

AIRLINE DATA ANALYSIS USING SPARK TECHNOLOGIES

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KHAJJAYAM SUMANTH KUMAR (Reg No.37110342)

KOLIPAKULA KARTHIK (Reg No.37110349)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY(DEEMED TO BE
UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by
AICTE**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC |
Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KHAJJAYAM SUMANTH KUMAR (Reg.No.37110342)** and **KOLIPAKULA KARTHIK (Reg.No.37110349)** who carried out the project entitled "**AIRLINE DATA ANALYSIS USING SPARK TECHNOLOGIES**" Under our supervision from November 2020 to April 2021.

Internal Guide
Dr.S.PRINCE MARY,M.E.,Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **KHAJJAYAM SUMANTH KUMAR (Reg.No.37110342)** hereby declare that the Project Report entitled "**AIRLINE DATA ANALYSIS USING SPARK TECHNOLOGIES**" done by me under the guidance of **Dr.S.PRINCE MARY,M.E.,Ph.D.**, at **Sathyabama Institute of Science and Technology** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala.,M.E.,Ph.D.**, Dean, School of Computing , **Dr.S.Vigneshwari.,M.E.,Ph.D.**, and **Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.PRINCE MARY,M.E.,Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This project analyzes the US domestic flights dataset using pyspark data frames. This will aim to predict which flight/flight carrier is most likely to be cancelled or delayed. Using Pyspark, a big data technology tool, we can therefore work with RDDs/Data frames/Datasets in Python. In this analysis will help us find out what are the top airlines with the maximum delays/cancellations in the US. Also, this project will investigate whether there are any factors that cause flight delays/cancellations. Further, this project predicts if a flight will get cancelled or not. In order to build the predictive system Machine learning models were incorporated, where we modeled a binary classification problem with categorical variables, logistic regression, decision tree classifier, random forest and gradient boosted trees.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
	1.1 PYSPARK	1
2.	LITERATURE SURVEY	3
	2.1 DATA ANALYTICS	3
	2.2 BIG DATA ANALYTICS	3
	2.3 EFFICIENCY EVALUATION	4
3.	AIM &SCOPE	6
	3.1 AIM	6
	3.2 OBJECTIVES	6
	3.3 ADVANTAGES	6
	3.4 LANDING EFFICIENCY	7
4.	SYSTEMS & ALGORITHMS	9
	4.1 EXISISTING SYSTEM	9
	4.2 PROPOSED SYSTEM	10
	4.3 SYSTEM REQUIREMENTS	11
	4.3.1 HARDWARE REQUIREMENTS	11
	4.3.2 SOFTWARE REQUIREMENTS	11
	4.4 MODULES	12
	4.4.1 DATA PROFILING	12
	4.4.2 DATA CLEANING & PREPARATION	12
	4.4.3 STATISTICAL DATA ANALYSIS	12
	4.4.4 MODEL PREDICTION	13
	4.4.5 DECISION TREE CLASSIFIERS	13

5.	SOFTWARE DEVELOPMENT METHODOLOGY	13
5.1	DESCRIPTION DIAGRAM	13
5.2	DATA FLOW DIAGRAM	14
5.3	ALGORITHMS	15
5.3.1	RANDOM FOREST ALGORITHM	16
5.3.2	WORKING	16
5.3.3	DECISION TREE ALGORITHM	17
5.3.4	LINEAR REGRESSION	17
6.	RESULTS & DISCUSSION	18
7.	CONCLUSION	23
	REFERENCES	24
	APPENDIX	
	A. PLAGIARISM REPORT	25
	B. JOURNAL PAPER	26
	C. SOURCE CODE	31

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
5.1	ARCHITECTURE DIAGRAM	13
5.2	DATA FLOW DIAGRAM	14
6.1	COLLECTED DATA	19
6.2	GLOSSARY OF ATTRIBUTES	20
6.3	DATA LOADING	21
6.4	PERCENTAGE DELAYED	22
6.5	AVERAGE ARRIVAL DELAY	22
6.6	MODEL PREDICTION	23

CHAPTER 1

INTRODUCTION

The dataset was taken from Kaggle, comprised 7 CSV files containing data from 2009 to 2015, and was about 7GB in size. It contained information about airlines, delay information, location details (origins and destinations), and cancellation (reasons labelled as cancellation codes). It also had certain technical details such as the time that the plane was stationary on ground.

My approach revolved around Apache Spark, specifically using PySpark with AWS EMR cluster support. This virtual set-up uses distributed, in-memory data structures to improve speeds for many data processing workloads by several orders of magnitude. As a unified analytics engine, it allows us to parallelize large data processing tasks on a distributed cluster.

1.1 PYSPARK

PySpark is also great for exploratory data analysis at scale, building machine learning pipelines, and creating ETLs (extract, transform, load) tools. Once the EMR cluster was “running,” I connected to Jupyterlab available as notebooks under AWS services and loaded the data from S3 buckets by importing it through the Kaggle website. I performed exploratory data analysis and model prediction.

One of the main concerns in air transport, and specifically in Air Traffic Management (ATM), is the optimisation of trajectories to allow aircraft to reach their destinations in the shortest possible time. A suboptimal trajectory has clear economic consequences, both for passengers (i.e., longer travel times) and airlines (i.e., higher fuel and crew costs); but it also affects the environment, as more fuel burnt implies larger quantities of pollutant emissions - to illustrate, a medium range aircraft contributes to climate change with around 1.5Kg of CO₂ per minute of flight [1]. While *prima facie* the best trajectory is expected to be a straight (or geodesic) one, real operations are more complex, with multiple factors modifying this trivial solution. For instance, planners try to avoid enroute headwinds, and to take advantage of tailwinds; and they further have to comply with the structure of airspaces and routes, designed to ensure an effective air traffic control, and hence the safety of operations. Research efforts to define and measure efficiency in air transport have been focused on three complementary aspects. On one hand, several works have tried to assess how efficient the actual system is

from a global perspective, i.e. its capacity of matching flights offer with demand. Examples of this include focusing on the transportation demand of the state of Indiana; focusing on the hub-and spoke network of the Java Island; and analysing the efficiency of 24 major international airports. Moving the focus to flights themselves, a second set of works have been focused on understanding the efficiency of different airspaces and on comparing them. Specifically, proposed a set of metrics to define the lateral and vertical deviation from an ideal trajectory, and to evaluate the additional fuel needed because of these deviations. A similar problem was tackled by finding that vertical deviations account for a 3% increase in fuel consumption in the US airspace. Comparable analyses were performed for Japan, US, Europe and China.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

2.1 Data analytics development of FDR (Flight Data Recorder) data for airline maintenance operations

In this article, we propose a data analytics development to detect unusual patterns of flights from a vast amounts of FDR (flight data recorder) data for supporting airline maintenance operations. A fundamental rationale behind this development is that if there are potential issues on mechanical parts of an aircraft during a flight, evidences for these issues are most likely included in the FDR data. Therefore, the data analysis of FDR data enables us to detect the potential issues in the aircraft before they occur. To this end, in a data pre-processing step, a data filtering, a data sampling, and a data transformation are sequentially performed. And then, in this analysis, all-time series data in the FDR are classified into three types: a continuous signal, a discrete signal, and a warning signal. For each type of signal, a high-dimensional

vector by arranging the time series data is chosen as features. In the feature section process, a correlation analysis, a correlation relaxation, and a dimension reduction are sequentially conducted. Finally, a type of k-nearest neighbor approach is applied to automatically identify the FDR data in which the unusual flight patterns are recorded from a large amount of FDR data. The proposed method is tested with using a realistic FDR data from the NASA's open database.

2.2 Big Data Analytics on Aviation Social Media: The Case of China Southern Airlines on Sina Weibo

The proposed model; (3) using sentiment analysis to study the case of China Southern Airlines on Sina Weibo, and illustrating Weibo users' attitudes toward China Southern Airlines. This study also discusses the practical implications for airlines to manage their social media platforms. With the combination of a passenger's social media value and passenger's other offline behavior information, we can establish a comprehensive profile for travelers.

2.3 Big Data Analytics in Airlines: Efficiency Evaluation using DEA

This study aims to quantitatively evaluate the operational efficiency in the airline scheduling and execution process by implementing big data analytics approach. Parameters for calculation are obtained from prior studies. These parameters are calculated using Data Envelopment Analysis (DEA) method to get efficiency scores for each operation process every month. Finally, we argue that the data analytics approach is beneficial to be implemented in airlines and find a decreasing trend in the efficiency score of the sample airline during 2017-2018.

CHAPTER 3

AIM &SCOPE

3.1 Aim

The main purpose of the project to explore detailed analysis on airline data sets such as listing airports in the India, list of airlines having zero stops, list of airlines operating with code share which country has highest airports and list of active airlines in united states.

3.2 Objectives

The main purpose of the project to explore detailed analysis on airline data sets such as listing airports in the India, list of airlines having zero stops, list of airlines operating with code share which country has highest airports and list of active airlines in united states. The main objective is the processing bigdata sets using spark technology.

3.3 Advantages

- By using airline data analysis users time can be saved.
- Cost Savings

3.4 Landing Efficiency

As a first overview to the results, reports the distribution of the landing time for all airports. As previously detailed, here landing time refers to the time from crossing an altitude of 10; 000 feet to touchdown. This time is of importance in operations, as flying at low altitude is slower and implies higher fuel consumption; hence, a reduction of this time implies both savings in costs, and a reduction of the environmental impact. Grey bands of correspond to the 15th-85th percentiles range, while black and red horizontal lines respectively to the average and median of the distribution. Two relevant observations can already be made. First of all, and opposed with what may be intuitive ,airports are characterised by a great variability in the landing time - for instance, it varies between 12 and 20 minutes for aircraft landing at London Heathrow (EGLL).

This suggests that landing procedures are not fixed, and that important variations in efficiency can be experienced. Secondly, large airports (e.g., EGLL, EHAM) are generally less efficient than small ones (as, for instance, LSZH or EDDL), pointing towards the complexity of the airspace as the main cause behind inefficiencies. The case of Zurich Airport (LSZH) is especially noteworthy, as it presents a small average landing time and dispersion, even though it is surrounded by mountains and its approach is considered one of the most complex in Europe. If traffic seems to be globally associated to less efficient approaches, the same may hold true at a more micro-level; in other words, one may expect a larger landing time in those days with a higher volume of traffic.

To test this hypothesis, reports a scatter plot for each airport, depicting the median landing time of each day as a function of the number of operations. Note that, in order to obtain a more reliable estimation, this latter number includes all detected landings, thus not just those fulfilling the filters described in Sec. II. If there was a relationship between landing times and traffic volumes, this would appear as structures in the scatter plot; instead, points are forming clouds, thus suggesting no clear relationship. At the same time, a test indicates that, at least for 7 of the considered airports, such hypothesis can be accepted in a statistically significant way, with increases in the average landing time of around a 10%. We further analyse whether the hour of the day and the season have an effect on the average and spread of the landing time. As can be seen in, depicting the distribution of landing times as a function of respectively the hour of the day and the season, clear trends appear, at least in some airports. For instance, Amsterdam Airport Schiphol (EHAM) and Barcelona-El Prat (LEBL) present a clear positive correlation between mean landing time and traffic, with peaks in the 8h-12h time band. On the other hand, the situation is less clear in the case of seasons, with some airports even presenting a negative correlation; this is probably due to the too low resolution of an analysis based on four consecutive months. In order to better spot the presence of an overall correlation, reports two scatter plots, with the deviation of landing time .

CHAPTER 4

SYSTEMS & MODULES

4.1 EXISISTING SYSTEM

In Existing system, an airport has huge amount of data related to number of flights, data and time of arrival and dispatch, flight routes, number of airports operating in each country, list of active airlines in each country. The problem they faced till now it's, they have ability to analyze limited data from databases.

This study proposes a passenger value model based on a social media platform. Using the case of China Southern Airlines on Sina Weibo, this study illustrates how big social data analysis can help an airline company to understand better the passengers and improve customer relationship management. The objectives of this study are threefold: (1) developing a model of passengers' social media value, (2) discussing the possible application scenarios for the proposed model, (3) using sentiment analysis to study the case of China Southern Airlines on Sina Weibo, and illustrating Weibo users' attitudes toward China Southern Airlines. This study also discusses the practical implications for airlines to manage their social media platforms. With the combination of a passenger's social media value and passenger's other offline behavior information, we can establish a comprehensive profile for travelers.

Airline schedule planning is a crucial issue in airline operation typically designed most efficiently and effectively. Even so, the plan execution, including recovery actions to any prevailed irregularities, has a considerable impact on airline efficiency. Those activities produce a vast amount of operational data which contains valuable insights for businesses. This study aims to quantitatively evaluate the operational efficiency in the airline scheduling and execution process by implementing big data analytics approach. Parameters for calculation are obtained from prior studies. These parameters are calculated using Data Envelopment Analysis (DEA) method to get efficiency scores for each operation process every month. Finally, we argue that the data analytics approach is beneficial to be implemented in airlines and find a decreasing trend in the efficiency score of the sample airline during 2017-2018.

4.2 PROPOSED SYSTEM

We want to analyze data regarding domestic flights to predict flight cancellations in the

US. We will analyze the cancellation including delay time, number of passengers, airline carrier, scheduled and actual departure and arrival times, and reason of delay.

Using these values, we plan to future calculate delays caused by the airport, get the count of the flights delayed with respect to the total number of flights etc. We also aim to calculate the average airline delay time, which airline has the greatest number of delays, at what time of the day do we see a greater number of delays in flights. We also want to see if there are any seasonal delays i.e. if delays are mostly during the holiday season. We will use Spark's MLlib to create powerful machine learning models alongside the Data Frame syntax and spark. We will set up on Amazon Web Services EC2 for big data analysis.

4.3 SYSYTEM REQUIREMENTS

4.3.1 *HARDWARE REQUIREMENTS*

- System: Pentium Dual Core.
- Hard Disk: 120 GB.
- Monitor: 15" LED
- Input Devices: Keyboard, Mouse
- Ram: 4 GB.

4.3.2 *SOFTWARE REQUIREMENTS*

- Operating system: Windows 7/10.
- Coding Language :Python
- Database: MYSQL

4.4 MODULES

- Data Profiling
- Data Wrangling
- Data Cleaning and Preparation

- Statistical Data Analysis
- Model Prediction
- Decision Tree classifiers

4.4.1 Data Profiling

To get a quick glance at the data, we loaded the 2009 csv file into a Resilient Distributed Dataset (RDD) which is Spark's representation of a dataset that is distributed across RAM, in the memory of a cluster of many machines. Through this Spark session, I found that there was about 27 unnamed variables and multiple null values. After combining the dataset (2009 to 2015 data), I created 3 major categorical CSV files namely: flight- cancellation, flight- delays, and flight-profile. This was essential to debug.

4.4.2 Data Cleaning and Preparation

Originally, there was 28 variables to work with. After removing the unnamed columns, the remaining 19 were checked for null values (as shown below). In addition, only columns with relevant data about the flight profile, delay, and cancellation information were kept. With time sensitive data, it was hard to recreate the null values with either mean or median of the data, so I had to drop these records which boils down to over 6+ million records to still work with. After cleaning out the date column, it contained only the yearly information which was used in the analysis.

4.4.3 Statistical Data Analysis

- From the profile sub-dataset, there were 61,556,964 flight profile records including 7605 US domestic flights in total, around 380 unique origins and 378 unique destinations. This information was populated in temporary tables ins 378 unique destinations. This information was populated in temporary tables inside the Spark session to query specifics.
- From the delay sub-dataset, there were negative values in the departure delay column indicating that around 3,1204,918 flights were actually ahead of their scheduled time; that is roughly 50% of the dataset. This means that 50% of the time flights were delayed.
- Within the cancellation sub-dataset, the cancel column is categorical data represented

by either a 0 or a 1 (not cancelled or cancelled respectively).

4.4.4 Model Prediction

To predict whether the flight would be cancelled or not, the dataset was modeled as a binary classification problem with categorical variables.

- Prepare the data for machine learning: use String Indexer, One Hot Encoder, and Vector Assembler to transform our features.
- Split the data into a 70/30 test and train ratio.
- Apply models: logistic regression, decision tree classifier, random forest and gradient boosted trees
- Compare all the models' accuracy for predicting cancellations.

4.4.5 Decision Tree Classifiers

Decision Tree Classifiers can handle categorical variables well and are able to capture non-linearity. With the aid of the pyspark.ml classification library, the Decision Tree Classifier was imported. This was essential to show the probability of each prediction and evaluate the model.

CHAPTER 5

SOFTWARE DEVELOPMENT METHODOLOGY

5.1 DESCRIPTION DIAGRAM

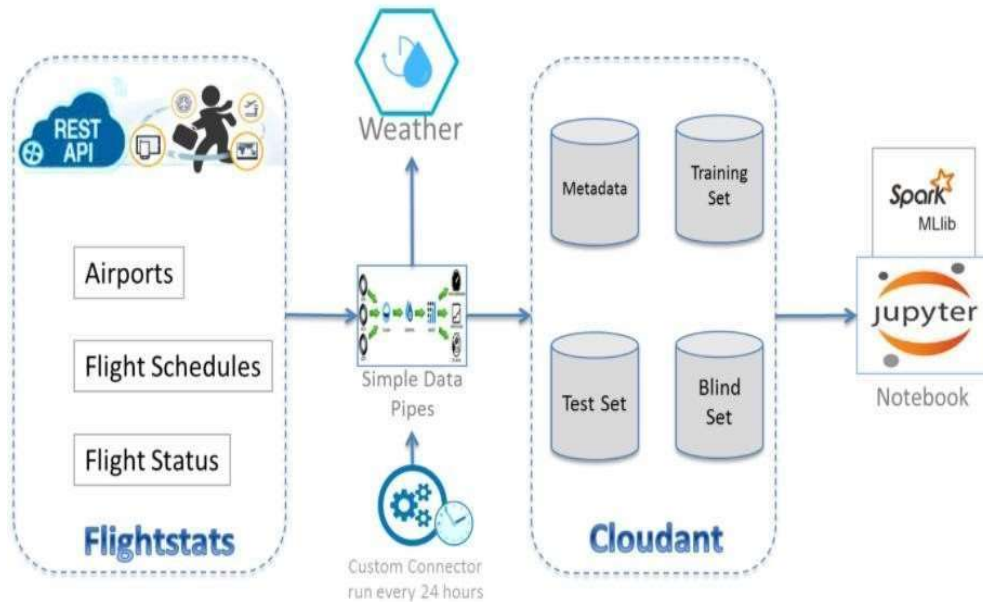


Fig 5.1: Architecture

In fig 5.1 The data set is taken from Kaggle and then send these data to the simple data pipes in that they will see whether the weather is good or suitable for travelling and the data set will be trained these can be sent to jupyter notebook.

5.2 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. A DFD is created to provide an overview of the system

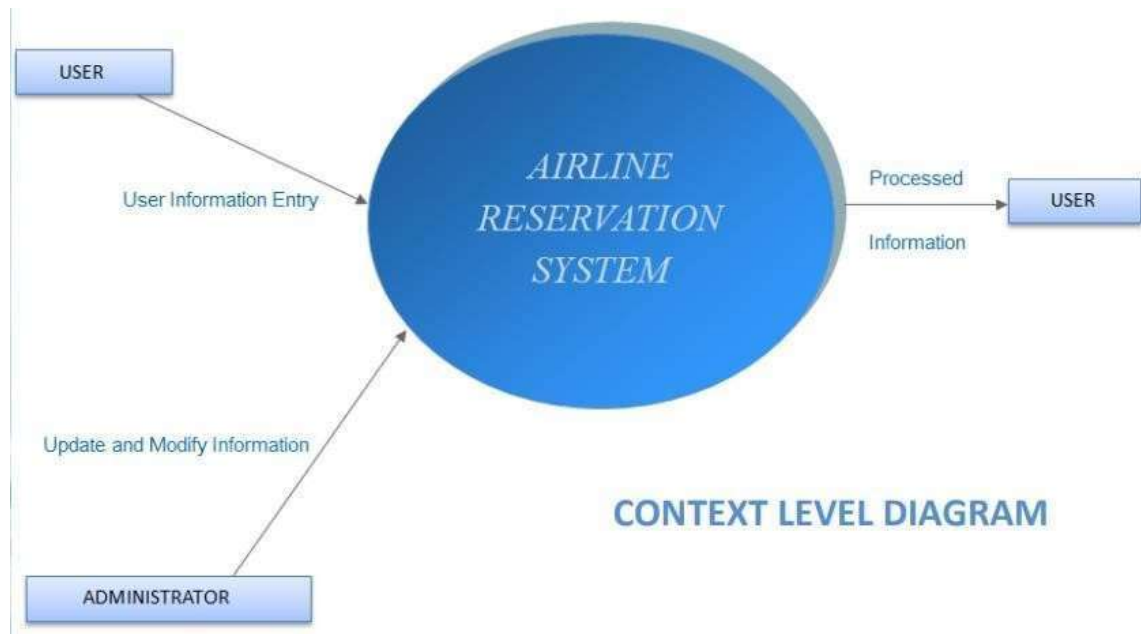


Fig 5.2. Context Level Diagram

In fig 5.2 represents the context level diagram of airline data user will give information to the system and then the administrator will update or modify the information these processed information will get the user.

5.3 ALGORITHMS

5.3.1 Random Forest Algorithm:

Random forest is a supervised classification machine learning algorithm which uses ensemble method. Simply put, a random forest is made up of numerous decision trees and helps to tackle the problem of overfitting in decision trees. These decision trees are randomly constructed by selecting random features from the given dataset. Random forest arrives at a decision or prediction based on the maximum number of votes received from the decision trees. The outcome which is arrived at, for a maximum number of times through the numerous decision trees is considered as the final outcome by the random forest.

5.3.2 Working of Random Forest:

Random forests are based on ensemble learning techniques. Ensemble, simply means a group or a collection, which in this case, is a collection of decision trees, together called as random forest. The accuracy of ensemble models is better than the accuracy of individual models due to the fact that it compiles the results from the individual models and provides a final outcome.

How to select features from the dataset to construct decision trees for the Random Forest?

Features are selected randomly using a method known as bootstrap aggregating or bagging. From the set of features available in the dataset, a number of training subsets are created by choosing random features with replacement. What this means is that one feature may be repeated in different training subsets at the same time.

5.3.3 Decision Tree Algorithm:

A decision tree is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret. Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

How does the Decision Tree algorithm work?

- The basic idea behind any decision tree algorithm is as follows:

- Select the best attribute using Attribute Selection Measures (ASM) to split the records.
- Make that attribute a decision node and breaks the dataset into smaller subsets.
- Starts tree building by repeating this process recursively for each child until one of the conditions will match:
 - All the tuples belong to the same attribute value.
 - There are no more remaining attributes.
 - There are no more instances.

5.3.4 Linear Regression:

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g., sales, price) rather than trying to classify them into categories (e.g., cat, dog). There are two main types:

Simple linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to “learn” to produce the most accurate predictions. x represents our input data and y represents our prediction.

$$y = mx + b \quad \dots 1$$

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

$$f(x,y,z) = w_1x + w_2y + w_3z \quad \dots 2$$

The variables x, y, z represent the attributes, or distinct pieces of information, we have about each observation. For sales predictions, these attributes might include a company's advertising spend on radio, TV, and newspapers.

$$\text{Sales} = w_1\text{Radio} + w_2\text{TV} + w_3\text{News} \quad \dots 3$$

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results

YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN	DESTINATION	SCHEDULE	DEPARTURE	TAXI_OUT	WHEELS_SCHEDULED	ELAPSED_AIR_TIME	DISTANCE	WHEELS_TAKEN			
2015	1	1	4	AS	98	N407AS	ANC	SEA	5	2354	-11	21	15	205	194	169	1448	404
2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	10	2	-8	12	14	280	279	263	2330	737
2015	1	1	4	US	840	N171US	SFO	CLT	20	18	-2	16	34	286	293	266	2296	800
2015	1	1	4	AA	258	N3HYAA	LAX	MIA	20	15	-5	15	30	285	281	258	2342	748
2015	1	1	4	AS	135	N527AS	SEA	ANC	25	24	-1	11	35	235	215	199	1448	254
2015	1	1	4	DL	806	N3730B	SFO	MSP	25	20	-5	18	38	217	230	206	1589	604
2015	1	1	4	NK	612	N635NK	LAS	MSP	25	19	-6	11	30	181	170	154	1299	504
2015	1	1	4	US	2013	N584UW	LAX	CLT	30	44	14	13	57	273	249	228	2125	745
2015	1	1	4	AA	1112	N3LAAA	SFO	DFW	30	19	-11	17	36	195	193	173	1464	529
2015	1	1	4	DL	1173	N826DN	LAS	ATL	30	33	3	12	45	221	203	186	1747	651
2015	1	1	4	DL	2336	N958DN	DEN	ATL	30	24	-6	12	36	173	149	133	1199	449
2015	1	1	4	AA	1674	N853AA	LAS	MIA	35	27	-8	21	48	268	266	238	2174	746
2015	1	1	4	DL	1434	N547US	LAX	MSP	35	35	0	18	53	214	210	188	1535	601
2015	1	1	4	DL	2324	N3751B	SLC	ATL	40	34	-6	18	52	215	199	176	1590	548
2015	1	1	4	DL	2440	N651DL	SEA	MSP	40	39	-1	28	107	189	198	166	1399	553
2015	1	1	4	AS	108	N309AS	ANC	SEA	45	41	-4	17	58	204	194	173	1448	451
2015	1	1	4	DL	1560	N3743H	ANC	SEA	45	31	-14	25	56	210	200	171	1448	447
2015	1	1	4	UA	1197	N78448	SFO	IAH	48	42	-6	11	53	218	217	199	1635	612
2015	1	1	4	AS	122	N413AS	ANC	PDX	50	46	-4	11	57	215	201	187	1542	504
2015	1	1	4	DL	1670	N806DN	PDX	MSP	50	45	-5	9	54	193	186	171	1426	545

Fig 6.1 Collected Data

In Fig 6.1 This is the collected data from Kaggle and then load the data.

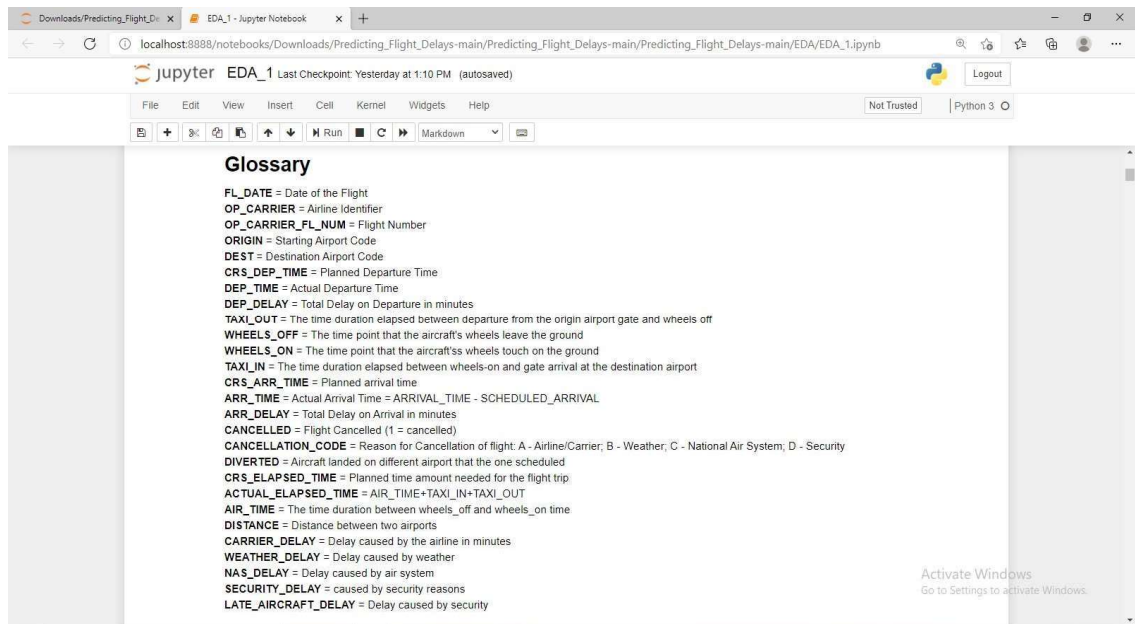


Fig 6.2 Glossary of Attributes

In fig 6.2 explains the glossary of the attributes the attributes are

FL_DATE=Date of Flight

OP_CARRIER=Airline Identifier, etc.

These are the some of the attributes used in this project.

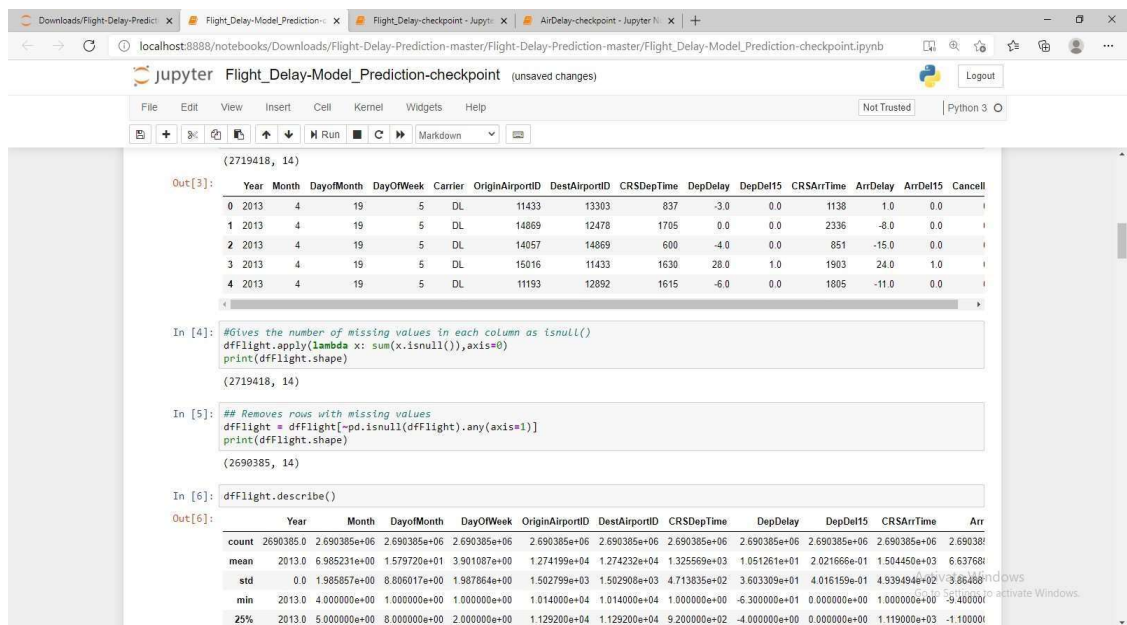


Fig 6.3 Data Loading

In fig 6.3 The collected data has been loaded.

Downloads/Predicting_Flight_D... x EDA_1 - Jupyter Notebook x +

localhost:8888/notebooks/Downloads/Predicting_Flight_Delays-main/Predicting_Flight_Delays-main/Predicting_Flight_... Logout

jupyter EDA_1 Last Checkpoint: Last Monday at 1:10 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

df_fp.head(20)

```
Out[14]:
```

	OP_CARRIER	DELAYED_FLIGHTS	TOTAL_FLIGHTS	PERCENTAGE_DELAYED
0	Alaska Airlines	89483	242979	36.83
1	Allegiant Air	39754	95192	41.76
2	American Airlines	355079	899527	39.47
3	Delta Airlines	278386	943837	29.50
4	Endeavor Air	74553	236403	31.54
5	Envoy Air	111435	284497	39.17
6	ExpressJet	71497	196583	36.37
7	Frontier Airlines	54203	117523	46.12
8	Hawaiian Airlines	30381	83375	36.44
9	JetBlue Airways	131673	297758	44.22
10	Mesa Airline	84277	209094	40.31
11	PSA Airlines	113097	265784	42.55
12	Republic Airways	105360	305251	34.52
13	SkyWest Airlines	260846	760796	36.91
14	Southwest Airlines	543141	1331609	40.79
15	Spirit Airlines	59486	174129	34.16
16	United Airlines	225978	614915	36.75

Type here to search 09:18 24-02-2021

Fig 6.4 Percentage Delayed by Airlines

In fig 6.4 These are the Percentage delayed by the airlines with flight identifier, delayed flights, total flights.

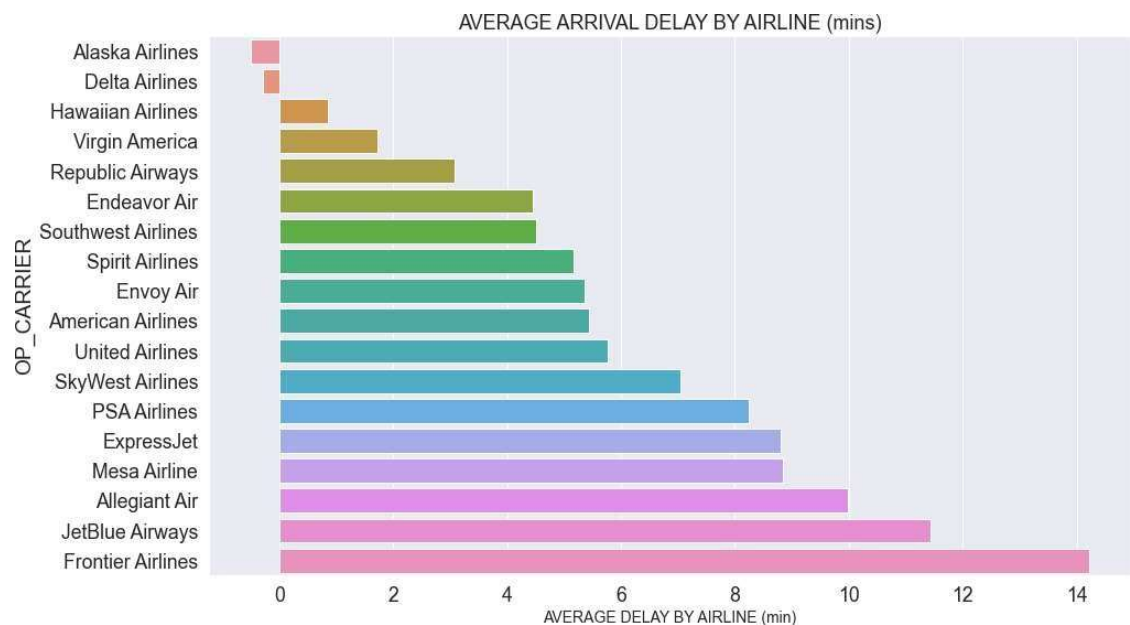


Fig 6.5 Average Arrival Delay by airline

In fig 6.5 explains the average delay by airline

X axis-average delay by airline

Y axis-Flight Identifier

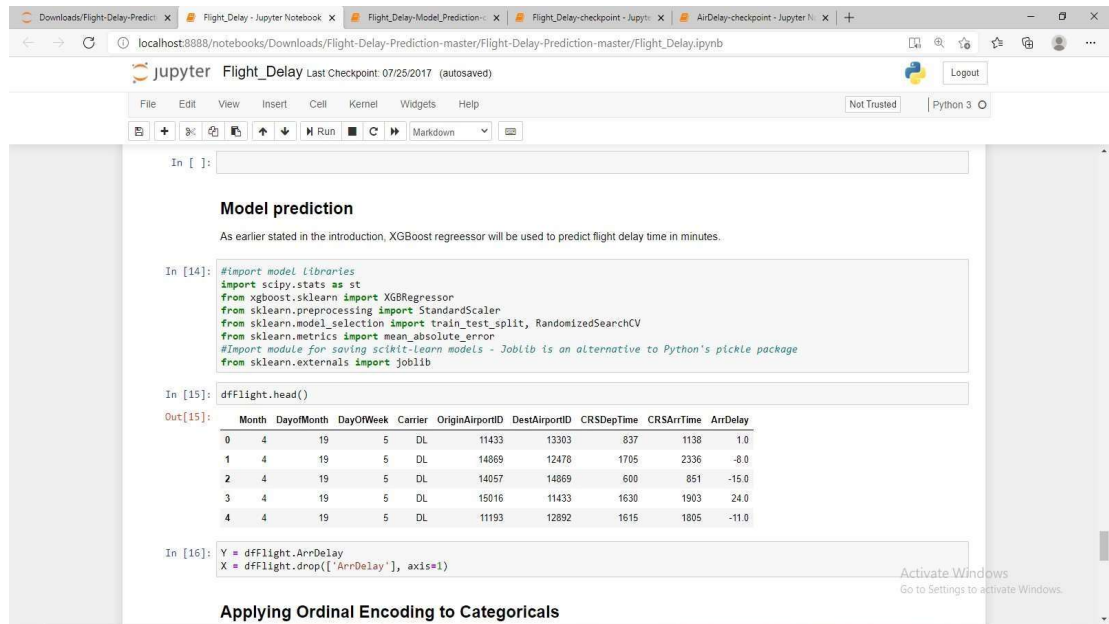


Fig 6.6 Model Prediction

In fig 6.6 Prediction of flight whether it will cancel or delayed.

CHAPTER 7

CONCLUSION

CONCLUSION

The project was able to identify whether the airline is likely to cancel with an accuracy of 63%. GBT performed the best for this case. From the data, the most likely flight to be cancelled was ExpressJet Airlines and the least likely was Delta Airlines. However, airline identity was not a meaningful predictor of delay time. This suggests that delay of flight is likely to be for reasons other than the airline, such as weather conditions or airport.

REFERENCES

- [1] Bureau of Transportation Statistics. (2016). Airline On-Time Performance and Causes of Flight Delays. Retrieved from <https://catalog.data.gov/dataset/airline-on-time-performance-and-causes-of-flight-delays-on-time-data>
- [2] Belcastro, L. & Marozzo, Fabrizio & Talia, Domenico & Trunfio, Paolo. (2016). Using Scalable Data Mining for Predicting Flight Delays. ACM Transactions on Intelligent Systems and Technology. 8. 10.1145/2888402. Retrieved from <https://dl.acm.org/doi/10.1145/2888402>
- [3] Chakrabarty, Navoneel, et al. "Flight Arrival Delay Prediction Using Gradient Boosting Classifier." Emerging Technologies in Data Mining and Information Security. Springer, Singapore, 2019. 651-659. Retrieved from https://www.researchgate.net/publication/327389509_Flight_Arrival_Delay_Prediction_Using_Gradient_Boosting_Classifier
- [4] Deshpande, V., & Arikan, M. (2011). The Impact of Airline Flight Schedules on Flight Delays. *Manufacturing & Service Operations Management*, 14, 423-440. Retrieved from <https://pubsonline.informs.org/doi/10.1287/msom.1120.0379>
- [5] Mu, Y. (2019, August). Airline Delay and Cancellation Data, 2009 - 2018. Retrieved April 2020 from <https://www.kaggle.com/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018/data>
- [6] Yi Ding "Predicting flight delay based on multiple linear regression", IOP Conference Series: Earth and Environmental Science. Retrieved from <https://iopscience.iop.org/article/10.1088/1755-1315/81/1/012198>

APPENDIX

A. PLAGARISM REPORT

JOURNAL PAPER FOR AIRLINE DATA ANALYSIS.pdf - Adobe Acrobat Reader DC (32-bit)

File Edit View Sign Window Help

Home Tools JOURNAL PAPER F... x

Search 'Strikethrough'

Export PDF

Adobe Export PDF

Convert PDF Files to Word or Excel Online

Select PDF File

JOURNAL P...ALYSIS.pdf

Convert to

Microsoft Word (*.docx)

Document Language:

English (U.S.) Change

Convert, edit and e-sign PDF forms & agreements

Free 7-Day Trial

JOURNAL PAPER FOR AIRLINE DATA ANALYSIS

ORIGINALITY REPORT

13% 10% 7% 8%

SIMILARITY INDEX INTERNET SOURCES PUBLICATIONS STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.net	Internet Source	3%
2	Submitted to Humble Independent School District	Student Paper	2%
3	Chang-Hun Lee, Hyo-Sang Shin, Antonios Tsourdos, Zakwan Skaf. "Data analytics development of FDR (Flight Data Recorder) data for airline maintenance operations", 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2017	Publication	1%
4	Submitted to BITS, Pilani-Dubai	Student Paper	1%
5	Submitted to University of Edinburgh	Student Paper	1%
6	na'il mitchell. "Modeling Variance: A Variance-Motivated Approach to Molecular Prediction", American Chemical Society (ACS), 2020	Publication	1%

Type here to search

10:32 10-03-2021

B. JOURNAL PAPER

AIRLINE DATA ANALYSIS USING SPARK TECHNOLOGIES

Karthik K¹, Sumanth Kumar K², Dr S.Prince Mary³

^{1*,2}UG Student, Department of Computer Science and Engineering,
Sathyabama Institute Of Science and Technology, Chennai ,India

³Assistant Professor, Department of Computer Science and Engineering ,
,Sathyabama Institute Of Science and Technology, Chennai ,India

Abstract— Our project analyzes the US domestic flights dataset using pyspark dataframes. We aim to predict which flight/flight carrier is most likely to be cancelled or delayed. Using Pyspark, a big data technology tool, we can therefore work with RDDs/Dataframes/Datasets in Python. Our analysis will help us find out what are the top airlines with the maximum delays/cancellations in the US. Also, our project will investigate whether there are any factors that cause flight delays/cancellations. Further, our project predicts if a flight will get cancelled or not. In order to build the predictive system Machine learning models were incorporated, where we modeled a binary classification problem with categorical variables, logistic regression, decision tree classifier, random forest and gradient boosted trees.

Keywords—Jupyterlab,Data,Airline,Logistic Regression,

cancellation (tagged as a new reason code) It also contained technical details such as the time it was empty and the aircraft on the ground. In my approach and Apache Spark, especially when using PySpark AWS EMR cluster support. But this can be the power of the use of it is, it is divided falls, for many orders of magnitude better than in the knowledge of the structure of memory struggled with a lot of laborious literature. As an analysis engine that allows us to parallelize an important role in information processing, clusters are distributed. Even the great football PySpark Exploratory Data Machine Learning caurūvadiem division, construction and creation of etls tools (extract, transform, load). Once the dried EMR is “running” it is not available on AWS Jupyterlab Notebooks and Data Services from the loaded S3 trails as designated by the Kaggle website. I performed exploratory analysis and model prediction.

I. INTRODUCTION:

And Kaggle on NBC from 2009 to 2015, included 7 CSV files with data and a size of 7 GB. Airlines subject to delay information, location details (origins and locations) and

II. EXISTING SYSTEM:

In Existing system, an airport has huge amount of data related to number of flights, data and time of arrival and dispatch, flight routes, number of airports operating in each country, list of active airlines in each country. The

problem they faced till now it's, they have ability to analyse limited data from databases.

III. LITERATURE SURVEY:

Studying the literature is the most important step in the software development process. Before any explanation is necessary to determine the time of the iron element, economy and the power of the county court. With all of these things, the next step is to figure out which operating system language will be a development tool. When programmers need to start building a tool for programmers, a lot of outside help. This assistance can be obtained from older programmers, from a book or from websites. Before considering a building system developer to develop the argument. It considers the needs and development of the development project of the region of the Most of everything is a necessary condition to fully study the position on the project. The standard area of the survey project is particularly important in the software development process. Before developing the tools they need to determine design time and component resource requirements, a utility company saves money. When all of this is they have it fully in mind, and the next step is to establish the truth about the software specifications in both systems, such as what type of operating system the project does not even have one that they are all necessary software are the work of each other as a stage in the development of technology and things associated.

Data analytics development of FDR (Flight Data Recorder) data for airline maintenance operations:

Our data was based on the development of new flight models detecting huge amounts of FDR (Flight Data Recorder) intentionally promoting the airline's departure. By their fruits you after the things of this kind of promotion is the rational that if the potential to be in the flight of the aircraft by means of mechanical parts, it can be most likely to be confirmed, especially in the case of things included in the FDR data. Therefore, analyzing the FDR data gives us the information to detect potential problems as possible before the competition. To this end, the step of data preprocessing, data filtering, data transformation and data sampling are continuously perfected. In addition, with regard to all the given series, the analysis of the FDR in the discrete types of the sign of

the three types of signal being immediately warned. By arranging the space of time of great dimension, a vector is chosen, the sign of each type of the nature of the given series. In the section of the process function, a correlation analysis, the nature of the relaxation and a reduction of the pattern is performed. In short, knowing the nature of things, very close to the k-power approach to data, is sufficiently FDR FDR avoiding evil, of which it is an example for our customers. The proposed method is tested by a realistic FDR of data from the NASA database is open.

Big Data Analytics on Aviation Social Media: The Case of China Southern Airlines on Sina Weibo:

The proposed model; (3) the meaning of the China Southern Airlines case study analysis on WeiBer Mountain, WeiB users and illustrating attitudes towards China Southern Airlines. The study of this attempt, too, to provide and in letters that descend to the custom of airlines, and that the use of their social media platforms. Through the combination of traveling travelers and other social media, more information online, including passengers, can create a profile.

Big Data Analytics in Airlines: Efficiency Evaluation using DEA

The process of achieving his knowledge of the implementation analysis at the end of the execution of the large scheduling of this quantitatively assesses the effectiveness of his innovative research. A set of calculations obtained from sad studies before. These parameters are calculated from the data envelopment analysis (DEA) to measure the efficiency scores for each of the processing operations. Finally, that it is beneficial to the trend data analysis approach will be implemented to decrease the efficiency score will find a sample of airlines, airline from 2017-2018.

IV. PROPOSED SYSTEM:

We want to analyze data regarding domestic flights to predict flight cancellations in the US. We will analyze the cancellation including delay time, number of passengers, airline carrier, scheduled and actual departure and arrival

times, and reason of delay. Using these values we plan to future calculate delays caused by the airport, get the count of the flights delayed with respect to the total number of flights etc. We also aim to calculate the average airline delay time, which airline has the most number of delays, at what time of the day do we see more number of delays in flights. We also want to see if there are any seasonal delays i.e if delays are mostly during the holiday season. We will use Spark's MLlib to create powerful machine learning models alongside the DataFrame syntax and spark. We will set up on Amazon Web Services EC2 for big data analysis.

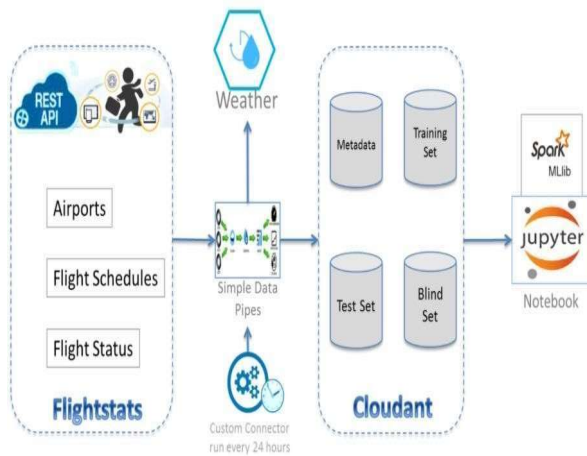


Fig. 1 Proposed System Architecture

A. ADVANTAGES OF PROPOSED SYSTEM:

- By using airline data analysis we can save time of users
- Cost Savings

V. MODULES:

- Data Profiling
- Data Wrangling
- Data Cleaning and Preparation
- Statistical Data Analysis

- **Model Prediction**
- Decision Tree classifiers

1. Data Profiling

For a quick overview that was given to us in 2009, the CSV file loaded into the Spark Distributed Dataset (RDD) is an image, which is distributed by NBC, across the memory cluster to many machines. During the Spark session where I found that a large portion of the 27 variables with no name zero value. After combining NBC (data from 2009 to 2015), created a categorical CSV file greater than 3, the new flight, the flight delays and the flight profile. This is essential for debugging.

2. Data Wrangling

First, the 28 variables as needed. After removing the unnamed columns, the remaining 19 were checked for null values (see below). In addition, the relevant data and columns regarding the flight delay profile data records were. With time sensitive data that it would be difficult to recreate the mean or the median, or zero values in the data, so I had the records, which are still injuries to lose on the job and over 6 million d 'records. After a day in a mainland cleaning pillar, but throughout the year said that as far as, not only in the data analysis.

3. Data Cleaning and Preparation

First, the 28 variables as needed. After removing the unnamed columns, the remaining 19 were checked for null values (see below). In addition, the relevant data and columns regarding the flight delay profile data records were. With time sensitive data that it would be difficult to recreate the mean or the median, or zero values in the data, so I had the records, which are still injuries to lose on the job and over 6 million d 'records. After a day in a mainland cleaning pillar, but throughout the year said that as far as, not only in the data analysis.

4. Statistical Data Analysis

A sub-NBC profile, there were 61,556,964 profile records including the US domestic flight (7) Flight 605 in a single sum of approximately 380 origins and 378 unique bursts. This information was disseminated during the session to query the specifics spark tables. The delay of the data subset, negative values of Ω delay the departure flight actually ahead of the column, indicating that their 3,1204,918 about an hour; or about 50% of the dataset. This means 50% of the time

the flight was delayed. In the money sub-NBC, the column is adamant that the invalid data is either 0 or a 1 (not canceled or canceled respectively).

5. Model Prediction

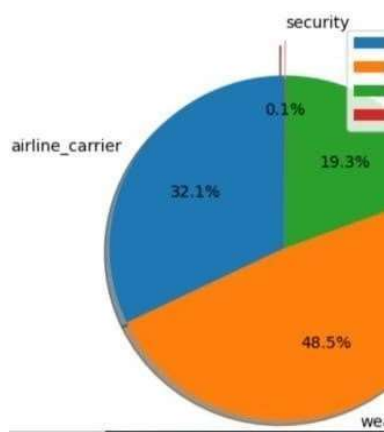
To predict whether the flight would be cancelled or not, the dataset was modeled as a binary classification problem with categorical variables.

5. Prepare the data for machine learning: use StringIndexer, OneHotEncoder, and VectorAssembler to transform our features
6. Split the data into a 70/30 test and train ratio
7. Apply models: logistic regression, decision tree classifier, random forest and gradient boosted trees
8. Compare all the models' accuracy for predicting cancellations.

6. Decision Tree classifiers

Tree classifiers will not be able to fully understand and process categorical variables. With the help of the spark.ml.classification library, which is imported Decision Tree Classifier. This, however, belongs to the nature of the probability of showing that he spoke well, and I'll give you an example.

VI. SCREENSHOTS:



VII. CONCLUSION:

The project is to try to identify if the airline cancels 63% accuracy. GBT best thing to do in this case. Data is likely to be cut. The Greek means the bow and arrow of the ExpressJet flight. However, the identity of the airline is not a long-to-say significant predictor. In this way, the future, to the violence of the flight delay, which airline or success.

VIII. REFERENCES:

- [1] Bureau of Transportation Statistics. (2016). Airline On-Time Performance and Causes of Flight Delays. Retrieved from <https://catalog.data.gov/dataset/airline-on-time-performance-and-causes-of-flight-delays-on-time-data>
- [2] Deshpande, V., & Arikan, M. (2011). The Impact of Airline Flight Schedules on Flight Delays. *Manufacturing & Service Operations Management*, 14, 423-440. Retrieved from <https://pubsonline.informs.org/doi/10.1287/msom.1120.0379>
- [3] Mu, Y. (2019, August). Airline Delay and Cancellation Data, 2009 - 2018. Retrieved April 2020 from <https://www.kaggle.com/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018/data>
- [4] Chakrabarty, Navoneel, et al. "Flight Arrival Delay Prediction Using Gradient Boosting Classifier." *Emerging Technologies in Data Mining and Information Security*. Springer, Singapore, 2019. 651-659. Retrieved from https://www.researchgate.net/publication/327389509_Flight_Arrival_Delay_Prediction_Using_Gradient_Boosting_Classifier

[5] Yi Ding "Predicting flight delay based on multiple linear regression", IOP Conference Series: Earth and Environmental Science. Retrieved from <https://iopscience.iop.org/article/10.1088/1755-1315/81/1/012198>

[6] Belcastro, L. & Marozzo, Fabrizio & Talia, Domenico & Trunfio, Paolo. (2016). Using Scalable Data Mining for Predicting Flight Delays. ACM Transactions on Intelligent Systems and Technology. 8. 10.1145/2888402. Retrieved from <https://dl.acm.org/doi/10.1145/2888402>

[7] Li, S. "Machine Learning with PySpark and MLlib — Solving a Binary Classification Problem". Towards Data Science. Retrieved from <https://towardsdatascience.com/machine-learning-with-pyspark-and-mllib-solving-a-binary-classification-problem-96396065d2aa>

C. Source Code

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "## Exploratory Data Analysis for 2009 - 2015 Flight Delays and
        Cancellations"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "#### Data Profiling "
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {},
```

```

"source": [
  "##### Import required libraries"
]
},
{
  "cell_type": "code",
  "execution_count": 1,
  "metadata": {},
  "outputs": [],
  "source": [
    "import findspark\n",
    "findspark.init()\n",
    "\n",
    "import pyspark\n",
    "from pyspark.sql.session import SparkSession\n",
    "sc                                     =
pyspark.SparkContext(appName=\"myAppName2\").getOrCreate()\n",
    "spark = SparkSession(sc)"
  ]
},
{
  "cell_type": "code",

```

```

"execution_count": 2,

"metadata": {},

"outputs": [],

"source": [

    "from pyspark.sql.types import *\n",

    "from pyspark.sql import functions as F\n",

    "from pyspark.sql.functions import isnan, when, count, col\n",

    "import pandas as pd "

]

},

{

    "cell_type": "markdown",

    "metadata": {},

    "source": [

        "##### Create Schema"

    ]

},

{

    "cell_type": "markdown",

    "metadata": {},

    "source": [

        "##### Flight Profile"

```

```

]
},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [
    "##### Chunking Data"
  ]

```

```

},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [

```

"Our null analysis shows that the following features have a lot of null values: We will delete these rows, and omit some unnecessary columns as well. "

```

]
},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [

```

```

    "Verifying that there are no more null columns"

]
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "##### Flight Cancellations"
    ]
},
{
    "cell_type": "code",
    "execution_count": 3,
    "metadata": {},
    "outputs": [],
    "source": [
        "cancel_schema = StructType([StructField('date', StringType(),
True),\n",
        "    StructField('flight_number', StringType(), True),\n",
        "    StructField('flight_identifier', StringType(), True),\n",
        "    StructField('cancel', StringType(), True),\n",
        "    StructField('reason', StringType(), True)])"
    ]
}

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "root\n",
        " |-- date: string (nullable = true)\n",
        " |-- flight_number: string (nullable = true)\n",
        " |-- flight_identifier: string (nullable = true)\n",
        " |-- cancel: string (nullable = true)\n",
        "\n"
      ]
    }
  ],
  "source": [

```

```

cancel_df = spark.read.format('csv').option(\"mode\",
\"DROPMALFORMED\").load('flight-cancellation.csv', header='true',
schema=cancel_schema) \n",

cancel_df = cancel_df.drop('reason')\n",

cancel_df = cancel_df.dropna()\n",

cancel_df = cancel_df.sample(.1)\n",

cancel_df.printSchema()

]

},

{

"cell_type": "code",

"execution_count": 5,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"6151854"

]

},

"execution_count": 5,

"metadata": {},

```

```

    "output_type": "execute_result"

}

],

"source": [

    "cancel_df.count()"

]

},

{

    "cell_type": "code",

    "execution_count": 6,

    "metadata": {},

    "outputs": [],

    "source": [

        "from pyspark.ml.feature import OneHotEncoderEstimator,\nStringIndexer, VectorAssembler\n",

        "categoricalColumns = [\"date\", \"flight_number\",\n\"flight_identifier\"]\n",

        "#indexers = [StringIndexer(inputCol=column, outputCol=column +\n\"_index\").fit(cancel_df) for column in [\"date\", \"flight_number\",\n\"flight_identifier\", \"cancel\", \"reason\"]]"

    ]

},

{

```

```

"cell_type": "code",

"execution_count": 7,

"metadata": {},

"outputs": [],

"source": [

    "stages = []\n",

    "\n",

    "for categoricalCol in categoricalColumns:\n",

        "    stringIndexer = StringIndexer(inputCol = categoricalCol, outputCol
= categoricalCol + 'Index')\n",

        "    encoder =
OneHotEncoderEstimator(inputCols=[stringIndexer.getOutputCol()],
outputCols=[categoricalCol + \"classVec\"])\n",

        "    stages += [stringIndexer, encoder]\n",

        "\n",

        "label_stringIdx = StringIndexer(inputCol = 'cancel', outputCol =
'label')\n",

        "stages += [label_stringIdx]"

    ]

},

{

    "cell_type": "code",

    "execution_count": 8,

```

```

"metadata": {},

"outputs": [],

"source": [

    "assemblerInputs = [c + \"classVec\" for c in categoricalColumns]\\n",

    "assembler      =      VectorAssembler(inputCols=assemblerInputs,
outputCol=\"features\\")\\n",

    "stages += [assembler]"

]

},

{

    "cell_type": "code",

    "execution_count": 9,

    "metadata": {},

    "outputs": [],

    "source": [

        "cols = ['date', 'flight_number',\\n",

        " 'flight_identifier',\\n",

        " 'cancel']"

    ]

},

{

    "cell_type": "code",

```

```

"execution_count": 10,

"metadata": {},

"outputs": [

{

  "name": "stdout",

  "output_type": "stream",

  "text": [

    "root\n",

    " |-- label: double (nullable = false)\n",

    " |-- features: vector (nullable = true)\n",

    " |-- date: string (nullable = true)\n",

    " |-- flight_number: string (nullable = true)\n",

    " |-- flight_identifier: string (nullable = true)\n",

    " |-- cancel: string (nullable = true)\n",

    "\n"

  ]

}

],

"source": [

  "from pyspark.ml import Pipeline\n",

  "pipeline = Pipeline(stages = stages)\n",

  "pipelineModel = pipeline.fit(cancel_df)\n",

```

```

"cancel_df = pipelineModel.transform(cancel_df)\n",
"selectedCols = ['label', 'features'] + cols\n",
"cancel_df = cancel_df.select(selectedCols)\n",
"cancel_df.printSchema()"
]
},
{
"cell_type": "code",
"execution_count": 11,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"+___+_____+__+_____+_____+___+\n",
"|label|          features|date|flight_number|flight_identifier|cancel|\n",
"+___+_____+__+_____+_____+___+\n",
"| 0.0|(7566,[2,560,7555...|2009|      1212|          XE|    0|\n",
"| 0.0|(7566,[2,1076,755...|2009|      1236|          XE|    0|\n",
"| 0.0|(7566,[2,1494,755...|2009|      2002|          XE|    0|\n",
"| 0.0|(7566,[2,1827,755...|2009|      2010|          XE|    0|\n",

```

```

"| 0.0|(7566,[2,1629,755...|2009|      2029|      XE|  0|\n",
"+____+_____+__+_____+_____+____+\n",
"only showing top 5 rows\n",
"\n"
]
}
],
"source": [
  "cancel_df.show(5)"
]
},
{
  "cell_type": "code",
  "execution_count": 12,
  "metadata": {},
  "outputs": [],
  "source": [
    "train, test = cancel_df.randomSplit([0.7, 0.3], seed = 2018)"
  ]
},
{
  "cell_type": "code",

```

```
"execution_count": 13,  
"metadata": {},  
"outputs": [],  
"source": [  
    "from pyspark.ml.classification import LogisticRegression\n",  
    "lr = LogisticRegression(featuresCol = 'features', labelCol = 'label')\n",  
    "lrModel = lr.fit(train)"  
]  
},
```