Flight Price prediction

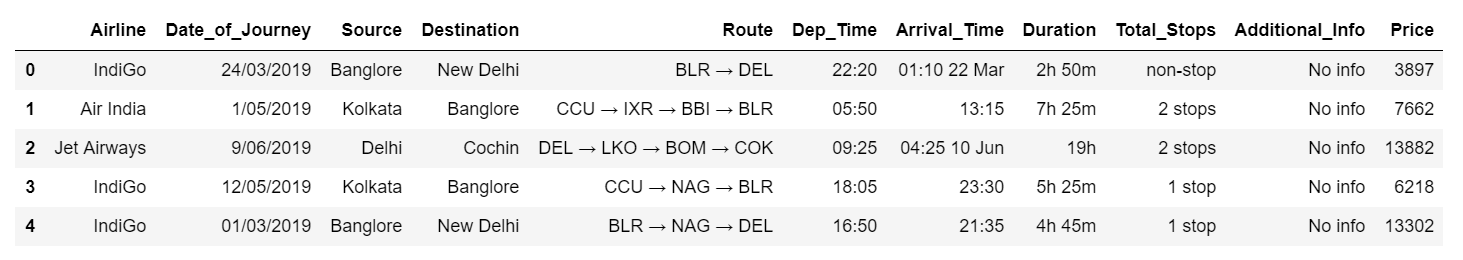
This Blog is generally on ‘Prediction on flight price’ helps you understand the whole ML model building process. So, let’s get started.

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. As data scientists, we are gonna 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.

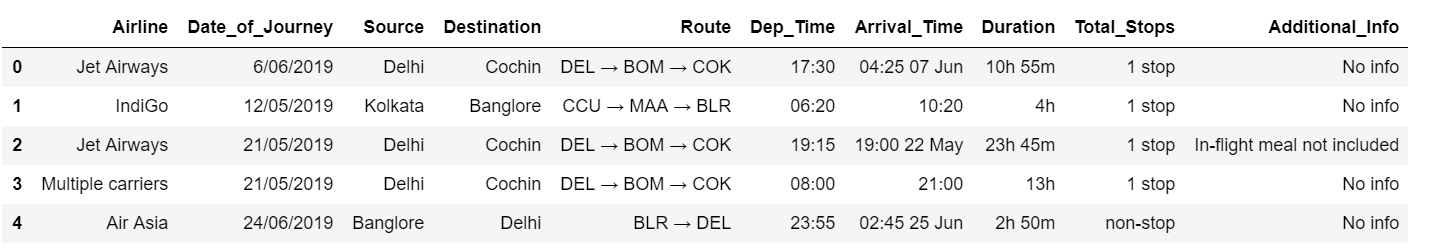
Datasets

We will be using two datasets — Train data and Test data



Screenshot of the **Training data** (10683 rows): Training datarefers to that portion of data used to fit a model.

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

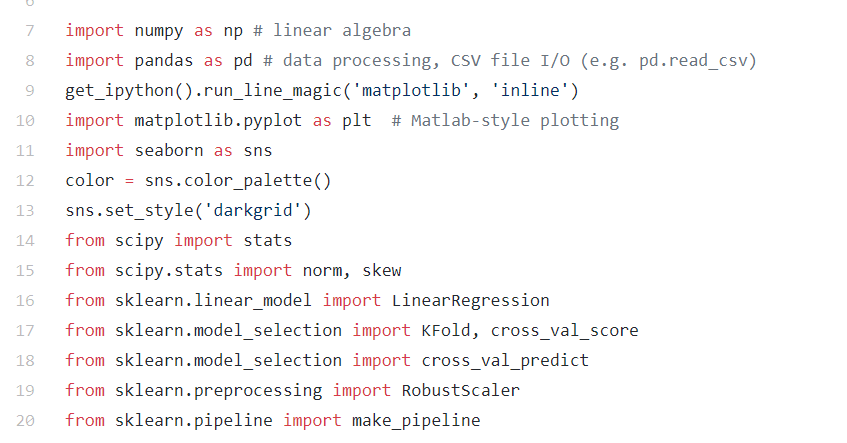


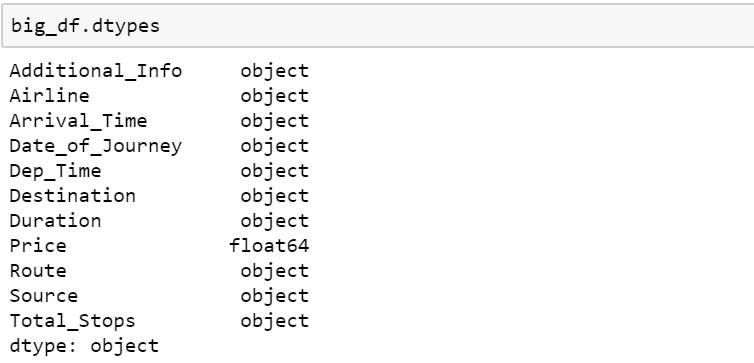
Screenshot of the **Test data** (2671 rows)

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

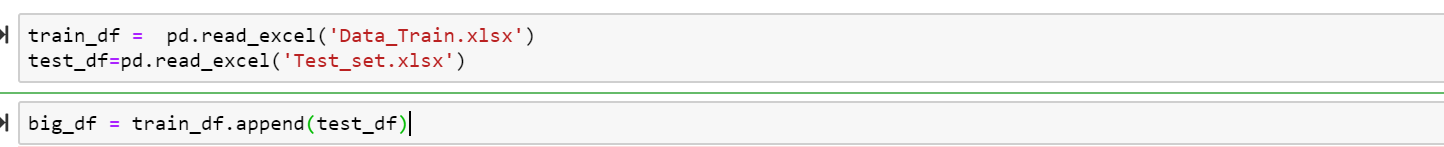
Python Coding

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





**Step 2: Import Train and Test data sets and append them**



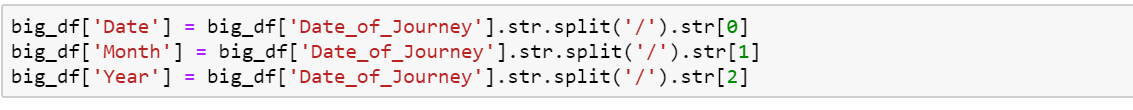
Appending of the data set is done to work together with both train and test at a same time and don,t have to make changes separately.After we apply the transformation then we can separate them again into test and train

**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:**

In the column ‘Date\_of\_Journey’, we can see the date format is given as dd/mm/yyyy and as you can see the datatype is given as object So there is two ways to tackle this column, either convert the column into Timestamp or divide the column into date,Month ,Year. Here , i am splitting the columns

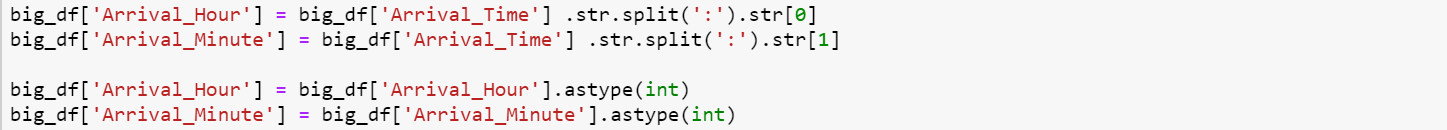


Date\_of\_Journey split into 3 variables (Date, Month, Year )

**Arrival\_Time:**

In the column ‘**Arrival\_Time’,**if we see we have combination of both time and month but we need only the time details out of it so we split the time into ‘Hours’ and ‘Minute’.

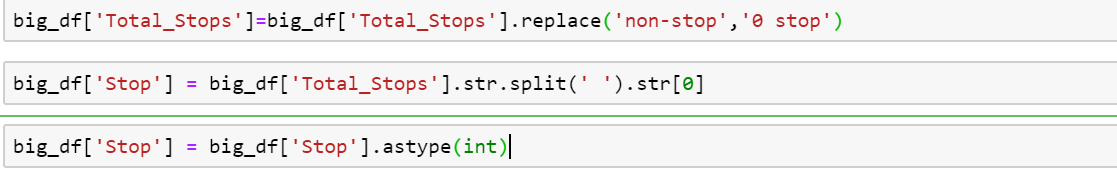




Arrival\_Time split into 2 variables (Hour, Minute)

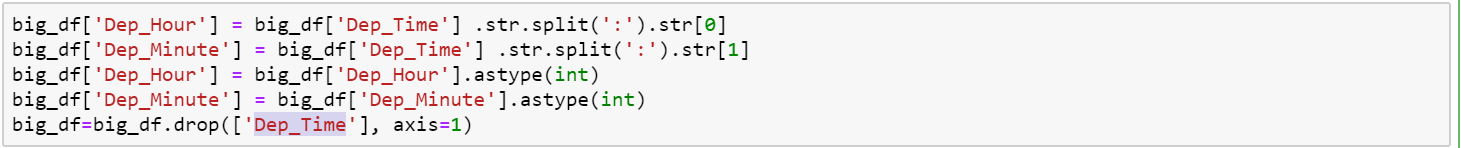
**Total\_Stops:**

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



**Dep\_Time:**

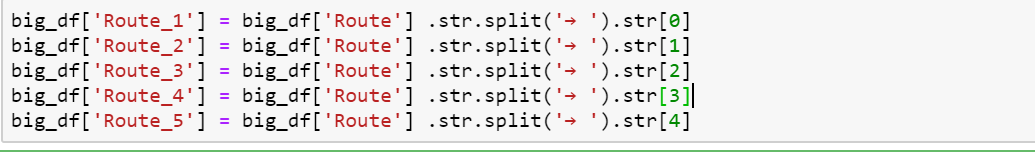
As same as ‘Arrival\_time’. we split this column also in hour and minute and convert it into integer



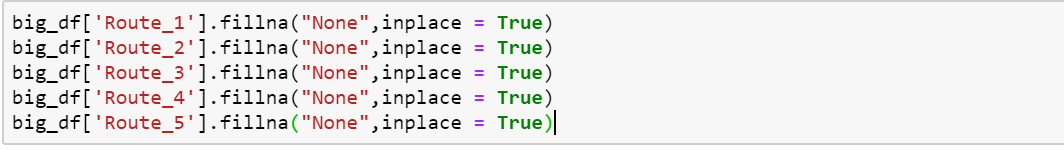
Dep\_Time split into 2 variables (Hour, Minute)

**Route:**

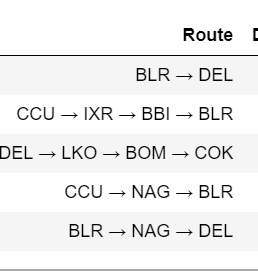
The ‘Route’ columns mainly tell us that how many cities they have taken to reach from source to destination. This column is very important because based on the route they took will directly effect the price of the flight So We split the Route column to extract the information .Regarding the ‘Nan’ values we replace those ‘Nan’ values with ‘None’ .



Route split into 5 variables



Replacing the Nan values with ‘None’

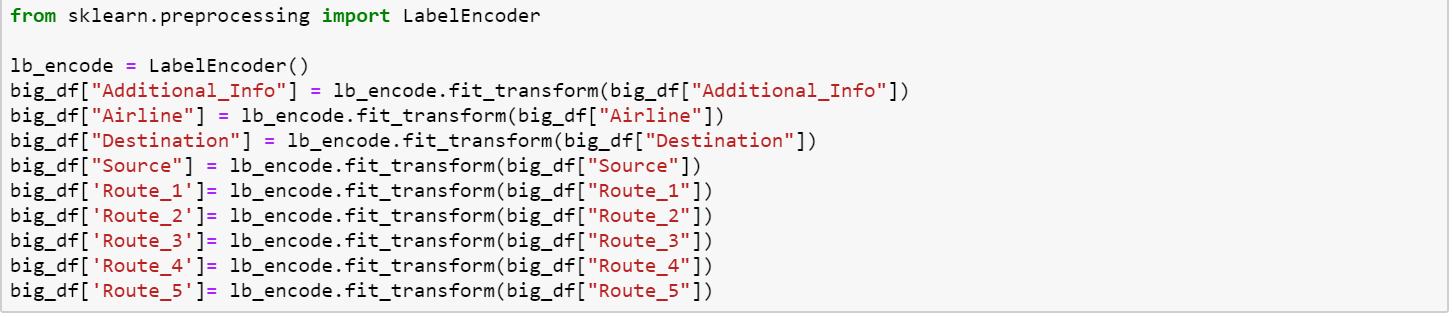




After splitting

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

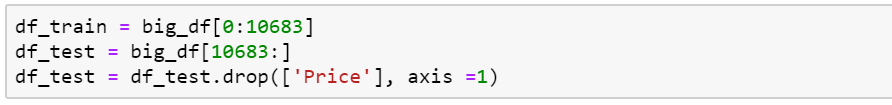
To convert categorical text data into model-understandable numerical data, we use the Label Encoder class. So, all we have to do, to label encode a column is import the LabelEncoder class from the sklearn library, fit and transform the column of the data, and then replace the existing text data with the new encoded data.

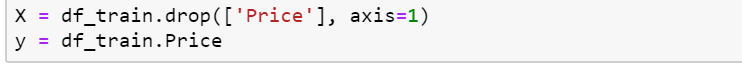


Label encoding of Categorical variables

**Step 5:** **Divide the data set into test and train**

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



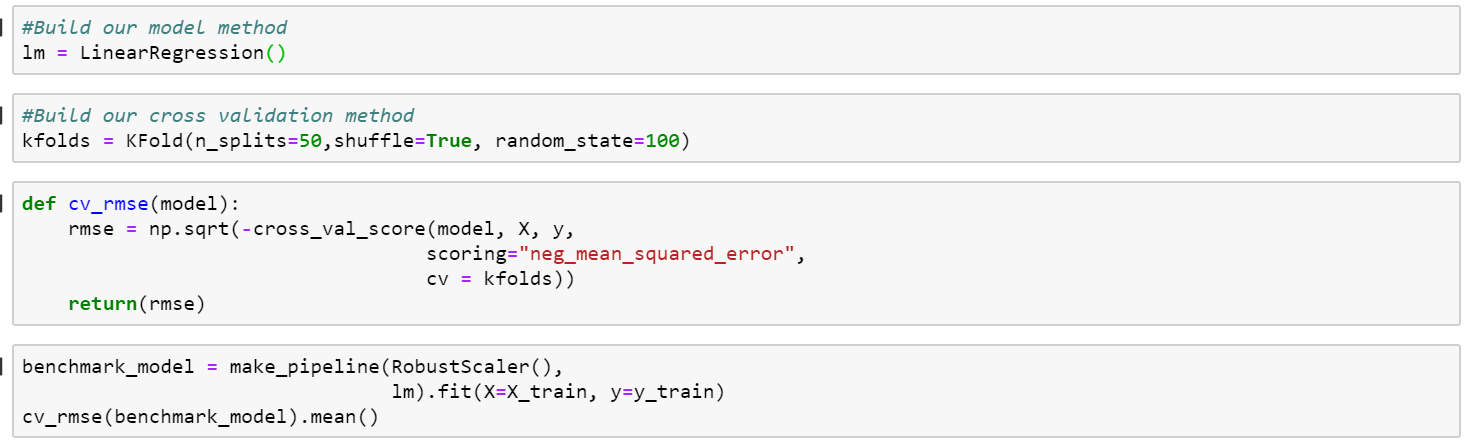


X — independent variables; y — dependent variable

**Step 6: 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 than other and At the end we will combine all of them using Stacking and see how our model is predicting

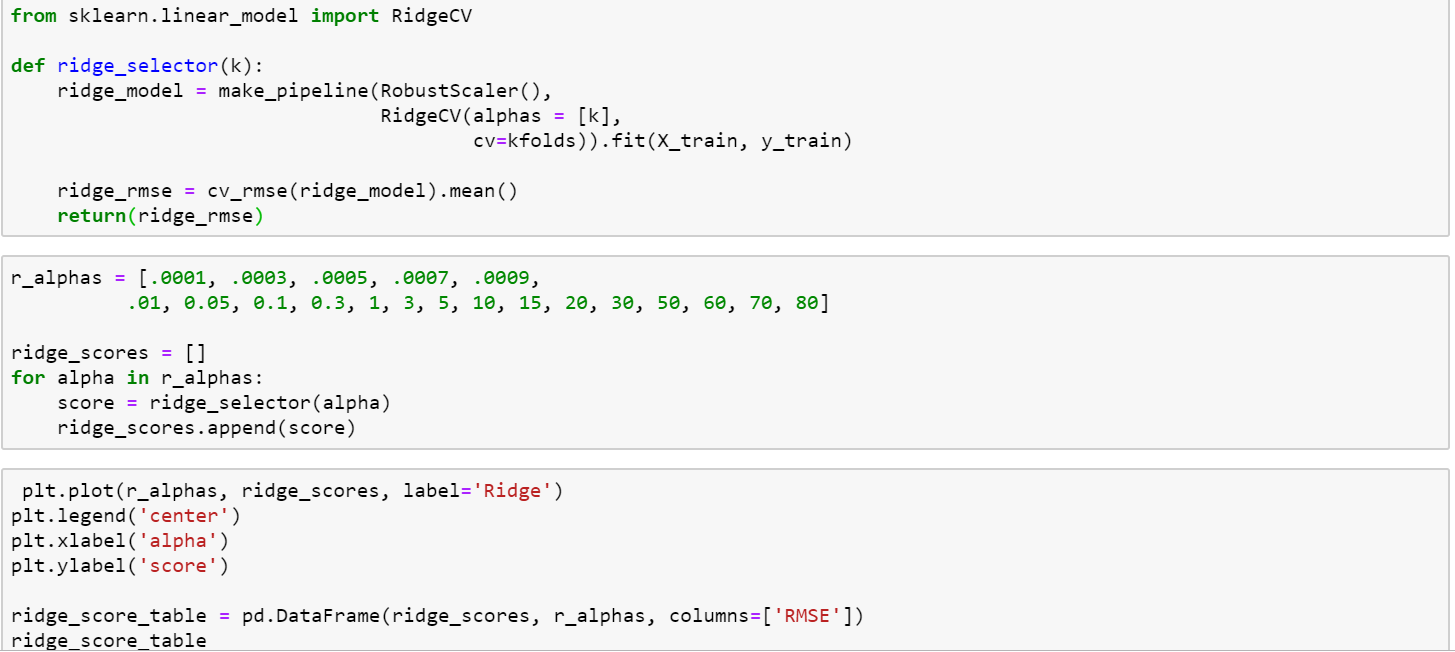
1. **Linear Regression:** You can check the below link for more details on the Regression Technique that we are using

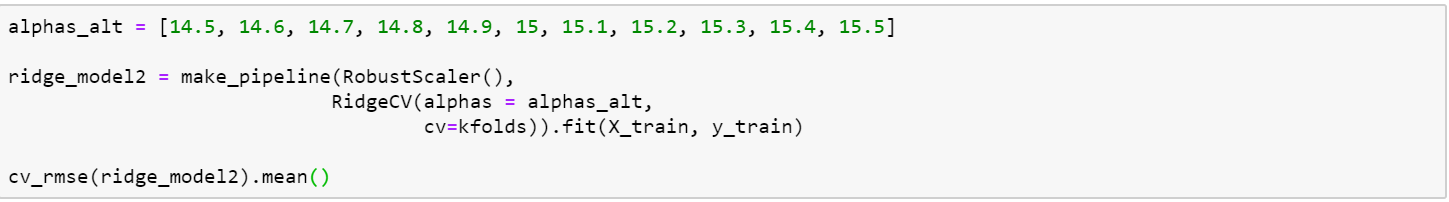


**RMSE (Root Mean Square Error):**3238.316987636252

**2. Ridge Regression:** You can check the below link for more details on the Regression Technique that we are using

[A comprehensive beginners guide for Linear, Ridge and Lasso Regression](https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/" \t "_blank)

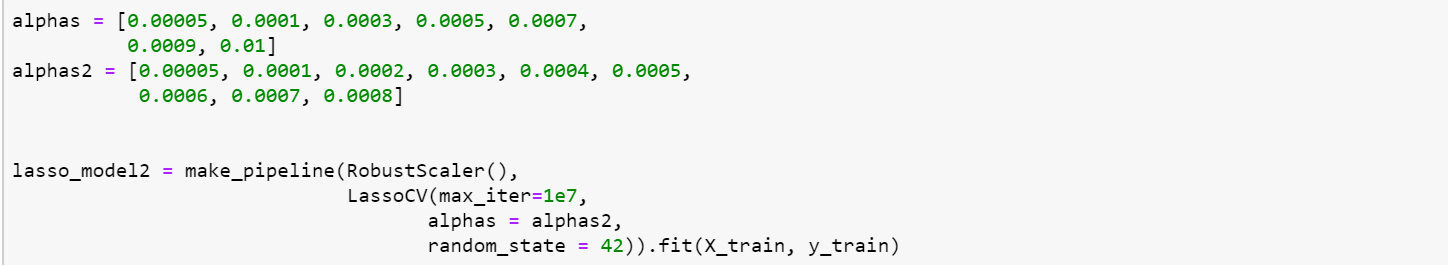




**RMSE (Root Mean Square Error):**3238.153926834792

**3. Lasso Regression:** You can check the below link for more details on the Regression Technique that we are using

[A comprehensive beginners guide for Linear, Ridge and Lasso Regression](https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/" \t "_blank)



**RMSE (Root Mean Square Error):**3273.005929514414

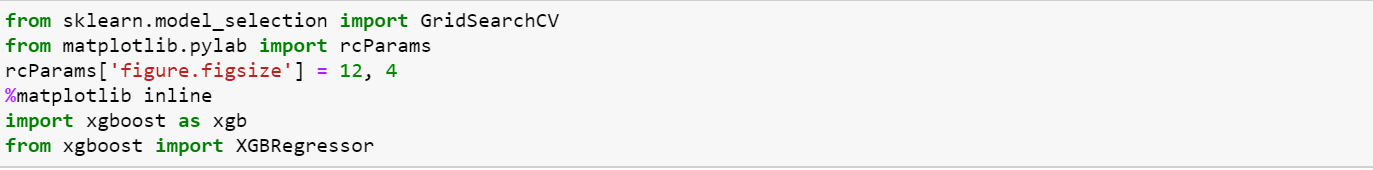
**4. Elastic Net Regularization:**You can check the below link for more details on the Regression Technique that we are using

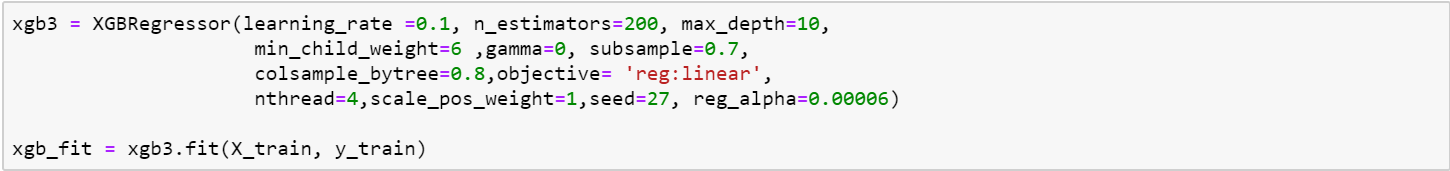
[Lasso, Ridge and Elastic Net Regularization](https://medium.com/@jayeshbahire/lasso-ridge-and-elastic-net-regularization-4807897cb722" \t "_blank)

**RMSE (Root Mean Square Error):**3238.296057360342

**5.Extreme Gradient Boosting (XGBoost):** You can check the below link for more details on the Regression Technique that we are using

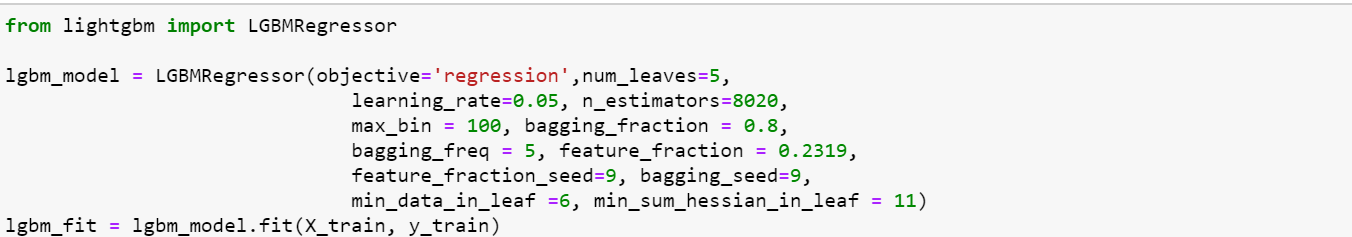
[An End-to-End Guide to Understand the Math behind XGBoost](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/" \t "_blank)





**RMSE (Root Mean Square Error):**1281.0225332975244

**6.Light GBM:** You can check the below link for more details on the Regression Technique that we are using

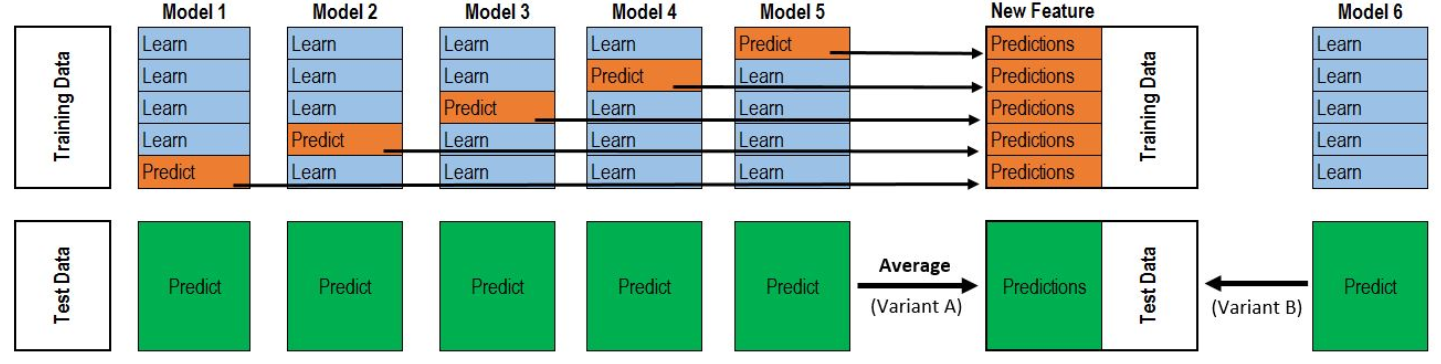


**RMSE (Root Mean Square Error):**1747.2331238078746

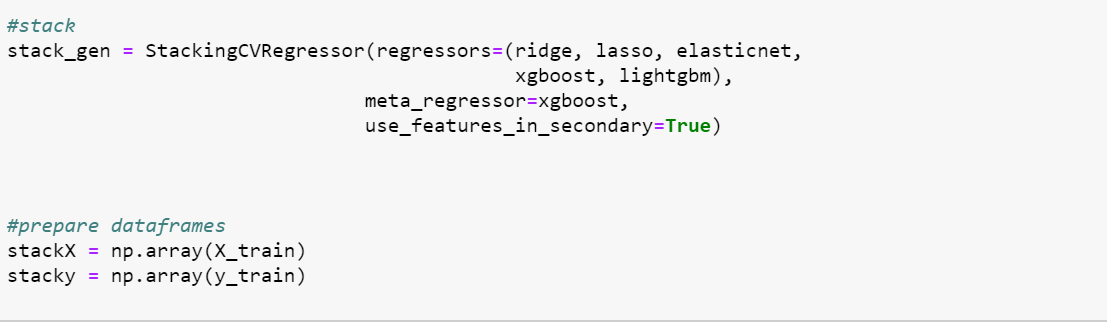
**7. STACKING:**

Stacking is an ensemble learningtechnique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The base level models are trained based on a complete training set, then the meta-model is trained on the outputs of the base level model as features.

[A Comprehensive Guide to Ensemble Learning (with Python codes)](https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/" \t "_blank)

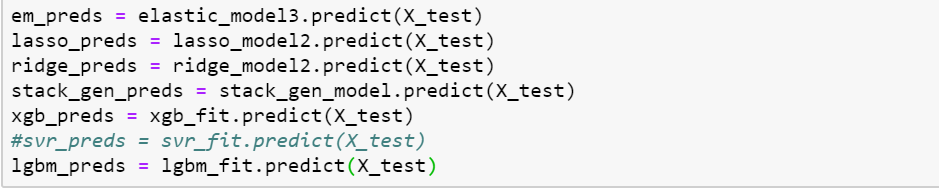


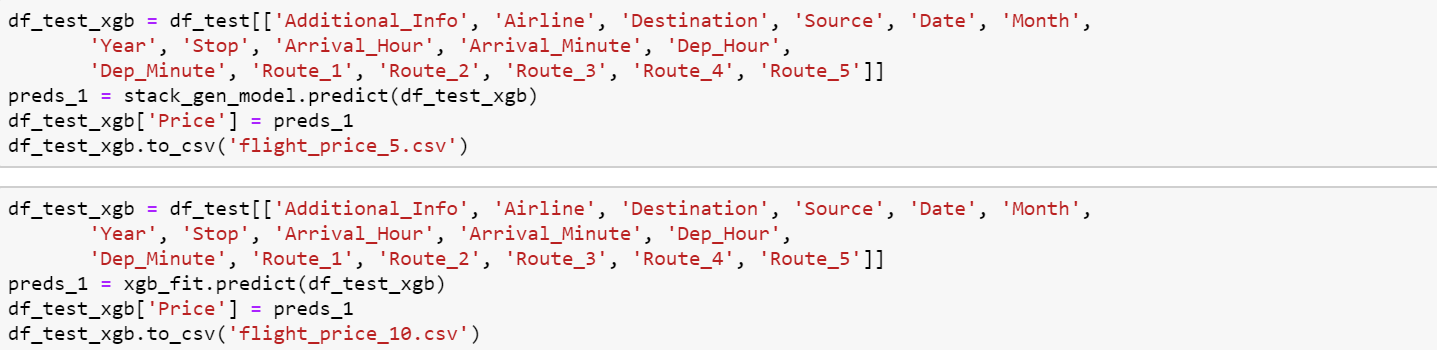




**RMSE (Root Mean Square Error):**1372.627469

From the above different Regression Technique, we can see XGboost is performing really good in regards to all. Finally, we will use this to predict our test data





Export it into a csv file and submit it

**Final Word**

In this type of problem Feature Engineering is the most crucial think. You can see how we have handled the categorical and numerical data and also how we build different ML model on the same dataset. We also check the RMSE score of each model so that we can understand how it should perform in our test dataset. At last, you can also further improve the Model by Tunning different parameters which are being used in the model.