Pattern Recognition and Machine Learning

Predict the flight ticket price

Name: Sakshi Todi (B20EE088)

Introduction

Machine learning has a wide variety of applications in today's world such as classifying emails as spam or not-spam, etc. Nowadays predicting flight ticket prices is something hard to guess, today we might see a price, check out the price of the same flight tomorrow and it will be a totally different story. So, in this project our main motive is to predict flight ticket price on the basis of various different parameters such as duration, day, route, etc.

Dataset

For this project we will be using a dataset which has 11 columns and 10683 rows.

The column names are:

- ➔ Airline (categorical feature)
- ➔ Date_of_Journey
- ➔ Source (categorical feature)
- ➔ Destination (categorical feature)
- ➔ Route
- ➔ Dep_Time
- ➔ Arrival_Time
- ➔ Duration
- ➔ Total_Stops (categorical feature)
- ➔ Additional_Info (categorical feature)
- ➔ Price (dependent variable)

The dataset is split in the ratio 70:30 (train:test).

Process

1) PRE-PROCESSING

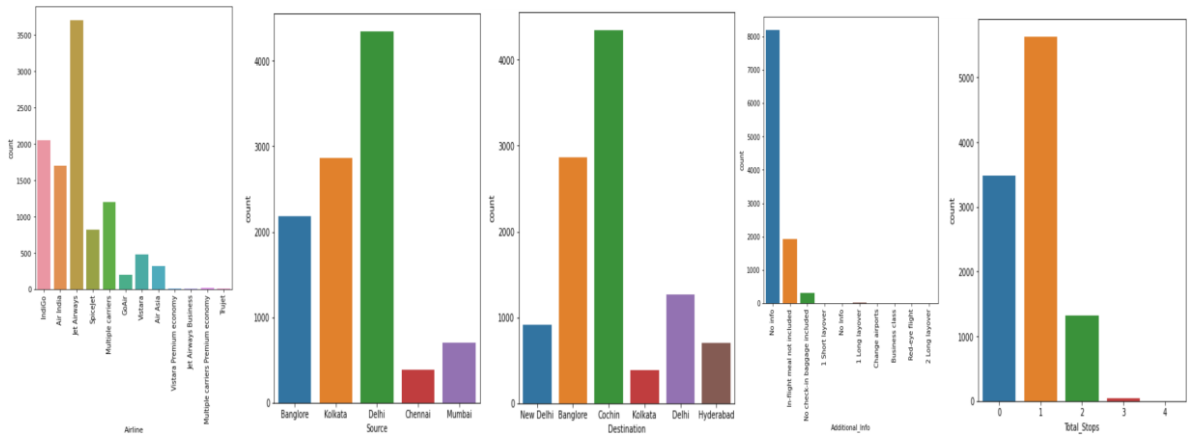
The very first step is to explore and pre-process the data. We begin with dropping the row which has NA values and the duplicate rows. We observe that the number of stops is directly proportional to the route. So, we can drop the Route column and we apply One hot encoding on the Total_stops column. Now for the column Date_of_Journey, we observe

that the year is same for all. So, we can ignore year and we split date and month in new columns and drop Date_of_Journey. Similarly for Dep_Time and Arrival_Time, we split it into two new columns with hour and min and drop the original columns. For the duration column, we convert all the data into just minutes for simplification. Next, we apply label encoding on the columns Source, Destination and Additional Info. Also, we apply one hot encoding on the Airline column. At last we apply standard scaler.

2) DATA VISUALIZATION

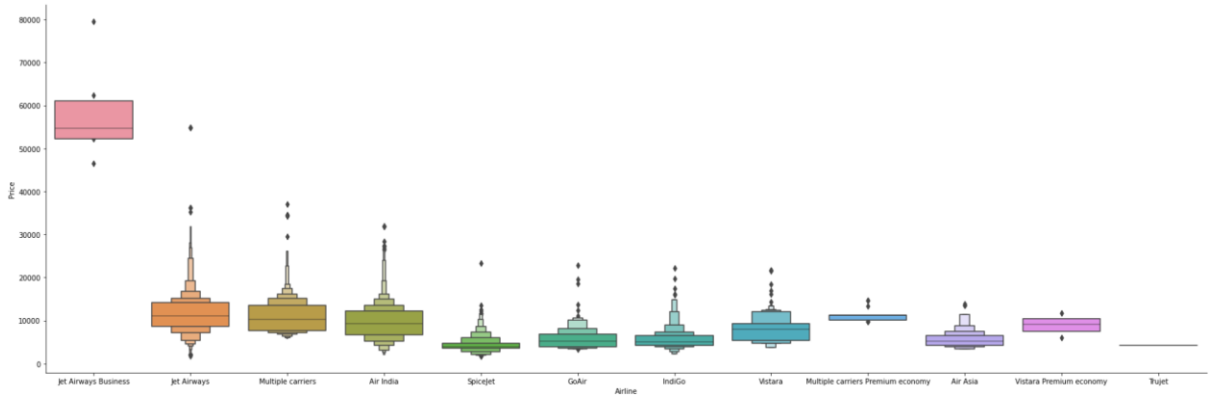
The next important step to explore the data is to visualize it.

First of all, we visualize the countplot of different columns:



From the above plots we observe that the count for Jet Airways is maximum in the Airline column. For the Source column we observe that maximum flights take off from Delhi and the destination for maximum flights is Cochin. For the Additional info column we observe that most of the flights to don't have any additional information. Number of stops is interrelated to route. So, we see that most of the flights have 1 stoppage.

Next we try to compare different attributes with price(targeted variable) using catplot and we conclude with similar type of results as in the case of countplot. We make an important observation that almost all the Jet Airways flight costs between 5k-6k. On such plot is:



We also try to visualize distribution plot for all the attributes. We observe that more people travel during May and June. Maximum flights depart during early morning and arrive in the evening. There are very few flights whose departure day and arrival day is different.

Duration of maximum flights is about 120-200 minutes. For the targeted variable column, maximum flight costs between 2k-20k.

Also, we visualize the independent variable with dependent variable using scatter plots to establish relation between price and different attributes similar to catplot.

3)TRAINING DIFFERENT MODELS

a) RandomForestRegressor

Random Forest Regressors use boosting to train upon various decision trees and then combine them together to give the results. It gives an accuracy of about 88%.

b) DecisionTreeRegressor

Decision Tree is a decision-making tool that uses a flowchart-like tree structure to give us the best results. It gives an accuracy of about 77%.

c)KNeighborsRegressor

It is the regression based on K-nearest neighbors. It is basically an algorithm which uses distance between two points. It gives an accuracy of about 75%.

d) GradientBoostingRegressor

Gradient boosting builds an additive model in a forward stage-wise fashion and it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. It gives an accuracy of about 81%.

COMPARISON TABLE:

	R2_score	Mean_absolute_error	Mean_squared_error
RFR	0.8861440573087951	714.0794939509094	2430881.013025968
DTR	0.7747869798588183	837.1451099076139	4808410.010114687
KNR	0.7509613040419841	1270.0046511627907	5317100.040662631
GBR	0.811234356475521	1275.9205363678388	4030240.3889430948

4)HYPER PARAMETER TUNING

Next, we apply RandomizedSearchCV to find out the best parameters for all the above models. After hyper parameter tuning we train our new models with the best parameters returned by the randomized search model. After tuning the parameters we observe that the r2_score increased whereas both the type of errors decreased.

Results:

From all the above analysis we choose gradient boosting regressor as the best model as it gives the highest r2_score and less errors. The final accuracy we reported is about 88%.