FLIP ROBO

PROJECT REPORT ON:

"Flight Price Prediction Project"

SUBMITTED BY

RAHUL M

ACKNOWLEDGMENT

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with this dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Shwetank Mishra (SME Flip Robo), is the person who has guided this Project.

A huge thanks to “Data trained” as well.

Contents:

1.INTRODUCTION

1.1 Business Problem Framing:

The tourism industry is changing fast and this is attracting a lot more travelers 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, 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 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.

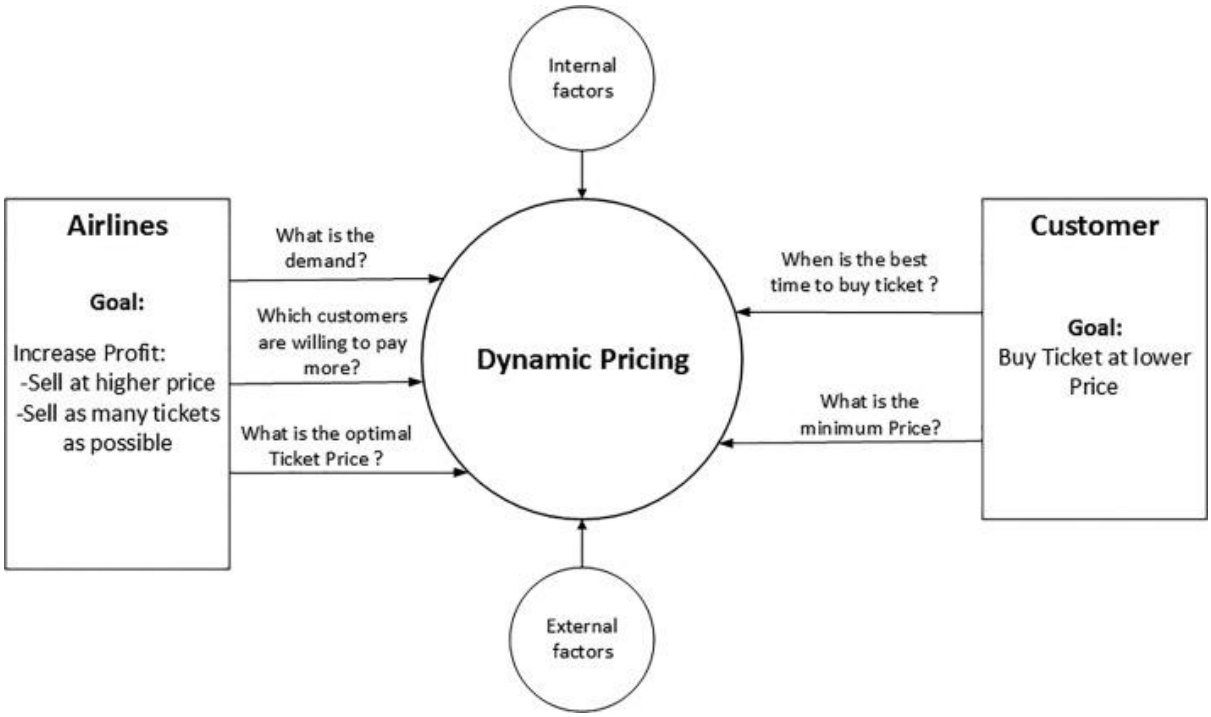
1.2 Conceptual Background of the Domain Problem

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 **80.3% 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.



1.3 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 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

2.1 Mathematical/ Analytical Modeling of the Problem

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

In this perticular 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 no null values in the dataset. To get better insight on the features I have used ploting like distribution plot, bar plot, strip plot and count plot. With these ploting 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 tunned 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 9095 rows and columns including target. In this p erticular 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.
- Journey_date: The date of the journey
- From: The source from which the service begins.
- Destination: The destination where the service ends.
- Dep_Time: The time when the journey starts from the source.
- Arrival_TimeTime of arrival at the destination.
- Total_Stops: Total stops between the source and destination.
- Price: The price of the ticket

2.3 Data Preprocessing Done

- ✓ As a first step I have scrapped the required data using selenium from makemytrip website.
- ✓ And I have imported required libraries and I have imported the dataset which was in csv format.
- ✓ I have droped Unnamed:0 column as I found it was the index column of csv file.
- ✓ Then I did all the statistical analysis like checking shape, nunique, value counts, info etc…..

- ✓ While checking for null values I found there was multiple rows in the dataset with '-' and I dropped that row as it will not help our analysis.
- ✓ 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.

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 bar plot for each pair of categorical features that shows the relation between target and independent features.
- ✓ I have used strip plot to see the relation between numerical columns and target column.
- ✓ I can notice there is a good relationship between maximum columns and target.

2.5 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 — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1.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 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 AdaBoostRegressor
- ✓ from xgboost import XGBRegressor
- ✓ from sklearn.ensemble import GradientBoostingRegressor
- ✓ from sklearn.ensemble import ExtraTreesRegressor
- ✓ from sklearn.neighbors import KNeighborsRegressor as KNN
- ✓ from sklearn.linear_model import LinearRegression
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.metrics import accuracy_score
- ✓ from sklearn.model_selection import cross_val_score

and also Datetime and pickle, 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. there was outliers in duration and skewness in duration_hour, duration_minutes the dataset. So removed them with the outliers using Zscore function and skewness is removed using PowerTransformer. 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. And checking the feature score as well Then followed by model building with all Regression algorithms.

3.2 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 XGBRegressor as the best model with least difference between r2_score and cross validation score. 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.

- LinearRegression
- RandomForestRegressor
- DecisionTreeRegressor
- XGBRegressor
- ExtraTreesRegressor
- GradientBoostingRegressor
- KNeighborsRegressor

3.3 Key Metrics for success in solving problem under consideration

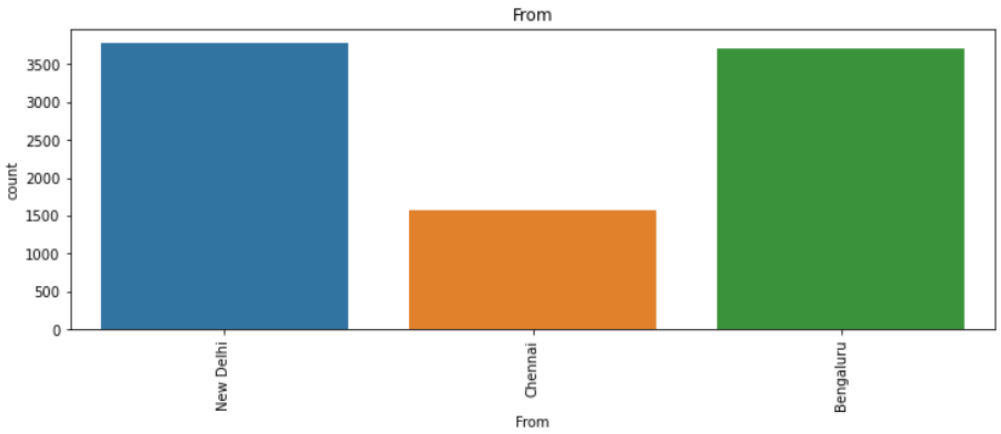
I have used the following metrics for evaluation:

- 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

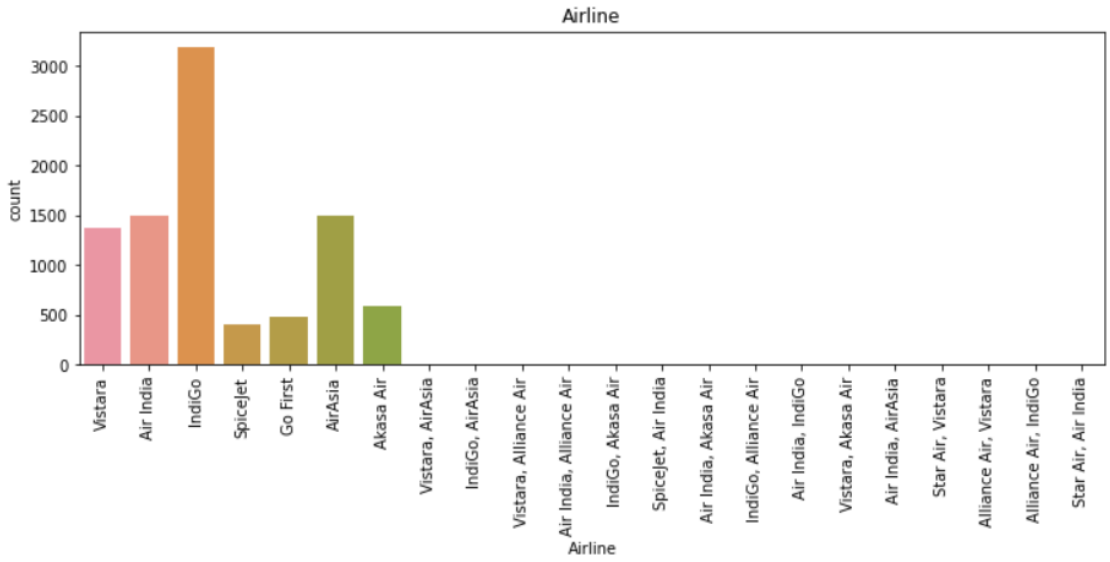
I have used bar plots to see the relation of categorical feature with target and I have used 2 types of plots for numerical columns one is distribution plot for univariate and strip plot for bivariate analysis.

1. Univariate Analysis for Categorical columns:



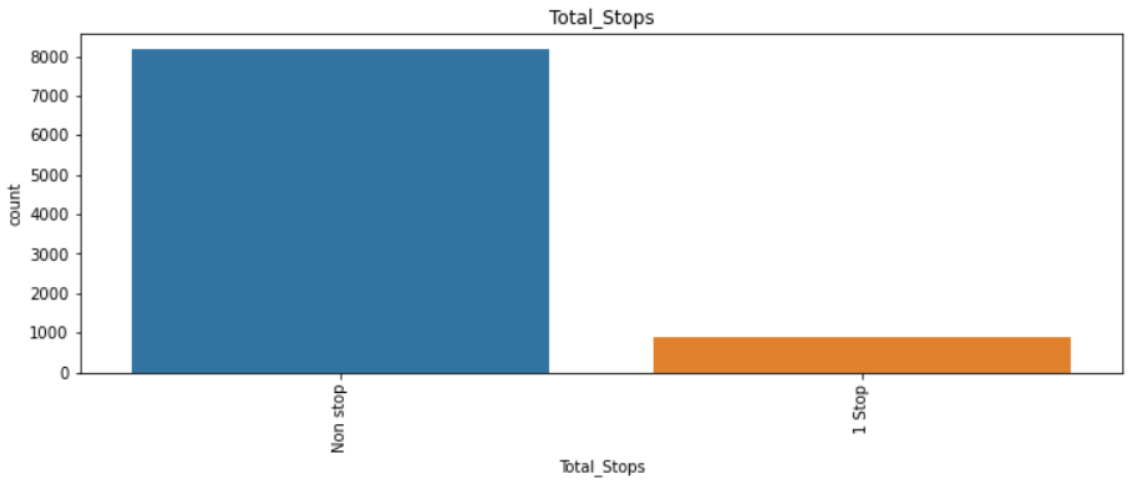
Observations:

- ✓ New Delhi has 3777 flights in the dataset
- ✓ Bengaluru has 3708 flights in the dataset
- ✓ Chennai has 1579 flights in the dataset.



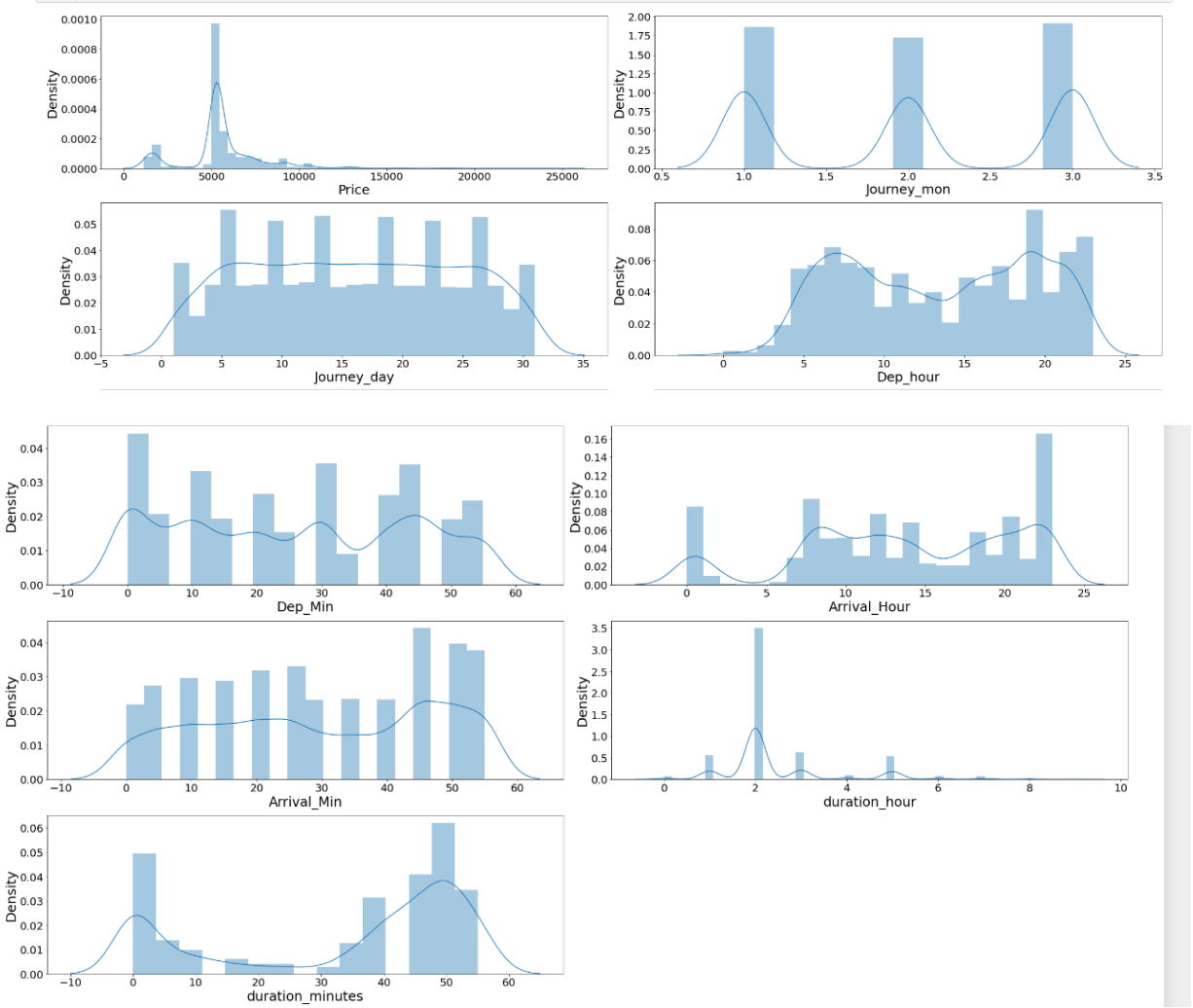
The airlines

- ✓ IndiGo has 3187 flights in the dataset
- ✓ AirAsia has 1503 in the dataset
- ✓ Air India has 1500 in the dataset
- ✓ Vistara has 1368 in the dataset
- ✓ Akasa Air has 585 in the dataset
- ✓ Go First has 476 in the dataset
- ✓ SpiceJet has 407 in the dataset
- ✓ Alliance Air, Vistara has 11 in the dataset
- ✓ IndiGo, AirAsia has 6 in the dataset
- ✓ rest of the fights are having less than 4 flights



- ✓ the nonstop flights with has 8166 flights in the dataset
- ✓ Bengaluru has 898 flights flights with has 898 flights in the dataset

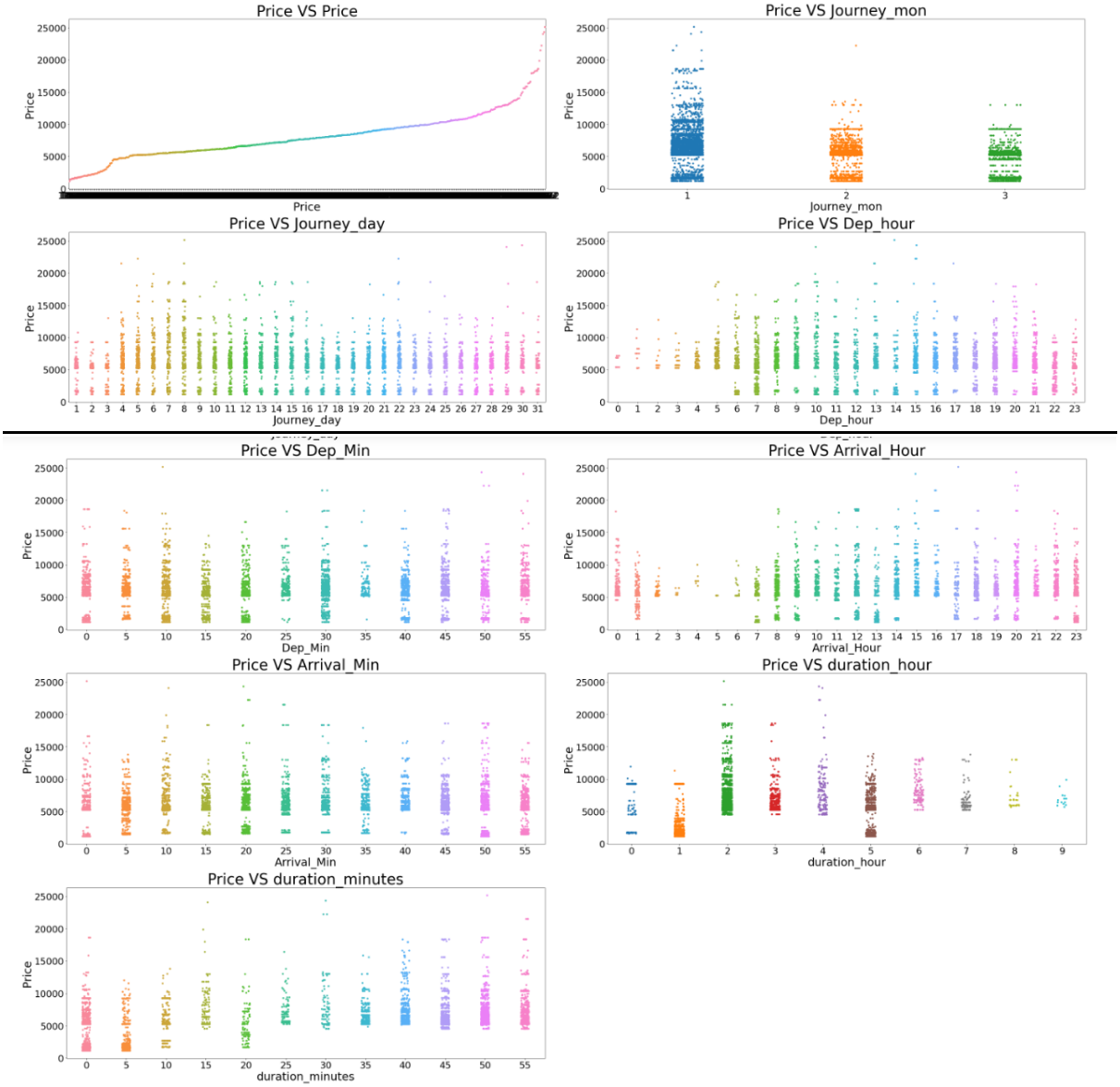
2. Univariate analysis for Numerical column:



Observations:

- ✓ There is skewness in duration minutes and duration hour in the dataset in the numerical columns.

3. Bivariate analysis for numerical columns:



Observations In Price Vs journey month:

- ✓ we are able to observe Higher prices in the initial months and reduction in prices as the months increases.
- ✓ The price is increasing near to 25000 in the month of january and
- ✓ we can observe price reduction in month of feb and march.
- ✓ The difference in the prices in Feb and march are Not big in values.
- ✓ The difference in the prices in Feb and march are Not big in values.

Observations In Price Vs Journey day

- ✓ We can observe higher prices in some days in the month and out of 31 days we can observe prices
- ✓ reaching 10000 and more in some days and upto 25000.
- ✓ in journey day 8 we can see the value reaching upto 25000 and near to these price in 29th and 30 of the months as well.

Observations In Price Vs Departure hour

- ✓ we can observe least price increment in the hour zero in the dataset and
- ✓ in hour 10 and 14 and 15 it is reaching near to peak of 25000 as well.
- ✓ From 6th hour to 23 we can observe the prices varying from min to max in the each hours.
- ✓ We can observe less increase in the price in morning flights compared to flights past the morning hours

Observations In Price Vs Departure minute

- ✓ we can observe in the 10th and 5oth and 55th min the prices are reaching till the max values for flights upto 25000.
- ✓ We can observe high data points in the price range of 5000 and 12000 for flights in the 5min intervals.

Observations In Price Vs Arrival hour

- ✓ when the arrival hour is at hour zero,1,2,3,4,5,10,15,16,21 we can see higher flight prices for all flights in that day.
- ✓ Except these hours we can observe lowest flight price and the price reaching upto max as well in the flight hours other than these as well
- ✓ in 15th 17,20th hours we can observe the price reaching to highest for flights

Observations In Price Vs Arrival Minutes

- ✓ Except the 5th min the flight prices are going Above 15000 and reaching 25000 as well for these minutes as well.

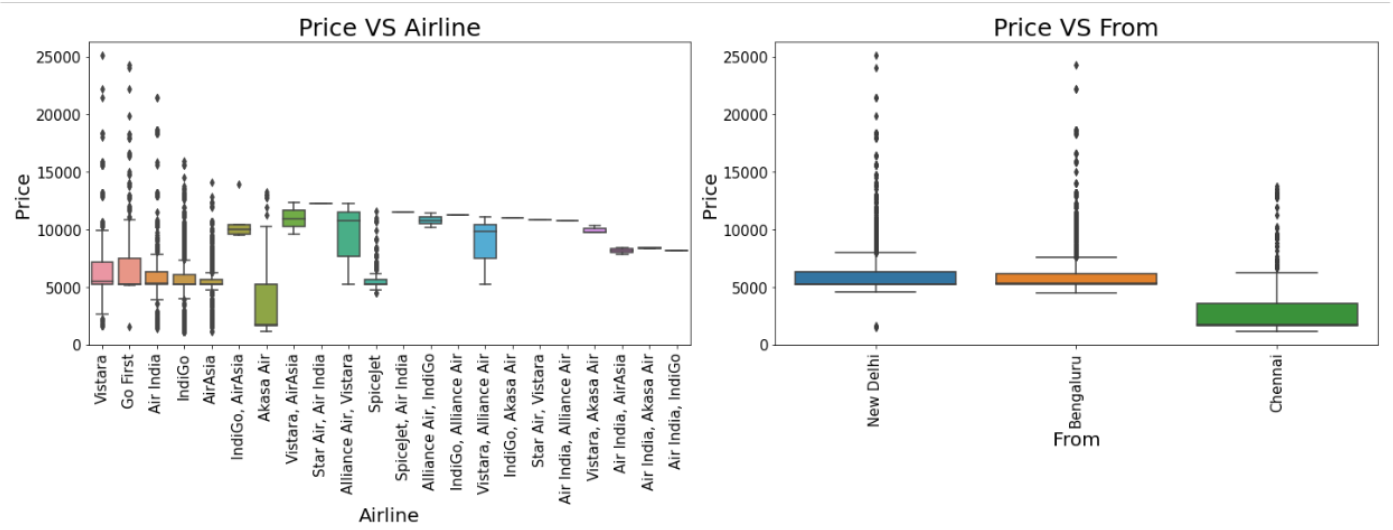
Observations In Price Vs Duration hours

- ✓ we can observe in the duration less than 2hours and more than 4 and less than 5 has the flight prices starting from least
- ✓ We can observe the highest price increment in the duration hour of 2 and 4 has the highest for the flights in the datset.
- ✓ when the duration is 9 we can see least increment in the prices for flights.

Observations In Price Vs Duration Minutes

- ✓ we can observe highest prices in the duration minute is at 15 and 30 and 50 for the flights.
- ✓ the prices for flights start at the very least in the duration of Zero and 5,10,20 they have the flughts with least prices.

✓ Bivariate Analysis for categorical columns:



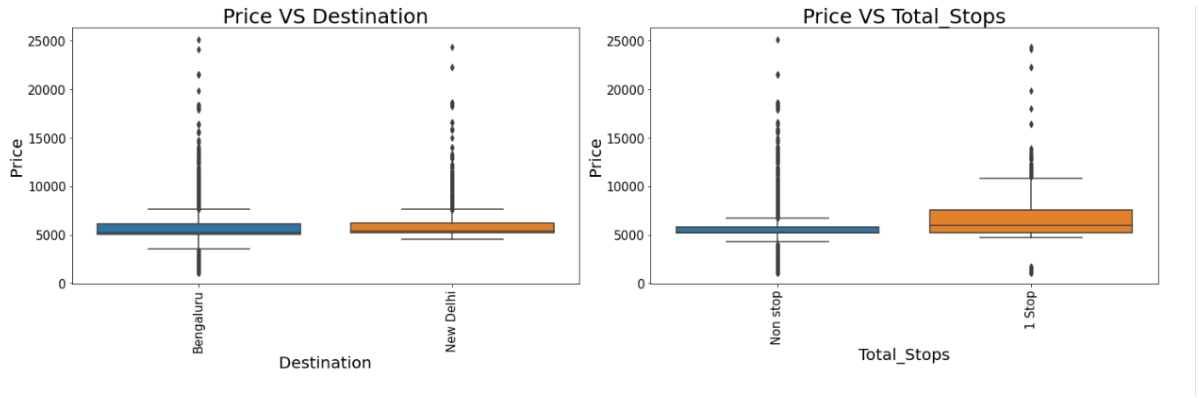
Observations In Price Vs Airlines:

- ✓ We can observe the Airlines Vistara,Go first ,Air India having the highest price increment for the flights in price.
- ✓ we can observe the spice jet being cheaper than indigo for flight price in the Airlines.
- ✓ we can observe the airlines vistara,Go first ,Air india,Indigo,Airasia,Akash Air having fligt prices starting at the very least.

Observations In Price Vs From:

- ✓ we can observe the flight prices are cheaper in chennai as the boarding station and
- ✓ Highest in new delhi as the boarding station starting with smaller prices with few flights as well.
- ✓ bangalore is also having high prices for flights as well reaching near to 25000 as well.

- ✓ we can observe the flight prices of chennai as the boarding station only reaching below 15000 increment



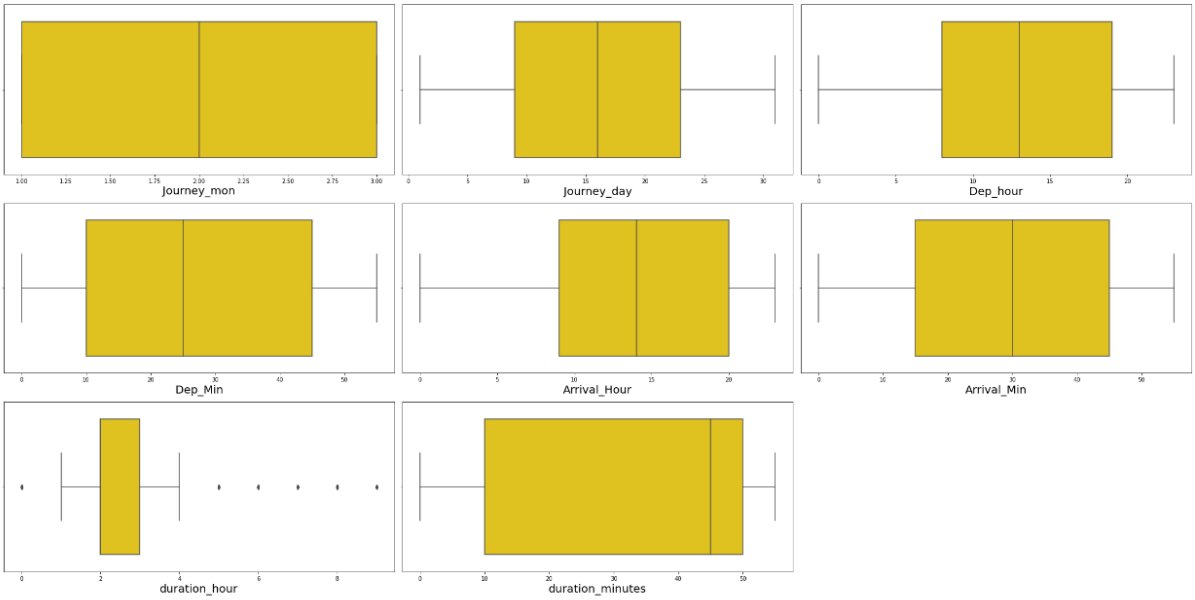
Observations In Price Vs Destination

- ✓ When the destination is banglore we can obsevre the highest price range reaching to 25000 and
- ✓ starting with smaller prices for flights as well compared to new delhi as the Destination.

Observations In Price Vs Destination

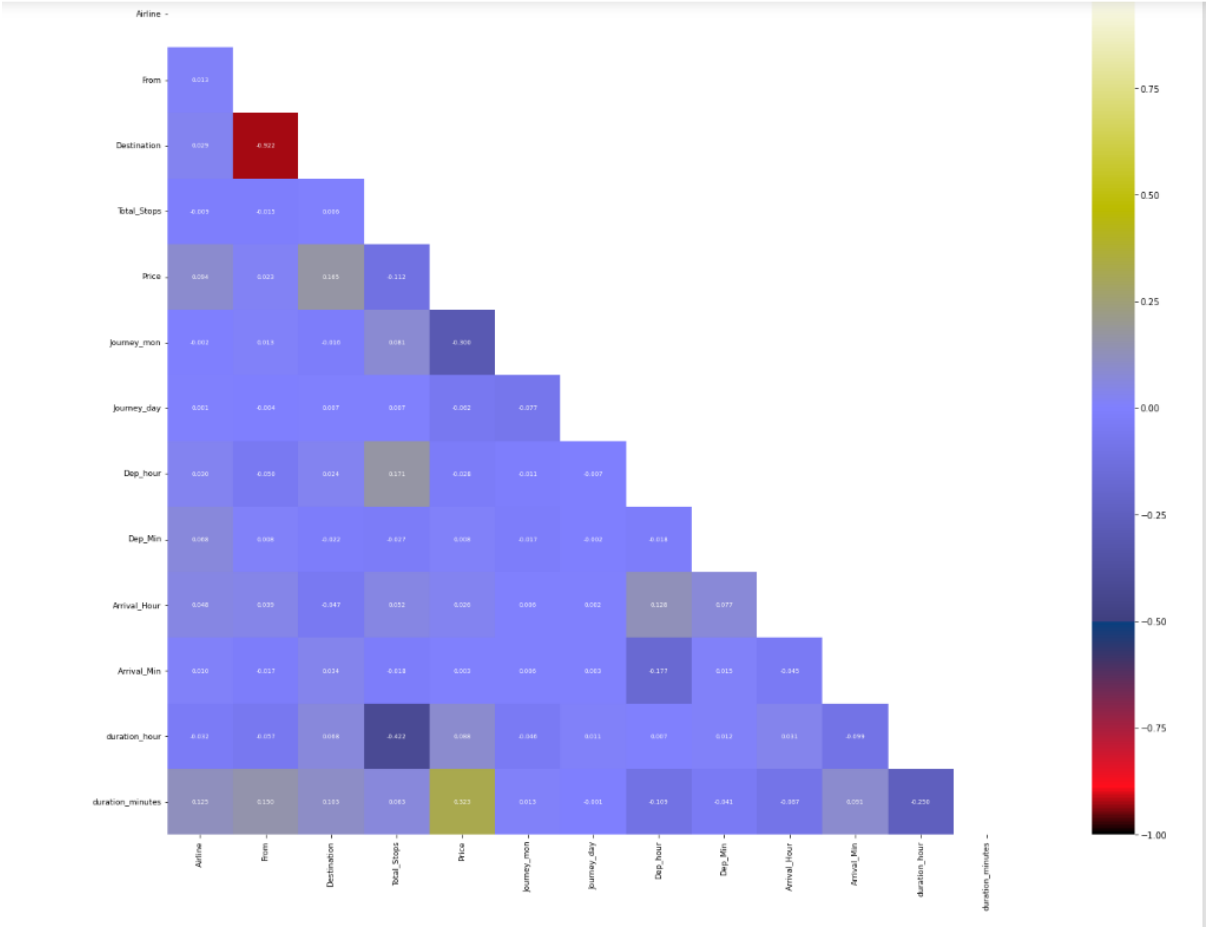
- ✓ we can observe the flight prices for non stop increasing from the very least and max price for flights as well.
- ✓ We ca observe fewe flight having price above 20000 in Non stop flights and
- ✓ for the flights having single stops have few flights having price greater than 15000 in the datset.

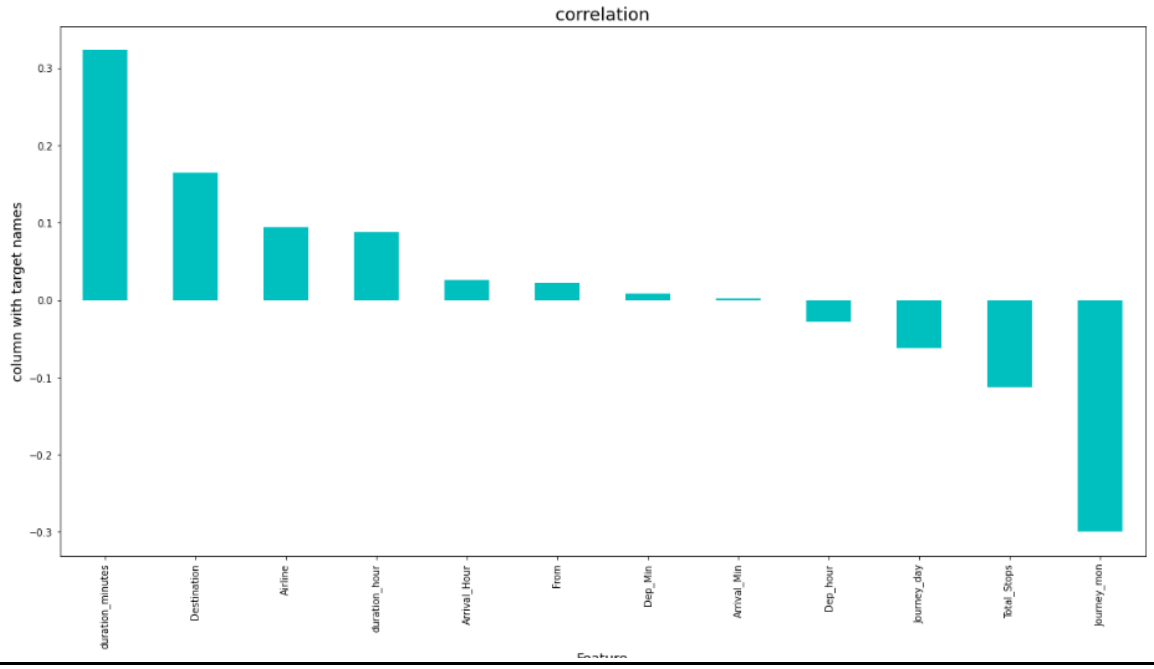
Checking Outliers



There are outliers in the column duration hour in the dataset

Heatmap


Airline
From
Destination
Total_Stops
Price
Journey_mon
Journey_day
Dep_hour
Dep_Min
Arrival_Hour
Arrival_Min
duration_hour
duration_minutes
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


correlation
column with target names
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
duration_minutes
Destination
Airline
duration_hour
Arrival_Hour
From
Dep_Min
Arrival_Min
Dep_hour
Journey_day
Total_Stops
Journey_mon

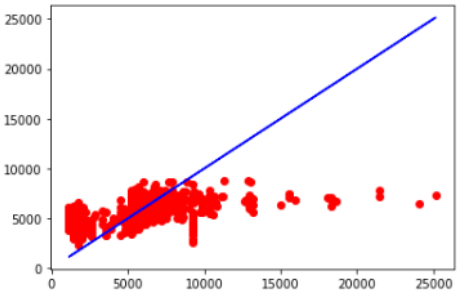
feature selection

5. Run and Evaluate selected models

1. Model Building:

1) LinearRegression:

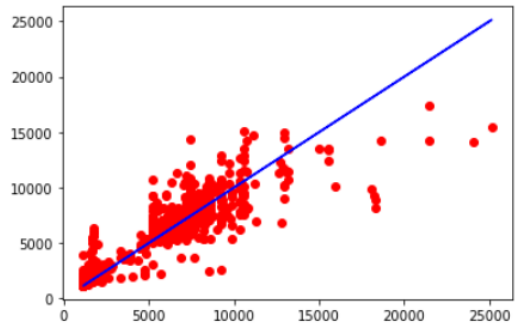
```
LinearRegression()
r2_score for train data is 20.81%


r2_score for  test data is  26.79%


Error
mean absolute error :  1272.1115654835892
mean squared error :  4010027.756134298
mean squared error is:  2002.5053698140982

 LinearRegression() Cross val score is  [-0.25019991 -1.43895655 -0.96504014 -0.27203274 -5.33806513]
mean is  165.2858895233469


difference b/w accuracy and crossval score is 138.4937067807673
```



LinearRegression model has given me 26.79% r2_score, we have to look into multiple models

2) RandomForestRegressor:

```
RandomForestRegressor()
r2_score for train data is 96.57%


r2_score for  test data is  78.10%


Error
mean absolute error :  482.05616183265226
mean squared error :  1199326.3803484843
mean squared error is:  1095.1376079509298

 RandomForestRegressor() Cross val score is  [ 0.22444536 -0.20344781 -1.19766795  0.08834865 -0.39020975]
mean is  42.082390466675875


difference b/w accuracy and crossval score is 36.02248266038812
```



- RandomForestRegressor has given me 78.10% r2_score, but still we have to look into multiple models.

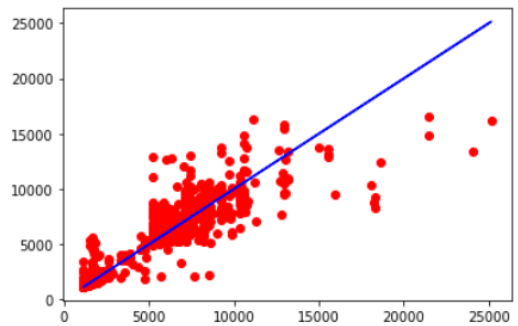
3) XGBRegressor:

```
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, ...)
r2_score for train data is 94.03%


r2_score for  test data is  79.13%


Error
mean absolute error :  514.6221337952242
mean squared error :  1142936.2275278876
mean squared error is:  1069.0819554776367

 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, ...) Cross val score is  [ 0.10550326 -0.76305382 -0.88954762  0.1575655  -2.11419324]
mean is  80.5972688892975


difference b/w accuracy and crossval score is 1.4629265780132101
```

- XGBRegressor is giving me 79.13% r2_score.

4) ExtraTreesRegressor:

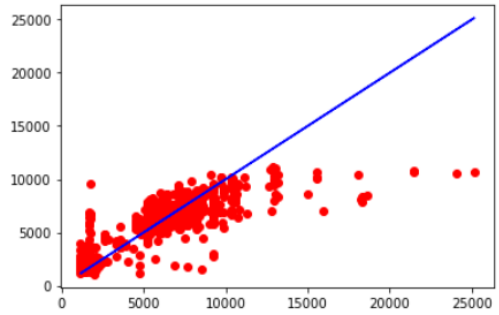
```
ExtraTreesRegressor()
r2_score for train data is 99.98%


r2_score for  test data is  78.17%


Error
mean absolute error :  456.1042484576556
mean squared error :  1195601.741763923
mean squared error is:  1093.4357510909927

 ExtraTreesRegressor() Cross val score is  [ 0.11404642  0.10890283 -1.21695118  0.10040774 -1.60011067]
mean is  62.80837670702306


difference b/w accuracy and crossval score is 15.364494119166949
```



- ExtraTreesRegressor is giving me 78.17% r2_score.

5) GradientBoostingRegressor:

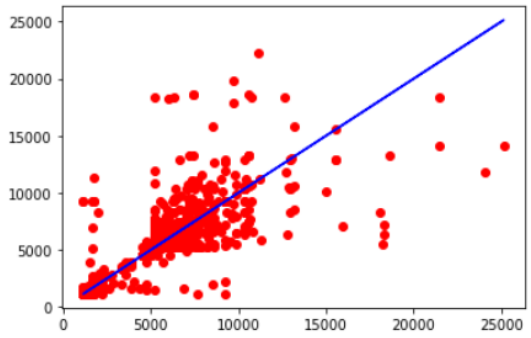
```
GradientBoostingRegressor()
r2_score for train data is 67.78%


r2_score for  test data is  66.26%


Error
mean absolute error :  696.6818496911934
mean squared error :  1848115.4169790715
mean squared error is:  1359.4540878525731

 GradientBoostingRegressor() Cross val score is  [-0.04072451 -0.32068103 -0.8424366  -0.19517848 -0.69618616]
mean is  41.90413570759179


difference b/w accuracy and crossval score is 24.356323335401925
```



- GradientBoostingRegressor is giving me 66.26% r2_score.

6) DecisionTreeRegressor:

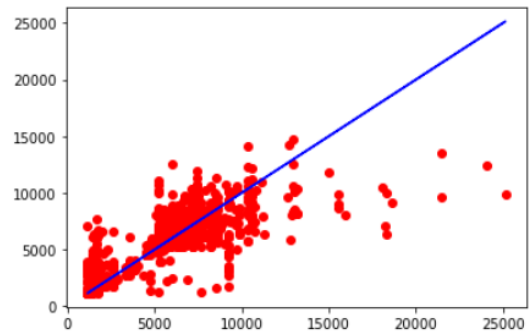
```
DecisionTreeRegressor()
r2_score for train data is 99.98%


r2_score for  test data is  53.29%


Error
mean absolute error :  598.3480089736399
mean squared error :  2558328.6468031406
mean squared error is:  1599.4776168496828

 DecisionTreeRegressor() Cross val score is  [-0.35664533 -0.86566627 -1.3857023  -0.07541639 -2.18621924]
mean is  97.39299072887665


difference b/w accuracy and crossval score is 44.098317337422834
```



- DecisionTreeRegressor is giving me 53.29% r2_score

7) KNN:

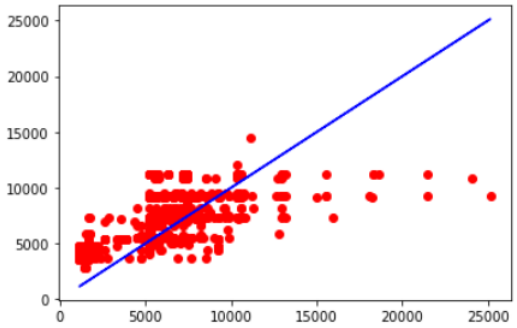
```
KNeighborsRegressor()
r2_score for train data is 72.73%


r2_score for  test data is  59.91%


Error
mean absolute error :  679.8720134604599
mean squared error :  2195732.7503533373
mean squared error is:  1481.8005096345923

 KNeighborsRegressor() Cross val score is  [-0.47871595 -1.51434729 -0.97842846 -0.2404914  -3.7080772 ]
mean is  138.4012059171416


difference b/w accuracy and crossval score is 78.48691397442437
```



- KNN is giving me 59.91% r2_score.

8) AdaboostRegressor:

```
AdaBoostRegressor()
r2_score for train data is 31.72%


r2_score for  test data is  32.74%


Error
mean absolute error :  1233.2182197511122
mean squared error :  3684438.731913629
mean squared error is:  1919.4891851515156

 AdaBoostRegressor() Cross val score is  [ -0.25853703 -15.91818394  -2.3497211    0.0266846  -11.19757406]
mean is  595.0140147216974


difference b/w accuracy and crossval score is 562.2778178137031
```



- AdaboostRegressor is giving me 32.74% r2_score.

✓ **By looking into the model r2_score and error I found XGBRegressor as the best model with highest r2_score and least difference between cross validation and accuracy.**

4. <u>Hyper Parameter Tunning:</u>

- ✓ **booster: Select the type of model to run at each iteration**
- ✓ **gbtree: tree-based models**
- ✓ **gblinear: linear models**
- ✓ **nthread: default to maximum number of threads available if not set**
- ✓ **objective: This defines the loss function to be minimized**
- ✓ **Parameters for controlling speed**
- ✓ **subsample: Denotes the fraction of observations to be randomly samples for each tree**
- ✓ **colsample_bytree: Subsample ratio of columns when constructing each tree.**
- ✓ **n_estimators: Number of trees to fit.**
- ✓ **Important parameters which control overfiting**

- ✓ **learning_rate: Makes the model more robust by shrinking the weights on each step**
- ✓ **max_depth: The maximum depth of a tree.**
- ✓ **min_child_weight: Defines the minimum sum of weights of all observations required in a child.**

```
#hypertuning parameters
param_tuning = {
        'learning_rate': [0.001, 0.01,0.1,1],
        'max_depth': [3, 5, 7, 10],
        'min_child_weight': [1, 3, 5],
        'subsample': [0.5,0.7],
        'colsample_bytree': [0.5, 0.7],
        'n_estimators' : [100, 200, 300,400],
        'objective': ['reg:squarederror']
    }
```

```
#importing gridsearch
from sklearn.model_selection import GridSearchCV

Grid_sear= GridSearchCV(estimator=xgbr,param_grid=param_tuning,cv= 5,n_jobs=-1)
```

```
#training the model with paramters
Grid_sear.fit(x_train,y_train)
```

```
GridSearchCV(cv=5,
             estimator=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...
                                    monotone_constraints='()', n_estimators=100,
                                    n_jobs=0, num_parallel_tree=1,
                                    predictor='auto', random_state=0,
                                    reg_alpha=0, reg_lambda=1, ...),
             n_jobs=-1,
             param_grid={'colsample_bytree': [0.5, 0.7],
                         'learning_rate': [0.001, 0.01, 0.1, 1],
                         'max_depth': [3, 5, 7, 10],
                         'min_child_weight': [1, 3, 5],
                         'n_estimators': [100, 200, 300, 400],
                         'objective': ['reg:squarederror'],
                         'subsample': [0.5, 0.7]})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
[136]: #Best parameteters in the hypertuned model
       Grid_sear.best_params_
```

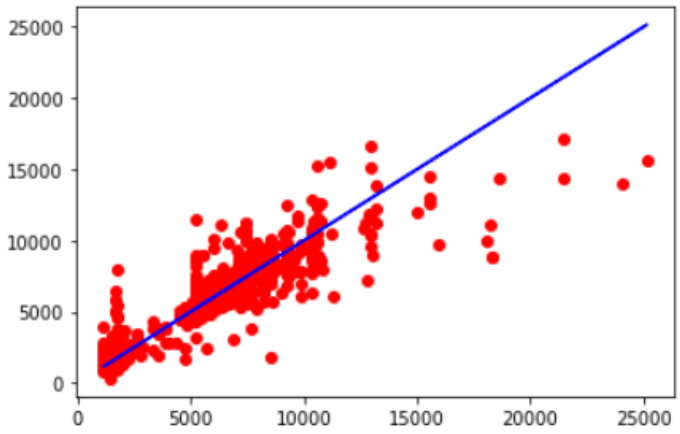
```
t[136]: {'colsample_bytree': 0.7,
         'learning_rate': 0.1,
         'max_depth': 7,
         'min_child_weight': 1,
         'n_estimators': 200,
         'objective': 'reg:squarederror',
         'subsample': 0.7}
```

```
[137]: #training the model with gridsearch parameters

       h_xgbr = XGBRegressor(learning_rate=0.1 , max_depth= 7, min_child_weight= 1, subsample= 0.7,
                             colsample_bytree= .7,n_estimators=200,objective='reg:squarederror')

```

```
#training the model with gridsearch parameters

h_xgbr = XGBRegressor(learning_rate=0.1 , max_depth= 7, min_child_weight= 1, subsample= 0.7,
                              colsample_bytree= .7,n_estimators=200,objective='reg:squarederror')

```

```
r2_score for train data is 95.11%


r2_score for  test data is  80.30%


Error
mean absolute error :  499.10857322466086
mean squared error :  1079226.9681460583
mean squared error is:  1038.858492840126
```

```
]: [<matplotlib.lines.Line2D at 0x28f34940280>]
```



- **I have choosed all parameters of XGBRegressor, after tunning the model with best parameters I have incresed my model accuracy from 79.13% to 80.30%.**

5. 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.

Predicting Flight Price for test dataset using Saved model of trained dataset:

In [150]:
```
# Loading the saved model
model=joblib.load("Flight_Price.pkl")

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

Out[150]:
```
array([6458.977 , 5330.366 , 5665.5713, ..., 4367.1304, 5144.3887,
       7335.446 ], dtype=float32)
```

In [154]:
```
pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[154]:

	0	1	2	3	4	5	6	7	8	9	10
Predicted	6458.977051	5330.366211	5665.571289	5962.687012	1343.678223	5210.253418	5951.603516	5155.138184	5192.196777	5325.219238	7832.333008	82
Actual	5236.000000	5250.000000	5482.000000	6617.000000	1454.000000	5354.000000	5236.000000	5237.000000	5237.000000	5482.000000	8113.000000	82

3.6 Interpretation of the Results

- ✓ The dataset was scrapped from makemytrip website.
- ✓ The dataset was very challenging to handle it had 9 features with 9095 samples.
- ✓ the datasets was having '-' values in some columns, so I have dropped that rows.
- ✓ 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 ploting 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 choosen strip plot and bar plot to see the relation between target and features.
- ✓ Removed outliers using Zscore method and skewness removed using PowerTransformer in the dataset.
- ✓ Encoded the data which are categorical columns 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 XGBRegressor as the best model with 80.30% 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 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 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 makemytrip 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. Data cleaning is one of the most important steps to remove unrealistic values and null values. This study is an exploratory attempt to use eight 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.

4.3 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, 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.