FLIP ROBO

NAME OF THE PROJECT:

A Project Report on Housing Price Prediction

Submitted by:

A KISHORE KUMAR

Batch No.:

Internship_33

Under the guidance of:

Mr. Shwetank Mishra

(FlipRobo SME)

ACKNOWLEDGMENT

I wish to express my sincere thanks and deep sense of gratitude to “Flip Robo” team, who has given me this opportunity to deal with an informative dataset and it has helped me to improve my problem analyzation skills and Machine Learning modelling.

Also, I want to express my huge gratitude to Mr. Shwetank Mishra (SME Flip Robo) for his tremendous support, to get me out of all the difficulties I faced while going through this project and for the successful completion of this work. He has been a great source of inspiration to work with and I shall always cherish my association with him with immense pleasure.

Finally, A huge thanks to “Data trained” who gave me the opportunity to get the Internship at Flip-Robo.

References use in this project:

1. SCIKIT Learn Library Documentation

2. Blogs from towardsdatascience, Analytics Vidya, Medium.

3. Andrew Ng Notes on Machine Learning (GitHub).

4. Data Science Projects with Python Second Edition by Packt.

5. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron.

6. B. Smith, J. Leimkuhler, R. Darrow, and Samuels, “Yield management at American airlines, “Interfaces, vol. 22, pp. 8–31, 1992.

7. William Groves, Maria Gini, “An agent for optimizing airline ticket purchasing”, in international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2013).

8. Chen, Y., Cao, J., Feng, S., Tan, Y., 2015. An ensemble learning based approach for building airfare forecast service. In: 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 964-969.

9. Yeamduan Narangajavana, Fernando.J. Garrigos-Simon, Javier Sanchez García, Santiago Forgas-Coll, “Prices, prices and prices: A study in the airline sector”, Tourism Manage., 41 (2014), pp. 28-42.

10. Bo An, Haipeng Chen, Noseong Park, V.S. Subrahmanian MAP: Frequency-Based Maximization of Airline Profits based on an Ensemble Forecasting Approach Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), ACM, New York, NY, USA (2016), pp. 421-430.

11. R. Ren, Y. Yang, and S. Yuan, “Prediction of airline ticket price,” University of Stanford, 2014.

12. T. Janssen, T. Dijkstra, S. Abbas, and A. C. van Riel, “A linear quantile mixed regression model for prediction of airline ticket prices,” Radboud University, 2014.

13. K. Tziridis, T. Kalampokas, G. A. Papakostas, and K. I. Diamantaras, “Airfare prices prediction using machine learning techniques,” in the 25th IEEE European signal processing conference, 2017, pp. 1036– 1039.

14. C. Koopmans and R. Lieshout, “Airline cost changes: To what extent are they passed through to the passenger?” Journal of Air Transport Management, vol. 53, pp. 1–11, 2016.

15. G. Francis, A. Fidato, and I. Humphreys, “Airport–airline interaction: the impact of low-cost carriers on two European airports,” Journal of Air Transport Management, vol. 9, no. 4, pp. 267–273, 2003.

16. Boruah A., Baruah K., Das B., Das M.J., Gohain N.B. (2019) “A Bayesian Approach for Flight Fare Prediction Based on Kalman Filter,” https://doi.org/10.1007/978-981-13-0224-4_18

17. G.A. Papakostas, K.I. Diamantaras and T. Papadimitriou, “Parallel pattern classification utilizing GPU-Based kernelized slackmin algorithm,” doi:10.1016/j.jpdc.2016.09.001

18. T. Janssen, “A linear quantile mixed regression model for prediction of airline ticket prices,” Bachelor Thesis, Radboud University, 2014.

Chapter 1: Introduction

1.1. Business Problem Framing:

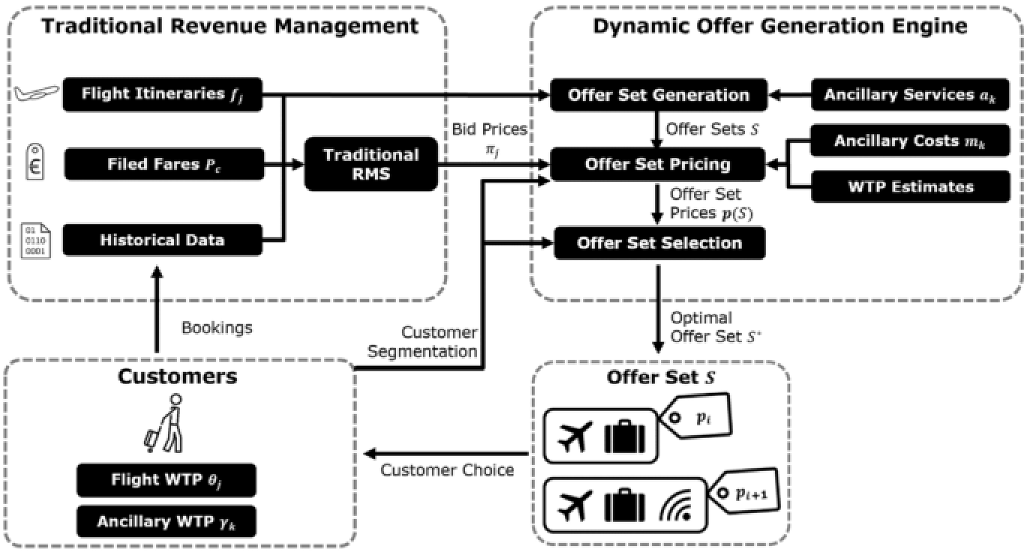
The Airline Companies is considered as one of the most enlightened industries using complex methods and complex strategies to allocate airline prices in a dynamic fashion. These industries are trying to keep their all-inclusive revenue as high as possible and boost their profit. Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible and maximize their profit. However, mismatches between available seats and passenger demand usually leads to either the customer paying more or the airlines company losing revenue. Airlines companies are generally equipped with advanced tools and capabilities that enable them to control the pricing process. However, customers are also becoming more strategic with the development of various online tools to compare prices across various airline companies. In addition, competition between airlines makes the task of determining optimal pricing is hard for everyone. 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

- Time of purchase patterns (making sure last-minute purchases are expensive)
- 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, this project involves collection of data for flight fares with other features and building a model to predict fares of flights.

1.2. Conceptual Background of the Domain Problem:

A report says India's affable aeronautics industry is on a high development movement. India is the third-biggest avionics showcase in 2020 and the biggest by 2030. Indian air traffic is normal to cross the quantity of 100 million travellers by 2017, whereas there were just 81 million passengers in 2015. Agreeing to Google, the expression" Cheap Air Tickets" is most sought in India. At the point when the white-collar class of India is presented to air travel, buyers searching at modest costs. Any individual who has booked a flight ticket previously knows how dynamically costs change. Aircraft uses advanced strategies called Revenue Management to execute a distinctive valuing strategy [6]. The least expensive accessible ticket changes over a period the cost of a ticket might be high or low. This valuing method naturally modifies the toll as per the time like morning, afternoon or night. Cost may likewise change with the seasons like winter, summer and celebration seasons.



The extreme goal of the carrier is to build its income yet on the opposite side purchaser is searching at the least expensive cost. Purchasers generally endeavor to purchase the ticket in advance to the take-off day. From the customer point of view, determining the minimum price or the best time to buy a ticket is the key issue. The conception of ''tickets bought in advance are cheaper" is no longer working (William Groves and Maria Gini, 2013) [7]. It is possible that customers who bought a ticket earlier pay more than those who bought the same ticket later. Moreover, early purchasing implies a risk of commitment to a specific schedule that may need to be changed usually for a fee. Most of the studies performed on the customer side focus on the problem of predicting optimal ticket purchase time using statistical methods. As noted by Y. Chen et al. (2015) [8], predicting the actual ticket price is a more difficult task than predicting an optimal ticket purchase time due to various reasons: absence of enough datasets, external factors influencing ticket prices, dynamic behaviour of ticket pricing, competition among airlines, proprietary nature of airlines ticket pricing policies etc. Early prediction of the demand along a given route could help an airline company pre-plan the flights and determine appropriate pricing for the route. Existing demand prediction models generally try to predict passenger demand for a single flight/route and market share of an individual airline. Price discrimination allows an airline company to categorize customers based on their willingness to pay and thus charge them different prices. Customers could be categorized into different groups based on various criteria such as business vs leisure, tourist vs normal traveler, profession etc. For example, business customers are willing to pay more as compared to leisure customers as they rather focus on service quality than price. In a less competitive market, the market power of a given airline is stronger, and thus, it is more likely to engage in price discrimination. On the other hand, the higher the level of competition, the weaker of the market power of an airline, and then the less likely the chance of the airline fare increases.

1.3. Review of Literature:

On the airlines side, the main goal is increasing revenue and maximizing profit. According to (Narangajavana et al., 2014) [9], airlines utilize various kinds of pricing strategies to determine optimal ticket prices: long-term pricing policies, yield pricing which describes the impact of production conditions on ticket prices, and dynamic pricing which is mainly associated with dynamic adjustment of ticket prices in response to various influencing factors. Among the recent work performed on route demand and market share prediction is the study done by (Bo An et al., 2016) [10]. The authors proposed a data mining technique designed for Maximizing Airline Profits (MAP) through prediction of total route demand and market share of an individual airline. Unlike most other works, this work considers a broad set of routes (around 700 routes) across 13 airlines operating in those routes. The training dataset spans 10 years (40 quarters) while the testing set includes the first quarter of 2015 (a total of 9100 predictions). However, the prediction is performed quarterly and not for a short period of time which might not consider dynamic demand changes. Moreover, the routes considered are only national routes in the US. Ren et al. [11] proposed using LR, Naive Bayes, Soft-max regression, and SVMs to build a prediction model and classify the ticket price into five bins (60% to 80%, 80% to 100%, 100% to 120%, and etc.) to compare the relative values with the overall average price. More than nine thousand data points, including six features (e.g., the departure week begin, price quote date, the number of stops in the itinerary, etc.), were used to build the models. The authors reported the best training error rate close to 22.9% using LR model. Their SVM regression model failed to produce a satisfying result. Instead, an SVM classification model was used to classify the prices into either “higher” or “lower” than the average. In [12], four LR models were compared to obtain the best fit model, which aims to provide an unbiased information to the passenger whether to buy the ticket or wait longer for a better price. The authors suggested using linear quantile mixed models to predict the lowest ticket prices, which are called the “real bargains”. However, this work is limited to only one class of tickets, economy, and only on one direction single leg flights from San Francisco Airport to John F. Kennedy Airport. Tziridis et al. [13] applied eight machine learning models, which included ANNs, RF, SVM, and LR, to predict tickets prices and compared their performance. The best regression model achieved an accuracy of 88%. In their comparison, Bagging Regression Tree is identified as the best model, which is robust and not affected by using different input feature sets. Macroeconomic data, such as crude oil price and Consumer Price Index (CPI), can also be utilized to uncover the hidden trend in airline fares. Fuel costs can take up to 50% of the total operating cost of an airline [14].

Hence, the level of crude oil price plays an essential rule of formulating the airline's pricing strategy. It is a common practice for airlines to pass the cost of aviation fuel to the customer by adjusting the fare to compensate for the fluctuation of crude oil price. The emergence of Low-Cost Carrier (LCC) has revolutionized the entire operating model of the airline industry. The presence of LCC in a market has had a substantial impact on the total passenger volume and the air ticket price [15]. In detail monitoring, the passenger gets an approximation of plane price with date to choose the best blend of date and price. The price for weekend on Sunday is not possible to calculate in this presented model, as weekend on Sundays the most accidental price difference compared to other days in the week and needs more elements, nonlinear model for successful forecast which will be the upcoming range of study to be done for this presented technique [16]. To forecast the mean plane ticket amount on the business area, machine learning support was evolved. Selecting feature techniques authors have presented model to forecast the mean flight amount with R squared score of 80% accuracy. The accuracy of logistic regression model is up to 70-75%. The conclusion of the given model is that most of the plane ticket price vary from day to day. Authors have reported that the ticket price is high for a certain period and then it gradually decreases to a certain level. When the flight is at a difference of 2-3 days' time the ticket price starts increasing again [17]. Janssen [18] built up an expectation model utilizing the Linear Quantile Blended Regression strategy for San Francisco to New York course with existing every day airfares given by www.infare.com. The model utilized two highlights including the number of days left until the take-off date and whether the flight date is at the end of the week or weekday. The model predicts airfare well for the days that are a long way from the take-off date, anyway for a considerable length of time close the take-off date, the expectation isn't compelling. Business class flights are more inelastic as compared to leisure class as business customers have less flexibility to change or cancel their travel date (Mumbower et al., 2014) [19]. In contrast, short distance flights are more elastic (more price sensitive) than long distance flights because of the availability of other travel options (e.g., bus, train, car etc.). Airlines use price elasticity information to determine when to increase ticket prices or when to launch promotions so that the overall demand is increased.

1.4. Motivation for the Problem Undertaken:

The project was the first provided to me by Flip Robo Technologies as a part of the internship program. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary motivation. Early prediction of the demand along a given route could help an airline company pre-plan the flights and determine appropriate pricing for the route. In addition, competition between airlines makes the task of determining optimal pricing is hard for everyone. So prime motive is to build flight price predication system based on short range timeframe (7- 14 days) data available prior to actual take-off date.

Chapter 2 : Analytical Problem Framing

2.1. Mathematical / Analytical Modelling of the Problem:

First phase of problem modelling involves data scraping of flights from internet. For that purpose, flight data is scrap from www.yatra.com for timeframe of 12 Jan 2023 to 23 Jan 2023. Data is scrape for flights on route of New Delhi to Mumbai. Data is scrap for Economy class, Premium Economy class & Business class flights. Next phase is data cleaning & pre-processing for building ML Model. Our objective is to predict flight prices which can be resolve by use of regression-based algorithm. Further Hyperparameter tuning performed to build more accurate model out of best model.

2.2. Data Sources and their formats:

Data is collected from www.yatra.com for timeframe of 12 Jan 2023 to 23 Jan 2023 using selenium and saved in CSV file. Data is scrape for flights on route of New Delhi to Mumbai. Data is scrap for Economy class, Premium Economy class & Business class flights. Around 2500 flights details are collected for this project.

```
# Importing dataset excel file using pandas:
df= pd.read_csv('Flight_dataset_.csv')
```

```
print('No. of Rows :',df.shape[0])
print('No. of Columns :',df.shape[1])
pd.set_option('display.max_columns',None) # This will enable us to see truncated columns
df.head()
```

```
No. of Rows : 2271
No. of Columns : 12
```

	Unnamed: 0	Airline	Class	Aeroplane	Date	Departure_Time	Arrival_Time	Source	Destination	Stops	Duration	Price
0	0	SpiceJet	Economy	SG-2976/279	Thu, 12 Jan 2023	20:10	10:30\n+ 1 day	New Delhi	Mumbai	1 Stop	14h 20m	5270
1	1	Air Asia	Economy	I5-743/762	Thu, 12 Jan 2023	04:55	13:05	New Delhi	Mumbai	1 Stop	8h 10m	5383
2	2	Air Asia	Economy	I5-1228/613	Thu, 12 Jan 2023	09:30	19:00	New Delhi	Mumbai	1 Stop	9h 30m	5383
3	3	Air Asia	Economy	I5-740/613	Thu, 12 Jan 2023	08:10	19:00	New Delhi	Mumbai	1 Stop	10h 50m	5383
4	4	Air Asia	Economy	I5-1228/1782	Thu, 12 Jan 2023	09:30	21:00	New Delhi	Mumbai	1 Stop	11h 30m	5383

Unnecessary column of index name as ‘Unnamed: 0’ is drop out. There are 12 features in dataset including target feature ‘Price’. The data types of different features are as shown below:

```
# lets sort columns by their datatype
df.columns.to_series().groupby(df.dtypes).groups
```

```
{int64: ['Price'], object: ['Airline', 'Class', 'Aeroplane', 'Date', 'Departure_Time', 'Arrival_Time', 'Source', 'Destination',
'Stops', 'Duration']}
```

2.3. Data Pre-processing

The dataset is large and it may contain some data error. In order to reach clean, error free data some data cleaning & data pre-processing performed data.

- **Data Integrity check –**

• **Dataset can contain whitespaces, missing value, duplicates entries, let investigate integrity of data before proceeding for further analysis.**

```
df.duplicated().sum()
```

```
0
```

```
df.isin([' ','?','-','null','NA']).sum().any()
```

```
False
```

No missing values or duplicate entries present in dataset.

- **Conversion of Duration column from hr & Minutes format into Minutes –**

By default, Duration of flights are given in format of [(hh) hours:(mm)minute] which need to convert into uniform unit of time. Here we have written code to convert duration in terms of minute. For example,

```
df['Duration'] = df['Duration'].map(lambda x : x.replace('05m','5m'))
```

```
# Conversion of Duration column from hr & Minutes format to Minutes
df['Duration'] = df['Duration'].str.replace('h','*60').str.replace(' ','+').str.replace('m','*1').apply(eval)

# convert this column into a numeric datatypes
df['Duration']= pd.to_numeric(df['Duration'])
```

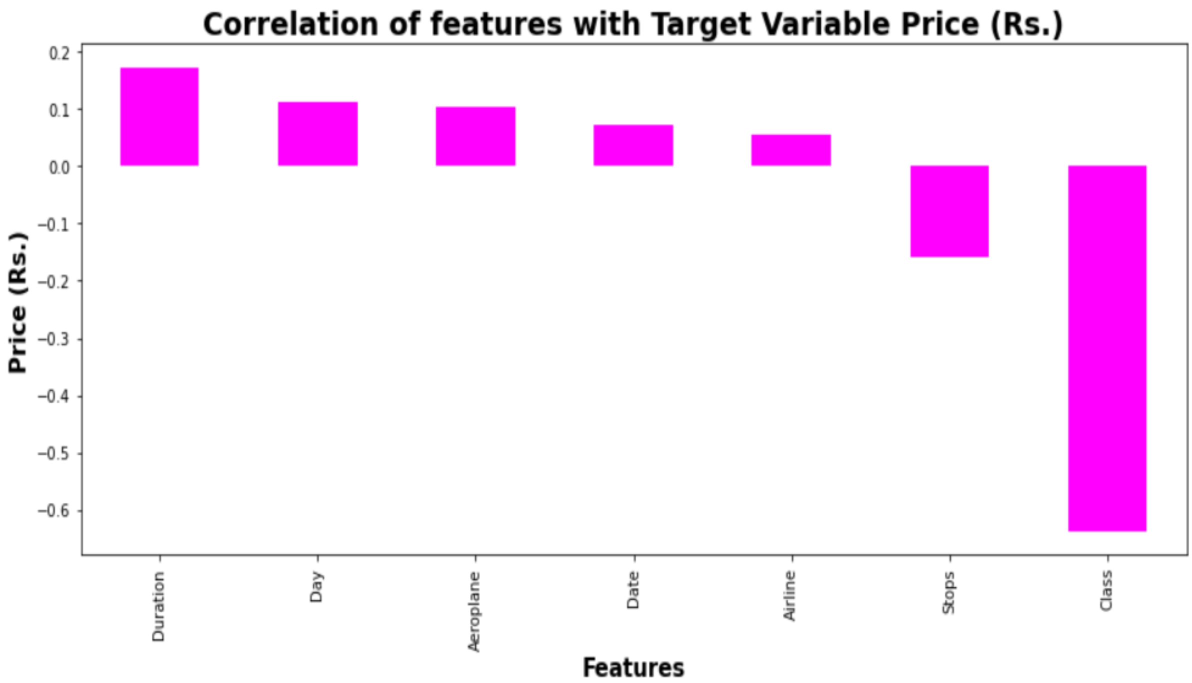
- **Create new column for day & date –**

New column for 'Day' & 'Date' is extracted from Date column.

```
df['Day']= df['Date'].map(lambda x :x[:3])
```

```
df['Date']= df['Date'].map(lambda x :x[4:])
```

2.4. Data Inputs- Logic- Output Relationships



Correlation heatmap is plotted to gain understanding of relationship between target features & independent features. We can see that class feature is correlated for more than -0.6 with target variable Price. Remaining feature are poorly correlated with target variable price.

2.5. Hardware & Software Requirements with Tool Used

Hardware Used -
Processor — AMD Ryzen 5.
RAM — 8 GB
GPU —AMD Radeon(TM) Vega 8.
Software Used -
Anaconda – Jupyter Notebook
Different libraries are used while building ML model and Visualisation of data.
Libraries Used – General libraries used for data wrangling

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # ploting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

Libraries used for web scraping data from website are:

```
# Let's import all required Libraries:
import pandas as pd
import numpy as np
from selenium.webdriver.common.keys import Keys
import selenium # Library that is used to work with selenium.
from selenium import webdriver # Importing webdriver module from selenium to open automated chrome window
import pandas as pd # to create Dataframe
from selenium.webdriver.common.by import By # importing inbuilt class by
import warnings # to ignore any sort of warning
warnings.filterwarnings("ignore")

import requests
import re # regular expression

# importing delays:

import time # used to stop engine for few seconds
from selenium.webdriver.support.ui import WebDriverWait # used to stop engine for few seconds
from selenium.webdriver.support import expected_conditions as EC # EC contains a set of predefined conditions to use with WebDriv

# importing exceptions:
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import StaleElementReferenceException
from selenium.common.exceptions import ElementNotVisibleException
from selenium.common.exceptions import ElementNotInteractableException
from selenium.common.exceptions import SessionNotCreatedException
from selenium.common.exceptions import TimeoutException
```

Libraries used for machine learning model building

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import ExtraTreesRegressor
from xgboost import XGBRegressor
```

Chapter 3:
Models Development & Evaluation

3.1. Identification Of Possible Problem-Solving Approaches (Methods)

First part of problem solving is to scrap data from www.yatra.com website which we already done. Next part of problem solving is building machine learning model to predict flight price. This problem can be solve using regression-based machine learning algorithm like linear regression. For that purpose, first task is to convert categorical variable into numerical features. Once data encoding is done then data is scaled using standard scalar. Final model is built over this scaled data. For building ML model before implementing regression algorithm, data is split in training & test data using train_test_split from model_selection module of sklearn library. After that model is train with various regression algorithm and 5-fold cross validation is performed. Further Hyperparameter tuning performed to build more accurate model out of best model.

3.2. Testing of Identified Approaches (Algorithms)

Web Scraping Strategy employed in this project as follow:

I. Selenium will be used for web scraping data from www.yatra.com
II. Flights on route of New Delhi to Mumbai in duration of 12 Jan 2023 to 23 Jan 2023.
III. Data is scrap in three parts:
    - Economy class flight price extraction
    - Business class flight price extraction
    - Premium Economy class price extraction
IV. Selecting features to be scrap from website.
V. In next part web scraping code executed for above mention details.
VI. Exporting final data in Excel file.

The different regression algorithm used in this project to build ML model are as below:

I. Linear Regression
II. Random Forest Regressor
III. Decision Tree Regressor
IV. XGB Regressor
V. Extra Tree Regressor

3.3. KEY METRICS FOR SUCCESS IN SOLVING PROBLEM

Following metrics used for evaluation:

I. Mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.
II. Root mean square error is one of the most commonly used measures for evaluating the quality of predictions.
III. R2 score which tells us how accurate our model predict result, is going to important evaluation criteria along with Cross validation score.

3.4. RUN AND EVALUATE SELECTED MODELS

1. Linear Regression:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 70, test_size=0.33)
lin_reg= LinearRegression()
lin_reg.fit(X_train, Y_train)
y_pred = lin_reg.predict(X_test)
print('\033[1m'+ 'Error :'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred))
print('Mean squared error :', mean_squared_error(Y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred)))
print('\033[1m'+' R2 Score :'+'\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error :
Mean absolute error : 6926.525438402277
Mean squared error : 81737772.95136161
Root Mean squared error : 9040.894477393353
 R2 Score :
62.21866081364096
```

2. Random Forest Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 70, test_size=0.33)
rfc = RandomForestRegressor()
rfc.fit(X_train, Y_train)
y_pred = rfc.predict(X_test)
print('\033[1m'+ 'Error of Random Forest Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred))
print('Mean squared error :', mean_squared_error(Y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred)))
print('\033[1m'+'R2 Score of Random Forest Regressor :'+'\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error of Random Forest Regressor:
Mean absolute error : 2881.78448
Mean squared error : 18859615.945131734
Root Mean squared error : 4342.765932574738
R2 Score of Random Forest Regressor :
91.28259161928123
```

3. Decision Tree Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 70, test_size=0.33)
dtc = DecisionTreeRegressor()
dtc.fit(X_train, Y_train)
y_pred = dtc.predict(X_test)
print('\033[1m'+ 'Error of Decision Tree Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred))
print('Mean squared error :', mean_squared_error(Y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred)))
print('\033[1m'+'R2 Score of Decision Tree Regressor :'+'\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error of Decision Tree Regressor:
Mean absolute error : 3441.0253333333335
Mean squared error : 33560112.017333336
Root Mean squared error : 5793.1090113455775
R2 Score of Decision Tree Regressor :
84.48763736181593
```

4. Extra Trees Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 70, test_size=0.33)
etc = ExtraTreesRegressor()
dtc.fit(X_train, Y_train)
y_pred = dtc.predict(X_test)
print('\033[1m'+ 'Error of Extra Tree Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred))
print('Mean squared error :', mean_squared_error(Y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred)))
print('\033[1m'+'R2 Score of Extra Tree Regressor :'+'\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error of Extra Tree Regressor:
Mean absolute error : 3398.848
Mean squared error : 33945659.816
Root Mean squared error : 5826.290399216297
R2 Score of Extra Tree Regressor :
84.3094270726434
```

5. XGB Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 70, test_size=0.33)
xgb = XGBRegressor()
xgb.fit(X_train, Y_train)
y_pred = xgb.predict(X_test)
print('\033[1m'+ 'Error of XGB Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred))
print('Mean squared error :', mean_squared_error(Y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred)))
print('\033[1m'+'R2 Score of XGB Regressor :'+'\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error of XGB Regressor:
Mean absolute error : 2971.5372174479166
Mean squared error : 18562267.77679951
Root Mean squared error : 4308.395034905633
R2 Score of XGB Regressor :
91.42003372956343
```

3.5. Cross validation:

Fold cross validation performed over all models. We can see that XGB Regressor gives maximum R2 score of 91.42 and maximum cross validation score. Among all model we will select XGB Regressor as final model and we will perform hyper parameter tuning over this model to enhance its R2 Score.

```
from sklearn.model_selection import GridSearchCV
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state=70, test_size=0.33)
```

```
parameter = {'n_estimators':[400,500],'gamma':np.arange(0,0.2,0.1),
             'booster' : ['gbtree','dart','gblinear'], 'max_depth':[6,8],
             'eta' : [0.01, 0.1] }
```

```
GCV = GridSearchCV(XGBRegressor(),parameter,verbose =10)
```

```
GCV.fit(X_train,Y_train)
```

```
GCV.best_params_
```

```
{'booster': 'gbtree',
 'eta': 0.01,
 'gamma': 0.0,
 'max_depth': 6,
 'n_estimators': 500}
```

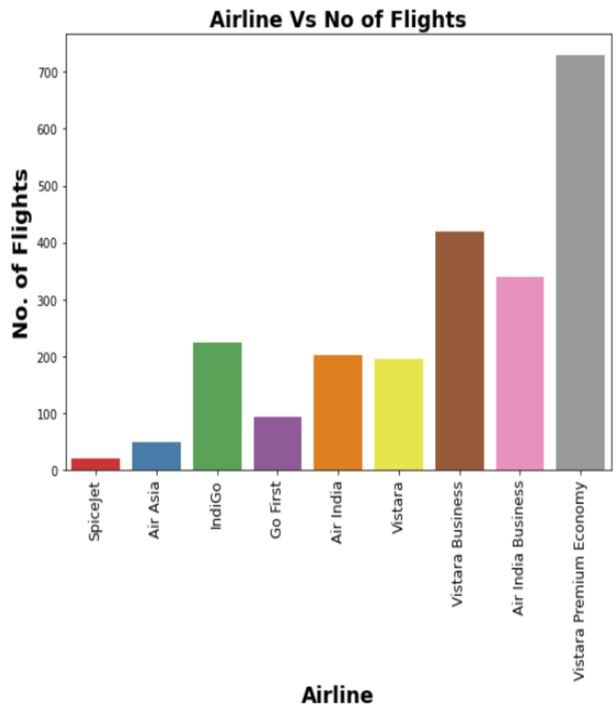
Final model is built with best params got in hyper parameter tuning.

```
Final_mod=XGBRegressor(booster='gbtree', max_depth=6, eta=0.01,
                       gamma=0.0, n_estimators=500)
Final_mod.fit(X_train,Y_train)
pred=Final_mod.predict(X_test)
print('R2_Score:',r2_score(Y_test,pred)*100)
print('mean_squared_error:',mean_squared_error(Y_test,pred))
print('mean_absolute_error:',mean_absolute_error(Y_test,pred))
print("RMSE value:",np.sqrt(mean_squared_error(Y_test, pred)))
```

```
R2_Score: 91.04201895960865
mean_squared_error: 19380081.176330384
mean_absolute_error: 3033.594888671875
RMSE value: 4402.281360423296
```

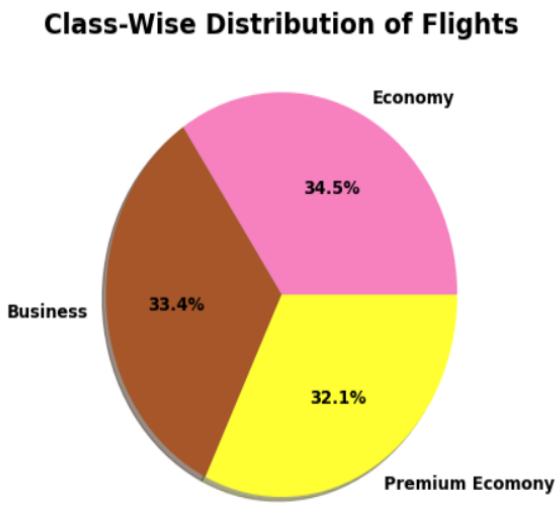
3.6. VISUALIZATIONS

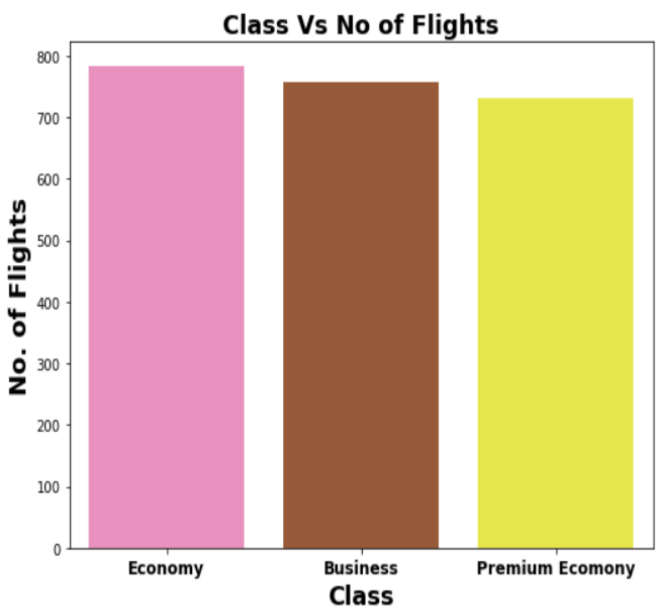
Let see key result from EDA, start with flight-wise distribution of airlines.



Observation:

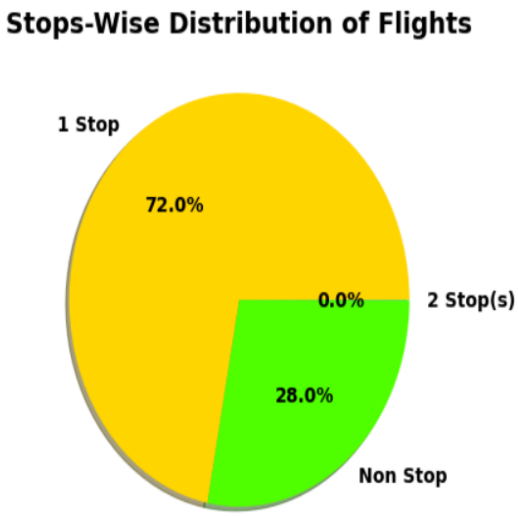
- We can see maximum number of flights run by Vistara Premium Economy while minimum Flights run by Spicejet.

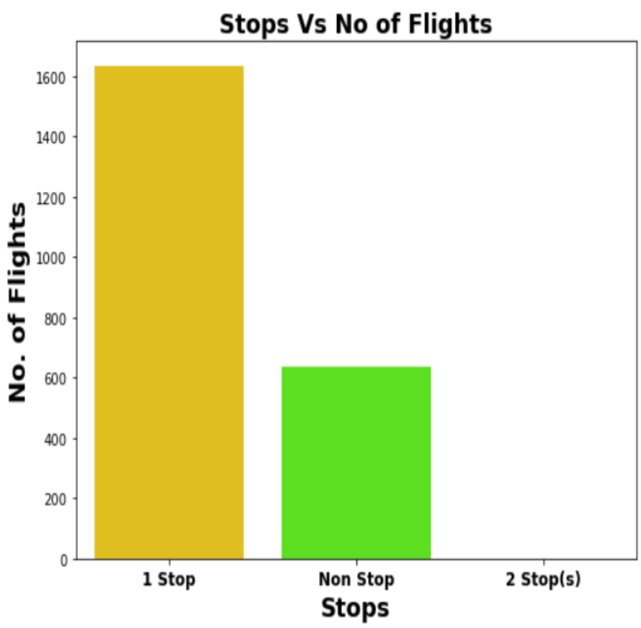




Observation:

- To have a balanced dataset for price prediction all the three classes are being integrated into the dataset in equal proportions.

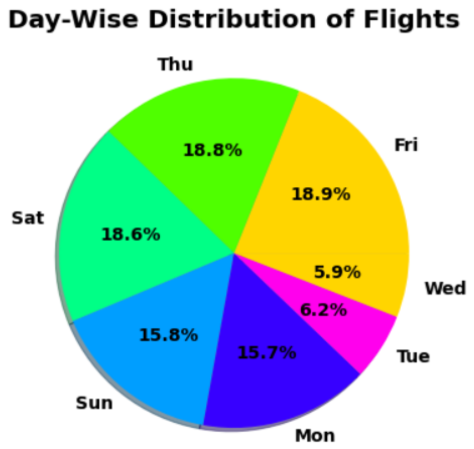


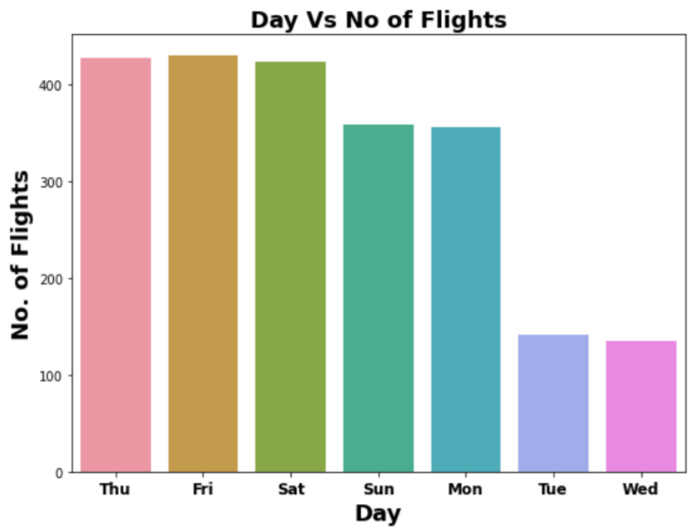


Observation:

- 72% flights take single stop in there way from New Dehli to Mumbai.It is also possible that these flights may have high flight duration compare to Non-stop Flight.

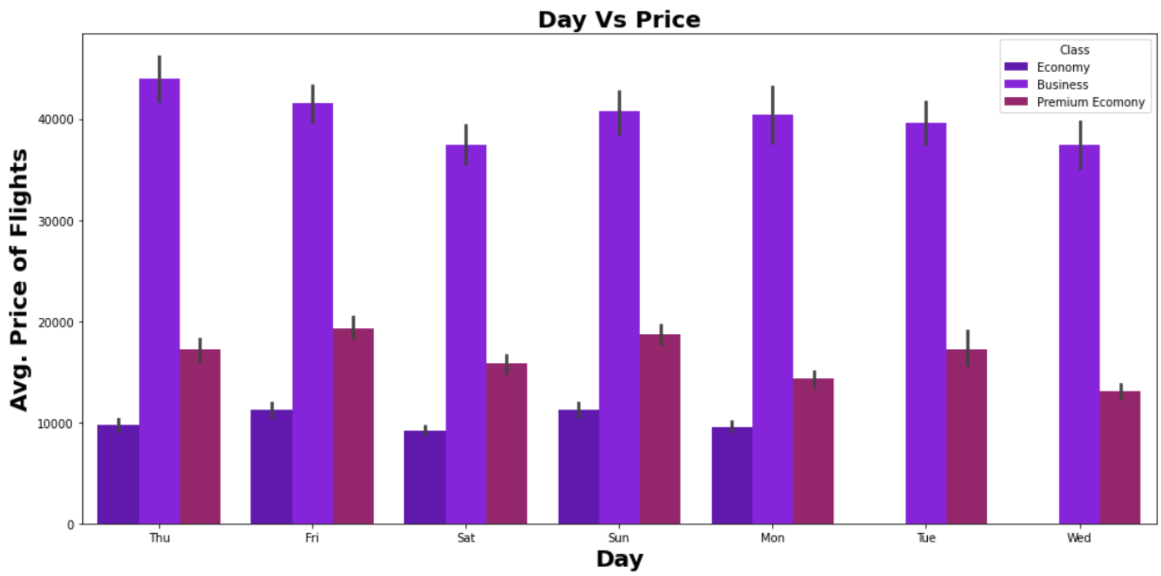
- 28% of flights do not have any stop in their route.





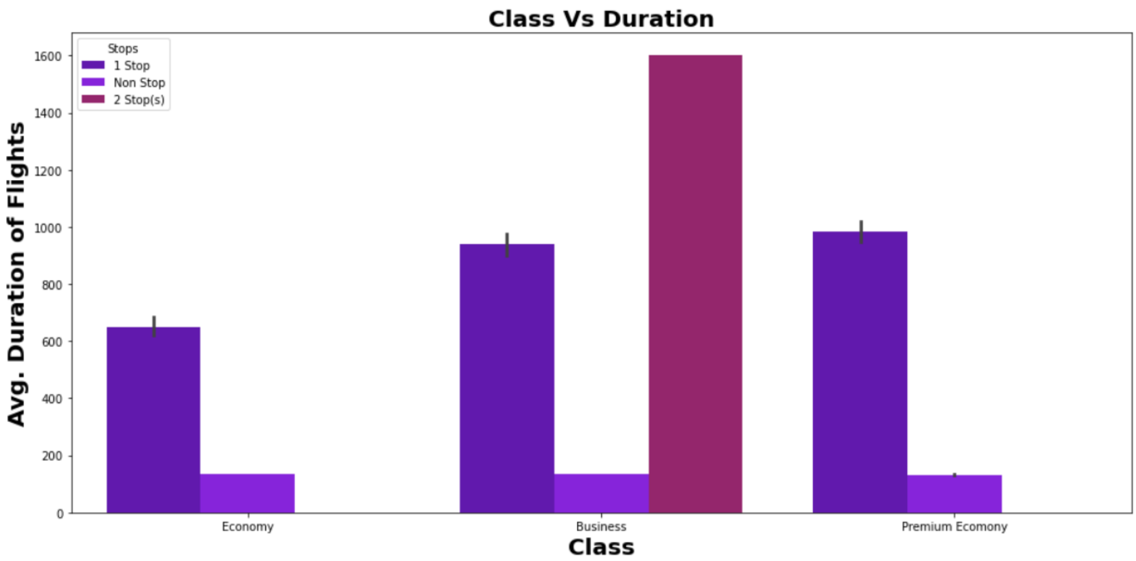
Observation:

- On Friday Maximum flights run while on Wednesday minimum flights run.



Observation:

- Maximum Avg. Fare for Business Flights is on Thursday while minimum Avg. Fare for Business flights on Saturday.
- Maximum Avg. Fare for Ecomony Flights is on Sunday.
- Maximum Avg. Fare for Premium Ecomony Flights is on Friday while minimum Avg. Fare for Premium Ecomony Flights on Wednesday.



Observation:

- As Number of Stops increase the duration of flights increases.
- As per Class of flight Maximum Avg. Duration of flight is for Business class.

Observation:

- We can see those Maximum flights schedule on 15 Jan 2023 & Minimum flights schedule on 22 & 23 Jan 2023.

Chapter 4: Conclusion

4.1. Key Findings and Conclusions of the Study:

Algorithm	R2 Score	CV Score
Random Forest Regressor	91.28	0.28
XGB Regressor	91.42	0.19
Linear Regression	62.22	-2.54
Decision Tree Regressor	84.49	-0.009
Extra Tree Regressor	84.31	0.078
XGB Hyper Parameter Tuned (Final Model)	92.05	0.83

- XGB Regressor giving us maximum R2 Score, so XGB Regressor is selected as best model.
- After hyper parameter tuning Final Model is giving us R2 Score of 99.046% which is slightly improved compare to earlier R2 score of 99.013%.

4.2. Limitations of this work and Scope for Future Work

- In this study we focus on flights on route of New Delhi to Mumbai, more route can incorporate in this project to extend it beyond present investigation.
- This investigation focuses on short timeframe (14 days prior flights take off) which can be extended variation over larger period.
- Time series analysis can be performed over this model.