FLIP ROBO

PROJECT REPORT ON:

“Flight Price Prediction Project”

SUBMITTED BY

Mr. Vikram Purohit

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 **Mr. Mohd. Kashif** (SME Flip Robo), he is the person who has helped me to get out of all the difficulties I faced while doing the project. He has inspired me in so many aspects and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

A huge thanks to my academic team "**Data Trained**" who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

Contents:

1.INTRODUCTION

1.1 Business Problem Framing:

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

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

As retrieving price metrics through aggregation techniques and business intelligence tools alone could lead to incorrect conclusions – for example, in cases where 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.

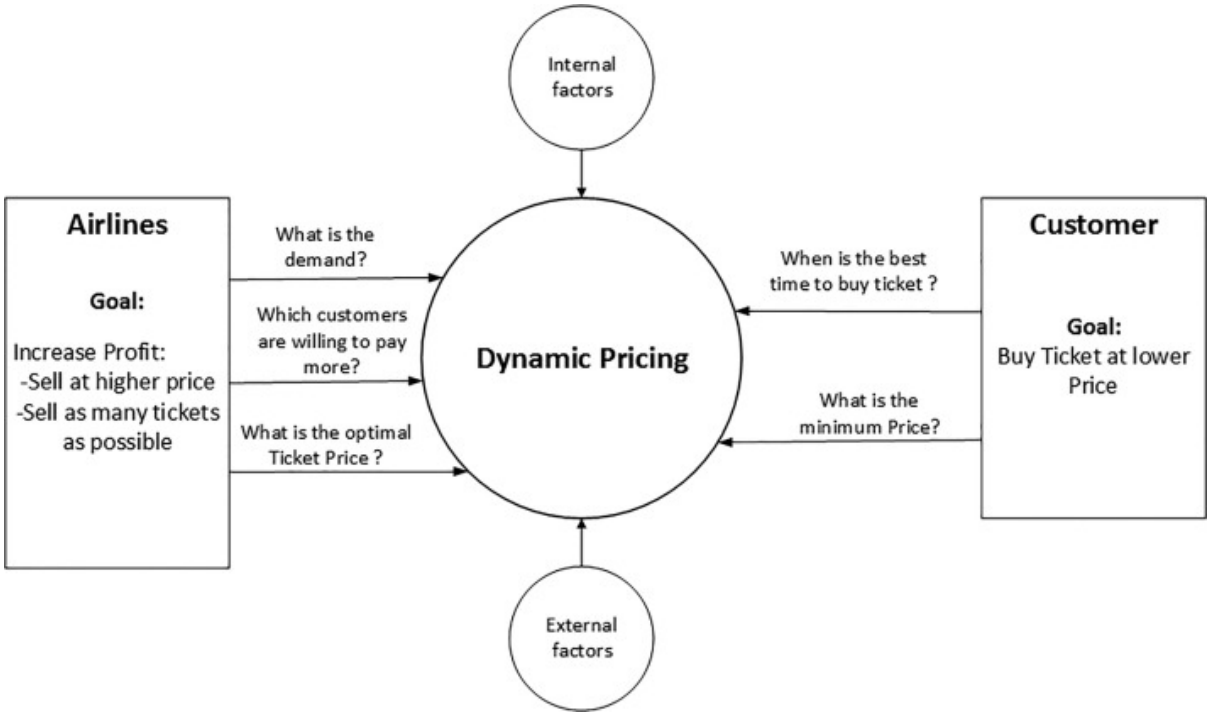
2. Conceptual Background of the Domain Problem

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

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

model that has **83.34% accuracy**. So let's understand what all the steps we did to reach this good accuracy.

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



3. Review of Literature

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

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

1. Time of purchase patterns (making sure last-minute purchases are expensive)

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

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

1.4 Motivation for the Problem Undertaken

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

2.Analytical Problem Framing

1. Mathematical/ Analytical Modelling of the Problem

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

In this particular problem I have Price as my target column and it was a continuous column. So clearly it is a regression problem and I have to use all regression algorithms while building the model. There was null values in the dataset. Since we have scrapped the data from Easemytrip website the raw data was not in the format, so we have use feature engineering to extract the required feature format. To get better insight on the features I have used plotting like distribution plot, bar plot, strip plot and count plot. With these plotting I was able to understand the relation between the features in better manner. I did not found any skewness or outliers in the dataset. I have used all the regression

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

2.2 Data Sources and their formats

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

Also, my dataset was having 4480 rows and 10 columns including target. In this particular datasets I have object type of data which has been changed as per our analysis about the dataset. The information about features is as follows.

Features Information:

- Airline: The name of the airline.
- date: The date of the journey
- Source: The source from which the service begins.
- Destination: The destination where the service ends.
- Departure Time: The time when the journey starts from the source.
- Arrival Time: Time of arrival at the destination.
- Duration : Time duration for flight to travel from source to destination.
- Stops: Total stops between the source and destination.
- Price: The price of the ticket

3. Data Preprocessing Done

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

- ✓ I have also droped 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 date-time columns and I have extracted useful information from the raw dataset. Thinking that this data will help us more than raw data.

2.4 Data Inputs- Logic- Output Relationships

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

5. Hardware and Software Requirements and Tools Used

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

Hardware required: -

1. Processor — 2.9 GHz Dual-Core Intel Core i5

2. RAM — 8 GB 2133 MHz LPDDR3

3. SSD — Intel Iris Graphics 550, 1536 MB

Software/s required: -

1.Anaconda

Libraries required :-

```
#IMPORTING NEEDED LIBRARIES
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

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

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

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

3.Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)

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

2. Testing of Identified Approaches (Algorithms)

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

- ➢ RandomForestRegressor
- ➢ Ridge Regression
- ➢ GradientBoostingRegressor
- ➢ DecisionTreeRegressor
- ➢ BaggingRegressor

3. Key Metrics for success in solving problem under consideration

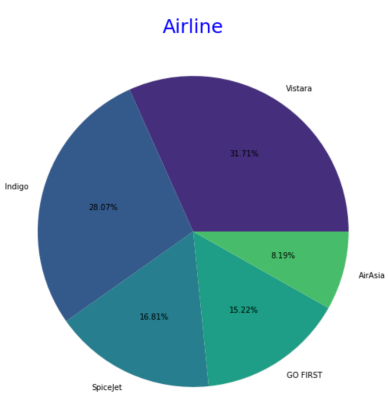
I have used the following metrics for evaluation:

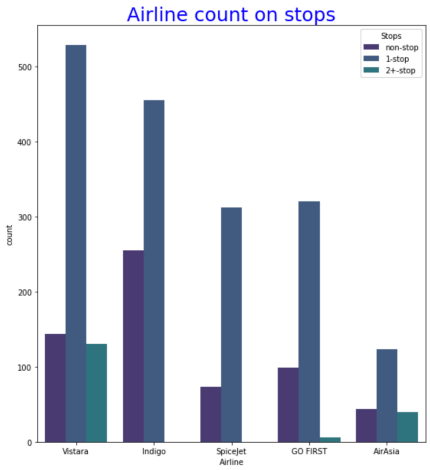
- I have used mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.
- I have used root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.
- I have used r2 score which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

3.4 Visualizations

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

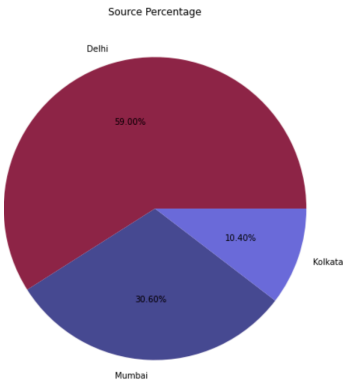
✺Univariate Analysis for Categorical columns:

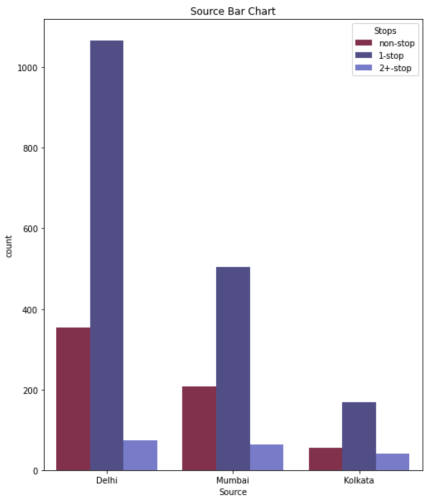




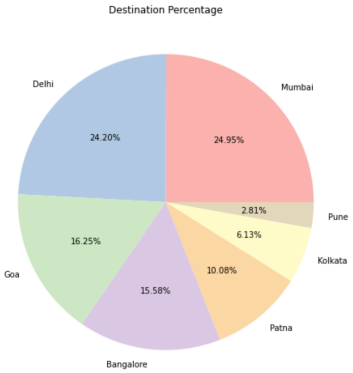
OBSEVATION:

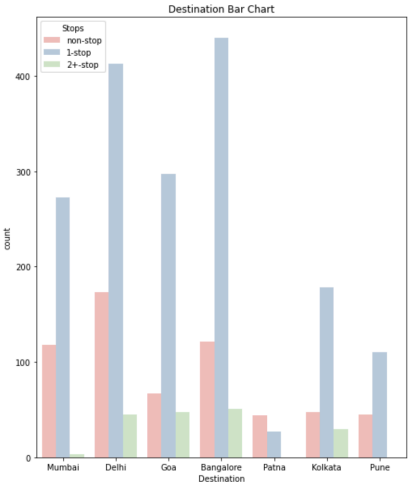
- Our data contains 36% of vistara flights followed by indigo
- Vistara flights are having more number of flights with 1 stop followed by indigo
- Air Asia is having the lowest count amongst all





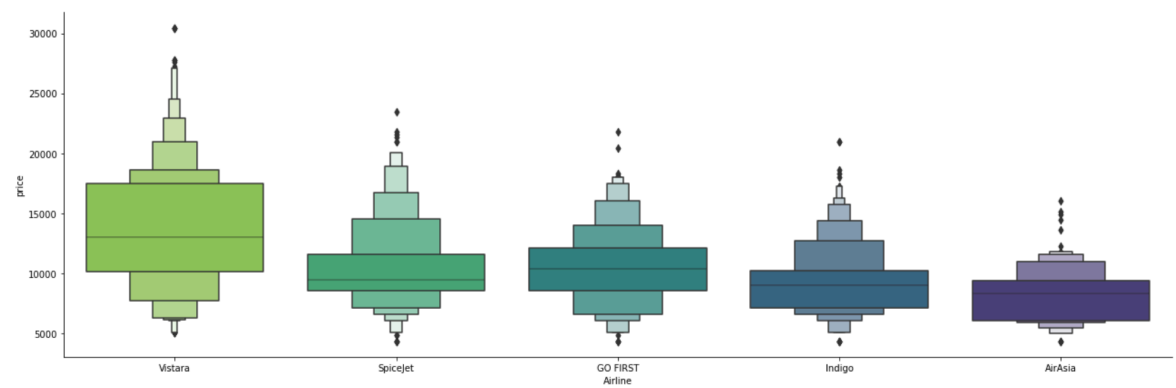
- FROM THE ABOVE PIE CHART AND COUNT PLOT WE CAN SEE THAT
- MOST FLIGHTS ARE FROM DELHI FOLLOWED BY MUMBAI
- MAXIMUM FLIGHTS TO THEIR DESTINATION ARE WITH 1 STOP

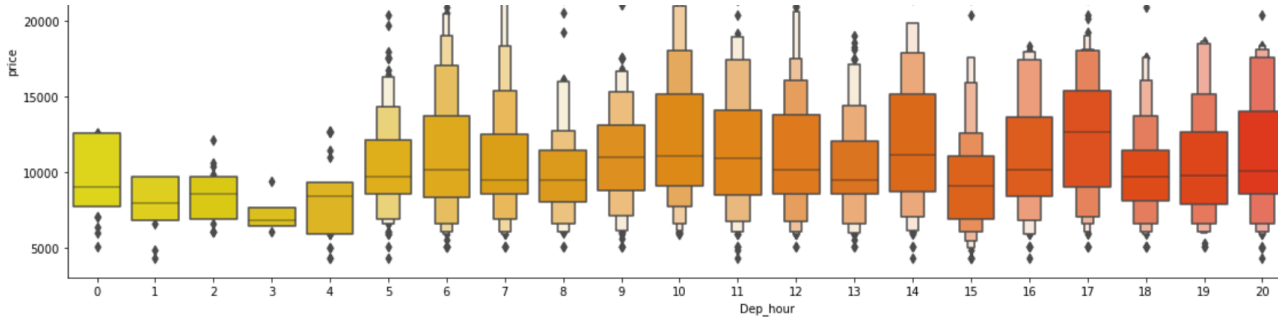


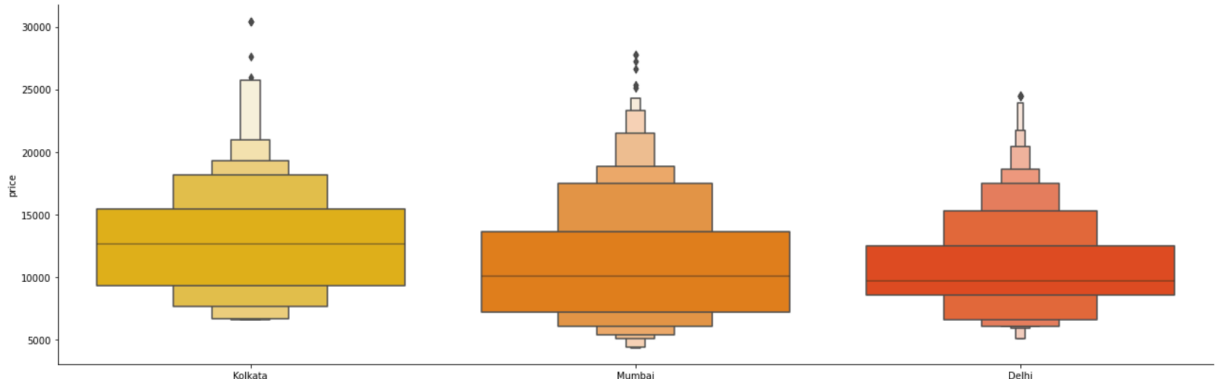


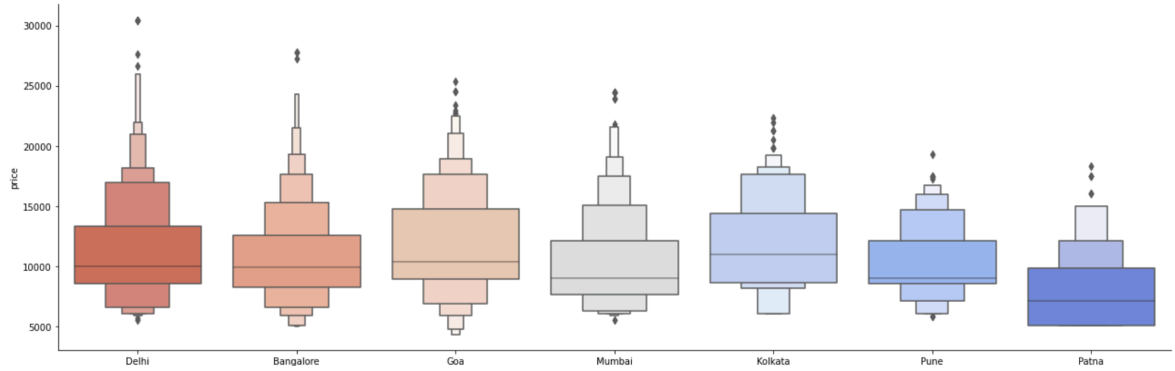
- MUMBAI AND DELHI ARE HAVING THE HIGHEST FLIGHT DESTINATION AND MOSTLY WITH 1 STOP FOLLOWED BY GOA AND BANGALORE.
- THEIR ARE LESS FLIGHTS TO PUNE.

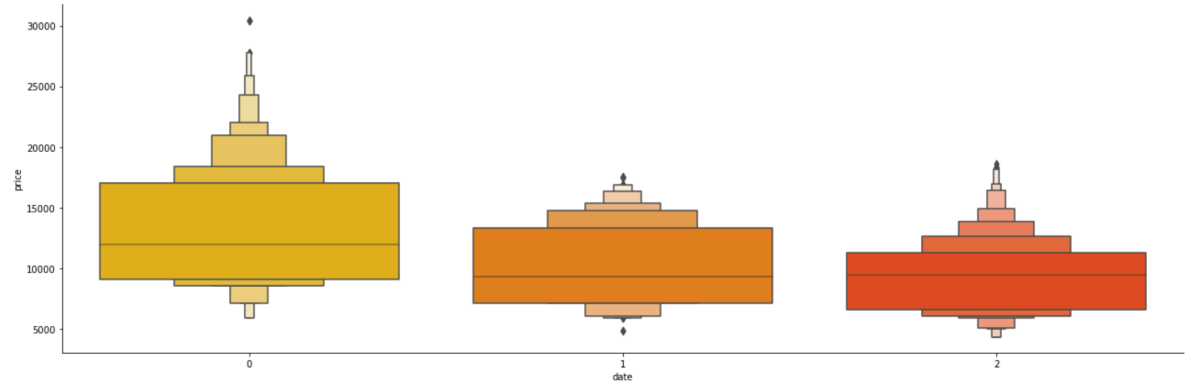
✺Bivariate analysis for numerical columns:

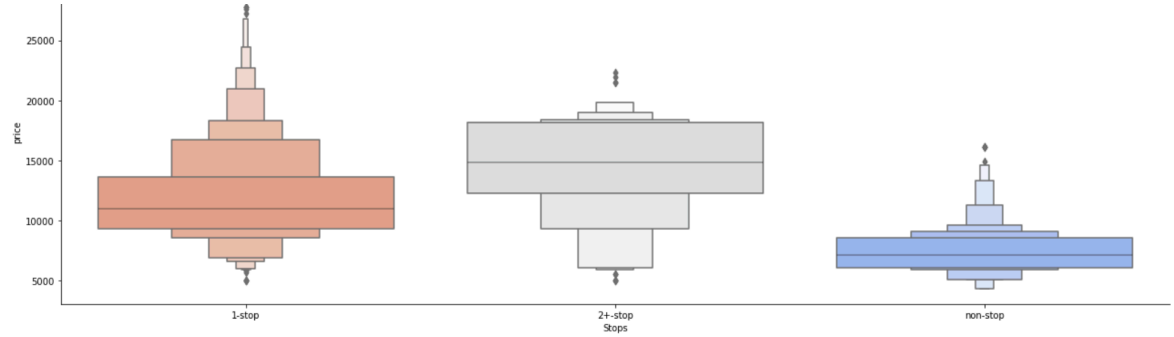



price
20000
15000
10000
5000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Dep_hour


price
30000
25000
20000
15000
10000
5000
Kolkata
Mumbai
Delhi


price
30000
25000
20000
15000
10000
5000
Delhi
Bangalore
Goa
Mumbai
Kolkata
Pune
Patna


price
30000
25000
20000
15000
10000
5000
0
1
2
date


price
25000
20000
15000
10000
5000
1-stop
2+-stop
non-stop
Stops

Observations:

- ✓ Flights with 2 stops costs more price compared to other flights.
- ✓ In all the dates the price is almost same.
- ✓ Morning to noon 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.
- ✓ Vistara flights are more costly than others and vistara flights have a high frequency too.
- ✓ Cheapest flights are air asia or spice jet .
- ✓ Flights from kolkata are marked a high price followed by mumbai and delhi.
- ✓ Flights for goa delhi and kolkata are having a bit higher price than others.
- ✓ Date of travel also decides the fare price.
- ✓ Tickets after one month are comparatively low followed by tickets after 15 days.

- ☑ After Visualization the nest step is to find **Correlation** , where I didn't found any correlating column, only data is negatively correlated to price.
- ☑ Next step I moved into **Encoding** where I changed categorical column to numerical by one hot encoding and label encoding.
- ☑ Found **Outliers** in data in 2 columns : Duration Hours and Price ,so removed only from Duration Hours by Percentile method.
- ☑ Checked **Skewness** ,where I removed skewness from arrival hour and Duration hour by Power Transform yeo-johnson method.
- ☑ Features were first checked for presence of **Multicollinearity** by VIF(Variance Inflation Factor), but not found any.
- ☑ Using **StandardScaler**, the features were scaled by resizing the distribution values so that mean of the observed values in each feature column is 0 and standard deviation is 1.
- ☑ From sklearn.model_selection's **train_test_split**, the data was divided into train and test data. Training data comprised 78% of total data where as test data comprised 22% based on the best random state that would result in best model accuracy.

5. Run and Evaluate selected models

1.**Model Building:**

```
from sklearn.ensemble import RandomForestRegressor
maxAcc = 0
maxRS=0
for i in range(1,100):
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = .25, random_state = i)
    modRF =  RandomForestRegressor()
    modRF.fit(x_train,y_train)
    pred = modRF.predict(x_test)
    acc  = r2_score(y_test,pred)
    if acc>maxAcc:
        maxAcc=acc
        maxRS=i
print(f"Best Accuracy is: {maxAcc} on random_state: {maxRS}")
```

```
Best Accuracy is: 0.8423638392528663 on random_state: 54
```

Best random state is determined to be 54.

1) Decision Tree Regressor:

```
from sklearn.tree import DecisionTreeRegressor

dtc=DecisionTreeRegressor()
dtc.fit(x_train,y_train)
dtc.score(x_train,y_train)
dtc_pred=dtc.predict(x_test)


print('score : ',dtc.score(x_train,y_train))
print('r2 score : ',r2_score(y_test,dtc_pred))

# printing errors

print('Mean absolute error:', mean_absolute_error(y_test,dtc_pred))
print('Mean squared error:', mean_squared_error(y_test,dtc_pred))
print('Root mean squared error:', np.sqrt(mean_squared_error(y_test,dtc_pred)))
```

```
score :  1.0
r2 score :  0.6915353116814669
Mean absolute error: 1247.475763016158
Mean squared error: 5211646.078994614
Root mean squared error: 2282.9029937766986
```

- Decision Tree Regressor has given me 69.15% r2_score.

2) Ridge Regressor:

```
from sklearn.linear_model import Ridge
rr=Ridge()
rr.fit(x_train,y_train)
predrr=rr.predict(x_test)
print('Score: ',rr.score(x_train,y_train))
print('r2 score: ', r2_score(y_test,predrr))

# printing errors
print('Mean absolute error:', mean_absolute_error(y_test,predrr))
print('Mean squared error:', mean_squared_error(y_test,predrr))
print('Root mean squared error:', np.sqrt(mean_squared_error(y_test,predrr)))
```

```
Score:  0.6057788138544564
r2 score:  0.6136358758449897
Mean absolute error: 1835.8791694720746
Mean squared error: 6527791.1831429135
Root mean squared error: 2554.9542428667705
```

- Ridge Regressor is giving me 61.36% r2_score.

3) Random Forest Regressor:

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
predrf=rf.predict(x_test)
print('Score: ',rf.score(x_train,y_train))
print('r2 score: ', r2_score(y_test,predrf))

# printing errors

print('Mean absolute error:', mean_absolute_error(y_test,predrf))
print('Mean squared error:', mean_squared_error(y_test,predrf))
print('Root mean squared error:', np.sqrt(mean_squared_error(y_test,predrf)))
```

```
Score:  0.9731384529950098
r2 score:  0.8367409083876516
Mean absolute error: 1068.5859425493713
Mean squared error: 2758333.893256194
Root mean squared error: 1660.8232576816215
```

- Random Forest Regressor is giving me 83.67% r2_score.

4) GradientBoostingRegressor:

```
from sklearn.ensemble import GradientBoostingRegressor
gbr= GradientBoostingRegressor()
gbr.fit(x_train,y_train)
predgbr= gbr.predict(x_test)
print('Score: ',gbr.score(x_train,y_train))
print('r2 score: ', r2_score(y_test,predgbr))

# printing errors

print('Mean absolute error:', mean_absolute_error(y_test,predgbr))
print('Mean squared error:', mean_squared_error(y_test,predgbr))
print('Root mean squared error:', np.sqrt(mean_squared_error(y_test,predgbr)))
```

```
Score:  0.7755458186052308
r2 score:  0.7722400328204475
Mean absolute error: 1399.7687123728222
Mean squared error: 3848104.450378799
Root mean squared error: 1961.658596794763
```

- GradientBoostingRegressor is giving me 77.22% r2_score.

5) BaggingRegressor:

```
from sklearn.ensemble import BaggingRegressor
br= BaggingRegressor()
br.fit(x_train,y_train)
predbr= br.predict(x_test)
print('Score: ',br.score(x_train,y_train))
print('r2 score: ', r2_score(y_test,predbr))

# printing errors

print('Mean absolute error:', mean_absolute_error(y_test,predbr))
print('Mean squared error:', mean_squared_error(y_test,predbr))
print('Root mean squared error:', np.sqrt(mean_squared_error(y_test,predbr)))
```

```
Score:  0.9598595617807153
r2 score:  0.8124928170432592
Mean absolute error: 1130.3371633752245
Mean squared error: 3168016.022082586
Root mean squared error: 1779.8921377663833
```

- BaggingRegressor is giving me 81.24% r2_score.

✓ **By looking into the model r2_score and error I found RandomForest Regressor as the best model with highest r2_score and least errors.**

3. Hyper Parameter Tunning:

Hyperparameter tuning

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

```
param={'n_estimators':[10,50,70,100,140,200]}
```

```
rf=RandomForestRegressor()
gscv=GridSearchCV(rf,param)
gscv.fit(x,y)
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'n_estimators': [10, 50, 70, 100, 140, 200]})
```

```
gscv.best_params_
```

```
{'n_estimators': 100}
```

```
gscv.best_estimator_
```

```
RandomForestRegressor()
```

```
rf=RandomForestRegressor(n_estimators=100)
rf.fit(x_train,y_train)
rf.score(x_train,y_train)
y_pred=rf.predict(x_test)
print("Score of Model is",r2_score(y_test,y_pred))
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))))
```

```
Score of Model is 0.8384227683578752
Mean Absolute Error 1054.7162836624775
Mean Squared Error 2729918.1320648114
Root Mean Squared Error 1652.2463896358834
```

- **I have chosen all parameters of Random Forest Regressor, after tuning the model with best parameters. The best score is 83.34%.**

5. Saving the model and Predictions:

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

Saving Model

```
import pickle
pickle.dump(rf,open("Flight_price_Project.pkl",'wb'))
```

- Now loading my saved model and predicting the price values.

```
import numpy as np
z = np.array(y_test)
predicted = np.array(rf.predict(x_test))
df = pd.DataFrame(zip(z,predicted), columns = ["Original","Predicted"])
df
```

	Original	Predicted
0	9734	9895.40
1	18158	18613.11
2	10522	11581.11
3	6269	6129.24
4	14868	13538.50
...	...	...
552	8892	12573.49
553	5103	5659.89
554	9132	16433.34
555	10391	10952.62
556	6059	6139.21

557 rows × 2 columns

Actual vs Predicted, To get better insight. Blue line is the actual line and red dots are the predicted values.

6. Interpretation of the Results

- ✓ The dataset was scrapped from Easemytrip website.
- ✓ The dataset was very challenging to handle it had 10 features with 4480 samples.
- ✓ Firstly, the datasets was having a complete row as nan values, so I have dropped that row.
- ✓ And there was 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 proper type of features will help us to get better insight on the data. I found both numerical columns and categorical columns in the dataset so I have chosen cat-plot to see the relation between target and features.
- ✓ I found outliers and skewness in the dataset which I then removed using specific methods.
- ✓ Then scaling dataset has a good impact like it will help the model not to get biased.
- ✓ We have to use multiple models while building model using dataset as 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 the best model.
- ✓ I found Random Forest Regressor as the best model with 83.34% r2_score. Also I have improved the accuracy of the best model by running hyper parameter tunning.
- ✓ 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.

4.CONCLUSION

4.1 Key Findings and Conclusions of the Study

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

4.2 Learning Outcomes of the Study in respect of Data Science

I found that the dataset was quite interesting to handle as it contains all types of data in it and it was self scrapped from Easemytrip website using selenium. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques of machine learning can be used in flight price research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values and null values. This study is an exploratory attempt to use seven machine learning algorithms in estimating flight price prediction, and then compare their results.

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

3. Limitations of this work and Scope for Future Work

- ✓ First drawback is scrapping the data as it is a fluctuating process.
- ✓ Followed by raw data which is not in format to analyse.
- ✓ Also, we have tried best to deal with improper format data and null values. So it looks quite good that we have achieved a accuracy of 83.34% 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.