



# **PROJECT REPORT**

## **ON**

### **FLIGHT DATA ANALYSIS AND FLIGHT DELAY PREDICTION**

**Submitted in partial fulfilment for the award of  
PGDiploma in Big Data Analytics from C-DAC ACTS  
(Pune)**

**Presented by:**

- 1. Krunal Padade-223334(PRN 220341225031)**
- 2. Sujith Nair-223358(PRN 220340125054)**

**Guided by:**

**Mr. Akshay Tilekar**

# CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

**This is to certify that**

1.Krunal Padade	-	223334
2. Sujith Nair	-	223358

have successfully completed their project on  
**FLIGHT DATA ANALYSIS AND FLIGHT  
DELAY PREDICTION**

under the guidance of **Mr. Akshay Tilekar**

**Project Guide**

**Project Supervisor**

**Mr. Prashant Karhale  
Centre Coordinator**

# ACKNOWLEDGEMENT

This project 'FLIGHT DATA ANALYSIS AND FLIGHT DELAY PREDICTION' was a great learning experience for us and we are submitting this work to Institute for Advanced Computing & Software Development (**CDAC IACSD**).

We all are very glad to mention the name of **Mr. Akshay Tilekar** for his valuable guidance to work on this project. The guidance and support from him helped us to overcome various obstacles and intricacies during the course of project work.

Our most heartfelt gratitude goes to **Mr. Prashant karale** (Center coordinator), (Program Head, IACSD) and **Mr. Shantanu Pathak** (Course Coordinator, PGDBDA) who gave us all the required support and coordinated with us to provide all the necessities we needed to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

**From**

<b>1.Krunal Padade</b>	<b>-</b>	<b>220341225031</b>
<b>2. Sujit Nair</b>	<b>-</b>	<b>220340125054</b>

# CONTENTS

1. Abstract
2. Introduction and Overview of Project 5 Objectives of the project:
3. Data Description and Technologies Implemented
4. Data Description
5. Information on the technologies being used:
6. Flow Diagram
7. Control Flow Diagram
8. Algorithm used
  - 1 K Nearest Neighbors :
  - 2 Logistic Regression
  - 3 Random Forest
  - 4 Decision Trees
  - 5 XGBoost
  - 6 Linear Regression
9. Classification
10. Metrics

Inference

Conclusion

Future scope

BIBLIOGRAPHY

References

# 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 is able to classify delayed flights and predict the arrival delay period after takeoff, using corresponding flight information along with the relevant weather forecast

# 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 also 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 weather 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 2016 - 2017. 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 Pre-Processing

#### 1. Flight On-time Performance Data

This data set contains the On-time performance for various flights over the years 2016 and 2017. The airports and flight attributes taken into consideration are given in Table 1 and Table 2 respectively,

**Table 1** Airports taken into consideration

ATL	CLT	DEN	DFW	EW
IAH	JFK	LAX	LAX	MCO
MIA	ORD	PHX	SEA	SFO

**Table 2** Flight attributes taken into consideration

FlightDate	Quarter	Year	Month	DayofMonth
DepTime	DepDelay15	CRSDepTime	DepDelayMinutes	OriginAirportID
DestAirportID	ArrTime	CRSArrTime		

#### 2. Weather Data

Weather data for the airports in interest was collected hourly. The weather features under consideration are shown in table 3

**Table 3** Weather Features taken into consideration

WindSpeedKmph	WindDirDegree	WeatherCode	precipMM	Visibility
Pressure	Cloudcover	DewPointF	WindGustKmph	tempF
WindChillF	Humidity	date	time	airport

From the data collected only the years 2016 and 2017 were taken into consideration because we are examining the performance of the flights for those years alone. Yes, due to the late arrival of the previous flight. It hurts airports, airlines, and affects a company's marketing strategies as companies rely on customer loyalty to support their frequent flying programs.

### 3.Merging the data

Finally, after the required data was collected the flight and weather data were merged on the - Airport the flight is departing from, the date the flight is de- parting and the time at which the flight is departing.



### 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

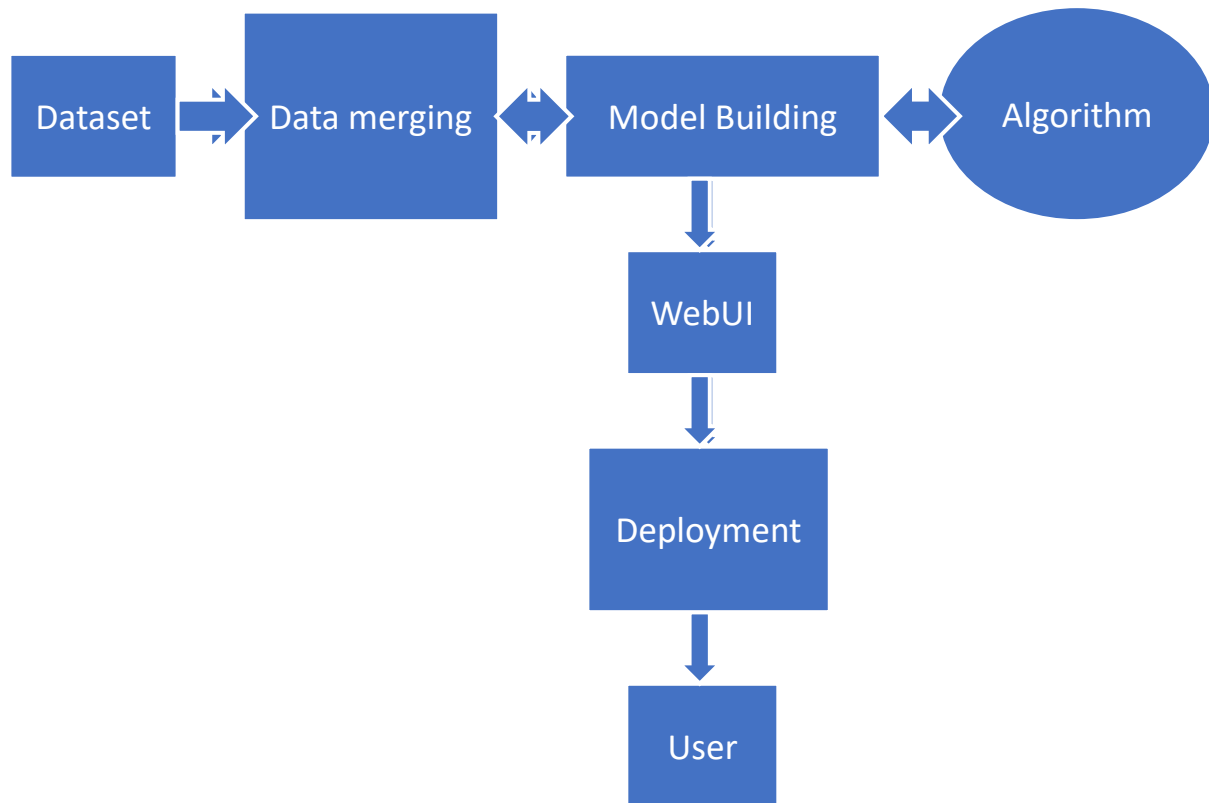
[Pandas](#) is a Python library for data analysis. Started by [Wes McKinney](#) in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries. It has an extremely active [community of contributors](#).

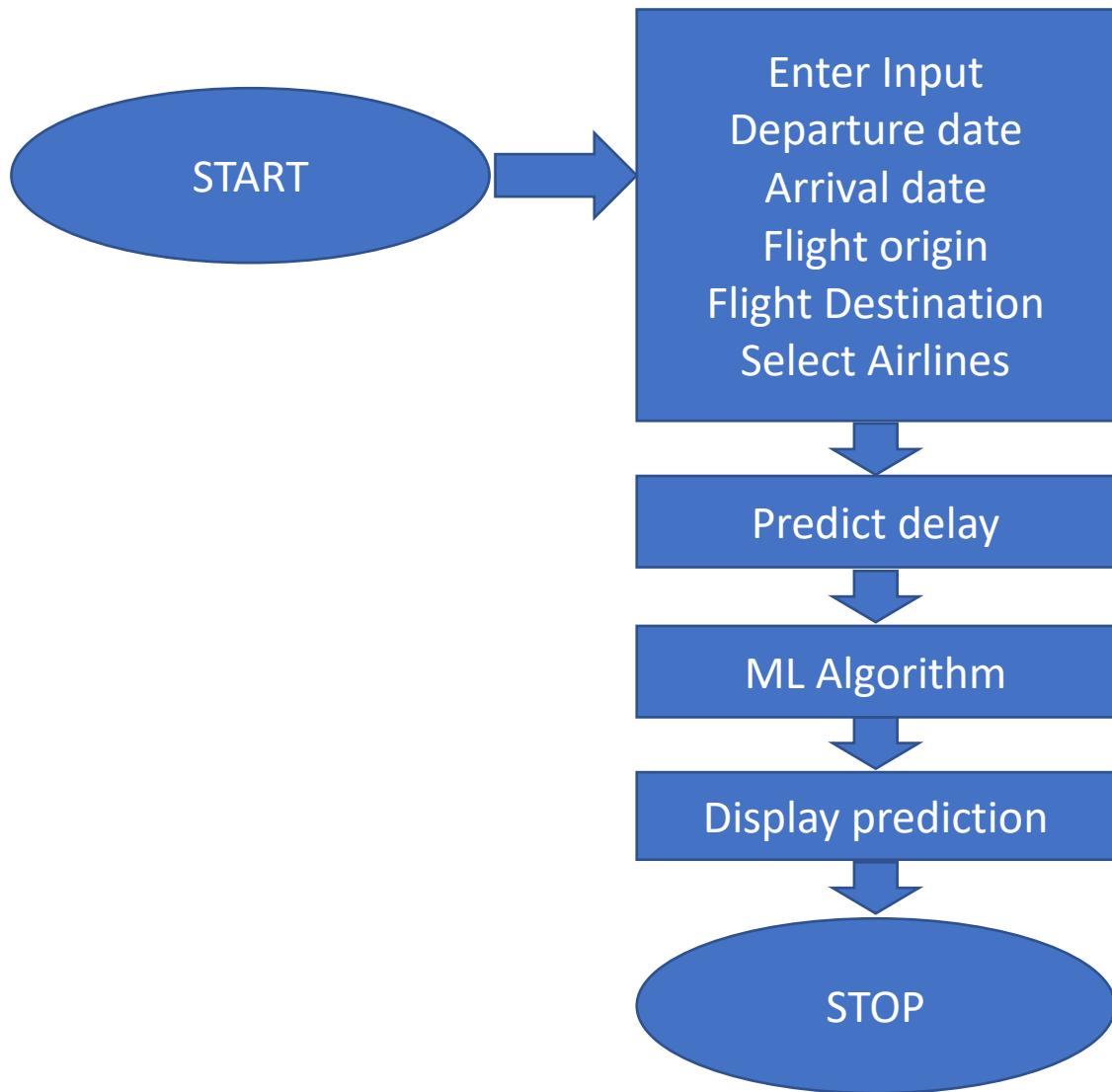
Pandas is built on top of two core Python libraries—[matplotlib](#) for data visualization and [NumPy](#) for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, pandas' `.plot()` combines 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.

# Flow Diagram





# Algorithm used

## Algorithms Used

The following algorithms have been used and evaluated.

1. K-Nearest Neighbors
2. Logistic Regression
3. Random Forest
4. Decision Trees
5. XGBoost

## 1. K-Nearest Neighbors :

Who is KNN? Is it some underground band that is bringing a [new sound](#) with the banjo that lives next door? Steve Martin, watch out now. What is it? Does it make a pie better than Patti LaBelle's [sweet potato pie](#)? Well, no. KNN is a classification algorithm. It uses the k value and distance metric([Euclidean distance](#)) to measure the distance of new points to nearest neighbors. The pros include that it is: simple to implement, training is easier and it has few parameters. The cons include: high prediction cost, doesn't play well with numerous features and doesn't like categorical features. Sorry qualitative features, only quantitative welcome here. The figure below displays a new point — identified by the star — that we want to classify. If we use a k value of 3, we would classify the new point with the gray elements. However, if we use a k value of 10, we would then classify the new point with the orange elements. The smaller the k value the greater the noise with the data; however, we can smooth this out by increasing the value of k.

What are some use Cases?

There are times where data may be given that is anonymized and the goal is to attempt to classify it, without really knowing the context of the data. Think group customer personalities to recommend certain products. Some of the 'use cases' of this are for the following:

- Recommender Systems
  - Theft prevention in the modern retail business
- Detect patterns in credit card usage and much more

## 2. Logistic Regression

What is 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

## 3. 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.

## 4. 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 take a 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.

### Introduction to Decision Trees Classifiers

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 actually 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.



## 5. XG Boost

How does the XGBoost algorithm work?

- Consider a function or estimate  $F$ . To start, we build a sequence derived from the function gradients. The equation below models a particular form of gradient descent.  $\frac{\partial F}{\partial x}$  represents the Loss function to minimize hence it gives the direction in which the function decreases.  $\epsilon_{x_t}$  is the rate of change fitted to the loss function, it's equivalent to the learning rate in gradient descent.  $F_{x_t+1}$  is expected to approximate the behaviour of the loss suitably.

$$F_{x_t+1} = F_{x_t} + \epsilon_{x_t} \frac{\partial F}{\partial x}(x_t)$$

- To iterate over the model and find the optimal definition we need to express the whole formula as a sequence and find an effective function that will converge to the minimum of the function. This function will serve as an error measure to help us decrease the loss and keep the performance over time. The sequence converges to the minimum of the function  $F$ . This particular notation defines the error function that applies when evaluating a gradient boosting regressor.

$$f(x, \theta) = \sum l(F((X_i, \theta), y_i))$$

Other Gradient Boosting methods

Gradient Boosting Machine (GBM)

GBM combines predictions from multiple decision trees, and all the weak learners are decision trees. The key idea with this algorithm is that every node of those trees takes a different subset of features to select the best split. As it's a Boosting algorithm, each new tree learns from the errors made in the previous ones.

*Useful reference* -> [Understanding Gradient Boosting Machines](#)

Light Gradient Boosting Machine (LightGBM)

LightGBM can handle huge amounts of data. It's one of the fastest algorithms for both training and prediction. It generalizes well, meaning that it can be used to solve similar problems. It scales well to large numbers of cores and has an open-source code so you can use it in your projects for free.

*Useful reference* -> [Understanding LightGBM Parameters \(and How to Tune Them\)](#)

Categorical Boosting (CatBoost)

This particular set of Gradient Boosting variants has specific abilities to handle categorical variables and data in general. The CatBoost object can handle categorical variables or numeric variables, as well as datasets with mixed types. That's not all. It can also use unlabelled examples and explore the effect of kernel size on speed during training.

## 6. Linear Regression

### Introduction

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 its mathematics and underlying principles.

### What is meant by Linear Regression?

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  $\hat{y}$ ).

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

### 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).

## Classification

Within this section we will look at - **classification**, where the classifier must predict if the flight will arrive 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, **ArrDel15** 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. Extra Trees
4. Decision Trees
5. XGBoost

### 1.3 Splitting the Data into Train and Test Data

**ArrDel15** (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 70:30 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 terminology 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 indicate 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 instances 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.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

## Metrics

**Precision** Precision quantifies the number of positive class predictions that actually 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 respect to the total number of classified flights.

**Recall** Recall quantifies the number of positive class predictions made out of all positive examples in the data-set.

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

## Class 0 (Arrived on Time) Unsampled Data

MODEL	Precision	Recall	f1 Score
KNeighborsClassifier	0.90	0.98	0.94
Logistic Regression	0.92	0.98	0.95
Random Forest	0.92	0.98	0.95
Decision Trees	0.92	0.91	0.92
XGBoost	0.92	0.98	0.95

## Class 1 (Arrived Late) Unsampled Data

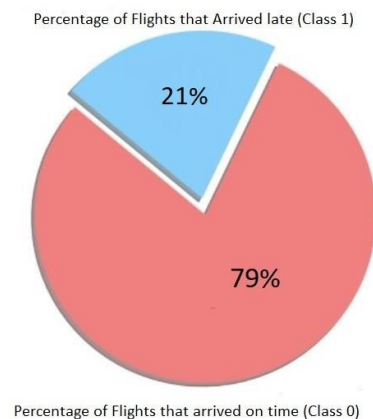
MODEL	Precision	Recall	f1 Score
KNeighborsClassifier	0.88	0.59	0.70
Logistic Regression	0.89	0.68	0.77
Random Forest	0.89	0.70	0.78
Decision Trees	0.68	0.70	0.69
XGBoost	0.90	0.70	0.78

# Results for Regressor (ArrDelayMinutes)

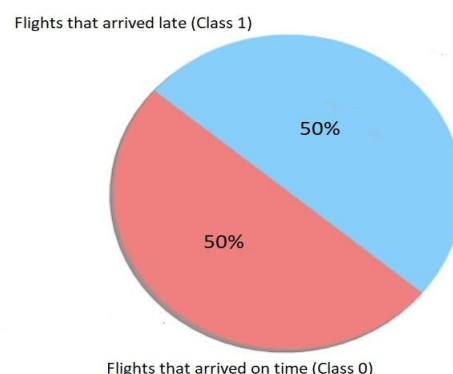
MODEL	MAE	MSE	RMSE	R-Squared
Linear Regression	12.28	315.77	17.77	0.938
Xgboost Regressor	11.26	266.93	16.33	0.947
Random forest Regressor	11.85	287.01	16.94	0.944
Decision Trees	16.65	591.05	24.31	0.884

# Class imbalance and Sampling methods

The performance of the classifier on Class 1 is weaker than the performance of Class 0 and one of the major reasons for this could be attributed to the class imbalance between the two of them.



## Flight Delay Prediction



To tackle the problem of Class imbalance there are standard sampling methods. Some of which used here are

### **SMOTE**

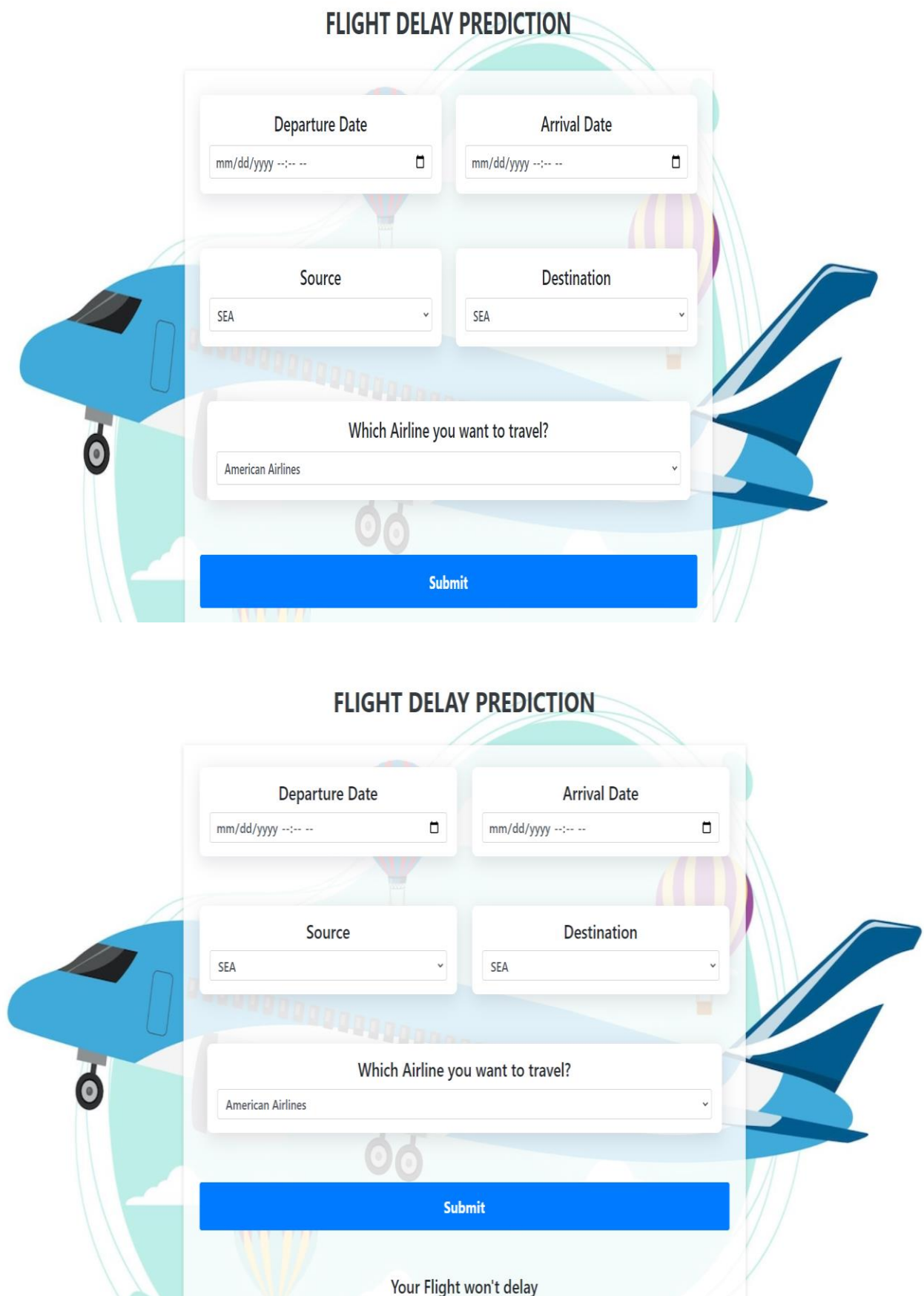
SMOTE (Synthetic Minority Oversampling Technique) is an oversampling technique that generates synthetic samples from the minority class. Rather than replicating the minority observations, SMOTE works by creating synthetic observations based upon the existing minority observations .

### **Random Undersampling (RUS)**

Random under-sampling involves randomly selecting examples from the majority class and deleting them from the training data-set.

# Demo

## FLIGHT DELAY PREDICTION



The form is titled "FLIGHT DELAY PREDICTION" and is set against a background illustration of a blue airplane flying over a globe with hot air balloons. The form contains several input fields and a submit button.

Departure Date

Arrival Date

Source

Destination

Which Airline you want to travel?

**Submit**

**FLIGHT DELAY PREDICTION**

Departure Date

Arrival Date

Source

Destination

Which Airline you want to travel?

**Submit**

Your Flight won't delay



## Inference

The regression model returns a higher Mean Absolute Error and also a higher Root Mean Squared Error with respect to the regression results from Section 4, Table 7. This can be due to the fact that the classifier may have incorrectly classified some of the flights as delayed or not delayed.

## Conclusion

The data for Flight attributes for the selected airports and also the weather features pertaining to these airports was collected. Both these datasets were processed, then merged to a single data-set that contains the features in interest, for further analysis. Using the XG Boost classifier the flights were classified as arrived late or on time. The XG Boost Classifier has performed with a Precision of 0.90, Recall of 0.73 and an f1-Score of 0.79 for the flights belonging to class 1 (flights that Arrived late).

A pipeline was performed with the best performing classifier, XGBoost, classifying if the flights would arrive late or on time. For those flights that were predicted to have arrived late, the XG Boost regressor, the comparatively best performing regressor, predicted the number of minutes by which the flight will arrive late with an MAE of 12.99, RMSE of 17.75 and R2 of 0.95. Hence a 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 arrives late

## FUTURE WORK

This project is based on data analysis of year 2016-17. A large dataset is available from 1987-2017 but handling a bigger dataset requires a great amount of preprocessing and cleaning of the data. Therefore, the future work of this project includes incorporating a larger dataset. There are many different ways to preprocess 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.

# BIBLIOGRAPHY

- [1] A. B. Guy, "Flight delays cost \$32.9 billion, passengers foot half the bill". [Online] Available : [https://news.berkeley.edu/2010/10/18/flight\\_delays/3/](https://news.berkeley.edu/2010/10/18/flight_delays/3/). [Accessed on June 2017].
- [2] M. Abdel-Aty, C. Lee, Y. Bai, X. Li and M. Michalak, "Detecting periodic patterns of arrival delay", Journal of Air Transport Management,, Volume 13(6), pp. 355– 361, November, 2007.
- [3] S. AhmadBeygi, A. Cohn and M. Lapp, "Decreasing Airline Delay Propagation By Re-Allocating Scheduled Slack", Annual Conference, Boston, 2008.
- [4] A. A. Simmons, "Flight Delay Forecast due to Weather Using Data Mining", M.S. Disseration, University of the Basque Country, Department of Computer Science, 2015.
- [5] L. Schaefer and D. Millner, "Flight Delay Propagation Analysis With The Detailed Policy Assessment Tool", Man and Cybernetics Conference, Tucson, AZ, 2001.
- [6] B. Bailey, "Data Cleaning 101". [Online]. Available: <https://towardsdatascience.com/data-cleaning-101-948d22a92e4>. [Accessed on March 2018].
- [7] Bureau of Transportation Statistics. [Online]. Available: <https://www.transtats.bts.gov/carriers.asp>. [Accessed on 2 April 2017].
- [8] How to Predict Yes/No Outcomes Using Logistic Regression. [Online]. Available: <https://blog.cleaarbrain.com/posts/how-to-predict-yesno-outcomes-using-logisticregression> [Accessed on 3 February 2018].
- [9] S. Polamuri, "How The Random Forest Algorithm Works In Machine Learning". [Online]. Available: <https://medium.com/@Synced/how-random-forest-algorithmworks-in-machine-learning-3c0fe15b6674>. [Accessed January 2018].
- [10] S. Ray, "Understanding Support Vector Machine algorithm". [Online]. Available: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vectormachine-example-code/>. [Accessed November 2017]. 56
- [11] OneHotEncoder. [Online]. Available: <http://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>. [Accessed on March 2018].

## References:

<https://github.com/Akashamba/Flight-Delay-Prediction>