FLIP ROBO

Project Report On

“Flight Price Prediction”

Submitted by:

Rahul Singh

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Flight Price Prediction”

Project also huge thanks to my Academic team “DataTrained”. Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

References:

I have also used few external resources that helped me to complete this project successfully.

Below are the external resources that were used to create this project.

1. https://www.google.com/
2. https://scikit-learn.org/stable/index.html
3. https://github.com/
4. https://www.analyticsvidhya.com/
5. www.researchgate.net/

INTRODUCTION

- Business Problem Framing

Airline industry is one of the most sophisticated in its use of dynamic pricing strategies to maximize revenue, based on proprietary algorithms and hidden variables. That is why the airline companies use complex algorithms to calculate the flight ticket prices. There are several different factors on which the price of the flight ticket depends. The seller has information about all the factors, but buyers are able to access limited information only which is not enough to predict the airfare prices. Considering the features such as departure time, arrival time and time of the day it will give the best time to buy the ticket.

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.

Business goal: The main aim of this project is to predict the price of flight tickets based on various features. The purpose of the paper is to study the factors which influence the fluctuations in the airfare prices and how they are related to the change in the prices. Then using this information, build a system that can help buyers whether to buy a ticket or not. So, we will deploy an Machine Learning

- Conceptual Background of the Domain Problem
- Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable.
- 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).
- Here we are trying to help the buyers to understand the price of the flight tickets by deploying machine learning models. These models would help the sellers/buyers to understand the flight ticket prices in market and accordingly they would be able to book their tickets.

- Review of Literature

Literature review covers relevant literature with the aim of gaining insight into the factors that are important to predict the flight ticket prices in the market. In this study, we discuss various applications and methods which inspired us to build our supervised ML techniques to predict the price of flight tickets in different locations. We did a background survey regarding the basic ideas of our project and used those ideas for the collection of data information by doing web scraping from www.yatra.com **website which is a** web platform where buyers can book their flight tickets.

This project is more about data exploration, feature engineering and pre-processing that can be done on this data. Since we scrape huge amount of data that includes more flight related features, we can do better data exploration and derive some interesting features using the available columns. Different techniques like ensemble techniques, and decision trees have been used to make the predictions.

The goal of this project is to build an application which can predict the price of flight tickets with the help of other features. In the long term, this would allow people to better explain and reviewing their purchase in this increasing digital world.

- Motivation for the Problem Undertaken

  Air travel is the fastest method of transport around, and can cut hours or days off of a trip. But we know how unexpectedly the prices vary. So, I was interested in Flight Fares Prediction listings to help individuals and find the right fares based on their needs. And also, to get hands on experience and to know that how the data scientist approaches and work in an industry end to end.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

We need to develop an efficient and effective Machine Learning model which predicts the price of flight tickets. So, “Price” is our target variable which is continuous in nature. Clearly it is a Regression problem where we need to use regression algorithms to predict the results. This project is done on three phases:

- **Data Collection Phase: I have done web scraping to collect the data of flights from the well-known website** www.yatra.com **where I found more features of flights compared to other websites and I fetch data for different locations. As per the requirement we need to build the model to predict the prices of flight tickets.**
- **Data Analysis:** After cleaning the data I have done some analysis on the data by using different types of visualizations.
- **Model Building Phase:** After collecting the data, I built a machine learning model. Before model

building, have done all data pre-processing steps. The complete life cycle of data science that I have used in this project are as follows:

- Data Cleaning
- Exploratory Data Analysis
- Data Pre-processing
- Model Building
- Model Evaluation
- Selecting the best model

- Data Sources and their formats
- We have collected the dataset from the website www.yatra.com **which is a web platform where** the people can purchase/book their flight tickets. The data is scraped using Web scraping technique and the framework used is Selenium. We scrapped nearly 5303 of the data and fetched the data for different locations and collected the information of different features of the flights and saved the collected data in excel format. The dimension of the dataset is 5303 rows and 9 columns including target variable "Price". The particular dataset contains both categorical and

numerical data type. The data description is as follows:

Variables	Definition
Airline	The Name of airline
Departure_time	The time when the journey starts from the source
Time_of_arrival	Time of arrival at the destination
Duration	Total duration taken by the flight to reach the destination from the source
Source	The source from which the service begins
Destination	The destination where the service ends
Meal_availability	Availability of meals in the flight
Number_of_stops	Total stops between the source and destination
Price	The price of the flight ticket

- Data Preprocessing Done

Data pre-processing is the process of converting raw data into a well-readable format to be used by

Machine Learning model. Data pre-processing isan integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model. I have used following pre-processing steps:

- Importing necessary libraries and loading collected dataset as a data frame.
- Checked some statistical information like shape, number of unique values present, info, unique (), data types, value count function etc.
- Checked null values and found no missing values in the dataset.
- Taking care of Timestamp variables by converting data types of “Departure_time” and “Time_of_arrival” from object data type into datetime data types.
- Done feature engineering on some features as they had some irrelevant values like “,”, “:” and replaced them by empty space.
- The column Duration had values in terms of minutes and hours. Duration means the time taken by the plane to reach the destination and it is the difference between the arrival time

and Departure time. So, I have extracted proper duration time in terms of float data type from arrival and departure time columns.

- Extracted Departure_Hour, Deparutre_Min and Arrival_Hour, Arrival_Min columns from Departure_time and Time_of_arrival columns and dropped these columns after extraction.
- The target variable "price" should be continuous numeric data but due to some string values like “,” it was showing as object data type. So, I replaced this sign by empty space and converted into float data type.
- From the value count function of Meal_availability we observed "eCash 250" entry which does not belong to meals so I have replaced it as "None" and grouped same categories.
- From the value count function of Number_of_stops I found categorical data so replaced them with numeric data according to stops.
- Checked statistical description of the data and separated categorical and numeric features.
- Performed univariate, bivariate and multivariate analysis to visualize the data.

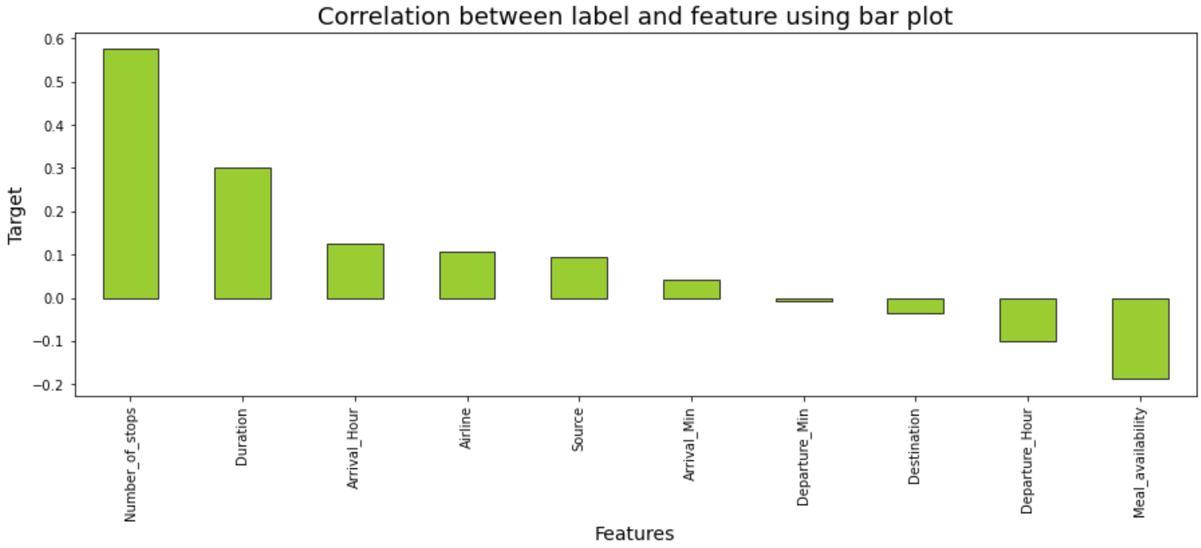
Visualized each feature using seaborn and matplotlib libraries by plotting several categorical and numerical plots like pie plot, count plot, bar plot, reg plot, strip plot, line plot, box plot, boxen plot, distribution plot, and pair plot.

- Identified outliers using box plots and found no outliers.
- Checked for skewness and removed skewness in numerical column “Duration” using square root transformation method.
- Encoded the columns having object data type using Label Encoder method. Used Pearson’s correlation coefficient to check the correlation between label and features. With the help of heatmap and correlation bar graph was able to understand the Feature vs Label relativity.
- Separated feature and label data and feature scaling is performed using Standard Scaler method to avoid any kind of data biasness.

- Data Inputs- Logic- Output Relationships

The dataset consists of label and features. The features are independent and label is dependent as the values of our independent variables changes as our label varies.

- Since we had both numerical and categorical columns, I checked the distribution of skewness using dist plots for numerical features and checked the counts using count plots & pie plots for categorical features as a part of univariate analysis.
- To analyse the relation between features and label I have used many plotting techniques where I found numerical continuous variables having some relation with label Price with the help of categorical and line plot.
- I have checked the correlation between the label and features using heat map and bar plot. Where I got both positive and negative correlation between the label and features. Below is the bar graph to know the correlation between features and label.



- Hardware and Software Requirements and Tools Used

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

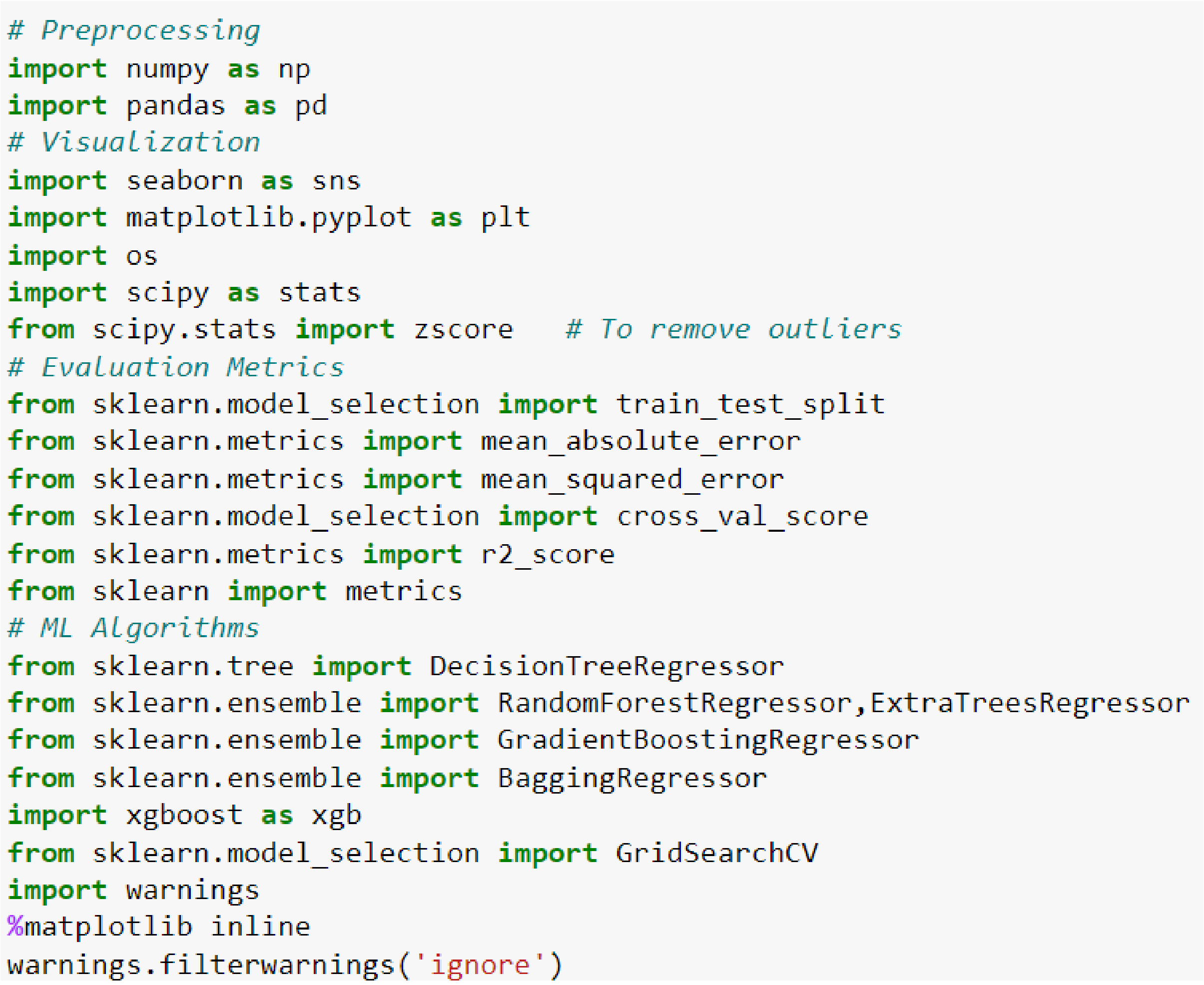
Hardware required:

- Processor: core i5 or above
- RAM: 8 GB or above
- ROM/SSD: 250 GB or above

Software required:

- Distribution: Anaconda Navigator
- Programming language: Python

- Browser based language shell: Jupyter Notebook
- Chrome: To scrape the data



```
# Preprocessing
import numpy as np
import pandas as pd
# Visualization
import seaborn as sns
import matplotlib.pyplot as plt
import os
import scipy as stats
from scipy.stats import zscore   # To remove outliers
# Evaluation Metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import cross_val_score
from sklearn.metrics import r2_score
from sklearn import metrics
# ML Algorithms
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
import xgboost as xgb
from sklearn.model_selection import GridSearchCV
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

✓**import numpy as np:** It is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single dimensional array elements. The calculations using Numpy arrays are faster than the normal Python array.

- ✓**import pandas as pd:** Pandas is a Python library that is used for faster data analysis, data cleaning and data pre-processing. The data-frame term is coming from Pandas only.
- ✓**import matplotlib.pyplot as plt:** Matplotlib and Seaborn acts as the backbone of data visualization through Python.

  Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays.
- ✓**import seaborn as sns: Seaborn** is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data.

With the above sufficient libraries, we can perform pre-processing, data cleaning and can build ML models.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
- I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data also used EDA techniques and heat map to check the correlation of independent and dependent features. Removed skewness using square root transformation. Encoded data using Label Encoder. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details. Finally created multiple

regression models along with evaluation metrics.

- For this particular project we need to predict flight ticket prices. In this dataset, “Price” is the target variable, which means our target column is continuous in nature so this is a regression problem. I have used many regression algorithms and predicted the flight ticket price. By doing various evaluations I have selected Extra Trees Regressor as best suitable algorithm to create our final model as it is giving high R2 score and low evaluation error among all the algorithms used. Performed hyper parameter tuning on best model. Then I saved my final model and loaded the same for predictions.

- Testing of Identified Approaches (Algorithms)

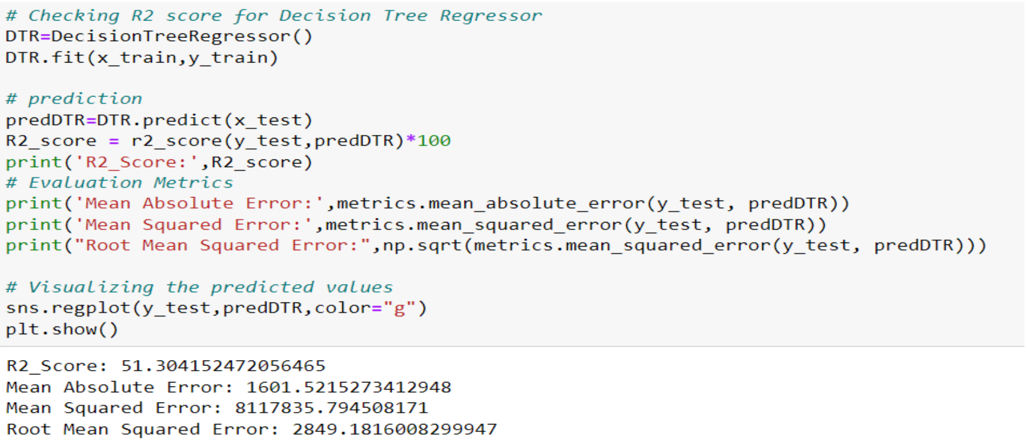
Since “Price” is my target variable which is continuous in nature, from this I can conclude that it is a regression type problem hence I have used following regression algorithms. After the pre-processing and data cleaning I left with 11 columns including target and with the help of feature importance bar graph I used these independent features for model building and prediction. The algorithms used on training the data are as follows:

1. Decision Tree Regressor
2. Random Forest Regressor
3. Extra Trees Regressor
4. Gradient Boosting Regressor
5. Extreme Gradient Boosting Regressor (XGB)
6. Bagging Regressor

- Run and Evaluate selected models

1. **Decision Tree Regressor:**

   Decision Tree Regressor is a decision-making tool that uses a flowchart like tree structure. It



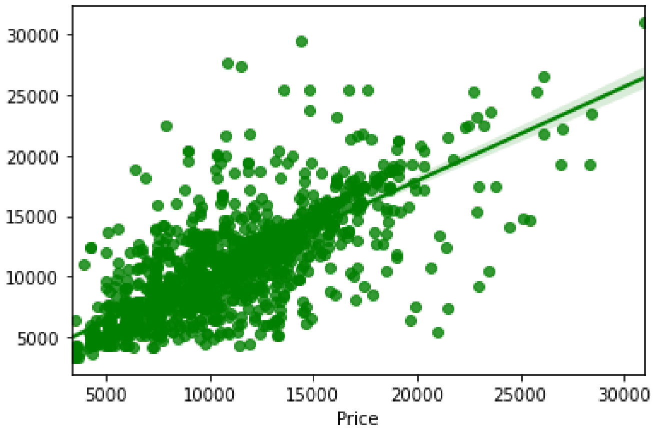
```
# Checking R2 score for Decision Tree Regressor
DTR=DecisionTreeRegressor()
DTR.fit(x_train,y_train)

# prediction
predDTR=DTR.predict(x_test)
R2_score = r2_score(y_test,predDTR)*100
print('R2_Score:',R2_score)
# Evaluation Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predDTR))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predDTR))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predDTR)))

# Visualizing the predicted values
sns.regplot(y_test,predDTR,color="g")
plt.show()
```

```
R2_Score: 51.304152472056465
Mean Absolute Error: 1601.5215273412948
Mean Squared Error: 8117835.794508171
Root Mean Squared Error: 2849.1816008299947
```

observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output.



❖ Created Decision Tree Regressor model and checked for its evaluation metrics. The model is giving R2 score as 51.30%.

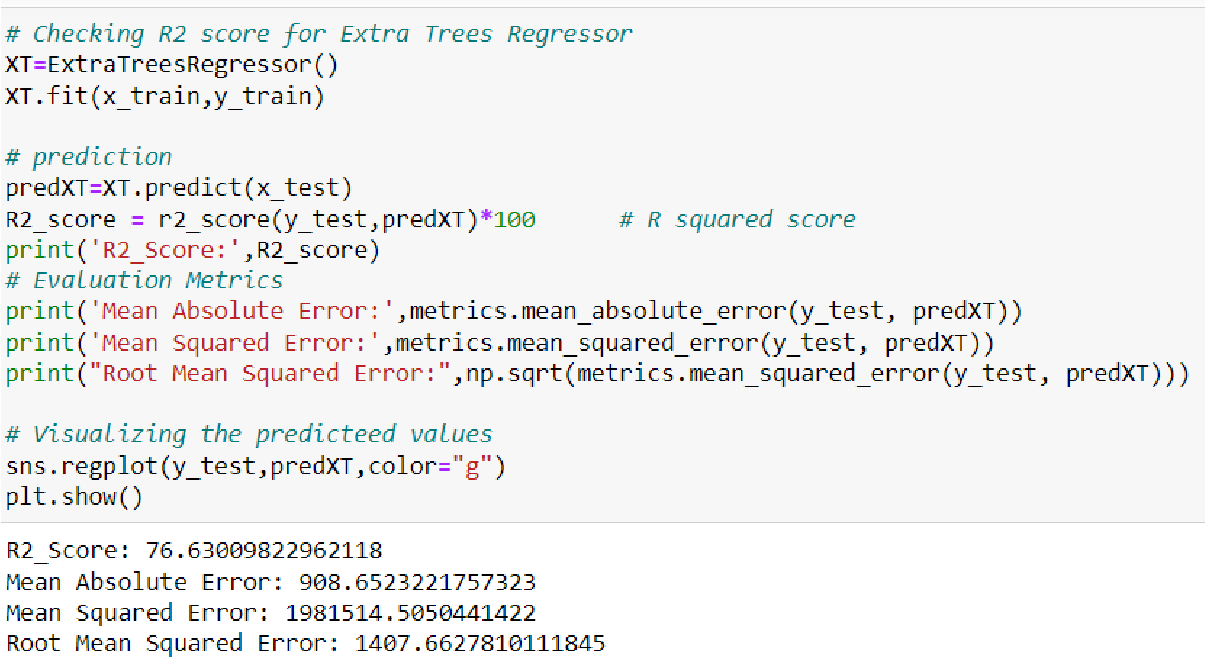
❖ From the graph we can observe how our model is mapping. In the graph we can observe the straight line which is our actual dataset and dots are the predictions that the model has given.

2. Random Forest Regressor:

Random forest is an ensemble technique capable of performing both regression and classification tasks with use of multiple decision trees and a technique called Bootstrap Aggregation. It improves the predictive accuracy and control over-fitting.

3.Extra Trees Regressor:

The **Extra Trees** implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

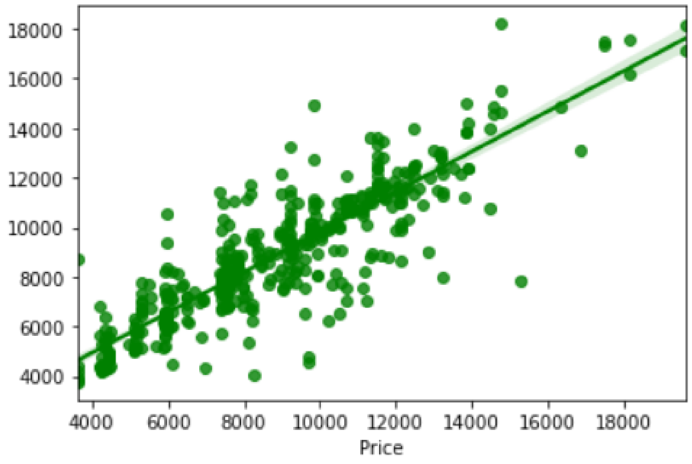


```
# Checking R2 score for Extra Trees Regressor
XT=ExtraTreesRegressor()
XT.fit(x_train,y_train)

# prediction
predXT=XT.predict(x_test)
R2_score = r2_score(y_test,predXT)*100      # R squared score
print('R2_Score:',R2_score)
# Evaluation Metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, predXT))
print('Mean Squared Error:',metrics.mean_squared_error(y_test, predXT))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test, predXT)))

# Visualizing the predicteed values
sns.regplot(y_test,predXT,color="g")
plt.show()
```

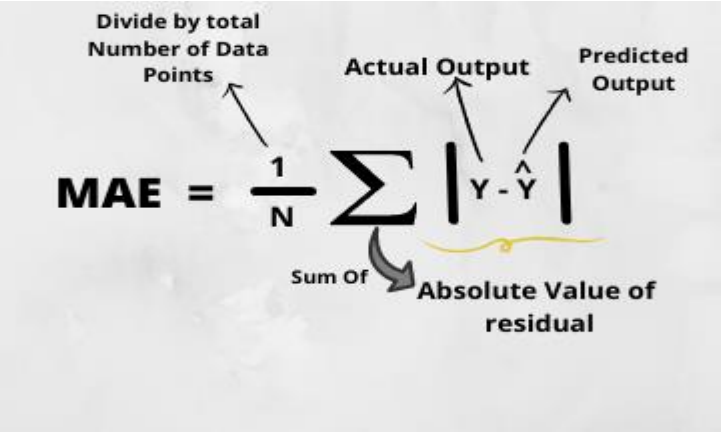
```
R2_Score: 76.63009822962118
Mean Absolute Error: 908.6523221757323
Mean Squared Error: 1981514.5050441422
Root Mean Squared Error: 1407.6627810111845
```



- Key Metrics for success in solving problem under consideration

The essential step in any machine learning model is to evaluate the accuracy and determine the metrics error of the model. I have used Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R2 Score metrics for my model evaluation:

❖ **Mean Absolute Error (MAE):** MAE is a popular error metric for regression problems which gives magnitude of absolute difference between actual and predicted values. The MAE can be calculated as follows:


Divide by total Number of Data Points
Actual Output
Predicted Output
MAE = 1/N ∑ |Y - Ŷ|
Sum Of
Absolute Value of residual

- **Mean Squared Error (MSE):** MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value. We perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

- **Root Mean Squared Error (RMSE):** RMSE is an extension of the mean squared error. The square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.

$$\mathbf{RMSE} = \sqrt{\mathbf{MSE}}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)}$$

- Visualizations

  I used pandas profiling to get the over viewed visualization on the pre-processed data. Pandas is an open-source Python module with which we can do an exploratory data analysis to get detailed description of the features and it helps in visualizing and understanding the distribution of each variable. I have analysed the data using univariate, bivariate and multivariate analysis. In univariate analysis I have used distribution plot, pie plot and count plot and in bivariate analysis I have used bar plots, strip plots, box plots and boxen plots to get the relation between categorical variables and target column Price and used line plots, reg plot, box plot, bar plot, boxen plot and factor plot to understand the relation between continuous numerical variables and target variable. Apart from these plots I have used pair plot (multivariate analysis) and box plots to get the insight from the features.

- **Univariate Analysis:** Univariate analysis is the simplest way to analyse data. “Uni” means one and this means that the data has only one kind of variable. The major reason for univariate analysis is to use the data to describe. The analysis will take data, summarise it, and then find some pattern in the data. Mainly we will get the counts of the values present in the features.
- Interpretation of the Results
- **Visualizations:** In univariate analysis I have used count plots and pie plots to visualize the counts in categorical variables and distribution plot to visualize the numerical variables. In

bivariate analysis I have used bar plots, strip plots, line plots, reg plots, box plots, and boxen plots to check the relation between label and the features. Used pair plot to check the pairwise relation between the features. The heat map and bar plot helped me to understand the correlation between dependent and independent features. Detected outliers and skewness with the help of box plots and distribution plots respectively. And I found some of the features skewed to right as well as to left. I got to know the count of each column using bar plots.

- **Pre-processing:** The dataset should be cleaned and scaled to build the ML models to get good predictions. I have performed few processing steps which I have already mentioned in the pre-processing steps where all the important features are present in the dataset and ready for model building.
- **Model building:** After cleaning and processing data, I performed train test split to build the model. I have built multiple regression models to get the accurate R2 score, and evaluation

metrics like MAE, MSE and RMSE. I got Extra Trees Regressor as the best model which gives 77% R2 score. After tuning the best model, the R2 score of Extra Trees Regressor has been increased to 77% and also got low evaluation metrics. Finally, I saved my final model and got the good predictions results for price of flight tickets.

CONCLUSION

- Key Findings and Conclusions of the Study
- The case study aims to give an idea of applying Machine Learning algorithms to predict the price of the flight tickets. After the completion of this project, we got an insight of how to collect data, pre-processing the data, analyse the data, cleaning the data and building a model.
- In this study, we have used multiple machine learning models to predict the flight ticket price. We have gone through the data analysis by performing feature engineering,

finding the relation between features and label through visualizations. And got the important feature and we used these features to predict the car price by building ML models. Performed hyper parameter tuning on the best model and the best model's R2 score increased and was giving R2 score as 77%. We have also got good prediction results of ticket price.

- Learning Outcomes of the Study in respect of Data Science
- While working on this project I learned many things about the features of flights and about the flight ticket selling web platforms and got the idea that how the machine learning models have helped to predict the price of flight tickets. I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features. This graphical representation helped me to understand which features are important and how these features describe price of tickets. Data cleaning was one of the important and crucial things in this project

where I dealt with features having string values, features extraction and selection. Finally got Extra Trees Regressor as best model.

- The challenges I faced while working on this project was when I was scrapping the real time data from yatra website, it took so much time to gather data. Finally, our aim was achieved by predicting the flight ticket price and built flight price evaluation model that could help the buyers to understand the future flight ticket prices.

- Limitations of this work and Scope for Future Work
- **Limitations:** The main limitation of this study is the low number of records that have been used. In the dataset our data is not properly distributed in some of the columns many of the values in the columns are having string values which I had taken care. Due to some reasons our models may not make the right patterns and the performance of the model also reduces. So that issues need to be taken care.

- **Future work:** The greatest shortcoming of this work is the shortage of data. Anyone wishing to expand upon it should seek alternative sources of historical data manually over a period of time. Additionally, a more varied set of flights should be explored, since it is entirely plausible that airlines vary their pricing strategy according to the characteristics of the flight (for example, fares for regional flights out of small airports may behave differently than the major, well flown routes we considered here). Finally, it would be interesting to compare our system's accuracy against that of the commercial systems available today (preferably over a period of time).