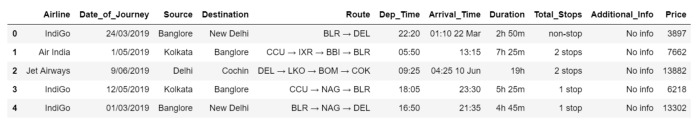
# Flight Price Prediction

Problem Statement

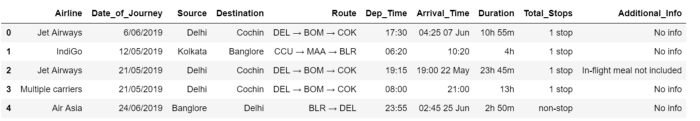
Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travelers saying that flight ticket prices are so unpredictable. As data scientists, we are going to prove that given the right data anything can be predicted. 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.

Datasets

We are going to use two Datasets: - Train Data and Test Data



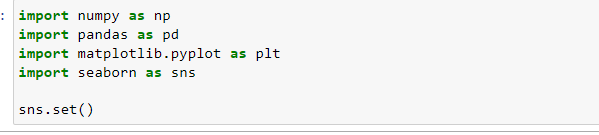
Training data is combination of both categorical and numerical also we can see some special character also being used because of which we have to do data Transformation on it before applying it to our model.

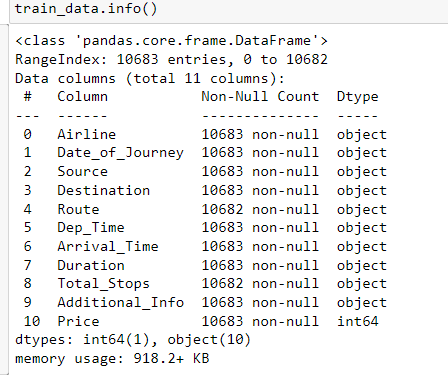


The test data is similar to the training data set, minus the ‘Price’ column (To be predicted using the model).

## **Python Coding**

**Step 1: Import the relevant libraries in Python**





**Step 2: Import Train Data**



We will import the data one by one, so just importing Train Dataset only.

**Step 3: Feature Generation**

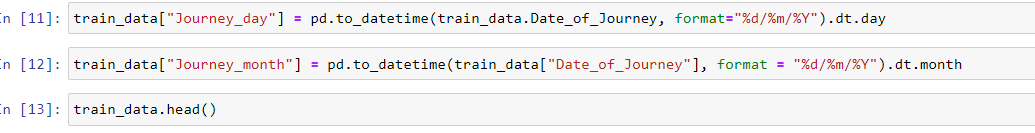
In this step we mainly work on the data set and do some transformation like creating different bins of particular columns, clean the messy data so that it can be used in our ML model. This step is very important because for a high prediction score you need to continuously make changes in it.

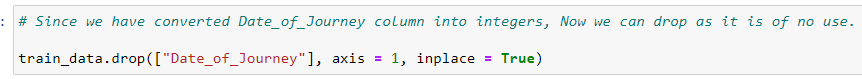
**Date\_of\_Journey: -**

From description we can see that Date\_of\_Journey is an 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 dtype.

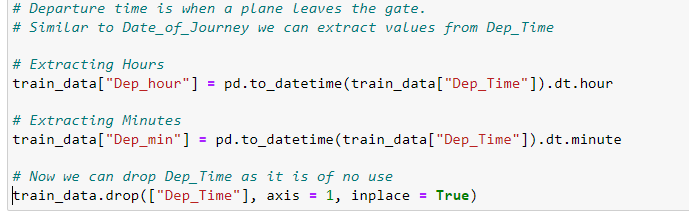
\*\*\*.dt.day\*\*\*\* method **will extract only day of that date\ .dt.month method will extract only month of that date**

Since we have converted the Date\_of\_Journey column into integers, we can drop as it is of no use,



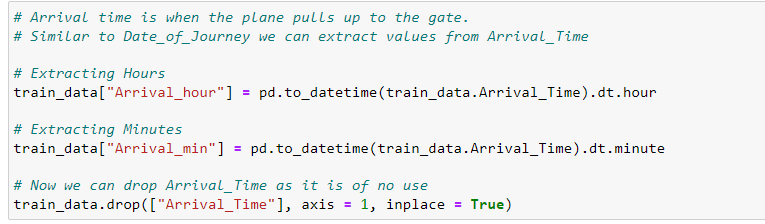
**Dept\_time:-**

Departure time is when a plane leaves the gate. Similar to Date\_of\_Journey we can extract values from Dep\_Time and then drop Dep\_time as it is of no use.



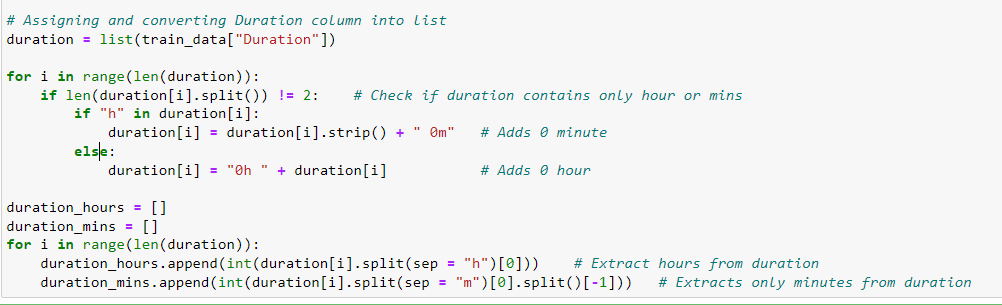
**Arrival\_time:-**

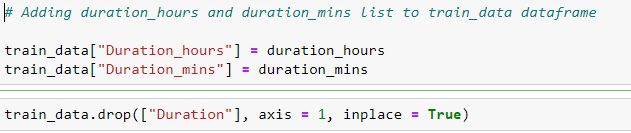
Arrival time is when a plane pulls up to the gate. Similar to Date\_of\_Journey we can extract values from Arrival\_time and then drop Arrival\_time as it is of no use.



**Duration: -**

Time taken by plane to reach destination is called Duration. It is the difference between Departure Time and Arrival time.





**Step 4: Prepare categorical variables for model using label encoder**

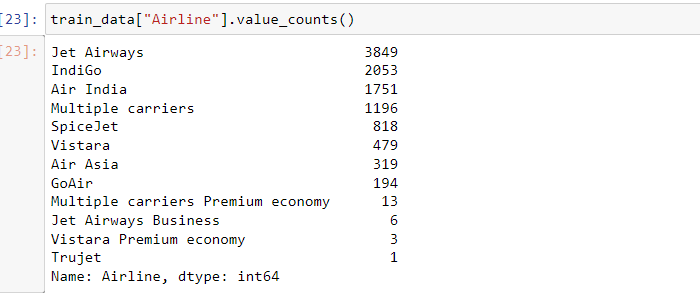
One can find many ways to handle categorical data. Some of them categorical data are,

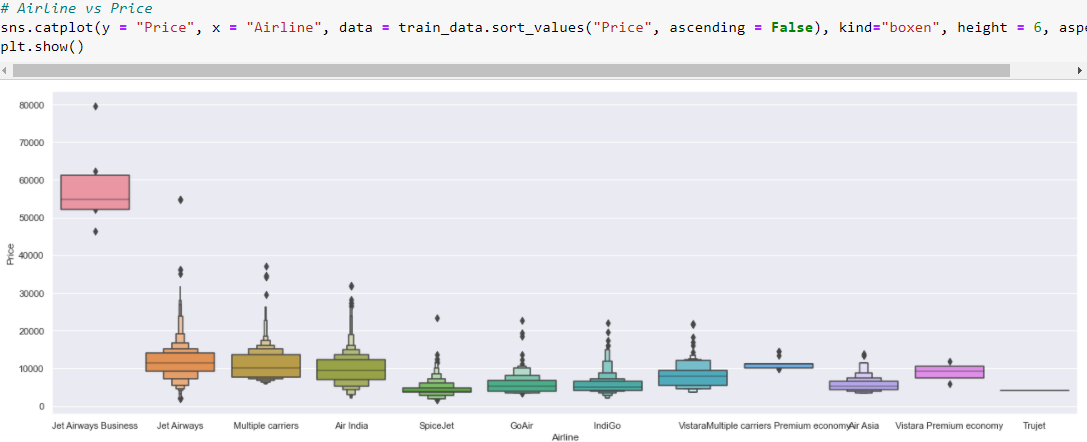
\*\* Nominal data \*\* --> data are not in any order --> \*\* OneHotEncoder \*\* is used in this case

\*\* Ordinal data \*\* --> data are in order --> \*\* LabelEncoder \*\* is used in this case

**Airline: -**

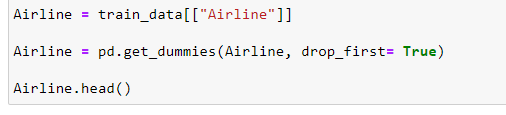
Airline column defines the number of airlines and the count of each flight with the airlines.





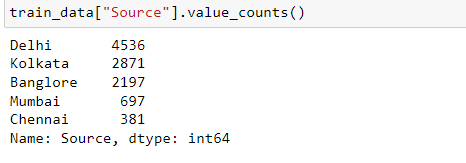
From graph we can see that Jet Airways Business have the highest Price. Apart from the first Airline almost all are having similar median.

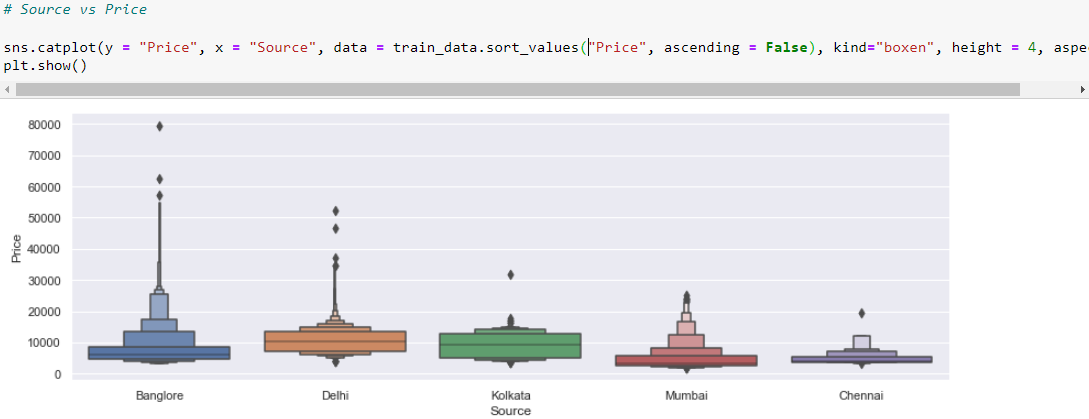
As Airline is Nominal Categorical data we will perform OneHotEncoding.

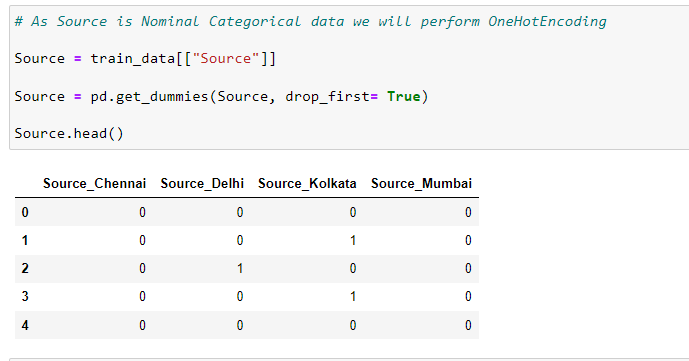


**Source: -**

As Source is Nominal Categorical data we will perform OneHotEncoding



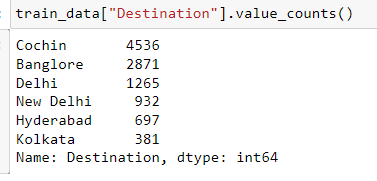


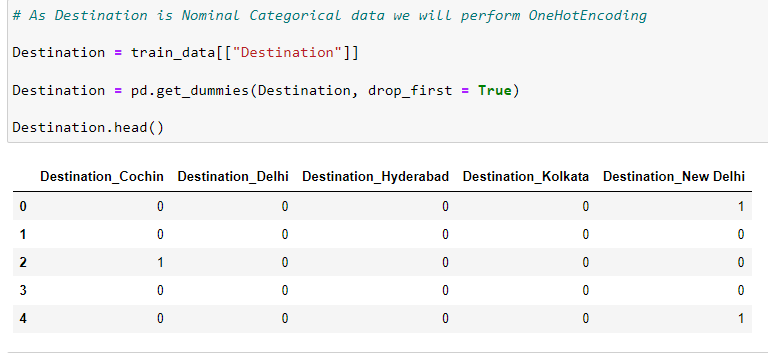


**Destination: -**

Destination column gives the detail information about the count of flights in each of the cities flights are schedules.

Destination is Nominal Categorical data we will perform OneHotEncoding.

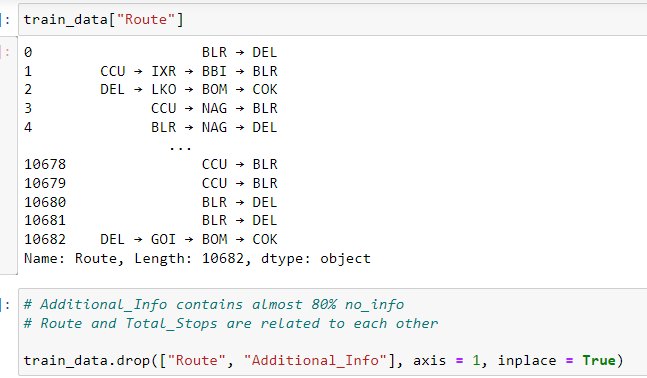




**Route: -**

The ‘Route’ columns mainly tell us that how many cities they have taken to reach from source to destination This column is very important because based on the route they took will directly affect the price of the flight.

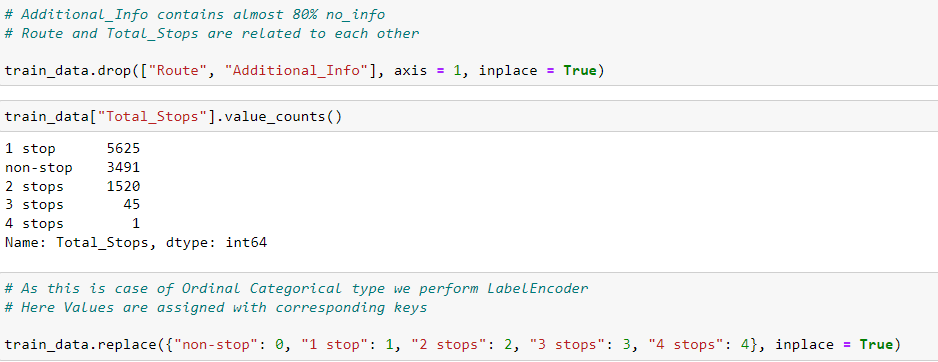
It is an ordinal categorical type data so we have to use LableEncoder.



**Total\_Stops: -**

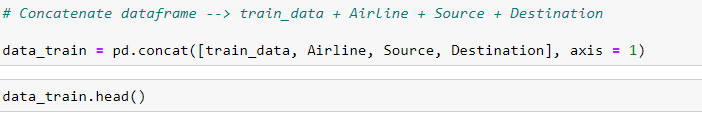
This column is combination of number and a categorical variable like ‘1 stop’ . So we need only the number details from this column so we split that and take the number details only also we change the ‘non stop’ into ‘0 stop’ and convert the column into integer type.

It is an ordinal categorical type data so we have to use LableEncoder



**Step 5: Concatenate the Data frame and drop unwanted data**

We will concatenate the data after labeling and drop the unwanted columns

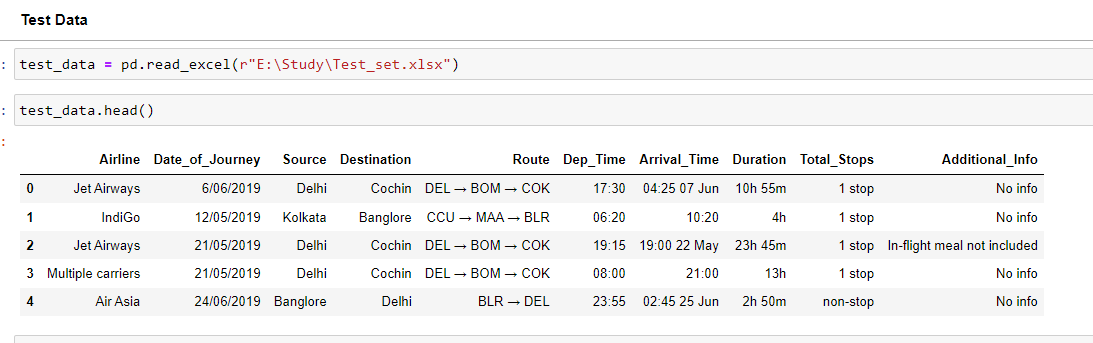




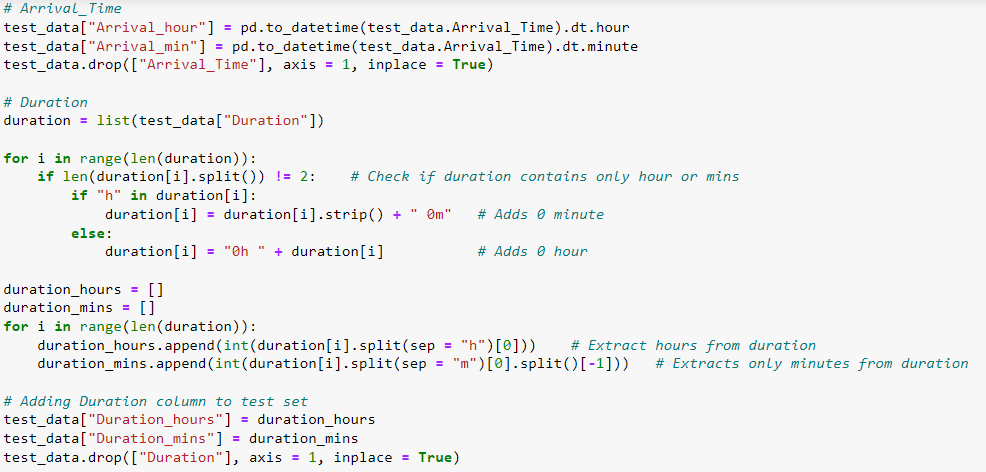
**Step 6: Follow the Same for the Test Dataset**

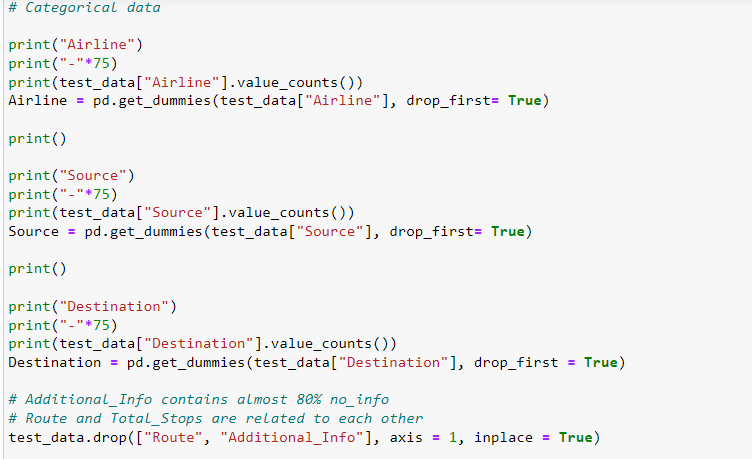
Now we will repeat the steps performed above to analyses the Test Dataset.

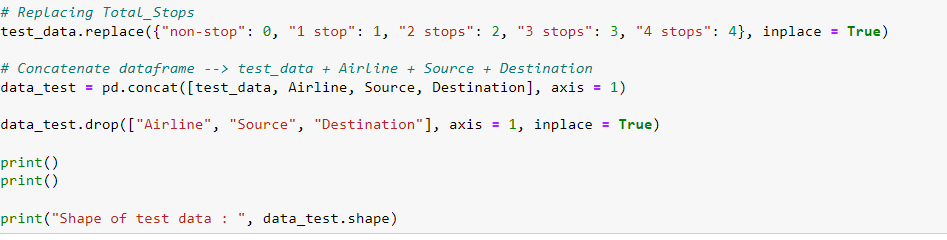
1. Importing Dataset
2. Featuring the data
3. Labeling the data
4. Concatenating the Data









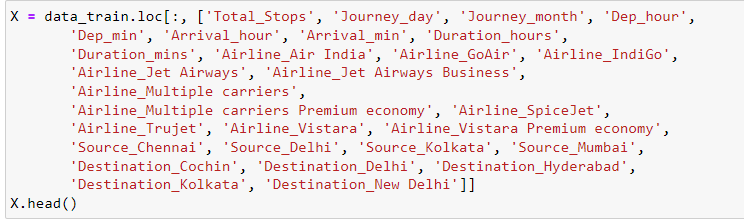


**Step 6: Feature Selection**

Finding out the best feature which will contribute and have good relation with target variable.

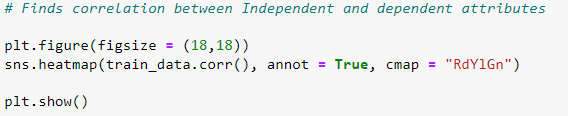
Following are some of the feature selection methods,

1. \*\* heatmap \*\*
2. \*\* feature\_importance\_ \*\*
3. \*\* SelectKBest \*\*

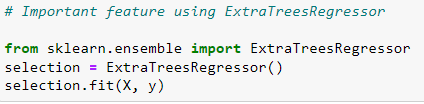




Finding Correlation between the dependent and Independent Attributes and selecting the Best feature using Ensemble technique.

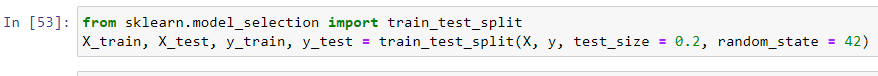


Using ExtraTreesRegressor to find out the best feature:



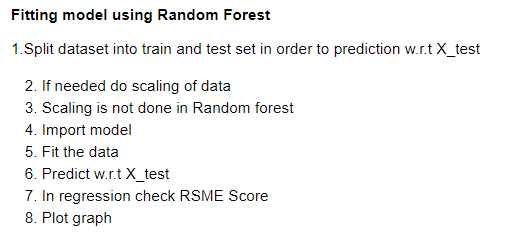
**Step 7: Divide the Dataset into Train and Test**

Now that all our data is numerical after label encoding so we split the data into test and train and drop the price column from the test set because we have to predict the price with our test data set

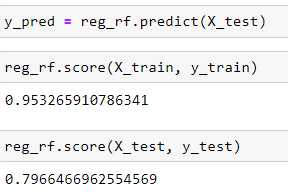


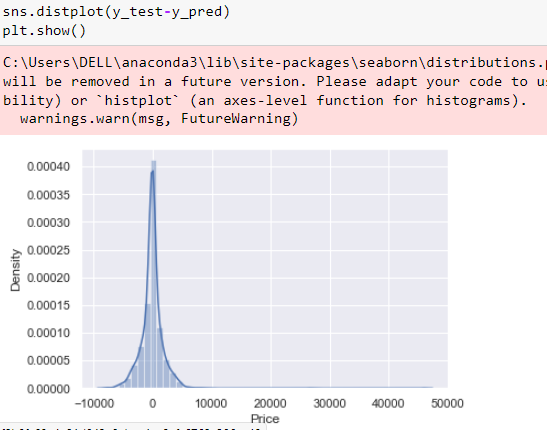
**Step 8: Build Model**

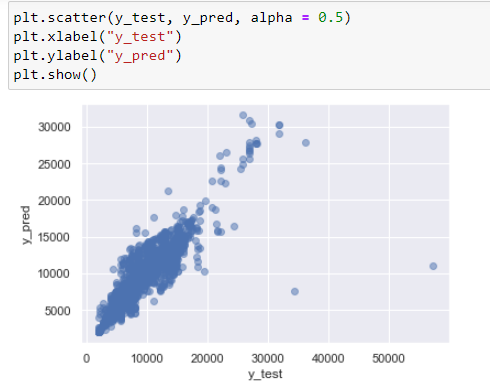
The goal in this step is to develop a benchmark model that serves us as a baseline, upon which we will measure the performance of a better and more tuned algorithm. We are using different Regression Technique and comparing them to see which algorithm is giving better performance then other and At the end we will combine all of them using Stacking and see how our model is predicting

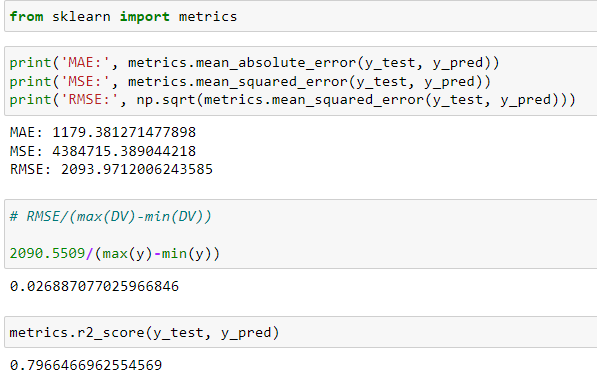






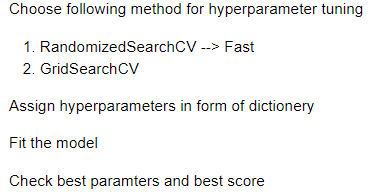


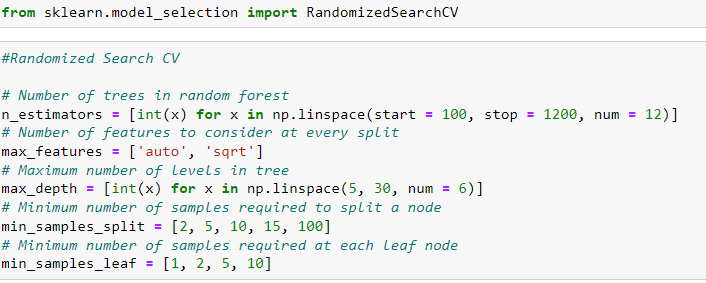


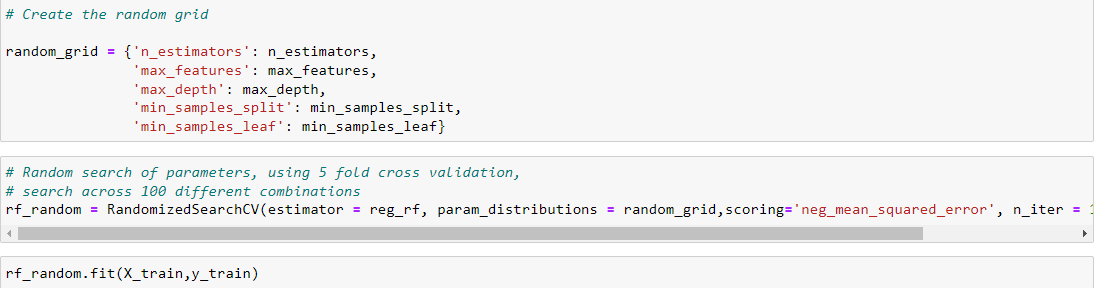


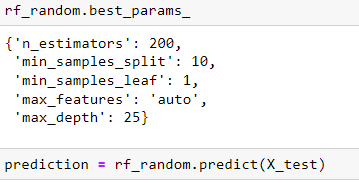
**RMSE(Root Mean Square Error): 0.026887077**

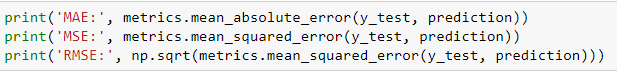
**Step 9: Tuning the model using Hyper parameter Tuning**







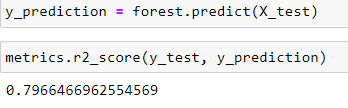




**MAE: 1147.1944005019561**

**MSE: 4039308.629058206**

**RMSE: 2009.8031319157124**



**Conclusion**

In this type of problem, Feature Engineering is the most crucial thing. You can see how we have handled the categorical and numerical data and also how we build ML model on the same dataset. We also check the RMSE score of model so that we can understand how it should perform in our test dataset. At last You can also further improve the Model by Tuning different parameters which are being used in the model. Please let me know your thoughts about this article and do comment if you face any issues.

As always, I welcome feedback and constructive criticism. I can be reached on Mahajan\_atul23@yahoo.co.in