

**FLIGHT PRICE PREDICTION**

**BHAGYASHREE A MOURYA**

**DATE**

**01-05-2022**

**REPORT**

**&**

**ANALYSIS**

|  |  |
| --- | --- |
| **Sr.no**. | **CONTENT** |
| **1.** | **Introduction** |
| **2.** | **Objectives** |
| **3.** | **Problem Definition** |
| **4.** | **EDA Concluding Remarks** |
| **5.** | **Data Analysis** |
| **6.** | **Pre-processing** |
| **7.** | **Building Machine Learning Models** |
| **8.** | **Concluding Remarks** |



**INTRODUCTION**

Machine Learning Algorithms have a wide application, like fraud detections, email filtering etc.

One such application in Machine Learning lies in the Aviation Industry, to predict the price of the flights.

There are various features/factors which impact the prices of flights.

These factors help create a pattern to decide the price of flight, and the machine learning models get trained on this pattern to make the predictions in future, automating the process and making the process quicker.

**Objectives:**

**There are basically two approaches to solve this problem. These involve considering it as a regression or classification problem. Algorithms can be applied to predict whether the price of the ticket will drop in the future. In this project, I will consider it as a regression problem, thus predicting the ticket price.**

**Problem Definition:**

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. 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.

Size of training set: **10683** records

Size of test set: **2671** records

**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.

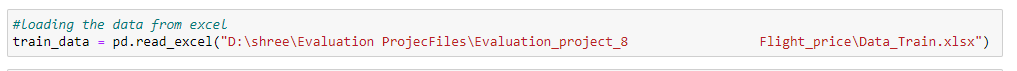
**Duration**: Total duration of the flight.

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

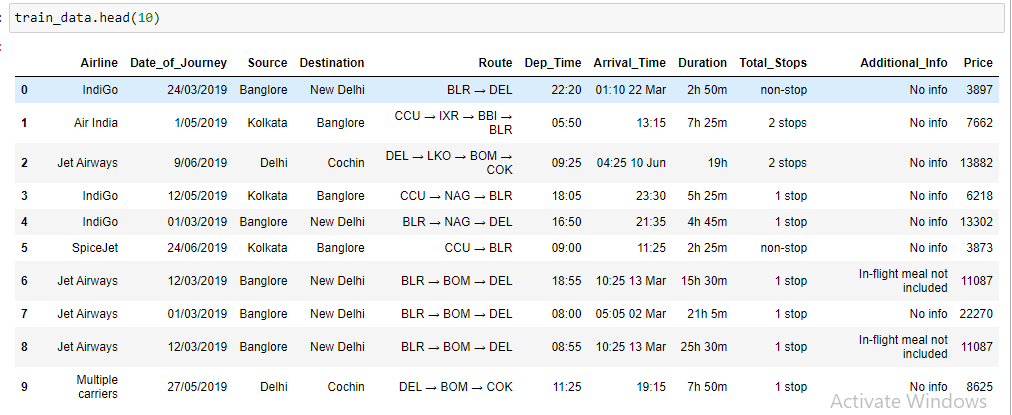
**Additional\_Info**: Additional information about the flight

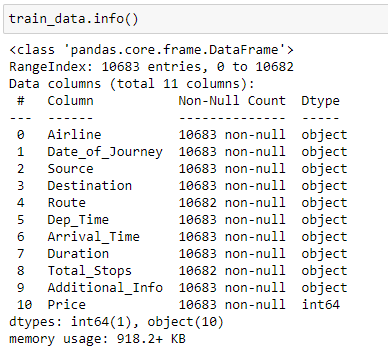
**Price**: The price of the ticket

**EDA Concluding Remarks**



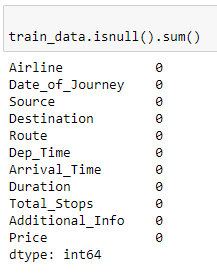
Sample of the data



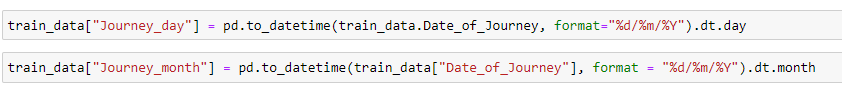
Proceed to explore the data 

We observe that all the columns have Object data types except the Price column which have data types integer.

Now we check the counts for the null values in our datasets.



Treating **Date\_of\_Journey column:**

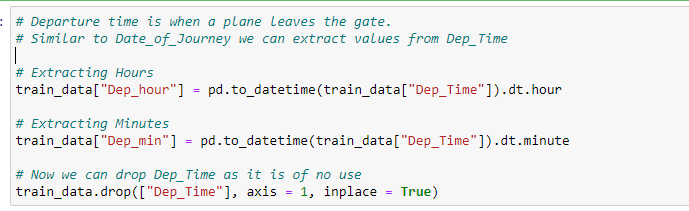


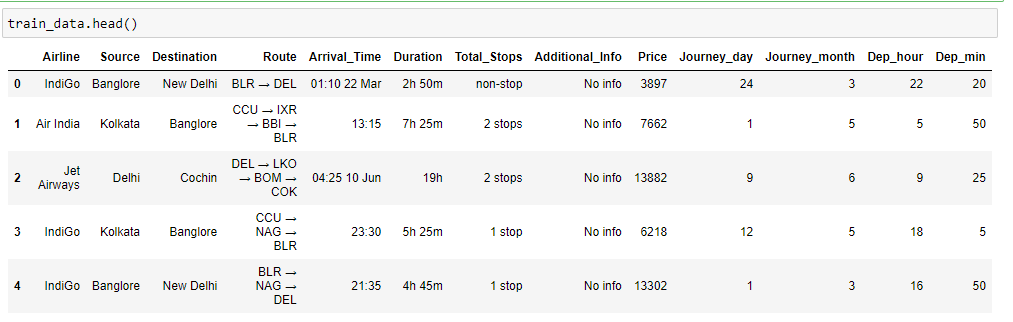
**Output:**

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

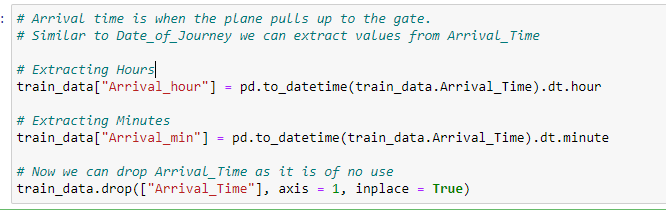


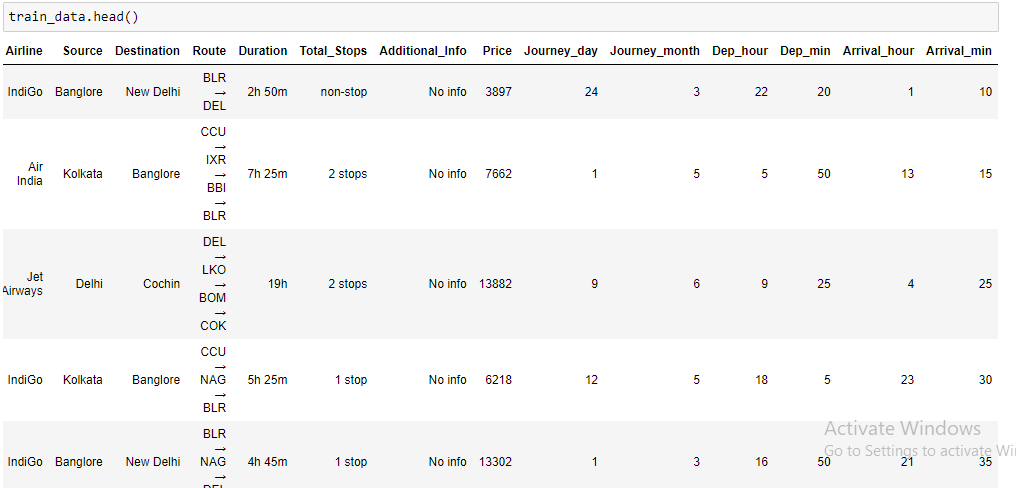
**Treating Dep\_Time column:**





**Treating Arrival\_Time column:**

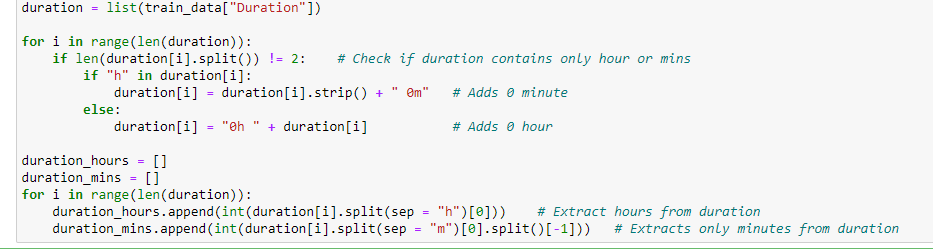




**Time taken by plane to reach the destination is called Duration**

**It is the difference between Departure Time and Arrival time**

**Assigning and converting the Duration column into a list**

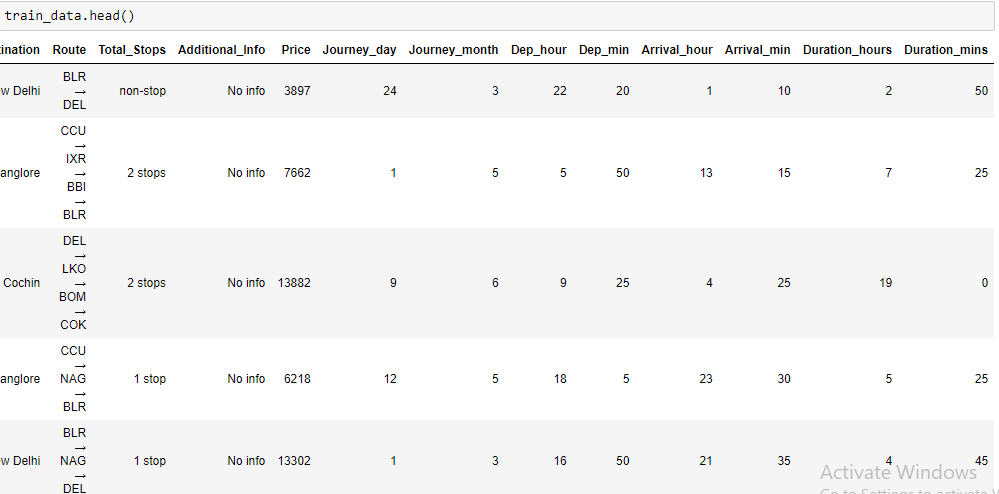


**Adding duration hours and duration\_mins list to train data frame**



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



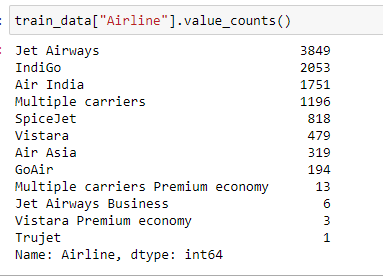


# **Handling Categorical Data:**

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

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

2 **Ordinal data** --> data are in order --> **Label Encoder** is used in this case

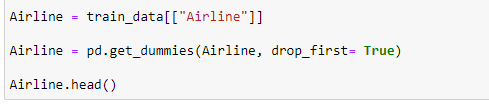


**Airline vs Price:**

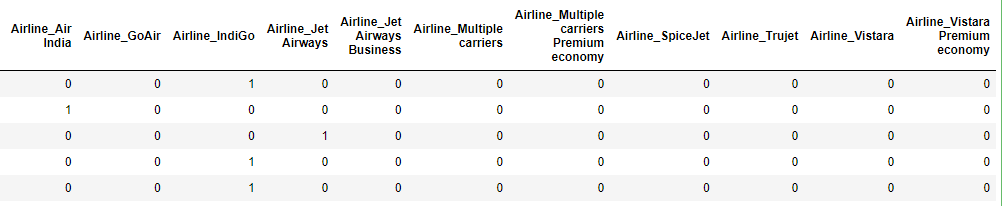


**From the graph, we can see that Jet Airways Business has the highest Price. Apart from the first Airline, almost all are having a similar median.**

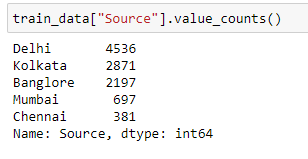
**As Airline is Nominal Categorical data we will perform One Hot Encoding:**



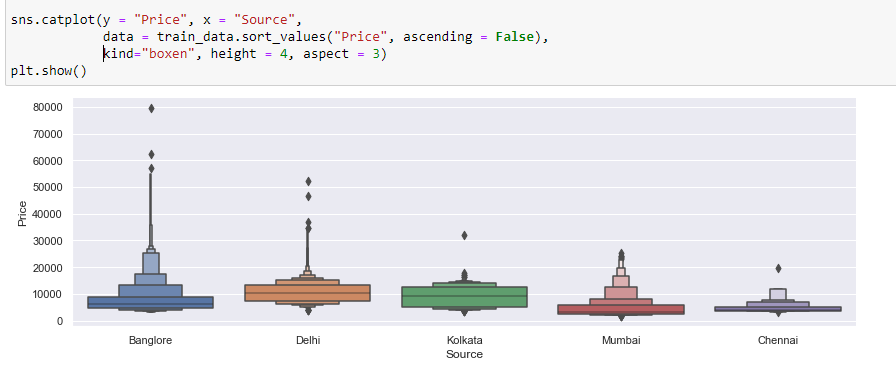
**OUTPUT:**



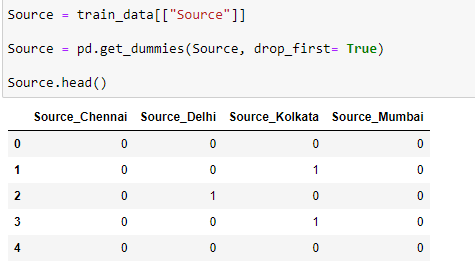
**Treating Source column:**



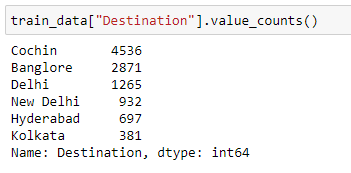
**Source vs Price:**



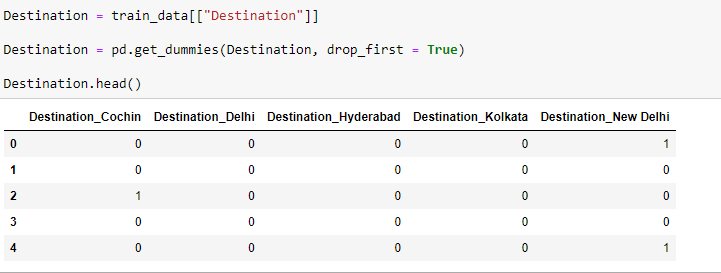
**As Source is Nominal Categorical data we will perform OneHotEncoding:**



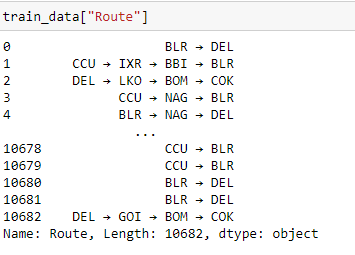
**Treating Destination column:**



**As Destination is Nominal Categorical data we will perform OneHotEncoding:**

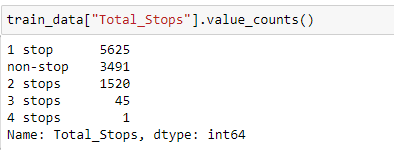


**Treating Route,** **Additional\_Info,** **Total\_Stopscolumn:**



**Additional\_Info contains almost 80% no\_info.Route and Total\_Stops are related to each other.**

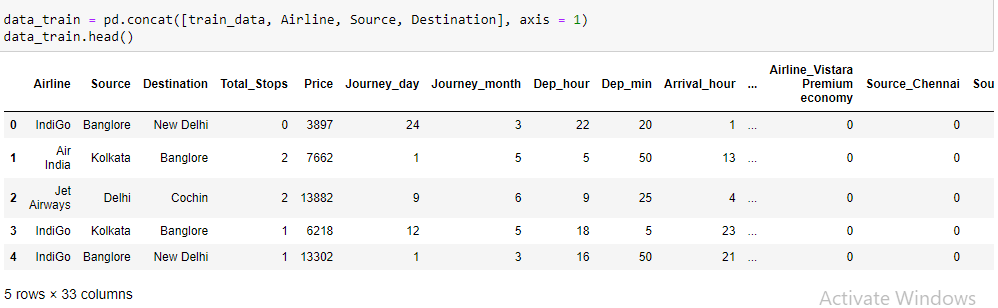




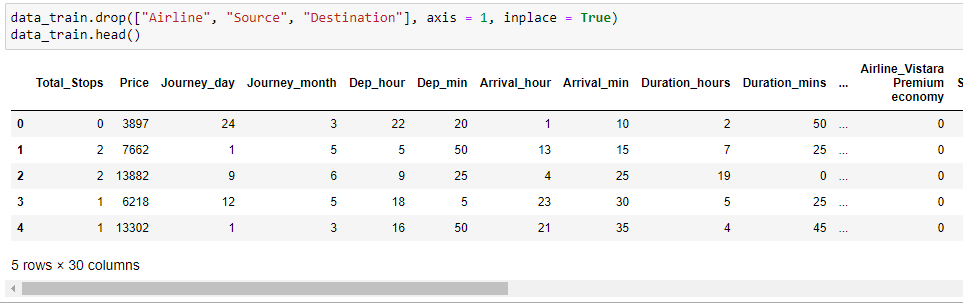
**As this is a case of Ordinal Categorical type we perform Label Encoder. Here Values are assigned with corresponding keys**



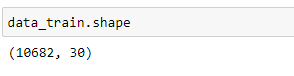
**Concatenate data frame --> train\_data + Airline + Source + Destination:**

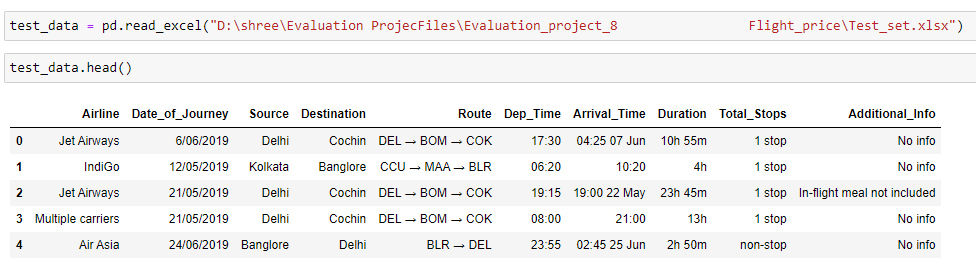


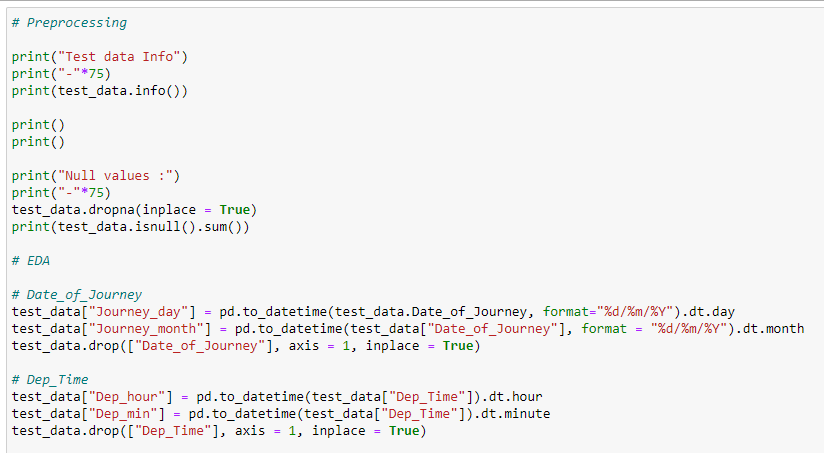
**Since we have converted the Airline,** **Source,** **Destination column into integers, now we can drop it as it is of no use.**

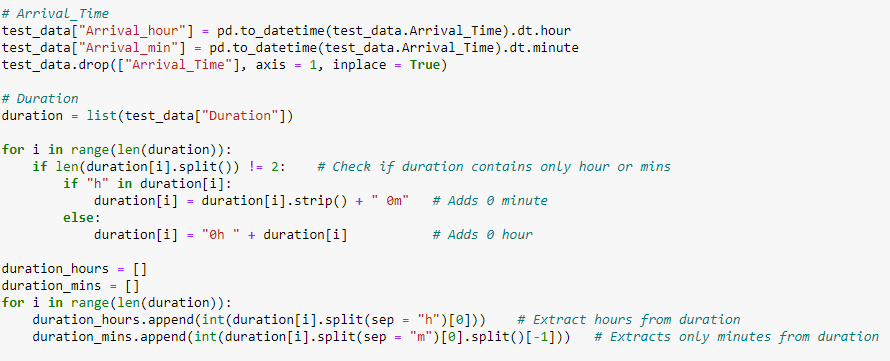


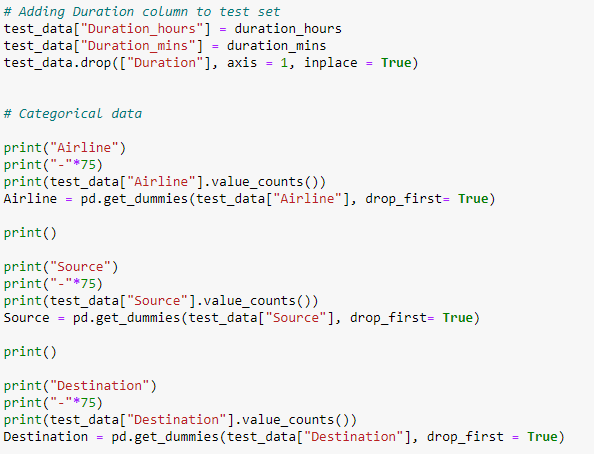
**Final Dataset Created with shape**

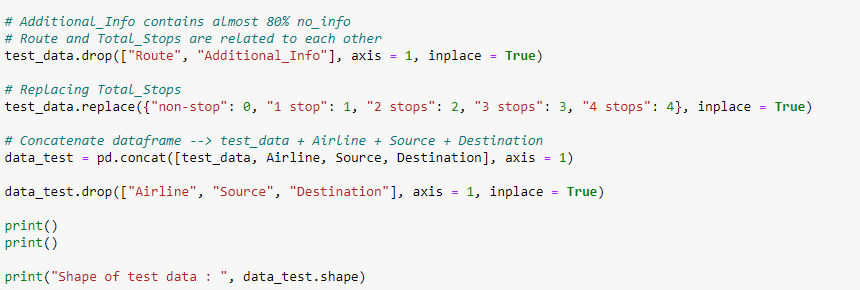


**Now calling the test dataset and applying the same procedure of columns as done for train data.**

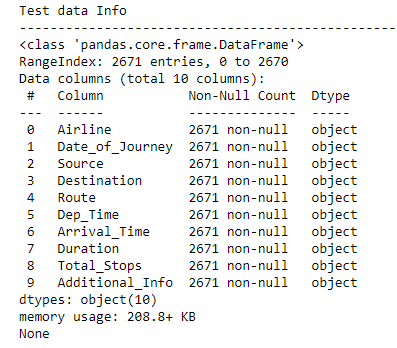


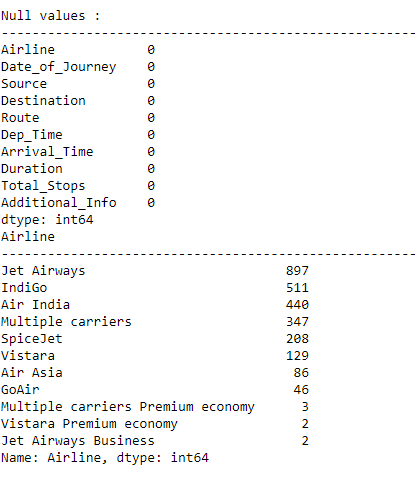


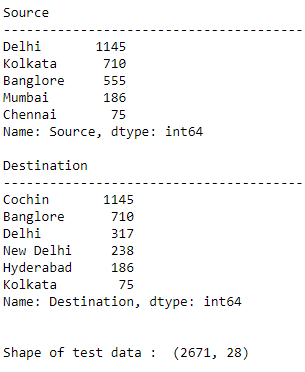


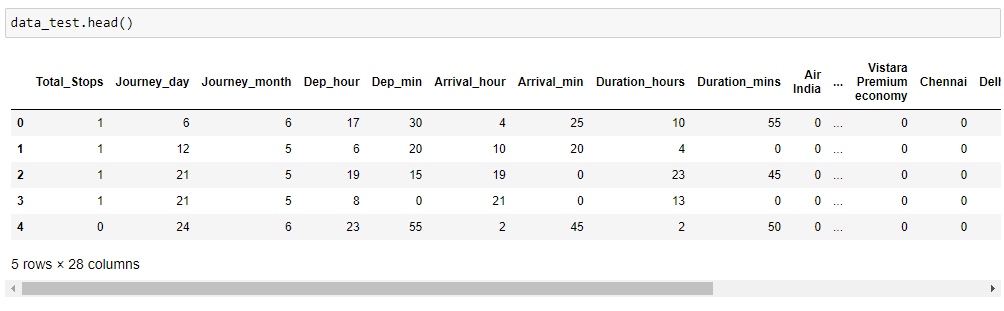


**OUTPUT:**





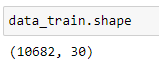




# Feature Selection

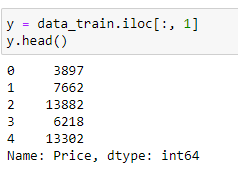
Finding out the best feature which will contribute and have good relation with the target variable. Following are some of the feature selection methods,

heatmap

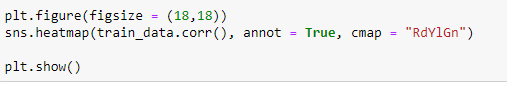






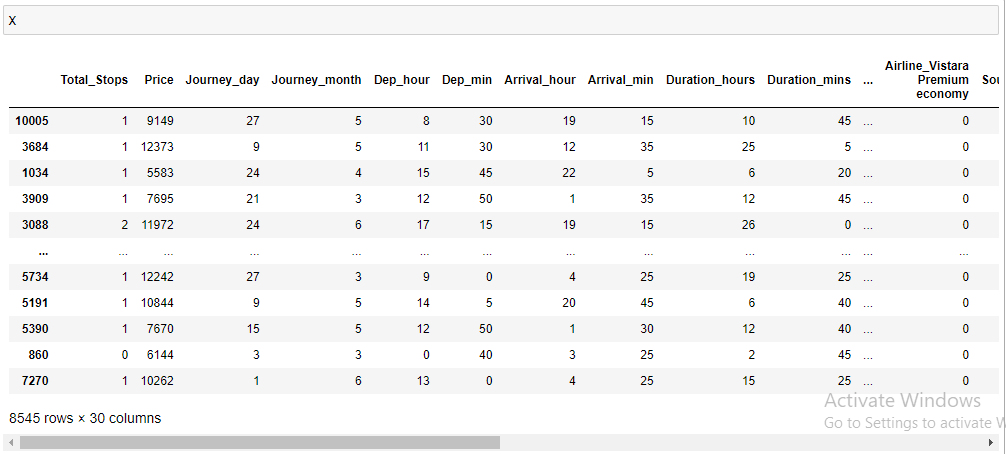


**Finds correlation between Independent and dependent attributes:**

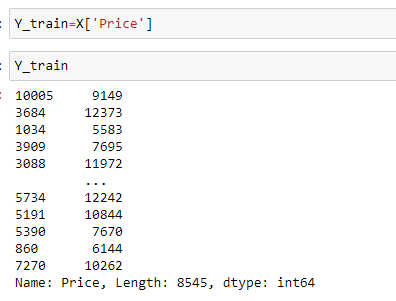


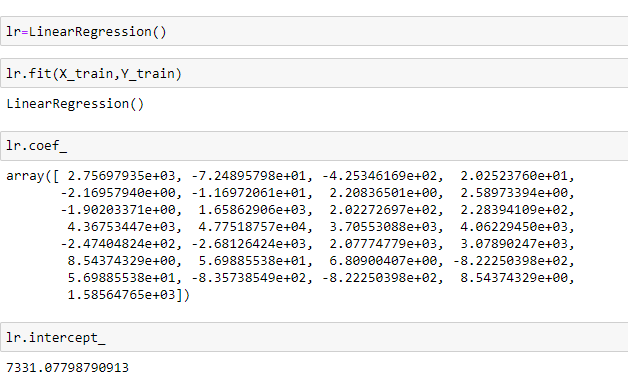
# Linear Model:

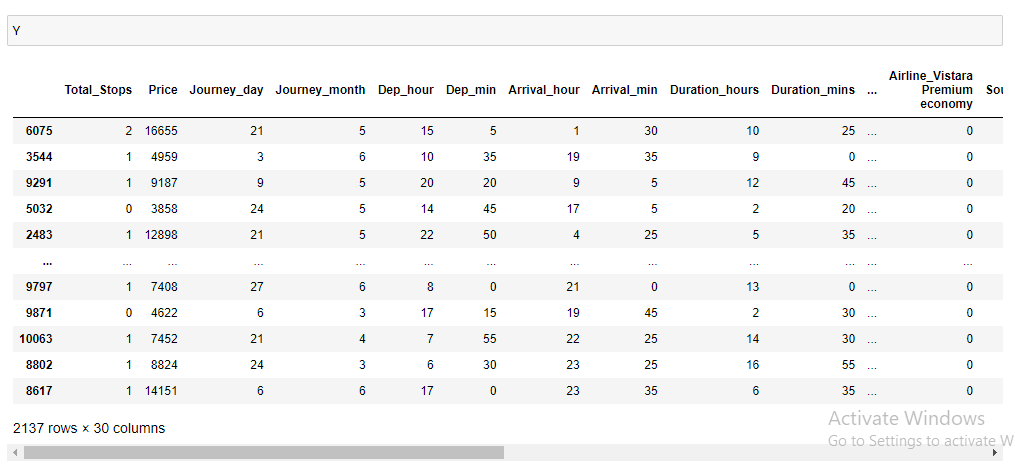






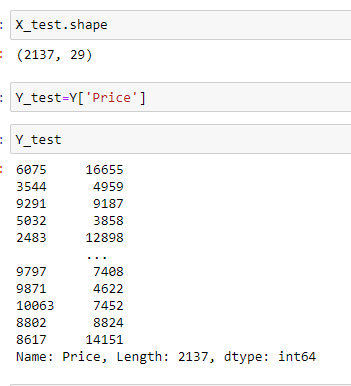


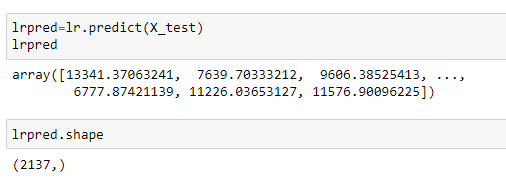




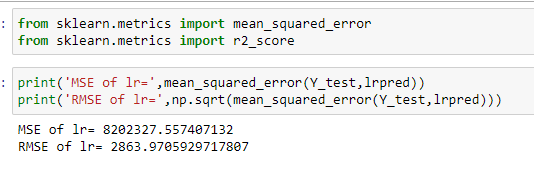


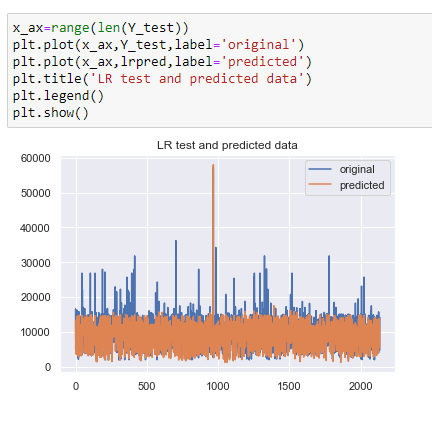






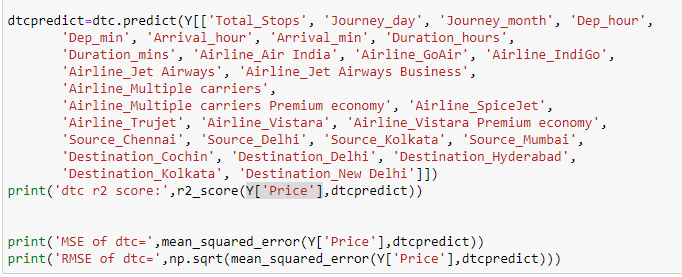
# Looking for patterns in the residuals



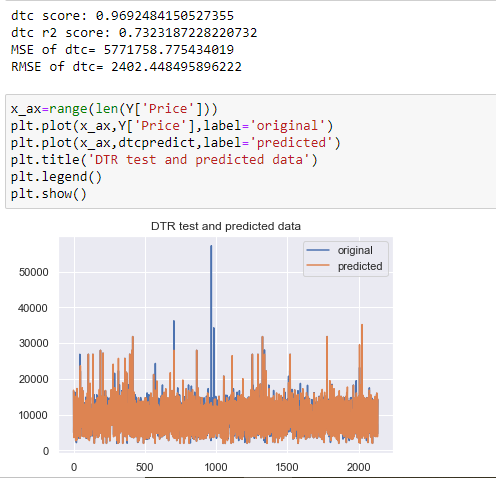


# Approaching more Repressors:

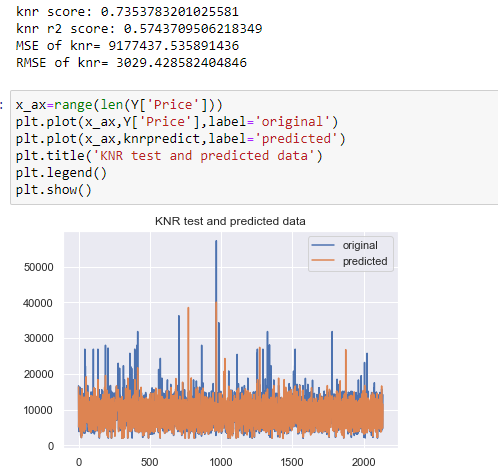




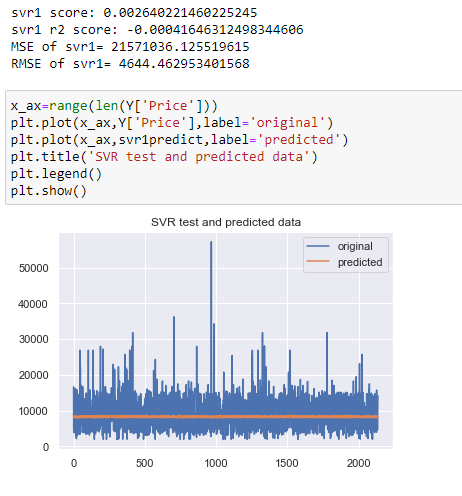
**OUTPUT:**



# KNeighborsRegressor:



# SVR:

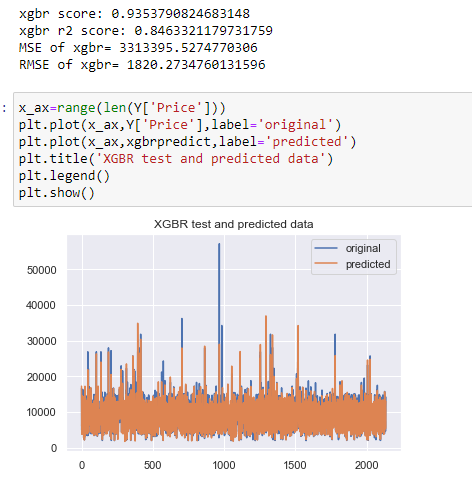


**Achieved the best, using DecisionTreeRegressor with minimum MSE and RMSE of 5771758.775434019 and 2402.448495896222 respectively.**

# Now, we will try to use Ensemble Models to check if our performance improves using ensemble models.

Models:

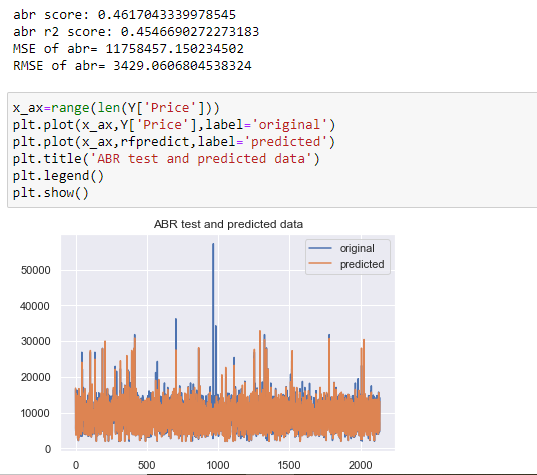
**XGBRegressor**



# Random Forest:



# AdaBoostRegressor



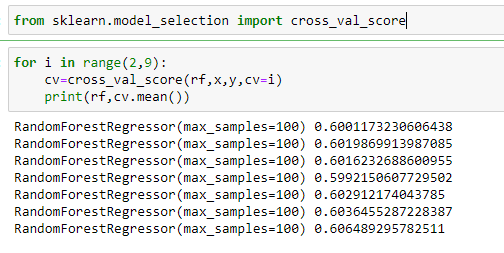
XGBRegressor gives the best accuracy of an RMSE of **1820.2734760131596.**

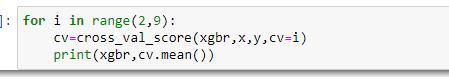
**On the other hand, Random Forest gives the second minimum value for RMSE**

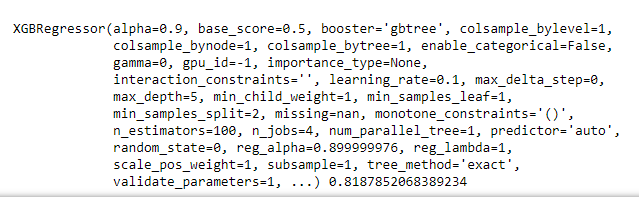
**(2093.0633177936515) which is less than Decision Tree RMSE**

**(2402.448495896222).**

# Cross Validation:

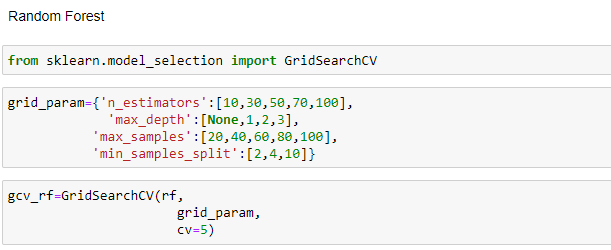


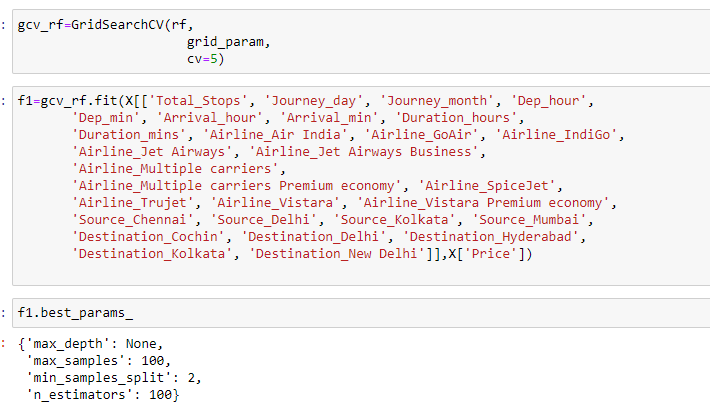


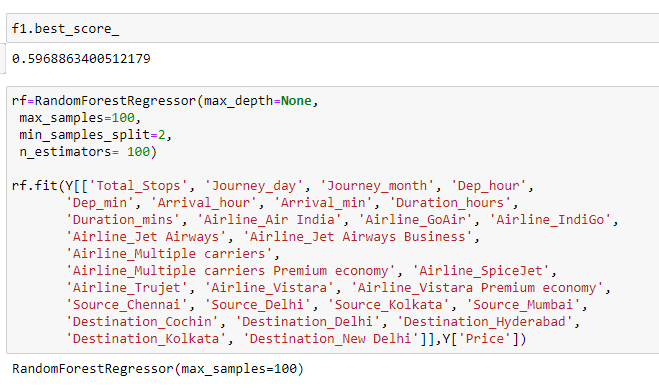


# HYPERTUNING THE MODEL

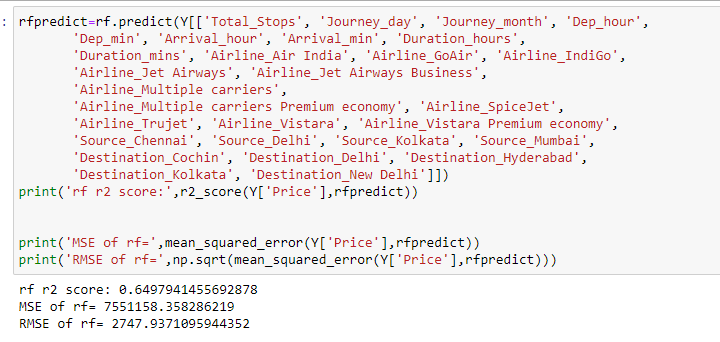
Random Forest



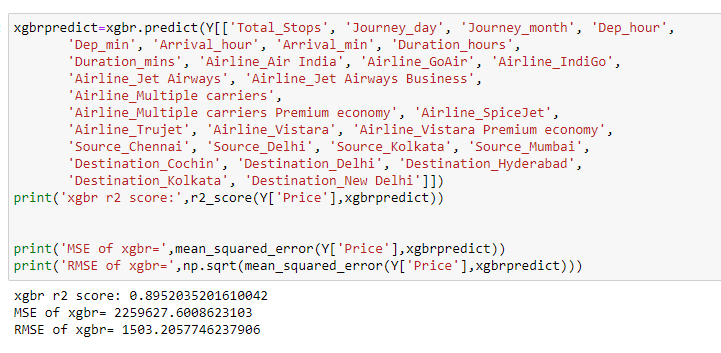






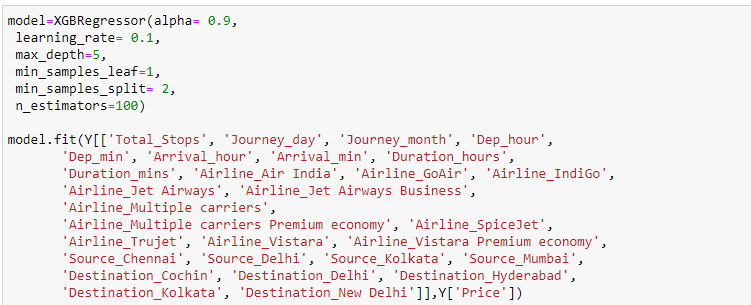


XGB Regressor

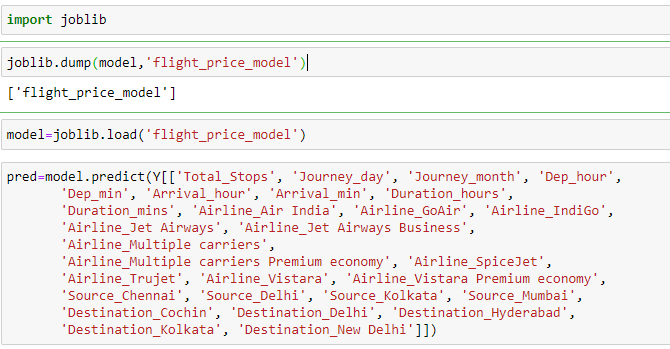


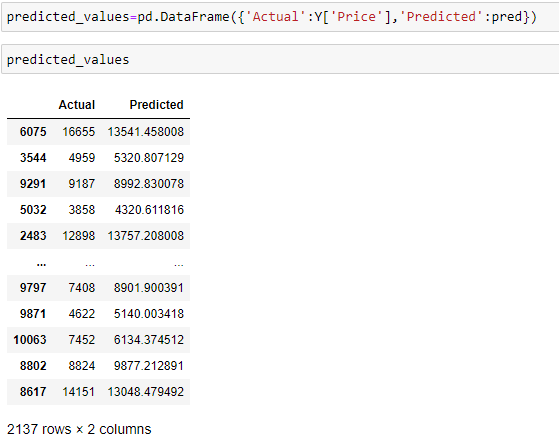
# The RMSE receive for XGBRegressor comes out to be better after hyper tuning.

Hence we select XGBRegressor as our final model.



**To load and predict the values:**





**These are the predictions on the training data. We can use the model to predict price value for given independent variables.**

**THANK YOU**