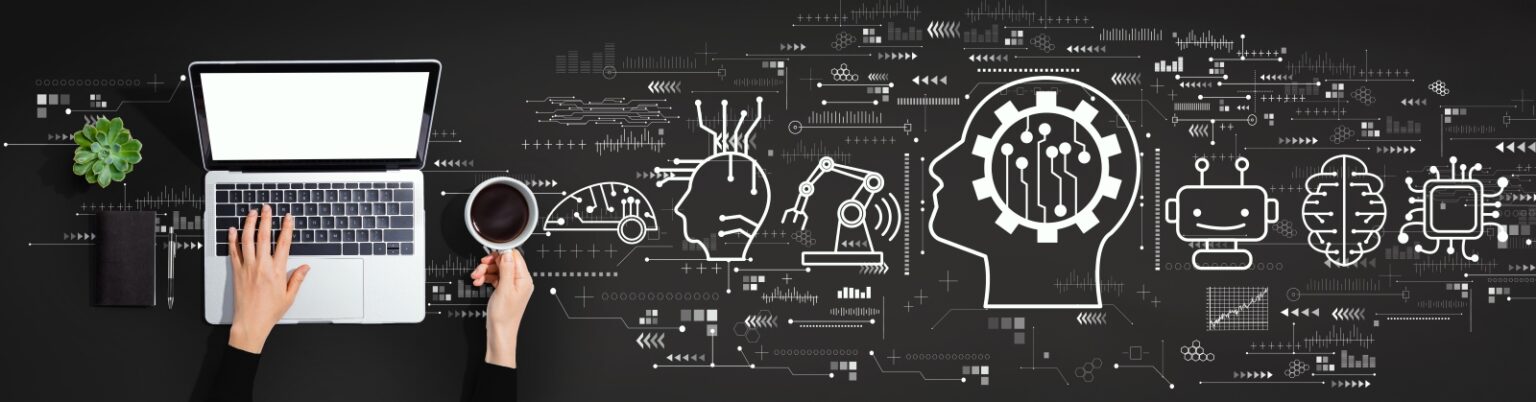
Flight Price Prediction : Machine Learning Project

Machine Learning



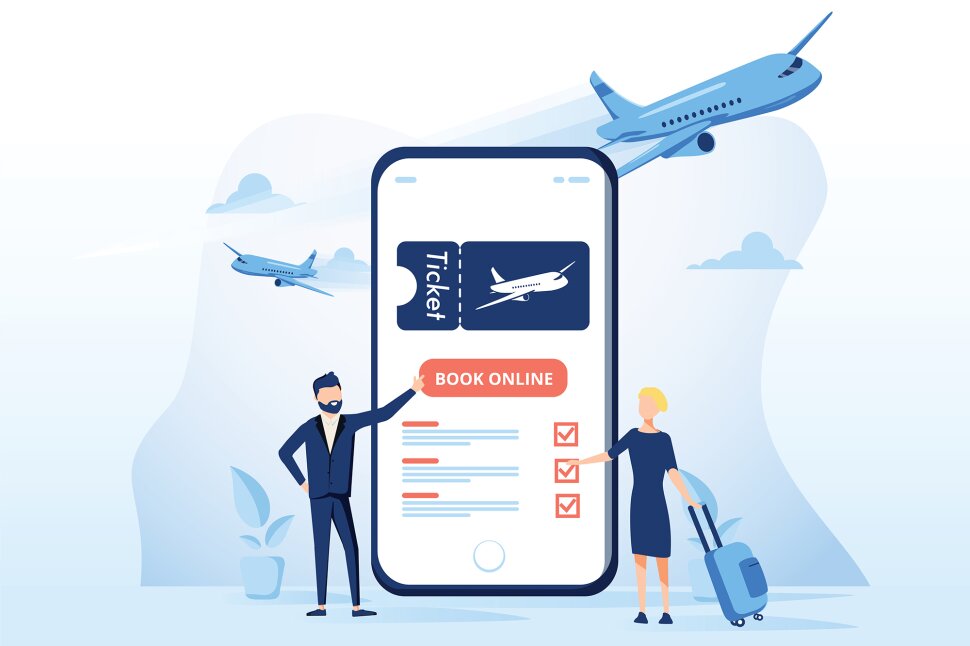
Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves.

Machine learning (ML) is the study of computer algorithms, which improve with experience and use of data. Machine learning algorithms build a model based on sample data (training data), and make predictions or decisions using this model without being programmed to do so.

Machine learning algorithms have a wide variety of applications, like fraud detections, email filtering etc. One such application of machine learning lies in the ‘Aviation industry’, to predict the prices of flights. There are various factors/features which impact the prices of flights — distance, flight time, number of stops etc. These factors help create a pattern to decide the price of a 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.

Regression —

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, and it will be a different story.

To solve this problem, we have been provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, using which we aim to build a model which predicts the prices of the flights using various input features. input features.

The Dataset

* Link for the dataset — <https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects>

We have 2 datasets here — training set and test set.

The training set contains the features, along with the prices of the flights. It contains 10683 records, 10 input features and 1 output column — ‘Price’.

The test set contains 2671 records and 10 input features. The output ‘Price’ column needs to be predicted in this set. We will use Regression techniques here, since the predicted output will be a continuous value.

Following is the description of features available in the dataset –

1. **Airline**: The name of the airline.

2. **Date\_of\_Journey**: The date of the journey

3. **Source**: The source from which the service begins.

4. **Destination**: The destination where the service ends.

5. **Route**: The route taken by the flight to reach the destination.

6. **Dep\_Time**: The time when the journey starts from the source.

7. **Arrival\_Time**: Time of arrival at the destination.

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

Contents of the article

This article explains the complete process to build a machine learning model. Below mentioned are the various phases that we will go through, throughout the project.

 1. Exploratory data analysis and Data modeling

2. Outlier detection and skewness treatment

3. Encoding the data — Ordinal Encoder

4. Scaling the data — Standard scaler

5. Fitting the machine learning models

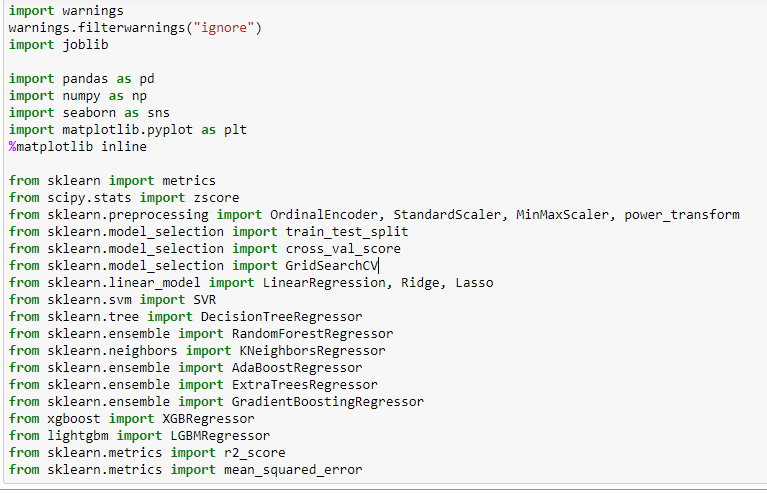
6. Cross-validation of the selected model

7. Model hyper-tuning

8. Saving the final model and prediction using saved model.

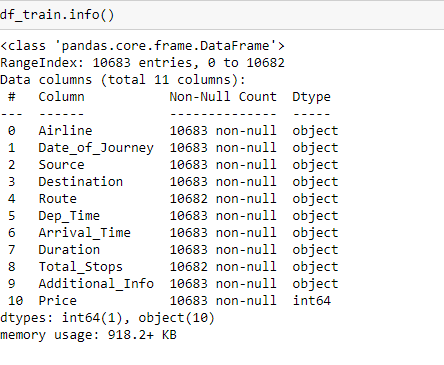
So let’s begin exploring our data set and start building a prediction model.

**Importing all the necessary Libraries :**

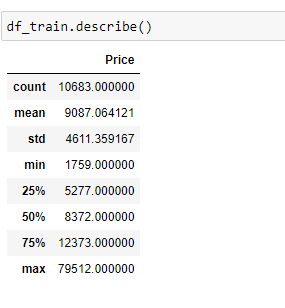
****

We have imported all the necessary dependencies which are required in our project. We are doing this project in Jupyter notebook.

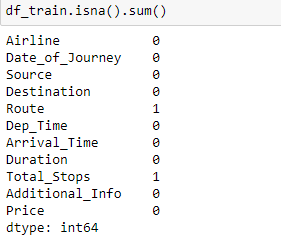
**Exploratory Data Analysis (EDA)**



Using the info method we can see that there is only 1 column with integer data type and 10 columns that have object data type.

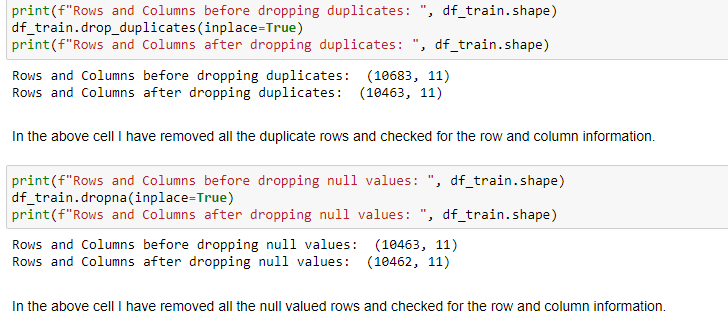


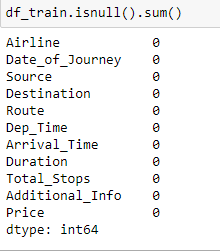
With the help of describe method we can see details for only numeric data type and since we have just 1 column with integer data type all the object data type columns got ignored. In the "Price" column we are able to take a look at the number of rows covered in our dataset being displayed in the count area, then we have the mean and standard deviation being reflected for our label column. Later on the values of minimum, 25% quartile, 50% quartile, 75% quartile and maximum number are being displayed above.



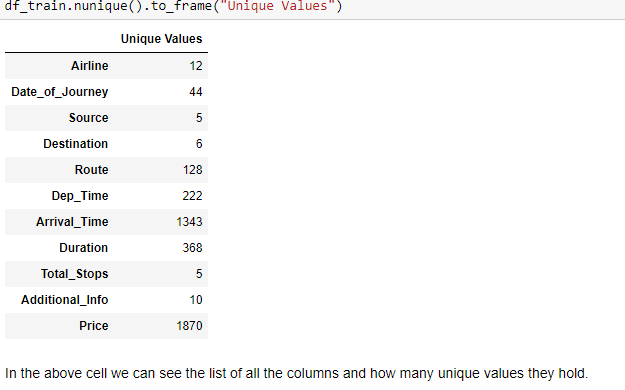
In the above cell we are taking a look at null values and there are 2 columns "Route" and "Total\_Stops" that have missing data.

**Removing the null values**



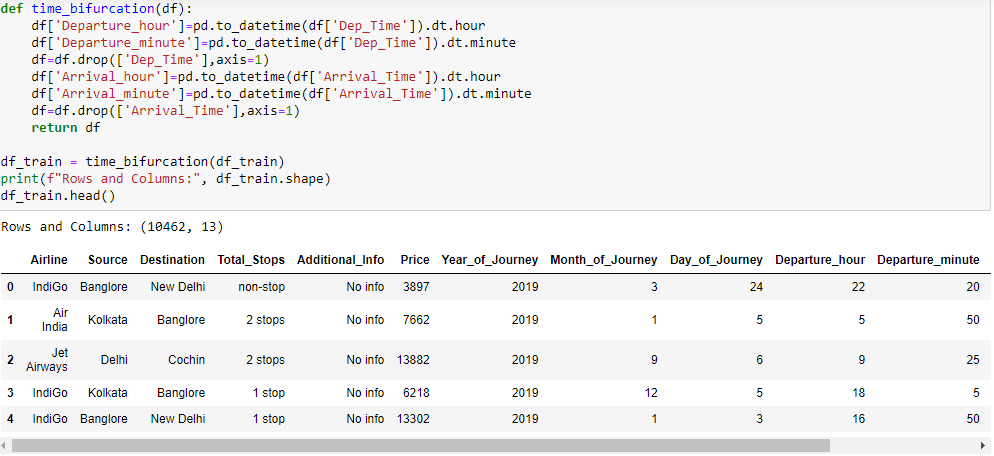


Great now we have no missing data present in our training dataset. Also we have taken care of duplicate rows that were present in our dataset.





I have created a function that basically separates the date information into proper numerical format instead of making it an object datatype. Also I have dropped the Date of Journey column since it's data was already bifurcated, then I removed the Route column as we have source and destination data so Route was not adding much of an insight and finally I removed the Duration column as later on I have time separations to deal with as well.

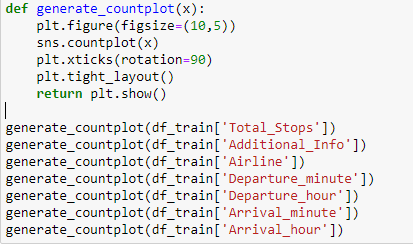


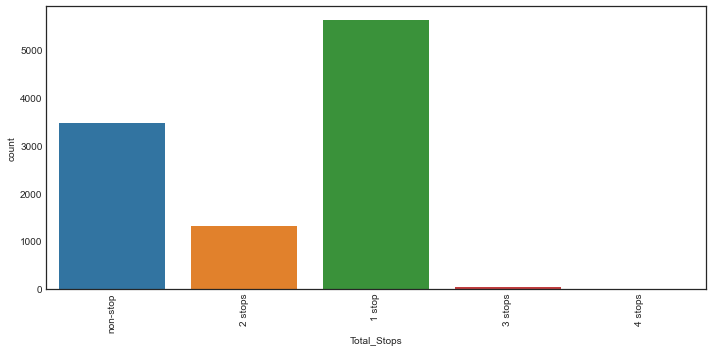
In the above cell I have created another function that deals with the separation on timings for arrival and departure. This allows me to get a proper insight on the flight duration details as well therefore I got rid of the duration column earlier.

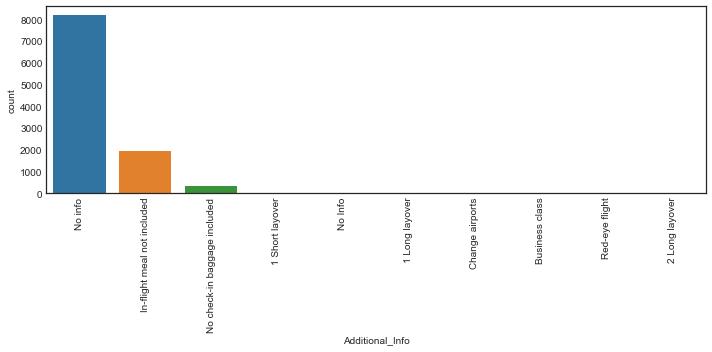


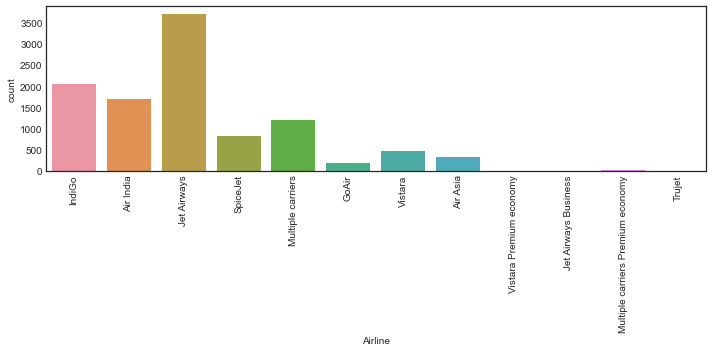
I am dropping the column named "Year\_of\_Journey" as it only has one uniqye value covering all the rows in the column and therefore it wont add much value to our prediction model later.

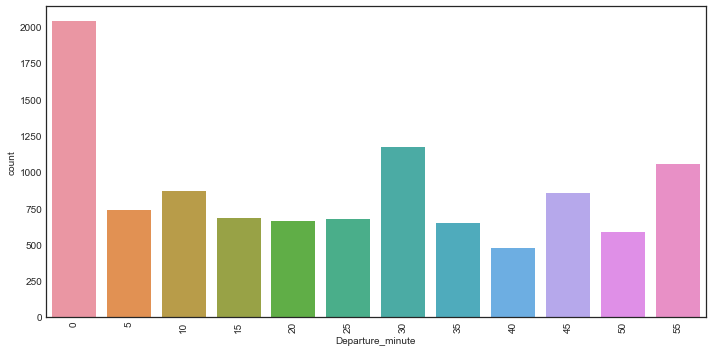
**Visualization**

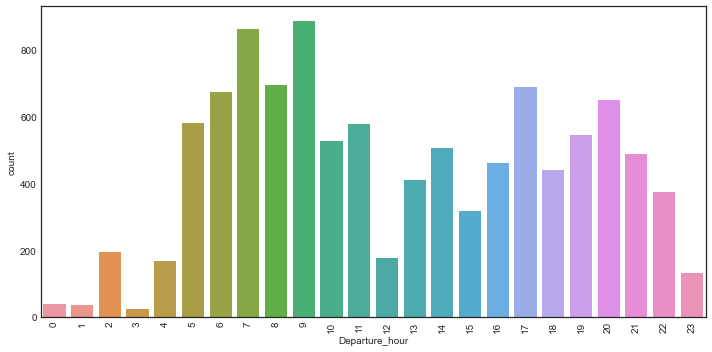


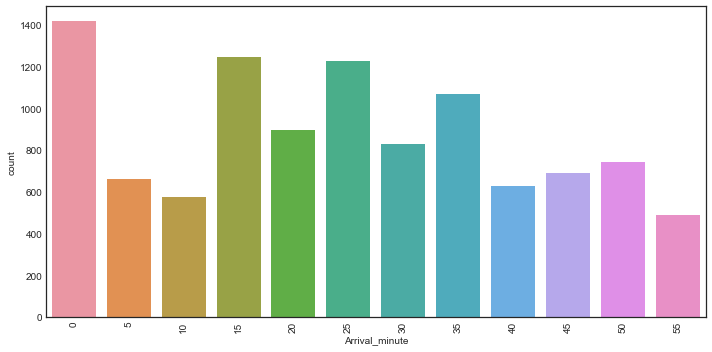


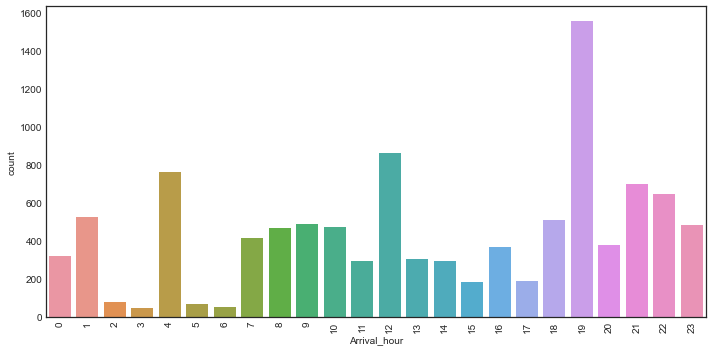


****



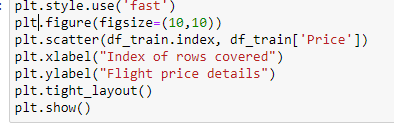


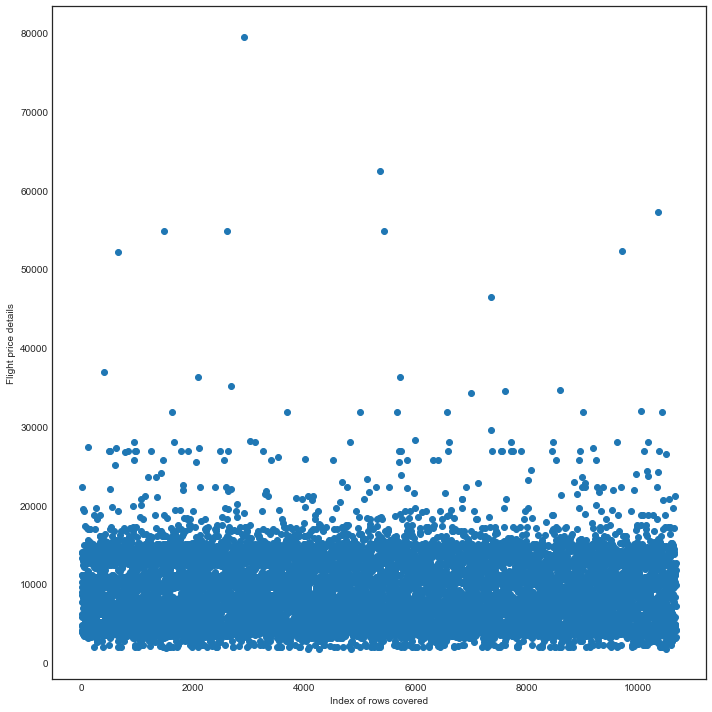




I have created a function to generate count plots for our feature columns. My observation for them are:

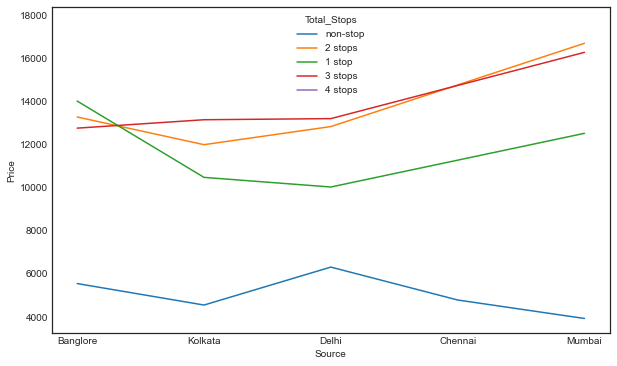
1. In the total stops column we see highest count of rows covered by 1 stop flight hauls and the least numbers are for 3 and 4 stops
2. In additional information column most number of rows are covered by no info values and rest of the values cover very less to negligible data points
3. The airline column shows that highest number of flight details are present in our dataset for Jet Airways followed by Indigo and Air India
4. The departure minute column gives us the indication that most number of flights get scheduled at 0 minutes for departure
5. The departure hour column gives us the indication that most number of flights get scheduled at 7 and 9 hour morning time and then there is a spike at 17 and 20 hour evening time
6. The arrival minute column gives us the indication that most number of flights get scheduled at 0 minutes for arrival
7. The arrival hour column gives us the indication that most number of flights get scheduled at 19 hour in the evening and then the chosen option for arrivals are 12 in the noon or 4 in the night



****

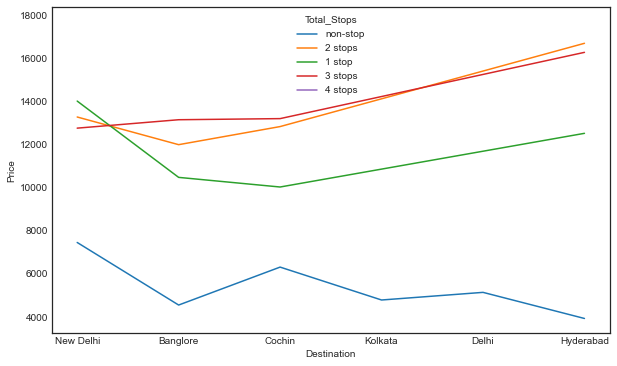
In the above scatter plot we are able to see that most of the flight price values are accumulated between 0-20000 and very rare data points are distributed above that number.

.



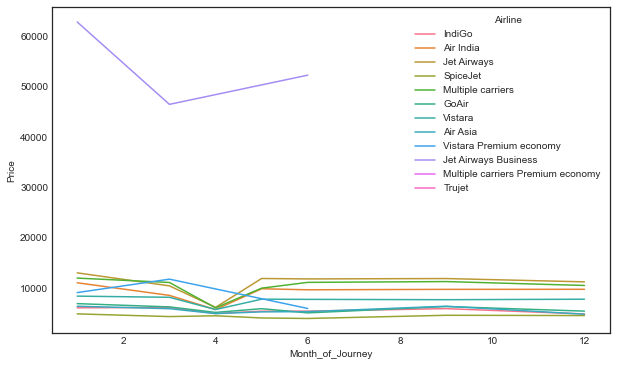
In the above line plot we see that non stop flights have lower price irrespective of the source as compared to flights that have 1 or more than 1 stops in the flight haul.





In the above line plot we see that non stop flights have lower price irrespective of the destination as compared to flights that have 1 or more than 1 stops in the flight haul.

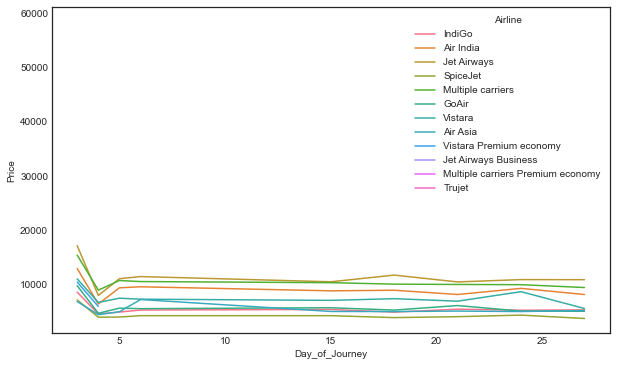




In the above line plot we see that Jet Airways Business class has the highest price than the rest possibly because the remaining offer the economy class data.

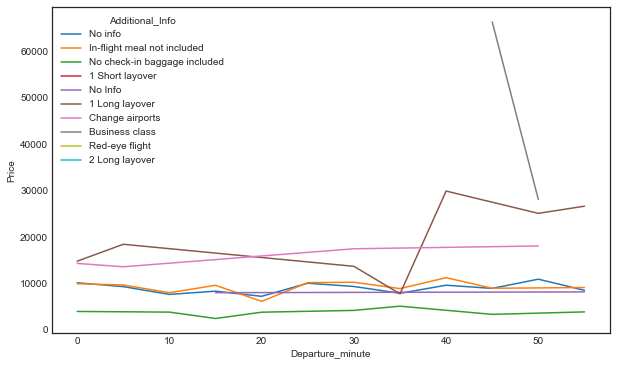
In [22]:





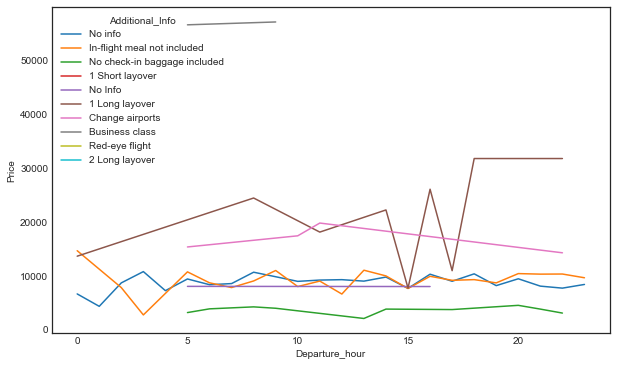
In the above line plot we see that all the airlines have high price between 1-5 days of a month and that reduces a bit on rest of the days apart from them.





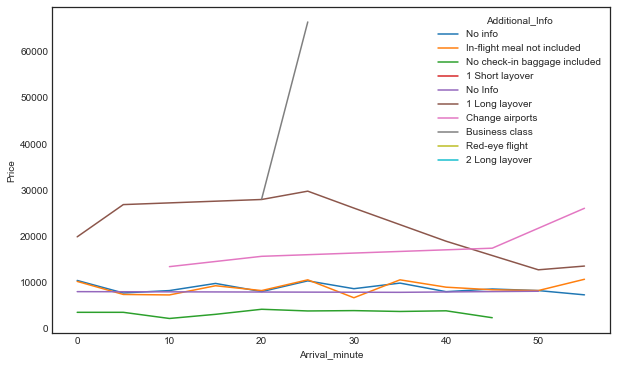
In the above line plot we see that Business class has high price and has data coverage for departure minutes between 45-50 minutes roughly.





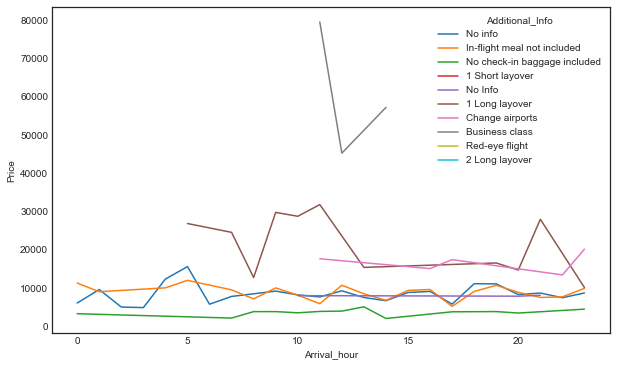
In the above line plot we see that business class has high price and it's departure hour is between 5-10 but the second highest pricing is for 1 long layover type with spike in between 17-22 departure hour



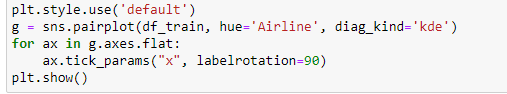


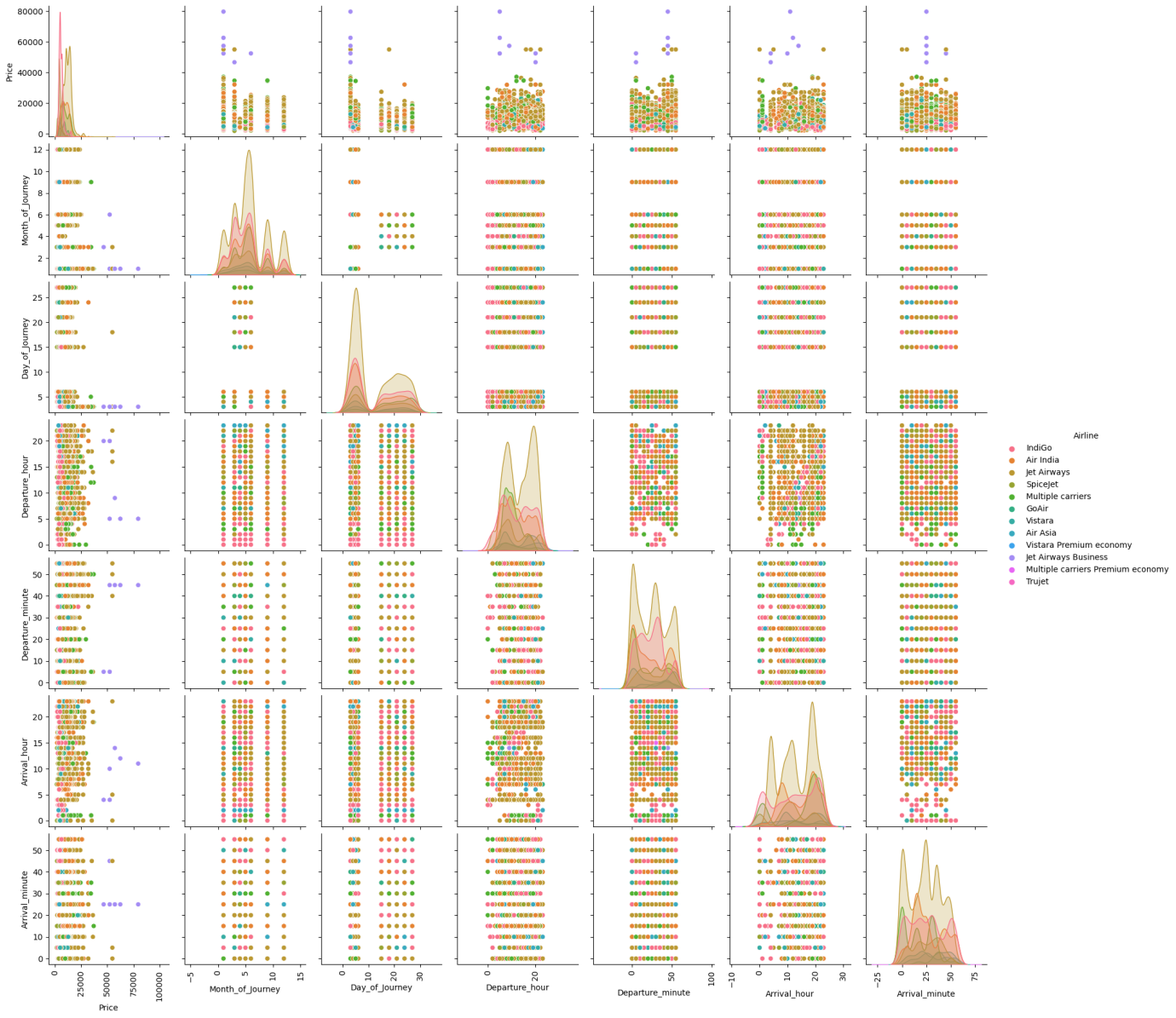
In the above line plot we see that business class again has an exponential price rise and the arrival minutes mostly range between 20-30 minutes





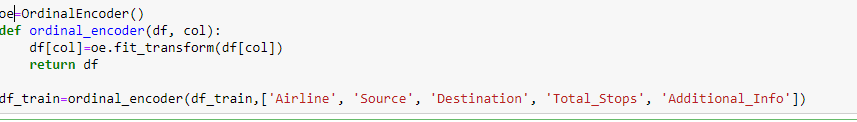
In the above line plot we see that price for no check-in bag included is least as compared to the business class being highest and the arrival hour for business class is spread only between 10-15 minutes.





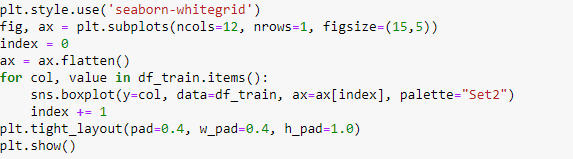
The above pairplot gives us an indication on the numerical data considering the different airlines present in our dataset.

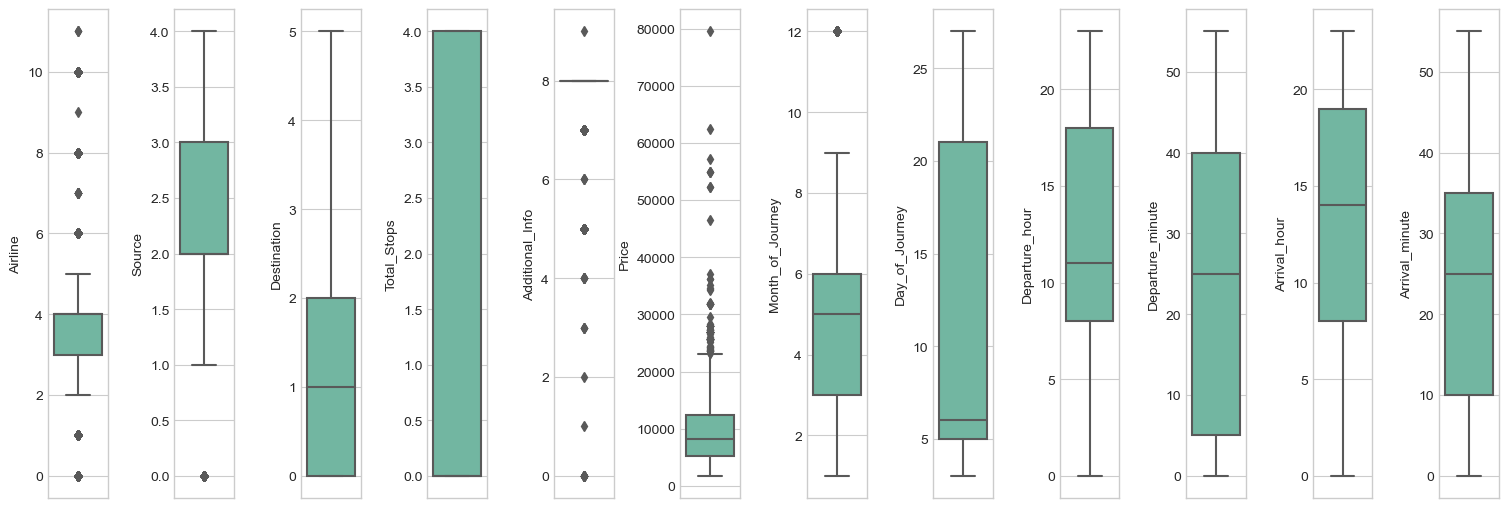
**Encoding the categorical object datatype columns**



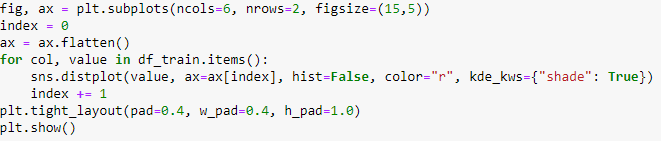
I am using the OrdinalEncoder method for encoding my categorical features since they all are present in an orderly format and will not increase the number of columns like in the usage of One Hot Encoding.

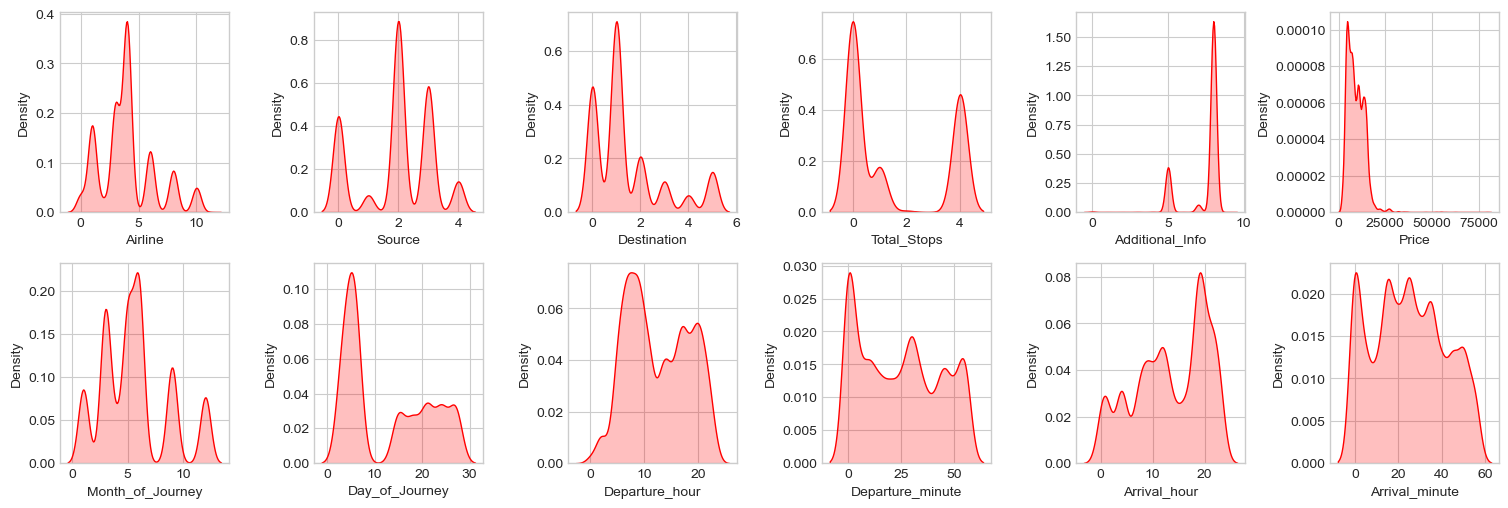
**Outlier detection and skewness treatment**

****

****

Using the box plot we are able to notice the outliers present in our dataset but since all of the feature columns are categorical data we will not have to worry about the presence of outliers here

****

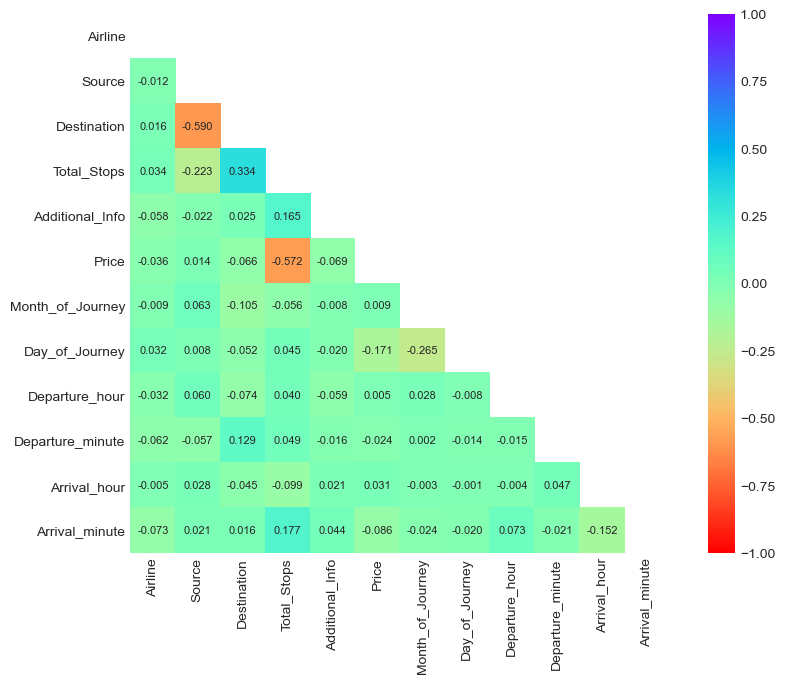
****

Just like in the case of outliers with the help of above distribution plots on our dataset gives us an indication on presence of skewness in our columns however for categorical data columns we do not have to worry about the outliers or the skewness in them.

**Correlation using a Heatmap**

* Positive correlation - A correlation of +1 indicates a perfect positive correlation, meaning that both variables move in the same direction together.
* Negative correlation - A correlation of –1 indicates a perfect negative correlation, meaning that as one variable goes up, the other goes down.

****

****

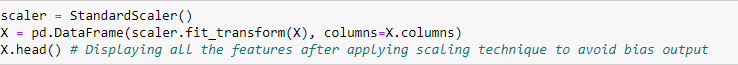
With the help of about heatmap we are able to notice the correlation details between our label and feature and also amongst our labels. After eye balling the above heatmap we can see that there are no multi collinearity concerns in our dataset so we won't have to worry about dealing with them.

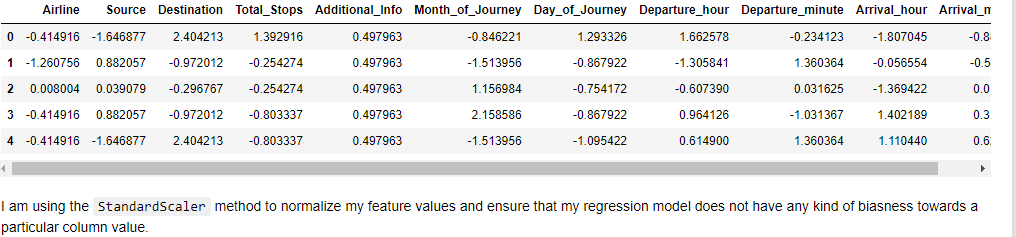
**Splitting the dataset into 2 variables namely 'X' and 'Y' for features and label**

****

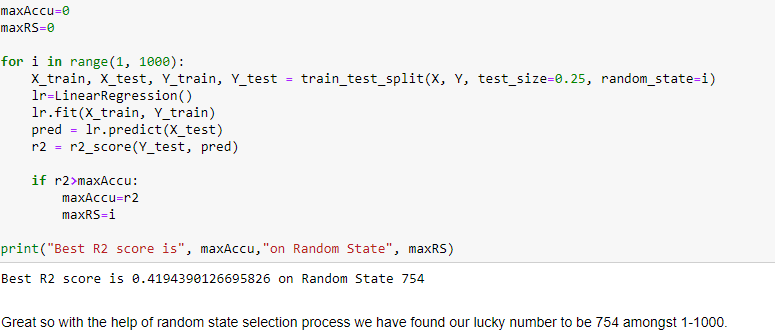
I have separated the dataset into features and label where X represents all the feature columns and Y represents the regression target label column.

**Feature Scaling**

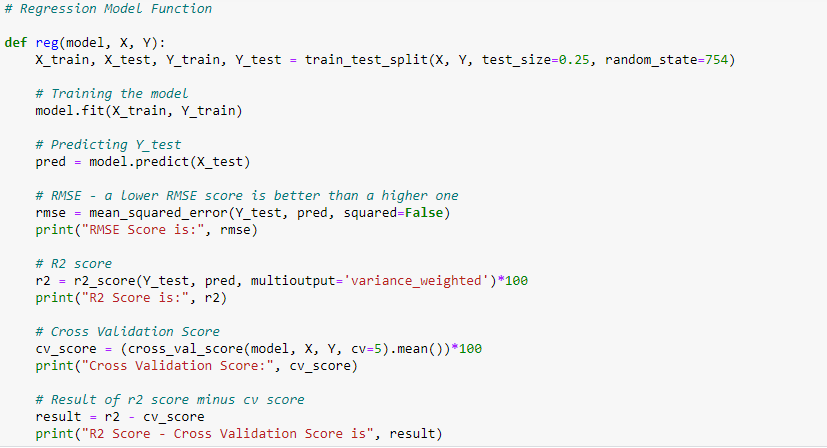
****

****

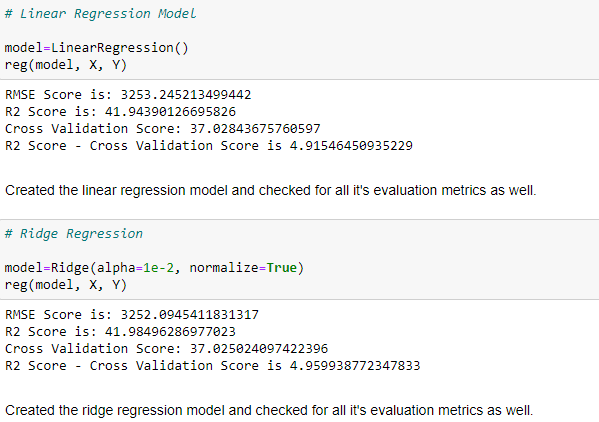
**Finding the best random state for building Regression Models**

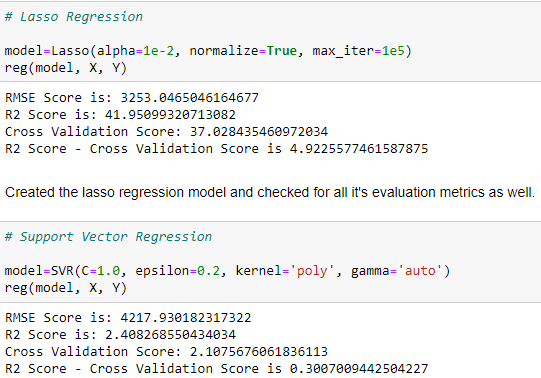
****

**Machine Learning Model for Regression with Evaluation Metrics**

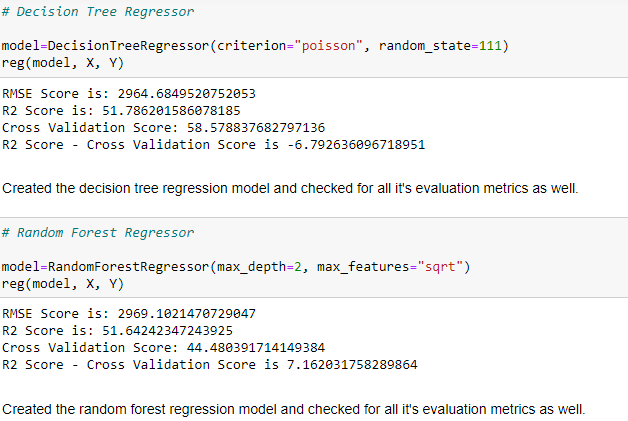
****

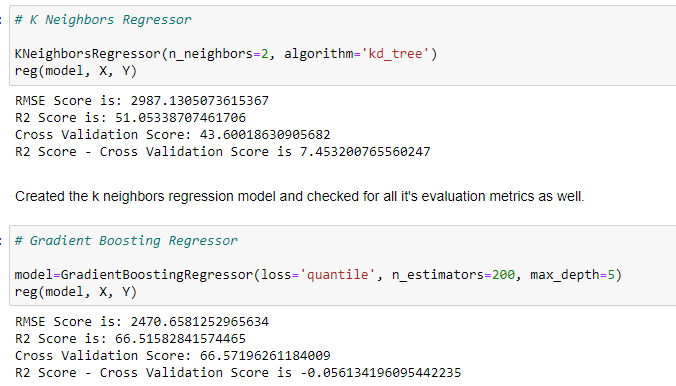
I have built a regression function that splits the training and testing features and labels, then trains the model, predicts the label, calculates the RMSE score, generates the R2 score, calculates the Cross Validation score and finally finds the difference between the R2 score and Cross Validation score.

****

****

Created the support vector regression model and checked for all it's evaluation metrics as well.

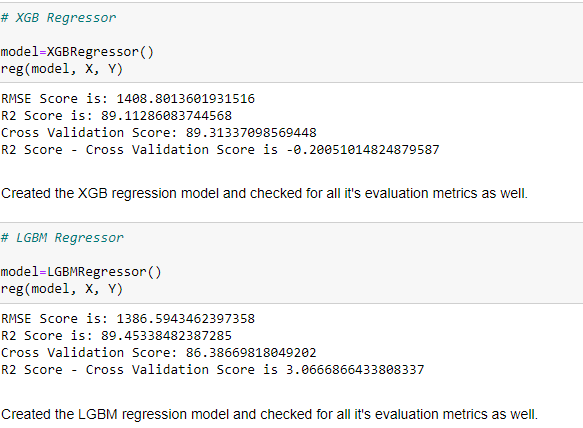
****

****

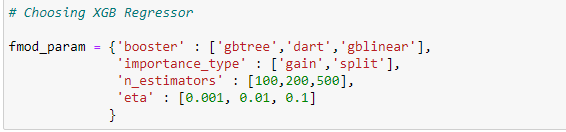
Created the gradient boosting regression model and checked for all it's evaluation metrics as well.

****

Created the extra trees regression model and checked for all it's evaluation metrics as well.

****

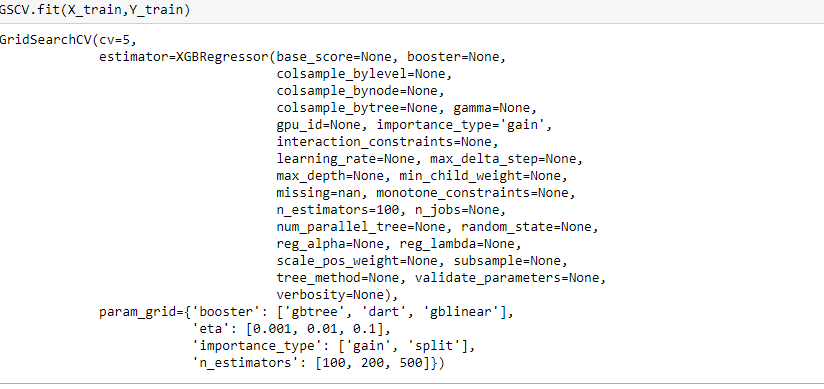
**Hyper parameter tuning on the best Regression ML Model**

****

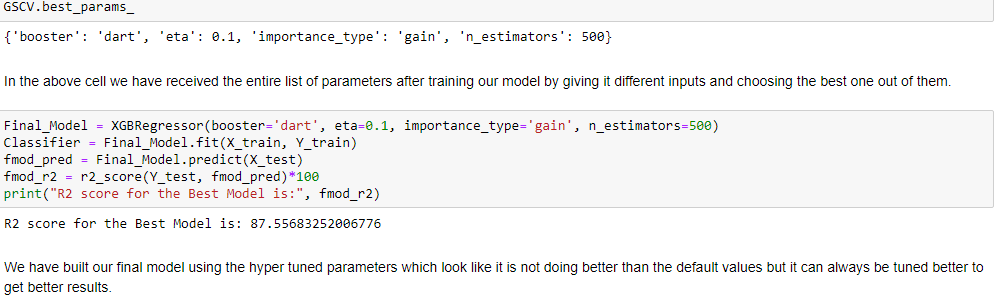
I have chosen the XGB regressor as my best model since it is able to provide the highest R2 score plus the model is doing better in Cross validation score too. In the above cell I have listed all the parameters for XGB regressor that can be used for hyper tuning our final model.

****

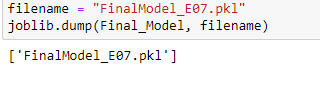
Here we are using the Grid Search CV method to perform the hyper tuning mechanism.

****

We are training our model by providing all the parameters we deem fit so it can go through all the permutations and combinations and identify the best value.

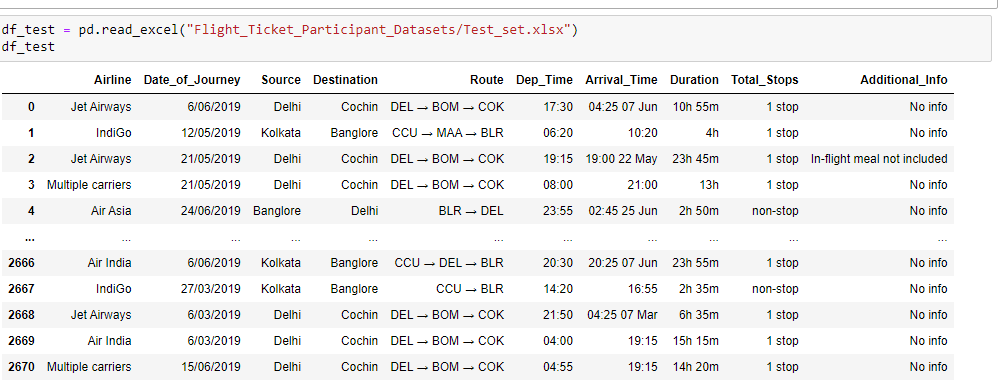
****

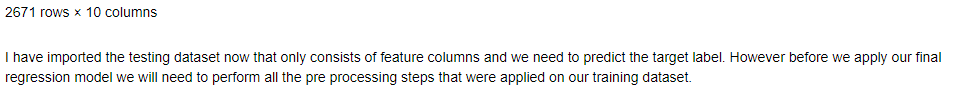
**Saving the best Regression ML model**

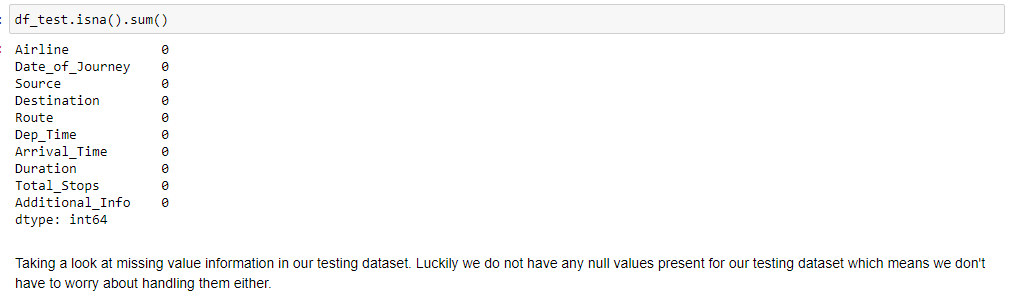
****

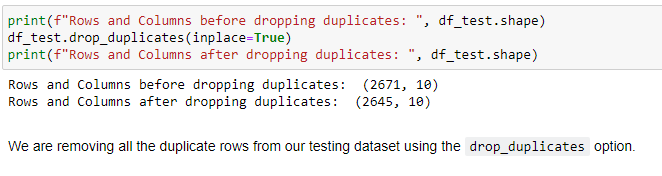
Finally I have saved my best Regression model using the joblib library.

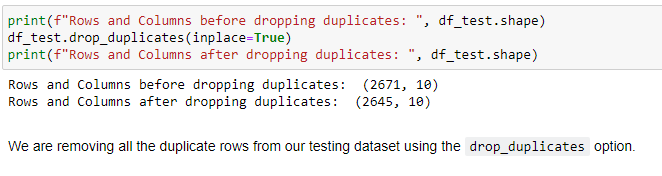
**Loading testing data**

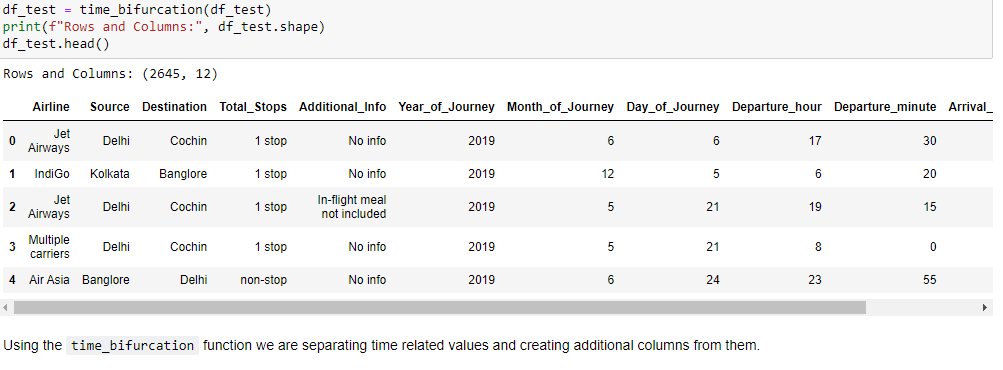
****









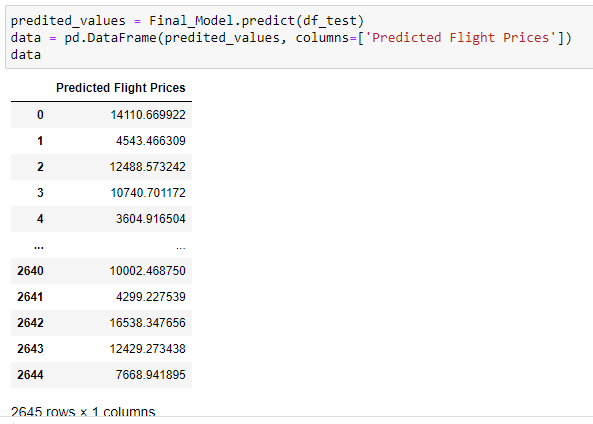




Now we are applying Ordinal Encoder technique since we have all our feature columns with an order category and it becomes easier to convert than the usage of One Hot Encoding method.



With the usage of StandardScaler we have transformed all our feature values in a normalized format to avoid biases in our regression model.



Conclusion

Hence, at the end, we were successfully able to train our regression model ‘Gradient Boosting Regressor’ to predict the flights of prices with an r2\_score of 87%, and have achieved the required task successfully.