Airline Ticket Price Prediction System

Department of Computer Science & Engineering
The LNM Institute of Information Technology, Jaipur

Group Members:-

19ucs090 Niket Mittal
19ucs110 Divyansh Bhadauria
19ucs111 Rohan Singh Chauhan
19ucs165 Vinayak Singh
19ucs182 Sagar Mittal

Abstract

Flight ticket prices are highly dynamic and its the main source of revenue for airlines thus prices are increased or decreased based on the revenue management and on the demand over a period. Each carrier has its own proprietary rules to set prices accordingly. Advancement in the field of Machine Learning (ML) and Artificial Intelligence (AI) has made it possible to infer condition rules and model it to price variations. To estimate the minimum possible airfare, a dataset with features like arrival time and departure time, flight time, date, month, etc. are taken under consideration for the Machine Learning (ML) model.

1. Introduction

The flight ticket booking system to purchase a ticket many days prior to flight takeoff so as to stay away from the extreme charges. Most aviation organizations don't follow this procedure anymore.These organizations may diminish the cost accordingly with time,demand,for market build up,etc.This change leads to the cost of ticket being highly dynamic.Customer usually tries to buy the ticket well in advance to prevent any extra hike but nowadays this might lead customer to wind up more charges then what they should had for the same seat. Most studies on airfare prediction are based on national level or focused towards a specific market , research at market and airport level is still limited. Earlier models of predicting the airfare upto a good approximation were based upon heuristic-based statistical models.Thus , we address this problem by inferring rules based on the defining parameters uncovering the hidden relationship among different features , further using this data on several machine learning frameworks to predict the airfare with r-squared value to achieve high prediction accuracy.

2. Task Definition

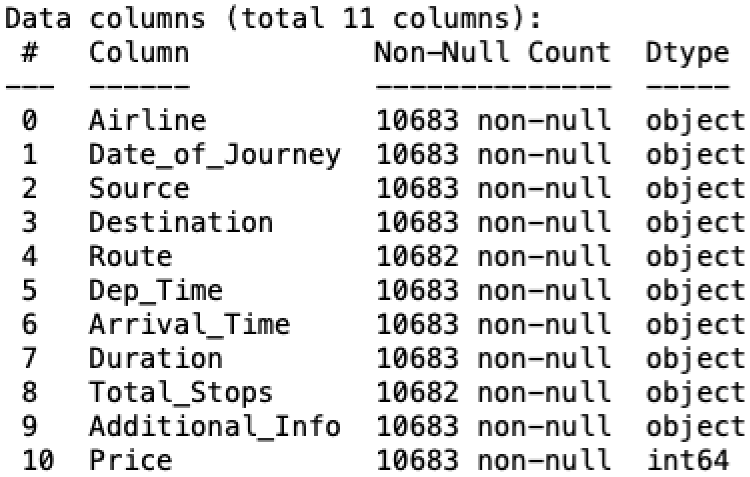
Our main task is to use the dataset of past flight prices to predict flight prices and then test them against actual flight prices.In order to so we shall implement different algorithms in our software to accurately predict flight prices.The first task is to read the data from the csv file which is basically a file in which data is present.The data is organised using several attributes hence our next task is to segregate it into different variables start the process of learning.During this process it becomes important to remove the values in which certain attributes are null or incorrect.Further we will plot the data using certain libraries so that data analysis can be performed.Bringing the data to a standard scale since there exist some column with low values and some with very high values therefore bringing them to a common scale will make learning easy for the model.After we successfully complete these tasks our model is ready for training.For this purpose we must apply the correct libraries with suitable splitting of data for training and testing purposes.The most important task here is to obtain high accuracy and for this purpose we must try out different approaches.Lastly we must plot the graph of predicted vs actual flight prices and evaluate the accuracy of our software.

3. Infrastructure

To build this project we have used python language.For statistical and numerical analysis , best considered libraries numpy,pandas are used.To plot the graphs for a better understanding we have used matplot library.Since , we are using a large dataset thus to get the optimum results we are using LazyPredict library which is one of the best library that can help to semi-automate machine learning tasks , with the help of this library we were able to get the the performance measure of multiple regression models based on the multiple performance parameters.Thus helping us to get the best performing Machine Learning model with respect to required processing time.

4. Dataset

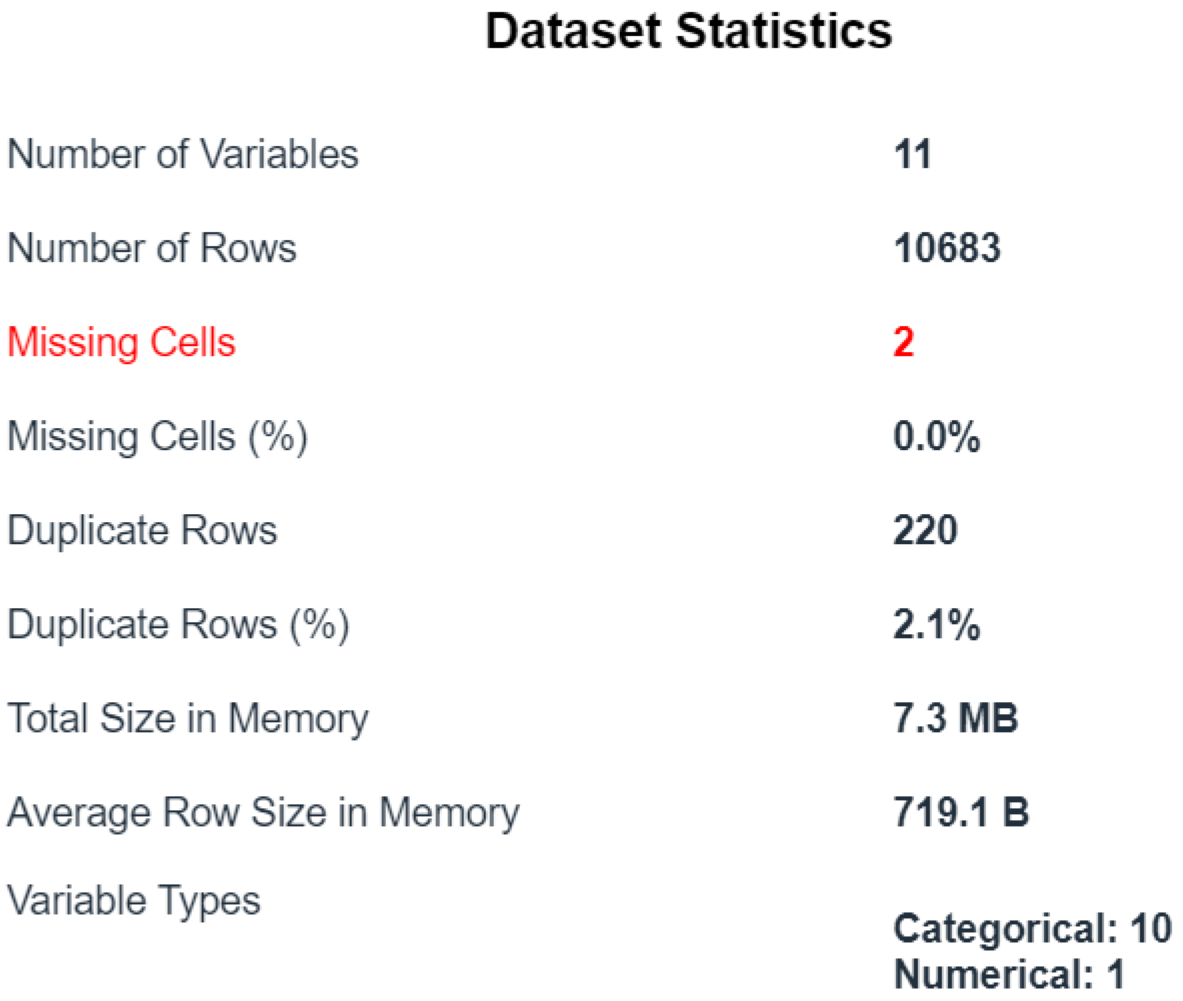
To get the most accurate prediction of airfare , the first step is to use a dataset with enough values covering every parameter under observation.Thus , we are using the dataset containing the price of tickets for various flight tickets for various airlines between the months of March and June 2019. Dataset contains 10683 records.We have splitted our dataset between training and testing set.



```
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64
```

Parameters of Dataset are :-

- **Airline -** The name of the Airline.
- **Date_of_Journey -** The recorded data of journey.
- **Source -** Place from where flight takes off.
- **Destination -** Place where flight landed.
- **Route -** The route taken by flight for the duration.
- **Dep_Time -** Departure time from origin.
- **Arrival_Time -** Time of arrival at the destination.
- **Duration -** Total time taken to reach destination.
- **Total_Stops -** Totale stops taken between source and destination.
- **Additional_Info -** Any additional information about the flight.
- **Price -** Price of the ticket.



Dataset Statistics

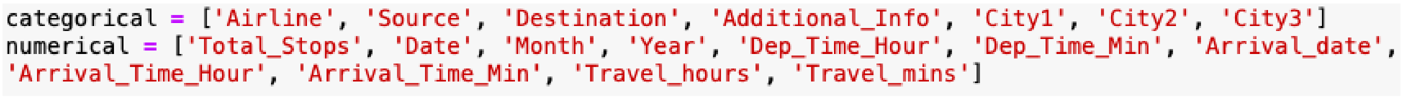
Number of Variables	11
Number of Rows	10683
Missing Cells	2
Missing Cells (%)	0.0%
Duplicate Rows	220
Duplicate Rows (%)	2.1%
Total Size in Memory	7.3 MB
Average Row Size in Memory	719.1 B
Variable Types	Categorical: 10 Numerical: 1

5. Approach

In order to predict the prices of the flight we are initially training our model using a model present in the sklearn library. The models we used were Random Forest Regressor, AdaBoost Regressor and Gradient Boost Regressor. Initially we found that the accuracy given by Random Forest Regressor is around 68% which is much greater than the AdaBoost Regressor and Gradient Boost Regressor, so we choose Random Forest Regressor our model and use it for predicting the flight prices, in order to further increase the accuracy of our model we are hypertuning the model in order to find the best parameter that would give us the best accuracy, after hypertuning the model we train it and then predict the prices of the flight. In order to find other models that could give us much higher accuracy than our current model we use LazyPredict Library. On using the library we find that Extra Tree Regressor model which has a training accuracy of 98% and test accuracy of 69% and XGBRegressor which has training accuracy of 90% and test accuracy of 66%, we use these model to train on our data set and then predict house prices with it.

6. Implementation

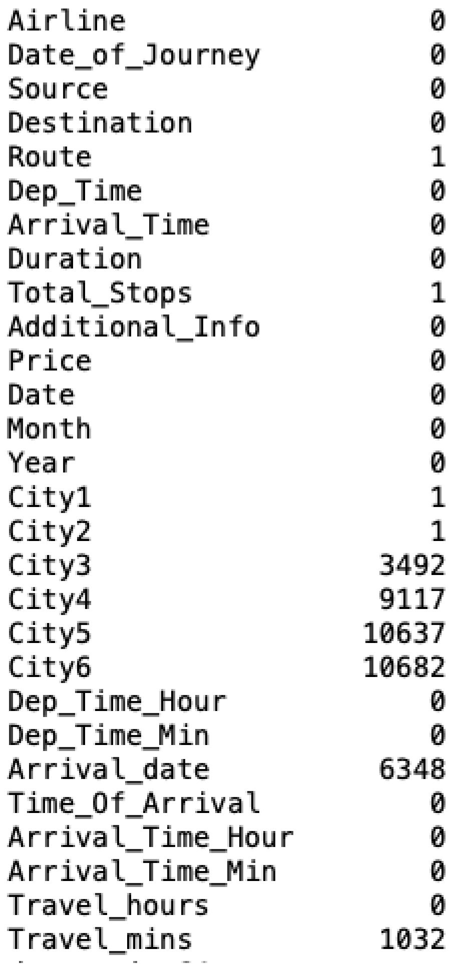
Implementation can be divided in total 3 steps:-



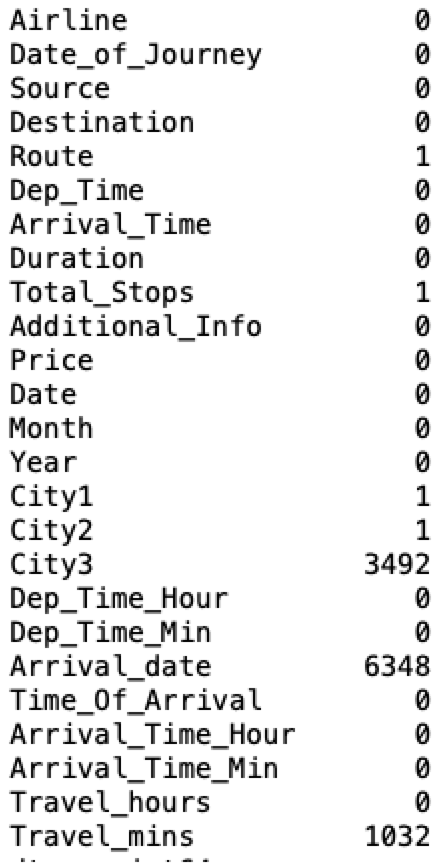
```
categorical = ['Airline', 'Source', 'Destination', 'Additional_Info', 'City1', 'City2', 'City3']
numerical = ['Total_Stops', 'Date', 'Month', 'Year', 'Dep_Time_Hour', 'Dep_Time_Min', 'Arrival_date',
'Arrival_Time_Hour', 'Arrival_Time_Min', 'Travel_hours', 'Travel_mins']
```

Monitoring Parameters

- **Cleaning and Data Preprocessing** - Process of transforming the raw data into an understandable format of our requirement , thus removes all inconsistencies, inaccurate and incomplete data.For cleaning our dataset we have first considered string parameter like month,date,time and converted them into integer data type , through this it will help us to calculate the duration of flight.Flight with no-stop are marked as 0 in dataset.



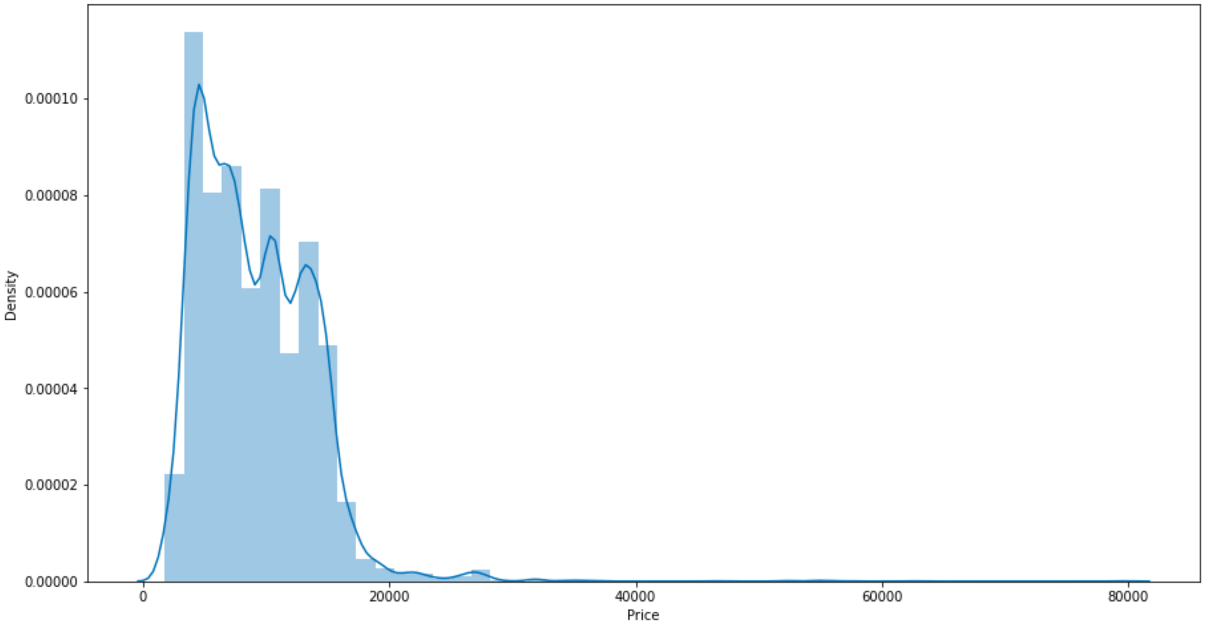
```
Airline                  0
Date_of_Journey          0
Source                   0
Destination              0
Route                    1
Dep_Time                 0
Arrival_Time             0
Duration                 0
Total_Stops              1
Additional_Info          0
Price                    0
Date                     0
Month                    0
Year                     0
City1                    1
City2                    1
City3                 3492
City4                 9117
City5                10637
City6                10682
Dep_Time_Hour            0
Dep_Time_Min             0
Arrival_date          6348
Time_Of_Arrival          0
Arrival_Time_Hour        0
Arrival_Time_Min         0
Travel_hours             0
Travel_mins           1032
```



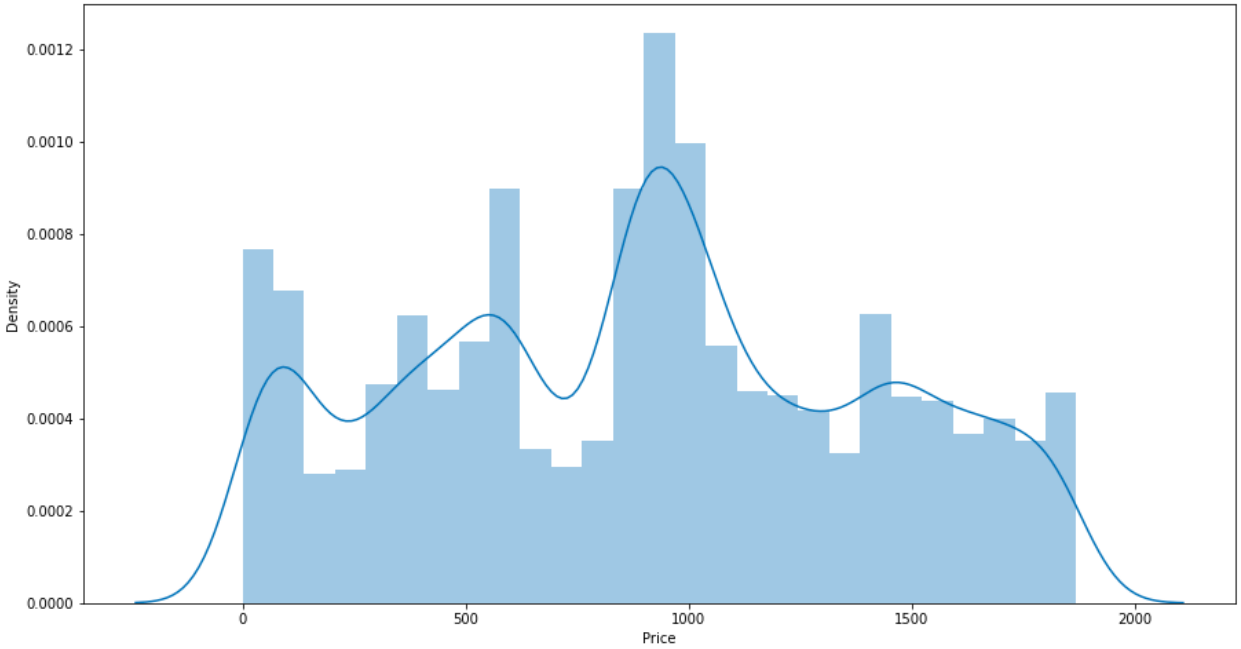
```
Airline                0
Date_of_Journey        0
Source                 0
Destination            0
Route                  1
Dep_Time               0
Arrival_Time           0
Duration               0
Total_Stops            1
Additional_Info        0
Price                  0
Date                   0
Month                  0
Year                   0
City1                  1
City2                  1
City3               3492
Dep_Time_Hour          0
Dep_Time_Min           0
Arrival_date        6348
Time_Of_Arrival        0
Arrival_Time_Hour      0
Arrival_Time_Min       0
Travel_hours           0
Travel_mins         1032
```

Removing Parameters with NULL values(Here City4,5,6)

To prevent inconsistencies we decided to drop data of city 4,5,6 since majority of them are NULL values.After removing all NULL values we split the whole dataset into different rows based on cities,arrival time,duration,etc.



Reducing the Skewness of the plot

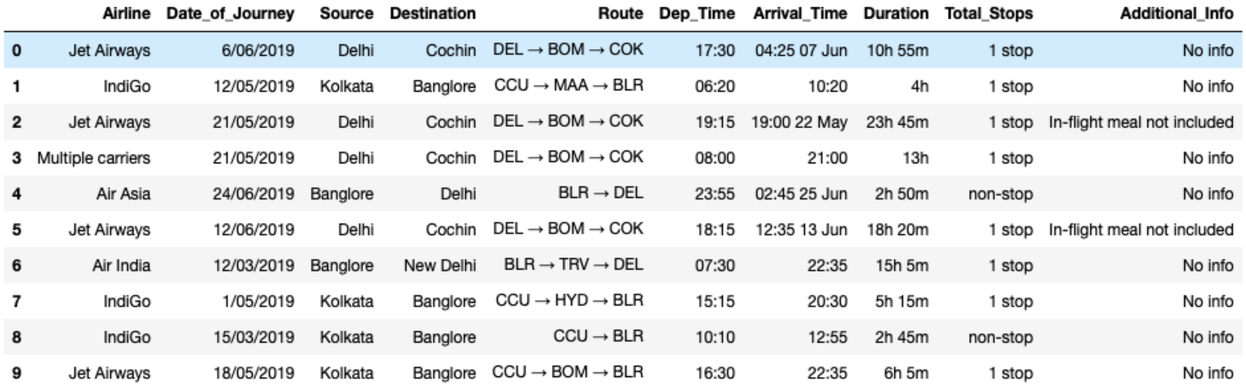


Decreasing Skewness to make the plot more symmetric

After cleaning the data we checked possible counterplots including distribution of price with other categorical data leading to highly skewed graph thus we tried to removed the skewness from dataset to make it more symmetric.Last step of data preprocessing we did was , our whole dataset is too varying with respect to different columns thus we scaled all columns using StandardScaler from sklearn library.

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. We have an encoder from sklearn inorder to encode our data. In our dataset we have a column ‘Airline’ where the name of the flights are mentioned so we have converted the names of flights from string data type to integer data type using encoding.
Our data is sparsely distributed, this makes it difficult for our machine to learn, therefore bringing our data to a common scale will make learning easy for the model, for scaling our data we have used standard scaler.

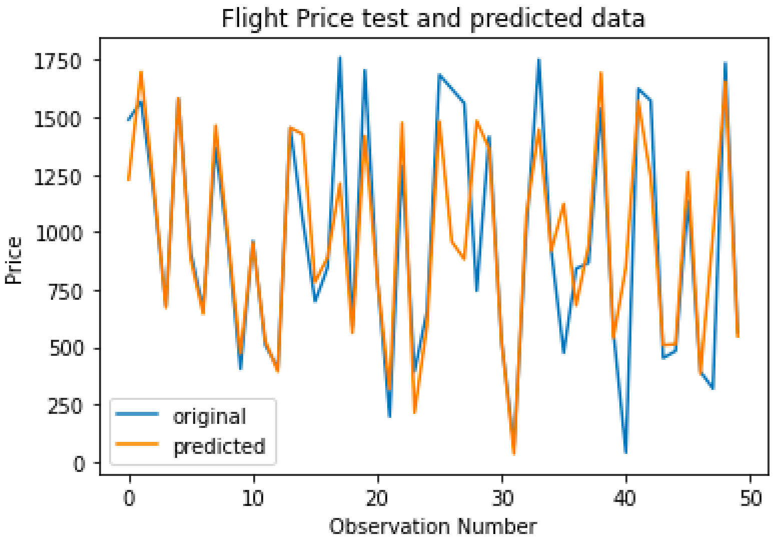
- **Model Training** - Training model is a dataset that is used to train an ML algorithm , this training dataset consists of input and correct output , on the basis of this fed dataset model have to predict the influence output.Whole purpose of training model is to run the input data through the algorithm to correlate the processed output against the given sample output.To training the model we have split are whole dataset in to 7:3 split where 70% of the dataset is used to training the model and rest 30% will be used for the testing of the model.



	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL → BOM → COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info
5	Jet Airways	12/06/2019	Delhi	Cochin	DEL → BOM → COK	18:15	12:35 13 Jun	18h 20m	1 stop	In-flight meal not included
6	Air India	12/03/2019	Banglore	New Delhi	BLR → TRV → DEL	07:30	22:35	15h 5m	1 stop	No info
7	IndiGo	1/05/2019	Kolkata	Banglore	CCU → HYD → BLR	15:15	20:30	5h 15m	1 stop	No info
8	IndiGo	15/03/2019	Kolkata	Banglore	CCU → BLR	10:10	12:55	2h 45m	non-stop	No info
9	Jet Airways	18/05/2019	Kolkata	Banglore	CCU → BOM → BLR	16:30	22:35	6h 5m	1 stop	No info

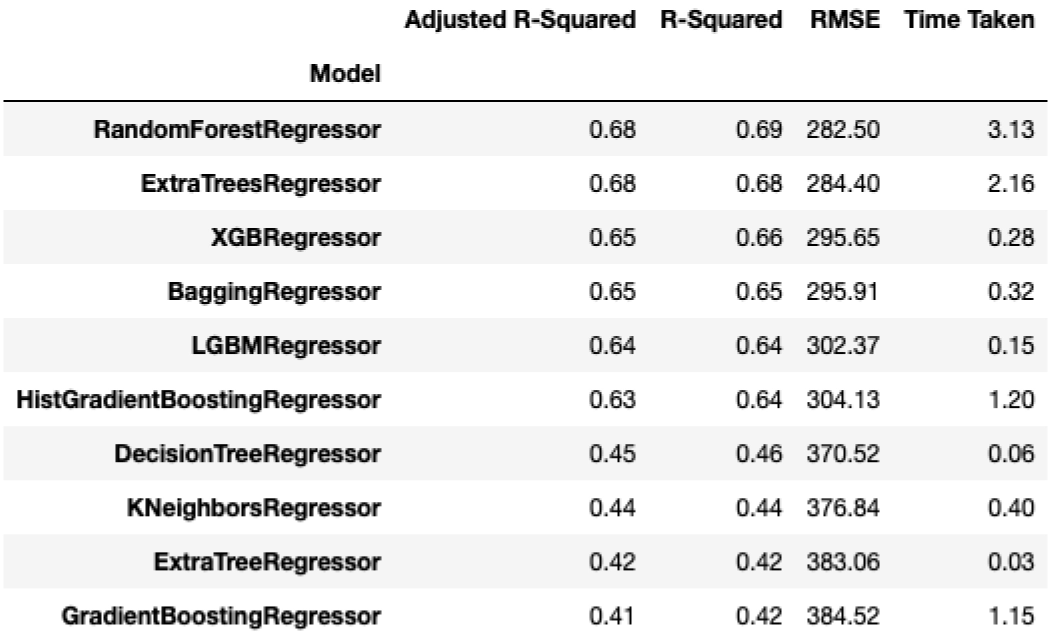
Test DataSet(Head)

To get the best results one have to compare the dataset on the different types of models thus to do this we used sklearn library for the models like linear regression, decision tree, SVR, KNN,random forest, adaboost regressor,etc.For advanced models we have used few parameters for optimal results , process called as hypertuning.After all setting we have on total ten different models and among them Random forest and Extra tree regressor has maximum adjusted r squared while gradient boosting regressor has minimum.



Model-ExtraTreesRegressor
(Test and Predicted Data Plot)

Main parameters to decide which model to use were r-squared error and RMSE , using the lazy predict library one can easily compare the values like r-squared and RMSE to compare the performance of different models.



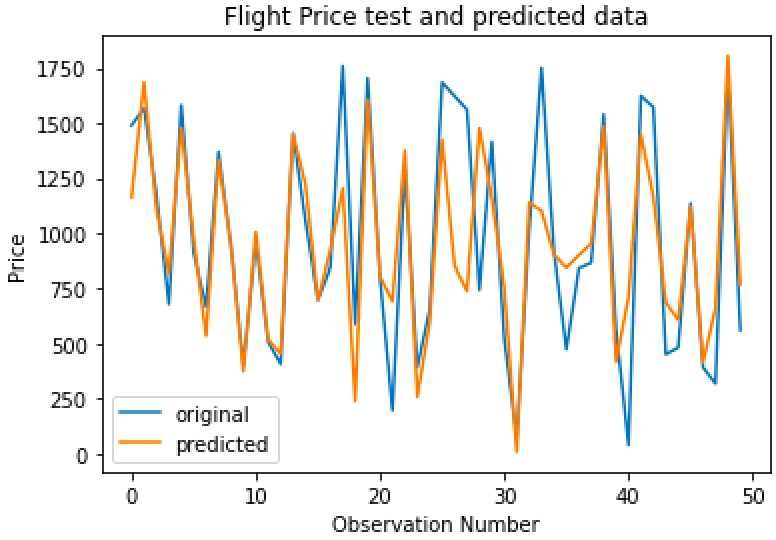
	Adjusted R-Squared	R-Squared	RMSE	Time Taken
Model				
RandomForestRegressor	0.68	0.69	282.50	3.13
ExtraTreesRegressor	0.68	0.68	284.40	2.16
XGBRegressor	0.65	0.66	295.65	0.28
BaggingRegressor	0.65	0.65	295.91	0.32
LGBMRegressor	0.64	0.64	302.37	0.15
HistGradientBoostingRegressor	0.63	0.64	304.13	1.20
DecisionTreeRegressor	0.45	0.46	370.52	0.06
KNeighborsRegressor	0.44	0.44	376.84	0.40
ExtraTreeRegressor	0.42	0.42	383.06	0.03
GradientBoostingRegressor	0.41	0.42	384.52	1.15

Models Accuracy Comparison(Library Used - LazyPredict)

In comparison of best we used the Extra Tree regressor for our final model since it yielded the same r values with less time as of random forest but XGB regressor gave us little more RMSE but it took minimal time in comparison of the above two.Thus XGBregressor is used.Fitting the data frame further to predict the results.

- **Prediction** - Model prediction is a fast technique and often calculation is completed in realtime.It works by analyzing the historical data and making assumptions on what happened in the past tand what will happen now.Thus found to be best for prediction.We are using XGB regressor thus on running this model on test dataset we were able to receives accuracy of around 65% , although extra tree regressor gave use accuracy of around 67% but it took too much time with respect to XGB regressor. On further using joblib we were able to get accuracy of 79% , thus our model is able to predict nearly correct prices.

7. Results



Plot showing prediction curve(in orange) and real curve(in blue)

After the training of our model is complete we used the matplotlib library to plot our predictions against the actual data that we have obtained from splitting the dataset for testing purposes.On studying the whole dataset and after many steps of cleaning and preprocessing we were able to achieve an accuracy of 79% for our model.Thus on final testing we found that our predictions were nearly correct.This accuracy was achieved using 7:3 split of dataset for the training and testing part which might change on a different split.

8. Analysis and Conclusion

While studying over the factors determining the parameters affecting the airfare we found that there is no such fixed parameter but a group of parameter exists which are highly dynamic all over the time , causing the scale of data to rise and lower down at any time.Thus after considering the parameters we found that our considered data is highly skewed thus curve moved away from the gaussian curve.Making further changes , we require several different models in ordered to get the best results , to compare the performances of all the models we used LazyPredict library which made our work a lot easier to an high extent , outputting all the performance measures of different model in a single table.

Also we found that few models were able to yield more accurate results on setting parameters in them , but our aim was to get maximum accuracy with minimum processing time.Therefore we used XGBRegressor which yielded optimum results and on further train with joblib providing pipelining utilities for the processing increased accuracy to 79%.