1 1 1 1 1		13 11 14 13 10 12 10	
aate DOS 07-07 DOS 07-07 DOS 07-02 DOS 07-03 DOS 07-03 DOS 07-06 DOS 07-06 DOS 07-07 DOS 07-09 DOS 07-09 DOS 07-09 DOS 07-09	INTEGR	store	Rem 1	13 11 14 13 10 12	

Node 2 - RowFilter

• Filters data by row with respect to store and item



Node 3 - Prophet

Used Facebook Prophet to create the ML model.

General Section of Prophet Model

- Set Date column in DS column field
- Y is the target variable. Set it to the Sales column
- Set Growth as linear or logistic
- We are using prophet model so it is sufficient to select seasonality in auto mode
- Set mode of seasonality as additive or multiplicative
- Set confidence Interval (0 to 1) which gives a range of plausible values for the parameter of interest.

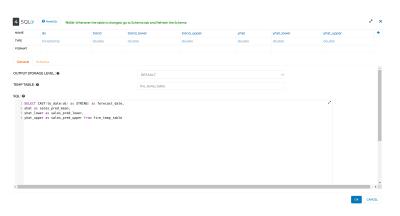
Future Data section of Prophet model

NAME date	store		item	sales		*
TYPE date	integer		integer	integer		
FORMAT						
General Future Data						
DUTPUT STORAGE LEVEL : O		DEFAULT		~		
DS COLUMN : 0		date : date		~		
0		sales : integer		Ý		
GROWTH:0		linear		~		
FEARLY SEASONALITY : 0		auto		Ý		
WEEKLY SEASONALITY : 0		auto		~		
DAILY SEASONALITY : O		auto		Ý		
SEASONALITY MODE : 0		additive		~		
NTERVAL WIDTH : 0		0.95				
<					CK CAN	cu
1				of step	JRE PERIOD	block gives the number
				of stej • FREQ • Set IN	URE PERIOD ps we want to QUENCY can	block gives the number predict
Const Future Data				of stej • FREQ • Set IN	URE PERIOD ps we want to QUENCY can	 block gives the number predict be Monthly or Daily STORY to true for testing
Contail Future Data		10		of stej • FREQ • Set IN	URE PERIOD ps we want to QUENCY can	 block gives the number predict be Monthly or Daily STORY to true for testing
		20 D		of stej • FREQ • Set IN	URE PERIOD ps we want to QUENCY can	 block gives the number predict be Monthly or Daily STORY to true for testing

Node 4 - SQL

General Section of SQL node

• Renames columns forecasted by Prophet



Schema Section of SQL node

• Refreshes Schema and sets data type with respect to columns

Node 5 - JoinUsingSQL

General Section of JoinUsingSQL node

DUTPUT COLUMN NAMES O	OUTPUT COLUMN TYPES O	OUTPUT COLUMN FORMATS O	
forecast_date	DATE	✓ format.	•
sales_pred_mean	DOUBLE	✓ format	•
sales.pred_lower	DOUBLE	✓ format	•
sales.pred.upper	DOUBLE	✓ format	•

• Joins Prediction (from SQL node) and Historical Data(from RowFilter node)

ots date date date othe othe othe www date may m		
data data data data FE data ringar integar integar Seat Integar integar Integar Integar Seat Integration Integration Integration Integration Seat Integration Integration Integration Integration Integration Seat Integration Integration Integration Integration Integration Seat Integration Integration Integration Integration Integration Integration Integration Seat Integration Integration Integration Integration Integration Integration Integration Integration Integration Integrat	double double	
PE data reager reager reager savet data data data data savet data data data data savet data data data data		
Search Schwarz	item sales	
Second TPUT STORAGE LEVEL © OPTALE MARES, 0 tempfabel tempfabe	integer integer	
TVT STOMAC LUVEL • O OFFALT		
P TAULE NAMES • • • • • • • • • • • • • • • • • • •		
i velocit i selocit i selocit	DEFAULT v	
select selectt selectt selectt selectt selectt selecttt selecttt selecttt selectttt selectttt selectttt selecttttt selecttttt selectttttt selecttttttttttttttttttttttttttttttttttt		
2 elect 2 5 elect 5	in province	
2 foreast, date, 3 also prof here, 4 aske prof here, 5 aske prof here, 5 aske prof here, 6 date, 7 store, 5 store, 6 date, 5 store, 5 st		
s Iren, 9 sales D from tempTablel a LEFT JOIN tempTable2 b OW(a.forecast date = b.date)		

Schema Section of JoinUsingSQL node

• Follow the same steps as in Schema Section of SQL node

CHEMA COLUMNS : O REFRESH SCHEMA O			
DLUNN NAMES FOR THE CSV Ø	COLUMN TYPES FOR THE CSV 0	COLUMN FORMATS FOR THE CSV 0	•
precast_date	DATE	 format 	•
ales_pred_mean	DOUBLE	✓ format	•
ales_pred_lower	DOUBLE	✓ format	•
late	DOUBLE	✓ format	•
tore	INTEGER	✓ format	•
em	INTEGER	✓ format	•
ales	INTEGER	✓ format	•

Node 6 - SaveCSV

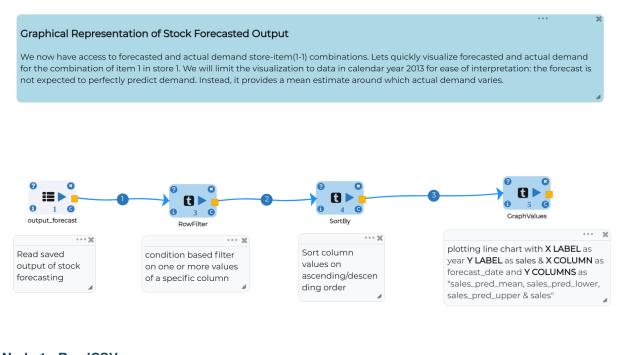
• Sets path where you want to save the final output

Output Visualization

Graphical representation is the best way to understand insights from data. It refers to the use of charts and graphs to visually display, analyze, clarify, and interpret numerical data, functions and other qualitative structures.



Below is the workflow for Visualizing it:



Node 1 - ReadCSV

• Reads output CSV which we have saved from Stock Forecasting.

Node 2 - RowFilter

• Filters dataframe with categorical variables like store and item

Node 3 - SortBy

• Gives options to sort our Dataset based on columns in ascending and descending order

SEPARATOR : O			/tmp/data/outputdemandforeca	st/butput_forecast		BROWSE HDPS VIEW FILE		
			true v					
ROP MALFORMED : 0			false					
CHEMA COLUMNS : 0	REFRESH SCHEMA							
COLUMN NAMES FOR THE CSV			COLUMN TYPES FOR THE CSV O		COLUMN FORMATS FOR THE CSV			
forecast_date			STRING	~	format		•	
sales_pred_mean			DOUBLE	~	format			
sales.pred.lower			DOUBLE	~	format		•	
sales_pred_upper			DOUBLE	~	format			
date			STRING	~	format		•	
store			INTEGER	~	format			_
item			INTEGER	J	format			
							OK C/	NCEL
RowFilterg 0	NodelhowFilter Details							, A
	NodellowFilter Details	store		item		sales		1
AME date	Nssellow-Filter Details	store integer		item Integer		sales		,
AME date	Notellow-Filter Details							,
came date			DEFAULT					,

Node 4 - GraphValue

- Defines labels for X-axis and Y-axis
- Sets columns for X-axis and Y-axis

ITTLE :	Graph			
KLABEL:	year			
LABEL:	sales			
MAX VALUES TO DISPLAY : O	10000			
CHART TYPE :	Line Chart		~	
S STREAMING? : 0	false		~	
COLUMN:	forecast_date : string		~	
COLUMNS: Annual States		÷	Selected sales, pred_mean : double sales, pred_upper : double sales, pred_upper : double sales : integer	* *

Graph obtained

- Sales_pred_mean Blue line
- Sales_pred_lower Red line
- Sales_pred_upper Magenta line
- Sales Yellow line
- Now have a look into graph

Air Passengers Forecasting

Objective

The objective is to develop a time series model to predict future demand of air pas-



sengers which helps Airline company to take decision on aircraft fleet management.

Dataset

Dataset contains 2 columns as follows:-

Month - Month of the year

Passengers - Total number of passengers travelled in that particular month

Air Passengers Occupancy Prediction

Time Series Modelling Workflow on Univariate Data

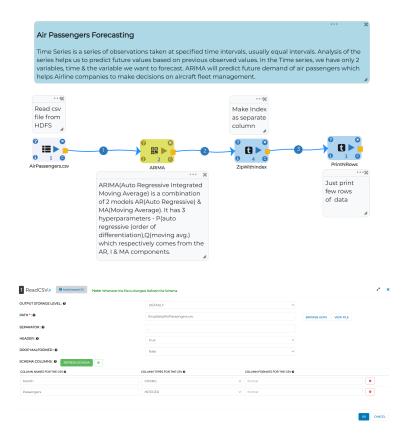
The auto_arima work to fit the best ARIMA(Autoregressive Integrated Moving Average) model to a univariate time arrangement is indicated by either AIC, AICc, BIC or HQIC. The capacity plays out an inquiry (either stepwise or parallelized) over conceivable model requests inside the require-

The auto_arima capacity can be overwhelming. There are a ton of boundaries to tune, and the result is vigorously subject to various themes. In this segment, we spread out a few contemplations you'll need to make when you fit your ARIMA models.

Node 1 - ReadCSV

ments given.

• Reads the given CSV file : AirPassengers.csv



Node 2 - ARIMA

- p The number of lag observations included in the model, also called the lag order.
- d The number of times that the raw observations are different, also called the degree of differencing.
- q The size of the moving average window, also called the order of moving average.
 - Not to worry about p,d,q in this case because we have an interesting model called - AUTO-ARIMA (Able to select automatically optimal value)
- Y Target Variable (Passengers Per Month)
- SEASONAL Automatically True but you can change as false if you want as non-seasonal
- SCORING How do you want to evaluate your model performance like MSE, MAE
- FORECAST Number of steps you want to forecast

2 ARIMA	NodeAutoARIMA		
NAME	Month		yers 1
TYPE FORMAT	string		
OUTPUT STORA	GE LEVEL : ₽	DEFAULT	
(:0		Passengers : integ	
EASONAL : 0		true	
STEPWISE : 0		true	
RACE : 0		true	
SUPPRESS WAR	RNINGS : Ø	true	~
ERROR ACTION	:0	ignore	
CORING : 0		mse	
ORECAST : 0		15	
Summa	ıry		• The model summary reveals a lot of info mation
lode 3	- ZipWithIndex		
			• Creates new column from index of Dataset
lode 4	- PrintNRows		
			• Number of rows you want to print to see the final result
inal R	esult		
			Lets check a few rows of forecasted data ARIMA Model
ime Se	eries Feature Engineeri	ing	
Objecti	ve		
			It is a process of extracting new featur from raw data via data mining technique These features can be used to improve t

Summary:										
				SA	RIMAX	Resul	.ts 			
Dep. Varia	ble:				у	No.	Observations	:	144	
Model:		SA	RIMAX(4, 1	, 3)	Log	Likelihood		-674.913	
Date:		Fr	i, 30	0ct	2020	AIC			1365.825	
Time:				12:3	3:24	BIC			1389.528	
Sample:					Θ	HQIC	;		1375.457	
				-	144					
Covariance	Type:				opg					
	cc	e===== Def	std	err			P> z	[0.025	0.975]	
ar.L1	-0.55	582	 0.	 117			0.000	-0.787	-0.329	
ar.L2	0.49	935	0.	113	4	.375	0.000	0.272	0.715	
ar.L3	0.12	238	0.	128	Θ	.970	0.332	-0.126	0.374	
ar.L4	-0.52	213	0.	085	- 6	.136	0.000	-0.688	-0.355	
ma.L1	0.90	969	0.	094	9	.657	0.000	0.723	1.091	
ma.L2	-0.55	590	0.	145	- 3	.866	0.000	-0.842	-0.276	
ma.L3	-0.73	385	0.	109	- 6	.778	0.000	-0.952	-0.525	
sigma2	724.17	724	85.	616	8	. 458	0.000	556.369	891.976	
Ljung-Box	(Q):				256	.02	Jarque-Bera	(JB):	1	.4.5
Prob(Q):					Θ	.00	Prob(JB):			0.0
Heterosked	asticity	(H):			5	.66	Skew:			0.7
Prob(H) (t	wo-sided)):			Θ	.00	Kurtosis:			3.5

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

3 PrintNI	ROWS NodePrintFirstNRows			×
NAME	Forecast_ARIMAX		Index	•
TYPE	double	long		
FORMAT				
OUTPUT STOR	AGE LEVEL : O	DEFAULT	\checkmark	
TITLE :		Row Values		
NUM ROWS TO	D PRINT : 0	10		

Forecast_ARIMAX	Index
DoubleType	LongType
467.573805910366	0
490.49456508436225	1
509.1369442565889	2
492.55476914877306	3
495.3059691093772	4
475.94780369392686	5
476.3398372576565	6
475.5521380450703	7
472.35382516685223	в
483.8896762403088	9

OK CANCEL

Dataset

Dataset contains 4 columns as below:

- Date Date when product was sold
- Store Store id from where product got sold
- Item Item id
- Sales Quantity of product sold

Create new feature from existing table to improve performance of models

Feature Engineering Workflow

run and more handily.

Moving average

Each column is a feature. But all features may not produce the best results from models, so feature engineering plays an important role in choosing the right features. A model will not entirely improve its prescient force, yet will offer the adaptability to utilize less unpredictable models that are quicker to

One step moving average

- Moving average is commonly used to streamline short-period fluctuations in time series data and feature long-term patterns.
- For one step, window size will be from -1 to 1 for sales data

Seven step moving average

- For seven step, window size will be from -7 to 7 for sales data
- Moving average output

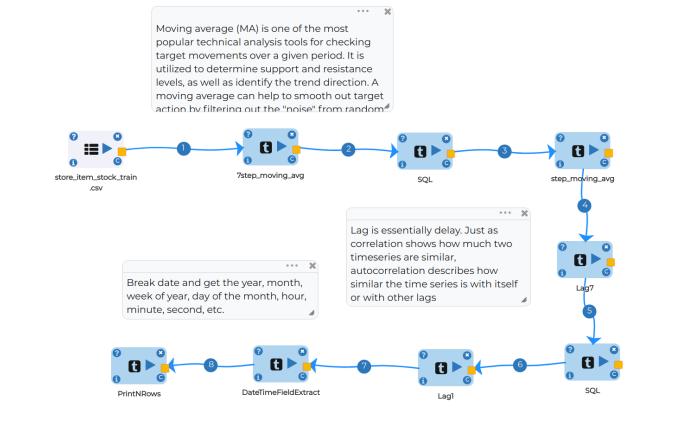
Extract Date Time Features

- Break date and get the year, month, week of year, day of the month, hour, minute, second, etc.
- Output of Date Time Features

Time Series Feature Engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. Feature engineering tries to come up with the right set of predictors for a model. We can do some feature engineering for time series data as:

- Rolling mean, min, max, etc. statistics
- \cdot Bollinger bands and statistics
- \cdot Rolling entropy, or rolling majority, for categorical features



WINDOW START : 0	-1	
WINDOW END : 0	1	
PARTITION COLUMN NAME : 🚱	store : integer 🗸	
ORDER COLUMN NAME : 0	sales : integer 🗸	
VARIABLES LIST:		
INPUT COLUMNS Ø	FUNCTIONS 🛛	
sales	avg	~ •
		OK CANCEL

date	store	item	sales	mean_sales	7_mean_sales
DateType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType
2013-01-08	1	1	9	9.0	10.75
2013-01-10	1	1	9	9.333333333333333334	11.0
2013-01-05	1	1	10	9.6666666666666666	11.3
2013-01-07	1	1	10	10.3333333333333334	11.3
2013-01-02	1	1	11	11.0	11.3
2013-01-06	1	1	12	11.6666666666666666	11.3
2013-01-09	1	1	12	12.33333333333333334	11.3
2013-01-01	1	1	13	12.666666666666666	11.3
2013-01-04	1	1	13	13.33333333333333334	11.5555555555555555
2013-01-03	1	1	14	13.5	11.875

COLUMN : 😡	date : date	\sim
EXTRACT YEAR : 🛛	true	~
EXTRACT MONTH : 0	true	~
EXTRACT DAY OF MONTH : 0	true	\sim
EXTRACT HOUR : 0	true	~
EXTRACT MINUTE : 0	true	~
EXTRACT SECOND : 🚱	true	~
EXTRACT WEEKOFYEAR : 0	true	~

date	store	item	sales	date_year	date_month	date_dayofmonth	date_hour	date_minute	date_second	date_weekofyear
DateType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType
2013-01-01	1	1	13	2013	1	1	0	0	0	1
2013-01-02	1	1	n	2013	1	2	0	0	0	1
2013-01-03	1	1	14	2013	1	3	0	0	0	1
2013-01-04	1	1	13	2013	1	4	0	0	0	1
2013-01-05	1	1	10	2013	1	5	0	0	0	1
2013-01-06	1	1	12	2013	1	6	0	0	0	1
2013-01-07	1	1	10	2013	1	7	0	0	0	2
2013-01-08	1	1	9	2013	1	8	0	0	0	2
2013-01-09	1	1	12	2013	1	9	0	0	0	2
2013-01-10	1	1	9	2013	1	10	0	0	0	2

Lags Feature

- Lag is used to make non-stationary data into stationary data
- Outliers are easily discernible on a lag plot
- acf and pacf plot is used to calcluate best lags

Lag one

- The most commonly used lag is 1, called a first-order lag
- Window shift is one

PARTITIONBY : O	store	
ORDERBY : 🚱	date	
	lag	/
ANALYTICS COLUMN :	sales : integer	/
WINDOW OFFSET : 🛛	1	

Lag seven

• Window shift is seven

date	store	item	sales	lag	7_lag
DateType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType
2013-01-01	1	1	13	None	None
2013-01-02	1	1	11	13	None
2013-01-03	1	1	14	11	None
2013-01-04	1	1	13	14	None
2013-01-05	1	1	10	13	None
2013-01-06	1	1	12	10	None
2013-01-07	1	1	10	12	None
2013-01-08	1	1	9	10	13
2013-01-09	1	1	12	9	11
2013-01-10	1	1	9	12	14

New feature data

Anamoly Detection for IOT Devices

Objective

Anomaly detection issue for time arrangement can be planned as discovering exception information guides relative toward some

date	store	item	sales	date_year	date_month	date_dayofmonth	date_hour	date_minute	date_second	date_weekofyear	lag	7_lag	mean_sales	7_mean_sales
DateType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	IntegerType	DoubleType	DoubleType
2013-01-01	1	1	13	2013	1	1	0	0	0	1	None	None	12.6666666666666666	11.3
2013-01-02	1	1	11	2013	1	2	0	0	0	1	13	None	11.0	11.3
2013-01-03	1	1	14	2013	1	3	0	0	0	1	11	None	13.5	11.875
2013-01-04	1	1	13	2013	1	4	0	0	0	1	14	None	13.33333333333333334	11.555555555555555555555555555555555555
2013-01-05	1	1	10	2013	1	5	0	0	0	1	13	None	9.666666666666666	11.3
2013-01-06	1	1	12	2013	1	6	0	0	0	1	10	None	11.666666666666666	11.3
2013-01-07	1	1	10	2013	1	7	0	0	0	2	12	None	10.33333333333333334	11.3
2013-01-08	1	1	9	2013	1	8	0	0	0	2	10	13	9.0	10.75
2013-01-09	1	1	12	2013	1	9	0	0	0	2	9	n	12.333333333333333334	11.3
2013-01-10	1	1	9	2013	1	10	0	0	0	2	12	14	9.333333333333333334	11.0

norm or common sign. Our center will be from a machine persopective, for example, surprising spikes, level move highlighting disintegrating soundness of a machine.

Dataset

Dataset contains 4 columns as follows:-

- Datetime 10 mins time interval of accelerometer data
- 4-Bearings Contains reading of devices

Anamoly Detection using Prophet Time Series Model Workflow

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends fit with yearly, weekly, daily, seasonality and holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Time-series anomaly detection

is a feature used to identify unusual patterns that do not conform to expected behavior, called outliers.

Data Preprocessing

- Column Filter convert multivariate data into univariate for prophet model
- Output Univariate data
- **Prophet** Model for anomaly detication using mean as threshold value

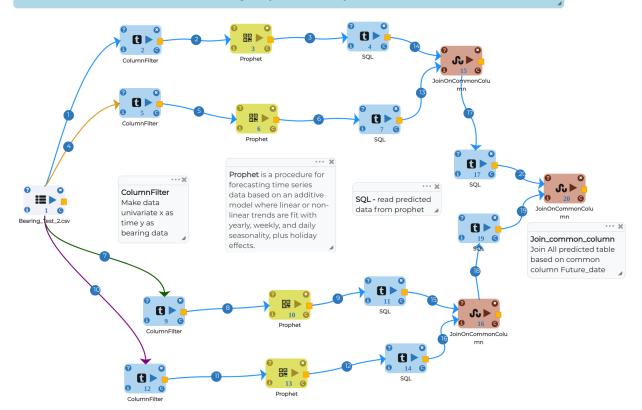
General Section of Prophet Model

• Set Datetime column in DS column field

Data Modeling

Anamoly Detection For IOT Devices

Generally, conditioning monitoring of a machine is done by looking at a sensor mesurement (Eg. Temperature, Vibration) and imposing bounds to it, i.e. under normal operating conditions, the measurement values are bounded by a maximum and minimum value (similar to control charts). Any deviation is the defined bounds sends an alarm. This is often generally defined *as anamoly detection*.





Datetime	Bearing_1
TimestampType	DoubleType
2004-02-12 10:32:39	0.058332877581913
2004-02-12 10:42:39	0.058995214610088
2004-02-12 10:52:39	0.060236437326041
2004-02-12 11:02:39	0.061455442160261
2004-02-12 11:12:39	0.061360759802725
2004-02-12 11:22:39	0.061664827384149
2004-02-12 11:32:39	0.061943893744811
2004-02-12 11:42:39	0.061230528834415
2004-02-12 11:52:39	0.062279749987792
2004-02-12 12:02:39	0.059890277845597

- Y is the target variable. Set it to the reading of bearings
- Set Growth as linear or logistic
- We are using prophet model so that it is selfsufficient to select seasonality in auto mode
- Set mode of seasonality as additive or multiplicative
- Set confidence Interval (0 to 1) which gives a range of plausible values for the parameter of interest.

General Potore Data		
OUTPUT STORAGE LEVEL : 0	DEFAULT	~
DS COLUMN : O	Datetime : timestamp	~
Y:0	Bearing_1: double	~
GROWTH : 0	linear	~
YEARLY SEASONALITY : 0	auto	~
WEEKLY SEASONALITY : 0	auto	~
DAILY SEASONALITY : 0	auto	~
SEASONALITY MODE : 0	additive	~
INTERVAL WIDTH : 0	0.95	

Future Data section of Prophet model

• FUTURE PERIOD block gives the number of steps we want to predict

General Future Data		
FUTURE PERIOD : 0	24	
	24	
FREQUENCY: 0	D	
INCLUDE HISTORY: 0	false	~

• SQL set mean column to set threshold

Model prediction

· Threshold to compare anomaly

ds	trend	trend_lower	trend_upper	yhat	yhat_lower	yhat_upper
TimestampType	Double?ype	DoubleType	DoubleType	Double?ype	DoubleType	Double7ype
2004-02-13 12:02:39	0.0065570403065542335	-0.4314989436380987	0.443870248355367	0.0065570403065542335	-0.4314334302089222	0.44348778316922943
2004-02-14 12:02:39	-0.04765893255461476	-1.2551897482733952	1.235643745446955	-0.04765893255461476	-1.2552038753818562	1,2358456648999048
2004-02-1512:02:39	-0.10187490541578374	-2.4618087335060457	2.188138763682763	-0.10187490541578374	-2.4616337516841176	2.187445509713839
2004-02-1612:02:39	-0.15609087827695273	-3.7689904345524265	3.35756280666189	-0.15609087827695273	-3.7689098347803065	3.357621979300629
2004-02-17 12:02:39	-0.2103068511381217	-5.3647822962412555	4.605712801394639	-0.2103068511381217	-5.3638858389282635	4.605428933121821
2004-02-1812:02:39	-0.2645228239992907	-6.981627822573064	6.1714043515849175	-0.2645228239992907	-6.98272353067758	6377796056104398
2004-02-1912:02:39	-0.3187387968604697	-8.94688944368949	7.832083459660579	-0.3187387968604997	-8.947444893896066	7.832509699896468
2004-02-2012:02:39	-0.37295476972162867	-10.606858671880655	10.032112103876145	-0.37295476972162867	-10.606690639862599	10.032533196920488
2004-02-2112:02:39	-0.42717074258279764	-12.637549731463602	11.838283845322355	-0.42717074258279764	-12.636874299933673	11.838282437376805
2004-02-2212:02:39	-0.4813867154439667	-14.90960174003428	14.331184407396112	-0.4813867154439667	-14.909209104792312	14.331133324256857

Future_time	Bearing_3_pred	Bearing_4_pred	Bearing_2_pred	Bearing_1_pred
TimestampType	DoubleType	DoubleType	DoubleType	DoubleType
2004-02-13 12:02:39	0.12892150390232607	0.018985275825913313	0.10739056982158202	0.0065570403065542335
2004-02-14 12:02:39	0.17377822690500572	-0.00595058722291632	0.1394695544580692	-0.04765893255461476
2004-02-15 12:02:39	0.21863494990768537	-0.030886450271745948	0.17154853909455636	-0.10187490541578374
2004-02-1612:02:39	0.263491672910365	-0.05582231332057558	0.2036275237310435	-0.15609087827695273
2004-02-17 12:02:39	0.3083483959130447	-0.08075817636940523	0.2357065083675307	-0.2103068511381217
2004-02-18 12:02:39	0.3532051189157243	-0.10569403941823484	0.2677854930040179	-0.2645228239992907
2004-02-1912:02:39	0.39806184191840394	-0.13062990246706446	0.299864477640505	-0.3187387968604597
2004-02-20 12:02:39	0.4429185649210836	-0.15556576551589413	0.33194346227699223	-0.37295476972162867
2004-02-2112:02:39	0.48777528792376323	-0.18050162856472377	0.3640224469134794	-0.42717074258279764
2004-02-2212:02:39	0.532632010926443	-0.2054374916135534	0.3961014315499666	-0.4813867154439667

CHAPTER 13

Troubleshooting

13.1 Troubleshooting

13.1.1 Installation

Installation Pre-requisites

Below are the Pre-requisites before installing Fire:

JDK 1.8+ installed on the machine java **and** jar have to be **in** the PATH If running on an Apache Spark cluster, Apache Spark 1.6+ **is** needed on the cluster. 3GB+ of RAM available on the machine.

With which user should Fire be installed

If Fire needs to be connected with an Apache Spark cluster the below is needed:

- Fire needs to be installed as a user which can impersonate other users. Impersonation for this user has to be set up in HDFS configs.
- If you disable impersonation in Fire, then the user with which Fire is installed needs to be able to submit jobs to the cluster.

More Details are available here : https://www.sparkflows.io/connecting-sparkflows-with-spark-cl

I do not see anything in my browser after I start Fire

Do check in the logs for exceptions and the root cause. On Linux and Mac, the log files are in nohup.out.

Possible causes are:

- The H2 database was not created and it is failing to the find the table.
- The server did not start properly because some other Application is running on the configured port. The default configured port for Fire is :8080

The http and https ports for Fire can be updated in conf/application.properties.

Fire UI does not get displayed when I go to :8080. Some other UI is displayed

Fire by default runs on port 8080. It is possible that you have some other application running on port 8080, and you are seeing its output. In this case, the solution is to run the Fire server on some other port which is not being used by any other application. Details for running Fire on another port is here : https://www.sparkflows.io/run-fire-on-different-port

13.1.2 LDAP

Fire can be configured to authenticate the user with LDAP. Below are some ways to troubleshoot the LDAP configurations.

Testing LDAP connection with Idapsearch

It is a good idea to test the ldap environment setup using ldapsearch. This ensures that the machine is setup correctly for LDAP - it can connect to the LDAP server, the LDAP username and passwords are correct, the SSL certificates are good if using LDAPS.

Testing Getting User Details from LDAP

- cd to your installation directory
- Create a properties file called ldaptestconfig.properties

Below is an example:

```
ldap_attributeUserName=myLdapUsername
ldap_Order="DB_LDAP";
ldap_URL="ldap://localhost:10389";
ldap_base="dc=example,dc=com";
ldap_userDn="uid=john,ou=bindusers,dc=example,dc=com";
ldap_password="johnspassword";
ldap_userSearchBase="ou=sparkflow";
ldap_userSearchFilter="(uid={0})";
ldap_groupSearchBase="ou=groups";
ldap_groupSearchFilter="member={0}";
```

Fetch the user details for the user xyz with the following command:

java -cp app/fire-ui-3.1.0.jar -Dloader.main=fireui.ldap.LDAPTest org.springframework. →boot.loader.PropertiesLauncher xyz

What if I get locked out

ldap.Order determines the order in which Fire tries to log in the user. In case you are locked out of Fire and are not able to log in, you can do the following:

• Add the below line to conf/configuration.properties

ldap.Order=DB

• Then restart the fire server. Now you should be able to log in with your admin account.

Once things are back to normal, you can remove the line you added to configuration.properties and restart the fire server.

13.1.3 Upgrade

Missing column: application_id in FIREDB.PUBLIC.ANALYSIS_FLOW_EXECUTION

After I upgrade to the latest Fire Release I get the error : Missing column: application_id in FIREDB.PUBLIC.ANALYSIS_FLOW_EXECUTION or something similar.

After upgrading the Fire Server, it is important to upgrade the Database Schema.

- Upgrade it by running create-h2-db.sh or create-mysql-db.sh from the Fire install directory.
- This would upgrade your DB schema to the latest.

Otherwise you can run into an error like below, when you start the Fire Server:

13.1.4 Dataset

I am getting an error when clicking 'Update' button on the Create/Update Dataset page

You may see the error below:

Unable to retrieve schema **for** this path :: Bad header **for** field, should start **with** a_ →character **or** _ **and** can contain only alphanumerics **and** _ 0:" id 1 "

- This is because one of the column names of the header is not in the right format. In this case the column name id 1 contains a space.
- Only alphanumerics and _ are permitted in the header and column names.
- If your data does not have a header column, set the Header field to false when defining the Dataset.

13.1.5 Running Workflows

Getting Exception : 'User: ec2-user is not allowed to impersonate ec2-user

Sparkflows impersonates the logged in user when submitting the jobs onto the Cluster.

So, the user with which Sparkflows is running has to be configured on HDFS as a proxy user.

Details for allowing the sparkflows user to impersonate other users is available at:

• ../installation-upgrading/connecting-spark-cluster

When running the workflows on my Spark Cluster, results are not showing up in the Browser

This is probably because there is some configuration error. Sparkflows uses spark-submit to submit the jobs to the cluster. The driver of the spark job posts back results to the Fire server.

- Check out the log for spark-submit for the workflow in /tmp/fire/workflowlogs to find the root cause. Maybe the spark job is just failing.
- It is also useful to ensure Spark jobs can be submitted to the Cluster from the machine on which Sparkflows is running with spark-submit. Submit the SparkPi job from spark-examples.jar to test it.
 - SparkPi can be run with a command like : spark-submit --class org.apache.spark. examples.SparkPi --master yarn --deploy-mode client spark-examples.jar 10
 - spark-examples.jar is in your Apache Spark install direction on the machine.
- If the Spark job is running successfully (according to the logs), but the results are still not showing up in the Browser, it could be because the fire spark job is unable to post results back to the Fire web server. You should see these failures in the logs.
 - Under Administration/Configuration, there is the config app.postMessageURL. It determines the Fire URL to which the results from the spark driver are posted back to the fire server. Ensure that it is set up correctly.

Getting Exception: org.apache.hadoop.security.AccessControlException: Permission:denied : user=admin

When running on the Cluster, you are running into the exception below:

- If the above exception is coming up when running the workflow, then it means that the logged in user does not exist on HDFS.
- In the above case, the user is logged into Fire as admin. So the jobs submitted by Fire on the cluster is as the user admin. But the user 'admin' does not exist on HDFS.
- Please make sure to log into Fire as a user which exists on HDFS.

When running the example workflows on the Spark Cluster it is not able to find the input files

The example workflows read in input files.

• They have to be on HDFS in the home directory of the logged in user.

- The data directory which comes with Sparkflows has to be uploaded onto HDFS.
- For example, if the logged in user is john, then the data directory would be on HDFS in the directory /user/john

Getting Exception : Server returned HTTP response code: 405 for URL: http://10.125.221.72:8080/ messageFromSparkJob

When submitting jobs to the cluster from Fire, you are running into the exception below:

Fire submits Spark jobs to the cluster. The spark driver, posts certain results back to the Fire server to be displayed to the user.

The cause of this error is that the postback-url has not been set correctly - http://10.125.221.72:8080/ messageFromSparkJob

There could be following issues with the URL:

```
The machine name/IP is wrong. It has to be the machine on which Fire is running.
The port number is wrong. Fire server is running on another port on the machine.
```

Getting Exception : java.lang.ClassNotFoundException: fire.execute.WorkflowExecuteFromFile

When running the jobs on the cluster, you are running into the exception below.

- The reason for it is that the app.sparkSubmitJar is not set up correctly. Fire comes with a jar file which gets submitted to the cluster with spark-submit. app.sparkSubmitJar has to correctly point to this jar file.
- You can go under Administration/Configuration to set it up correctly.

Exception:

```
Warning: Local jar /home/ec2-user/fire-2.1.0/fire-lib/fire-spark_1_6-core-2.1.0-jar-

→with-dependencies.jar does not exist, skipping.

java.lang.ClassNotFoundException: fire.execute.WorkflowExecuteFromFile at java.net.

→URLClassLoader.findClass(URLClassLoader.java:381) at
```

(continues on next page)

(continued from previous page)

```
java.lang.ClassLoader.loadClass(ClassLoader.java:424) at java.lang.ClassLoader.
→loadClass(ClassLoader.java:357) at
java.lang.Class.forName0(Native Method) at java.lang.Class.forName(Class.java:348) at
org.apache.spark.util.Utils$.classForName(Utils.scala:177) at
org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$
→$runMain(SparkSubmit.scala:688) at
org.apache.spark.deploy.SparkSubmit$$anon$1.run(SparkSubmit.scala:163) at
org.apache.spark.deploy.SparkSubmit$$anon$1.run(SparkSubmit.scala:161) at java.
⇔security.AccessController.doPrivileged(Native Method) at
javax.security.auth.Subject.doAs(Subject.java:422) at
org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1917)_
→at
org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:161) at
org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206) at org.apache.
⇔spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121) at
org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
```

Getting Exception on HDInsight : No FileSystem for scheme: wasbs

When running the jobs on the cluster, you are running into the exception below.

- The reason for it is that it is not understanding the scheme wasb. In order to fix it, run ./ run-fire-spark-submit.sh start instead of ./run-fire.sh start.
- This enables getting the distribution libraries into the executable.

Exception:

```
Error : java.io.IOException: No FileSystem for scheme: wasbs at
org.apache.hadoop.fs.FileSystem.getFileSystemClass(FileSystem.java:2586) at
org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:2593) at
org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:91) at
org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:2632)
```

13.1.6 Fire Server & Workflow Execution Logs

Where do I find the logs of the Fire Server

When running on linux or mac the logs of the Fire Process are in the file fire.log. Logs of the Fire Web Server are in the file fireserver.log under the directory where Fire has been installed. It would be something like/fire-2.1.0

Where do I find the logs of the workflows when running on my Cluster

The logs are in the directory /tmp/fire/workflowlogs on the machine on which the Fire server is running:

Each workflow execution has its own log file.

The json representation of the workflow is in /tmp/fire/workflows when running in YARN client mode. They are in .fireStaging directory under the users home directory on HDFS when running in YARN cluster mode.

13.1.7 Dashboards

When viewing the Dashboard the cells are showing up empty

Dashboards show output of Workflows.

If the corresponding workflow has not executed, the content in the Dashboard would show up as empty.

13.1.8 Kerberos

My cluster is Kerberised. How do I setup Sparkflows for it

The steps to setup Sparkflows on a Kerberised cluster are at:

• ../installation-upgrading/configuration/configuring-kerberos

13.1.9 Python Installation

Python installations from source with version 3.6.5

showing warning message with missing package while restarting pyspark server

showing warning message with missing package while restarting pyspark server:

UserWarning: Could not import the lzma module. Your installed Python is incomplete

Possible Solution

For centos: Install development tool:

```
sudo yum install -y xz-devel
```

Recompile python from source code:

```
cd Python-3.6.5
sudo ./configure --enable-optimizations
sudo make altinstall
```

CHAPTER 14

Frequently Asked Questions

14.1 FAQ

14.1.1 Scheduling Workflows

How can I schedule the workflows I create ?

Fire Insights saves workflow definitions as JSON files. These workflows are executed through spark-submit. Fire Insights has a scheduler which allows Workflows to be scheduled at regular intervals. Since the workflows are submitted with spark-submit, they can also be easily scheduled with Oozie, crontab etc.

14.1.2 Custom Nodes

Does Fire Insights allow me to create my own custom nodes?

Yes, new Nodes can be easily to added to Fire Insights. Develop nodes in Java or in Scala and dop the definition JSON for the node on the server. The newly added nodes will become visible in the Fire Insights User Interface.

14.1.3 Distributions Supported

What distributions or platforms are supported with Sparkflows?

Sparkflows Fire has been tested with CDH, Hortonworks, MapR, AWS EMR, Apache Spark distributions.

Note: Any cluster with Apache Spark 1.6+ will work fine with Sparkflows.

Can I run Sparkflows on my Amazon AWS cluster or Microsoft Azure or Google Cloud?

Yes, all Sparkflows needs for successful deployment is a Apache Spark cluster. Sparkflows is deployed on the edge node of the cluster.

14.1.4 Workflow Export - Import

How does one export/import workflows between instances?

Sparkflows allows workflows to be exported and imported. Workflows are represented as JSON files and hence can also be checked into github etc. for versioning.

Sparkflows also maintains the version history of the workflows.

14.1.5 Submit Apache Spark Jobs

When running on a Apache Spark cluster how does Sparkflows submit the spark jobs?

Fire Insights uses spark-submit to submit the Apache Spark jobs to the cluster. Hence it is important that spark-submit work from the machine on which Fire Insights is installed.

14.1.6 Multi User Support

How does the Sparkflows platform handle multi-user support (i.e. Can user 1 see or edit user 2's data sources, pipelines, etc)

Sparkflows supports various user types and enables users to easily share datasets and workflows with each other to foster collaboration.

14.1.7 Data Sources

How does one define a new data source and establish a connection?

Sparkflows platform has various OOTB connectors to HIVE, Flume, Kafka, HBase, Solr. For all other structured or unstructured datasets on HDFS or CloudBricks, Sparkflows platform can identify the schema on the fly when a new dataset is created in Sparkflows pointing to a data source. The schema can be updated right there as well. Sparkflows workflow execution writes a summary of its output to MySQL/Oracle/H2 which is accessible by the users of the system.

14.1.8 Hadoop Installation Pre-Requisites

Below are the pre-requisites for installing Hadoop:

- Linux
- JDK 1.8 installed
- IPV6 disabled
- Selinux disabled

Linux

Minimum machine configuration:

- vCPU : 8 vcores
- RAM: 32 GB

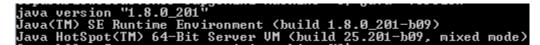
JDK

JDK 8 is needed on the Linux Machine. Below are the steps for installing oracle java:

- Install java 8 as the root user
- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- wget –no-cookies –no-check-certificate –header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "https://download.oracle.com/otn-pub/java/jdk/8u201-b09/ 42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.rpm"
- yum localinstall jdk-8u201-linux-x64.rpm

Ensure that java 8 is installed properly:

• java -version



Set the below in .bash_profile

• export JAVA_HOME=/usr/java/jdk1.8.0_201-amd64/

Disable IPV6

• Edit file /etc/sysctl.conf - vi /etc/sysctl.conf

Add the following lines:

- net.ipv6.conf.all.disable_ipv6 = 1
- net.ipv6.conf.default.disable_ipv6 = 1

Execute the following command to reflect the changes.

• sysctl -p

Selinux

Just ensure that selinux should be disabled so that it cant impact Hadoop performance.

• sudo setenforce 0

To disable it permanently

edit /etc/selinux/config

SELINUX=disabled

reboot

Steps Involved in Installing Hadoop

- Install bind-utils : Otherwise Cloudera Manager gives host not found
 - yum install bind-utils
- Install Cloudera Manager
 - cd
 - wget https://archive.cloudera.com/cm5/installer/latest/cloudera-manager-installer.bin
 - chmod u+x cloudera-manager-installer.bin
 - ./cloudera-manager-installer.bin
 - Accept Licenses
- Open ports on Linux Machine
 - Open the ports 7180 and 8080

After Installation of Cloudera Manager

- go to http://host-ip:7180/
 - Log in with admin/admin
 - Select Cloudera Express Installation
 - For host, give the hostname IP (private IP)
 - Install using Parcels
 - Include the Kafka parcels
 - User : sparkflows (As per as updated on machine while creating Linux Machine)
 - Supply the private key
 - Install Core with Spark
 - Update default Configurations in it.

Add proxy user in HDFS

- Add sparkflows as proxy user in HDFS
 - https://www.sparkflows.io/connecting-sparkflows-with-spark-cl
 - Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml
 - * hadoop.proxyuser.sparkflows.hosts
 - * hadoop.proxyuser.sparkflows.groups
- Restart Cluster services

Root access to your hosts i password-less sudo/pbrun		ra packages. This installer will connect to your hosts via SSH and log in either directly as
Login To All Hosts As:	o root	
	 Another user 	
	ec2-user	(with password-less sudo/pbrun to root)
You may connect via passw	ord or public-key authenticati	on for the user selected above.
Authentication Method:	 All hosts accept same pas 	ssword
	 All hosts accept same priv 	/ate key
Private Key File:	Choose File sparkflows.pem	
Enter Passphrase:		
Confirm Passphrase:		
SSH Port:	22	
Number of Simultaneous Installations:	10 (Running a lar	ge number of installations at once can consume large amounts of network bandwidth and

Create HDFS directory

Create HDFS directory for sparkflows user (we can create as per as requirements)

- sudo su
- su hdfs
- hadoop fs -mkdir /user/sparkflows
- hadoop fs -chown sparkflows:sparkflows /user/sparkflows

Install Spark2

spark2 is installed using CSD or Parcels

- https://www.cloudera.com/documentation/spark2/latest/topics/spark2_installing.html
 - cd /opt/cloudera/csd
 - sudo su
 - wget http://archive.cloudera.com/spark2/csd/SPARK2_ON_YARN-2.1.0.cloudera2.jar
 - chown cloudera-scm:cloudera-scm SPARK2_ON_YARN-2.1.0.cloudera2.jar
 - chmod 644 SPARK2_ON_YARN-2.1.0.cloudera2.jar
 - service cloudera-scm-server restart

Login Again into Cloudera Manager

- In Cloudera Manager:
 - Go to Hosts/Parcels

- Download Spark2
- Distribute Spark2
- Activate Spark2
- Add Spark2 service in Cloudera Manager
 - Go to Cluster/Add Service
 - Add Spark2 Service
 - For dependency select one with HIVE etc.
 - Select the host

In YARN increase Container memory to 8GB

- yarn.scheduler.maximum-allocation-mb
- yarn.nodemanager.resource.memory-mb

AFTER INSTALLATION GET CDH TO USE JAVA 8

- In Spark configuration in Cloudera Manager set the below for spark-defaults.conf
 - spark.executorEnv.JAVA_HOME=/usr/java/jdk1.8.0_201-amd64/
 - then redeploy the client configurations
 - Restart the cluster service

Install Sparkflows

- ssh to the machine
- wget https://s3.amazonaws.com/sparkflows-release/fire/rel-x.y.z/2/fire-x.y.z.tgz
- tar xvf fire-x.y.z.tgz
- cd fire-x.y.z
- ./create-h2-db.sh
- ./run-fire.sh start
- ./run-fire-server.sh start

Upload the Fire Insights example data directory onto HDFS

- As sparkflows user
- cd fire-x.y.z
- hadoop fs -put data

Log into Fire Insights

- http://host-ip:8080/#/dashboard
 - Log in with admin/admin
 - Create user sparkflows in Sparkflows. Give it admin rights. Add to group default, save it.
 - Again Login with sparkflows user.
 - Go to Configurations under administration and click on infer hadoop cluster config and save it.
 - open spark and update spark2-submit under "spark.spark-submit" and save it.
 - Create a workflow and execute it.

CHAPTER 15

Administration

15.1 Administration Guide

15.1.1 User Administration

Fire allows you to create and manage

- Users
- Groups
- Roles

These are accessible under the Administration Menu.

Users

- Fire allows you to create and edit users
- Users belong to groups and have roles
- A user can be a designated as a superuser
- The user should exist on HDFS (when running against a Hadoop Cluster). Fire can run independent of a Hadoop Cluster.

Groups

- Fire allows you to create and edit groups
- Groups allow users to share Datasets, Workflows and Dashboards with other groups

Roles

- Fire allows you to create and edit roles.
- A role has various permissions associated with it.

Permissions

Fire has the following permissions defined.

- ✓users.manage
- groups.manage
- ✓roles.manage
- Configurations.manage
- datasets.view
- datasets.modify
- Workflows.view
- Workflows.modify
- Workflows.execute

CHAPTER 16

Databricks Integration

16.1 Databricks Guide

16.1.1 Databricks Prerequisites

Below are the Prerequisites for installing Fire Insights on a Databricks Cluster:

Package	Description	Value
Python	python version on Databricks	3.6.0 or above
version	Cluster	
pip version	pip version on Databricks	20.0 or above
	Cluster	
Spark version	Spark Version on Databricks	2.4
	Cluster	
Fire Running	Port on Which Fire is Running	Accessible from databricks Cluster
Port		

Table 1: Below are the Needed Package

16.1.2 Databricks Integration Steps

Fire Insights integrates with Databricks. It submits jobs to the Databricks clusters using the REST API of Databricks and have the results displayed back in Fire Insights.

Fire also fetches the list of Databases and Tables from Databricks, making it easier for the user to build their workflows and execute them. In addition fire displays the list of Databricks clusters running for the user.

Databricks can be running on Azure or on AWS.

 Running Databricks on Azure : quickstart-create-databricks-workspace-portal

https://docs.microsoft.com/en-us/azure/azure-databricks/

• Running Databricks on AWS : https://databricks.com/aws

Below are the steps for Integrating Fire Insights with your Databricks Clusters.

Install Fire Insights

Install Fire Insights on any machine. The machine has to be reachable from the Databricks cluster.

Upload Fire Core Jar to Databricks

Fire Insights jar has to be uploaded to Databricks. Fire Insights jobs running on Databricks make use of this jar file.

Upload fire-x.y.z/fire-core-lib/fire-spark_2_3-core-3.1.0-jar-with-dependencies. jar to Databricks. Upload it under Workspace as a Library on to Databricks.

- 1. Login to Databricks Cluster
- 2. Click on workspace in the left side pane



3. Create a new Library



- 4. Upload fire-spark_2_4-core-3.1.0-jar-with-dependencies.jar from your machine by Clicking on Drop JAR here
- 5. Once fire-spark_2_4-core-3.1.0-jar-with-dependencies.jar is uploaded, click on Create
- Check the box with Install automatically on all clusters, in order to avoid installing it manually to every cluster.

Configure the Uploaded Library in Fire Insights

Configure the path of the uploaded fire core jar library in Databricks in Fire Insights.

This has to be done under Administration/Configuration.

Micros	soft Azure
Azure Databricks	Create Library Library Source
	Upload DBFS PyPI Maven CRAN
Home	Library Type
Ъ	Jar Python Egg Python Whl
Workspace	Library Name
٥	Optional
Recents	
Data	
	Drop JAR here
Clusters	
Jobs	
Q Search	Create Cancel

Micros	oft Azure
8	Create Library
Azure Databricks	Library Source
_	Upload DBFS PyPI Maven CRAN
(*)	
Home	Library Type
Ъ	Jar Python Egg Python Whl
Workspace	Library Name
0	fire-spark 2.3-core-3.1.0-jar-with-dependencies.jar
Recents	
Data Clusters	fire-spark_2 core-3.1.0-jar- with- dependencies.j;
Jobs	Remove file
Q Search	Create Cancel

Micros	oft Azure	
ATUR	fire-spark_2.3-core-3.1.0-jar-with-dependencies.jar	
Databricks Mome	fire-spark_2.3-core-3.1.0-jar-with-dependencies.jar	encies.jar
Workspace	Source	
0	dbfs:/FileStore/jars/ccbd9fc8_e256_4b10_9ab3_6562fdf4b9ce-fire_spark	伦 Copy
Recents	Status on running clusters	
	Install automatically on all clusters	
Data	S Uninstall % Install	
Configuration	ons	
SAIT CONFIGUR	VIONS INTER INCODE CURTER CONTIS	
APP ARFLON	V SPARK HDPS HADOOP YARN HIVE KERBEROS LDAP ALERT DATABROXS	

Configure app.postMessageURL in Fire Insights

Configure app.postMessageURL to be the IP of the machine on which Fire Insights is installed. Jobs running on Databricks would post back results to Fire Insights using this URL.

ARFLOW	SPARK HDPS HADOOP	YARN HIVE REPORTED LOAP ALERT DATAG	ROG AWS PLUGINS
NAME	10.4	WER	OSSCRIPTION
opp.poeMessogeURL	Fire of portback URL	http://Fublic.lp.8083./messageFromSponijab	Fire all postbock URL for the messages of executing jobs

Install Databricks JDBC Driver

Fire needs the Databricks JDBC Driver to be installed. Install it in the fire-user-lib and fire-server-lib folder of the Fire installation.

You can download the Databricks JDBC Driver from the Databricks site :

- https://docs.databricks.com/bi/jdbc-odbc-bi.html
- · https://databricks.com/spark/odbc-driver-download

The driver is available as a zip file. eg: SimbaSparkJDBC-2.6.3.1003.zip

- Unzip the downloaded file. It will create a directory like SimbaSparkJDBC-2.6.3.1003
- Copy the jdbc jar file named SparkJDBC4.jar into fire-x.y.z/fire-user-lib and fire-x.y. z/fire-server-lib

Create your REST API token in Databricks

Create your token in Databricks. It would be used in making REST API calls to Databricks from Fire Insights.

- 1. Login to your Databricks Account
- 2. Click on Account icon in right corner top
- 3. Click on User Settings
- 4. Click on Generate New Token
- 5. Add comment & Lifetime (days) for token expiry & Click on Generate
- 6. Copy the token generated. Click on DONE

Create Databricks Connection in Fire Insights

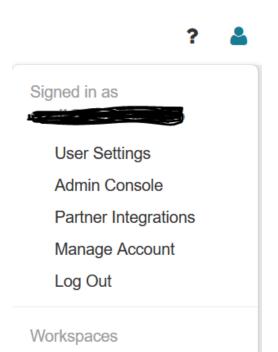
Create a connection in Fire Insights to Databricks.

It can be created by the Administrator under Administration/Global Connections. These connections are available for everyone to use.

It can also be created by any user with their Application. In this case, it is only available to the Application and its users.

- Specify your Databricks Token.
- Specify the Databricks JDBC URL of your cluster in Databricks.

Now we are ready to start using the Databricks Connection in Fire Insights to:



User Settings Admin Console Partner Integrations Manage Account Log Out

Workspaces

User Settings		
Access Tokens Git Integration Notebook Settings		
Personal access tokens can be used for secure authentication to the Data	abricks API instead o	of passwords.
Generate New Token		
Comment		
databricks development		
Lifetime (days) 😧		
90		
	Cancel	Generate

Generate N	ew Token
Your token has been cre	ated successfully.
dapif34e533561efd5b5	f6c6ae986e13df82
A Make sure to copy the	e token now. You won't be able to see it again.
	Done
Add Connection	
CONNECTION TYPE @ *	Databricks
CONNECTION NAME @*	Test_databricks
TOKEN @*	
TITLE 😧 *	databricks
DESCRIPTION @	test
URL 🎯 *	jdbc:spark://eastus.azuredatabricks.net:443/default;tr
	SAVE CANCE

- Browse DBFS
- · View your Databricks Clusters
- · Browse your Databricks Databases & Tables
- · Create Workflows which Read from and Write to Databricks

16.1.3 Databricks Python Integration Steps

Fire Insights integrates with Databricks and can submit Python jobs. It submits jobs to the Databricks clusters using the REST API of Databricks and have the results displayed back in Fire Insights.

Below are the steps for Integrating Fire Insights with your Databricks Clusters for running Python jobs.

Note: The Machine on which Fire Insights is installed should have Python 3.7.0 or above.

Python Installation Steps:

https://docs.sparkflows.io/en/latest/installation/python-install-linux.html

Install Fire Insights

Install Fire Insights on your machines. The machine has to be reachable from the Databricks cluster.

Upload Fire wheel file to Databricks

Fire Insights wheel file has to be uploaded to Databricks. Fire Insights jobs running on Databricks make use of this wheel file.

Upload fire-x.y.z/dist/fire-3.1.0-py3-none-any.whl to Databricks. Upload it under Workspace as a Library on to Databricks under DBFS or even in S3 Bucket which is accessible from the Databricks Cluster.

1. Login to Databricks Cluster



2. Click on workspace in the left side pane



3. Create a new Library

You can select Library Source as DBFS, Library Type as Python Whl, provide any Library Name field, & add File Path of fire-3.1.0-py3-none-any.whl located in DBFS.

\bigcirc	Create L	ibrary				
databricks	Library Sour	ce				
	Upload	DBFS/S3	PyPI	Maven	CRAN	
A Home	Library Type					
Þ	Jar Py	/thon Egg	Python V	Vhl		
Workspace	Library Nam	е				
0	fire-python-	-wheel				
Recents						
	File Path					
	dbfs:/FileSt	tore/jars/fire-3.	1.0-py3-n	one-any.wh		
Data						
(H	Create	Cancel				
Clusters						
Jobs						

On Clicking on Create button it will ask to install on specific databricks Cluster, select cluster on which you want to install.

On Successfull installation of wheel file on Databricks Cluster, it would be displayed under Libraries.

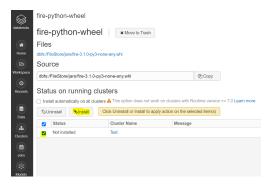
Another option is to upload fire-3.1.0-py3-none-any.whl file to s3 Bucket which is accessible from Databricks Cluster.

Once you upload fire-3.1.0-py3-none-any.whl file to s3 Bucket, login to Databricks Cluster & inside Libraries tab.

Install New Library & select DBFS/S3 in Library Source, Python Whl in Library Type and copy paste the location of python wheel file available in s3 in File Path & Click on Install.

Once it is installed successfully, you can see the python wheel inside Library is up.

	fire-python-wheel				
databricks	fire-python-whee	× Move to Trash			
	Files				
Home	dbfs:/FileStore/jars/fire-3.1.0-py3-n	one-any.whi			
B	Source				
Workspace	dbfs:/FileStore/jars/fire-3.1.0-py	3-none-any.whl		f2) Copy	
O Rocents	Status on running clu Install automatically on all cluster SUninstall % Install	usters ars 🛦 This option does not work on cl	usters with R	untime version	>= 7.0 Learn more
_	Status	Cluster Name	Message		
Clusters	Not installed	Test			
Jobs					



	Clusters / Test		
adabricas	🔵 Test 🕴 🕼 Clone 🖉 Re	estart Terminate	* Delete
e Home	Configuration Notebooks (0) Libraries Event	Log Spark UI Drive	r Logs Metrics Apps Spark Cluster UI - Master +
ь	Si Uninstall % Install New		
Workspace	Name	Type Status	Source
۲	fire-spark_2.4-core-3.1.0-jar-with-dependencies	JAR • Installed	s3.//sparkflow-sample-data/sparkflowsampledata/
Recents	dbfs:/FileStore/jars/fire-3.1.0-py3-none-any.whl	Wheel Installed	dbfs:/FileStore/jars/fire-3.1.0-py3-none-any.whl
B Dels			

Clusters / Test	
🔵 Test 🕴 🕼 Edit 🖉 Ci	Install Library ×
Configuration Notebooks (0) Libra	Ubrary Source Upload DBFS/S3 PyPI Maven CRAN Workspace
Name fire-spark_2.4-core-3.1.0-jar-with-de	Library Type Jar Python Egg Python WH
	File Path
	s3//sparkflow-sample-data/sparkflowsampledata/databricks_a
	Cancel Install

luste	ers / Test					
• 1	Test	Gr Eck	fg Cione	2 Rostart	Tarminato	a R Delote
Cont	Iguration	Notebooks (1)	Ubraries	Event Log	Sperk UI Dr	itver Logs Metrics Apps Spark Cluster UI - Master +
	ninstal	% Instal Now				
SU	Name	% install Now		Тур	e Status	Source
	Name	% Install Now	r-with-depende			

Install Python dependencies

You need to install the python dependencies required by Fire Insights on the machine by running below Command from fire-x.y.z/dist/fire/directory:

```
pip install -r requirements.txt
```



Note: Make sure that pip etc. is already installed on that machine

Install dependency for AWS

Copy the jars hadoop-aws and aws-java-sdk to pyspark jar path.

murmurni, 1 ec2-usen ec2-usen	11948376 Aug 31 12:36 /home/ec2-user/veru/lib/python3.6/site-packages/pyspark/jars/aws-java-sdk-1.7.4.jar
(venv) [ec2-usenRip-172-31-77-18	1 ~15 1s -1 /home/ec2-user/verw/lib/python3.6/site-packages/pyspark/lars/hadoop-aus-2.7.3.1ar
-nw-nw-n, 1 ec2-usen ec2-usen	126287 Aug 31 12:36 /home/ec2-user/venv/lib/python3.6/site-packages/pyspark/jars/hadoop-aus-2.7.3.jar
(very) (ec2-usenitin-172-31-77-18	1~15 1s -1 /home/ec2-user/verw/lib/python3.6/site-packages/pyspark/lars
total 217396	
-rw-rw-r 1 ec2-user ec2-user	69409 Aug 31 12:10 activation-1.1.1.jar
mu-mu-n 1 ec2-user ec2-user	134944 Aug 31 12:10 aircompressor-0.10.jar
mi-mi-n 1 ec2-user ec2-user	445285 Aug 31 12:10 antln-2.7.7.jan
milminica 1 ec2-user ec2-user	334662 Aug 31 12:10 antlr4-runtime-4.7.jan
mirminnin, 1 ec2-usen ec2-usen	164368 Aug 31 12:10 antlr-runtime-3.4.jar
-nu-nu-n 1 ec2-usen ec2-usen	4467 Aug 31 12:10 appalliance-1.0.jan
-nw-nw-n 1 ec2-usen ec2-usen	14766 Aug 31 12:10 appalliance-repackaged-2.4.0-b34.jan
-nw-nw-n 1 ec2-usen ec2-usen	44925 Aug 31 12:10 apacheds-i18n-2.0.0-M15.jan
-rw-rw-r, 1 ec2-user ec2-user	691479 Aug 31 12:19 apacheds-kerberos-codec-2,0.0-MIS, jan
interneties 1 activities activities	448794 Ave 31 12:10 spacks-logic-extract 2 17 day

Install any specific package of python, if Need to use in Custom Processors on databricks Cluster aswellas Fire Insights Machine.

Use the command below to install it on the Fire Insights machine:

```
pip install scorecardpy
```



Install it on your Databricks cluster with the below:

```
* Open a Notebook and attach to Databricks Cluster.
* %sh pip install scorecardpy
```

Upload Fire workflowexecutedatabricks.py file to DBFS

For Python Job submission to Databricks Cluster.

Upload fire-x.y.z/dist/workflowexecutedatabricks.py, file to DBFS or even S3 Bucket too.

You can UPLOAD it, using DBFS Browser too.

da 🔹 Test	1v	9+ 0?+ 0	a+ n ¢								2 B	•	4
1 lish pip install	scorecardp												~ -
Collecting scorecas													
Requirement already				/databric	ks/python3	l/lib/pythor	13.7/site-p	packages (1	ros scorec	ardpy) (0.2	1.3)		
Collecting pandas>													
Downloading http:		thoshosted.or	T/backaBes/g	1/#d/1011	12535645ad7	ratd26c3b9de	+fd20a32d9v	#42340619c4	1731126cca	d0Gee/panda	8-1.1.1.	cp37+cp	078
anylinux1_sD6_64.wd Requirement already													
Requirement already										0.3)			
Requirement already										-0.10.1-1-0-0			
Requirement already													
Requirement already													
Requirement already													0.8
0)													
Requirement alread	satisfied:	kfwiselver>:	1.8.1 in /dz	rabricks/	python3/li	b/python3.1	/site-pack	kapes (from	matplotl1	b->scorecar	(by) (1.)	1.0)	
Requirement alread													
Requirement already	satisfied:	pyparsingled	1.0.4,1=2.1.2	,112-1-6,	>=2.8.1 in	/databrick	cs/python3,	/lib/pythor	3.7/site-p	ackages (fr	on natpl	otlib->	-500
cardpy) (2.4.6)													
Requirement already	satisfied:	sfa>=1.5 in	/databricks/	python3/1	1b/python3	1.7/site-pac	skages (fri	on python-c	lateutil>=2	.T.3->panda	82-01521	3->scor	ieco.
py) (1.12.0)													
Requirement already	satisfied:	setuptcols :	in /usr/14b/p	ython1.7/	site-packa	iges (from k	cturisolver:	>=1.0.1->m	rtplat15b->	scorecardpy) (40.8.0	3)	
												_	
		77-101 fin											

Configure the Uploaded Library in Fire Insights

Configure the path of the uploaded fire python wheel package file & workflowexecutedatabricks.py under databricks.pythonFile & databricks.pythonPackages respectively in Fire Insights.

It can be two source either DBFS or S3 path.

If you have Uploaded in DBFS path.

If you have Uploaded in S3 path.

Job Submission using Pyspark Engine

Now You can submit pyspark jobs to Databricks Cluster from Fire Insights.

16.1.4 Databricks User Guide

Browsing Databricks Tables

Fire Insights allows you to Browse your Databricks Databases & Tables.

Go to Data/Databricks DB

It will display the Databricks DB page.

Select the Tables

Once you select the Tables, right click on it to get the query to view the first few records from the table. Execute the sql query to view records from the table selected.

Databricks File System				
Alfoma		UNLOSO FUE Connection	Test Ouster 1	ype-dotobricks 🛛 🛩
HAME	IS DIRECTORY	NDI .	920	ACTION
D Riddee	true .	/TieDore		•
Do Atraducidos envelto	true .	Adotabricke results		•
D modelsmpath	5.00	./modelsowepath		•
D /mitmedelpath	100	/steinodelpath		•
D recipiermodel	the second se	/steelploterrodel		•
C fest_workflee.pas	false	/het_workflow.juan	6083	•
D rep	5.00	/http		•
D Awar	100	Aper		•
D /vsekdevenani olariskekdis.gv	folse	/wohflowenecutedatabricks.py	1813	•

		NEDOCP CLUTTE								٩		
APP DATAB	ROS AR	NOW ANS	SPARK	HOPS HADOG	DP YARN	HWE KE	REEROS LI	DAP ALERT	PLUGINS	MODULE	UISETTINGS	
NAME	π	na -		MAGE					DESCRIPT	CN .		
databricks.encb	ind in	Databrics enabled		5140					Of Vitation in	regration with Da	nobricha is enable	
databe ida. auber	*)*	th to Fire care (or) viccoment	n Detobricka		-somple-doto's with-dependen		oko Marabertoka)	o/fire-spork_2.4	C Peth of the data.rind dependent	e spork core jor w tankrijanu Tire-spo cies jar	th dependencies ark_2.3-core-3.10	on DBMS, egy Departmenter
databricks.pytho	ntie h	th to main python visconnext	file in Osfsbricks	this work for	vereculeddiator	chi py			G Path of Bri	gypork mein fi	le on DEFE. eg	
databricks.pytho		t of python librarie eculion in dotobric		o dife. Telbore	verec fediration	py)-none-ony whi					i os CBF3, eg: 04_4533_4e3F_1	
ME CONFIGURATI												
Configure	ations											
SANS CONFIDER	NTOIS B	FER HADOOP CU	INTER CONFIG							۹		
APP DATAD		now Aws	SPARK	HOFS HADO	OP YARN	HNT K	LRBCROS	LDAP ALER	T PLUGNS	MODULE	UI SETTING	8
NAME		TITLE	ta bled" tu						DESCRIPTION	tere with the 1	ricks is enabled	
dotobricka.aub		Path to fire con Databricks Envi		Naporkflow-samp ark_3.4-cane-3.1				1790- 127 (Path of Fire spor Sblc /FileStore/j Seperchencies (a	k core jor with d pru/Tre-spark_ r	lependencies or 2.3-core-3.1.0-j	offs. og ar with-
dotobricks.pyt	hon/ile	Path to main py Databricks Envi	thon file in 13	Aport/low-samp angorit/lowsomp	ole- niediana (doneb	nas jaowaat	owner and on	anicke py 12	Path of the pysp Bals:/jobs/fire/C	ark main file on latebricks/work	DBIS.og Nov_execute p	w.
dotobricka.py/f	hos/fackages	List of python 18 needs during er	marties that	Appent flow-some	sk-dofe/spork	Novampledete	aldanobyicka je	effre-110.	Path of the pysp dblc/FileStore/9	ark libraries on pre/d29c9494	CEPS. eg: 402d Hedt Mr	15_077487415d34/3re-
		danabricas							110 gy3-rore	orywhi		
	_											
ANE CONFIGURA	TIONS											
ONLOV D. O			858							0.00	- Cook	15
ORFLOW NAME	0		het_bet							_	_	
MAK-SUBHITIOS	9.740											
MAK-SUBHITIOS	an 6											
MAK SUBATIO	an.e		ag rassalar r	www.bj.ann.do	er osena 2 - altis	ar man or y 2g						
NARK SUBHIT-CC					er ocenen 3 - altitu	an manut jij						
NOOSE JAR FILES	e		ng manadar a		er ssamet 2 – abba	ar maray 3g						
NOOSE JAR FILES					er somen 2 - ables	ar man any 2g						
NOOSE JAR FILES					er somen 2 mable	ar manay 2g						
NOCES HE FLES		~			er somen 2 – alvis	ar manay by						
NOOSE JAR FILES		94			or comes 2 - ghio	ar war ory by						
NOCES HE FLES		91			er somen 2 - akto	an manané jip						
NOCESSAN PRESERVE NOCESS (RE RELOCESS NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME OF TRACTOR	e e e leul Lo	94 estridu distan			er usenn 2 – skin	ar easery by						
NOCESSAN PRESERVE NOCESS (RE RELOCESS NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME CONTRACTOR NAME OF TRACTOR	e e e leul Lo				er osmes 2 – gille	an man wy by						
NOCESSAN PREMAM INCOME JAR FREDS MINE ON RECORDS MORE ON PREMAM INCOME JAR Scholmaling (3)	e la te te la	estricia cluster	Означурася		er norma 2 - alter Seetland + 1970	ar ran ay by						
NOCESSAN PREMAM INCOME JAR FREDS MINE ON RECORDS MORE ON PREMAM INCOME JAR Scholmaling (3)	e la te te la	estricia cluster	Означурася	the	er norma 2 - akter	e. 29° 398						
NOCESSAN PREMAM INCOME JAR FREDS MINE ON RECORDS MORE ON PREMAM INCOME JAR Scholmaling (3)	e las te the la	estricia cluster	Означурася	the	er usenen 2 - aktor Nocalizati + (*rus	-C14-1318						
ROOM PRIME PROOF AN FILSE PRICE OF PRICESS PRICE OF PRICESS PRICE OF PRICESS PRICE OF PRICESS PRICESS Sciences of the Sciences of the		estricia cluster	Означурася	the	er some 2 – skier local net 1 (*Pre	n nonvy by						
BOORDAN HARANA HOODE JAAR FALLOS HARA CONTRACTOR HOLE POWER Schemeling th Schemeling th Schemeling th Schemeling th		estricia cluster	Означурася	the	er somet 2 - skile legel ost 1 ("ru	n. 19~198						
BOORDAN HARANA HOODE JAAR FALLOS HARA CONTRACTOR HOLE POWER Schemeling th Schemeling th Schemeling th Schemeling th		estricia cluster	Означурася	the	er wordt 7 - dele	n. 30°-398						
BOORAN HARANG HOOSE JAR FILLS: HINE CONTROLLING: BOORT/Dow F Scherman (1) Scherman (1) Scherman (1) Scherman (1) P Application app: 2008027	e las te tre la Mi	estricia cluster	Означурася	the	ne mont 2 - dela	n, 34°-398						
Contrast resources and	11111 0 0 10 10 10 10 10 10 10 10 10	nierisis claster	Означурася	the	er onne 2 - dile	a						
Control of the second		nderlicks (Laster telerlicks (Laster	Означурася	the	er anna 2 - dila	no one of by						
Control of the second		n perios cluster exercise Cluster A M M Inteline	Означурася	the	e con 1-obs	n (1999)	Reference of the second s	Alexe	810/a		3	
In COLOMP AND		ndertois claster ndertois Claster ndertois Claster North C		459) : forgering (stories Strigbou	diamy biologiae	neres	Admon	Shinghow		Saultin	
IN DOCUMENT REALMENT REAL OF THE REAL OF T		n estricto closter estricto closter estricto closter marcino closter participante sentence sentence sentence	Stateging 2 Interface Interface Interface Interface	1593 - Reports R	station	drivery Sringige pti	10	914	Shinghipe No	10	Shing"ype	-
BOOLEN FRANKLER BOOLEN FRANKLE		natricka claster natricka claster NATRIA Claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster		ASSIS - Reporter B Solutions So	stantes Serigipo 2	dianay Shafas pi	10	y88 r0	thingSigne real	10	Shing"ype	N
BOOLEN FRANK		natricka claster natricka claster NATRIA Claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster	Bit decent Bit decent Image: Second Seco	1947 - 19	1 1 2 1 2	Channey Strigtype pril pril	10 10 10	948 70 70 70	57193cm	10 10 10	5%ingType 1 0 0 0 0	00 10 10 10
COLUMNIAL AND A COLUMNIAL CONTRACTOR OF CONTRACTOR CONTRACTOR OF CONTR		natricka claster natricka claster NATRIA Claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster	Imaginary management	409 begins 6 409 begins 6 5 5 5 5 5 5 5 5 5 5 5 5 5	stockes Schingliges 2 1 2 1 2 1 2 1	Crimmy Snighge pti pti pti pti pti pti pti	10 10 10 10	999 70 70 70 70	57193pe	10 10 10 10	5ningType 1 0 0 0 0	N
SOCIAL PROVIDENT		natricka claster natricka claster NATRIA Claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster Satricka claster	Badecess Badecess Image: Second	2	100000 20102500 2 1 1 2 1 1	driveway grin	00 00 100 100 100 100	999 70 70 70 70 999	5711g3gau 10 10 10 10 10 10 10	10 10 10 10 10	5tingType 1 0 0 0 0 0 0 0 0 0	00 10 10 10
SOCIALM FRANCESCON CONTRACTOR OF THE SOCIAL SOCIALISTICOCIAL SOCIAL SOCIAL SOCIAL SOCI			Image: constraint of the second sec	409 begins 6 409 begins 6 400 begins 6	stockes Schingliges 2 1 2 1 2 1 2 1	Orienwey Shing/gas pti	10 10 10 10 10 10 10 10	999 70 70 70 999 999	inglipe na na na na na na na na na	10 10 10 10 10 10 10 10	5ningType 1 0 0 0 0	N
SOCIAL PROVIDENT AND		nettis (low k	Badecess Badecess Image: Second	2	status 2nglps 2 1 2 1 2 1 2 1 2	driveway grin	00 00 100 100 100 100	944 no no no 944 944 no 944 944 944	5711g3gau 10 10 10 10 10 10 10	10 10 10 10 10 10 10 10 10	5mglipe 1 0 0 0 0 0 0 2	N N N N N N N N N N N N N N N N N N N
CONTRACT PROVIDENT CONTRACT CONTR			Imaginary Imaginary	2552 + September 45 2552 + Se	stacks Simples 2 1 2 1 2 1 2 3 2 3	Observey StringType pringType pringType <td< td=""><td>00 00 961 00 961 00 961 00</td><td>988 100 100 100 100 100 100 100 1</td><td>Smighpa M M M M M M M M M M M M M M M M M M M</td><td>10 10 10 10 10 10 10 10</td><td>SningTipe 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td><td>N N N N N N N N N N N N N N N N N N N</td></td<>	00 00 961 00 961 00 961 00	988 100 100 100 100 100 100 100 1	Smighpa M M M M M M M M M M M M M M M M M M M	10 10 10 10 10 10 10 10	SningTipe 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	N N N N N N N N N N N N N N N N N N N

Databricks DB			Connection	Test Cluster Ty	oe-databricks	×	
SCHEMAS Q Search ▼ E default ▶ ∰ Tobles	Ø	50C L6Hw 1		Limit to 100 -	ows v (•
SCHEMA INFO Schema: default							

Databricks DB		Connection :	Test Cluster Type-databricks	×	
SCHEMAS C Brench Second Sec	0 }	90, fater	Limit to 100 rows	*	
SCHEMA INFO Table: cars_csv					

Databricks DB				Connection : Test Cluster Type-databr	icks 💉
SCHEMAS	C SQL Editor			Limit to 100 rows 💌	(† (III)
Q Search ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓		00M cars_csv LIMIT 10			
	cl	ci	ci	c4	
	1	0.0	2.3	3.0	
	2	10	3.0	2.0	
SCHEMA INFO	3	0.0	u	1.0	
Table: cars_csv	4	0.0	4,1	5.0	
	5	0.0	3.1	6.0	
	6	1.0	2.1	2.0	

Running DDL Commands

Fire Insights allows you to run DDL commands on Databricks.

With this one can:

- Create New Databases
- Create New Tables
- View the schema of the tables
- And many more

Go to DATABROWSERS/Databricks DB. Then click on DDL.

Databricks has a good page on Creating New Tables:

https://docs.databricks.com/spark/latest/spark-sql/language-manual/create-table.html

Below are example of running DDL

Creating Table

• DDL Statement:

```
CREATE TABLE `employee` (`id` INT, `name` STRING) USING com.databricks.spark.csv.

→OPTIONS (`multiLine` 'false', `escape` '"', `header` 'true', `delimiter` ',',

→path 'dbfs:/FileStore/tables/employee.csv' );
```

Location of the data could be changed to S3 location.

mection :	global connection Type-databricks 😆
	DDL
PE DATABRI	XX DDL COMMAND HERE:
	ex de commune riske. E 'employee' ('Id' INT, 'name' STRING) USING com.databricks.spark.csv OPTIONS ('multilune' false; 'escape' '', 'header' 'true, 'delimiter' ", path 'dbfs:/FileStore/tables/employee.csv/);
	- and all of a well and a second second second control and a second second second second second second second s
ECUTE COP	
	xxxxxx xuccent/ly
KICUTI COP	

Running SQL

• Select SQL Statement:

select count(*) as count from employee;
	Databricks DB Connection : Type-databricks 2
	DATABASES DOL TYPE DATABASES DOL COMMAND HEEE:
	select count(*) as count from <u>employage</u> :
	EXECUTE COMMANDS
	3

Sample Data:

• Select SQL Statement:

select * from employee;

By default first 100 rows of data is displayed.

Databricks DB			
Connection : global connection Type-databricks	¢		
DATABASES DDL			
TYPE DATABRICKS DDL COMMAND HERE:			
EXECUTE COMMANDS			
	name		
	nome 600		
id			

Drop Table

• Drop Statement:

drop table employee;

Viewing Databricks Clusters

Fire Insights enables you to view your Databricks Clusters. You can also Start and Stop the Databricks clusters from Fire Insights.

Databricks DB		
Connection : global connection Type-databricks ¢		
DATABASES DOL		
TYPE DATABRICKS DDL COMMAND HERE:		
		6
EXECUTE COMMANDS		
Table dropped successfully		

Go to Data Browsers/Databricks Clusters

It will display the various Databricks Clusters available.

Databricks Cluste	ers						
Connection : Test Cluster	Type-databricks 👻						
CLUSTER ID	CLUSTER NAME	CLUSTER SOURCE	STATE	START TIME	TERMINATED TIME	LAST STATE LOSS TIME	ACTIONS
0826-092903-ook501	Test	UI	RUNNING	2020/08/26 14:59:03 IST	null IST	2020/08/27 17:43:24 IST	۲

If you want to see Cluster Details, Click on CLUSTER NAME, it will display all informations.

Connection : Test Clush	ers r Type-databricks						
CLUSTER ID	CLUSTER NAME	CLUSTER SOURCE	STATE	START TIME	TERMINATED TIME	LAST STATE LOSS TIME	ACTIONS
0626-092903-oak501	Test	UI	RUNNING	2020/08/26 14:59:03 IST	null IST	2020/08/27 17:43:24 IST	۲

You can also Start and Stop the Databricks clusters from Fire Insights, using ACTIONS button.

Browse DBFS

Fire Insights enables you to browse your DBFS & UPLOAD FILE & Delete file and directory in DBFS.

Go to DATA BROWSERS/DBFS

It will display the Databricks File System list page.

UPLOAD FILE in DBFS

You can upload file in DBFS from local pc.

On clicking on UPLOAD FILE button, it will ask you to select file from local pc and UPLOAD.

On successful UPLOAD, it will show successful informations and file can be viewed inside the folder in DBFS.

🌢 Fire Insights 🛛 🖓 DATA BROWSERS	- ## APPLICATIONS -	🔿 SCHEDULER 🕘 🐵 EXECUTIONS - 🔹 MODELS - 📑 DATA QUALITY - 😝 ADMINISTRATION - 🛛	PROCESSO
Databricks Clusters	Cluster Details		1
	SPARK CONTEXT ID :	6133712372202850000	
Connection : Test Cluster Type-data	SPARK VERSION :	6.4.x-scala2.11	
CLUSTER ID CLUSTER NAM	NODE TYPE ID :	m4.large	TIME
0826-092903-oak501 Test	DRIVER NODE TYPE ID :	m4.large	43:24
	AUTOTERMINATION MINUTES	15	
	ENABLE ELASTIC DISK:	false	
	STATE MESSAGE :		
	JDBC URL:		5-
			_
	MIN WORKERS :		_
	MAX WORKERS :		_
	JDBC PORT :	10000	_
	CREATOR USER NAME :		_
			ж

Databricks Clusters

Connection : Test Cluster	Type-dotabricks						
CLUSTER ID	CLUSTER NAME	CLUSTER SOURCE	STATE	START TIME	TERMINATED TIME	LAST STATE LOSS TIME	ACTIONS
0826-092903-ook501	Test	UI	RUNNING	2020/08/26 14:59:03 IST	null IST	2020/08/27 17:43:24 IST	<mark>©</mark>

Databricks File System

#Home		UPLOAD FILE Connection :	Test Cluster T	ype-databricks 👻
NAME	IS DIRECTORY	PATH	SIZE	ACTION
C /FileStore	true	/FileStore		0
🗅 /databricks-results	true	/databricks-results		0
🗅 /modelsavepath	true	/modelsavepath		0
🗅 /steelmodelpath	true	/steelmodelpath		0
🗅 /steelplatemodel	true	/steelplatemodel		•

Databricks File System

#Home		UPLOAD FILE Connection :	Test Cluster T	ype-databricks 👻
NAME	IS DIRECTORY	PATH	SIZE	ACTION
C /FileStore	true	/FileStore		8
🗅 /databricks-results	true	/databricks-results		0
🗀 /modelsavepath	true	/modelsavepath		0
🗅 /steelmodelpath	true	/steelmodelpath		0
🗅 /steelplatemodel	frue	/steelplatemodel		0

🕘 Fire Insights 🛛 🖓 DATA BROWSERS 👻	III APPLICATIONS * @ SCHEDULER * @ EXE	CUTIONS 👻 💩 MODELS 👻 🧮 DATA QUALITY 👻 😁 ADMI	NISTRATION	 PRO 	CESSORS *	o;- 0- III-
Databricks File System	Upload File		×			
Home /FileStore/tables	Choose File cors.csv			Connection :	Test Cluster Ty	pe-databricks 👻
NAME					SIZE	ACTION
Clickthru.cov		UPLOAD	CLOSE		2989	0
hike_sharing_sample_dataset.csv	false	/FileStore/tables/bike_sharing_sample_dataset.csv			608761	0

🌢 Fire Insights 🖓 DATA BROWSERS - III APPLICATIONS -	© sc	HEDULER - © EXECU	itions - 🖓 models -	🖹 DATA QUALITY 👻		in - 🛢 proc	ESSORS +	o:- ⊖- ≡- 4
Databricks File System		Message		×				
Home /FileStore/tables		File uploaded suc	cessfully		UPLOAD FILE	Connection :	Test Cluster Type	e-dotabricks 👻
NAME	IS DIRE			ox			SIZE	ACTION
Chickthra.cvv	false						2989	0
hike_sharing_sample_dataset.csv	false		/FileStore/tables/bike_sha	ing_sample_dataset.csv			608761	
C /cars.ev	false		/FileStore/tables/cars.csv				264	

Databricks File System					
Home /FileStore/tables			UPLOND FILE Connection :	Test Cluster Type	e-databricks 🛩
NAME	IS DIRECTORY	PATH		SIZE	ACTION
Clickthru.csv	folse	/FileStore/tables/Clickthru.csv		2989	8
hike_sharing_sample_dataset.csv	folse	/FileStore/tables/bike_sharing_sample_dataset.csv		608761	0
D /cars.cov	false	/FileStore/tables/cars.csv		264	0

Delete file and directory in DBFS

You can delete file and directory in DBFS using delete ACTION button.



On successful deletion, it will show successful informations and file can be viewed inside the folder in DBFS.

Databricks File System		Success	×					
#Home /FileStore/tables		Successfully delete	d the file.	UPLOAD FILE	Connection :	Test Cluster Ty	ype-databricks 🗸	
NAME	IS DIRE	b	ок			SIZE	ACTION	
Clickthrucev	false					2959	0	
🗅 /bike_sharing_sample_dataset.csv	faise		/FileStore/tables/bike_sharing_sample_dataset.csv			608761	8	

Reading Databricks Tables

Fire Insights enables you to read from and write to Databricks tables.

Below is a workflow which reads data from the Databricks table xyz. It then processes the data and finally writes out the result to the Databricks table abc.

Read Databricks table in Workflow

In the workflow use the processor 'ReadDatabricksTable'. It will allow you to read tables from Databricks.

Then use the other processors in Fire for processing the data read from the Databricks Table.

Workflow



Processor Configurations for ReadDatabricksTable

Refresh schema for processor ReadDatabricksTable

ReadDatabricksTablez				
ITPUT STORAGE LEVEL : 😡	DEFAULT		T	
TABRICKS DATABASE : O	default		BROWSE DATABRICKS DB	
TABRICKS TABLE : 😡	flights_data_csv		BROWSE DATABRICKS TABLE	
TABRICKS QUERY (OPTIONAL) : O				
1				
			I	OK CANC
ReadDatabricksTablez	uffonkristikk		✔ Schema Re	
ReadDatabricksTablez Orean	utionalestation		V Schema Ru	
ReadDatabricksTablez Orienter	iffundestation		🖍 Scherna Ri	
		COLUMN FORMA		ofreshed.
EMA COLUMNSE : Q REFREEN SCHEMA Q		COLUMN FORMA		

Processor executions for ReadDatabricksTable

ReadDatabricks	Table										
Reading Databricks t	table										
Row Values											
Row Values											
DAY_OF_MONTH	DAY_OF_WEEK	CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_D
IntegerType	IntegerType	StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	Integer
	3	AA	N338AA	1	12478	JFK	12892	LAX	900	914	14
1	3	АА	N339AA	2	12892	LAX	12478	JEK	930	1132	122
1	3	AA	N335AA	3	12478	JEK	12892	LAX	1200	1157	0
1	3	АА	N367AA	5	11298	DFW	12173	HNL	1305	1307	2
1	3	AA	N364AA	6	13830	ogg	11298	DFW	1755	1753	0
1	3	АА	N364AA	7	11298	DFW	13830	OGG	1200	1205	5
								0.014			

Databricks Workflow execution

Below is the output of executing the above workflow which reads data from a Databricks table.

Writing to Databricks Tables

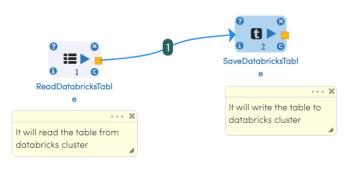
Fire Insights enables you to write to Databricks tables.

In the workflow use the processor 'SaveDatabricksTable'. It will allow you to save data to tables to Databricks. Below is a workflow which writes data to the Databricks table test_save.

Workflow

Processor Configurations for SaveDatabricksTable

RK-SUBMIT-CONF: 🖗											
		eg:exe	cutor-memo	ry 2gexecut	tor-cores 2driver-me	mory 2g					
IGRAM PARAMETERS:	Ð										
DOSE JAR FILES: 🛛		🗏 SPARKJI	DBC41.JAR								
ALL ON SUCCESS: @											
VIL ON FAILURE: 0											
rkflow Result	Logs										
Submitting the Job (to the Databricks c	luster									
	to the Databricks c	luster									
Submitting the Job (to the Databricks c	luster									
Submitting the 7ab (Row Values Row Values	to the Databricks c		TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	DEST_AIRPORT_ID	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY
submitting the 7ab (Row Values Row Values		CARRIER		FL_NUM IntegerType	ORIGIN_AIRPORT_ID IntegerType		DEST_AIRPORT_ID	DEST StringType	CRS_DEP_TIME IntegerType	DEP_TIME IntegerType	
Submitting the Job (Row Values Row Values DAY_OF_MONTH	DAY_OF_WEEK	CARRIER									DEP_DELAY IntegerType 14
Submitting the Job 1 Row Values Row Values DAY_OF_MONTH IntegerType	DAY_OF_WEEK	CARRIER StringType	StringType	IntegerType	IntegerType	StringType	IntegerType	StringType	IntegerType	IntegerType	IntegerType
Submitting the Job f Row Values Row Values DAY_OF_MONTH IntegerType 1	DAY_OF_WEEK IntegerType 3	CARRIER StringType AA	StringType N338AA	IntegerType	IntegerType 12478	StringType JFK	IntegerType 12892	StringType LAX	IntegerType 900	IntegerType 914	IntegerType
Submitting the Job f Row Values Row Values DAY_OF_MONTH IntegerType 1	DAY_OF_WEEK IntegerType 3 3	CARRIER StringType AA AA	StringType N338AA N339AA	IntegerType 1 2	IntegerType 12478 12892	StringType JFK LAX	IntegerType 12892 12478	StringType LAX JFK	IntegerType 900 930	IntegerType 914 1132	IntegerType 14 122
Soluting the 366 (Row Values Row Values DAY_OF_MONTH IntegerType 1 1 1	DAY_OF_WEEK IntegerType 3 3 3	CARRIER StringType AA AA AA	StringType N338AA N339AA N335AA	IntegerType 1 2 3	IntegerType 12478 12892 12478	StringType JFK LAX JFK	IntegerType 12892 12478 12892	StringType LAX JFK LAX	IntegerType 900 930 1200	IntegerType 914 1132 1157	IntegerType 14 122 0
Soluting the 366 (Row Values Row Values DAY_OF_MONTH IntegerType 1 1 1	DAY_OF_WEEK IntegerType 3 3 3 3 3 3	CARRIER StringType AA AA AA AA	StringType N338AA N339AA N335AA N367AA	IntegerType 1 2 3 5	IntegerType 12478 12892 12478 11298	StringType JFK LAX JFK DFW	IntegerType 12892 12478 12892 12892 12173	StringType LAX JFK LAX HNL	IntegerType 900 930 1200 1305	IntegerType 914 1132 1157 1307	IntegerType 14 122 0 2
Additing the 24b is A Row Values Row Values DAY_OF_MONTH IntegerType 1 1 1 1 1 1 1	DAY_OF_WEEK IntegerType 3 3 3 3 3 3 3 3 3 3	CARRIER StringType AA AA AA AA AA	StringType N338AA N339AA N335AA N367AA N367AA	IntegerType 1 2 3 5 6	IntegerType 12478 12892 12478 11298 113830	StringType JFK LAX JFK DFW OGG	IntegerType 12892 12478 12892 12173 11298	StringType LAX JFK LAX HNL DFW	IntegerType 900 930 1200 1305 1755	IntegerType 914 1132 1157 1307 1753	IntegerType 14 122 0 2 0 2 0
Additing the 24b is A Row Values Row Values DAY_OF_MONTH IntegerType 1 1 1 1 1 1 1	DAY_OF_WEEK IntegerType 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	CARRIER StringType AA AA AA AA AA AA	StringType N338AA N339AA N335AA N367AA N364AA N364AA	IntegerType 1 2 3 5 6 7	IntegerType 12478 12892 12478 11298 13830 11298	StringType JFK LAX JFK DFW OGG DFW	IntegerType 12892 12478 12892 12173 11298 13830	StringType LAX JFK LAX HNL DFW OGG	IntegerType 900 930 1200 1305 1755 1200	IntegerType 914 1132 1157 1307 1753 1205	IntegerType 14 122 0 2 0 5
Additing the 24b is A Row Values Row Values DAY_OF_MONTH IntegerType 1 1 1 1 1 1 1	DAY_OF_WEEK IntegerType 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	CARRIER StringType AA AA AA AA AA AA AA AA	StringType N338AA N339AA N335AA N367AA N364AA N364AA N372AA	IntegerType 1 2 3 5 6 7 8	IntegerType 12478 12892 12478 11298 13830 11298 1298 12173	StringType JFK LAX JFK DFW OGG DFW HNL	IntegerType 12892 12478 12892 12478 12892 12173 11298 13830 11298	StringType LAX JFK LAX LAX HNL DFW OGG DFW	InlegerType 900 930 1200 1305 1755 1200 1800	IntegerType 914 1132 1157 1307 1753 1205 1839	IntegerType 14 122 0 2 0 5 39



NAME	Timestamp	UserId	IP Address	Product Id	
PE	string	string	string	string	
ORMAT					
NUTPUT STORAGE LEVEL : 😡		DEFAULT		*	
DATABRICKS DATABASE * : 😡		default		BROWSE DATABRICKS DB	
ATABRICKS TABLE * : O		Test_save		BROWSE DATABRICKS TABLE	
ARTITION BY : O					
DRMAT : 😡		CSV		•	
WE MODE : O		Append		•	

Databricks Workflow execution

Below is the output of executing the above workflow which saves the data to Databricks table.

WORKFLOW NAME: Ø			SaveDatabricksTable								
SPARK-SUBMIT-CONF: 0											
				ag:executor-memory 2gexecutor-cores 2driver-memory 2g							
PROGRAM PARAMETERS: Ø											
сно	DOSE JAR FIL	is: 0		SPARKJDBC4LJAR							
EMV	AIL ON SUCCI	iss: 0									
EMV	ALL ON FAILUI	IE: 0									
Wo	rkflow	Result	Logs								
•											
	Submitting	the Job to	the Databricks clast	ar.							
	ReadDo	tabricksTa	ble								
	Descuting	Databricks	weny : select * from	<pre># default.clickthrw_csv</pre>							
(Output	Schema									
	Executing	Node fire.	nodes.databricks.N	lodeSaveDatobrick:Table : 2 : Mar 12, 2020 7:12:51 AM							
	Input Sc	hema									
-	Output	Schema									
-											
	Successfully finished executing the workflow										

• Verify the Table

Table: test_save		
test_save 2 Refresh		
Databricks_cluster		
Schema:		
col_name -	data_type	comment
Timestamp	string	null
UserId	string	null
IP Address	string	null
Product Id	string	null

Sample Data:									
Timestamp	Userld -	IP Address	Product Id						
9:03 AM	275	207.51.113.192	1						
12:57 AM	586	62.34.98.94	2						
2:45 AM	508	20.237.172.182	3						
2:13 PM	378	69.215.255.150	4						
9:27 AM	965	56.101.183.251	5						

File Formats

The tables can be saved into CSV, JSON, Parquet and ORC file formats.

If the file format is not specified, the data in tables is stored in Parquet format.

Reading S3 files

https://docs.databricks.com/_static/notebooks/data-import/s3.html

There are two ways in Databricks to read from S3. You can either read data using an IAM Role or read data using Access Keys.

Databricks recommends leveraging IAM Roles in Databricks.

Fire Insights allows you to browse your Data in S3 and create workflows using them. When the job is submitted to Databricks, the job reads data from the S3 location and processes them.

You can also create external tables in Databricks over data in S3. Fire Insights can process data from Databricks tables.

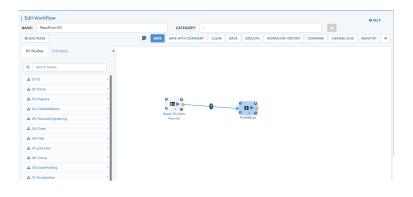
Accessing S3 buckets from Databricks

This document from Databricks has very good information on the setup for accessing S3 buckets from Databricks. https://docs.databricks.com/security/credential-passthrough/iam-passthrough.html

Read the data from S3 in Workflow

In Sparkflows, user can read the data from S3 location using processors like ReadCSV, ReadParquet, ReadJson etc.

Workflow



Browse S3 Path and Refresh schema for processor ReadCSV

	DEFAULT		0	f
ath • : Ø	s3a://sparkflow-sample-d	ata/data/Clickthru.csv	BROWSE \$3	
IPARATOR : O				
EADER : @	frue		\$	
ROP MALFORMED : 0	false		•	
CHEMA COLUMNS : O REFRESH SCHEMA				
OLUMN NAMES FOR THE CSV O	COLUMN TYPES FOR THE CSV O	COLUMN FORMATS FOR	THE CSV O	
Timestamp	STRING	¢ format		۰
	STRING	¢ format ¢ format		•

Workflow executions Results

Writing to S3 files

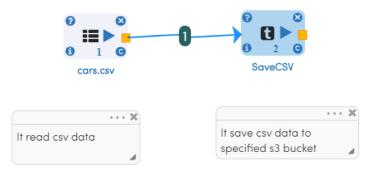
https://docs.databricks.com/_static/notebooks/data-import/s3.html

Fire Insighs workflows can write data to S3 locations.

ixecuting Node fire.nodes.dataset.NodeDatasetCSV : 1 : Apr 16, 2020 4:13:48 AM									
Output Schema									
ixecuting Node fire.node	xeculing Node fire nodes util NodePrintFirstNRows : 2 : Apr 16, 202 413:49 AM								
Input Schema									
Row Values	Userid	IP Address	Product Id						
tow Values Timestamp	Userid IntegerType	IP Address StringType	Product Id IntegerType						
tow Values Timestamp StringType									
tow Values Timestamp StringType 9:03 AM	IntegerType	StringType	IntegerType						
Row Values Row Values Timestamp StringType 9:03 AM 12:57 AM 2:45 AM	IntegerType 275	StringType 207.51.113.192	IntegerType 1						

Below is an example workflow which writes data to S3. When the workflow is executed, the Dataframe is saved to the S3 location.

In the dailog box of the save CSV processor the path is specified as s3a://sparkflow-sample-data/write/



Browse S3 specified Path & other parameter for processor SaveCSV

AME	c1	c2		c3	c4
PE	integer	doubl	le	double	double
ORMAT					
TPUT STOR	RAGE LEVEL : 🔞		DEFAULT	•	
H*:00			s3a://sparkflow-sample-data/write/		BROWSE HDFS BROWSE S3 VIEW FILE
E MODE : 🕻	0		Append	v	
ADER : 😡		true			

Execution Result

Once the above workflow successfully completed, the save data can be viewed using DATABROWSERS/AWS S3 Location with specified path

16.1.5 Troubleshooting Fire/Databricks Integration

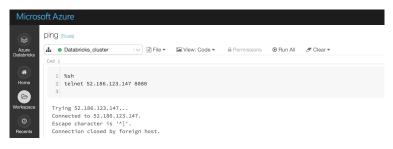
http://ip-172-31-71-252.ec2.internal:4041						
Executing Node fire.nodes.dataset.Nod	eDatasetCSV : 1 : Apr 9, 2020 8:02:14 AM					
Output Schema						
Executing Node fire.nodes.save.NodeSaveCSV : 2 : Apr 9, 2020 8:02:15 AM						
Input Schema						
Output Schema						
Successfully finished executing the work	flow					
ne s3a://sparkflow-sample-data/write/						
E	PATH	SIZE	IS DIRECTORY	OWNER	GROUP	LAST UPDATED
CCESS	write/_SUCCESS	0 bytes	false	support	d41d8cd98f00b204e9800998ecf8427e	04/09/2020 at 1:32
00000-2bbc7cb2-fd96-41c7-afcc-6009f4f20a0c-c000.csv	write/part-00000-2bbc7cb2-fd96-41c7-afcc-6009f4f20a0c-c000.csv	264 bytes	false	support	49a113f4421549cc2ab046497e4a80b5	04/09/2020 at 1:32

When the workflow is executed, nothing shows up in Fire

One problem might be that the postbackURL is not configured right in Fire Insights under Administration/Configuration.

The other problem can be that the machine running Fire Insights is not accessible from the Databricks Cluster. Test connectivity to the Fire Insights machine from Databricks.

Connectiving from Databricks to Fire postbackURL can be done in Databricks via Notebooks using the telnet command.



When the workflow is executed, nothing shows up in Fire

Another reason might be that you are using the Databricks High Concurrency cluster. Ensure that you are connecting Fire to Databricks Standard cluster.

When accessing most Databricks pages in Fire, it gives Simba JDBC error

The reason for it is that the Databricks Simba JDBC jar file is not deployed in Fire.

https://docs.sparkflows.io/en/latest/databricks/databricks-installation.html#install-databricks-jdbc-driver

In the workflow editor, it shows 'Cannot connect to Fire'

Ensure that under Administration/Configuration, app.runOnCluster is set to false.

Checking the cluster logs in Databricks

There are times when it is helpful to look at the Cluster logs in Databricks when running Fire with Databricks.

The following logs under Driver Logs are useful:

• log4j-active.log

Search for WorkflowExecuteDatabricks in the logs to view if the Fire Insights Job is running in Databricks.

java.lang.Exception: An error occurred while initializing the REPL. Please check whether there are conflicting Scala libraries of at com.databricks.backend.daemon.driver.DatabricksILoop\$class.initSpark(DatabricksILoop.scala:98)

This error can happen when running spark 2.3 version of Fire with spark 2.4 cluster on Databricks. Either upgrade Fire to spark 2.4 version, or create another Databricks cluster which supports spark 2.3.

Databricks Cluster Versions Support

Databricks Runtime Version Spark Version Scala Version

6.2 2.4.4 2.11
6.3 2.4.4 2.11
6.4 2.4.5 2.11
6.5 2.4.5 2.11

CHAPTER 17

AWS Integration

17.1 AWS Guide

17.1.1 Introduction

Fire Insights is the flagship product from Sparkflows. It is seamlessly integrated with AWS. With Fire Insights you can perform self-serve data processing, analytics and machine learning on AWS.

Fire Insights integrates with EMR, S3, Redshift, SageMaker, HIVE and Kinesis.

Fire Insights comes with a number of components including:

- Workflow Editor : To create workflows for data processing, analytics and machine learning.
- **260+ Processors** : These include reading data from various stores, data processing, machine learning and visualizations.
- Execution Engine : For executing the workflow on EMR
- · Scheduler : For scheduling running the workflows at certain time intervals

Sparkflows Fire Insights can be deployed to an existing Amazon EMR cluster, or you can use one of our CloudFormation templates to set up a new Amazon EMR Cluster. If you use our provided CloudFormation templates we'll create an EMR cluster for you or even an EMR cluster and MySQL instance running in RDS, depending on which template you choose.

Pre-requisites and Requirements

Fire Insights needs EMR for running the workflows. So, you need a running EMR cluster for using Fire Insights.

You also need ssh access to one of the machines of the EMR cluster for installing Fire Insights. This machine is typically an edge node or a master node of the EMR cluster.

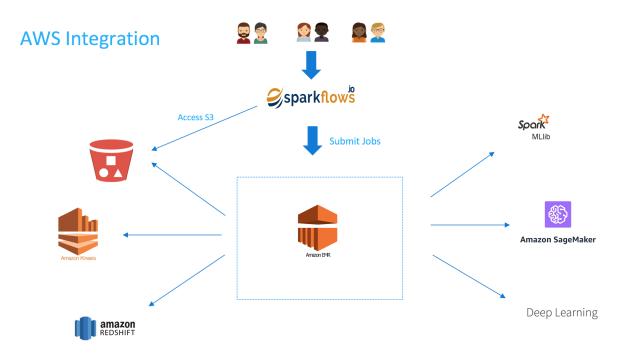
· Getting started with EMR - https://aws.amazon.com/emr/getting-started/

 Opening SSH access to the EMR master node - https://docs.aws.amazon.com/emr/latest/ManagementGuide/ emr-connect-master-node-ssh.html

Architecture

Fire Insights runs on the edge node or one of the master nodes of the EMR cluster. It submits the processing jobs onto the cluster. By default it runs on port 8080. This port needs to be changed to some port which is available on the machine as it is in use by default. Lets assume we will use port 8085.

When the jobs are fired onto the EMR cluster, it can read/write data from S3/HDFS/Redshift/Kinesis. It can also fire Machine Learning modeling jobs to SageMaker.



17.1.2 Planning Guide

This document describes details to help you plan on deploying and using Fire Insights on AWS.

Security

Fire Insights is installed onto the edge node or master node of the EMR cluster. The jobs fired by the users would be able to access and process data on S3, HDFS, Redshift, Kinesis.

Costs

The main costs involved when using Fire Insights are around the EMR cluster. EMR cluster has master nodes and workflow nodes.

Pricing for EMR can be found here : https://aws.amazon.com/emr/pricing/

The more processing capacity needed, the larger should be the size of the EMR cluster.

Fire Insights can also run Machine Learning Modeling jobs onto SageMaker. If this is used, there would be cost associated with using AWS SageMaker. Amazon SageMaker Pricing details are here : https://aws.amazon.com/sagemaker/ pricing/

Sizing

EMR cluster normally starts with a mimumum of 1 master node and 2 worker nodes.

We recommend using at least 16GB machines for the master and worker nodes.

As your data volume and the number of concurrent users increases, we recommend increasing the size of the EMR cluster. Memory for the worker nodes can be increased to 32GB to 64GB to 512GB. Since Apache Spark has the ability to use as much memory you provide, its a good idea to give it more memory.

Same goes for the number of disks and vcores.

17.1.3 Deployment Guide

Fire can be easily installed on an AWS EMR Cluster. Fire can be installed on the master node of an EMR cluster. It would then submit the jobs to the EMR cluster.

Below are the overall steps for installing Fire Insights on EMR.

- ssh into the Master node
- Download Fire Insights from https://www.sparkflows.io/download
- Unzip it
- Create H2 Database
- Start Fire

Steps

• Start your EMR cluster on AWS:

Start your EMR cluster on AWS **if** you do **not** already have it running.

• Update the inbound rules for the Master Node:

```
We would have Fire listening on ports 8085 and 8086
Fire by default listens on 8080 and 8443. But EMR clusters have other processes.
→listening on these ports.
So we will later change it to listen on ports 8085 and 8086
Update the inbound rules for the Master Node to allow ports 8085 and 8086
```

• ssh into the Master EMR node as the hadoop user:

ssh -i my.pem hadoop@ec2-xx-yyy-zz-aaa.compute-1.amazonaws.com

- Download the fire tgz file by one of the following options:
 - https://www.sparkflows.io/download OR
 - https://www.sparkflows.io/archives OR
 - wget https://s3.amazonaws.com/sparkflows-release/fire/rel-x.y.z/2/fire-x.y.z.tgz

• Unpack it:

```
tar xvf fire-x.y.z.tgz
```

· Copy hadoop-lzo.jar:

```
cp /usr/lib/hadoop-lzo/lib/hadoop-lzo.jar /home/hadoop/fire-3.1.0/fire-user-lib
```

• Configure Fire to listen on ports 8085 and 8086:

```
    cd <fire install_dir>
    Edit conf/application.properties
    Update the last two lines to below:
http.port=8085
https.port=8086
```

• Create H2 DB:

• Launch Fire Server:

cd <fire install_dir>
./run-fire-server.sh start

• Open your web browser and navigate to:

```
<machine_name>:8085/index.html
```

• Login with the following default username and password:

```
username : admin
password : admin
```

• Connect Fire with the EMR Cluster:

```
Go to Administration/ConfigurationClick on 'Infer Hadoop Configs'Save
```

- If your EMR cluster **is not** running HIVE, update 'spark.sql-context = SQLContext'

• Create the hadoop user in Fire:

```
- Under Administration/Users, add the 'hadoop' user
```

Loading Example Workflows

- From the home page of Fire Insights, click on *Load Example Applications*
- Upload the Fire examples data onto HDFS:

```
cd <fire install_dir>
hadoop fs -put data /tmp
```

Install and Running Example Workflows

• Start off with executing the example workflows:

```
    Fire comes pre-packaged with a number of example workflows
    You can install them by clicking on the 'Install example workflows' link in the_
    Ianding page when logged in as the `admin` user.
```

- · Logout from the current session and login again with the 'hadoop' user
 - Execute the workflows

Adding a new user

Create the home directory on HDFS for the new user.

For example, for user 'test':

- · hadoop fs -mkdir /user/test
- hadoop fs -chown test:test /user/test

Create the user in Fire Insights if not already created.

Extra configuration for running PySpark

EMR needs extra configurations when running PySpark. In the below the python 3.6 virtual environment is installed in the directory /home/hadoop/venv

- export SPARK_HOME=/usr/lib/spark/
- export PYSPARK_PYTHON=/home/hadoop/venv/bin/python
- export YARN_CONF_DIR=/etc/hadoop/conf

17.1.4 S3 Integration

Fire Insights allows you to access your files on S3. This page describes S3 integration of Fire.

We recommend controlling access to S3 using IAM Roles.

- Run Fire Insights on an EC2 machine with the appropriate S3 IAM Role.
- Run the EMR cluster with the appropriate S3 IAM Role.

If you are running Fire Insights on a independent machine, you can also use aws configure to set the AWS Access Key and Secret Access Key on the machine.

AWS CLI S3 Reference : https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html

Installing aws cli

- http://docs.aws.amazon.com/cli/latest/userguide/installing.html
- pip install awscli -upgrade -user

Configuring AWS access key and password

Run aws configure to configure your credentials on the machine on which Fire Insights is running.

Access S3 in fire-ui

In Fire Insights, you can browse S3 under the menu Browser/AWS S3.

Fire Insights	街 DATA BROWSERS 👻	② APPLICATIONS		▼		III PROCESSORS -	≡ • o\$ • ≜ •
	I HDFS						
Dashboard	I HIVE Tables						
	I AWS S3		\frown				
놀프		16	\mathcal{O}		2 302		241
-6-	А	PPLICATIONS !	VD	EXECUTED WORKFLOWS			PROCESSORS !

• Click on AWS S3 to view the files on S3.

• Fire Insights	쉽 DATA BROWSERS 🝷	⊘ APPLICATIONS ▼		ED 🔹 🕞	WORKFLOW EXECUTIONS 🝷	伯 ADMINISTR	ATION - II	PROCESSORS •	•	o; -	. -
AWS S3											
NAME			PATH	SIZE	IS DIRECTORY	OWNER	GROUP	LAST UPDATED			
aws-glue-scripts-0043	331324847-us-east-1			0 bytes	true	support		01/01/2019 at 4:0	DIAM		

Protecting Data Using Server Side Encryption

Data encryption settings on S3 buckets: https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption. html

REFERENCE : Creating Access Key & Secret Key

- 1. You'll need create a user with programmatic access by following the steps here (https://docs.aws.amazon.com/ IAM/latest/UserGuide/id_users_create.html).
- 2. Next, you'll create an IAM policy that defines what this user has access to in your AWS account. It's important to only grant this user minimal access within your account. See this documentation for how to create IAM policies (https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html).
- 3. Finally, you'll create an access key and secret key for this user (https://docs.aws.amazon.com/IAM/latest/ UserGuide/id_credentials_access-keys.html#Using_CreateAccessKey).

Note It's important to regularly rotate your access and secret keys. See this documentation for more information (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#rotating_access_keys_console)

17.1.5 Testing Fire Insights on AWS

After you have deployed Fire Insights on AWS, it is a good idea to test the things.

Below are a few good steps for achieving it:

- Ensure you can log into the sytem
- View the Sample Applications
- Execute a workflow on EMR

Log into the System

• Log into the system as the hadoop user which you had created during the Deployment process. * http://docs. sparkflows.io/en/latest/aws/running-on-emr.html

View the Sample Applications

- Go to the "Applications/List' page.
- If you loaded the Sample Applications during the deployment process you would see a number of Sample Applications listed.
- Click on any of them to view their Datasets/Workflows etc.

Execute a workflow on EMR

- From the Applications/Workflows page.
- Click on the Execute icon next to any workflow
- This will open up the Execute page.
- Click on Execute to execute the workflow on the EMR cluster
- The results of execution would get displayed on the page.

17.1.6 Operational Guide

This document describes details for operating Sparkflows when running on AWS.

Onboarding New Users

New users can be created in Sparklows by logging into it. Then go to Administration/Users.

Health Check

The main server process which handles the web requests is fire-ui. This is a long running process and very stable. This process can be checked for responsiveness for any health checks.

Backup and Recovery

Fire Insights stores the metadata into a Relational Database.

It comes with an embedded H2 database. It scales well for pretty heavy loads and upto 50 users. Sparkflows can be easily configured to run with an MySQL database.

When running with H2 database, Sparkflows by default stores the db files in the user home directory which is running Sparkflows. There are 2 files:

- firedb.mv.db
- firedb.trace.db

For backup, just copying these files to a backup location is enough. There is no need to stop Sparkflows. It is a good idea to copy it to another maching.

When running with MySQL running on the same or different machine, the MySQL database named fire needs to be backed up.

Routing Maintenance

Apart from backups of the database, Fire does not need much of routine maintenance.

Fire stores the details of the job executions in the relational database. Over time, you may have too many jobs executed. Deleting old jobs from the Workflow Executions page is a good idea so as not to fill up the database too much. But it has the ability to handle millions of jobs, so you do not have to worry too much about it.

Support

For support, you can contact Sparkflows at support@sparkflows.io. We will guide you through the process.

Sparkflows can also support you though Zendesk tickets. Get in touch with us for guidance and setup.

17.1.7 Copying files to S3 with aws-cli

There would be times when you want to upload multiple files from your laptop to S3. This document describes the process for it.

Installing aws-cli on mac

brew install awscli

Configure AWS Credentials

aws configure:

```
– Enter your awsAccessKeyId
– Entery your awsSecretAccessKey
```

View S3 Buckets

• aws s3 ls

View S3 Directory

• aws s3 ls s3://bucket_name/dir1/

Copy files to S3

Copy all files from local_direcory to s3://bucket-name/dir1:

aws s3 cp local_directory s3://bucket-name/dir1 --recursive

Delete All Files in Directory

• aws s3 rm s3://bucket_name/dir1/ -recursive

Setting Roles and Policies for EMR

In order to be able to access S3 files from the EMR cluster, attach the AmazonS3FullAccess Policy to the EMRDe-faultRole.

Now the EMR cluster would have access to the S3 buckets.

REFERENCE : Creating Access Key & Secret Key

- 1. You'll need create a user with programmatic access by following the steps here (https://docs.aws.amazon.com/ IAM/latest/UserGuide/id_users_create.html).
- Next, you'll create an IAM policy that defines what this user has access to in your AWS account. It's important to only grant this user minimal access within your account. See this documentation for how to create IAM policies (https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html).
- 3. Finally, you'll create an access key and secret key for this user (https://docs.aws.amazon.com/IAM/latest/ UserGuide/id_credentials_access-keys.html#Using_CreateAccessKey).

Note It's important to regularly rotate your access and secret keys. See this documentation for more information (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#rotating_access_keys_console)

17.1.8 Reading/Writing from S3

Fire is fully integrated with AWS S3. The Dataset Processors of Fire, can directly read data from S3 if the policies allow them to.

Dataset Processors

Dataset Processors include:

- Read CSV
- Read Parquet
- Read JSON
- Read XML

The path specified for reading from S3 would be s3://...

Reading from S3

Below is an example Workflow. It reads a CSV file from S3, parses it and prints out the first 10 records.

In the dialog box of the Read CSV processor the path is specified as s3a://sparkflow-sample-data/data/ Clickthru.csv

	° ≡ ► <mark>-</mark>	
	2 CClickthru.csv	PrintNRows
	Reading csv file from s3 bucket location	
2 ReadCSV		 ×
OUTPUTSTORAGE LEVEL : 🕑	DEFAULT	
PATH * : 🛛	s3a://sparkflow-sample-data/da	ta/Clickthru.csv BROWSE HDFS BROWSE S3 VIEW FILE
EPARATOR : 🛛	,	
HEADER : 😧	true	τ
DROP MALFORMED : 🛛	false	•
SCHEMA COLUMNS : Ø REFRESH SCHEMA		
COLUMN NAMES FOR THE CSV 🕑	COLUMN TYPES FOR THE CSV @	COLUMN FORMATS FOR THE CSV 🛛
Timestamp	STRING	format
Heorid	INITEGED	v format
		OK CANCEL

Writing to S3

Below is an example Workflow. It reads a CSV file and save it to S3 path specified.

In the dailog box of the save CSV processor the path is specified as s3a://sparkflow-sample-data/write/

Execution Result

Once the above workflow successfully completed, the save data can be viewed using DATABROWSERS/AWS S3 Location with specified path

17.1.9 Saving ML Model to S3

.dataset.NodeDatasetCSV : 2 : Apr 9, 2020	12:21:59 PM		
UserId	IP Address	Product Id	
IntegerType	StringType	IntegerType	
275	207.51.113.192	1	
586	62.34.98.94	2	
508	20.237.172.182	3	
378	69.215.255.150	4	
965	56.101.183.251	5	
263	9.151.97.180	6	
670	101.195.1.186	7	
	UserId IntegerType 275 586 508 378 965 263	Userid IP Address IntegerType StringType 275 207.51.113.192 586 62.34.98.94 503 20.237.172.182 378 69.215.255.150 965 56.101.183.251 263 9.151.97.180	Vertical Series PAdress Product Id IntegerType StringType IntegerType 275 207.51.113.192 IntegerType 586 62.34.98.94 2 503 20.237.172.182 3 784 69.215.255.150 4 655 61.01.183.251 5 6263 9.51.97.180 5

ReadCSV

0		$\mathbf{\otimes}$	-	0		8
•		G G	0		2	G
car	s.cs	sv		Sc	aveCS	V



AME	c1	c2		c3	c4	
PE	integer	double	9	double	double	
ORMAT						
TPUT STOR	AGE LEVEL :		DEFAULT	v	BROWSE HDFS BROWSE S3	VIEW FILE
E MODE : 6	0		Append	v		
ADER : 😧			true	Ţ		

	APPLICATIONS - 🔿 SCHEDULED - 🚱 EXECUTIONS - 🗳		-0			
http://ip-172-31-71-252.ec2.internal:4041						
Executing Node fire.nodes.dataset.Nod	eDatasetCSV : 1 : Apr 9, 2020 8:02:14 AM					
Output Schema						
Calparschema						
Executing Node fire.nodes.save.NodeS	aveCSV : 2 : Apr 9, 2020 8:02:15 AM					
🕀 Input Schema						
Output Schema						
Successfully finished executing the wor	kflow					
s3a://sparkflow-sample-data/write/						
	PATH	SIZE	IS DIRECTORY		GROUP	LAST UPDATED
ESS	write/_SUCCESS	0 bytes	false	support	d41d8cd98f00b204e9800998ecf8427e	04/09/2020 at 1:32P
000-2bbc7cb2-fd96-41c7-afcc-6009f4f20a0c-c000.csv	write/part-00000-2bbc7cb2-fd96-41c7-afcc-6009f4f20a0c-c000.csv	264 bytes		support	49a113f4421549cc2ab046497e4a80b5	04/09/2020 at 1:32P

Saving Spark ML Model

Below is an example workflow in sparkflows, where data is read from S3 and the final Spark ML model is saved to S3 location.

Workflow:

Configure ReadCSV

Configure SaveMlModel

Execution Result:

Saving H20 ML Model

Below is an example workflow in sparkflows, where final H20 ML model is saved to S3 location.

Workflow:

Configure Save H20 ML Model

Execution Result:

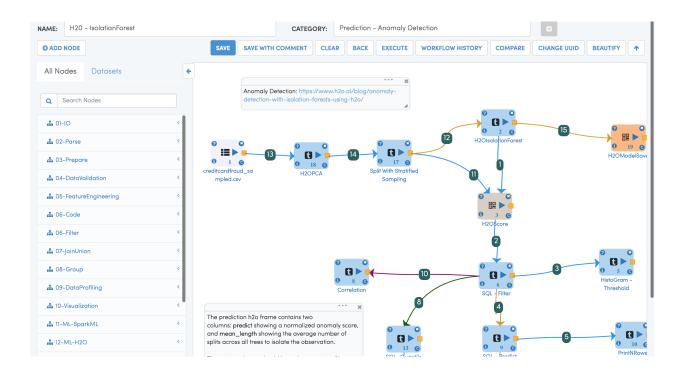
17.1.10 Fire Integration with HIVE

Fire seamlessly integrates with HIVE when running on AWS.

15 ReadCSV		* ×
OUTPUT STORAGE LEVEL : 🕑	DEFAULT	
• PATH * : @	s3a://sparkflow-sample-data/sparkflowsampledata/data/housing.cs	BROWSE HDFS BROWSE 53
SEPARATOR : 0		
HEADER : 🚱	true \$	
DROP MALFORMED : 😡	false \$	
SCHEMA COLUMNS : REFRESH SCHEMA		
COLUMN NAMES FOR THE CSV @ COL	JMN TYPES FOR THE CSV 😧 COLUMN FORMATS FOR TH	HE CSV 😧
	EGER format	
		OK CANCEL
A 07-joinUnion	5 OneHofthicoder	4
4.08-Group		

19 MI	_ Mode	el Save	eg 0	NodeModelSave	9										2	×
																•
NAME	id	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea	is_price_outlier	driveway_idx	recrooi
ТҮРЕ	integer	double	integer	integer	integer	integer	string	string	string	string	string	integer	string	string	double	double
FORMAT																
OUTPUT ST	FORAGE LE	VEL : 😡				DEFAU	LT						*			
PATH :						s3a://s	sparkflow-s	ample-date	a/write/sa	vesparkm	nodel			BROWSE HDFS	ROWSE S3	
														VIEW FILE		
OVERWRIT	E OUTPUT	:				O TRUE										
															ОК СА	NCEL
👬 06-Code				<		Pro	ədict				17 G					
👍 06-Filter				<												
📥 07-JoinUn	ion			<												
🔒 08-Group				<												
🚓 09-DataP	rofiling			<												

	40750.0	5200	4	1	3	yes	no	no	no	no	0	no
19	45000.0	3450	1	1	1	yes	no	no	no	no	0	no
20	45000.0	3986	2	2	1	no	yes	yes	no	no	1	no
Output	Schema											
Executing	Node fire nodes	ml NodeMor	delSave · 19 · /	Apr 12 2020 8.15	49 AM							
Executing	Node fire.nodes.	.ml.NodeMoo	delSave : 19 : /	Apr 12, 2020 8:15	:49 AM							
		.ml.NodeMoo	delSave : 19 : /	Apr 12, 2020 8:15	:49 AM							
) Node fire.nodes. del Save	.ml.NodeMod	delSave : 19 : /	Apr 12, 2020 8:15	:49 AM							
ML Mc						2c440-f352-46	22-82b5-7553c	lbabb9d2				
ML Mc	del Save					2c440-f352-46	22-82b5-7553d	lbabb9d2				
ML Mc	del Save					2c440-1352-46	22-82b5-7553d	lbabb9d2				
ML Mc	del Save	to : s3a://sį	parkflow-sampl			2c440-f352-46	22-82b5-7553c	lbabb9d2				
ML Mc	idelSave he Spark ML model	to : s3a://sį	parkflow-sampl			2c440-f352-46	22–82b5–7553c	Ibabb9d2				



19 H2	OMod	elSave	93 0	NodeH2OM	lodelSave													2	×
																			•
NAME	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19
ТҮРЕ	double	double	double	double	double	double	double	double	double	double	double	double	double	double	double	double	double	double	dout
FORMAT																			
OUTPUT ST	ORAGE LE	VEL: 😧				DE	EFAULT							\$					
PATH * : 😧						sa	3a://spark	flow-sam	nple-data	/write/sa	veh20mo	del			BROWSE	HDFS	BROWSE	S3	
															VIEW FIL	.E			
									_								0	CAN	CEL
06-Code				<										0 3					
06-Filter				<										H2OScor	0				
07-JoinUni	on			<										9				0	0
08-Group				<					e ()	© ▶ ■(10		- C >	0	3			G
, 09-DataPr	ofiling			<					Correlat					6 6 SQL - Filt				HistoGra Thresha	
				U								81							

Executing Node fire.nodes.h2o.NodeH2OModelSave : 19 : Apr 12, 7	2020 8:03:42 AM
H2OModelSave	
Saving H2O Model to : s3a://sparkflow-sample-data/write/saveh20mo	del/3ff95ba3-8c35-4523-b737-405d6cfa6b3c
H2OModelSave Successfully saved H2O Model to : s3a://sparkflow-sample-data/wri	te/saveh20model/3ff95ba3-8c35-4523-b737-405d6cfa6b3c
Executing Node fire.nodes.doc.NodeStickyNote : 15 : Apr 12, 2020	8:03:42 AM
Executing Node fire.nodes.doc.NodeStickyNote : 20 : Apr 12, 2020	8:03:42 AM
Successfully finished executing the workflow	

Overview

On AWS, the data normally resides in S3 buckets. HIVE tables are created pointing to data in the S3 buckets.

Details

- Fire would run on the master node of the EMR cluster, or on an Edge node with the cluster contiguration files.
- HIVE can be running on the same EMR cluster on running on another EMR cluster.
- Make sure to have the correct hive-site.xml on the cluster where the Spark jobs are running.
- Fire will automatically pick it up and be able to process it.

Writing to **HIVE**

Below is a workflow for writing to HIVE.

It reads housing.csv, creates a DataFrame and writes it out to a HIVE table.

17.1.11 Fire Integration with Redshift

Fire is fully integrated with Redshift. Fire has a number of Processors specifically for Redshift.

Redshift Processors

Fire has processors for reading from and writing to Redshift. They include:

- Read Redshift AWS
- Write Redshift AWS

17.1.12 Fire Integration with SageMaker

Fire is fully integrated with AWS SageMaker. Fire provides a number of processors for doing model building with SageMaker.

You can do Data Preparation and Feature Engineering with Sparkflows doing compute with Apache Spark. Sparkflows then seamlessly enables you to do your model training and deployment with SageMaker.

The above forms a very powerful combinations for end to end Machine Learning.

Spark Sagemaker Examples

There are a number of SageMaker-Spark examples by AWS here :

- https://github.com/aws/sagemaker-spark
- https://docs.aws.amazon.com/sagemaker/latest/dg/apache-spark-example1.html

Fire SageMaker Processors

SageMaker Processors include:

- KMeansSageMakerEstimator
- XGBoostSageMakerEstimator
- LDASageMakerEstimator
- LinearLearnerBinaryClassifier
- LinearLearnerRegressor
- PCASageMakerEstimator
- SaveSageMaker

AWS Provided Policies

AWS provides managed policies for SageMaker. Example : AmazonSageMakerFullAccess

Launching EMR

When launching the EMR Cluster make sure that the Role (eg: EMR_EC2_DefaultRole) used has the AmazonSage-MakerFullAccess policy.

Now that the Roles and Policies are in place, start up your EMR cluser with the EMR_DefaultRole and EMR_EC2_DefaultRole Roles.

Create New Role

Create a new Role called **aws-sagmaker-full-access** with the below Policy. It would be used in the Apache Spark job when accessing SageMaker.

AmazonSageMakerFullAccess

Use ARN of the new Role in the Workflow

We now use the ARN of the new Role when we use the SageMaker KMeans Estimator Node in the Workflow.

arn:aws:iam::account_id:role/aws-sagemaker-full-access

AWS Instance Types

AWS has various instance types:

- p : GPU Instances
- c : Compute Instances
- r : Memory Optimized Instances
- m : General Instances

Amazon SageMaker Instance Types details are here : https://aws.amazon.com/sagemaker/pricing/instance-types/

Dataset Column Names for Training with Sagemaker

Sagemaker needs the following columns to exist in the Dataset.

- label : label column
- features : features column, this column can also be set

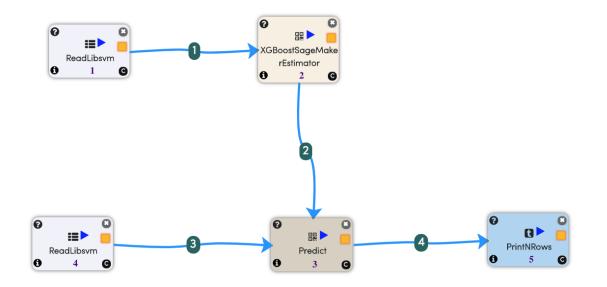
Flow with Sparkflows and AWS

- We do the Data Preparation and Feature Generation in EMR with Sparkflows.
- When Sparkflows invokes the SageMakerEstimator, it calls SageMaker for Training and Deployment.
- Once the model is deployed on SageMaker, the endpoint can be used for realtime predictions.

XGBoost Sagemaker Workflow

Below is a workflow which:

- · Reads in a libsvm file as input
- Performs XGBoost Modeling
- Reads in another libsvm file
- Performs predictions with the model built in the previous step
- Prints out the result



XGBoost Configuration

Below are the configuration setup details of the XGBoost Processor.

OUTPUT STORAGE LEVEL : 0	DEFAULT	\$
ROLE ARN * : 🕢	arn:aws:iam::004331324847:role/aws-segmaker-full-access	
	ml.c.4.xlarge	
	1	
	ml.c4.xlarge	
	1	
BOOSTER ★: €	gbtree	\$
SILENT *: 0	1	\$
NTHREAD *: 0	2	
OBJECTIVE *: 😡	multi:softmax	\$
NUM TREES *: @	2	
NUM CLASSES *: O	10	

OK CANCEL

Executing the Workflow

Below are the results of executing the workflow.

Executing Node fire.nodes.dataset.NodeDatasetLibsvm : 1 Dec 30, 2018 7:27:42 AM
ReadLibsvm
Reading LibSVM File
Output Schema
Executing Node fire.nodes.sagemaker.NodeXGBoostSageMakerEstimator : 2 Dec 30, 2018 7:27:42 AM
🕒 Input Schema
CBoostSageMakerEstimator
Endpoint Name is endpoint-df4c1966a59a-2018-12-30T07-27-42-915

17.1.13 Fire Integration with Kinesis

This document described Fire integration with Kinesis. Fire uses Apache Spark Structured Streaming Connector from Qubole.

https://github.com/qubole/kinesis-sql

Install AWS CLI

Install AWS CLI:

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html

Create an access key and secret key

Create an access key and secret key for the user (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_ access-keys.html#Using_CreateAccessKey).

Note It's important to regularly rotate your access and secret keys. See this documentation for more information (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#rotating_access_keys_console)

Configure AWS CLI

Configure AWS CLI:

Create AWS Kinesis Stream

Create AWS Kinesis Stream:

aws kinesis create-stream --stream-name sparkflows_kinesis_test --shard-count 1

Send message to AWS Kinesis from AWS CLI

Sending message to Kinesis:

```
aws kinesis put-record --stream-name sparkflows_kinesis_test --data file://data.json -
→-partition-key uuidgen
```

Update EMR_EC2_Default_Role

Update EMR_EC2_DefaultRole with AmazonKinesisFullAccess Policy so that our EMR Cluster would have full access to Kinesis.

Or Create an IAM policy for accessing Amazon Kinesis

Create an IAM policy that defines what this user has access to in your AWS account. It's important to only grant this user minimal access within your account. See this documentation for how to create IAM policies (https://docs.aws. amazon.com/IAM/latest/UserGuide/access_policies_create.html).

Create EMR Cluster with the above Role

When we create the EMR Cluster with the above Role, it would have full access to Amazon Kinesis.

Pushing data to Kinesis

AWS provides a Kinesis Data Generator. It can be configured for pushing random data in specified format to Kinesis. https://awslabs.github.io/amazon-kinesis-data-generator/web/help.html

aws Servi	ces 👻 Resource Gro	oups 🗸 🍾			û jayant@spa	rkflows.io @ 0043 👻	• Oregon • Support •
CloudFormatio	n 🖌 Stacks						
	•	The redesigned AWS CloudFor We've completely redesigned the conso			now and provide us fe	Contraction Contra	
Drift detection le	on now available ats you detect whether a sta s menu. Learn more.	ack's actual configuration has been chan	ged outside of C	loudFormation. To detect drift o	a stack, select the st	ack, and then select $\mathbf{D}\mathbf{\epsilon}$	Setect drift for current stack
Create Stack	Actions - Design	a template					C \$
Filter: Active - By St	ack Name						Showing 1 stack
Stack Name		Created Time	Status	Drift Status	Des	cription	
C Kinesis-Data-Gene	rator-Cognito-User	2018-12-29 08:21:35 UTC-0800	CREATE_COM	IPLETE NOT_CHECKE	D This	template creates an Ar	mazon Cognito User Pool an
Kinesis-Data- Stack name: Stack ID: Status: Status reason: Termination protection:	Kinesis-Data-Generator-	-	Data-Generator-	Cognito-User/cf51a0e0-0b85-1	e9-9a6e-028572da1(Actions • Update Stack
		1.1.9.					
Drift status: Last drift check time:	NOT_CHECKED View of	Jetails					
IAM role:							
Description	This template creates an Data Generator tool.	Amazon Cognito User Pool and Identity	Pool, with a sin	gle user. It assigns a role to auth	enticated users in the	identity pool to enable	the users to use the Kinesis
 Outputs 							
Кеу		Value		Description		Export Name	
KinesisDataGeneratorUr	I	https://awslabs.github.io/amazon-kir erator/web/producer.html?upid=us-v C2sR&ipid=us-west-2:62243857-ece 14a60da643c4&cid=7g98rfkihbjr8jdr	vest-2_pBGPv eb-4d61-b6ef-	The URL for your Kinesis Data	Generator.		

Kinesis Workflow in Fire

Workflows can be easily built in Fire which read data from Kinesis, process them and save the results where needed.

REFERENCE : Creating Access Key & Secret Key

- 1. You'll need create a user with programmatic access by following the steps here (https://docs.aws.amazon.com/ IAM/latest/UserGuide/id_users_create.html).
- Next, you'll create an IAM policy that defines what this user has access to in your AWS account. It's important to only grant this user minimal access within your account. See this documentation for how to create IAM policies (https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html).

3. Finally, you'll create an access key and secret key for this user (https://docs.aws.amazon.com/IAM/latest/ UserGuide/id_credentials_access-keys.html#Using_CreateAccessKey).

Note It's important to regularly rotate your access and secret keys. See this documentation for more information (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#rotating_access_keys_console)

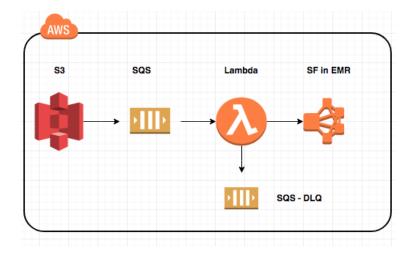
17.1.14 File Watcher with AWS & Sparkflows

Overview

There are many use cases where we have to process the incoming files on S3. This document describes one way to achieve it with SQS, Lambda and using the REST API of Fire Insights.

Design

The below diagram captures the high level design:



Below is the flow of execution:

- New files arrives on S3 in the directory location /sparklows-file-watcher/raw-data/iot/ 2019-08-2201
 - In the above design, all the raw data comes into the directory /sparklows-file-watcher/ raw-data
 - There are various types of raw data which can come.
 - iot is one type of raw data coming in. Each day we receive a number of iot files in the folder / sparklows-file-watcher/raw-data/iot/yyyy-MM-dd.
 - Once all the files for that date have been written to the appropriate folder, a _SUCCESS files is written into it.
- It triggers an event which is sent to a configured SQS queue.
- Once the event reaches SQS, it triggers an AWS Lambda.
- The AWS Lambda uses the Fire Insights REST API(http://docs.sparkflows.io/en/latest/rest-api-reference/ workflow.html#execute) to execute a workflow to process the new incoming files in the AWS S3 bucket.
- If AWS Lambda fails, it sends the event to DLQ (Dead Letter Queue). It can be further handled from there based on the requirements.

Create an SQS Queue

Create an SQS Queue for receiving the events from S3 and triggering the AWS Lambda function.

Below we see the SQS queue : sf-workflow-file-watcher-ql-dev.

It has the below permissions to receive the messages from S3 bucket and invoke the AWS Lambda function.

Create New Queue Actions	·						Q
ilter by Prefix: 🤍 sf-w		×					$\ \leqslant ~~\leqslant~$ 1 to 2 of 2 items $~\gg~$
Name			~ Queue Type ~	Content-Based Deduplication	Messages Available -	Messages in Flight-	Created
sf-workflow-file-watcher-dql-dev			Standard	N/A	0	0	2019-08-26 17:19:21 GMT+0
sf-workflow-file-watcher-ql-dev			Standard	N/A	0	1	2019-08-26 17:19:24 GMT+0
sf-workflow-file-watcher-ql-dev			Standard	N/A	0	1	2019-08-26 17:19:24 GMT+0
			000				
2S Queue selected Details Permissions Redr	ve Policy Monitoring Tags E	Encryption Lambda Triggers					
Add a Permission Edit Policy Document (A	Ivanced) What's an SQS Queue Access Policy?						
Effect Principals Actions	Conditions						
Vlow • Everybody (*) • SQS:Send?	essage • AmEquals		/ X				
	 aws:SourceArn: "arn:aws:s3:::; 						
	ssage • AmEquals		/ X				
Allow • Everybody (*) • sqs:SendM • sqs:GetQue		-east-1:					

Configure AWS S3 bucket to generate events

Configure the AWS S3 bucket to send events for the new files coming in to AWS SQS queue.

Below, it looks for the new files with prefix of events and suffix of _SUCCESS. It sends these events to sf-workflow-file-watcher-ql-dev SQS Queue.

Create the AWS Lambda function

Create the AWS Lambda function to take the SQL Event and kick off the workflow in Fire Insights. This workflow would process the new files which came in.

First create an IAM role. An example is shown below.

We add 3 Environment variables as shown below. These get used by the Lambda functions in this example.

- SPARKFLOWS_TOKEN or KMS_ARN
- SPARKFLOWS_URL
- WORKFLOW_ID

Instead of the Sparkflows token, users can encrypt the token using KMS and use the kms arn as the Environment variable and decrypt the token using kms inside the Lamdba.

Upload the jar file for the RequestHandler. It can also be placed into S3 location and the Lambda configured for it.

	Even	ts		×
Add notification	Delete Edit			
Name	Events	Filter	Туре	
ile-watcher-events				\times
Name 🚯				
file-watcher-events	1			
Events ()				
PUT		Permanently de	eleted	
POST		Delete marker	created	
COPY		All object delet	e events	
Multipart upload	completed	Restore initiate	d	
All object create	events	Restore compl	eted	
Object in RRS los	st			
Prefix 🜖				
events/				
Suffix 🜖				
_SUCCESS				
Send to 🚯				
SQS Queue				~
sqs				
sf-workflow-file-wa	tcher-ql-dev			~
1 Active notificat	ions		Cancel	Save

	-workflow-execute-handler" is too large to enable inl Runtime Java 8 V		Handler Info	voke your function. orkflowExecuteHandler::handleF	
Code entry type Upload a .zip or .jar file Upload a .zip or .jar file Upload a file from Amazon S3	Runtime		Handler Info		·
Upload a.zip or.jar file ▼ Upload a.zip or.jar file Upload a file from Amazon S3]		orkflowExecuteHandler::handleF	i
Upload a .zip or .jar file Upload a file from Amazon S3 բոյ Օրսաս	Java 8 🛛 🔻		com.sf.handler.W	orkflowExecuteHandler::handleF	
Upload a file from Amazon S3					
[M] obroad					
or nies larger than 10 MB, consider uploading using Amazon 55.					
invironment variables ou can define environment variables as key-value pairs that are accessible from your fi SPARKFLOWS_TOKEN	unction code. These are useful to store configuration settings 60efbdd9-089c-4dff-a12d-2c82a7c08413	without the need to char	nge function code. Lear Remove	n more	
SPARKFLOWS_URL	http://ec2-54-158-230-0.compute-1.amazonaws.c	com:8083	Remove		
WORKFLOW_ID	225		Remove		
Key	Value		Remove		
ncy little littt	Pulluc				

WorkflowExecuteHandler

package com.sf.handler

```
import com.amazonaws.services.lambda.runtime.events.SQSEvent
import com.amazonaws.services.lambda.runtime.events.SQSEvent.SQSMessage
import com.amazonaws.services.lambda.runtime.{Context, LambdaLogger, RequestHandler}
import com.amazonaws.services.s3.event.S3EventNotification
import com.amazonaws.services.s3.event.S3EventNotification.S3EventNotificationRecord
import com.sf.WorkflowExecute
import scala.collection.JavaConverters._
class WorkflowExecuteHandler extends RequestHandler[SQSEvent, Unit] {
 private val token = System.getenv("SPARKFLOWS_TOKEN")
 private val sparkflowsURL = System.getenv("SPARKFLOWS_URL")
 private val workflowId = System.getenv("WORKFLOW_ID")
 def handleRequest(sqsEvent: SQSEvent, context): Unit = {
   implicit val logger: LambdaLogger = context.getLogger
    logger.log(s"sparkflowsURL: $sparkflowsURL")
   logger.log(s"workflowId: $workflowId")
   sqsEvent
      .getRecords
      .asScala.map(sqsMessageToS3Event)
      .foreach(_.getRecords.asScala.foreach(processS3Record))
  }
 private[handler] def sqsMessageToS3Event(sqsMessage: SQSMessage):...
→S3EventNotification = {
   S3EventNotification.parseJson(sqsMessage.getBody)
  }
 private[handler] def processS3Record(s3EventRecord: S3EventNotificationRecord)
                                      (implicit logger: LambdaLogger): Unit = {
   val s3Entity = s3EventRecord.getS3
   val inputBucketName: String = s3Entity.getBucket.getName
   val inputObjectKey: String = s3Entity.getObject.getUrlDecodedKey
   val eventName: String = s3EventRecord.getEventName
   val path = s"s3://$inputBucketName/$inputObjectKey".replace("/_SUCCESS", "")
    logger.log(s"Event record $eventName; path $path")
   val body = s"""
                  | {
                    "workflowId": "${workflowId}",
                    "parameters": "--var datapath=${path}"
                  | }
       """.stripMargin
     val workflowStatus = WorkflowExecute.executeWorkflow(body, token, sparkflowsURL)
```

(continues on next page)

(continued from previous page)

```
logger.log(s"Status of workflow $workflowStatus")
```

} }

WorkflowExecute

```
package com.sf
import com.mashape.unirest.http.Unirest
object WorkflowExecute {
  def executeWorkflow(body: String, token: String, sparkflowsHostName: String) = {
   val workflow = Unirest.post(s"$sparkflowsHostName/api/v1/workflow/execute")
      .header("Content-Type", "application/json")
      .header("Cache-Control", "no-cache")
      .header("Authorization", s"Bearer $token")
      .body (body)
      .asString
    workflow match {
      case s if workflow.getStatus >= 200 && workflow.getStatus <= 300 => workflow.
→getBody
      case f => throw SubmissionFailedException(s"Job submissions failed, status code,
→is ${f.getStatus}")
   }
  }
  case class SubmissionFailedException (message:String) extends Exception (message)
}
```

17.1.15 CloudFormation Template with Embedded H2 DB

Overview

Using CloudFormation Templates, Fire can be easily installed on AWS. This CFT works with EMR 5.8 onwards. The below steps would allow you to start up an EMR Cluster and have Fire setup on it.

The CFT does the following:

- Creates EMR cluster with 1 master node and 2 worker nodes by default.
- Once the cluster is ready it runs the job/script to deploy Fire (takes around 1-1:30 min for deploying app!).

Relevant Files

Title	Description	File
emr-file-	CloudFormation Template	https://s3.amazonaws.com/sparkflows-cft/h2-db/emr-fire-h2.
h2.json		json
deploy-fire-	Script for deploying Fire	https://s3.amazonaws.com/sparkflows-cft/h2-db/deploy-fire-h2.
h2.sh		sh
script-	Script Runner	https://s3.amazonaws.com/sparkflows-cft/h2-db/script-runner.
runner.jar		jar

Table 1: Below are the Relevant Files

Ports

• With this CFT and deploy-fire-h2.sh, when Fire comes up, it would be listening on ports 8085 and 8086.

Download Files and Upload to your S3 Bucket

- Download CFT emr-fire-h2.json from the above link.
- Download deploy-fire-h2.sh and script-runner.jar from the above links and upload them to your s3 bucket

Update Cloudformation template based on your environment

Update the CFT emr-fire-h2.json according to your requirement and environment in which you are deploying.

• ElasticMapReduce-Master-SecurityGroup under mastersg:

From AWS console -> EC2 -> Security Groups -> search for "ElasticMapReduce-master"

• ElasticMapReduce-Slave-SecurityGroup under slavesg:

From AWS console -> EC2 -> Security Groups -> search for "ElasticMapReduce-slave"

• Applications:

```
By default the CFT deploys Hadoop, Hive & Spark. Add any other Applications which, {\hookrightarrow} you need.
```

• EbsRootVolumeSize:

```
If required change the root(/) ebs volume size. By default CFT has 50GB disk \begin{subarray}{c} \begin{subarray}{c} \begin{subarray}{c} \begin{subarray}{c} \end{subarray} \end{subarray} \end{subarray}
```

• SizeInGB for Master and Core Instances:

```
If required change the SizeInGB under EbsConfiguration. By default CFT has 50GB _{\rm \ominus} disk volume (used for hdfs)
```

• VolumesPerInstance for Master and Core Instances:

```
If required change the VolumesPerInstance under EbsConfiguration By default cft_

→has 1. It means one additional disk of 50GB added to each instance(for hdfs). e.

→g. If you change it 2, two 50GB (SizeInGB size) disks will be added to each_

→instances.
```

• deploy-fire-h2.sh and script-runner.jar:

```
Change the s3 bucket path for these two files, this s3 bucket must be same_

→bucket as S3Bucket. You'll pass the S3Bucket value while creating the_

→cloudformation stack.
```

Steps to Create EMR Cluster and Deploy Fire

- AWS web Console -> Management tools -> CloudFormation
 - Click on Create Stack.
- Next page is Select Template
 - Select the radio-button Upload a template to Amazon S3
 - Select the updated emr-fire-h2.json from your system
 - Click Next
- Next page is **Specify Details**
 - Enter CloudFormation stack name

Name of Parame-	Description		
ter			
AdditionalSecurityGroupsom the list choose the additional security group(sg), it's required because default emr			
	ports are not opened for ssh, fire & etc		
AmiId	EMR cluster can be launched using Custom AMI, pass the value if you have a Custom AMI		
ClusterName	Name for EMR Cluster		
CoreInstanceType	Provide the required instance type for core nodes, default instance type is m4.xlarge		
CoreNodes	Choose the required number of core nodes, by default it's 2		
EmrVersion	Choose the required EMR version, it's should be above EMR v.5.8.x		
Environment	By default dev		
FireVersion	Enter the required version of Fire		
KeyName	Enter the valid pem key name to connect to emr nodes		
MasterInstanceType	Provide the required instance type for master nodes, default instance type is m4.xlarge		
MasterNodes	By default 1		
Owner	provide the name of a team or person creating the cluster		
ReleaseVersion	Eversion Enter the required ReleaseVersion, it has to match with fire version		
S3Bucket	Provide the s3 bucket name, this s3 bucket should be same s3 bucket where deploy-fire-		
	h2.sh and script-runner.jar are uploaded		
Subnet	Provide the proper subnet name, which has sufficient resources to create emr cluster		
TaskInstanceType	Optional, required only if you're choosing TaskNodes. Provide the required instance type		
	for task nodes, default instance type is m4.xlarge		
TaskNodes	Optional, required only if you want to create the cluster with tasknodes.By default zero,		
	enter the required number of nodes		

Table 2:	Update Parameters where needed
----------	--------------------------------

- Click Next
- Next Page is **Options**
 - If required (not mandatory) enter tag details
 - Click Next
- Next Page is **Review**

- Review all the details provided to create an EMR stack
- Click on Create
- It will start creating the Stack
- Next page is back to Cloudformation Page
 - Choose your Stack name
 - Click on **Events** to check the process
 - Click on Resources to get the EMR Cluster id
- Once the stack runs successfully, your EMR Cluster and Fire is ready to use. Cluster creation time depends on your EMR cluster configuration
- To cross check the Fire installation
 - Go to EMR from AWS web console
 - Choose your EMR Cluster
 - Identify the Master Node Public DNS
 - Go to http://masternodeip:8085/index.html

Connect Fire to the New Cluster

- Go to User/Administration
- Click on Infer Hadoop Configuration
- Click on the Save button

Load Examples

- In Fire, click on Load Examples
- ssh to the master node
- cd /opt/fire/fire-3.1.0
- Upload the example data files to HDFS
 - hadoop fs -put data

Create hadoop user

- Go to Administration/User
- Click on Add User
- Create a new user with username hadoop
- Log out and log back in as user hadoop

Start running the Examples

- Go to Applications
- Start creating/using the Applications

Summary

Using the above CFT you have your EMR cluster with Fire running seamlessly.

17.1.16 CloudFormation Template with MySQL

Overview

Using CloudFormation Templates, Fire can be easily installed on AWS. This CFT works with EMR 5.8 onwards.

The below steps would allow you to start up an EMR Cluster and have Fire setup on it.

The CFT does the following:

- Creates External DB for Fire to be used as the metastore for Fire data
- Creates EMR cluster with 1 master node and 2 worker nodes by default.
- Once the cluster is ready it runs the job/script to deploy Fire (takes around 1-1:30 min for deploying app!).

Relevant Files

Title	Description	File
emr-file-	CloudFormation Template	https://s3.amazonaws.com/sparkflows-cft/mysql-db/
mysql.json		emr-fire-mysql.json
deploy-fire-	Script for deploying Fire with	https://s3.amazonaws.com/sparkflows-cft/mysql-db/
mysql.sh	MySQL	deploy-fire-mysql.sh
script-	Script Runner	https://s3.amazonaws.com/sparkflows-cft/mysql-db/
runner.jar		script-runner.jar

Table 3: Below are the Relevant Files

Ports

• With this CFT and deploy-fire-mysql.sh, when Fire comes up, it would be listening on ports 8085 and 8086.

Download Files and Upload to your S3 Bucket

- Download CFT **emr-fire-mysql.json** from the above link.
- Download deploy-fire-mysql.sh and script-runner.jar from the above links and upload them to your s3 bucket

Update Cloudformation template based on your environment

Update the CFT emr-fire-mysql.json according to your requirement and environment in which you are deploying.

• ElasticMapReduce-Master-SecurityGroup under mastersg:

From AWS console -> EC2 -> Security Groups -> search **for** "ElasticMapReduce-master"

• ElasticMapReduce-Slave-SecurityGroup under slavesg:

From AWS console -> EC2 -> Security Groups -> search for "ElasticMapReduce-slave"

• Applications:

```
By default the CFT deploys Hadoop, Hive & Spark. Add any other Applications which \begin{subarray}{c} \begin{subarray}{c} \end{subarray} \end{subarray} used.
```

• EbsRootVolumeSize:

```
If required change the root(/) ebs volume size. By default CFT has 50GB disk \begin{subarray}{c} \begin{subarray}{c} \begin{subarray}{c} \end{subarray} \end{subarray}
```

• SizeInGB for Master and Core Instances:

```
If required change the SizeInGB under EbsConfiguration. By default CFT has 50GB_

→disk volume (used for hdfs)
```

• VolumesPerInstance for Master and Core Instances:

```
If required change the VolumesPerInstance under EbsConfiguration By default cft_

\rightarrowhas 1. It means one additional disk of 50GB added to each instance(for hdfs). e.

\rightarrowg. If you change it 2, two 50GB (SizeInGB size) disks will be added to each_

\rightarrowinstances.
```

• deploy-fire-mysql.sh and script-runner.jar:

```
Change the s3 bucket path for these two files, this s3 bucket must be same.

→bucket as S3Bucket. You'll pass the S3Bucket value while creating the.

→cloudformation stack.
```

Steps to Create EMR Cluster and Deploy Fire

- AWS web Console -> Management tools -> CloudFormation
 - Click on Create Stack.
- Next page is Select Template
 - Select the radio-button Upload a template to Amazon S3
 - Select the updated emr-fire-mysql.json from your system
 - Click Next
- Next page is Specify Details
 - Enter CloudFormation stack name

Name of Parame-	Description
ter	
AdditionalSecurityGr	outprom the list choose the additional security group(sg), it's required because default emr sg's
	ports are not opened for ssh, fire & etc
AmiId	EMR cluster can be launched using Custom AMI, pass the value if you have a Custom AMI
ClusterName	Name for EMR Cluster
CoreInstanceType	Provide the required instance type for core nodes, default instance type is m4.xlarge
CoreNodes	Choose the required number of core nodes, by default it's 2
EmrVersion	Choose the required EMR version, it's should be above EMR v.5.8.x
Environment	By default dev
FireVersion	Enter the required version of Fire
KeyName	Enter the valid pem key name to connect to emr nodes
MasterInstanceType	Provide the required instance type for master nodes, default instance type is m4.xlarge
MasterNodes	By default 1
Owner	provide the name of a team or person creating the cluster
ReleaseVersion	Enter the required ReleaseVersion, it has to match with fire version
S3Bucket	Provide the s3 bucket name, this s3 bucket should be same s3 bucket where deploy-fire.sh
	and script-runner.jar are uploaded
Subnet	Provide the proper subnet name, which has sufficient resources to create emr cluster
TaskInstanceType	Optional, required only if you're choosing TaskNodes. Provide the required instance type
	for task nodes, default instance type is m4.xlarge
TaskNodes	Optional, required only if you want to create the cluster with tasknodes.By default zero,
	enter the required number of nodes

Table 4: Update Parameters where needed

- Click Next
- Next Page is **Options**
 - If required (not mandatory) enter tag details
 - Click Next
- Next Page is Review
 - Review all the details provided to create an EMR stack
 - Click on Create
 - It will start creating the Stack
- Next page is back to Cloudformation Page
 - Choose your Stack name
 - Click on Events to check the process
 - Click on Resources to get the EMR Cluster id
- Once the stack runs successfully, your EMR Cluster and Fire is ready to use. Cluster creation time depends on your EMR cluster configuration
- To cross check the Fire installation
 - Go to EMR from AWS web console
 - Choose your EMR Cluster
 - Identify the Master Node Public DNS
 - Go to http://masternodeip:8085/index.html

Connect Fire to the New Cluster

- Go to Administration/Configuration
- Click on Infer Hadoop Configuration
- Click on the Save button

Load Examples

- In Fire, click on Load Examples
- ssh to the master node
- cd /opt/fire/fire-3.1.0
- hadoop fs -put data

Create hadoop user

- Go to Administration/User
- Click on Add User
- Create a new user with username hadoop
- Log out and log back in as user hadoop

Start running the Examples

- Go to Applications
- Start building your Applications.

Summary

Using the above CFT you have your EMR cluster with Fire running seamlessly.

CHAPTER 18

AZURE Integration

18.1 AZURE Guide

18.1.1 Introduction

Fire Insights is the flagship product from Sparkflows. It is seamlessly integrated with Azure. With Fire Insights you can perform self-serve data processing, analytics and machine learning on Azure.

Fire Insights integrates with Azure Databricks, ADLS, HDInsight etc.

Fire Insights comes with a number of components including:

- Workflow Editor : To create workflows for data processing, analytics and machine learning.
- **300+ Processors** : These include reading data from various stores, data processing, machine learning and visualizations.
- Execution Engine : For executing the workflow on Azure VM or HDInsight
- Scheduler : For scheduling running the workflows at certain time intervals
- I-Dashboard : For Visualization using chart, dashboard

18.1.2 Deployment Guide

Fire Insights can be easily installed on an Azure Standalone VM.

prerequisite:

- java 8 should be installed
- if you do not already have it, Need to install
- Download it from below link:

https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html

• Install using below command (Centos):

yum localinstall jdk-8uxxx-linux-x64.rpm

• Set the below in .bash_profile:

export JAVA_HOME=/usr/java/jdk1.8.0_xxx-amd64/

Below are the overall steps for installing Fire Insights on VM.

- ssh into the Azure VM
- Download Fire Insights from https://www.sparkflows.io/download
- Unzip it
- Create H2 Database
- Start Fire

Steps

• Create a VM on Azure:

Create a vm if you do not already have it running.

• Update the inbound rule

```
    ssh port ie 22 should be accessible to ssh to Azure VM.
    We would have Fire listening on ports 8080, so just ensure its opened.
```

• ssh into the VM:

```
ssh -i my.pem userp@public ip.
```

• Just Confirm that java 8 is already installed, if not follow above steps:

java -version

- Download the fire tgz file by one of the following options:
 - https://www.sparkflows.io/download OR
 - https://www.sparkflows.io/archives OR
 - wget https://s3.amazonaws.com/sparkflows-release/fire/rel-x.y.z/2/fire-x.y.z.tgz
- Unpack it:

tar xvf fire-x.y.z.tgz

• Create H2 DB:

• Launch Fire Server:

```
cd <fire install_dir>
./run-fire-server.sh start
```

• Open your web browser and navigate to:

```
<machine_ip>:8080
```

• Login with the following default username and password:

```
username : admin
password : admin
```

Loading Example Workflows

- From the home page of Fire Insights, click on *Load Example Applications*
- Upload the Fire examples data with default or if data is available at anyother location, point to that location:

Install and Start Running Example Workflows

• Start off with executing the example workflows:

```
    Fire comes pre-packaged with a number of example workflows, you can start.
    A executing.
```

18.1.3 Azure Databricks Integration Steps

Fire Insights integrates with Databricks. It submits jobs to the Databricks clusters using the REST API of Databricks and have the results displayed back in Fire Insights.

Fire also fetches the list of Databases and Tables from Databricks, making it easier for the user to build their workflows and execute them. In addition Fire displays the list of Databricks clusters running for the user.

Running Databricks on Azure : https://docs.microsoft.com/en-us/azure/azure-databricks/
 quickstart-create-databricks-workspace-portal

Below are the steps for Integrating Fire Insights with your Databricks Clusters.

Install Fire Insights

Install Fire Insights on any machine. The machine has to be reachable from the Databricks cluster.

Upload Fire Core Jar to Databricks

Upload Fire Insights jar to Databricks. Fire Insights jobs running on Databricks make use of this jar file.

Upload fire-x.y.z/fire-core-lib/fire-spark_2_4-core-3.1.0-jar-with-dependencies. jar to Databricks. Upload it under Workspace as a Library on to Databricks.

- 1. Login to Databricks Cluster
- 2. Click on workspace in the left side pane



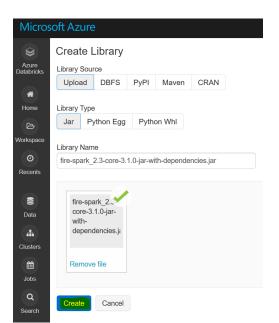
3. Create a new Library



4. Upload fire-spark_2_4-core-3.1.0-jar-with-dependencies.jar from your machine by Clicking on Drop JAR here

Micros	soft Azure
8	Create Library
Azure Databricks	Library Source
	Upload DBFS PyPI Maven CRAN
Home	Library Type
	Jar Python Egg Python Whl
Workspace	
	Library Name
O Recents	Optional
Recents	
Data	
	Drop JAR here
Clusters	
Jobs	
Q	Create Cancel
Search	Create

5. Once fire-spark_2_4-core-3.1.0-jar-with-dependencies.jar is uploaded, click on Create



• Check the box with Install automatically on all clusters, in order to avoid installing it manually to every cluster.



Configure the Uploaded Library in Fire Insights

Configure the path of the uploaded fire core jar library in Databricks in Fire Insights. This has to be done under Administration/Configuration.



Configure app.postMessageURL in Fire Insights

Configure app.postMessageURL to be the IP of the machine on which Fire Insights is installed. Jobs running on Databricks would post back results to Fire Insights using this URL.

_	INFER INDOOP CLUSTER CONTIS		Q pro	
VPP ARELOW SPA				
	VRK HDPS HADOOP 10	AAN HIVE KERBEROG LOAP ALERT DATABRICKS A	WS PLUGINS	
нине пл.е		ware	DESCRIPTION	
epp.peetMessageURL Fre.u	ri perihadi UNL	Mg./Tuble.g.8083/nessagePointgookjab	Fire al postoeck UR, for the messages of east	suffrig jobs

Install Databricks JDBC Driver

Fire needs the Databricks JDBC Driver to be installed. Install it in the fire-user-lib and fire-server-lib folder of the Fire installation.

You can download the Databricks JDBC Driver from the Databricks site :

- https://docs.databricks.com/bi/jdbc-odbc-bi.html
- https://databricks.com/spark/odbc-driver-download

The driver is available as a zip file. eg: SimbaSparkJDBC-2.6.3.1003.zip

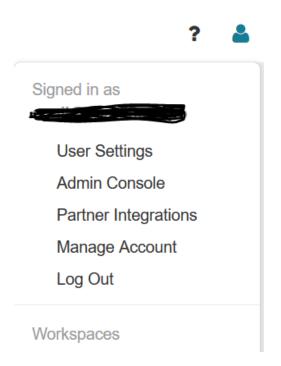
- Unzip the downloaded file. It will create a directory like SimbaSparkJDBC-2.6.3.1003
- Copy the jdbc jar file named SparkJDBC4.jar into fire-x.y.z/fire-user-lib and fire-x.y. z/fire-server-lib

Create your REST API token in Databricks

Create your token in Databricks. It would be used in making REST API calls to Databricks from Fire Insights.

1. Login to your Databricks Account

2. Click on Account icon in right corner top



3. Click on User Settings

User Settings Admin Console Partner Integrations Manage Account Log Out

Workspaces

4. Click on Generate New Token

User Settings							
Access Tokens	Git Integration	Notebook Settings					
Personal access to Generate New To	_	or secure authentication to the Databricks API instead of passwords.					

5. Add comment & Lifetime (days) for token expiry & Click on Generate

Generate New Toker	ו
Comment	
databricks development	
Lifetime (days) 🚱	
90	
	Cancel

6. Copy the token generated. Click on DONE

Create Databricks Connection in Fire Insights

Create a connection in Fire Insights to Databricks.

It can be created by the Administrator under Administration/Global Connections. These connections are available for everyone to use.

It can also be created by any user with their Application. In this case, it is only available to the Application and its users.

- Specify your Databricks Token.
- Specify the Databricks JDBC URL of your cluster in Databricks.

Your token has been created successfully.								
dapif34e533561efd5b5f6c6ae986e13df82								
A Make sure to copy the token now. You won't be able to see it again.								
		Dor						
Add Connection								
CONNECTION TYPE @*	Databricks	Ŧ						
CONNECTION NAME @*	Test_databricks							
TOKEN @*								
TOKEN @*		۲						
TOKEN 😡 *	databricks	۲						
TITLE @*		۲						
		۲						

Now we are ready to start using the Databricks Connection in Fire Insights to:

- Browse DBFS
- View your Databricks Clusters
- Browse your Databricks Databases & Tables
- · Create Workflows which Read from and Write to Databricks

18.1.4 ADLS Integration

Fire Insights integrated with azure data lake storage, once configured you can use the filesystem for accessing data from it.

Below are the steps to Configured adls using managed identity

Managed identity allow the users to access the azure resources without hardcoding any credentials in code.

System identity need to be enabled

System identity need to be enabled on vm where Fire Insights is running or need to be install

adls-test Identit Virtual machine	у
✓ Search (Ctrl+/)	« System assigned User assigned
T SIZE	System assigned User assigned
Security	A system assigned managed identity enak Vault) without storing credentials in code.
💠 Advisor recommendations	role-based-access-control. The lifecycle o
Extensions	Additionally, each resource (e.g. Virtual M more about Managed identities.
🐔 Continuous delivery	🔚 Save 🗙 Discard 💍 Refresh
S Availability + scaling	
💼 Configuration	Status ①
😢 Identity	Off On

In storage account, add the role to provide the access

In storage account, add the role to provide the access to Azure vm with needed access

 Degrate and same provensi. 	ur 😈 uni sasidossio/0.0748 unisas Urr	Owner ©	Subscription (Inherited)
Access Control (IAM)	Storage Account Contributor		
💕 Deta transfer	realized set		
🗲 Events	/subscriptions/Stbadai1-057e-4ot Virtual Machine	Storage Account Contributor 💿	This resource
📓 Storage Explorer (preview)	Storage Blob Data Contributor		
Settings	ads text	Storage Bob Data Contributor 💿	This resource
📍 Access keys			
🌻 Geo-replication	tpartflows ads /subscriptions/Stibeda31-057e-4e8	Storage Blob Data Contributor 💿	This resource
😵 CORS	Storage Blob Data Owner		

login to Fire Insights

login to Fire Insights application and add below parameter in Configuration under administration section for AZURE.

- azure.enabled to true
- azure.homeDir as abfs://containerName@storageAccountName.dfs.core.windows. net

CONFIGUR	ATIONS													
SHAR COMPOUND	NULL HOUSE CL	AFER COMPIC									5	eardh		۹
APP DATA	ROS ARROW	ANS	A2191	SPARK	HOPS	HADOOP	volter	HME	KEPBEPICS	LDAP	ALERT	PLUCING	MODULE	U SETTINGS
NUME	me			www						DESC	artos			
accrearabled	Enable dates			laue .						@ Insi	ite datae hir a	convergible La	re Data labe star	
azure-horneDir	Home Dir			attolive eats	(hfree-weige	ment afficience	ndowsnet			5000	Life ge Mil		age. The path m	
SHA CONFIGURATO	ю													

Save Configuration

Save the above configuration and refresh the page & Click on Data browser to see ADLS page

Fire Insights @ DATA and					(c) (0) (0) (4)
DASHBOARD					
DASHBOARD					
	APPLICATIONS	旦	EXECUTED WORKFLOWS	部	PROCESSORS
ŌŌ	25	00	4	100	312
Vev Ostala +		Verdenit +		VevOetels +	

Click on Data browser

Click on ADLS to see ADLS FILESYSTEM in DATA BROWSERS

Once the above configurations done, you can start using those file while creating dataset and workflow.

	Insights 🛞 Draw arrows				a • O pecunicia • d	6нооца · ≣она силл · ⊝л	CHENSTRATION ·	HOCKSON *	«· • =
ADI	LS FILE SYSTEM						UNDAD FILE	OTAL DISCOUP	•••
1 Ho	ene abts/fire-data@firedewek	pree	Leftsconsolindows.net					Search HDFS	•
	NAME		anter	928	 UNIT UPDATED 	E owner	6904		IONS ACTONS
	C) Test Folder	œ	/TextFolder		2020/1/09 10:09:21:57	speriflovo-ads	1005	dons-eds not e-	0 ≜ 4
	DWND, ARUNA, TLOW	œ	MIND, AGUNA, TUNY	14.44.93	2020/10/22 17:44:46 157	specificing adds	speri	dovsads ner	040
		œ	Amelitandan	97.48	2020/19/10/2010 107	1640360-606-488 4748-654M777	35 speci	idovsiads exerci-	040
	Etidete	0	Abeta		2020/02218-0679197	specificary with	speri	downeds maren	
	Codeg	8	ing.		2020/02/12/2020/17 07	samples and	14.40	downed more-	

CHAPTER 19

Load Balancer Integration

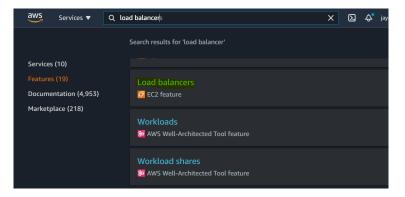
19.1 Load Balancer

Below are steps to Configure Network Load balancer and route using Route 53 in AWS

19.1.1 AWS Network Load balancer

It Explains about Creating Network Load balancer in AWS and Configuring it VM running with Fire Insights. Below are steps involved in Creating Network Load balancer in AWS.

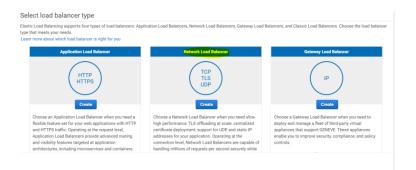
• Login with AWS Console and search for load balancer with EC2 feature.



- Create Load Balancer & select Network Load Balancer.
- Configure Load balancer

```
Add Name
Scheme : internet-facing
IP address type : ipv4
```

(continues on next page)



	(continued from previous page)
Listeners	
Load Balancer Protocol : TLS (SECURETCP) Port: 443	
Availability Zones	
VPC : select VPC where application vm is running.	
Availability Zones : select the specific zone.	
1. Configure Load Balancer 2. Configure Security Settings 3. Configure Routing 4. Register Targets 5. Review	
Step 1: Configure Load Balancer	

Step 1: Conligue	e Load Balancer	
Name (i)	sparkflows-dev	
Scheme (j)	● internet-facing ○ internal	
IP address type (j)	(ipv4 🗳	
Listeners		
A listener is a process that o	hecks for connection requests, using the protocol and port that you c	configured.
Load Balancer Protocol		Load Balancer Port
TLS (Secure TCP)]	443
Add listener		

• Configure Security Settings

Select default certificate.

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer.

Certificate type								
Certificate name								
Security policy								
	1. Configure Load Bulancer 2. Configure Security Settings 3. Configure Routing 4. Register Targets 5. Review							
	Step 2: Configure Security Settings							
	Select default certificate							
	AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more and certificate management.							
	Certificate type 👔 Choose a certificate from ACM (recommended) Ulpolad a certificate to ACM (recommended) Choose a certificate from AAM Ulpolad a certificate to IAM							
	Request a new certificate from ACM AWS Certificate Manager makes it easy to provision, manage, deploy, and renew SSL Certificates on the AWS platform. ACM manages certificate renewals for you. Learn more							
	Certificate name () No existing certificates							
	Select Security Policy							
	Security policy () ELESEcurityPolicy 2016 08 4							

Note: Make sure to add certificate either through ACM or IAM

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/configuring-https-ssl-upload.html

• Configure Routing

```
Target group
Name : A name of target group
Target type : Instance
Protocol : TCP
Port : 80
Register Target
```

Name	sparkflows-dev
ARN	$arn: aws: elastic load balancing: us-east-1:004331324847: load balancer/net/sparkflows-dev/3d556cd21439df 55 \ell^2_{\Box}$
DNS name	sparkflows-dev-3d556cd21439df55.elb.us-east-1.amazonaws.com 🖉 (A Record)
State	active
Туре	network
Scheme	internet-facing
IP address type	ipv4
	Edit IP address type
VPC	vpc-6cd7ad0a 🔀
Availability Zones	subnet-35a03809 - us-east-1e 🖓

· Port forwarding

As Fire Insights by default running on port 8080 for HTTP & 8443 for HTTPS, Make sure forward HTTP or HTTPS to specified port on which Fire Insights is running.

```
sudo firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8443 --permanent
sudo firewall-cmd --reload
```

19.1.2 Route 53

It Explains about Configuring Route 53 to Network Load balancer.

Below ares steps to follow:

• Login to AWS Console and Type R 53 in search box

Sign in to the AWS Management Console and open the Route 53 console at https://console.aws.amazon.com/route53/

aws	Services 🔻	Q R 53 X
Rout		Search results for 'R 53'
Dashb		Services
Hoste	Features (5)	
Health	Documentation (5,074)	Scalable DNS and Domain Name Registration
Traffic	Marketplace (1)	
Traffic		Route 53 Resolver
Policy		Resolve DNS queries in your Amazon VPC and on-premises network.
Domai		

• Get started with R 53 Dashboard

```
Register a domain
```



• Hosted zone

Create hosted zone

losted zones (1)						
View details			esults. To change modes g	o to settings.		
	Ealt	Delete	reate nosted zone			
Q Filter hosted zones by	property or value				< 1	>

• Create records

Create records and Registered Network load balancer to it.

```
Value/Route traffic to : Alias to Network LB
Select Zone
By default load balancer domain name should be populated.
Record type : A -Routes traffic to IPV4 address and some aws resources.
Routing policy : Simple Routing
```

Records (3) DNSSEC signing Hosted zone tags (0)	
Records (3) Info Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings. C Edit Delete record Import zone file Create record Q. Filter records by property or value Type ♥ Routing policy♥ Alias ♥ <1	۲
□ Record name v Type v Routing Differe policy v ntiator v Value/Route traffic to	▽

Record name Info To route traffic to a subdomain, enter the subdomain name. For example, to rou the domain.	te traffic to blog.example.com, enter blog. If you leave this field blank, the default r	ecord name is the name of
blog	.sparkflows.net	
Valid characters: a-z, 0-9, ! " # \$ % & ' () * + , - / : ; < = > ? @ [\] ^ _ ` { } . ~		
Value/Route traffic to Info The option that you choose determines how Route 53 responds to DNS queries.	For most options, you specify where you want to route internet traffic.	
Alias to Network Load Balancer		•
US East (N. Virginia) [us-east-1]		•
Q sparkflows-dev-3d556cd21439df55.elb.us-east-1.amazonaws.c	com.	×
Record type Info The DNS type of the record determines the format of the value that Route 53 re	turns in response to DNS queries.	
A – Routes traffic to an IPv4 address and some AWS resources		•
Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VP	C, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.	
Routing policy Info The routing policy determines how Amazon Route 53 responds to queries.		

CHAPTER 20

Superset

20.1 Superset

Superset enables powerful Visualizations. Superset can connect with Databricks clusters and display data from Tables in Databricks.

Below are steps involved in Installing Superset and Configuring to Databricks.

20.1.1 Installation

Ensure that Superset machine has python 3.6.0+ installed on it.

Steps involved in installing apache superset (centos7)

• Install Superset:

pip install apache-superset

• Initialize the database:

superset db upgrade

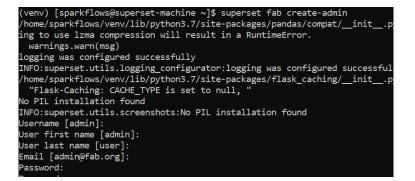
• Create an admin user (you will be prompted to set a username, first and last name before setting a password):

```
export FLASK_APP=superset
superset fab create-admin
```

• Load some data to play with:







superset load_examples

• Create default roles and permissions:

superset init

• Start a development web server on port 8088, using Gunicorn in background:

Once above command runs successfully, ensure that port 8088, on which Superset is running is accessible from your browser

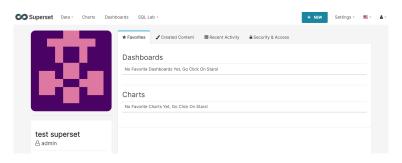
• Open browser and login with public ip and port:

http://public-ip:8088/login

 $\mathbf{\circ}$

Sign In
Enter your login and password below:
a,
SIGN IN

• Use your created credentials to login:



20.1.2 Connecting Superset with Databricks

Once Superset is running, you can configure Databricks database.

Note: Make sure that the Databricks cluster is up.

Install the Python dependencies

Install Needed python dependency for Databricks on the Superset VM:

```
pip install databricks-dbapi
pip install databricks-dbapi[sqlalchemy]
```

C

Once the above two python databricks dependencies have been installed successfully, restart superset server & Login to Superset UI & Click on database

👀 Superset	Data -	Charts	Dashboards	SQL Lab -	+ NEW	Settings -	-	A -
Database	s				T F	ILTER LIST	0	

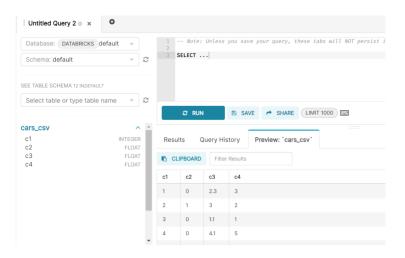
Now you can add databricks database by Clicking on NEW Tab & add Databricks *Database name & SQLAlchemy URI*:

Ø	Superset	Data =	Charts	Dashboards	SQL Lab =				
	Edit Databa	ase							
	Database *		de	default					
	SQLAIchemy	URI *		databricks+pyhive://token:XXXXXXXX@dbc-d5b87134-a Refer to the SqlAlchemy docs for more information on how to					
	Chart Cache	Timeout		hart Cache Time					
) of the caching tim I timeout if undefine	eout for charts of this ed.			

Click on TEST CONNECTION to test your connection. It should not throw any error and SAVE it, Once the database is saved successfully, it would be available in Superset database list page.

\$\$	Superset Data -	Charts	Dashboards S	QL Lab -			+ NEW Settings	; • · ·
	Databases						T FILTER LIST	0
	REFRESH 2							
		Database	I Backend	Expose in SQL Lab	1 Async Execution	1 Creator	Modified	I
		default	databricks	True	False	test supr	erset 20 hours ago	

Now You can start using databricks database tables for charts and visualizations



CHAPTER 21

Python

21.1 Python Integration

Sparkflows supports Python in Workflows in a few ways:

• PySpark Processor

The PySpark Processor allows writing PySpark/Python code to processes the incoming DataFrame and create a new DataFrame. It can also be used to build scikit-learn models etc.

· Jython Processor

The Jython Processor allows writing Jython code to processes the incoming DataFrame and create a new DataFrame.

• Pipe Python Processor

Pipe Python Processor allows writing Python script to process the incoming DataFrame.

The incoming DataFrame is piped to the python script.

The Python script takes in each record of the DataFrame as a comma separated string. It parses the string, processes the record and writes out the new record.

21.1.1 PySpark Processor

Fire Insights provides a PySpark processor for writing PySpark/Python code.

Interface

In the PySpark Processor, we have to implement the myfn function which gets invoked:

```
def myfn(spark: SparkSession, workflowContext: WorkflowContext, id: int, inDF:_
→DataFrame):
```

(continues on next page)

(continued from previous page)

```
* spark : SparkSession object
* workflowContext : Can be used for outputting results to the user
* id : id of the current processor
* inDF : Input PySpark dataframe
```

WorkflowContext

WorkflowContext provides the following methods for outputting data to the user:

```
* def outStr(self, text: str)
* def outNameValue(self, nm: str, val: str)
* def outSchema(self, id: int, title: str, df: DataFrame)
* def outDataFrame(self, id: int, title: str, df: DataFrame)
* def outPandasDataframe(self, id: int, title: str, df: pd.DataFrame)
* def outNumpy1darray(self, id: int, title: str, arr: np.ndarray)
* def outNumpy2darray(self, id: int, title: str, arr: np.ndarray)
```

Example 1

Below is an example code for the PySpark Node.

```
from pyspark.sql.types import StringType
   from pyspark.sql.functions import *
2
   from pyspark.sql import *
3
   from workflowcontext import *
4
5
   def myfn(spark: SparkSession, workflowContext: WorkflowContext, id: int, inDF:_
6
   →DataFrame):
    house_type_udf = udf(lambda bedrooms: "big house" if int(bedrooms) >2 else "small_
7
   →house", StringType())
    filetr_df = inDF.select("id", "price", "lotsize", "bedrooms")
     outDF = filetr_df.withColumn("house_type", house_type_udf(filetr_df.bedrooms))
     return outDF
10
```

Example 2

Below is another example which uses sklearn

```
from pyspark.sql.types import StringType
   from pyspark.sql.functions import *
2
   from pyspark.sql import *
   from workflowcontext import *
4
5
   import numpy as np
6
   import pandas as pd
7
8
   from sklearn.linear_model import LinearRegression
9
   from sklearn import datasets
10
   from sklearn.model_selection import train_test_split
11
   from sklearn import metrics
12
13
  from joblib import dump, load
14
```

(continues on next page)

```
(continued from previous page)
```

```
15
   def myfn(spark: SparkSession, workflowContext: WorkflowContext, id: int, inDF:...
16
   \rightarrow DataFrame):
     # Convert the Spark DataFrame to a Pandas DataFrame using Arrow
17
     dataset = inDF.select("*").toPandas()
18
19
     dataset = dataset.fillna(method='ffill')
20
21
     X = dataset[
22
            ['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
23
    → 'chlorides', 'free sulfur dioxide',
             'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']].values
24
25
     y = dataset['quality'].values
26
27
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
28
   \rightarrow state=0)
29
     # There are three steps to model something with sklearn
30
     # 1. Set up the model
31
     model = LinearRegression()
32
     # 2. Use fit
33
     ft = model.fit(X_train, y_train)
34
     print(ft)
35
     # 3. Check the score
36
37
     scr = model.score(X_test, y_test)
     workflowContext.outStr("Model Score : " + str(scr))
38
39
     # 4. Print model
40
     workflowContext.outStr("Model Coefficient : " + str(model.coef_))
41
     workflowContext.outStr("Model Intercept : " + str(model.intercept_))
42
43
     # 5. Predict test data
44
     y_pred = model.predict(X_test)
45
46
     # 6. See difference between actual and predicted value
47
     df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
48
49
     df1 = df.head(25)
50
     workflowContext.outPandasDataframe(id, "Actual - Predicted : ", dfl)
51
     # 7. Evaluate the performance
52
     workflowContext.outStr("Mean Absolute Error:" + str(metrics.mean_absolute_error(y_
53

→test, y_pred)))

     workflowContext.outStr("Mean Squared Error:" + str(metrics.mean_squared_error(y_
54
   →test, y_pred)))
     workflowContext.outStr("Root Mean Squared Error:" + str(np.sqrt(metrics.mean_
55

→squared_error(y_test, y_pred))))
56
     return inDF
57
```

21.1.2 Jython Processor

Sparkflows has a Jython Processor.

The Jython Processor allows writing Jython code to process the incoming DataFrame. It then produces a resulting DataFrame.

In the Jython node, the following variables are available:

- **inDF** : Incoming Spark DataFrame
- spark : The Spark Session object

Example Jython Code

Below are some example Jython code which can be used.

Select a specific column from the DataFrame

• outDF = inDF.select("c2")

Count the number of records after grouping them

• outDF = inDF.groupBy("c2").count()

Run a SQL on the input DataFrame

The Jython Processor registers the incoming dataframe as a temporary table with a configurable name.

The below SQL in Jython script, performs a SELECT on the registered temporary table.

• outDF = spark.sql("SELECT c1, c2 FROM fire_temp_table")

Run a SQL followed by further grouping and count

- outDF = spark.sql("SELECT c1, c2 FROM fire_temp_table")
- outDF = outDF.groupBy("c2").count()

Read from HDFS and create a new DataFrame

The below Jython script, reads a JSON file from HDFS.

• outDF = spark.read().json("data/people.json")

21.1.3 Pipe Python Processor

Fire Insights has a Pipe Python Processor.

It pipes the incoming DataFrame through pipe to the Python Script. It also passes the Schema of the DataFrame to the Python script through the command line argument. (argv[1])

The Python script is written in the Workflow Editor.

Below is an example workflow containing Pipe Python Processor.



Input DataFrame Schema

The schema of the incoming dataframe is also passed into the Python script as an argument. It can be used in the Python script as needed.

The format of the dataframe schema is below:

colname1:datatype1|colname2:datatype2|colname3:datatype3

Below is an example of printing the arguments and an example result:

Simple Example

The below example reads in the incoming records, parses them, adds a new column whose value is the sum of the first and second fields. Finally it write out the updated record back for Spark to read:

```
#!/usr/bin/python
import sys
for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    fields = line.split(",")
    total = str(float(fields[0]) + float(fields[1]))
    result = ",".join(fields) + "," + total
    print result
```

Below is the code in the Dialog box of the Pipe Python Processor of the Workflow.

Pipe Python @								
Θ								
SCHEMA :								
COLUMN NAME	c1	c2	c3	c4				
COLUMN TYPE	double	double	double	double				
COLUMN FORMAT								
<pre>PIPE PYTHON: ③ PIPE PYTHON: ③</pre>								
				OK CANCE	L			

Output Schema of the Python Script

The output schema of the Python Script is used in the Spark code for recreating the Spark DataFrame from the data received from running the Python script.

It has to be specified in the Pipe Python Processor Dialog.

Program Execution Output

Below is the output produced when executing the workflow.

21.1.4 Pipe Python2 Processor

Fire Insights has a Pipe Python2 Processor.

It pipes the incoming DataFrame through pipe to the Python Script. It also passes the Schema of the DataFrame to the Python script through the command line argument. (argv[1])

The Python script is written in the Workflow Editor.

Below is an example workflow containing Pipe Python2 Processor.

~

Pipe Python @

0

SCHEMA COLUMNS : 🕑 👴						
	OUTPUT COLUMN TYPES @	OUTPUT COLUMN FORMATS @				
c1	DOUBLE	♦ format	•			
c2	DOUBLE	♦ format	•			
c3	DOUBLE	♦ format	•			
c4	DOUBLE	♦ format	•			
c5	DOUBLE	♦ format	•			



CI	C2	C3	C4	C5
DoubleType	DoubleType	DoubleType	DoubleType	DoubleType
1.0	0.0	2.3	3.0	1.0
2.0	1.0	3.0	2.0	3.0
3.0	0.0	1.1	1.0	3.0
4.0	0.0	4.1	5.0	4.0
5.0	0.0	3.1	6.0	5.0
6.0	1.0	2.1	2.0	7.0
1.0	0.0	2.3	3.0	1.0
2.0	1.0	3.0	2.0	3.0
3.0	0.0	13	1.0	3.0
4.0	0.0	4.1	5.0	4.0



Input DataFrame Schema

The schema of the incoming dataframe is also passed into the Python script as an argument. It can be used in the Python script as needed.

The format of the dataframe schema is below:

```
colname1:datatype1|colname2:datatype2|colname3:datatype3
```

Below is an example of printing the arguments and an example result:

Reading in Data in Python into a Pandas DataFrame

Below is an example script which reads in the input lines and converts it to a Pandas DataFrame. It parses the schema passed in argv[1] to extract the column names which is used in creating the Pandas DataFrame:

```
#!/usr/bin/python
import sys
import pandas as pd
dataframe_list_of_rows = []
for line in sys.stdin:
   line = line.strip()
    if not line:
        continue
   row_list = []
    for field in line.split(","):
        row_list.append(field)
    # convert list to tuple
   row_tuple = tuple(row_list)
    dataframe_list_of_rows.append(row_tuple)
# generate column names
schema = sys.argv[1]
column_names = []
schema_columns = schema.split("|")
for column_name_with_type in schema_columns:
    column_name_with_type_split = column_name_with_type.split(":")
    column_names.append(column_name_with_type_split[0])
# create dataframe from the input rows
input_dataframe = pd.DataFrame.from_records(dataframe_list_of_rows, columns=column_
\rightarrownames)
```

Transform the Pandas DataFrame

Now that we have the Pandas DataFrame in input_dataframe, we can transform it to create the result DataFrame - output_dataframe. In the below example, we are just setting the output dataframe to the input dataframe:

```
output_dataframe = input_dataframe
```

Writing the Pandas DataFrame schema back to Spark

Below is an example code for writing the Pandas Schema back to Spark. It is used in inferring the schema output of the Python code. This way users do not have to reenter the schema of the output in the Workflow:

```
dataframe_dtypes = output_dataframe.dtypes
f = open(sys.argv[2],'w+')
f.write(str(dataframe_dtypes))
f.close()
```

Fire expects each line of the schema file to contain the following:

- Name of the column
- Data Type of the column

There can be multiple spaces between the name and the data type.

Fire uses the below for mapping from the data type to Spark DataFrames Data Types:

- int : integer
- float : float
- double : double
- boolean : boolean
- string : string

Writing the Pandas DataFrame back to Spark

Below is an example code for writing the Pandas DataFrame back to Spark:

```
# iterate over the dataframe created and return it to the pipeNode
for index, row in output_dataframe.iterrows():
    list = row.tolist()
    row_string = ','.join(str(e) for e in list)
    print(row_string)
```

Output Schema of the Python Script

The output schema of the Python Script is written to a file which is read by the Spark Code. Clicking on **Refresh** Schema infers the Python Schema output into Spark.

SCHEMA COLUMNS : REFRESH SCHEMA									
	OUTPUT COLUMN TYPES 🕑		OUTPUT COLUMN FORMATS 🕑						
c1	STRING		format	•					
c2	STRING		format	•					
c3	STRING		format	•					
c4	STRING		format	•					

OK CANCEL

CHAPTER 22

Performance

22.1 Performance Tuning

Performance is the cornerstone of any Big Data Processing. Fire is extremely optimized for best performance. Each of the Processors are written for extreme performance, the engine is optimized for the best performance.

There are certain things which need to be taken into account for any Spark job. Fire makes it extremely easy to apply them to a Workflow.

22.1.1 Caching Level

Setting the right caching level of the Dataset outputs of the nodes is very important for performance in Apache Spark. Fire allows you to set the caching output of the Dataset of any Processor.

When to use Caching

In general the default Caching does not have to be changed.

It is important to set Caching in the following scenarios:

- If the Dataset is going to be reused later. Below are some examples.
 - A Dataset is read from HBase. Then another dataset is read and the two are joined. In this case it is a good
 idea to Cache the dataset read from HBase.
 - A Dataset is joined with another Dataset. The result is then joing with another Dataset. In this case it is a
 good idea to cache the result of the first Join.
- A Dataset which is used in machine learning.
- Whenever a Dataset computation is expensive (JOIN etc.), caching can help in case the executor fails, the blocks are evicted from memory.

22.1.2 Executor Memory, vcores

When running Apache Spark jobs, we can define the number of executors, executor memory and number of vcores per executor.

Normally dynamic allocation of executors is enabled, and we do not need to specify the number of executors.

Certain jobs need higher executor memory and number of vcores. These can be specified with --executor-memory and --executor-vcores.

These additional spark configs can be specified in Fire in the Execute page. They can also be specified when the jobs are scheduled for execution.

22.1.3 Repartioning

Repartioning splits the datasets into the specified number of partitions.

This can help with performance

When saving to JDBC/File etc.

When saving a Dataset, the parallelism depends on the number of partitions of the Dataset. In case there are too few partitions, repartitioning the Dataset before saving would increase the parallelism.

Parallelism is also a double-edged sword. It is not a good idea to say have too many parallel connections to a Relational Database as it would put heavy load on the RDBMs.

22.1.4 Debug Mode

Fire Insights allows you to run the workflow in Debug mode.

In the debug mode a count is performed on the output from each Processor. This helps to know which Processor is exactly taking more time.

Apache Spark in general executes the DAG lazily. It starts the execution of the DAG only when it hits an Action. Hence, many times we do not know which Processor is actually taking more time.

Forcing Action with count in Debug mode forces execution of that step and insights into the time taken by the Processor.

CHAPTER 23

Developer Guide

23.1 Developer Guide

23.1.1 Custom Node Development in Browser

Fire Insights enables you to write custom nodes from your Browser.

You would provide the execute method for the Processor and the Schema update code. You would also provide the details of the widgets through which the user would provide the parameters for the new custom node.

Below are the steps for creating the custom node.

Once you login to Fire Insights application, there is PROCESSORS menu on top, select Custom Processors.

Fire Insights 🛛 DATA BROWSERS	- ■ APPLICATIONS - Ø	SCHEDULER - 🛛 EXECUTI	IONS - 🚓 MODELS - 🗃 DATA QUALITY -		- PROCESSORS -	o:• ⊕• ≣• ≛•
Dashboard					Category	
	24 APPLICATIONS	B	25 EXECUTED WORKFLOWS	.	List Custom Processors	296 processors
• My Applications	0	View Details	0	View Detoils		

Click on CREATE PROCESSORS

Click on CREATE PROCESSORS to start creating the new processor.

🕚 Fire Insights	(2) DATA BROWSERS		Ø SCHEDULER	O EXECUTIONS -	🗞 MODELS	-	🛢 DATA QUALITY	÷		l	PROCESSORS	-	¢-0-≣	1 × 4
Custom Pro	cessors					C	Search						REATE PROCESS	ons -
ID USER NAME	NAME	TYPE	DESCRIPTION		STATUS	DATI	E CREATED		LAST UPDATED		ACTIONS			

It would open up the Create Processor Page as below.

Enter the name and other details for the new processor.



Then provide details for the various fields of the new processors. These fields would appear in the processor dialog when used in the workflow editor.

rocessor C	ode Schema						_
IAME: *		TYPE: *	Transform	~	ENGINE: *	Pyspark	
NPUT:	Input Description	OUTPUT:	Output Description		DESCRIPTION:		

Click on the + sign to add a new field. For each field provide the following:

- WIDGET
- NAME
- TITLE
- VALUE
- DESCRIPTION

WIDGET: *	~	NAME: *	TITLE: *	
VALUE:	Use comma for multiple value	DESCRIPTION:		

Finally click on the Next button to go to the Code tab.

Execute Code

The Code tab is where you write the execution code for the new Custom Processor.

It shows the default template which you can update

Then click on Next button to go to the Schema tab.

Schema Update Code

The Schema tab is where you add the code which updates the incoming schema to produce the output schema from this processor.

It displays the default template code which you can update.

Finally click on the Submit button to finish creating the new custom processor.

Once the custom processor is submitted successfully, it will be visible in Custom Processors list page.

Create Processor						BACK
	DATASET:	Select Dataset	*	TEST	SAVE DRAFT	
Processor Code Schema						
<pre>i rom pryset.stl lopet * from fifseworthousetta lippet * from fifseworthousetta lippet * i for fifseworthousetta lippet * i for fifseworthousetta lippet i fifseworthousetta werklascenteet, idi int, indii Batafraea, parameters:</pre>						
					BACK	NEXT
Create Processor						BACK
	DATASET:	Select Dataset	~	TEST	SAVE DRAFT	
Processor Code Schema						
<pre>i from fire_avertilenengine import lokContext, FuputSchemaContext, FireSchema i for Schemac[poptSchema:FireSchema_popset[</pre>						

Testing the custom processor

Fire Insights enables you to seamlessly Test your custom processor.

When editing the custom processor, select the Dataset for the data you want to feed to the custom processor. Then click on Test to view the output of the new custom processor.

Using the new Processor

The processor is now available in the Workflow Editor.

You can click on the custom processor to start using it in your workflow.

You can also export & import them

Export Custom Processors

Fire Insights enables you to export Custom Processors from Browser to local machine.

Below are the steps to export Custom Processors.

Login to Fire Insights & go to Custom Processors list page.

Select the Custom Processors which you want to export and click on export.

NOTE: you can export multiple Custom Processors at a time too.

Once you click on export button, the selected Custom Processors will be downloaded to local machine in zip format.

Custom Processors		Q Search		CREATE PROCESSORS *
ID USER NAME NAME	TYPE DESCRIPTION	STATUS DATE CREATED	LAST UPDATED	ACTIONS
226 sparkflows FDS	pyspark	LIVE 2020/08/18 15:15:21 IST	2020/08/18 15:15:21 IST	• / B

Edit Custon	n Processor : FDS							BACK
				DATASET:	Housing	~	TEST	SAVE DRAFT
Processor Co	de Schema							
NAME: *	FDS	TYPE: *	Transform	~	ENGINE: *	Pyspark		~
INPUT:	Input Description	OUTPUT:	Output Description		DESCRIPTION:			
Fields								0
								NEXT

Application	Id											
	1596624144228 @	9957										
uiWebUrl												
xecuting No	de DatasetStruct	tured : 1 undefine	d									
xecuting No	de DatasetStruct	tured : 1 undefine	d									
-	de DatasetStruct	tured : 1 undefine	d									
Aug 18, 202	0 09:58:03 AM	tured : 1 undefine	d									
Aug 18, 202 Nug 18,	0 09:58:03 AM		d bedrooms	bathrms	stories	driveway	recroom	fullbase	gathw	airco	garagepl	prefarea
Aug 18, 202 Aug 18, 202 Aug 18,	0 09:58:03 AM 2020 09:5	58:03 AM		bothrms StringType	stories StringType	driveway StringType	recroom StringType	fullbase StringType	gashw StringType	airco StringType	garagepl StringType	prefarea StringType
Aug 18, 202	0 09:58:03 AM 2020 09:5 price	58:03 AM	bedrooms									

Create Workflow							•
AME: Enter Workflow Name		CATEGORY:	Enter Category				
O ADD NODE			📕 SAVE	SAVE WITH COMMENT	CLEAR BACK	EXECUTE	BEAUTIFY
All Nodes Datasets Hive DB	*						
Q FDS							
A CustomProcessors	¥						
di pyspark	~						
AL FDS							
iii 105							

Custom Processors		Q Search		CREATE PROC	ESSORS - IMPORT EXPOR
D ID USER NAME NAME	TYPE DESCRIPTION	STATUS	DATE CREATED	LAST UPDATED	ACTIONS
32 sporkflows Nest	transform	LIVE	2020/08/26 10:38:02 IST	2020/06/26 10:38:02 IST	• / 8
) 134 sparkflows FDS	transform	LIVE	2020/06/26 II:37:49 IST	2020/08/26 18:37:49 IST	• / 8

Fir	e Insig	ihts 🕲 DA1	A BROWSERS -	III APPLICATIONS - O SC	HEDULER -	is - 🚷 Models	- 🛢 DAT	a quality 🕘 😣 Admin	ISTRATION - I	PROCESSORS -	o¦+ ⊕+ ≡+ 4
Cu	storr	Processo	ors			Q Search				CREATE PROCESSORS -	
	ID.	USER NAME	NAME	TYPE	DESCRIPTION		STATUS	DATE CREATED	LAST UPDATED	ACTION	is .
•	132	sporkflows	testi	transform			UVE	2020/08/26 10:38:02 IST	2020/08/26	10:38:02 IST 🗶 🏉	18
	134	sporkflows	FDG	transform			LIVE	2020/08/26 11:37:49 IST	2020/08/26	11.37:49 IST 🗶 🏉	18

re Insig Istorr	Processor	s		Successi	×			CREATE PROC	ssors -	
ID	USER NAME	NAME	TYPE	Exported 2 Custom Processor Successfully			LAST UPDAT	то	ACTIONS	
132	sparkflows	test1	transform		ок	38:02 IST	2020/08/2	10:38:02 IST	• /	8
13.4	sparkflows		transform			J7:49 IST	2020/08/3	6 11:37:49 IST	• /	ŝ
										t of t \rightarrow \rightarrow

Import Custom Processors

Fire Insights enables you to import Custom Processors to different environment.

Below are the steps to Import Custom Processors.

Login to Fire Insights & go to Custom Processors list page.

Fire Insights 🔌 DATA BROWSERS 👻 🎟 .	APPLICATIONS - O SCHEDULER - O EXECUTI	ons - 💩 models - 🛢 data quality - 😁	IDMINISTRATION - 🗧 PROCESSOR	s - ot- e- ≡-
Custom Processors		Q Search	CREATE PRO	CESSORS - IMPORT EXPO
ID USER NAME NAME	TYPE DESCRIPTION	STATUS DATE CREATED	LAST UPDATED	ACTIONS
ID USER NAME NAME	TYPE DESCRIPTION	STATUS DATE CREATED	LAST UPDATED	ACTIONS

Select the IMPORT button, it will open a new windows to upload zip file from local machine.

сизтот Processors в изстание тите осс г с с с с	Q		STATUS	DATE CREATED	LAST	CREATE PRO	ACTION	INFORT S	DOOR
	CRIPTION	1	STATUS	DATE CREATED	Let	T UPDATED	ACTION	3	
Fre Insights 2: онланочаев - №Анистрол В Контолея									
Біне Інадала 2 сала шахнава - Шаймаллога — В контоков -									
Пое Insights фоллиночани и Шамисларов – В Контоляя									
Five Insights €10474.0000485 - 100.4994.201000- 100.001(194118-									
Fire Insights & DATA BROWSER - III APPLICATIONS - OF SCHEDULTE -									
Fire Insights ④ DATA BROWSERS - 111 APPL <u>CATIONS - © SCHEDULER</u> -									
Fire Insights @ DATA BROWSERS - III APPLICATIONS - @ SCHEDULER -									
	 Ø EXECUTIONS - 	A MODELS	- 🖀 DATA OU	liality - 🕅 AE		 PROCESSORS 	•	o°- 0-	B -
Custom Processors Upload File					×	CREATE PROC	ESSORS -		EXPO
D USER NAME NAME Choose File No file chosen						UPDATED	ACTIONS		
				IMPORT	CANCEL				

Once you upload zip file of Custom Processors from local machine, press IMPORT button to import it.

NOTE: You can import multiple Custom Processors at a time too.

Once you Click on IMPORT button, success message will display on imported Custom Processors.

After success import, you can view those Custom Processors in Custom Processors list page.

Now you can use those Custom Processors in your workflow.

23.1.2 Custom Node Development & Deployment (Java/Scala)

Fire Insights follows an open and extensible architecture allowing developers to add new custom nodes/processors that can be exposed in Fire UI and embedded into workflows.

The details for building new nodes are available at the URL below:

• https://github.com/sparkflows/writing-new-node

Examples of more complex nodes are at the URL below :

• https://github.com/sparkflows/sparkflows-stanfordcorenlp

Step 1 : Start by cloning the github repo: writing-new-node

The easiest way to start writing a new node or processor is by cloning the writing-new-node repo using the command below:

• git clone https://github.com/sparkflows/writing-new-node.git

Fire Insights	III APPLICATIONS - O SCHEDULER - O EXECUTIONS - ANN	AODELS - 🗏 DATA QUALITY -	 FA ADMINISTRATION 	 PROCESSORS 	- o;- ⊕- ≣-
Custom Processors	Upload File		×	CREATE PROCE	ESSORS - IMPORT EXPO
D ID USER NAME NAME	Choose File CustomProcessor_121238.zip CUSTOM PROCESSORS IN THE ZIP FILE			30 ACTI	ONS
	FDS fost				
		I	IMPORT CANCEL		
Fire Insights 91 DATA BROWSERS -	Шаррикалсак - Оксырлире - Фресплсак - Фа	KODELS - 🗮 DATA OKIALITY -		PROCESSORS	- α²-ρ-≡-
Fire Insights @DATA BROWRESS - Custom Processors	MAPLICATIONS · O SCHOOLE - O ROCUTIONS - & A	ncoels - 🗏 data quality - >	-	PROCESSORS CREATE PROCES	
Custom Processors			<	_	
Custom Processors	Import Information!	2 2	LAST	CREATE PROCES	SSORS - IMPORT EXPO
Custom Processors ID USER NAME NAME USER NAME FDS	TYPE TOTAL SUCCESS	2	<	CREATE PROCES	ACTIONS
Custom Processors ID USER NAME NAME IBI sportflows FDS	TYPE Socialism FALLINE FALLINE	2 2	<	CREATE PROCES	ACTIONS

Step 2 : Install the Fire core jar to the local maven repository

Insall the Fire core jar to your local maven repository. The pom.xml contains the dependency for it.

• mvn install:install-file -Dfile=fire-spark_2.4-core-3.1.0.jar -DgroupId=fire -DartifactId=fire-spark_2.4-core - Dversion=3.1.0 -Dpackaging=jar

Step 3 : Code the new custom node

The custom node might be a Dataset node or a Transform node.

A Dataset node reads data from some source into a Dataframe. It passes on this new Dataframe to the next node. Examples of data sources include:

- Files on HDFS
- HIVE tables
- HBase tables
- Cassandra
- MongoDB
- Salesforce / Marketo

A Transform node receives an input Dataframe(s), transforms it and sends the transformed Dataframe to the next node.

Writing a Dataset node

Create a new class that extends the NodeDataset class.

- Override the execute () method. The execute () method will read in data from the defined source into a Dataframe. It would then pass on the resulting DataFrame to output node(s).
- Override the getOutputSchema() method to return the schema of of the Dataframe created by the node.

Fir	e Insig	ihts 🕲 DATA	BROWSERS +	III APPLICATIONS - O SCHEE	IULER - © EXECUTION	s + 🚓 models + 🛢	DATA QUALITY 🔹 😐	ADMINISTRATION - 🔳	PROCESSORS -	o(+ ⊕+ ≣+
Cu	istom	Processo	'S			Q Search			CREATE PROCESSORS *	IMPORT EXPOR
	ю	USER NAME	NAME	TYPE	DESCRIPTION	STATUS	DATE CREATED	LAST UPDATED	ACTIO	res S
	161	sporkflows	FDS	transform		LIVE	2020/08/26 17:18)	01 IST 2020/08/26 17	18:01 IST 🛛 👁 🥖	• e
	162	sporkflows	test1	transform		LIVE	2020/08/26 17:18	01 IST 2020/08/26 17	r:18:01 IST 🛛 🗶 🥖	• e

Writing a Transform node

Create a new class that extends the Node class.

- Override the execute() method. The execute() method will transform the incoming DataFrame and then pass on the resulting DataFrame to output node(s).
- If the node is updating the incoming schema, also override the getOutputSchema() method. Otherwise the incoming schema to this node is sent to the next node(s).

Examples of Custom Nodes

Example of custom nodes are available at:

• https://github.com/sparkflows/writing-new-node/tree/master/src/main/java/fire/nodes/examples

Step 4 : Create the node JSON file

Create the JSON file for the new node. The JSON file is used for displaying the new node in the Workflow Editor and capturing the user inputs of the various fields of the node through a Dialog box. The JSON for the node also captures the name of the Java/Scala class which has the implementation code for the Node.

Fire supports various widgets types for capturing the details of the fields from the user through the Node Dialog Box.

Widget Types

The details of the various widget types is available at the URL below:

• https://github.com/sparkflows/writing-new-node/blob/master/docs/README_Processor_JSON.md

Examples of Node JSON

- https://github.com/sparkflows/writing-new-node/blob/master/json/nodes/testprintnrows.json
- https://github.com/sparkflows/writing-new-node/blob/master/json/nodes/testmovingaverage.json

Step 5 : Deploy the Custom Node in the Fire Server

Now that you have created a new node, follow the steps below to deploy it into the Fire Server:

- Create a jar file with mvn clean package
- Copy the jar file created in the previous step (target/writing-new-node-3.1.0.jar) into fire-user-lib directory of Fire Insights.
- Place the JSON file for the new node under the nodes directory.
- Restart the Fire Server.

The new node would be picked up by the Fire Server and be visible in the Workflow Editor. Check that new node is available as expected in the Workflow Editor.

Use the custom node in Spark submit when running on the Spark cluster

- Select the custom node jar checkbox when executing the workflow containing the custom node.
- You can also include the custom node with --jars <...> when running the workflow on the cluster

23.1.3 Databricks Custom Node Example JSON

Custom Nodes in Fire Insights can be exported as zip files and then subsequently imported into Fire Insights. Click on the clink below to download a custom node zip file containing scorecardpy binning custom node. Import it into Fire Insights by going to Processors/Custom Nodes.

The code looks like below:

Execution Code

```
from pyspark.sql import DataFrame, SparkSession
1
    from fire.workflowcontext import WorkflowContext
2
    import scorecardpy as sc
3
л
    def myfn(spark: SparkSession, workflowContext: WorkflowContext, id: int, inDF:_
5
   →DataFrame, parameters: dict):
        # Write your code here by using input dataframe i.e inDF and pass the output_
6
   ↔ result as outDF dataframe.
7
        pandas_df = inDF.toPandas()
8
        variables = ["purpose"]
9
        stopLimit = 0.1
10
        countDistrLimit = 0.05
11
12
        binNumLimit = 8
        method = "tree"
13
        positive = "bad|1"
14
        workflowContext.outStr(id, "Method: " + parameters['method'] + ", Positive:" +...
15

→parameters['positive'])

16
        bins = sc.woebin(pandas_df, y="creditability", x=variables, stop_
17
   →limit=float(stopLimit),
                      count_distr_limit=float(countDistrLimit),
18
                      bin_num_limit=int(binNumLimit), method=method, positive=positive)
19
20
        bins_ply = sc.woebin_ply(pandas_df, bins)
        spark_df = spark.createDataFrame(bins_ply)
21
        outDF = spark_df
22
23
        return outDF
```

Schema Propagation Code

```
from fire.workflowengine.workflow import JobContext
from fire.workflowengine.fireschema import FireSchema
def schema(inputSchema: FireSchema, parameters: dict):
    #to add new column
    #inputSchema.append("house_type", "string")
```

(continues on next page)

1

2

4

5

6

(continued from previous page)

```
#to drop a column
#inputSchema.drop("id")
inputSchema.append('purpose_woe', 'double')
return inputSchema
```

23.1.4 Building and Running Custom Node

Fire Insights allows you to build your own Custom Nodes.

In this tutorial we would build a custom node built upon scorecardpy.

Install the scorecardpy dependencies

Since we are using the library scorecardpy, we would install its packages both on the Fire Insights machine and on the Databricks cluster.

Use the command below to install it on the Fire Insights machines:

```
• pip install scorecardpy
```

Install it on your Databricks cluster with the below:

· Open a Notebook

8

9

10 11

12

%sh pip install scorecardpy

Go to Custom Processors

Once you login to Fire Insights application, there is PROCESSORS menu in top, select Custom Processors.



Click on CREATE PROCESSORS

Click on CREATE PROCESSORS to start creating the new processor.

Fire Insights	② DATA BROWSERS -	# APPLICATIONS •	Ø SCHEDULER	O EXECUTIONS -	& MODELS	- 🗃 DATA QUALITY -	O ADMINISTRATION ·	PROCESSORS -	o\$- ⊕- ≣- ≜
Custom Proc	cessors					Q Search			CREATE PROCESSORS -
ID USER NAME	NAME	TYPE	DESCRIPTION		STATUS	DATE CREATED	LAST UPDATED	ACTIONS	
Fire Insights	街 DATA BROWSERS 👒	# APPLICATIONS -	Ø SCHEDULER *		& MODELS	* 🛢 DATA QUALITY -	O ADMINISTRATION -	PROCESSORS •	¢;• ⊕• ≣• .
		III APPLICATIONS -	O SCHEDULER •		& MODELS	• 📑 DATA QUALITY •	O ADMINISTRATION -	PROCESSORS -	¢‡ · ⊕ · ■ - 4 Create processors ·
Fire Insights Custom Proc		HI APPLICATIONS -		● EXECUTIONS -	& MODELS		ADMINISTRATION -	PROCESSORS -	_
Custom Proc	essors			© EXECUTIONS -		Q Search			CREATE PROCESSORS -
Custom Proc	essors			• O EXECUTIONS •		Q Search			CREATE PROCESSORS -
Custom Proc	essors			© EXECUTIONS +		Q Search			CREATE PROCESSORS -

It would open up the Create Processor Page as below.

Enter the name and other details for the new processor.

Then provide details for the various fields of the new processors. These fields would appear in the processor dialog when used in the workflow editor.

				DATASET:	Select Dataset	~	TEST	E DRAFT
cessor Coo	le Schema							
AME: *		TYPE: *	Transform	~	ENGINE: *	Pyspark		
IPUT:	Input Description	OUTPUT:	Output Description		DESCRIPTION:			

Click on the + sign to add a new field. For each field provide the following:

- WIDGET
- NAME
- TITLE
- VALUE
- DESCRIPTION

WIDGET: *		~	NAME: *	тп	'LE: *		
ALUE:	Use comma for multiple value		DESCRIPTION:				

Finally click on the Next button to go to the Code tab.

Execute Code

The Code tab is where you write the execution code for the new Custom Processor.

Its updated for scorecardpy here.

It shows the default template which you can update for scorecardpy.

Create Processor					BAC
	DATASET:	Select Dataset	~	TEST SAVE I	IRAFT
Processor Code Schemo					
1 from paper-kal leget * * * * * * * * * * * * * * * * * * *					
					ACK NB

Then click on Next button to go to the Schema tab.

Schema Update Code

The Schema tab is where you add the code which updates the incoming schema to produce the output schema from this processor.

It displays the default template code which you can update.

Create Processor					BACK
	DATASET:	Select Dataset	~	TEST SAVE DRAFT	
Processor Code Schema					
i fra firs.orflaengin import Jukontart, Ispatishamidantart, Firsisham i fra firs.orflaengin import Jukontart, Ispatishamidantart, Firsisham i di genostianta apanol(Nous.type", "string") i di di genostianta apanol(Nous.type", "string") i di genostianta apan					
				BACK SU	UBMIT

Finally click on the Submit button to finish creating the new custom processor.

Once the custom processor submitted successfully, it will be vissible in Custom Processors list page.

Custom Processors		Q Search		CREATE PROCESS	SORS * IMPORT EXPORT
D ID USER NAME NAME	TYPE DESCRIPTION	STATUS	DATE CREATED	LAST UPDATED	ACTIONS
65 sparkflows ScoreCard_Binning	transform	LIVE	2020/08/28 19:04:54 IST	2020/08/28 19:04:54 IST	• / 8

Using the new Processor

The processor is now available in the Workflow Editor.



You can click on the custom processor to start using it in your workflow & submit the job.

	1087					··· 00	NG
METLOW NAME &	Test-Baser/Cord						
NRK-500HIT-00NT: 0							
	eg: executor-memory 2g	executor-cores 2 drive	e-memory 2p				
DORMAN PARAMETERS: Ø							
CORE JAN FEED @	- swedencertwa						
ALL ON SUCCESS: 0							
WE ON FAILURE O							
orkflow Result Logs							
Submitting the link to the Subdricks cluster							
0							
Submitted the Sub to the Establishs Electre :	H30-H3HH-LakME1 Response	Realized in (Desc.) (Col.)	30)				
Application Id							
#72-30098838130688 8000							
#P2-302986381506888 #600							
© www.or							
© www.or							
© www.or							
Intel I							
united t When (718-201-201-201-4002	1018		4	52	2		
Intel I	tore		•	53	2		
Intel I	59		4	53	2		
De united the second se							
De united the second se							
Johnson Markurstanderschuten 11 underheit T	nine		2	25	2		
De united the second se							
Constant May (15:20:20:00) Kenneling Truth Resultings (20:00) T T T T T T	Note Note		2	25	2		
Johnson Markurstanderschuten 11 underheit T	nóre		2	25	2		
Constant May (15:20:20:00) Kenneling Truth Resultings (20:00) T T T T T T	Note Note		2	25	2		
Constant May (15:20:20:00) Kenneling Truth Resultings (20:00) T T T T T T	Note Note		2	25	2		
Constant May (15-25) - 20-20 (102 May (15-25) - 20-20 (102 T T T T T T T	Note Note		2	25	2		
Constant May (15:20:20:00) Kenneling Truth Resultings (20:00) T T T T T T	Note Note		2	25	2		

CHAPTER 24

Processors

24.1 Processors

24.1.1 16-Utilities

03-Execution

ExecuteInLoop

Туре

transform

Class

fire.nodes.etl.NodeLoop

Fields

Name	Title	Description
loopCols	Loop Columns	

ReadParameters

Reads in the parameters from the given file.

Input

Input file has records in the following form on each line : name=value

Output

It adds the input parameters into the JobContext

Туре

shellcommand

Class

fire.nodes.util.NodeReadParameters

Fields

Name	Title	Description
path	Path	Path of the parameters file containing the pa-
		rameter name and value in each line

SpecifyParameters

Provides additional parameters to the workflow. When running with spark-submit, variables can also be given on the command line with -var name=value.

Туре

doc

Class

fire.nodes.util.NodeSpecifyParameters

Fields

Name	Title	Description
names	Parameter Names	Parameter Names
values	Parameter Values	Parameter Values

ExecuteWorkflow

Fires the given workflow. Does not wait for the workflow to complete to resume execution

Туре

transform

Class

fire.nodes.util.NodeExecuteWorkflow

Fields

02-Data-Partition

Coalesce

This node coalesces the DataFrame into specified number of Partitions

Input

This type of node takes in a DataFrame and transforms it to another DataFrame.

Output

The output DataFrame has the specified number of partitions

Туре

transform

Class

fire.nodes.etl.NodeCoalesce

Fields

Name	Title			Description
numPartitions	Number	of	Parti-	input for number of partitions
	tions			

Details

This node coalesces the DataFrame into specified number of Partitions.

It is specially helpful for the case when too many small files are being created. In such a scenario, the Coalesce node can be used to limit the number of output files produced.

Repartition

This node repartitions incoming dataframe into a specified number of partitions

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeRepartition

Fields

Name	Title		Description
numPartitions	Number of Pa	ti-	Number of Partitions
	tions		

NumberOfPartitions

This node will get the number partitions in input dataframe.

Туре

transform

Class

fire.nodes.util.NodeGetNumberOfPartitions

01-Spark-Performance

CacheDataFrame

Caches the DataFrame with the provided StorageLevel

Input

It takes in a DataFrame as input

Output

The input DataFrame is cached with the specified storage level and send to the output

Туре

transform

Class

fire.nodes.util.NodeCacheDataFrame

Fields

Name	Title	Description
storageLevel	Storage Level	storage level name

PrintSparkConfiguration

Print the all spark configuration used in workflow.

Туре

transform

Class

fire.nodes.util.NodeSparkConfiguration

UnpersistDataFrame

Unpersists the output DataFrames of the given Nodes

Input

It takes in a DataFrame as input

Output

The outputs the incoming DataFrame

Туре

transform

Class

fire.nodes.util.NodeUnpersistDataFrame

Fields

Name	Title	Description
nodeIdsToUnpersist	Node ID to Unper-	Output of node to unpersist
	sist	

24.1.2 09-DataProfiling

ColumnsCardinality

Distribution of categorical data. Calculates the count of records for each unique value for the column specified.

Туре

transform

Class

fire.nodes.ml. Node Columns Cardinality

Name	Title	Description
maxValuesToDisplay	Max Values To Dis-	Maximum number of values to display in re-
	play	sult.
inputCols	Column Names	Name of columns for the cardinality data

SummaryStatistics

Summary statistics provide useful information about sample data. eg: measures of spread.

Туре

transform

Class

fire.nodes.ml.NodeSummary

Fields

Details

Summary statistics provides useful information about sample data. eg: measures of spread.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-statistics.html#summary-statistics

Summary Node provides a table consist of informations such as number of non-null entries (count), mean, standard deviation, and minimum and maximum value for each numerical column.

SkewnessAndKurtosis

Туре

transform

Class

fire.nodes.etl.NodeSkewnessAndKurtosis

Fields

Name	Title	Description
inputCols	Column Names	Name of columns to get the skewness and
		kurtosis.

HistoGram

Computes a histogram of the data using number of bins evenly spaced between the minimum and maximum of the specific columns.

Туре

transform

Class

fire.nodes.ml.NodeHistoGramCal

Fields

Name	Title	Description
inputCols	Column Name	Name of column
bins	Number of Bins	Number of Bins

FlagOutlier

Flag the outlier based on the selected column using Box-and-Whisker technique.

Туре

transform

Class

fire.nodes.ml.NodeFlagOutlier

Fields

Name	Title	Description
inputCol	Input Column to	The Input Column to flag the outlier
	flag the outlier	
lowerQuantile	LowerQuantile	
upperQuantile	UpperQuantile	

DistinctValuesInColumn

Туре

transform

Class

fire.nodes.etl.NodeDistinctValues

Fields

Name	Title	Description
distinctCols	Column Names	Name of columns to get the distinct combi-
		nation of values.

NullValuesInColumn

Number of Null Values in Selected Columns.

Туре

transform

Class

fire.nodes.etl.NodeNullValuesInColumn

Fields

Name	Title	Description
inputCols	Column Names	Name of columns for Number of Null Val-
		ues Check

CrossTab

Categorical V.S. Categorical

Туре

transform

Class

fire.nodes.ml.NodeCrosstab

GraphWeekDayDistribution

This node Finds the distribution of Week Days from Date values

Туре

transform

Class

fire.nodes.graph.NodeGraphWeekDayDistribution

Fields

Correlation

calculates the correlation between two series of data.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

The input DataFrame is passed along to the next Processors

Туре

transform

Class

fire.nodes.ml.NodeCorrelation

Fields

Details

This node calculates the correlation between two series of data in a common operation in Statistics. More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-statistics.html#correlations

GraphYearDistribution

This node Finds the distribution of Years from Date values

Туре

transform

Class

fire.nodes.graph.NodeGraphYearDistribution

Fields

GraphMonthDistribution

This node Finds the distribution of months from Date values

Туре

transform

Class

fire.nodes.graph.NodeGraphMonthDistribution

Fields

24.1.3 05-FeatureEngineering

WordCount

Туре

transform

Class

fire.nodes.ml.NodeWordCount

Fields

MovingWindowFunctions

This node calculates the moving values of selected functions for the field(input column).

Input

It accepts a DataFrame as input from the previous Node

Output

A new columns is added which contains the results of applying the selected function on the given column of the input DataFrame

Туре

transform

Class

fire.nodes.etl.NodeMovingWindowFunctions

Fields

Name	Title		Description
windowStart	Window St	art	value to be used to calculate the window
			from
windowEnd	Window Er	nd	value to be used to calculate the window to
partitionCol	Partition	Column	partition column to split the incoming
	Name		dataframe for the sliding/window operation
orderCol	Order	Column	the order of the selected column for the slid-
	Name		ing/window operation
inputCols	Input Colu	mns	input column name for calc
functions	Functions		

DateToAge

This node converts a date-column into columns of age (both in years and in days).

Туре

transform

Class

fire.nodes.etl.NodeDateToAge

Name	Title	Description
inputColName	Input Column Name	Input Column Name
yearsOutputColName	Years Output Col-	Num Years Output Column Name
	umn Name	
daysOutputColName	Days Output Col-	Num Days Output Column Name
	umn Name	

Details

Calculates age from the given date or timestamp column. Age is calculated and displayed in years and days columns.

Examples

Examples when date is 06-25-2019

dd-MM-yyyy : 20-09-2018 , 0 year : 278 days MM-dd-yyyy : 09-30-2018 , 0 year : 268 days yyyy-MM-dd : 2012-01-31 , 7 year : 2702 days

24.1.4 01-IO

02-ReadStructured

ReadExcel

Dataset Node for reading Excel files

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetExcel

Name	Title	Description
path	Path	Path of the Excel file
sheetName	Sheetname	Excel Sheet Name
header	Header	Does the file have a header row
outputColNames	Column Names for	New Output Columns of the SQL
	the Excel	
outputColTypes	Column Types for	Data Type of the Output Columns
	the Excel	
outputColFormats	Column Formats for	Format of the Output Columns
	the Excel	

EmptyDataset

It creates an empty DataFrame

Input

It does not read any input

Output

It creates an empty DataFrame

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetEmpty

Fields

ReadCSV

It reads in CSV files and creates a DataFrame from it

Input

It reads in CSV text files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetCSV

Fields

Name	Title	Description
path	Path	Path of the Text file/directory
separator	Separator	CSV Separator
header	Header	Does the file have a header row
dropMalformed	Drop Malformed	Whether to drop Malformed records or error
outputColNames	Column Names for	New Output Columns of the SQL
	the CSV	
outputColTypes	Column Types for	Data Type of the Output Columns
	the CSV	
outputColFormats	Column Formats for	Format of the Output Columns
	the CSV	

ReadAvro

Dataset Node for reading Apache Avro files

Input

It reads in Avro files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetAvro

Name	Title	Description
path	Path	Path of the Avro file/directory
outputColNames	Column Names for	Output Columns of the Avro
	the Avro	
outputColTypes	Column Types for	Data Type of the Output Columns
	the Avro	
outputColFormats	Column Formats for	Format of the Output Columns
	the Avro	

ReadXML

It reads in XML files and creates a DataFrame from it

Input

It reads in XML text files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetXML

Fields

Name	Title	Description
path	Path	Path of the Text file/directory
rowTag	Row Tag	Row Tag
outputColNames	Column Names for	New Output Columns of the SQL
	the CSV	
outputColTypes	Column Types for	Data Type of the Output Columns
	the CSV	
outputColFormats	Column Formats for	Format of the Output Columns
	the CSV	

QueryJDBCConnection

This node executes query in Relational Databases using JDBC and creates a DataFrame from it

Input

It reads data from Relational Databases

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeJDBCQueryUsingConnection

Fields

Name	Title	Description
connection	Connection	The JDBC connection to connect
query	Query	
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

JDBCIncrementalLoad

This node is used to load incremental data from RDBMS to Hive.

Input

RDBMS detail like url, username , password, hivedb , hive table name

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetJDBCIncrementalLoad

Fields

Name	Title	Description
sqldb	SqlDB	
sqlServer	SqlServer	
sqlUser	SqlUser	
password	password	
sqltable	SqlTable	
sqlkeycolumn	SqlKeyColumn	
homeDirectory	Config Path	
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

DB2 JDBC

This node reads data from other databases using JDBC.

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetJDBC

Name	Title	Description
url	DB2 JDBC URL	The JDBC URL to connect to
user	User	User for connecting to the DB
password	Password	Password for connecting to the DB
dbtable	DB2 Table	The JDBC table that should be read. Note
		that anything that is valid in a FROM clause
		of a SQL query can be used. For example,
		instead of a full table you could also use a
		subquery in parentheses.
driver	DB2 Driver	The class name of the JDBC driver needed
		to connect to this URL
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

ReadParquet

Dataset Node for reading Apache Parquet files

Input

It reads in Parquet files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetParquet

Name	Title	Description
path	Path	Path of the Parquet file/directory
outputColNames	Column Names for	Output Columns of the Parquet
	the Parquet	
outputColTypes	Column Types for	Data Type of the Output Columns
	the Parquet	
outputColFormats	Column Formats for	Format of the Output Columns
	the Parquet	

ReadDatabricksTable

This node reads data from Relational Databases using JDBC and creates a DataFrame from it

Input

It reads data from Relational Databases

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeReadDatabricksTable

Fields

Name	Title	Description
db	Databricks	Databricks Database
	Database	
table	Databricks Table	Databricks Table from which to read the
		data
driver	Driver	The class name of the JDBC driver needed
		to connect to this URL
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

JDBCConnection

This node reads data from Relational Databases using JDBC and creates a DataFrame from it

Input

It reads data from Relational Databases

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetJDBCUsingConnection

Fields

Name	Title	Description
connection	Connection	The JDBC connection to connect
dbtable	DB Table	The JDBC table that should be read. Note
		that anything that is valid in a FROM clause
		of a SQL query can be used. For example,
		instead of a full table you could also use a
		subquery in parentheses.
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

CreateDataset

Creates a dataset with the specified number of Rows and 9 pre-defined columns

Input

It does not read data from any external source

Output

It creates a DataFrame with the specified number of Rows

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetCreate

Fields

Name	Title	Description
numRows	Number of Rows	Number of Rows in the Output Dataset

DatasetStructured

This Node creates a DataFrame by reading data from HDFS, HIVE etc. The dataset has been defined earlier in Fire by using the Dataset Feature. As a user, you just have to select the Dataset of your interest.

Input

It reads in data from HIVE or files HDFS

Output

It creates a DataFrame from the input data and sends it to its output.

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetStructured

Fields

Name	Title	Description
dataset	Dataset	Selected Dataset

Details

This Node creates a DataFrame by reading data from HDFS, HIVE etc.

The data has been defined earlier in Fire by using the Dataset Feature. As a user, you just have to select the Dataset of your interest.

ReadJDBC

This node reads data from Relational Databases using JDBC and creates a DataFrame from it

Input

It reads data from Relational Databases

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetJDBC

Fields

Name	Title	Description
url	URL	The JDBC URL to connect to
user	User	User for connecting to the DB
password	Password	Password for connecting to the DB
dbtable	DB Table	The JDBC table that should be read. Note
		that anything that is valid in a FROM clause
		of a SQL query can be used. For example,
		instead of a full table you could also use a
		subquery in parentheses.
driver	Driver	The class name of the JDBC driver needed
		to connect to this URL
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

ReadHanaCsv

It reads in Hana CSV files and creates a DataFrame from it

Input

It reads in CSV text files and sql file to create schema from it

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeReadHANACSVDump

Fields

Name	Title	Description
path	Path	Path of the Text file/directory
hdfsSqlFile	SQL File	Path of the sql file that contains create table
		script.
outputColNames	Column Names for	New Output Columns of the SQL
	the CSV	
outputColTypes	Column Types for	Data Type of the Output Columns
	the CSV	
outputColFormats	Column Formats for	Format of the Output Columns
	the CSV	

URLSingleRecordJSONReader

It reads in single record JSON from the given URL and creates a DataFrame from it

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetURLSingleRecordJsonReader

Name	Title	Description
URL	URL	URL from where to read the JSON string
		from
outputColNames	Column Names	Column Names
outputColTypes	Column Types	Data Types
outputColFormats	Column Formats	Formats

ReadLibsvm

It reads in Libsvm files and creates a DataFrame from it

Input

It reads in Libsvm text files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetLibsvm

Fields

Name	Title	Description
path	Path	Path of the Text file/directory
numFeatures	NumFeatures	Number of features in feature column
outputColNames	Column Names for	New Output Columns of the SQL
	the CSV	
outputColTypes	Column Types for	Data Type of the Output Columns
	the CSV	
outputColFormats	Column Formats for	Format of the Output Columns
	the CSV	

ReadJSON

Dataset Node for reading JSON files

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetJSON

Fields

Name	Title	Description
path	Path	Path of the JSON file/directory
multiLine	Multi Line	
outputColNames	Column Name	New Output Column Name
outputColTypes	Column Type	Data Type of the Output Column
outputColFormats	Column Format	Format of the Output Column

Details

It reads in JSON files. Each JSON record has to be on a separate line for Spark to handle it correctly.

There cannot be line break within a record.

URLTextFileReader

Reads text file from the given URL and creates a DataFrame from it. Each line in the file is a record in the DataFrame.

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetUrlTextFileReader

Fields

Name	Title	Description
url	URL	URL of the file

ReadShapeFile

It reads in Shape files and creates a DataFrame from it

Input

It reads in Shape files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetShapeFile

Fields

Name	Title	Description
path	Path	Path of the input directory

03-ReadUnstructured

TextFiles

Reads in Text Files from a given path and loads each line as a separate Row

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetTextFiles

Fields

Name	Title		Desc	ription				
path	Path		Path o	of the Te	xt file/direc	ctory		
outputCol	Output	Column	Text	Lines	Column	in	the	Output
	Name		DataF	Frame				

WholeTextFiles

Reads in Whole Text Files directory from a given path and loads each files as a separate Row with key(file name and values(file content)

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetWholeTextFiles

Fields

Name	Title	Description
path	Path	Path of the Text files directory

Tika

Reads in files from a given path and parses them with Apache Tika

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetTika

Fields

Name	Title	Description
path	Path	Path of the file/directory
fileNameCol	File Name Column	File Name Column in the Output DataFrame
contentCol	Content Column	Tika output Column in the Output
		DataFrame

PDF

Reads in PDF Files from a given path and extracts the text content from them

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetPDF

Fields

Name	Title	Description
path	Path	Path of the PDF file/directory
fileNameCol	File Name	File Name Column in the Output DataFrame
contentCol	File Content	File Content Column in the Output
		DataFrame

PDFImageOCR

Reads in PDF Files from a given path, extracts the images from them and converts them to text with Tesseract

Input

It reads in a PDF file or a directory containing PDF files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetPDFImageOCR

Fields

Name	Title	Description
path	Path of the PDF files	Path of the PDF file/directory
fileNameCol	File Name Column	File Name Column in the Output DataFrame
outputCol	Column Name	OCR output Column in the Output
	which contains the	DataFrame
	result of OCR	

BinaryFiles

Reads in Binary Files from a given path and loads them as FileName/Content

Туре

dataset

Class

fire.nodes.dataset.NodeDatasetBinaryFiles

Fields

Name	Title	Description
path	Path	Path of the Binary file/directory
fileNameCol	File Name Column	File Name Column in the Output DataFrame
binaryContentCol	Binary File Content	Binary File Content Column in the Output
	Column	DataFrame

Details

It creates a new Dataframe from some data. Data can be in binary, text, parquet, pdf, image files.

03-Save

SaveJDBC

This node writes data to databases using JDBC.

Туре

transform

Class

fire.nodes.save.NodeSaveJDBC

Name	Title	Description
url	URL	The JDBC URL to connect to
table	DB Table	The JDBC table to write to
driver	Driver	The class name of the JDBC driver needed
		to connect to the URL
user	User	Username with which to connect to the DB
password	Password	Password with which to connect to the DB
truncate	Truncate	Whether to truncate the table in case Save
		Mode is Overwrite
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the table Exists

UpsertJDBC

This node insert or update the data to databases using JDBC.

Туре

transform

Class

fire.nodes.save.NodeUpsertJDBC

Fields

Name	Title	Description
primaryKeyColumn	PrimaryKeyColumn	Key column name in table
url	URL	The JDBC URL to connect to
table	DB Table	The JDBC table to write to
driver	Driver	The class name of the JDBC driver needed
		to connect to the URL
user	User	Username with which to connect to the DB
password	Password	Password with which to connect to the DB

SaveCSV

Saves the DataFrame into the specified location in CSV Format

Туре

transform

Class

fire.nodes.save.NodeSaveCSV

Fields

Name	Title		Description
path	Path		Path where to save the CSV files
saveMode	Save Mode		Whether to Append, Overwrite or Error if
			the path Exists
header	Header		Should a Header Row be saved with each
			File?
partitionColNames	Partition Co	Column	Partition Column Names
	Names		

SaveJSON

Saves the DataFrame into the specified location in JSON Format

Туре

transform

Class

fire.nodes.save.NodeSaveJSON

Fields

Name	Title	Description
path	Path	Path where to save the JSON files
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists
partitionColNames	Partition Column	Partition Column Names
	Names	

KafkaProducer

Write out the Dataframe to a specified Apache Kafka Topic

Туре

transform

Class

fire.nodes.save.NodeKafkaProducer

Fields

Name	Title	Description
brokers	Kafka Brokers	Brokers
topic	Торіс	Kafka Topic to write out the incoming Dataframe to

SaveParquet

Saves the DataFrame into the specified location in Parquet Format. When running on Hadoop, it is saved onto HDFS.

Туре

transform

Class

fire.nodes.save.NodeSaveParquet

Fields

Name	Title	Description
path	Path	Path where to save the Parquet files
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists
partitionColNames	Partition Column	Partition Column Names
	Names	

SaveORC

Saves the DataFrame into the specified location in ORC Format

Туре

transform

Class

fire.nodes.save.NodeSaveORC

Name	Title	Description
path	Path	Path where to save the ORC files
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists

InsertIntoHIVETable

Saves the DataFrame into an Apache HIVE Table

Туре

transform

Class

fire.nodes.save.NodeInsertIntoTable

Fields

Name	Title	Description
database	HIVE Database	Name of the HIVE Database
table	HIVE Table	Name of the HIVE table
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists
partitionBy	Partition By	Partition By Column (can be empty)
bucketBy	Bucket By	Bucket By Column (can be empty)

Details

When using Insert Into Table, the HIVE table has to already exist.

Otherwise it throws the following exception:

org.apache.spark.sql.catalyst.analysis.NoSuchTableException: Table or view 'xyz' not found in database 'abc';

SaveAsHIVETable

Saves the DataFrame into an Apache HIVE Table

Туре

transform

Class

fire.nodes.save.NodeSaveAsTable

Fields

Name	Title	Description
database	HIVE Database	Name of the HIVE Database
table	HIVE Table	Name of the HIVE table
partitionBy	Partition By	List of columns to partition by - separated
		by space
format	Format	File format when saving to HIVE Table
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists

Details

If the HIVE table does not exist, it would create the table.

SaveAvro

Saves the DataFrame into the specified location in Apache Avro Format

Туре

transform

Class

fire.nodes.save.NodeSaveAvro

Fields

Name	Title	Description
path	Path	Path where to save the Avro files
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists

01-Connectors

Salesforce

This node reads data from Salesforce.

Туре

dataset

Class

fire.nodes.sales force.NodeReadSales force

Fields

Name	Title	Description
sql	SQL	Sql for reading salesforce data ex - select id,
		name, amount from opportunity
userNmae	User Name	UserName of Salesforce
password	Password	Password of Salesforce
readOption	Read Option	Pulling data/Object from salesforce
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

ReadMarketo

Node for reading Marketo files

Туре

dataset

Class

fire.nodes.marketo.NodeReadMarketo

Name	Title	Description
clientId	Client Id	Marketo account clientId
clientSecret	Client Secret	Marketo account clientSecret
instanceUrl	Instance Url	Instance URL to be used to access Mar- keto. It has to be specified without /rest. i.e it should be like https://119-AAA-888. mktorest.com
object	Object	Object to be queried from Marketo. ex. leads
filterType	Filter Type	Filter field to be used
filterValues	Filter Values	Comma separated filter values to be applied
fromDate	From Date	(Optional) Datatime from which the data has to be fetched. It has to be in ISO 8601 format
customObject	Custom Object	(Optional) Boolean to specify if the speci- fied object is custom object, Default value is false
apiVersion	Api Version	(Optional) API Version to be used. Default value is v1
modifiedFields	Modified Fields	(Optional) Fields to be considered for lead- Changes. It has to be comma separated field names
queryType	Query Type	Query Type of Marketo
outputColNames	Column Names for the Marketo	New Output Columns of the SQL
outputColTypes	Column Types for the Marketo	Data Type of the Output Columns
outputColFormats	Column Formats for the Marketo	Format of the Output Columns

SaveRedshift-AWS

This node save data to Redshift using JDBC.

Туре

transform

Class

fire.nodes.aws.NodeSaveRedshift

Name	Title	Description
url	URL	The JDBC URL to connect to
dbtable	Redshift Table	The Redshift table that should be write.
		Note that anything that is valid in a FROM
		clause of a SQL query can be used. For ex-
		ample, instead of a full table you could also
		use a subquery in parentheses.
awsAccessKeyId	AWS Access Key Id	AWS Access Key Id
awsSecretAccessKey	AWS Secret Access	AWS Secret Access Key
	Key	
tempS3Dir	Temporary S3 direc-	Temporary S3 directory
	tory	
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists

WriteToSnowFlake

Туре

transform

Class

fire.nodes.snowflake.NodeWriteToSnowFlake

Fields

Name	Title	Description
sfUrl	SF Url	SnowFlake URL to connect to
sfUser	SF User	User for connecting to the SnowFlake
sfPassword	SF Password	Password for connecting to the SnowFlake
sfDatabase	SF Database	Database for connecting to the SnowFlake
sfSchema	SF Schema	Schema for connecting to the SnowFlake
sfWarehouse	SF Warehouse	Warehouse for connecting to the SnowFlake
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the table Exists
dbtable	SF Table	

SaveCassandra

Saves the rows of the incoming DataFrame into Apache Cassandra

Туре

transform

Class

fire.nodes.cass and ra.Node Save Cass and ra

Fields

Name	Title	Description
table	Cassandra Table	Cassandra Table into which data gets loaded
	Name	
keyspace	Cassandra Keyspace	The keyspace where table is looked for
	Name	
host	Host	
username	Username	
password	Password	

ExecuteQueryInSnowFlake

Туре

dataset

Class

fire.nodes.snowflake.Node Execute Query In SnowFlake

Fields

Name	Title	Description
sfUrl	SF Url	SnowFlake URL to connect to
sfUser	SF User	User for connecting to the SnowFlake
sfPassword	SF Password	Password for connecting to the SnowFlake
sfDatabase	SF Database	Database for connecting to the SnowFlake
sfSchema	SF Schema	Schema for connecting to the SnowFlake
sfWarehouse	SF Warehouse	Warehouse for connecting to the SnowFlake
query	SF Query	
outputColNames	Output Column	Name of the Output Columns
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	

ReadMongoDB

Reads data from MongoDB

Туре

dataset

Class

fire.nodes.mongodb.NodeReadMongoDB

Fields

Name	Title	Description
mongoURI	MongoDB URI	URI of MongoDB to read from
mongoDBName	MongoDB Database	Name of the MongoDB database to read
		from
mongoTableName	MongoDB Table	Name of the MongoDB table to read from
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

SaveMongoDB

It Saves the incoming Dataframe into MongoDB

Input

It takes in a DataFrame as input

Output

Incoming dataFrame is passed along to the next nodes.

Туре

transform

Class

fire.nodes.mongodb.NodeSaveMongoDB

Name	Title	Description
mongoURI	mongo URI	URI of mongodb
mongoDBName	mongoDB Name	mongoDB Name
mongoTableName	mongo Table Name	mongo Table Name

ReadDatabricksTable

This node reads a table from Databricks

Input

It reads data from Databricks Table

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.databricks.NodeReadDatabricksTable

Fields

ReadHIVETable

This node reads data from Apache HIVE table and creates a DataFrame from it

Input

It reads in CSV text files

Output

It creates a DataFrame from the data read and sends it to its output

Туре

dataset

Class

fire.nodes.hive.NodeHIVE

Fields

SaveHBase

Saves all the rows in the incoming DataFrame onto Apache HBase using the specific field mapping

Input

It takes in a DataFrame as input

Output

Incoming dataFrame is passed along to the next nodes.

Туре

transform

Class

fire.nodes.hbase.NodeSaveHBase

Fields

Details

SaveHBase node saves all the rows in the incoming DataFrame onto HBase using the specific field mapping. The DataFrame columns which do not have to be loaded into HBase are left empty.

SaveElasticSearch

Stores the rows of the incoming DataFrame into Elastic Search

Туре

transform

Class

fire.nodes.elastics earch.NodeSaveElasticSearch

Name	Title	Description
indexName	Index Name	Name of the Elastic Search Index
elasticSearchHost	Elastic Search Host	Name of the Elastic Search Host
elasticSearchPort	Elastic Search Port	Port of Elastic Search
esIndexAutoCreate	es.index.auto.create	ES Index Auto Create
esNodesWANOnly	es.nodes.wan.only	ES Nodes WAN Only
esNodesIngestOnly	es.nodes.ingest.only	ES Nodes Ingest Only
esNodesDataOnly	es.nodes.data.only	ES Nodes Data Only
esNetHttpAuthUser	es.net.http.auth.user	Username
esNetHttpAuthPass	es.net.http.auth.pass	Password
esConfKeys	Config Key/Value	More Config Values
	Pairs	
esConfValues	Config Key/Value	More Config Values
	Pairs	

ReadFromSnowFlake

Туре

dataset

Class

fire.nodes.snowflake.NodeReadFromSnowFlake

Fields

Name	Title	Description
sfUrl	SF Url	SnowFlake URL to connect to
sfUser	SF User	User for connecting to the SnowFlake
sfPassword	SF Password	Password for connecting to the SnowFlake
sfDatabase	SF Database	Database for connecting to the SnowFlake
sfSchema	SF Schema	Schema for connecting to the SnowFlake
sfWarehouse	SF Warehouse	Warehouse for connecting to the SnowFlake
dbtable	SF Table	
outputColNames	Output Column	Name of the Output Columns
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	

SFTP

Secure file transfer protocol

Туре

dataset

Class

fire.nodes.sftp.NodeSftp

Fields

Name	Title	Description
sftpHost	Sftp Host	IP address of sftp
sftPort	Sft Port	Port no of SFTP. Default port is 22
sftpUser	Sftp User Name	SFTP User Name
sftpPass	Sftp Password	SFTP User Password
sftpUserDir	Sftp User Directory	user directory path(File take from)
sftpDirectory	Sftp Directory	server directory path(Inside SFTP uploads
		folder '/uploads')
pemKey	Pem Key	Path of pem key directory

ReadCassandra

This node reads data from Apache Cassandra

Туре

dataset

Class

fire.nodes.cass and ra.Node Read Cass and ra

Fields

Name	Title	Description
table	Cassandra Table	Cassandra Table from which to read the data
keyspace	Cassandra Keyspace	Cassandra Keyspace
host	Cassandra host	
username	Username	
password	Password	

SaveDatabricksTable

This node saves a input data as table in Databricks

Input

It take dataframe as input data.

Output

It creates a Table in Databricks from the dataframe(input data).

Туре

transform

Class

fire.nodes.databricks.NodeSaveDatabricksTable

Fields

Name	Title	Description
database	Databricks	Name of the Database
	Database	
table	Databricks Table	Name of the table
partitionBy	Partition By	List of columns to partition by - separated
		by space
format	Format	File format when saving to Table
saveMode	Save Mode	Whether to Append, Overwrite or Error if
		the path Exists

ReadRedshift-AWS

This node reads data from Redshift using JDBC.

Туре

dataset

Class

fire.nodes.aws.NodeReadRedshift

Name	Title	Description
url	URL	The JDBC URL to connect to
dbtable	Redshift Table	The Redshift table that should be read. Note
		that anything that is valid in a FROM clause
		of a SQL query can be used. For example,
		instead of a full table you could also use a
		subquery in parentheses.
awsAccessKeyId	AWS Access Key Id	AWS Access Key Id
awsSecretAccessKey	AWS Secret Access	AWS Secret Access Key
	Кеу	
tempS3Dir	Temporary S3 direc-	Temporary S3 directory
	tory	
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

ReadElasticSearch

Reads data from Elastic Search

Туре

dataset

Class

fire.nodes.elastics earch.NodeReadElasticSearch

Name	Title	Description
indexName	Index Name	Name of the Elastic Search Index
elasticSearchHost	Elastic Search Host	Name of the Elastic Search Host
elasticSearchPort	Elastic Search Port	Port of Elastic Search
temporaryTable	Spark Temporary	Spark Temporary Table to be used for read-
	Table for Reading	ing from Elastic Search
	from ES	
sql	SQL for reading	SQL for reading from Elastic Search.
	from Elastic Search	Where condition can be applied here for
		limiting the rows read from ES.
outputColNames	Column Names of	Output Columns Names of the Table
	the Table	
outputColTypes	Column Types of	Output Column Types of the Table
	the Table	
outputColFormats	Column Formats	Output Column Formats

24.1.5 11-ML-SparkML

12-FreqPatternMining

FPGrowth

Does Pattern Mining using FPGrowth Algorithm

Туре

transform

Class

fire.nodes.ml.NodeFPGrowth

Fields

Name	Title	Description
transactionCol	Transaction Column	Input data set, each element contains a trans-
		action
minSupport	Min Support	The minimum support for an itemset to be
		identified as frequent
numPartitions	Number of Parti-	The number of partitions used to distribute
	tions	the work

Details

This node does Pattern Mining using FPGrowth Algorithm.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-frequent-pattern-mining.html

04-FeatureTransformers

VectorAssembler

Merges multiple columns into a vector column

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

It adds a new column to the incoming DataFrame. The new column contains the values of the input columns concatenated into a vector in the specified order.

Туре

ml-transformer

Class

fire.nodes.ml.NodeVectorAssembler

Fields

Name	Title	Description
inputCols	Input Columns	Input column of type - all numeric, boolean
		and vector
outputCol	Output Column	Output column name

IDF

Compute the Inverse Document Frequency (IDF) given a collection of documents.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

The output DataFrame contains a new column of type vector, It takes feature vectors (generally created from HashingTF) as input and scales each column. Intuitively, it down-weights columns which appear frequently in a corpus.

Туре

ml-transformer

Class

fire.nodes.ml.NodeIDF

Fields

Name	Title	Description
inputCol	Input Column	Input Column Name
outputCol	Output Column	Output column name
minDocFreq	MinDocFreq	The minimum of documents in which a term
		should appear.

StopWordsRemover

Filters out stop words from input. Null values from input array are preserved unless adding null to stopWords explicitly.

Output

It adds a new column containing the sequence of strings from the input column but with the stop words removed, to the incoming DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeStopWordsRemover

Name	Title	Description
inputCol	Input Column	Column containing the array text from
		which the stop words have to be removed
outputCol	Output Column	Contains array of text by dropping list of
		stop words
caseSensitive	Case Sensitive	Case Sensitive
stopWords	Comma Separated	Custom List of Stop Words
	List of Custom Stop	
	Words. If not pro-	
	vided, the default	
	list of stop words	
	would be used.	

Details

Stop words filters out stop words from input. Null values from input array are preserved unless adding null to stop-Words explicitly.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#stopwordsremover

Tokenizer

A tokenizer that converts the input string to lowercase and then splits it by white spaces.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

It adds a new column containing the results of tokenization of the input column, to the incoming DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeTokenizer

Name	Title	Description
inputCol	Input Column	Column containing text (such as sentence)
outputCol	Output Column	Output column name

PolynominalExpansion

Perform feature expansion in a polynomial space

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

The output DataFrame contains a new column of type vector, Expanding your features into a polynomial space, which is formulated by an n-degree combination of original dimensions.

Туре

ml-transformer

Class

fire.nodes.ml.NodePolynominalExpansion

Fields

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
degree	Degree	The polynomial degree to expand, which
		should be $>= 1$. A value of 1 means no ex-
		pansion.

VectorIndexer

Vector Indexer indexes categorical features inside of a Vector. It decides which features are categorical and converts them to category indices. The decision is based on the number of distinct values of a feature.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

It indexes categorical features in datasets of Vectors and stores the result into a new column of the DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeVectorIndexer

Fields

Name	Title	Description
inputCol	Input Column	The Input column name
outputCol	Output Column	Output column name
maxCategories	Maximum Cate-	Threshold for the number of values a cate-
	gories	gorical feature can take. If a feature is found
		to have > maxCategories values, then it is
		declared continuous. Must be ≥ 2

Normalizer

Normalizer is a Transformer which transforms a dataset of Vector rows, normalizing each Vector to have unit norm.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

It adds a new column containing the normalized value of the input column, to the incoming DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeNormalizer

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
p	Р	Normalization in L^p space. Must be ≥ 1 .
		(default: $p = 2$)

Details

Normalizer is a Transformer which transforms a dataset of Vector rows, normalizing each Vector to have unit norm. More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#normalizer

OneHotEncoder

Maps a column of label indices to a column of binary vectors, with at most a single one-value

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

The output DataFrame contains a new column which contains the mapping of a column of label indices to a column of binary vectors, with at most a single one-value.

Туре

ml-transformer

Class

fire.nodes.ml.NodeOneHotEncoder

Fields

Name	Title	Description
inputCols	Input Columns	Input columns for encoding
outputCols	Output Columns	Output columns

NGramTransformer

Converts the input array of strings into an array of n-grams. Null values in the input array are ignored. It returns an array of n-grams where each n-gram is represented by a space-separated string of words. When the input is empty, an empty array is returned. When the input array length is less than n (number of elements per n-gram), no n-grams are returned

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

It adds a new column consisting of a sequence of nn-grams where each nn-gram is represented by a space-delimited string of nn consecutive words, to the incoming DataFrame

Туре

ml-transformer

Class

fire.nodes.ml.NodeNGramTransformer

Fields

Name	Title	Description
inputCol	Input Column	Contains sequence of strings
inputColStringArrCol	List of Words	Sequence of words
outputCol	Output Column	Consist of a sequence of n-grams where
		each n-gram is represented by a space-
		delimited string of n consecutive words
numberOfGrams	Number of Grams	Sequence of 'string array' for integer 'Num-
		ber of Grams'

Details

This node converts the input array of strings into an array of n-grams. Null values in the input array are ignored. It returns an array of n-grams where each n-gram is represented by a space-separated string of words. When the input is empty, an empty array is returned. When the input array length is less than n (number of elements per n-gram), no n-grams are returned"

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#n-gram

Binarizer

Binarize a column of continuous features given a threshold.

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

A new column containing the binarized values is added to the incoming DataFrame

Туре

ml-transformer

Class

fire.nodes.ml.NodeBinarizer

Fields

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
threshold	Threshold	The features greater than the threshold, will
		be binarized to 1.0. The features equal to or
		less than the threshold, will be binarized to
		0.0.

Details

This node binarizes a column of continuous features given a threshold.

More at Spark MLlib/ML docs page : https://spark.apache.org/docs/latest/ml-features.html#binarizer

VectorFunctions

Vector Functions for transforming Vectors

Туре

ml-transformer

Class

fire.nodes.ml. Node Vector Functions

Fields

Name	Title	Description
inputCol	Input Column	The Input column name
vectorFunction	Vector Function	Vector Function Name
parameter	Parameter	Parameter for the Function
outputCol	Output Column	Output column name

WordToScoreMapping

Map the original word of hashValue to score.

Туре

ml-transformer

Class

fire.nodes.ml.NodeWordToScoreMapping

Fields

Name	Title	Description
words	Words	Array of words
features	Features	Vector with hash value of words
output	Output	

IndexString

Maps a column of indices back to a new column of corresponding string values. The index-string mapping is either from the ML attributes of the input column, or from user-supplied labels

Туре

ml-transformer

Class

fire.nodes.ml.NodeIndexString

Name	Title	Description
inputCol	Input Column	Column containing label indices
outputCol	Output Column	Output column name

Details

This node maps a column of indices back to a new column of corresponding string values. The index-string mapping is either from the ML attributes of the input column, or from user-supplied labels

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#indextostring

QuantileDiscretizer

QuantileDiscretizer takes a column with continuous features and outputs a column with binned categorical features.

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

The output DataFrame contains a new column of binned categorical features.

Туре

ml-transformer

Class

fire.nodes.ml.NodeQuantileDiscretizer

Fields

Name	Title	Description
inputCol	Input Column	The Input column name
outputCol	Output Column	Output column name
numBuckets	NumBuckets	Maximum number of buckets (quantiles or
		categories) into which the data points are
		grouped. Must be ≥ 2 .

Details

QuantileDiscretizer takes a column with continuous features and outputs a column with binned categorical features. More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#quantilediscretizer

SQLTransformer

This node runs the given SQL on the incoming DataFrame using Spark ML SQLTransformer

Туре

transform

Class

fire.nodes.ml.NodeSQLTransformer

Fields

Name	Title	Description
tempTable	Temp Table	Temp Table Name to be used
sql	SQL	SQL to be run
outputColNames	Output Column	Name of the Output Columns
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	

StringIndexer

StringIndexer encodes a string column of labels to a column of label indices

Input

It takes in a DataFrame and transforms it to another DataFrame

Output

It adds a new column containing the encoding of the string column of labels to a column of label indices, to the incoming DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeStringIndexer

Fields

Name	Title	Description
handleInvalid	Handle Invalid	Invalid entries to be skipped or thrown error
inputCols	Input Columns	Input columns for encoding
outputCols	Output Columns	Output columns

03-FeatureExtraction

RFormula

RFormula feature selection, RFormula selects columns specified by an R model formula. Currently we support a limited subset of the R operators, including '~', '.', '+', and '-'

Туре

ml-transformer

Class

fire.nodes.ml.NodeRFormula

Fields

Name	Title	Description
featuresCol	Features Column	The features column name
formula	Formula	formula
labelCol	Label Column	The label column name

HashingTF

Maps a sequence of terms to term frequencies using the hashing trick.

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

A new column is added to the input DataFrame containing hashing of the bag of words into a feature vector

Туре

ml-transformer

Class

fire.nodes.ml.NodeHashingTF

Fields

Name	Title	Description
inputCol	Input Column	Contains sets of terms. In text processing, a
		'set of terms' might be a bag of words
outputCol	Output Column	Output column name

CountVectorizer

Extracts the vocabulary from a given collection of documents and generates a vector of token counts for each document.

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

It adds a new column to the incoming DataFrame containing the vector of token counts in the input column, to generate the output DataFrame

Туре

ml-transformer

Class

fire.nodes.ml.NodeCountVectorizer

Name	Title	Description
inputCol	Input Column	Input column name
outputCol	Output Column	Output column name
vocabularySize	Vocabulary Size	Max size of the vocabulary.

Details

CountVectorizer and CountVectorizerModel aim to help convert a collection of text documents to vectors of token counts. When an a-priori dictionary is not available, CountVectorizer can be used as an Estimator to extract the vocabulary and generates a CountVectorizerModel. The model produces sparse representations for the documents over the vocabulary, which can then be passed to other algorithms like LDA.

More at Spark MLlib/ML docs page : https://spark.apache.org/docs/latest/ml-features.html#countvectorizer

Word2Vec

Transforms vectors of words into vectors of numeric codes for the purpose of further processing by NLP or machine learning algorithms.

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

A new column containing feature vector is added to the incoming DataFrame

Туре

ml-transformer

Class

fire.nodes.ml.NodeWord2Vec

Fields

Name	Title	Description
inputCol	Input Column	Contains sequences of words
inputColStringArrCol	Text Array Column	The text array column which is produced
outputCol	Output Column	Output column name
vectorSize	Vector Size	Vector Size
minCount	Min Count	Min Count

Details

Word2Vec is an Estimator which takes sequences of words representing documents and trains a Word2VecModel. The model maps each word to a unique fixed-size vector. The Word2VecModel transforms each document into a vector using the average of all words in the document; this vector can then be used for as features for prediction, document similarity calculations, etc.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#word2vec

11-CollaborativeFiltering

ALS

Alternating Least Squares (ALS) matrix factorization.

Input

It takes in a DataFrame as input and performs ALS

Output

It generates the ALSModel and passes it to the next Predict and ModelSave Nodes. It also passes the incoming DataFrame to the next Nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeALS

Name	Title	Description
userCol	User Column	The column name for user ids.
itemCol	Item Column	The column name for item ids.
ratingCol	Rating Column	The column name for ratings.
predictionCol	Prediction Column	The prediction column created during model scoring
maxIter	Max iterations	The maximum number of iterations.
regParam	Regularization	The regularization parameter.(>=0)
	Param	
alpha	Alpha	The alpha parameter in the implicit prefer-
		ence formulation.(>=0)
checkpointInterval	Checkpoint Interval	The checkpoint interval.
nonnegative	Non negative	Whether to apply nonnegativity constraints.
numItemBlocks	Num Item Blocks	The number of item blocks.
numUserBlocks	Num User Blocks	The number of user blocks.
rank	Rank	The rank of the matrix factorization.
seed	Seed	Random Seed.
implicitPrefs	Implicit Prefs	whether to use implicit preference

Details

Collaborative filtering is commonly used for recommender systems. These techniques aim to fill in the missing entries of a user-item association matrix. spark.mllib currently supports model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries. spark.mllib uses the alternating least squares (ALS) algorithm to learn these latent factors.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html

09-Regression

GBTRegression

It supports both continuous and categorical features.

Input

This takes in a DataFrame and performs Logistic Regression

Output

It generates the GBTRegression and passes it to the next Predict and ModelSave Nodes. The input DataFrame is also passed along to the next nodes.

Туре

ml-estimator

Class

fire.nodes.ml. NodeGBTR egression

Fields

Details

GBT Regression supports both continuous and categorical features.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-classification-regression.html# gradient-boosted-trees-gbts

AFTSurvivalRegression

Accelerated failure time (AFT) model which is a parametric survival regression model for censored data.

Output

It generates the LAFTSurvivalRegressionModel and passes it to the next Predict and ModelSave Nodes. The input DataFrame is also passed along to the next nodes.

Туре

ml-estimator

Class

fire.nodes.ml. Node AFTS urvival Regression

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting
labelCol	Label Column	The label column for model fitting
censorCol	Censor Column	Indicator of the event has occurred or not. If
		the value is 1.O, it means the event has oc-
		curred i.e. uncensored; otherwise censored
fitIntercept	Fit Intercept	Whether to fit an intercept term
maxIter	Maximum Iterations	Maximum number of iterations (≥ 0)
tol	Tolerance	The convergence tolerance for iterative al-
		gorithms
quantileProbabilities	QuantileProbabilities	Values of the quantile probabilities array
		should be in the range $(0, 1)$
quantilesCol	Quantiles Column	The quantiles column created during model
		scoring
predictionCol	Prediction Column	The prediction column created during
		model scoring

Details

More at Spark MLlib/ML docs page : https://spark.apache.org/docs/latest/ml-classification-regression.html# survival-regression

XGBoostRegressor

Input

It takes in a DataFrame as input and performs XGBoost Regression

Output

The XGBoost Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeXGBoostRegressor

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting
labelCol	Label Column	The label column for model fitting
predictionCol	Prediction Column	The prediction column created during
		model scoring.
maxDepth	Max Depth	The Maximum depth of a tree
maxBins	Max Bins	The maximum number of bins used for dis-
		cretizing continuous features. Must be ≥ 2
		and >= number of categories in any categor-
		ical feature.
maxLeaves	Max Leaves	
numRound	Num Round	
numWorkers	Num Workers	
objective	Objective	
eta	Eta	
regLambda	Reg Lambda	
regAlpha	Reg Alpha	
subsample	Subsample	
sampleType	SampleType	
treeMethod	TreeMethod	
useExternalMemory	UseExternalMemory	
seed	Seed	
baseScore	Base Score	
minChildWeight	Min Child Weight	
colsampleBylevel	ColSampleByLevel	
colsampleBytree	ColSampleByTree	
minSplitLoss	MinSplitLoss	
maxDeltaStep	MaxDeltaStep	
sketchEps	SketchEps	
scalePosWeight	ScalePosWeight	
growPlicy	GrowPlicy	
normalizeType	NormalizeType	
skipDrop	SkipDrop	
rateDrop	RateDrop	

DecisionTreeRegression

It supports both continuous and categorical features.

Input

This takes in a DataFrame and performs Decision Tree Regression

Output

The Decision Tree Regression Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml. Node Decision Tree Regression

Fields

Details

Decision tree supports both continuous and categorical features.

More at Spark MLlib/ML docs page : https://spark.apache.org/docs/1.6.0/ml-classification-regression.html# decision-tree-regression

RandomForestRegression

It supports both continuous and categorical features.

Input

This takes in a DataFrame and performs Random Forest Regression

Output

It generates the Random Forest Regression Model and passes it to the next Predict and ModelSave Nodes. The input DataFrame is also passed along to the next nodes.

Туре

ml-estimator

Class

fire.nodes.ml.NodeR and omForest Regression

LinearRegression

The interface for working with linear regression models and model summaries is similar to the logistic regression case.

Input

This takes in a DataFrame and performs Logistic Regression

Output

It generates the LinearRegressionModel and passes it to the next Predict and ModelSave Nodes. The input DataFrame is also passed along to the next nodes.

Туре

ml-estimator

Class

fire.nodes.ml.NodeLinearRegression

Fields

Details

The interface for working with linear regression models and model summaries is similar to the logistic regression case.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-classification-regression.html# linear-regression

08-Clustering

LDA

LDA is given a collection of documents as input data, via the featuresCol parameter. Each document is specified as a Vector of length vocabSize, where each entry is the count for the corresponding term (word) in the document

Input

It takes in a DataFrame as input and performs LDA

Output

LDA Model is passed to the next Node for Prediction or Storing

Туре

ml-estimator

Class

fire.nodes.ml.NodeLDA

Fields

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting.
k	K	The number of topics to create.
maxIter	Max Iterations	The maximum number of iterations.
optimizer	Optimizer	Optimizer or inference algorithm used to es-
		timate the LDA model.
topicDistributionCol	TopicDistributionColu	nOutput column with estimates of the topic
		mixture distribution for each document
checkpointInterval	checkpointInterval	The checkpoint interval (>= 1) or disable
		checkpoint (-1). E.g. 10 means that the
		cache will get checkpointed every 10 iter-
		ations.
subsamplingRate	subsamplingRate	Fraction of the corpus to be sampled and
		used in each iteration of mini-batch gradient
		descent, in range (0, 1].
seed	Seed	Random Seed.
maxTermsPerTopic	MaxTermsPerTopic	Number of Terms in Topics

GaussianMixture

This class performs expectation maximization for multivariate Gaussian Mixture Models (GMMs). A GMM represents a composite distribution of independent Gaussian distributions with associated mixing weights specifying each's contribution to the composite.

Input

It takes in a DataFrame as input and performs GaussianMixture clustering

Output

The input DataFrame is passed along to the next Processors

Туре

ml-estimator

Class

fire.nodes.ml.NodeGaussianMixture

Fields

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting.
k	К	The number of clusters to create.
maxIter	Max Iterations	The maximum number of iterations.
predictionCol	Prediction Column	The prediction column created during
		model scoring.
seed	Seed	Random Seed.
tol	Tolerence	The convergence tolerance for iterative al-
		gorithms.

Details

GaussianMixture clustering will maximize the log-likelihood for a mixture of k Gaussians, iterating until the log-likelihood changes by less than convergenceTol, or until it has reached the max number of iterations. While this process is generally guaranteed to converge, it is not guaranteed to find a global optimum.

More at Spark MLlib/ML docs page : https://spark.apache.org/docs/2.2.0/mllib-clustering.html#gaussian-mixture

KMeans

K-means clustering with support for k-means|| initialization proposed by Bahmani et al

Input

It takes in a DataFrame as input and performs K-Means clustering

Output

The input DataFrame is passed along to the next Processors

Туре

ml-estimator

Class

fire.nodes.ml.NodeKMeans

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting.
k	K	The number of clusters to create.
maxIter	Max Iterations	The maximum number of iterations.
predictionCol	Prediction Column	The prediction column created during
		model scoring.
seed	Seed	Random Seed.
tol	Tolerence	The convergence tolerance for iterative al-
		gorithms.
initMode	initMode	The initialization algorithm mode.
initSteps	initSteps	The number of steps for the k-meansll ini-
		tialization mode. It will be ignored when
		other initialization modes are chosen.

Details

K-means clustering with support for k-means|| initialization proposed by Bahmani et al

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-clustering.html#k-means

05-DimensionalityReduction

SVD

Туре

transform

Class

fire.nodes.ml.NodeSVD

Fields

PCA

Trains a model to project vectors to a low-dimensional space using PCA.

Input

This takes in a DataFrame as input

Output

The output DataFrame is a projection of the vectors in the incoming DataFrame to a low-dimensional space using PCA

Туре

ml-transformer

Class

fire.nodes.ml.NodePCA

Fields

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
k	К	The number of principal components

Details

PCA trains a model to project vectors to a low-dimensional space using PCA.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#pca

02-FeatureScaler

MinMaxScaler

MinMaxScaler transforms a dataset of Vector rows, rescaling each feature to a specific range (often [0, 1])

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

A new column containing the scaled features is added to the incoming DataFrame

Туре

ml-transformer

Class

fire.nodes.ml.NodeMinMaxScaler

Fields

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
max	Max	The upper bound after transformation, shared by all features
min	Min	The lower bound after transformation, shared by all features

StandardScaler

StandardScaler transforms a dataset of Vector rows, normalizing each feature to have unit standard deviation and/or zero mean.

Input

It takes in a DataFrame as input and transforms it to another DataFrame

Output

It adds a new column containing the transform of the input Vector column to unit standard deviation and/or zero mean features to the incoming DataFrame.

Туре

ml-transformer

Class

fire.nodes.ml.NodeStandardScaler

Fields

Name	Title	Description
inputCol	Input Column	The input column name
outputCol	Output Column	The output column name
withMean	With Mean	Centers the data with mean before scaling.
withStd	With Standard Dev	Scales the data to unit standard deviation

Details

StandardScaler transforms a dataset of Vector rows, normalizing each feature to have unit standard deviation and/or zero mean.

StandardScaler is an Estimator which can be fit on a dataset to produce a StandardScalerModel; this amounts to computing summary statistics. The model can then transform a Vector column in a dataset to have unit standard deviation and/or zero mean features.

If the standard deviation of a feature is zero, it will return default 0.0 value in the Vector for that feature.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-features.html#standardscaler

17-Util

Spark ML Model Load

Туре

ml-modelload

Class

fire.nodes.ml.NodeModelLoad

Fields

TrainValidationSplit

This node represents Train Validation Split from Spark ML

Input

TrainValidationSplit takes an Estimator, a set of ParamMaps provided in the estimatorParamMaps parameter, and anEvaluator.

Output

The incoming DataFrame is passed to the output.

Туре

ml-trainvalidationsplit

Class

fire.nodes.ml.NodeTrainValidationSplit

Name	Title	Description
trainRatio	Train Ratio	Training Ratio

Details

This node represents Train Validation Split from Spark ML.

More at Spark MLlib/ML docs page : example-model-selection-via-train-validation-split

http://spark.apache.org/docs/latest/ml-guide.html#

Spark ML Model Save

This node saves the ML model generated at the specified path

Input

It takes in a Model and DataFrame as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-modelsave

Class

fire.nodes.ml.NodeModelSave

Fields

Spark ML ROC

Туре

transform

Class

fire.nodes.etl.NodeROC

Name	Title	Description
probabilityCol	Probability Column	
labelCol	Label Column	

CrossValidator

This node represents Cross Validator from Spark ML

Input

It takes in a DataFrame, Estimator and Evaluator as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-crossvalidator

Class

fire.nodes.ml.NodeCrossValidator

Fields

Name	Title	Description
numFolds	Num Folds	The number of folds

Details

This node represents Cross Validator from Spark ML.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-guide.html# example-model-selection-via-cross-validation

Spark Pipeline

This node represents Pipeline from Spark ML

Input

It takes in a DataFrame as input.

Output

The incoming DataFrame is passed to the output.

Туре

ml-pipeline

Class

fire.nodes.ml.NodePipeline

Fields

Details

This node represents Pipeline from Spark ML.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-guide.html#pipeline-components

07-SplitDataset

Split With Stratified Sampling

This node splits the incoming DataFrame into 2. It takes in the fraction to use in splitting the data by Stratified Sampling.

Input

It takes in a DataFrame as input

Output

The input DataFrame is split into 2 DataFrames and output

Туре

transform

Class

 $fire.nodes.util. \\ Split \\ With \\ Stratified \\ Sampling$

Fields

Name	Title	Description
keyInputCol	Column Name	column that defines strata
fraction	Fraction	sampling fraction for each stratum. If a stra- tum is not specified, we treat its fraction as zero
seed	Seed	random seed

Details

Split With Stratified Sampling, which is the preferred way to sample from populations with varing subpopulation sizes.

More details are available at : https://spark.apache.org/docs/latest/api/python/_modules/pyspark/sql/dataframe.html# DataFrame.sampleBy

Split

This node splits the incoming DataFrame into 2. It takes in the fraction to use in splitting the data. For example, if the fraction is .7, it would split the data into 2 DataFrames, one containing 70% of the rows and the other containing the remaining 30%.

Input

It takes in a DataFrame as input

Output

The input DataFrame is split into 2 DataFrames and output

Туре

transform

Class

fire.nodes.ml.NodeSplit

Name	Title	Description
fraction1	Fraction 1	Fraction to be used for Splitting the DataFrame into two. The first DataFrame would go to the lower edge output. The
		other would go to the higher edge output.

SplitProbabilityColumn

Туре

transform

Class

fire.nodes.ml.NodeSplitProbabilityCol

Fields

Name	Title	Description
probabilityColName	Probability Column	
numFields	NumFields	Number of fields in probability columns to
		extract

10-Classification

MultiLayerPerceptron

It supports creation of full connected neural network.

Туре

ml-estimator

Class

fire.nodes.ml. NodeMultilayer Perceptron

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting
labelCol	Label Column	The label column for model fitting
predictionCol	Prediction Column	The prediction column created during
		model scoring.
layers	Layers - comma	The integer array specifying the number of
	separated list of	activation units in each layer
	integers	
maxIter	Max number of iter-	Number of iterations to train the Neural net-
	ations	work
blockSize	Block Size	Block size
seed	Seed	The initial seed to initialise the neural net-
		work.
tol	Tol	
solver	Solver	solver
stepSize	Step Size	Step size

GBTClassifier

Gradient-Boosted Trees (GBTs) is a learning algorithm for classification. It supports binary labels, as well as both continuous and categorical features. Note: Multiclass labels are not currently supported.

Input

It takes in a DataFrame as input and performs GBT Classification

Output

The GBT Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeGBTClassifier

Fields

XGBoostClassifier

Input

It takes in a DataFrame as input and performs XGBoost Classification

Output

The XGBoost Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeXGBoostClassifier

Fields

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for model fitting
labelCol	Label Column	The label column for model fitting
predictionCol	Prediction Column	The prediction column created during model scoring.
numClass	Num Class	
maxDepth	Max Depth	The Maximum depth of a tree
maxBins	Max Bins	The maximum number of bins used for dis- cretizing continuous features.Must be >= 2 and >= number of categories in any categor- ical feature.
maxLeaves	Max Leaves	
numRound	Num Round	
numWorkers	Num Workers	
objective	Objective	
eta	Eta	
regLambda	Reg Lambda	
regAlpha	Reg Alpha	
subsample	Subsample	
sampleType	SampleType	
treeMethod	TreeMethod	
useExternalMemory	UseExternalMemory	
seed	Seed	
baseScore	Base Score	
minChildWeight	Min Child Weight	
colsampleBylevel	ColSampleByLevel	
colsampleBytree	ColSampleByTree	

Continued on next page

Name	Title	Description
minSplitLoss	MinSplitLoss	
maxDeltaStep	MaxDeltaStep	
sketchEps	SketchEps	
scalePosWeight	ScalePosWeight	
growPlicy	GrowPlicy	
normalizeType	NormalizeType	
skipDrop	SkipDrop	
rateDrop	RateDrop	

Table 1 – continued from previous page

LogisticRegression

Logistic regression. Currently, this class only supports binary classification.

Input

This takes in a DataFrame and performs Logistic Regression

Output

The Logistic Regression Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeLogisticRegression

Fields

Details

Logistic regression is a popular method to predict a categorical response.

It is a special case of Generalized Linear models that predicts the probability of the outcomes. In spark.ml logistic regression can be used to predict a binary outcome by using binomial logistic regression, or it can be used to predict a multiclass outcome by using multinomial logistic regression.

More details are available at : https://spark.apache.org/docs/2.3.0/ml-classification-regression.html# logistic-regression

Examples

The below example is available at : https://spark.apache.org/docs/2.3.0/ml-classification-regression.html#logistic-regression

 $import\ org. a pache. spark.ml. classification. Logistic Regression$

// Load training data val training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

val lr = new LogisticRegression() .setMaxIter(10) .setRegParam(0.3) .setElasticNetParam(0.8)

// Fit the model val lrModel = lr.fit(training)

// Print the coefficients and intercept for logistic regression println(s"Coefficients: \${lrModel.coefficients} Intercept: \${lrModel.intercept}")

// We can also use the multinomial family for binary classification val mlr = new LogisticRegression()

.setMaxIter(10) .setRegParam(0.3) .setElasticNetParam(0.8) .setFamily("multinomial")

val mlrModel = mlr.fit(training)

// Print the coefficients and intercepts for logistic regression with multinomial family println(s"Multinomial coefficients: \${mlrModel.coefficientMatrix}") println(s"Multinomial intercepts: \${mlrModel.interceptVector}")

DecisionTreeClassifier

It supports both binary and multiclass labels, as well as both continuous and categorical features.

Input

It takes in a DataFrame and performs Decision Tree Classification

Output

The Decision Tree Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeDecisionTreeClassifier

Fields

Details

Decision trees supports both binary and multiclass labels, as well as both continuous and categorical features.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-classification-regression.html# decision-tree-classifier

NaiveBayes

Creates a NaiveBayes model. Supports both Multinomial NB which can handle finitely supported discrete data. For example, by converting documents into TF-IDF vectors, it can be used for document classification. By making every vector a binary (0/1) data, it can also be used as Bernoulli NB.The input feature values must be nonnegative

Туре

ml-estimator

Class

fire.nodes.ml.NodeNaiveBayes

Fields

Name	Title	Description
featuresCol	Features Column	Features column of type vectorUDT for
		model fitting
labelCol	Label Column	The label column for model fitting
predictionCol	Prediction Column	The prediction column created during
		model scoring
modelType	modelType	The model type. Supported options: multi-
		nomial and bernoulli. (default = multino-
		mial)
smoothing	Smoothing	The smoothing parameter.

RandomForestClassifier

Supports both binary and multiclass labels, as well as both continuous and categorical features.

Input

Takes in a DataFrame and performs Random Forest Classification

Output

Random Forest Classification Model generated is passed along to the next nodes. The input DataFrame is also passed along to the next nodes

Туре

ml-estimator

Class

fire.nodes.ml.NodeR and omForestClassifier

Fields

Details

Random forests supports both binary and multiclass labels, as well as both continuous and categorical features.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/ml-classification-regression.html# random-forest-classifier

13-EvaluatePredict

MulticlassClassificationEvaluator

Evaluator for multiclass classification, which expects two input columns: score and label.

Туре

ml-evaluator

Class

fire.nodes.ml. Node Multiclass Classification Evaluator

Fields

Name	Title	Description
labelCol	Label Column	The label column for model fitting.
predictionCol	Prediction Column	The prediction column.
metricName	Metric Name	The metric used in evaluation.

Details

Evaluator for multiclass classification, which expects two input columns: score and label.

More at Spark MLlib/ML docs page :https://spark.apache.org/docs/1.6.0/mllib-evaluation-metrics.html# multiclass-classification

RegressionEvaluator

Evaluator for regression, which expects two input columns: prediction and label.

Input

It takes in a DataFrame as input

Output

The incoming DataFrame is passed to the output

Туре

ml-evaluator

Class

fire.nodes.ml.NodeRegressionEvaluator

Fields

Name	Title	Description
labelCol	Label Column	The label column for model fitting.
predictionCol	Prediction Column	The prediction column.
metricName	Metric Name	The metric used in evaluation.

Details

Evaluator for regression, which expects two input columns: prediction and label.

More at Spark MLlib/ML docs page:

http://spark.apache.org/docs/1.6.0/api/scala/index.html#org.apache.spark.ml.evaluation.RegressionEvaluator

Predict

Predict node takes in a DataFrame and Model and makes predictions

Input

It takes in a DataFrame and Model as input

Output

A new column containing the predictions is added to the input DataFrame

Туре

ml-predict

Class

fire.nodes.ml.NodePredict

Fields

BinaryClassificationEvaluator

Evaluator for binary classification, which expects two input columns: rawPrediction and label.

Output

It outputs the Probability for each class

Туре

ml-evaluator

Class

fire.nodes.ml.NodeBinaryClassificationEvaluator

Fields

Name	Title	Description
labelCol	Label Column	The label column for model fitting.
predictionCol	Prediction Column	The prediction column.
metricName	Metric Name	The metric used in evaluation.

Details

Evaluator for binary classification, which expects two input columns: rawPrediction and label.

More at Spark MLlib/ML docs page : http://spark.apache.org/docs/latest/mllib-evaluation-metrics.html# binary-classification

06-FeatureSelection

ChiSqSelector

Chi-Squared feature selection, which selects categorical features to use for predicting a categorical label.

Туре

ml-transformer

Class

fire.nodes.ml.NodeChiSqSelector

Fields

Name	Title	Description
featuresCol	Features Column	The features column name
outputCol	Output Column	The output column name
labelCol	Label Column	The label column name
numTopFeatures	NumTopFeatures	Number of features that selector will select
		(ordered by statistic value descending).

VectorSlicer

VectorSlicer feature selection, which takes a feature vector and outputs a new feature vector with a sub-array of the original features. It is useful for extracting features from a vector column

Туре

ml-transformer

Class

fire.nodes.ml.NodeVectorSlicer

Fields

Name	Title	Description
inputCol	Features Column	The features column name
outputCol	Output Column	The output column name

24.1.6 ML-TS

ARIMA

Туре

ml-transformer

Class

fire.nodes.ts.NodeAutoARIMA

Fields

Name	Title	Description
У	Y	The time-series to which to fit the ARIMA
		estimator
seasonal	Seasonal	Whether to fit a seasonal ARIMA. Default
		is True
stepwise	Stepwise	Whether to use the stepwise algorithm to
		identify the optimal model parameters.
trace	Trace	Whether to print status on the fits.
suppress_warnings	Suppress Warnings	If suppress_warnings is True, all of the
		warnings coming from ARIMA will be
		squelched.
error_action	Error Action	If unable to fit an ARIMA for whatever rea-
		son, this controls the error-handling behav-
		ior. One of (warn, raise, ignore)
scoring	Scoring	The metric to use for scoring the out-of-
		sample data. One of (mse, mae)
n_periods	Forecast	Int number of periods to forecast forward.

Prophet

Туре

ml-transformer

Class

fire.nodes.ts.NodeProphet

Fields

24.1.7 02-Parse

FieldSplitter

This node splits the string of the specified input column using the specified delimiter

Input

It accepts a DataFrame as input from the previous Node

Output

New columns are added to the incoming DataFrame with values from the result of splitting the value in the input column

Туре

transform

Class

fire.nodes.etl.NodeFieldSplitter

Fields

Name	Title	Description
inputCol	Input Column	input column name
outputCols	Output Columns	new column names separed by comma','.(eg: col1,co2,col3)
sep	Separator	separator to split the input column value(default: space)
onError	On Error	

RegexTokenizer

This node creates a new DataFrame by the process of taking text (such as a sentence) and breaking it into individual terms (usually words) based on regular express

Туре

transform

Class

fire.nodes.etl.NodeRegexTokenizer

Name	Title	Description
inputCol	Column	input column for tokenizing
outputCol	Tokenized Column	New output column after tokenization
pattern	Pattern	The regex pattern used to match delimiters
gaps	Gaps	Indicates whether the regex splits on gaps

Fixed Length Fields

Fixed Length

Туре

transform

Class

fire.nodes.etl.NodeFixedLength

Fields

Name	Title	Description
inputCol	Input Column	input column name
outputColNames	Column Names for	New Output Columns of the SQL
	the CSV	
outputColTypes	Column Types for	Data Type of the Output Columns
	the CSV	
colLengths	Length of each col-	Length of the columns in characters
	umn	
outputColFormats	Column Formats for	Format of the Output Columns
	the CSV	

ApacheLogs

Reads in Apache Log files from a given path, parses them and loads them into a DataFrame

Туре

dataset

Class

fire.nodes.logs.NodeApacheFileAccessLog

Name	Title	Description
path	Path	Full path for the directory or file for the
		Apache File Logs

ParseJSONCol

Parses JSON content in a given Col

Туре

transform

Class

fire.nodes.etl.NodeParseJSONColumn

Fields

Name	Title	Description
jsonColName	JSON Col Name	Column containing the JSON Content
inputCol	Input Col	Input Columns
jsonFieldNames	JSON Field names	JSON Field names
jsonFieldTypes	JSON Field Type	Data Type of the JSON field

OCR

Performs Optical Character Recognition using the Tesseract Library. Please make sure the TESSDATA_PREFIX environment variable is set to the parent directory of your 'tessdata' directory. Download the tessdata directory with git clone https://github.com/tesseract-ocr/tessdata.git

Туре

transform

Class

fire.nodes.ocr.NodeOCRTesseract

Name	Title	Description
imageNameCol	Image Name Col-	input image column name
	umn	
imageCol	Image Column	input image column name
outputCol	Output OCR Col-	output column name
	umn	

MultiRegexExtractor

This node to extract pattren from input columns

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node extract pattren from input columns as specified

Туре

transform

Class

fire.nodes.etl.NodeMultiRegexExtractor

Fields

Name	Title	Description
inputColNames	InputColumnsName	Columns
outputColNames	OuputColumnsName	name of the output column
patterns	Patterns	patterns or regex to extract the input column
		name
groups	Groups	An regular expression group number start-
		ing with 1, defining which portion of the
		matching string will be returned

24.1.8 06-Filter

FilterByDateRange

This node filters Rows within the given date range

Туре

transform

Class

fire.nodes.etl.NodeFilterByDateRange

Fields

Name	Title	Description
inputCol	Column	input column name
fromDateCol	From Date	Takes Start Date in the form of yyyy-MM- dd
toDateCol	To Date	Takes End Date in the form of yyyy-MM-dd

FilterByNumberRange

This node filter Rows in the given Number Range

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeFilterByNumberRange

Fields

Name	Title	Description
inputCol	Input Column Name	input column name
lowestValue	Lowest Value	input lowest value
highestValue	Highest Value	input highest value

ColumnFilter

This node creates a new DataFrame that contains only the selected columns

Input

This type of node takes in a DataFrame and transforms it to another DataFrame.

Output

This node filters the specified columns from the incoming DataFrame

Туре

transform

Class

fire.nodes.etl.NodeColumnFilter

Fields

Name	Title	Description
outputCols	Columns	Columns to be included in the output
		DataFrame

RowFilter

This node creates a new DataFrame containing only rows satisfying given condition

Input

It accepts DataFrame as input from the previous Node

Output

This node filters the rows based on the conditional expression to generate the output DataFrame

Туре

transform

Class

fire.nodes.etl.NodeRowFilter

Name	Title	Description
conditionExpr	Conditional Expres-	The filtering condition. Rows not satisfying
	sion	given condition will be excluded from out-
		put DataFrame. eg: usd_pledged_real > 0
		and (category = 1 or category == 2) and goal
		> 100

Details

This node creates a new DataFrame containing only rows satisfying the given condition.

Examples of Conditional Expression

col1 > 5 AND col2 > 3
name is not NULL
name is NULL
usd_pledged_real > 0 and (category = "Narrative Film" or category == "Music") and goal > 100
datetime > '2011-01-01 00:00:00.0' (datetime column is of type timestamp)
datetime > '2011-01-01 00:00:00.0' and datetime < '2016-01-01 00:00:00.0'</pre>

FilterByStringLength

This node filters the Rows within the given string length. The column to be used for determining the string length is specified

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeFilterByStringLength

Name	Title	Description
inputCol	Input Column Name	input column name
minLength	Minimum length	Minimum length of String
maxLength	Maximum length	Maximum length of String

NodeRowFilterByIndex

This node creates a new DataFrame containing only rows satisfying given condition

Input

It accepts DataFrame as input from the previous Node

Output

This node filters the rows based on the conditional expression to generate the output DataFrame

Туре

transform

Class

fire.nodes.etl.NodeRowFilterByIndex

Fields

Name	Title	Description
indexes	Indexes	Comma separated index values starts from
		0. ex: 0, 1, 2, 5
indexesRange	IndexesRange	Index ranges example like 10-14 i.e 10, 11,
		12, 13, 14.

DropColumns

This node creates a new DataFrame by deleting columns specified as an input

Input

It takes in a DataFrame as input

Output

The specified columns are dropped from the incoming DataFrame to generate the output DataFrame

Туре

transform

Class

fire.nodes.etl.NodeDropColumns

Fields

Name	Title	Description
dropCols	Columns	The columns to be excluded from the output
		DataFrame

24.1.9 18-OpenNLP

OpenNLPNameFinder

This node finds names using OpenNLP. It takes in the OpenNLP model. Models can be downloaded from http://opennlp.sourceforge.net/models-1.5/

Input

It takes in a DataFrame as input.

Output

It extracts the names from the specified column and stores the result in the specified output column.

Туре

transform

Class

fire.nodes.opennlp.NodeOpenNLPNameFinder

Name	Title	Description
model	Model	Path to the model file (on HDFS when run-
		ning on the cluster)
inputCol	Input Text Column	input column name
outputCol	Output Column	Output Column containing the results

Details

This node performs namefinder using OpenNLP to easily detect named entities and numbers in text.

To be able to detect entities the Name Finder needs a model. The model is dependent on the language and entity type it was trained for.

https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html#tools.namefind.recognition.cmdline

The OpenNLP project offers a number of pre-trained name finder models which are trained on various freely available corpora. They can be downloaded at the OpenNLP download page.

http://opennlp.sourceforge.net/models-1.5/

OpenNLPSentenceDetector

This node detects sentences using OpenNLP - https://opennlp.apache.org/documentation/1.7.2/manual/opennlp.html# tools.sentdetect. It takes in the OpenNLP model. Models can be downloaded from http://opennlp.sourceforge.net/ models-1.5/

Input

It takes in a DataFrame as input.

Туре

transform

Class

fire.nodes.opennlp.NodeOpenNLPSentenceDetector

Name	Title	Description
model	Model	Path to the model file (on HDFS when run-
		ning on the cluster)
inputCol	Input Text Column	input cpulmn name
outputCol	Output Column	Output Column containing the results

Details

This node detects sentences using OpenNLP -

https://opennlp.apache.org/documentation/1.7.2/manual/opennlp.html#tools.sentdetect.

It takes in the OpenNLP model. Models can be downloaded from http://opennlp.sourceforge.net/models-1.5/

NodeOpenNLPDocumentCategorizer

This node classifies text into pre-defined categories using OpenNLP - https://opennlp.apache.org/documentation/1.7.2/ manual/opennlp.html#tools.doccat. It takes in the OpenNLP model. Models can be downloaded from http://opennlp. sourceforge.net/models-1.5/

Input

It takes in a DataFrame as input.

Output

It finds the Document Category and stores the result in the specified output column.

Туре

transform

Class

fire.nodes.opennlp.NodeOpenNLPDocumentCategorizer

Fields

Name	Title	Description
model	Model	Path to the model file (on HDFS when run-
		ning on the cluster)
inputCol	Input Text Column	input cpulmn name
outputCol	Output Column	Output Column containing the results

Details

This node classifies text into pre-defined categories using OpenNLP

https://opennlp.apache.org/documentation/1.7.2/manual/opennlp.html#tools.doccat.

It takes in the OpenNLP model. Models can be downloaded from http://opennlp.sourceforge.net/models-1.5/

24.1.10 15-ScoreCardPy

Binning Scorecard

Туре

ml-transformer

Class

fire.nodes.scorecardpy.NodeBinning

Fields

Name	Title	Description
У	Y	
X	Х	
stopLimit	StopLimit	
countDistrLimit	CountDistrLimit	
binNumLimit	BinNumLimit	
method	Methos	
positive	Positive	

VariableSelection Scorecard

Туре

ml-transformer

Class

fire.nodes.scorecardpy.NodeVariableSelection

Name	Title	Description
У	Y	
ivLimit	IvLimit	
missingLimit	MissingLimit	
identicalLimit	IdenticalLimit	
positive	Positive	

24.1.11 03-Prepare

13-Others

MultiWindowAnalytics

Туре

transform

Class

fire.nodes.etl.NodeMultiWindowAnalytics

Fields

Name	Title	Description
analyticsCols	AnalyticsColumn	
windowFunctions	Window Function	Window Function Name
partitionByCols	PartitionBy	partition column names separated by comma(,)
orderByCols	OrderBy	order by column names separated by comma(,)
outPutColumns	OutPutColumn	Enter output field(column) name

RoundValue

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.etl.NodeRoundDouble

Name	Title	Description
inputCols	Input Column	The columns containing double or float val-
		ues to round.
precision	Precision	The scale of the double values to round to.

SortBy

It sorts the incoming DataFrame on the fields specified.

Туре

transform

Class

fire.nodes.etl.NodeSortBy

Fields

Name	Title	Description
description	Description	Description
sortByColNames	Columns	Columns on which to Sort By
ascDesc	Sorting Order	Whether to sort in ascending or descending
		order

Transpose

This node transposes a dataframe without performing aggregation function by given column(transposeby). ALL IN-PUT COLUMNS TO THIS NODE HAVE TO BE OF THE SAME TYPE

Input

It accepts a DataFrame as input from the previous Node

Output

Output dataframe consisting of three columns transposeBy, column_name, column_value

Туре

transform

Class

fire.nodes.etl.NodeTranspose

Name	Title	Description
transposeBy	TransposeByColumn	transposeBy column name
	Name	

WindowRanking

Туре

transform

Class

fire.nodes.etl.NodeWindowRanking

Fields

Name	Title	Description
partitionByCols	PartitionBy	partition column names separated by
		comma(,)
orderByCols	OrderBy	order by column names separated by
		comma(,)
windowFunction	Window Function	Window Function Name

GeoPoint

Туре

transform

Class

fire.nodes.etl.NodeGeoPoint

Name	Title	Description
longitude	Longitude	
latitude	Latitude	

MultiWindowRanking

Туре

transform

Class

fire.nodes.etl.NodeMultiWindowRanking

Fields

Name	Title	Description
windowFunctions	WindowFunction	Window Function Name
partitionByCols	PartitionBy	partition column names separated by comma(,)
orderByCols	OrderBy	order by column names separated by comma(,)
outPutColumns	OutputColumn	Enter output field(column) name

ColumnsRename

This node creates a new DataFrame by renaming existing columns with new name

Input

This type of node takes in a DataFrame and transforms it to another DataFrame.

Output

The specified columns are renamed to have the new names.

Туре

transform

Class

fire.nodes.etl.NodeColumnsRename

Name	Title		Description
currentColNames	Current	Column	Current Column Names
	Names		
newColNames	Columns	New	New name for existing columns
	Name		

RecoverHivePartitions

Node to recover the partitions of external hve table.

Туре

doc

Class

fire.nodes.etl.NodeRecoverHivePartitions

Fields

Name	Title	Description
databaseName	HIVE Database	Name of the HIVE Database
tableName	HIVE Table	Name of the HIVE table

Details

This node is used recover the partitions of external hve table.

It will run the command: "MSCK REPAIR TABLE \${databaseName}.\${tableName}"

CDCUsingFullTableMerge

CDC Using Full Table Merge

Туре

transform

Class

fire.nodes.etl.NodeCDCUsingFullTableMerge

Name	Title	Description
baseTable	Base Table Name	Name of the Base Table
idCols	ID Column Names	ID Column names
modifiedDateCol	Modified Date Col-	Modified Date Column
	umn	

Count

This node counts the number of records in the incoming Dataframe and puts the count into a variable to the used by subsequent Nodes

Input

It accepts a DataFrame as input from the previous Node

Output

The incoming DataFrame is sent to the output

Туре

transform

Class

fire.nodes.etl.NodeCount

Fields

Name	Title	Description
variable	Variable Name	Name of the Variable in which the count is
		stored

Sample

Samples the incoming DataFrame

Туре

fire.nodes.etl.NodeSample

Fields

Name	Title	Description
withReplacement	With Replacement	With or without Replacement
fraction	Fraction	Fraction
seed	Seed	Seed

SortColumns

It sort the columns selection.

Туре

transform

Class

fire.nodes.etl.NodeSortColumns

Fields

Name	Title	Description
sortColumnNames	Columns	Sort the Column Name

RegisterTempTable

This node registers the incoming DataFrame as a temporary table in Spark

Input

It accepts a DataFrame as input from the previous Node

Output

The incoming DataFrame is output without any changes

Туре

fire.nodes.etl.NodeRegisterTempTable

Fields

Name	Title	Description
tempTable	Temporary Table	Name of the temporary table to be created

GeolP

This node converts IP to geo location

Input

The input dataframe is passed in the variable inDF

Output

Transforms the IP to Geo location

Туре

transform

Class

fire.nodes.etl.NodeGeoIP

Fields

Name	Title	Description
ipCol	IP Column	IP Column in the DataFrame
databaseFilePath	Database File Path	Database File Path

WindowAnalytics

Туре

fire.nodes.etl.NodeWindowAnalytics

Fields

Name	Title	Description
partitionByCols	PartitionBy	partition column names separated by
		comma(,)
orderByCols	OrderBy	order by column names separated by
		comma(,)
windowFunction	Window Function	Window Function Name
analyticsCol	Analytics Column	
window_offset	Window Offset	It's used in lead and lag functions.

10-Condition

Assert

This Node takes in an expression. It evaluates the expression and based on the results sends the execution to the first or the second output Node

Input

It accepts a DataFrame as input from the previous Node

Output

The incoming DataFrame is sent to the output. Only one of the output Nodes receives the DataFrame based on the results of the expression

Туре

transform

Class

fire.nodes.etl.NodeAssert

Name	Title	Description
expression	Expression	Expression to be evaluated. It can use vari-
		ables computed in the previous Nodes

Decision

It computes expressions to determine if the condition is met or not. Accordingly proceeds to the next step or stops here.

Туре

transform

Class

fire.nodes.etl.NodeDecision

Fields

Name	Title	Description
description	Description	Description
inputCols	Columns	Columns
functions	Function	Function to apply
symbols	Symbol	Symbol to apply
values	Values	Values

09-Split

Split By Expression

This node splits the incoming DataFrame into two output DataFrames by applying the conditional logic

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeSplitByExpression

Name	Title	Description
conditionExpr	Conditional Expres-	Conditional Expression to be used for Split-
	sion to split the Data	ting the DataFrame into two. DataFrame
	on	which matches the condition will go to the
		lower edge output. The other would go to
		the higher edge output.

SplitByMultipleExpressions

Splits the incoming DataFrame into multiple output DataFrames by applying the conditional logic

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeSplitByMultipleExpressions

Name	Title	Description
conditionExpr1	Conditional Expres-	Conditional Expression 1 to be used for
	sion 1 to split the	Splitting the Dataset
	Data on	
conditionExpr2	Conditional Expres-	Conditional Expression 2 to be used for
	sion 2 to split the	Splitting the Dataset
	Data on	
conditionExpr3	Conditional Expres-	Conditional Expression 3 to be used for
	sion 3 to split the	Splitting the Dataset
	Data on	
conditionExpr4	Conditional Expres-	Conditional Expression 4 to be used for
	sion 4 to split the	Splitting the Dataset
	Data on	
conditionExpr5	Conditional Expres-	Conditional Expression 5 to be used for
	sion 5 to split the	Splitting the Dataset
	Data on	

CompareAllColumnsSingleOutput

Compares 2 incoming DataFrames. Outputs 1 DataFrame (A-B) or (B-A) or (A intersection B) based on user's input

Туре

join

Class

fire.nodes.etl.NodeCompareAllColumnsSingleOutput

Fields

Name	Title	Description
compareOption	Compare Type	Comparision options whether (A-B) or (B-
		A) or (A intersection B)

Compare Specific Columns

Compares 2 incoming DataFrames on specific columns. Outputs 1 DataFrame (A-B) or (B-A) or (A intersection B) based on user's input

Туре

join

Class

fire.nodes.etl.NodeCompareSpecificColumnsSingleOutput

Fields

Name	Title	Description
columnsToCompare	Columns to Com-	Columns to be used in the comparison
	pare	
compareOption	Compare Type	Comparision options whether (A-B) or (B-
		A) or (A intersection B)

CompareSpecificColumns

Compares 2 incoming DataFrames on specific columns. Outputs 3 DataFrames (A-B), (B-A), (A intersection B)

Туре

join

Class

fire.nodes.etl.NodeCompareSpecificColumns

Fields

Name	Title	Description
columnsToCompare	Columns to Com-	Columns to be used in the comparison
	pare	

Compare All Columns

Compares 2 incoming DataFrames. Outputs 3 DataFrames (A-B), (B-A), (A intersection B)

Туре

join

Class

fire.nodes.etl.NodeCompareAllColumns

Fields

11-AddColumn

Expressions

Expressions

Туре

transform

Class

fire.nodes.etl.Node Expressions

Name	Title	Description
description	Description	Description
outputCols	New Columns	New Columns Name
	Name	
expressions	Expressions	Expressions

AddColumns

This node allows adding new columns with certain values

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node adds the user specified columns to the DataFrame

Туре

transform

Class

fire.nodes.etl.NodeAddColumns

Name	Title	Description
addCurrentDateCol	Add Current Date	Whether to add the current date as a new
	Column	column
currentDateColName	Current Date Col-	Name of the new Current Date Column Cre-
	umn Name	ated
addCurrentTimeCol	Add Current Time	Whether to add the current time as a new
	Column	column
currentTimeColName	Current Time Col-	Name of the new Current Time Column
	umn Name	Created
addConstantStringCol1	Add Constant String	Whether to add a new columns with con-
	Column	stant string value
constantStringColName1	Constant String	Constant String Name
	Column Name	
constantStringColValue1	Constant String	Constant String Value
	Column Value	
addConstantIntCol1	Add Constant Inte-	Whether to add a new columns with con-
	ger Column	stant integer value
constantIntColName1	Constant Integer	Constant Integer Column Name
	Column Name	
constantIntColValue1	Constant Integer	Constant Integer Value
	Column Value	

GenerateUID

This node Generates a new column with unique Index/Value for each row in the Dataset for each partition. Each Partition starts a new range.

Туре

transform

Class

fire.nodes.etl.NodeGenerateUID

Fields

Name	Title	Description
outputCol	UID Column Name	UID column name

Hash

This node adds a new Columns which contains the Hash of the specified columns

Input

It accepts a DataFrame as input from the previous Node

Output

A new column is added to the incoming DataFrame by creating a Hash of the specified input columns.

Туре

transform

Class

fire.nodes.etl.NodeHash

Fields

Name	Title	Description
inputCols	Columns	Columns to be concatenated
hashingAlgorithm	Hashing Algorithm	Hashing Algorithm
outputCol	Output Column	Column name for Hash
	Name	
bitLength	Bit Length	Bit Length
sep	Separator	Separator to be used when concatenating the
		columns

GenerateUUID

This node Generates a Universally Unique ID

Input

It accepts a dataframe as input

Output

It adds a new column for UUID to the input DataFrame. This new DataFrame is sent to the output

Туре

fire.nodes.etl.NodeGenerateUUID

Fields

Name	Title	Description
outputCol	Output Column	Output Column Name

CaseWhen

Sets values based on conditions

Туре

transform

Class

fire.nodes.etl.NodeCaseWhen

Fields

Name	Title	Description
outputCol	Output Column	output column name
	Name	
whenConditions	When Condition	When Condition
values	Value	Value when this condition is true
finallyElse	Else	else

ConcatColumns

This node creates a new DataFrame by concatenating the specified columns of the input DataFrame

Input

It accepts a DataFrame as input from the previous Node

Output

A new column is added to the incoming DataFrame by concatenating the specified columns. The new DataFrame is sent to the output of this Node.

Туре

transform

Class

fire.nodes.etl.NodeConcatColumns

Fields

Name	Title	Description
inputCols	Columns	Columns to be concatenated
outputCol	Concatenated	Column name for the concatenated columns
	Column Name	
sep	Separator	Separator to be used when concatenating the
		columns

ZipWithIndex

This node Generates a new column with unique Index/Value for each row in the Dataset

Туре

transform

Class

fire.nodes.etl.NodeZipWithIndex

Fields

Name	Title		Description
indexColName	Index	Column	Index column name
	Name		

12-CastDataType

CastToSingleType

This node creates a new DataFrame by casting the specified input columns to a new data type

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node casts the data type of columns as specified

Туре

transform

Class

fire.nodes.etl.NodeCastColumnType

Fields

Name	Title	Description
inputCols	Columns	Columns to be cast to new data type
outputColType	New Data Type	New data type(INTEGER, DOUBLE,
		STRING, LONG, SHORT)
replaceExistingCols	Replace Existing	Whether to replace existing columns or cre-
	Cols	ate new ones

CastToDifferentTypes-2

This node creates a new DataFrame by casting the specified columns into new types

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node casts the data type of columns as specified

Туре

transform

Class

fire.nodes.etl.NodeMultiCastColumnType2

Name	Title	Description
inputColNames	Columns	Columns
outputColTypes	New Data Type	New data type(INTEGER, DOUBLE, STRING, LONG, SHORT)
replaceExistingCols	Replace Existing	Whether to replace existing Columns or cre-
	Cols	ate New Ones
formats	Formats	Formats like yyy-MM-dd used in input &
		output

CastToDifferentTypes-1

This node creates a new DataFrame by casting the specified columns into new types

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node casts the data type of columns as specified

Туре

transform

Class

fire.nodes.etl.NodeMultiCastColumnType

Fields

Name	Title	Description
inputColNames	Columns	Columns
outputColTypes	New Data Type	New data type(INTEGER, DOUBLE,
		STRING, LONG, SHORT)
replaceExistingCols	Replace Existing	Whether to replace existing Columns or cre-
	Cols	ate New Ones

06-Math

MathFunctions

This node performs specified math function on a row

Input

It accepts a DataFrame as input from the previous Node

Output

A new column is added which contains the results of applying the Math function on the given column of the input DataFrame

Туре

transform

Class

fire.nodes.etl. Node MathFunctions

Fields

Name	Title	Description
inputCol	Input Column Name	input column name
mathFunction	Math Function	Math Function Name
outputCol	Output Column	Output Column Name
scale	Scale	Scale to be used when applying the Math
		Function

MathFunctionsMultiple

Math Functions Multiple

Туре

transform

Class

fire.nodes.etl. Node Math Functions Multiple

Name	Title	Description
description	Description	Description
inputCols	Columns	Columns
functions	Function	Math Function to apply
replaceExistingCols	Replace Existing	Replace Existing Columns (true, false)
	Cols	
scales	Scale	Scale to be used when applying the Math
		Function

MathExpression

Туре

transform

Class

fire.nodes.etl.NodeMathExpression

Fields

Name	Title	Description
outputCols	OutPut Column	Output Column Name
expressions	Math Expression	Define math expression.

03-DateTime

DateDifference

This node finds difference between two dates

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeDateDiff

Name	Title	Description
fromDate	FromDate	From date column name
toDate	Todate	To date column name
useCurrentDateAsToDateCol	useCurrentDateAsTo	CoCurrent Date As ToDate
days	Days	Days difference
hours	Hours	Hours difference
minutes	Minutes	Minutes difference
seconds	Seconds	Seconds difference

Details

Calculates difference between 2 given dates. Difference between dates is displayed in days, hours, minutes, and seconds columns.

Examples

Format Examples

dd-MM-yy : 30-11-95 to 19-02-18 diff- 8608 days : 206609 hours : 12396574 min : 743794461 : second dd-MMyyyy : 10-02-1996 to 20-09-2017 diff- 8536 days : 204881 hours : 12292884 min : 737573070 : second MM-dd-yyyy : 19-10-1994 to 06-12-2017 diff- 9015 days : 216377 hours : 12982644 min : 778958670 : second yyyy-MM-dd : 1994-12-25 to 2019-01-16 diff- 8948 days : 214769 hours : 12886164 min : 773169870 : second yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 to 2010-12-30 22:59:59 diff-397 days: 1 hour: 0 minutes : 0 seconds

TimeFunctions

Туре

transform

Class

fire.nodes.etl.NodeTimeFunctions

Name	Title	Description
timeStampCol	TimeStamp Column	input column name
	Name	
timeFunctions	Time Functions	Time Functions Name

DateTimeFieldExtract

It creates a new DataFrame by extracting Date and Time fields.

Input

It takes in a DataFrame as Input

Output

Node to extract year/month/dayofmonth/hour/minute/seconad values from TimeStamp

Туре

transform

Class

fire.nodes.etl.NodeDateTimeFieldExtract

Fields

Name	Title	Description
inputCol	Column	The input column name
extractYear	Extract Year	Extract Year
extractMonth	Extract Month	Extract Month
extractDayOfMonth	Extract Day of	Extract Day of Month
	Month	
extractHour	Extract Hour	Extract Hour
extractMinute	Extract Minute	Extract Minute
extractSecond	Extract Second	Extract Second
extractWeekOfYear	Extract WeekO-	Extract WeekOfYear
	fYear	

Details

Extracts year, month, day of month, hour, minute, second and week of year in different columns.

StringToUnixTime

This nodes converts a string to Unix Time

Туре

fire.nodes.etl.NodeStringToUnixTime

Fields

Name	Title	Description
inputColName	Input Column Name	Input Column Name
inputColFormat	Input Column For-	Input Column Format (eg: yyyy-MM-dd
	mat	HH:mm:ss)
outputColName	Output Column	Output Column Name
	Name	

Details

This node converts a string column to unix time (seconds).

Examples

Format Examples

dd-MM-yy : 31-01-12 dd-MM-yyyy : 31-01-2012 MM-dd-yyyy : 01-31-2012 yyyy-MM-dd : 2012-01-31 yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 yyyy-MM-dd HH:mm:ss.SSS : 2012-01-31 23:59:59.999 yyyy-MM-dd HH:mm:ss.SSSZ : 2012-01-31 23:59:59.999+0100 EEEEE MMMMM yyyy HH:mm:ss.SSSZ : Saturday November 2012 10:45:42.720+0100

Example: Date (string), Format, Unix time (seconds)

2003-07-25, yyy-MM-dd, 1059091200

StringToDate

This node converts a string column to date using the given date/time format

Туре

transform

Class

fire.nodes.etl.NodeMultiStringToDate

Name	Title	Description
inputColNames	Columns	Columns
inputColFormats	Input Column For-	Input Column Formats. eg: yyyy-MM-dd
	mats	yyyy-MM-dd HH:mm:ss
outputColNames	Output Column	Output Column Names
	Names	
outputColTypes	New Data Types	New data types (DATE, TIMESTAMP)

Details

This node converts multiple string columns to date/time.

Examples

Format Examples

dd-MM-yy : 31-01-12 dd-MM-yyyy : 31-01-2012 MM-dd-yyyy : 01-31-2012 yyyy-MM-dd : 2012-01-31 yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 yyyy-MM-dd HH:mm:ss.SSS : 2012-01-31 23:59:59.999 yyyy-MM-dd HH:mm:ss.SSSZ : 2012-01-31 23:59:59.999+0100 EEEEE MMMMM yyyy HH:mm:ss.SSSZ : Saturday November 2012 10:45:42.720+0100

UnixTimeToString

This node converts Unix Time to String

Туре

transform

Class

fire.nodes.etl.NodeUnixTimeToString

Name	Title	Description
inputColName	Input Column Name	input column name
outputColName	Output Column	Output Column Name
	Name	
outputColFormat	Output Column For-	Output Column Format (eg: yyyy-MM-dd
	mat	HH:mm:ss)

Details

This node converts unix time (seconds) to string type.

Examples

Format Examples

dd-MM-yy : 31-01-12 dd-MM-yyyy : 31-01-2012 MM-dd-yyyy : 01-31-2012 yyyy-MM-dd : 2012-01-31 yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 yyyy-MM-dd HH:mm:ss.SSS : 2012-01-31 23:59:59.999 yyyy-MM-dd HH:mm:ss.SSSZ : 2012-01-31 23:59:59.999+0100 EEEEE MMMMM yyyy HH:mm:ss.SSSZ : Saturday November 2012 10:45:42.720+0100

Example: select an input column (long type), output column name and desired output column format. It will add one more column in string format.

If you input a date format like dd-MM-yy. It will add one column having value like 31-01-12.

DateToString

This node converts a date/time column to string with given format

Туре

transform

Class

fire.nodes.etl.NodeMultiDateToString

Fields

Name	Title	Description
inputColNames	Input Column Name	Input Column Name
outputColFormats	Output Column For-	Output Column Formats. eg: yyyy-MM-dd
	mats	yyyy-MM-dd HH:mm:ss
outputColNames	Output Column	Output Column Names
	Names	

Details

This node converts date/time column to string type with given format.

Examples

Format Examples

dd-MM-yy : 31-01-12 dd-MM-yyyy : 31-01-2012 MM-dd-yyyy : 01-31-2012 yyyy-MM-dd : 2012-01-31 yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 yyyy-MM-dd HH:mm:ss.SSS : 2012-01-31 23:59:59.999 yyyy-MM-dd HH:mm:ss.SSSZ : 2012-01-31 23:59:59.999+0100 EEEEE MMMMM yyyy HH:mm:ss.SSSZ : Saturday November 2012 10:45:42.720+0100

07-String

StringFunctions

This node performs specified String function on a row

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeStringFunctions

Fields

Name	Title	Description
inputCols	Input Column Name	input column name
stringFunction	String Function	String Function Name
replaceExistingCols	ReplaceExistingCols	replaceExistingCols

StringFunctionsMultiple

String Functions Multiple

Туре

transform

Class

fire.nodes.etl.NodeStringFunctionsMultiple

Name	Title	Description
description	Description	Description
inputCols	Columns	Columns
functions	Function	String Function to apply
replaceExistingCols	Replace Existing	Replace Existing Columns (true or false)
	Cols	

TextCaseTransformer

This node converts text to upper or lower case

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeTextCaseTransformer

Fields

Name	Title	Description
inputCol	Input Column Name	input column name
mode	Text Case Type	input to convert text to upper or lower case
outputCol	Output Column	Output Column

05-DataCleaning

DataWrangling

This node creates a new DataFrame by applying each of the Rules specified

Input

It takes in a DataFrame as Input

Output

It creates the output DataFrame by applying the data wrangling rules provided

Туре

transform

Class

fire.nodes.etl.NodeDataWrangling

Fields

Name	Title	Description
rules	Rules	Rules to be applied on column and rows

Details

Rename one column to another rename col:c1 to c2;

Drop Column drop col:col4

Delete columns with some condition delete col:col3 > 44

Substring col:col2 0,3 get substring between 0 and 3rd column from the column col2

Trim Values : Removes leading and trailing whitespace from a string value.

set col:Name value: trim(Name)

Sets the new value of Name column to be trim(Name)

RemoveUnwantedCharactersMult

This node removes unwanted characters

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeRemoveUnwantedCharactersMultiple

Name	Title	Description
inputCols	Input Columns	Input columns
removeWhitespaces	Remove Whites-	Removes white space
	paces	
removeLetters	Remove Letters	Removes letters
removeDigits	Remove Digits	Removes digits
removeSigns	Remove Signs	Removes signs
removeCommas	Remove Commas	Removes commas

ImputingWithMedian

Imputing with median

Туре

transform

Class

fire.nodes.ml.NodeReplaceMissingValueWithMedian

Fields

Name	Title	Description
colNames	Input Columns	Input column of type - all numeric for me-
		dian impute

DropRowsWithNull

This node creates a new DataFrame by dropping rows containing null values

Input

It accepts DataFrame as input from the previous Node

Output

This node drops rows containing null values

Туре

transform

Class

fire.nodes.etl.NodeDropRowsWithNull

Fields

DropDuplicateRows

1>When user don't select any column, returns a new Dataset that contains only the unique rows from this Dataset. 2> Returns a new Dataset with duplicate rows removed, considering only the subset of columns.

Туре

transform

Class

fire.nodes.etl.NodeDropDuplicateRows

Fields

Name	Title	Description
colNames	Columns	Columns to be used in determining if any two rows are duplication. No columns indi- cate to use all the available columns.

FindAndReplaceUsingRegexMultiple

This node finds and replaces text in a column containing string

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeFindAndReplaceUsingRegexMultiple

Name	Title	Description
inputCols	Input Columns	Columns on which to apply Regex
searchPatterns	Find	Enter Search Pattern
replacePatterns	Replace	Enter replacement Value

FindAndReplaceUsingRegex

This node finds and replaces text in a column containing string

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeFindAndReplaceUsingRegex

Fields

Name	Title	Description
inputCols	Input Columns	Columns on which to apply Regex
searchPattern	Find	Enter Search Pattern
replacePattern	Replace	Enter replacement Value

ImputingWithConstant

It imputes missing value with constant value. It fills missing values (None) in selected columns with given constant value for the corresponding column, in the incoming DataFrame.

Туре

transform

Class

fire.nodes.ml.NodeReplaceMissingValueWithConstant

Name	Title	Description
colNames	Columns	Columns to be processed for missing values
constants	Constants	Missing value will be replaced with constant

ImputingWithMeanValue

Imputing the continuous variables by mean.

Туре

transform

Class

fire.nodes.ml.NodeReplaceMissingValueWithMean

Fields

Name	Title	Description
inputCols	Column Names	Columns type should be continuous

RemoveDuplicateRows

This node take an array of fields, compare rows on those fields. If they full match then its a match. From the matches it would randomly take one row and drop the rest.

Input

It accepts a DataFrame as input from the previous Node

Output

The output Dataframe is the same as the input Dataframe with the duplicate rows removed

Туре

transform

Class

fire.nodes.etl.NodeRemoveDuplicateRows

Name	Title	Description
order	Order	Whether to take the first or last matching
		record when removing duplicates
inputCols	Columns	The columns to be selected for match

Dedup

This node is used for problems like entity resolution or data matching. Entity resolution or Data matching is the problem of finding and linking different mentions of the same entity in a single data source or across multiple data sources.

Input

It takes in a DataFrame as input

Output

Dataframe with confidence score field and other selected scores for entities

Туре

transform

Class

fire.nodes.ml.NodeDedup

Fields

Name	Title	Description
confidenceScore	Confidence Score	Confidence Score
lhsCols	LHS Variables	LHS columns for matching
rhsCols	RHS Variables	RHS columns for matching
matchingAlgorithms	Algorithm to use	Algorithm to use for matching
matchingWeights	Weights	Weights for matches
outputCols	Output Column	Output Column

Details

Levenstein

The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character.

How many char you change to make two strings equal.

JaroWinker

Jaro–Winkler distance for two strings is, the more similar the strings are. The Jaro–Winkler distance metric is designed and best suited for short strings such as person names. The score is normalized such that 0 equates to no similarity and 1 is an exact match.

Good for short words, typos and nikename.

Fullmatch

Fullmatch distance for two strings is, how two strings are match exactly. The score is assigned such that 1 is for exact match and 0 is for not match.

Jaccard

The Jaccard similarity measures similarity between finite sample sets, and is defined as the cardinality of the intersection of sets divided by the cardinality of the union of the sample sets. Suppose you want to find jaccard similarity between two sets A and B it is the ration of cardinality of A B and A B.

Sparkflows provide default 3-gram Jaccard similarity measures.

Longest common subsequences(LCS): LCS distance between strings s1 and s2, computed as |s1| + |s2| - 2 * |LCSfunction(s1, s2)| and distance is normalized between 0 and 1.

LCS function returns the length of Longest Common Subsequence (LCS) between strings s1 and s2.

Notional distance

Notional distance between two numbers X and Y, computed as abs(X - Y) / abs(x) + abs(Y).

Date Difference

Date Difference gives number of days between two dates(yyyy-MM-dd).

RemoveUnwantedCharacters

This node removes unwanted characters

Input

It accepts a DataFrame as input from the previous Node

Туре

transform

Class

fire.nodes.etl.NodeRemoveUnwantedCharacters

Fields

Name	Title	Description
inputCols	Input Columns	Input columns
removeWhitespaces	Remove Whites-	Removes white space
	paces	
removeLetters	Remove Letters	Removes letters
removeDigits	Remove Digits	Removes digits
removeSigns	Remove Signs	Removes signs
removeCommas	Remove Commas	Removes commas

ImputingWithModeValue

Imputing with most frequently observed value. It fills missing values (None) in selected columns with most frequently observed value in the corresponding column, in the incoming DataFrame.

Туре

transform

Class

fire.nodes.ml.NodeReplaceMissingValueWithMode

Fields

Name	Title	Description
colNames	Columns	Columns to be processed for imputing the
		missing values.

24.1.12 04-DataValidation

ValidateFieldsAdvanced

Validation Multiple Node

Туре

transform

Class

fire.nodes.etl. Node Validation Multiple

Fields

Name	Title	Description
description	Description	Validations being Performed
measureValue	Validation Success-	Condition for Validation Passing
	ful if Percent Good	
	Records >=	
inputCols	Columns	Columns
functions1	Function	Validation Function to apply
values1	Values	Values
conditions1	Condition	Validation Condition to apply
functions2	Function	Validation Function to apply
values2	Values	Values
conditions2	Condition	Validation Condition to apply
functions3	Function	Validation Function to apply
values3	Values	Values

CompareDatasets

Validate the input datasets

Туре

join

Class

fire.nodes.validation.NodeCompareDatasets

Fields

ValidateAddress

This node validate the USA address

Input

It accepts a DataFrame as input from the previous Node

Output

A new column isValidAddress is added which contains valid or inValid values

Туре

transform

Class

fire.nodes.etl.NodeValidateAddress

Fields

Name	Title	Description
inputColName	Input Column Name	input column name

ValidateFieldsSimple

Validation Node

Туре

transform

Class

fire.nodes.etl.NodeValidation

Fields

Name	Title	Description
description	Description	Validations being Performed
inputCols	Columns	Columns
functions	Function	Validation Function to apply
values	Values	Values

24.1.13 CustomProcessors

pyspark

ScoreCard_Binning

Туре

transform

Class

 $fire.nodes.etl.NodeCustomPySpark_dd281630\mbox{-}bf8f\mbox{-}4e04\mbox{-}8526\mbox{-}1cb555871c46$

Fields

24.1.14 17-Documentation

StickyNote

Allows capturing Notes on the Workflow

Туре

sticky

Class

fire.nodes.doc.NodeStickyNote

Fields

Name	Title	Description
bgColor	Bg Color	Background of note
width	Width	Width of note
height	Height	Height of note
comment	Comment	Comments for the Workflow

Notes

Allows capturing Notes on the Workflow

Туре

doc

Class

fire.nodes.doc.NodeDocLarge

Name	Title	Description
comment	Comment	Comments for the Workflow

24.1.15 12-ML-H2O

H2OWord2Vec

H2O Word2Vec

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OWord2vec

Fields

Name	Title	Description
min_word_freq	Min Word Freq	Specify an integer for the minimum word
		frequency. Word2vec will discard words
		that appear less than this number of times.
vec_size	Vec Size	Specify the size of word vectors.
window_size	Window Size	This specifies the size of the context window
		around a given word.
epochs	Epochs	Specify the number of training iterations to
		run.
init_learning_rate	Init Learning Rate	Set the starting learning rate.
sent_sample_rate	Sent Sample Rate	Set the threshold for the occurrence of
		words. Those words that appear with higher
		frequency in the training data will be ran-
		domly down-sampled. An ideal range for
		this option 0, 1e-5.
aggregateMethod	AggregateMethod	Specifies how to aggregate sequences of
		words.

Details

The Word2vec algorithm takes a text corpus as an input and produces the word vectors as output. The algorithm first creates a vocabulary from the training text data and then learns vector representations of the words.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/word2vec.html#

H2OScore

Туре

join

Class

fire.nodes.h2o.NodeH2OScore

Fields

H2OModelSave

Saves an H2O binary ML model

Туре

ml-modelsave

Class

fire.nodes.h2o.NodeH2OModelSave

Fields

Name	Title	Description
path	Path	Absolute Path for saving the H2O Mojo

H2OPCA

H2O PCA

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OPCA

Fields

Details

Principal Components Analysis (PCA) is closely related to Principal Components Regression. The algorithm is carried out on a set of possibly collinear features and performs a transformation to produce a new set of uncorrelated features.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/pca.html

H2OGLM

H2O GLM

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OGlm

Fields

Details

Generalized Linear Models (GLM) estimate regression models for outcomes following exponential distributions. In addition to the Gaussian (i.e. normal) distribution, these include Poisson, binomial, and gamma distributions. Each serves a different purpose, and depending on distribution and link function choice, can be used either for prediction or classification.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html

H2OScore

Туре

ml-predict

Class

fire.nodes.h2o.NodeH2OScore

Fields

Name	Title	Description
isTestData	isTestData	To enable the test metrics.

H2OMojoLoad

Loads an H2O Mojo ML model

Туре

ml-modelload

Class

fire.nodes.h2o.NodeH2OMojoLoad

Fields

Name	Title	Description
path	Path	Absolute Path for loading the H2O Mojo

H2OXGBoostScore

Туре

ml-predict

Class

fire.nodes.h2o.NodeH2OXGBoostScore

Name	Title	Description
isTestData	isTestData	To enable the test metrics.

H2O Model Load

This node load the H2O model.

Туре

ml-modelload

Class

fire.nodes.h2o.NodeH2OModelLoad

Fields

H2OXGBoostWithGridSearch

H2O XGBoost

Input

It takes in a DataFrame as input

Туре

join

Class

 $fire.nodes.h2o.node_h2oxgboost_gridsearch$

Fields

H2OXGBoost

H2O XGBoost

Input

It takes in a DataFrame as input

Туре

join

Class

 $fire.nodes.h2o.node_h2oxgboost$

Fields

Name	Title	Description
responseCol	Response Column	
featureCols	Feature Columns	Specify the column or columns to be in-
		cluded for feature.
ntrees	NTrees	Specify the number of trees to build
tree_method	Tree Method	Specify the construction tree method to use.
grow_policy	Grow Policy	
max_depth	Max Depth	Specify the maximum tree depth (Setting
		this value to 0 specifies no limit)
max_leaves	Max Leaves	Specify the maximum number of leaves to
		include each tree
col_sample_rate_per_tree	Col Sample Rate	
	Per Tree	
sample_rate	Sample rate	Specify the row sampling rate (x-axis).
		(Note that this method is sample without re-
		placement)
learn_rate	Learn Rate	Specify the learning rate (The range is 0.0 to
		1.0)
stopping_rounds	Stopping Rounds	
stopping_metric	Stopping Metric	Specify the construction tree method to use.
seed	Seed	

H2OXGBoost

H2O XGBoost

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OXGBoost

Details

XGBoost is a supervised learning algorithm that implements a process called boosting to yield accurate models. Boosting refers to the ensemble learning technique of building many models sequentially, with each new model attempting to correct for the deficiencies in the previous model.

More details are available at : https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/docs-website/h2o-docs/ data-science/xgboost.html

H2O Model Save

This node saves the H2O model at the specified path.

Input

It takes in a Model and DataFrame as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-modelsave

Class

fire.nodes.h2o.NodeH2OModelSave

Fields

H2ONeuralNetwork

H2O Deep Learning/Neural Network

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2ONeuralNetwork

Fields

Details

H2O's Deep Learning is based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons with tanh, rectifier, and maxout activation functions.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html

H2ONaiveBayes

H2O Naive Bayes

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2ONaiveBayes

Fields

Details

Naïve Bayes is a classification algorithm that relies on strong assumptions of the independence of covariates in applying Bayes Theorem. The Naïve Bayes classifier assumes independence between predictor variables conditional on the response, and a Gaussian distribution of numeric predictors with mean and standard deviation computed from the training dataset.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/naive-bayes.html

H2OGLRM

H2O GLRM

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OGlrm

Fields

Details

Generalized Low Rank Models (GLRM) is an algorithm for dimensionality reduction of a dataset. It is a general, parallelized optimization algorithm that applies to a variety of loss and regularization functions. Categorical columns are handled by expansion into 0/1 indicator columns for each level. With this approach, GLRM is useful for reconstructing missing values and identifying important features in heterogeneous data.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glrm.html

H2OGBM

H2O GBM

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OGbm

Fields

Details

Gradient Boosting Machine (for Regression and Classification) is a forward learning ensemble method. The guiding heuristic is that good predictive results can be obtained through increasingly refined approximations. H2O's GBM

sequentially builds regression trees on all the features of the dataset in a fully distributed way - each tree is built in parallel.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html

H2OKMeans

H2O KMeans

Input

It takes in a DataFrame as input

Туре

ml-estimator

Class

fire.nodes.h2o.NodeH2OKMeans

Fields

Details

K-Means falls in the general category of clustering algorithms. Clustering is a form of unsupervised learning that tries to find structures in the data without using any labels or target values. Clustering partitions a set of observations into separate groupings such that an observation in a given group is more similar to another observation in the same group than to another observation in a different group.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/k-means.html

H2OIsolationForest

Isolation Forest is similar in principle to Random Forest and is built on the basis of decision trees.

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2OI solationForest

Details

Isolation Forest is similar in principle to Random Forest and is built on the basis of decision trees. Isolation Forest, however, identifies anomalies or outliers rather than profiling normal data points. Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. This split depends on how long it takes to separate the points.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/if.html

H2ODRF

H2O DRF

Input

It takes in a DataFrame as input

Туре

transform

Class

fire.nodes.h2o.NodeH2ODrf

Fields

Details

Distributed Random Forest (DRF) is a powerful classification and regression tool. When given a set of data, DRF generates a forest of classification or regression trees, rather than a single classification or regression tree. Each of these trees is a weak learner built on a subset of rows and columns. More trees will reduce the variance. Both classification and regression take the average prediction over all of their trees to make a final prediction, whether predicting for a class or numeric value.

More details are available at : http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html

H2OMojoSave

Saves an H2O MOJO ML model

Туре

ml-modelsave

Class

fire.nodes.h2o.NodeH2OMojoSave

Fields

Name	Title	Description
path	Path	Path for saving the H2O Mojo

H2OModelLoad

Loads an H2O binary ML model

Туре

ml-modelload

Class

fire.nodes.h2o.NodeH2OModelLoad

Fields

Name	Title	Description
path	Path	Path for loading the H2O Mojo

24.1.16 13-ML-AWSSagemaker

KMeansSageMakerEstimator

Туре

ml-estimator

Class

fire.nodes.sagemaker.NodeKMeansSageMakerEstimator

Name	Title	Description
roleArn	Role Arn	Role arn to use sagemaker
trainingInstanceType	Training Instance	InstanceType for training
	Туре	
trainingInstanceCount	Training Instance	Number of Instance for training
	Count	
endpointInstanceType	Endpoint Instance	InstanceType for Endpoint
	Туре	
endpointInitialInstanceCount	Endpoint Initial In-	Number of Instance for Endpoint
	stance Count	
k	К	The number of clusters to create.
featureDim	Feature Dim	The number of dimensions in dataset

XGBoostSageMakerEstimator

Туре

ml-estimator

Class

fire.nodes.sagemaker.NodeXGBoostSageMakerEstimator

Name	Title	Description
roleArn	Role Arn	Role arn to use sagemaker
trainingInstanceType	Training Instance	InstanceType for training
	Туре	
trainingInstanceCount	Training Instance	Number of Instance for training
	Count	
endpointInstanceType	Endpoint Instance	InstanceType for Endpoint
	Туре	
endpointInitialInstanceCount	Endpoint Initial In-	Number of Instance for Endpoint
	stance Count	
booster	Booster	Select the type of model to run at each iter-
		ation. It has 2 options: gbtree: tree-based
		models & gblinear: linear models
silent	Silent	Silent mode is activated is set to 1, i.e. no
		running messages will be printed
nthread	NThread	If you wish to run on all cores, value should
		not be entered and algorithm will detect au-
		tomatically
objective	Objective	This defines the loss function to be mini-
		mized
numTrees	Num Trees	The number of rounds for boosting
numClasses	Num Classes	For Objective: multi:softmax, you also need
		to set an additional num_class (number of
		classes) parameter defining the number of
		unique classes
seed	Seed	Can be used for generating reproducible re-
		sults and also for parameter tuning

PCASageMakerEstimator

Туре

ml-estimator

Class

fire.nodes.sagemaker.NodePCAS ageMakerEstimator

Name	Title	Description
roleArn	Role Arn	Role arn to use sagemaker
trainingInstanceType	Training Instance	InstanceType for training
	Туре	
trainingInstanceCount	Training Instance	Number of Instance for training
	Count	
endpointInstanceType	Endpoint Instance	InstanceType for Endpoint
	Туре	
endpointInitialInstanceCount	Endpoint Initial In-	Number of Instance for Endpoint
	stance Count	
numComponents	Num Components	The number of principal components to
		find.
featureDim	Feature Dim	The number of dimensions in dataset

SageMakerLinearLearnerBinaryClassifier

Туре

ml-estimator

Class

fire.nodes.sage maker.Node Linear Learner Binary Classifier

Fields

Name	Title	Description
roleArn	Role Arn	Role arn to use sagemaker
trainingInstanceType	Training Instance	InstanceType for training
	Туре	
trainingInstanceCount	Training Instance	Number of Instance for training
	Count	
endpointInstanceType	Endpoint Instance	InstanceType for Endpoint
	Туре	
endpointInitialInstanceCount	Endpoint Initial In-	Number of Instance for Endpoint
	stance Count	

SageMakerLinearLearnerRegressor

Туре

ml-estimator

Class

fire.nodes.sagemaker.NodeLinearLearnerRegressor

Fields

Name	Title	Description
roleArn	Role Arn	Role arn to use sagemaker
trainingInstanceType	Training Instance	InstanceType for training
	Туре	
trainingInstanceCount	Training Instance	Number of Instance for training
	Count	
endpointInstanceType	Endpoint Instance-	InstanceType for Endpoint
	Туре	
endpointInitialInstanceCount	Endpoint Initial In-	Number of Instance for Endpoint
	stance Count	

SaveSageMakerFormat

Saves the DataFrame into the specified location in Sagemaker Format

Туре

transform

Class

fire.nodes.sagemaker.NodeSaveSagemaker

Fields

Name	Title		Description
path	Path		Path where to save the Sagemaker files
saveMode	Save Mode		Whether to Append, Overwrite or Error if
			the path Exists
labelColumnName	Label	Column	label column name
	Name		
featuresColumnName	Features	Column	features column name
	Name		

24.1.17 14-ML-Sklearn

SklearnPredict

Predict node takes in a DataFrame and Model and makes predictions

Input

It takes in a DataFrame and Model as input

Output

A new column containing the predictions is added to the input DataFrame

Туре

ml-predict

Class

fire.nodes.sklearn.NodeSklearnPredict

Fields

SklearnRegressionEvaluator

Evaluator for regression, which expects two input columns: prediction and label.

Input

It takes in a DataFrame as input

Output

The incoming DataFrame is passed to the output

Туре

transform

Class

fire.nodes.sklearn.NodeSklearnRegressionEvaluator

Fields

Name	Title	Description
targetCol	Label Column	The label column for model fitting.
predictCol	Prediction Column	The prediction column.

Sklearn Model Load

This node load the Sklearn model stored in the pickel file.

Туре

ml-modelload

Class

fire.nodes.sklearn.NodeModelLoad

Fields

CustomMetrics

Туре

transformer

Class

fire.nodes.sklearn.NodeCustomMetrics

Fields

Name	Title	Description
actualCol	ActualCol	
predictedCol	PredictedCol	
aggregatedAt	AggregatedAt	
metricsType	metricsType	Window Function Name

SkLearnRidgeRegression

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnRidgeRegression

Name	Title	Description
targetCol	Target Column	The label column for model fitting
alpha	Alpha	
fitintercept	Fitintercept	
normalize	Normalize	
maxiter	Maxiter	
tol	Tolerence	
solver	Solver	
randomstate	randomstate	Random state

SklearnRandomForestClassifier

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnRandomForestClassifier

Fields

Name	Title	Description
targetCol	Target Column	The label column for model fitting
n_estimators	NEstimators	
criterion	Criterion	
max_depth	MaxDepth	Default value is None i.e -1
min_samples_split	MinSamplesSplit	
min_samples_leaf	MinSamplesLeaf	
min_weight_fraction_leaf	MinWeightFractionLeaf	
max_features	MaxFeatures	
max_leaf_nodes	MaxLeafNodes	Default value is None i.e -1
min_impurity_decrease	MinImpurityDecrease	
min_impurity_split	MinImpuritySplit	
bootstrap	Bootstrap	
oob_score	OobScore	
random_state	RandomState	Default value is None i.e -1
warm_start	WarmStart	

SklearnRandomForestRegression

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnRandomForestRegression

Fields

Name	Title	Description
targetCol	Target Column	The label column for model fitting
n_estimators	NEstimators	
criterion	Criterion	
max_depth	MaxDepth	Default value is None i.e -1
min_samples_split	MinSamplesSplit	
min_samples_leaf	MinSamplesLeaf	
min_weight_fraction_leaf	MinWeightFractionLeaf	
max_features	MaxFeatures	
max_leaf_nodes	MaxLeafNodes	Default value is None i.e -1
min_impurity_decrease	MinImpurityDecrease	
min_impurity_split	MinImpuritySplit	
bootstrap	Bootstrap	
oob_score	OobScore	
random_state	RandomState	Default value is None i.e -1
warm_start	WarmStart	

SklearnGradientBoostingRegression

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnGradientBoostingRegressor

Name	Title	Description
targetCol	Target Column	The label column for model fitting
loss	Loss	
learning_rate	LearningRate	
n_estimators	NEstimators	
subsample	Subsample	
criterion	Criterion	
min_samples_split	MinSamplesSplit	
min_samples_leaf	MinSamplesLeaf	
min_weight_fraction_leaf	MinWeightFractionLeaf	
max_depth	MaxDepth	Default value is None i.e -1
min_impurity_decrease	MinImpurityDecrease	
min_impurity_split	MinImpuritySplit	
random_state	RandomState	Default value is None i.e -1
max_features	MaxFeatures	
alpha	Alpha	
verbose	Verbose	
max_leaf_nodes	MaxLeafNodes	Default value is None i.e -1
warm_start	WarmStart	
presort	Presort	
validation_fraction	ValidationFraction	
n_iter_no_change	NIterNoChange	Default value is None i.e -1
tol	Tol	

SklearnGradientBoostingClassifier

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnGradientBoostingClassifier

Name	Title	Description
targetCol	Target Column	The label column for model fitting
loss	Loss	
learning_rate	LearningRate	
n_estimators	NEstimators	
subsample	Subsample	
criterion	Criterion	
min_samples_split	MinSamplesSplit	
min_samples_leaf	MinSamplesLeaf	
min_weight_fraction_leaf	MinWeightFractionLeaf	
max_depth	MaxDepth	
min_impurity_decrease	MinImpurityDecrease	
min_impurity_split	MinImpuritySplit	
random_state	RandomState	Default value is None i.e -1
max_features	MaxFeatures	
verbose	Verbose	
max_leaf_nodes	MaxLeafNodes	Default value is None i.e -1
warm_start	WarmStart	
presort	Presort	
validation_fraction	ValidationFraction	
n_iter_no_change	NIterNoChange	Default value is None i.e -1
tol	Tol	

SkLearnLassoRegression

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnLassoRegression

Name	Title	Description
targetCol	Target Column	The label column for model fitting
alpha	Alpha	
fit_intercept	Fitintercept	
normalize	Normalize	
precompute	Precompute	
max_iter	Maxiter	
tol	Tol	
warm_start	WarmStart	
positive	Positive	
random_state	RandomState	Default value is None i.e -1
selection	Selection	

SklearnLogisticRegression

Туре

ml-estimator

Class

fire.nodes.sklearn.NodeSklearnLogisticRegression

Fields

Name	Title	Description
targetCol	Target Column	The label column for model fitting
penalty	Penalty	
dual	Dual	
tol	Tol	
С	С	
fit_intercept	Fitintercept	
intercept_scaling	InterceptScaling	
class_weight	ClassWeight	
random_state	RandomState	
solver	Solver	
max_iter	Maxiter	
multi_class	MultiClass	
verbose	Verbose	
warm_start	WarmStart	
11_ratio	L1Ratio	

Sklearn Model Save

This node saves the Sklearn model generated at the specified path in pickle file.

Input

It takes in a Model and DataFrame as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-modelsave

Class

fire.nodes.sklearn.NodeModelSave

Fields

Sklearn Model Load From S3

This node load the Sklearn model stored in the pickel format in S3.

Input

It takes in a Model and DataFrame as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-modelsave

Class

fire.nodes.sklearn.NodeSklearnModelLoadFromS3

Fields

SklearnClassificationEvaluator

Evaluator for classification, which expects two input columns: prediction and label.

Input

It takes in a DataFrame as input

Output

The incoming DataFrame is passed to the output

Туре

transform

Class

fire.nodes.sklearn.NodeSklearnClassificationEvaluator

Fields

Name	Title	Description
targetCol	Label Column	The label column for model fitting.
predictCol	Prediction Column	The prediction column.

Sklearn Model Save To S3

This node saves the Sklearn model generated at the specified path in S3 in pickle format.

Input

It takes in a Model and DataFrame as input.

Output

The incoming dataframe is passed to the output.

Туре

ml-modelsave

Class

fire.nodes.sklearn.NodeSklearnModelSaveToS3

CategoryEncoders

Туре

ml-transformer

Class

fire.nodes.sklearn.NodeCategoryEncoders

Fields

Name	Title		Description
category_features_column	Category Column	Features	

24.1.18 08-Group

GroupBy

Grouper Node

Туре

transform

Class

fire.nodes.etl.NodeGroupBy

Fields

Cube

Cube Node generates a result set that shows aggregates for all combinations of values in the selected columns.

Туре

transform

fire.nodes.etl.NodeCube

Fields

Name	Title	Description
cubeCols	Cube Columns	
aggregateCols	Aggregate Columns	Aggregate Columns
aggregateOperations	Aggregate Opera-	Aggregate Operation
	tion to use	

Rollup

Rollup Node generates a result set that shows aggregates for a hierarchy of values in the selected columns.

Туре

transform

Class

fire.nodes.etl.NodeRollup

Fields

Name	Title	Description
rollupCols	Rollup Columns	
aggregateCols	Aggregate Columns	Aggregate Columns
aggregateOperations	Aggregate Opera- tion to use	Aggregate Operation

PivotBy

Pivot Node

Туре

transform

Class

fire.nodes.etl.NodePivotBy

24.1.19 06-Code

SQLExecuter

This node runs the given SQL query

Input

This type of node takes the sql query of any statement type

Output

This node execute the given SQL query

Туре

dataset

Class

fire.nodes.runrdbmssql.NodeSqlExecuter

Fields

Name	Title	Description
url	Db Url	Url of SQL
driver	driver class name	driver class name for SQL
user name	User Name	User name of SQL
password	password	password of SQL
statementType	Statement Type	statementType of SQL
query	query	write query to wxecute

PipePython2

This node runs any given Python code. It pipes the incoming DataFrame through pipe to the Python Script. Output back to Spark has to be written out using print.

Input

It pipes the incoming DataFrame through pipe to the Python Script. It also passes the Schema of the DataFrame to the Python script through the command line argument - argv[1]

Output

Output back to Spark has to be written out using print.

Туре

transform

Class

fire.nodes.etl.NodePipePython 2

Fields

Name	Title	Description
codeHeader	Pipe Header Code	Header part of the Python code to be run. It
		receives each record as a string
codeBody	Pipe Body Code	Body part of the Python code to be run.
codeFooter	Pipe Footer Code	Footer part of the Python code to be run. It
		should write out each resulting record back
		as a string.
outputColNames	Output Column	Output Schema of Pipe Python Processor
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	

ScalaUDF

This node runs any given Scala code for UDFs

Input

Туре

·

scala

Class

fire.nodes.etl. Node UDFS cala

Name	Title	Description
code	Scala	Scala code to be run.

Jython

This node runs any given Jython code. The input dataframe is passed in the variable inDF. The output dataframe should be placed in the variable outDF

Input

The input dataframe is passed in the variable in DF

Output

The output dataframe should be placed in the variable outDF

Туре

transform

Class

fire.nodes.etl.NodeJython

Fields

Details

This node runs any given Jython code.

Below is an example jython code. It takes the input dataframe 'inDF', and returns the new dataframe 'outDF' outDF = inDF.groupBy("c2").count()

UnixShellCommands

This node execute shell command

Туре

shellcommand

fire.nodes.etl.NodeShellCommand

Fields

Name	Title	Description
shellCommand	shell Command	Unix Shell Command

SQL

This node runs the given SQL on the incoming DataFrame

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node runs the given SQL on the incoming DataFrame to generate the output DataFrame

Туре

transform

Class

fire.nodes.etl.NodeSQL

Fields

Scala

This node runs any given Scala code. The input dataframe is passed in the variable inDF. The output dataframe is passed back by registering it as a temporary table.

Input

The input dataframe is passed in the variable inDF.

Output

The output dataframe is passed back by registering it as a temporary table

Туре

scala

Class

fire.nodes.etl. NodeScala

Fields

PipePython

This node runs any given Python code. It pipes the incoming DataFrame through pipe to the Python Script. Output back to Spark has to be written out using print.

Input

It pipes the incoming DataFrame through pipe to the Python Script. It also passes the Schema of the DataFrame to the Python script through the command line argument - argv[1]

Output

Output back to Spark has to be written out using print.

Туре

transform

Class

fire.nodes.etl.NodePipePython

Fields

Name	Title	Description
code	Pipe Python	Python code to be run. It receives each
		record as a string and outputs records back
		as a string.
outputColNames	Output Column	Output Schema of Pipe Python Processor
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	

PySpark

This node runs any given PySpark code. The input dataframe is passed in the variable inDF. The output dataframe is passed back by registering it as a temporary table.

Input

The input dataframe is passed in the variable inDF.

Output

The output dataframe is passed back by registering it as a temporary table

Туре

pyspark

Class

fire.nodes.etl.NodePySpark

Fields

RunHIVEQL

This node runs the given SQL on the incoming DataFrame

Input

This type of node takes in a DataFrame and transforms it to another DataFrame

Output

This node runs the given SQL on the incoming DataFrame to generate the output DataFrame

Туре

transform

Class

fire.nodes.etl.NodeRunHiveQL

Name	Title	Description
hql	HiveQL - HIVE	HiveQL
	Query Language	

24.1.20 10-Visualization

GraphRegionGeo

This node displays values on a Map

Туре

transform

Class

fire.nodes.graph.NodeGraphRegionGeo

Fields

PrintNRows

Prints the specified number of records in the DataFrame. It is useful for seeing intermediate output

Туре

transform

Class

fire.nodes.util.NodePrintFirstNRows

Fields

GraphValues

Туре

transform

fire.nodes.graph.NodeGraphValues

Fields

GraphGroupByColumn

Groups the data by the given column and plots the number of records in each group

Туре

transform

Class

fire.nodes.graph.NodeGraphGroupByColumn

Fields

Sample PrintNRows

Prints the specified number of records in the DataFrame. It is useful for seeing intermediate output

Туре

transform

Class

fire.nodes.util.NodeSamplePrintFirstNRows

Fields

24.1.21 19-Deprecated

StringToDate

This node converts a string column to date using the given date/time format

Туре

transform

fire.nodes.etl.NodeStringToDate

Fields

Name	Title	Description
inputColName	Input Column Name	Input Column Name
inputColFormat	Input Column For-	Input Column Format. eg: yyyy-MM-dd
	mat	yyyy-MM-dd HH:mm:ss
outputColName	Output Column	Output Column Name
	Name	
outputColType	Output Column	Output Column Type
	Туре	

Examples

Format Examples

dd-MM-yy : 31-01-12 dd-MM-yyyy : 31-01-2012 MM-dd-yyyy : 01-31-2012 yyyy-MM-dd : 2012-01-31 yyyy-MM-dd HH:mm:ss : 2012-01-31 23:59:59 yyyy-MM-dd HH:mm:ss.SSS : 2012-01-31 23:59:59.999 yyyy-MM-dd HH:mm:ss.SSSZ : 2012-01-31 23:59:59.999+0100 EEEEE MMMMM yyyy HH:mm:ss.SSSZ : Saturday November 2012 10:45:42.720+0100

OUTPUT COLUMN NAME: - If user inputs an existing column name, it overrides the column otherwise it will add a new column.

24.1.22 15-Streaming

StreamingSocketTextStream

Reads in streaming text from a socket

Input

It does not take any DataFrame as input

Output

It creates DataFrame from reading data from a socket. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

fire.nodes.streaming.NodeStreamingSocketTextStream

Fields

Name	Title	Description
batchDuration	Batch Duration in	Batch Duration in Seconds
	Seconds	
hostname	Hostname	Host to connect to for listening
port	Port	Port to connect to

Details

This Processor reads in messages from a Socket

Key Fields

Below are the key fields of this Processor.

- hostname: this is the name of the host from where to read in the messages
- port: this is the port number from where to read in the messages

Examples

Below is an example of the fields:

- hostname: localhost
- port: 8099

StreamingKafka

Reads in streaming text from topics in Apache Kafka

Input

It does not take any DataFrame as input

Output

It reads events from Kafka and creates DataFrame from the resulting rows. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.streaming.NodeStreamingKafka

Fields

Name	Title	Description
batchDuration	Batch Duration in	Batch Duration in Seconds
	Seconds	
brokers	Kafka Brokers	Kafka Brokers
group	Consumer Group	Consumer Group
topics	Kafka Topics	List of Topics separated by , (comma)
autoOffsetReset	auto.offset.reset	Auto Offset Reset
enableAutoCommit	enable.auto.commit	Enable Auto Commit
kafkaParamsKeys	Params Key/Value	More Config Values
	Pairs	
kafkaParamsValues	Parms Key/Value	More Config Values
	Pairs	

StreamingTextFileStream

It monitors a specified directory for new files. It keeps reading in any new files created in the directory.

Input

It does not take any DataFrame as input

Output

It reads the new files and creates DataFrame from the content of the text files. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.streaming.NodeStreamingTextFileStream

Name	Title	Description
path	Path	Directory from where to pick up files from
batchDuration	Batch Duration in	Batch Duration in Seconds
	Seconds	
outputCol	Output Column	Output Column

24.1.23 15-StructuredStreaming

StructuredStreamingCSV

It monitors a specified directory for new files. It keeps reading in any new files created in the directory.

Input

It does not take any DataFrame as input

Output

It reads the new files and creates DataFrame from the content of the text files. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingCSV

Fields

Name	Title	Description
path	Path	Path of the Text file/directory
separator	Separator	CSV Separator
outputColNames	Column Names for	Output Column Names
	the CSV	
outputColTypes	Column Types for	Output Column Types
	the CSV	
outputColFormats	Column Formats for	Output Column Formats
	the CSV	

StructuredStreamingHiveSink2

Saves the streaming data into an Apache HIVE Table

Туре

transform

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingHiveSink2

Fields

Name	Title	Description
databaseName	HIVE Database	Name of the HIVE Database
tableName	HIVE Table	Name of the HIVE table

StructuredStreamingFileSink

It writes the DataFrame to files with Structured Streaming

Input

It takes in DataFrame as input

Output

It writes the incoming DataFrame to files.

Туре

transform

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingFileSink

Name	Title	Description
path	Path	Path where to write the files
outputMode	Output Mode	Output Mode for saving to Files
checkpointLocation	Checkpoint Loca-	Checkpoint Location on HDFS compatible
	tion	file system for Streaming
format	Format	File Format
partitionBy	Partition By	Partition By Columns separated by space
	Columns	(can be empty in which case partitionBy
		would not be applied)

StructuredStreamingSocket

Reads in streaming text from a socket

Input

It does not take any DataFrame as input

Output

It reads events a socket and creates DataFrame from the resulting rows. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingSocket

Fields

Name	Title	Description
host	Hostname	Host to connect to for listening
port	Port	Port to connect to

StructuredStreamingHiveSink

Saves the streaming data into a HIVE Table

Туре

transform

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingHiveSink

Fields

Name	Title	Description
databaseName	HIVE Database	Name of the HIVE Database
tableName	HIVE Table	Name of the HIVE table

StructuredStreamingKinesis

Reads in streaming text from Kinesis stream

Input

It does not take any DataFrame as input

Output

It reads events from Kinesis and creates DataFrame from the resulting rows. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.structureds treaming.NodeStructuredStreamingKines is

Fields

Name	Title	Description
streamName	Stream Name	Kinesis Stream Name
endpointUrl	Endpoint Url	Kinesis Endpoint Url
editorData	Editor Data	Data to be used for testing in the Workflow
		Editor

StructuredStreamingKafka

Reads in streaming text from topics in Apache Kafka

Input

It does not take any DataFrame as input

Output

It reads events from Kafka and creates DataFrame from the resulting rows. This DataFrame is passed to the output Nodes.

Туре

sparkstreaming

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingKafka

Fields

Name	Title	Description
batchDuration	Batch Duration in	Batch Duration in Seconds
	Seconds	
brokers	Kafka Brokers	Kafka Brokers
group	Consumer Group	Consumer Group
topics	Kafka Topics	List of Topics separated by , (comma)
autoOffsetReset	auto.offset.reset	Auto Offset Reset
enableAutoCommit	enable.auto.commit	Enable Auto Commit
kafkaParamsKeys	Params Key/Value	More Config Values
	Pairs	
kafkaParamsValues	Parms Key/Value	More Config Values
	Pairs	

StructuredStreamingConsoleSink

It output the DataFrame to the console

Input

It takes in DataFrame as input

Output

It writes the incoming DataFrame to the console.

Туре

transform

Class

fire.nodes.structuredstreaming.NodeStructuredStreamingConsoleSink

Fields

Name	Title	Description
outputMode	Output Mode	Output Mode for saving to Files

24.1.24 14-DL

KerasModelFit

Туре

ml-estimator

Class

fire.nodes.dl.NodeModelFit

Name	Title	Description
targetCol	Target Column	The label column for model fitting
batch_size	BatchSize	Default value is None i.e -1
epochs	Epochs	
verbose	Verbose	
callbacks	Callbacks	Default value is None i.e -1
validation_split	ValidationSplit	
validation_data	ValidationData	Default value is None i.e -1
shuffle	Shuffle	
class_weight	ClassWeight	Default value is None i.e -1
sample_weight	SampleWeight	Default value is None i.e -1
initial_epoch	InitialEpoch	
steps_per_epoch	StepsPerEpoch	Default value is None i.e -1
validation_steps	ValidationSteps	Default value is None i.e -1
validation_freq	ValidationFreq	
max_queue_size	MaxQueueSize	
workers	Workers	
use_multiprocessing	UseMultiprocessing	

KerasPredict

Туре

ml-predict

Class

fire.nodes.dl.NodePredict

Fields

Name	Title	Description
targetCol	Target Column	The label column for model fitting
batch_size	BatchSize	Default value is None i.e -1
verbose	Verbose	
steps	Steps	Default value is None i.e -1
callbacks	Callbacks	Default value is None i.e -1
max_queue_size	ValidationFreq	
workers	Workers	
use_multiprocessing	UseMultiprocessing	

KerasModelCompile

Туре

transform

Class

fire.nodes.dl.NodeModelCompile

Fields

Name	Title	Description
optimizer	Optimizer	
loss	Loss	
metrics	Metrics	
loss_weights	LossWeights	
sample_weight_mode	SampleWeightMode	
weighted_metrics	WeightedMetrics	
target_tensors	TargetTensors	

DenseLayer

Туре

transform

Class

fire.nodes.dl.NodeDense

Fields

Name	Title	Description
units	Units	
activation	Activation	
use_bias	Use Bias	
kernel_initializer	Kernel Initializer	
bias_initializer	Bias Initializer	
kernel_regularizer	Kernel Regularizer	
bias_regularizer	Bias Regularizer	
activity_regularizer	Activity Regularizer	
kernel_constraint	Kernel Constraint	
bias_constraint	Bias Constraint	

KerasModelSequential

Туре

transform

Class

fire.nodes.dl.NodeModelSequential

Fields

Name	Title	Description
layers	Layers	

24.1.25 07-JoinUnion

UnionAll

This node creates a new DataFrame by merging all the rows without removing the duplicates

Input

It accepts a DataFrame as input from the previous Node

Output

This node does union of all the rows without removing the duplicates

Туре

join

Class

fire.nodes.etl.NodeUnionAll

Fields

GeoJoin

This node joins the incoming dataframes

Input

This node takes in 2 DataFrames as input and produces one DataFrame as output

Туре

join

Class

fire.nodes.etl.NodeGeoJoin

Fields

Name	Title	Description
latitudeCol	Latitude Column	Latitude Column from first DataFrame
longitudeCol	Longitude Column	Longitude Column from first DataFrame
polygonCol	Polygon Column	Polygon Column from second DataFrame

JoinOnCommonColumns

This node joins the incoming dataframes on 1 or more columns

Input

It takes in 2 DataFrames as input and produces one DataFrame as output by joining on the specified columns

Output

The output DataFrame produced as a result of joining the incoming DataFrames on the specified columns

Туре

join

Class

fire.nodes.etl.NodeJoinUsingColumns

Name	Title	Description
joinCols	Common Join	Space separated list of columns on which to
	Columns	join
joinType	Join Type	Type of Join
outputColNames	Output Column	Name of the Output Columns
	Names	
outputColTypes	Output Column	Data Type of the Output Columns
	Types	
outputColFormats	Output Column For-	Format of the Output Columns
	mats	
whereClause	Where Clause	where condition after join function

JoinOnColumns

Туре

join2inputs

Class

fire.nodes.etl.JoinOnColumns

Fields

Name	Title	Description
joinType	Join Type	Type of Join
leftTableJoinColumn	LeftTableJoinColu	mn
rightTableJoinColumn	RightTableJoinCol	lumn

JoinUsingSQL

This node registers the incoming DataFrames as temporary tables and executes the SQL provided

Input

It takes in 2 DataFrames as input and produces one DataFrame as output by executing the provided SQL.

Output

The DataFrame created as a result of executing the join SQL

Туре

join

Class

fire.nodes.etl.NodeJoinUsingSQL

Fields

UnionDistinct

This node creates a new DataFrame by performing a DISTINCT on the result set, eliminating any duplicate rows

Input

It takes in multiple DataFrames as input

Output

This node does union of all the rows from the incoming DataFrames to generate the output DataFrame

Туре

join

Class

fire.nodes.etl.NodeUnionDistinct

Fields

JoinOnCommonColumn

This node joins the incoming dataframes on a joinCol

Input

This node takes in 2 DataFrames as input and produces one DataFrame as output

Output

The output DataFrame is the result of joining the 2 incoming DataFrames on the join column

Туре

join

Class

fire.nodes.etl.NodeJoinUsingColumn

Fields

Name	Title	Description
joinCol	Common Join Col-	column on which to join
	umn	

CHAPTER 25

Release Notes

25.1 Release Notes

25.1.1 Upcoming Features

Below are the upcoming features in Fire Insights.

Installer

A one-click installer and update for Fire Insights. Users would be able to install and update Fire Insights on their laptops with one click.

25.1.2 Aug 2020

New Features

- Time Series Modeling with Prophet
- Time Series Modeling with Arima
- Building Custom Nodes in Python

UI Improvements

• Upgraded look and feel

25.1.3 May 2020

New Features

- · Added viewing of Fire Insights logs under Administration Menu
- Added more details to Data Profiling
- · Added file upload and delete capabilities in DBFS browser
- Added ability to create datasets for data on AWS S3.
- · Added configurations for AWS Home Directory to restrict access of other bucket or folder
- · Added interactive dashboards
- · Added ability to view workflows by type : Normal, Data Profiling, Dataset Cleaning

UI Improvements

• Each workflow list page now displays up to 50 workflows

25.1.4 April 2020

New Features

- Added browsing of AWS S3 file system under Data Browser
- Added uploading files to S3
- Added creating folder on S3
- Added deleting files on S3
- Added reporting for Total Users, Groups, Projects, Workflows & Workflows Executions

UI Improvements

· Autocomplete feature added to SQL editor in workflows

25.1.5 March 2020

New Features

- Integration with Databricks
- Added Browsing Databricks DB, Databricks Cluster & DBFS
- Added Scheduling in Standalone Mode
- Compatible with Amazon Aurora Database

UI Improvements

- Improvement of Metrics which include stage information
- Improvement in JOIN USING SQL Processors

25.1.6 February 2020

New Features

- Job Metrics Integration with improvements.
- SUPERUSER to have more rights when elevated access is enabled.
- If user is inactive, he is unable to login.
- Added Runtime Statistics.
- Added Compare Model.

UI Improvment

• Improvement to Connection page.

25.1.7 January 2020

New Features

- Integrated with yarn which enable us to see detail information of job submitted to cluster
- Integrated with Job Metrics
- Added plugins for GoogleRestApiKey in Configurations
- Added Geo chart: Country & Geo chart: Lat, Lon features in Interactive Dahboard
- Integrated with Model List and Summary Page for viewing detail information about the model
- Added Reload Sample Application Features

25.1.8 September 2019

New Processors Added For Scala Engine

- MultiWindowAnalytics
- MultiWindowRanking

New Processors Added For Pyspark Engine

- SaveAvro
- SaveJSON

Improvement of RESTAPI

New Features

- Integrated File Watcher with AWS
- Database Cleanup for workflow execution & workflow execution results

- Export of all users implemented
- Added search help with search option to Quickstart Guide, Tutorials & FAQ

Upgrades for Security Vulnerabilties

• All the dependencies have been upgraded to handle security vulnerabilities.

UI Improvement

• Improvement of WorkflowEeditor Page to make it easy to add the workflow parameters.

25.1.9 August 2019

New Processors Added For Scala Engine

- WindowAnalytics
- WindowRanking
- H2OGLRM
- H2OWord2Vec

New Processors Added For Pyspark Engine

- ZipWithIndex
- ReadAvro
- ReadJSON
- ReadParquet

UI Improvements

- Drag and drop function for node in workflow editor
- Improvement of workflow editor page look & feel.

25.1.10 July 2019

Integration of H2O

- The following New H2O Processors have been added :
 - H2ODRF
 - H2OGBM
 - H2OGLM
 - H2OIsolationForest
 - H2OKMeans

- H2OModelLoad
- H2OModelSave
- H2OMojoLoad
- H2OMojoSave
- H2ONaiveBayes
- H2ONeuralNetwork
- H2OPCA
- H2OScore

Improvements in UI

- Login Page of Fire Insights has been upgraded.
- Scatter Plot look and feel has been upgraded.

Improvements to HDFS Browser

• Ability to edit files and directories.

Improvements in Home Dashboard Page

Added New Features

- Added Search Box to Search Workflow, Node, Dataset & Dashboard available in an application.
- Added Self-Registration to create a user directly from Login page.

Upgradation of Running Server on Ports

• Fire Insights now enable us to run Fire & Pspark server on different ports.

25.1.11 June 2019

The following features have been released in June 2019.

Improvements in UI

- Displaying text in Workflow Execution Page with more details visible.
- CSV and other read file nodes, now display the name of the file.
- When cloning a node in the editor, the cloned node is created close to the original node.

Improvements to HDFS Browser

• Fire Insights now allows moving multiple files from one directory to another.

Support Of Authentication Using Token

• Fire now supports two methods Of getting tokens to access Fire

Grant Types – Password.

Grant Types - Authorization code.

Improvements in Dataset

• Look and feel of the edit Dataset page has been upgraded.

Running Applications Locally

• Workflows when running locally are now executed as separate Java or Python processes.

Node Updates

• JoinUsingSQL now allows joining multiple datasets at a time.

25.1.12 May 2019

The following features have been released in May 2019.

PySpark Engine

• New Engine for running PySpark

New Processors

Outlier Detection

• New Node for Outlier Detection

Improvements to HDFS Browser

- · Displaying user permission for each file/directory
- · Displaying an icon indicating whether it is file or directory
- · Better display of error messages

Applications

• Datasets tab is the first tab now

Datasets

- Better display of the Create/Edit dataset page
- Do not display JDBC passwords

Workflow Editor

- Ability to create DataSet Nodes by browsing the list of datasets
- HIVE DB Browser on the LHS
- Better display of the processors
- Fix for tabs in dialogs not showing up (eg. in Logistic Regression Processor)

25.1.13 April 2019

New Processors Added For Scala Engine

- MultiWindowAnalytics
- MultiWindowRanking

New Processors Added For Pyspark Engine

- SaveAvro
- SaveJSON

Improvement of RESTAPI

New Features

- Integrated File Watcher with AWS
- Database Cleanup for workflow execution & workflow execution results
- · Export of all users
- Added search help with search option to quickstart guide, tutorials & FAQ

Upgrades for Security Vulnerabilties

• All the dependencies have been upgraded to handle security vulnerabilities.

UI Improvement

• Improvement of workflow editor page to make it easy to add the workflow parameters.

25.1.14 February 2019

The following features have been released in Feb 2019.

Correlation Node Output

In Heatmap the colors are not repeated.

Scheduled Workflow Edit

Fire now enables editing of already scheduled workflows for executions.

Multiple users in a Group

Fire now enables you to add multiple users to a group.

SaveMongoDB Node

Fire now enables you to save your data to MongoDB using this node.

Interactive Dashboard Improvements

- Allows 2 items or more in y-axis in Histogram Chart.
- When there are 2 items on x-axis, only one item is allowed on the y-axis.

25.1.15 January 2019

The following features have been released in Jan 2019.

Interactive Dashboards

Fire now enables you to create Interactive Dashboards. Interactive Dashboards pull data from JDBC sources.

Workflow Wizard

Workflow Wizard enables you to quickly create workflows of various kinds. These could be data cleaning, reporting, spam detection, churn prediction etc.

Pipelines

Fire now supports Pipelines. Pipelines allow creating a DAG of workflows. In the future it would allow adding more types of nodes to the DAG.

Charts Improvements

- Ability to display more than 1 heatmap in a workflow
- Display of X-values and X-axis in the Charts

Processor Improvements

• In RowFilter Processor, the size of conditional expression textfield has been increased.

Support for Uploading Large Files

Fire now supports uploading very large files.

25.1.16 November 2018

The following features have been released in Nov 2018.

Support for Applications

You can now create Applications in Fire. Applications can contain:

- Datasets
- Workflows
- Dashboards
- · Sharing information

This allows you to easily create complex Big Data and ML Applications and work in groups.

Structured Streaming

Fire now supports Structured Streaming. It provides a number of Processors for Structured Streaming. These include Processors for reading from Kafka, reading from files etc. There are also a number of Processors for writing to files etc.

25.1.17 3.1.0 Release Notes

- Release Date: 09/01/2018
- Download TGZ name: sparkflows-fire-3.1.0.tgz
- TGZ Size: 505 MB

Contents of this release

- New Processors Added
 - Decision Node Processor
 - JSON Parser Processor
 - SortBy Processor
 - Empty Dataset Processor
 - Multi Validation Processor
 - String Function Multiple Processor

- Math Function Multiple Processor
- Case When Processor
- Remove Duplicate Processor
- Support for uploading files to HDFS
- Support for LDAP
- Support for running the workflows in debug mode.
 - In debug mode, the number of records processed at each Node are printed.
 - SQL executed is printed where relevant
- Various Workflow Editor Upgrades
 - Ability to rename the Nodes
 - Richer support in JDBC Processor for interactive execution
 - Save Warning when moving away from the Workflow Editor
 - Rich widget support for Multi-Validations Processor
- Support for Caching Datasets in any Processor
- Support for Workflow Cloning
- Richer support in Dashboard Editor for drag and drop of Processors

25.1.18 2.1.0 Release Notes

- Release Date: 04/01/2018
- Download TGZ name: sparkflows-fire-2.1.0.tgz
- TGZ Size: 508 MB

Contents of this release

- Separation of Workflow Server from Workflow Engine
- Support for HDFS File Upload
- New Processors
 - HBase Read Processor
 - HBase Write Processor
 - Split by Multiple Expressions Processor
 - Fixes to Node Correlation
- Support for Rich REST API's

25.1.19 1.4.0 Release Notes

- Release Date: 11/29/2017
- Download TGZ name: sparkflows-fire-1.4.3.tgz
- TGZ Size: 485 MB

Contents of this release

- Scheduling Workflows
- Support for ORC files
- Support for ElasticSearch
- Running in YARN Cluster Mode
- Better browsing experience
- Support for more widget types
- Fixes to Node Correlation Matrix
- Elastic Search Integration
- Support for OpenNLP

25.1.20 1.3.0 Release Notes

- Release Date: 1/8/2017
- Download TGZ name: sparkflows-fire-1.3.0.tgz
- TGZ Size: 485 MB

Contents of this release

- Interactive Workflow Execution
- Streaming Workflow Engine
- Saving & Loading Models
- Support for Jython Nodes
- Many new Machine Learning Nodes added
- Many User Interface Improvements

CHAPTER 26

REST API Authentication

26.1 REST API Authentication

Sparkflows provides REST API for interacting with it.

Swagger is also enabled and is available at http://<machine-name>:8080/swagger-ui.html

To authenticate and access Fire Insights REST APIs, you can use personal access tokens or passwords. We strongly recommend that you use tokens. Like passwords, tokens should be treated with care. Unlike passwords, tokens expire and can be revoked.

Tokens can be generated using Postman.

You can also log in with your username/password, get a session cookie, store it into a file and use it in subsequent requests.

26.1.1 Acquire Session Cookie Using CURL

When invoking the REST APIs of Fire Insights with curl, the first step is to log in and save the incoming cookie into a text file. This file would then be used in making subsequent REST calls via curl.

Save the incoming cookies using the -c option of curl into a file.

In the below example, the Fire Insights web server is running on the local machine at : localhost:8080

You can replace it with your machine name and port.

CURL:

```
curl -i -X POST -d username=admin -d password=admin -c /tmp/cookies.txt_

→localhost:8080/login
```

In the above:

- username = admin
- password = admin

- Incoming cookie gets saved into : /tmp/cookies.txt
- REST API endpoint : localhost:8080/login

26.1.2 Acquire Session Cookie in Python

Fire Insights REST API's can be accessed with Python. Session Cookie can be acquired using username and password and used in the subsequent calls.

Get List of Processors

The below code in Python logs in the user and acquires the session cookie via the Fire Insights REST API.

It then gets the list of Processors in Fire Insights using the REST API and prints them.

```
#!/usr/bin/python
1
2
   # This python script logs into an instance of sparkflows, and then gets the list of
3
   ↔ Processors/Operators supported
4
   # -*- coding: utf-8 -*-
5
   import json
6
   import requests
7
8
   payload = {'username':'admin', 'password':'admin'}
9
10
   # login url
11
   urllogin = 'http://localhost:8080/login'
12
13
   # get list of processors url
14
   urlprocessors = 'http://localhost:8080/nodeList'
15
16
17
   with requests.session() as s:
18
     # log into sparkflows
19
     r = s.post(urllogin, data=payload)
20
21
     # get list of processors
22
     resp = s.get(urlprocessors)
23
24
     parsed_resp = json.loads(resp.text)
25
26
     for i in parsed_resp:
27
       print (i['name'])
28
```

26.1.3 Acquire Token Using CURL

Tokens can be acquired from Fire Insights using curl. They would then be used in making subsequent curl requests. This page work is in progress...

26.1.4 Acquire Token using Postman and Grant Type - Password

This document describes the steps to obtain and use OAuth 2.0 access tokens using Postman.

Overview of Grant Type – Password

The Password grant is used when the application presents a traditional username and password login form to collect the user's credentials and makes a POST request to the server to exchange the password for an access token. The POST requests that the application made looks like the example below.

Form the Post Request

The POST Request method requests that a web server accepts the data enclosed in the body of the request message, most likely for storing it

Table 1: Below are the Relevant Request

Title	URL	
POST	http://hostname:port/oauth/token?grant_type=password&username= <username>&password</username>	= <password></password>

Update the username and password in URL and use as request header.

Click on Authorization tab and select Type - Basic Auth

Basic Auth is an authorization type that requires a verified username and password to access a data resource.

Use default Username sparkflows and Password secret for client authentication. Click on Send to authorize the user and get the access token.

Example

POST http://localhost:8080/oauth/tok • +	***	No Environment 🔻 👁
POST • http://localhost:8080/oa	uth/token?grant_type=password&us	ername= <username>&password=<passwor save<="" send="" th=""></passwor></username>
TYPE	Username	sparkflows
Basic Auth	Password	Show Password
	euler.	Status: 200 OK Time: 110 ms Size: 795 B Download
dy Cookies (1) Headers (18) Test Re		

Now use access_token from previous step to access the REST API

An Access Token is a credential that can be used by an application to access an API. Below is an example to invoke the nodeList REST API of Fire Insights.

26.1.5 Acquire token using Postman - Authorization code

The Authorization Code grant type is used to exchange an authorization code for an access token.

POST http://localhos	t:8080/oauth/tok	GET	http://localhost:8080/nodeList	• +	***	NO Environme	nt	• •	1
GET 👻	http://localho	t:8080/nd	odeList				Send 🔹	Save	Ŧ
TYPE OAuth 2.0 The authorization value of the request Learn authorization Add authorization Request Headers Preview Request	erated when you more about data to	* send	Access Token		763a0af7-7890-4b97-a44 Get New Access Token	b-883902342e65	Available	Tokens '	r
Body Cookies (1)	Headers (16)	Test Re	sults		Status: 200 C	DK Time: 80 ms 5	Size: 455.58 KB	Downloa	
Pretty Raw	Preview	JSON 🔻	. ⇒						۹
3 4	"id": "3", "path": "/AW	s/".							

Get the access token

The app can obtain an access token that provides temporary, secure access to it. Below are steps involved to Request an Access_token

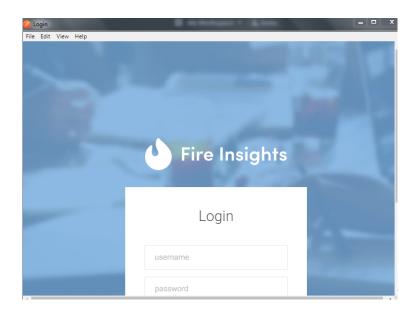
Click on Authorization tab

• Select Type OAuth 2.0

GET NEW ACCESS TOKEN		×
Token Name	sparkflows	
Grant Type	Authorization Code	Ŧ
Callback URL 🕕	http://localhost:8080/callback	
Auth URL 🕕	http://localhost:8080/oauth/authorize	
Access Token URL 🕕	http://localhost:8080/oauth/token	
Client ID 🕕	sparkflows	
Client Secret 🜒	******	
Scope 🕕	write	
State 🚺	State	
Client Authentication	Send as Basic Auth header	Ŧ
	Request Token	

Click on Request Token

It will redirect to sparkflows login URL Page.

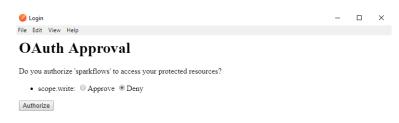


Fill the username and password and click on signIn

It will then display the OAuth Approval page.

OAuth Approval

OAuth is an authentication protocol that allows you to approve one application interacting with another on your behalf without giving away your password. Below is the Screenshot for updating the Oauth approval.



Click on Use token

A security token (sometimes called an authentication token) is a small hardware device that the owner carries to authorize access to a network service.

MANAGE ACCESS TOKENS		×
ALL TOKENS	Token Name	sparkflows
sparkflows	Access Token	a9da183e-d65a-427b-9f68-6174aa4d0e3b
sparkflows	Token Type	bearer
Token Name	refresh_token	9c5e9112-a4ef-422e-a60e-2fd5a3af2475
Token Name	expires_in	49999
Token Name	scope	write
Token Name		
Token Name		Use Token

Using tokens for accessing REST API

Using above token we can access the REST API.

POST http://localhos	st:8080/oauth/tok 🖲 GET	http://localhost:8080/nodeList	• + …	NO Environment	• •
GET 👻	http://localhost:8080/n	odeList		Se	end 🔻 Save 💌
TYPE OAuth 2.0 The authorization automatically gent the request. Learn authorization Add authorization Request Headers Preview Request	erated when you send more about data to s v	Access Token	e9ds183e-d65 Get New Acces	a-427b-9468-6174aa4d0e3b	Available Tokens 🔻
ody Cookies (1) Pretty Raw	Headers (17) Test R		St	atus: 200 OK Time: 46 ms Size:	455.67 KB Download
1 • [2 • { 3 4 5 6 7	"id": "3", "path": "/AHS/", "name": "ReadAVRO-53 "iconImage": null, "description": "Read	-AWS", the AVRO file from spec	ified S3 location",		

26.1.6 Acquire Token in Python - Grant Type Password

Below are examples of Python code for accessing the Fire REST API using Python.

Get Processor Count

The below code in Python does the following:

- · Acquires the token using Grant Type Password
- Invokes the Fire Insights REST API to get the number of processors list available in Fire Insights.

```
#step B, C - single call with resource owner credentials in the body and client.
→credentials as the basic auth header will return #access_token
data = {'grant_type': 'password', 'username': RO_user, 'password': RO_password}
access_token_response = requests.post(token_url, data=data, verify=False, allow_
→redirects=False, auth=(client_id, client_secret))
print(access_token_response.headers)
print(access_token_response.text)
tokens = json.loads(access_token_response.text)
print( "access token: " + tokens['access_token'])
# Step C - now we can use the access_token to make another rest api call to get_
→the processor count
api_call_headers = {'Authorization': 'Bearer ' + tokens['access_token']}
print( api_call_headers)
api_call_response = requests.get(processor_count_api_url, headers=api_call_
→headers, verify=False)
print(api_call_response.text)
```

After running above REST API code in Python, we get the below results.

(nemu)	[zbarkf]ows@develowment-build-maching negral]5 withon processor.pv
	Mon. 12 Ruy 2019 09:25:58 GHT', 'Access-Control-Allow-Origin': '*', 'Access-Control-Allow-Methods': 'POST.GET.DELETE.PUT.OPTIONS', 'Access-Control-Allow-Hethods': 'POST.GET.DELETE.PUT.OPTIONS', 'Access-Control-Allow-Hethods'; 'POST.GET
	-Allow-Gredentials': 'true', 'X-Content-Type-Options': 'nosniff', 'X-XSS-Protection': '1; mode=block', 'Cache-Control': 'no-cache, no-store, max-age=0, must-
	: '0', 'X-Frame-Options': 'DBWY', X-Application-Context': 'application.db.logd/, 'Content-Tuye': 'application.jpon:charset witf B', 'Transfer-Encoding': 'c _ token': Blo55dad 522; Algoshodicate2, ''cohent type': 'bearery''refresh.token': '5222d55' FR2-91ce-b51'/Ad08189766', ''acpirez.in': 40979, ''cope''
	isation': /Barbon Bit65Han-522-4831-8915-3965adciec2f')

Infer Hadoop Cluster Configurations

The below code in Python invokes the Fire Insights REST API to infer Hadoop cluster configurations. It then saves the infer cluster Hadoop configurations as updated values.

```
#!/usr/bin/python
import requests
import json
token_url = "http://hostname:8080/oauth/token"
infer_configuration_api_url = "http://hostname:8080/api/v1/configurations/infer"
save_configuration_api_url = "http://hostname:8080/api/v1/configurations"
#Step A - resource owner supplies credentials
#Resource owner (enduser) credentials
RO_user = 'admin' #input your own username
RO_password = 'admin' #input your own password
#client (application) credentials
```

```
client_id = 'sparkflows'
client_secret = 'secret'
#step B, C - single call with resource owner credentials in the body and client.
\hookrightarrow credentials as the basic auth header will return \#access\_token
data = {'grant_type': 'password', 'username': R0_user, 'password': R0_password}
access_token_response = requests.post(token_url, data=data, verify=False, allow_
→redirects=False, auth=(client_id, client_secret))
print(access_token_response.headers)
print(access_token_response.text)
tokens = json.loads(access_token_response.text)
print( "access token: " + tokens['access_token'])
#Step- now use the access_token to call infer configuration api and its save api.
api_call_headers = {'Authorization': 'Bearer ' + tokens['access_token']}
print( api_call_headers)
#infer the hadoop configuration
infer_configuration_api_response = requests.get(infer_configuration_api_url,
print(" infer configuration response : "+ infer_configuration_api_response.text)
#save the hadoop configuration
save_configuration_api_response = requests.post(save_configuration_api_url,json=infer_
--configuration_api_response.json(), headers=api_call_headers, verify=False)
print(" configuration after save : "+save_configuration_api_response.text)
```

After running above REST API code using Python, Will get the results as below



CHAPTER 27

REST API's using Python

27.1 REST API Examples using Python

Sparkflows provides REST API for interacting with it.

Below are examples using tokens. The first step is to log in with your username and password and acquire the token.

Swagger is also enabled and is available at http://<machine-name>:8080/swagger-ui.html

27.1.1 Accessing REST API using Python & Session

Fire Insights REST APIs can be accessed with Python. This page provides 2 examples of accessing the REST API's with Python.

Get List of Processors

The below code in Python gets the list of Processors in Fire Insights using the REST API and prints them.

```
#!/usr/bin/python
1
2
   # This python script logs into an instance of sparkflows, and then gets the list of
3
   ↔ Processors/Operators supported
4
   # -*- coding: utf-8 -*-
5
   import json
6
   import requests
7
8
   payload = {'username':'admin', 'password':'admin'}
9
10
11
   # login url
   urllogin = 'http://localhost:8080/login'
12
13
```

```
# get list of processors url
14
   urlprocessors = 'http://localhost:8080/nodeList'
15
16
   with requests.session() as s:
17
18
      # log into sparkflows
19
     r = s.post(urllogin, data=payload)
20
21
      # get list of processors
22
     resp = s.get(urlprocessors)
23
24
     parsed_resp = json.loads(resp.text)
25
26
27
     for i in parsed_resp:
       print (i['name'])
28
```

Create a New Workflow

The Workflow JSON is saved in a file called workflow.json.

The below code in Python creates a new Workflow in the Project with id 1.

```
#!/usr/bin/python
1
2
   # This python script logs into an instance of sparkflows, and then gets the list of
3
    ↔ Processors/Operators supported
4
   # -*- coding: utf-8 -*-
5
   import json
6
   import requests
7
8
   payload = {'username':'admin', 'password':'admin'}
9
10
   # login url
11
   urllogin = 'http://localhost:8080/login'
12
13
   # save workflow url
14
   urlsaveworkflow = 'http://localhost:8080/saveWorkflow'
15
16
   # read workflow json
17
   wf = open("workflow.json", "r", encoding='utf8')
18
   workflow = wf.read()
19
20
   # define other parameters
21
   analysisFlowId = "null"
22
23
   projectId = "1"
   engine = "scala"
24
25
   with requests.session() as s:
26
27
      # log into sparkflows
28
     s.get(urllogin)
29
30
     r = s.post(urllogin, data=payload)
31
32
```

27.1.2 Accessing REST API using Python & Tokens

Below are examples of Python code for accessing the Fire REST API using Python.

Get Processor Count

The below code in Python invokes the Fire Insights REST API to calculate number of processors list available in Fire Insight.

```
#!/usr/bin/python
import requests
import json
import getpass
token_url = "http://localhost:8080/oauth/token"
processor_count_api_url = "http://localhost:8080/getNodeCount" # processor.
⇔list count api of sparkflows
#Step A - resource owner supplies credentials
#Resource owner (enduser) credentials
RO_user = 'admin'
RO_password = 'admin'
#client (application) credentials
client_id = 'sparkflows'
client_secret = 'secret'
#step B, C - single call with resource owner credentials in the body and_
→client credentials as the basic auth header will return #access_token
data = {'grant_type': 'password', 'username': RO_user, 'password': RO_
⇔password}
access_token_response = requests.post(token_url, data=data, verify=False,_

wallow_redirects=False, auth=(client_id, client_secret))

print(access_token_response.headers)
print(access_token_response.text)
tokens = json.loads(access_token_response.text)
print( "access token: " + tokens['access_token'])
```



After running above REST API code using Python, will get the results as below:

Chates '1 Sun, '12 Sun 2019 '07:518' 071', 'Reset: 'derive', 'ller's's', 'lecen-Castra', Allar Kthaf: '1071', 'Reset: 'derive', 'lecen-Castra', 'ller's', 'lecen-Castra', 'ller's', 'll

Infer Hadoop Cluster Configurations

The below code in Python invokes the Fire Insights REST API to infer Hadoop cluster configurations. It then saves the infer cluster Hadoop configurations as updated values.

```
#!/usr/bin/python
import requests
import json
token_url = "http://localhost:8080/oauth/token"
infer_configuration_api_url = "http://localhost:8080/api/v1/configurations/infer"
save_configuration_api_url = "http://localhost:8080/api/v1/configurations"
#Step A - resource owner supplies credentials
#Resource owner (enduser) credentials
RO_user = 'admin' #input your own username
RO_password = 'admin' #input your own password
#client (application) credentials
client_id = 'sparkflows'
client_secret = 'secret'
#step B, C - single call with resource owner credentials in the body and client.
Generalized as the basic auth header will return #access_token
data = {'grant_type': 'password', 'username': RO_user, 'password': RO_password}
access_token_response = requests.post(token_url, data=data, verify=False, allow_
→redirects=False, auth=(client_id, client_secret))
print(access_token_response.headers)
print(access_token_response.text)
```

After running above REST API code using Python, will get the results as below



CHAPTER 28

REST API's using Java

28.1 REST API Examples using Java

Fire Insighs provides REST API for interacting with it.

Below are examples using tokens. The first step is to log in with your username and password and acquire the token. Swagger is also enabled and is available at http://<machine-name>:8080/swagger-ui.html

CHAPTER 29

REST API's using curl

29.1 REST API Examples using curl

This topic contains a range of examples that demonstrate how to use the Fire Insights API using curl.

Acquire Session Cookie Using Curl

When invoking the REST APIs of Fire Insights with curl, the first step is to log in and save the incoming cookie into a text file. This file would then be used in making subsequent REST calls via curl.

Save the incoming cookies using the -c option of curl into a file.

In the below examples, the Fire Insights web server is running on the local machine at : localhost:8080

You can replace it with your machine name and port.

Login and save the session cookie into /tmp/cookies.txt:

```
curl -i -X POST -d username=admin -d password=admin -c /tmp/cookies.txt_
→localhost:8080/login
```

In the above:

- username = admin
- password = admin
- Incoming cookie gets saved into : /tmp/cookies.txt
- REST API endpoint : localhost:8080/login

There are various categories of REST API's available:

29.1.1 Processors REST API's

Overview

The Processors REST APIs, allow you to get the list of available Processors and details regarding each Processor.

Below are the various Processor APIs available in Fire Insights.

They should be executed after you have logged into Fire Insights. Use the -b option to use the cookies previously saved.

GET Processors List

Gets the list of processors available.

An example request for getting list of processors:

An example response:

```
Γ
{
 "id": "3",
 "path": "/01-Connectors/",
 "name": "ReadCassandra",
 "iconImage": null,
 "description": "This node reads data from Apache Cassandra",
 "details": "",
 "examples": "",
 "type": "dataset",
 "nodeClass": "fire.nodes.cassandra.NodeReadCassandra",
 "x": null,
 "y": null,
 "fields": [
   {
      "name": "storageLevel",
      "value": "DEFAULT",
      "widget": "array",
      "title": "Output Storage Level",
      "description": "Storage Level of the Output Datasets of this Node",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": [
       "DEFAULT",
       "NONE",
        "DISK_ONLY",
        "DISK_ONLY_2",
        "MEMORY_ONLY",
        "MEMORY_ONLY_2",
        "MEMORY_ONLY_SER"
        "MEMORY_ONLY_SER_2",
        "MEMORY_AND_DISK"
        "MEMORY_AND_DISK_2",
        "MEMORY_AND_DISK_SER",
        "MEMORY_AND_DISK_SER_2",
        "OFF_HEAP"
      ],
      "required": false,
```

```
"display": true,
     "editable": true,
     "disableRefresh": false
   },
   {
     "name": "table",
     "value": "",
     "widget": "textfield",
     "title": "Cassandra Table",
     "description": "Cassandra Table from which to read the data",
     "optionsMap": null,
     "datatypes": null,
     "optionsArray": null,
     "required": true,
     "display": true,
     "editable": true,
     "disableRefresh": false
   },
   {
     "name": "keyspace",
     "value": "",
     "widget": "textfield",
     "title": "Cassandra Keyspace",
     "description": "Cassandra Keyspace",
     "optionsMap": null,
     "datatypes": null,
     "optionsArray": null,
     "required": true,
     "display": true,
     "editable": true,
     "disableRefresh": false
   },
   {
     "name": "cluster",
"value": "",
     "widget": "textfield",
     "title": "Cassandra Cluster",
     "description": "The group of the Cluster Level ",
     "optionsMap": null,
     "datatypes": null,
     "optionsArray": null,
     "required": false,
     "display": true,
     "editable": true,
     "disableRefresh": false
   }
],
 "engine": "scala"
},
```

GET Node Count

Gets the count of the processors.

An example request for getting count of the processors:

An example response:

266

GET Processors list for Engine

Gets the list of processors for the specified engine(scala or pyspark or empty-field for all).

An example request for getting list of processors for scala

An example response:

```
[
{
 "id": "3",
 "path": "/01-Connectors/",
 "name": "ReadCassandra",
 "iconImage": null,
 "description": "This node reads data from Apache Cassandra",
 "details": "",
 "examples": "",
 "type": "dataset",
 "nodeClass": "fire.nodes.cassandra.NodeReadCassandra",
  "x": null,
  "y": null,
  "fields": [
   {
     "name": "storageLevel",
     "value": "DEFAULT",
     "widget": "array",
     "title": "Output Storage Level",
     "description": "Storage Level of the Output Datasets of this Node",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": [
       "DEFAULT",
       "NONE",
        "DISK_ONLY",
        "DISK_ONLY_2",
       "MEMORY_ONLY",
        "MEMORY_ONLY_2",
        "MEMORY_ONLY_SER",
        "MEMORY_ONLY_SER_2",
        "MEMORY_AND_DISK",
        "MEMORY AND DISK 2",
        "MEMORY_AND_DISK_SER",
        "MEMORY_AND_DISK_SER_2",
        "OFF_HEAP"
     ],
      "required": false,
```

```
"display": true,
      "editable": true,
      "disableRefresh": false
    },
    {
      "name": "table",
      "value": "",
      "widget": "textfield",
      "title": "Cassandra Table",
      "description": "Cassandra Table from which to read the data",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": null,
      "required": true,
      "display": true,
      "editable": true,
      "disableRefresh": false
    },
    {
      "name": "keyspace",
      "value": "",
      "widget": "textfield",
      "title": "Cassandra Keyspace",
      "description": "Cassandra Keyspace",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": null,
      "required": true,
      "display": true,
      "editable": true,
      "disableRefresh": false
    },
    {
      "name": "cluster",
"value": "",
      "widget": "textfield",
      "title": "Cassandra Cluster",
      "description": "The group of the Cluster Level ",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": null,
      "required": false,
      "display": true,
      "editable": true,
      "disableRefresh": false
    }
 ],
  "engine": "scala"
},
```

GET Processor Details by Name

Gets Processor Details by Name

An example request for getting Processor Details by Name:

An example response:

{

```
"id": "17",
 "path": "/02-ReadStructured/",
"name": "ReadCSV",
"iconImage": null,
"description": "It reads in CSV files and creates a DataFrame from it",
"details": "",
"examples": "",
"type": "dataset",
"nodeClass": "fire.nodes.dataset.NodeDatasetCSV",
"x": null,
"y": null,
"fields": [
{
  "name": "storageLevel",
  "value": "DEFAULT",
  "widget": "array",
  "title": "Output Storage Level",
  "description": "Storage Level of the Output Datasets of this Node",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": [
    "DEFAULT",
    "NONE",
    "DISK_ONLY",
    "DISK_ONLY_2",
    "MEMORY_ONLY",
    "MEMORY_ONLY_2",
    "MEMORY_ONLY_SER",
    "MEMORY_ONLY_SER_2",
    "MEMORY_AND_DISK",
    "MEMORY_AND_DISK_2",
    "MEMORY_AND_DISK_SER",
    "MEMORY_AND_DISK_SER_2",
    "OFF_HEAP"
  ],
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "path",
  "value": "",
  "widget": "textfield",
  "title": "Path",
  "description": "Path of the Text file/directory",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": null,
  "required": true,
  "display": true,
```

```
"editable": true,
  "disableRefresh": false
},
{
  "name": "separator",
  "value": ",",
  "widget": "textfield",
  "title": "Separator",
  "description": "CSV Separator",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": null,
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "header",
  "value": "false",
  "widget": "array",
  "title": "Header",
  "description": "Does the file have a header row",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": [
   "true",
    "false"
  ],
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "dropMalformed",
  "value": "false",
  "widget": "array",
  "title": "Drop Malformed",
  "description": "Whether to drop Malformed records or error",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": [
   "true",
    "false"
  ],
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "outputColNames",
  "value": "[]",
  "widget": "schema_col_names",
  "title": "Column Names for the CSV",
  "description": "New Output Columns of the SQL",
```

```
"optionsMap": null,
    "datatypes": null,
    "optionsArray": null,
    "required": false,
    "display": true,
    "editable": true,
    "disableRefresh": false
  },
  {
    "name": "outputColTypes",
    "value": "[]",
    "widget": "schema_col_types",
    "title": "Column Types for the CSV",
    "description": "Data Type of the Output Columns",
    "optionsMap": null,
    "datatypes": null,
    "optionsArray": null,
    "required": false,
    "display": true,
    "editable": true,
    "disableRefresh": false
  },
  {
    "name": "outputColFormats",
    "value": "[]",
    "widget": "schema_col_formats",
    "title": "Column Formats for the CSV",
    "description": "Format of the Output Columns",
    "optionsMap": null,
    "datatypes": null,
    "optionsArray": null,
    "required": false,
    "display": true,
    "editable": true,
    "disableRefresh": false
  }
],
"engine": "all"
```

Node Rules

}

Gets the node rules used in the workflow editor.

An example request for getting the node rules:

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8080/api/v1/node-
→rules' -b /tmp/cookies.txt
```

An example response:

```
{
 "nodeType": "dataset",
 "possibleSources": [
```

(continues on next page)

ſ

```
"shellcommand"
  ],
  "minNumOfInputs": 0,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": null,
  "sourceRestrictions": [],
  "backgroundColor": "#F0F1F9",
  "nodeIcon": "fa-th-list",
  "nodeShape": "rectangle"
},
{
  "nodeType": "shellcommand",
  "possibleSources": [
   "dataset",
   "scala",
    "pyspark",
    "transform",
   "join",
    "ml-transformer",
    "ml-predict",
    "sparkstreaming"
 ],
  "minNumOfInputs": 0,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": null,
  "sourceRestrictions": [],
  "backgroundColor": "#F0F1F9",
  "nodeIcon": "fa-th-list",
  "nodeShape": "rectangle"
},
{
  "nodeType": "sparkstreaming",
  "possibleSources": [],
  "minNumOfInputs": 0,
  "maxNumOfInputs": 0,
  "maxNumOfOutputs": null,
  "sourceRestrictions": [],
  "backgroundColor": "#FFEB94",
  "nodeIcon": "fa-external-link",
  "nodeShape": "rectangle"
},
{
  "nodeType": "transform",
  "possibleSources": [
   "dataset",
    "scala",
    "pyspark",
    "transform",
    "join",
    "ml-transformer",
    "ml-predict",
    "sparkstreaming",
    "shellcommand"
  ],
  "minNumOfInputs": 1,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": null,
```

```
"sourceRestrictions": [],
 "backgroundColor": "#AFD4F0",
 "nodeIcon": "fa-tumblr-square",
  "nodeShape": "rectangle"
},
{
  "nodeType": "scala",
  "possibleSources": [
   "dataset",
   "transform",
   "join",
   "ml-transformer",
   "ml-predict",
    "sparkstreaming",
    "shellcommand"
 ],
  "minNumOfInputs": 0,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": null,
  "sourceRestrictions": [],
  "backgroundColor": "#AFD4F0",
  "nodeIcon": "fa-tumblr-square",
  "nodeShape": "rectangle"
},
{
 "nodeType": "pyspark",
 "possibleSources": [
   "dataset",
   "transform",
   "join",
    "ml-transformer",
    "ml-predict",
    "sparkstreaming",
    "shellcommand"
 ],
 "minNumOfInputs": 0,
 "maxNumOfInputs": 1,
 "maxNumOfOutputs": null,
 "sourceRestrictions": [],
 "backgroundColor": "#AFD4F0",
 "nodeIcon": "fa-tumblr-square",
  "nodeShape": "rectangle"
},
{
 "nodeType": "join",
  "possibleSources": [
   "dataset",
    "transform",
   "join",
    "shellcommand",
    "sparkstreaming"
 ],
 "minNumOfInputs": 2,
 "maxNumOfInputs": 8,
  "maxNumOfOutputs": null,
  "sourceRestrictions": [],
  "backgroundColor": "#D4A190",
```

```
"nodeIcon": "fa-stumbleupon",
  "nodeShape": "rectangle"
},
{
 "nodeType": "ml-transformer",
  "possibleSources": [
   "dataset",
   "transform",
    "ml-transformer",
   "join",
   "shellcommand"
 ],
 "minNumOfInputs": 1,
 "maxNumOfInputs": 1,
 "maxNumOfOutputs": "2",
  "sourceRestrictions": [],
 "backgroundColor": "#dfe166",
 "nodeIcon": "fa-qrcode",
  "nodeShape": "rectangle"
},
{
  "nodeType": "ml-estimator",
 "possibleSources": [
   "dataset",
   "transform",
   "ml-transformer",
   "join",
   "shellcommand"
 ],
  "minNumOfInputs": 1,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": "2",
  "sourceRestrictions": [],
  "backgroundColor": "#F7EFE2",
 "nodeIcon": "fa-qrcode",
  "nodeShape": "rectangle"
},
{
 "nodeType": "ml-predict",
 "possibleSources": [
   "dataset",
   "transform",
   "join",
   "ml-estimator",
    "ml-transformer",
    "ml-pipeline",
    "ml-crossvalidator",
    "ml-modelload"
 ],
 "minNumOfInputs": 1,
 "maxNumOfInputs": 2,
 "maxNumOfOutputs": null,
 "sourceRestrictions": [],
 "backgroundColor": "#D7CFC2",
 "nodeIcon": "fa-grcode",
  "nodeShape": "rectangle"
},
```

```
"nodeType": "ml-evaluator",
  "possibleSources": [
   "ml-predict",
    "ml-estimator"
    "ml-pipeline"
 ],
  "minNumOfInputs": 1,
  "maxNumOfInputs": 1,
 "maxNumOfOutputs": "1",
 "sourceRestrictions": [],
 "backgroundColor": "#ff9900",
 "nodeIcon": "fa-qrcode",
 "nodeShape": "rectangle"
},
{
  "nodeType": "ml-pipeline",
  "possibleSources": [
   "ml-estimator",
    "ml-transformer"
 ],
  "minNumOfInputs": 1,
  "maxNumOfInputs": 1,
 "maxNumOfOutputs": "1",
 "sourceRestrictions": [],
 "backgroundColor": "#1FFF62",
 "nodeIcon": "fa-grcode",
 "nodeShape": "rectangle"
},
{
  "nodeType": "ml-crossvalidator",
  "possibleSources": [
   "ml-evaluator"
 ],
  "minNumOfInputs": 1,
 "maxNumOfInputs": 1,
 "maxNumOfOutputs": null,
 "sourceRestrictions": [],
 "backgroundColor": "#F9FC81",
 "nodeIcon": "fa-grcode",
 "nodeShape": "rectangle"
},
{
  "nodeType": "ml-trainvalidationsplit",
  "possibleSources": [
   "ml-evaluator"
 1,
  "minNumOfInputs": 1,
 "maxNumOfInputs": 1,
 "maxNumOfOutputs": null,
 "sourceRestrictions": [],
 "backgroundColor": "#B681FC",
 "nodeIcon": "fa-grcode",
 "nodeShape": "rectangle"
},
{
  "nodeType": "ml-modelsave",
```

```
"possibleSources": [
    "ml-estimator",
    "ml-pipeline",
    "ml-crossvalidator",
    "ml-trainvalidationsplit"
 ],
  "minNumOfInputs": 1,
  "maxNumOfInputs": 1,
  "maxNumOfOutputs": "1",
 "sourceRestrictions": [],
 "backgroundColor": "#FCB881",
 "nodeIcon": "fa-qrcode",
 "nodeShape": "rectangle"
},
{
  "nodeType": "ml-modelload",
  "possibleSources": [
   "dataset",
    "transform",
   "join",
    "ml-estimator",
    "ml-transformer",
    "ml-pipeline",
    "ml-crossvalidator",
    "ml-modelsave"
 ],
 "minNumOfInputs": 0,
 "maxNumOfInputs": 1,
 "maxNumOfOutputs": "1",
 "sourceRestrictions": [],
 "backgroundColor": "#FCB881",
 "nodeIcon": "fa-qrcode",
  "nodeShape": "rectangle"
},
{
 "nodeType": "doc",
 "possibleSources": [
   "doc"
 ],
 "minNumOfInputs": 0,
 "maxNumOfInputs": 0,
 "maxNumOfOutputs": null,
 "sourceRestrictions": [],
 "backgroundColor": "#FFFF88",
 "nodeIcon": "fa-file-text",
  "nodeShape": "rectangle"
},
{
  "nodeType": "sticky",
  "possibleSources": [],
 "minNumOfInputs": 0,
 "maxNumOfInputs": 0,
 "maxNumOfOutputs": null,
  "sourceRestrictions": [],
 "backgroundColor": "#FFFF88",
  "nodeIcon": "fa-file-text",
  "nodeShape": "rectangle"
```

},

(continued from previous page)

29.1.2 Datasets REST API

Overview

The Dataset REST APIs, allow you to manage the Datasets.

Below are the various Dataset APIs available in Fire Insights, They should be executed after you have logged into Fire Insights.

GET List of Datasets by Application

Returns the list of Datasets for the logged in user for a given application id:

```
curl -X GET --header 'Accept: application/json' --header 'api_key: cookies' 'http://

→localhost:8080/api/v1/datasets?sortPara=dsc&projectId=1'
```

Create / Update Dataset

If id value is not passed, new dataset will be created:

JSON

```
{
  "id": 13,
  "version": 0,
 "name": "spam",
 "header": true,
  "path": "data\/spam.csv",
  "delimiter": ",",
  "schemaModel": {
    "schemaColList": [
     {
        "colName": "label",
       "colType": "DOUBLE",
       "colFormat": "",
        "colMLType": "NUMERIC"
      },
      {
        "colName": "message",
       "colType": "STRING",
       "colFormat": "",
        "colMLType": "TEXT"
      },
      {
        "colName": "id",
        "colType": "DOUBLE",
        "colFormat": "",
        "colMLType": "NUMERIC"
```

```
}
]
}
```

Curl

Delete Dataset

- "datasetId": "98"
- "projectId": "33"

An example request for Deleting dataset:

```
curl -X DELETE --header 'Accept: text/plain' 'http://localhost:8080/api/v1/datasets/

→98?projectId=33'
```

An example response:

```
Dataset with id 98 deleted successfully
```

Get Dataset by Id

- "datasetId": "65"
- "projectId": "33"

An example request for Getting dataset by Id:

An example response:

```
{
   "id": 65,
   "userId": 33,
   "uuid": "1e13ec2a-4094-405e-a6e7-ffed3bd027f7",
   "version": 0,
   "name": "Test-dataset",
   "category": null,
   "description": "Test",
   "header": true,
   "readOptions": null,
   "path": "/user/sparkflows/Clickthru.csv",
```

```
"delimiter": ",",
  "datasetType": "CSV",
  "filterLinesContaining": null,
  "datasetSchema": "{colNames:[\"Timestamp\",\"UserId\",\"IP Address\",\"Product Id\
→"],colTypes:[\"STRING\",\"INTEGER\",\"STRING\",\"INTEGER\"],colFormats:[\"\",\",\
→"\",\"\"],colMLTypes:[\"TEXT\",\"NUMERIC\",\"TEXT\",\"NUMERIC\"]}",
  "dateCreated": 1566880637842,
  "dateLastUpdated": 1566880637846,
  "permission": null,
  "readOptionsModel": null,
  "schemaModel": {
   "schemaColList": [
   {
     "colName": "Timestamp",
     "colType": "STRING",
     "colFormat": "",
     "colMLType": "TEXT"
   },
   {
     "colName": "UserId",
     "colType": "INTEGER",
     "colFormat": "",
     "colMLType": "NUMERIC"
   },
   {
     "colName": "IP Address",
     "colType": "STRING",
     "colFormat": "",
     "colMLType": "TEXT"
   },
   {
     "colName": "Product Id",
     "colType": "INTEGER",
     "colFormat": "",
     "colMLType": "NUMERIC"
   }
 ]
 },
   "sampleData": {
   "headers": [
   "Timestamp",
   "UserId",
   "IP Address",
   " Product Id"
 ],
 "cells": [
   [
     "9:03 AM",
     "275",
     "207.51.113.192",
     "1"
   ],
   [
     "12:57 AM",
     "586",
     "62.34.98.94",
     "2"
```

```
],
[
  "2:45 AM",
  "508",
 "20.237.172.182",
  "3"
],
[
  "2:13 PM",
  "378",
  "69.215.255.150",
  "4"
],
[
  "9:27 AM",
  "965",
  "56.101.183.251",
  "5"
],
[
  "8:18 AM",
  "263",
  "9.151.97.180",
  "6"
],
[
  "9:40 AM",
 "670",
 "101.195.1.186",
  "7"
],
[
  "7:14 AM",
 "447",
  "232.29.216.95",
  "8"
],
[
  "12:57 AM",
  "33",
  "85.119.50.62",
  "9"
],
[
  "12:56 AM",
  "589",
  "185.132.243.178",
  "10"
],
[
  "11:04 PM",
  "22",
  "120.212.232.218",
  "11"
],
[
  "8:29 PM",
```

"504",

"13"

"15"

"16"

"17"

"18"

"429",

"20"

"115.123.246.193",

], [

], [

], [

], Γ

], [

], Γ

], [

], Γ

1] },

"226.70.25.117", "12" "5:18 PM", "228", "213.53.100.18", "2:56 PM", "536", "60.65.25.167", "14" "3:57 AM", "46", "149.156.17.120", "8:05 AM", "812", "23.213.182.107", "12:02 PM", "980", "93.20.165.16", "12:53 PM", "915", "24.180.112.147", "11:32 AM". "814", "110.81.139.11", "19" "11:01 PM",

(continued from previous page)

"ison": "{\"id\":65, \"userId\":33, \"uuid\": \"1e13ec2a-4094-405e-a6e7-ffed3bd027f7\", \ ↔ "version\":0, \"name\":\"Test-dataset\", \"description\":\"Test\", \"header\":true, \ →"path\":\"/user/sparkflows/Clickthru.csv\", \"delimiter\":\", \", \"datasetType\":\ →"CSV\", \"datasetSchema\":\"{colNames:[\\\"Timestamp\\\", \\\"UserId\\\", (@@ntinucson next page)

"projectId": 33
},

Get Dataset Count

Returns the count of datasets available:

```
curl -X GET --header 'Accept: application/json' --header 'api_key: cookies' 'http://

→localhost:8080/api/v1/datasets/count'
```

Get sample data

Delimiter and header are optional values

- · path: data/spam.csv
- schema: {"colNames":["0.0","this is not a spam","3.0"],"colTypes":["DOUBLE","STRING","DOUBLE"],"colFormats":["","",""

CURL:

Returns schema of the files in the given path using the given delimiter

- · delimiter and header are optional values
- path:data/spam.csv
- schema: {"colNames":["0.0","this is not a spam","3.0"],"colTypes":["DOUBLE","STRING","DOUBLE"],"colFormats":["","","",""

CURL:

Get Latest Five Datasets

Returns the latest updated datasets:

Get the list of files/directories in the given path

• path:data/transaction.csv

CURL:

```
curl -X GET --header 'Content-Type: application/json' --header 'Accept: application/

→json' -d 'data/transaction.csv' 'http://localhost:8080/filesInPathJSON -b /tmp/

→cookies.txt'
```

29.1.3 Workflow REST API

The Workflow REST API's, allow you to interact with the Workflows.

Below are the various Workflow API's available in Fire Insights. They should be executed after you have logged into Fire Insights.

Create Workflow

Create a new Workflow.

An example request for creating workflow:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept:
→application/json' -d '{
"analysisflowId": 1,
"comment": "string",
"projectId": 33,
"workflow": {
 "category": "string",
 "dataSetDetails": [
   {
      "datasetSchema": "string",
     "datasetType": "CSV",
      "delimiter": "string",
      "description": "string",
      "filterLinesContaining": "string",
     "header": true,
      "id": 0,
     "name": "string",
      "path": "string",
     "readOptions": "string",
     "uuid": "string",
     "version": 0
   }
 ],
 "description": "string",
  "edges": [
   {
      "id": 0,
      "source": "string",
      "target": "string"
   }
 ],
  "engine": "string",
 "h2OWorkflow": true,
```

```
"name": "string",
  "nodes": [
   {
      "description": "string",
      "details": "string",
      "engine": "string",
      "examples": "string",
      "fields": [
        {
          "datatypes": [
            "string"
          ],
          "description": "string",
          "disableRefresh": true,
          "display": true,
          "editable": true,
          "name": "string",
          "optionsArray": [
            "string"
          ],
          "optionsMap": {},
          "required": true,
          "title": "string",
          "value": "string",
          "widget": "string"
        }
      ],
      "iconImage": "string",
      "id": "string",
      "name": "string",
      "nodeClass": "string",
      "path": "string",
      "type": "string",
      "x": "string",
      "v": "string"
   }
 ],
 "parameters": "string",
 "uuid": "string"
}' 'http://hostname:port/api/v1/workflows' -b /tmp/cookies.txt
```

An example response:

193

Execute Workflow

Execute specified Workflow.

An example request for Executing specified workflow:

```
"emailOnSuccess": "string",
"libJars": "string",
"programParameters": "string",
"sparkConfig": "string",
"workflowId": 1
}' 'http://hostname:port/api/v1/workflow/execute' -b /tmp/cookies.txt
```

An example response:

162

Update Workflow

Update specified Workflow.

An example request for updating workflow:

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/
⇔json' -d '{
"analysisflowId": 1,
"comment": "string",
"projectId": 33,
"workflow": {
 "category": "string",
  "dataSetDetails": [
   {
      "datasetSchema": "string",
     "datasetType": "CSV",
      "delimiter": "string",
     "description": "string",
      "filterLinesContaining": "string",
      "header": true,
      "id": 0,
      "name": "string",
      "path": "string",
      "readOptions": "string",
      "uuid": "string",
      "version": 0
   }
 ],
  "description": "string",
  "edges": [
   {
     "id": 0,
     "source": "string",
      "target": "string"
    }
 ],
  "engine": "string",
 "h2OWorkflow": true,
  "name": "string",
  "nodes": [
    {
      "description": "string",
      "details": "string",
```

```
"engine": "string",
      "examples": "string",
      "fields": [
        {
          "datatypes": [
            "string"
          ],
          "description": "string",
          "disableRefresh": true,
          "display": true,
          "editable": true,
          "name": "string",
          "optionsArray": [
            "string"
          ],
          "optionsMap": {},
          "required": true,
          "title": "string",
          "value": "string",
          "widget": "string"
        }
     ],
      "iconImage": "string",
      "id": "string",
      "name": "string",
      "nodeClass": "string",
      "path": "string",
     "type": "string",
      "x": "string",
      "y": "string"
   }
 ],
  "parameters": "string",
  "uuid": "string"
}
}' 'http://hostname:port/api/v1/workflows' -b /tmp/cookies.txt
```

An example response:

131

Get workflow by Id

Gets the workflow with the specified id.

• id: 1

An example request for getting workflow by id:

```
curl -X GET --header 'Accept: application/json' 'http://hostname:port/api/v1/

→workflows/id/1' -b /tmp/cookies.txt
```

An example response:

{ "id": 1,

```
"userId": 1,
"uuid": "3a3dfa34-bbd7-4c05-8745-55628d90cbf6",
"name": "Analyze Flights Delay",
"category": "Analytics",
"content": "{\"name\":\"Analyze Flights Delay\",\"uuid\":\"3a3dfa34-bbd7-4c05-8745-
→55628d90cbf6\",\"category\":\"Analytics\",\"description\":\"Find Flights which are
→delayed by more than 40 minutes.\", \"nodes\":[{\"id\":\"1\", \"name\":\
→ "DatasetStructured\", \"description\":\"This Node creates a DataFrame by reading.
→data from HDFS, HIVE etc. The dataset has been defined earlier in Fire by using the
\rightarrowDataset Feature. As a user, you just have to select the Dataset of your interest.\",
→\"details\":\"This Node creates a DataFrame by reading data from HDFS, HIVE etc.
→\\u003cbr\\u003cbr\\u003cbr\\u003cbr\\u003e\\nThe data has been defined earlier in Fire by_
\rightarrowusing the Dataset Feature. As a user, you just have to select the Dataset of your

winterest.\\u003cbr\\u003e\", \"examples\":\"\", \"type\":\"dataset\", \"nodeClass\":\

↔"fire.nodes.dataset.NodeDatasetStructured\", \"x\":\"38.9492px\", \"y\":\"275.613px\",
→\"fields\":[{\"name\":\"storageLevel\",\"value\":\"DEFAULT\",\"widget\":\"array\",\
→"title\":\"Output Storage Level\",\"description\":\"Storage Level of the Output_
→Datasets of this Node\",\"optionsArray\":[\"DEFAULT\",\"NONE\",\"DISK_ONLY\",\"DISK_
→ONLY_2\", \"MEMORY_ONLY\", \"MEMORY_ONLY_2\", \"MEMORY_ONLY_SER\", \"MEMORY_ONLY_SER_2\
→",\"MEMORY_AND_DISK\",\"MEMORY_AND_DISK_2\",\"MEMORY_AND_DISK_SER\",\"MEMORY_AND_
→DISK_SER_2\", \"OFF_HEAP\"], \"required\":false, \"display\":true, \"editable\":true, \
→ "disableRefresh\":false}, {\"name\":\"dataset\", \"value\":\"2ff32692-9b3c-49de-91a7-
→401daf2590c1\",\"widget\":\"dataset\",\"title\":\"Dataset\",\"description\":\
→"Selected Dataset\", \"required\":true, \"display\":true, \"editable\":true, \
→ "disableRefresh\":false}], \"engine\": \"scala\"}, {\"id\": \"2\", \"name\": \"PrintNRows\
→":\"transform\",\"nodeClass\":\"fire.nodes.util.NodePrintFirstNRows\",\"x\":\"38.
→4336px\",\"y\":\"59.1094px\",\"fields\":[{\"name\":\"storageLevel\",\"value\":\
→ "DEFAULT\", \"widget\":\"array\", \"title\":\"Output Storage Level\", \"description\":\
→"Storage Level of the Output Datasets of this Node\", \"optionsArray\":[\"DEFAULT\", \
→ "NONE\", \"DISK_ONLY\", \"DISK_ONLY_2\", \"MEMORY_ONLY\", \"MEMORY_ONLY_2\", \"MEMORY_QNLY_2\", \"MEMORY_QNLY_2\", \"MEMORY_QNLY_2\", \"MEMORY_QNLY_2\", \"MEMORY_QNLY_2\", \"MEMORY_QNLY_Z\", \"MEMORY_
→ONLY_SER\", \"MEMORY_ONLY_SER_2\", \"MEMORY_AND_DISK\", \"MEMORY_AND_DISK_2\", \"MEMORY_
→AND_DISK_SER\", \"MEMORY_AND_DISK_SER_2\", \"OFF_HEAP\"], \"required\":false, \"display\
→":true, \"editable \":true, \"disableRefresh \":false}, { \"name \": \"title \", \"value \": \
→ "display\":true, \"editable\":true, \"disableRefresh\":false}, {\"name\":\"n\", \"value\
→":\"10\",\"widget\":\"textfield\",\"title\":\"Num Rows to Print\",\"description\":\
→":true, \"disableRefresh\":false}], \"engine\": \"scala\"}, {\"id\": \"3\", \"name\": \
→"CastColumnType\", \"description\":\"This node creates a new DataFrame by casting.
→input columns with a new data type\",\"details\":\"\",\"examples\":\"\",\"type\":\
→"transform\",\"nodeClass\":\"fire.nodes.etl.NodeCastColumnType\",\"x\":\"313.223px\
→",\"y\":\"61.8633px\",\"fields\":[{\"name\":\"storageLevel\",\"value\":\"DEFAULT\",\
→"widget\":\"array\",\"title\":\"Output Storage Level\",\"description\":\"Storage_
→Level of the Output Datasets of this Node\", \"optionsArray\":[\"DEFAULT\", \"NONE\", \
→ "DISK_ONLY\", \"DISK_ONLY_2\", \"MEMORY_ONLY\", \"MEMORY_ONLY_2\", \"MEMORY_ONLY_SER\", \
→ "MEMORY_ONLY_SER_2\", \"MEMORY_AND_DISK\", \"MEMORY_AND_DISK_2\", \"MEMORY_AND_DISK_
→SER\",\"MEMORY_AND_DISK_SER_2\",\"OFF_HEAP\"],\"required\":false,\"display\":true,\
→"editable\":true,\"disableRefresh\":false}, {\"name\":\"inputCols\", \"value\":\"[\\\
→"CRS_DEP_TIME\\\",\\\"CRS_ARR_TIME\\\",\\\"CRS_ELAPSED_TIME\\\"]\",\"widget\":\
-- "variables\", \"title\": \"Columns\", \"description\": \"Columns to be cast to new data_
→type\", \"required\":false, \"display\":true, \"editable\":true, \"disableRefresh
→":false}, {\"name\":\"outputColType\", \"value\":\"DOUBLE\", \"widget\":\"array\", \
→"title\":\"New Data Type\",\"description\":\"New data type(INTEGER, DOUBLE, STRING,
→LONG, SHORT)\",\"optionsArray\":[\"BOOLEAN\",\"BYTE\",\"DATE\",\"DECIMAL\",\"DOUBLE\
↔", \"FLOAT\", \"INTEGER\", \"LONG\", \"SHORT\", \"STRING\", \"TIMESTAMP\"], \"required
 -: false, \"display\":true, \"editable\":true, \"disableRefresh\":false}, { (continues on next page)
→ "replaceExistingCols\", \"value\": \"true\", \"widget\": \"array\", \"title\": \"Replace...
    xisting Cols\",\"description\":\"Whether to replace ex
758nes\", \"optionsArray\":[\"true\", \"false\"], \"required hapters 29 \ "REST ARI's using curl
→"editable\":true,\"disableRefresh\":false}],\"engine\":\"scala\"}, {\"id\":\"4\",\
→"name\":\"CastColumnType\",\"description\":\"This node creates a new DataFrame by_
→casting input columns with a new data type\",\"details\":\"\",\"examples\":\"\",\
```

```
"description": "Find Flights which are delayed by more than 40 minutes.",
"version": 1,
"dateCreated": 1566551544583,
"dateLastUpdated": 1566551544583,
"lockedByUserId": null,
"permission": null,
"workflow": {
 "name": "Analyze Flights Delay",
 "uuid": "3a3dfa34-bbd7-4c05-8745-55628d90cbf6",
 "category": "Analytics",
 "description": "Find Flights which are delayed by more than 40 minutes.",
 "parameters": null,
 "nodes": [
   {
      "id": "1",
      "path": null,
      "name": "DatasetStructured",
      "iconImage": null,
      "description": "This Node creates a DataFrame by reading data from HDFS, HIVE_
→etc. The dataset has been defined earlier in Fire by using the Dataset Feature. As,
\rightarrowa user, you just have to select the Dataset of your interest.",
      "details": "This Node creates a DataFrame by reading data from HDFS, HIVE etc.
\rightarrow shows the data has been defined earlier in Fire by using the Dataset Feature.
\hookrightarrowAs a user, you just have to select the Dataset of your interest.<br/>dr>",
      "examples": "",
      "type": "dataset",
      "nodeClass": "fire.nodes.dataset.NodeDatasetStructured",
      "x": "38.9492px",
      "v": "275.613px",
      "fields": [
        {
          "name": "storageLevel",
          "value": "DEFAULT",
          "widget": "array",
          "title": "Output Storage Level",
          "description": "Storage Level of the Output Datasets of this Node",
          "optionsMap": null,
          "datatypes": null,
          "optionsArray": [
            "DEFAULT",
            "NONE",
            "DISK ONLY",
            "DISK ONLY 2",
            "MEMORY_ONLY",
            "MEMORY_ONLY_2",
            "MEMORY_ONLY_SER"
            "MEMORY_ONLY_SER_2",
            "MEMORY AND DISK",
            "MEMORY_AND_DISK_2",
            "MEMORY_AND_DISK_SER",
            "MEMORY_AND_DISK_SER_2",
            "OFF_HEAP"
          ],
          "required": false,
          "display": true,
          "editable": true,
          "disableRefresh": false
```

```
},
   {
     "name": "dataset",
     "value": "2ff32692-9b3c-49de-91a7-401daf2590c1",
     "widget": "dataset",
     "title": "Dataset",
     "description": "Selected Dataset",
     "optionsMap": null,
     "datatypes": null,
     "optionsArray": null,
     "required": true,
     "display": true,
     "editable": true,
     "disableRefresh": false
  }
],
 "engine": "scala"
},
```

Delete Workflow

Deletes a workflow with the given workflowId.

• workflowId: 1955

An example request for deleting workflow:

```
curl -X DELETE --header 'Accept: application/json' 'http://localhost:8080/api/v1/

↔workflows/id/1955' -b /tmp/cookies.txt
```

An example response:

```
Workflow deleted successfully.
```

Get Latest WorkFlows

Gets the latest workFlows available in the given application:

An example request for getting Latest WorkFlows availble in application:

An example response:

```
"id": 1954,
"userId": 3,
"uuid": "0e119cf1-2833-4c62-8466-21853fc1fb21",
"name": "aaaaawqw",
"category": "-",
"content": "{\"name\":\"aaaaawqw\",\"uuid\":\"0e119cf1-2833-4c62-8466-21853fc1fb21\",\
→"category\":\"-\",\"description\":\"1111\",\"parameters\":\"2222@1111\",\"nodes\":[
→{\"id\":\"1\",\"name\":\"ReadCSV\",\"description\":\"It reads in CSV files and
→",\"nodeClass\":\"fire.nodes.dataset.NodeDatasetCSV\",\"x\":\"243.5px\(cohlinubs'on\next page)
→"206px\",\"fields\":[{\"name\":\"storageLevel\",\"value\":\"DEFAULT\",\"widget\":\
760utput Datasets of this Node\", \"optionsArray\":[\"DEFAChapter 290 REST APL's using curl
→\"DISK_ONLY_2\",\"MEMORY_ONLY\",\"MEMORY_ONLY_2\",\"MEMORY_ONLY_SER\",\"MEMORY_ONLY_
→SER_2\", \"MEMORY_AND_DISK\", \"MEMORY_AND_DISK_2\", \"MEMORY_AND_DISK_SER\", \"MEMORY_
→AND_DISK_SER_2\",\"OFF_HEAP\"],\"required\":false,\"display\":true,\"editable\
                Dofmoob \ U.fo
                                ( \setminus \mathbb{I}_{p \rightarrow m} ) \setminus \mathbb{I}_{p \rightarrow m} 
  Intrana Alldiashl
```

```
"description": "1111",
"version": 4,
"dateCreated": 1566395460079,
"dateLastUpdated": 1566395644690,
"lockedByUserId": null,
"permission": null,
"workflow": {
 "name": "aaaaawqw",
 "uuid": "0e119cf1-2833-4c62-8466-21853fc1fb21",
 "category": "-",
 "description": "1111",
 "parameters": "2222@1111",
 "nodes": [
   {
      "id": "1",
      "path": null,
      "name": "ReadCSV",
      "iconImage": null,
      "description": "It reads in CSV files and creates a DataFrame from it",
      "details": "",
      "examples": "",
      "type": "dataset",
      "nodeClass": "fire.nodes.dataset.NodeDatasetCSV",
      "x": "243.5px",
      "y": "206px",
      "fields": [
        {
          "name": "storageLevel",
          "value": "DEFAULT",
          "widget": "array",
          "title": "Output Storage Level",
          "description": "Storage Level of the Output Datasets of this Node",
          "optionsMap": null,
          "datatypes": null,
          "optionsArray": [
            "DEFAULT",
            "NONE",
            "DISK_ONLY",
            "DISK_ONLY_2",
            "MEMORY_ONLY",
            "MEMORY ONLY 2",
            "MEMORY_ONLY_SER",
            "MEMORY_ONLY_SER_2",
            "MEMORY_AND_DISK",
            "MEMORY_AND_DISK_2"
            "MEMORY_AND_DISK_SER",
            "MEMORY_AND_DISK_SER_2",
            "OFF HEAP"
          ],
          "required": false,
          "display": true,
          "editable": true,
          "disableRefresh": false
        },
        {
          "name": "path",
          "value": "/user/sparkflows/Clickthru.csv",
```

```
"widget": "textfield",
  "title": "Path",
  "description": "Path of the Text file/directory",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": null,
  "required": true,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "separator",
  "value": ",",
  "widget": "textfield",
  "title": "Separator",
  "description": "CSV Separator",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": null,
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "header",
  "value": "true",
  "widget": "array",
  "title": "Header",
  "description": "Does the file have a header row",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": [
    "true",
    "false"
  ],
  "required": false,
  "display": true,
  "editable": true,
  "disableRefresh": false
},
{
  "name": "dropMalformed",
  "value": "false",
  "widget": "array",
  "title": "Drop Malformed",
  "description": "Whether to drop Malformed records or error",
  "optionsMap": null,
  "datatypes": null,
  "optionsArray": [
    "true",
    "false"
  ],
  "required": false,
  "display": true,
  "editable": true,
```

```
(continued from previous page)
```

```
"disableRefresh": false
    },
    {
      "name": "outputColNames",
      "value": "[\"Timestamp\",\"UserId\",\"IP Address\",\" Product Id\"]",
      "widget": "schema_col_names",
      "title": "Column Names for the CSV",
      "description": "New Output Columns of the SQL",
      "optionsMap": null,
      "datatypes": null,
      "optionsArray": null,
     "required": false,
     "display": true,
      "editable": true,
      "disableRefresh": false
    },
    {
      "name": "outputColTypes",
      "value": "[\"STRING\", \"INTEGER\", \"STRING\", \"INTEGER\"]",
      "widget": "schema_col_types",
      "title": "Column Types for the CSV",
      "description": "Data Type of the Output Columns",
      "optionsMap": null,
      "datatypes": null,
     "optionsArray": null,
     "required": false,
     "display": true,
      "editable": true,
      "disableRefresh": false
    },
    {
      "name": "outputColFormats",
      "value": "[\"\",\"\",\"\",\"\"]",
      "widget": "schema_col_formats",
      "title": "Column Formats for the CSV",
      "description": "Format of the Output Columns",
      "optionsMap": null,
     "datatypes": null,
     "optionsArray": null,
     "required": false,
      "display": true,
      "editable": true,
      "disableRefresh": false
    }
  ],
  "engine": "all"
},
```

Get Workflow Count

Gets the count of the workflows in the given application.

An example request for getting count of the Workflow:

An example response:

92

Get Workflow Versions

Gets the versions of workflow.

• workflowId: 1

An example request for getting workflow by id:

An example response:

```
Γ
 {
 "id": 1,
 "analysisflowId": 1,
 "content": "{\"name\":\"Analyze Flights Delay\", \"uuid\":\"3a3dfa34-bbd7-4c05-8745-
→55628d90cbf6\",\"category\":\"Analytics\",\"description\":\"Find Flights which are
→"DatasetStructured\", \"type\":\"dataset\", \"nodeClass\":\"fire.nodes.dataset.
→NodeDatasetStructured\",\"x\":\"38.9492px\",\"y\":\"275.613px\",\"fields\":[{\"name\
→":\"dataset\",\"value\":\"2ff32692-9b3c-49de-91a7-401daf2590c1\",\"widget\":\
→"dataset\", \"title\":\"Dataset\", \"description\":\"Selected Dataset\", \"required\
→",\"name\":\"PrintNRows\",\"description\":\"Prints the specified number of records_
→in the DataFrame\", \"type\":\"transform\", \"nodeClass\":\"fire.nodes.util.
→NodePrintFirstNRows\",\"x\":\"38.4336px\",\"y\":\"59.1094px\",\"fields\":[{\"name\
→":\"n\",\"value\":\"10\",\"widget\":\"textfield\",\"title\":\"Num Rows to Print\",\
→":false,\"editable\":true,\"disableRefresh\":false}]},{\"id\":\"3\",\"name\":\
→"CastColumnType\",\"description\":\"This node creates a new DataFrame by casting_
→input columns with a new data type\",\"type\":\"transform\",\"nodeClass\":\"fire.
→nodes.etl.NodeCastColumnType\",\"x\":\"313.223px\",\"y\":\"61.8633px\",\"fields\":[
→{\"name\":\"inputCols\",\"value\":\"[\\\"CRS_DEP_TIME\\\",\\\"CRS_ARR_TIME\\\",\\\
→"CRS_ELAPSED_TIME\\\"]\", \"widget\":\"variables\", \"title\":\"Columns\", \
→":\"New data type(INTEGER, DOUBLE, STRING, LONG, SHORT)\",\"optionsArray\":[\
→ "BOOLEAN\", \"BYTE\", \"DATE\", \"DOUBLE\", \"FLOAT\", \"INTEGER\", \"LONG\", \"SHORT\", \
→"STRING\", \"TIMESTAMP\"], \"required\":false, \"display\":false, \"editable\":true, \
→ "disableRefresh\":false}]}, {\"id\":\"4\", \"name\":\"CastColumnType\", \"description\
\rightarrow":\"This node creates a new DataFrame by casting input columns with a new data type
→",\"type\":\"transform\",\"nodeClass\":\"fire.nodes.etl.NodeCastColumnType\",\"x\":\
→"322.949px\",\"y\":\"275.633px\",\"fields\":[{\"name\":\"inputCols\",\"value\":\
→"Columns\",\"description\":\"Columns to be cast to new data type\",\"required\
→":false,\"display\":false, \"editable\":true, \"disableRefresh\":false}, {\"name\":\
→Type\",\"description\":\"New data type(INTEGER, DOUBLE, STRING, LONG, SHORT)\",\
→"optionsArray\":[\"BOOLEAN\",\"BYTE\",\"DATE\",\"DOUBLE\",\"FLOAT\",\"INTEGER\",\
→"LONG\", \"SHORT\", \"STRING\", \"TIMESTAMP\"], \"required\":false, \"display\":false, \
 →"editable\":true,\"disableRefresh\":false}]}, {\"id\":\"5\",\"name\":\"StringIndexer
→label indices\", \"type\":\"ml-transformer\", \"nodeClass\":\"fire.nodes.ml.
764

"handleInvalid\",\"x\":\"630.238px\",\"y\":\"272.879px\","fielde\".[]"name\":\"

"handleInvalid\",\"value\":\"skip\",\"widget\":\"array\",\"title\":\"Handle Invalid\
\rightarrow", \"description \": \"Invalid entries to be skipped or thrown error \", \"optionsArray \
→":[\"skip\",\"error\"],\"required\":false,\"display\":false,\"editable\":true,\

→ "disableRefresh\":false}, {\"name\":\"inputCols\", \"value\":\"[\\\"DAY_OF_MONTH\\\",
```

```
"dateLastUpdated": 1566551544603,
  "userName": null,
 "userId": null,
  "userComment": null
],
```

29.1.4 Workflow Execution REST API

Overview

}

The Workflow Execution REST API's, allow you to execute Workflows, get results etc.

Below are the various Workflow Execution API's available in Fire Insights, They should be executed after you have logged into Fire Insights.

List all the Executions

List all the workflow executions.

An example request for List all the executions:

```
curl -X GET --header 'Accept: application/json' 'http://hostname:port/api/v1/workflow-
→executions?page=0&size=1000' -b /tmp/cookies.txt
```

An example response:

```
Γ
{
"id": 135,
"analysisFlowId": 161,
"userId": 33,
"projectId": 33,
"analysisFlowScheduleId": null,
"status": 2,
"name": "Test_csv",
"category": "-",
"description": "Fired Manually",
"logs": "/tmp/fire/workflowlogs/workflow-5342148677548385044.log",
"fireJobId": "02aedbe5-0713-4172-9f7c-c63272f7cbd9",
"applicationId": "application_1560754639341_5932",
"uiWebUrl": "http://hostname:4042",
"metrics": null,
"startTime": 1566821007783,
"endTime": 1566821024075,
"emailOnSuccess": null,
"emailOnFailure": null
},
```

List Executions of a Workflow

Return the list of Executions for given workflowId. workflowId = 131:

An example request for List executions of a Workflow:

```
curl -X GET --header 'Accept: application/json' 'http://hostname:port/api/v1/workflow-

→executions/workflows/131' -b /tmp/cookies.txt
```

An example response:

```
[
{
"id": 99,
"analysisFlowId": 131,
"userId": 33,
"projectId": 33,
"analysisFlowScheduleId": null,
"status": 2,
"name": "Test_workflow",
 "category": "-",
"description": "Fired Manually",
"logs": "/tmp/fire/workflowlogs/workflow-4439919411814145818.log",
"fireJobId": "7b7b7dd5-b27b-419e-b853-794b5f53a5b8",
"applicationId": "application_1560754639341_5929",
"uiWebUrl": "http://hostname:4041",
"metrics": null,
"startTime": 1566795625424,
"endTime": 1566795650970,
"emailOnSuccess": null,
"emailOnFailure": null
],
```

GET Status of Workflow Execution

Return status of workflow execution for given workflowId.

workflowId = 193:

An example request for status of workflow execution

An example response:

KILLED

Stop the Execution of workflow

Stops the execution of workflow with specified workflowExecutionId.

Workflow Execution Id = 100:

An example request for Stopping specified workflow:

An example response:

{"status":"ok", "message":"Stopping Analysis Flow Execution"}

Kill the Execution of workflow

Kill the execution of workflow with specified workflowExecutionId.

Workflow Execution Id = 100:

An example request for Killing specified workflow:

An example response:

```
Killed YARN application : yarn application -kill application_1560754639341_5930,Exit_
→Value : 0
```

Delete Workflow Executions by days

Delete Workflow Executions by days

"days": "7":

An example request for deleting workflow executions by days:

An Example response:

Workflow executions deleted successfully

Get Executed Task Count

Get Executed Task Count:

An example request for Getting Executed Task Count:

An example response:

92

Get Latest Executions

Get Latest Executions:

An Example request for Getting Latest Executions:

An example response:

```
"id": 162,
"analysisFlowId": 131,
"userId": 33,
"projectId": 33,
"analysisFlowScheduleId": null,
"status": 2,
"name": "Test_workflow",
"category": "-",
"description": "Fired Manually",
"logs": "/tmp/fire/workflowlogs/workflow-3535160145732140945.log",
"fireJobId": "7b456feb-22fe-474e-a0c6-f31c40a1a9cd",
"applicationId": "application_1560754639341_5934",
"uiWebUrl": "http://hostname:4040",
"metrics": null,
"startTime": 1566834233892,
"endTime": 1566834262432,
"emailOnSuccess": null,
"emailOnFailure": null
},
```

29.1.5 Dashboard REST API

Overview

The Dashboards REST API's, allow you to interact with the Dashboards. Below are the various Dashboard API's available in Sparkflows They should be executed after you have logged into Sparkflows

Get List of Dashboards for the user

Returns the list of dashboards for the logged in user.

• Header: sortPara:asc/desc.

```
curl -i --header "Accept:application/json" -H "Content-Type:application/json" -H

→"sortPara:desc" -X GET -b /tmp/cookies.txt localhost:8080/dashboardsJSON
```

Create New Dashboard / Save Dashboard

Set dashboardId value null to create new dashboard:

```
curl - X POST --header 'Content-Type: application/json' --header 'Accept: text/plain'_

--header 'dashboardId: null' -d '{"category": "string", "description": "string",

--mame": "string", "sheets": [{"description": "string", "idx": "string", "items": [{

--mame": "string", "id": 0, "name": "string", "nodeId": "string", "type": "string

--mame": "string", "workflowName": "string", "x": "string", "y": "string"]],

--mame": "string", "type": "string"]], "uuid": "string"} ' 'http://localhost:8080/

--saveDashboard' -b /tmp/cookies.txt
```

Get Dashboard by Id

• id:1(Url Parameter)

Get dashboard results

- dashboardId:1
- sheetId:0

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8080/api/v1/

→dashboards/results?dashboardId=1&sheetId=0' -b /tmp/cookies.txt
```

update dashboard

- dashboardContent: abcd,
- dashboardId: 1,

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: */*' -d 'abcd

→' 'http://localhost:8080/api/v1/dashboards/1'
```

Delete Dashboard

- dashboardId: 1,
- projectId: 1,

```
curl -X DELETE --header 'Accept: text/plain' 'http://localhost:8080/api/v1/dashboards/

→1?projectId=1' -b /tmp/cookies.txt
```

29.1.6 HDFS REST API

Overview

The HDFS REST API's, allow you to interact with the HDFS of the Hadoop Cluster Sparkflows is connected to.

Below are the various HDFS API's available in Sparkflows

They should be executed after you have logged into Sparkflows

Get List of Files in Directory

Returns list of all the files on hdfs in the users home directory

curl -X GET --header 'Accept: application/json' 'http://localhost:8080/api/v1/hdfs'

Create HDFS directory

Get list of files in HDFS in the specified directory

Returns list of files in HDFS in the specified directory(/user/sparkflows/)

Get list of all the files on hdfs in the users home directory in sorted order

*sortPara: alphabetical

*path: /user/sparkflows/

Upload file

Deletes a file from HDFS

*path: /user/sparkflows/Airline.csv

Download HDFS file

*path: /user/sparkflows/Airline.csv

Rename HDFS File

*sourceFilePath: /user/sparkflows/Airline.csv

*destinationFilePath: /user/sparkflows/airline.csv

```
curl -X GET --header 'Accept: text/plain' 'http://localhost:8080/api/v1/hdfs/files/

→rename?sourceFilePath=%2Fuser%2Fsparkflows%2FAirline.csv&destinationFilePath=%2Fuser

→%2Fsparkflows%2Fairline.csv'
```

Get first X bytes of content of a file

*path: /user/sparkflows/Airline.csv

```
curl -X GET --header 'Accept: text/plain' 'http://localhost:8080/api/v1/hdfs/files/
→open?path=%2Fuser%2Fsparkflows%2FAirline.csv'
```

29.1.7 HIVE REST API

Overview

The HIVE REST API's, allow you to interact with the HIVE of the Hadoop Cluster Sparkflows is connected to.

Below are the various HIVE REST API's available in Sparkflows

They should be executed after you have logged into Sparkflows

Get all Hive Databases

Get Table for a given Database

- "db": "default",
- "table": "sample_07"

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8080/api/v1/hive/

→tables?db=default&table=sample_07' -b /tmp/cookies.txt
```

Get all Hive Databases

29.1.8 Scheduler REST API

Overview

The Scheduler REST API's, allow you to schedule some jobs once Sparkflows connected to Hadoop Cluster.

Below are the various Scheduler REST API's available in Sparkflows

They should be executed after you have logged into Sparkflows

Get list of all Workflows Scheduled

• analysisflowId = 1

Schedule new Workflow

curl:

JSON:

```
"analysisFlowId": 0,
 "cronPattern": "string",
"dateCreated": "2019-08-06T11:77:17.2212",
 "dateLastUpdated": "2019-08-06T11:77:17.221Z",
 "day": "string",
 "dayOfTheMonth": 0,
 "description": "string",
 "emailonFailure":"string",
 "emailonSuccess": "string"
 "endTime": "2019-08-06T11:77:17.221Z",
"fireEvery": "string",
 "firedTime": "2019-08-06T11:77:17.221Z",
"hour": 0,
 "id": 0,
"Libjars": "string",
"minute": 0,
"name": "string",
"sparkSubmitOptions": "string",
"startTime": "22019-08-06T11:77:17.221Z",
"userId": "0",
"id": "string",
} '
     'http://137.117.83.79:8080/api/v1/workflow-schedules' -b /tmp/cookies.txt
```

Delete Scheduled Workflow

It deletes a scheduled instance of a workflow:

```
curl -X GET --header 'Accept: application/json' --header 'id: 1' 'http://
→localhost:8080/api/v1/workflow-schedules/1' -b /tmp/cookies.txt
```

CHAPTER 30

Third Party Acknowledgements

30.1 Third Party Acknowledgements

Sparkflows uses and distributes the following third party software. These are open source software licensed as mentioned.

30.1.1 Server Libraries

- Apache Spark
 - https://spark.apache.org/
 - Copyright © 2018 The Apache Software Foundation
 - License: Apache-2.0
- Apache Avro
 - https://avro.apache.org/
 - Copyright © 2012 The Apache Software Foundation
 - License: Apache-2.0
- Apache Commons
 - https://commons.apache.org/
 - Copyright © 2019 The Apache Software Foundation.
 - License: Apache-2.0
- Apache Hadoop
 - https://hadoop.apache.org/
 - Copyright © 2018 The Apache Software Foundation.
 - License: Apache-2.0

- Apache HBase
 - https://hbase.apache.org/
 - Copyright ©2007–2019 The Apache Software Foundation
 - License: Apache-2.0
- Apache Hive
 - https://hive.apache.org/
 - Copyright © 2011-2014 The Apache Software Foundation
 - License: Apache-2.0
- Apache HTTP
 - https://hc.apache.org/
 - Copyright © 1999–2019 The Apache Software Foundation.
 - License: Apache-2.0
- spark-streaming-kafka
 - http://spark.apache.org/
 - © 2017 Apache Software Foundation
 - License: Apache-2.0
- Apache pdfbox
 - https://pdfbox.apache.org
 - Copyright © 2009–2019 The Apache Software Foundation
 - License: Apache-2.0
- Apache OpenNLP
 - https://opennlp.apache.org/
 - Copyright © 2017 The Apache Software Foundation
 - License: Apache-2.0
- Apache Tika
 - https://tika.apache.org/
 - Copyright © 2019 The Apache Software Foundation
 - License: Apache-2.0
- Apache Tomcat
 - http://tomcat.apache.org/
 - Copyright © 1999-2019, The Apache Software Foundation
 - License: Apache-2.0
- AWS Java SDK
 - https://aws.amazon.com/
 - Copyright © 2019, Amazon Web Services, Inc. or its affiliates
 - License: Apache-2.0

- Eclipse jetty
 - https://www.eclipse.org/jetty/
 - Copyright © 2016 The Eclipse Foundation.
 - License: EPL- v 2.0
- Elasticsearch-spark-20_2.11
 - https://github.com/elastic/elasticsearch-hadoop
 - © 2019. Elasticsearch B.V.
 - License: Apache-2.0
- Guava
 - https://github.com/google/guava
 - https://github.com/google/guava/blob/master/COPYING
 - License: Apache-2.0
- H2O
 - https://www.h2o.ai/
 - © Copyright 2013, 0xdata, Inc.
 - License: Apache-2.0
- Json Java
 - http://www.json.org
 - Copyright (c) 2002 JSON.org
 - License: BSD-style with "no evil" clause
- Log4J
 - http://logging.apache.org/log4j/2.x/
 - Author: The Apache Software Foundation
 - License: Apache-2.0
- Sagemaker-spark_2.11
 - https://github.com/aws/sagemaker-spark
 - Author: The Apache Software Foundation
 - License: Apache-2.0
- Mongo_spark_connector_2.11
 - http://github.com/mongo-spark
 - Author: The Apache Software Foundation
 - License: Apache-2.0
- Python
 - https://www.python.org/
 - Copyright ©2001-2019. Python Software Foundation
 - License: PSFL2

- Quartz
 - http://www.quartz-scheduler.org/
 - Copyright© Terracotta, Inc., a wholly-owned subsidiary of Software AG USA, Inc. All rights reserved
 - License: Apache-2.0
- Spring Framework
 - https://spring.io/
 - Copyright © 2019 Pivotal Software, Inc. All Rights Reserved
 - License: Apache-2.0
- SLF4J
 - http://www.slf4j.org/
 - Copyright (c) 2004-2017 QOS.ch
 - License: MIT

30.1.2 Frontend Libraries

- angularjs
 - https://angularjs.org/
 - Copyright (c) 2010-2014 Google, Inc.
 - License: MIT
- bootstrap
 - http://getbootstrap.com/2.3.2/
 - Copyright 2011-2014 Twitter, Inc
 - License: MIT
- jquery
 - https://jquery.com/
 - Copyright 2019 The jQuery Foundation. jQuery License
 - License: MIT
- rxjs
 - https://rxjs-dev.firebaseapp.com/
 - Copyright 2015-2018 Google, Inc., Netflix, Microsoft Corp.
 - License: Apache License 2.0

30.1.3 Definitions

- Apache-2.0 : Apache License, Version 2.0 : http://www.apache.org/licenses/LICENSE-2.0.html
- MIT : MIT License : https://en.wikipedia.org/wiki/MIT_License#Relation_to_Patents
- BSD-style: BSD-style License : http://json.org/license.html
- EPL: EPL v 2.0 License: https://www.eclipse.org/legal/epl-2.0/

• PSFL2 : Python Software Foundation License Version 2

CHAPTER 31

Indices and tables

- genindex
- modindex
- search