FLIGHT DELAY PREDICTION

Project

submitted in partial fulfillment of the

requirements towards the degree of

BACHELOR OF TECHNOLOGY (B. TECH)

Submitted by:

Nakshatra Garg	**180060103045**
Shiv Tyagi	**180060103070**
Ananya Tyagi	**180060103013**

Under the supervision of

Mr. Kshitij Jain
(Asst. Professor, IT Department)

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER SCIENCE ENGINEERING
COLLEGE OF ENGINEERING ROORKEE, ROORKEE-247667 (UTTARAKHAND) INDIA

MAY, 2022

ABSTRACT

Anyone who has ever booked a flight ticket knows how unexpectedly the flight timing varies. Airlines use using sophisticated tactics which they call "yield management".

According to a recent study, India is currently the 3rd largest civil aviation market in the world. According to IATA (International Air Transport Association) the number of flyers globally could double to 8.2 billion in 2037. Nowadays, in this ever advancing world time management has become a prominent factor in all business operations. We can't afford to lose time on unnecessary delays. Using this as motivation we have developed a ML Model which predicts arrival flight delay based on various factors that include airport score, general airline trend, air traffic etc. The reasons for these delays vary a lot going from air congestion to weather conditions, mechanical problems, difficulties while boarding passengers, and simply the airlines inability to handle the demand given its capacity. By using Machine Learning (ML) Algorithms we can try to predict if your flight will be delayed in many ways. Of course, all of these different algorithms will have pitfalls and a certain degree of accuracy which will be associated to the data that they are fed.

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Mr. Kshitij Jain**, Department of Information Technology Engineering for his/her generous guidance, help and useful suggestions.

I express my sincere gratitude to **Mr. Sharad Kumar Singh, HoD** in Department of Information Technology Engineering for his/her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

I am extremely thankful to **College of Engineering Roorkee** for providing me infrastructural facilities to work in, without which this work would not have been possible.

Nakshatra Garg **180060103045**

Shiv Tyagi **180060103070**

Ananya Tyagi **180060103013**

CERTIFICATE OF APPROVAL

This is to certify the report entitled the "Flight Delay Prediction" is record of bonafide work, carried out by **Nakshatra Garg, Shiv Tyagi, Ananya Tyagi** under my supervision and guidance.

In my opinion, the report in its present form is in partial fulfillment of all the requirement for the award of **Bachelor of Technology (B.Tech), in Information Technology Engineering** at **College Of Engineering Roorkee**, is an authentic work carried by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted for any other degree or diploma.

Mr. Kshitij Jain (Supervisor)

Mr. Sharad Kumar Singh (HOD)

The B.Tech Viva –Voice Examination of **Nakshatra Garg, Shiv Tyagi, Ananya Tyagi** has been held on ..... and is accepted.

External Examiner

TABLE OF CONTENTS

LIST OF FIGURES

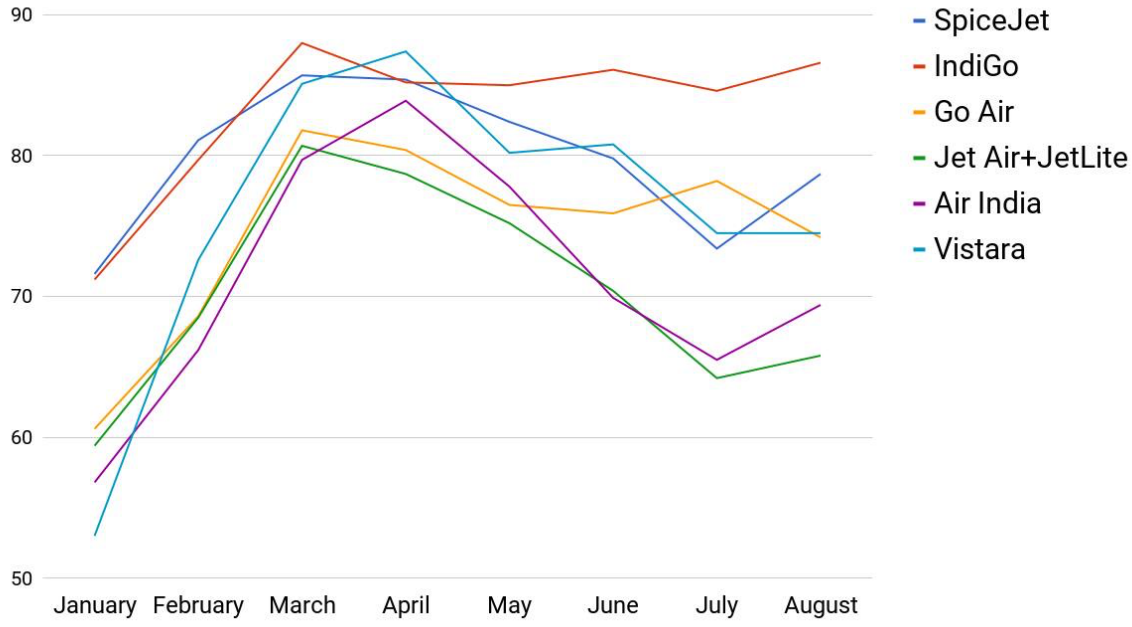
CHAPTER 1: INTRODUCTION

Flight delays has become a very important subject for air transportation all over the world because of the associated financial loses that the aviation industry is going through. According to a recent study, India is currently the 3rd largest civil aviation market in the world. According to IATA (International Air Transport Association) the number of flyers globally could double to 8.2 billion
in 2037. According to data from the Directorate General of Civil Aviation (DGCA) Air India Stated, over 21% of flights were cancelled during 2018, which impacted nearly 10,00,000 passengers.

1.1 Punctuality

About 24 percent of the flights run by Indian domestic carriers are never on time. Almost all the major airlines performed more or less the same. The best on-time performance has been of IndiGo with a rate of 83 percent. The worst among the major airlines is Jet Airways (data includes JetLite's performance) with only 70 percent of the flights being on time. Air India follows Jet Airways with an on-time rate of 71 percent.

Fig 1.1 – Rate of on-time performance of domestic carriers



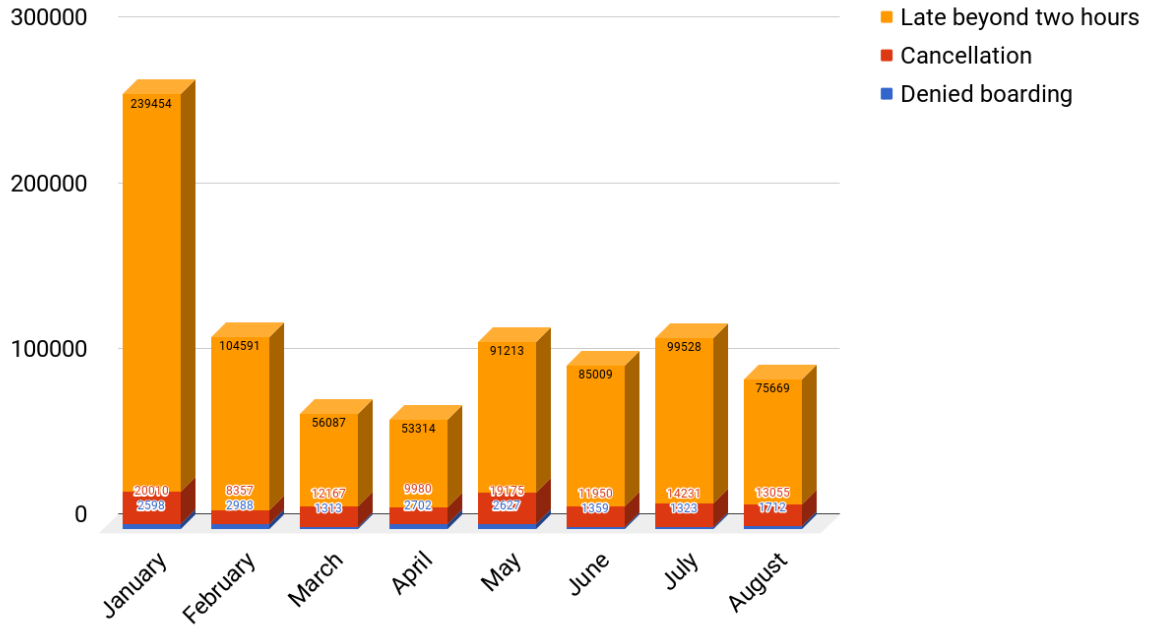
1.2 Effect on passengers

Passengers are not only marred by late flights but cancellation and boarding being denied due to various reasons also create problems for lakhs of passengers. In the first eight months, almost 10 lakh passengers were affected due to these reasons.

Over 16,600 passengers were denied boarding by airlines operating in the country. Among the major airlines, Jet Airways, with 14,104 passengers, denied the maximum number of passengers to board the aircraft. Whereas Air India and SpiceJet denied 2,054 and 293 passengers, respectively. IndiGo denied just three passengers as per submitted data.

Close to 1.1 lakh passengers were affected due to flights getting cancelled. Passengers of IndiGo, which has the largest market share among domestic carriers, were the most affected lot because of cancellations. Nearly 43,000 IndiGo passengers suffered, whereas, for SpiceJet and Air India, the numbers were 2,609 passengers and 21,538 passengers, respectively.

Fig 1.2 – Passengers affected due to flights being cancelled, late and denied boarding



1.3 Costly affair

The inefficiency in services not just affects passengers but also air carriers. They not only lose credibility but also must pay a hefty amount as compensation and towards providing facilities to affected passengers.

In first eight months, airlines have already spent Rs 34 crore towards compensation and facilities. Carriers paid Rs 22.7 crore to compensate the passengers which were denied boarding, translating into an average of Rs 13.66 thousand per passenger. Airlines also incurred expenses of Rs 4.15 crore towards facilities and compensation for passengers whose flights got cancelled and Rs 7.13 crore towards passengers whose flights got delayed by more than two hours.

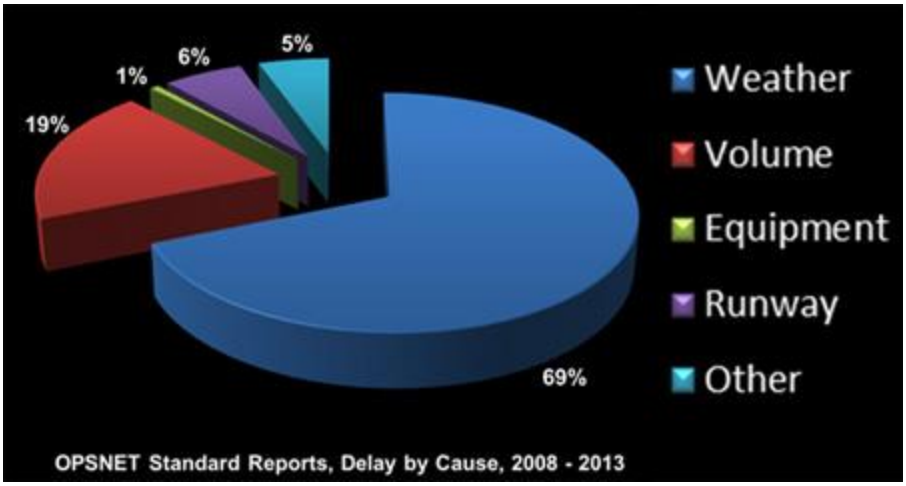
It is high time, the air carriers along with authorities running the industry in India start working towards providing better services or else passengers, as well as airlines, will keep bleeding money and, more importantly, time. [1]

1.4 Current Scenario

Nowadays, in this ever-advancing world time management has become a prominent factor in all business operations. We can't afford to lose time on unnecessary delays. Using this as motivation we have developed a ML Model which predicts arrival flight delay based on various factors that include airport score, general airline trend, air traffic etc.

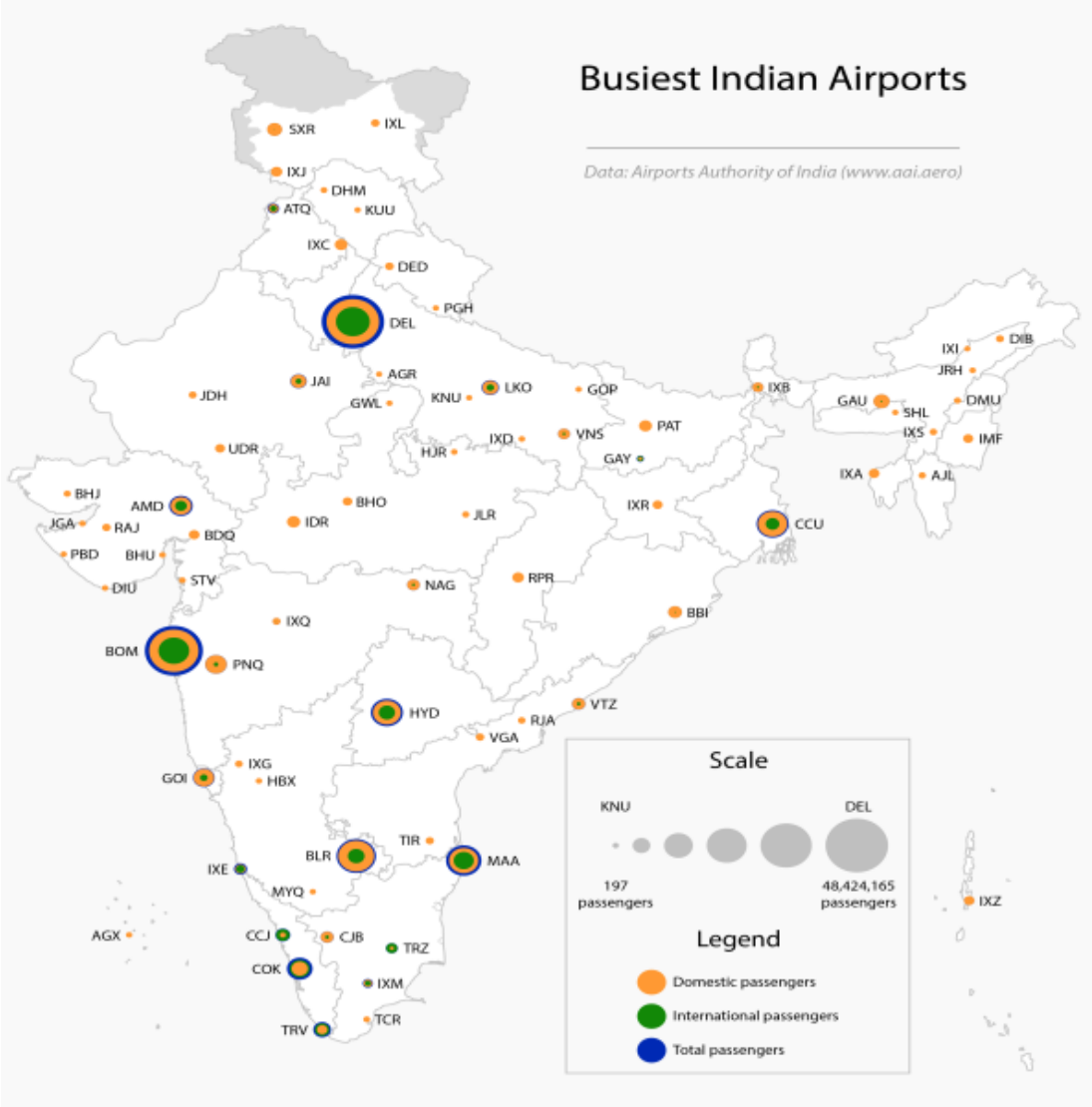
It is clearly visible from the pie chart that weather is one of the most prominent factors causing delay, hence we have extracted weather features separately.

Fig 1.3 – Flights Delay by Cause



The Busiest Airports are the ones where the probability of flight delay is high, hence we chose Bombay, Delhi, Bengaluru, Chennai, Hyderabad.

Fig 1.4 – Busiest Indian Airports



1.5 Related Work

The main concern of the researchers and analysts is to predict the reasons for flight delays and for that they have put in their efforts on collecting data about flight and the weather. Mohamed et al. [2] have studied the pattern of arrival delay for non-stop domestic flights at the Orlando International Airport. They focused primarily on the cyclic variations that happen in the air travel demand and the weather at that particular airport.

In Shervin et al.'s work [3], their motive of research is to propose an approach that improves the operational performance without hampering or effecting the planned cost.

Adrian et al. [4] have created a data mining model which enables the flight delays by observing the weather conditions. They have used WEKA and R to build their models by selecting different classifiers and choosing the one with the best results. They have used different machine learning techniques like Naïve Bayes and Linear Discriminant Analysis classifier.

Choi et al. [5] have focused on overcoming the effects of the data imbalancing caused during data training. They have used techniques like Decision Trees, AdaBoost, and K-Nearest Neighbors for predicting individual flight delays. A binary classification was performed by the model to predict the scheduled flight delay.

Schaefer et al. [6] have made Detailed Policy Assessment Tool (DPAT) that is used to stimulate the minor changes in the flight delay caused by the weather changes.

Bing Liu [7] has done a sentiment analysis and opinion mining that analyzes people's opinions, sentiments, and studies their behavior. The output of the research is a feature-based opinion summary which is also known as sentiment classification.

Using techniques such as Natural Language Processing, Naïve Bayes, and Support Vector Machine, researchers built algorithms for analysis that helped them in extracting features in the model. Most of them focused on predicting overall flight delays. Our research concentrated mainly on predicting flight delays for a particular airport over a specific period of time. First, we used a regression model to examine the significance of each feature and then, a feature selection approach to examine the impact of feature combination. These two techniques determined the features to retain in the model. Instead of using the whole set, we sampled 5,000 records at a time to run through different machine learning models. The machine learning models implemented here were Random Forest classifier and Support Vector Machine (SVM) classifier. Further, we applied an approach called One-Hot-Encoder to create a variant of the model for evaluating potential prediction performance.

CHAPTER 2: PROJECT DEVELOPMENT

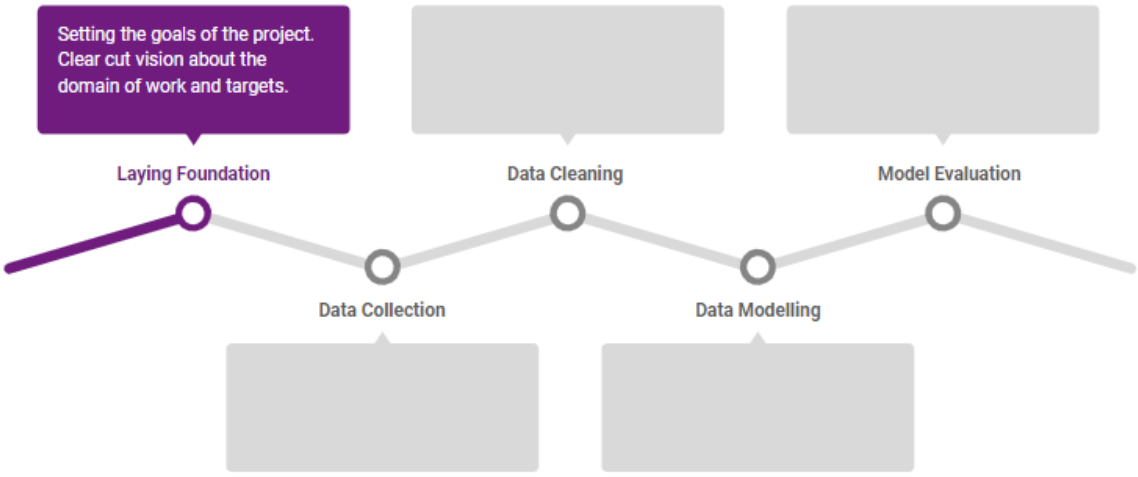
One important thing that we wanted to highlight to our audience is that though the fancy ML algorithms generally fill up most the web resources available on the Internet, however, almost 80% of the time goes in actually gaining basic understanding and cleaning the data. The predictive modelling constituted just about 15–20% of our overall time in the Project.

2.1 About the project

We selected six airlines and five airports and tried to study the factors affecting flight schedules on these specific routes only. The idea behind this was the fact that the five airports selected, namely Delhi, Mumbai, Bangalore, Hyderabad, and Kolkata together contributed more than fifty percent of the volume of domestic passengers. The locations of these airports also reflect the geographical diversity of the country. Hence study of flight schedules from these airports will in general reflect the characteristics of the population data.

The six airlines considered (Indigo, Air India, SpiceJet, GoAir, Air Asia and Vistara) together acquire 99.4% of the domestic market.

Fig 2.1 – Laying Foundation



2.2 Data collection

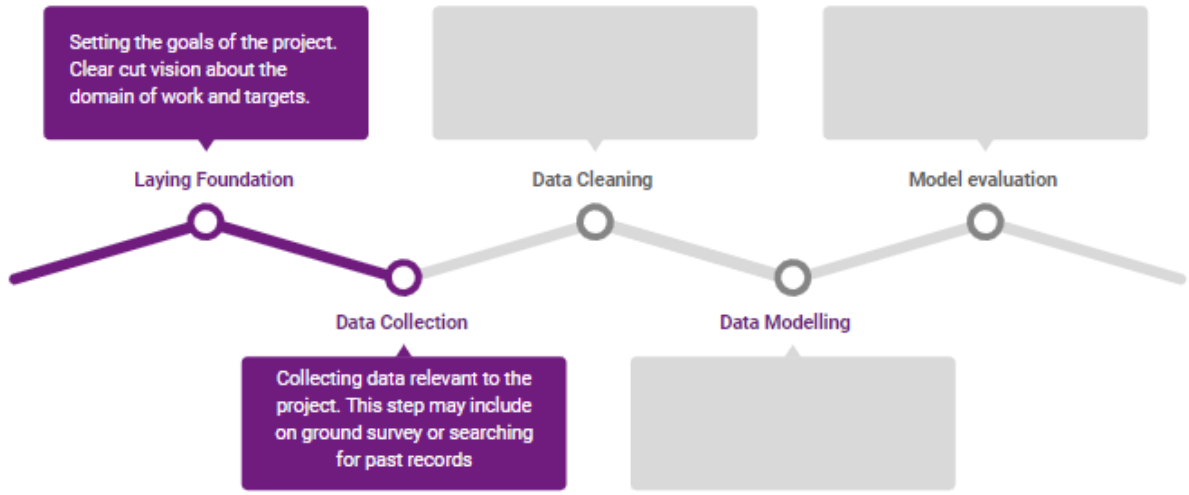
After laying down the foundations, we started looking for data relevant to our project. This data collection had two parts. The first part revolved around gathering historical data of flights like departure airport, arrival airport, expected time of departure and arrival and actual time of departure and arrival. While such datasets are available on various sites for money, being

students, it was not feasible for us to buy. We stuck to the organic method of "data scraping" using python package Beautiful Soup.
This was by far the most important and tedious part of the project. Important because this was the part which determined the number of data points in our data set and tedious because the airline websites blocked the IP address within an hour as we start to scrape out data.

The second part involved selecting features to include in the data set. While weather details were obvious parameters to include and easy to scrape from sites like accuweather.com, we had to decide on the non-weather features which could affect the arrival delays. Within a few days of continual efforts, we gathered about 15k data points with all required flight parameters from various websites. We included 45 features which we thought would contribute to delays. Excluding 22 weather features, others covered many aspects of an airport, airline carrier and the flight itself.

> "It's not who has the best algorithm that wins, it's who has the most data"
> - Andrew NG

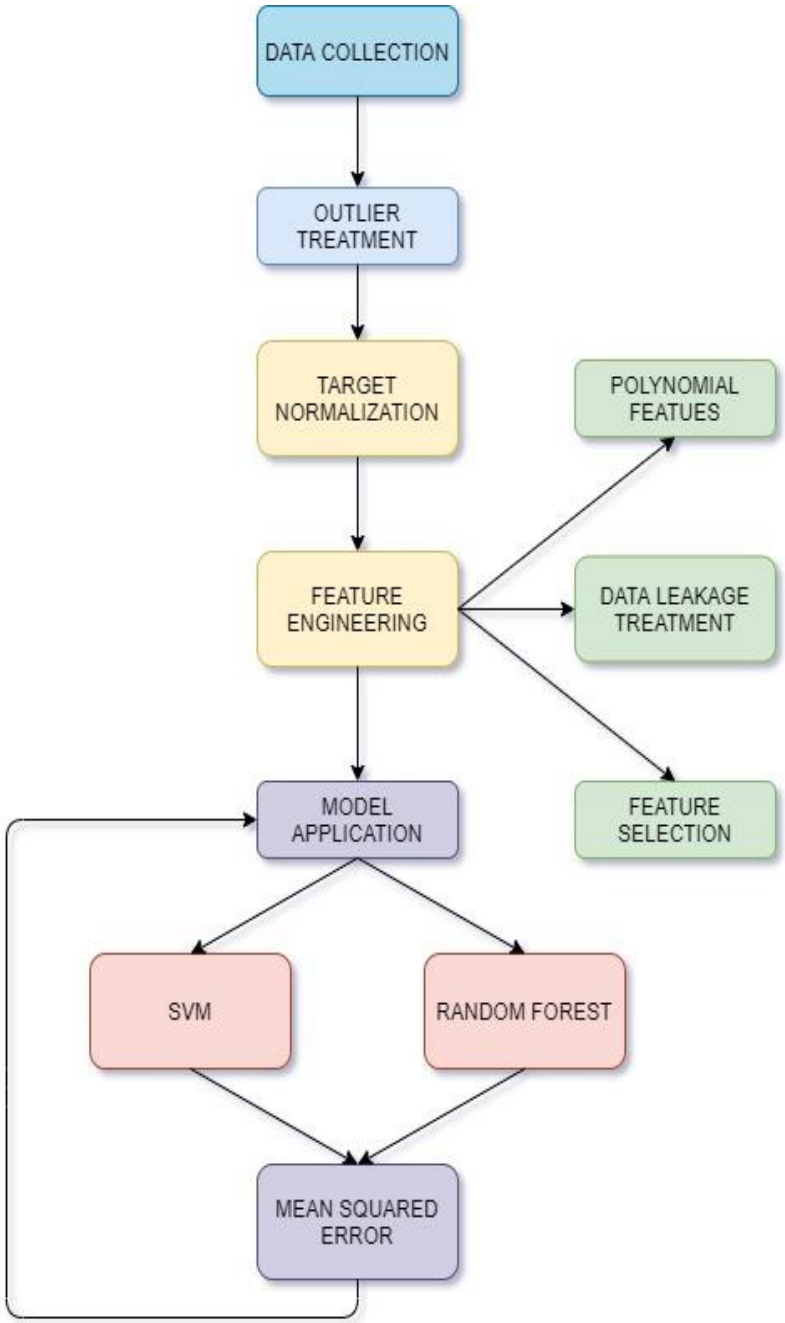
Fig 2.2 – Data Collection



2.3 Data cleaning

We then proceeded with cleaning the data set by imputing missing values, deleting rows with lots of missing values, removing the outliers, text features cleaning and other issues due to joining of multiple databases.

Fig 2.3 – Flow Chart



There were many categorical columns in the data set which were taken care of by using One Hot Encoding as most of them had less than ten unique values.

Fig 2.4 – Data Cleaning

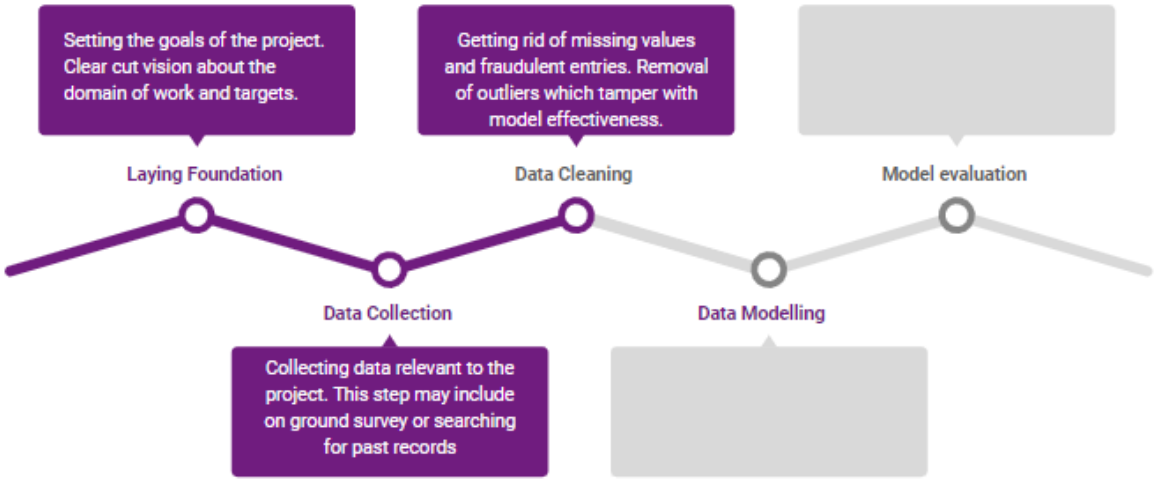
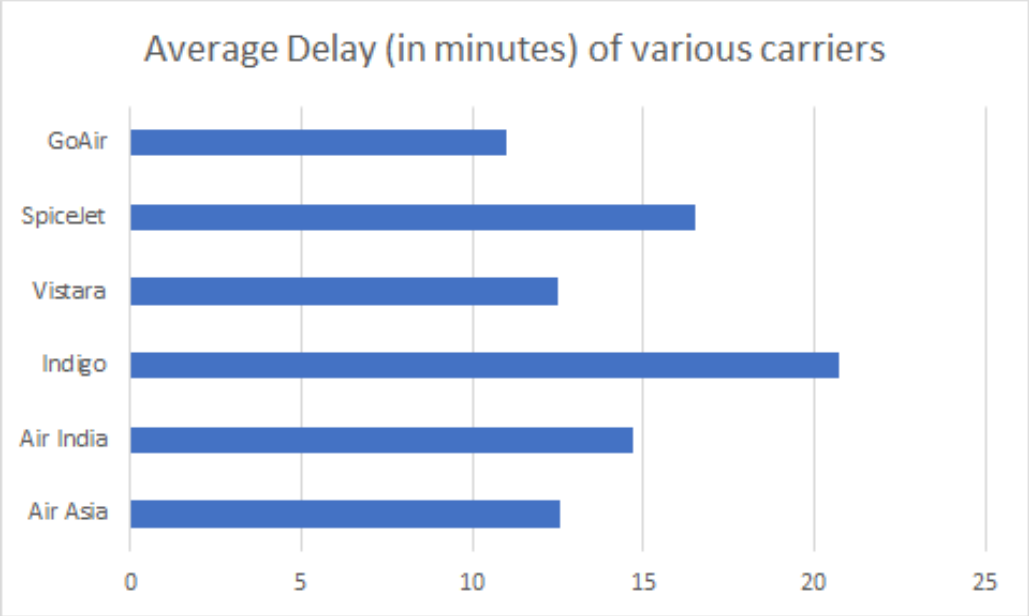


Fig 2.5 – Average Delay (in minutes) of various carriers



2.4 Data modelling

We started off with simple linear models and a lot to our astonishment achieved great R2 scores (Training R2 score=0.92, Validation R2 score=0.89). Such results on a crude real life data set were unreal. Later while going through the coefficients which the algorithm allotted to various features, we found that one of our features i.e., Departure delay (Note: Our target was to find arrival delays) was allotted very high importance. Its coefficient was so high that all other features got ignored.

This situation is called Data Leakage, where the model doesn't consider the effect of parameters other than one or two. As our prediction feature (i.e., arrival delay) is a actually also a cause of Departure delays, including such feature in the modelling would be inappropriate as we are giving higher weights to a feature that itself depends on the target feature.

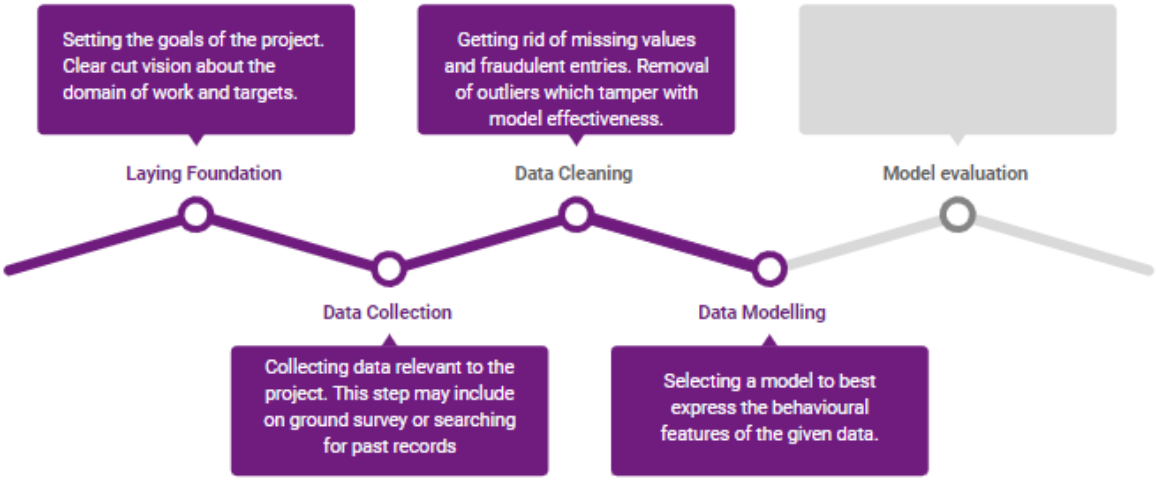
> This experience gathered here emphasized that data science is not only about how accurate the results are, but rather how models capture the effects produced by various factors in real life. In short, its more about the approach than the result.

Therefore, this feature had to be removed. As we removed the departure delay column, the linear as well as polynomial models failed miserably. The low R2 score was indicative of the fact that the model was unable of capture the variance in the targets as the features varied.

We moved on to ridge regression and then lasso regression as well, but results were unsatisfactory. We then deployed tree-based ensemble models and they performed way better than previous ones, but it still needed refinement.

We adopted various approaches to refine the model. We normalized the target variables so that the model generalizes well in the real world. There is never an end to the refinement process. We continued to refine our model with SVMs, feature selections and careful hyper-parameter tuning by randomized cross search validation and finally came up with an ensemble of various models.

Fig 2.6 – Data Modelling



2.5 Model evaluation

After realizing many drawbacks in R2 as an evaluation metric, we decided to change our metric to Mean Squared Error (MSE). At last, our best fit model produced a training set error of 0.13 while the test set error was 0.25 (the test data set was not seen by the model ever before the last step).

Fig 2.7 – Predicted vs Actual Values

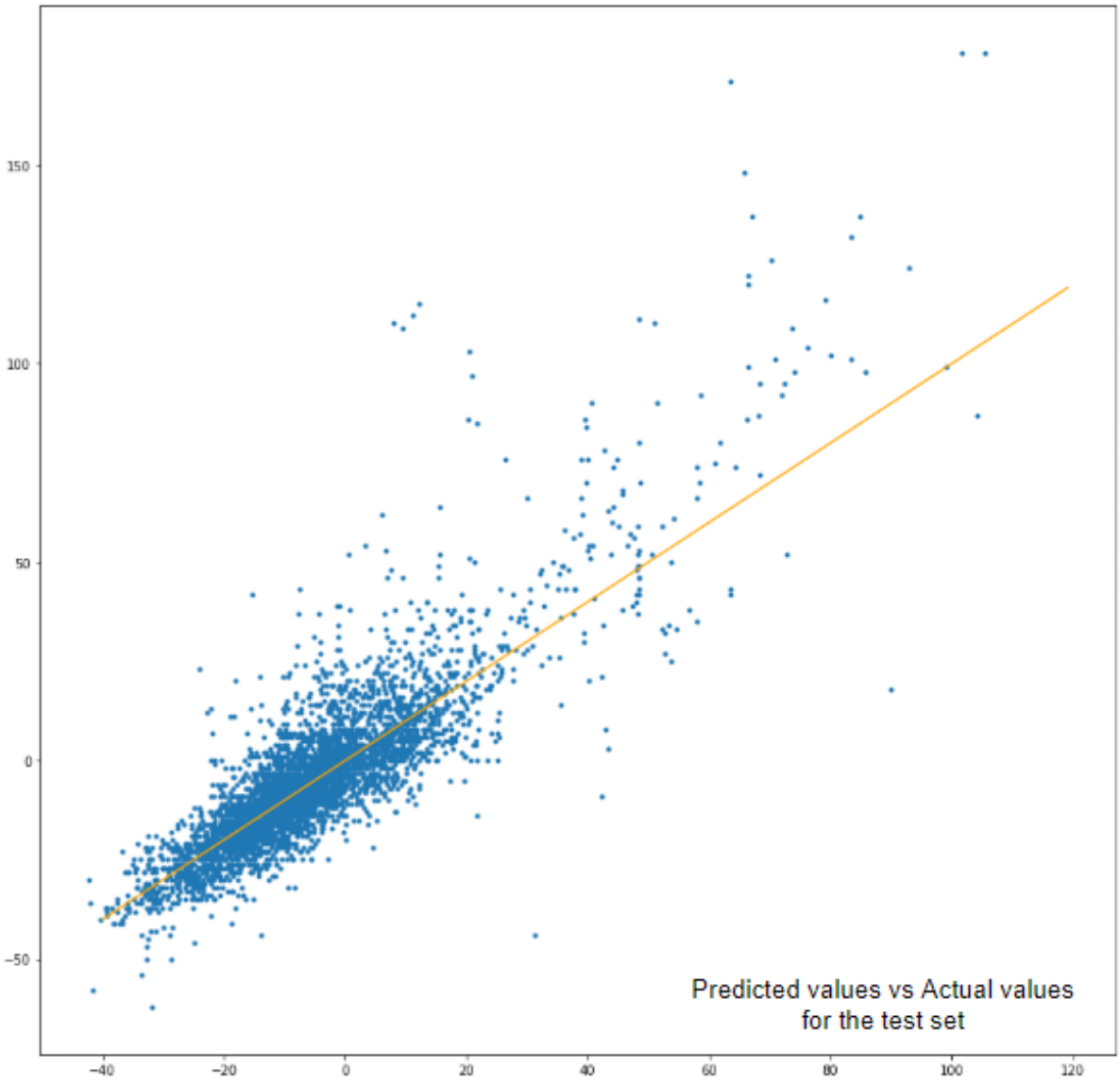
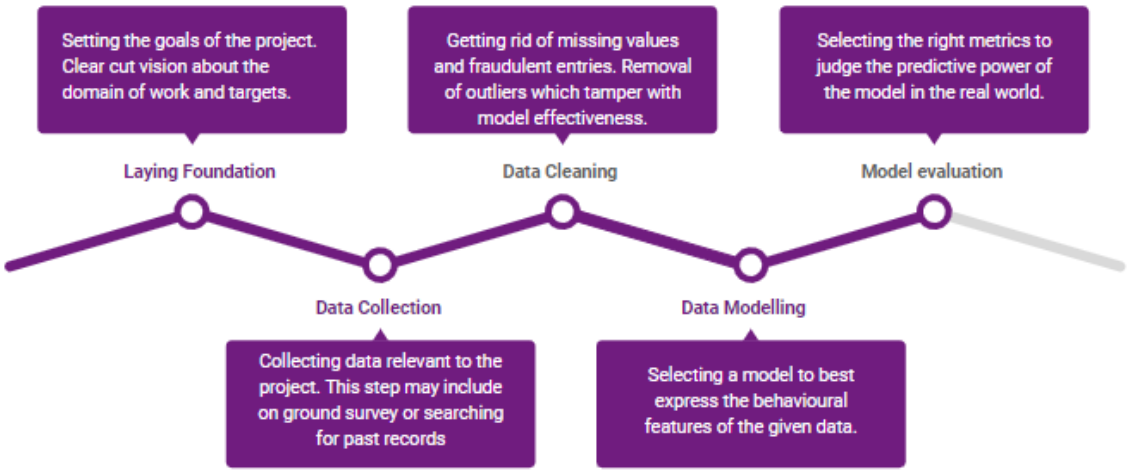


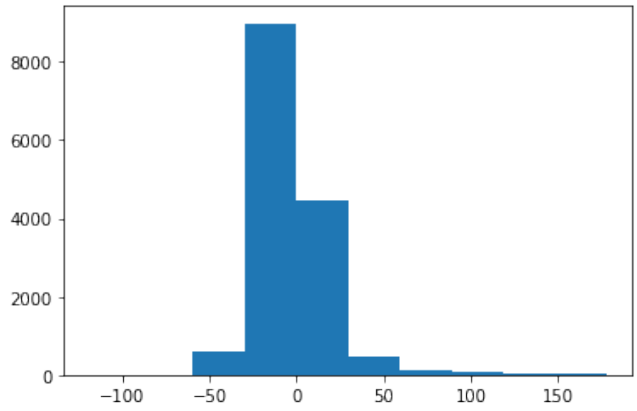
Fig 2.7 – Model Evaluation



CHAPTER 3: METHODOLOGY

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_csv('fit.csv')

# One Hot Encoding for cataegorical features like route
from sklearn.preprocessing import LabelEncoder

data = df['route']
values= np.array(data)

label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(values)

from sklearn.preprocessing import OneHotEncoder

onehot_encoder = OneHotEncoder(sparse=False)
integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
onehot_encoded = onehot_encoder.fit_transform(integer_encoded)

df1 = pd.DataFrame.from_records(onehot_encoded)
df = pd.concat((df,df1),axis=1)

# One Hot Encoding for cataegorical features like carrier
from sklearn.preprocessing import LabelEncoder

data1 = data['Carrier']
values1 = np.array(data1)

label_encoder1 = LabelEncoder()
integer_encoded1 = label_encoder.fit_transform(values1)

from sklearn.preprocessing import OneHotEncoder

onehot_encoder1 = OneHotEncoder(sparse=False)
integer_encoded1 = integer_encoded1.reshape(len(integer_encoded1), 1)
onehot_encoded1 = onehot_encoder1.fit_transform(integer_encoded1)

df2 = pd.DataFrame.from_records(onehot_encoded1)
df = pd.concat((df,df2),axis=1)
```

```
df.head

import pandas as pd
data = pd.read_csv("fit.csv")

# Feature Importance Score
from sklearn import preprocessing

X = data.drop('A_Delay', axis =1)
y = data['A_Delay'].to_numpy().reshape(-1,1)

lab_enc = preprocessing.LabelEncoder()
y_encoded = lab_enc.fit_transform(y)

from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt

model = ExtraTreesClassifier(n_estimators = 5,criterion ='entropy', max_features = 2)
model.fit(X,y_encoded)

print(model.feature_importances_ * 100)

# Removing Data Leakage
data = data.drop("D_Delay",axis = 1)

# Outlier Removal
data = data[data['A_Delay'] < 180]
data = data[data['A_Delay'] > -180]

# Looking at the Target
plt.hist(data['A_Delay'])
plt.show()
```




```
# Data Description post out-lier removal
pd.options.display.max_columns = None
data.describe()
```

```
# Looking at the columns..
data.columns
```

Index(['D_Airport Rating', 'D_Airport On Time Rating', 'D_Airport Service Rating', 'A_Airport Rating', 'A_Airport On Time Rating', 'A_Airport Service Rating', 'Duration', 'A_Delay', 'C_Rating', 'C_Market Share', 'C_Load Factor', 'C_On Time Performance Rating', 'D_DewPointC', 'D_WindGustKmph', 'D_cloudcover', 'D_humidity', 'D_precipMM', 'D_pressure', 'D_tempC', 'D_visibility', 'D_winddirDegree', 'D_windspeedKmph', 'D_Time', 'A_DewPointC', 'A_WindGustKmph', 'A_cloudcover', 'A_humidity', 'A_precipMM', 'A_pressure', 'A_tempC', 'A_visibility', 'A_winddirDegree', 'A_windspeedKmph', 'A_Time', 'BLR-BOM', 'BLR-DEL', 'BOM-DEL', 'CCU-DEL', 'DEL-HYD', 'Air Asia', 'Air India', 'Go Air', 'Indigo', 'Spicejet', 'Vistara'], dtype='object')

```
# AVERAGE DELAYS OF AIRLINES
print(abs(data[data['Air Asia'] == 1]['A_Delay']).sum()/1873)
print(abs(data[data['Air India'] == 1]['A_Delay']).sum()/3529)
print(abs(data[data['Indigo'] == 1]['A_Delay']).sum()/2296)
print(abs(data[data['Vistara'] == 1]['A_Delay']).sum()/2825)
print(abs(data[data['Spicejet'] == 1]['A_Delay']).sum()/1723)
print(abs(data[data['Go Air'] == 1]['A_Delay']).sum()/2487)
```

12.59586225306994
14.678379144233494
20.692073170731707
12.523893805309735
16.52930934416715
11.02854845195014

```
# WE MAKE 4 POSSIBLE SETS OF DATA
# 1 - NO CHANGE IN FEATURES, NO CHANGE IN TARGET
# 2 - POLYNOMIAL FEATURES OF DEGREE 2, NO CHANGE IN TARGET
# 3 - NO CHANGE IN FEATURES, TARGET NORMALIZED
# 4 - POLYNOMIAL FEATURES OF DEGREE 2, TARGET NORMALIZED

from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split as tts
from sklearn.preprocessing import PowerTransformer
from sklearn.metrics import mean_squared_error
PT = PowerTransformer(method = "yeo-johnson")
X_train,X_test,y_train,y_test = tts(data.drop('A_Delay', axis =
1),data['A_Delay'],test_size = 0.3,random_state = 1)

PT.fit(y_train.to_numpy().reshape(-1,1))
lamb = PT.lambdas_
```

```
yt_train = PT.transform(y_train.to_numpy().reshape(-1,1))
yt_test = PT.transform(y_test.to_numpy().reshape(-1,1))

from sklearn.preprocessing import PolynomialFeatures
PF = PolynomialFeatures(2);

data_p = PF.fit_transform(data.drop('A_Delay',axis = 1))

Xp_train,Xp_test,y_train,y_test = tts(data_p,data['A_Delay'],test_size =
0.3,random_state = 1)

# Printing all shapes to be sure!
print(X_train.shape) # NO CHANGE APPLIED
print(X_test.shape)  # NO CHANGE APPLIED
print(y_train.shape) # NO CHANGE APPLIED
print(y_test.shape)  # NO CHANGE APPLIED
print(Xp_train.shape)# POLYNOMIAL FEATURES
print(Xp_test.shape) # POLYNOMIAL FEATURES
print(yt_train.shape)# TARGET NORMALIZED
print(yt_test.shape) # TARGET NORMALIZED
```

(10313, 44)
(4420, 44)
(10313,)
(4420,)
(10313, 1035)
(4420, 1035)
(10313, 1)
(4420, 1)

```
# Applying random forest regression on the data.
# Evaluation Metric used is mse.

from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor

# NO CHANGE DATA
RFR = RandomForestRegressor(n_estimators = 100, min_samples_split = 10,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(X_train,y_train)
print(mean_squared_error(RFR.predict(X_train),y_train))
print(mean_squared_error(RFR.predict(X_test),y_test))

# TARGET NORMALIZED
```

```
RFR = RandomForestRegressor(n_estimators = 100, min_samples_split = 10,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(X_train,yt_train)
print(mean_squared_error(RFR.predict(X_train),yt_train))
print(mean_squared_error(RFR.predict(X_test),yt_test))

# POLYNOMIAL FEATURES AND NO NORMALIZATION OF TARGET
RFR = RandomForestRegressor(n_estimators = 100, min_samples_split = 10,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(Xp_train,y_train)
print(mean_squared_error(RFR.predict(Xp_train),y_train))
print(mean_squared_error(RFR.predict(Xp_test),y_test))

# BOTH POLYNOMIAL AND NORMALIZATION APPLIED
RFR = RandomForestRegressor(n_estimators = 100, min_samples_split = 10,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(Xp_train,yt_train)
print(mean_squared_error(RFR.predict(Xp_train),yt_train))
print(mean_squared_error(RFR.predict(Xp_test),yt_test))

# SELECTING BEST 60 FEATURES BY SelecKBest

from sklearn.feature_selection import SelectKBest ,mutual_info_regression
SKB = SelectKBest(mutual_info_regression,60)

SKB.fit(Xp_train,y_train)
Xp60_train = SKB.transform(Xp_train)
Xp60_test = SKB.transform(Xp_test)

SKB.fit(Xp_train,yt_train)
Xpt60_train = SKB.transform(Xp_train)
Xpt60_test = SKB.transform(Xp_test)

print(Xp60_train.shape)
print(Xp60_test.shape)
print(Xpt60_train.shape)
print(Xpt60_test.shape)
```

(10313, 60)
(4420, 60)
(10313, 60)
(4420, 60)

```
print(Xp60_train.shape)
print(Xp60_test.shape)
(Xp60_train == Xpt60_train)
```

(10313, 60)
(4420, 60)
array([[True, True, True, ..., True, True, True],
[True, True, True, ..., True, True, True],
[True, True, True, ..., True, True, True],
...,
[True, True, True, ..., True, True, True],
[True, True, True, ..., True, True, True],
[True, True, True, ..., True, True, True]])

```
# POLYNOMIAL REDUCED TO 60 FEATURES AND Y WITHOUT
NORMALIZATION
from sklearn.ensemble import RandomForestRegressor
RFR = RandomForestRegressor(n_estimators = 500, min_samples_split = 20,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(Xp60_train,y_train)
print(mean_squared_error(RFR.predict(Xp60_train),y_train))
print(mean_squared_error(RFR.predict(Xp60_test),y_test))

# POLYNOMIAL REDUCED TO 60 FEATURES AND Y WITH
NORMALIZATION
RFR = RandomForestRegressor(n_estimators = 500, min_samples_split = 20,
max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(Xpt60_train,yt_train)
print(mean_squared_error(RFR.predict(Xpt60_train),yt_train))
print(mean_squared_error(RFR.predict(Xpt60_test),yt_test))
```

60.217648962875295
105.9436047999162

0.13783012802053835
0.2532220962569266

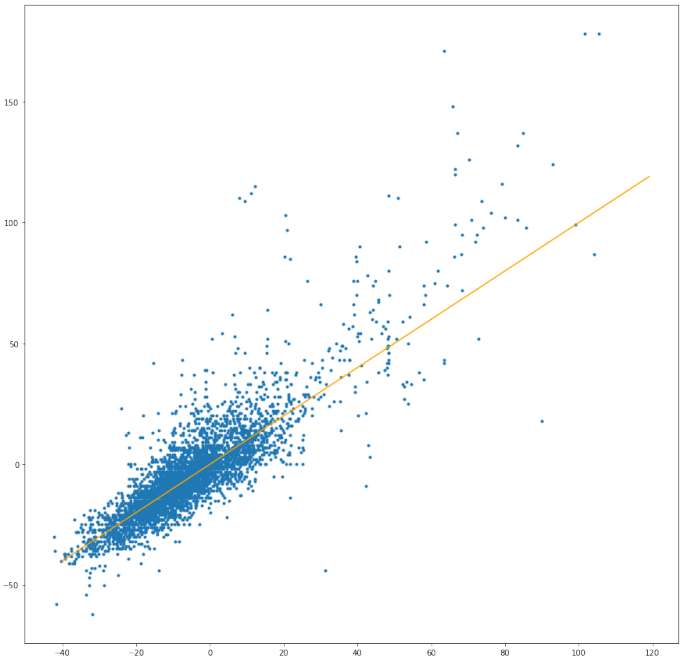
```
# TRYING SUPPORT VECTOR MACHINE ON THE DATA

from sklearn.svm import SVR
svr = SVR(kernel = 'poly',degree = 1, C = 1.0 , max_iter = 10000)

svr.fit(Xp60_train,yt_train)
print(mean_squared_error(svr.predict(Xp60_train),yt_train))
print(mean_squared_error(svr.predict(Xp60_test),yt_test))
```

1.1667758409532987
0.8210629135613802

```
# WE TAKE THE PREDICTED VALUES FROM THE SVR AND USE IT AS A FEATURE IN RANDOM FOREST REGRESSION

X_train_svm = np.concatenate([Xp60_train,svr.predict(Xp60_train).reshape(10313,1)],axis = 1)
X_test_svm = np.concatenate([Xp60_test,svr.predict(Xp60_test).reshape(4420,1)],axis = 1)

data_svm = np.concatenate([X_train_svm,X_test_svm],axis = 0)

# TRAINING RFR ON THE MODIFIED DATASET
RFR = RandomForestRegressor(n_estimators = 500, min_samples_split = 20, max_features = 'auto',n_jobs = -1, random_state = 1)
RFR.fit(X_train_svm,yt_train)
print(mean_squared_error(RFR.predict(X_train_svm),yt_train))
print(mean_squared_error(RFR.predict(X_test_svm),yt_test))
```

0.1264725628541487
0.24969577516553274

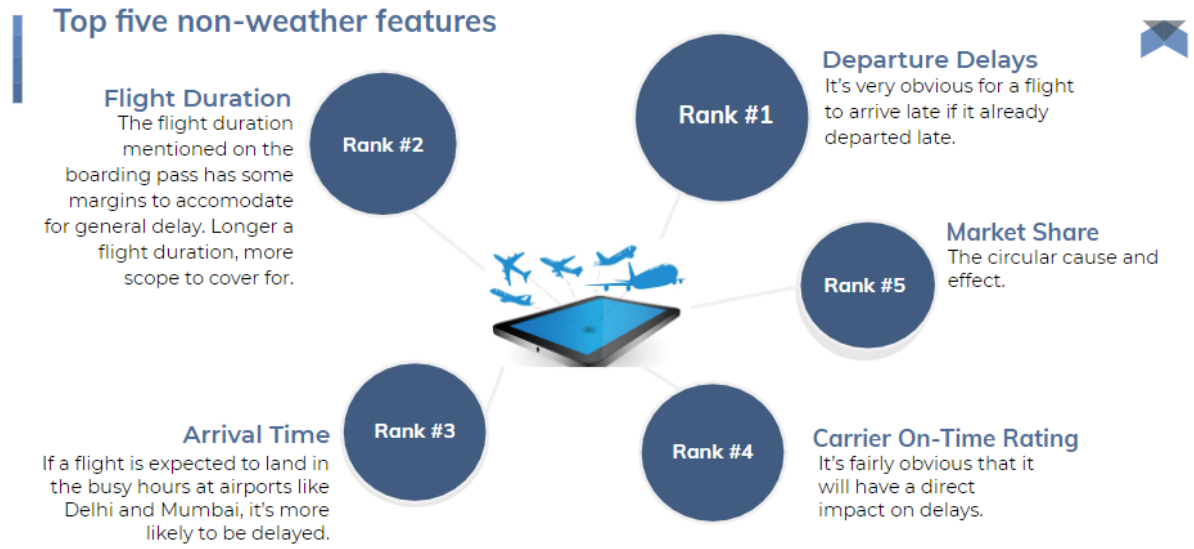
```
g1 = PT.inverse_transform(RFR.predict(X_test_svm).reshape(-1,1)).reshape(-1,)
g2 = PT.inverse_transform(yt_test.reshape(-1,1)).reshape(-1,)

# LOOKING AT THE PREDICTIONS VISUALLY
# PREDICTED VS ACTUAL GRAPH
# y = x line (ORANGE)

plt.figure(figsize=(15,15))
plt.plot(np.arange(-40,120),np.arange(-40,120), color = 'orange')
plt.scatter(g1,g2, marker = '.')
```



CHAPTER 4: CONCLUSION

When feature scores were computed for the tree-based ensemble model, market share and carrier on-time rating (OTR) were the most important airline features. In fact, among the 23 non-weather features, market share ranked 5th. While it's obvious that OTR will have a direct impact on arrival delays, we tried to reason ourselves on why market share was so important.

Fig 4.1 – Top five non-weather features



Note: We dropped the departure delay because it was consuming way too much feature importance, hence it is still the most important non-weather feature.

According to data, higher the market share of an airline carrier, less likely it was to delay. A customer is more likely to choose a better on-time performing airline based on experience and imitation (word of mouth effect).

Indigo being highly on-time over many years makes it more likely for passengers to opt. Thus, market share automatically boosts. Again, once you are a market leader, you always have an extra edge over competitors for provisions to further reduce delays. This is a circular cause and effect.

- Go Air was found to be the airline with least delay with an average delay of 11 minutes.
- SpiceJet was found to be the airline with most delay with an average delay of 18 minutes.
- The Airport which had the most Delay was Bangalore, and the airport with the least delay was Hyderabad.

- Of the given 44 parameters, Wind Gust, Cloud Cover and Airport Location were the most important which affected the delay the most.
- As expected, the Delay is high on Public Holidays and major religious festivals

CHAPTER 5: FUTURE WORK

This project is based on data analysis from year 2019. A large dataset is available from 1987-2019 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 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 India, but we can include more countries like China, United States, 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.

CHAPTER 6: BIBLIOGRAPHY

[1] Shubham Raj, “DATA STORY: The state of India's airlines and the one that's the most punctual of all”, moneycontrol.com, October, 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] S. Choi, Y. J. Kim, S. Briceno and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms", Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th, Sacramento, CA, USA, 2016.

[6] L. Schaefer and D. Millner, "Flight Delay Propagation Analysis with The Detailed Policy Assessment Tool", Man and Cybernetics Conference, Tucson, AZ, 2001.

[7] B. Liu "Sentiment Analysis and Opinion Mining Synthesis", Morgan & Claypool Publishers, p. 167, 2012.

CHAPTER 7: REFERENCES

Airline Data: flightradar24.com

Weather Data: accuweather.com

Scikit-Learn: scikit-learn.org

NumPy: numpy.org

SciPy: scipy.org

Pandas: pandas.org

Matplotlib: matplotlib.org