PROJECT REPORT ON:

“Flight Price Prediction Project”

SUBMITTED BY

SAKSHI SHUKLA

ACKNOWLEDGEMENT

Firstly, I would like to thank FlipRobo Technologies for giving me the opportunity to work on this project. Also, I would like to thank the DataTrained team, especially Deepika Sharma Ma’am for providing me the knowledge and guidance which helped me a lot to work on this project.

CONTENTS

INTRODUCTION

Business Problem Framing:

The tourism industry is changing fast and this is attracting a lot more travelers each year. The airline industry is considered one of the most sophisticated industries in using complex pricing strategies. Nowadays flight prices are quite unpredictable. The ticket prices change frequently. Customers are seeking to get the lowest price for their tickets, 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's 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 the 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 have insufficient data points to compute specific price statistics – 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.

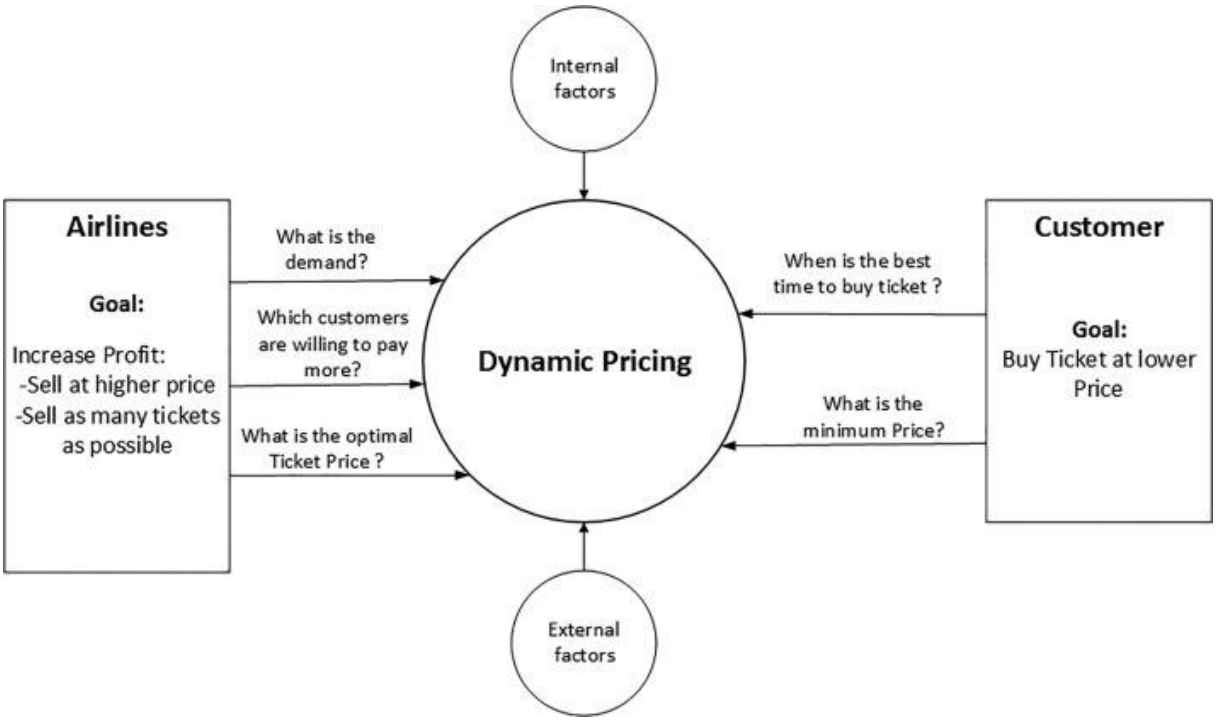
Conceptual Background of the Domain Problem

Flight prices are something unpredictable. It's more than likely that we spent hours on the internet researching flight deals, trying to figure out 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 the 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 analysing the data I have achieved a good model that has **82% accuracy**. So let's understand 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.

Review of Literature

It is hard for the client to buy an all ticket at the most reduced cost. For this few procedures are explored to determine the time and date to grab air tickets with a 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 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 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 that 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 on flight fares with other features and work to make a model to predict the fares of flights.

Motivation for the Problem Undertaken

The flight Price Prediction project help tourists find the right flight price based on their needs and also it gives various options and flexibility for traveling.
Different features (airline, source, destination, departure and arrival timings, Journey date, etc.) help to understand the flight price variations. Using it airlines also get benefits and required passengers. Also, they will get benefit from scheduling also.

ANALYRICAL PROBLEM FRAMING

Mathematical/ Analytical Modeling of the Problem

As a first step, I have scrapped the required data from the MakeMyTrip website. I have fetched data for different sources and destinations and saved it in 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 were no null values in the dataset. Since we have scrapped the data from the yatra website the raw data was not in the format, so we have used feature engineering to extract the required feature format. To get a deeper insight into the features I have used plotting like distribution plot, bar plot, strip plot, and count plot. With this plotting, I was able to understand the relation between the features in a better manner. I did not find any skewness or outliers in the dataset. I have used all the regression algorithms while building the model then tunned the best

model and saved the best model. At last, I have predicted the Price using the saved model.

Data Sources and their formats

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

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

Data Preprocessing Done

- As a first step I have scrapped the required data using selenium from the MakeMyTrip website.
- And I have imported the required libraries and I have imported the dataset which was in CSV format.
- Then I did all the statistical analysis like checking shape, unique 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 the 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. Think that this data will help us more than raw data.

Data Inputs- Logic- Output Relationships

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

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: -

Processor — core i5 and above
RAM — 8 GB or above

Software/s required: -

Anaconda

Libraries required:

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

```
In [1]: #importing required libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import datetime as dt

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

import pandas as pd: **pandas** is a popular Python-based data analysis toolkit that 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 the exploration and understanding of data.

Import matplotlib.pyplot as plt: matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each plot 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 xgboost import XGBRegressor

- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.ensemble import ExtraTreesRegressor
- ✓ from sklearn.neighbors import KNeighborsRegressor as KNN
- ✓ from sklearn.ensemble import BaggingRegressor
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.metrics import accuracy_score
- ✓ from sklearn.model_selection import cross_val_score

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

DATA ANALYSIS & VISUALIZATION

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 were 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 Standardisation to scale the data. After scaling we have to check multicollinearity using VIF. Then followed by model building with all Regression algorithms.

Testing of Identified Approaches (Algorithms)

Since Price was my target and it was a continuous column with an improper format which has to be changed to a continuous float datatype column, this particular problem was a Regression problem. And I have used all Regression algorithms to build my model. By looking into the r2 score and error values I found ExtraTreesRegressor as the best model with the highest r2_score and least error values. Also to get the best model we have to run through multiple. Below is the list of Regression algorithms I have used in my project.

- RandomForestRegressor
- XGBRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor
- DecisionTreeRegressor
- KNN
- BaggingRegressor

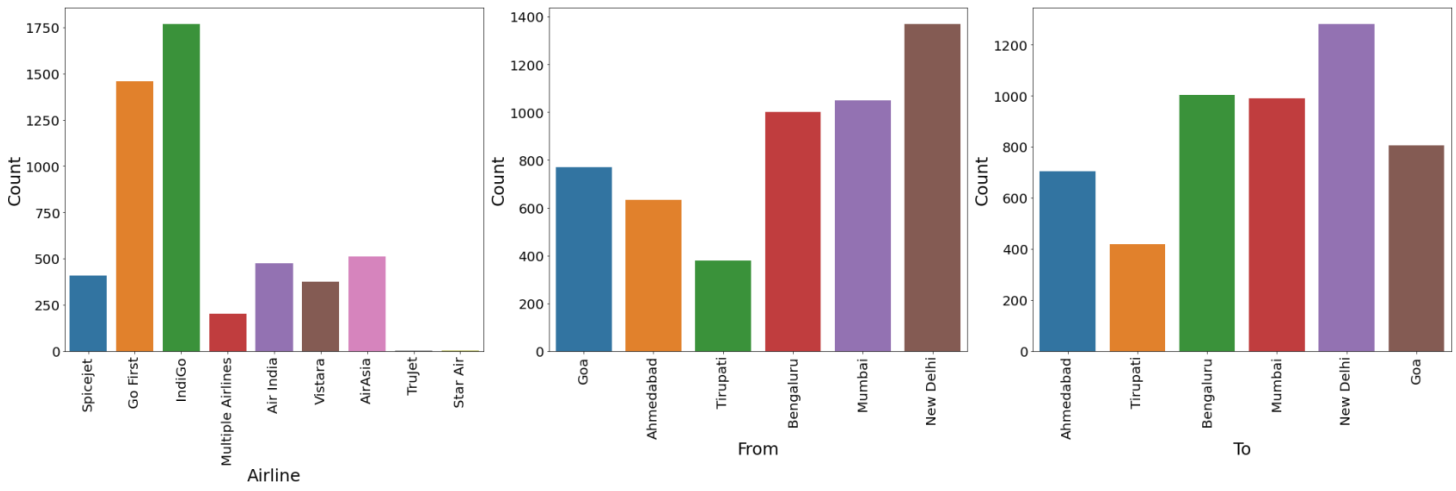
I have used the following metrics for evaluation:

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

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 a dist plot for univariate and strip plot for bivariate analysis.

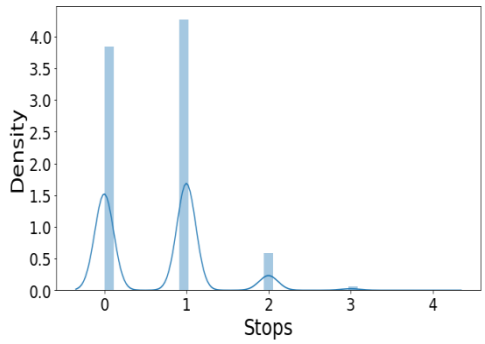
Univariate Analysis for Categorical columns:

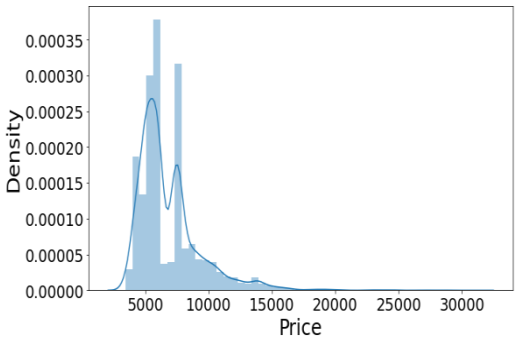


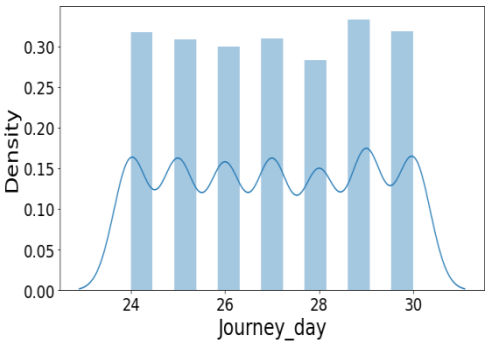
Observations:

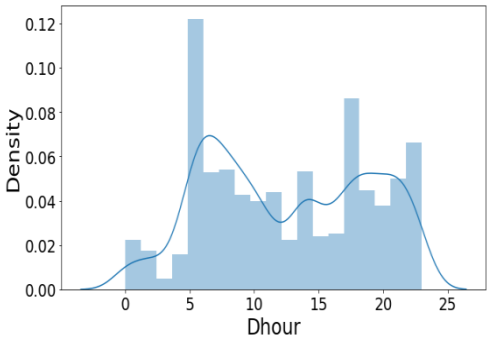
- Indigo has a maximum count which means most of the passengers preferred Indigo for their traveling.
- New Delhi has a maximum count for source which means maximum passengers are choosing New Delhi as their source.
- New Delhi has a maximum count for Destination which means maximum passengers are choosing New Delhi as their Destination.

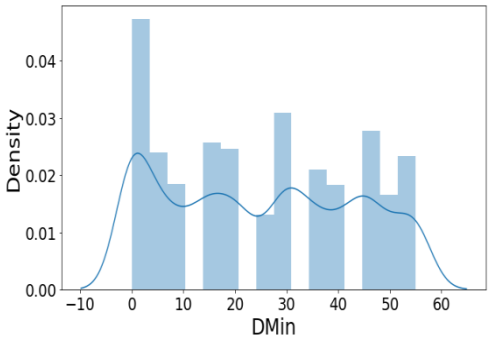
Univariate analysis for Numerical column:

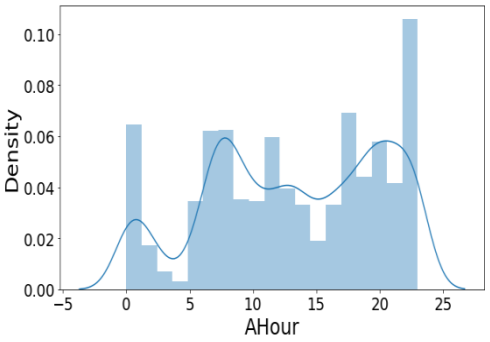


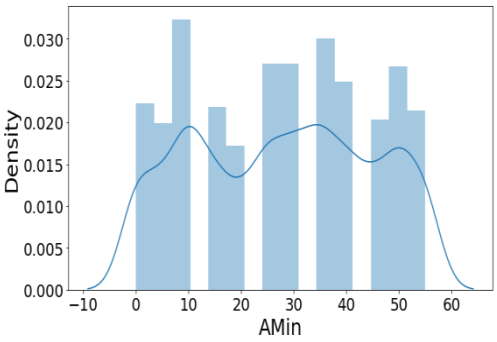








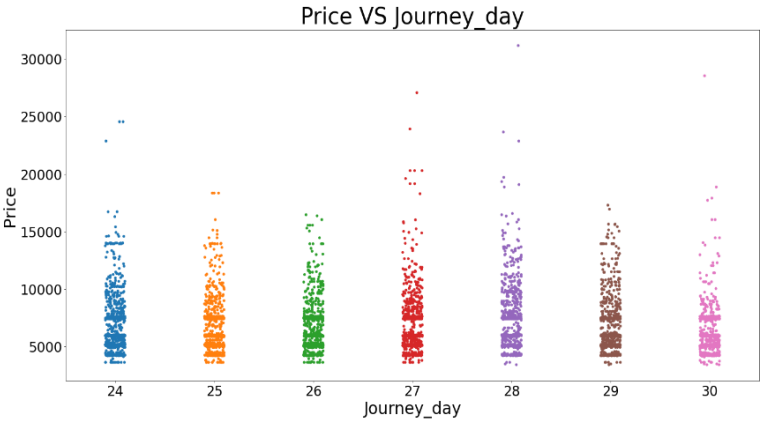


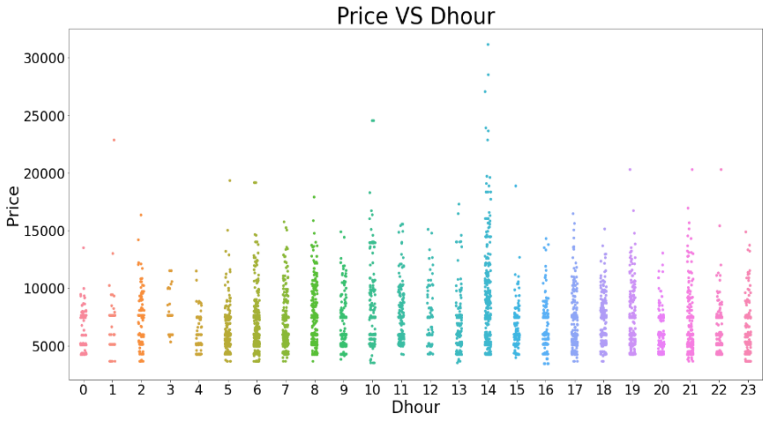


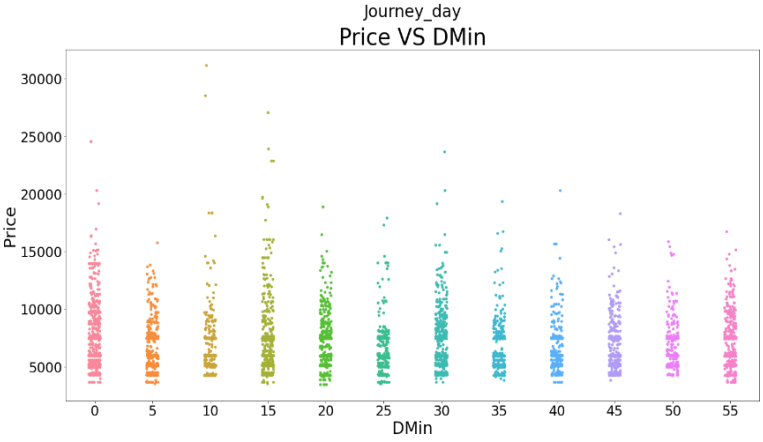
Observations:

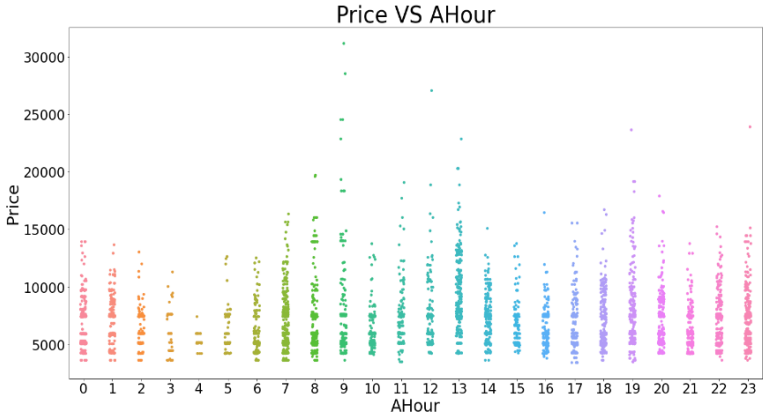
There is no skewness in any of the numerical columns.

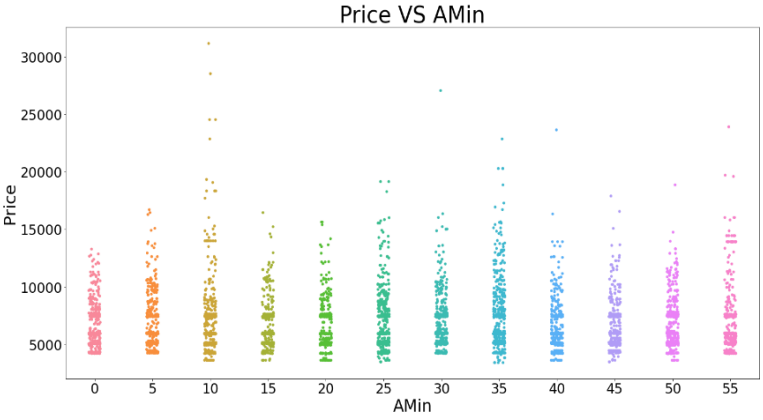
Bivariate analysis for numerical columns:







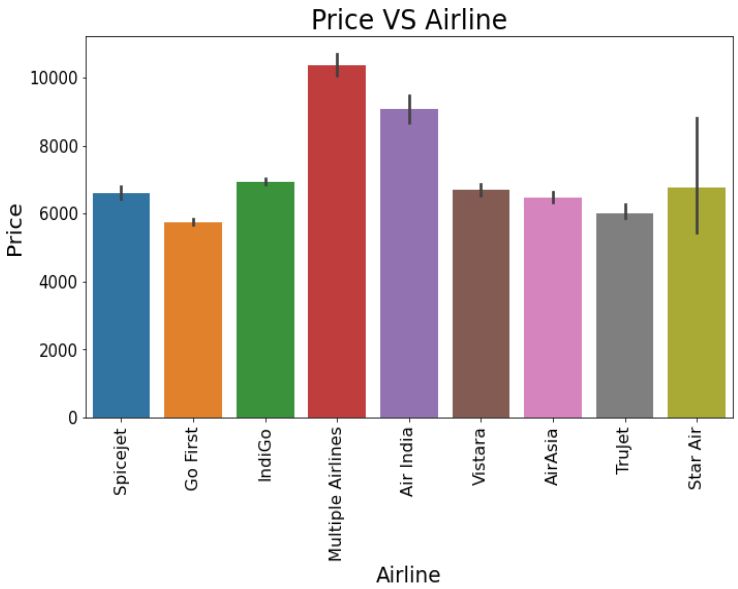


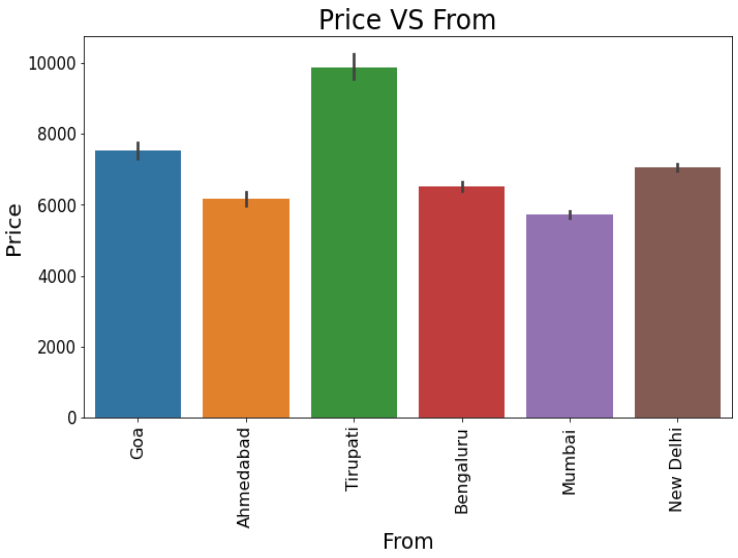


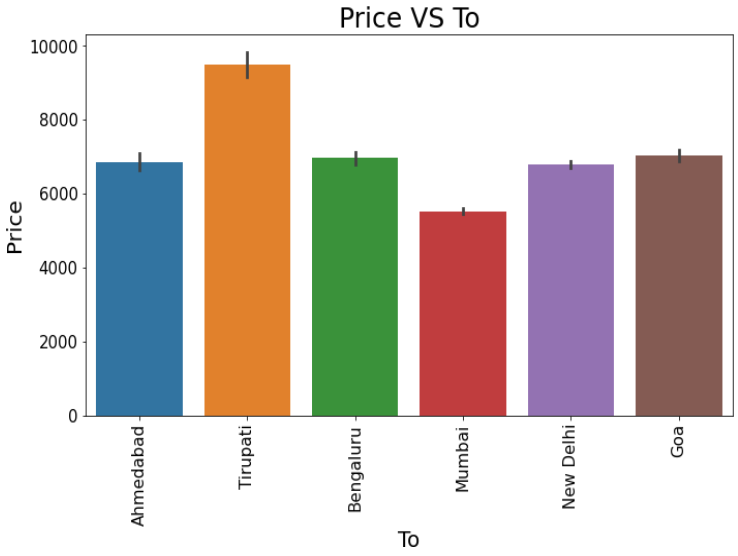
Observations:

- Flights with 2 stops cost more price compared to other flights.
- In all the dates the price is almost the same.
- At 2 PM 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 7 AM to 1 PM 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.

Bivariate Analysis for categorical columns:







Observations:

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

Run and Evaluate selected models

Model Building:

RandomForestRegressor:

In [65]:
```
RFR=RandomForestRegressor()
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 81.05921939883352
mean_squared_error: 1178818.6868681058
mean_absolute_error: 582.8659834965376
root_mean_squared_error: 1085.7341695222206
```

RandomForestRegressor has given me 81.06% r2_score, but still we have to look into multiple models.

XGBRegressor:

In [66]:
```
XGB=XGBRegressor()
XGB.fit(X_train,y_train)
pred=XGB.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 79.56952342348205
mean_squared_error: 1271532.9994655694
mean_absolute_error: 678.8862794219951
root_mean_squared_error: 1127.6227203571102
```

XGBRegressor is giving me 79.57% r2_score.

ExtraTreesRegressor:

In [67]:
```
ETR=ExtraTreesRegressor()
ETR.fit(X_train,y_train)
pred=ETR.predict(X_test)
print('R2_score:',r2_score(y_test,pred))
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.811783877185204
mean_squared_error: 1171401.9998219074
mean_absolute_error: 535.5772720478325
root_mean_squared_error: 1082.3132632569498
```

ExtraTreesRegressor is giving me 81.18% r2_score.

GradientBoostingRegressor:

In [68]:
```
GBR=GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred=GBR.predict(X_test)
print('R2_score:',r2_score(y_test,pred))
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.6569134340645822
mean_squared_error: 2135270.259733484
mean_absolute_error: 952.091629491256
root_mean_squared_error: 1461.2563976706772
```

GradientBoostingRegressor is giving me 65.69% r2_score.

DecisionTreeRegressor:

In [69]:
```
DTR=DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred=DTR.predict(X_test)
print('R2_score:',r2_score(y_test,pred))
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
R2_score: 0.6459859162056638
mean_squared_error: 2203279.92905189
mean_absolute_error: 647.523062139654
root_mean_squared_error: 1484.34494948172
```

DecisionTreeRegressor is giving me 64.60% r2_score.

KNN:

```
In [70]: knn=KNN()
         knn.fit(X_train,y_train)
         pred=knn.predict(X_test)
         print('R2_score:',r2_score(y_test,pred))
         print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
         print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
         print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

         R2_score: 0.532582749003129
         mean_squared_error: 2909068.013837284
         mean_absolute_error: 1092.0563741191545
         root_mean_squared_error: 1705.599019065526
```

KNN is giving me 53.26% r2_score.

BaggingRegressor:

```
In [71]: BG=BaggingRegressor()
         BG.fit(X_train,y_train)
         pred=BG.predict(X_test)
         print('R2_score:',r2_score(y_test,pred))
         print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
         print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
         print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

         R2_score: 0.7897988903795836
         mean_squared_error: 1308230.1159525483
         mean_absolute_error: 606.6656438180654
         root_mean_squared_error: 1143.7788754617513
```

BaggingRegressor is giving me 78.98% r2_score.

By looking into the model r2_score and error I found ExtraTreesRegressor as the best model with highest r2_score and least errors.

Hyper Parameter Tunning:

In [72]:
```
#importing necessary libraries
from sklearn.model_selection import GridSearchCV
```

In [73]:
```
parameter = {'max_features':['auto','sqrt','log2'],
             'min_samples_split':[1,2,3,4],
             'n_estimators':[20,40,60,80,100],
             'min_samples_leaf':[1,2,3,4,5],
              'n_jobs':[-2,-1,1,2]}
```

Giving ETR parameters.

In [74]:
```
GCV=GridSearchCV(ExtraTreesRegressor(),parameter,cv=5)
```

Running grid search CV for ETR.

In [75]:
```
GCV.fit(X_train,y_train)
```

Out[75]:
```
GridSearchCV(cv=5, estimator=ExtraTreesRegressor(),
             param_grid={'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 3, 4, 5],
                         'min_samples_split': [1, 2, 3, 4],
                         'n_estimators': [20, 40, 60, 80, 100],
                         'n_jobs': [-2, -1, 1, 2]})
```

In [76]:
```
GCV.best_params_
```

Out[76]:
```
{'max_features': 'auto',
 'min_samples_leaf': 2,
 'min_samples_split': 2,
 'n_estimators': 80,
 'n_jobs': 1}
```

Got the best parameters for ETR.

In [77]:
```
Best_mod=ExtraTreesRegressor(max_features='auto',min_samples_leaf=2,min_samples_split=2,n_estimators=80,n_jobs=1)
Best_mod.fit(X_train,y_train)
pred=Best_mod.predict(X_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2_Score: 82.01424634259344
mean_squared_error: 1119380.6081810314
mean_absolute_error: 550.2649657480705
RMSE value: 1058.0078488277068
```

I have chosen all parameters of ExtraTreesRegressor, after tunning the model with best parameters I have increased my model accuracy from 81.18% to 82.01%.

Saving the model and Predictions:

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

In [78]:
```
# Saving the model using .pkl
import joblib
joblib.dump(Best_mod,"Flight_Price.pkl")
```

Out[78]:
```
['Flight_Price.pkl']
```

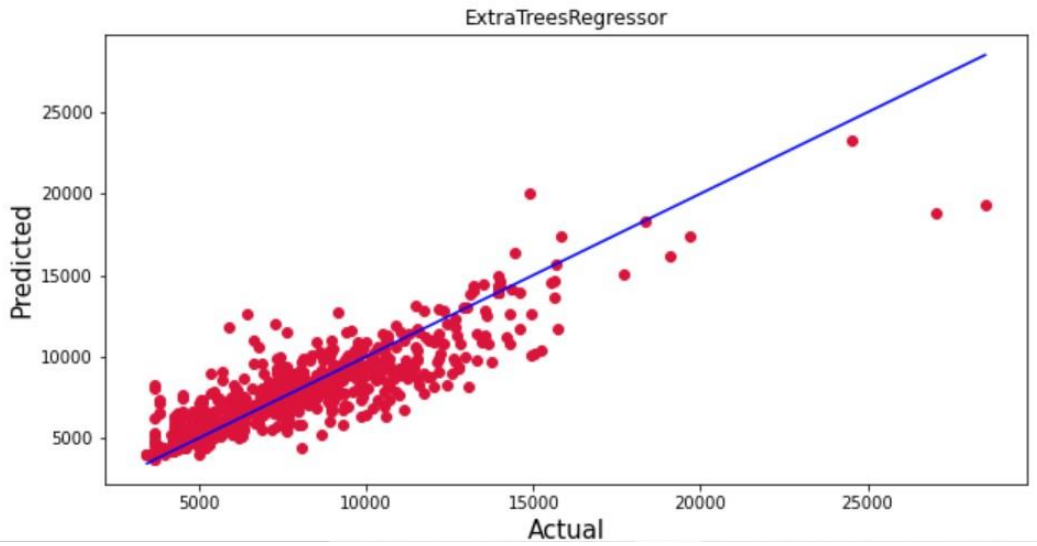
Now loading my saved model and predicting the price values.

```
In [79]: # Loading the saved model
         model=joblib.load("Flight_Price.pkl")

         #Prediction
         prediction = model.predict(X_test)
         prediction

Out[79]: array([7103.75875   , 7220.78854167, 5102.40416667, ..., 5921.47083333,
                5940.91875   , 7698.17083333])

In [80]: pd.DataFrame([model.predict(X_test)[:],y_test[:]],index=["Predicted","Actual"])

Out[80]:
```

	0	1	2	3	4	5	6	7	8	9	...	1551	
Predicted	7103.75875	7220.788542	5102.404167	6034.927917	8315.571875	5057.31875	4718.034375	8345.773958	7916.926042	6082.25625	...	5256.193333	54
Actual	7671.00000	5566.000000	5061.000000	5900.000000	7425.000000	5060.00000	4264.000000	8265.000000	7487.000000	5955.00000	...	4845.000000	51

2 rows × 1561 columns

```
In [81]: plt.figure(figsize=(10,5))
         plt.scatter(y_test, prediction, c='crimson')
         p1 = max(max(prediction), max(y_test))
         p2 = min(min(prediction), min(y_test))
         plt.plot([p1, p2], [p1, p2], 'b-')
         plt.xlabel('Actual', fontsize=15)
         plt.ylabel('Predicted', fontsize=15)
         plt.title("ExtraTreesRegressor")
         plt.show()
```



Plotting Actual vs Predicted, to get a better insight. The blue line is the actual line and the red dots are the predicted values.

Interpretation of the Results

- The dataset was scraped from the MakeMyTrip website.
- The dataset was very challenging to handle it had 9 features with 5204 samples.
- Firstly, the datasets was having a complete row as nan values, so I dropped that row.

- And there was a huge number of unnecessary entries in all the features so I have used feature extraction to get the required format of variables.
- And proper plotting for the proper type of features will help us to get a better insight into the data. I found both numerical columns and categorical columns in the dataset so I have chosen strip plot and bar plot to see the relation between target and features.
- I did not find any outliers or skewness in the dataset.
- Then scaling the dataset has a good impact like it will help the model not to get biased. Since we did not have outliers and skewness in the dataset so we have to choose Standardisation.
- We have to use multiple models while building a model using a dataset to get the best model out of it.
- And we have to use multiple metrics like mse, mae, rmse, and r2_score which will help us to decide on the best model.
- I found ExtraTreesRegressor as the best model with an 81.18% r2_score. Also, I have improved the accuracy of the best model by running hyper- parameter tuning.
- At last I have predicted the used flight price using saved model. It was good!! that I was able to get the predictions near to actual values.

CONCLUSION

Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict flight prices. We have mentioned the step-by-step procedure to analyze the dataset and find the correlation between the features. Thus, we can select the features which are correlated to each other and are independent in nature.
These feature sets 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 the predicted and actual values were almost same.

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-scrapped from the 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 understand what data is trying to say. Data cleaning is one of the most important steps to removing 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 prices 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 prices. The future direction of research may consider incorporating additional used flight data from a larger economical background with more features.

Limitations of this work and Scope for Future Work

- The First drawback is scrapping the data as it is a fluctuating process.
- Followed by raw data which is not in format to analyze.
- Also, we have tried our best to deal with improper format data and null values. So it looks quite good that we have achieved a accuracy of 82.01% even after dealing all these drawbacks.
- Also, this study will not cover all Regression algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones