**((((( FLIGHT PRICE PREDICTION )))))**

**<<PROBLEM STATEMENT>>**

A flight price prediction problem is to predict the price an airline will charge for a given flight from point A to point B. The problem is complicated by the fact that airlines can and do change prices at any time. Additionally, airlines often use different pricing methods, making it difficult to predict the final price.

There is a problem in the airline industry whereby customers cannot predict flight prices accurately. Airlines have been able to keep prices high by making small increases in price after a long period of no change. This has resulted in customers not being able to anticipate flight prices and has made it difficult for them to plan their travel.

Airlines have also been able to keep prices high by refusing to give discounts to customers who book their flights more than a few weeks in advance. This has resulted in customers having to pay full price for their tickets even when the airline knows that they will not be able to use the tickets.

Airlines could solve the price prediction problem by making small increases in price all the time, giving discounts to customers who book their flights more than a few weeks in advance, and allowing customers to change their tickets up to a few days before their flight.

**<<THE DATA SET>>**

We use the data set “ flight\_price\_prediction”. The data set has two set

One is for training and Second is for test purpose.

Size of training set: **10683** records , Size of test set: **2671** records , **11** features .

**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.

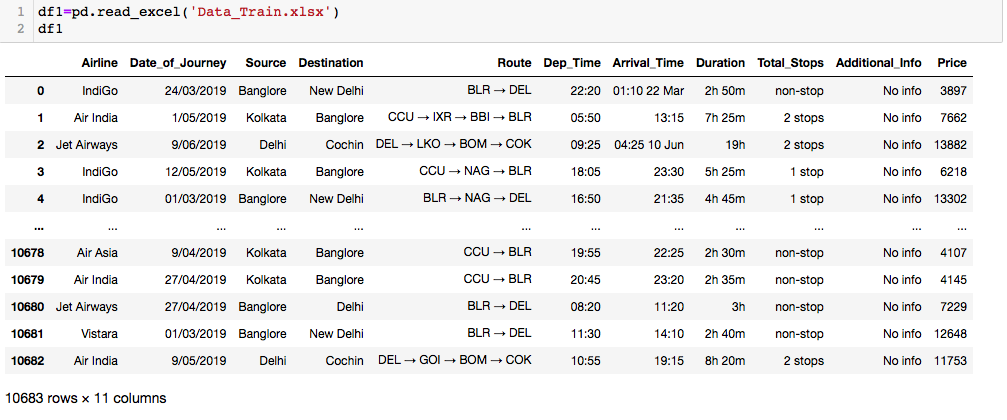
**Duration**: Total duration of the flight.

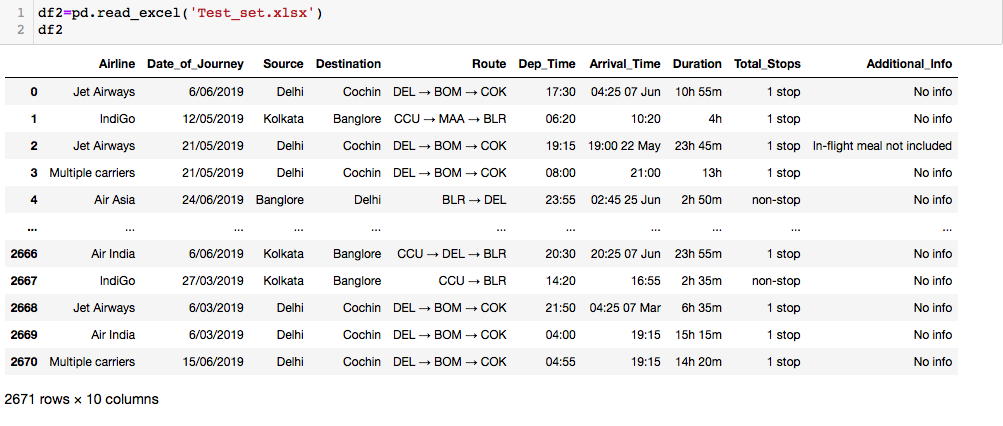
**Total\_Stops**: Total stops between the source and destination.

**Additional\_Info**: Additional information about the flight

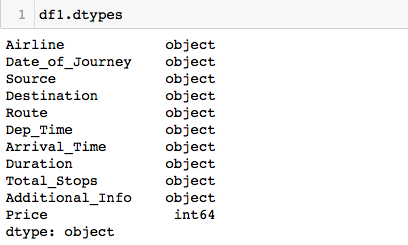
**Price**: The price of the ticket

**<<EDA OF DATA SET>>**

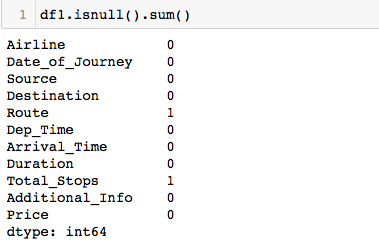
****



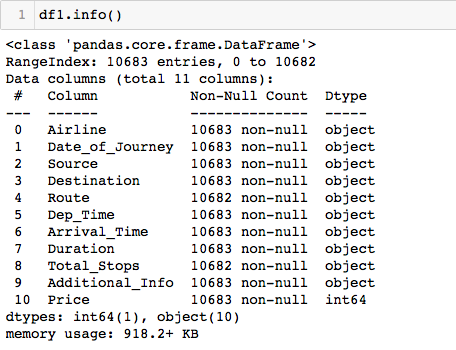
**Data Set has only two data types 10 features are of “Object” and one is “Int” type .**



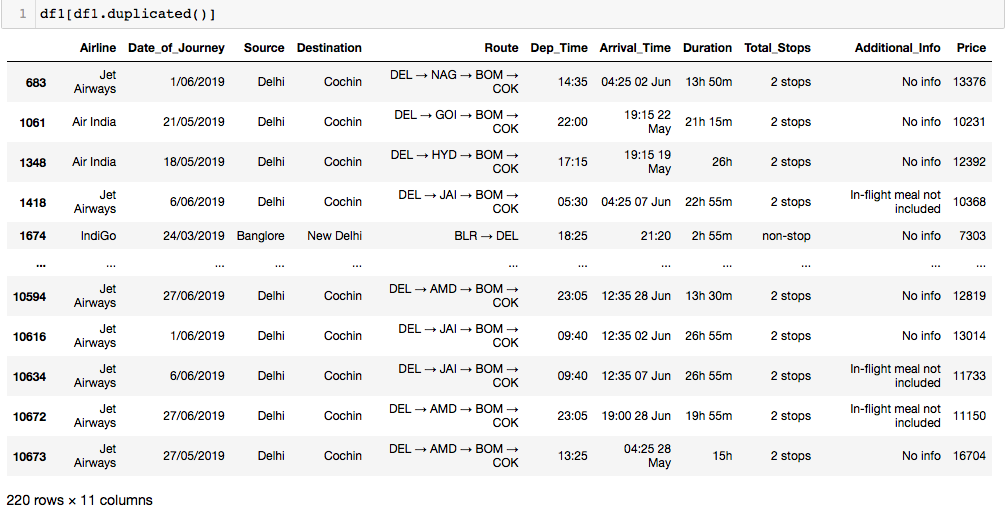
**Data Set has only Two Null values.**



**Some extra information is given below -**



**The data has some duplicate values also**

****

**<<PREPROCESSING PIPELINE>>**

Removing Duplicate Values .

Convert Duration Time into minute.

Split Journey Date into Day and Month Separately.

Convert Departure Time into hour and minute.

Convert Arrival Time into hour and minute.

Use Label Encoding to replace the Categorical values with a Numerical values.

Checking if Skewness is present in data.

Checking if Outliers present in data.

**Removing Duplicate Values :**

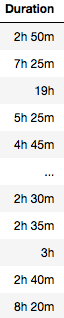
Screen Shot 2022-04-27 at 5.26.14 PM

Duplicate values can affect the accuracy of the model . The data have 220 duplicate values . So we remove that from main data set.

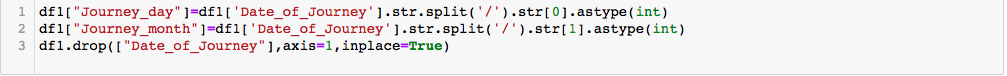
**Convert Duration Time into minute** **:**

Screen Shot 2022-04-28 at 4.20.44 PM

We convert the time column in to minute individually.

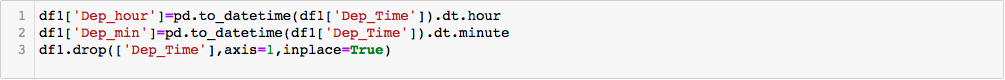
 

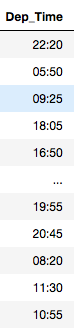
**Split Journey Date into Day and Month Separately** **:**



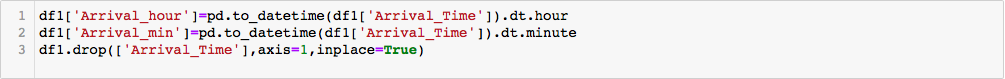
 

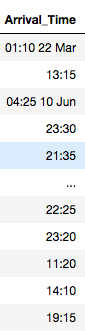
**Convert Departure Time into hour and minute :**



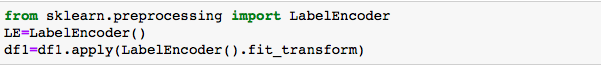
 

**Convert Arrival Time into hour and minute:**

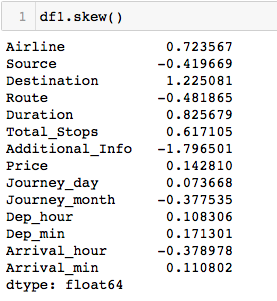


**Use Label Encoding to replace the Categorical values with a Numerical values :**

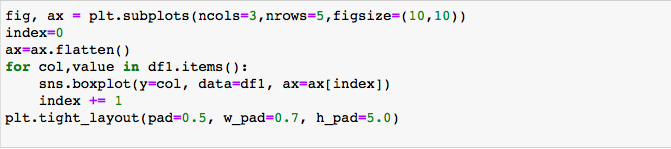


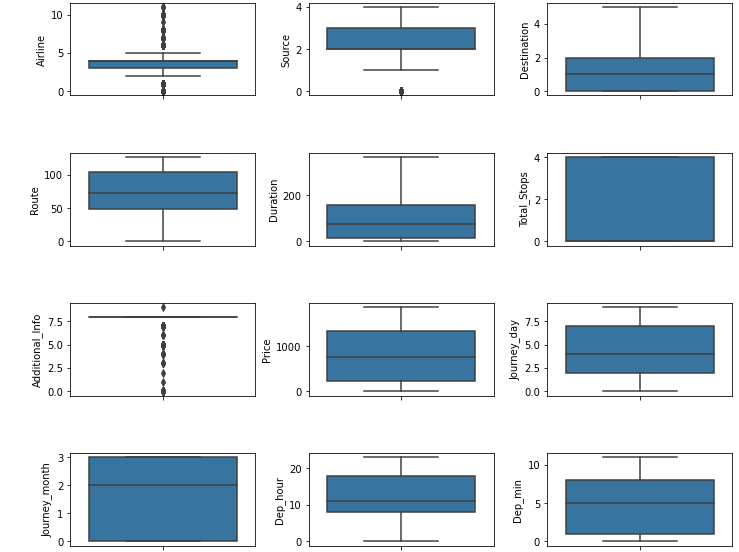
**Checking if Skewness is present in data :**



As we can see that there are minor skewness present in range of -1 to 1 in data but it’s not so much that affect the accuracy. So we can ignore it .

**Checking if Outliers present in data** **:**





Some of the variables have present outliers in it but that are not very much related with the target variable , so there is no need to treat that.

**Do Some Scalling if Needed :**

In the machine learning algorithms if the values of the features are closer to each other there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower.

So if the data in any conditions has data points far from each other, scaling is a technique to make them closer to each other or in simpler words, we can say that the scaling is used for making data points generalized so that the distance between them will be lower.

As we know, most of the machine learning models learn from the data by the time the learning model maps the data points from input to output. And the distribution of the data points can be different for every feature of the data. Larger differences between the data points of input variables increase the uncertainty in the results of the model.

The machine learning models provide weights to the input variables according to their data points and inferences for output. In that case, if the difference between the data points is so high, the model will need to provide the larger weight to the points and in final results, the model with a large weight value is often unstable. This means the model can produce poor results or can perform poorly during learning.

**BUILDING MACHINE LEARNING MODEL :**

**In this data set our target is to predict the prices of flight tickets for the next upcoming days on the basis of previous days data . Our target variable is a continuous variable i.e. it is regression based problem.**

**Building a model needs some more additional steps :-**

**Splitting the data into two sets ( train & test ).**

**Find the best suitable parameters for better accuracy of the model.**

**Then use different related models suitable for problem.**

**Splitting the data into two sets ( train & test ) :**

For Machine learning model we have to split the data into two parts . First one is for Training purpose and Second one is for Testing purpose. The standard ratio is 80:20 i.e. 80% is for training and 20% is for testing. The main purpose behind data splitting is to make model used to data for future testing.

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.

Screen Shot 2022-04-29 at 11.12.49 AM

Screen Shot 2022-04-29 at 11.13.27 AM

**Find the best suitable parameters for better accuracy of the model :**

Machine learning involves predicting and classifying data and to do so, you employ various machine learning models according to the dataset. Machine learning models are parameterized so that their behavior can be tuned for a given problem. These models can have many parameters and finding the best combination of parameters can be treated as a search problem.

A model parameter is a configuration variable that is internal to the model and whose value can be estimated from the given data.

* They are required by the model when making predictions.
* Their values define the skill of the model on your problem.
* They are estimated or learned from data.
* They are often not set manually by the practitioner.
* They are often saved as part of the learned model.

A model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data.

* They are often used in processes to help estimate model parameters.
* They are often specified by the practitioner.
* They can often be set using heuristics.
* They are often tuned for a given predictive modeling problem.

You cannot know the best value for a model hyperparameter on a given problem. You may use rules of thumb, copy values used on other issues, or search for the best value by trial and error. When a machine learning algorithm is tuned for a specific problem then essentially you are tuning the hyperparameters of the model to discover the parameters of the model that result in the most skillful predictions.

Hyperparameter tuning is a final step in the process of applied machine learning before presenting results.Although there are many hyperparameter optimization/tuning algorithms.

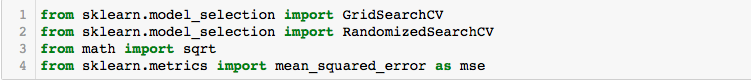
two simple strategies:

1. grid search

2. Random Search.

Grid search is an approach to hyperparameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid.

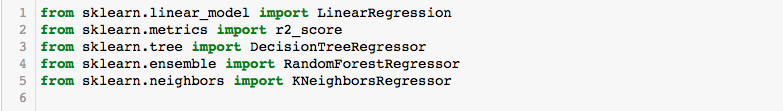
Random search differs from a grid search. In that you longer provide a discrete set of values to explore for each hyperparameter rather, provide a statistical distribution for each hyperparameter from which values may be randomly sampled.



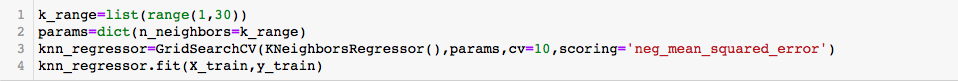
**Select the Best Model :**

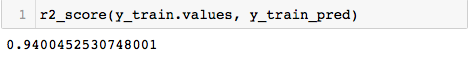
As we know our target is to predict the price of flight ticket . This problem is regression based

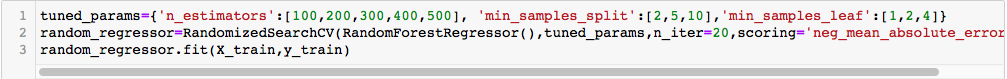
So we use some regression based algorithms .

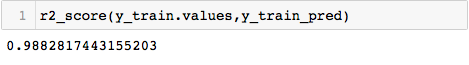


We use Linear regression, knn regressor, Decision tree regressor, random forest regressor.











Screen Shot 2022-04-30 at 4.19.51 PM

**We got 97% accuracy and 89% CrossValidation Score with Decision Tree Regressor and 98% accuracy with Random Forest Regressor. So we save Decision Tree as our model .**

**Screen Shot 2022-04-30 at 4.45.08 PM**

--------------------------------(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)---------------------------------