**Flight Price Prediction using Machine Learning**

##### **Problem Statement:**

The variation in prices of flight tickets has always been very confusing for customers and is very difficult to guess. The airline companies implement dynamic strategies to assign pricing for airfare tickets to increase demand for their seats and maximize their revenue, hence it becomes difficult for consumers to buy tickets at a minimum price. The closely connected terms with the airline prices such as commercial, financial, social factors and marketing is under consideration while practicing these dynamic strategies. The airline companies are trying their best to keep their revenue high and increase their profit. The travellers often find the flight prices unpredictable as the flight prices tomorrow will not be the same as the flight prices of today. The system is complicated because each flight has a limited number of seats to be sold. In case the demand for air tickets is high, then the prices will increase and on the other hand if the seats are left unsold then the cost of air tickets might decrease as it represents a loss of revenue. To solve this problem of predicting flight prices, Machine Learning is a great idea to learn from historical data of the past flight prices and build logic on the given data. We will use Linear Regression which will help us in predicting the flight prices on the basis of certain factors which will involve data extracting, data analyzing and data interpretation.

**Dataset Information**

The prices of flight tickets for various airlines are between the months of March and June of 2019 and between various cities.

We have two datasets i.e. Train Data and Test Data.

The size of the **Training Set** is 10,683 records which consists of both categorical and numeric data. Some special characters are also seen within the data to which we will apply data transformation before using it on the Model.

The **features** considered initially for each flight are:

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

The size of the **Testing Set** is 2671 records. The testing data is similar to the training data, except for the “Price” column which will be predicted using the model.

**Data Analysis**

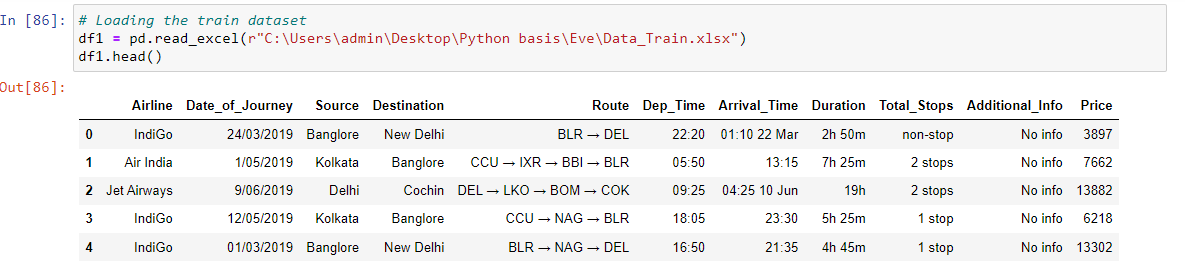
In this project, we are provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities. Also we are provided with two datasets- Train data and Test data. At first we import all the required libraries that we will need and then load the datasets into the notebook.

Source:: <https://github.com/mygithub9319/Datatrained-Project-Evaluation-Phase>

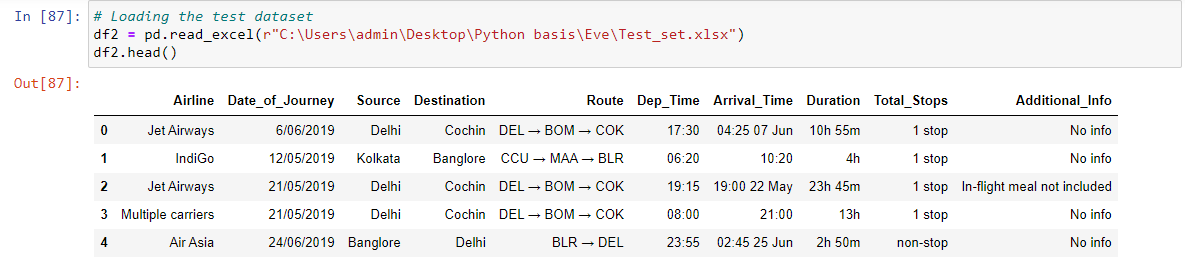
**Importing the libraries:**



**Loading the dataset df1**



**Loading the dataset df2**



Both the datasets are similar, with the Test data not having the ‘price’ column. Using the Train dataset we have to train and validate our model, and using that model we have to predict the price in the test dataset. The features that are present in the datasets are:

**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.

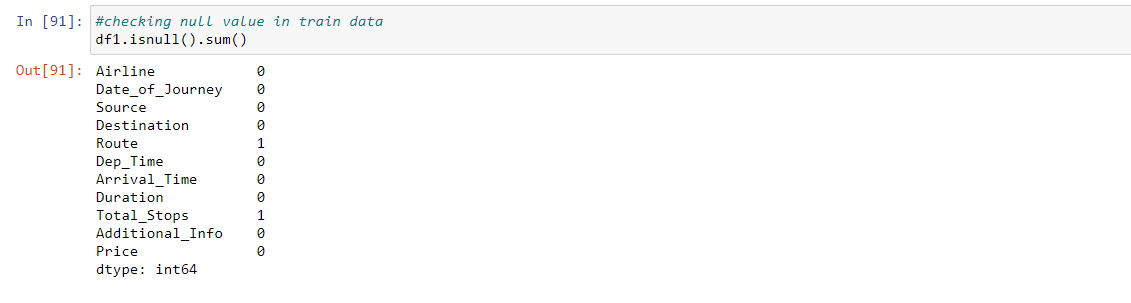
**Additional\_Info**: Additional information about the flight

**Price**: The price of the ticket

The price column is the target column here.

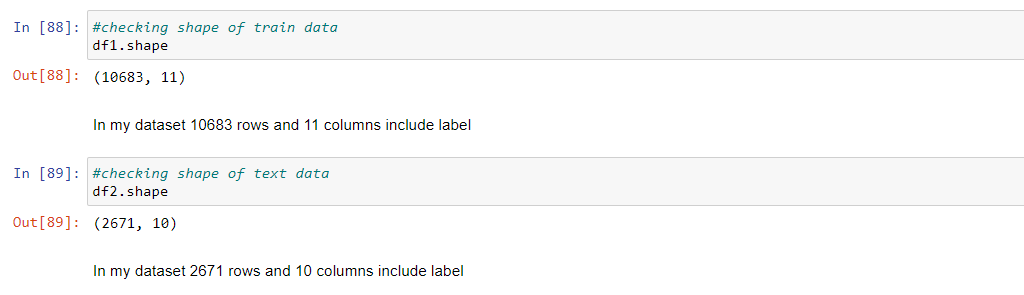
**EDA Concluding Remarks:**

First of all we check the shape of the dataset to get an idea about the size of the data. Then we check for any missing values in the dataset, so that we can treat them.



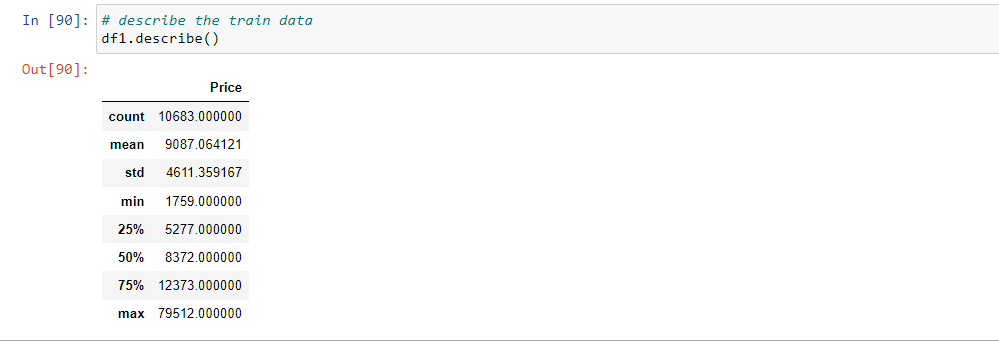
We have found the shape of the train data to be of 11 columns and 10683 rows. Also we have found two missing values in two columns each.

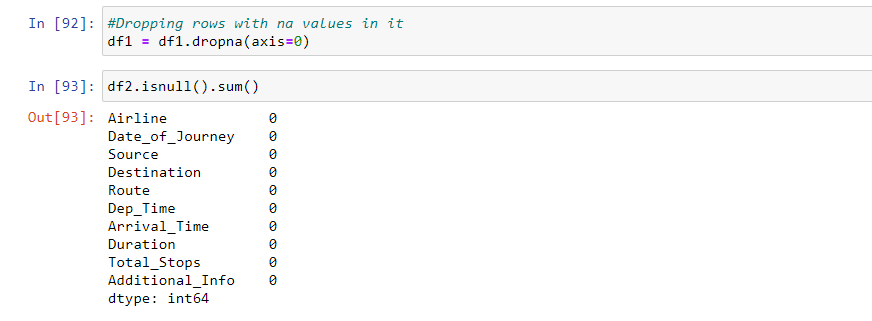
For checking shape of both df1 and df2



In our df1 10683 rows and 11 columns include target column.and df2 2671 rows and 10 columns include target column

Describe the train data(df1)



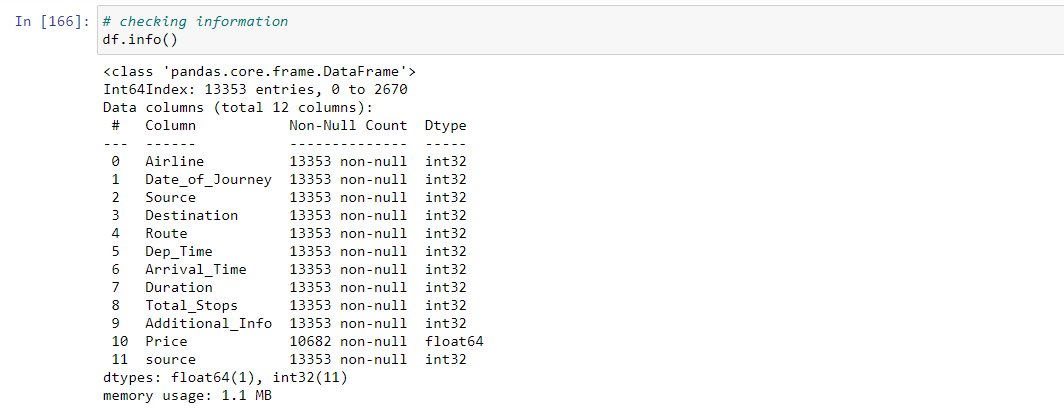


There is no any missing value in our df2 data.

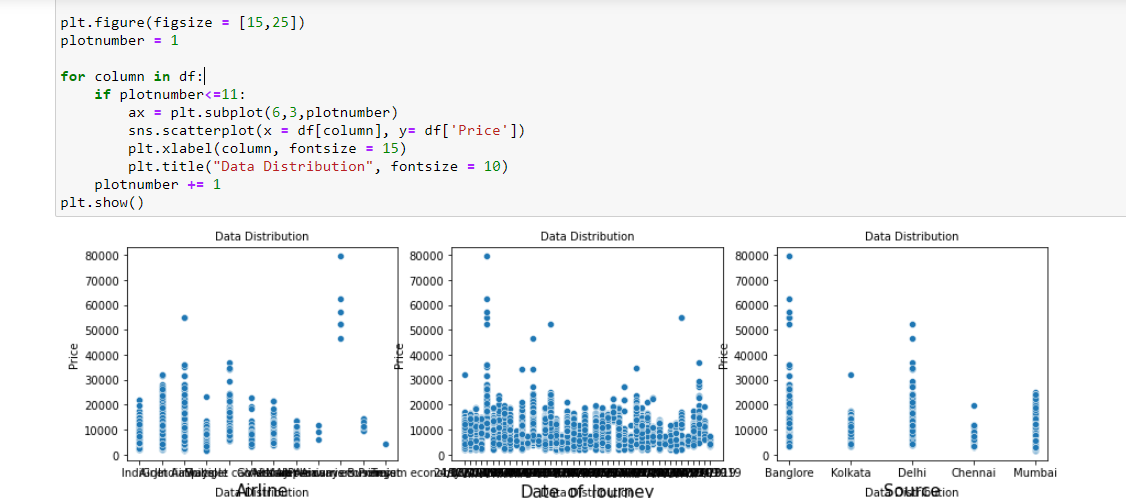
**Now Adding both train and test data(df1+df2)**

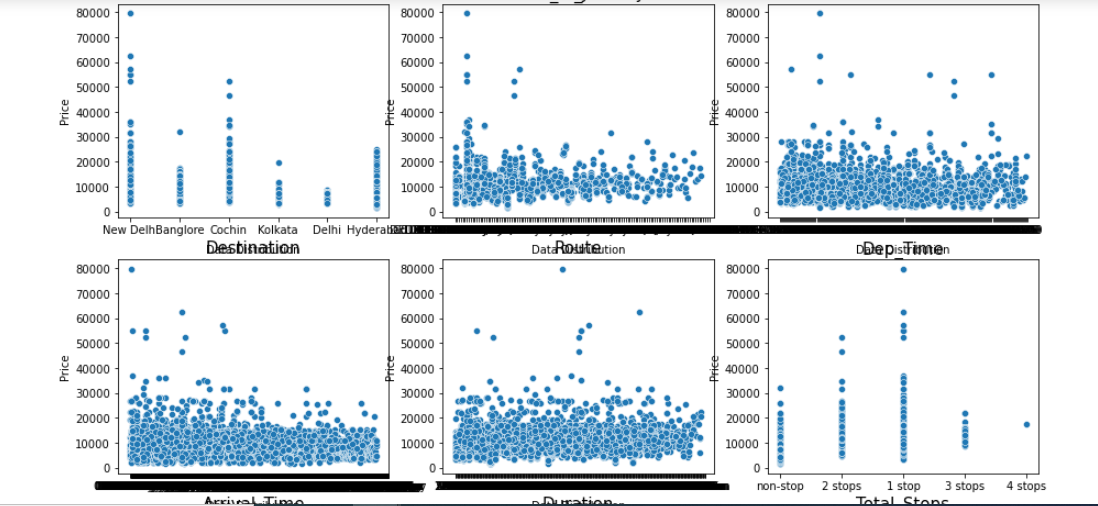


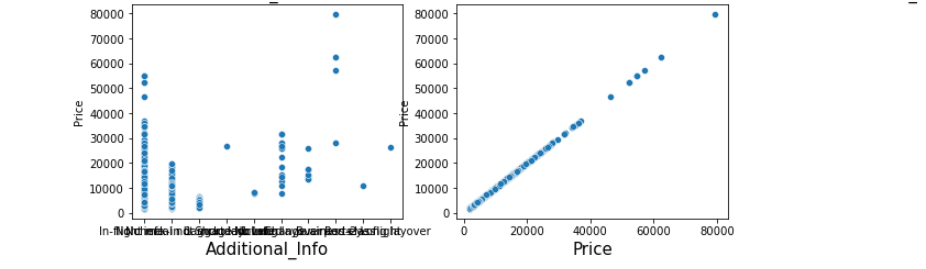
Then we check the info of the data and the unique values present in it.



**Now using scatter plot to see the relation of features with label**

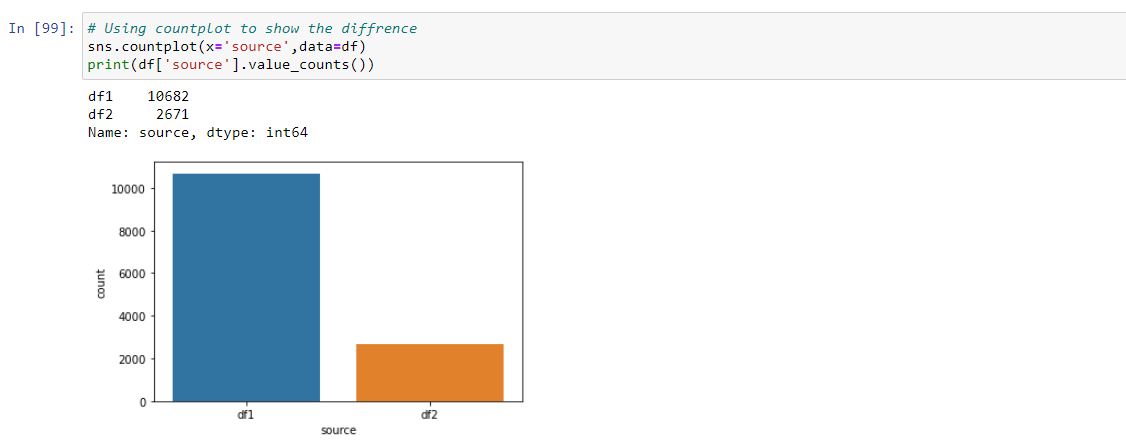






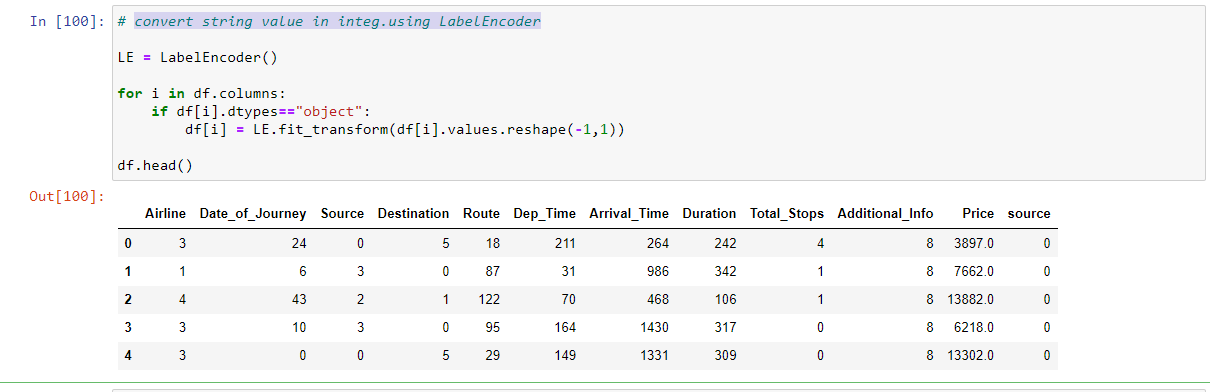
In the data set there is positive and negative both relationship with features



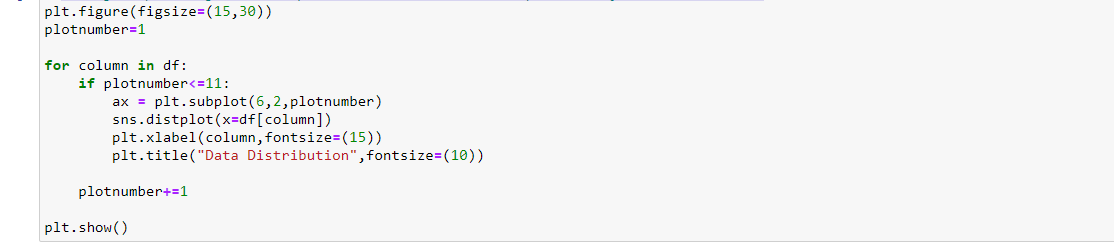


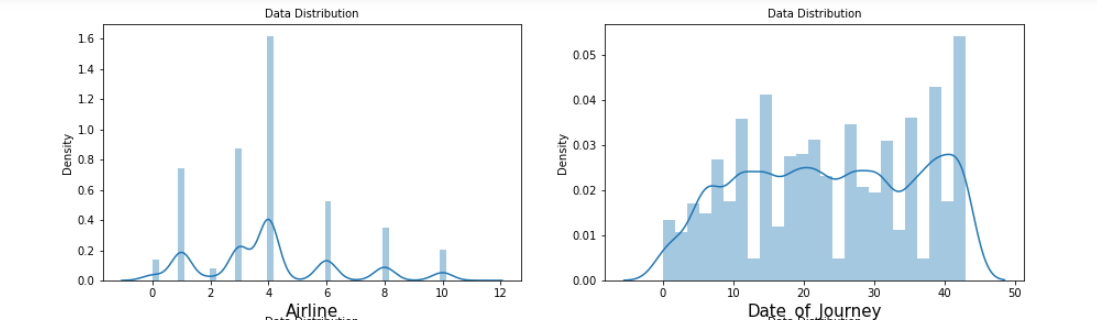
Train and test data are imbalanced (df1 and df2)

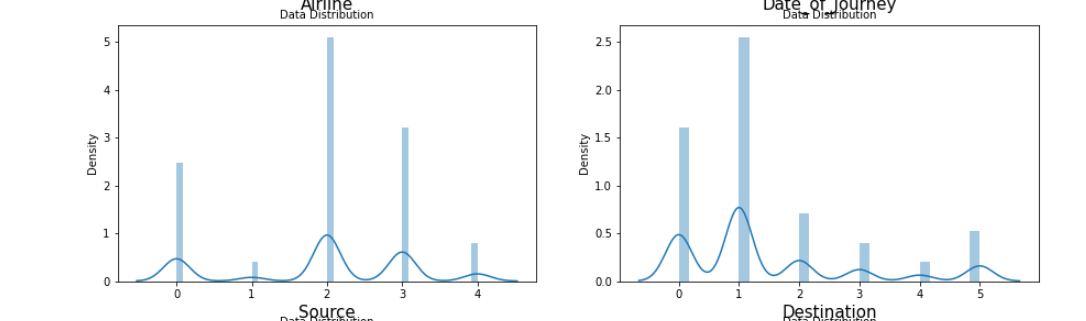
**Convert string value in integer. Using LabelEncoder**

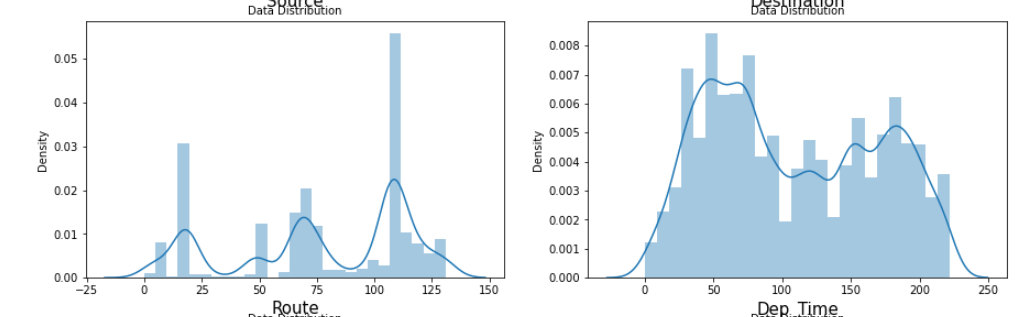


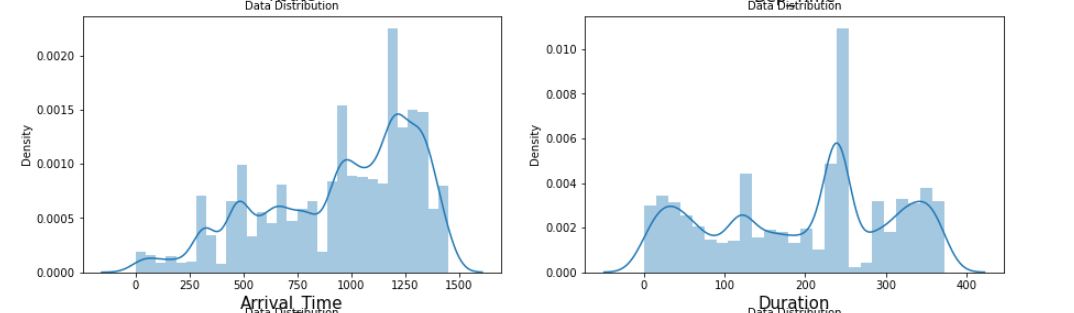
**Now again plotting the dist plot to see the relationship between features and label**

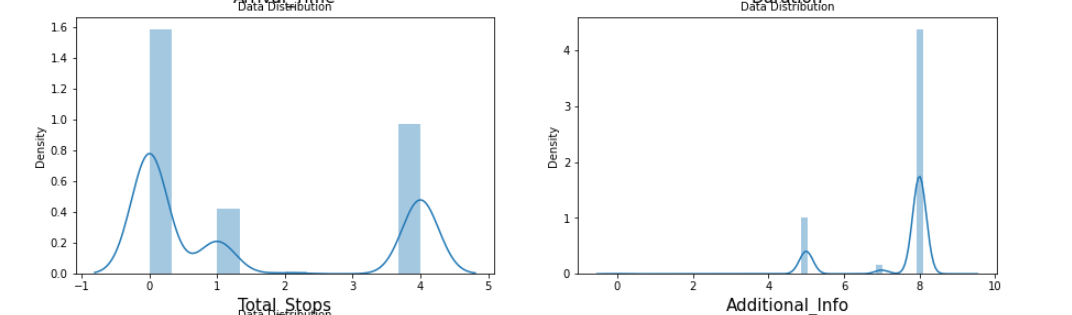


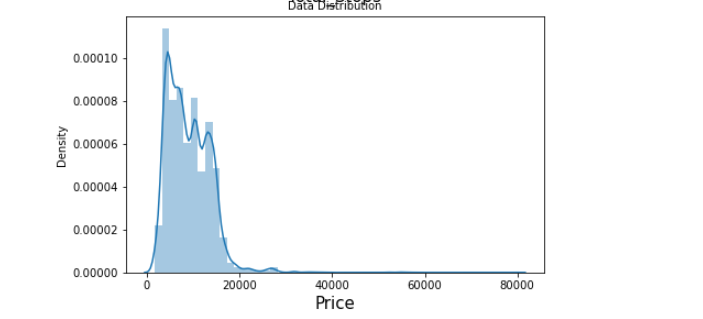






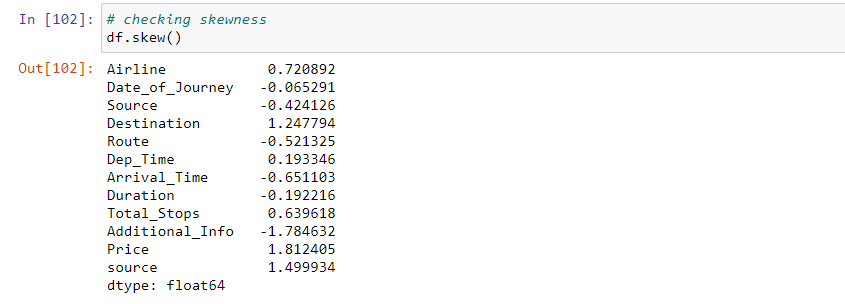






**Only Price showing some normal distribution. Rest not any showing normal distribution**

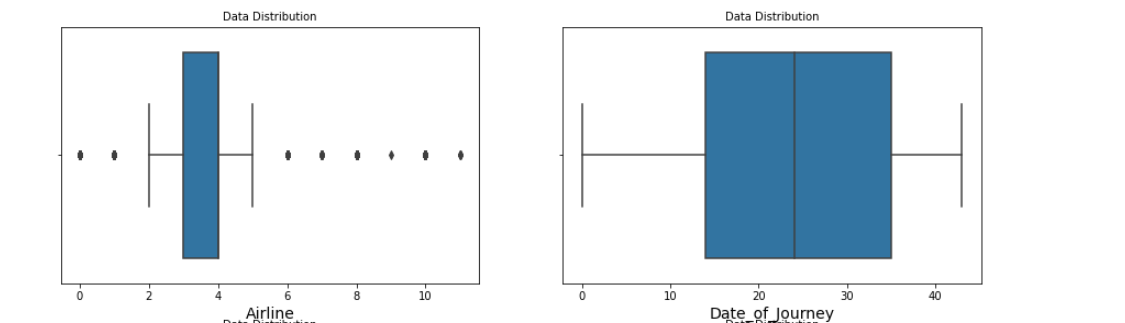
**Checking skewness**

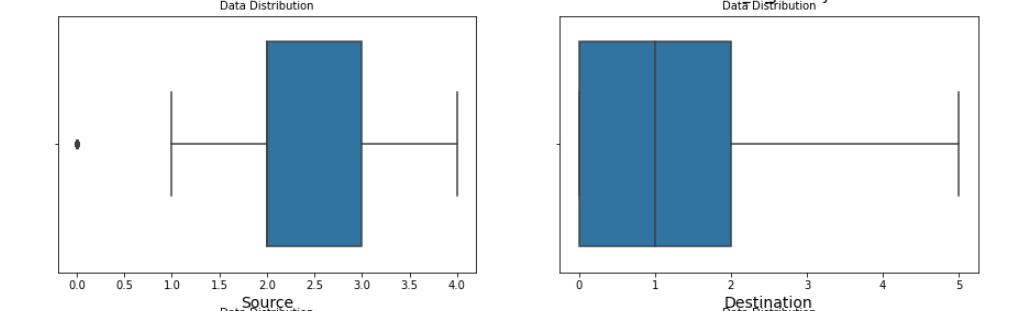


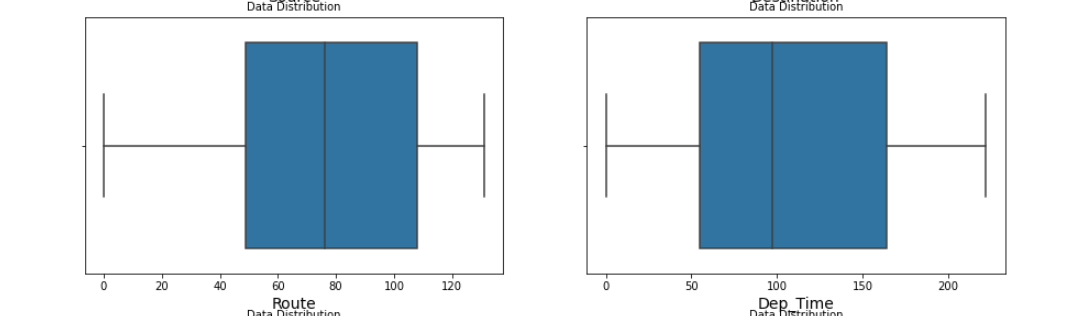
Some skewness present in our dataset

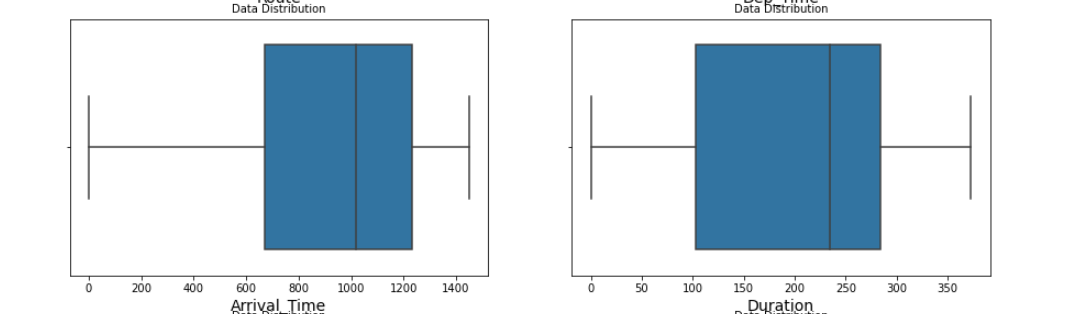
**Checking outlier in our dataset**

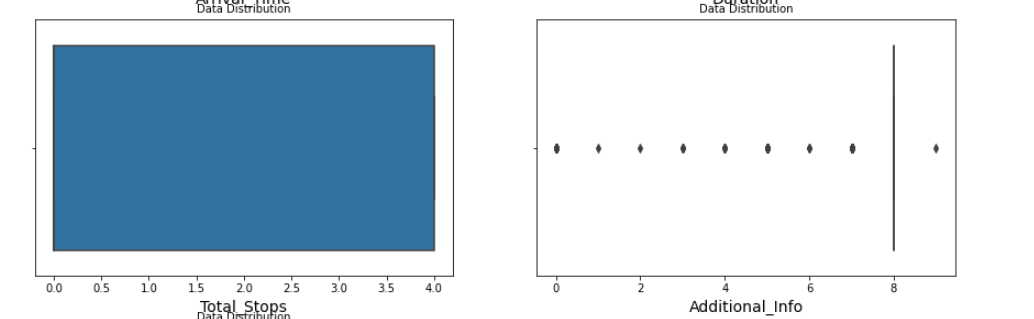


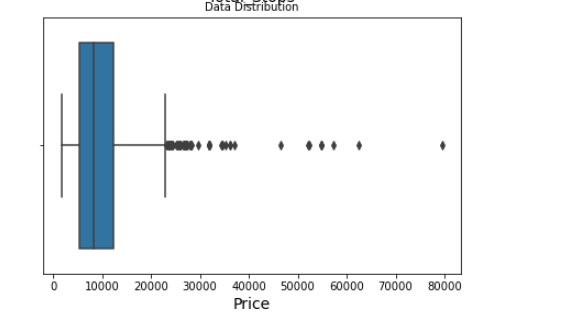






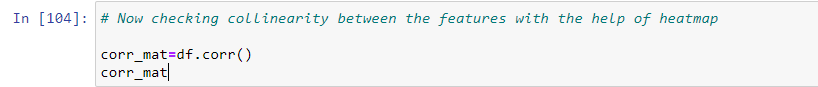


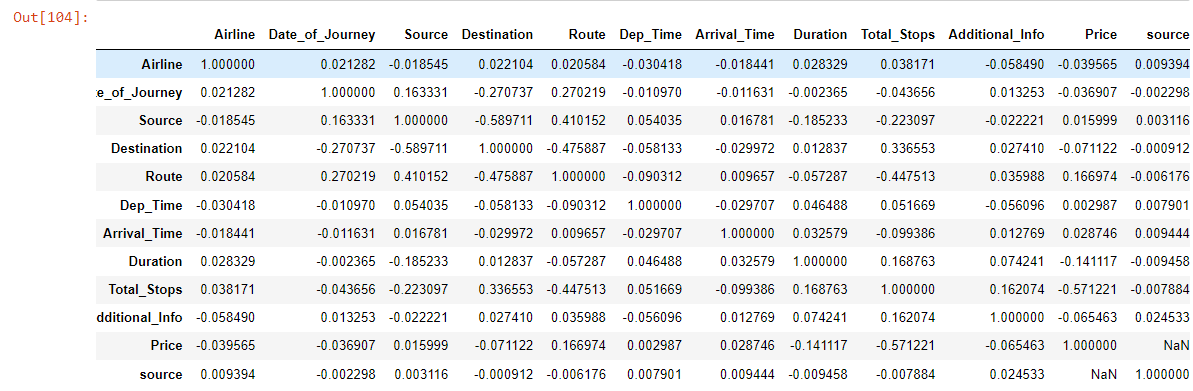


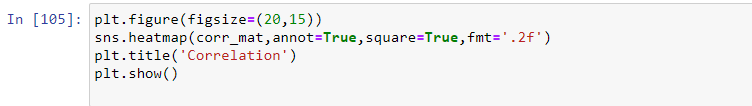


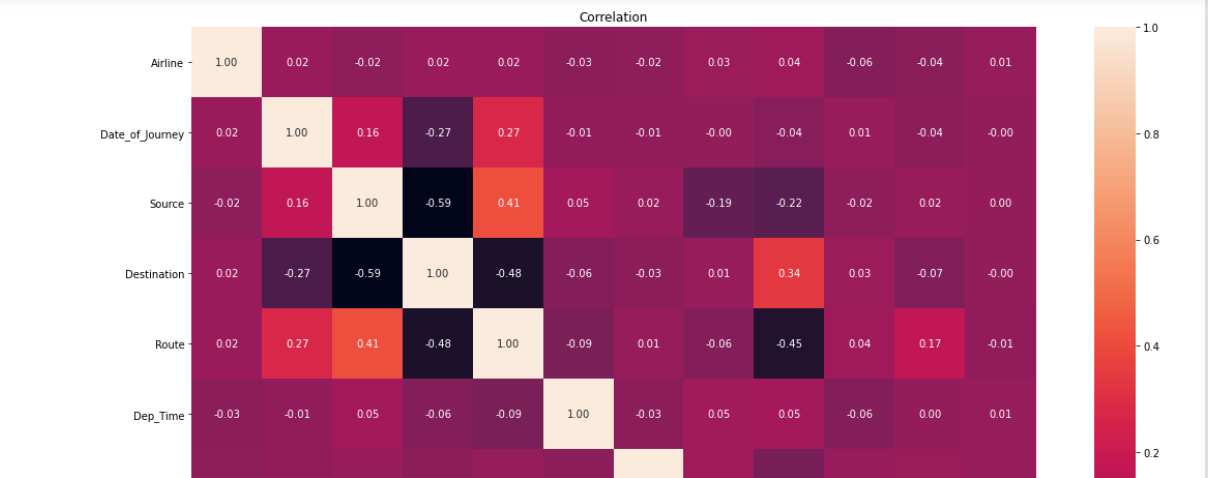
Skewness present in our dataset

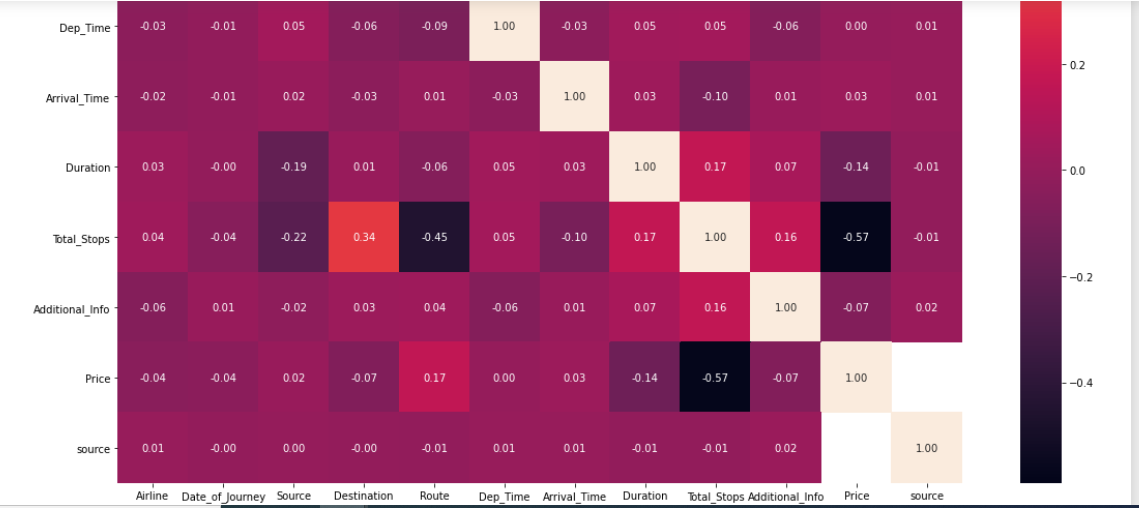
**Now checking collinearity between the features with the help of heatmap**



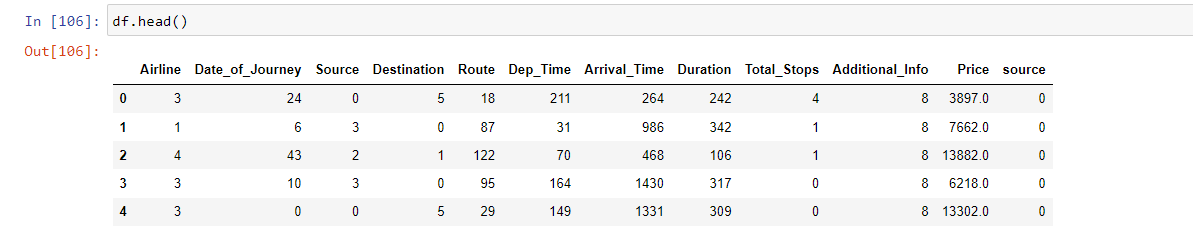


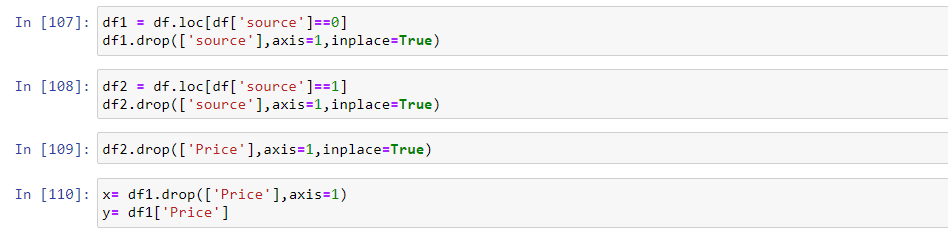




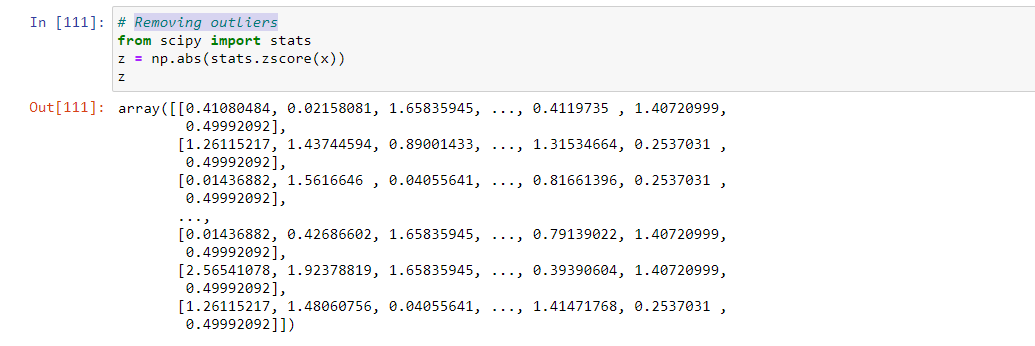


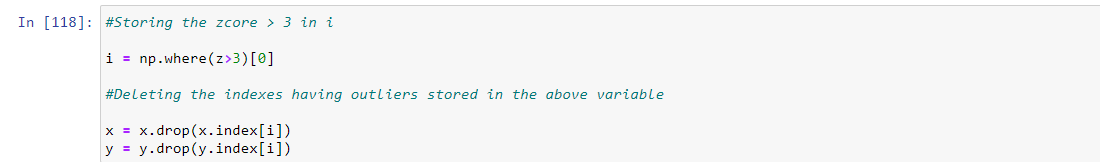
There are very less relationship between source vs other column.and total\_stops vs price is very good relationship



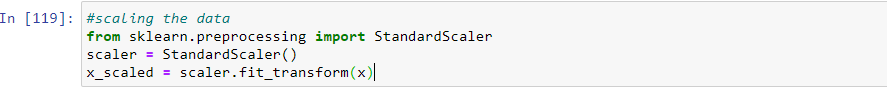


**Removing outliers**



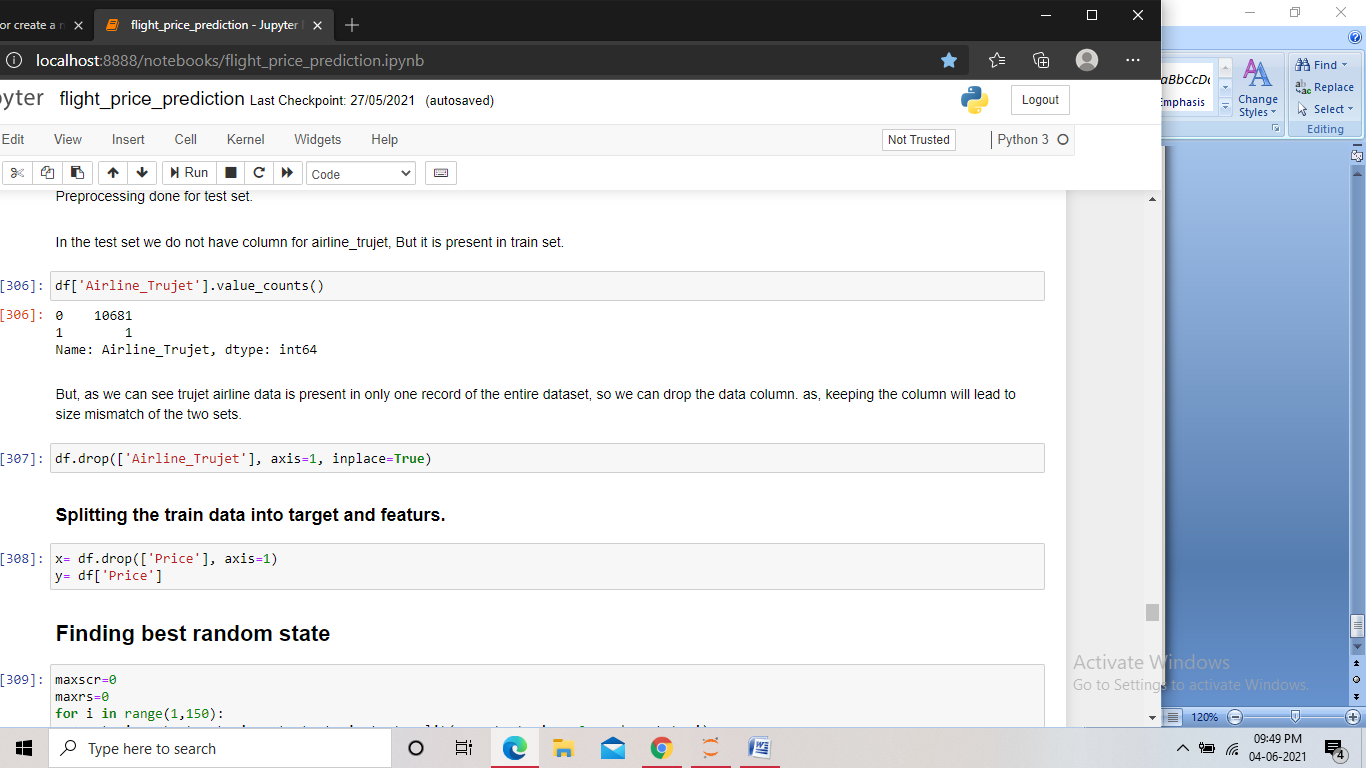


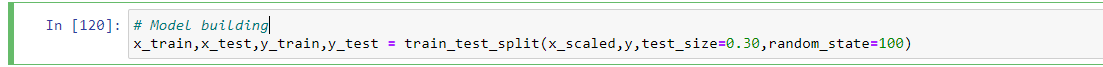
**Scaling the data**



**Building Machine Learning models:**

At first we split the train data into target and features.

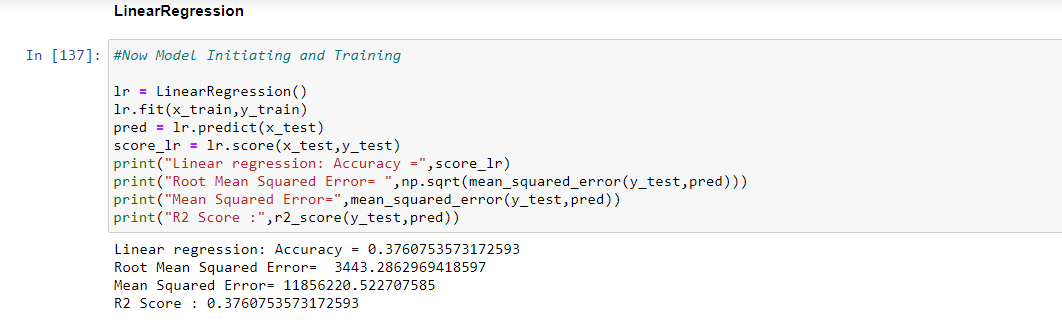




Building a model that will help to measure the performance of a better and more refined algorithm is the major goal here. We have used different Regression and Ensemble Techniques to compare and check which algorithm gives better performance and stack them all at the end to see how the model is giving predictions.

**Linear Regression**

Linear Regression is a Supervised Machine Learning algorithm that performs Regression tasks. Simple Linear Regression analysis is used to identify the correlation between two continuous variables. Prediction error is minimum when we find the best fit line for the given data using Linear Regression Algorithm



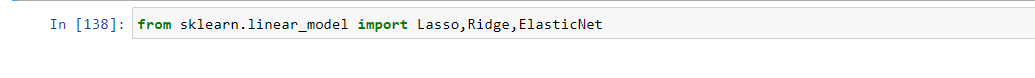
The error rate of the model after using Linear Regression was found to be-

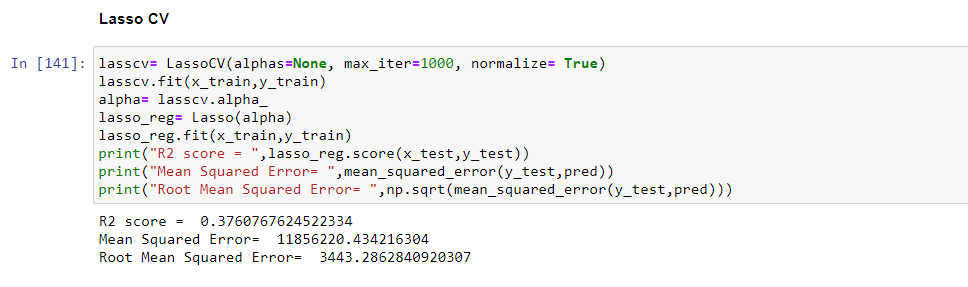
Accuracy=0.376

Root Mean Squared = 3443.286

Mean Squared Error: 11856220.522

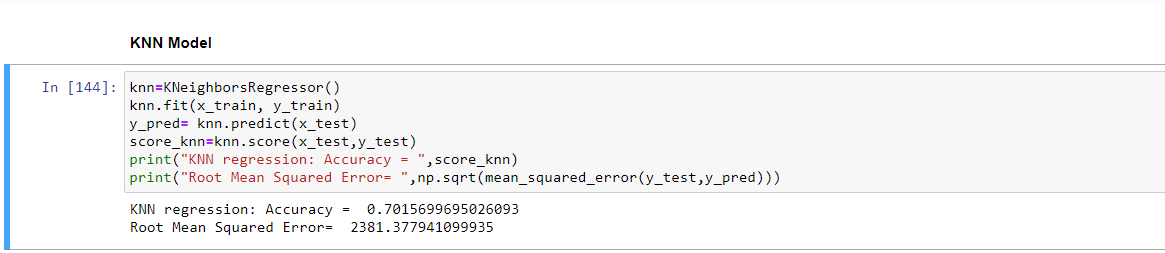
R2 Score: 0.37

**Regularation Process**



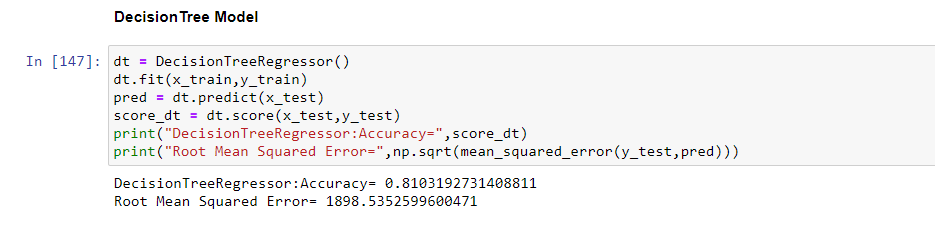
Using Lassocv we found

R2 Score=0.37, Mean Squared Error=11856220 and Root Mean Squared Error=3443.286



Using KNN we found

Accuracy=0.70 and Root Mean Squared Error = 2380.377



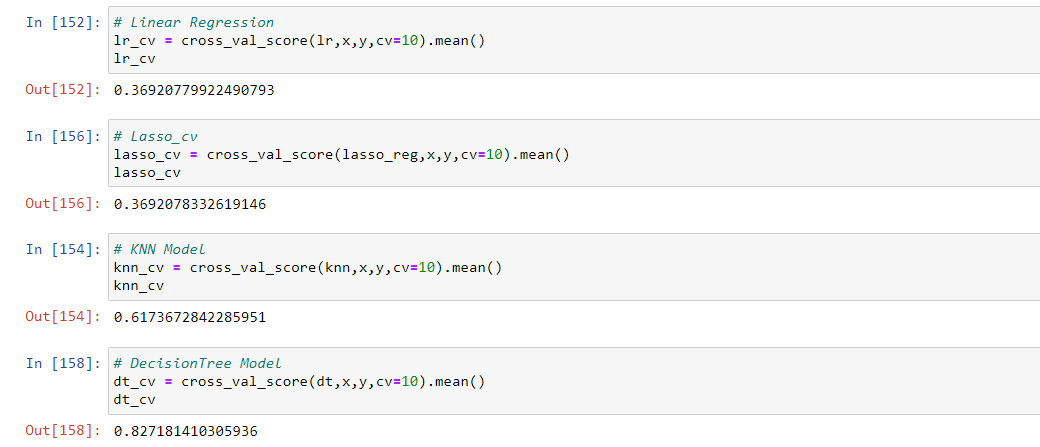
Using DecisionTree we found

Accuracy = 0.810 and Root Mean Squared Error=1898.535

**Model Evaluation**

A **cross-validation** technique was applied to all the samples and the mean performance of the model is produced.

The **best fit line** of the model was seen covering almost all data points which is an illustration of getting the best accuracy and indicates that the model has studied all the points in the data and there are no chances of having overfitting and underfitting issues.



DecisionTree is best model because higher cv score and comparatively less rmse

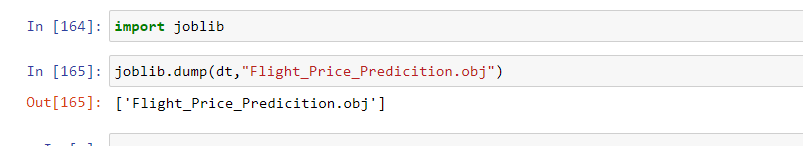
Again we found the CV scores of the Random forest and DecisionTree to be higher as compared to the other models. They also showed very less difference between the CV score and the r2 score. Now we performed hyper parameter tuning on both these models to find the best performing model among the two.



We have found Accuracy 0.815, Mean Squared Error=5670960 and Root Mean Squared Error=2381.377

**Saving the model**

We then loaded the saved model into a variable. And using that model we predicted the price for the test dataset. Which is then stored in a dataframe for submission.



**Concluding Remarks:**

In this project we found the feature engineering play a crucial role in the performance of the model. We treated columns containing date and time data and categorical data.

While performing Bi-variate analysis we found the Duration of the flight being very positively correlated with the price, we also found jet airways business flight being the most expensive among all, with spice jet being the least expensive. Also we found that as the number of stops increases the price of the flights also increases.

With this project we also got an idea about preprocessing the train data and test data separately and different encoding techniques to be used for different types of data. We Also saw how we can check different model’s performances, and to select and finalize the best performing model.