FLIP ROBO

PROJECT REPORT
ON
Flight Price Prediction Project

SUBMITTED BY
MAHESHKUMAR OTA

ACKNOWLEDGMENT

It gives me immense pleasure to deliver this report. Working on this project was a great learning experience that helped me attain in-depth knowledge on data analysis process.

Flip Robo Technologies (Bangalore) provided all of the necessary information and datasets, required for the completion of the project.

I express my gratitude to my SME, Ms. Khushboo Garg, for providing the dataset and directions for carrying out the project report procedure.

My heartfelt gratitude to DataTrained institute and FlipRobo company for providing me this internship opportunity. Last but not least to my sincere thanks to my family and all those who helped me directly or indirectly in completion this project.

CONTENTS

1.INTRODUCTION

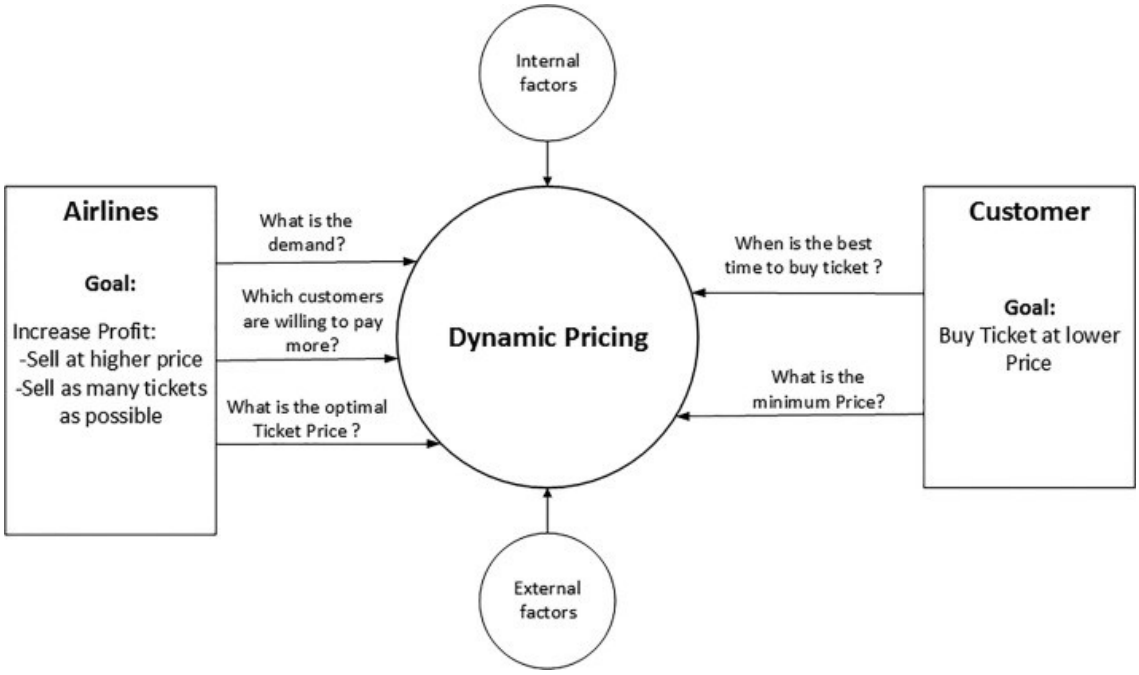
- **Business Problem Framing:**
  The tourism industry is changing fast and this is attracting a lot more travellers each year. The airline industry is considered as one of the most sophisticated industries in using complex pricing strategies. Now-a-days flight prices are quite unpredictable. The ticket 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.

- **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 information and analysing the data I have achieved a good model that has **99.9% 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.



- **Review of Literature:**

- It is hard for the client to buy an air ticket at the most reduced cost. For these 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.

- **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 timeings, 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

- **Mathematical / Analytical Modelling of the Problem:**
  As a first step I have scrapped the required data from yatra.com website. I have fetched data for different locations and saved it to excel 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 yatra 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 dist plot, and count plot which enabled better understanding of relation between the features. Also, I checked for outliers and skewness 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.

- **Data Sources and their formats:**
  The data was collected from yatra.com website in excel format. The data was scrapped using selenium. After scrapping required features the dataset is saved as excel file.
  Also, my dataset is having 5997 rows and 10 columns including target. In this dataset I ha ve object type of data which has been changed as per our analysis about the dataset. The information about features is as follows.

Features Information:

- Unnamed: - s.no of the dataset
- Airline - The name of the airline
- Departure_Time - The time when the journey starts from the source.
- Arrival_Time - Time of arrival at the destination.
- Duration – Travel time
- Source- The source from which the service begins.
- Destination - The destination where the service ends.
- Meal_availability – Availability of food in flight.
- Total_Stops - Total stops between the source and destination.
- Price - The price of the ticket

- **Data Pre-processing Done:**

- As a first step I have scrapped the required data using selenium from yatra 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.
- 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 datetime columns and I have extracted usefull information from the raw dataset. Thinking that this data will help us more than raw data.

- **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 bar plot for each pair of categorical features that shows the relation between label and independent features.

- **Hardware and Software Requirements and Tools Used:**

  Hardware technology being Used:-
  - CPU: HP Pavilion
  - Chip: intel core13 8th Gen
  - RAM: 8 GB

  Software Technology being Used:-

  - Programming language: Python
  - Distribution: Anaconda Navigator
  - Browser based language shell: Jupyter Notebook

  Libraries/Packages Used:-

  Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas_profiling.

3.DATA ANALYSIS AND VISUALIZATION

- **Identification of possible problem-solving approaches (methods):**

  - To tackle the problem, I employed both statistical and analytical methodologies, which mostly included data pre-processing and EDA to examine the connection of independent and dependent characteristics. In addition, before feeding the input data into the machine learning models, I made sure that it was cleaned and scaled. We need to anticipate the ticket price of the used cars for this project, which implies our goal column is continuous, making this a regression challenge. I evaluated the prediction using a variety of regression methods. After a series of assessments, I determined that ExtraTree Regressor is the best method for our final model since it has the best r2-score and the smallest difference in r2-score and CV-score of all the algorithms tested. Other regression methods are similarly accurate.
  - I used K-Fold cross validation to gain high performance and accuracy. Then hyper parameter tweaked the final model.

- Once I had my desired final model, I made sure to save it before loading the testing data and beginning to do data pre-processing as the training dataset and retrieving the anticipated selling price values from the Regression Machine Learning Model.

- **Testing of Identified Approaches (Algorithms):**

Since Car_price is the target and is a continuous column so given problem is regression problem. And I have used all regression algorithms to build my model. By looking into the difference of r2 score and cross validation score I found XGB Regressor as a best model with least difference. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have to go through cross validation. Below is the list of regression algorithms I have used in my project.

- RandomForestRegressor
- XGBRegresso
- ExtraTreesRegressor
- DecisionTreeRegressor

- **Key Metrics for success in solving problem under consideration:**

  r2 score, cross val score, MAE, MSE, and RMSE were the main metrics employed in this study. We used Hyperparameter Tuning to identify the optimal parameters and to improve our results, and we'll be utilising the GridSearchCV technique to do it.

  - Cross Validation:

  Cross-validation aids in determining the model's overfitting and underfitting. The model is constructed to run on several subsets of the dataset in cross validation, resulting in numerous measurements of the model. If we fold the data five times, it will be separated into five parts, each representing 20% of the whole dataset. During the Cross-validation, the first part (20%) of the 5 parts will be left out as a holdout set for validation, while the rest of the data will be utilised for training. We'll acquire the initial estimate of the dataset's model quality this way.

  Further rounds are produced in the same way for the second 20% of the dataset, which is kept as a holdout set while the remaining four portions are utilised for training data during the process. We'll acquire the second estimate of the dataset's model quality this way. During the cross-validation procedure, these stages are repeated to obtain the remaining estimate of model quality.

  - R2 Score:

  It is a statistical metric that indicates the regression model's quality of fit. The optimal r-square value is 1. The closer the r-square value is to 1, the better the model fits.

  - Mean Squared Error:

  The average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated is measured by the MSE of an estimator (of a process for estimating an unobserved variable). MSE is a risk function that represents the squared error loss's anticipated value. The Root Mean Squared Error is abbreviated as RMSE.

- Mean Absolute Error:

MAE is a statistic that assesses the average magnitude of mistakes in a set of forecasts without taking into account their direction. It's the average of the absolute differences between forecast and actual observation over the test sample, where all individual deviations are given equal weight.

- Hyperparameter Tunning:

There is a list of several machine learning models available. They're all distinct in some manner, yet the only thing that distinguishes them is the model's input parameters. Hyperparameters are the name given to these input parameters. These hyperparameters will establish the model's architecture, and the greatest thing is that you get to choose the ones you want for your model. Because the list of hyperparameters for each model differs, you must choose from a distinct list for each model.

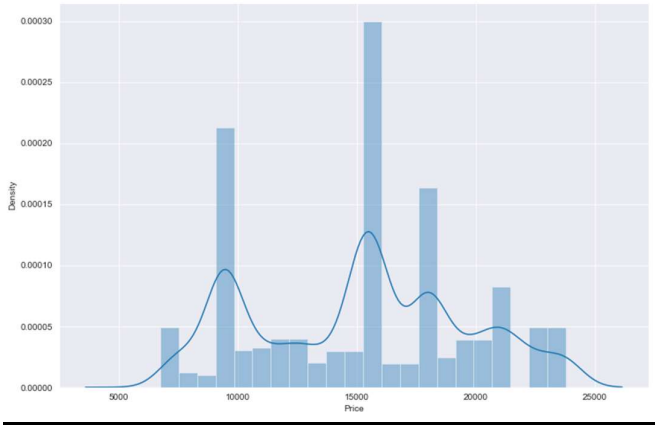
We are unaware of the ideal hyperparameter settings that would produce the best model output. So we instruct the model to automatically explore and choose the best model architecture. Hyperparameter tuning is the term for the method of selecting hyperparameters. GridSearchCV may be used to tune the system.

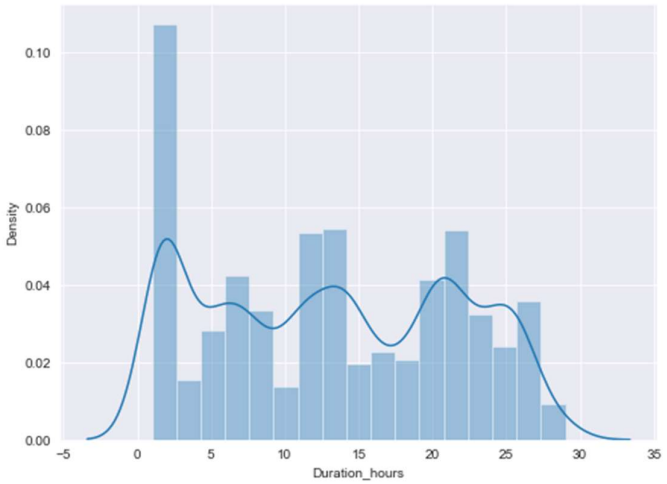
GridSearchCV is a model selection function in the Scikit-learn (or SK-learn) package. It is vital to remember that the Scikit-learn library must be installed on the PC. This function aids in fitting your estimator (model) to your training set by looping over specified hyperparameters. Finally, we may choose the optimal settings from the hyperparameters presented.

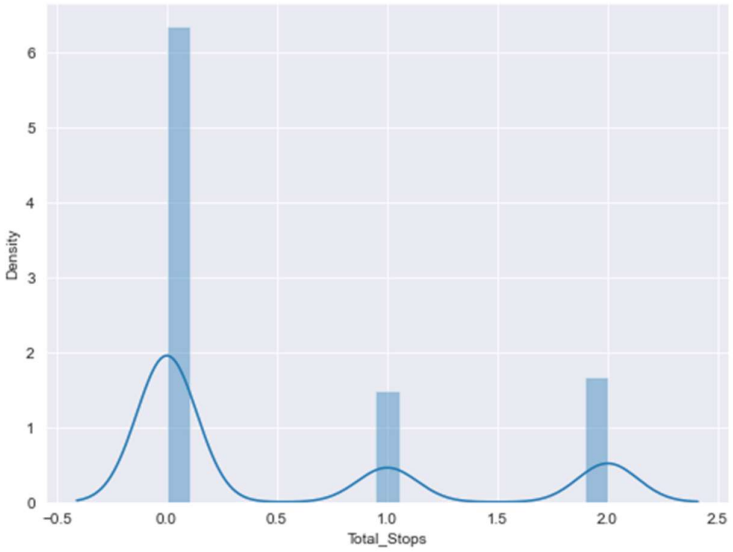
- **Visualization:**

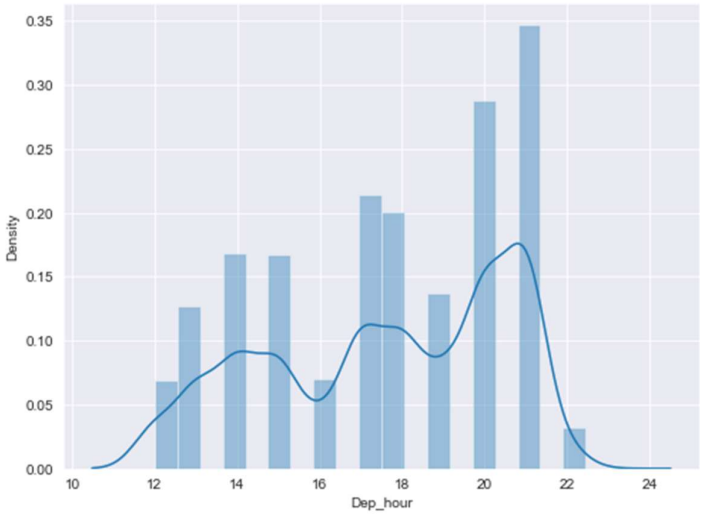
I have used bar plots to see the relation of categorical feature and I have used dist plots for numerical features.

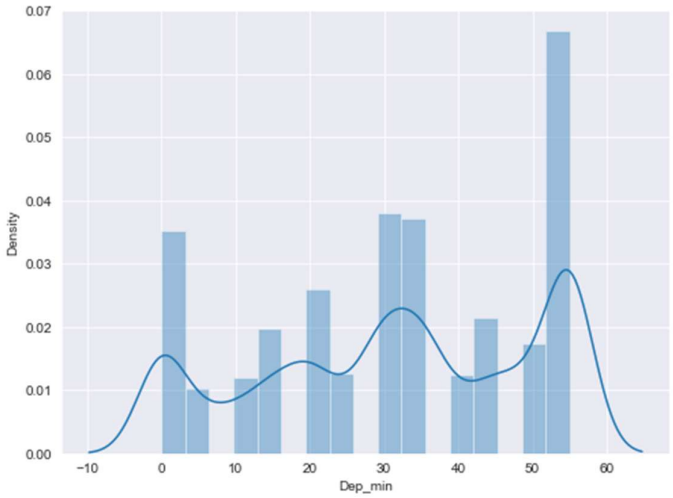
- **<u>Visualization of numerical features:</u>**

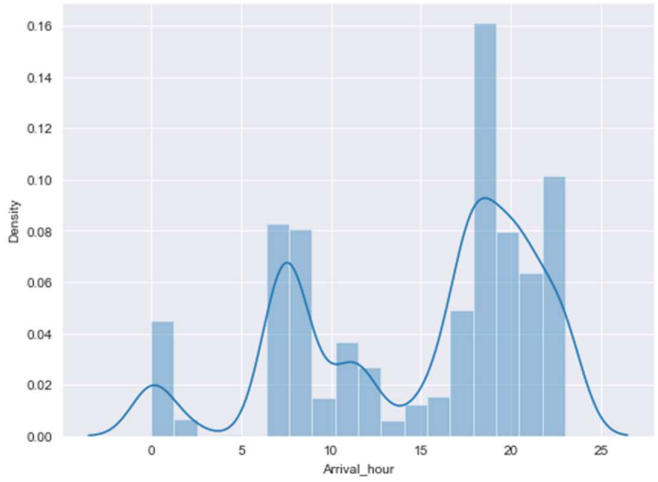


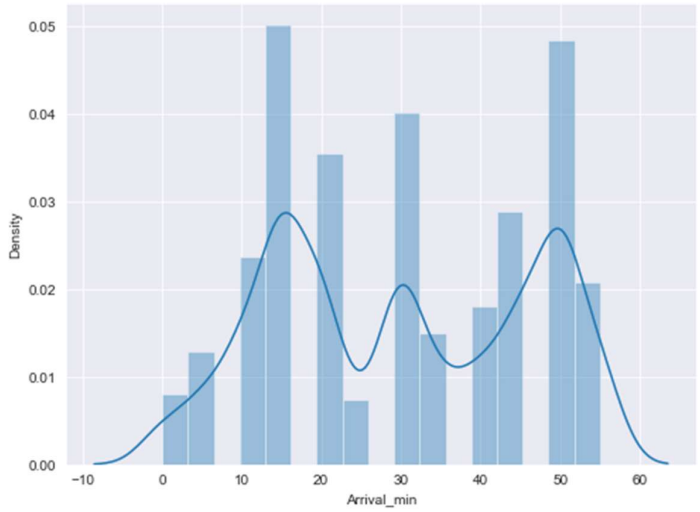

Density
0.10
0.08
0.06
0.04
0.02
0.00
−5
0
5
10
15
20
25
30
35
Duration_hours

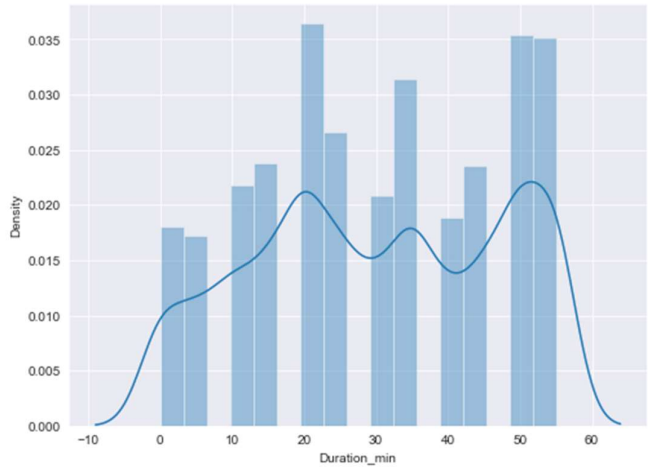

Density
6
5
4
3
2
1
0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Total_Stops


Density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
10
12
14
16
18
20
22
24
Dep_hour


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
−10
0
10
20
30
40
50
60
Dep_min


0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Density
0
5
10
15
20
25
Arrival_hour

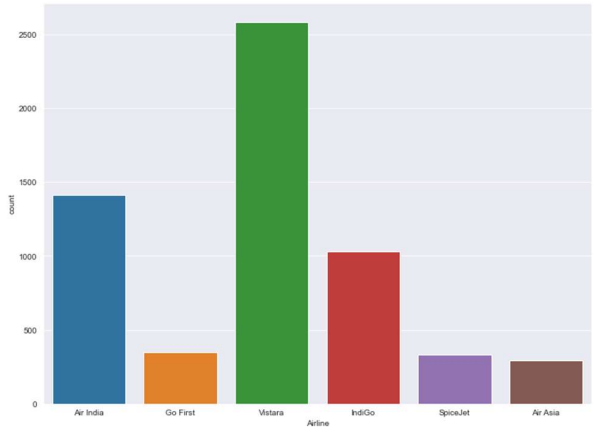

0.05
0.04
0.03
0.02
0.01
0.00
Density
−10
0
10
20
30
40
50
60
Arrival_min

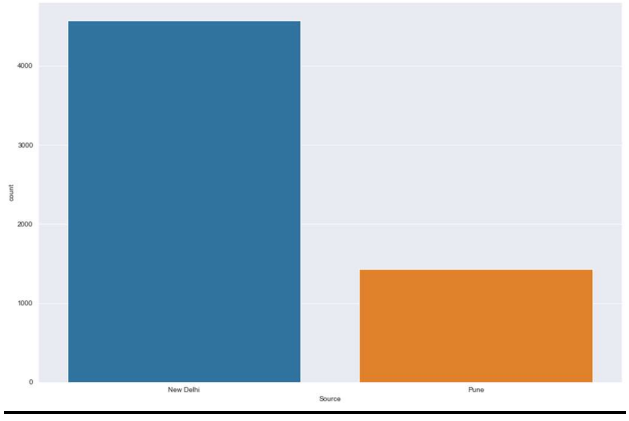


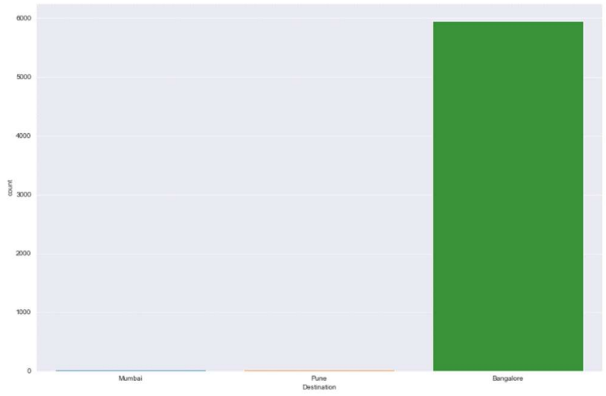
Observations:

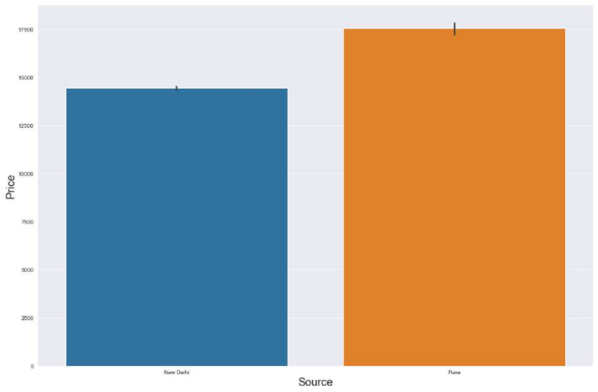
- From the distribution plot we can observe that most of the columns are somewhat distributed normally as they have no proper bell shape curve.
- "Price" is widely distributed between the range of 5000 - 25000.we can observe that the greatest number of tickets are priced at 15000 and 10000.
- The data in the column Arrival_Hour and Arrival_min skewed to left since the mean values is less than the median.

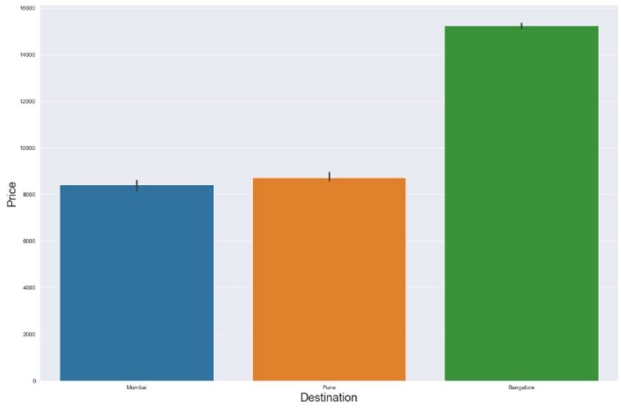
2. Visualization of categorical features with target:

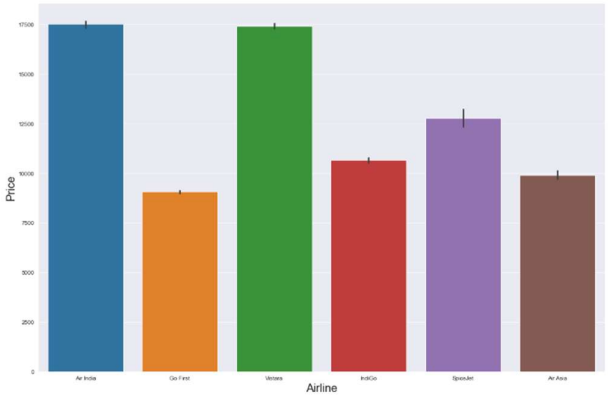


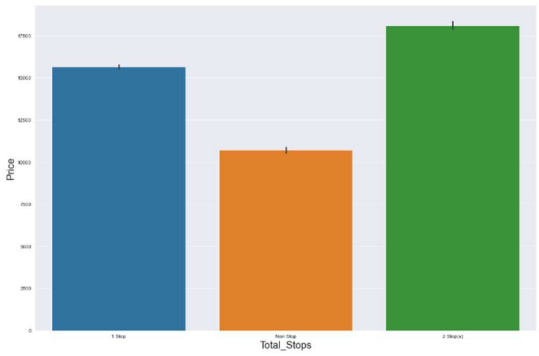


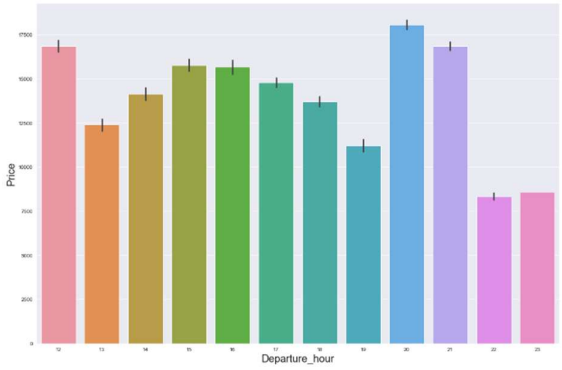

count
Mumbai
Pune
Bangalore
Destination

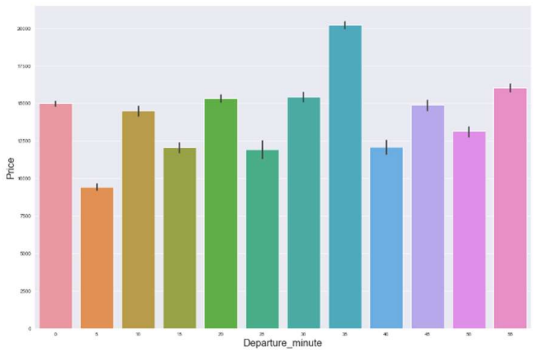

Price
New Delhi
Pune
Source

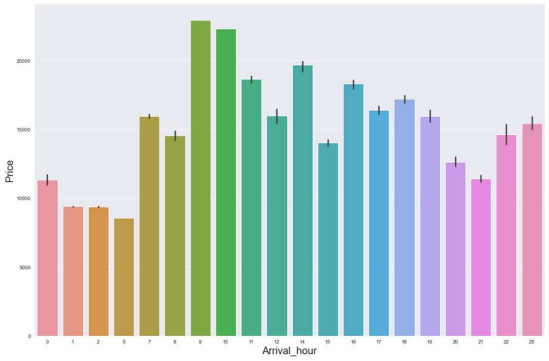

Price
Mumbai
Pune
Bangalore
Destination

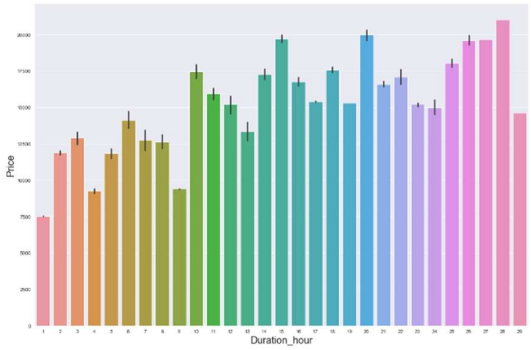

Price
Air India
Go First
Vistara
IndiGo
SpiceJet
Air Asia
Airline

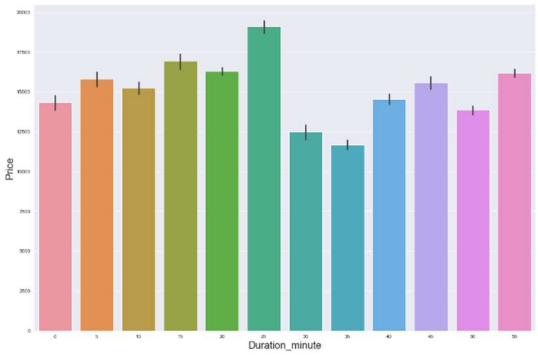

Price
1 Stop
Non Stop
2 Stop(s)
Total_Stops

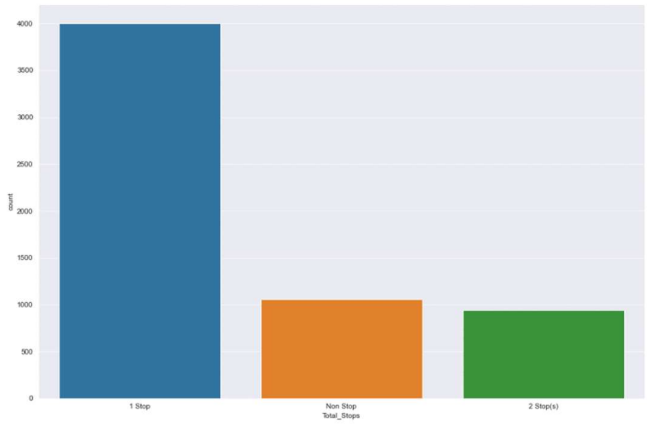

Price
12
13
14
15
16
17
18
19
20
21
22
23
Departure_hour











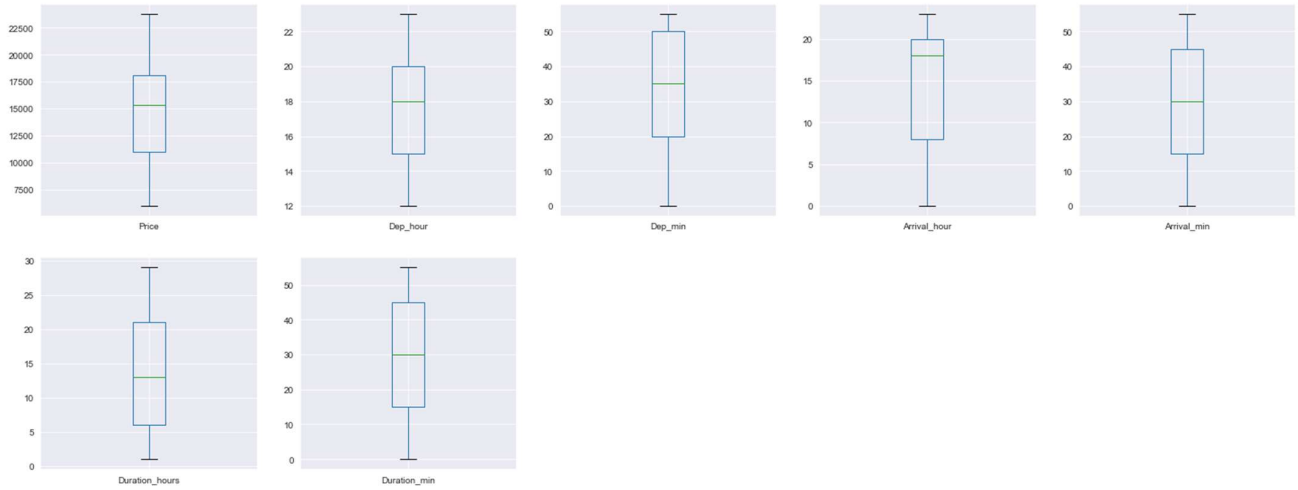
Observations:

- Vistara has largest share in market followed by AirIndia and Indigo.
- In our given dataset almost 76% of fight starts from New Delhi and rest from Pune
- In our given dataset almost 99% of fights destination is to Bangalore
- Ticket price of filghts starting from Pune is costlier than New Delhi
- Ticket price of filghts to Bangalore is costlier than mumbai and pune
- Ticket prices of AirIndia and Vistara are the costliest followed by SpiceJet,IndiGo,AirAsia and GoFirst
- Ticket prices are high for fights with 2 stops.NonStop flight ticket prices are low.
- "Departure_Hour vs Price": From the bar plot and line plot we can see that there are some flights departing in the noon 12 AM having most expensive ticket prices compared to late

evening flights. We can also observe the flight ticket prices are higher during evening (may fluctuate) and it decreases in the late night.

- Ticket prices are very high during 9-10am where as early morning flights have low prices.
- Long duration flights have higher ticket price and short duration flights have low ticket price

Boxplot



Observations:

There are no outliers in the data

Dep_hour has high multicollinearity, which I have check with VIF method too.so, i can drop this column.

- **Run and evaluate selected models**

1.Model building:

Ridge Regressor

```
R = Ridge()
R.fit(X_train,y_train)
```

```
Ridge()
```

```
R.score(X_train,y_train)
```

```
0.4685951990858602
```

```
pred_r = R.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_r))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_r))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_r)))
```

```
R2_SCORE: 0.46097941551020305
Mean_Squared_Error: 11049540.310245022
Root Mean_Squared_Error: 3324.084883128742
```

- Ridge regressor has given 46.09% accuracy.

DecisionTreeRegressor

```
DTR = DecisionTreeRegressor()
DTR.fit(X_train,y_train)
```

```
DecisionTreeRegressor()
```

```
DTR.score(X_train,y_train)
```

```
1.0
```

```
pred_dtr = DTR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_dtr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_dtr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_dtr)))
```

```
R2_SCORE: 0.9995971922114072
Mean_Squared_Error: 8257.274444444445
Root Mean_Squared_Error: 90.86954629822053
```

- Decision tree regressor has given 99.9% accuracy.

RandomForestRegressor

```
RFR = RandomForestRegressor()
RFR.fit(X_train,y_train)
```

```
RandomForestRegressor()
```

```
RFR.score(X_train,y_train)
```

```
0.9996439977513557
```

```
pred_rfr = RFR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_rfr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_rfr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_rfr)))
```

```
R2_SCORE: 0.9998614408424088
Mean_Squared_Error: 2840.3646191111116
Root Mean_Squared_Error: 53.29507124595211
```

- RandomForest regressor has given 99.9% accuracy.

XGBRegressor

```
XGR = XGBRegressor()
XGR.fit(X_train,y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)
```

```
XGR.score(X_train,y_train)
```

```
0.9999999995351071
```

```
pred_xgr = XGR.predict(X_test)
```

```
print('R2_SCORE:',r2_score(y_test,pred_xgr))
print('Mean_Squared_Error:',mean_squared_error(y_test,pred_xgr))
print('Root Mean_Squared_Error:',np.sqrt(mean_squared_error(y_test,pred_xgr)))
```

```
R2_SCORE: 0.999340998643385
Mean_Squared_Error: 13509.061182359723
Root Mean_Squared_Error: 116.2284869658025
```

- XGB regressor has given 99.9% accuracy.

ExtraTrees regressor

```
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: 0.9997489902913556
mean_squared_error: 5145.521291277779
mean_absolute_error: 4.691927777777779
root_mean_squared_error: 71.73228904250706
```

From the above created models,we can conclude that "Random Tree Regressor" as the best fitting model.

- ExtraTree regressor has given 99.9% accuracy.

    - By looking into the difference of model accuracy and cross validation score I found ExtraTrees Regressor as the best model.

2. Hyper Parameter Tunning:

Hyper Parameter Tuning:

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

```
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]}
```

```
GCV=GridSearchCV(ExtraTreesRegressor(),parameter,cv=5)
```

```
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]})
```

```
GCV.best_params_
```

```
{'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 4,
 'n_estimators': 20,
 'n_jobs': -2}
```

```
FlightPrice=ExtraTreesRegressor(max_features='sqrt',min_samples_leaf=1,min_samples_split=4,n_estimators=20,n_jobs=-2)
FlightPrice.fit(X_train,y_train)
pred=FlightPrice.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: 99.96076442994769
mean_squared_error: 8043.0140399691345
mean_absolute_error: 5.99649074074074
RMSE value: 89.68285254143701
```

- I have choosed all parameters of ExtraTreesRegressor, after tunning the model with best parameters, model accuracy from remained same at 99.9%

3. Saving the model and Predicting Ticket Price for test data:

Predicting the saved model

```
# Loading the saved model
model=joblib.load("Prediction_of_FlightPrice.pkl")

#Prediction
prediction = model.predict(x_test)
prediction
```

```
array([ 9419., 15615.,  9419., ..., 11940., 15571.,  7938.])
```

- I have saved my best model using .pkl as follows.
- Now loading my saved model and predicting the test values.

Predicting the saved model

```
# Loading the saved model
model=joblib.load("Prediction_of_FlightPrice.pkl")

#Prediction
prediction = model.predict(x_test)
prediction
```

```
array([ 9419., 15615.,  9419., ..., 11940., 15571.,  7938.])
```

```
pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Original"]).T
```

	Predicted	Original
0	9419.0	23688.0
1	15615.0	16873.0
2	9419.0	17400.0
3	17913.0	12664.0

- Plotting Actual vs Predicted values.

Predicting fight ticket Price for test dataset:

- I have predicted the Price to save model . I have also saved my predictions for further analysis.

- **Interpretation of the Results:**

  - The dataset was scrapped from yatra website.
  - The dataset was very challenging to handle it had 10 features with 5997 samples.
  - There are no null values in the dataset.
  - 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.
  - Ploting of different types 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 dist plot,pairplot and bar plot to see the relation between target and features.
  - I did not find any outliers or skewness in the dataset.
  - Then scaling 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 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 ExtraTreesRegressor as the best model with 99.9% r2_score. Also I checked the accuracy of the best model by running hyper parameter tunning.
  - At last, I have predicted the used flight price using saved model.

4.CONCLUSION

- **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 analyse 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 tunning 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 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 yatra.com 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.

- **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, 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.

5.REFERENCE

www.yatra.com