NAME OF THE PROJECT

FLIGHT PRICE PREDICATION USING MACHINE LEARNING TECHNIQUES

SUBMITTED BY:

MRS. SWATI AMIT MOTUGADE

FLIPROBO SME:

GULSHANA CHAUDHARI

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. And I want to express my huge gratitude to Ms. Gulshana Chaudhari (SME Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project.

A huge thanks to “Data trained” who are the reason behind my Internship at Fliprobo. Last but not least my parents who have been my backbone in every step of my life.

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. 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.

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

9. 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.

10. 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

11. William Groves and Maria Gini "An agent for optimizing airline ticket purchasing" in proceedings of the 2013 international conference on autonomous agents and multi-agent systems.

12. J. Santos Dominguez-Menchero, Javier Rivera and Emilio Torres Manzanera "Optimal purchase timing in the airline market".

13. Supriya Rajankar, Neha sakhrakar and Omprakash rajankar “Flight fare prediction using machine learning algorithms” International journal of Engineering Research and Technology (IJERT) June 2019.

14. Abdella, Juhar Ahmed, et al. "Airline ticket price and demand prediction: A survey." Journal of King Saud University-Computer and Information Sciences 33.4 (2021): 375-391.

Mrs. Swati Amit Motugade

CHAPTER 1

INTRODUCTION

1.1 BUSINESS FRAMING PROBLEM

Currently, everyone loves to travel by flights. Going along with the study, the charge of travelling through a plane change now and then which also includes the day and night time. Additionally, it changes with special times of the year or celebration seasons. There are a few unique elements upon which the cost of air transport depends. The salesperson has data regarding each of the variables, however, buyers can get confined information which is not sufficient to foresee the airfare costs. Considering the provisions, for example, time of the day, the number of days remaining and the time of take-off this will provide the perfect time to purchase the plane ticket. The motivation behind this project is to concentrate on every component that impacts the variations in the costs of this means of transport and how these are connected with the diversity in the airfare. Subsequently, at that point, utilizing this data, construct a framework that can help purchasers when to purchase a ticket. Machine Learning algorithms prove to be the best solution for the above-discussed problems. In this project, there is an implementation of LR (Linear Regression), DT (Decision Tree), RF (Random Forest), Extra Tree Regressor and XGB Regressor algorithms.

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 endeavour 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 traveller, 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.

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.

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 programme. 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 20 Jan 2023. Data is scrape for flights on route of Mumbai to Bangalore. 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 20 Jan 2023 using selenium and saved in CSV file. Data is scrape for flights on route of Mumbai to Bangalore. Data is scraped for Economy class, Premium Economy class & Business class flights. Around 2200 flights details are collected for this project.

load the dataset

```
In [2]: data= pd.read_excel('FlightDetails_Dataset.xlsx')
```

check for shape of dataset

```
In [3]: print('No. of Rows :',data.shape[0])
        print('No. of Columns :',data.shape[1])
        pd.set_option('display.max_columns',None)
        data.head()

        No. of Rows : 2222
        No. of Columns : 12
```

Unnecessary column of index name as ‘Unnamed: 0’ is dropped out. There are 11 features in dataset including target feature ‘Price’. The data types of different features are as shown below:

```
# drop the unnecessary column 'Unnamed: 0'
data.drop(columns='Unnamed: 0', inplace =True)
```

The columns of dataset are grouped according to their datatypes by using groupby method

```
# lets sort columns by their datatype using groupby method
data.columns.to_series().groupby(data.dtypes).groups
```

And the output shows that only the target variable is numerical and all the other variables are of object type.

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 –

```
data.isnull().sum().sum()
```

```
0
```

```
data.duplicated().sum().sum()
```

```
0
```

```
data.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,

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

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

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

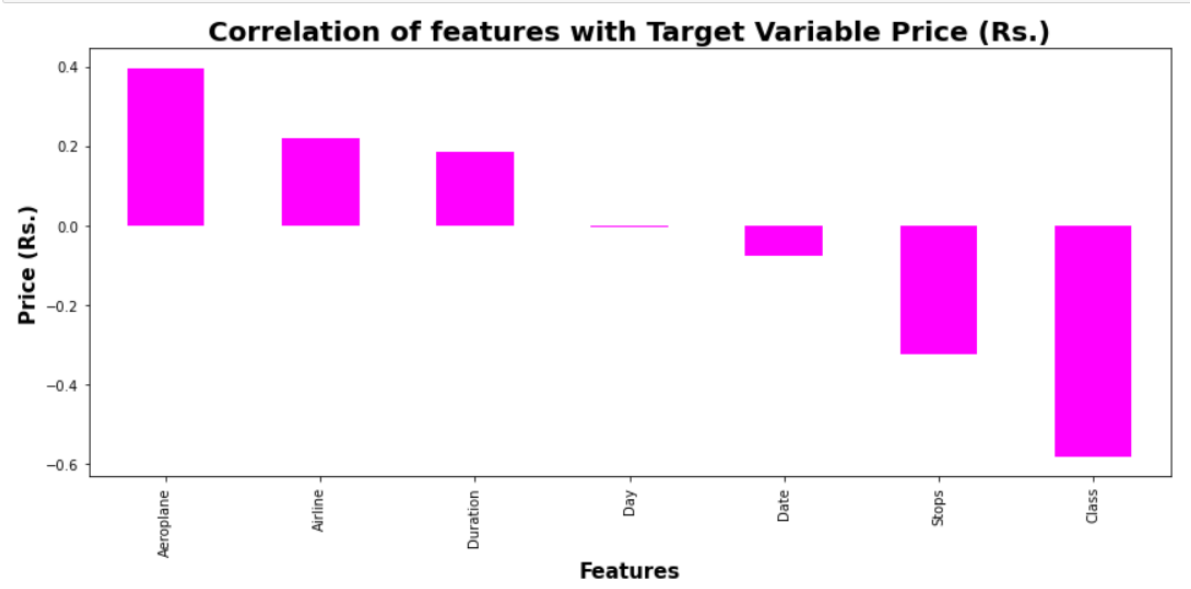
• Create new column for day & date –

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

```
#first first 3 characters are showing day so here we will take only those
data['Day']= data['Date'].map(lambda x :x[:3])
```

```
#the remaining characters are showing date of the flifgt so we will select only these characters now
data['Date']= data['Date'].map(lambda x :x[4:])
```

4.Data Inputs- Logic- Output Relationships



Correlation heatmap is plotted to gain understanding of relationship between target features & independent features. We can see that feature 'Class' is correlated for more than -0.6 with target variable

Price. Remaining features are poorly correlated with target variable price.

5.HARDWARE & SOFTWARE REQUIREMENTS WITH TOOL USED

Hardware Used –

1. Processor — 11th Gen Intel ® Core™ i5-1155G7 @ 2.50GHz 2.50GHz

2. RAM — 8 GB

Software utilised –

1.Anaconda – Jupyter Notebook

2. Selenium – Web scraping

Libraries Used – General library for data wrangling & visualisation

```
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 yatra.com website are

```
# here we will use selenium web driver to scrape the flight details hence importing required librari
import pandas as pd
import numpy as np
import time
import selenium
from selenium import webdriver
from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC

import warnings
warnings.filterwarnings('ignore')
```

Libraries used for machine learning model building

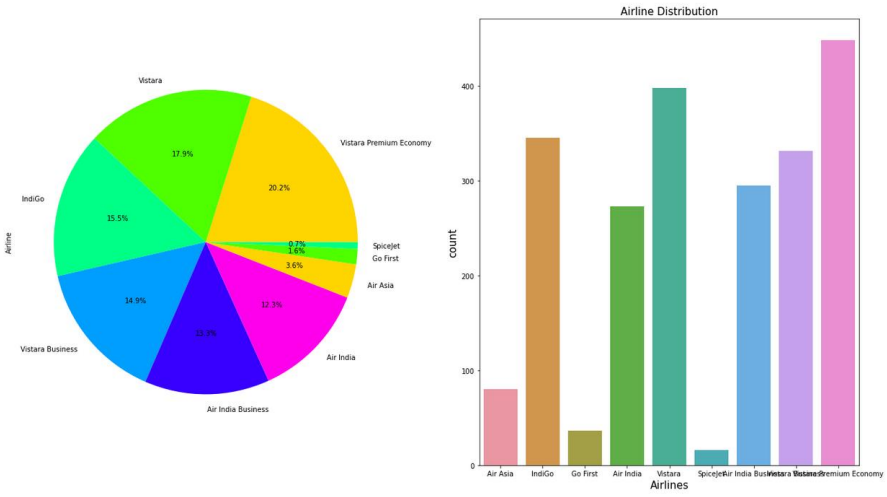
```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import ExtraTreesRegressor
from xgboost import XGBRegressor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

CHAPTER 3

EDA

3.1 AIRLINE WISE DISTRIBUTION

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

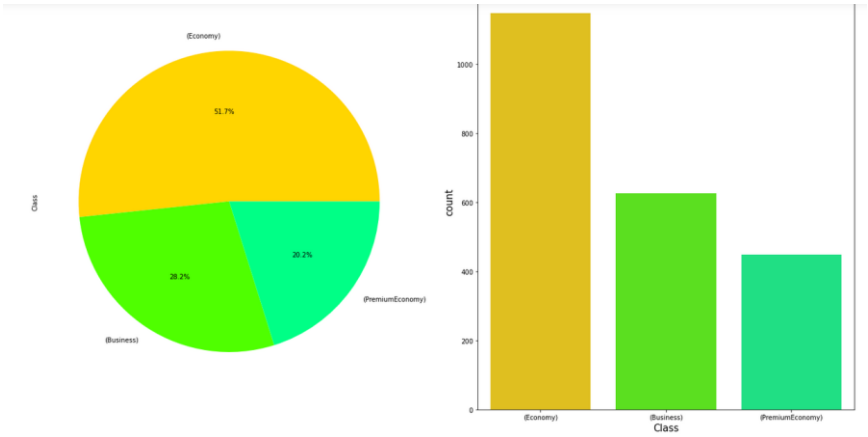


Observation:

1.We can see maximum number of flights are of Business Class. Maxi mum number of flights run by Vistara Premium Economy while minimum Flights run by SpiceJet.

2. Around 28% of flights of Business Class.

3.2 CLASS WISE DISTRIBUTION

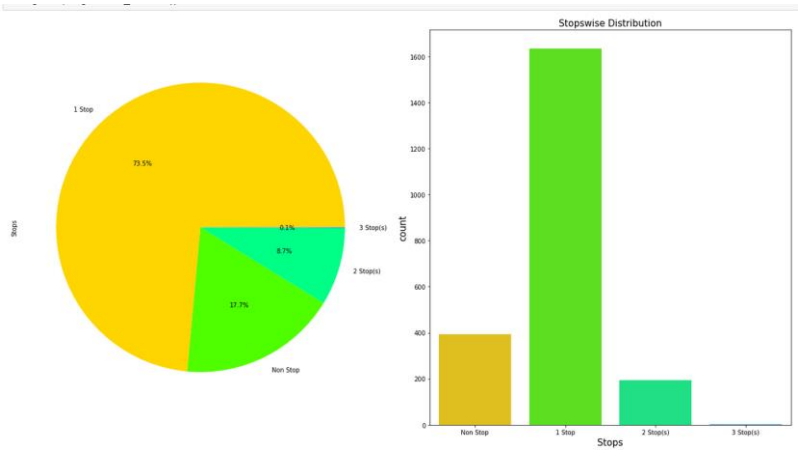


Observation:

1.51.7% flights are of Economy class, as they are low cost of flight & most of people prefer it.

2.There are more business class flights than Premium Economy flights. It strange because Business class is costlier than Premium Economy class.

3.3 STOP WISE DISTRIBUTION

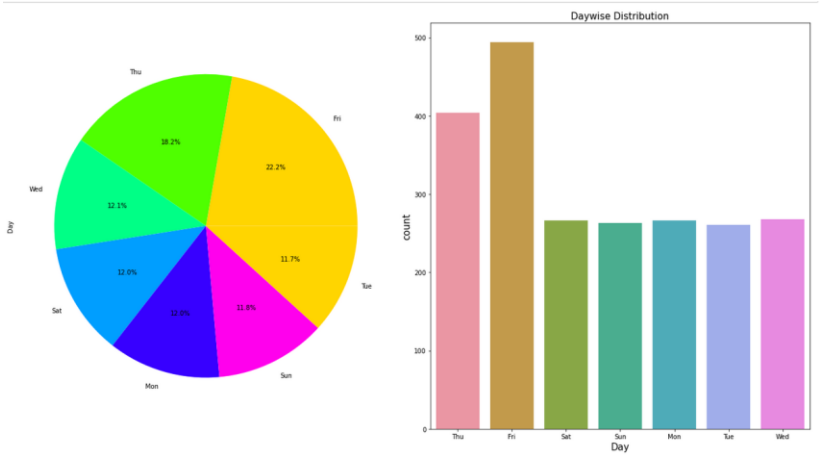


Observation:

1. 73.5% flights take single stop in there way from Mumbai to Bangalore. It is also possible that these flights may have high flight duration compare to Non-stop Flight

2. 17.7% of flights do not have any stop in their route.

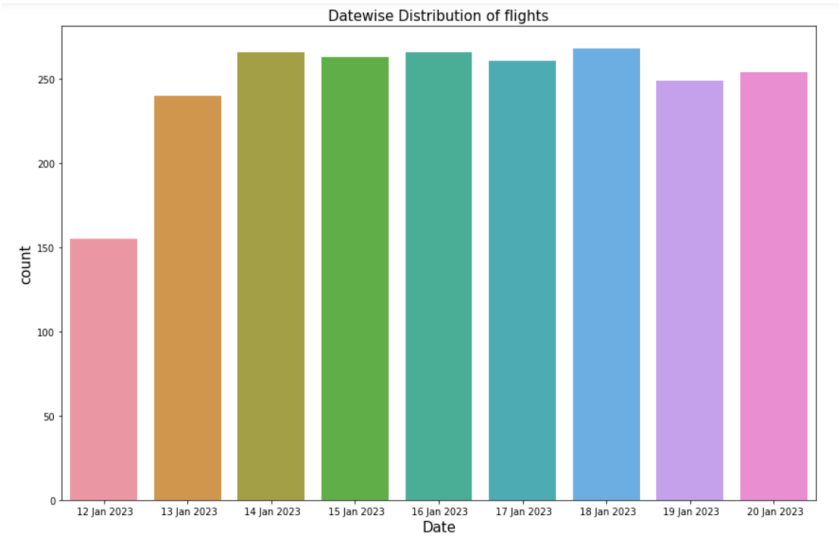
3.4 DAY WISE DISTRIBUTION



Observation:

1.Maximum number flights runs on Friday where minimum number of flights runs on Tuesday.

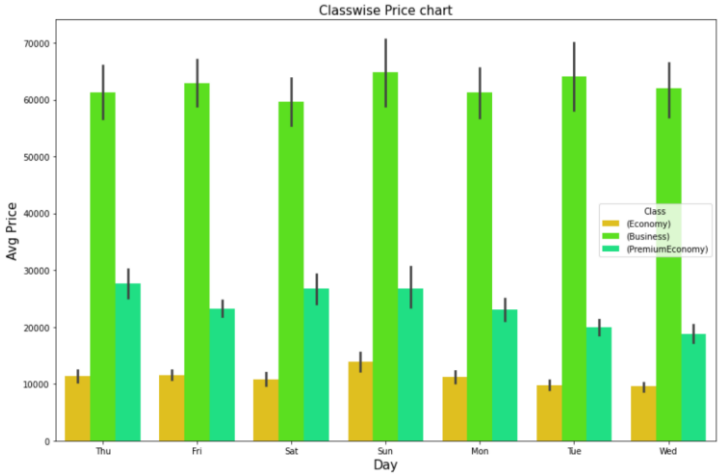
3.5 DATE WISE DISTRIBUTION



Observation:

1. Maximum number of flights are available on 18 Jan 2023 and minimum flights are available on 12 Jan 2023.

3.6 DAY VS PRICE



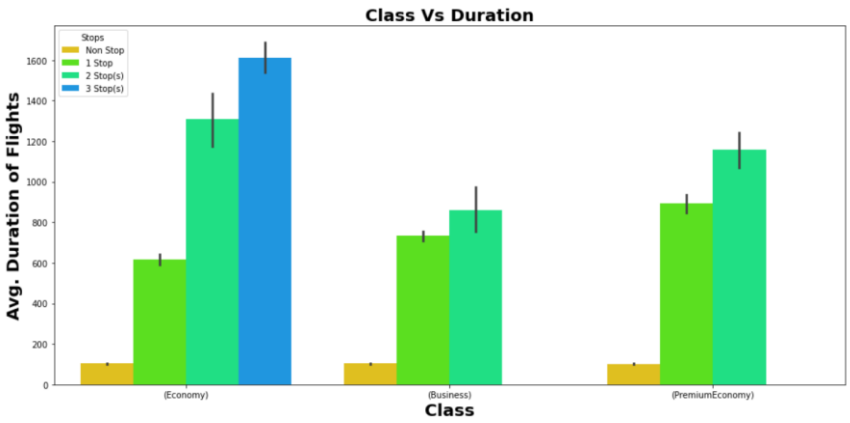
Observation:

1. Maximum average price for Business class is for Sunday and minimum for Saturday.

2. Maximum average price for Economy class is for Sunday and minimum for Tuesday.

3. Maximum average price for Premium Economy class is for Thursday and minimum for Wednesday.

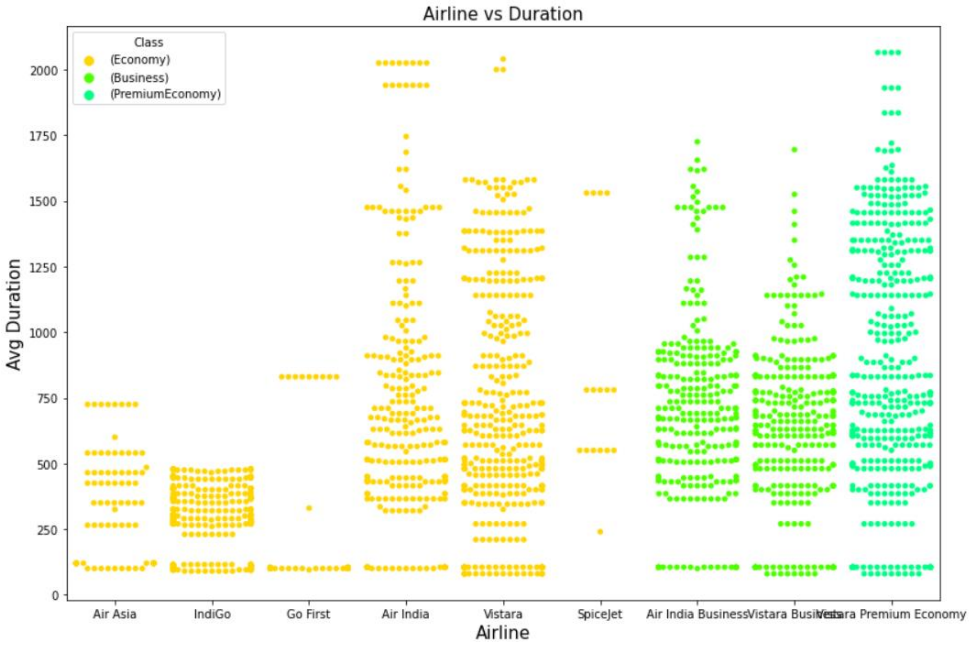
3.7 CLASS VS DURATION



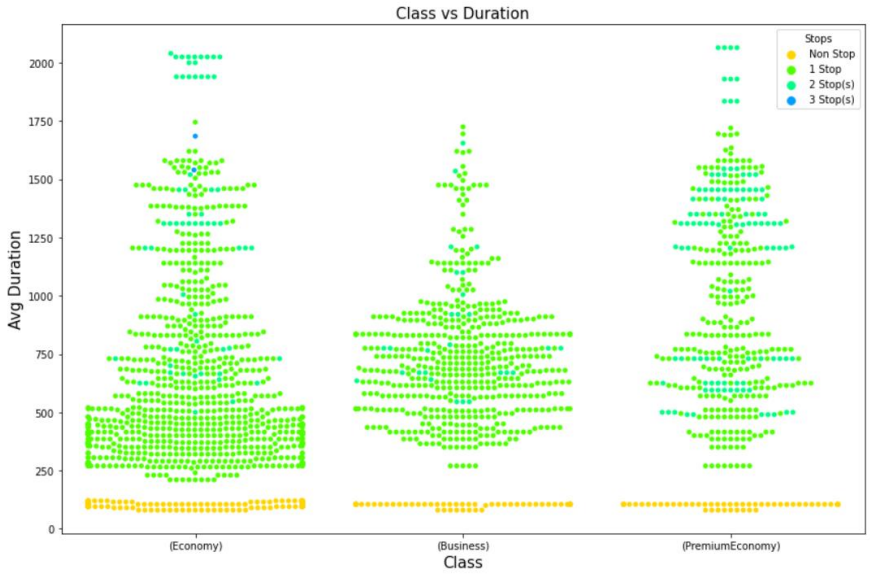
Observation:

As the number of stops increases the duration also increases.

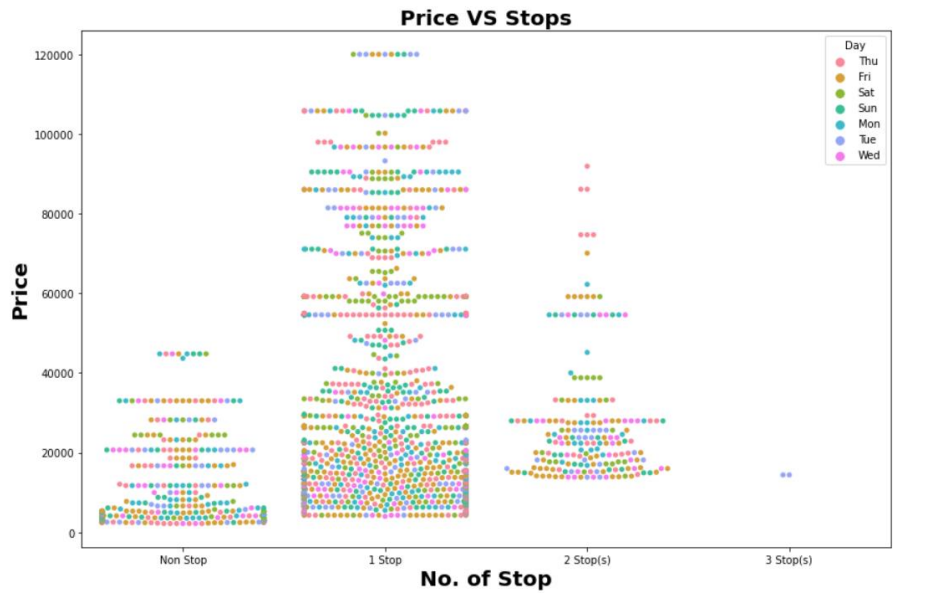
3.8 AIRLINE VS DURATION



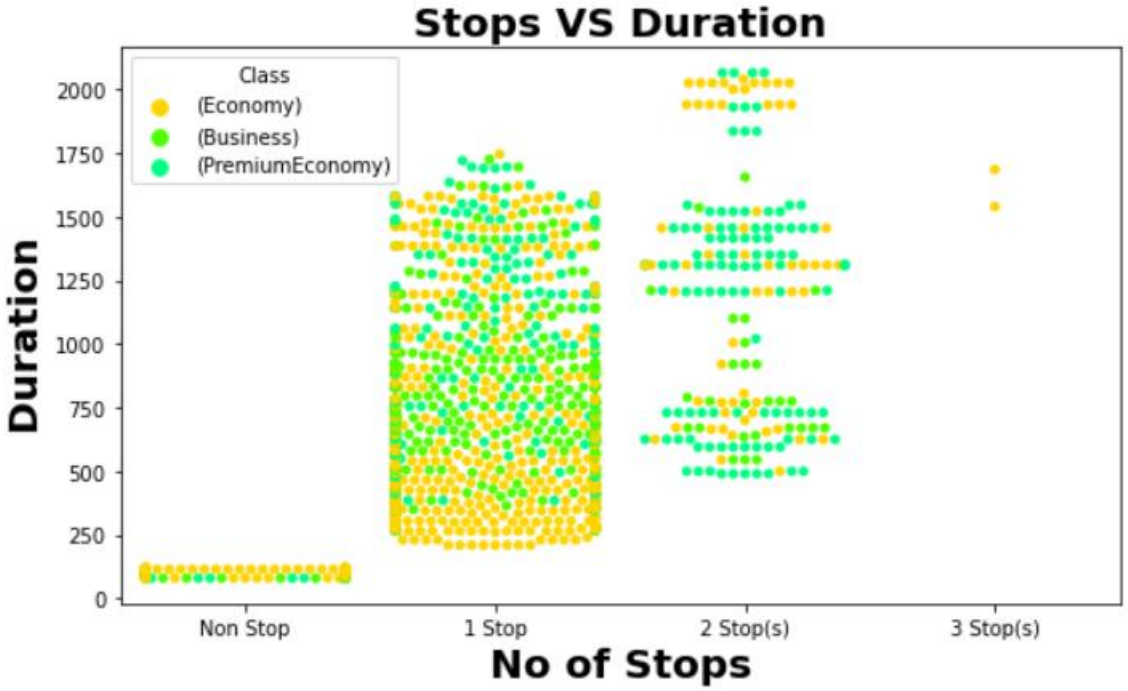
3.9 CLASS VS DURATION



3.10 PRICE VS STOPS



3.11 STOPS VS DURATION



CHAPTER 4

MODEL DEVELOPMENT AND EVALUATION

4.1 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.

4.2 TESTING OF IDENTIFIED APPROACHES (ALGORITHMS)

Phase 1 Web Scraping Strategy employed in this project as follow:

1. Selenium will be used for web scraping data from www.yatra.com

2. Flights on route of New Delhi to Mumbai in duration of 23 Jan 2022 to 4 Feb 2022.

3. Data is scrap in three parts:

1. ▪ Economy class flight price extraction

2. ▪ Economy class flight price extraction
3. ▪ Premium Economy class price extraction

4. Premium Economy class price extraction

5. In next part web scraping code executed for above mention details.

6. Exporting final data in Excel file.

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

❖ Linear Regression

❖ Random Forest Regressor

❖ Decision Tree Regressor

❖ XGB Regressor

❖ Extra Tree Regressor

4.3 KEY METRICS FOR SUCCESS IN SOLVING PROBLEM UNDER CONSIDERATION

Following metrics used for evaluation:

1. Mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.

2. Root mean square error is one of the most commonly used measures for evaluating the quality of predictions.

3. R2 score which tells us how accurate our model predict result, is going to important evaluation criteria along with Cross validation score.

4.4 RUN AND EVALUATE SELECTED MODELS

1. Linear Regression:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 366, test_size=0.3)
lr= LinearRegression()
lr.fit(X_train, Y_train)
y_pred_lr = lr.predict(X_test)
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred_lr))
print('Mean squared error :', mean_squared_error(Y_test, y_pred_lr))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred_lr)))
print('R2 Score_lr :',r2_score(Y_test,y_pred_lr)*100)
print('CV Score_lr :',cross_val_score(lr, X_scale, Y, cv=10).mean())
```

```
Mean absolute error : 11316.880252988574
Mean squared error : 202319273.08602673
Root Mean squared error : 14223.89795681995
R2 Score_lr : 70.79928855339242
CV Score_lr : -4.459568312889918
```

2. Extra Trees Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 366, test_size=0.3)
ext = ExtraTreesRegressor()
ext.fit(X_train, Y_train)
y_pred_ext = ext.predict(X_test)
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred_ext))
print('Mean squared error :', mean_squared_error(Y_test, y_pred_ext))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred_ext)))
print('R2 Score_ext :',r2_score(Y_test,y_pred_ext)*100)
print('CV Score_ext :',cross_val_score(ext, X_scale, Y, cv=10).mean())
```

```
Mean absolute error : 3970.71107946027
Mean squared error : 48079349.04292368
Root Mean squared error : 6933.927389504715
R2 Score_ext : 93.06071450075753
CV Score_ext : 0.20531465598490817
```

3. Random Forest Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 366, test_size=0.3)
rf = RandomForestRegressor()
rf.fit(X_train, Y_train)
y_pred_rf = rf.predict(X_test)
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred_rf))
print('Mean squared error :', mean_squared_error(Y_test, y_pred_rf))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred_rf)))
print('R2 Score_rf :',r2_score(Y_test,y_pred_rf)*100)
print('CV Score_rf :',cross_val_score(rf, X_scale, Y, cv=10).mean())
```

```
Mean absolute error : 3723.8526836581714
Mean squared error : 41291618.334776916
Root Mean squared error : 6425.855455484267
R2 Score_rf : 94.04038669294454
CV Score_rf : 0.4316889418684208
```

4. Decision Tree Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 366, test_size=0.3)
dt = DecisionTreeRegressor()
dt.fit(X_train, Y_train)
y_pred_dt = dt.predict(X_test)
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred_dt))
print('Mean squared error :', mean_squared_error(Y_test, y_pred_dt))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred_dt)))
print('R2 Score_dt :',r2_score(Y_test,y_pred_dt)*100)
print('CV Score_dt :',cross_val_score(dt, X_scale, Y, cv=10).mean())
```

```
Mean absolute error : 4367.607196401799
Mean squared error : 70655354.17091455
Root Mean squared error : 8405.67392723002
R2 Score_dt : 89.80232294317574
CV Score_dt : 0.10380831327204805
```

5. XGB Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 366, test_size=0.3)
xgb = XGBRegressor()
xgb.fit(X_train, Y_train)
y_pred_xgb = xgb.predict(X_test)
print('Mean absolute error :', mean_absolute_error(Y_test,y_pred_xgb))
print('Mean squared error :', mean_squared_error(Y_test, y_pred_xgb))
print('Root Mean squared error :', np.sqrt(mean_squared_error(Y_test, y_pred_xgb)))
print('R2 Score_xgb :',r2_score(Y_test,y_pred_xgb)*100)
print('CV Score_xgb :',cross_val_score(xgb, X_scale, Y, cv=10).mean())
```

```
Mean absolute error : 3692.5184939268647
Mean squared error : 38099746.23646836
Root Mean squared error : 6172.499188859271
R2 Score_xgb : 94.50106913162429
CV Score_xgb : 0.48723279151304916
```

10-Fold cross validation performed over all models. We can see that XGB Regressor gives maximum R2 score of 94.501069 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=366, test_size=0.3)
```

```
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': 'dart',
 'eta': 0.1,
 'gamma': 0.0,
 'max_depth': 6,
 'n_estimators': 400}
```

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

```
final_model=XGBRegressor(booster='gbtree', max_depth=6, eta=0.1,
                         gamma=0.1, n_estimators=400)
final_model.fit(X_train,Y_train)
pred=final_model.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: 94.09917724102289
mean_squared_error: 40884283.70619134
mean_absolute_error: 3867.83636758281
RMSE value: 6394.081928329612
```

Final model is saved using joblib library.

```
# Saving the model using .pkl
import joblib
joblib.dump(final_model,"Flight_Price_Prediction.pkl")
```

```
['Flight_Price_Prediction.pkl']
```

CHAPTER 5.

CONCLUSION

5.1 RESULT

The R2 Scores and CV scores of the algorithms implemented in this project are as below in the table

ALGORITHM	R2 SCORE	CV SCORE
Linear Regression	70.79928	-4.45956
Extra Trees Regressor	93.06071	0.20531
Random Forest Regressor	94.04038	0.43168
Decision Tree Regressor	89.80232	0.10380
XGB Regressor	94.50106	0.48723
Final Model (XGB Hyperparameter Tuned)	94.09917	

5.2 CONCLUSION

➢ 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 94.09917% which is slightly decreased compare to earlier R2 score of 94.50106%.

5.3. LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE

➢ 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 focus on short timeframe (8 days prior flights take off) which can be extended variation over larger period.

➢ Time series analysis can be performed over this model.