Data Science full-stack developer – Case Study

As a full stack Data science engineer I would like to build/test & support large-scale data processing systems and be an expert in the latest Big Data, NoSQL, advanced analytic tools & AI concepts. Here I will be demonstrating a complete data application lifecycle – data ingestion, modeling, processing & prediction solutions using AI techniques (Deep Learning/Machine Learning) to meet business needs of the organization. From my experienced I came up with the following problem solving solution.

Use-Case: Airline on-time performance

Have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted it if you'd had more data?

Well as a Data Science Engineer I would like to come up with the solution by analyzing all the existing data (flight arrival and departure information from October 1987 to April 2008) and streaming real time data and come up with prediction and updated our customer with the real time data, so customer can benefitted from it and avoid all the consciences.

The Framework Implementations:

For implementing the framework I have used the following lists of tools:

OS – Mac
Bigdata tool -

- Spark 2.1.1
- Kafka 1.0.0
- Flume 1.8.0
- Hadoop 2.7.2
- Pig 0.17.0
- Zeppelin 0.7.1
- R Studio 1.1.423

Step by Step guide to install and configured these tools:

1.1 Install Spark

Here is the official site for downloading Spark

- https://spark.apache.org/downloads.html
- tar –xvzf spark-2.2.1-bin-hadoop2.7.tgz

Start Spark Master and slave by using below command

sbin/start-master.sh -h 127.0.0.1 -p 7077
sbin/start-slave.sh spark://127.0.0.1:7077

1.2 Install kafka

Here is the official site for downloading Kafka

- https://kafka.apache.org/
- http://redrockdigimark.com/apachemirror/kafka/1.0.0/kafka_2.11-1.0.0.tgz
- tar -xvzf kafka_2.11-1.0.0.tgz

Start Zookeeper server and Kafka Server

bin/zookeeper-server-start.sh config/zookeeper.properties
bin/kafka-server-start.sh config/server.properties

Create Topic in Kafka

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic Airports
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic Carriers
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic Planedate
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic OTP

Read the source data in topics

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Airports < /Users/cruise/Documents/developer/emirates/airports.csv
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Planedate < /Users/cruise/Documents/developer/emirates/plane-data.csv
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Carriers < /Users/cruise/Documents/developer/emirates/carriers.csv

Get data into consumer

bin/kafka-console-consumer.sh --zookeeper localhost:2181 --bootstrap-server localhost:9092 --topic Airports --from-beginning
bin/kafka-console-consumer.sh --zookeeper localhost:2181 --bootstrap-server localhost:9092 --topic Planedate --from-beginning
bin/kafka-console-consumer.sh --zookeeper localhost:2181 --bootstrap-server localhost:9092 --topic Carriers --from-beginning

1.3 Install Flume

Here is the official site for downloading Flume

- https://flume.apache.org/download.html
- apache-flume-1.8.0-bin.tar.gz

- tar -xvzf apache-flume-1.8.0-bin.tar.gz

create conf file and put it in conf directory of flume

bin/flume-ng agent --conf conf --conf-file conf/Airports.conf --name flume1 -Dflume.root.logger=INFO,console

bin/flume-ng agent --conf conf --conf-file conf/Planedate.conf --name flume1 -Dflume.root.logger=INFO,console

1.4 Install Hadoop

Here is the official site for downloading Hadoop

- http://hadoop.apache.org/releases.html

After making change in conf file (core-site.xml, hdfs-site.xml, hadoop-env.sh) of hadoop, start namenode and datanode by using below command

sbin/start-dfs.sh

bin/hdfs dfs -mkdir -p /data/raw/
bin/hdfs dfs -mkdir -p /data/decomposed/
bin/hdfs dfs -mkdir -p /data/modelled/
bin/hdfs dfs -mkdir -p /data/schema/

1.5 Install Pig

Here is the official site for downloading Pig

- http://www-eu.apache.org/dist/pig/pig-0.17.0/

- download pig-0.17.0.tar.gz
- tar -xvzf pig-0.17.0.tar.gz

Created pig script and execute with below command

bin/pig -x local /Users/cruise/Documents/developer/emirates/add_uuid_timestamp.pig

1.6 **Install Zeppelin**

Here is the official site for downloading Zeppelin

- http://www.apache.org/dyn/closer.cgi/zeppelin/zeppelin-0.7.3/zeppelin-0.7.3-bin-all.tgz
- tar -xvzf **zeppelin-0.7.3-bin-all.tgz**

Start Zeppelin by using below command

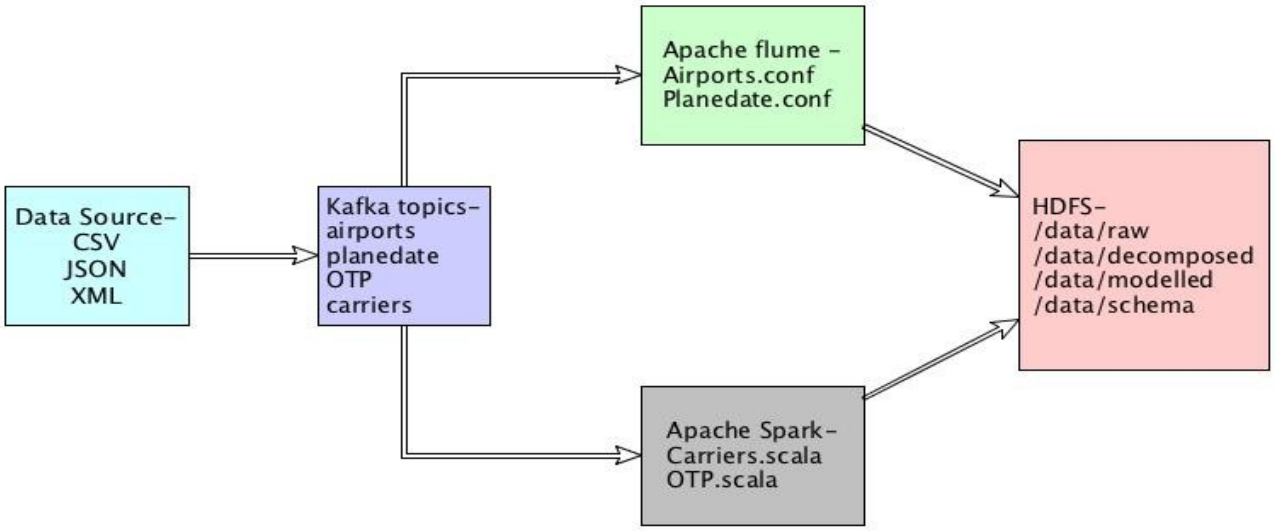
bin/zeppelin-daemon.sh start

1.7 **Install Rstudio**

Donwload Binary package of R and Rstudio

- https://www.rstudio.com/products/rstudio/download/

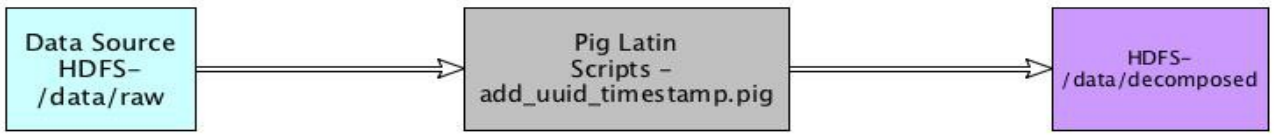
Data Pipeline:



Implementation of the data Pipeline:

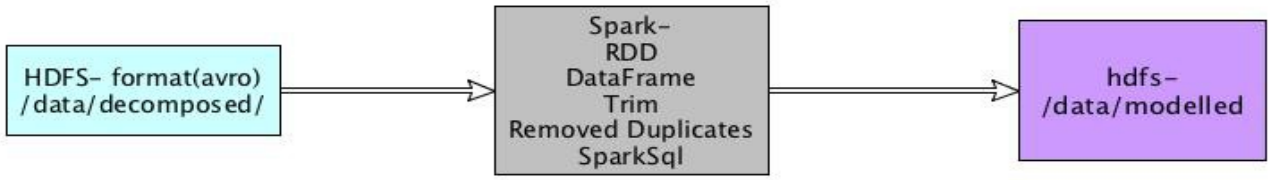
- 1st of all I read the data from the csv file, which is proved in the case study.
- Next I have ported those data into the Kafka topics
- From topics Airport and planedate I have read the data through Apache Flume and save the data into HDFS under the directory /data/raw
- From topics OTP and Carriers I have read the data using Apache Spark(Spark Streaming) and save those data into HDFS under the directory /data/raw

Data decomposition:



For adding timestamp and UUID, I have used Pig Latin Scripts
In which, I've read the contain from HDFS data/raw directory and those decomposed data I have added back to HDFS under the directory /data/decomposed in AVRO format.

Data Modeling and Processing:



- In data modeling, I have read the data from data/decomposed directory and convert the data into data frame by using the package com.databricks.spark.avro
- Next I trimmed those data, remove special characters and null
- For removing duplicate data I have used dataframe.distinct and save the refined data into HDFS into parquet format

Develop a solution using R/Python/Spark MLlib tools and Big Data platform to answer the following questions. Demonstrate your expertise in each tools.

· **Which carrier performs better?**

Based on the data following are the carriers perform better.

See more details in zeppelin notebook. It's included in github directory.

1. Delays Chart based upon Carriers

FINISHED

```
%sql
select c.description,sum((d.ArrDelay + d.DepDelay)) as delay from data_2007_2008 d join carriers c on c.code = d.UniqueCarrier group by c.description order by delay
    desc
```

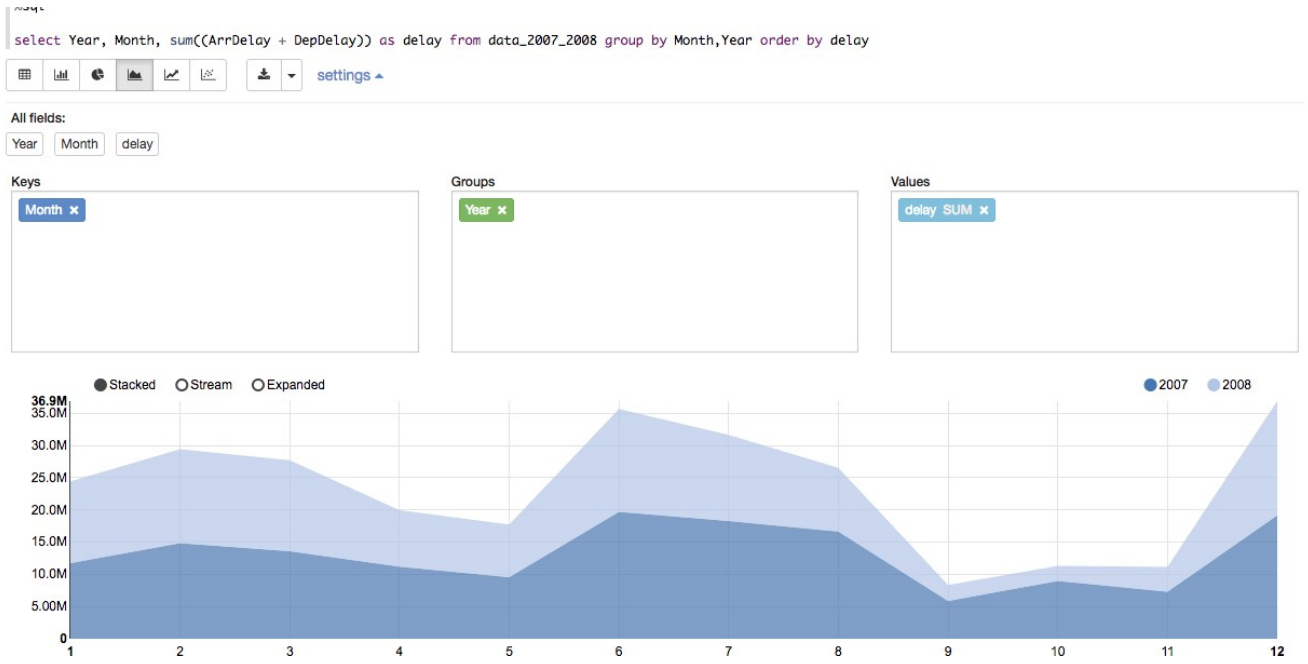
description	delay
Southwest Airlines Co.	3.6758695E7
American Airlines Inc.	3.2996502E7
United Air Lines Inc.	2.3811063E7
American Eagle Airlines Inc.	2.3155757E7
Skywest Airlines Inc.	1.7734136E7
Expressjet Airlines Inc.	1.7204505E7
Atlantic Southeast Airlines	1.6447223E7
US Airways Inc. (Merged with America West 9/05. Reporting for both starting 10/07.)	1.5042951E7
Continental Air Lines Inc.	1.4117233E7

Took 2 min 29 sec. Last updated by cruise at February 24 2018, 10:21:12 PM. (outdated)

· When is the best time of day/day of week/time of year to fly to minimize delays?

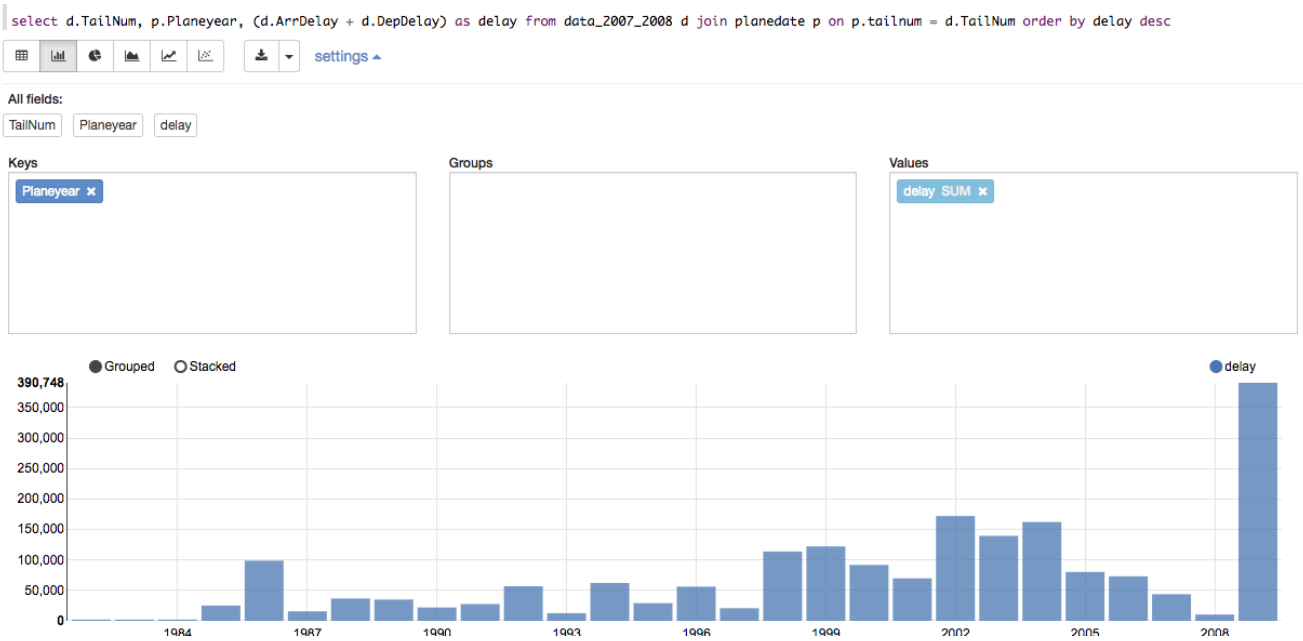
Following are the best time of year to fly to minimize delays.

You can find more details in zeppelin notebook.



· **Do older planes suffer more delays?**

By analyzing the following data you cannot say that older planes suffer more delay.



· **Create a model to predict flight delays**

To predict flight delay I have created a data model in Rstudio using SparkR and included this in github repository.

· **How well does weather predict plane delays?**

I wasn't able to create data model to predict plane delays seems OTP data wasn't available.

For complete code, please visit github link

Repo link-
https://github.com/Chetan8256/assignment

Repo zip link-
https://github.com/Chetan8256/assignment/archive/master.zip

Repo clone link
https://github.com/Chetan8256/assignment.git

Note: For this project, I did not use any cluster tool like (**Hortonworks**, C**loudera**).
However I have experience in working on Hortonworks and cloudera.
I have also worked on complete projects in **Python** using **Pyspark** and pandas.