



**PROJECT REPORT ON:**  
**“Flight Price Prediction Project”**

**SUBMITTED BY**  
**TANUJ SWARNKAR**

# **ACKNOWLEDGMENT**

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills.

# **Contents:**

## **1. Introduction**

- 1.1 Business Problem Framing:
- 1.2 Conceptual Background of the Domain Problem
- 1.3 Review of Literature
- 1.4 Motivation for the Problem Undertaken

## **2. Analytical Problem Framing**

- 2.1 Mathematical/ Analytical Modeling of the Problem
- 2.2 Data Sources and their formats
- 2.3 Data Preprocessing Done
- 2.4 Data Inputs-Logic-Output Relationships
- 2.5 Hardware and Software Requirements and Tools Used

## **3. Data Analysis and Visualization**

- 3.1 Identification of possible problem-solving approaches (methods)
- 3.2 Testing of Identified Approaches (Algorithms)
- 3.3 Key Metrics for success in solving problem under consideration
- 3.4 Visualization
- 3.5 Run and Evaluate selected models
- 3.6 Interpretation of the Results

## **4. Conclusion**

- 4.1 Key Findings and Conclusions of the Study
- 4.2 Learning Outcomes of the Study in respect of Data Science
- 4.3 Limitations of this work and Scope for Future Work

# 1. INTRODUCTION

## 1.1 Business Problem Framing:

The tourism industry is changing fast and this is attracting a lot more travelers each year. The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Now-a-days flight prices are quite unpredictable. The ticket prices change frequently. Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible. Using technology it is actually possible to reduce the uncertainty of flight prices. So here we will be predicting the flight prices using efficient machine learning techniques.

When booking a flight, travelers need to be confident that they're getting a good deal. The Flight Price Analysis API uses an Artificial Intelligence algorithm trained on Amadeus historical flight booking data to show how current flight prices compare to historical fares. More precisely, it shows how a current flight price sits on a *distribution* of historical airfare prices.

As retrieving price metrics through aggregation techniques and business intelligence tools alone could lead to incorrect conclusions – for example, in cases where insufficient data points to compute specific price statistics have – we used machine learning to forecast prices. This provides an elegant way to interpolate missing data and predict coherent prices. Moreover, we Confirmed the forecast decisions using state of the art Explainable AI techniques.

## 1.2 Conceptual Background of the Domain Problem

Flight prices are something unpredictable. It's more than likely that we spent hours on the internet researching flight deals, trying to figure an airfare pricing system that seems completely random every day. Flight price appears to fluctuate without reason and longer flights aren't always more expensive than shorter ones.

But now the question is how to know proper Flight price, for that I have built a Machine learning model which can predict the Flight price. Using various features like **Airline, Source, Destination, Arrival time, Departure time, Stops, Travelling date and the Price for the same travel**. So using all these previously

known information and analyzing the data I have achieved a good model that has **66% accuracy**. So let's understand what all the steps we did to reach this good accuracy.

Nowadays, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. That's why we will try to use machine learning to solve this problem. This can help airlines by predicting what prices they can maintain. It can also help customers to predict future flight prices and plan their journey accordingly.

### 1.3 Review of Literature

It is hard for the client to buy an air ticket at the most reduced cost. For this few procedures are explored to determine time and date to grab air tickets with minimum fare rate. The majority of these systems are utilizing the modern computerized system known as Machine Learning. The model guesses airfare well in advance from the known information. This framework is proposed to change various added value arrangements into included added value arrangement heading which can support to solo gathering estimation.

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on

1. Time of purchase patterns (making sure last-minute purchases are expensive)
2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

So, we have to work on a project where we collect data of flight fares with other features and work to make a model to predict fares of flights.

## 1.4 Motivation for the Problem Undertaken

Flight Price Prediction project help tourists to find the right flight price based on their needs and also it gives various options and flexibility for travelling.

Different features (airline, source, destination, departure and arrival timings, Journey date etc.) helps to understand the flight price variations. Using it airlines also get benefits and required passengers. Also they will get benefit in scheduling also.

# 2. Analytical Problem

## 2.1 Mathematical/ Analytical Modeling of the Problem

As a first step I have scrapped the required data from makemytrip website. I have fetched data for different source and destinations and saved it to csv format.

In this particular problem I have Price as my target column and it was a continuous column. So clearly it is a regression problem and I have to use all regression algorithms while building the model. There was no null values in the dataset. Since we have scrapped the data from yatra website the raw data was not in the format, so we have use feature engineering to extract the required feature format. To get better insight on the features I have used plotting like distribution plot, bar plot, strip plot and count plot. With these plotting I was able to understand the relation between the features in better manner. I did not found any skewness or outliers in the dataset. I have used all the

regression algorithms while building model then tuned the best model and saved the best model. At last I have predicted the Price using saved model.

## 2.2 Data Sources and their formats

The data was collected from makemytrip.com website in csv format. The data was scrapped using selenium. After scrapping required features the dataset is saved as csv file.

Also, my dataset was having 5204 rows and 9 columns including target. In this particular datasets I have object type of data which has been changed as per our analysis about the dataset. The information about features is as follows.

### **Features Information:**

- Airline: The name of the airline.
- Journey\_date: The date of the journey
- From: The source from which the service begins.
- To: The destination where the service ends.
- Route: The route taken by the flight to reach the destination.
- D\_Time: The time when the journey starts from the source.
- A\_Time: Time of arrival at the destination.
- Stops: Total stops between the source and destination.
- Price: The price of the ticket

## 2.3 Data Preprocessing Done

- ✓ As a first step I have scrapped the required data using selenium from makemytrip website.
- ✓ And I have imported required libraries and I have imported the dataset which was in csv format.
- ✓ Then I did all the statistical analysis like checking shape, nunique, value counts, info etc.....
- ✓ While checking for null values I found there was a row full of null values in the dataset and I dropped that row as it will not help our analysis.
- ✓ I have also dropped Unnamed:0 column as I found it was the index column of csv file.

- ✓ Next as a part of feature extraction I converted the data types of datetime columns and I have extracted useful information from the raw dataset. Thinking that this data will help us more than raw data.

## 2.4 Data Inputs- Logic- Output Relationships

- ✓ Since I had numerical columns I have plotted dist plot to see the distribution of skewness in each column data.
- ✓ I have used bar plot for each pair of categorical features that shows the relation between target and independent features.
- ✓ I have used strip plot to see the relation between numerical columns and target column.
- ✓ I can notice there is a good relationship between maximum columns and target.

## 2.5 Hardware and Software Requirements and Tools Used

While taking up the project we should be familiar with the Hardware and software required for the successful completion of the project. Here we need the following hardware and software.

### **Hardware required: -**

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

### **Software/s**

#### **required: -**

1. Anaconda

### **Libraries required:-**



To run the program and to build the model we need some basic libraries as follows:

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 from scipy import stats
5 import warnings
6 warnings.filterwarnings('ignore')
7 from matplotlib import pyplot as plt
8 from scipy.stats import zscore
```

- ✓ **Import pandas as pd:** pandas is a popular Python-based data analysis toolkit which can be imported using `import pandas as pd`. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.
- ✓ **Import numpy as np:** NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
- ✓ **Import seaborn as sns:** Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.
- ✓ **Import matplotlib.pyplot as plt:** matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.
- ✓ `from sklearn.preprocessing import LabelEncoder`
- ✓ `from sklearn.preprocessing import StandardScaler`
- ✓ `from sklearn.ensemble import RandomForestRegressor`
- ✓ `from sklearn.tree import DecisionTreeRegressor`

- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.neighbors import KNeighborsRegressor as KNN
- ✓ from sklearn.metrics import classification\_report
- ✓ from sklearn.metrics import accuracy\_score
- ✓ from sklearn.model\_selection import cross\_val\_score

With this sufficient libraries we can go ahead with our model building.

## **3. Data Analysis and**

### **3.1 Identification of possible problem-solving approaches (methods)**

- ✓ Since the data collected was not in the format we have to clean it and bring it to the proper format for our analysis. Since there was no outliers and skewness in the dataset no need to worry about that. We have dropped all the unnecessary columns in the dataset according to our understanding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used Standardization to scale the data. After scaling we have to check multicollinearity using VIF. Then followed by model building with all Regression algorithms.

### **3.2 Testing of Identified Approaches (Algorithms)**

Since Price was my target and it was a continuous column with improper format which has to be changed to continuous float datatype column, so this particular problem was Regression problem. And I have used all Regression algorithms to build my model. By looking into the r2 score and error values I found RandomForest Regressor as a best model with highest r2\_score and least error values. Also to get the best model we have to run through multiple. Below are the list of Regression algorithms I have used in my project.

- RandomForestRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor
- DecisionTreeRegressor
- KNN

### 3.3 Key Metrics for success in solving problem under consideration

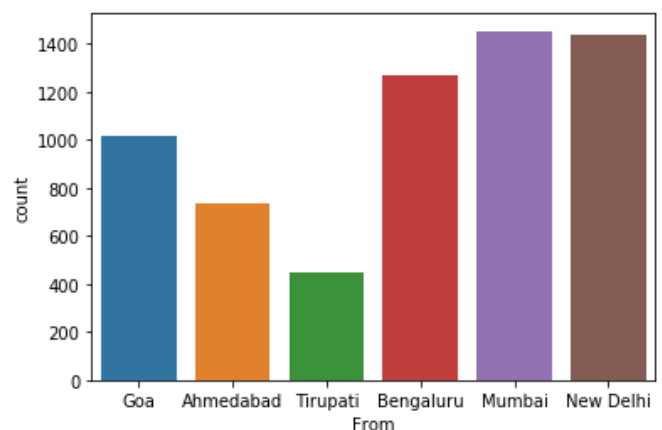
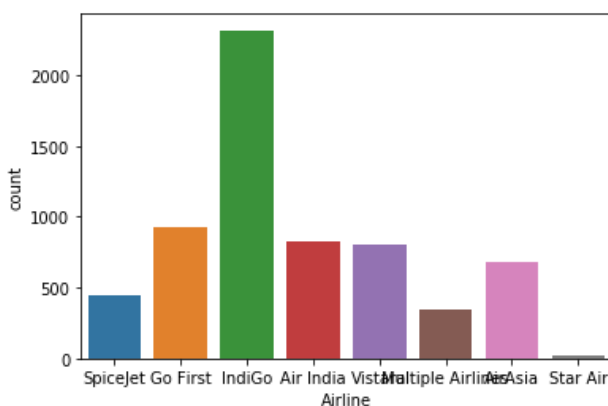
I have used the following metrics for evaluation:

- I have used mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.
- I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.
- I have used r2 score which tells us how accurate our model is.

### 3.4 Visualizations

I have used bar plots to see the relation of categorical feature with target and I have used 2 types of plots for numerical columns one is disp plot for univariate and strip plot for bivariate analysis.

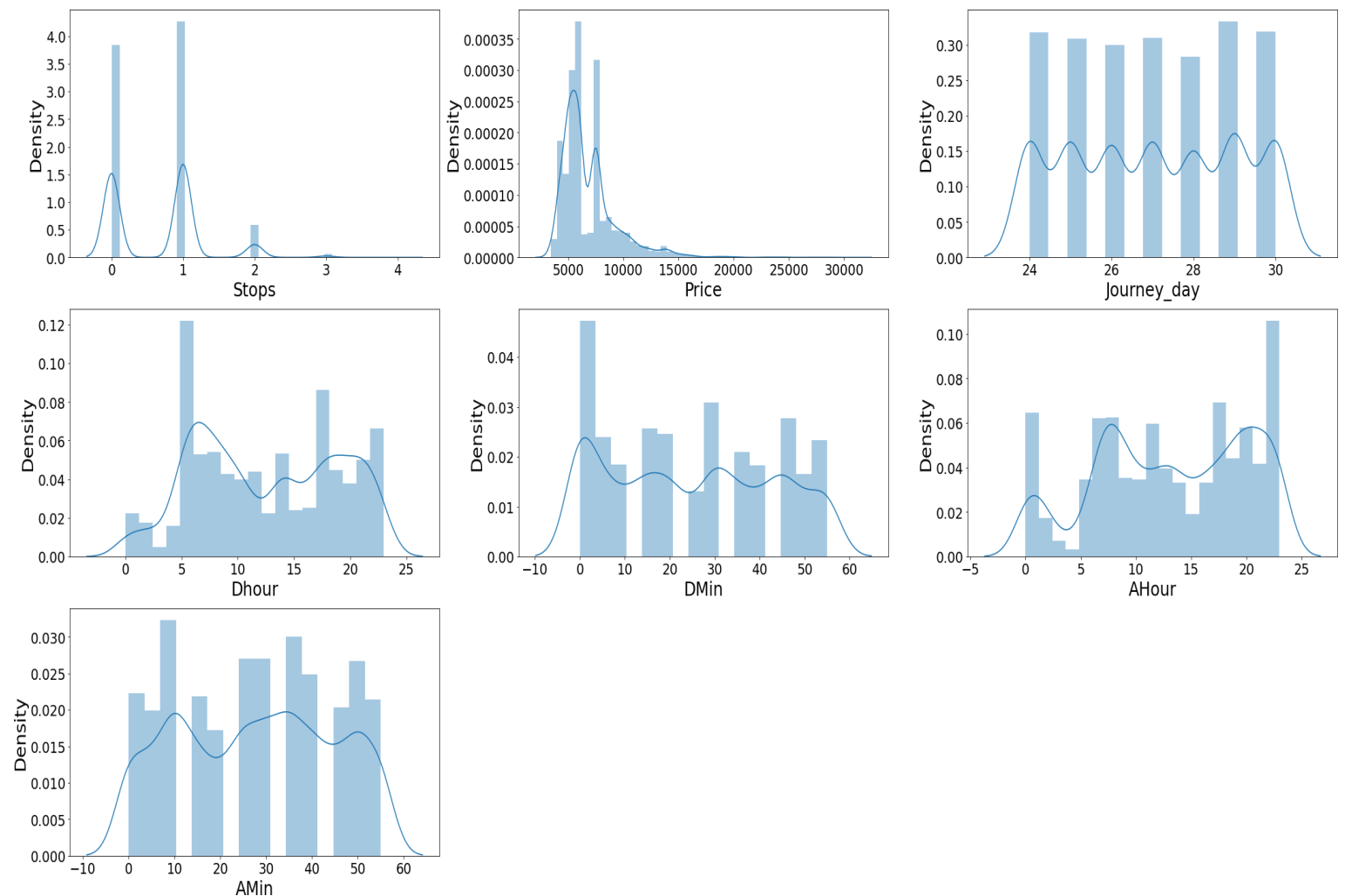
#### 1. Univariate Analysis for Categorical columns:



## Observations:

- ✓ Indigo has maximum count which means most of the passengers preferred Indigo for their travelling.
- ✓ mumbai has maximum count for source which means maximum passengers are choosing New Delhi as there source.

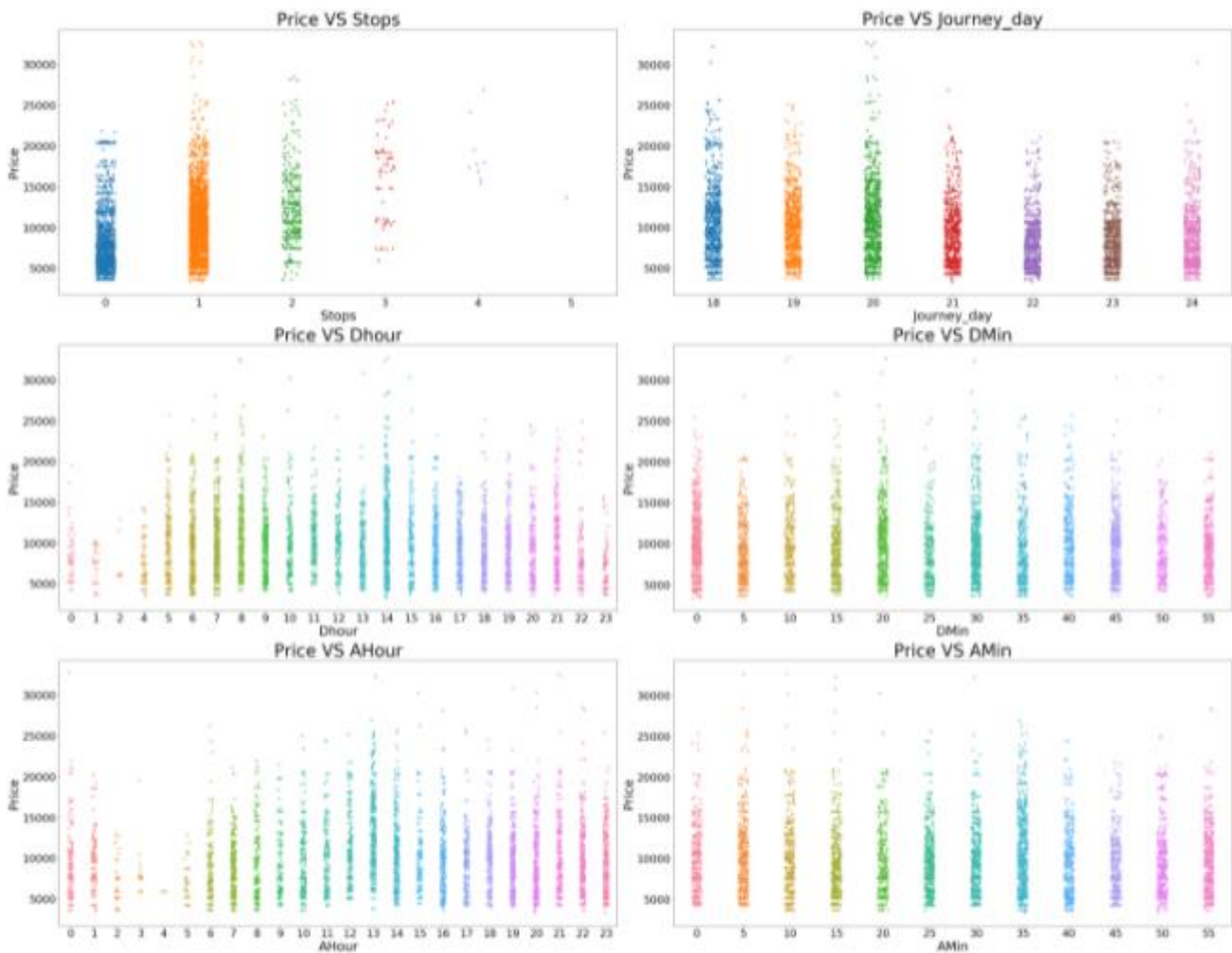
## 2. Univariate analysis for Numerical column:



## Observations:

- ✓ There is no skewness in any of the numerical columns.

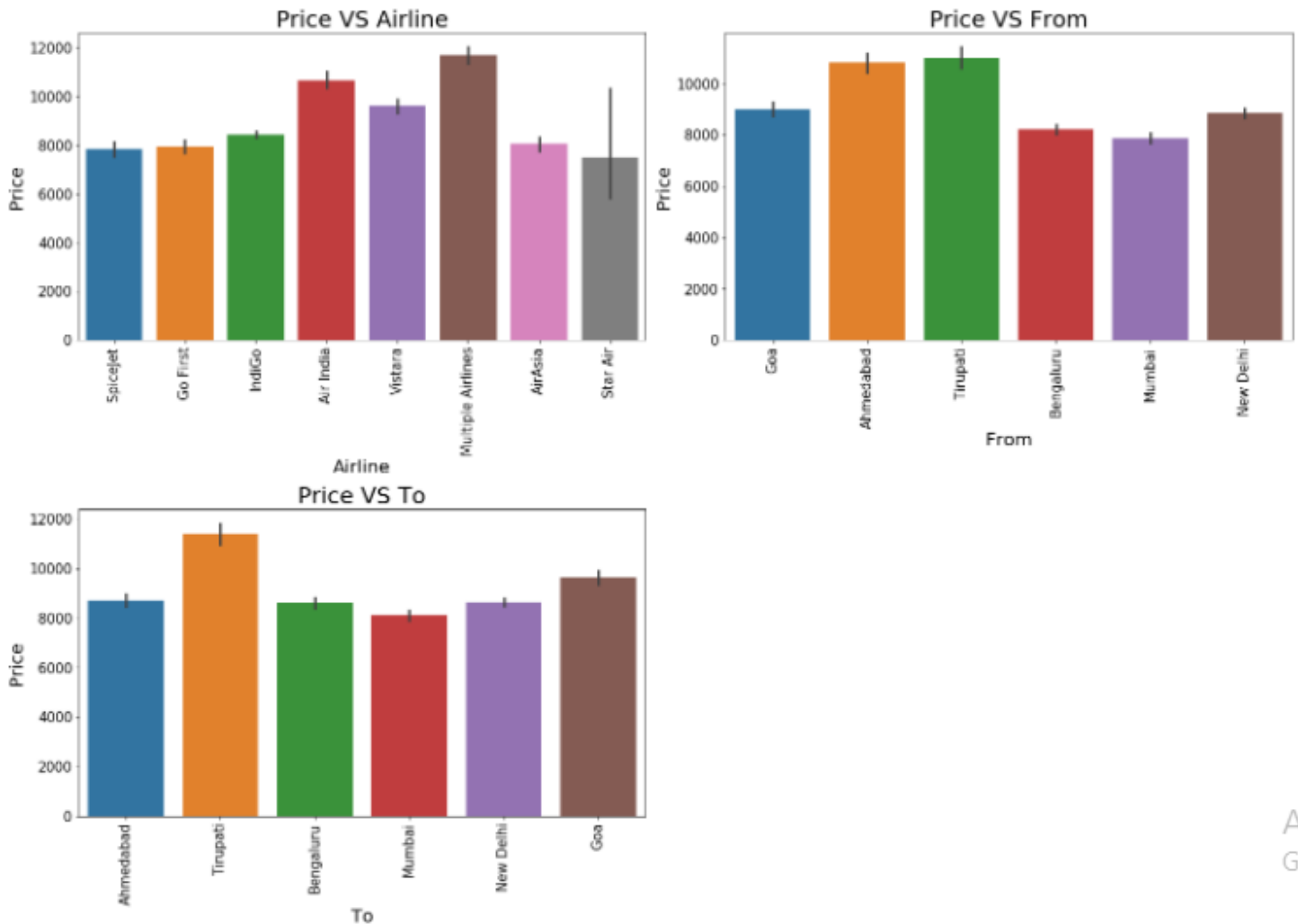
### 3. Bivariate analysis for numerical columns:



### Observations:

- ✓ Flights with 1 stops costs more price compared to other flights.
- ✓ In all the dates the price is almost same.
- ✓ At 2PM departure time of every day the flight Prices are high so it looks good to book flights rather than this departure time.
- ✓ And Departure minute has less relation with target Price.
- ✓ At 7AM to 1PM Arrival time of every day the flight Prices are high so it looks good to book flights rather than this arrival time.
- ✓ And Arrival minute has less relation with target Price.

#### 4. Bivariate Analysis for categorical columns:



#### Observations:

- ✓ For Multiple Airlines the Price is high compared to other Airlines.
- ✓ Taking Tirupati as Source costs highest Price Compared to other Source points.
- ✓ Taking Tirupati as Destination costs highest Price Compared to other Destination points.

## 5. Run and Evaluate selected models

### 1. Model Building:

```
1 x1=df.drop('Price',axis=1)
2 from sklearn.preprocessing import MinMaxScaler
3 scaler=MinMaxScaler()
4 scaled = scaler.fit_transform(x1)
```

```
1 y1=df['Price']
```

```
1 from sklearn.neighbors import KNeighborsRegressor
2 from sklearn.svm import SVR
3 from sklearn.tree import DecisionTreeRegressor
4 from sklearn.linear_model import LinearRegression
5 from sklearn.linear_model import Lasso
6 from sklearn.ensemble import RandomForestRegressor
7 from sklearn.ensemble import GradientBoostingRegressor
8 from sklearn.model_selection import train_test_split
9 x=scaled
10 y=y1
```

```
1 xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.33,random_state=7)
2 from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
3 models=[KNeighborsRegressor(),SVR(),DecisionTreeRegressor(),LinearRegression(),Lasso(),
4          RandomForestRegressor(),GradientBoostingRegressor()]
5 maelist=[]
6 mselist=[]
7 rmselist=[]
8 r2list=[]
```

```

1 def create_model(model):
2     m=model
3     m.fit(xtrain,ytrain)
4     p=m.predict(xtest)
5
6     mae=mean_absolute_error(p,ytest)
7     mse=mean_squared_error(p,ytest)
8     rmse=np.sqrt(mean_squared_error(p,ytest))
9     r2=r2_score(ytest,p)
10
11     maelist.append(mae)
12     mselist.append(mse)
13     rmselist.append(rmse)
14     r2list.append(r2)
15
16     print(m)
17
18     print('Mean absolute error',mae)
19     print('Mean squared error',mse)
20     print('Root Mean squared error',rmse)
21     print('R2 Score',r2)
22     print('-----')
23 for i in models:
24     create_model(i)
25 print('Minimum Mean Absolute error is shown by ',models[maelist.index(min(maelist))],min(maelist))
26 print('Minimum Mean squared error is shown by ',models[mselist.index(min(mselist))],min(mselist))
27 print('Minimum Root Mean squared error is shown by ',models[rmselist.index(min(rmselist))],min(rmselist))
28 print('Maximum R2 Score is shown by ',models[r2list.index(max(r2list))],max(r2list))

```

Activate

```

KNeighborsRegressor()
Mean absolute error 2049.296854146806
Mean squared error 9155849.384442326
Root Mean squared error 3025.8634113988564
R2 Score 0.4073247128942179

```

```

-----
SVR()
Mean absolute error 2850.6045912825916
Mean squared error 17111882.61917947
Root Mean squared error 4136.651135783566
R2 Score -0.1076842265967839

```

```

-----
DecisionTreeRegressor()
Mean absolute error 1625.001906577693
Mean squared error 8791395.857006673
Root Mean squared error 2965.028812171422
R2 Score 0.4309164726469147

```

```

-----
LinearRegression()
Mean absolute error 2558.5505583255185
Mean squared error 12563200.96281796
Root Mean squared error 3544.4606025202143
R2 Score 0.18676046044860783

```

```

-----
Lasso()
Mean absolute error 2558.2790681239194
Mean squared error 12558197.85357431
Root Mean squared error 3543.7547676968716
R2 Score 0.1870843210848958

```

```

-----
RandomForestRegressor()
Mean absolute error 1457.33936423578
Mean squared error 5574366.696306543
Root Mean squared error 2361.0096773004857
R2 Score 0.6391607983656644

```

```

-----
GradientBoostingRegressor()
Mean absolute error 1958.9060547882316
Mean squared error 8015756.862442889
Root Mean squared error 2831.2112006070633
R2 Score 0.48112503817603525

```

```

-----
Minimum Mean Absolute error is shown by RandomForestRegressor() 1457.33936423578
Minimum Mean squared error is shown by RandomForestRegressor() 5574366.696306543
Minimum Root Mean squared error is shown by RandomForestRegressor() 2361.0096773004857
Maximum R2 Score is shown by RandomForestRegressor() 0.6391607983656644

```



I have choose all parameters of RandomForest, after tuning the model with best parameters I have increased my model accuracy from 63% to 66%.

## 5. Saving the model and Predictions:

- I have saved my best model using .pkl as follows.

```
1 # Saving the model using .pkl
2 from joblib import dump
3 dump(model1, 'flight.joblib')

['flight.joblib']
```

# 4. CONCLUSION

## 4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the flight prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. Thus we can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to seven algorithms and a hyper parameter tuning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on those metrics. Then we have also saved the best model and predicted the flight price. It was good the predicted and actual values were almost same.

## 4.2 Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it and it was self-scraped from makemytrip website using selenium. Improvement in computing technology has made it

possible to examine social information that cannot previously be captured, processed and analyzed. New analytical techniques of machine learning can be used in flight price research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values and null values. This study is an exploratory attempt to use seven machine learning algorithms in estimating flight price prediction, and then compare their results.

To conclude, the application of machine learning in predicting flight price is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting online platforms, and presenting an alternative approach to the valuation of flight price. Future direction of research may consider incorporating additional used flight data from a larger economical background with more features.

