**Blog Submission**

**Flight Price Prediction**

****

**Submitted By:**

**Arpan Pattanayak**

**Batch no:1842**

**INTRODUCTION**

Optimal timing for airline ticket purchasing from the consumer’s perspective is challenging principally because buyers have insufficient information for reasoning about future price movements.

Since the deregulation of the airline industry, airfare pricing strategy has developed into a complex structure of sophisticated rules and mathematical models that drive the pricing strategies of airfare. Although still largely held in secret, studies have found that these rules are widely known to be affected by a variety of factors. Traditional variables such as distance, although still playing a signiﬁcant role, are no longer the sole factor that dictate the pricing strategy. Elements related to economic, marketing, and societal trends have played increasing roles in dictating the airfare price

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. Airlines use using sophisticated quasi-academic tactics known as "revenue management" or "yield management". The cheapest available ticket for a given date gets more or less expensive over time. This usually happens as an attempt to maximize revenue based on -

1. Time of purchase patterns (making sure last-minute purchases are expensive)

2. Keeping the flight as full as they want it (raising prices on a flight which is filling up to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

The price of an airline ticket is affected by several factors, such as ﬂight distance, purchasing time, fuel price, etc. Each carrier has its own proprietary rules and algorithms to set the price accordingly. Recent advance in Artificial Intelligence (AI) and Machine Learning (ML) makes it possible to infer such rules and model the price variation.

## In this article, we will predict the flight price by using various machine learning models. Below are the takeaways:

## 1.      Problem Definition 2.      Data Analysis 3.      EDA 4.      Pre-processing Data 5.      Building Machine Learning Models 6.     Concluding Remarks

**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, 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.

## The problem statement explains that the target variable is continuous and it’s a “Regression type problem” since we need to predict the price of the flight tickets. In this project we will be using many regression models that can help the consumers to make purchasing decisions by predicting how flight ticket prices will evolve in the future.

## Attribute Information:

## 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

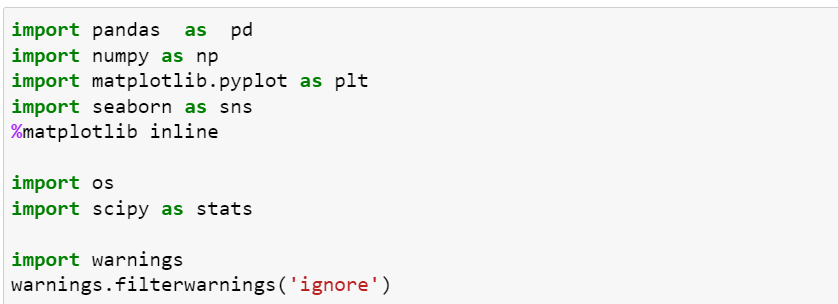
## Price: The price of the ticket

## In the dataset, there are 11 attributes where “Price” being the target variable and remaining 10 columns being independent variables.

## 2.Data Analysis

## The process of cleaning, transforming and extracting data to discover the useful information for business decision making is called data analysis.

**Importing necessary libraries and dataset**

****

There are two separate datasets,

* Train dataset: Train file will be used for building the ML models. It consists of 10 independent variables and 1 target variable.

Size of training set: 10683 records

* Test dataset: Test file will be used for getting the prediction from the trained model. It consists of only independent variables.

Size of test set: 2671 records.

Reading the training data of our dataset

**Table

Description automatically generated with medium confidence**

I have imported the training dataset. Training dataset contains 10683 rows and 11 columns. Price is our target variable.

## 3.Exploratory Data Analysis (EDA)

## Here we can get more information about our dataset

Graphical user interface, table

Description automatically generated

Looking at the overview of the dataset, only one integer column which is 'Price'. Remaining all columns are object type

The columns Date\_of\_Journey, Dept\_Time and Arrival\_Time showing object data type which means python is not able to understand the type of data in this column.

Therefore, we must convert this datatype into timestamp to use them properly for prediction.

Table

Description automatically generated

Now we have splitted Date\_of\_Journey column into 3 new columns of Year, Month and Day of Journey. All 3 new columns are integers data type. We can now drop the Date\_of\_Journey, Duration column.

Here we are dealing with time separation of Departure and Arrival column.

Text, email

Description automatically generated

Now while using the IsNull function, we will going to see the number of null values in our dataset

Text

Description automatically generated with medium confidence

We can find missing values in Route and Total\_Stops column, they both must be from the same row since we get the values of total stops from route column only. We can directly use dropna method but these two columns have categorical data so we will use mode method to fill the missing values.

## Treating null values using imputation techniques

## Text Description automatically generated

Here I have filled the null values using mode of that column. Mode means the frequently occurred values in the columns.

Let's see if any duplicate value present in our dataset and drop it.

Text

Description automatically generated with low confidence

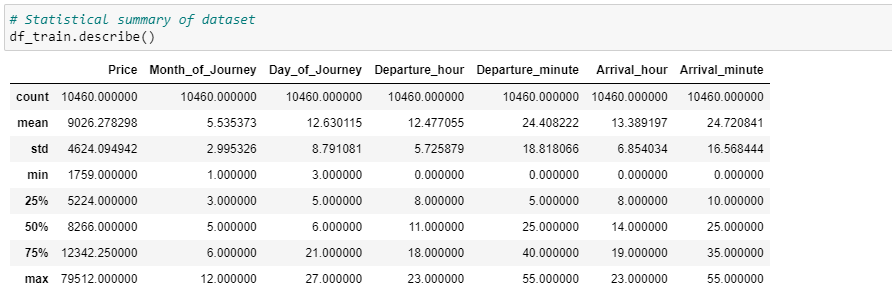
Let’s check the unique values

Table

Description automatically generated

Dropping 'Year\_of\_Journey' as it has only 1 value.

Statistical Summery:



By looking at the above statistics, Target variable 'Price' has huge data variance between mean and max. Mean value is greater than the median (50%) in the columns Price, Day\_of\_Journey, Departure\_hour so we can say they are skewed to right. Median (50%) is bit greater than mean in Departure\_minute, Arrival\_hour and Arrival\_minute which means they are skewed to left.

Now Let's separate the Numerical and Categorical columns

Graphical user interface, text, application, email

Description automatically generated

Data Visualization

Plotting categorical columns

Chart

Description automatically generated

On studying the graph of Airline column, we can observe that Jet Airways column has the highest number of flights followed by Indigo. In comparison, the number of flights of GoAir, Air Asia and Trujet are almost negligible.

Chart, bar chart

Description automatically generated

Most of the flights seems to take off from Delhi, while Mumbai and Chennai seem to have very low number of flights compared to others.

Chart, bar chart

Description automatically generated

Cochin has the highest count in the Destination column, which is not even present in Source. While Kolkata and Hyderabad have the lowest count compared to others. Also, Mumbai and Chennai which had low values in Source column, they are absent in Destination column.

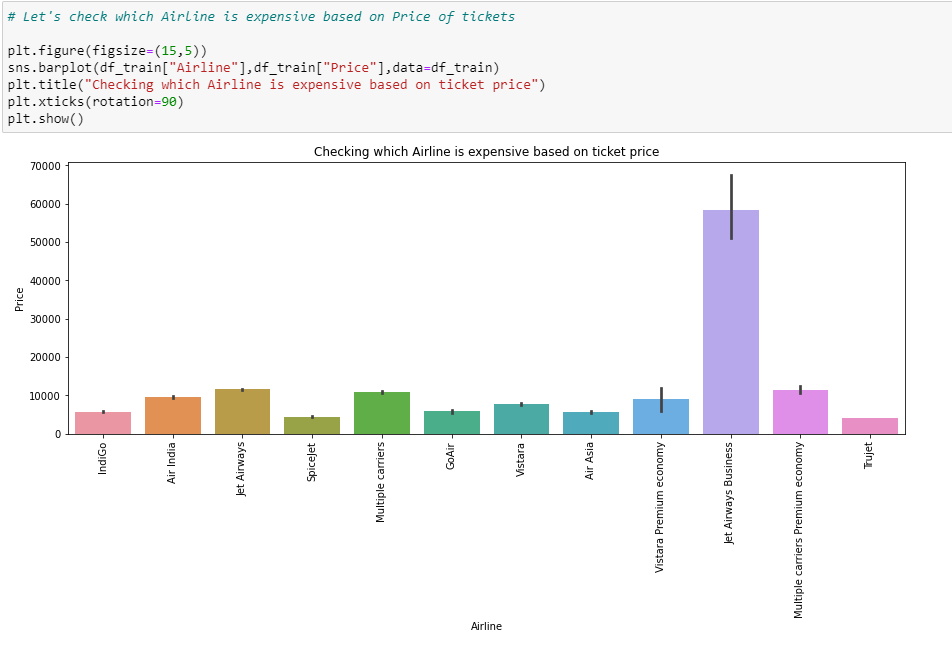
Chart, bar chart

Description automatically generated

The count is high in 1 stop followed by non-stop. Most of the flights have only 1 stop between the source and the destination. No flights have 4 stops between the source and destination.

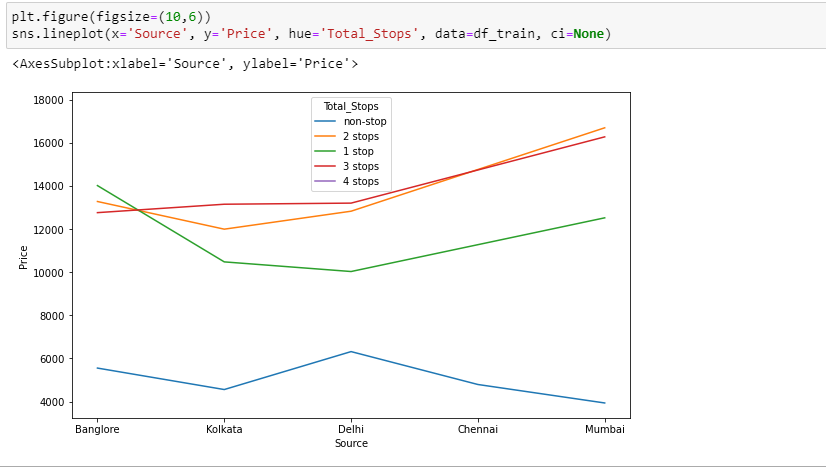
Plotting numerical columns

Airline vs Price



* The bar plot shows that the Jet Airways is most expensive Airline followed by Multiple carriers and Air India.
* The Trujet and Spicejet have very cheap ticket prices compared to others.

Source vs Price



The ticket prices are expensive in Delhi region compared to others and price is cheaper in Chennai and Mumbai sources.

Destination vs Price

Chart, line chart

Description automatically generated

The ticket price is high in Cochin destination followed by Bangalore which means they have long distance from the source.

Additional\_Info vs Price

Chart, bar chart

Description automatically generated

The plot shows that the Business class flights are more expensive compared to others and the flights having the class No check-in baggage included has very least ticket price.

In a similar manner we can plot the graphs for other columns too and we can make a good visualization by them.

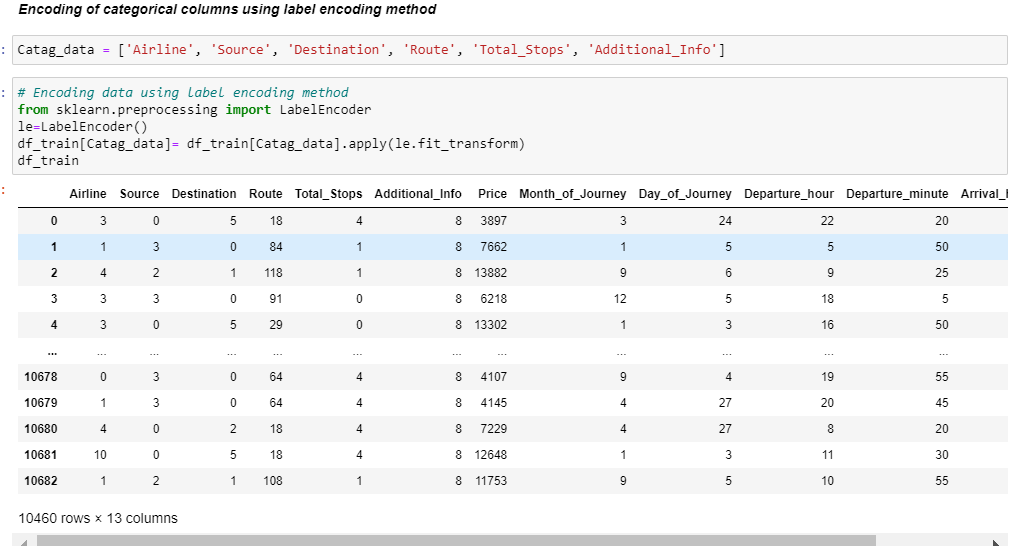
Checking skewness in the data

Table

Description automatically generated with medium confidence

Presence of skewness more than +0.5 and -0.5 is not acceptable as it will impact on our accuracy. Here we can find Price, Journey\_Month have skewness above the acceptable range. But the column Price is our target so I am keeping it untouched. Since we have both numerical and categorical data in the dataset, it is time to convert the object data type into numerical data type with the help of label encoding. There are many ways to convert categorical into numerical data but here I am using label encode the data.

Taking care of categorical columns using label encoding



Now we have converted the categorical columns into numerical columns using label encoding method.

Correlation between the label and features using HEAT map

## 

This heatmap shows the correlation matrix by visualizing the data. we can observe the relation between one feature to other.

* This heat map contains both positive and negative correlation.
* The feature Duration\_hours is highly positively correlated with the target variable Price.
* The feature Total\_Stops is highly Negatively correlated with the label.
* The features Duration\_ours and Total\_Stops, Duration\_hours and Destination are highly negatively correlated with each other. This may lead to multicollinearity problem so we will check the VIF value to solve this, if we get the features having VIF more than 10 then we can drop those columns. But it is very important to scale the data before checking VIF values.

Chart, waterfall chart

Description automatically generated

Here we can easily observe the positive and negative correlation between the label and the features.

**4. Pre-Processing Data**

## Separating the feature and label into x and y

## Graphical user interface, text Description automatically generated with medium confidence

## I have separated feature and label into x and y and checked for their shapes.

## Since the skewness of the data is in the acceptable range and the data is also normally distributed in the columns, in such case we can make use of Standard Scaler method else we can make use of Min Max scaler method.

## Standard Scaler method

## Standard Scaler helps to get standardized distribution, which makes mean = 0 and scales the data to unit variance. It helps in improving our model accuracy also solve the issue of data biasness.

## Table Description automatically generated

We have scaled the data using standard scaler method to overcome with the issue of data biasness.

In the heat map we have found some features having high correlation between each other which leads to multicollinearity problem. To solve multicollinearity problem, we will check VIF values. If we find VIF values greater than 10 in any features that means the features causing multicollinearity issue in the data. To overcome with this problem, we need to drop that feature column.

## Checking Variance Inflation Factor (VIF)

## Graphical user interface, text, application, email Description automatically generated

We can observe the none of the columns have VIF above 10 which means our data is free from multicollinearity problem.

Till here, same steps must be done for the testing dataset. I will make you understand better how to use test dataset to predict the values using saved trained model after building the machine leaning models.

Since we have done all the data analysis, EDA and pre-processing, now it is time to build our machine learning models.

**5.Building Machine Learning Models**

Before building the models, first we need to find the best random state and accuracy using any one of the regression models. We will import few basic libraries and find the best random state.

Graphical user interface, text, application, email

Description automatically generated

Here we are getting the maximum r2 score as 92.61% on the random state 71.

After we have found the value for best random state, we proceeded with the train test split function to create new training and testing datasets and fit them into the models to find our ideal models.

Random Forest Regressor

Graphical user interface, text, application, email

Description automatically generated

Random Forest Regressor giving R2 score as 92.57%

Decision Tree Regressor

Graphical user interface, text, application, email

Description automatically generated

Decision Tree Regressor giving R2 score as 86.92%.

Gradient Boosting Regressor

Graphical user interface, text, application, email

Description automatically generated

Created Gradient Boosting Regressor model and got the R2 score as 83.79%.

Bagging Regressor

Graphical user interface, text, application, email

Description automatically generated

We have created Bagging Regressor model and got the R2 score as 91.62%.

**Extra Trees Regressor**

Graphical user interface, text, application, email

Description automatically generated

Using Extra Tress Regressor model, we got the predicted R2 score as 91.05%.

Extreme Gradient Boosting Regressor (XGBoost)

Graphical user interface, text, application, email

Description automatically generated

Extreme Gradient Boosting Regressor giving R2 score as 92.34%.

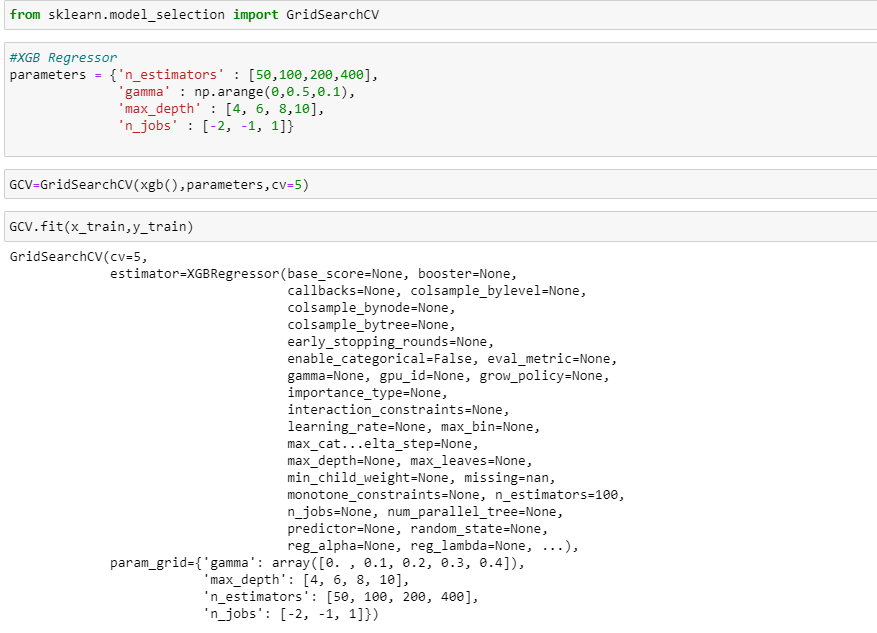
We have successfully built the models by training the data using x\_train and y\_train and with the help of x\_test we have got the prediction for every models. Also, we have got the R2 score with the help of prediction test and y\_test.

To check if the model is overfitted or not, we have also checked the cross-validation score.

The model XGB Regressor giving very less difference compared to other models.

Since XGB Regressor is giving best in R2 score and CV score difference, Evaluation metrics, so we choose XGB Regressor as best fitting model. Let’s check whether we can increase the R2 score by using hyper parameter tuning.

Hyper Parameter Tuning (Using GridSearchCV)



Graphical user interface, text, application, email

Description automatically generated

We tried to increase the model accuracy by using best parameters of XGB Regressor and again trained the model to get an increased R2 score. After tuning the model, we are getting R2 score as 92.49%

We have built the models and performed the hyper parameter tuning, now we will save the model to reuse it again while processing test data.

Graphical user interface, text, application, email

Description automatically generated

**Loading and predicting the saved model**

Graphical user interface, text, application, email

Description automatically generated

We have loaded the saved model to predict the ticket price using x\_test.

Test Data

Graphical user interface, table

Description automatically generated with medium confidence

* The test dataset contains 2671 rows and 10 columns. It is having all the same columns as train data except target column.
* You need to perform same steps that we have performed for train dataset, that includes Data Analysis, EDA, Scaling the data and Pre-Processing. After doing all these steps, no need to build any model using test data, only cleaning the data is required.
* Once you have done all these steps, all you need to do is to predict the flight price by using loaded trained data.

Prediction Results

I have used the loaded Model to predict the flight ticket price of the test dataset and compare the actual and predicted ticket price by creating dataframe.

Graphical user interface, text, application, email

Description automatically generated

I have got the predicted flight price ticket for testing dataset and by using above code, I have added the predicted price output to our original test dataset to complete it with feature and target column.

**6.Concluding Remark**

In this project we have gone through the feature engineering which is the most crucial thing and removed the outliers and skewness. Also handled the categorical columns by encoding the data, scaled the data and at last, we built different regression models to predict the flight price and performed the hyper tuning to improve the model by using different parameters.

With the help of above techniques, our model able to predict the flight ticket price with R2 score of 92%. Also, we have seen the actual and predicted values are almost same that means our model predicted is correct. This can help airlines by predicting what prices they can maintain. It can also help customers to predict future flight prices and plan their journey accordingly. So, Machine learning techniques are very useful to solve this kind of problems.