**FLIGHT PRICE PREDICTION – MACHINE LEARNING PROJECT**

**PROBLEM STATEMENT:**

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. Airline companies use complex algorithms to calculate the flight prices given in various conditions present at that particular time. These methods take financial, marketing, and various social factors into account to predict flight prices.

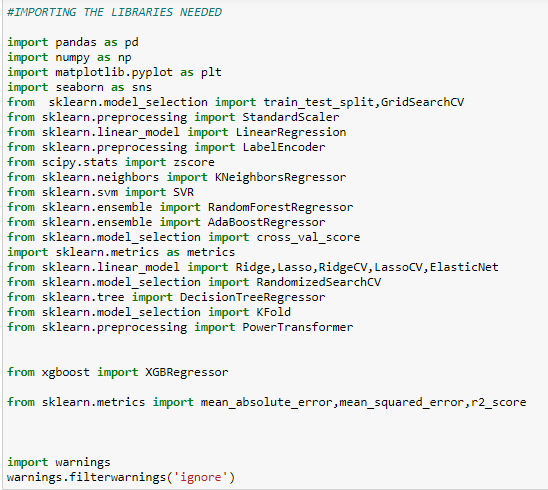
Nowadays, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. That’s why we will try to use machine learning to solve this problem. This can help airlines by predicting what prices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

In this blog post, I will go through the whole process of creating a machine learning model on the dataset. The dataset has the information on the price the ticket was purchased along with source, destination and route the aircraft has taken.

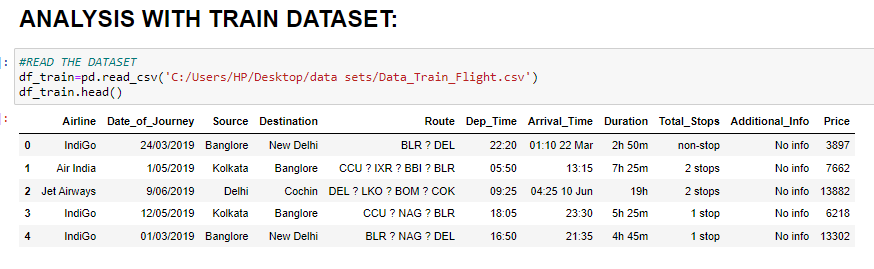
*Flight prices are based on demand and availability. You may see that prices are high during weekend or during vacations as there is an increase in demand for holiday or leisure travel. Sometimes we notice that airline will offer promotional fare as they are expecting low availability during that period and want to lure the customer by offering discounts.*

Since the price is continuous, we will require regression models to predict the prices.

***Importing the Libraries, we will use:***



***Now we will Load the data***:



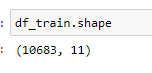
Someone who purchase flight tickets frequently would be able to predict the right time to procure a ticket to obtain the best deal. Many airlines company change ticket prices for their revenue management. The airline may increase the prices when the demand is to be expected to increase the capacity. To estimate the minimum airfare, data for a specific air route has been collected including the features like departure time, arrival time and airways over a specific period. Features are extracted from the collected data to apply Machine Learning (ML) models.

The document contains the data with features and its details. A significant perspective is to choose the features required for calculation of expected flight price. Output gathered by reading the dataset contains number of parameters for each flight, so just accompanying the components are -

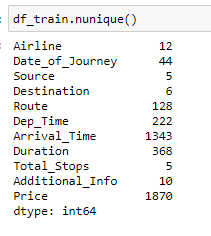
* Date of journey- The date of the journey
* Source- The source from which the service begins.
* Destination- The destination where the service ends.
* Route- The route taken by the flight to reach the destination.
* Time of Departure- The time when the journey starts from the source.
* Time of Arrival- Time of arrival at the destination.
* Total stops- Total stops between the source and destination.
* Airline- The name of the airline.
* Additional info- Additional information about the flight
* Price- The price of the ticket(target).

In this train dataset we got all the independent features as categorical form and only the target column is in continuous form. We will encode all the independent features to numerical form before putting it into the model.

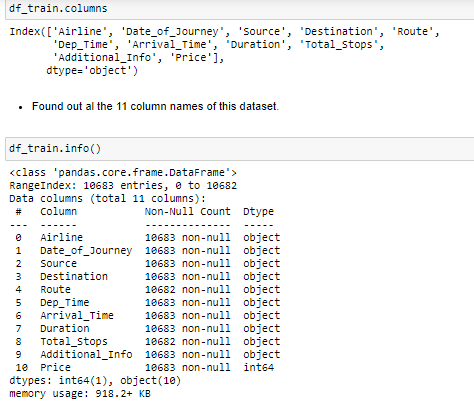
* ***Data Analysis:***



We can see that we have 10683 rows and 11 columns in this dataset, that means we have 10683 entries in total for analysis and predict the flight price.

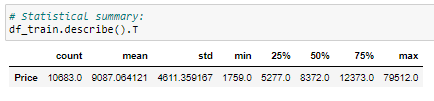


* There are twelve unique airline companies given in this dataset.
* There are five sources given, from where the flight starts.
* There are six destinations where the flight reaches after starting from the source.
* There 128 unique routes from where the flights move to and fro.
* There are five kinds of stops in the dataset, which are 1-stops,2-stops,3-stops0,4-stop and nonstop.



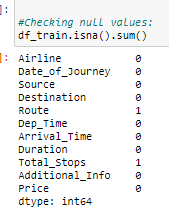
* In the above image, we used df. columns to find out all the column names present in this dataset.
* We can also see that each and every column is object in nature except the target column "Price".

Checking the Statistical Summary:

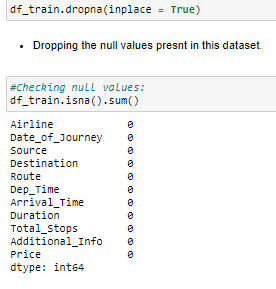


* By df. describe (), we can check the statistical summary of the continuous features.
* We can see that only Price feature is showing in statistical summary as it is the only continuous feature of the dataset. It may have Right skewness because the difference between the 75% value and the max value is very high.

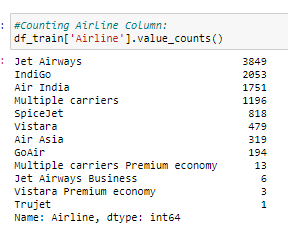
Checking for Null Values:



While checking the null values we can see that there is one null value each in Route and Total stops column, we will remove the null values now.



Dropped the null values by using .dropna function,used it because there was only one null values each in two columns.

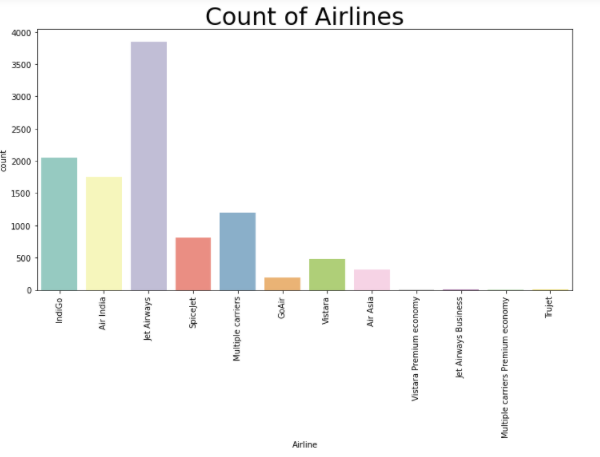


We can see from the above image that; the train dataset contains the information about twelve airline company from which Jet Airways is having the highest number of flights moving and Trujet Airline Company have only one flight moving.

**EDA CONCLUDING REMARK**:

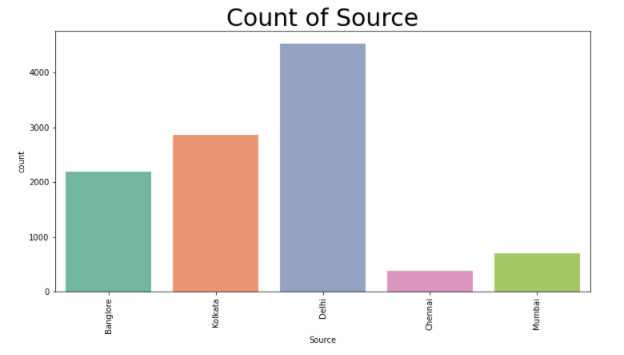
**UNIVARIATE ANALYSIS:**

PLOTTING THE AIRLINE COLUMN:

****

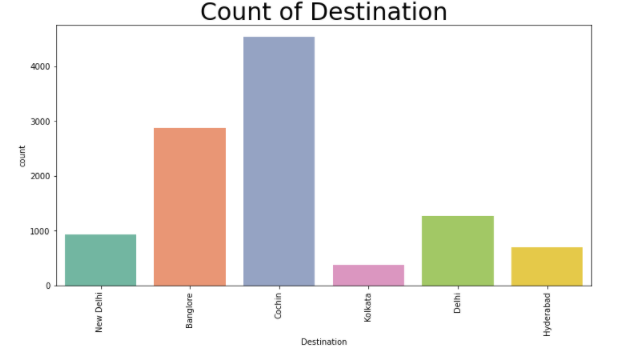
We can see from the above image that; the train dataset contains the information about twelve airline company from which Jet Airways is having the highest number of flights moving and Trujet Airline Company have only one flight moving.

PLOTTING THE SOURCE COLUMN:



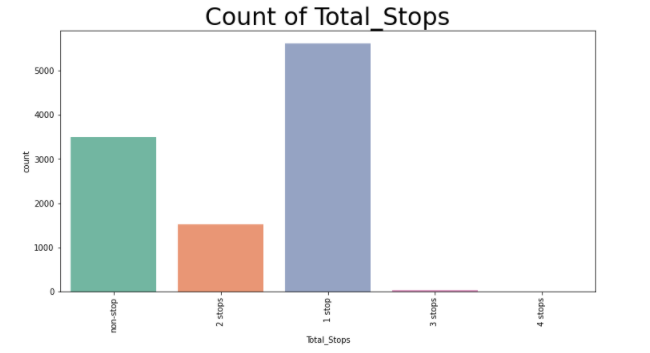
* There is main five sources from which the flights start, they are Delhi, Kolkata, Bangalore, Mumbai and Chennai.
* Most of the flights starts from Delhi. Delhi is the busiest airport found in this dataset.
* Source Chennai is having the least flights.

PLOTTING THE DESTINATION COLUMN:

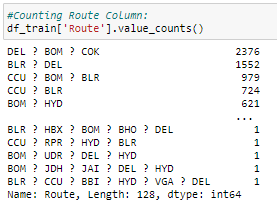


* There are five destinations, the last stop, found in this dataset, which are Cochin, Bangalore, Delhi, New Delhi, Hyderabad and Kolkata.
* The highest destination found is of Cochin Airport.
* The least destination found is of Kolkata Airport.

PLOTTING THE TOTAL STOPS COLUMN:

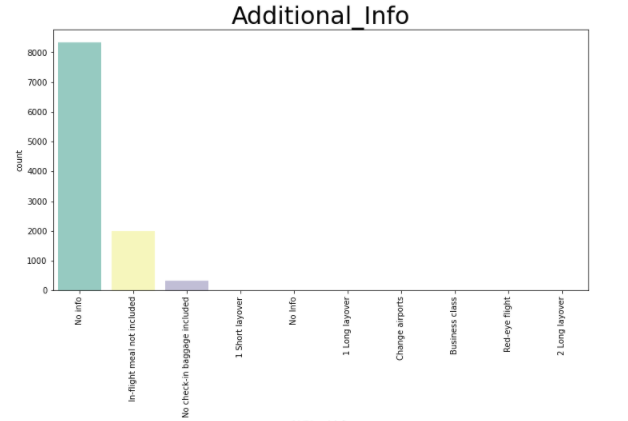


* There are five types of stops found, they are 1 stop, non-stop,2 Stops,3 stops and 4 Stops.
* Most of the flights travel from source to destination by taking only one stop.
* There is only one flight which is having four stops.



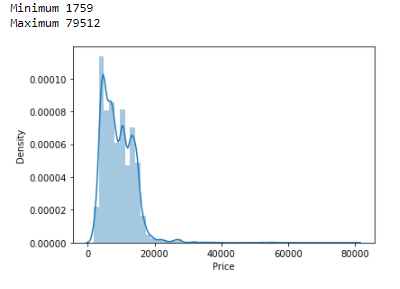
We can see that there are 128 unique routes in this dataset.

PLOTTING THE ADDITIONAL INFO COLUMN:



We can see that 80% of the column is having ‘no info’.

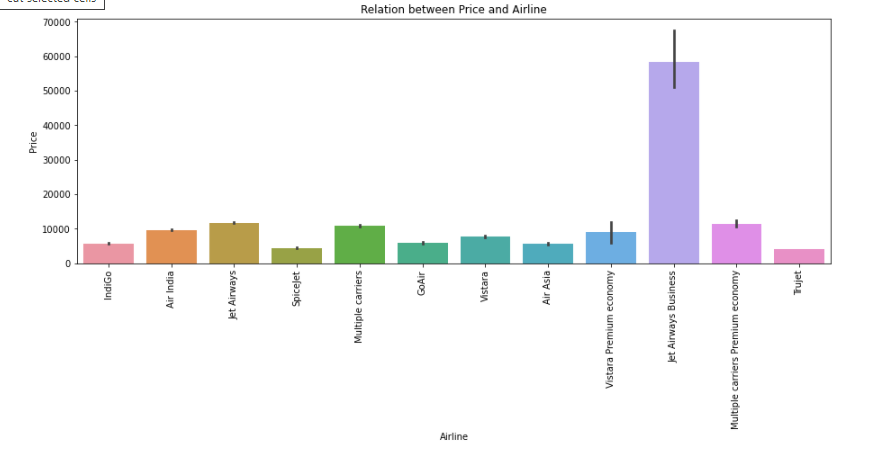
PLOTTING THE PRICE COLUMN:



We can see that the maximum price is 79512 and the minimum price is 1759.we can also see that there is Right skewness in the column, we will remove the skewness latter by outlier removal techniques.

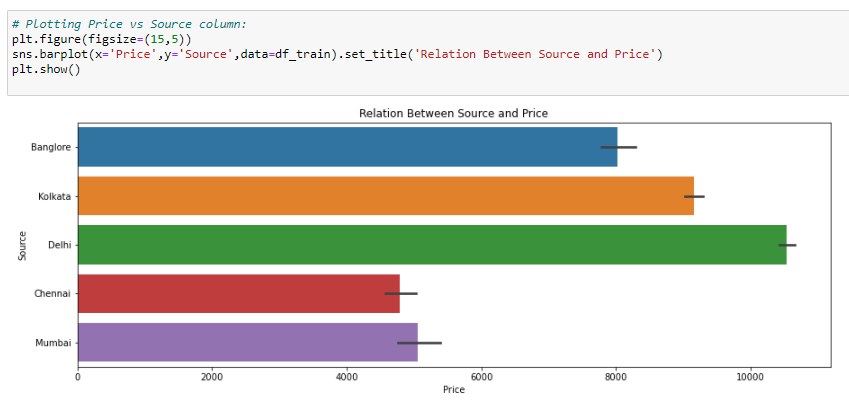
BI VARIATE ANALYSIS:

PLOTTING PRICE VS AIRLINE FEATURE:



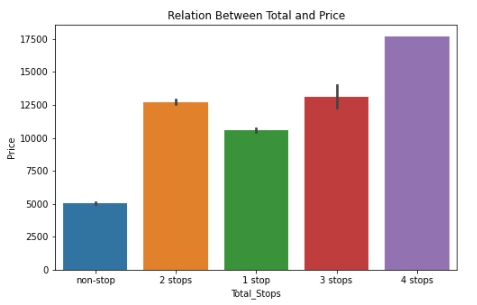
It is clear that Jet Airways Business class is having the highest prices, because it is the business class and it is so high compared to other classes because business fares usually include **an element of cancellability** so you are not just paying for 50% more leg room and larger peanuts but the freedom to abandon the booking at the last minute, forcing the airline to discount it heavily.

PLOTTING PRICE VS SOURCE COLUMN:



Indira Gandhi International Airport (DELHI) isn’t just the busiest airport in India but also ranks 12th among the busiest airports in the world. In the year 2018, the aviation hub catered to more than 69 million people, besting airports in Singapore, Dallas/Fort-Worth and Guangzhou. The airport is functioning with three runways, out of which one is the longest runway in the country, stretching 4430 feet. The IGI Airport has three terminals. Moreover, the airport is also accredited as the largest airport in India, hence it is sure that most flights are having source from this Airport and Chennai is least busy airport from the above five cities.

PLOTTING TOTAL STOPS VS PRICE FEATURE:



For the airlines, 1stop is theoretically a cheaper and more **reliable system because it avoids the risk of a nonstop route with unreliable demand**. And instead of flying long-distance routes that necessitate larger aircraft, carriers can fly short routes and use smaller (and subsequently fuller) planes. But here in this data it is not following the thumb rule, we can see that 4 stops flights are more expensive, may be because of the more fuel consumption.

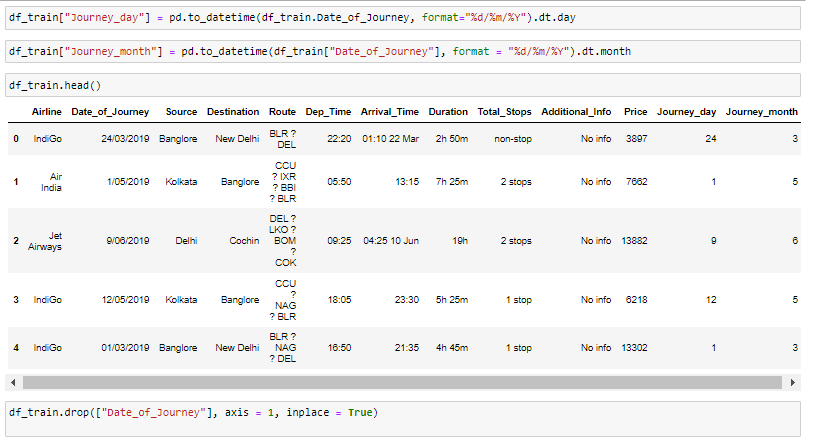
PLOTTING DESTINATION VS PRICE COLUMN:



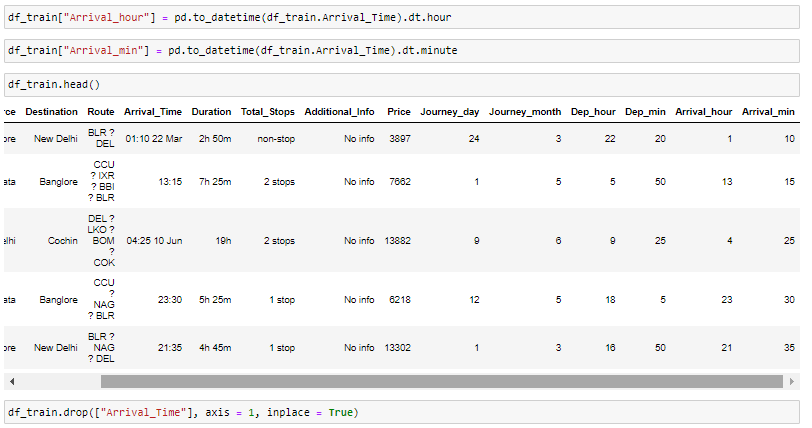
Same as the source as Indira Gandhi Airport is the busiest airport in India it is quite expected that the Destination data too will be highest. Less people choose Kolkata as their destination.

FEATURE ENGINEERING:

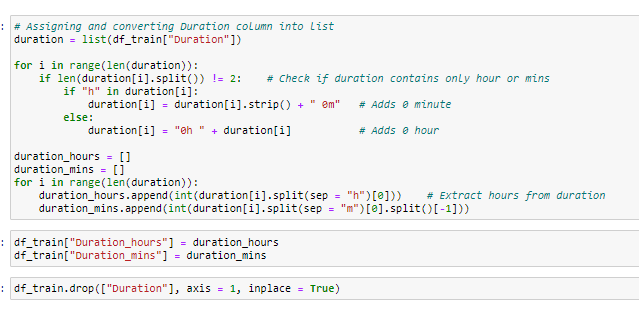
* There is lot of data cleaning and data processing we are going to perform, from description we can see that Date\_of\_Journey, Dep Time and Arrival Time is object data type; Therefore, we have to convert this datatype into timestamp so as to use this column properly for prediction.
* For this we require pandas to\_datetime to convert object data type to datetime datatype.
* .dt.day method will extract only day of that date and .dt.month method will extract only month of that date.
* .dt.hour will only extract the hour and .dt.minute will only extract the minutes.



Separated the Date of journey column to create Journey Day and journey month and deleted the Date of Journey column.

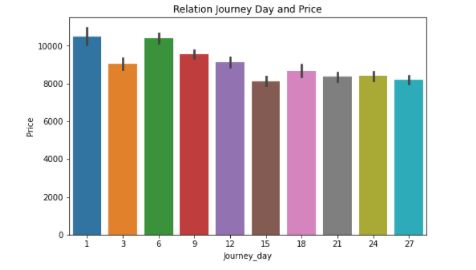


Similarly ,Separated the Arrival Time column into Arrival hour and Arrival minute column and deleting the Arrival time column.



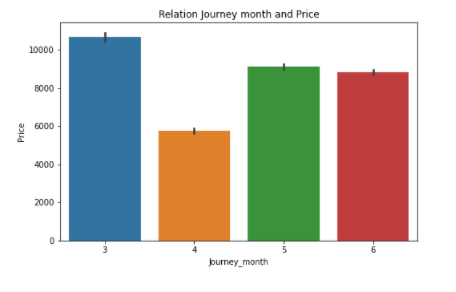
Moving on now we will check the Duration column. The duration column contains the total travelling time in hours and minute i.e., 2h 50m. I will separate the Duration column into Duration hours and Duration Minutes by splitting it with. split function. Which will in whole help to scale up, so that it will be helpful for prediction. And after that we have deleted the Duration column. Now I will plot relations with the target feature price with the newly made column.

PLOTTING PRICE VS JOURNEY DAY COLUMN:



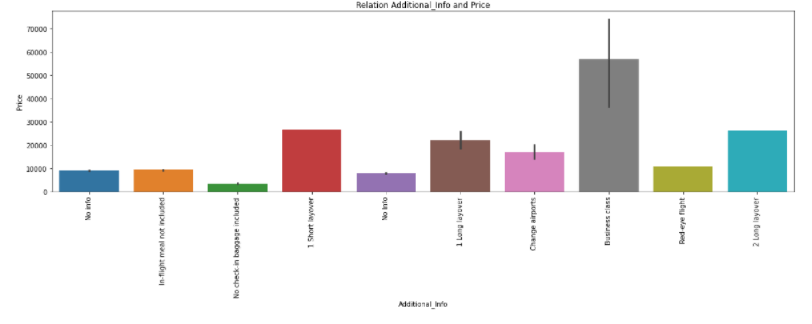
We can come to the point that the prices starting of the month is higher than the latter days of the month.

PLOTTING THE JOURNEY MONTH VS PRICE COLUMN:



In this dataset we got the data of the month March, April, May, June (3,4,5,6), we can see that the prices in the month of March is highest and the prices in the month of April is lowest. Remember I said in the beginning, that generally the flight prices are higher during vacation or weekend and lower during the non-vacation period. Vacations in India are during Christmas, Diwali and summer holidays. Hence, we will update them as high season. Rainy season will be marked as Low and remaining period we will update that as Shoulder season.

PLOTTING ADDITIONAL INFO VS PRICE COLUMN:



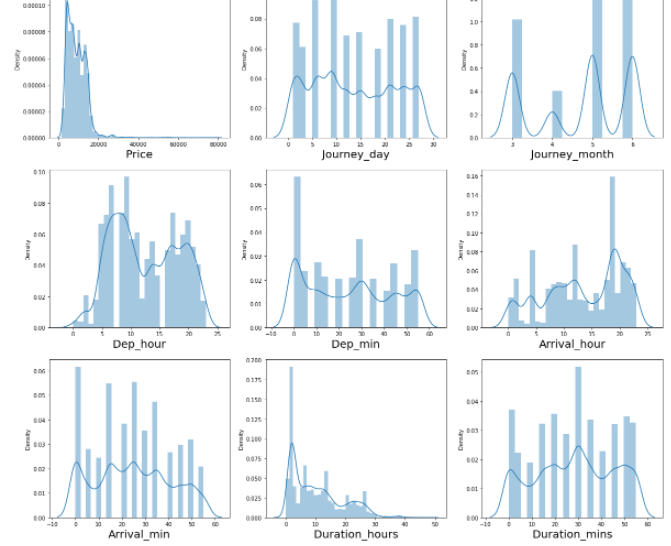
* Additional Info also show that price is higher for business class and the lowest for No check-in baggage.

**Remarks:**

1. Since there is no data for the month of Low season, we can clearly see that the price is higher during the High season.
2. Flight with 1 stop has the higher price. This could be because of the cabin customer is flying on. Maybe business class.
3. We can observe that fares are high for Jet airways business. True Jet has the lowest price.
4. Price will be higher if the Source is BLR.
5. Destination Delhi will have the higher ticket Price.
6. Additional Info also show that price is higher for business class and the lowest for No check-in baggage.
7. Flights departing early morning tend to have the higher price. Lowest price is available for the flights departing late night.
8. Flights arriving in the morning have higher price.
9. There is no correlation we can find with travel time and price.
10. We should have more information on the type of fare customer has purchased and also the cabin the customer is using.

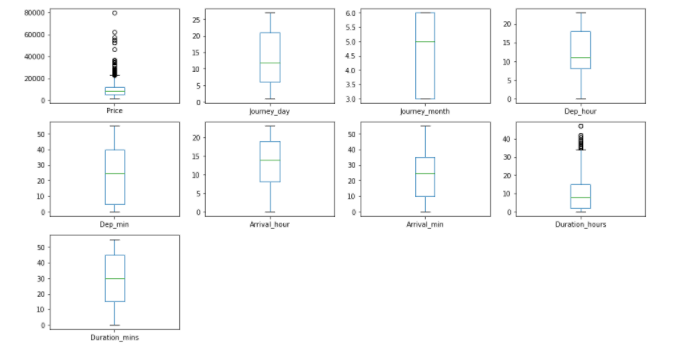
PRE-PROCESSING PIPELINE:

CHECKING FOR NORMAL DISTRIBUTION:



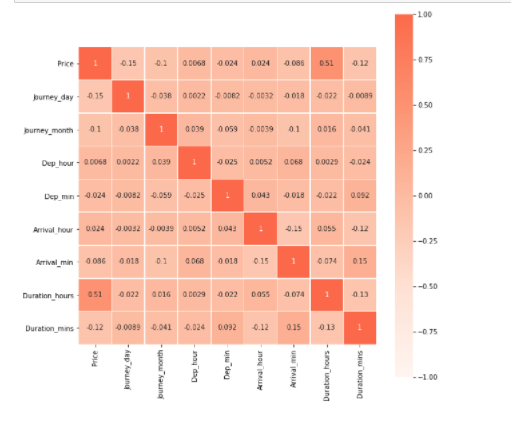
We can see that Price column is having Right skewness and more or less each and every continuous column is not in shape of normal distribution.

CHECKING FOR OUTLIERS USING BOXPLOT:



We can see that Price column and Duration hours is having outliers present, we will remove the outliers before putting it into the model.

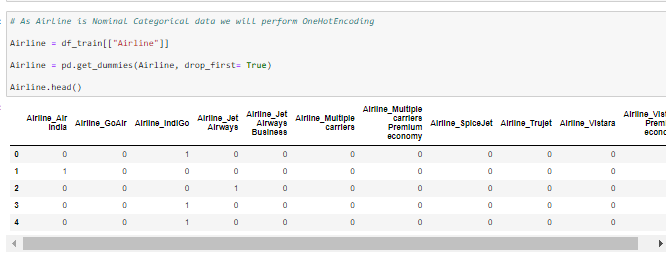
CHECKING FOR MULTI-COLINIEARITY BY USING HEAT MAP:

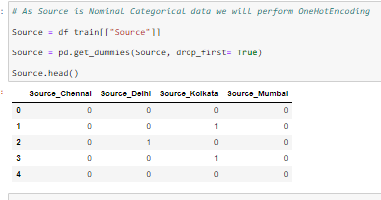


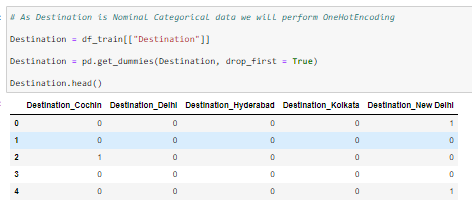
We can see that there is no such co-relation between the features, only price and Duration hours are somewhat co-related to each other.

CONVERTING THE CATEGORICAL COLUMNS INTO NUMERICAL:

As Airline, source, and destination is nominal in nature we will convert these columns into Numerical by using One hot encoding.





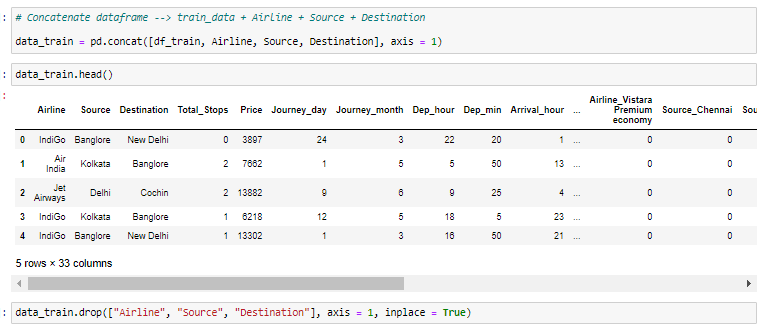


From the above images we can see that all the three columns are converted into numerical data.

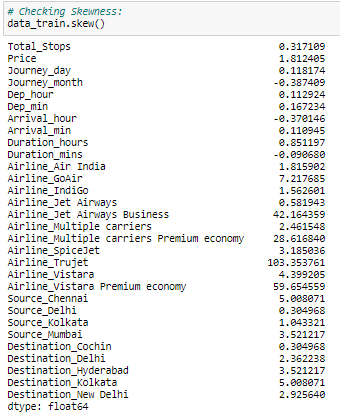
Now we will convert the Total Stops column as this is case of Ordinal Categorical type, we will perform LabelEncoder.Here Values are assigned with corresponding keys to maintain the unique type.



Now we will concatenate the data frame after converting the columns into numerical form and drop the columns which is already converted.

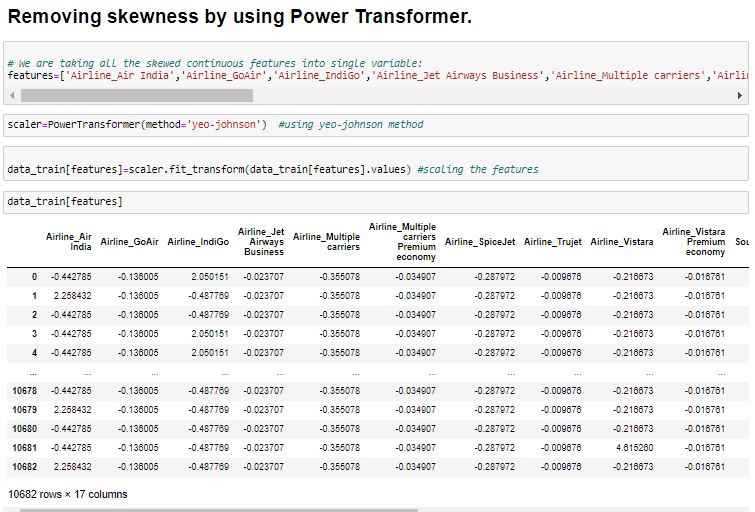


CHECKING SKEWNESS:

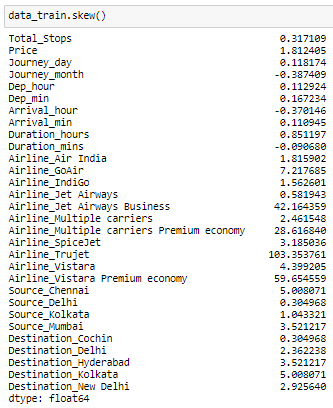


We can see huge skewness in Airline\_Air India, Airline\_GoAir, Airline\_IndiGo, Airline\_Jet Airways Business, Airline\_Multiple carriers, Airline\_Multiple carriers Premium economy,Airline\_SpiceJet,Airline\_Trujet,Airline\_Vistara,Airline\_Vistara Premium economy,Source\_Chennai,Source\_Kolkata,Source\_Mumbai,Destination\_Delhi,Destination\_Hyderabad,Destination\_Kolkata and Destination\_New Delhi columns,We will remove the skewness by using the power transfomer.

REMOVING SKEWNESS WITH POWER TRANSFORMER:

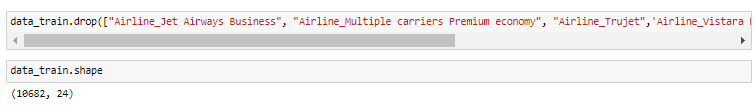


CHECKING SKEWNESS AGAIN:



We still have skewness in Airline\_Jet Airways Business, Airline\_Multiple carriers’ Premium economy, Airline\_Trujet, Airline\_Vistara Premium economy, Source Chennai and Destination\_Kolkata column. we will be dropping this column below.

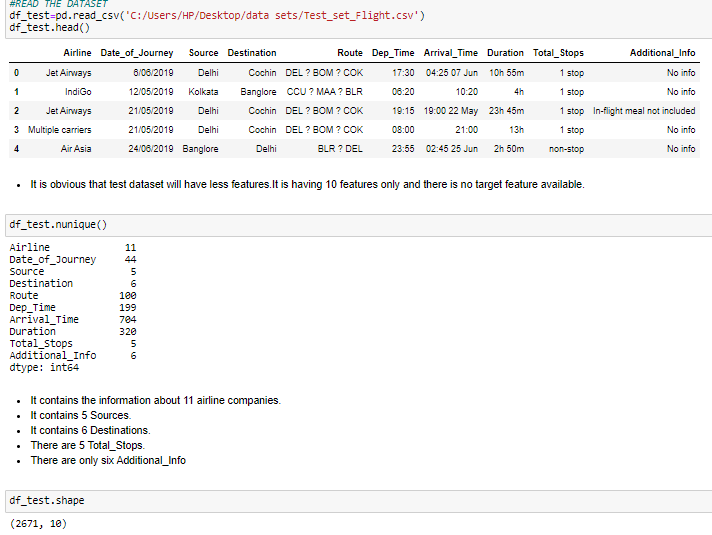
DROPPING THE SKEWED COLUMNS:



Dropped all the skewed columns in the train dataset and checked the shape of the train data set and found that now there are 10682 rows and 24 columns in the dataset.

NOW WE WILL ANALYZE THE TEST DATASET:

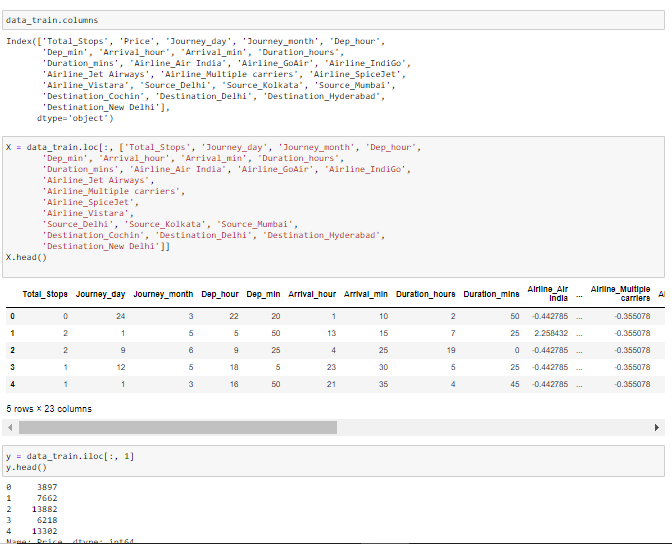
READING THE TEST DATASET:



As we all know that Test data contains lesser features and lesser data when compared to the train data, we can see that there is 2671 rows and 10 columns in the test data, The test data serves to give an unbiased estimation of your model learner's performance on unseen data, and more test data only gives you a more accurate estimate. You should test on as much data as possible. Normally this comes as a trade off when splitting between training and testing data (more test data means less training data, which is arguably more important), but in this case you are purposefully reducing the training set size to analyse the effect of that reduction. In this case, we are not combining the train and test data because it causes to data leakage. Data leakage is when information from outside the training dataset is used to create the model, this results in unreliable and bad prediction outcomes after model deployment.

We will perform the same feature engineering process with the test dataset as we have done with the train dataset while putting it into the model for prediction.

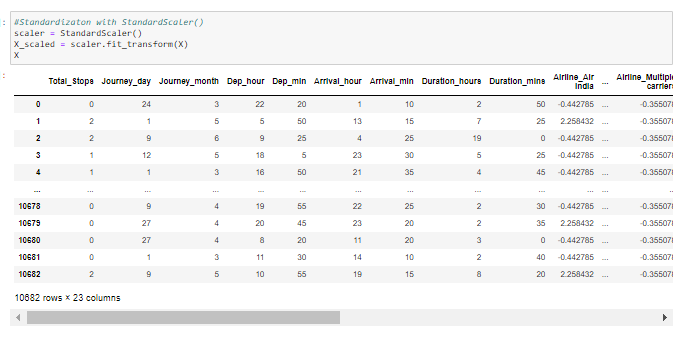
SEPARATING THE FEATURES AND TARGET VARIABLE:



STANDARDIZATION:

Data standardization is about making sure that data is internally consistent and it comes to the common scale.

We will use StandardScaler to standardize the data.



**HOLD- OUT -METHOD-** The **holdout method** is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The function approximator fits a function using the training set only.



**BUILDING MODELS:**

Before going deep into building models, you need to know about some model building features of Regression models.

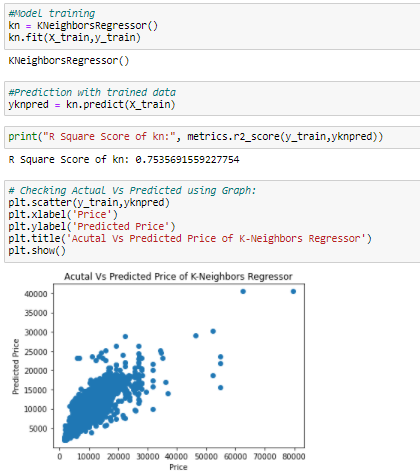
1)R2score- R2 score is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

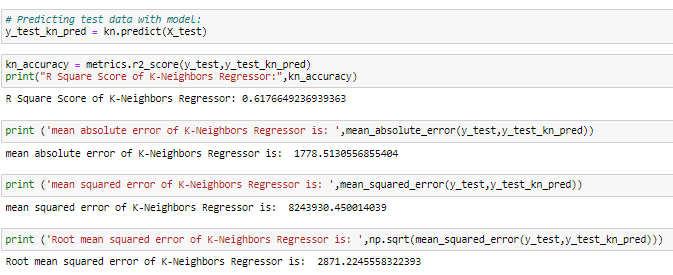
2)Mean Absolute error- In the context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group.

3)Mean squared Error - It is simply the average of the square of the difference between the original values and the predicted values.

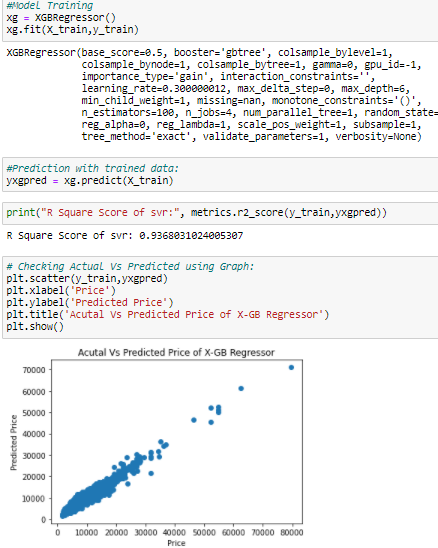
4) RMSE is calculated as the square root of the mean of the squared differences between actual outcomes and predictions. Squaring each error forces, the values to be positive, and the square root of the mean squared error returns the error metric back to the original units for comparison.

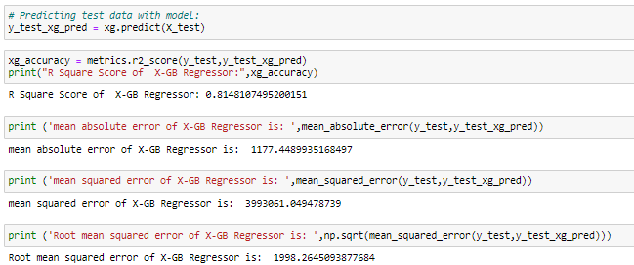
# KNeighborsRegressor:



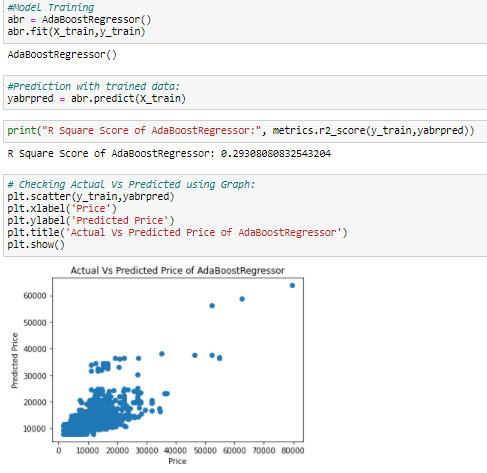


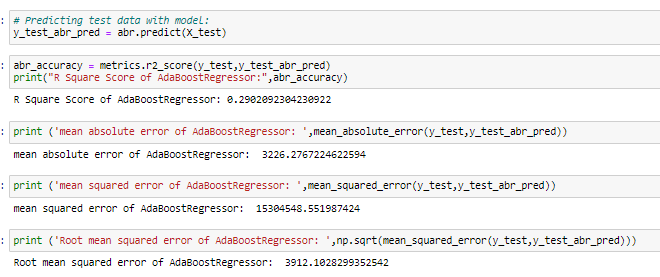
# XGB Regressor:



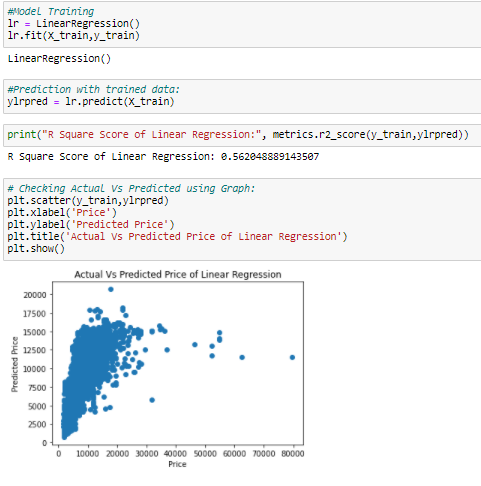


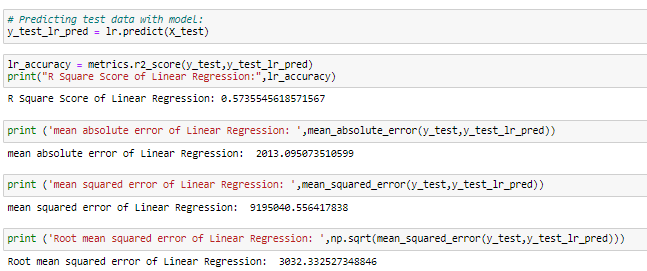
# 3.Adaboost Regressor:



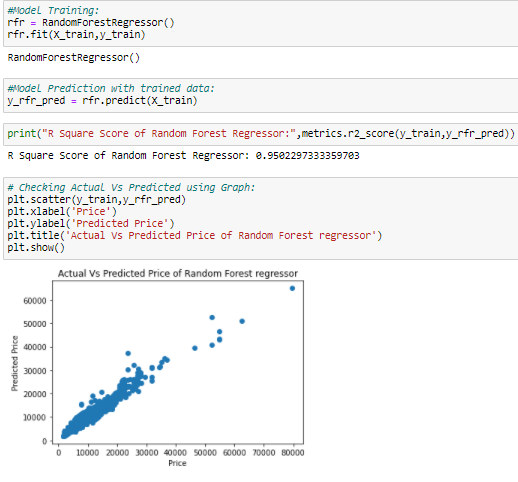


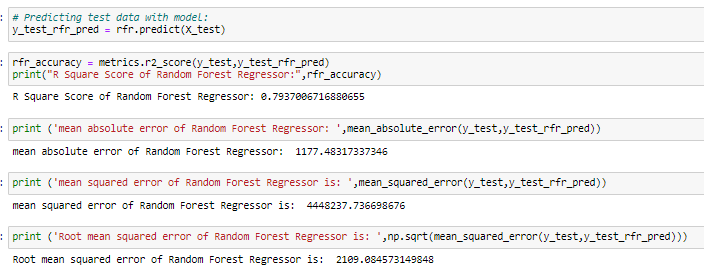
# Linear Regression:





# 5.Random Forrest Regressor:

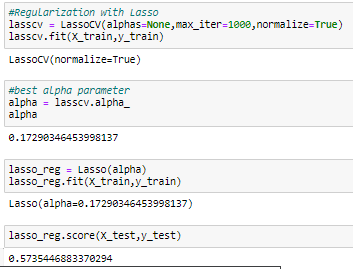




Regularization: It is highly important to restrict the features while modelling to minimize the risk of overfitting and this process is called **regularization**.

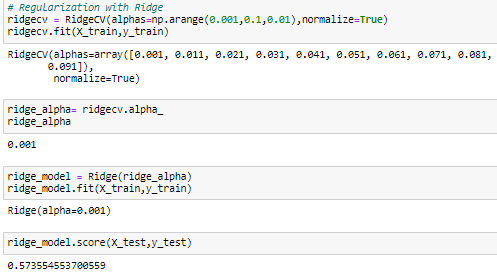
In **regression** we know that the features are estimated using **coefficients** and these estimates are the real game changes in modelling. If there is a possibility to ‘restrict’ or ‘shrink’ or ‘regularize’ the estimates towards zero, then the effect of the non- impactful features is reduced and it saves the model from high variance with a stable fit.

REGULARIZATION WITH LASSO:



Lasso regression is a regularization technique. It is **used over regression methods for a more accurate prediction**. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e., models with fewer parameters). Here we got the lasso regression score is only 57.3%.

REGULARIZATION WITH RIDGE:



Ridge regression is a model tuning method that is **used to analyse any data that suffers from multicollinearity**. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values. Here we got the score of 57.3% only.

CROSS VALIDATION SCORE:



Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

From the above metrics we can conclude that:

1. Kneighbors Regressor has the accuracy score of 61.7% and the cross-validation score of 56%.

2. XGB Regressor has the accuracy score of 81.4.% and the cross-validation score of 80.3%.

3. Adaboost Regressor has the accuracy score of 29% and the cross-validation score of 35%.4.

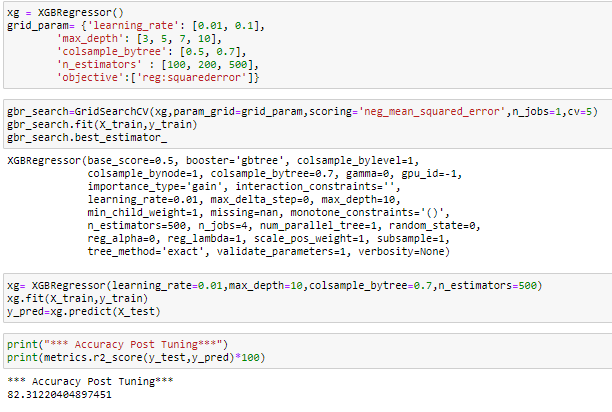
4. Linear Regression has the accuracy score of 57% and the cross-validation score of 56%.

7. Random Forrest Regressor has the accuracy score of 79.3% and the cross-validation score of 77.5%.

To select the final model, we don’t only have to look at the test accuracy score, we also need to check the Cross validation (CV) score. The least difference between the test and CV score indicates that the model is performing well without being underfit or overfit.

From the above performance metrics, we can conclude that XGB Regressor has performed well with test accuracy of 81.4% and CV score of 80.3%. Hence, we will select XGBRegressor as our final model since it has least difference between test and CV score.

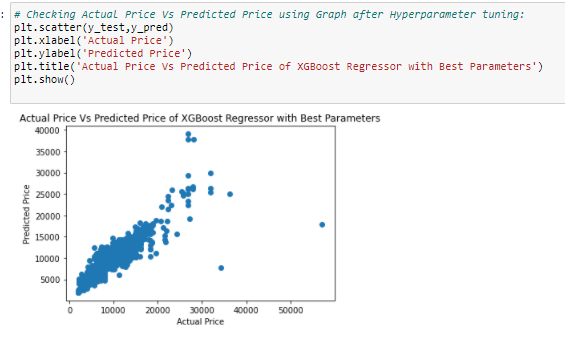
**HYPER PARAMETER TUNING:-**



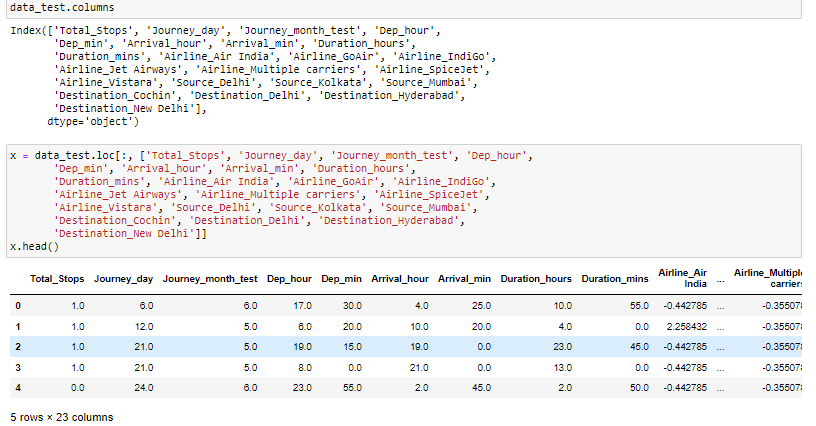
GREAT!!!!!

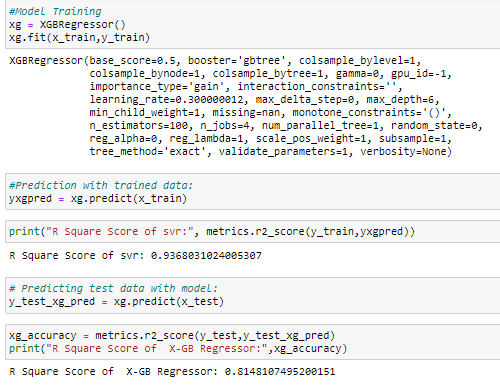
After Hyperparameter tuning of XGB Regressor we got the Accuracy score of 82.31%.

PLOTTING THE ACTUAL PRICE VS PREDICTED PRICE USING GRAPGH AFTER HYPER PARAMETER TUNING:



# Now putting the Test data into the model:

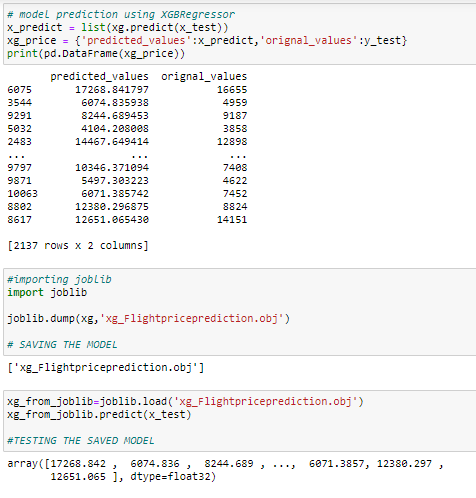




We got the Accuracy score of 81.4% with the test Data.

**Hence, we will select XGBRegressor as our final model since it has least difference between test and CV score.**

**SAVING AND TESTING THE MODEL:**

****

**Summary:**

We started with the data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data pre-processing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features. Afterwards we started training 5 different machine learning models, picked one of them (XGBRegressor) and tuned its performance through optimizing it’s hyperparameter values and finally selected the XGBRegressor Model.

# **Conclusion:**

In the proposed project the overall survey for the dynamic price changes in the flight tickets is presented. this gives the information about the highs and lows in the airfares according to the days, weekend and time of the day that is morning, evening and night. also, the machine learning models in the computational intelligence field that are evaluated before on different datasets are studied. their accuracy and performances are evaluated and compared in order to get better result. For the prediction of the ticket prices perfectly different prediction models are tested for the better prediction accuracy. As the pricing models of the company are developed in order to maximize the revenue management. So, to get result with maximum accuracy regression analysis is used. From the studies, the feature that influences the prices of the ticket are to be considered. In future the details about number of available seats can improve the performance of the model.

Results we can find after doing the analysis:

* The trend of flight prices varies over various months and across the holiday.
* The airfare varies depending on the time of departure, making timeslot used in analysis an important parameter.
* Airfare varies according to the day of the week of travel. It is higher for weekends and Monday and slightly lower for the other days.
* There are a few times when an offer is run by an airline because of which the prices drop suddenly. These are difficult to incorporate in our mathematical models, and hence lead to error.

**Concluding Remarks:**

Of course, there is still room for improvement, like doing a more extensive feature engineering, by comparing and plotting the features against each other and identifying and removing the noisy features. Also, more details about the flight price will certainly help. Like the cabin customer is flying and fare category, special fare has lot of discounts but we don’t enjoy free meal on board or maybe no flexibility to change or refund the booking. The additional info feature did have for some of the flights but we should have that for most of the flights.