

Flight Price Prediction Project

Submitted by:

ROHIT KATTEWAR

ACKNOWLEDGEMENT

I would like to thank and express my sincere gratitude to Flip Robo Technologies for giving me the opportunity to work on this project named ‘Flight Price Prediction Project’ using Machine Learning algorithms.

Primarily, I would like to thank to the author of the paper titled: “Airline ticket price and demand prediction: A survey” as well as “Trying to Predict Airfares When the Unpredictable Happens” for providing me invaluable knowledge and insights in determining the prices of flight tickets.

Finally, I will thank my mentors, under whose guidance I learned a lot about Machine Learning, Natural Language Processing and much more.

INTRODUCTION

* **Business Problem Framing**

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on – Time of purchase patterns (making sure last-minute purchases are expensive) and Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

Therefore, a predictive model is required to predict accurately predict Flight ticket fares.

* **Conceptual Background of the Domain Problem**

Predictive modelling, Regression algorithms are some of the machine learning techniques used for predicting Flight Ticket prices. Identifying various relevant attributes like Airline Brand, flight duration, source and destination etc are crucial for working on the project as they determine the valuation of air fare.

* **Review of the Literature**

A Research paper titled: “Airline ticket price and demand prediction: A survey” by Juhar Ahmed Abdella and online article titled: “Trying to Predict Airfares When the Unpredictable Happens” were reviewed and studied to gain insights into all the attributes that contribute to the pricing of flight tickets.

It is learnt that deterministic features like Airline Brand, flight number, departure dates, number of intermediate stops, week day of departure, number of competitors on route and aggregate features – which are based on collected historical data on minimum price, mean price, number of quotes on non-stop,1-stop and multi-stoppage flights are some the most important factors that determine the pricing of Flight Tickets.

* **Motivation for the Problem Undertaken**

With airfares fluctuating frequently, knowing when to buy and when to wait for a better deal to come along is tricky. The fluctuation in prices is frequent and one has limited time to book the cheapest ticket as the prices keep varying due to constant manipulation by Airline companies. Therefore, it is necessary to work on a predictive model based on deterministic and aggregate feature data that would predict with good accuracy the most optimal Air fare for a particular destination, route and schedule.

ANALYTICAL PROBLEM FRAMING

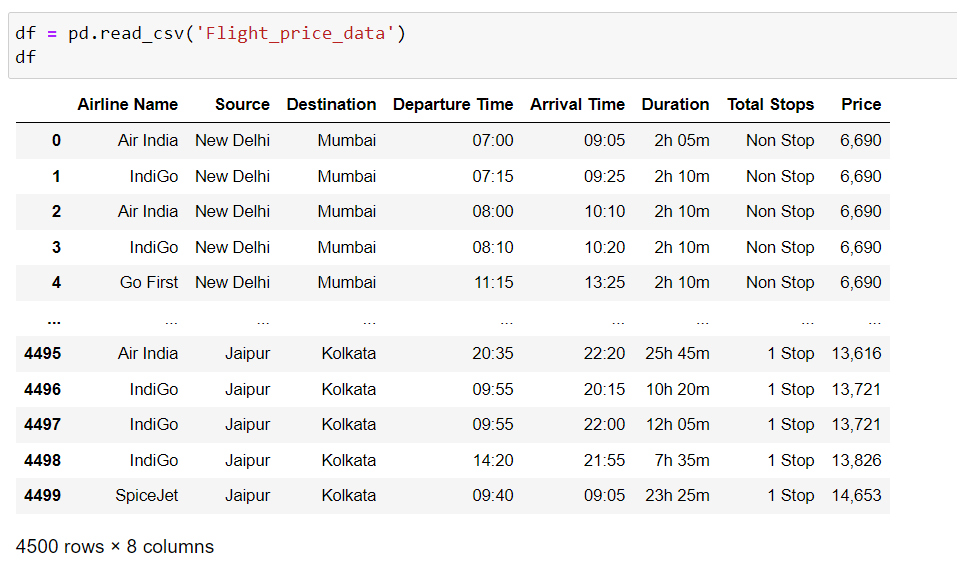
* **Mathematical/Analytical Modeling of the Problem:**

In order to forecast Flight Ticket price, predictive models such as ridge regression Model, Random Forest Regression model, Decision tree Regression Model, Support Vector Machine Regression model and Extreme Gradient Boost Regression model were used to describe how the values of Flight Ticket Price depended on the independent variables of various Air Fare attributes.

Various Regression analysis techniques were used to build predictive models to understand the relationships that exist between Flight ticket price and Deterministic and Aggregate features of Air travel. The Regression analysis models were used to predict the Flight ticket price value for changes in Air travel deterministic and aggregate attributes. Regression modelling techniques were used in this Problem since Air Ticket Price data distribution is continuous in nature.

* **Data Sources and their Formats**

The Dataset was created by automated test software named Selenium using web scrapping techniques for various Flight ticket attributes including prices from <https://www.yatra.com/>

****

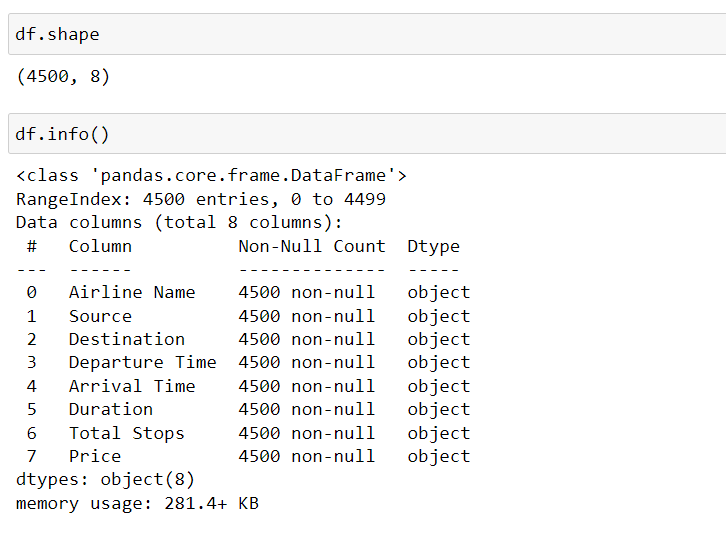
* **Dataset Description:**

**The Independent Feature columns are:**

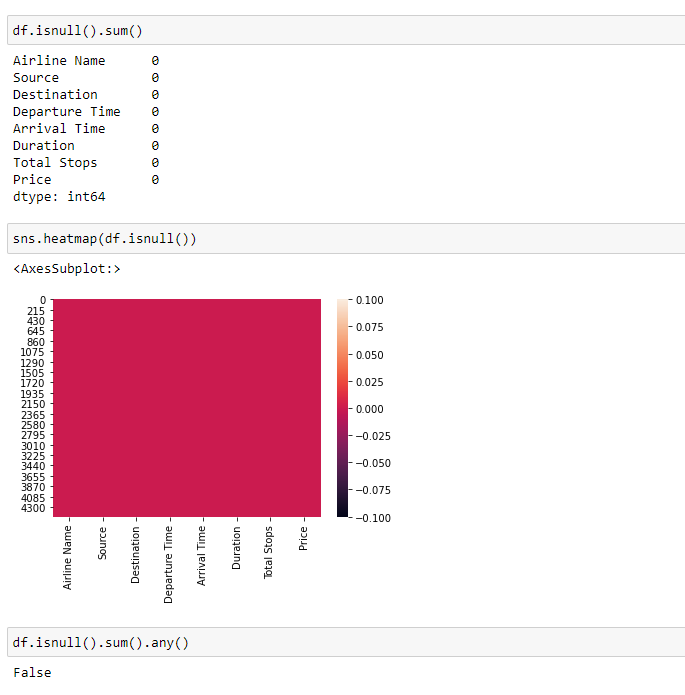
* Airline: The name of the airline.
* Flight Number: Number of Flight
* Date of Departure: The date of the journey
* From: The source from which the service begins
* To: The destination where the service ends
* Duration: Total duration of the flight
* Total Stops: Total stops between the source and destination.

**Target / Label Column:**

* Price: The Price of the Ticket



* **Data Pre-processing Done**

****

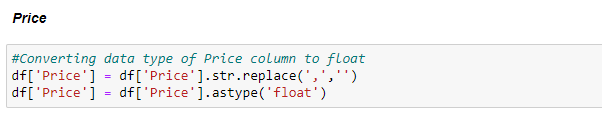
There are no null values present in the dataset.

Feature engineering:



Converted stops into numbers as shown in the above image.





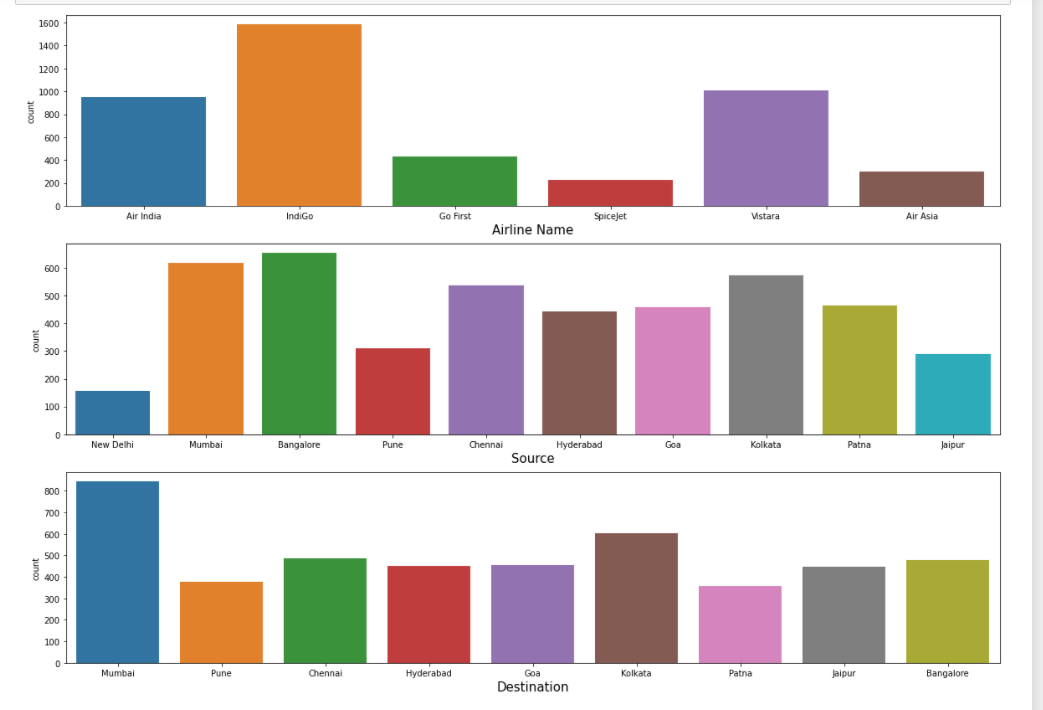
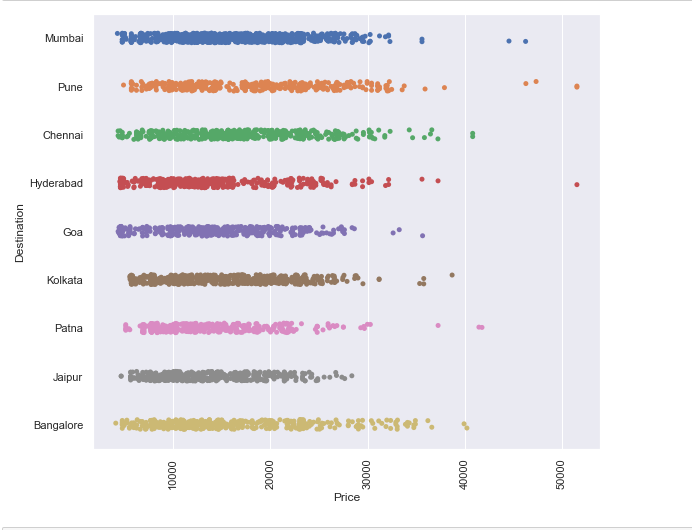
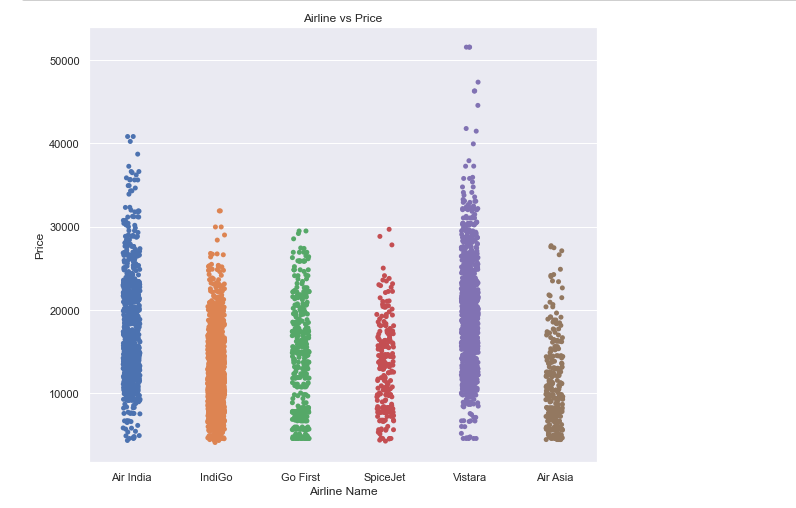
As shown in above images, we processed the data and converted in order to achieve the data cleansing.

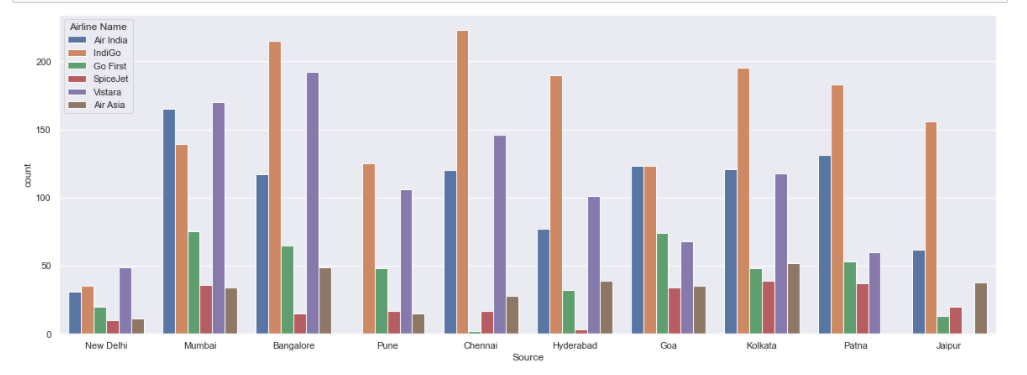
* **Data Inputs- Logic- Output Relationships:**

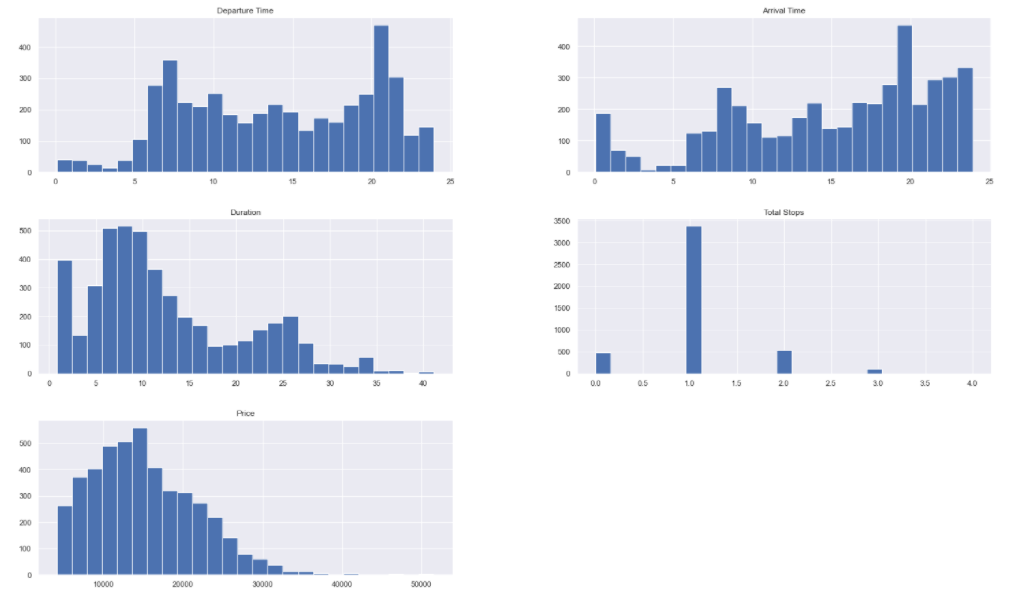
The Datasets consist mainly of Int and Object data type variables. The relationships between the independent variables and dependent variable were analysed.

* **Exploratory Data Analysis:**

**Visualizations:** Bar plots, Count plots, Box plots were used to visualize the data of all the columns and their relationships with Target variable.

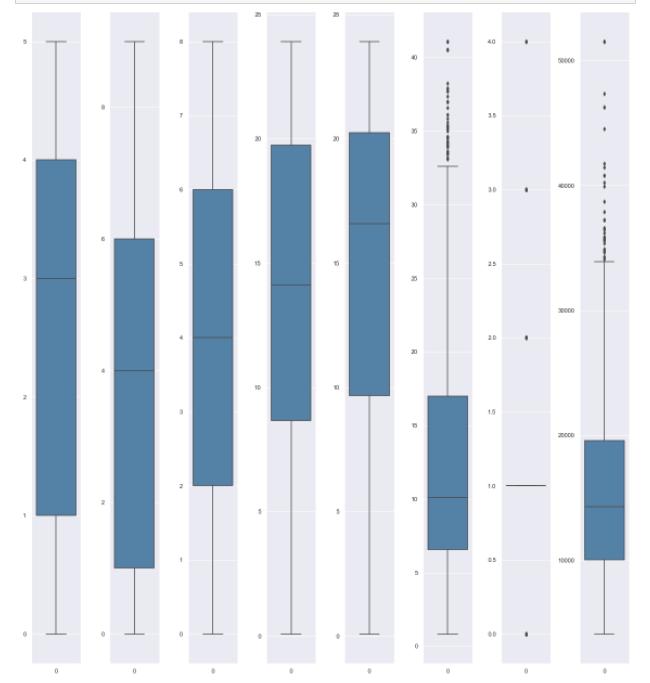








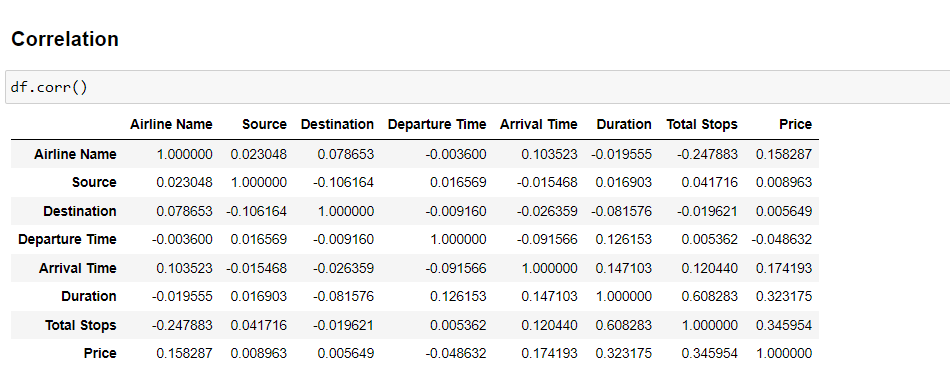
Outliers:

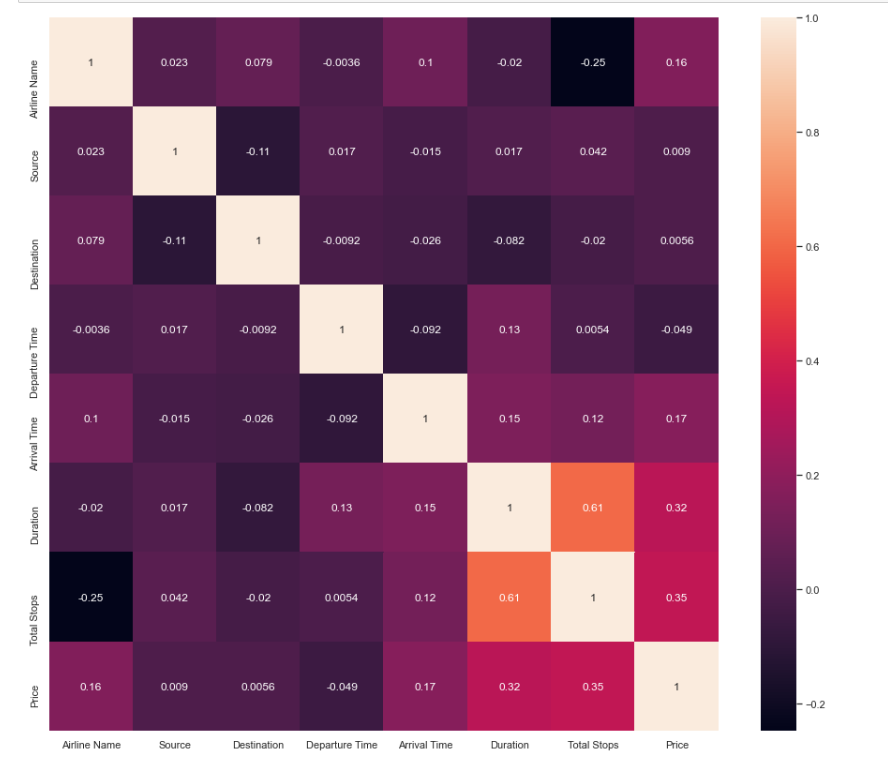


As we can see in the image, there are outliers present in the dataset. So, the following image shows we removed the outliers and after removal of the outliers we lost the 3.533% data.



Correlation:





* **Model Building:**

Finding the best Random state and accuracy:



The model algorithms used were as follows:

**Linear Regression:**

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, it refers to the method as multiple linear regression.

**Lasso Regression:**

Lasso regression is a type linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models. This particular type of regression is well-suited for models showing high levels of multicollinearity.

**Ridge Regression:**

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

**Decision Tree Regressor:**

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

**Random Forest Regressor:**

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

**Gradient Boosting Regressor:**

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

**XGBRegressor:**

XGBoost uses decision trees as base learners; combining many weak learners to make a strong learner. As a result, it is referred to as an ensemble learning method since it uses the output of many models in the final prediction. It uses the power of parallel processing, supports regularization, and works well in small to medium dataset.

**Support Vector Regressor:**

SVR works on the principle of SVM with few minor differences. Given data points, it tries to find the curve. But since it is a regression algorithm instead of using the curve as a decision boundary it uses the curve to find the match between the vector and position of the curve. Support Vectors helps in determining the closest match between the data points and the function which is used to represent them. SVR is robust to the outliers. SVR performs lower computation compared to other regression techniques.

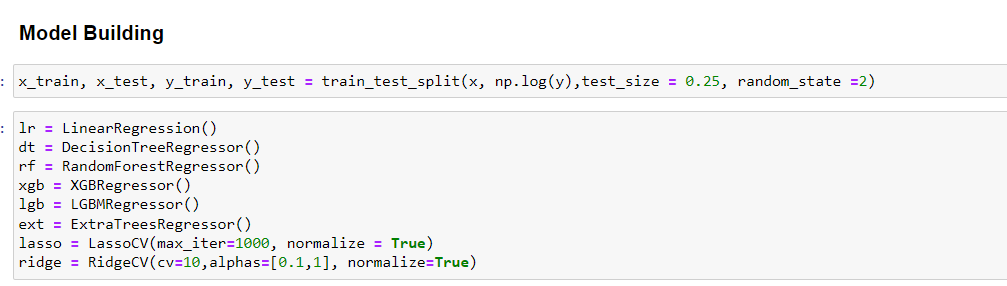
**Light GBM:**

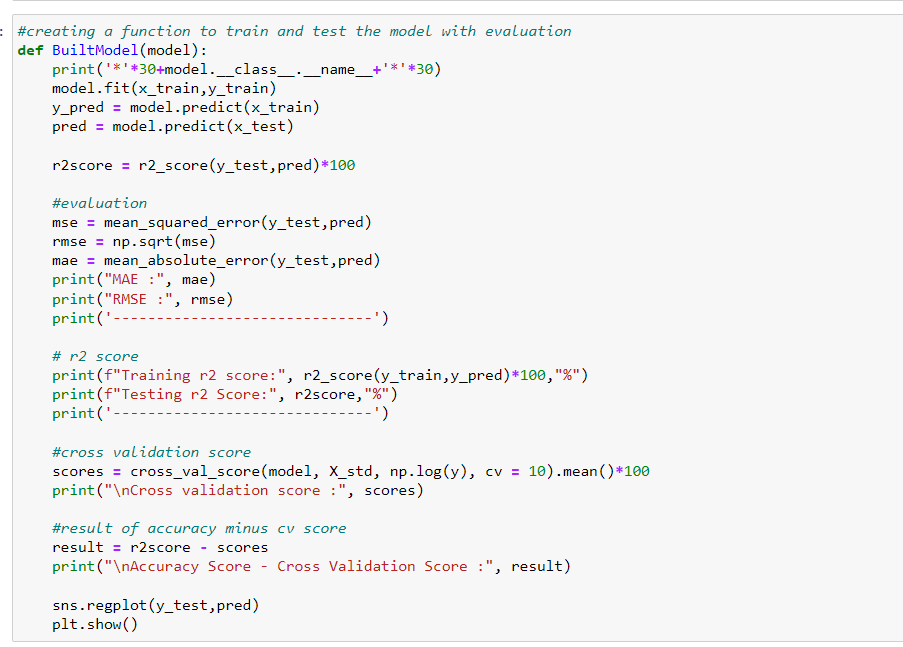
Light GBM is a gradient boosting framework that uses tree-based learning algorithm. Light GBM grows tree **leaf-wise**while another algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.

**Run and Evaluate selected models:**

The above-mentioned algorithms have been run in the jupyter notebook and the performance metrics are found as shown in further below figures:

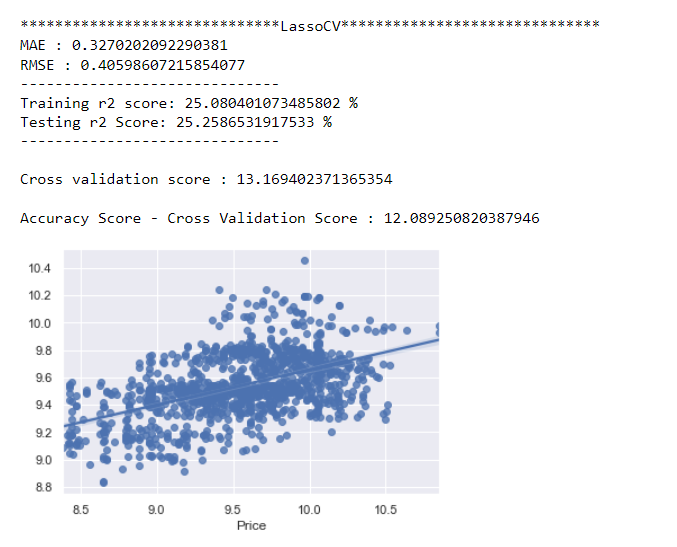
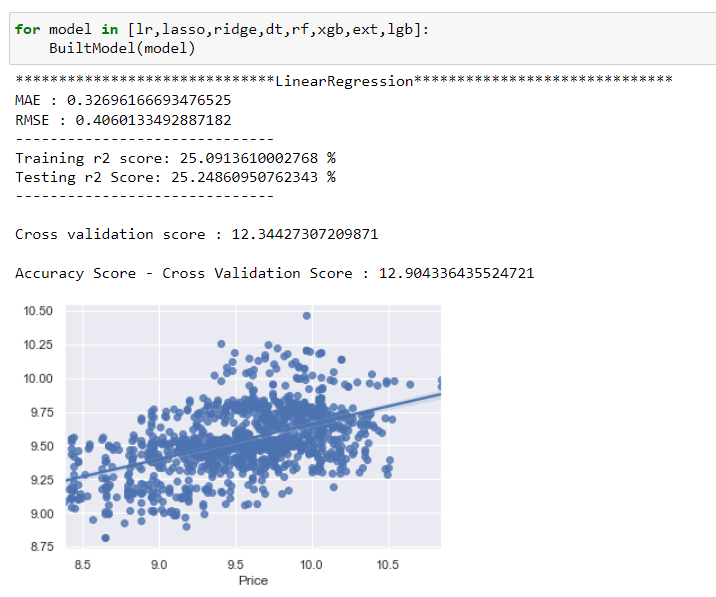
To check whether the model is overfitting/underfitting GridSearchCV is used and cross validated the models as shown in below figures.

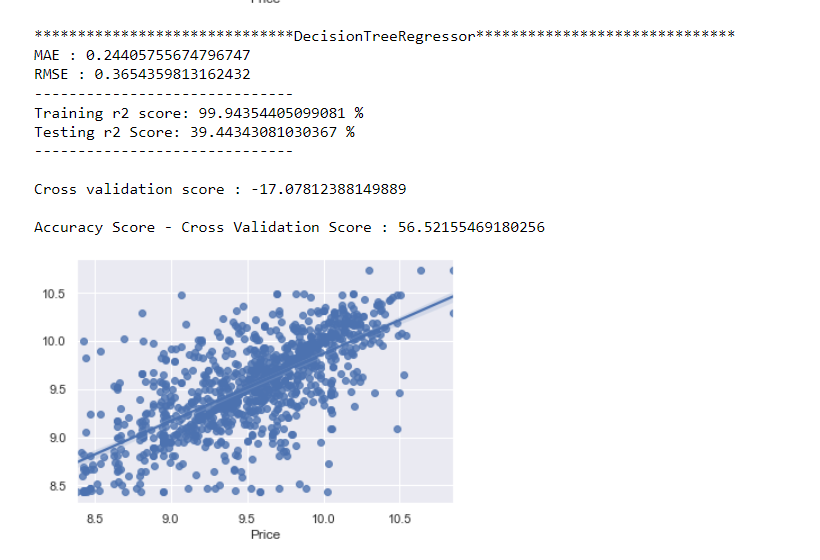
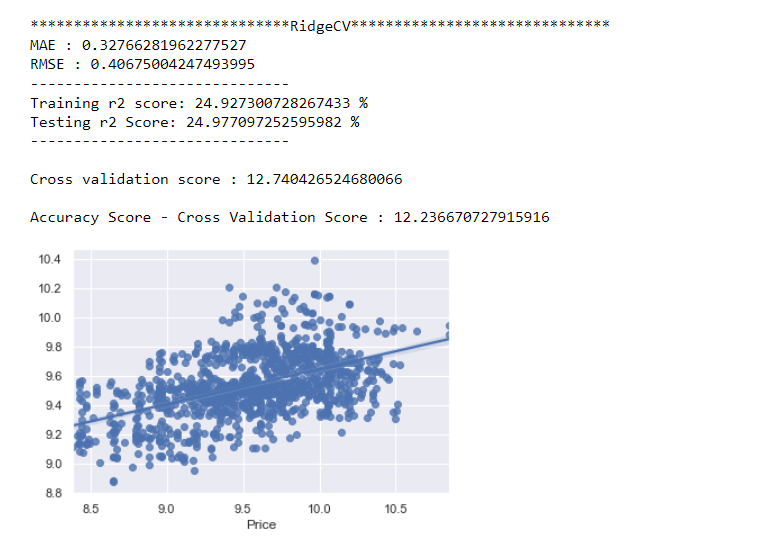


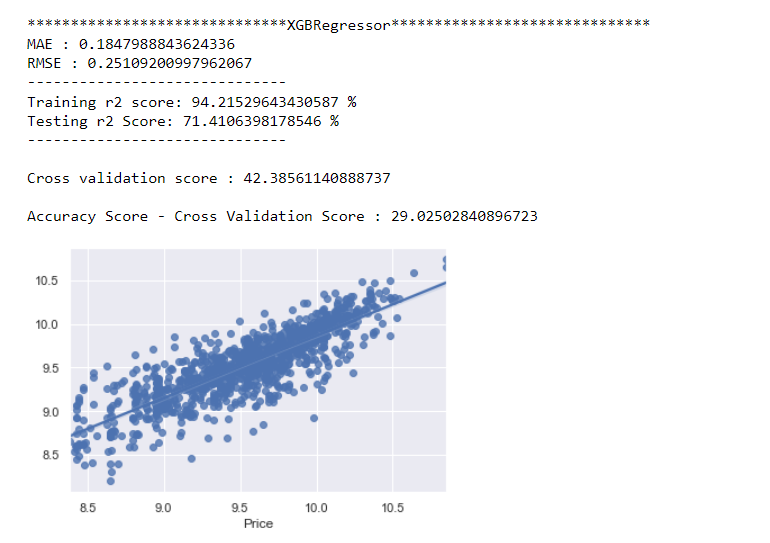
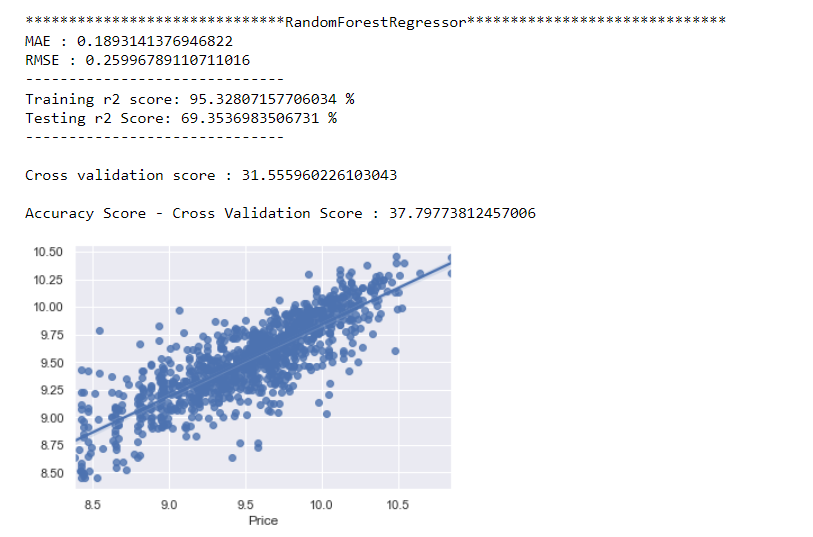


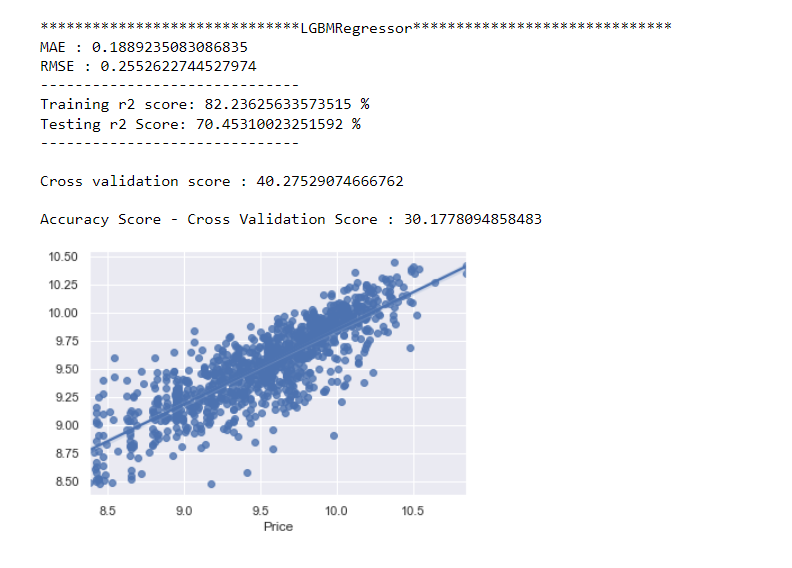
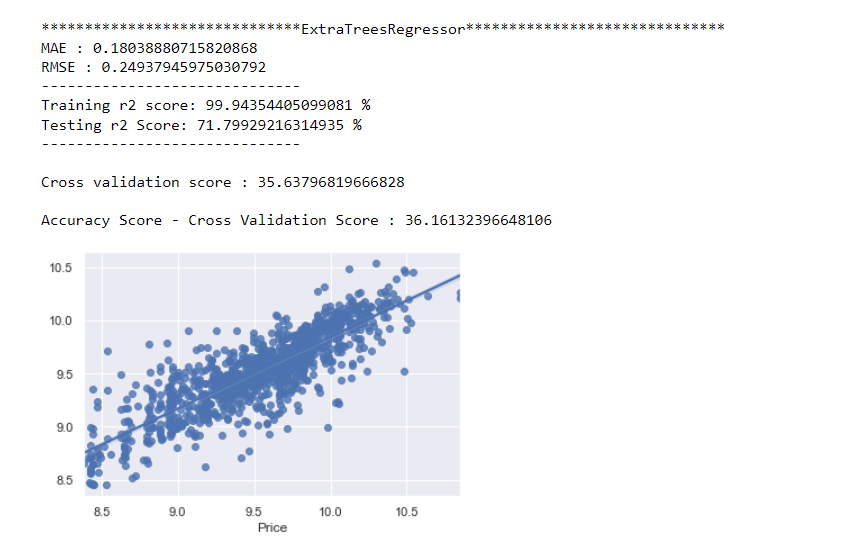
Created a Function called BuiltModel to train further test the model and gives the evaluation.

Following is the model’s evaluation:

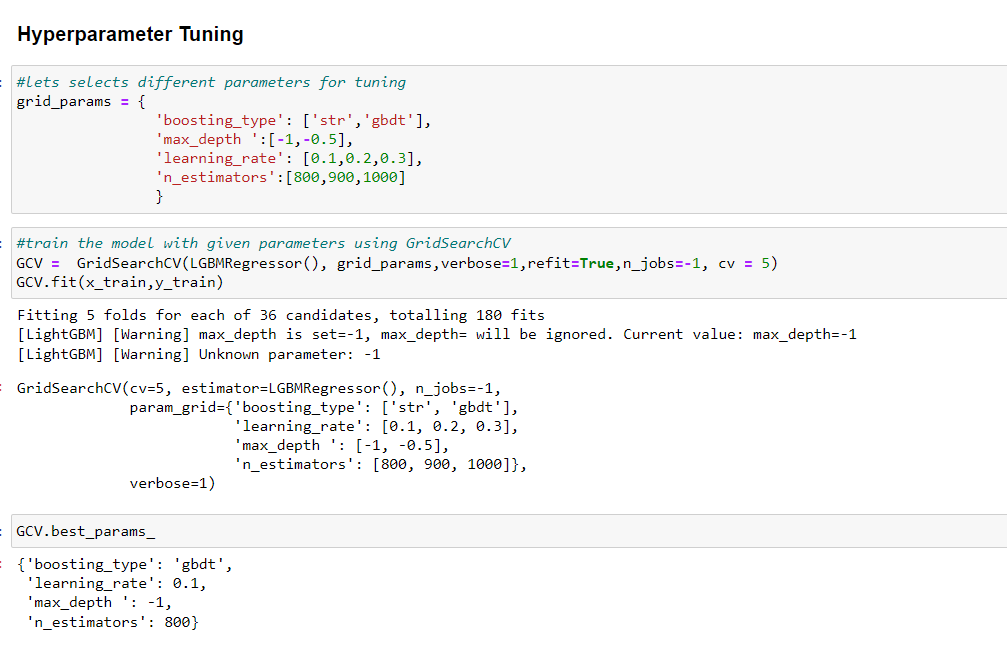








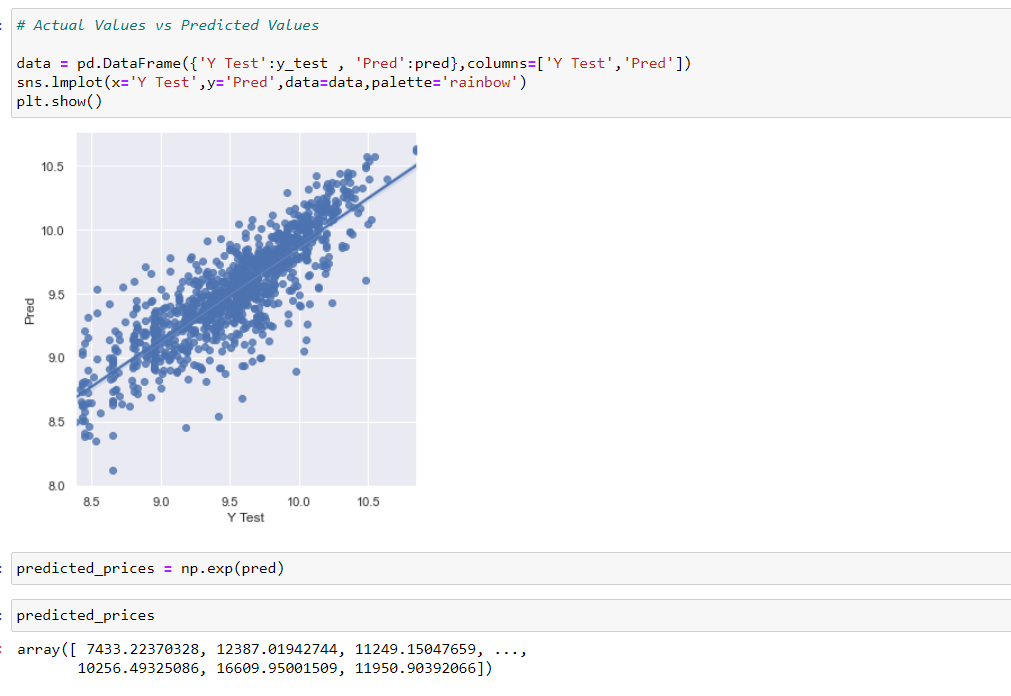
**Hyperparameter Tuning:**



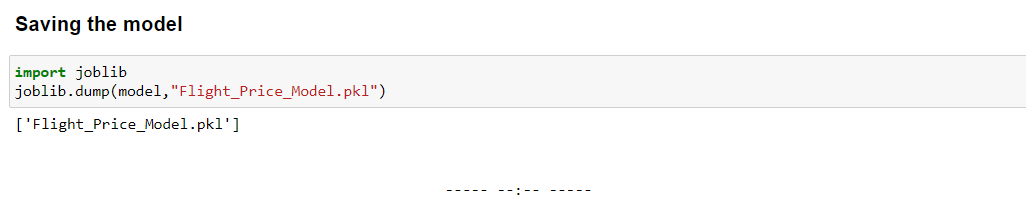


I selected Light GBM as the best model and from above we can say that, our model is giving 72.33% accuracy.

**Actual Vs Predicted Values:**

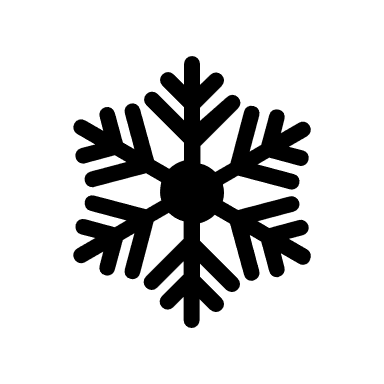


**Saving the model:**

****

**Conclusion:**

We scraped the flight attributes data from yatra.com. Then the csv file is loaded into a dataframe. The dataset has no missing values. Looking at the data set we understand that there are some features needs to be processed like converting the data types, and get the actual value from the string entries from the time related columns. After the data is been processed I have done some EDA to understand the relation among features and the target variable. Features like flight duration, number of stops during the journey and the availability of meals are playing major role in predicting the prices of the flights as looking at the features we came to know that the number of features is very less, due to which we are getting somewhat lower r2-scores. Some algorithms are facing over-fitting problem which may be because of a smaller number of features in our dataset. We can get a better r2 score than now by fetching some more features from the web scraping by that we may also reduce the over fitting problem in our models.

----- --:-- ----- ----- --:-- -----  ----- --:-- ---------- --:-- -----