**Flight Price Prediction**

As the above heading depicts we will be studying about the Reason involved behind the huge variation in the price of the flight . We will create different Machine Learning models to predict the price of the flight tickets. We will go step by step in creating a good machine learning model which will help us to understand in details about the main reasons involved in these price variation and also we will understand the whole machine learning model building process.



**INTRODUCTION:**

Pricing in the airline industry is often compared to a brain game between carriers and passengers where each party pursues the best rates. Carriers love selling tickets at the highest price possible — while still not losing consumers to competitors. Passengers are crazy about buying flights at the lowest cost available — while not missing the chance to get on board. All this makes flight prices fluctuant and hard to predict.

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.

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable.

Problem Statement:

Here you will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

**FEATURES:**

Airline: The name of the airline.

Date\_of\_Journey: The date of the journey

Source: The source from which the service begins.

Destination: The destination where the service ends.

Route: The route taken by the flight to reach the destination.

Dep\_Time: The time when the journey starts from the source.

Arrival\_Time: Time of arrival at the destination.

Duration: Total duration of the flight.

Total\_Stops: Total stops between the source and destination.

Additional\_Info: Additional information about the flight

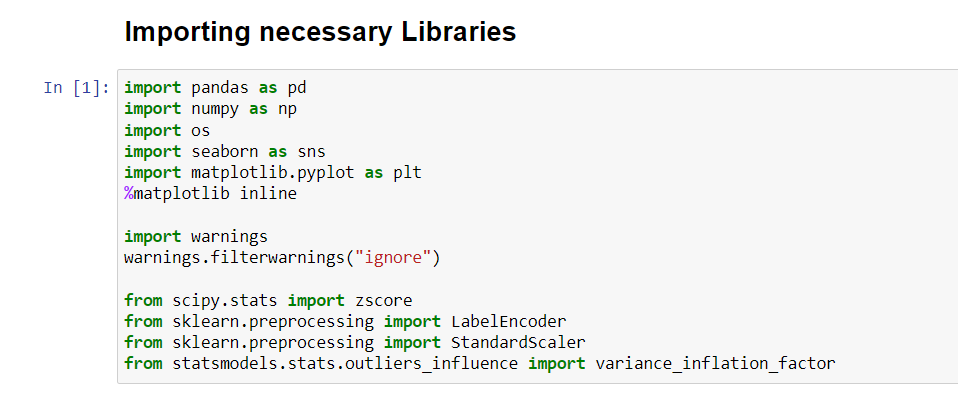
Price: The price of the ticket

As we can see above this is our Problem statement which gives us the information about the Dataset. In this dataset we can see that there are in total 11 Attributes in which “Price” is our Target Variable and remaining 10 columns are the independent variables.

Since our target variable “Price” has continuous data which means it’s a “Regression type Problem”. As we need to make a model for flight price prediction so that it can help consumers in making better purchasing decisions.

For Data Analysis , Firstly we need to import all the necessary libraries so that we could open and read the Dataset.

Data Analysis:



After importing all the necessary libraries, we can see that we have two separate dataset.

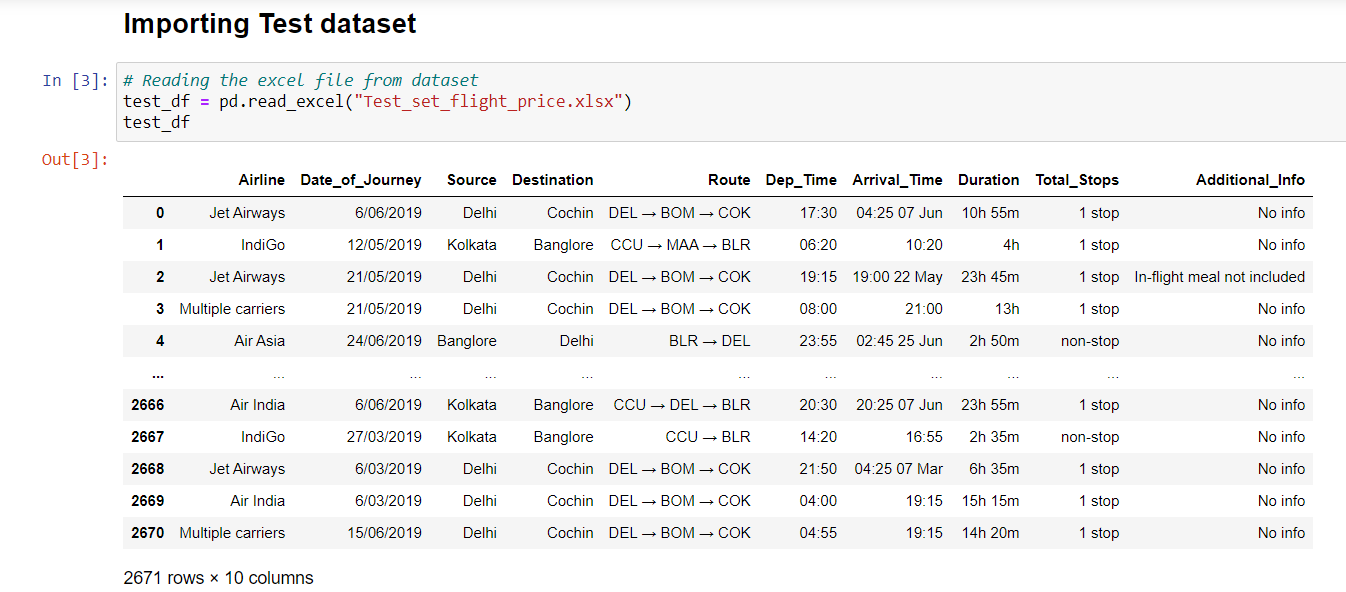
1. **Train Dataset**

This dataset will be used for building our Machine Learning Model as it contains 10 independent variables and 1 target variable. It consists of 10683 rows and 11 columns.



1. **Test Dataset**

This dataset will be used for getting prediction by using our trained model as it contains only independent variables. It consists of 2671 rows and 10 columns.



As we know that whenever we are provided with two sets of data i.e training and testing data. There are two ways for processing our data:

1. We need to perform all the Preprocessing and Data cleaning process for training dataset to build machine learning model and doing the same process again for the testing dataset for **getting prediction from the trained model by loading the saved trained model.**
2. Another way is in which we can concatenate and merge both the dataset into single dataset and then processing it to build machine learning model.

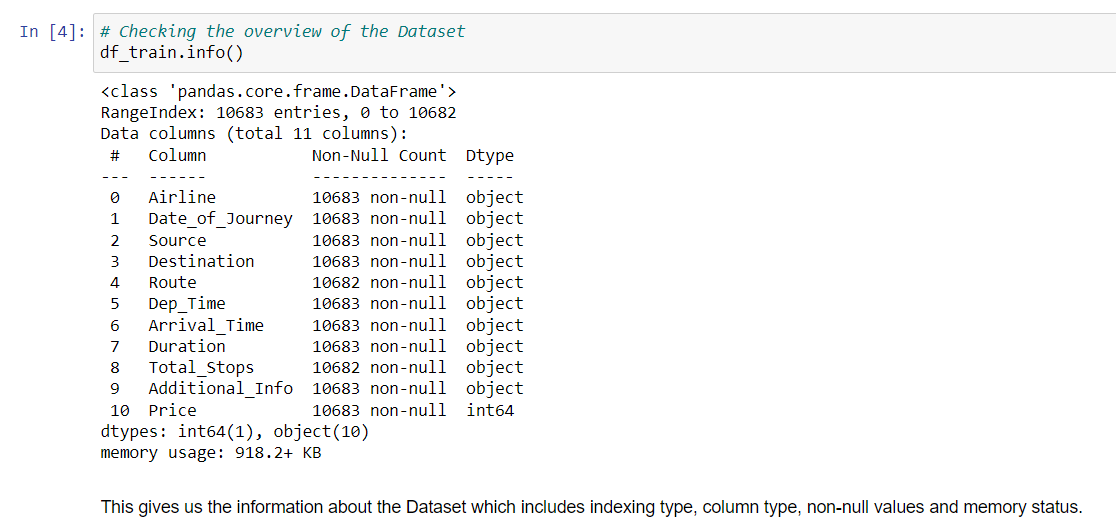
After looking at the first and last 5 rows of both the training and testing dataset, we can see that we have both Numerical and Categorical data. We can also see some special characters used in the data for which we need to do data transformation.

Even though the problem statement specifically mentioned about the months, but there are no particular columns for months. So, we will extract the values of month and day from date of journey column and make a separate column for them to study the prices of flight tickets for various airlines based on month.

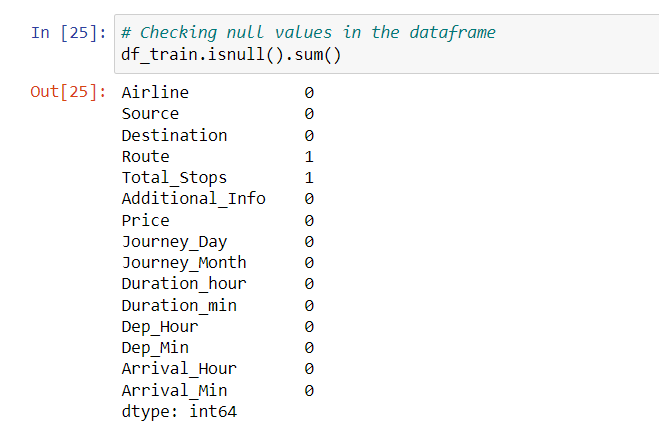
The arrival time column contains date mentioned with time, so we need to make a separate column for that too. Same things to be done in the case of departure time.

Duration is the difference between arrival time and departure time. Since the duration column contains both hours and minutes data, we can extract the values from this column.

**EDA (Exploratory Data Analysis)**

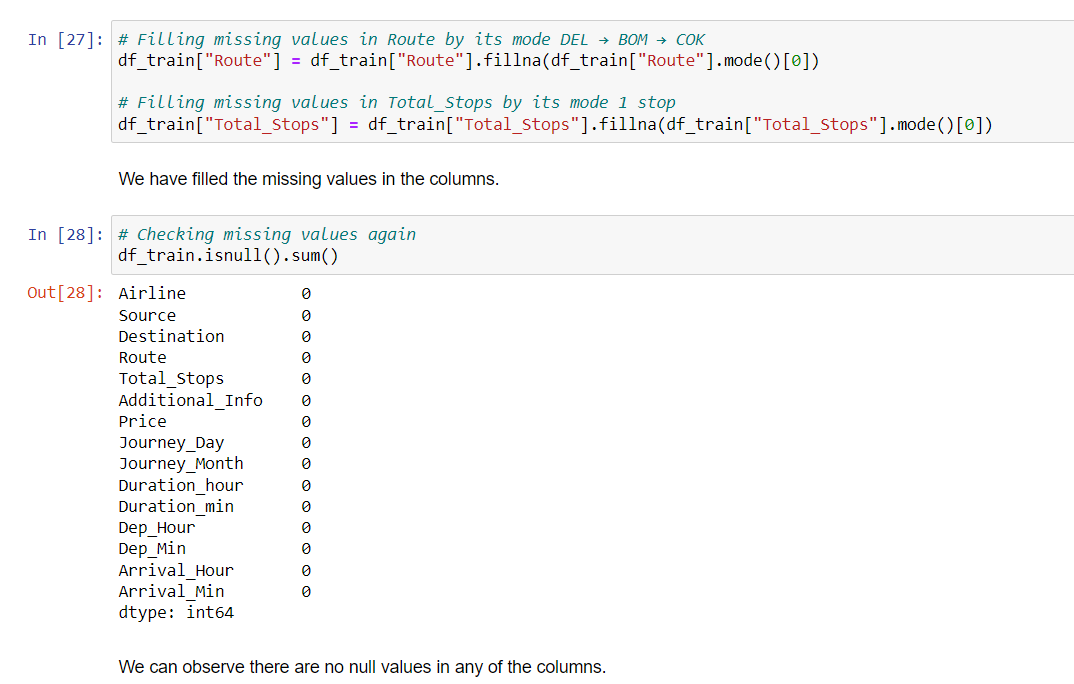


We run df\_train.info() this gives us the information about the columns present in our dataset , number of values present, type of data present in each columns and the memory usage of the dataset.

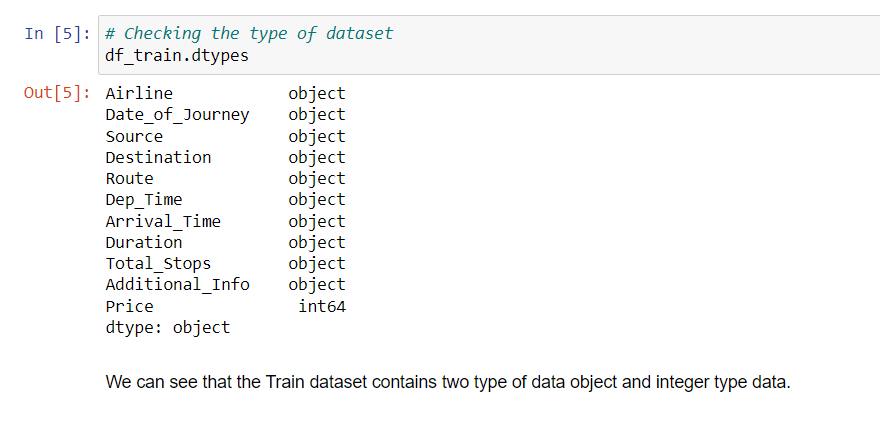


In this we are checking if there are any missing values present in our dataset. After running df\_train.isnull().sum() we can see that Route and Total\_stops column have 1 missing values they might be from the be from the same row since we get the values of total stops from route column only. We can directly use dropna method but these two columns have categorical data so we will use mode method to fill the missing values.

NULL VALUES CAN BE TREATED BY USING IMPUTATION TECHNIQUE



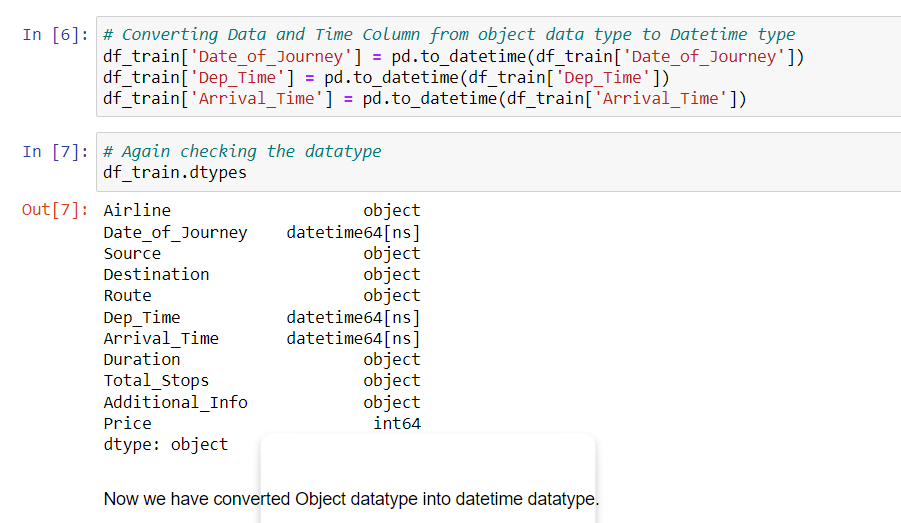
Let’s check the type of the data present in our Dataset.



As we can see that all of our independent variables are of object datatype and our target variable is integer datatype.

Feature Engineering:

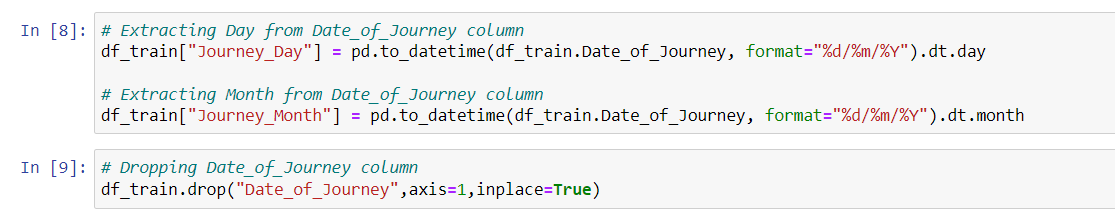
As we can see that in the columns Date\_of\_Journey, Dept\_Time and Arrival\_Time contains special characters due to which its showing object data type which means python is not able to understand this type of data in the columns.

Therefore, we need to convert object datatype into timestamp to use them for proper prediction.

We have converted object datatype into datetime datatype.

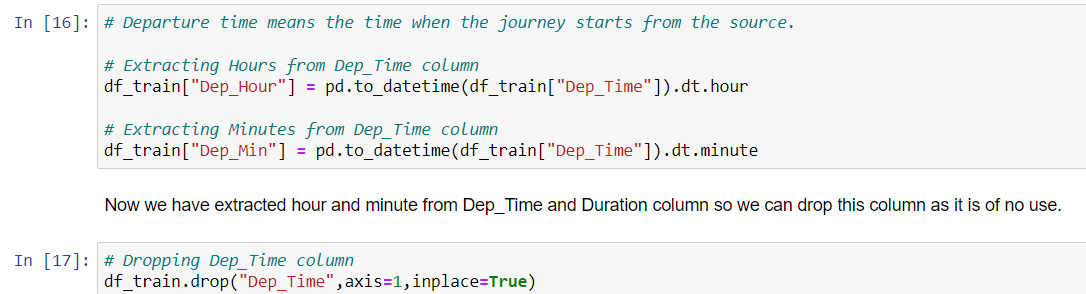
**Date\_of\_Journey:**

So, from the problem statement we get to know that the provided data is from a single year i.e. 2019 so we don’t need to extract it. We will only extract date and month from the column and then we will drop the Date\_of\_Journey column after extraction of the required data.



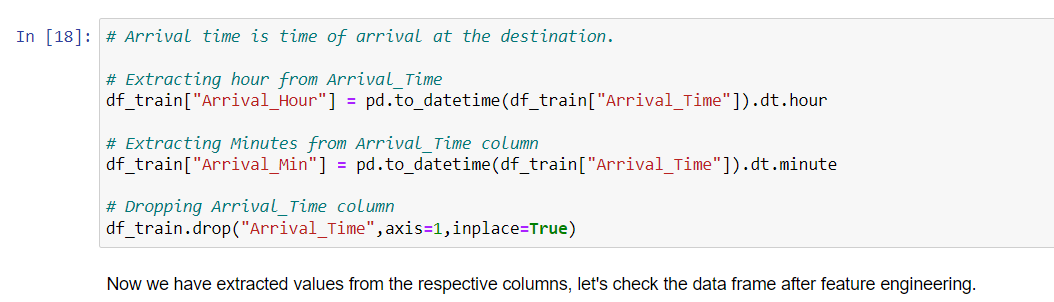
Dep\_Time:

Departure time means the time at which the flight takes off. This column contains hours and minutes so we need to extract hours and minutes separately and then we will drop the Dep\_Time column.



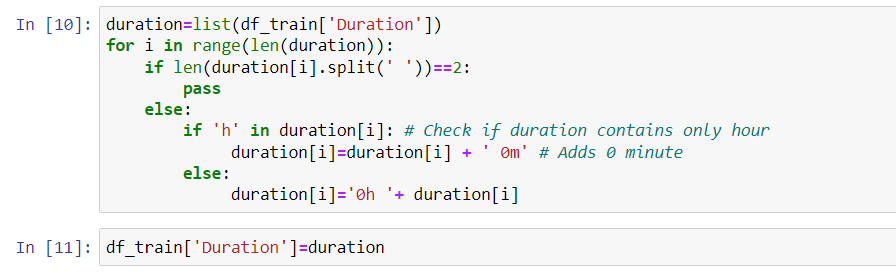
Arrival\_time:

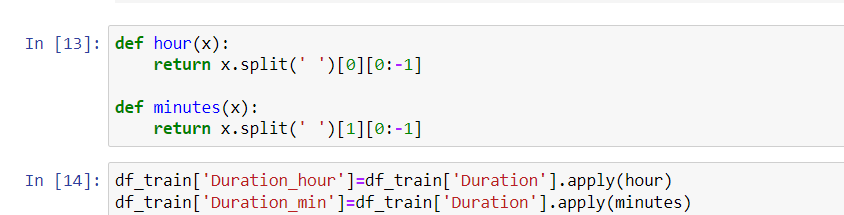
Arrival time means the time at which the flight lands. Similarly we will extract hours and minutes from the Arrival\_time and then we will drop Arrival\_time column also.

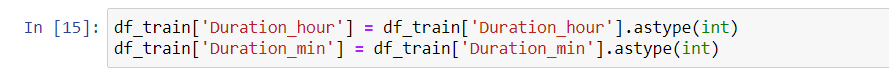


Duration:

The column Duration has values in terms of minutes and hours. Duration means the time taken by the plane to reach the destination. It is basically the difference between arrival and departure time. We will extract hours and minutes from this Duration column.

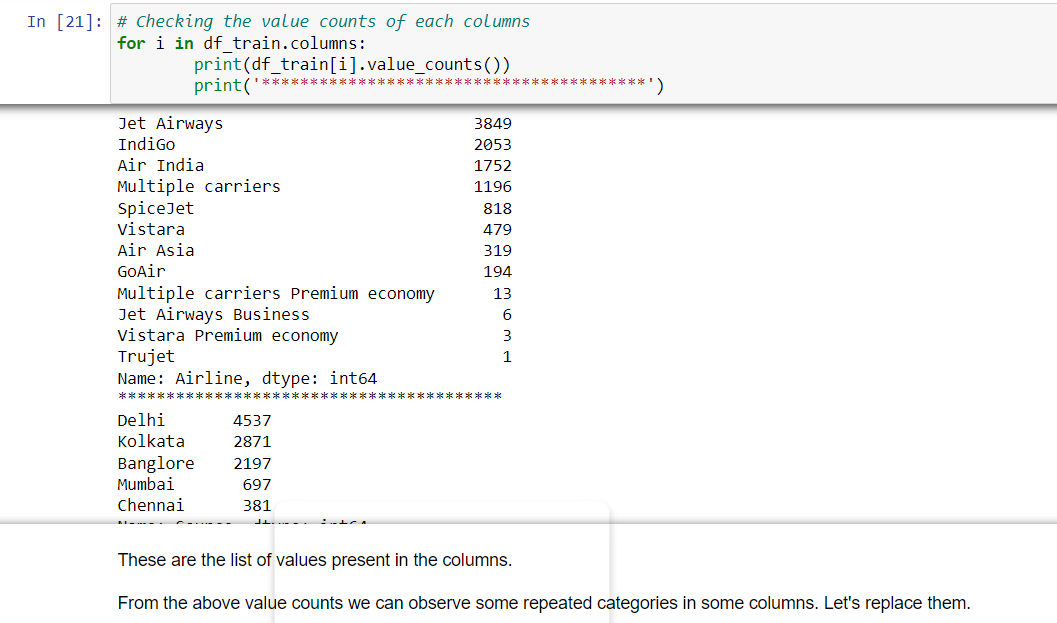




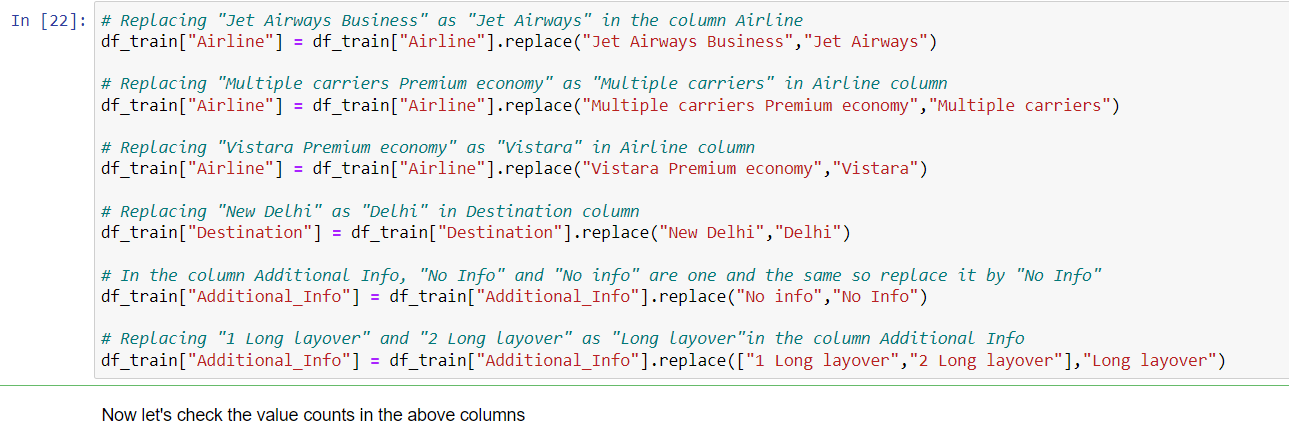




We have extracted all the required data and made columns out of it. After looking into the dataset I have found that some of the columns i.e. Airline, Destination and Additional Info have some repeated categories. I have to check the value counts of the columns and then replace it with the required category.

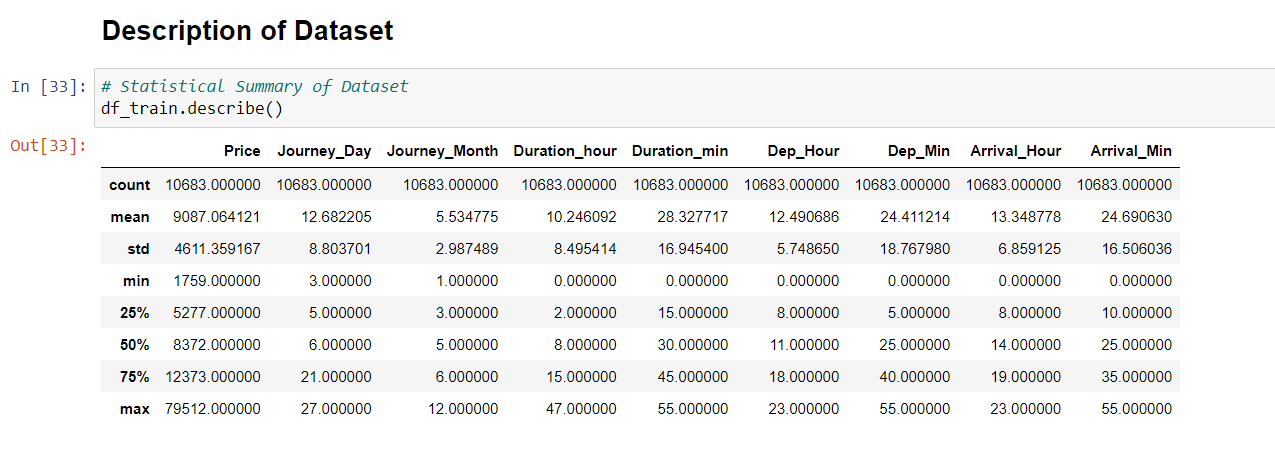


As we can see we have some repeated categories so we need to replace it with required categories.





Now we have successfully replaced the required category now we will move forward towards statistical summary of the Dataset.



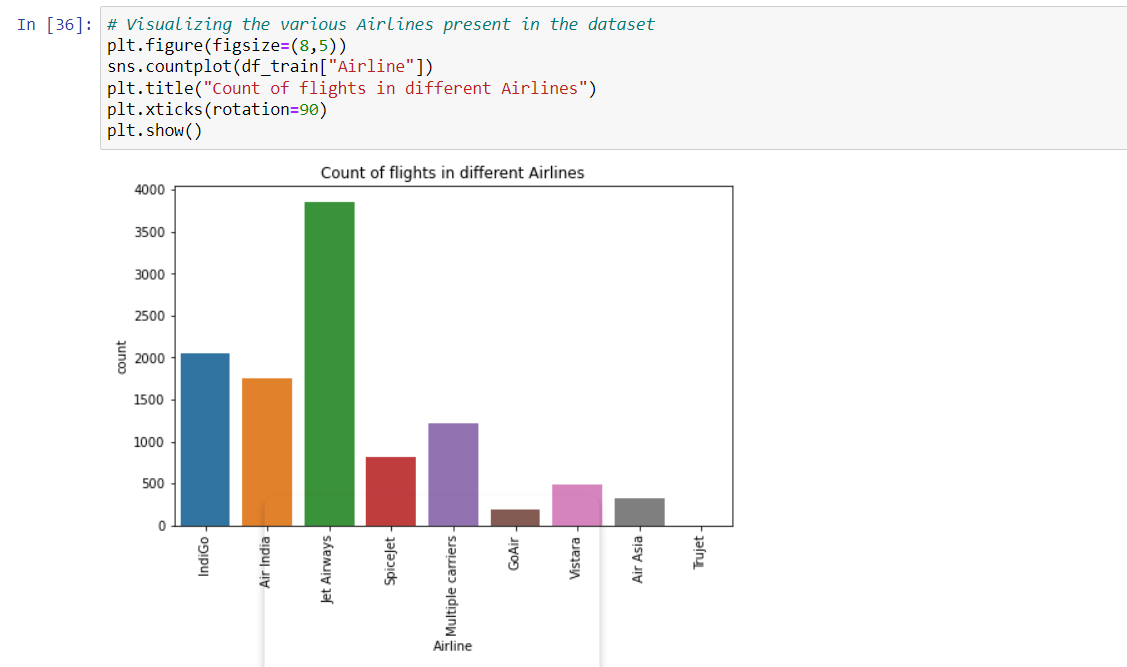
We use df\_train.describe() method for getting the statistical summary of the dataset. The summary of this dataset looks perfect since there is no negative/ invalid values present. It gives the summary of numerical data.

From the above description we can observe the following things:

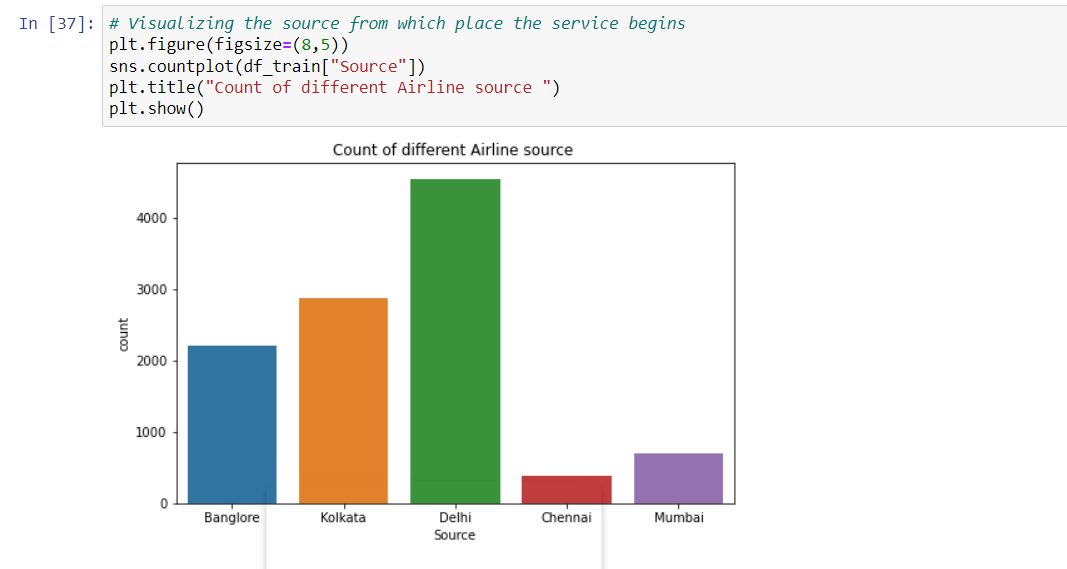
* The counts of every column is same which means there are no missing values present in the dataset.
* The mean value is greater than the median(50%) in the columns Price, Journey\_Day, Duration\_hours and Dep\_Hour so we can say they are skewed to right.
* The median(50%) is bit greater than mean in Duration\_mins, Dep\_Min, Arrival\_Hour and Arrival\_Min which means they are skewed to left.
* From the description we can say the minimum price of the flight ticket is Rs.1759 and maximum price is Rs.79512 also the mean is 9087.
* In summarizing the data we can observe that there is huge difference in maximum and 75% percentile in the columns Price, Arrival\_Min, that means huge outliers present in those columns. These differences can also be seen in many other columns.

Data Visualization:

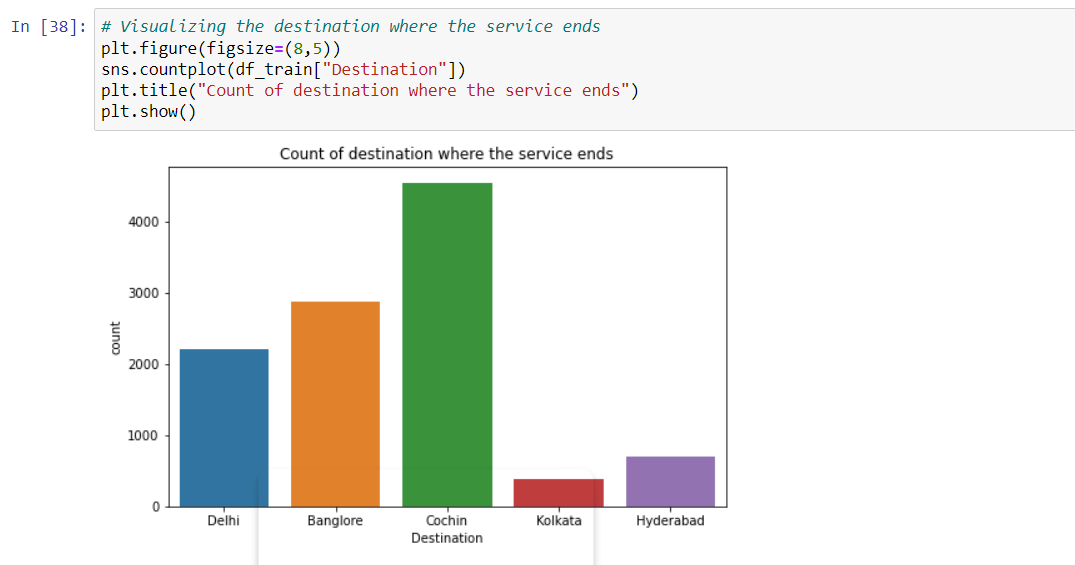
Plotting Categorical Columns:



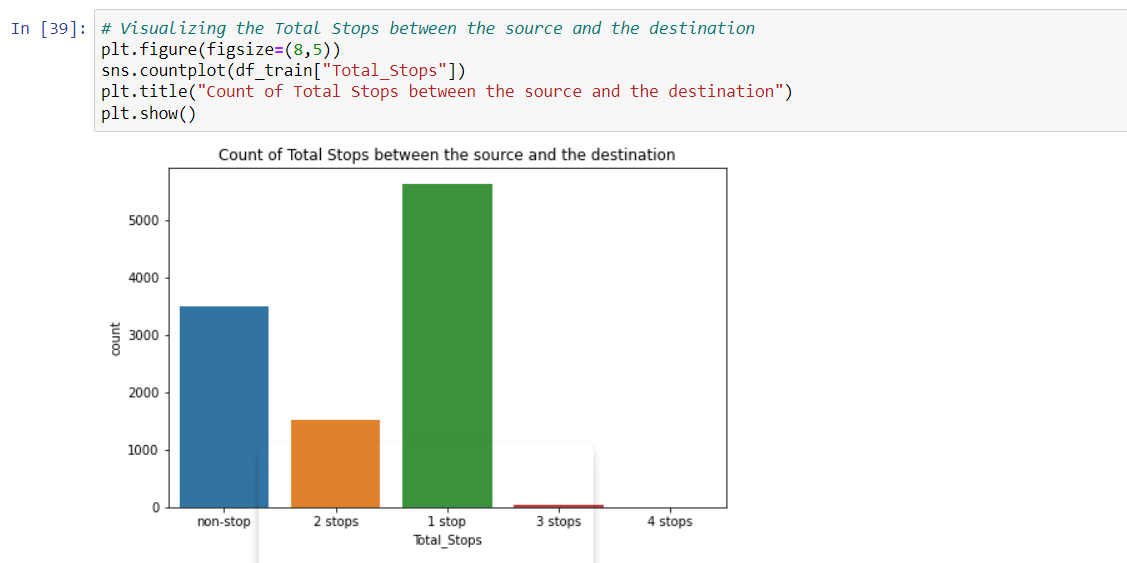
After looking into the above plotted graph of the Airlines column we can observe that JetAirways has the highest number of flights followed by Indigo then Air India and others.



Above plotted graph clearly depicts that Delhi has the highest number of flights taking off in comparison with the other souces like Kolkata, Bangalore, Mumbai and Chennai.

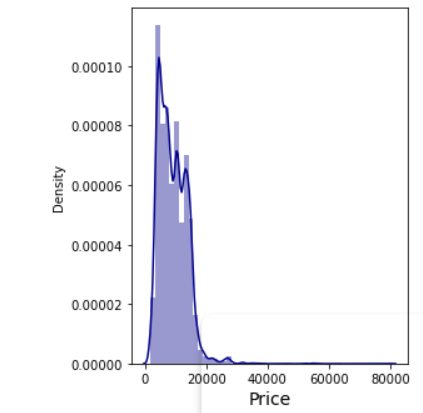


The graph gives us the info that Cochin has the highest number of landing flights followed by Bangalore and then Delhi which had the highest flights take off.



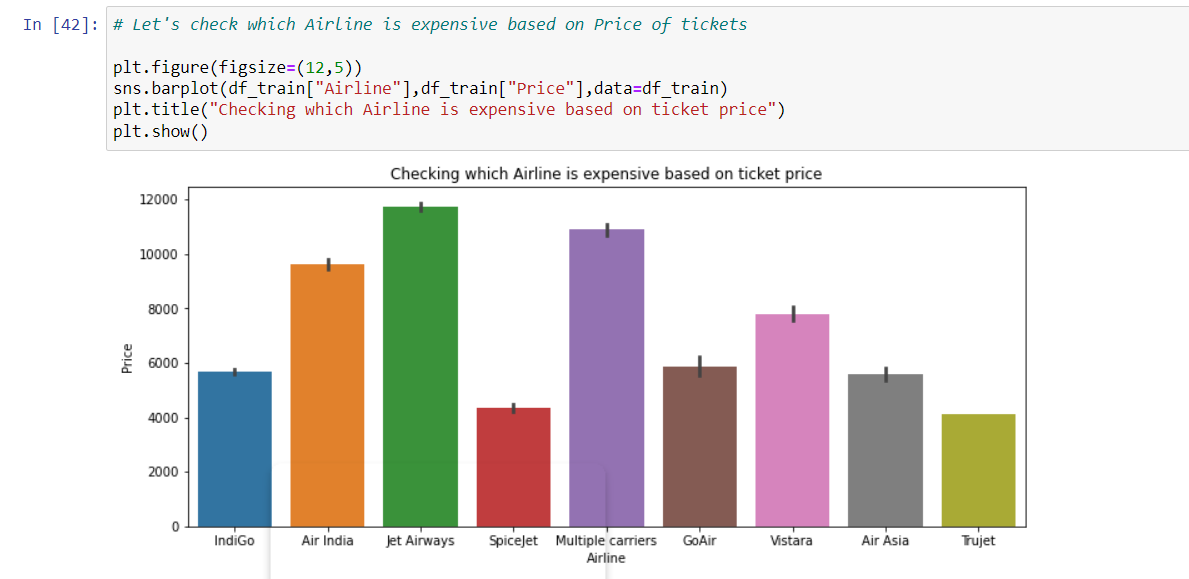
So we can see that most of the flights are 1 stop flight between source and the destination followed by non-stop flights. No flights have 4 stops and very few have 2 stops and very few have 3 stops.





I have showed the above graph plot as it is our target variable and we get the graph of all the variables using the above code.

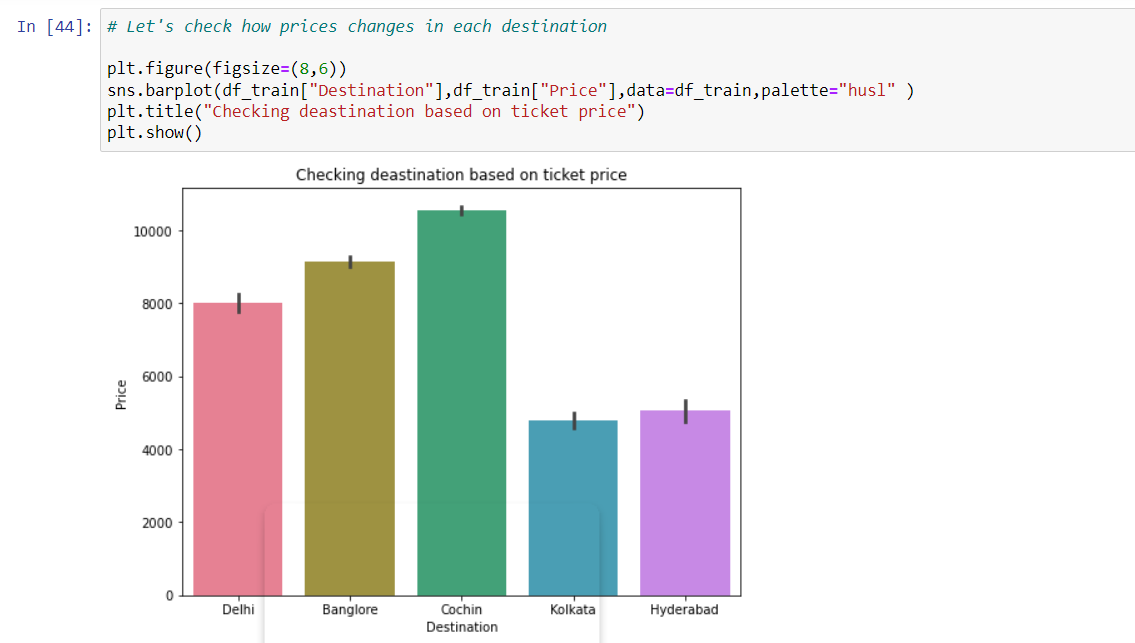
We can see that our target variable is not normally distributed its mean is more than that of median so the data looks to be right skewed. I will removed the skewness later now lets visualize our feature and label.



As we can see in the above graph plots that JetAirways has the most expensive flights followed by Mutiple Carriers. Whereas Truejet and Spicejet has the cheapest compared to all other.



In this plot we can see that price of ticket of Delhi is highest followed by Kolkata, Bangalore, Mumbai and Chennai.

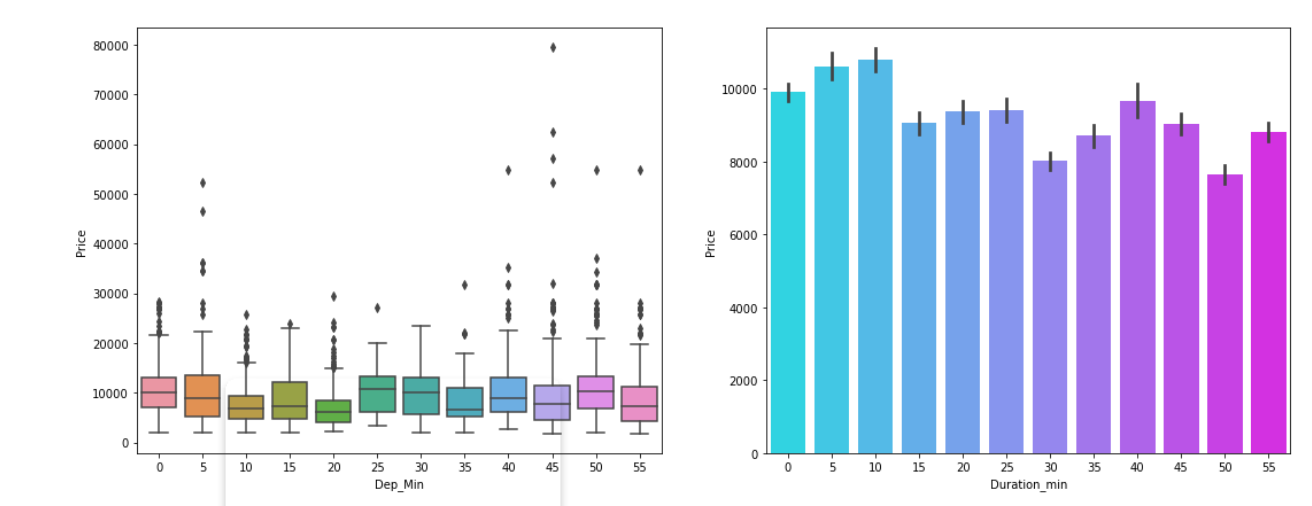
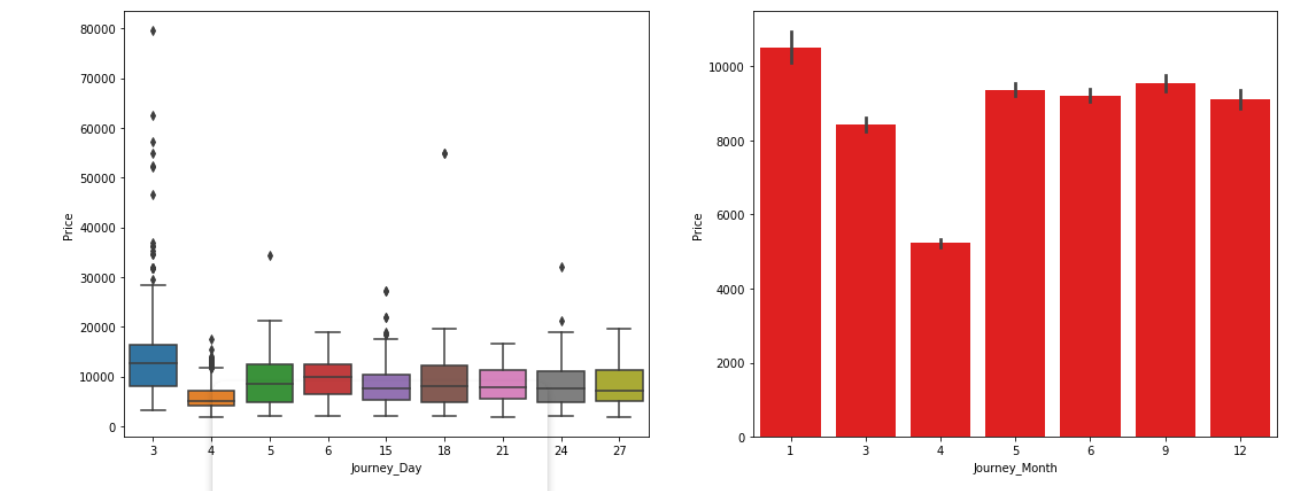


The ticket price is high in Cochin destination followed by Bangalore which means they have long distance from the source.



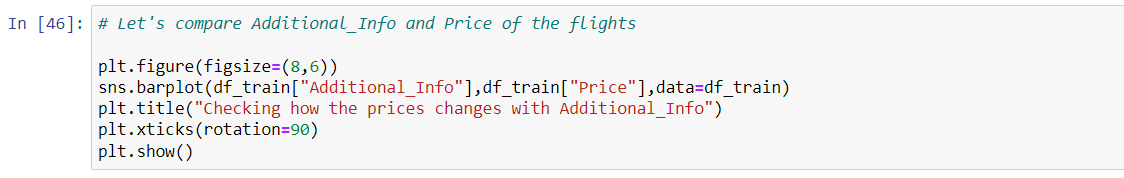
Here the flights with 4 stops have highest price followed by flights having 3 stops and the flights which have no stops is having very less ticket price compared to others.

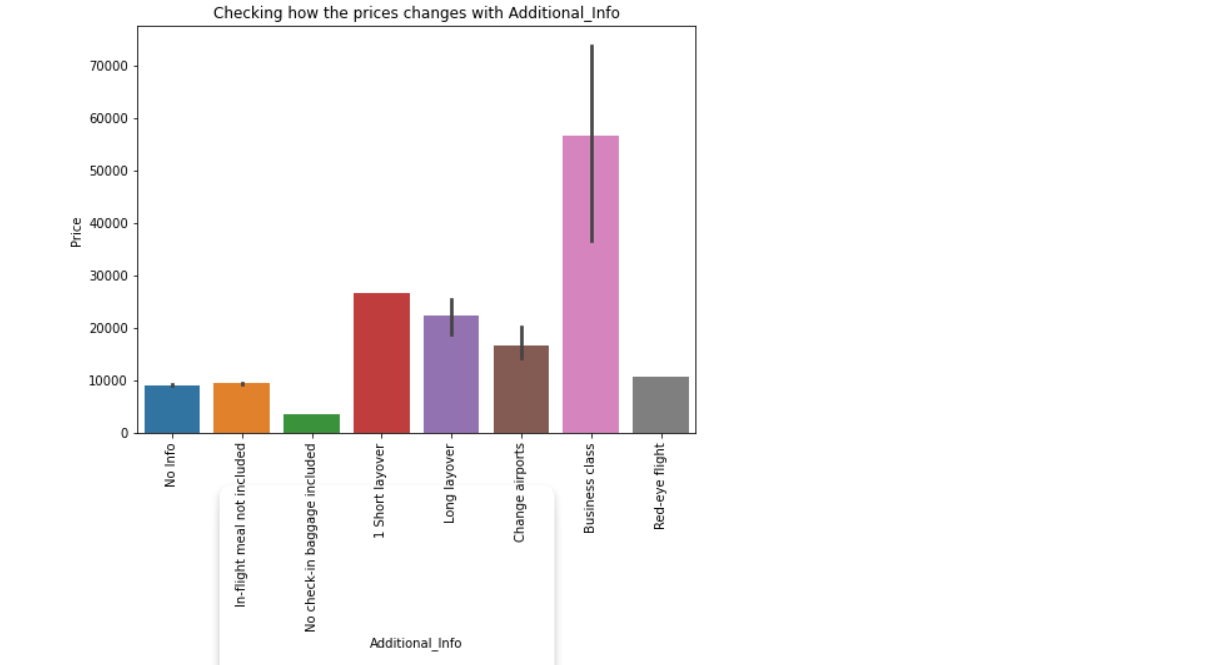




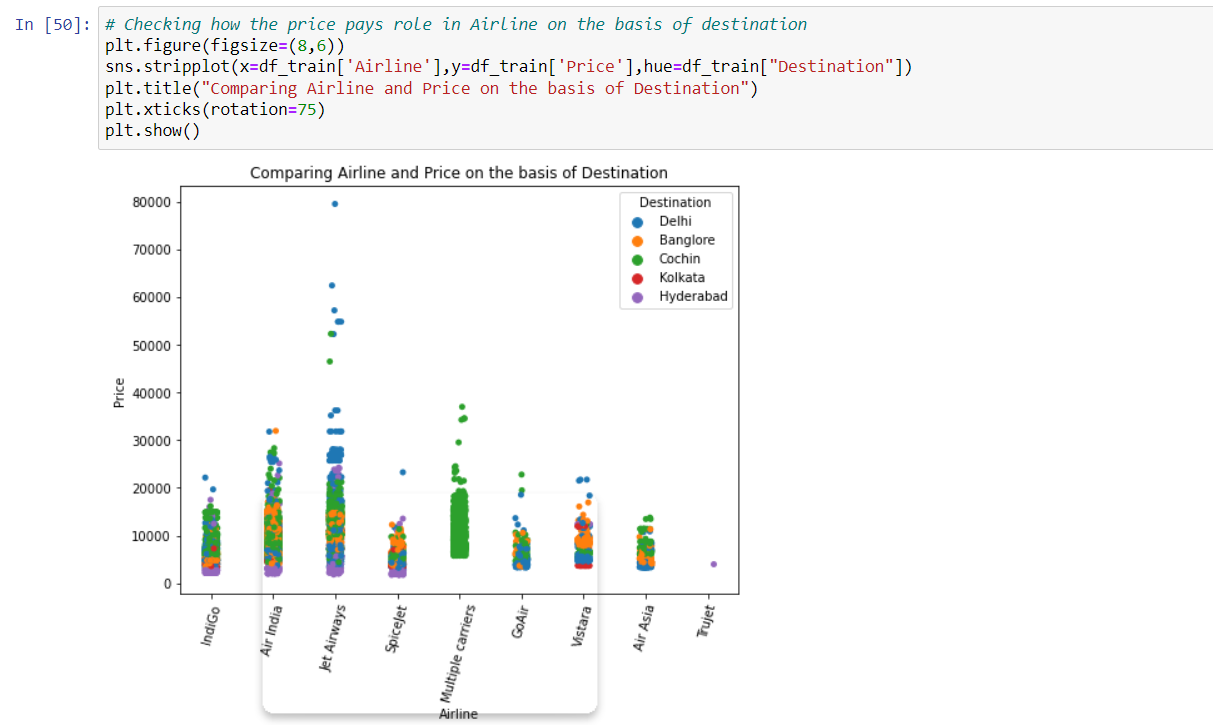
From the above plots we can observe the following

* While comparing Journey\_Day and Price we can see the price of ticket is high in day 3 apart from this there is no much impact of day on ticket price.
* While comparing Journey\_Month and Price it can be inferred that the flights travelling in the January month are more expensive than others and the flights travelling in April month have very cheap ticket prices.
* There in no significance relation between Dep\_MIn and Price of the tickets.
* In the fourth graph also we can say there is no much impact of Duraation\_mins on Price. But we can say duration minutes 10 and 5 have bit high prices compared to others.





The plot shows that the Business class flights are more expensive compared to others and the flights having the class No check-in baggage included has very least ticket price.



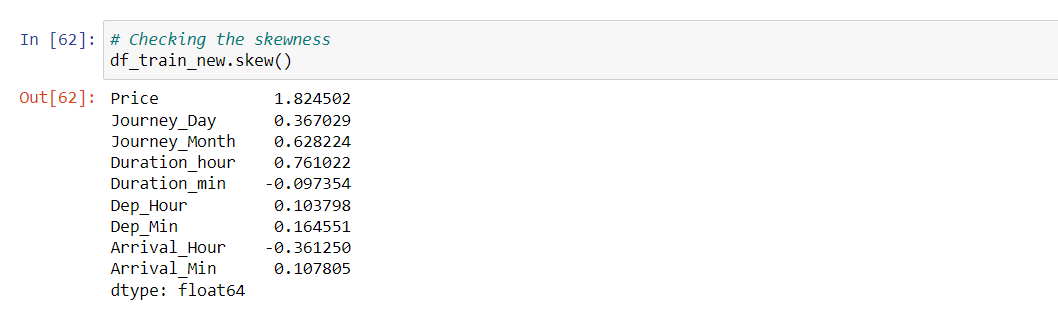
Here we can conclude that the Jet Airways flights that are destined to Delhi are have more expensive ticket prices compared to others.

EDA Concluding Remarks:

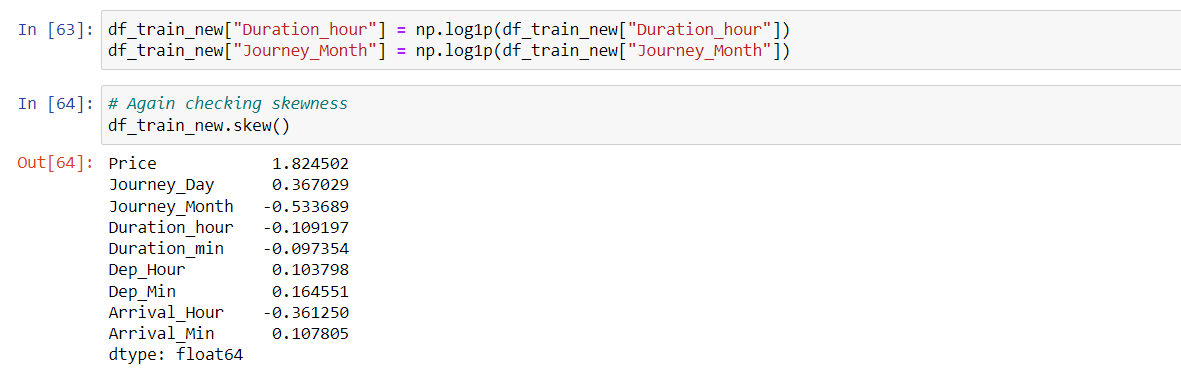
I have done all the Data transformation introduced new required columns in our Dataset. Removed all the irrelevant data which was not needed.

After doing that I have visualized the data in which we observed that flight prices depends on many factors such as prices of tickets are cheaper in flights which do not provide certain services like check in baggage and flight meals. Also, the number of flights varied from city to city and Delhi being the source of most flights and Cochin seems to be the destination of most flights. The number of flights is highest in the month of January and the ticket price is also expensive in this month compared others.

I used boxplots to identify the outliers and found outliers in Duration\_hours and Journey\_Month, so I decided to remove the outliers using Zscore method as the data loss using this method is less compared to IQR method and got new dataframe. You can use any of these methods based on data loss.

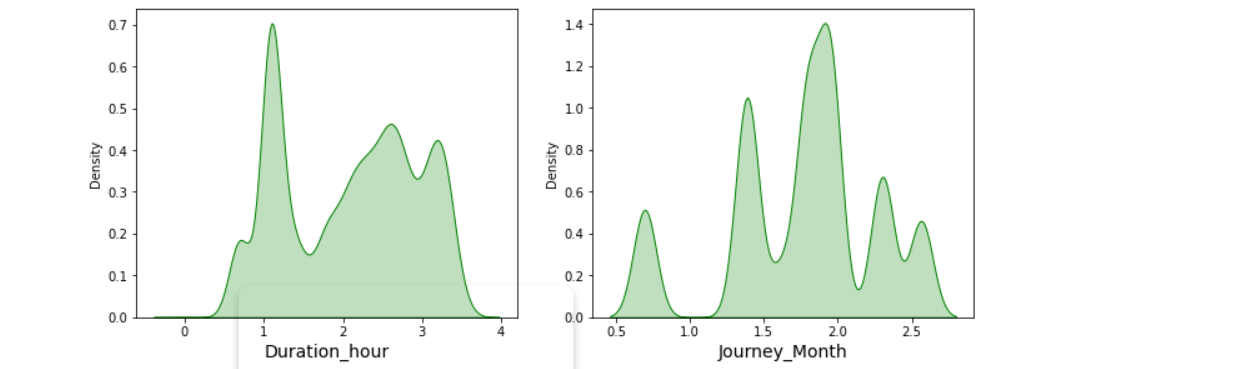
Checking Skewness in Dataset

Presence of skewness more than +0.5 and -0.5 is not acceptable as it will impact on our accuracy. Here we can find Price, Journey\_Month and Duration\_hours have skewness above the acceptable range. But the column Price is our target so I am keeping it untouched and removing skewness in the remaining columns using log transformation method.



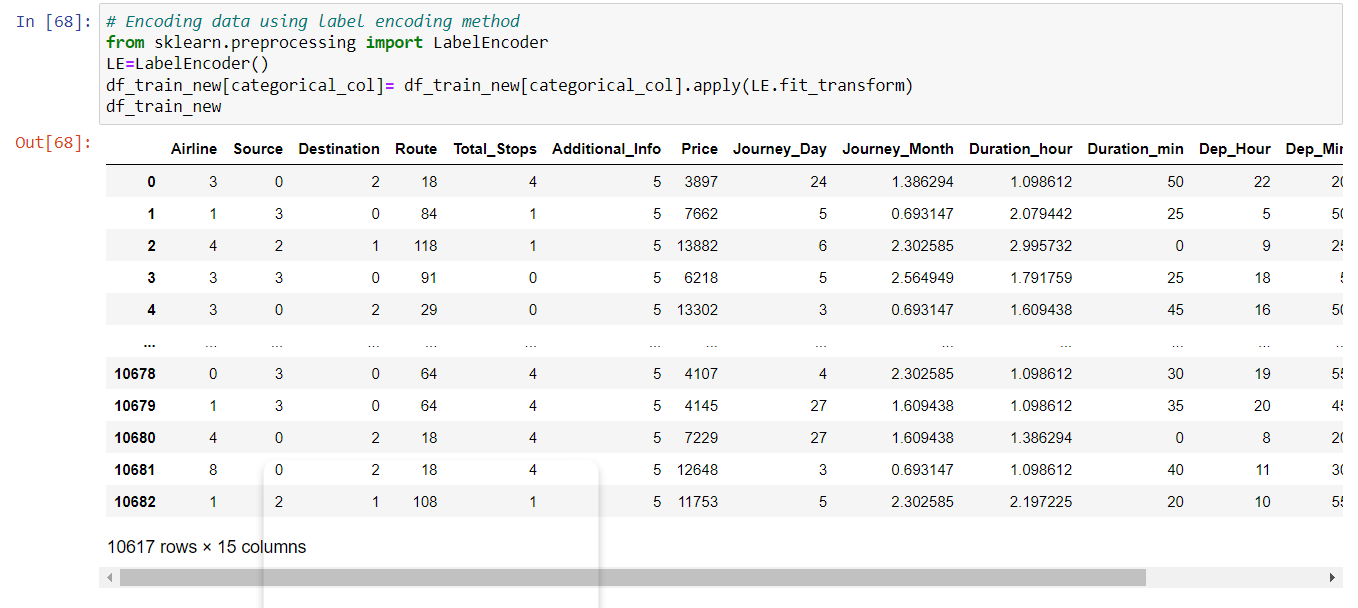
The skewness has been removed in Journey\_Month and Duration\_hours.





I have removed the skewness and using distribution plot we can that the data looks normal. As the dataset contains numerical and categorical data so now I have changed the categorical data into numerical form using Label encoding.



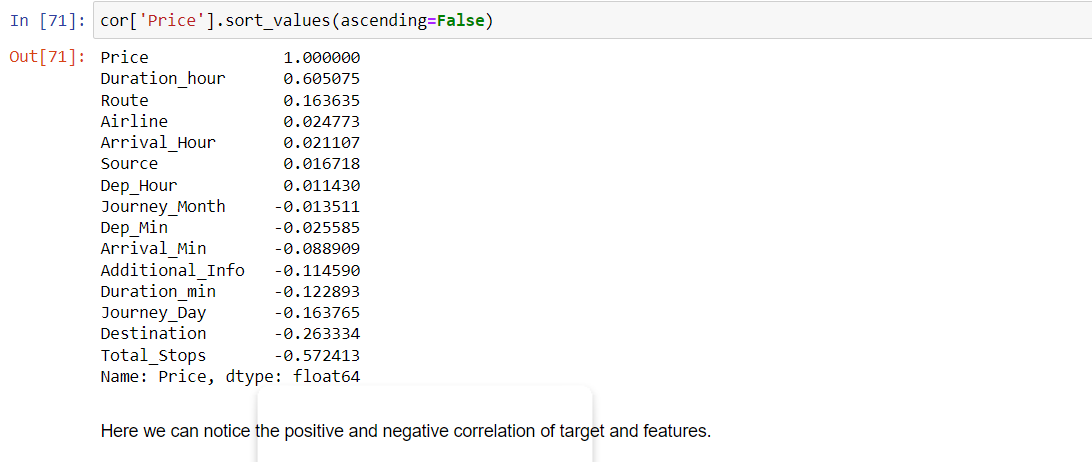


Now I have converted categorical data into numerical data we now get the heat map of our dataset.

This heatmap shows the correlation matrix by visualizing the data. we can observe the relation between one feature to other.

* This heat map contains both positive and negative correlation.
* The feature Duration\_hours is highly positively correlated with the target variable Price.
* The feature Total\_Stops is highly Negatively correlated with the label.
* The features Duration\_hours and Total\_Stops, Duration\_hours and Destination are highly negatively correlated with each other. This may lead to multicollinearity problem so we will check the VIF value to solve this, if we get the features having VIF more than 10 then we can drop those columns.

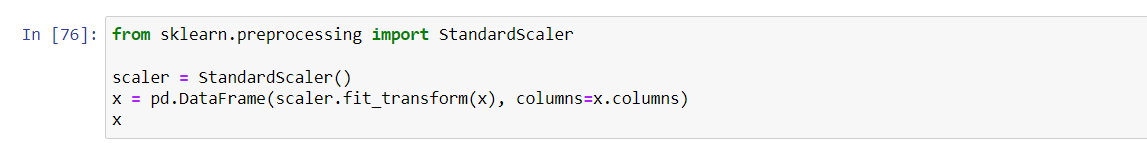


Preprocessing Pipeline:



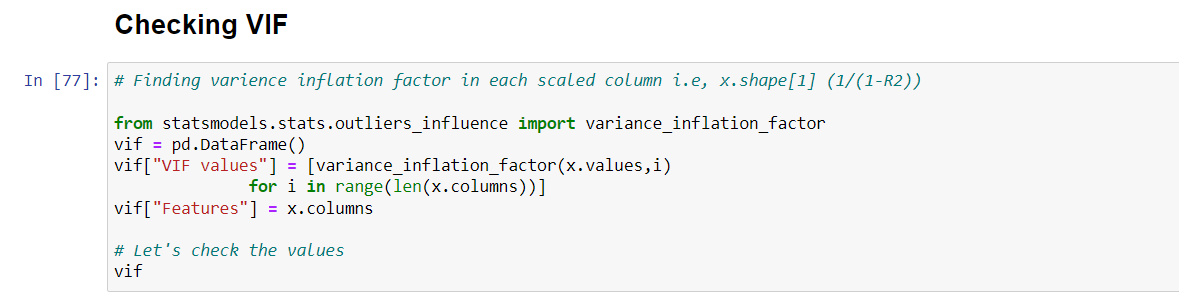
## I have separated feature and label into x and y and checked for their shapes.

## Since the skewness of the data is in the acceptable range and the data is also normally distributed in the columns, in such case we can make use of Standard Scaler method else we can make use of Min Max scaler method.



We have scaled the data using standard scaler method to overcome with the issue of data biasness.

In the heat map we have found some features having high correlation between each other which leads to multicollinearity problem. In order to solve multicollinearity problem, we will check VIF values. If we find VIF values greater than 10 in any features that means the features causing multicollinearity issue in the data. To overcome with this problem, we need to drop that particular feature column.





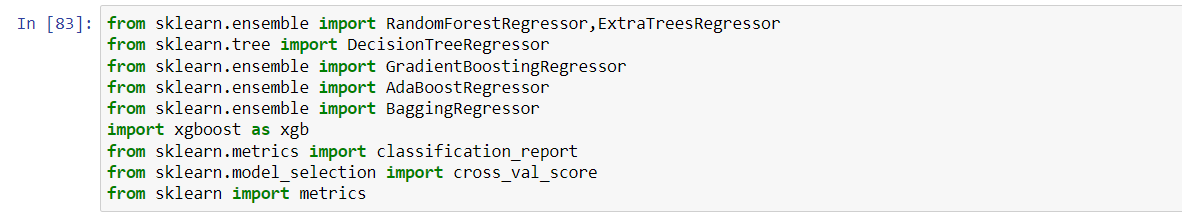
In the heat map we have found the multicollinearity problem but after scaling data using standard scaler method, we can observe the none of the columns have VIF above 10 which means our data is free from multicollinearity problem.

Since we have done all the data analysis, EDA and pre-processing, now it is time to build our machine learning models.

Building Machine Learning Model:

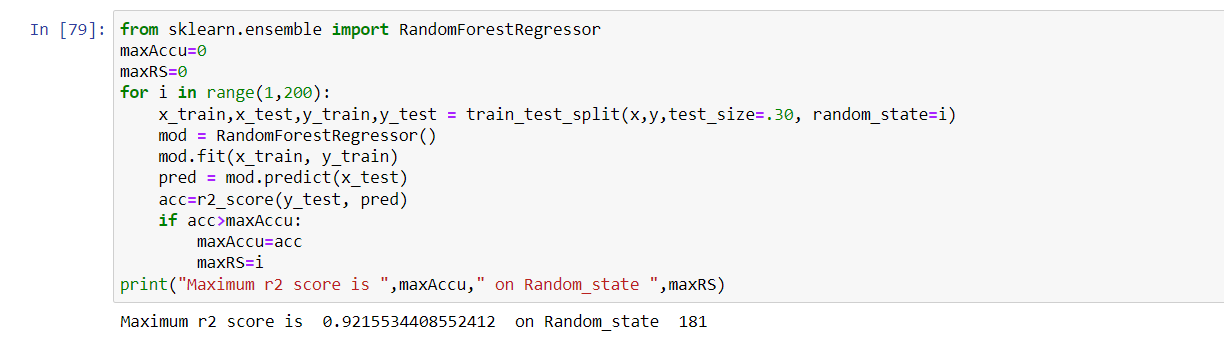
Machine learning (ML) is a type of artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning [algorithms](https://whatis.techtarget.com/definition/algorithm) use historical data as input to predict new output values.

The main goal in this step is to develop a benchmark model serves us as a baseline, upon which we will measure the performance of a better and more tuned algorithm. We are using different Regression Technique and comparing them to see which algorithm is giving better performance.





All the useful libraries have been imported using sklearn library. In this I am using 6 machine learning algorithm to predict the flight price prediction. The model which will give us the best performance that will be considered best model for flight price prediction. Now lets start with finding the best random state and check the R2 score using the Regression algorithm.

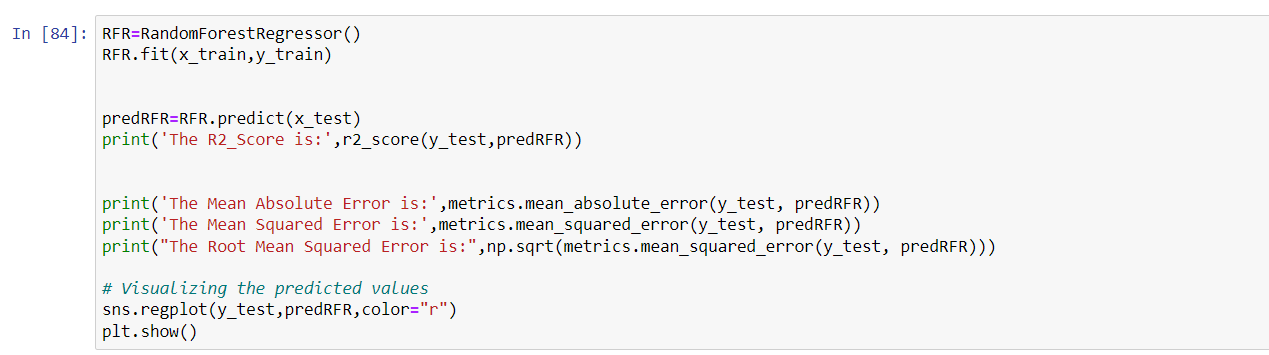


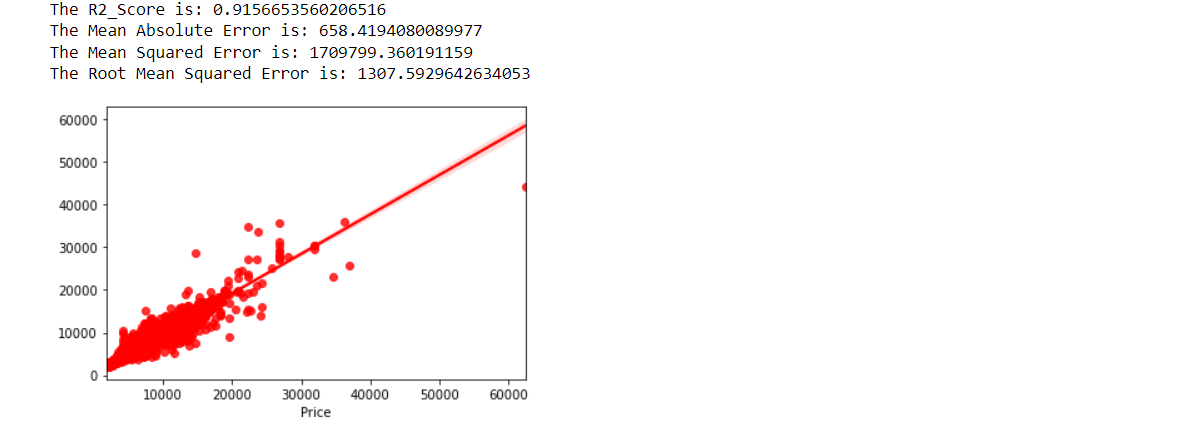


After finding the best Random state we will move further with train test split function to create training and testing dataset so that we can use it in our models.

**Random Forest Regressor**

Random Forest is an ensemble machine learning technique capable of performing both regression and classification tasks using multiple decision trees and a statistical technique called **bagging.** All calculations are run in parallel and there is no interaction between the Decision Trees when building them.

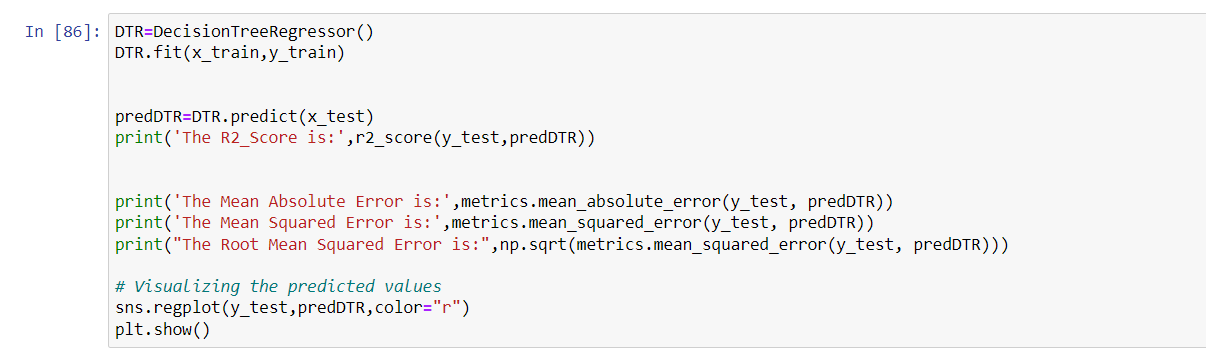


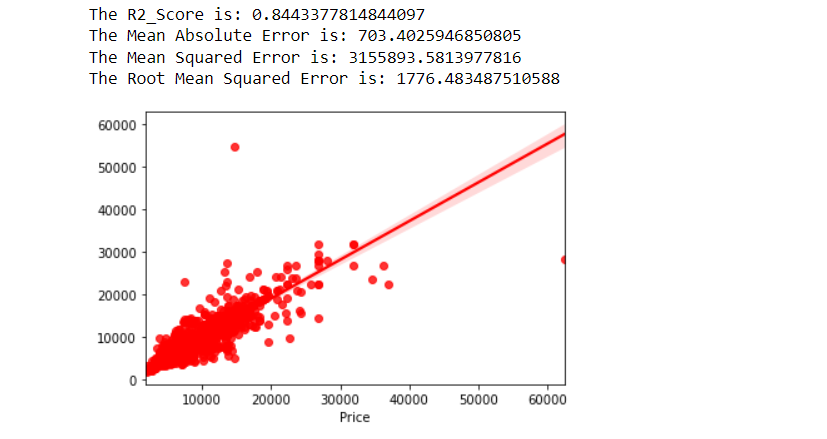


So we have got 91.56% score in our Random Forest Regressor algorithm.

**Decision Tree Regressor**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too. In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record’s attribute.

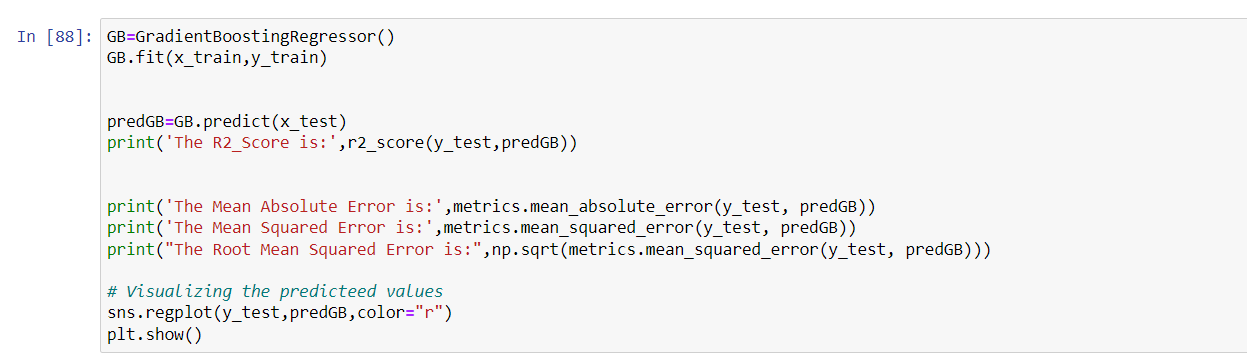


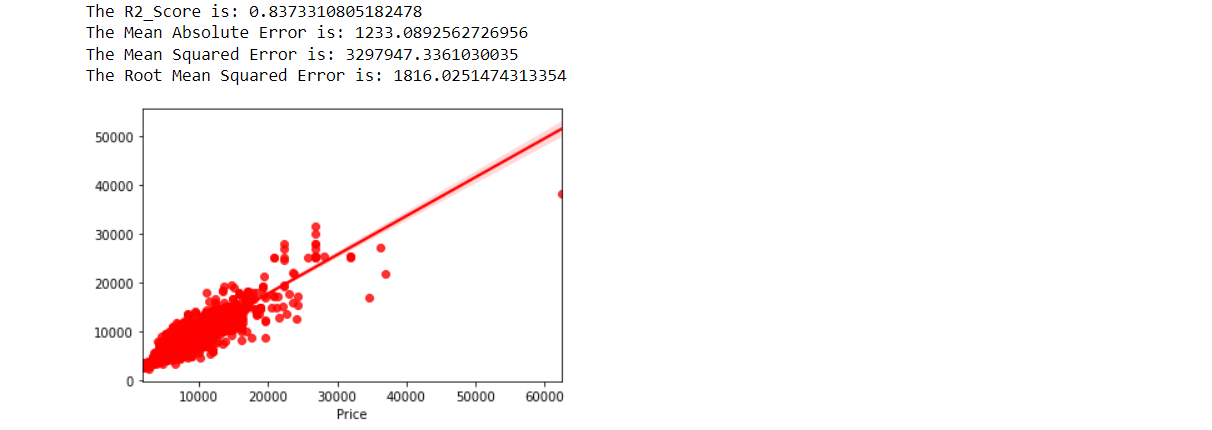


So, we got 84.43% score in our Decision Tree regressor algorithm.

**Gradient Boosting Regressor**

Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting algorithm can be used to train models for both regression and classification problem. Gradient Boosting Regression algorithm is used to fit the model which predicts the continuous value.

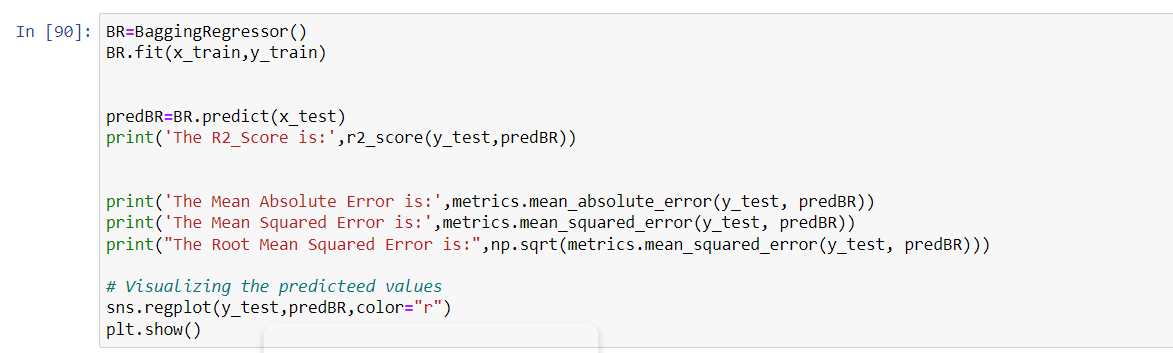


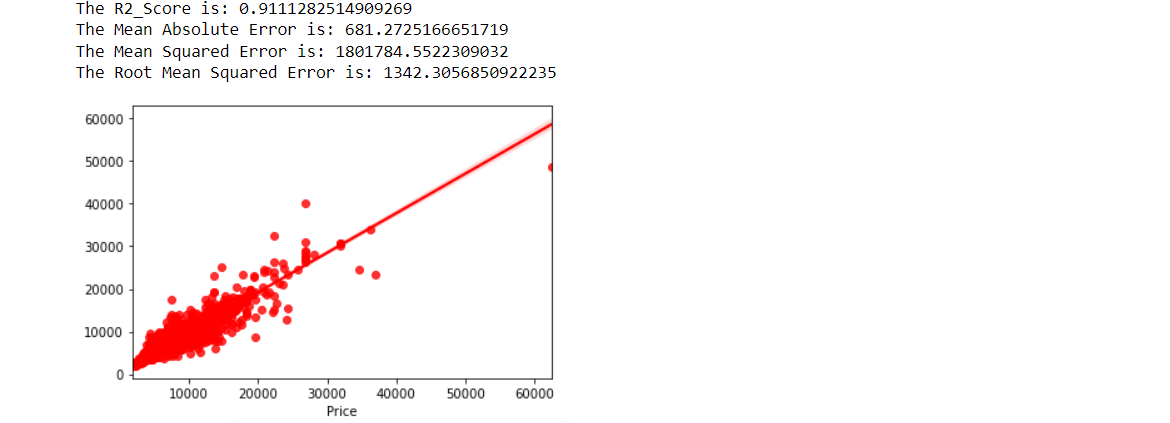


So we got 83.73% score for our Gradient Boosting Regressor algorithm.

**Bagging Regressor**

A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions to form a final prediction.

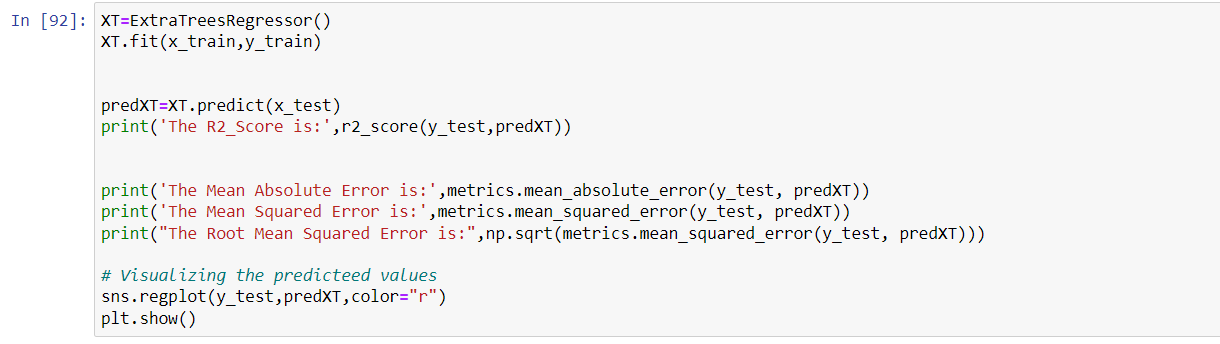


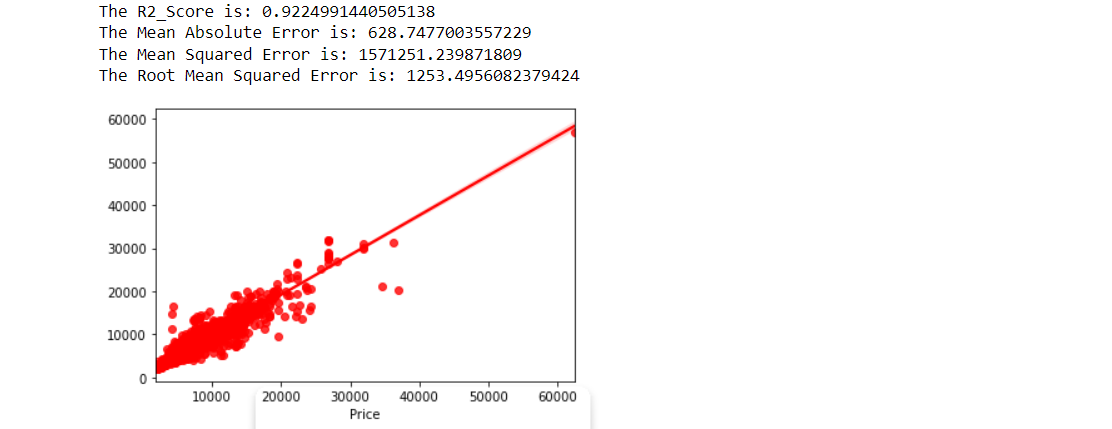


So we got score of 91.11% for Bagging Regressor.

**Extra Tree Regressor**

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.





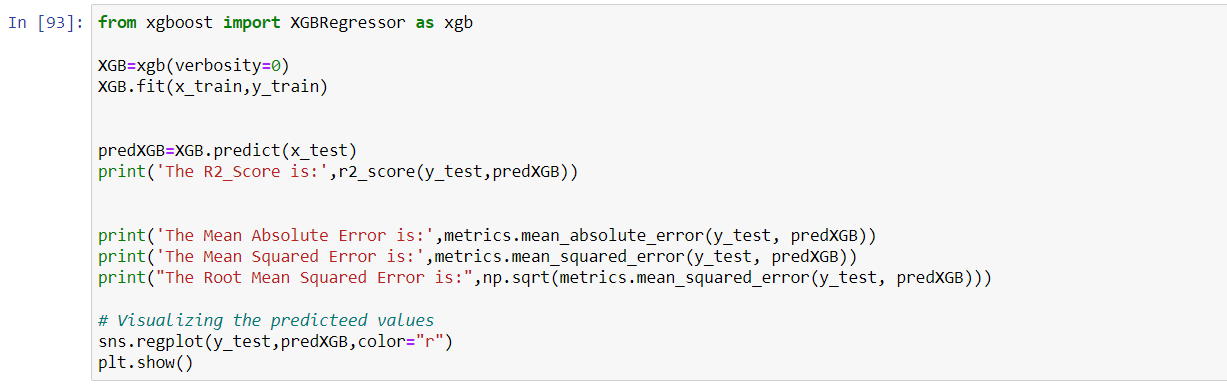
We got score of 92.24% for ExtraTrees Regressor.

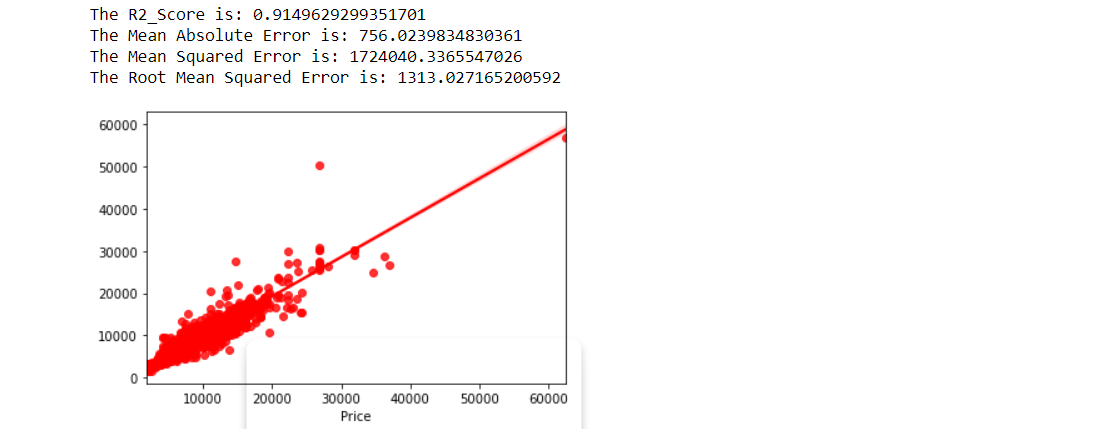
**Extreme Gradient Boosting Regressor (XGB Regressor)**

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.

Shortly after its development and initial release, XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions.

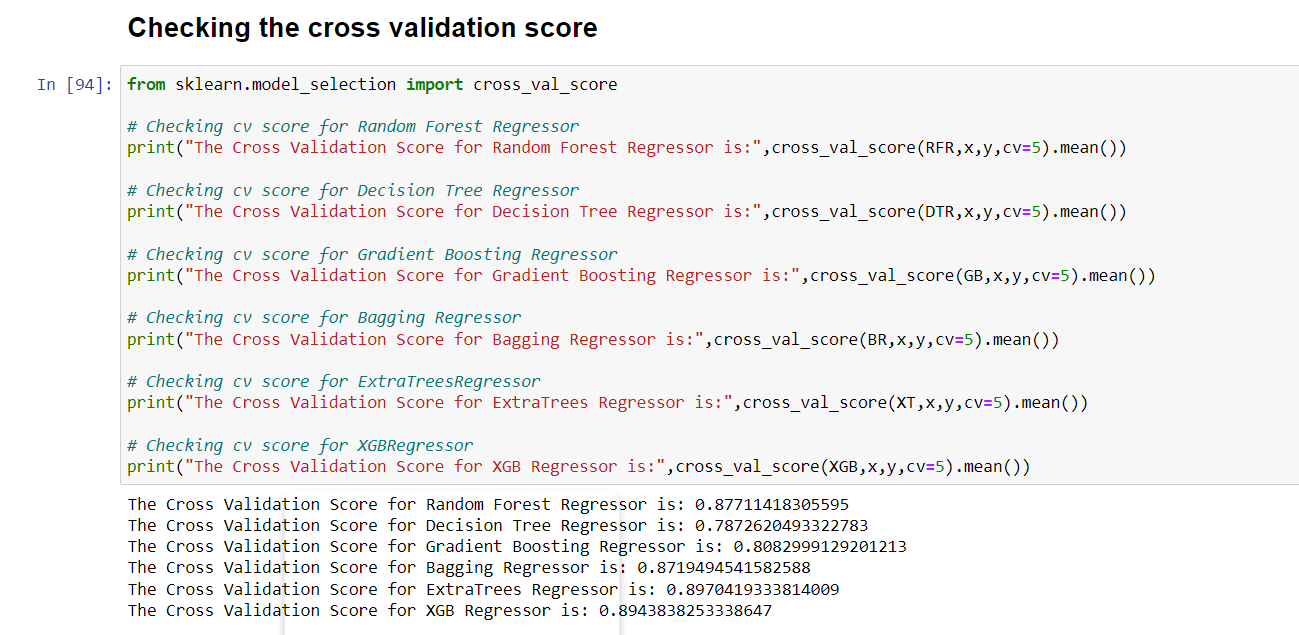
Regression predictive modeling problems involve predicting a numerical value such as a dollar amount or a height. **XGBoost** can be used directly for **regression predictive modelling.**





We got score of 91.49% for XGB Regressor.

We have successfully built the models by training the data using x\_train and y\_train and with the help of x\_test we have got the prediction for every models. Also, we have got the R2 score with the help of prediction test and y\_test. Based on every model, Extra Tress Regressor has best R2 score as 92%. This could be because of overfitting. In order to check if the model is overfitted or not, we need to perform cross validation.



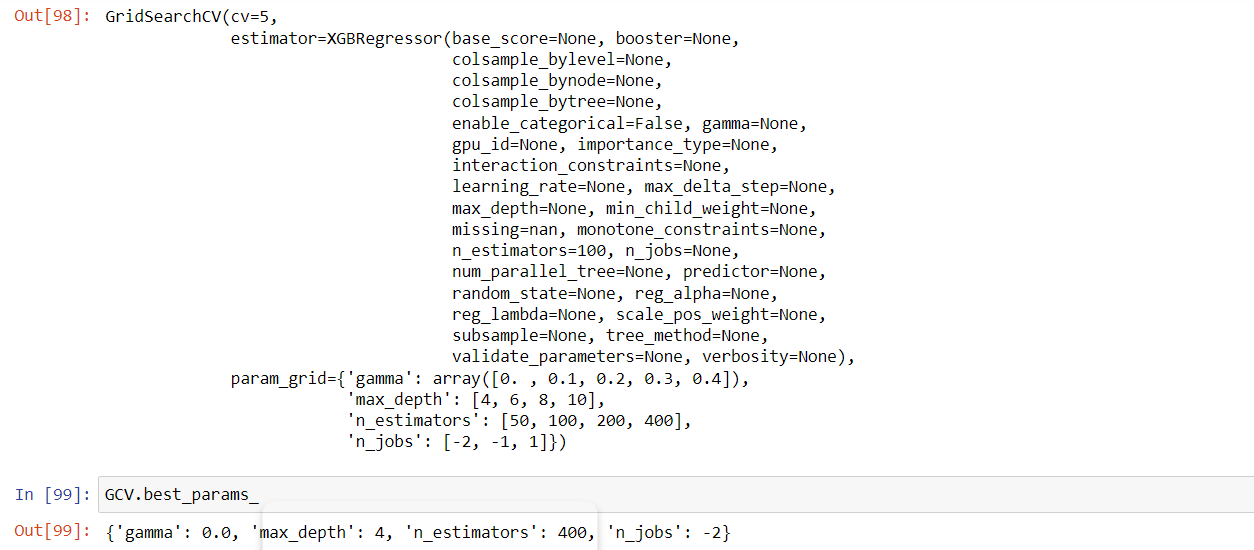
**Difference between R2 Score and Cross Validation Score:**

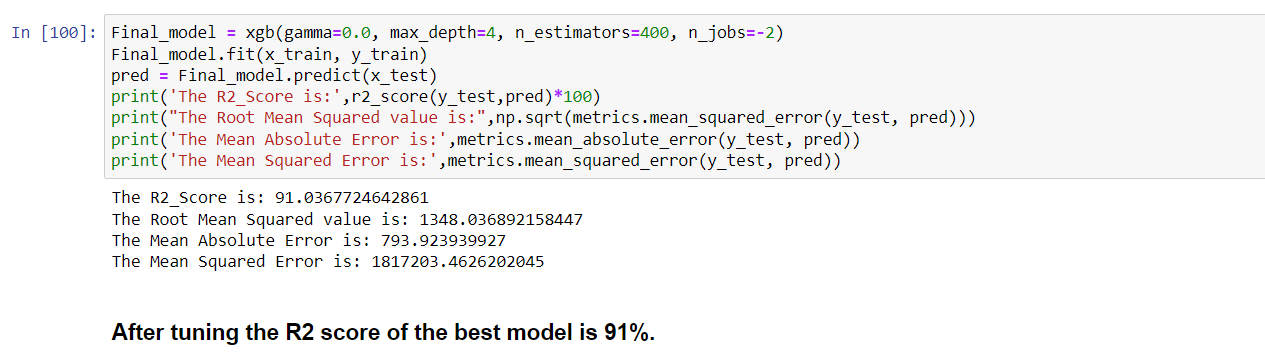
|  |  |  |  |
| --- | --- | --- | --- |
| **Models** | **R2 Score** | **CV Score** | **Difference** |
| Random Forest Regressor | 91.56 | 87.71 | 3.85 |
| Decision Tree Regressor | 84.43 | 78.72 | 5.71 |
| Gradient Boosting Regressor | 83.73 | 80.82 | 2.91 |
| Bagging Regressor | 91.11 | 87.19 | 3.92 |
| Extra Tress Regressor | 92.24 | 89.43 | 2.81 |
| XGB Regressor | 91.49 | 89.43 | 2.06 |

The model XGB Regressor giving very less difference compared to other models.

Since XGB Regressor is giving best in R2 score and CV score difference, Evaluation metrics, so we choose XGB Regressor as best fitting model. Let’s check whether we can increase the R2 score by using hyper parameter tuning.



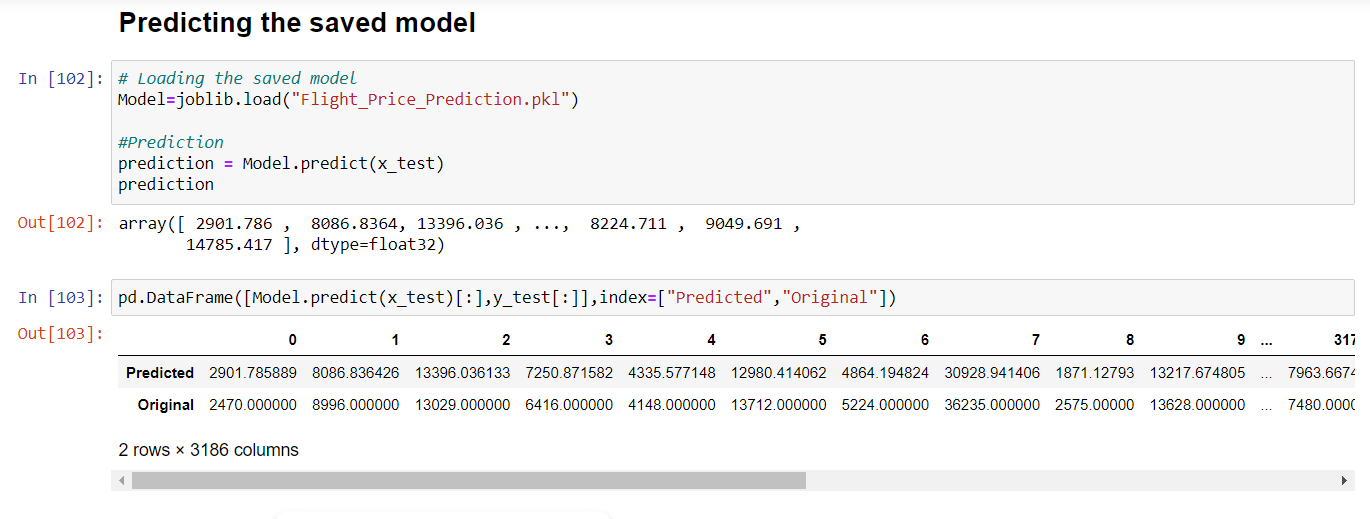




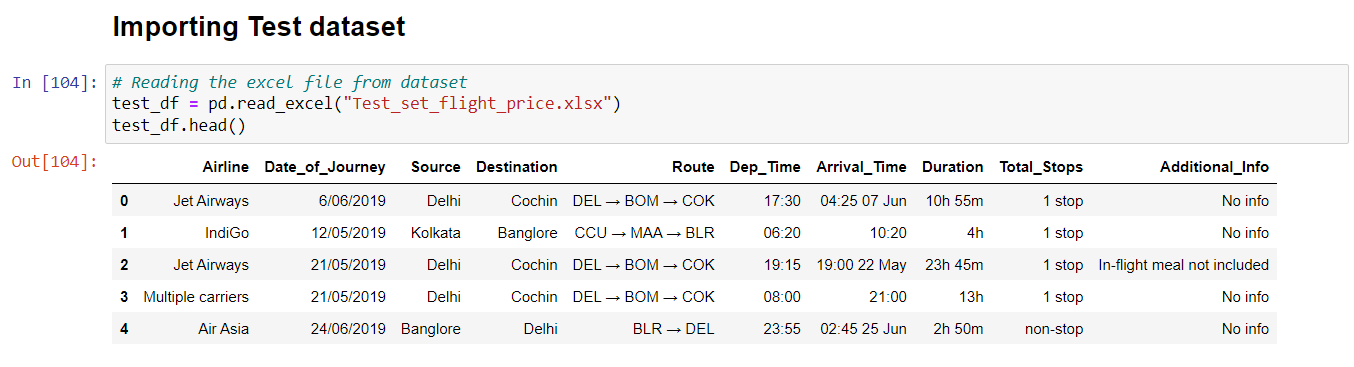
So as we can see above that we tried increasing our model score by doing Hyper Parameter Tuning with the best parameters for increasing our R2 Score but we can see that we have got 91% score after using the best parameters which is almost the same this may be because of the parameter which we used.

So, We have done the model building and also performed the Hyper Parameter Tuning .Now we just need to save the model and then reuse it for processing Test Data.



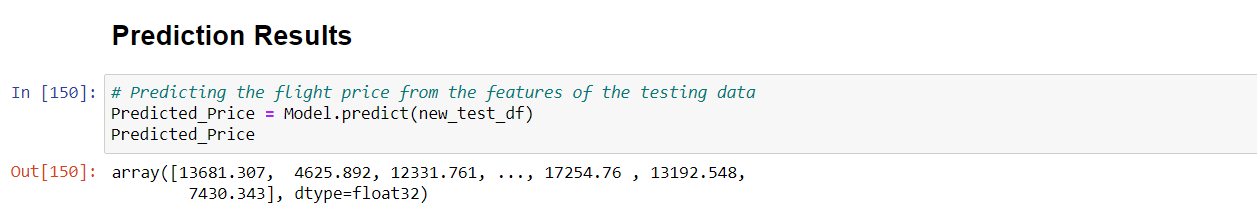


**Test Data:**



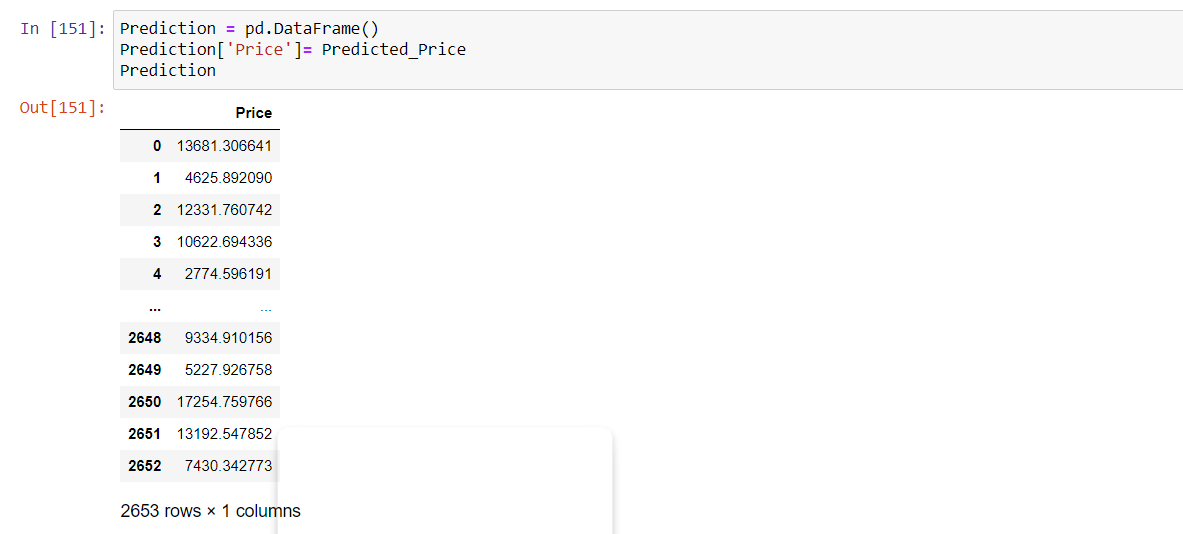
* The test dataset contains 2672 rows and 10 columns. It is having all the same columns as train data except target column.
* You need to perform same steps that we have performed for train dataset, that includes Data Analysis, EDA, Scaling the data and Pre-Processing. After doing all these steps, no need to build any model using test data, only cleaning the data is required.
* Once you have done all these steps, all you need to do is to predict the flight price by using loaded trained data.

**Prediction Result:**



I have used the loaded Model to predict the flight ticket price of the test dataset.

Now let’s compare the actual and predicted ticket price by creating dataframe.



I have got the predicted flight price ticket for testing dataset and by using above code, I have added the predicted price output to our original test dataset to complete it with feature and target column.

Conclusion Remarks:

So we have successfully done the Feature engineering process which is very important for finding the correct information about our Dataset. We have removed the Outliers and Skewness in our dataset and also we have converted our categorical data into numerical data . We have scaled our data and then we moved forward towards creating different regression models to predict flight price and also I have done the hyper parameter tuning to improve the model by using different parameters.

With the help of above methods I can say that my model is able to predict the flight price by R2 Score of 91%. Also we have seen that our predicted price and actual price are almost same which means our model prediction is correct. This can be very helpful for the customers for predicting flight prices and then planning their journey accordingly. It will also help airlines in maintaining flight prices so that they can have maximum reach to their customers . So we can now see that how much Machine Learning technique can be very helpful in solving lots of this kind of problems.