Project Report

On

An Intelligent Approach to Flight Delay Prediction & Analysis

Submitted in partial fulfilment for the award of

Post Graduate Diploma in Big Data Analytics (PG-DBDA)

From Know-IT (Pune)

Guided by:

Mr. Tushar Kute

Mrs. Trupti Joshi

Submitted By:

Chaitanya Larokar (220943025010)
Vaibhav Gavli (220943025052)
Pushpak Paunikar (220943025029)
Akash Kunjir (220943025016)

Centre of Development of Advanced Computing (C-DAC), Pune

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that.

Chaitanya Larokar (220943025010)

Vaibhav Gavli (220943025052)

Pushpak Paunikar (220943025029)

Akash Kunjir (220943025016)

Have Successfully completed their project on

An Intelligent Approach to Flight Delay Prediction & Analysis

Under the guidance of Mr. Tushar Kute and

Mrs. Trupti Joshi

Centre of Development of Advanced Computing (C-DAC), Pune

ACKNOWLEDGEMENT

This project "An Intelligent Approach to Flight Delay Prediction & Analysis" was a great learning experience for us and we are submitting this work to CDAC Know-IT (Pune).

We all are very glad to mention the name of **Mr. Tushar Kute** and **Mrs. Trupti Joshi** for their valuable guidance to work on this project. Their guidance and support helped us to overcome various obstacles and intricacies during project work.

We are highly grateful to **Mr. Vaibhav Inamdar** Manager (Know-IT), C-DAC, for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data Analytics (PG-DBDA) through C-DAC ACTS, Pune.

Our most heartfelt thanks goes to **Mrs. Bakul Joshi** (Course Coordinator, PG-DBDA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC Know-IT, Pune.

From:
Chaitanya Larokar (220943025010)
Vaibhav Gavli (220943025052)
Pushpak Paunikar (220943025029)
Akash Kunjir (220943025016)

Centre of Development of Advanced Computing (C-DAC), Pune

TABLE OF CONTENTS

Abstract

Flight scheduling has been a problem since the dawn of air travel and is something that airline companies wish to tackle. For an airport to be able to schedule the flights such that they reach on time, they must be able to tell if the flight will arrive on time or not. A flight is said to be delayed if the flight either takes off or arrives later than the scheduled time. This Project predicts whether if the flight will arrive delayed or not, after the flight's departure, and if the flight is classified as arriving late, then the arrival delay in minutes is predicted. This project proposes a Two Stage Predictive Machine Learning Engine that can classify delayed flights and predict the arrival delay period after take-off using corresponding flight information.

1. Introduction and Overview of Project

Scope

Since the inception of commercial air travel, the number of people travelling by air has increased drastically, with an increase of 42 % in the last decade alone. This means that there will be even more air traffic than usual at a given point of time and hence scheduling flights will be a colossal problem for the Aviation Department.

When a flight is delayed it will cause issues for the customers in the form of loss of money and time. Not only does it disturb the lives of the customers travelling by air commercially, but it also destroys the integrity of the airline company. Flights can be delayed due to various reasons, one of them being, extreme weather conditions. Since it is possible for the Aviation Department to estimate the weather conditions after the flight departs it may help them schedule flights better and hence reduce air traffic and make commercial air travel smooth.

Hence it is critical to be able to predict if a flight will be delayed or not and if delayed by how long.

About the Project

This project examines the impact of various conditions on the arrival delay for 15 domestic flights in the United States. It uses a two-stage machine learning model to classify and predict the arrival delays of various flights in 15 different airports during the years 2019 - 2021. The machine learning engine’s Classification and Regression algorithms are then evaluated with standard metrics and hence compared.

Research Motivation

Average aircraft delay is regularly referred to as an indication of airport capacity. Flight delay is a prevailing problem in this world. It's very tough to explain the reason for a delay. A few factors responsible for the flight delays like runway construction to excessive traffic are rare, but bad weather seems to be a common cause. Some flights are delayed because of the reactionary delay.

2. Data Description and Technologies Implemented

Data used in the project is structured in nature from year 2019 to 2021. It was collected from Bureau of Transportation and Statistic (BTS) The main goal of the analysis is to be build accurate and robust classification and Regression models to predict the outcome of a Delay Time. This research uses Logistic Regression, Linear Regression, Random Forest, Decision Tree.

Flight Delay Data

This data set contains the performance for various flights over the years 2019 and 2021. The airports and flight attributes taken into consideration are given in Table 1 and Table 2 respectively.

Table 1 Airports taken into consideration.

ABY	AEX	BDL	BHM	CAE
DAY	EWR	FSD	GRR	ILM
JAX	MDT	MQT	OMA	SDF

Table 2 Flight attributes taken into consideration.

Carrier_Delay	Arr_Camcelled	Year	Month	Arr_Diverted
Airport_Name	Depdel15	Weather_Delay	Depdelayminutes	Originairportid
Late_Aircraft_Delay	Arrtime	Weather_Ct	Nas_Delay	Security_Delay

3. Information on the technologies being used

1. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

2. Machine Learning

Machine Learning tutorial provides basic and advanced concepts of machine learning. Our machine learning tutorial is designed for students and working professionals.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

This machine learning tutorial gives you an introduction to machine learning along with the wide range of machine learning techniques such as Supervised, Unsupervised, and Reinforcement learning.

You will learn about regression and classification models, clustering methods, hidden Markov models, and various sequential models.

3. Pandas

Pandas is a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors.

Pandas is built on top of two core Python libraries matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, pandas.plot() combines multiple matplotlib methods into a single method, enabling you to plot a chart in a few lines.

Before pandas, most analysts used Python for data munging and preparation, and then switched to a more domain specific language like R for the rest of their workflow. Pandas introduced two new types of objects for storing data that make analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and DataFrames, which have a tabular structure.

4. Spark

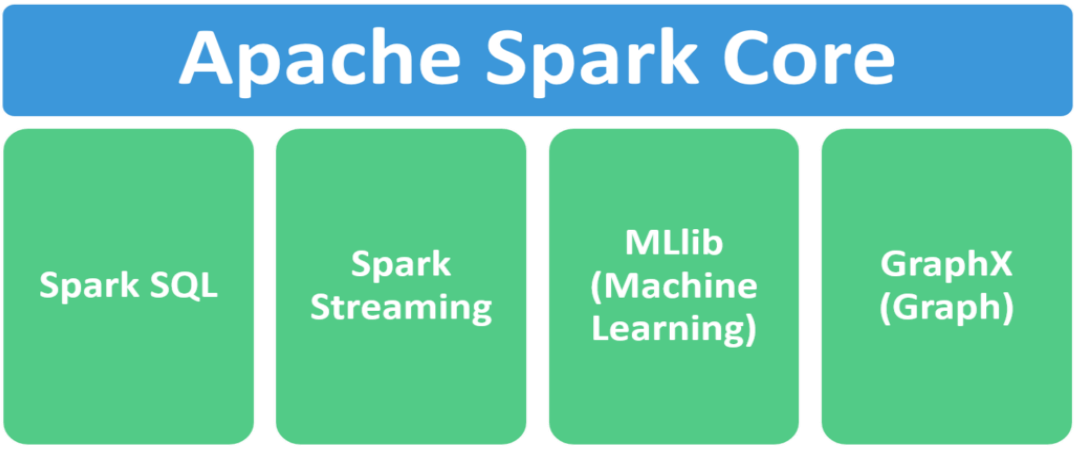
Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching and optimized query execution for fast queries against data of any size. Simply put, Spark is a fast and general engine for large-scale data processing.

The fast part means that it’s faster than previous approaches to work with Big Data like classical MapReduce. The secret for being faster is that Spark runs on memory (RAM), and that makes the processing much faster than on disk drives.

The general part means that it can be used for multiple things like running distributed SQL, creating data pipelines, ingesting data into a database, running Machine Learning algorithms, working with graphs or data streams, and much more.

Components of Spark:

1. **Apache Spark Core** – Spark Core is the underlying general execution engine for the Spark platform that all other functionality is built upon. It provides in-memory computing and referencing datasets in external storage systems.
2. **Spark SQL** – Spark SQL is Apache Spark's module for working with structured data. The interfaces offered by Spark SQL provides Spark with more information about the structure of both the data and the computation being performed.
3. **Spark Streaming** – This component allows Spark to process real-time streaming data. Data can be ingested from many sources like Kafka, Flume, and HDFS (Hadoop Distributed File System). Then the data can be processed using complex algorithms and pushed out to file systems, databases, and live dashboards.
4. **MLlib (Machine Learning Library)** – Apache Spark is equipped with a rich library known as MLlib. This library contains a wide array of machine learning algorithms- classification, regression, clustering, and collaborative filtering. It also includes other tools for constructing, evaluating, and tuning ML Pipelines. All these functionalities help Spark scale out across a cluster.
5. **GraphX** – Spark also comes with a library to manipulate graph databases and perform computations called GraphX. GraphX unifies ETL (Extract, Transform, and Load) process, exploratory analysis, and iterative graph computation within a single system.



Features of Spark

1. **Fast processing** – The most important feature of Apache Spark that has made the big data world choose this technology over others is its speed. Big data is characterized by volume, variety, velocity, and veracity which needs to be processed at a higher speed. Spark contains Resilient Distributed Dataset (RDD) which saves time in reading and writing operations, allowing it to run almost ten to one hundred times faster than Hadoop.
2. **Flexibility** – Apache Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python.
3. **In-memory computing** – Spark stores the data in the RAM of servers which allows quick access and in turn accelerates the speed of analytics.
4. **Real-time processing** – Spark can process real-time streaming data. Unlike MapReduce which processes only stored data, Spark can process real-time data and is, therefore, able to produce instant outcomes.
5. **Better analytics** – In contrast to MapReduce that includes Map and Reduce functions, Spark includes much more than that. Apache Spark consists of a rich set of SQL queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed in a better fashion with the help of Spark.

5. Tableau

Data visualization is the graphical representation of information and data. It helps create interactive elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Tableau is widely used for Business Intelligence but is not limited to it. It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights. All of this is made possible with gestures as simple as drag and drop.

4. Data Cleaning Process

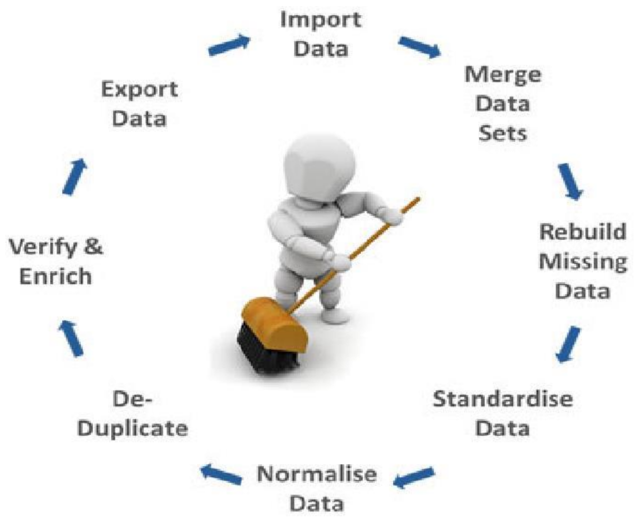
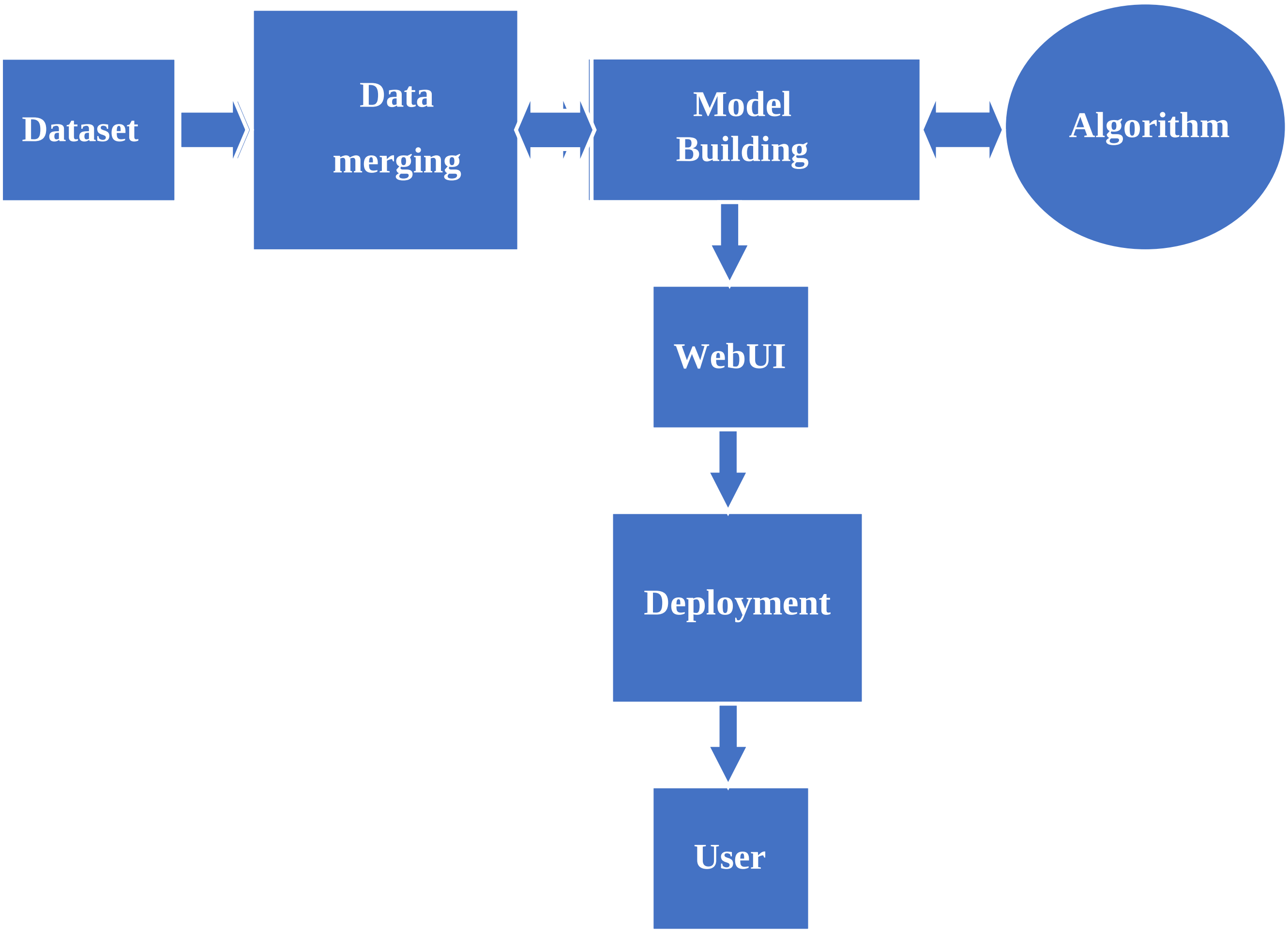


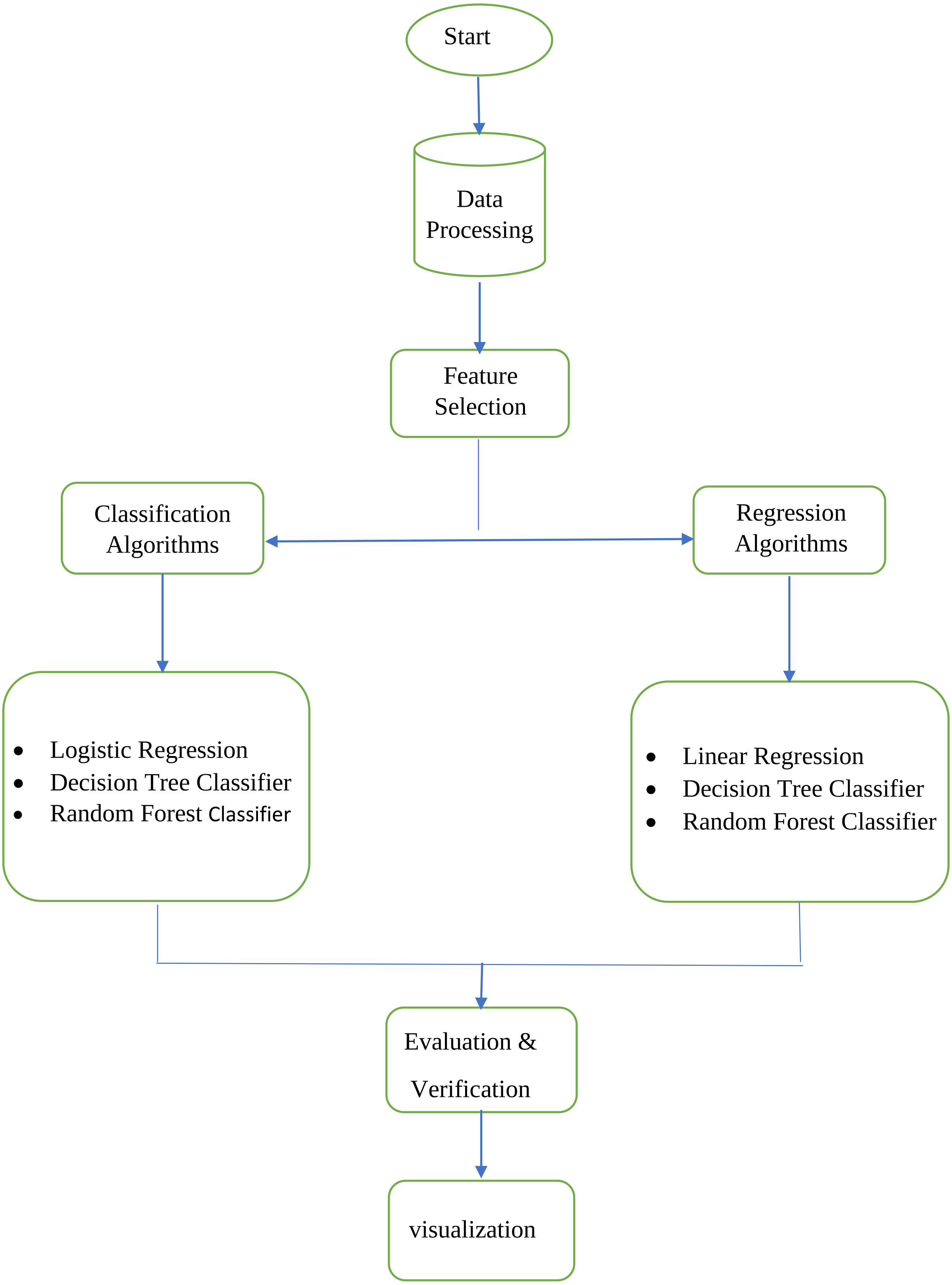
Fig: Data Cleaning Process

Data cleansing or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting. After cleansing, a data set should be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores. Data cleaning differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at the time of entry, rather than on batches of data.

5. Flow Diagram



Methodology



6. Machine Learning Algorithms

The following algorithms have been used and evaluated.

1. Logistic Regression
2. Random Forest
3. Decision Trees
4. Linear Regression

1. Logistic Regression

- Logistic Regression is a Supervised statistical technique to find the probability of dependent variable (Classes present in the variable).
- Logistic regression uses functions called the logit functions, that helps derive a relationship between the dependent variable and independent variables by predicting the probabilities or chances of occurrence.
- The logistic functions (also known as the sigmoid functions) convert the probabilities into binary values which could be further used for predictions.

Types of Logistic Regression:

1. Binary Logistic Regression: The dependent variable has only two 2 possible outcomes/classes. Example-Male or Female.
2. Multinomial Logistic Regression: The dependent variable has only two 3 or more possible outcomes/classes without ordering. Example: Predicting food quality (Good, Great and Bad).
3. Ordinal Logistic Regression: The dependent variable has only two 3 or more possible outcomes/classes with ordering. Example: Star rating from 1 to 5

2.Random Forest

Random forest is different from the vanilla bagging in just one way. It uses a modified tree learning algorithm that inspects, at each split in the learning process, a random subset of the features. We do so to avoid the correlation between the trees. Suppose that we have a very strong predictor in the data set along with a number of other moderately strong predictors, then in the collection of bagged trees, most or all of our decision trees will use the very strong predictor for the first split! All bagged trees will look similar. Hence all the predictions from the bagged trees will be highly correlated.

Correlated predictors cannot help in improving the accuracy of prediction. By taking a random subset of features, Random Forests systematically avoids correlation and improves model's performance.

The example below illustrates how Random Forest algorithm works. Let's look at a case when we are trying to solve a classification problem. As evident from the image above, our training data has four features- feature1, feature 2, feature 3 and feature 4. Now, each of our bootstrapped sample will be trained on a particular subset of features. For example, Decision Tree 1 will be trained on features 1 and 4. DT2 will be trained on features 2 and 4, and finally DT3 will be trained on features 3 and 4. We will therefore have 3 different models, each trained on a different subset of features. We will finally feed in our new test data into each of these models and get a unique prediction.

The prediction that gets the maximum number of votes will be the ultimate decision of the random forest algorithm. For example, DT1 and DT3 predicted a positive class for a particular instance of our test data, while DT2 predicted a negative class. Since, the positive class got the majority number of votes (2), our random forest will ultimately classify this instance as positive. Again, I would like to stress on how the Random Forest algorithm uses a random subset of features to train several models, each model seeing only specific subset of the dataset. Random forest is one of the most widely used ensemble learning algorithms.

Why is it so effective? The reason is that by using multiple samples of the original dataset, we reduce the variance of the final model. Remember that the low variance means low overfitting. Overfitting happens when our model tries to explain small variations in the dataset because our dataset is just a small sample of the population of all possible examples of the phenomenon we try to model. If we were unlucky with how our training set was sampled, then it could contain some undesirable (but unavoidable) artifacts: noise, outliers and over- or underrepresented examples. By creating multiple random samples with replacement of our training set, we reduce the effect of these artifacts.

3. Decision Tree Classifier

Decision Tree Classifier is a simple Machine Learning model that is used in classification problems. It is one of the simplest Machine Learning models used in classifications, yet done properly and with good training data, it can be incredibly effective in solving some tasks, sometimes the simplest models are the best of certain tasks. So, in this article, we are going to look at the logic and the maths behind the Decision Trees Classifiers and analyze it by looking at a simple dataset. Make sure to visit this blog if you want to read more stories of this kind.

In a previous article, we defined what we mean by classification tasks in Machine Learning. If you've already seen that or you're familiar with classification tasks, let's see again our simple dataset that we can use better understand decision trees.

Decision Trees Classifiers are a type of Supervised Machine Learning meaning we build a model, we feed training data matched with correct outputs and then we let the model learn from these patterns. Then we give our model new data that it hasn't seen before so that we can see how it performs. And because we need to see what exactly is to be trained for a Decision Tree, let's see what exactly a decision tree is.

A decision tree consists of 3 types of components:

- Nodes — Decision over a value of a certain attribute ("is age over 50?", "is salary higher than $2000?")
- Edges — An edge is one of the answers from a node ("yes", "no") and build the connection to the next nodes.

Leaf nodes — Exit points for the outcome of the decision tree — for example, in our case, we can have multiple "Yes" and "No" leaf nodes meaning there are multiple ways we can exit the decision trees with the information that there will be or there will not be a traffic jam.

4.Linear Regression

This article attempts to be the reference you need when it comes to understanding the Linear Regression algorithm using Gradient Descent. Although this algorithm is simple, only a few truly understand it’s mathematics and underlying principles.

Linear means in a particular line and Regression means a measure of the relationship hence Linear Regression is a linear relationship of the data (independent variable) with the output (target variable).

Types of Linear Regression

- Simple Linear Regression
- Multiple Linear Regression

Simple Linear Regression is used for finding the relationship between two continuous variables i.e. finding the relationship between the independent variable (predictor) and the dependent variable (response). The crux of the Simple Linear Regression algorithm is to obtain a line that best fits the data. This is done by minimizing the loss function. What is the loss function? will be discussed later in this blog. Figure 2 shows the equation of the regression line. Where c is the y-intercept, m is the slope of the line with respect to independent feature x and y is the predicted value (also denoted as ŷ).

$$y_{predicted} \; or \; \hat{y}_i = mx + c$$

What does ‘m’ denote?

- If m > 0, then X (predictor) and Y (target) have a positive relationship. This means the value of Y will increase with an increase in the value of X.
- If m < 0, then X (predictor) and Y (target) have a negative relationship. This means the value of Y will decrease with an increase in the value of X.

What does ‘c’ denote?

- It is the value of Y when X=0. Suppose if we plot a graph in which the X-axis consists of Years of Experience (independent feature) and Y-axis consists of Salary (dependent feature). For Years of Experience = 0 what will be the Salary, this is what is denoted by ‘c’.

Now that you have understood the theory about the regression line, let’s discuss how can we select the best-fit regression line for a particular model using loss functions.

The loss function is the function that computes the distance between the current output of the algorithm and the expected output. It’s a method to evaluate how your algorithm models the data. It can be categorized into two groups. One for classification (discrete values, 0,1,2…) and the other for regression (continuous values).

7. Classification

Within this section we will look at - classification, where the classifier must predict if the flight will be arrived late or on time.

1.1 What is Classification?

Classification is an instance of supervised learning. Within classification we aim to predict a class under which an object will fall into.

With respect to the problem statement at hand, ArrDel is a binary categorical variable that holds a value of 0 for flights that arrived on time and a value of 1 for flights that arrived late. The classifier will need to predict if the flight will fall into class 0 (On-time) or class 1 (Delayed).

1.2 Algorithms Used

The following algorithms have been used and evaluated.

1. Logistic Regression
2. Random Forest
3. Decision Trees

1.3 Splitting the Data into Train and Test Data

ArrDel (Which tells us if the flight is delayed or not) and ArrDelayMinutes (Which gives us the number of minutes by which the flight is delayed) were removed for our independent variable, because these two are considered ground truth features which we will not know beforehand. The data was split into test and train in a 75:25 ratio.

1.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known.

The following are some terminologies related to a confusion matrix.

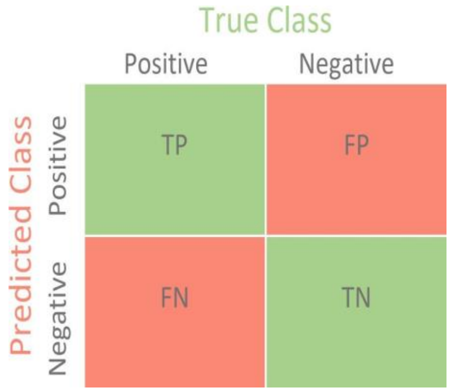
TP - True Positive, which means the number instances that were classified correctly. In the current use case, it refers to the number of flights that were classified correctly as Delayed.

FP - False Positive, which refers to the number instances that incorrectly in- dictates the occurrence of an instance. In the current use case, it refers to the number of flights that were classified as Delayed but were actually On-time.

TN - True Negative, which is the number instances that were classified correctly for the non-occurrence of an instance. In the current use case, it refers to the number of flights that were correctly classified as On-time.

FN - False Negative, which refers to the number instances that were classified incorrectly for the non-occurrence of an event. In the current use case, it refers to the number of flights that were classified as on-time for flights that were Delayed.

A confusion matrix is drawn with each row of the matrix represents the in- stances in a predicted class while each column represents the instances in an actual class (or vice versa). A representation of one can be seen in Figure 1.



8. Metrics

Precision Precision quantifies the number of positive class predictions that ac- tually belong to the positive class. Therefore, it tells us how many of the classified items are relevant.

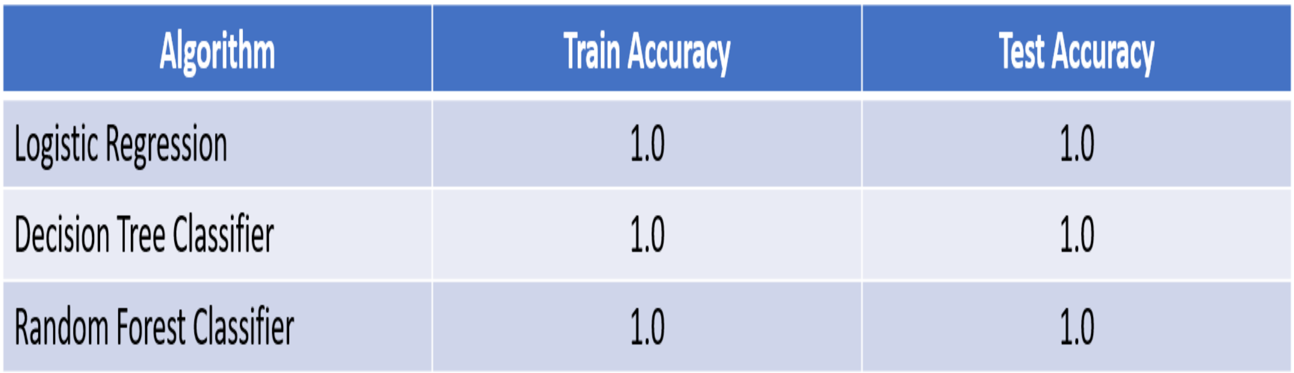
With respect to our problem at hand it gives us the proportion of the flights which have been classified correctly, either as delayed or not delayed, with re- spect to the total number of classified flights.

Recall Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.

With respect to our problem at hand it gives us the proportion of flights it has classified as delayed with respect to the total number of delayed Flights.

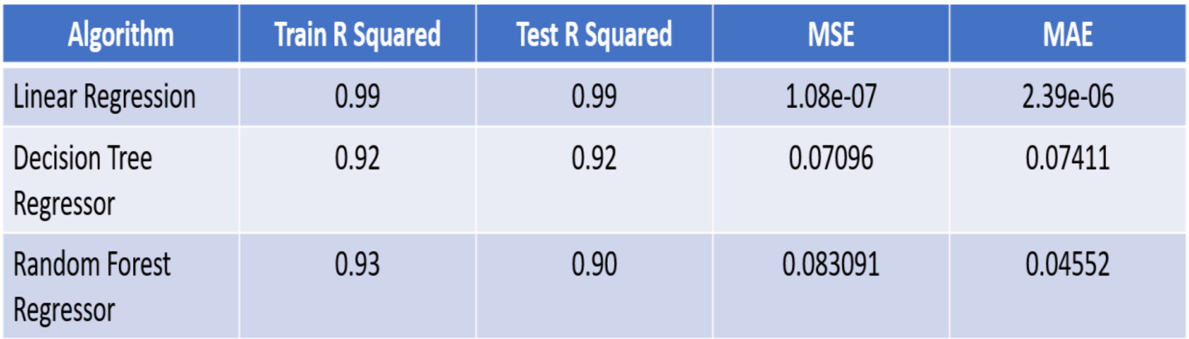
F1 Score or F- Measure F1 Score or F-Measure provides a single score that balances both the concerns of precision and recall in one number. It is evaluated as the harmonic mean of Precision and Recall.

Results for Classification



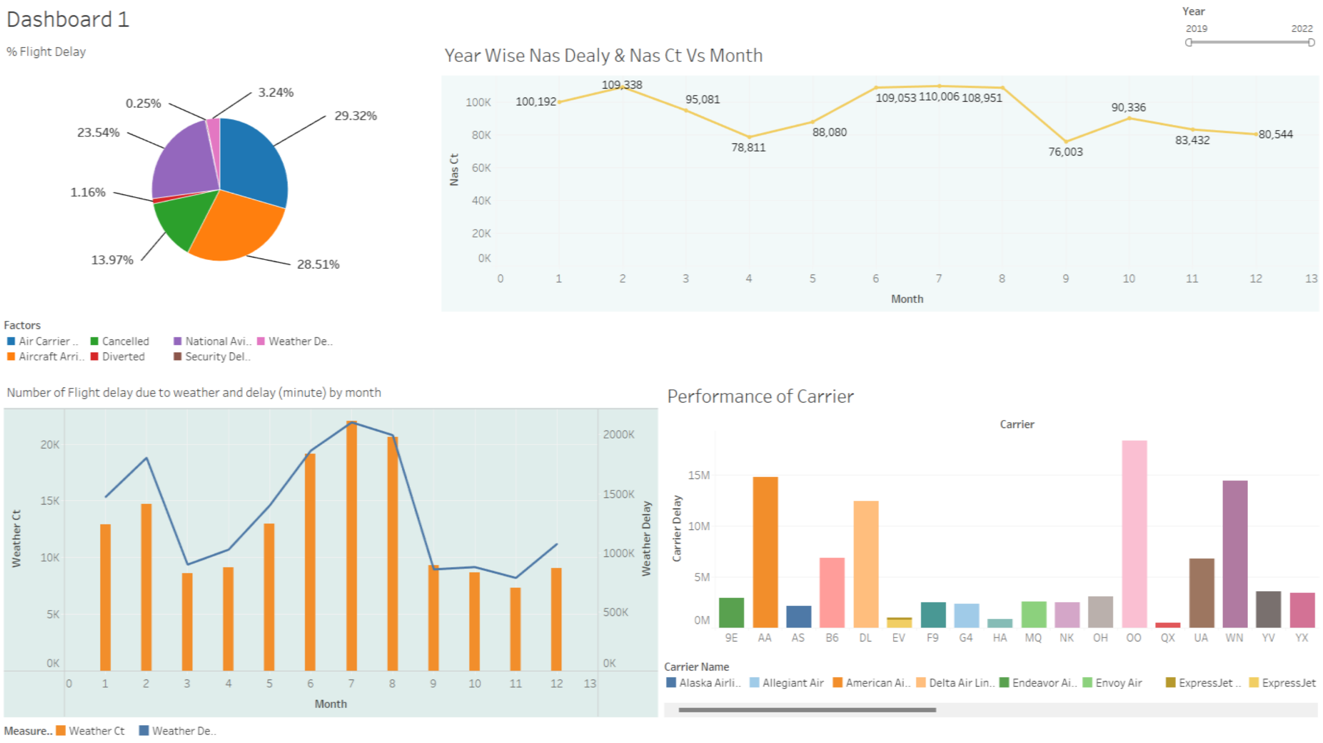
Algorithm	Train Accuracy	Test Accuracy
Logistic Regression	1.0	1.0
Decision Tree Classifier	1.0	1.0
Random Forest Classifier	1.0	1.0

Results for Regressor



Algorithm	Train R Squared	Test R Squared	MSE	MAE
Linear Regression	0.99	0.99	1.08e-07	2.39e-06
Decision Tree Regressor	0.92	0.92	0.07096	0.07411
Random Forest Regressor	0.93	0.90	0.083091	0.04552

9. Analysis using Tableau



- Among all the factors affecting flight delay, carrier delay, delay due to National Aviation System, Aircraft Arriving late are the major factors.
- In month of February maximum number of flights were delayed due to heavy air traffic which is controlled by NAS.
- The numbers of flights delayed and the total time of delay due to Weather is maximum from June to august.
- SkyWest Airline has maximum numbers of flights delay due to Air Carrier (performance of crew members).

10. Inference

The regression model returns a higher Mean Absolute Error and a higher Root Mean Squared Error with respect to the regression. This can be since the classifier may have incorrectly classified some of the flights as delayed or not delayed.

11. Conclusion

The data for Flight attributes for the selected airports that contains the features in interest, for further analysis. Using the XG Boost classifier the flights were classified as arrived late or on time.

Overall, the flight delay prediction and analysis project has provided valuable insights into the potential of data analytics and machine learning in the aviation industry. By developing accurate and efficient prediction models, this technology has the potential to transform the way airlines and airport authorities manage air travel disruptions, thereby improving the overall customer experience.

Hence two stage, classification and regression, machine learning engine was designed and built to classify whether if a flight will arrive late or on time and predict the number of minutes by which a flight arrive late.

12. Future Scope

This project is based on data analysis of year 2019-21 A large dataset is available from 1987-2017 but handling a bigger dataset requires a great amount of pre-processing and cleaning of the data. Therefore, the future work of this project includes incorporating a larger dataset. There are many different ways to pre-process a larger dataset like running a Spark cluster over a server or using a cloud-based services like AWS and Azure to process the data. With the new advancement in the field of deep learning, we can use Neural Networks algorithm on the flight and weather data. Neural Network works on the pattern matching methodology. It is divided into three basic parts for data modelling that includes feed forward networks, feedback networks, and self-organization network. Feed-forward and feedback networks are generally used in the areas of prediction, pattern recognition, associative memory, and optimization calculation, whereas self-organization networks are generally used in cluster analysis. Neural Network offers distributed computer architecture with important learning abilities to represent nonlinear relationships. Also, the scope of this project is very much confined to flight and weather data of United States, but we can include more countries like China, India, and Russia. Expanding the scope of this project, we can also add the flight data from international flights and not just restrict our self to the domestic flights