**Flight Price Prediction**

Hello everyone, we are here to learn about how can we make a model to predict various flight prices.

**Introduction:**

As we know that nowadays, many numbers of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different situations. That is why we will try to use machine learning to solve this problem. This can help airlines by predicting how they can maintain their prices. It can also help customers to predict future flight prices so that they can plan their journey accordingly.

**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, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable. Here we will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

**Data Overview:**

Here we are having the train dataset as well as test dataset for testing the prediction of our model.

Size of training set: 10683 rows and 10 columns

Size of test set: 2671rows and 9 columns

#### Features 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.

**Data Analysis:**

Let's start with our most first step which is data analysis and the most important step also. And one more point is that whatever we will do with our train dataset we will do the same with test dataset also so we could make the prediction at the end.

* In train dataset all the columns are object type, only price column (which is our target column) is of integer type. In test dataset all the columns are object type.
* As we know that price is continuous type and it is our target variable so our problem will be of regression type.
* If we talk about null values in dataset, train data have 2 null values in two columns. one null value in each column. Since the null value is only 2 therefore, we can drop it easily instead of replacing.
* Since all the columns are categorical type and there are some special symbol presents in column so we have to transform them so that they can easily applicable for model.
* There are some date and time columns, first we have to convert it into datetime type.

**Exploratory Data Analysis:**

Now, we are going to do some visualization and statistical analysis for better understanding of data.

First, we did some **Univariate Analysis** so that we came to know some facts.

We did some observation from univariate analysis which are given as.

* Jet airways are highest in count in both datasets, but in train data trujet airline has only one count while in test data it is not present. Either we can drop this airline or we can keep it, according to the requirement. But we did not want to lost any data so we will not drop it.
* In both the datasets, source column has Delhi as highest count and Chennai as least count.
* And destination column has cochin as highest count and Kolkata as least count.
* In both the dataset, total stops column has 1 stops as highest count and 4 stops as least count.
* Mainly price is in between 0 to 40000 after that it becomes almost zero, we have to remove that part so that price column could not contain any unnecessary data.

Now, it’s time to do some **Bivariate Analysis** so that we came to know relationship about independent features and with dependent column also.

We did some observation from bivariate analysis which are given as.

* Jet airways, Vistara premium economy and multiple carriers premium economy has highest price among all.
* Among all the source city Delhi has highest price, and for destination city also Delhi has highest price.
* More no. of stops between source and destination leads to increase in the price.
* The month of January, May, June and October has highest price among all the months.

Now, we did **Multivariate Analysis** and analyzes the correlation between features and dependent column and among features also (i.e., about multicollinearity).

We use heatmap for analyze the correlation because it is very easy to understand from heatmap. And we came to know that,

* Duration hour is very much correlated to target column. Other than this rest columns do not influence much to target column.
* There is no multicollinearity between columns.

We did some statistical analysis and we came to know that,

* Some columns have mean >> standard deviation and some have mean < standard deviation.
* Some columns have 75% value << max value, there may be outliers present in these columns.

After all these analyses of data we are going for data preprocessing which is very important to do, because model accept data in some standard format it can not process the data as it was collected.

**Preprocessing of Data:**

* We extracted all the informative values from above date time columns and then we can drop original columns.
* Likewise, we have duration column and we have to make it in proper form so it could not be a problem.
* From this column we extracted the hour and min and made two separate columns for that and dropped original column.
* We have column named route which have multiple places and name separated by arrows for which we splitted every route and we did some feature engineering to make the information valuable. This leads to increase number of columns.
* As we already filled all the null values so we are going to do encoding of categorical columns.
* We did ordinal encoding for all categorical columns.
* We checked for the outliers. Outliers present in both datasets which are in categorical columns so we should not remove that, and from target column also. Therefore, we kept as it is.
* After doing all above processes our dataset shapes became 10681 rows and 18 columns in train dataset and 2671 rows and 17 columns in train dataset.

After all these pre-processing we are ready for splitting the data into training and testing part.

And now, we did some scaling, we scale the data so that all the features came onto the same scale otherwise our model will be biased towards high range of feature column.

We used MinMaxScaler to scale our feature columns.

**Building Machine Learning Model:**

Since our problem is given which has continuous type of target column so we model our data with the help of different regression techniques and comparing them to see which algorithm is giving better performance.

We used several models for training our data such as Linear Regression, Support Vector Machine, Decision Tree Regressor, KNeighbors Regressor, Random Forest Regressor, AdaBoost Regressor, XGB Regressor and we also used some Regularization technique such as Lasso, Ridge and ElasticNet.

**Linear Regression:**

To determine the correlation between two continuous variables, simple linear regression analysis is used. It gives the statistical relationship not the deterministic relationship between two variables. Linear regression algorithm gives the best fit line to the given data for which the prediction error is minimum.

**Support Vector Machine:**

Support vector machine is a supervised machine learning algorithm which can be used for both classification and regression problems. For classification we use support vector classifier and for regression we use support vector regressor.

It separates the classes with the help of maximum margin hyperplane.

**Decision Tree Regressor:**

Decision Tree Regressor is a supervised machine learning algorithm which can be used for both classification and regression problems. It splits the population to make a tree and it uses the criterion Gini or Entropy to know the impurity level on the basis of impurity it decided the root node and its child node.

**KNeighbors Regressor:**

In kneighbors regression analysis, the output is mean of its kneighbors. Kneighbor is a supervised classification algorithm that can also be used as a regressor. It assigns a new data point to the class. It is non-parametric because it does not take any assumption. It calculates the distance between every training point and a new data point.

**Random Forest Regressor:**

This is an algorithm which ensembles the less predictive model to produce better predictive models. It aggregates the base model to create a large model. The features are sampled and passed to trees without replacement to obtain the highly uncorrelated decision trees.

**AdaBoost Regressor:**

AdaBoost algorithm, short for Adaptive Boosting. It is a boosting technique that is used as ensemble method. It works on the principle where learners are grown sequentially, except for first each learner grown from previous grown learners. All means weak learners are converted into strong ones.

**XGB Regressor:**

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.

Extreme Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling problems.

Ensembles are constructed from decision tree models. Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models.

**Regularization:**

We used few regularization techniques which are Lasso, Ridge and ElasticNet.

These are the form of regression, that regularizes the coefficient estimates towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

* All these models are used for the training purpose and we also introduced the cross-validation score so that we can understand the overfitting or underfitting nature of our model.
* We used metrices like r2 score, root mean squared error etc. for the analysis of accuracy of our model prediction.
* On the basis of r2 score and cross validation score we select few models such as Random Forest Regressor, KNeighbors Regressor, AdaBoost Regressor, XBG Regressor for hyper parameter tuning.
* Rest of the models have high difference between r2 score and cross validation score and high root mean squared error as well, that is why we did not use them for further process.

It's time to do hyper parameter tuning of rest of the models, it need to be tuned so that the model can optimally solve the machine learning problems.

* There are different types of technique available to so hyper parameter tuning and here we did hyper parameter tuning using Grid Search CV.
* After doing hyper parameter tuning, we could see that XBG Regressor have highest r2 score and least root mean squared error value.
* So, it will be our final model.

Now, we have to save our model for further predictions, we can use pickle and joblib to save our model, here we are going to use joblib to save the final model.

As we have test dataset also so for that, first we have to load the saved model.

After loading the model, we did prediction with test dataset.

**Conclusion:**

Machine Learning algorithms are applied on the dataset to predict the dynamic fare of flights. This gives the predicted values of flight fare to get a flight ticket at minimum cost.

As we did so much of processes and we came to know that,

* The flight prices vary at different months and also at holidays.
* The flight prices vary according to the departure time also.
* The flight prices vary according to the days of week such that at the weekends prices are high as compare to other days.

Since this was a small dataset but if we provided large dataset and with some more information then prediction will be better.