

FLIGHT PRICE PREDICTION

**By Ankit soran**

**1/29/2022**

INTRODUCTION:

Flight prices are something unpredictable. It’s more than likely that we spent hours on the internet researching flight deals, trying to figure an airfare pricing system that seems completely random every day. Flight price appears to fluctuate without reason and longer flights aren’t always more expensive than shorter ones.

But now the question is how to know proper Flight price, for that I have built a Machine learning model which can predict the Flight price. Using various features like Airline, Source, Destination, Arrival time, Departure time and the Price for the same travel. So using all these previously known information and analysing the data I have achieved a good model that has 85% accuracy. So let’s understand what all the steps we did to reach this good accuracy

Libraries used:

* Python
* Numpy
* Matplotlib
* Seaborn

Now let’s get into the problem and build a best possible model to predict Flight price. In this particular problem we are going to deal with two sets of data one is train and other is test. Let’s have a look.

1. 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, and 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.

Size of training set: 10683 records

Size of test set: 2671 records

Attribution Information:

Airline: The name of 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.

Departure 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.

In this particular problem we have two datasets so we have to build model using train dataset and save the best model, after that using the best model saved we have to predict the price for test dataset.

Let’s do it step by step firstly analysing the dataset and doing exploratory data analysis, data visualization, data cleaning, pre-processing, model building, finally predictions and saving model.

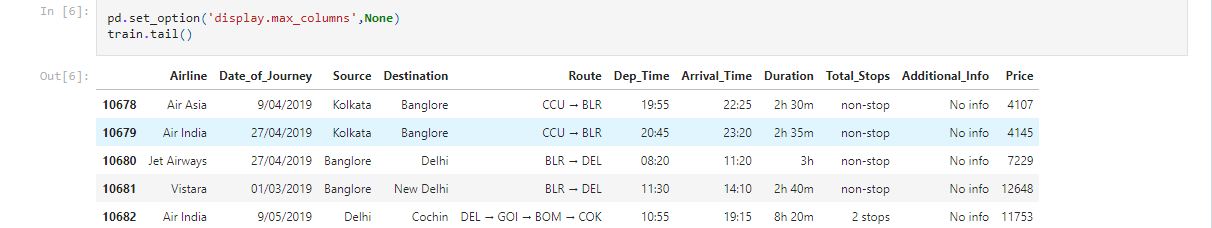
2. Data Analysis:

Let’s import the datasets first. I have two datasets one is train dataset and other is test dataset. I can merge both the sets, but I haven’t done that. Because this merging may cause data leakage so to avoid that we have to clean both the datasets separately with same steps did for first dataset. Now looking into the target ‘Price’ and I have to make sure the data type of target column to decide the type of problem.

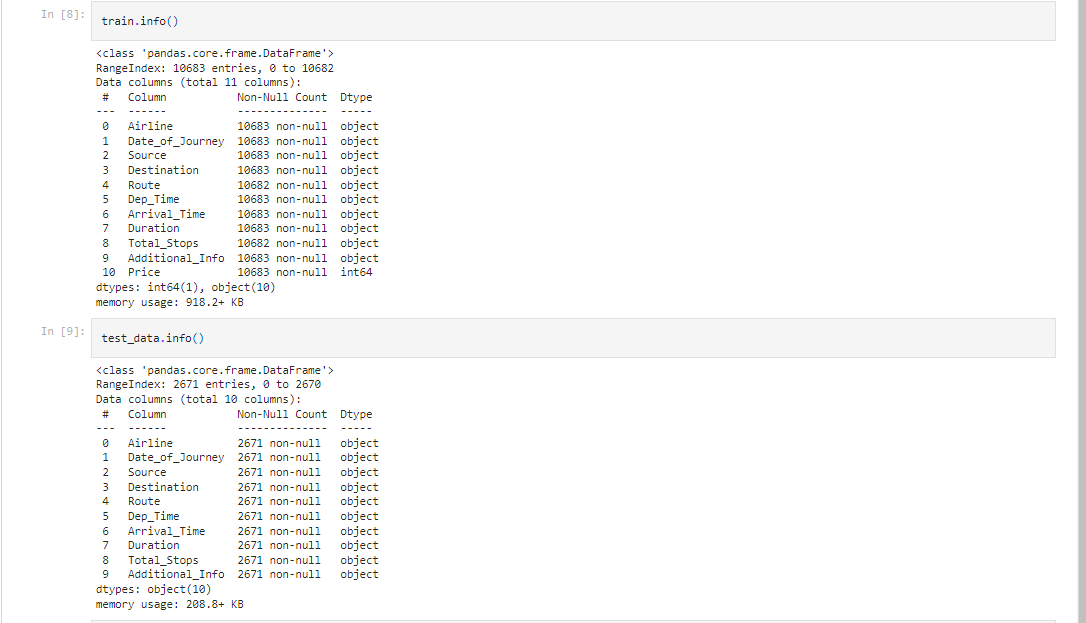
Since Price is my target and it is a numerical column with continuous entries. So it looks quite clear that this particular problem is a Regression problem and I have to use all regression algorithms while building the model.



· Firstly I have imported both datasets which were in excel format as train for train dataset and test\_data for test dataset. Below is the train datasets.



* Both datasets have same features except the target.So now we have to clean this data.



**Data Preparation and cleaning**:

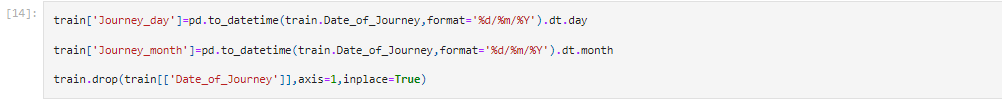
* Firstly we have to do some statistical analysis like describe, info etc…..
* There are very less null values in dataset.
* We found that both the missing values belong to the same record. So, we can drop the record to make the dataset free of null values.

**Pre-Processing Pipeline:**

* We now perform feature engineering on the features in the dataset. As we saw the features “Date of journey”, ”Departure time” and “Arrival time” contain date and time in the data, we need to treat these columns and extract the info from them.

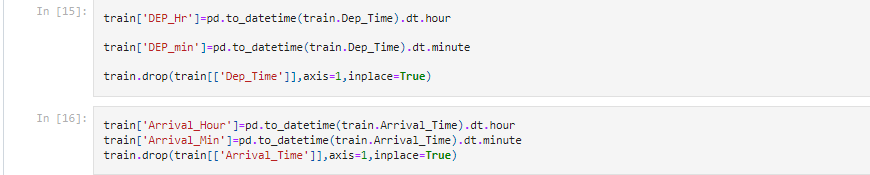
Date of journey:

* We first convert the object data type of the column to date and time.
* Then we extract the day and month from the data and store it in separate columns.
* And then we drop the base column, as we no longer need



Departure Time:

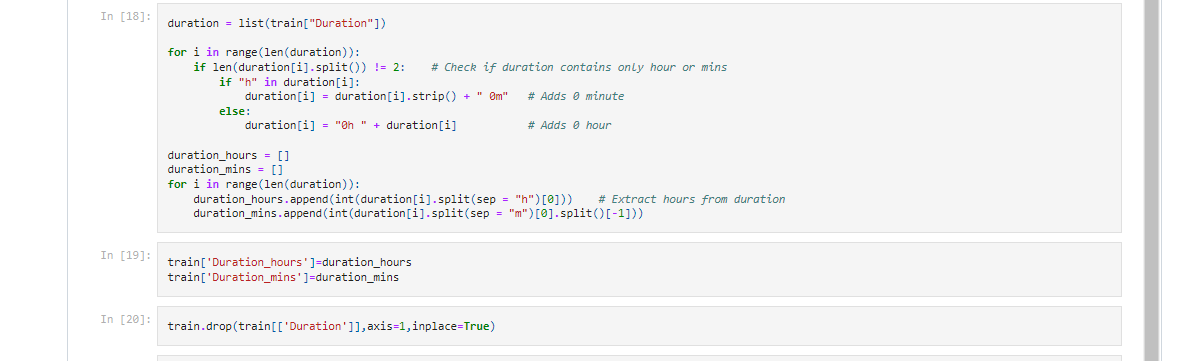
* First we convert the object data type to date and time.
* Then we extract the hour and minute from the column and store it in separate columns.
* And then we drop the base column, as we no longer need it.
* Same is followed for the Arrival time



Duration:

The column contains the duration of the flight in, but is contains string in it. So we try a different approach here.

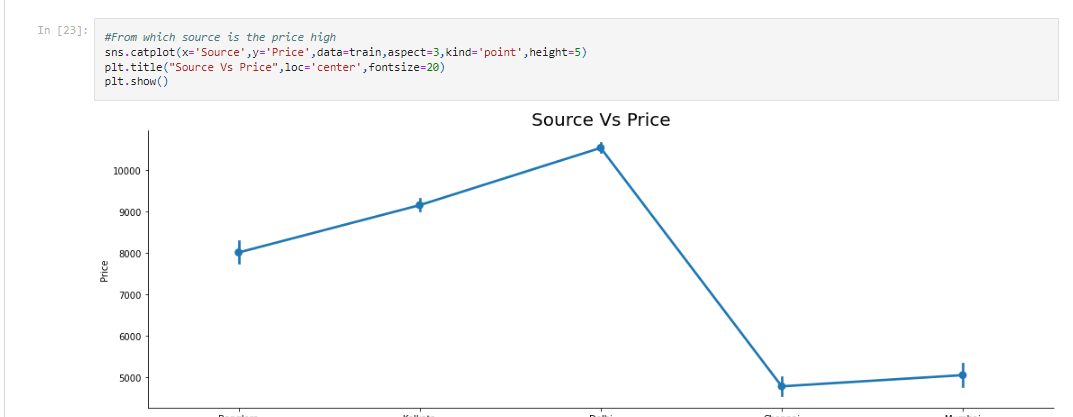
* We first create a list and store the column data in it.
* Then we add ‘0 m’ or ‘0h’ in the records that contain only the hour or the minute respectively.
* We then add the list to the data frame.
* After this we split the data in the column, keeping the hour and minute without any string in different columns.
* We drop all the unwanted columns created in this process along with the base column.



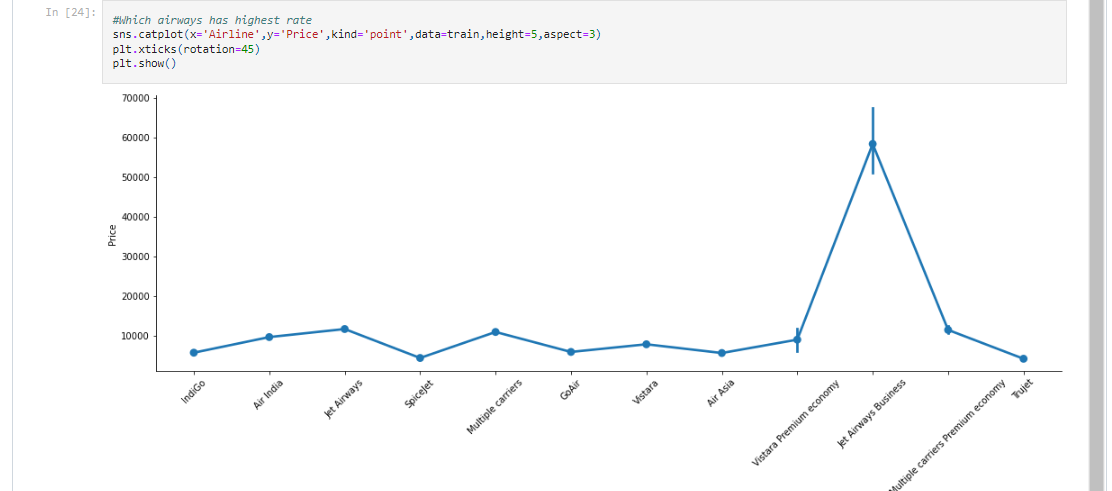
**Visualization:**

After the feature engineering, we visualized few features which may can affect our model, including our target feature.

****

****

I have plotted catplot for destination, source and airline since Price is my target column .

****

# As route is the path b/w source and destination and if there is a stop b/w source and destination, it is seen in total stops columns.

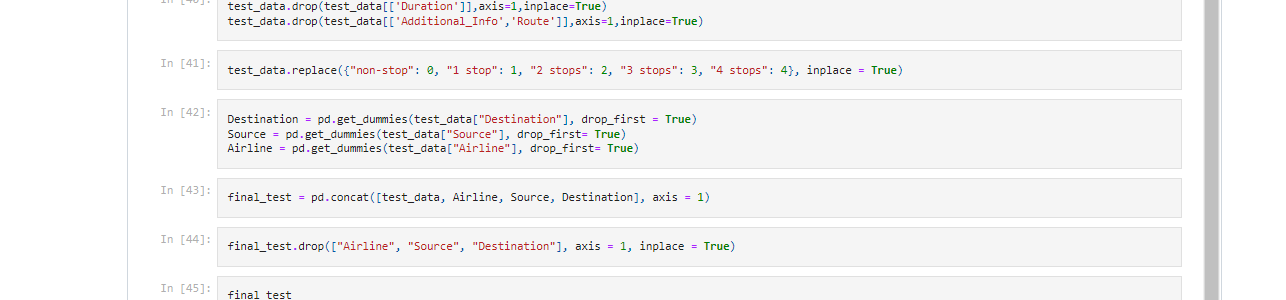
So we can drop the Route column. Column Additional Info column has more than 80% of columns with 'No Info' which potentially is a NA value in terms.so we can drop them.

Next looking at Total stops column there are specific entries like non-stop, 1-stop, 2-stop, 3-stop, 4-stop so I can use particular labels for this column like 0, 1,2,3,4 respectively to avoid misconception on the analysis.



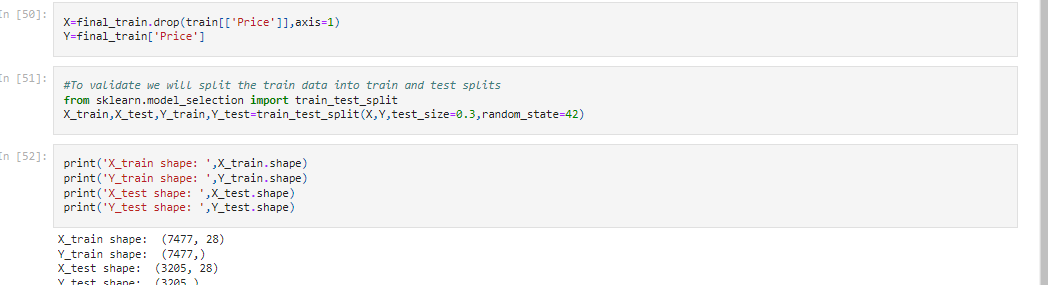
Afterwards doing same feature engineering for Test dataset as train data set we move forward.

After this, we perform encoding on the dataset as there were still some columns left with object data types. The columns left were categorical in nature. For the columns having nominal data we performed one-hot encoding as the number of categories were also less.



**Building Machine Learning models:**

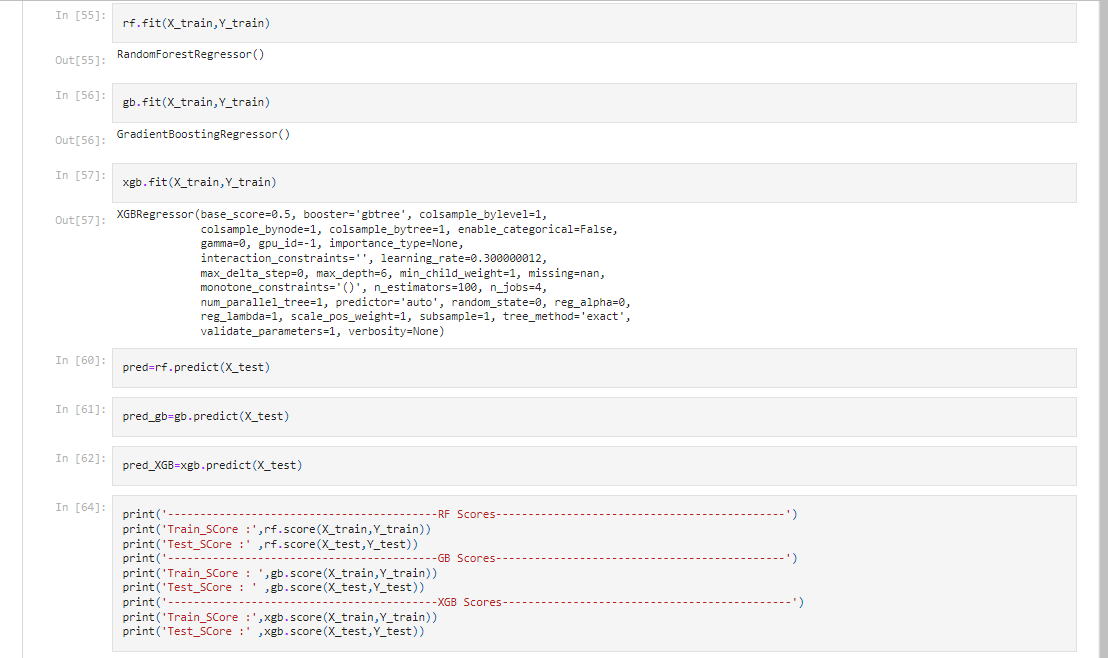
At first we split the train data into target and features.

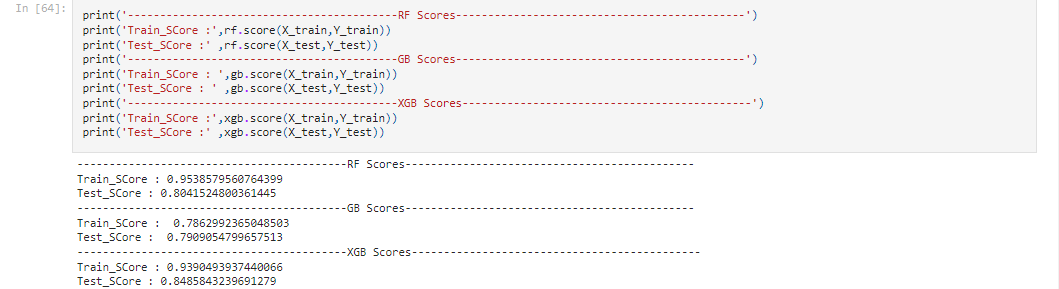


Checking for the accuracy of Train and Test datasets with different models.

We send the x-validation split to the models for prediction.

* RandomForestRegressor
* GradientBoostingRegressor
* XGBRegressor

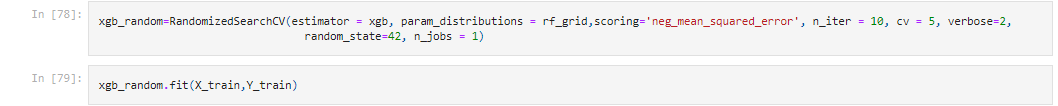
****

****

* **For the** Random Forest Regressor we get the accuracy of 95% for train data and 80% for test data.
* **Using** Gradient Boosting Regressor we get the accuracy of 78% for train data and 79% for test data.
* **And for** XGB Regressor we can see the accuracy of 93% for train data and 84% for test data.

We found the **Random** **forest** **model** and the **XGB** model giving the best score among all the models.

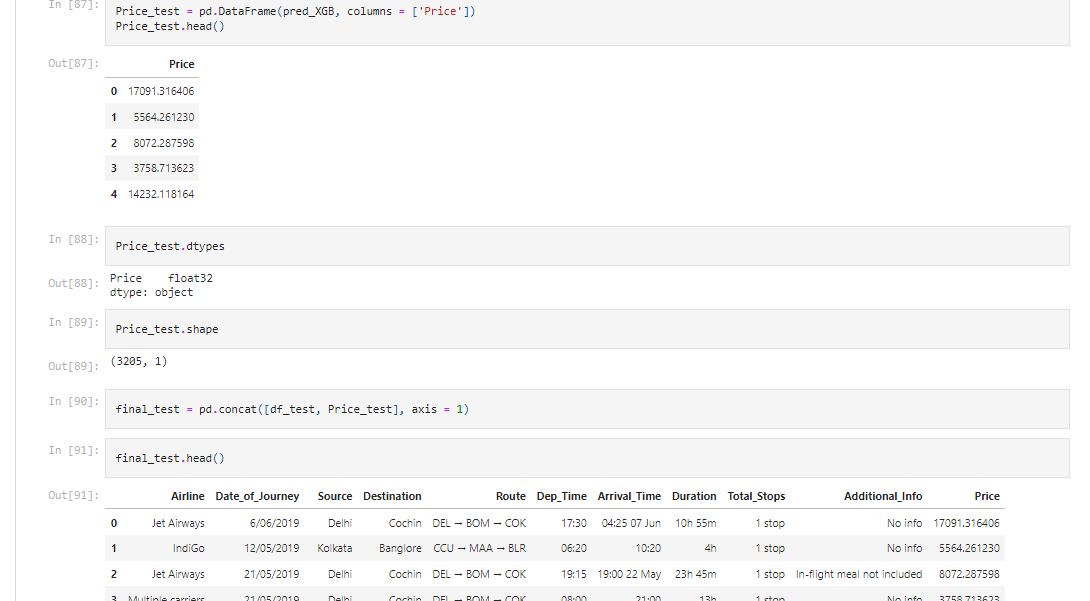
**Hyper parameter tuning for better result:**

****

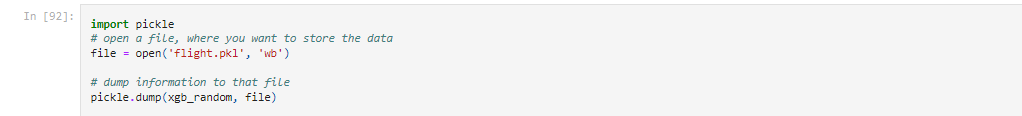
By analysing both the models, we found XGBoost model to be performing slightly better. Hence we made the XGBoost model our final model and saved the model.

**Predictions**:

· Now using the model and cleaned test dataset I have to predict price for test dataset using best model of train dataset.



**Saving Model:**

****

**Concluding Remarks:**

* This particular problem needs a good vision on data, and in this problem Feature Engineering is the most crucial thing.
* You can see how we have handled numerical and categorical data and also how we build different machine learning models on the same dataset.
* Using hyper parameter tuning we can improve our model accuracy, for instance in this model the accuracy remained same.
* Using this machine Learning Model we passengers can easily predict the flight price and we could also save some amount while booking the Flight smartly.

Thank you