**Flight Price Prediction**

**Problem Definition**

The variation in prices of flight tickets has always been very confusing for customers and is very difficult to guess. The airline companies implement dynamic strategies to assign pricing for airfare tickets to increase demand for their seats and maximize their revenue, hence it becomes difficult for consumers to buy tickets at a minimum price. The closely connected terms with the airline prices such as commercial, financial, social factors and marketing is under consideration while practicing these dynamic strategies. The airline companies are trying their best to keep their revenue high and increase their profit. The travellers often find the flight prices unpredictable as the flight prices tomorrow will not be the same as the flight prices of today. The system is complicated because each flight has a limited number of seats to be sold. In case the demand for air tickets is high, then the prices will increase and on the other hand if the seats are left unsold then the cost of air tickets might decrease as it represents a loss of revenue. To solve this problem of predicting flight prices, Machine Learning is a great idea to learn from historical data of the past flight prices and build logic on the given data. We will use Linear Regression which will help us in predicting the flight prices on the basis of certain factors which will involve data extracting, data analysing and data interpretation.

**Dataset Information**

The prices of flight tickets for various airlines are between the months of March and June of 2019 and between various cities.

We have two datasets i.e. Train Data and Test Data.

The size of the **Training Set** is 10,683 records which consists of both categorical and numeric data. Some special characters are also seen within the data to which we will apply data transformation before using it on the Model.

The **features** considered initially for each flight are:

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

The size of the **Testing Set** is 2671 records. The testing data is similar to the training data, except for the “Price” column which will be predicted using the model.

**Data Analysis**

**Data Collection**

From numerous sources the data was collected. The flight ticket detailed information is retrieved from an online data source (github.com). We took out this data from the website which is in the form of a csv record. The file consists of the information with input features and its target variable required for analyzing data. We have retrieved additional features from the existing variables to get more accuracy in the results.

Features such as “Arrival\_Time”, “Arrival\_Date”, “Arrival\_Month”, “Day”, “Month” & “Year”

are generated to make analysis of data.

An important perspective is to wisely choose the necessary features required for calculating the flight prices as per expectations. All the existing variables and the retrieved features for each flight may not be required in making an accurate analysis of data, hence we only select variables that are significant and remove features with less importance for analyzing data accurately.

**Cleaning and Data Preparation**

All the accumulated data needs a great deal of work. So after information gathering, all the irrelevant data (features such as “Arrival\_Date” & “Arrival\_Month”), duplicate features and invalid qualities (features such as “Additional\_Info” & “Year”) are deleted. As the dataset contains missing values in variables such as “Route” and “Total\_Stops”, these values are resolved using SimpleImputer function. One of the necessary steps during data preparation is data transformation, hence the data present in string format is transformed to float data type using Encoding techniques such as OrdinalEncoder(). For example, “Airline” is a string data type and not an integer. This is done to construct a final dataframe from the original data and make a proper analysis of data. In all ML projects, this is the most important and time consuming procedure.

**Data Analysis Using Visualizations**

Data preparation is done by breaking down the information, understanding patterns and then applying different ML algorithms. In this case, we divide our data into three different dataframes on the basis of its type such as nominal, ordinal or continuous types of data.

**Visualization of Nominal Data**

We use countplot for nominal/categorical data as it gives the frequency of the columns. The following observations were made while analysing data:

* The maximum flights flying are from Jet Airways, Indigo and Air India and least are from Trujet & Vistara Premium economy.
* Most of the flights are flying from Delhi and the least are flying from Chennai.
* Majority of the flights are landing at the Cochin Airport and least are landing at the Kolkata Airport.
* The majority flights have only one stop in between the journey and most of them are also nonstop but very few flights have 3 & 4 stops.
* The flights have journeys mostly in the month of June, May and March.

**Visualization of Continuous Type of Values**

We use distplot and scatter plot graphs to understand the continuous type of data as distplot displays the density of values and scatter plot helps us find relations between the variables.

* The data is broadly scattered in all columns except for the price column.
* The data in the price column is right skewed but it is a target variable.
* In the Day column, the maximum number of flights are flying between the dates 3 and 7.

**EDA Concluding Remarks**

After performing all the transformations, integrations and cleaning of data, we get all the relevant variables and significant information required for building an ML model. We end up having 11 variables and 10,683 records in the dataset. The final dataset consists of important features used for analysis are:

* Airline
* Source
* Destination
* Route
* Dep\_Time
* Arrival\_Time
* Duration
* Total\_Stops
* Price
* Month
* Day

**Concluding Observation Remarks-**

* The standard deviation in the “Route”, “Dep\_Time”, “Arrival\_time”, “Duration” and “Price” column is too high which means that the values in these columns are largely scattered and are not near to the mean value.
* The standard deviation of other columns is not too high which shows us a normal distribution of data and less chances of having skewness.
* The value in the target variable (“Price”) has its minimum price at 1759 and maximum price at 79512. The range is too high.
* The most negatively correlated column is that of the “Total\_Stops” which means more the number of stops less the flight price.
* The most positively correlated variable is “Route”.
* The variables “Route”, “Arrival\_Time”, “Source”, “Month'' and “Dep\_Time” are positively correlated with the Target Variable and variables “Airline”, “Destination”, “Duration”, “Day” and “Total\_Stops” are negatively correlated.

**Pre-processing Pipeline**

**Skewness Correction**

As the range for skewness is threshold +/-0.5, we did not see much skewness in our data. Only variables “Airline”, “Destination”, “Price” and “Month” showed skewness wherein all the input features are of the Object data type and “Price” is the Target Variable. Hence, there is no skewness found in the data.

**Outliers Detection and Cleaning**

Statistical methods are used and factor analysis is done to detect outliers. In this case, we use the z-score function for detecting outliers. The outliers were observed only in the “Price” column which is the Target Variable and cannot be detected as an outlier. Therefore, there are no outliers in the data.

**Normalization**

In order to build and train a Machine Learning model, we have to standardize the data and get the values within a particular range for the model to understand data. We have used StandardScaler() technique for normalization because the value ranges are high in the data. Standard Scaler function will get all the values in the dataset within the range of 0 to 1. This will help the ML algorithms to learn data better.

Since the Target Variable is of the continuous type of values, we use Regression Algorithms. In this case, we have used **Linear Regression**.

**Building Machine Learning Models**

Building a model that will help to measure the performance of a better and more refined algorithm is the major goal here. We have used different Regression and Ensemble Techniques to compare and check which algorithm gives better performance and stack them all at the end to see how the model is giving predictions.

**Linear Regression**

Linear Regression is a Supervised Machine Learning algorithm that performs Regression tasks. Simple Linear Regression analysis is used to identify the correlation between two continuous variables. Prediction error is minimum when we find the best fit line for the given data using Linear Regression Algorithm.

The error rate of the model after using Linear Regression was found to be-

**Mean absolute error (MAE):** 2.81%

**Mean squared error (MSE):** 1.36%

**Root Mean squared error (RMSE):** 3.69%

A **cross-validation** technique was applied to all the samples and the mean performance of the model is produced.

The **best fit line** of the model was seen covering almost all data points which is an illustration of getting the best accuracy and indicates that the model has studied all the points in the data and there are no chances of having overfitting and underfitting issues.

**ElasticNet Regression**

ElasticNet Regression is a regularization regression technique which uses the best of both lasso and ridge regression models by learning from their drawbacks to better the regularization of statistical methods. Hence, we have applied this regularization technique to our regression model to avoid the risk of overfitting. The Regularization technique was put in use with the help of GridSearchCV.

**AdaBoostRegressor**

AdaBoost Algorithm is a boosting algorithm that is used as an Ensemble Technique. AdaBoost is a short form of Adaptive Boosting and is used to boost the performance of ML algorithms. It helps in combining multiple weak trainees and merge their performance to form one strong act. We do this to achieve a model with higher stability overcoming all the issues within the model by decreasing the variance of a single measure and combine several measures from variant models.

**Concluding Remarks**

* The r2 score achieved for Linear Regression is 100%.
* The accuracy achieved after applying cross validation and ElasticNet Regularization Regression technique comes out to be 99.99%.
* Applying the Ensemble Techniques to these Regression Models in order to achieve stability in the performance of the model, we get r2 score at 98.21% and CV val score at 97.66%.
* As we observe stability in performance and overfitting/underfitting issues resolved using the AdaBoostRegressor Ensemble Technique, we save it as the best model.
* Following the same procedures for the testing file as done for the training file (the complete EDA process), we then used the best saved model of the training file to predict the analysis of the testing file.
* Majority of the predicted outputs were found to be similar to the actual outputs at 98% accuracy.