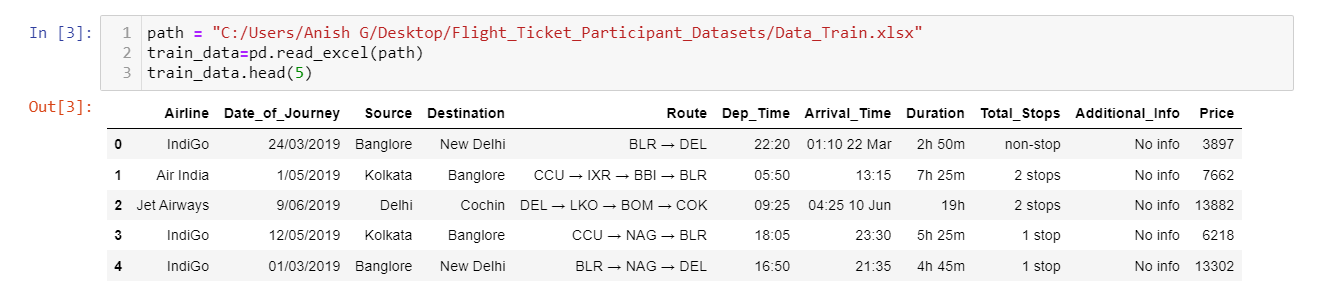
**Flight Price Prediction**

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. 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.

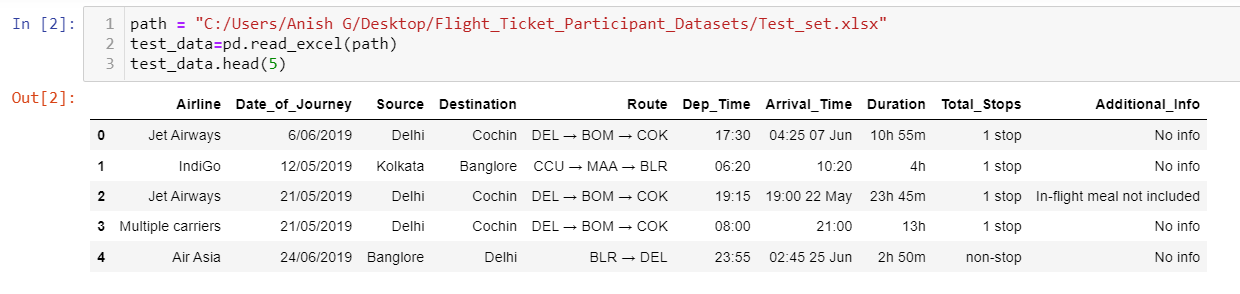
**Datasets**

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



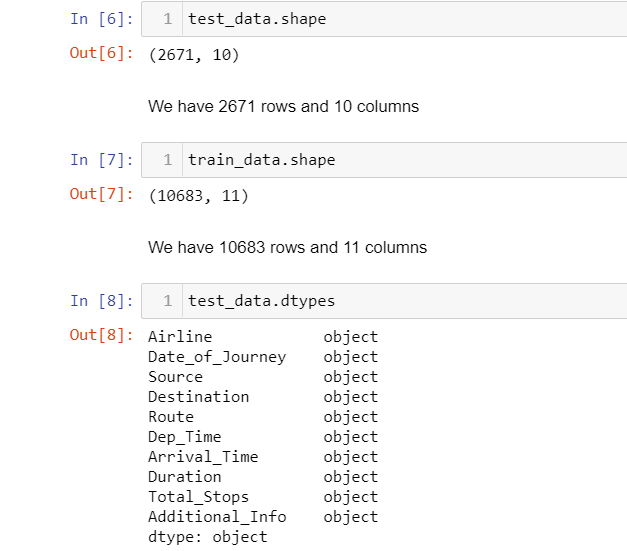
Screenshot of the **Training data** (10683 rows): Training datarefers to that portion of data used to fit a model.

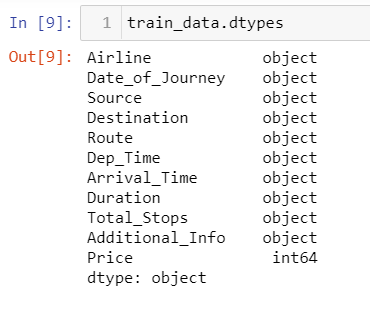
Training data is combination of both categorical and numerical also we can see some special character also being used because of which we have to do data transformation on it before applying it to our model.



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).

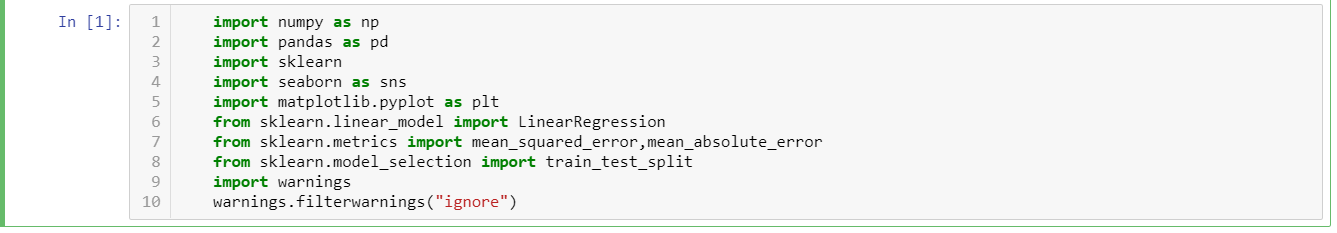




The details show the test and train data’s number of rows, number of columns and its datatypes.

**Python Coding**

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

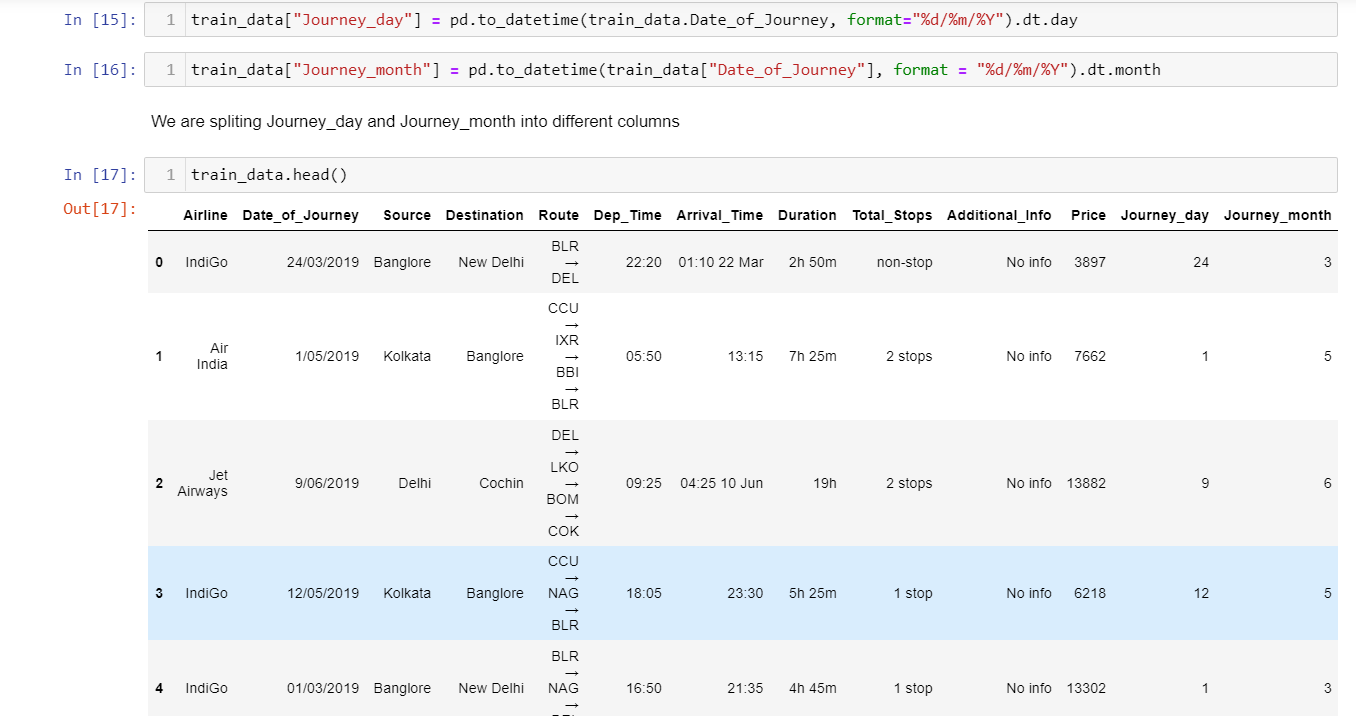


**Step 2: Feature Generation**

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 important because for a high prediction score you need to continuously make changes in it.

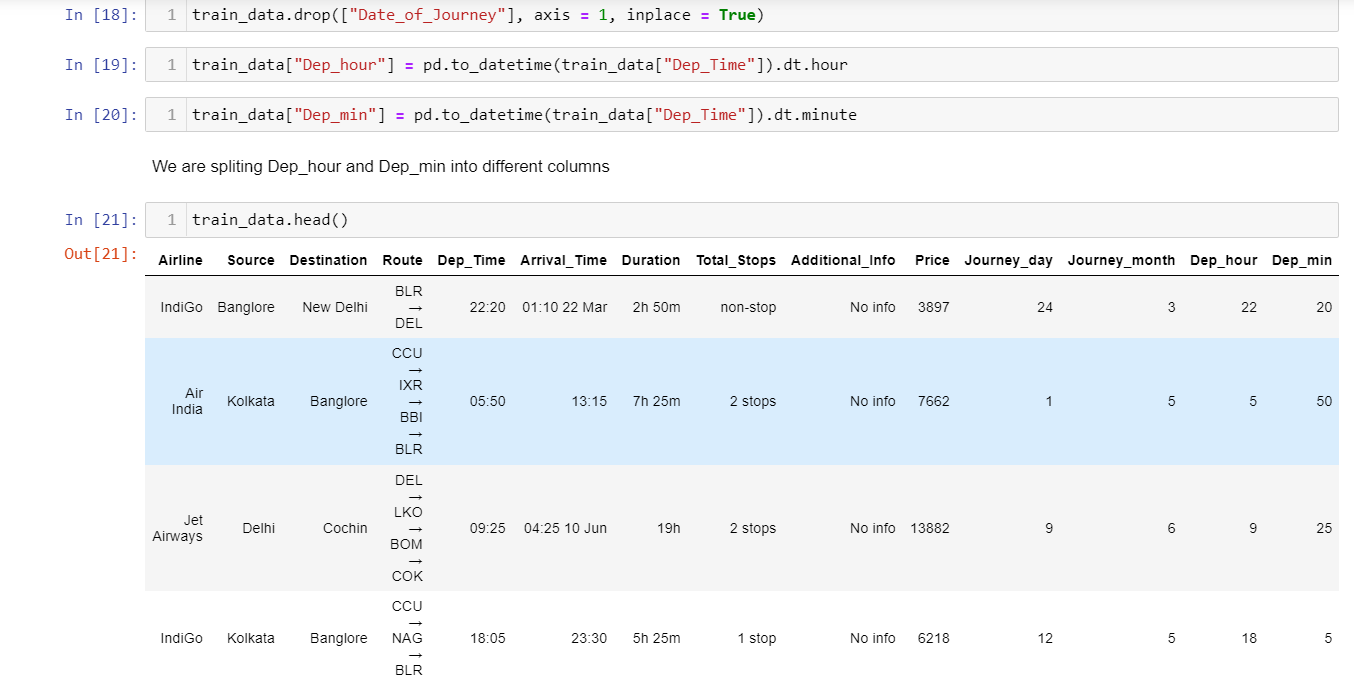
* **Date of Journey:**

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 Journey Day, Journey Month, and Journey year. Here, I am splitting the columns.



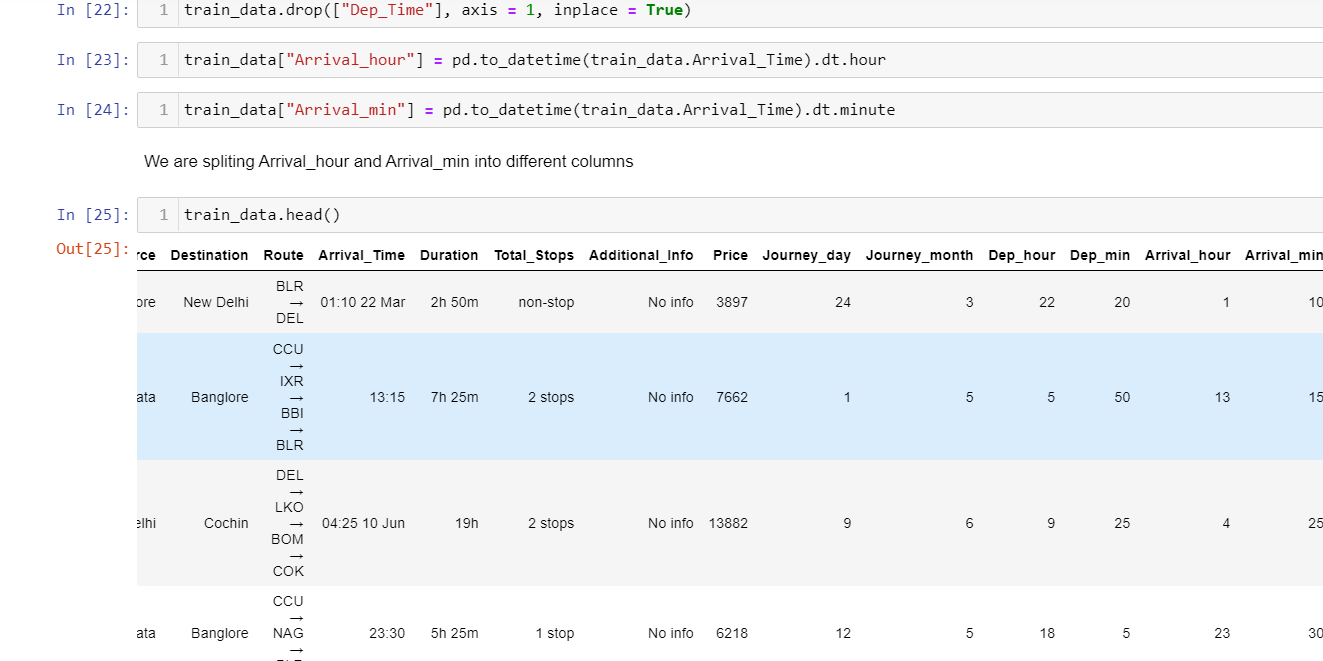
* **Departure time:**

In the column ‘Departure Time’**,** if we see we have combination of both Hours and minutes. so, we split the time into ‘Hours’ and ‘Minutes’.



* **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’.



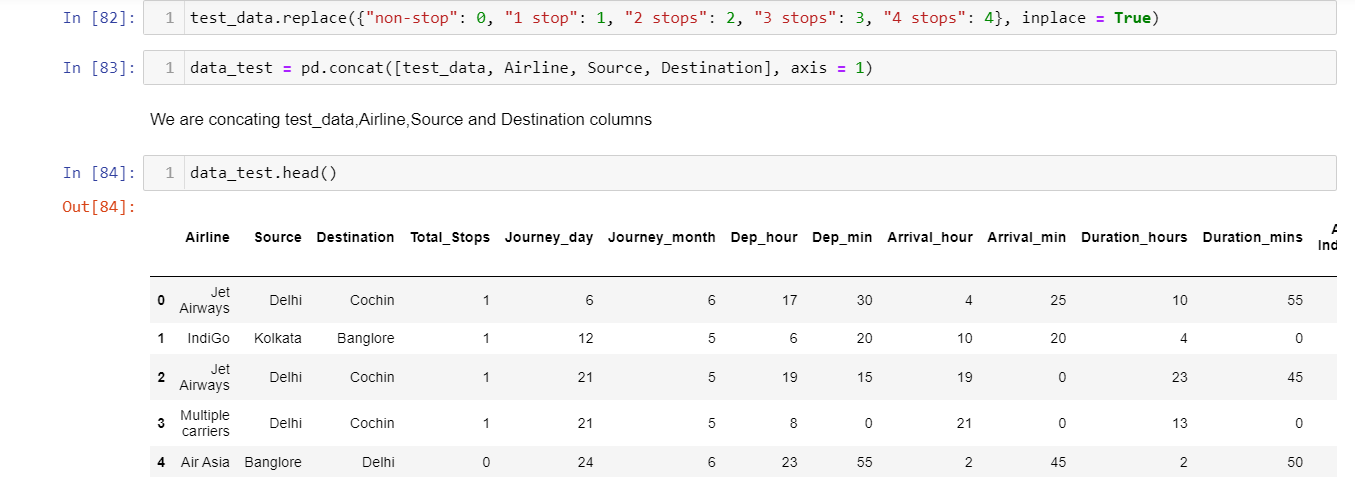
* **Duration hours:**

In the column ‘Duration hours**,** if we see we have combination of both hours and minute, but we need only the time details out of it, so we split the time into ‘Hours’ and ‘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 ‘nonstop’ into ‘0’ and convert the column into integer type.

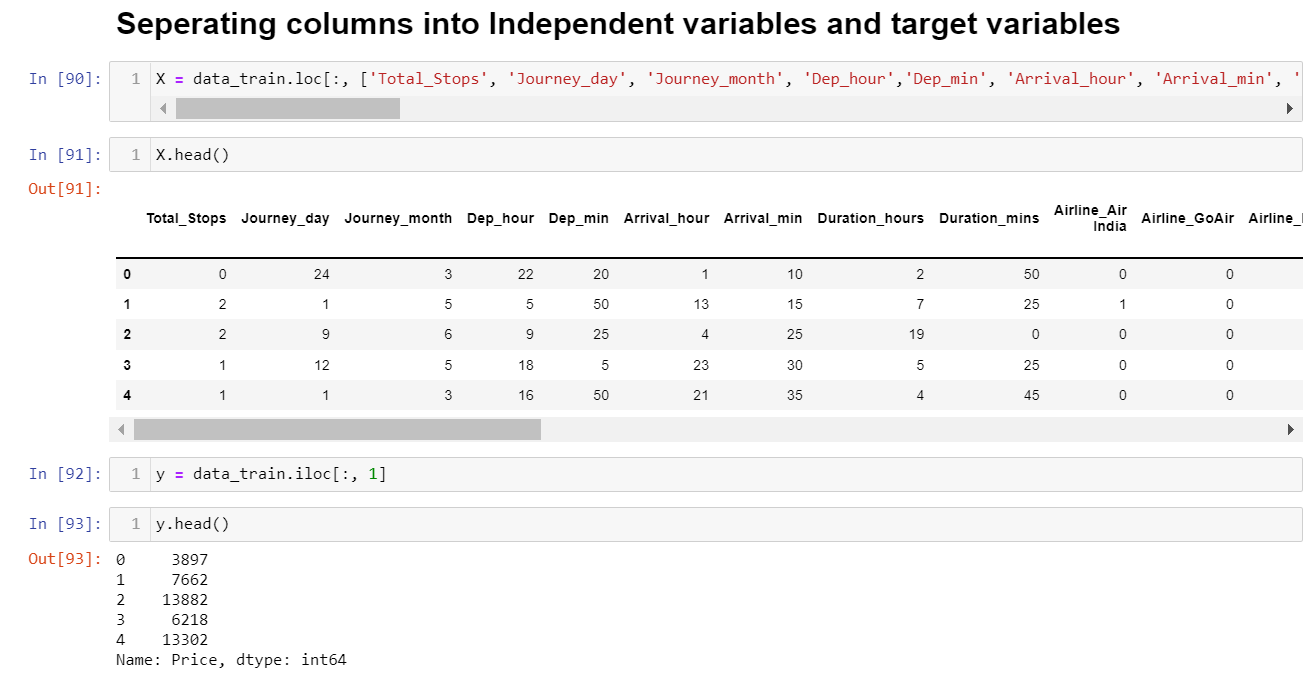


**Step 3: Prepare categorical variables for model using label encoder.**

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 Label Encoder 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.

**Step 4:** **Divide the data set into test and train.**

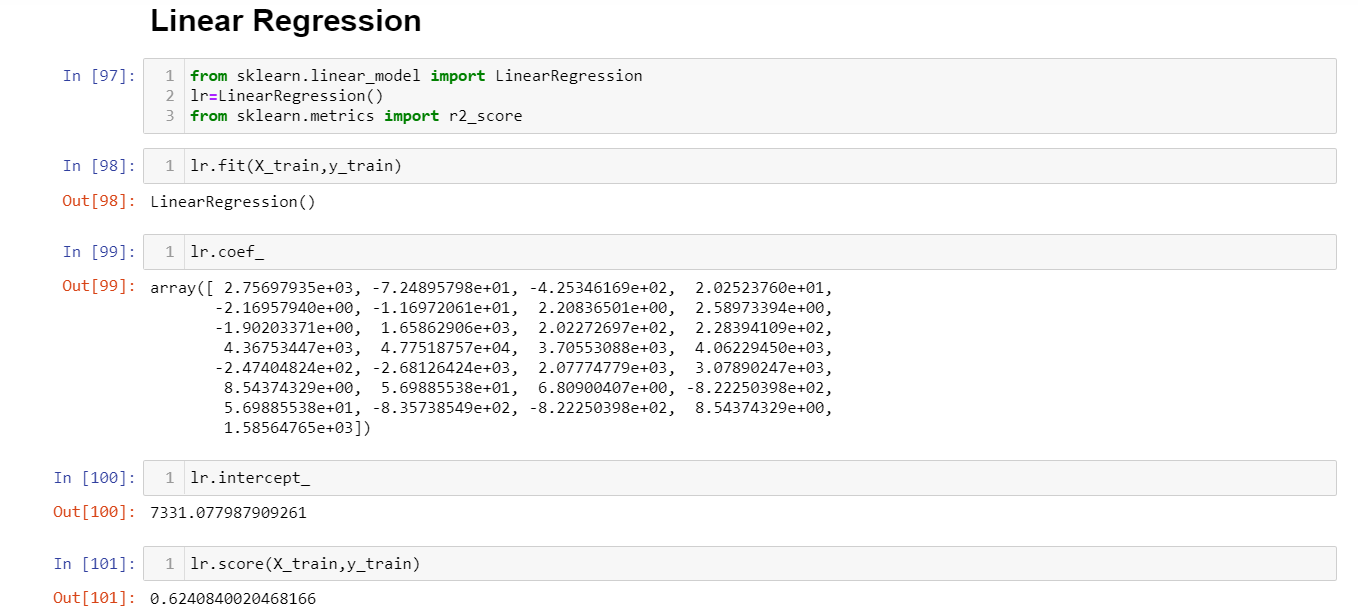
Now that all our data is numerical after label encoding, so we split the data into test and train and drop the price column from the test set because we have to predict the price with our test data set.

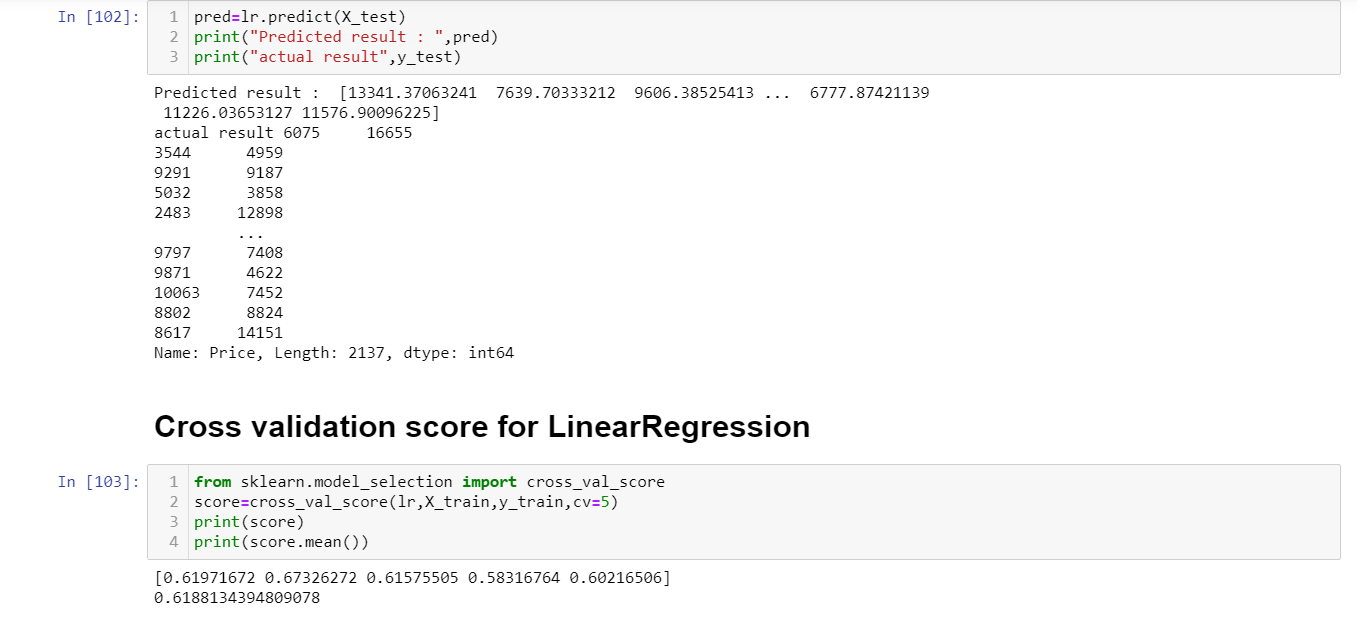


**Step 5: Building Machine Learning Models**

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.

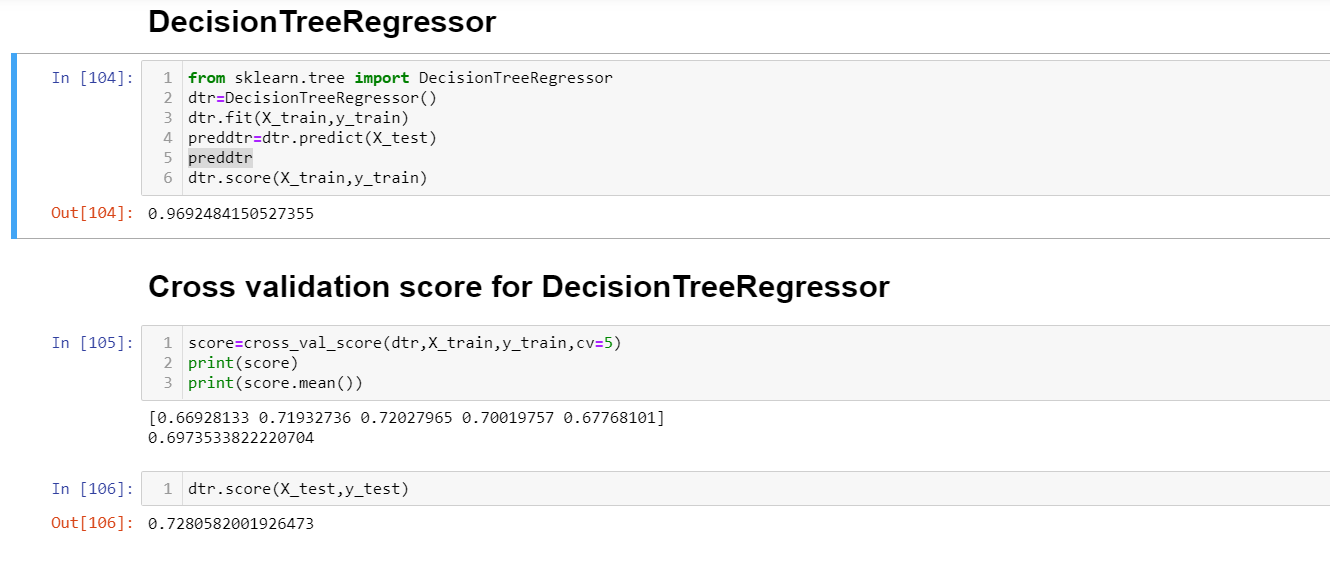
1. **Linear Regression:**





From above we come to know that Linear Regression score is 0.6240 and its cross-validation score is 0.6188.

1. **Decision Tree Regressor:**



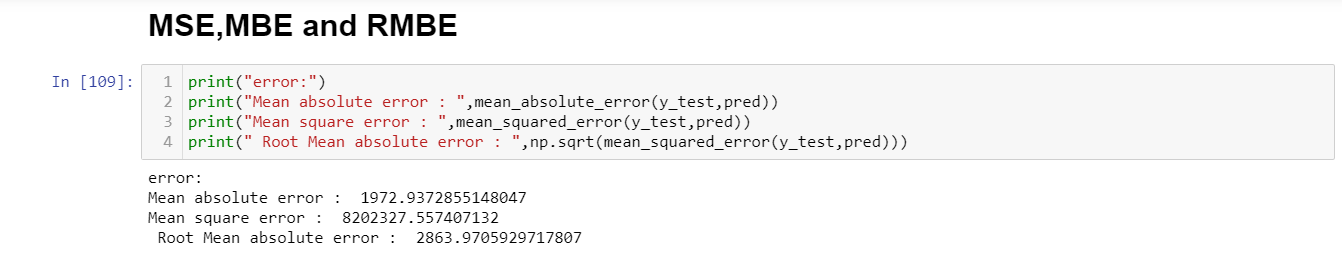
From above we come to know that Decision Tree score is 0.7280, cross-validation score is 0.6973 and its accuracy score is 0.9692.

1. **Support Vector Regressor:**

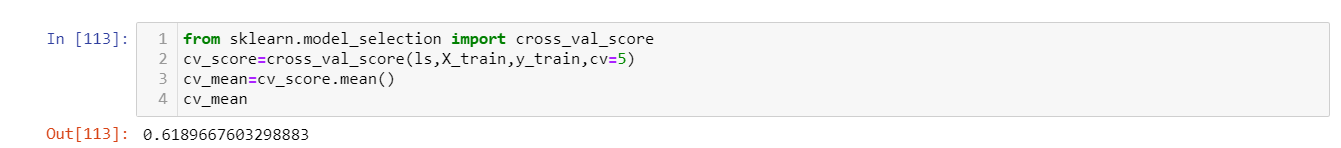


From above we come to know that Support Vector accuracy score is 0.0026 and its cross-validation score is 0.0023. We find that model is not performing well with SVR.

1. **Mean Absolute Error, Mean Square Error and Root Mean Absolute Error:**

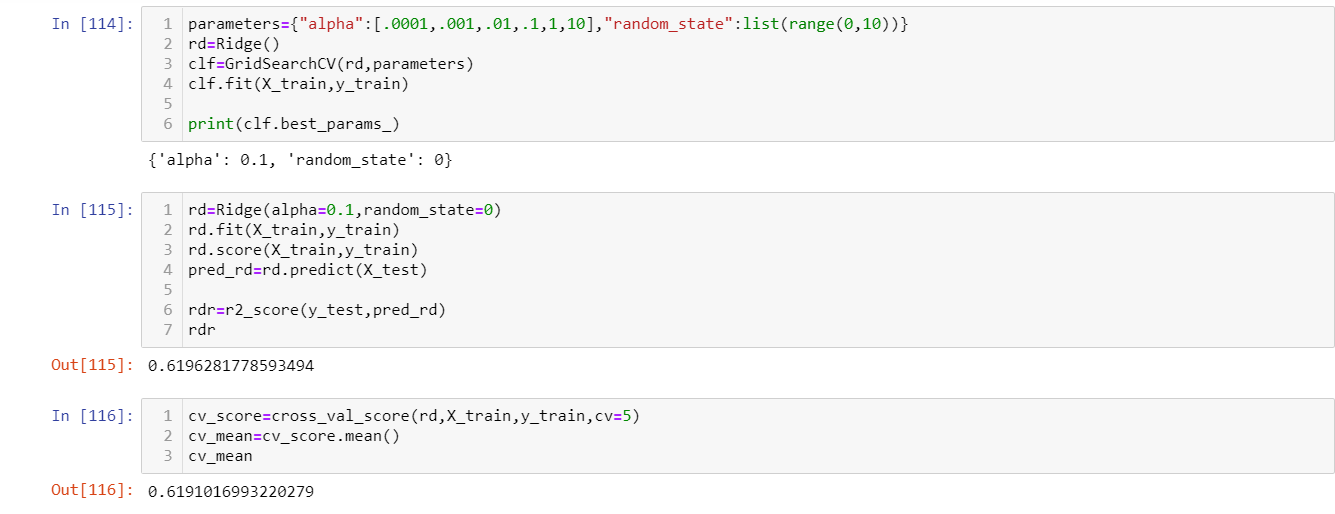


1. **Regularization:**
   1. **Lasso Regressor**



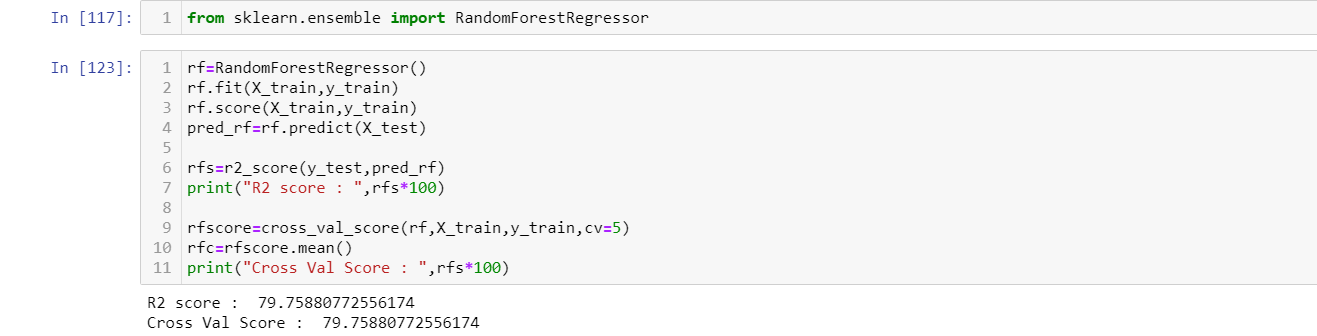
From above we come to know that Lasso accuracy score is 0.6190 and its cross-validation score is 0.6189.

* 1. **Ridge Regressor:**



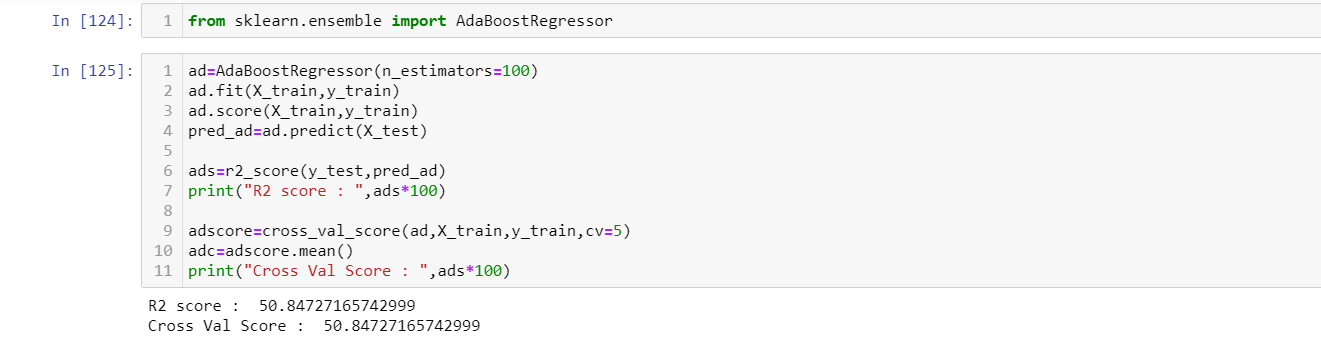
From above we come to know that Ridge accuracy score is 0.6196 and its cross-validation score is 0.6191.

1. **Ensemble Methods:**
   1. **Random Forest:**



From above we come to know that Random Forest accuracy score is 79.75 and its cross-validation score is 79.75.

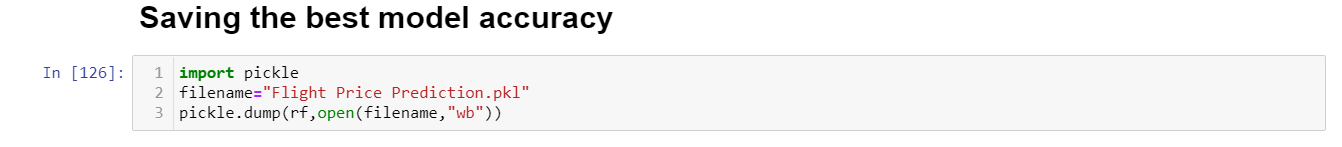
* 1. **AdaBoost Regressor:**



From above we come to know that AdaBoost Regressor accuracy score is 50.84 and its cross-validation score is 50.54.

1. **Saving the best model accuracy:**

As we can find from above Build model Random Forest is performing well and has good accuracy score with cross validation. So, we are going to save Random Forest as the best model.



1. **Conclusion:**

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 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.

