



# **INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT**

**AKURDI, PUNE**

Documentation On

## **“Flight Delay Analysis using Apache Spark and Building Flight Delay Prediction Machine Learning Model”**

PG-eDBDA SEPT 2020-21

Submitted By:

**Group No: 06**

**Sandeep Bayas - 1508**

**Nipun Jaiswal– 1520**

**Project Guide:**

**Mr. Akshay Tilekar (External Guide)**

**Mr. Rahul (Internal Guide)**

**Mr. Manish (Internal Guide)**

**Mr. Prashant Karhale**

**Centre Coordinator**

# INDEX

## Table of Contents

Contents	Page no
<b>Acknowledgement</b>	3
<b>Introduction</b>	4
<b>Purpose</b>	4
<b>Scope</b>	6
<b>Proposed Methodology for Machine Learning Model</b>	7
<b>Big Data Methodology</b>	10
<b>Overall Description</b>	12
<b>Data Collection</b>	14
<b>Data Processing and Data Visualization</b>	15
<b>System Design</b>	26
<b>Model Building</b>	29
<b>Results</b>	41
<b>Future Scope</b>	42
<b>Conclusion</b>	42
<b>References</b>	43

## **Acknowledgement**

This work could not have been completed without the guidance and encouragement of many people. We would like to particularly acknowledge those below.

We pay our humble regards and gratitude to Prof. Prashant Karhale, Centre Coordinator IACSD CDAC for guiding us and giving moral support and timely boost.

We wish to express our special thanks to project guides Mr. Rahul, Mr. Manish, and Mr. Akshay Tilekar project evaluators, who helped us a lot in the preparation of our project.

## CHAPTER 1

# INTRODUCTION

Delay is one of the most remembered performance indicators of any transportation system.

Notably, commercial aviation players understand delay as the period by which a flight is late

or postponed. Thus, a delay may be represented by the difference between scheduled and actual times of departure or arrival of a plane.

Flight delays have negative impacts, mainly economic, for passengers, airlines, and airports.

Given the uncertainty of their occurrence, passengers usually plan to travel many hours earlier for their appointments, increasing their trip costs, to ensure their arrival on time. Furthermore, from the sustainability point of view, delays may also cause environmental damage by increasing fuel consumption and gas emissions.

Delays also relies on airlines marketing strategies, since carriers rely on customers' loyalty

to support their frequent-flyer programs and the consumer's choice is also affected by reliable performance.

ARRIVALS				
AIRLINE	FLIGHT	ORIGIN	STD	STA
IC	932	Dubai	05:00	06:06
5S	481	Mumbai	06:15	05:59
GB	341	Delhi	06:20	06:22
IC	564	Shanghai	06:30	
9W	537	Bangalore	07:25	08:10
52	493 X	Chennai	07:25	07:13
IT	431	Bangalore	07:30	07:43
6E	151	Jalpur	07:40	07:34
9W	453	Mumbai	07:40	07:57
IT	161	Mumbai	07:45	08:02
9W	827	Delhi	07:50	08:05
IT	463	Chennai	08:00	08:00
IT	2472	Chennai	08:30	08:30
6E	301	Delhi	08:35	08:44
IT	434	Kolkata	08:35	08:31
IC	940	Delhi	08:40	08:23
EK	526	Dubai	08:55	09:15
DR	617	Bangalore	09:00	08:45
6E	351	Bangalore	09:05	08:58
IC	945	Chennai	09:10	08:59
9W	457	Mumbai	09:15	08:59
5N	878	Delhi	09:15	09:07
IC	615	Mumbai	09:15	08:59
IT	2812	Rajpur	09:30	09:50
SV	728	Jeddah-Riyadh	09:30	09:41
SG	226	Vishakhapatnam	09:45	09:38

DEPARTURES				
AIRLINE	FLIGHT	DESTINATION	STD	STA
52	261	Bangalore	06:30	25F
GB	342	Delhi	06:55	22
IC	937	Mumbai	07:50	30A
9W	518	Bangalore	07:55	30B
52	484	XChennai	08:00	
IT	431	GUWAHATI	08:05	24
9W	454	Mumbai	08:15	26
6E	151	Cochin	08:20	25B
DR	406	Bangalore	08:20	25A
IC	735	Bangkok	08:20	33A
9W	828	Delhi	08:25	28
IT	162	Mumbai	08:25	22
IT	463	Vishakhapatnam	08:30	25C
IT	2473	Chennai	09:00	25C
6E	302	Delhi	09:10	26
IT	434	Bangalore	09:20	25D
DR	417	Rajmundry	09:35	25A
IC	916	Bangalore	09:40	30B
6E	351	Kolkata	09:50	25E
DR	678	Chennai	09:50	24
9W	458	Mumbai	10:00	25B
IT	2831	Vijayawada	10:00	25C
IC	946	Chennai	10:10	30A
5G	226	Delhi	10:15	22
EK	527	Dubai	10:20	32A
IC	616	Mumbai	10:30	28

HAND BAGGAGE ONLY  
DOMESTIC : B INTERNATIONAL : G

## **1.1 Purpose**

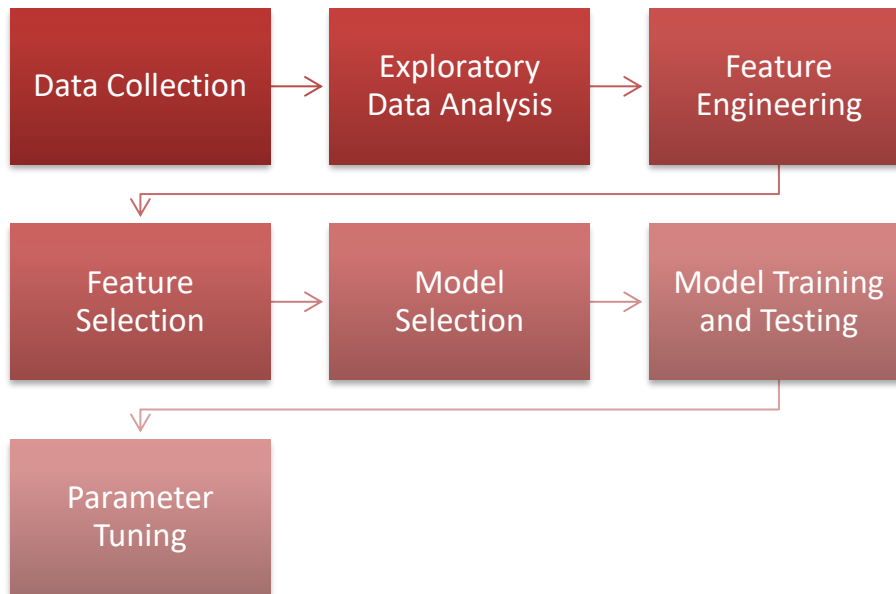
So the primary goal of this project is to collect and analyze the data. Provide insights from data to Business Stakeholders and help them for making a well-assessed decision to increase operation efficiency and growth in business. Alongside building a predictive machine learning model to reduce inconvenience happens to passengers while travelling and inform the passenger by how much time is flight being delayed or not.

## **1.2 Scope**

### **Initial functional requirement will be:-**

- Understanding the data to get the proper Insights.
- Selecting the algorithm meeting requirement.
- Parameter tuning of the algorithm.
- Cluster setup for data storage and data processing
- Visualization of data for better understanding.

### 1.3 Proposed Methodology for Machine Learning Model



#### 1.3.1 Proposed Methodology Model Gathering Data:

Data Gathering is the first step of the machine learning life cycle.

In this step, we need to identify the different data sources, as data can be collected from various sources such as files, database, internet, or mobile devices. It is one of the most important steps of the methodology. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction. We also need to collect the data from authenticated sites so that we get the original dataset.

### **1.3.2 Feature Engineering: -**

#### **Exploratory Data Analysis:**

Data exploration:

It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data. A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.

Data pre-processing:

Here we need to find the missing values or is there any Na value present or any invalid value. Putting NA where there is missing values. Remove unnecessary columns (as per the requirement).Format or Binarize the target or unknown variable. Check data types, handle columns with Object / string type (either converting to number / removing them).Removing columns which have all missing / NA values. Removing rows with missing / NA values greater than 50%.Filling missing / NA values using central tendencies.

So, we use various filtering techniques to clean the data. It is mandatory to detect and remove the above issues because it can affect the quality of the outcome.

#### **Feature Selection:**

Now the cleaned and prepared data is passed on to feature selection step. This step involves selection of analytical techniques.

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification, Regression, Cluster analysis etc. then build the model using prepared data, and evaluate the model. The two main type of feature selection is supervised and unsupervised methods. This step also helps to predict the target variable

Hence, in this step, we take the data and use machine learning algorithms to build the model.



## **Model Selection:**

Model Selection is the process of selecting one final model for machine learning model among the various present out there. It provides conceptual framework for determining a good model. If we are in a data-rich situation, the best approach is to randomly divide the dataset into three parts: a training set, a validation set, and a test set. The training set is used to fit the models; the validation set is used to estimate prediction error for model selection; the test set is used for assessment of the generalization error of the final chosen model.

## **Model Training and Testing:**

### **Training Model:**

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem. We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

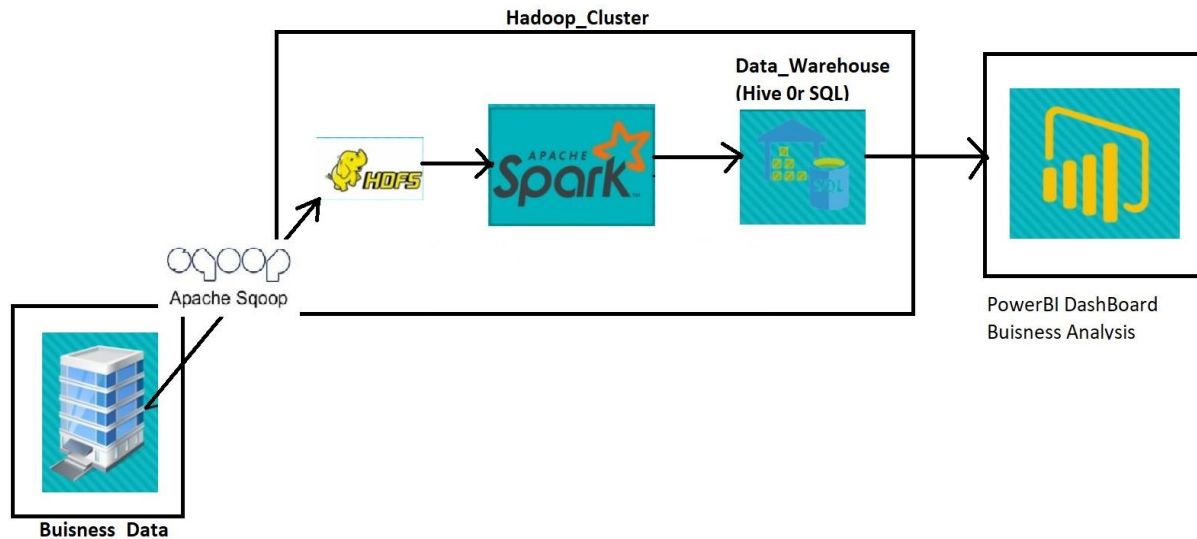
### **Testing Model:**

Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project .

### **Parameter Tuning:**

It is the process of selecting the optimal or best hyper parameters for a learning algorithm. It also helps to optimize the performance of the model. Hyper parameters are important because they directly control the behaviour of the training algorithm and have a significant impact on the performance of the model is being trained. A hyper parameter is a hyper parameters whose value is used to control the learning process and increase the model performance by choosing best parameters .Some example of hyper parameters are number of leaves or depth of a tree, or max split required ,criteria as well. It can be decided by setting different values, training different models and choosing the values that test better.

## BIG Data Methodology:



## Data Collection:

Collected data from RDBMS system we are going to store it into simple storage service called as s3 bucket. We using this space as our staging area for data and from this point we fetching it into our cluster

## Data Processing:

Once the data is fetched from staging area it stored into memory of a cluster and ready for starting the analysis and processing. We are using spark\_sql for processing the data. In it first we create data frame which holds our data and after creating table view for that data frame we performing sql queries on that table.

**Storing the results:**

Output of the sql analysis is going to stored into another sql server data warehouse from where we can connect our visualization tools to warehouse and doing the job visualization.

**Data Visualization:**

For doing the visualization of our processed data. Here we are using powerbi for connecting to our data warehouse and fetching the data from it. Once the data is fetched by powerbi from warehouse we are proceeding for visualization and creation of dashboards for better representation and understanding of data.

## CHAPTER 2

### OVERALL DESCRIPTION

Our aim from the project is to make use of pandas, matplotlib, & seaborn libraries from python to extract the meaningful insights from the data and use classifier models like Random forest, Decision Trees libraries for machine learning. We also used Hadoop ecosystem tools such as Sqoop, Pyspark, Sql, data warehousing and PowerBi for storage processing and visualization.

Secondly, to learn how to hyper tune the parameters using grid search for every machine learning model.

And in the end, to visualize the data.

#### Attributes in the dataset

FL\_DATE = Date of the Flight

OP\_CARRIER = Airline Identifier

OP\_CARRIER\_FL\_NUM = Flight Number

ORIGIN = Starting Airport Code

DEST = Destination Airport Code

CRS\_DEP\_TIME = Planned Departure Time

DEP\_TIME = Actual Departure Time

DEP\_DELAY = Total Delay on Departure in minutes

TAXI\_OUT = The time duration elapsed between departure from the origin airport gate and wheels off

WHEELS\_OFF = The time point that the aircraft's wheels leave the ground

WHEELS\_ON = The time point that the aircraft's wheels touch on the ground

TAXI\_IN = The time duration elapsed between wheels-on and gate arrival at the destination airport

CRS\_ARR\_TIME = Planned arrival time

ARR\_TIME = Actual Arrival Time = ARRIVAL\_TIME - SCHEDULED\_ARRIVAL

ARR\_DELAY = Total Delay on Arrival in minutes

CANCELLED = Flight Cancelled (1 = cancelled)

CANCELLATION\_CODE = Reason for Cancellation of flight: A - Airline/Carrier; B - Weather; C -National Air System; D - Security

DIVERTED = Aircraft landed on different airport than the one scheduled(1 = Diverted)

CRS\_ELAPSED\_TIME = Planned time amount needed for the flight trip

ACTUAL\_ELAPSED\_TIME = AIR\_TIME+TAXI\_IN+TAXI\_OUT

AIR\_TIME = The time duration between wheels off and wheels on time  
DISTANCE = Distance between two airports  
CARRIER\_DELAY = Delay caused by the airline in minutes  
WEATHER\_DELAY = Delay caused by weather  
NAS\_DELAY = Delay caused by air system  
SECURITY\_DELAY = caused by security reasons  
LATE\_AIRCRAFT\_DELAY = Delay caused by security\.

## **2.1 Data Collection:**

The dataset come from the website of bureau of transportation and statistics United States ([www.bts.gov](http://www.bts.gov)) OR you can get it on kaggle also. The dataset provide information about flight records per airline across United States. which include 28 features and around 57.6 million rows.

## **2.2 Functional requirement specification:**

**Use case: - Prediction of Delay of flight**

### **Brief Description:**

Understanding the problem statement is the first and foremost step. This would help you give an intuition of what you will face ahead of time. Let us see the problem statement .It is a classification problem where we have to predict whether the flight is delayed or not. In a classification problem, we have to predict discrete values based on a given set of independent variables. And for the regression problem we have to predict the actual time by which the flight is delayed i.e the continuous variable.

## CHAPTER 3

### Data Processing and Data Visualization

#### Data Processing:-

The preview of data:-

Airline Data 2017 contains about 56 million rows and 28 features.

	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	...
0	2017-01-01	AA	1	JFK	LAX	800	831.0	31.0	25.0	856.0	...
1	2017-01-01	AA	2	LAX	JFK	900	934.0	34.0	34.0	1008.0	...
2	2017-01-01	AA	4	LAX	JFK	1130	1221.0	51.0	20.0	1241.0	...
3	2017-01-01	AA	5	DFW	HNL	1135	1252.0	77.0	19.0	1311.0	...
4	2017-01-01	AA	6	OGG	DFW	1855	1855.0	0.0	16.0	1911.0	...

## Descriptive Statistics:

Here we can see that FL\_DATE variable is in Object format, so while loading the data we have to take care of the date variable, it should be in python date time format. Further we can see that some of other features are also in object format so we need to remove them as they are not important for analysis.

```
FL_DATE          object
OP_CARRIER      object
OP_CARRIER_FL_NUM  int64
ORIGIN           object
DEST            object
CRS_DEP_TIME     int64
DEP_TIME        float64
DEP_DELAY        float64
TAXI_OUT         float64
WHEELS_OFF       float64
WHEELS_ON        float64
TAXI_IN          float64
CRS_ARR_TIME     int64
ARR_TIME         float64
ARR_DELAY        float64
CANCELLED        float64
CANCELLATION_CODE object
DIVERTED         float64
CRS_ELAPSED_TIME float64
ACTUAL_ELAPSED_TIME float64
AIR_TIME         float64
DISTANCE         float64
CARRIER_DELAY   float64
WEATHER_DELAY    float64
NAS_DELAY        float64
SECURITY_DELAY   float64
LATE_AIRCRAFT_DELAY float64
Unnamed: 27      float64
dtype: object
```



## Target Feature 1

ARR\_DELAY is our Target Feature. which gives information like flight is delayed or it arrived before scheduled arrival time and if it delayed then by how much time and arrived early then by how much time.

## Target Feature 2

We are creating another target variable (Binary Classification) from ARR\_DELAY  
If ARR\_DELAY having value equal to 0 or less than 0 then there is not delayed flight =0  
If ARR\_DELAY having value greater than 0 then it consider to be a delayed flight =1

```
delay = []

for i in df['ARR_DELAY']:
    if i <= 0:
        delay.append(0) # No delay
    else:
        delay.append(1) #delay
df['Target'] = delay
```

Here you can observed that we have created a new column that is target variable and cross check with arrival delay column.

	<b>ARR_DELAY</b>	<b>Target</b>
<b>0</b>	27.0	1
<b>1</b>	42.0	1
<b>2</b>	42.0	1
<b>3</b>	97.0	1
<b>4</b>	42.0	1
<b>5</b>	394.0	1
<b>6</b>	58.0	1
<b>7</b>	-22.0	0
<b>8</b>	-30.0	0
<b>9</b>	-24.0	0

## Dealing With Null Values and Object Type Data

There are total 4 features which have object data type

- 1)OP\_CARRIER
- 2)ORIGIN
- 3)DEST
- 4)CANCELLATION\_CODE

We are going to exclude those feature because, like CANCELLATION\_CODE is for the flights who get cancelled so there is no sense to predict the delay for already cancelled flight. and we won't found any impactful relation of other 3 features with target variable that's why we decided to drop those features.

```
df = df.select_dtypes(exclude=['object'])
```

Checking is null column which are having all null values. We found that we have null column named Unnamed: 27 (which comes to be True).So we are dropping the column Unnamed: 27

```
#Checking columns which are having all null values.  
df.isnull().all()
```

FL_DATE	False
OP_CARRIER_FL_NUM	False
CRS_DEP_TIME	False
DEP_TIME	False
DEP_DELAY	False
TAXI_OUT	False
WHEELS_OFF	False
WHEELS_ON	False
TAXI_IN	False
CRS_ARR_TIME	False
ARR_TIME	False
ARR_DELAY	False
CANCELLED	False
DIVERTED	False
CRS_ELAPSED_TIME	False
ACTUAL_ELAPSED_TIME	False
AIR_TIME	False
DISTANCE	False
CARRIER_DELAY	False
WEATHER_DELAY	False
NAS_DELAY	False
SECURITY_DELAY	False
LATE_AIRCRAFT_DELAY	False
Unnamed: 27	True
Target	False
dtype:	bool

```
#Dropping all null values column  
df.drop('Unnamed: 27',inplace=True,axis=1)
```

Then checking the total null values present all the columns of dataset. We surprisingly found that CARRIER\_DELAY, WEATHER\_DELAY, NAS\_DELAY, SECURITY\_DELAY AND LATE\_AIRCRAFT\_DELAY have more than 80 percent null values. It makes no sense if column have more than 80 percent null values, so come the conclusion that we would drop those column as well.

```
df1=df.isnull().sum()/df.shape[0]*100  
df1
```

FL_DATE	0.000000
OP_CARRIER_FL_NUM	0.000000
CRS_DEP_TIME	0.000000
DEP_TIME	1.415213
DEP_DELAY	1.415830
TAXI_OUT	1.447586
WHEELS_OFF	1.447515
WHEELS_ON	1.492153
TAXI_IN	1.492153
CRS_ARR_TIME	0.000000
ARR_TIME	1.492153
ARR_DELAY	1.677839
CANCELLED	0.000000
DIVERTED	0.000000
CRS_ELAPSED_TIME	0.000123
ACTUAL_ELAPSED_TIME	1.677839
AIR_TIME	1.677839
DISTANCE	0.000000
CARRIER_DELAY	81.858295
WEATHER_DELAY	81.858295
NAS_DELAY	81.858295
SECURITY_DELAY	81.858295
LATE_AIRCRAFT_DELAY	81.858295
Target	0.000000

dtype: float64

```
df.drop(['CARRIER_DELAY', 'WEATHER_DELAY', 'NAS_DELAY', 'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY'],axis=1,inplace=True)
```

Here we have deleted the rows of column CANCELLED = 1(yes) and DIVERTED=1(yes). As there is no sense to predict the flight delays for already cancelled and diverted flights, and these features won't be helpful for the further delay predictions.

```
df = df[df.CANCELLED != 1]
```

```
df.CANCELLED.unique()  
array([0.])
```

```
df.DIVERTED.value_counts(normalize=True)
```

```
0.0    0.997762  
1.0    0.002238  
Name: DIVERTED, dtype: float64
```

```
df = df[df.DIVERTED != 1]
```

```
df.DIVERTED.unique()  
array([0.])
```

We have filled the Na values with the help of central tendency. The 3 most common measure of central tendency are the mode, median and mean where the mode is the most frequent value ,median is the middle number in ordered data set and mean is the sum of all the values divided by the total numbers of value. We have used mode central tendency as it performs best with respect to others.

## Handelling the NA Values

```
#Min one na value is present or not in columns  
df.columns[df.isna().any()]
```

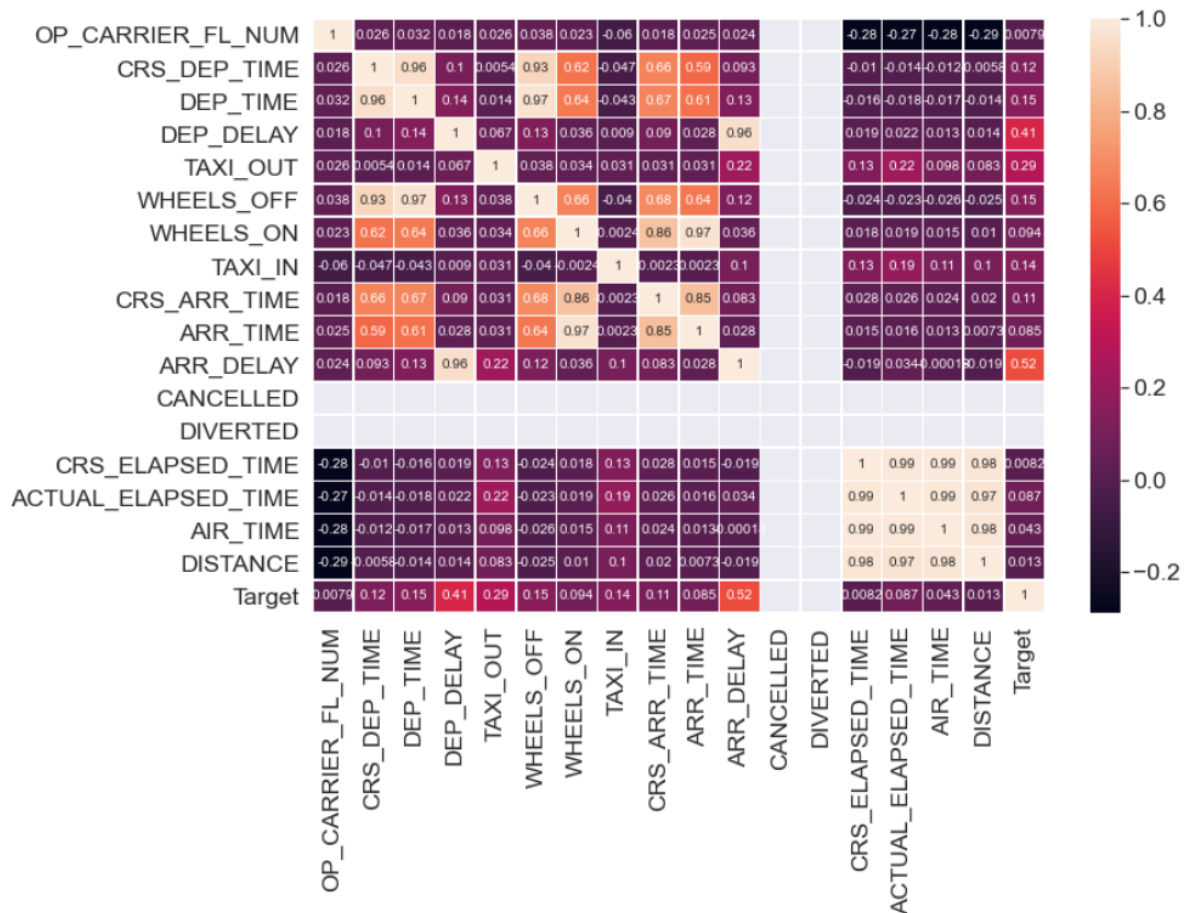
```
Index(['WHEELS_ON', 'TAXI_IN', 'ARR_TIME', 'ARR_DELAY', 'ACTUAL_ELAPSED_TIME',  
      'AIR_TIME'],  
      dtype='object')
```

```
for col in df.columns[df.isnull().any()]:  
    df[col].fillna(df[col].mode()[0],inplace=True)  
df.columns[df.isnull().any()]
```

```
Index([], dtype='object')
```

### 3.1 Data Visualization:-

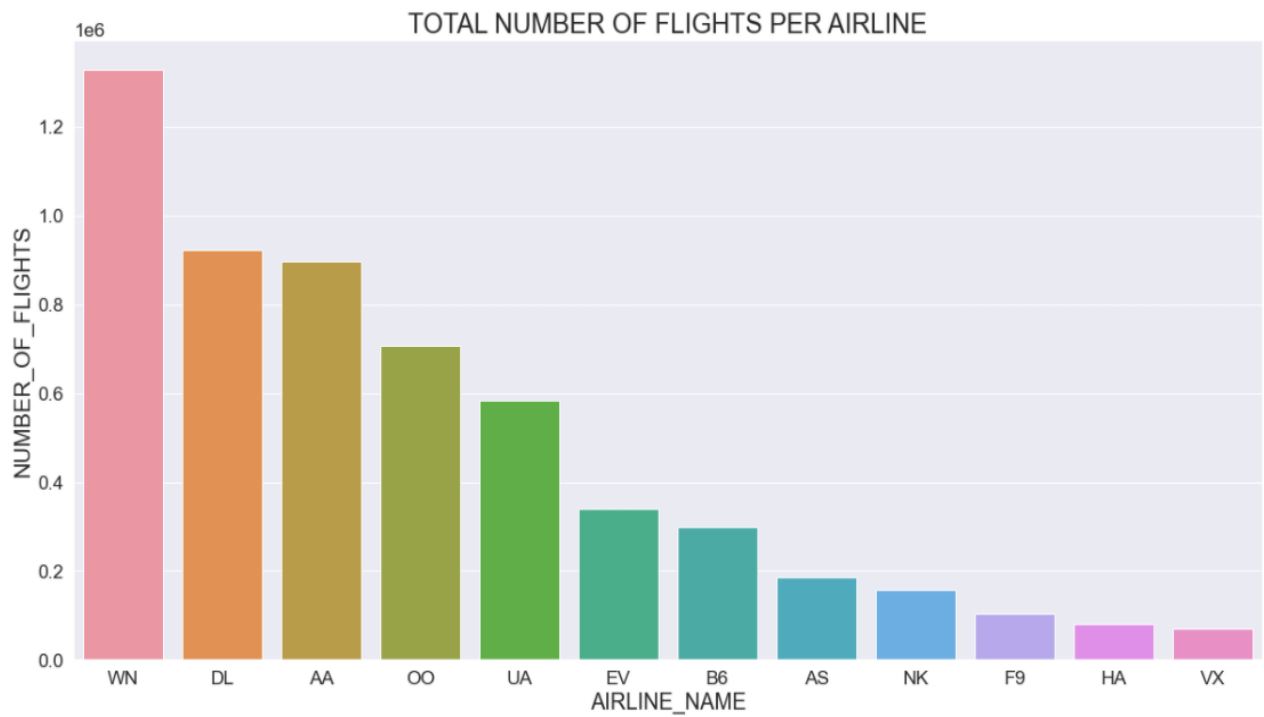
#### a. Correlation Heat map:-



In correlation heat map the light variable shows the highly correlated where as the dark color shows the least correlated with each other as you can observed the CANCELLED and DIVERTED has no impact on other features so we can drop the both columns.

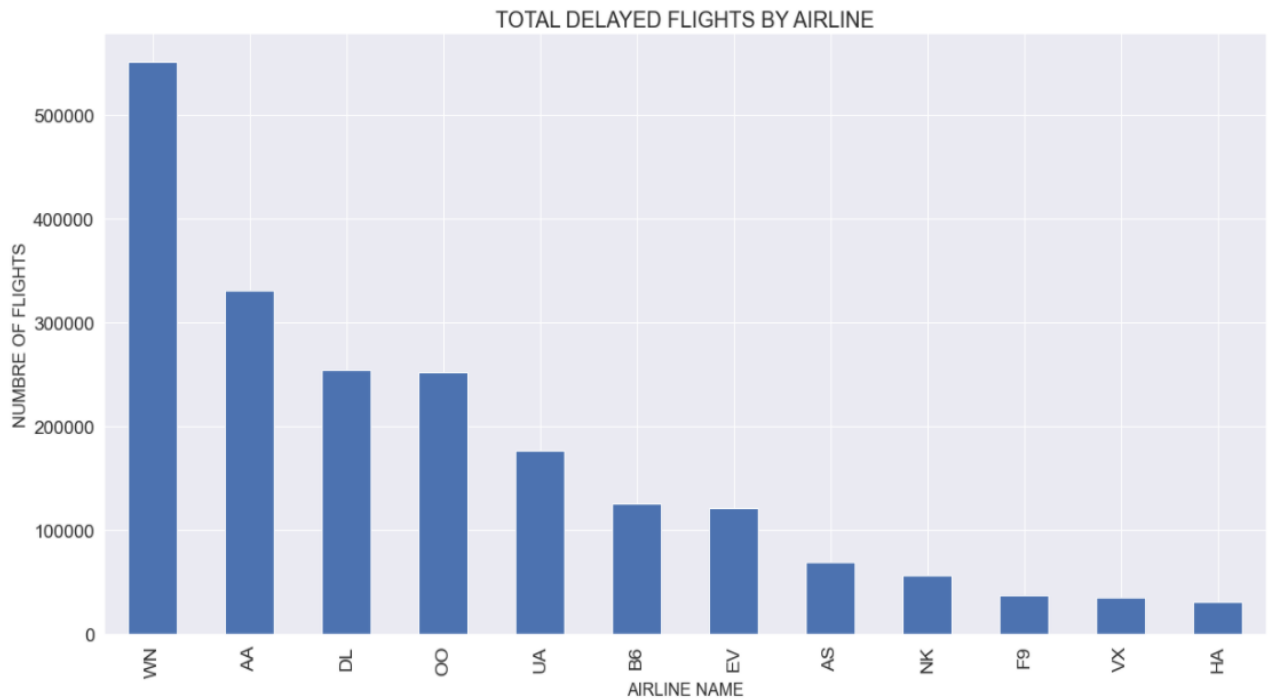
```
df.drop(['CANCELLED','DIVERTED'],axis=1,inplace=True)
```

**b. Total Number of flights per Airline:-**



**Fig 1. Bar plot**

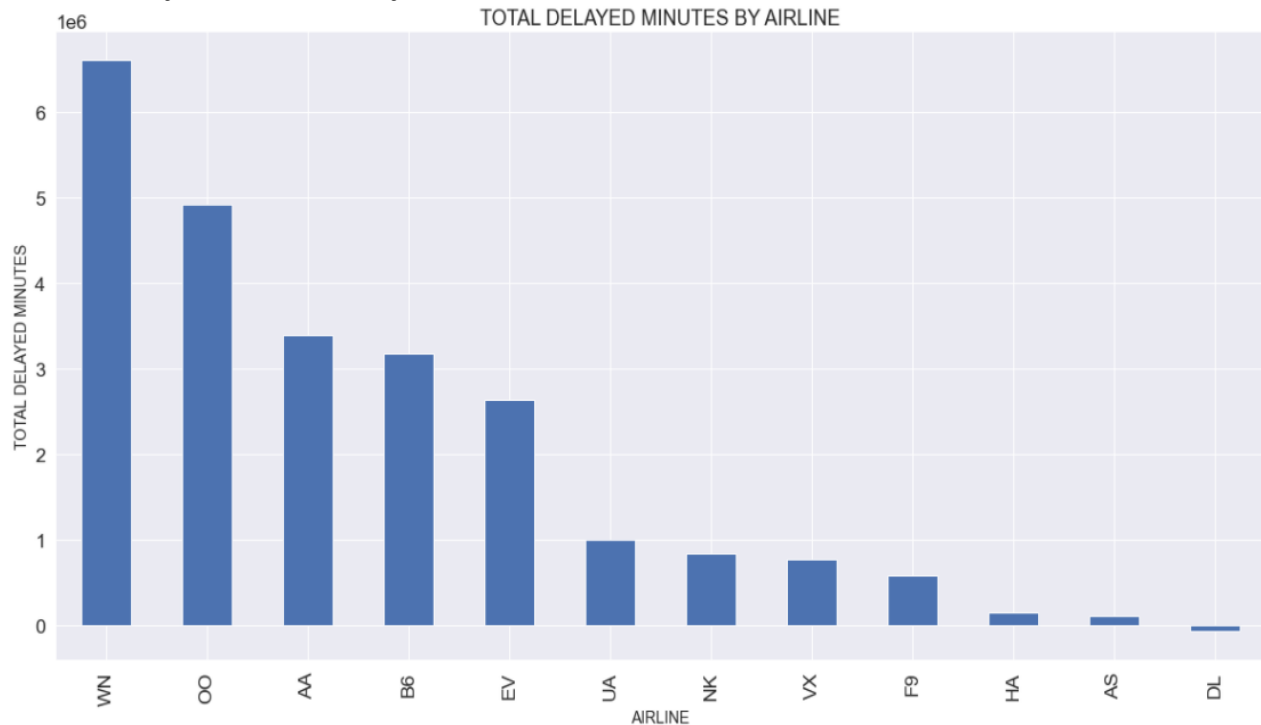
**c. Total Delayed Flights by Airline:-**



**Fig 2. Bar plot**



**d. Total Delayed Minutes by Airlines:**



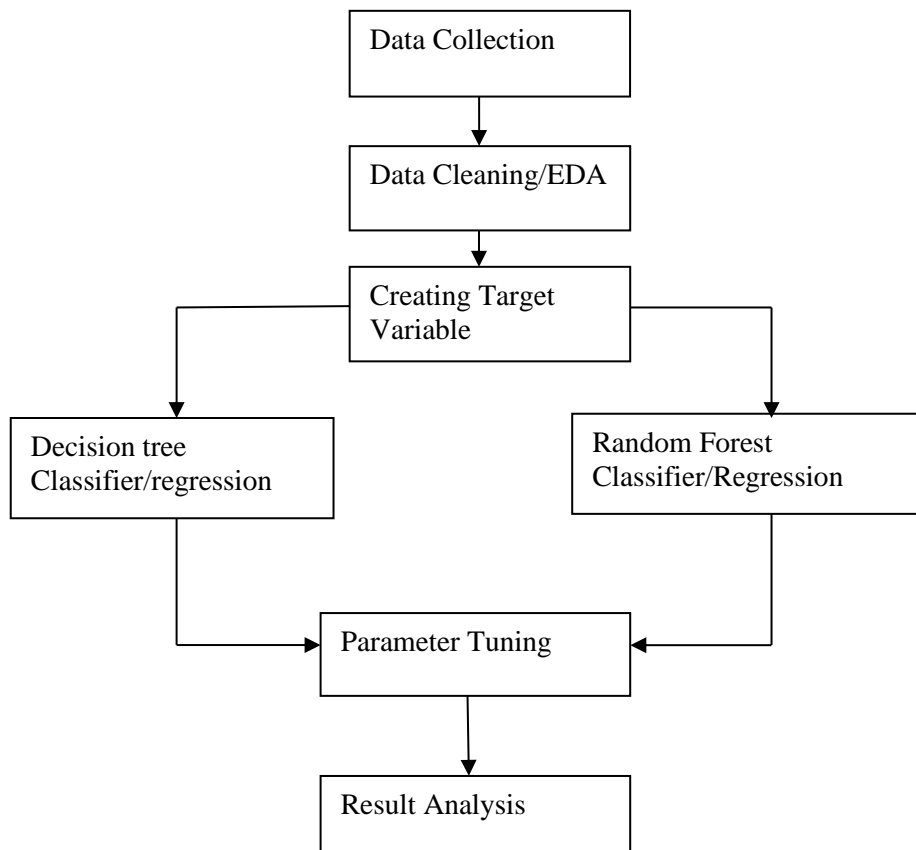
**Fig 3. Bar plot**

With reference of above plots we come to know that B6 (JetBlue Airways) operates less number of flights as compare to others (Top 5) . Along with there flights are delayed also in large number as compare to other 4 flight operators. So passengers must be cautious while selecting the airlines.

## Chapter 4:- System Design

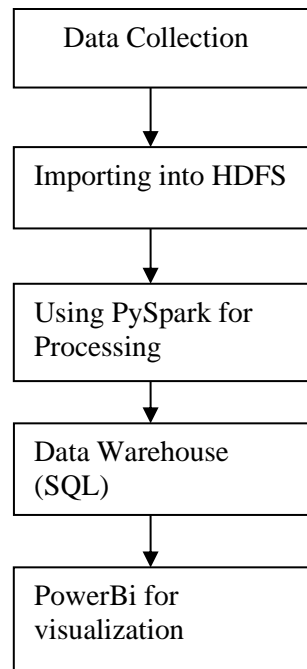
### 4.1 Flowchart of System:

#### Flowchart for the Machine Learning algorithm



Above is the flow chart for machine leaning algorithm. By performing all the steps of the flow chart we would able to get the appropriate result for our dataset.

## System Design for analyzing and visualizing the data:



We are setting up our Hadoop cluster along with spark on AWS where we call it EMR (Elastic Map Reduce) service. The cluster have one master node (no high availability) and two worker nodes. Generally specification for cluster is like 4core cpu and 7.5gb memory and 80gb base storage for each machine and this configurations are scalable according to the requirements.

For data storage (staging) we used simple storage service(s3 bucket)

For data warehousing mysql engine running on single machine and Powerbi running on local system.

ID	Status	Node type & name	Instance type
▶ <a href="#">ig-2F4B1DJDQICIY</a>	Terminated (1 Requested)	<b>MASTER</b> Master - 1	<b>c3.xlarge</b> 4 vCore, 7.5 GiB memory, 80 SSD GB storage EBS Storage: none
▶ <a href="#">ig-CZFUXWZYSTTH</a>	Terminated (2 Requested)	<b>CORE</b> Core - 2	<b>c3.xlarge</b> 4 vCore, 7.5 GiB memory, 80 SSD GB storage EBS Storage: none

## Configuration details

**Release label:** emr-5.32.0






**Hadoop distribution:** Amazon 2.10.1


**Applications:** Hive 2.3.7, Hue 4.8.0, Spark 2.4.7, Sqoop 1.4.7





**Log URI:** --


**EMRFS consistent view:** Disabled

**Custom AMI ID:** --

**Buckets (1)**   Copy ARN  Empty  Delete  Create bucket

Buckets are containers for data stored in S3. [Learn more](#) 

 Find buckets by name  1  

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
	iacsd	Asia Pacific (Singapore) ap-southeast-1	<u>Objects can be public</u>	February 18, 2021, 12:14:10 (UTC+05:30)




RDS > Databases > database-1

## database-1

Modify

Actions ▼

### Summary

DB identifier database-1	CPU  1.69%	Status  Available	Class db.t2.micro
Role Instance	Current activity  0 Connections	Engine MySQL Community	Region & AZ ap-southeast-1b

## **Chapter 5**

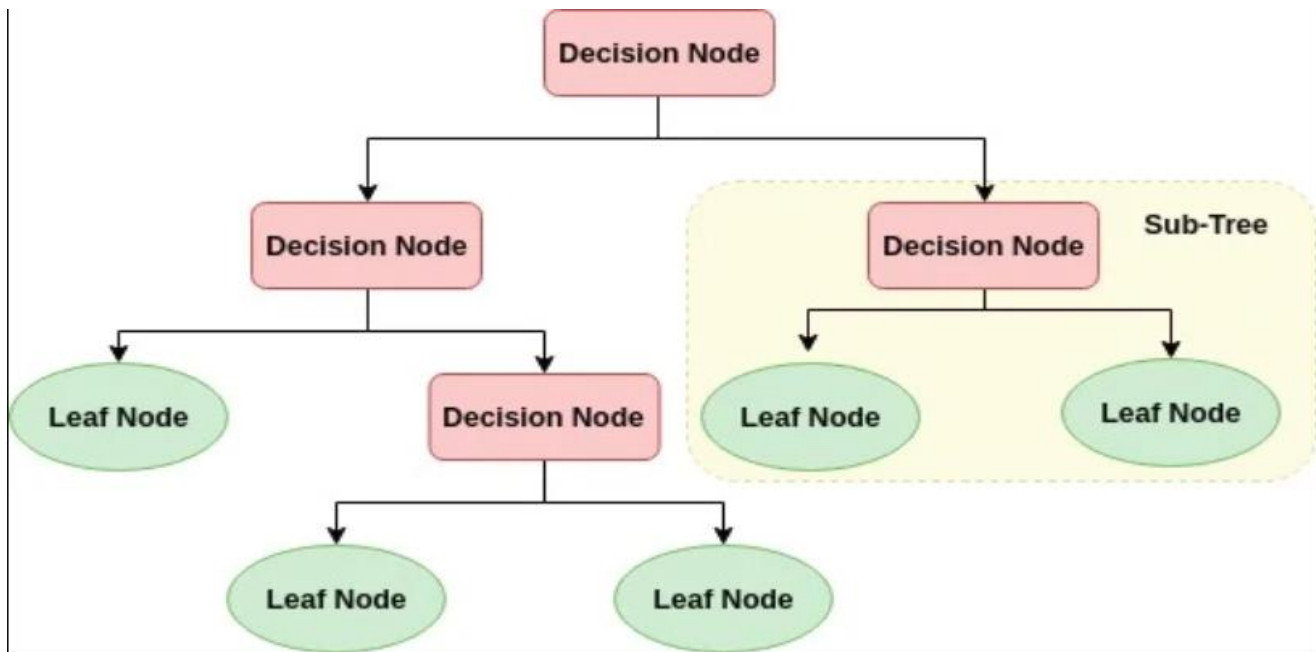
### **Model Building**

#### **5.1 Algorithm research and Selection:-**

The machine learning classification models have been used for prediction of delay of flights. The brief details of each model is described below.

##### **5.1.1 Decision Tree:-**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.



**Pros:**

- 1 Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
- 2 A decision tree does not require normalization of data.
- 3 A decision tree does not require scaling of data as well.
- 4 Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
- 5 A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

## Cons:

- 1 A small change in the data can cause a large change in the structure of the decision tree causing instability.
- 2 For a Decision tree sometimes, calculation can go far more complex compared to other algorithms.
- 3 Decision tree often involves higher time to train the model.
- 4 Decision tree training is relatively expensive as the complexity and time has taken are more.
- 5 The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

## Decision Tree for Classification:

```
#Decision Tree for Classification
Y = df['Target'] #Testing data
X = df.drop(['Target', 'FL_DATE'],axis=1) #Train
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.3,random_state =7)
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
model = tree.DecisionTreeClassifier(random_state=7)
diag=model.fit(X_train,Y_train)
diag
model.score(X_test,Y_test)
pred= model.predict(X_test) #Accuracy
pred
```

## Decision tree for Regression:

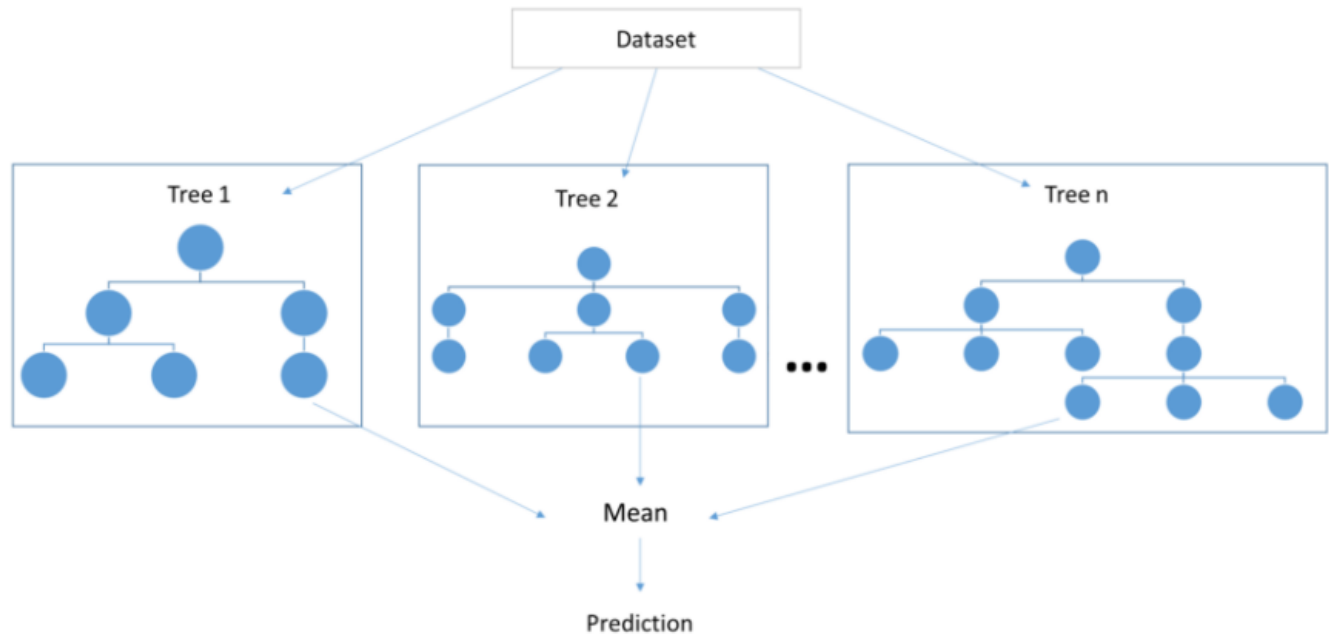
```
Y1 = df['ARR_DELAY']
X1 = df.drop(['Target', 'FL_DATE', 'ARR_DELAY'], axis=1)
from sklearn.model_selection import train_test_split
X1_train, X1_test, Y1_train, Y1_test = train_test_split(X1, Y1, test_size = 0.3, random_state = 7)
X1_train.shape, X1_test.shape
from sklearn.tree import DecisionTreeRegressor
model_reg = DecisionTreeRegressor(random_state=7)
model_reg.fit(X1_train, Y1_train)
model_reg.score(X1_test, Y1_test)
pred1 = model_reg.predict(X1_test)
pred1
```

### 5.1.2 Random Forest:

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forests has a variety of applications, such as recommendation engines, image classification and feature selection.



**Pros:**

1. Random forests is considered as a highly accurate and robust method because of the number of decision trees participating in the process.
2. It does not suffer from the over fitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.
3. The algorithm can be used in both classification and regression problems.

**Cons:**

1. Random forests is slow in generating predictions because it has multiple decision trees. Whenever it makes a prediction, all the trees in the forest have to make a prediction for the same given input and then perform voting on it. This whole process is time-consuming.
2. The model is difficult to interpret compared to a decision tree, where you can easily make a decision by following the path in the tree.

## Random Forest For Classification:

```
from sklearn.ensemble import RandomForestClassifier
model_Class = RandomForestClassifier(random_state=7)
model_Class.fit(X_train, Y_train)
model_Class.score(X_test, Y_test)
pred2=model_Class.predict(X_test)
pred2|
```

## Random Forest For Regression:

```
: from sklearn.ensemble import RandomForestRegressor
model_reg1 = RandomForestRegressor(n_estimators=50)
model_reg1.fit(X1_train, Y1_train)
model_reg1.score(X1_test, Y1_test)
pred3 = model_reg1.predict(X1_test)
pred3|
```

## Hyper Parameter Tuning:

Hyper parameters are important because they directly control the behavior of the training algorithm and have a significant impact on the performance of the model is being trained

A hyper parameter is a hyper parameters whose value is used to control the learning process and increase the model performance by choosing best parameters . Hyper parameter tuning works by running multiple trails in a single training job. Each trial is a complete execution of your training application with values for your chosen hyper parameters, set within limits you specify.

Parameters for Decision tree:

```
param = {  
    'criterion': ['gini', 'entropy'],  
    'max_depth': [1, 2, 3, 4, 5, None]    #max split  
}
```

```
from sklearn.model_selection import GridSearchCV  
GCV_grid=GridSearchCV(estimator=model,param_grid=param,cv=10,verbose=2)
```

```
GCV_grid.fit(X_train, Y_train)
```

Best parameters for decision tree is:

```
{'criterion': 'gini', 'max_depth': 1}  
[0.99999974 0.99999974 0.99999974 0.99999949 0.99999949 0.99999949  
 0.99999974 0.99999974 0.99999974 0.99999949 0.99999949 0.99999949]
```

**Criterion :**

**Gini :**

Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. The degree of Gini index varies between 0 and 1.

While building the **Decision Tree**, we would prefer choosing the feature with the least Gini index as the root node.

## Entropy:

Entropy can be defined as a measure of the purity of the sub split. Entropy always lies between 0 to 1. It also helps for selecting the best splitting.

Parameters for Random Forest:

```
: param1 ={
    'n_estimators':[10,50,100],
    'criterion' :['gini', 'entropy'],    #how many decision tree to keep
    'max_depth':[1,2,3,4,5,None]        #max split
}

: from sklearn.model_selection import GridSearchCV
  GCV_grid1=GridSearchCV(estimator=model_Class,param_grid=param1,cv=5,verbose=1)
```

We got the best parameters for tuning for random forest is given below:

---

```
{'criterion': 'gini', 'max_depth': 3, 'n_estimators': 100}
[0.96102481 0.90858711 0.88651744 0.99995186 0.96677708 0.9645536
 0.99948868 0.99467916 0.99999974 0.97478381 0.99954783 0.99999974
 0.99999974 0.99999974 0.99999974 0.99999974 0.99999974 0.99999974
 0.95715164 0.90410635 0.88225662 0.99986865 0.9699067 0.97111753
 0.99970248 0.99618444 0.99999974 0.9856398 0.99994726 0.99999974
 0.99999974 0.99999974 0.99999974 0.99999974 0.99999974 0.99999974]
```



```

from pyspark.sql import SparkSession
from pyspark.sql.functions import *

spark = SparkSession.builder.appName("AirlinesDataset").getOrCreate()

Airdata = spark\
    .read\
    .option("inferSchema", "true")\
    .option("header", "true")\
    .csv("s3://iacsd/2017.csv")

spark.conf.set("spark.debug.maxToStringFields", 10000)
Airdata.createOrReplaceTempView("Airdata")

```

## Data Processing and Storing Results into Data warehouse:

```

spark.conf.set("spark.debug.maxToStringFields", 10000)
Airdata.createOrReplaceTempView("Airdata")

sqlproperties = {"user":"admin", "password":"admin123", "driver":"com.mysql.jdbc.Driver"}

distance=spark.sql("select op_carrier,sum(distance) from Airdata group by op_carrier")
distance.write.jdbc(url="jdbc:mysql://database-1.cdzt5btssyqo.ap-southeast-1.rds.amazonaws.com/delaydb",table="distance",properties=sqlproperties)

```

```

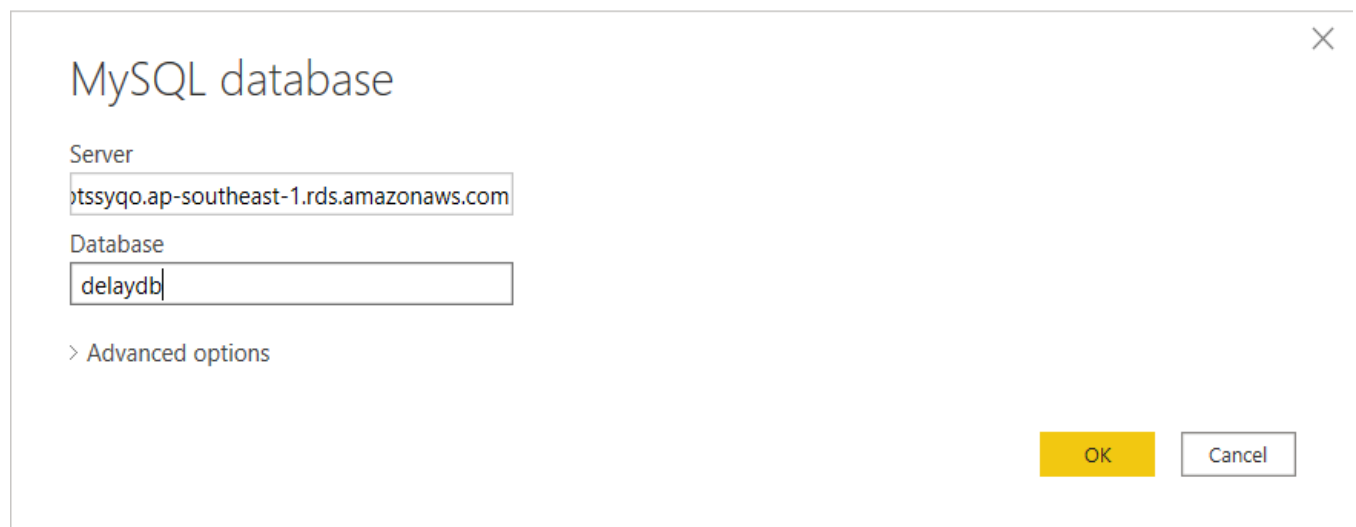
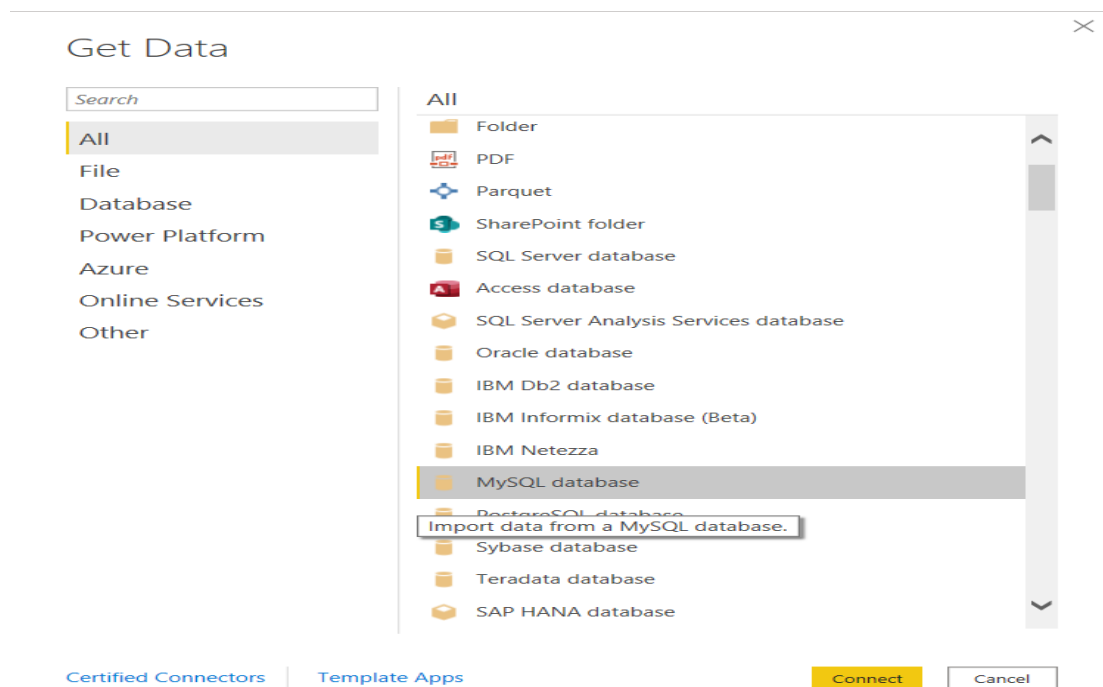
MySQL [delaydb]> show tables;
+-----+
| Tables_in_delaydb |
+-----+
| avg_delay          |
| cancel_count       |
| delay_count        |
| destination        |
| distance            |
| divert_count       |
| flights_count      |
| nondelay_count     |
| origin             |
| total_delay        |
+-----+
10 rows in set (0.00 sec)

MySQL [delaydb]> select * from cancel_count;
+-----+-----+
| count(op_carrier_fl_num) | op_carrier |
+-----+-----+
| 5959                     | UA         |
| 5141                     | NK         |
| 12138                    | AA         |
| 9966                     | EV         |
| 8288                     | B6         |
| 8960                     | DL         |
| 10458                    | OO         |
| 966                      | F9         |
| 230                      | HA         |
| 18046                    | WN         |
| 1508                     | AS         |
| 1033                     | VX         |
+-----+-----+
12 rows in set (0.00 sec)

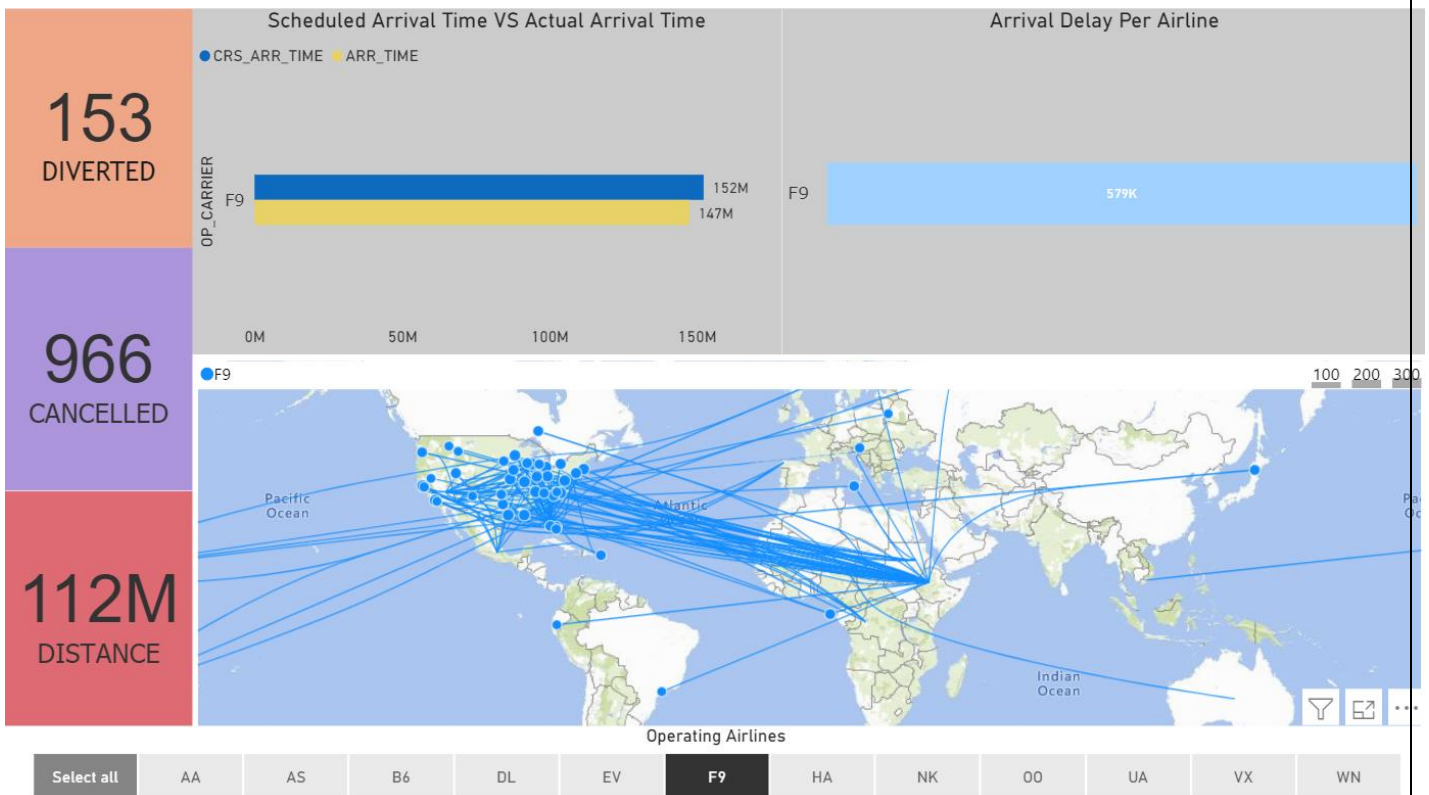
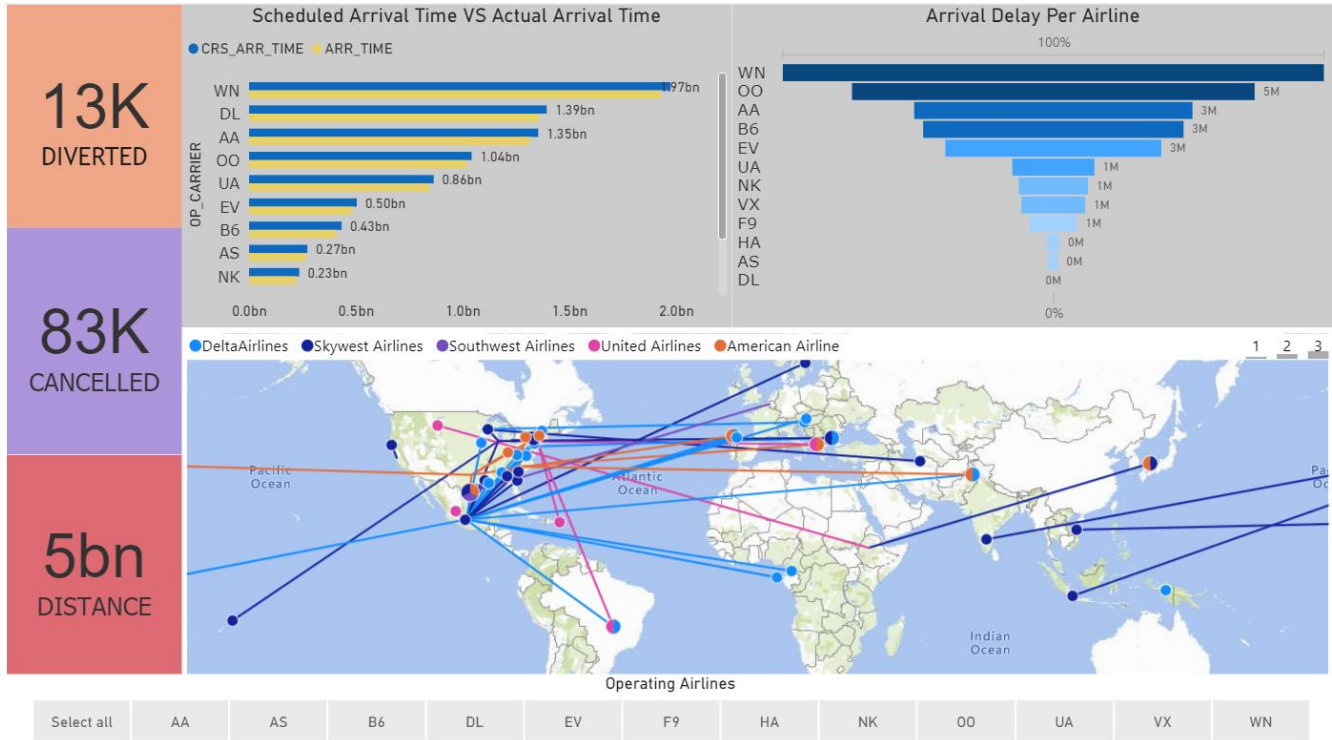
MySQL [delaydb]>

```

## Pulling the Data from Warehouse using Powerbi:



## Visualization:





## **Chapter 6**

### **Results**

- 1.All the models which we have fitted Random forest is the best model with best accuracy around 99% for classification and for regression we got 99.77% of accuracy.
- 2.The accuracy of each and every algorithms are high , so over fitting criteria are also check to diagnose the problem but, model perform in same manner for training and testing dataset.
- 3.From analysis and visualization of data with help of spark and powerbi we get great insights like operating efficiency, count of total number flights, count of diverted and cancelled flights along with their reason behind cancellation and many more other aspects from which we easily select the flights with better, best and worst operations efficiency.

## **Chapter 7**

### **Future Scope**

The scope of Machine Learning is not limited to only predictions, but with the help of this project we can upgrade this into application which will provide passenger insight details for caused of delay of flight and it will help the customer as well the airlines for better understanding. With new advancement in the field of Deep learning we can use Neural Networks algorithm on the flight and weather data. As neural Network works on pattern matching methodology. Also the scope of the project is very much confined to flight and weather delay of US so here we can also take into count of other countries like India, Japan and many more

## **CHAPTER 8**

### **Conclusion**

Here we utilized machine learning capabilities to predicting the flight delays. This model is based on a simple classification and regression techniques using Decision Tree and Random Forest algorithms. The model has achieved an overall 99% testing accuracy on publicly accessible dataset. It is concluded from accuracy that Random Forest is highly suitable for solving this kind of problem statements. For storing and processing this kind of large datasets Spark with Hadoop cluster doing a great job by harnessing the capabilities of Parallel processing. And for better insights and visuals from data Powerbi has done the great job.

In near future this module of prediction can be integrate with the module of automated processing system. The system is trained on old training dataset in future can be made such that new testing data in real time flight delay prediction

## References:-

1. Flight Delay Prediction Based on Aviation Big Data and Machine Learning  
Authors: Guan Gui ; Fan Liu ; China ; Jan.2020  
(<https://ieeexplore.ieee.org/document/8903554> )
2. Prediction model and algorithm of flight delay propagation based on integrated consideration of critical flight resources  
Authors: Rong Yao ; Wang Jiandong ; China ; 2009  
(<https://ieeexplore.ieee.org/document/5267970>)
3. <https://www.transtats.bts.gov>
4. <http://stat-computing.org/dataexpo/2009/>
5. <https://packages.revolutionanalytics.com/datasets/>

