ML Mini Project

# Flight Fare Prediction

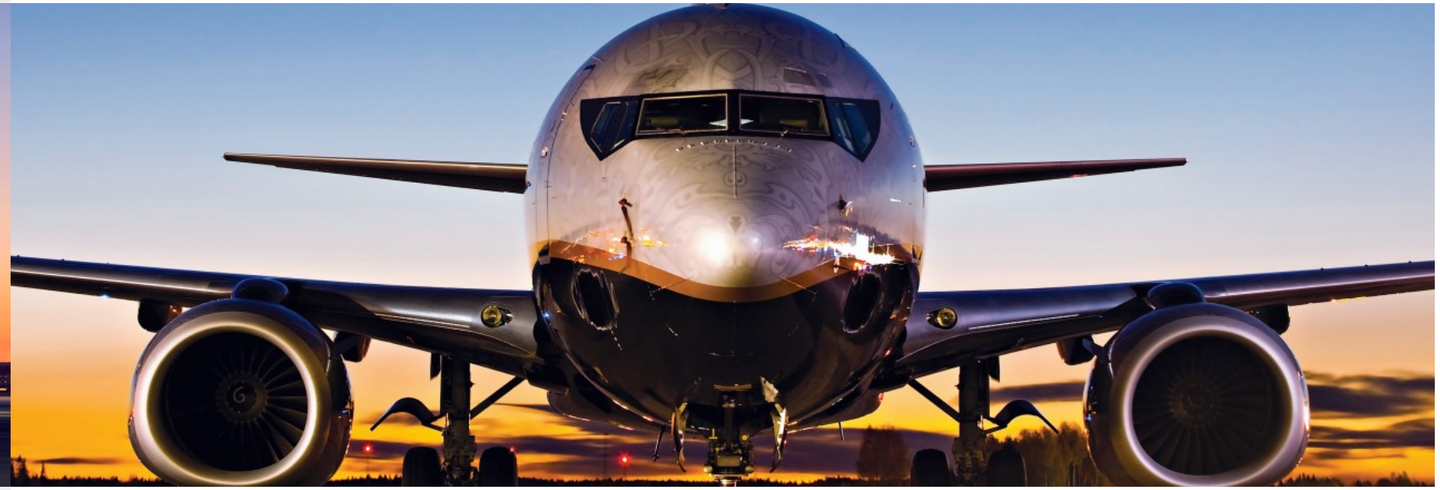
Uvej Pawne 20070122508
Yash Mathur 19070122201
Sakshi Sonawane 20070122517

# Table of Contents

# The Solution

Airfares are constantly fluctuating now more than ever. That's because airlines have access to better technology and more real-time information on passengers than ever before. The more a prospective passenger looks up airfares, the more airlines will tweaking prices based on shifts in demand.

The aim is on doing analysis past airfares and provide a good estimate of what the fares of each flight will be based of past data. We will develop an application for the same where the user can view these fares and book tickets in the minimum possible fare



PART 1

# Engineering tools For design and devlopment and proposed solution

Our coding Environment- Google Colab

Language - Python

Libraries -

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
```

# Our 2 Main Datasets

DATA_TRAIN.XLSX(530.39 KB)

Rows: 10683

Columns: 11

TEST_SET.XLSX(120.77 KB)

Rows: 2671

Columns: 10

Moving forward to find criteria's that affect the price and as we further explore our dataset we find that there are columns with which can deflect the flight prices.

PART 2

# Identify suitable criteria for proposed solution

# Original Columns

| | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info | Price |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | IndiGo | 24/03/2019 | Banglore | New Delhi | BLR → DEL | 22:20 | 01:10 22 Mar | 2h 50m | non-stop | No info | 3897 |
| 1 | Air India | 1/05/2019 | Kolkata | Banglore | CCU → IXR → BBI → BLR | 05:50 | 13:15 | 7h 25m | 2 stops | No info | 7662 |
| 2 | Jet Airways | 9/06/2019 | Delhi | Cochin | DEL → LKO → BOM → COK | 09:25 | 04:25 10 Jun | 19h | 2 stops | No info | 13882 |
| 3 | IndiGo | 12/05/2019 | Kolkata | Banglore | CCU → NAG → BLR | 18:05 | 23:30 | 5h 25m | 1 stop | No info | 6218 |
| 4 | IndiGo | 01/03/2019 | Banglore | New Delhi | BLR → NAG → DEL | 16:50 | 21:35 | 4h 45m | 1 stop | No info | 13302 |

# Data Info

```
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

# Feature Engineering

So after looking at the dataset it was visible that many columns that can change the scenario of the model and most of the features which were absolutely necessary were of object datatype which we converted to Integer Dtype. Reason behind doing this is to transform raw data into features to make it work well on new tasks and train better features.

Since the final goal is to calculate most accurate prices with the reach of the dataset we manipulated such features.

| | Airline | Source | Destination | Route | Total_Stops | Additional_Info | Price | Journey_day | Journey_month | Dep_hour | Dep_min | Arr_hour | Arr_min | duration_hours | duration_min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | IndiGo | Banglore | New Delhi | BLR → DEL | 0 | No info | 3897 | 24 | 3 | 22 | 20 | 1 | 10 | 2 | 50 |
| **1** | Air India | Kolkata | Banglore | CCU → IXR → BBI → BLR | 2 | No info | 7662 | 1 | 5 | 5 | 50 | 13 | 15 | 7 | 25 |
| **2** | Jet Airways | Delhi | Cochin | DEL → LKO → BOM → COK | 2 | No info | 13882 | 9 | 6 | 9 | 25 | 4 | 25 | 19 | 0 |
| **3** | IndiGo | Kolkata | Banglore | CCU → NAG → BLR | 1 | No info | 6218 | 12 | 5 | 18 | 5 | 23 | 30 | 5 | 25 |
| **4** | IndiGo | Banglore | New Delhi | BLR → NAG → DEL | 1 | No info | 13302 | 1 | 3 | 16 | 50 | 21 | 35 | 4 | 45 |

PART 3

# Design identified solution using modern tools and methods

Now that the solution is identified then it is time to apply machine learning techniques for prediction.

# Key ML Techniques Used

In time series type data is always well structured for which regression will be our main statistical method for predicting a future response based on the response history

1

### FITTING MODEL USING RANDOM FOREST

Split dataset into train and test set in order to prediction w.r.t test data
Importing of model, Fit the data, Predict w.r.t X_test
In regression check RSME Score, and plotting appropriate graph

2

### REGRESSION

Splitting the dataset into the Training set and Test set
Training the Random Forest Regression model on the training set
Predicting the Results, and plotting Gaussian distribution to check accuracy

3

### HYPERPARAMETER TUNING

There are two techniques of Hyperparameter tuning i.e 1) RandomizedSearchCv 2) GridSearchCV We use RandomizedSearchCv because it is much faster than GridSearchCV to increase accuracy.

4

### GRADIENT BOOSTING

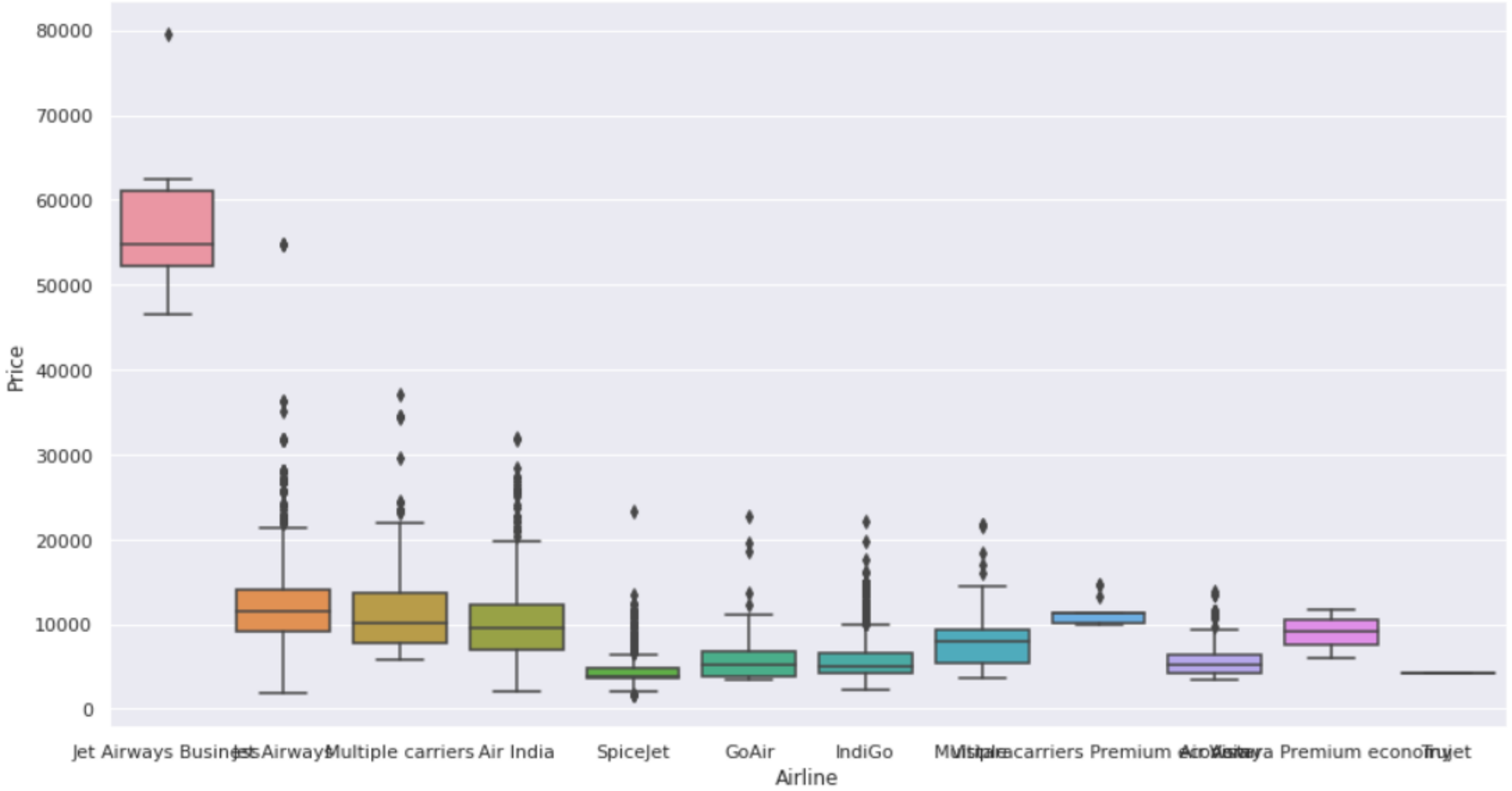
Gradient boosting involves three elements:
A loss function to be optimized, A weak learner to make predictions and an additive model to add weak learners to minimize the loss function.

Data visualization enables an accessible way to see and understand trends, patterns in data, and outliers.

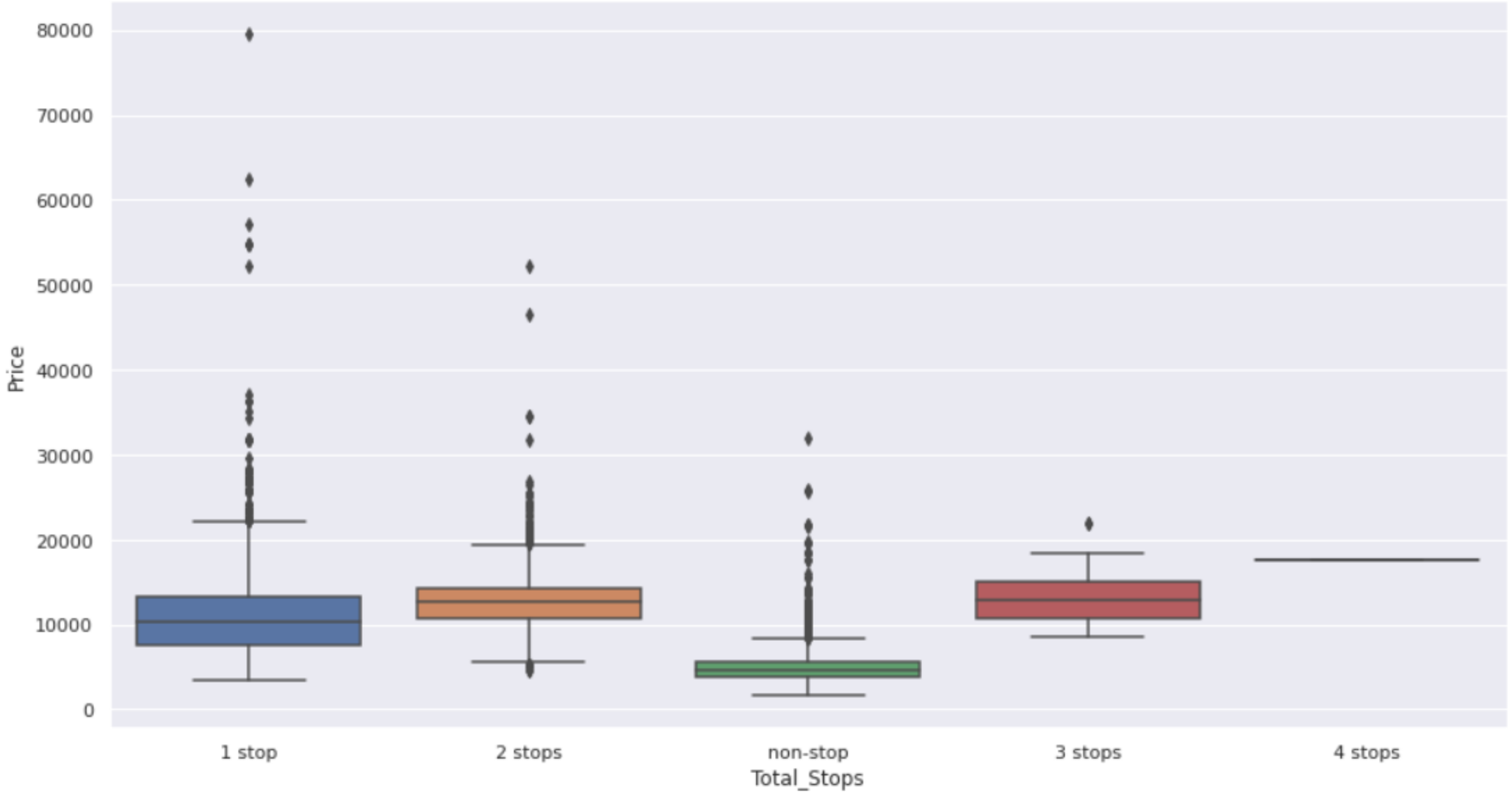
PART 4

# Generate visualizations to understand data and validate solution

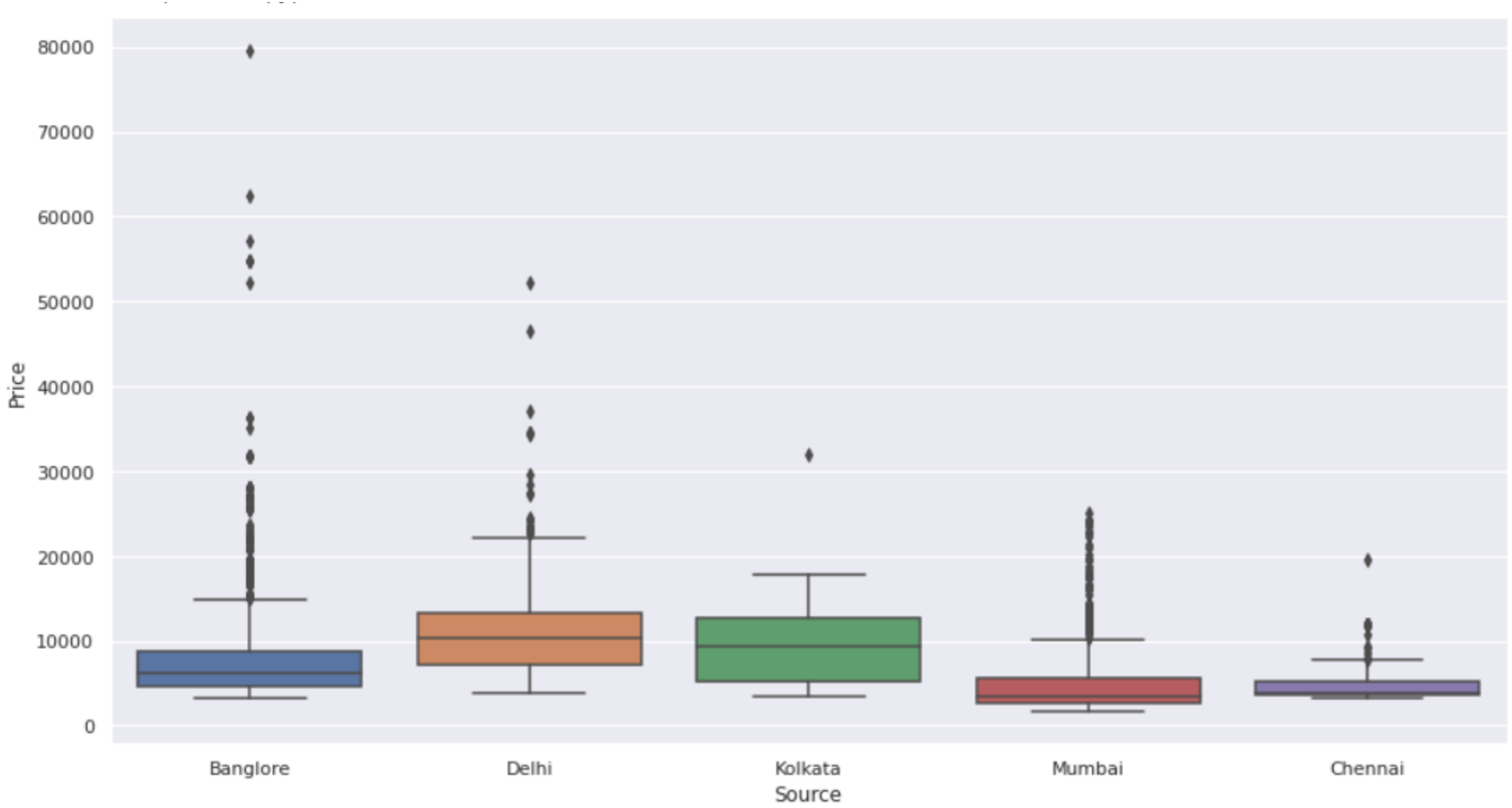
# AIRLINE VS PRICE BOXPLOT



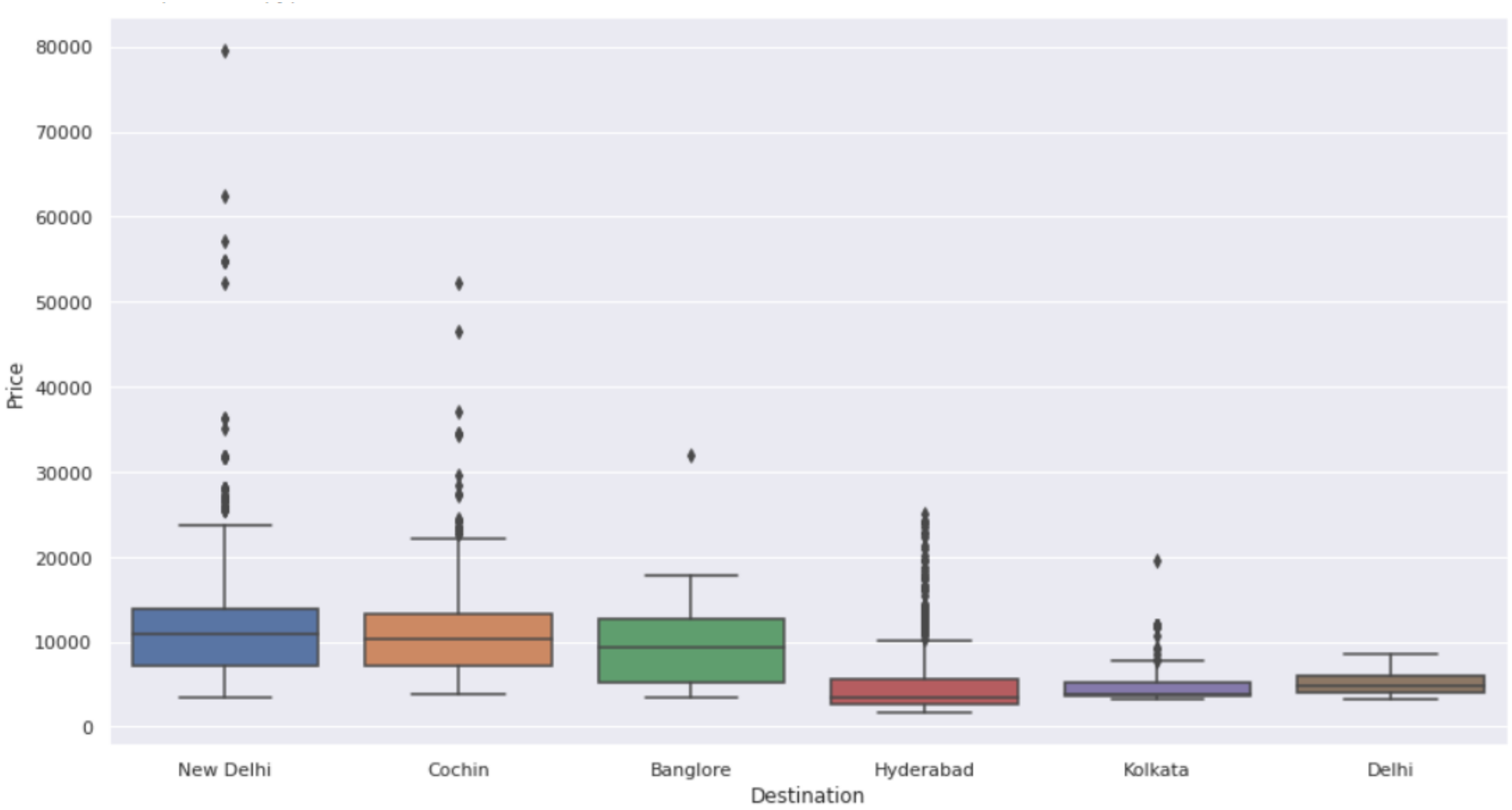
# PRICE VS TOTAL STOPS

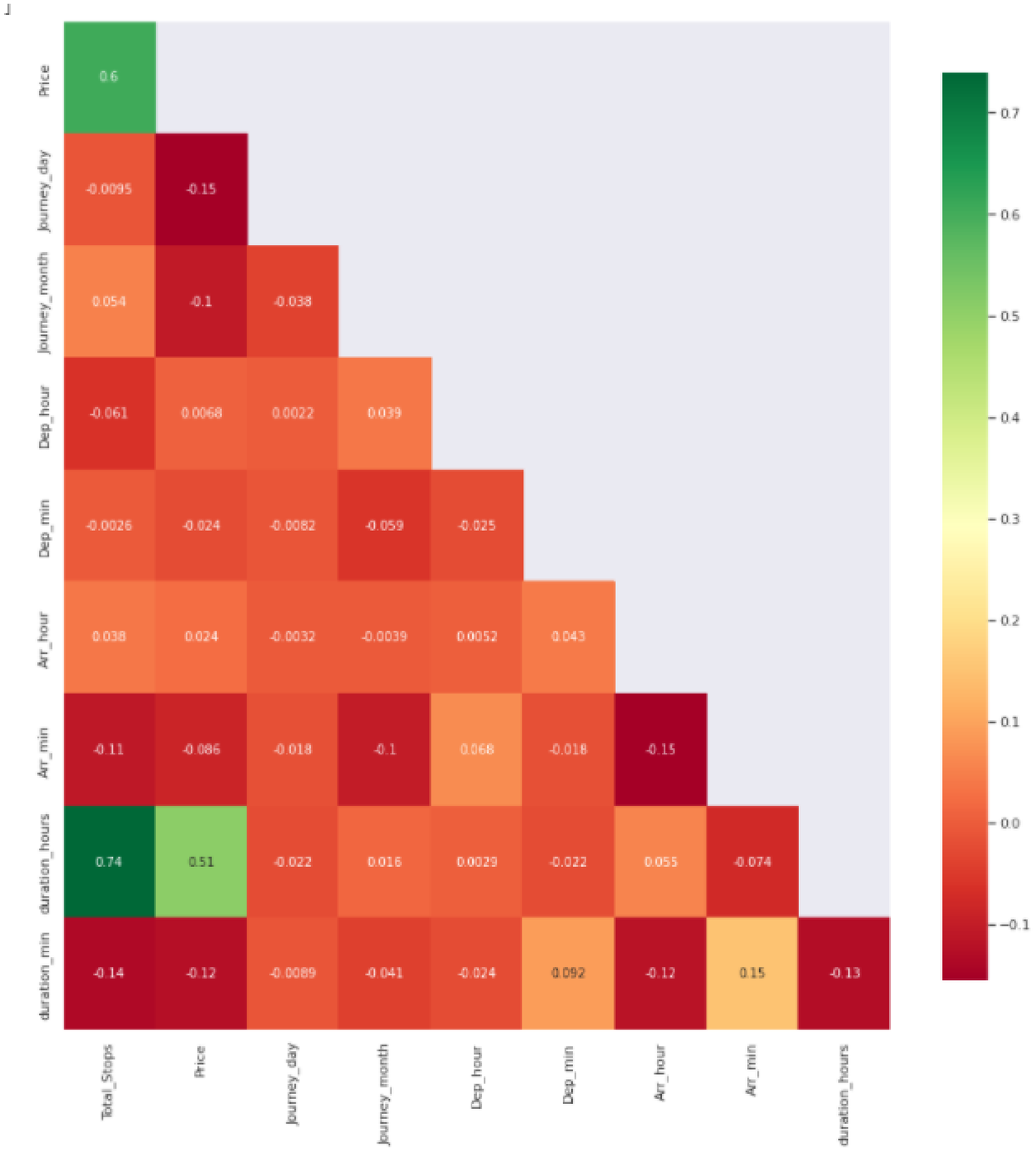


# PRICE VS SOURCE



# PRICE VS DESTINATION






# Heat map for finding correlation between Independent and Dependent Feature

Extreme green means highly correlated, Extreme red means negatively correlated.

If two independent features are highly correlated , then we can drop any one of them as both are doing almost same task.

# Plan of Work( Timeline)

**Target 1**

First we shall frame a Problem statement and obtain a suitable dataset for it.

**Target 2**

We plan choosing a suitable platform to execute our code and work at a collaborative environment. We upload the dataset

**Target 3**

Then we classify and clean the data by removing all the null values and removing the unrequired columns. We organize it and give it a desirable structure for our problem statement

**Target 4**

Via proper visualizations we try to analyze the data and find relations in it and Ultimately, maximizing our insights of a dataset and minimizing potential error that may occur later in the process.

**Target 5**

We model the data and derive predictions from it.

# Random Forest Regressor

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
from sklearn.ensemble import RandomForestRegressor
reg_rf = RandomForestRegressor()
reg_rf.fit(X_train, y_train)
```

```
RandomForestRegressor()
```

```
[ ] y_pred = reg_rf.predict(X_test)
```

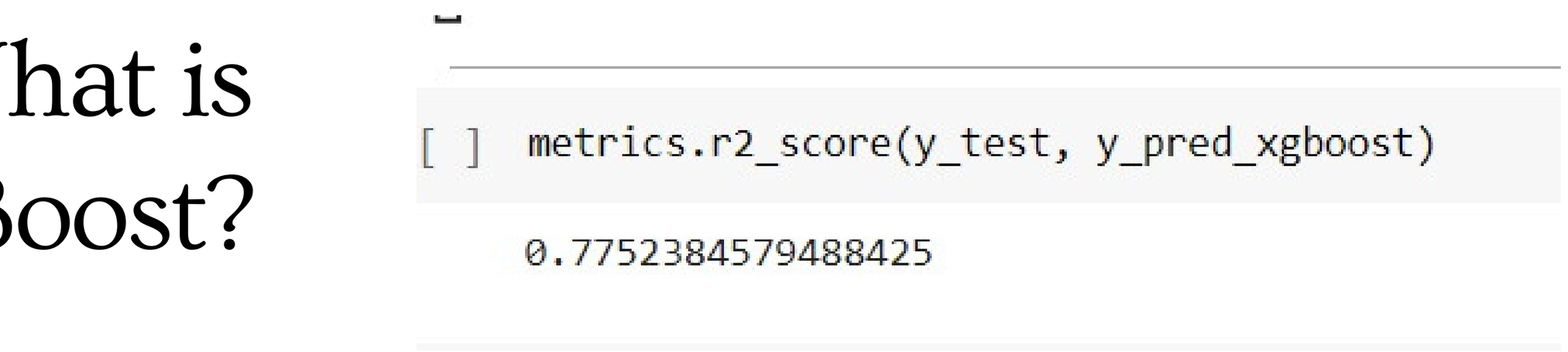
```
[ ] reg_rf.score(X_train, y_train)
```

```
0.9538542119883145
```

```
[ ] reg_rf.score(X_test, y_test)
```

```
0.7977151692684881
```

# What is XGBoost?




XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boostingframework.

The above picture depicts XGBoost's predicted score implemented in our project..

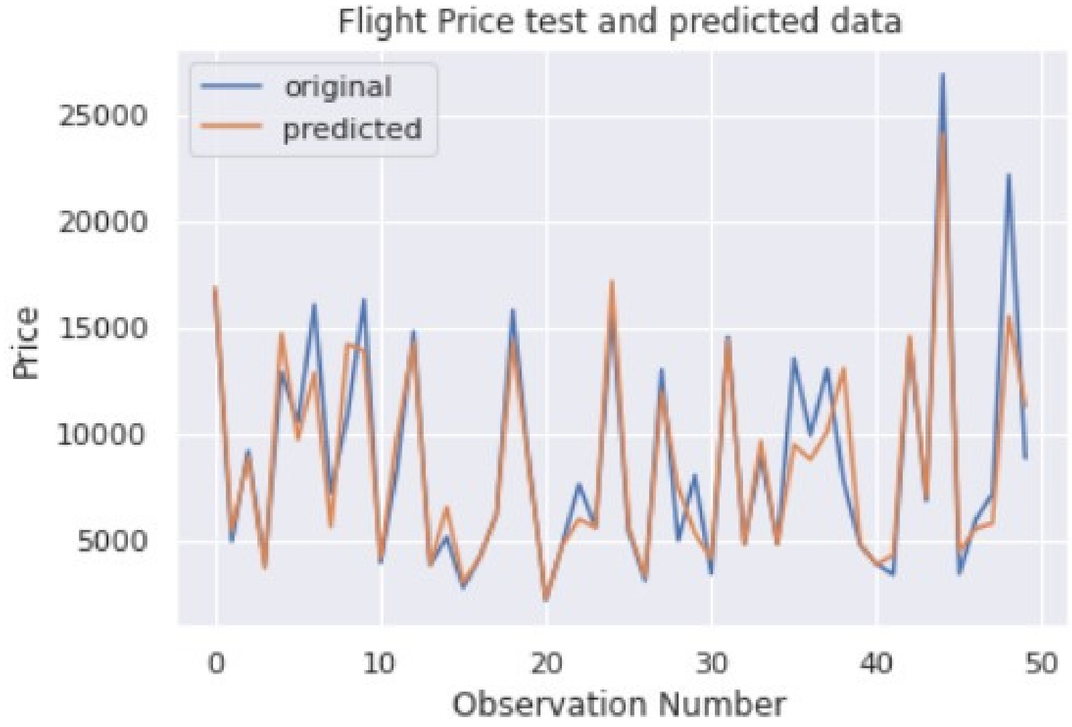
# Regression analysis

Time series regression is a statistical method for predicting future responses based on response history (also known as autoregressive dynamics) and dynamics transfer from relevant predictors.

Creating the Training and Test sets from the dataset
The training set is used to train the Random Forest Regression model.
Predicting the outcome and visualising the Gaussian distribution to ensure accuracy.

# Final Prediction

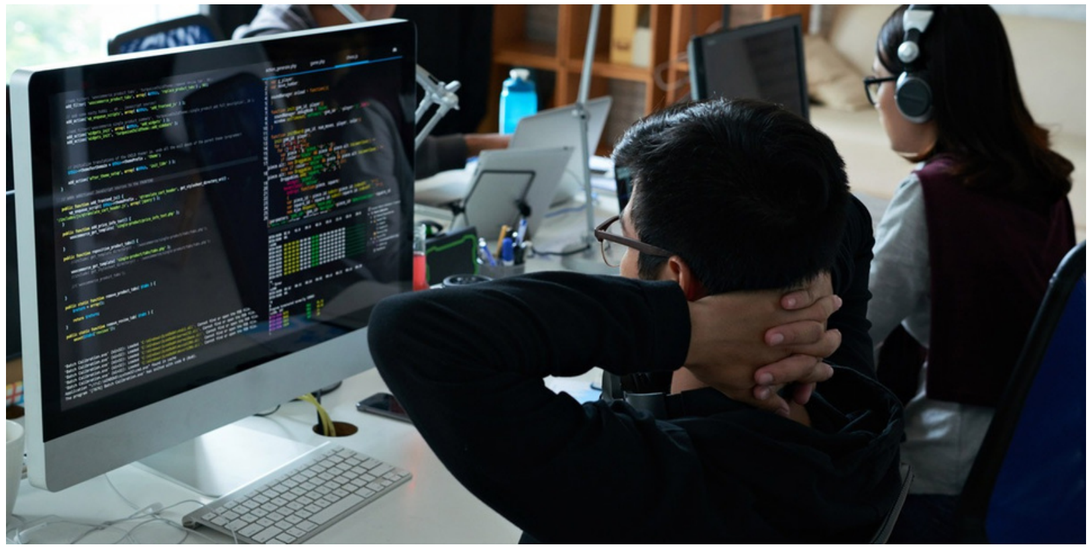
According to this statistical visualization it is clear that ourmodel predictions and original prices are overlapping.



# Future Scope

We tend to make a web application for this project in the future.

Time series analysis is yet one of goals we need to accomplish in the near future for this project.



# Thank you!