Predict The Flight Ticket Price

Divyam Patel (B20EE082)

Abstract

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. For the calculation of flight ticket fares, airlines now employ complicated strategies. Because the pricing fluctuates dynamically, this highly intricate system makes it impossible for customers to estimate the flight ticket fare. Therefore, I have tried to construct a model that predicts the approximate ticket price so that customers may determine when the best time to purchase a ticket is.

Index Terms

Flight Ticket, Random Forest, Regression, KNeighbours, Feature Selection, Encoding, Support Vector Regression, XGBoost Regression, r2 score, Gradient Boosting

I. Introduction

THE major goal of this project is to predict travel prices using regression-based Machine Learning algorithms and to assist consumers in finding the best time and pricing to book flights.

II. Data Description And Exploratory Data Analysis

The dataset provides prices of flight tickets for various airlines along with the time of Departure and Arrival. The features included in the dataset 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) and **Price** (The price of the ticket).

Exploratory Data Analysis is primarily used to see what the data can reveal beyond the formal modeling or hypothesis testing task and to provide a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate or not. Since there were a lot of features, it was important to understand the important features and the relationships between them.

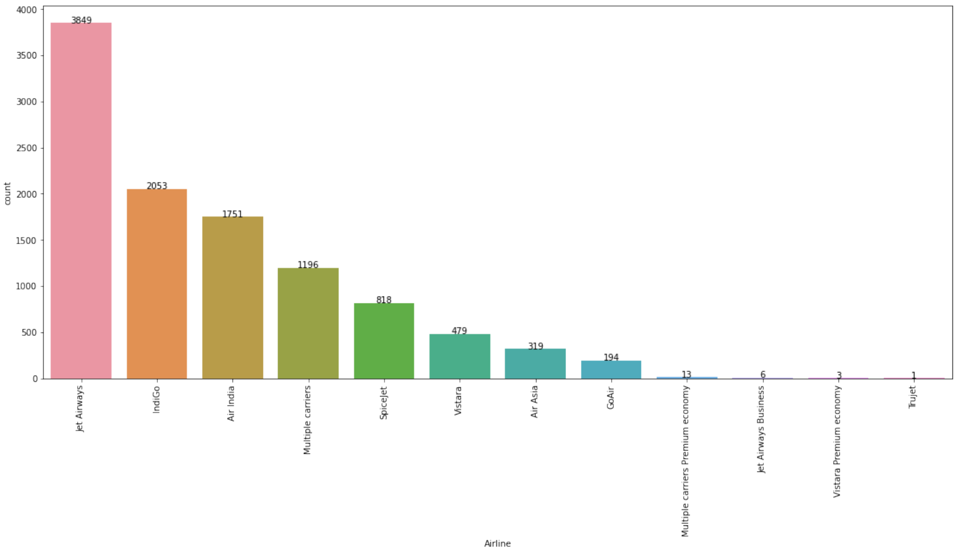
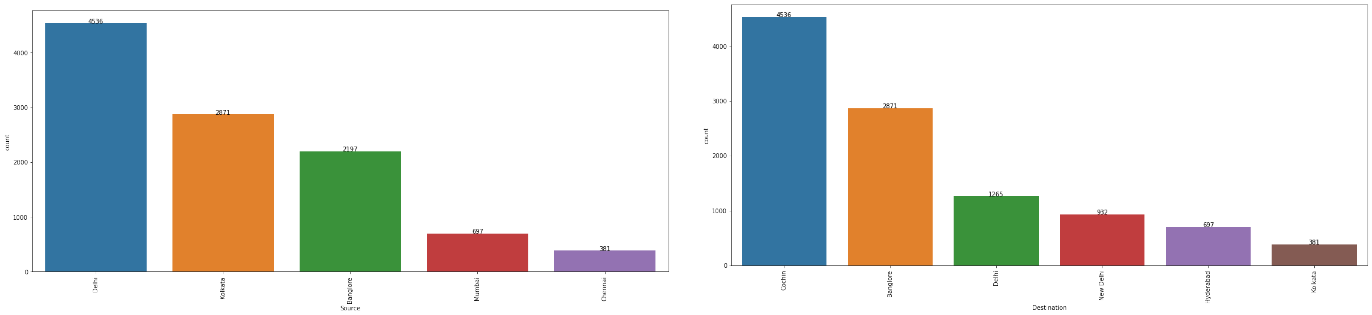


Fig. 1: Count of Airlines



(a) Source

(b) Destination

Fig. 2: Count of Source and Destination

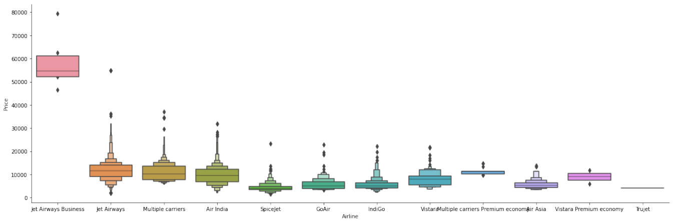


Fig. 3: Plot of Airline vs Price
The figure illustrates Jet Airways Business have the highest Price. Apart from the first Airline almost all are having similar median.

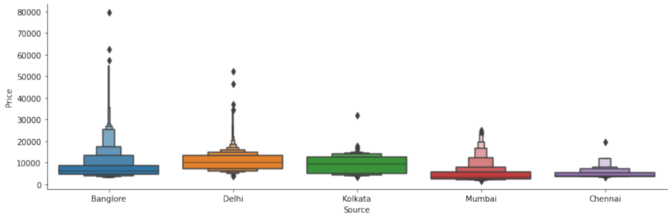


Fig. 4: Plot of Source vs Price

III. Data Preprocessing

1) **Null Value Treatment:** I identified only two missing values overall which were in the columns *Route* and *Total_Stop*s. On exploring a little further I found both the values to be missing in the same row. So, I dropped the row.
2) **Handling Date-Time Data:**
   - *Date_of_Journey* is a object data type. So I converted *Date_of_Journey* to timestamp to use it properly for prediction. Created 2 new features ***Journey_Day*** and ***Journey_Month*** to store the Date and Month of the Journey.
   - Converted *Dep_Time* and *Arrival_Time* to date time format and created 4 new features (***Dep_hour***, ***Dep_min***, ***Arrival_hour***, ***Arrival_min***) to store the hour and minute.
   - Duration is the time taken by plane to reach destination. It is the difference between Departure Time and Arrival time. I converted *Duration* into ***Duration_hours*** and ***Duration_mins***
3) **Handling Categorical Data:**
   - *Total_Stops* is the case of Ordinal Categorical type. So, performed **LabelEncoding** to assign values with corresponding keys.
   - Changed ”Multiple carriers Premium economy”, ”Jet Airways Business”, ”Vistara Premium economy” to ”Multiple carriers”, ”Jet Airways”, ”Vistara” respectively. Further, I dropped ”Trujet” because *Airline* feature has high cardinality and this record is present only 1 time so the model can't learn enough.
   - As *Airline*, *Source* and *Destination* had Nominal Categorical Data, so I performed **OneHotEncoding**
   - *Total_Stops* and *Route* are correlated. So I dropped *Route*. Also, in the Additional_Info column, No info Category is present more than 75 percent of total records this means it is not a significant feature so dropped *Additional_Info*

IV. Machine Learning Models

Before implementing the Machine Learning Models, feature selection was done to find out the best features and have good relation with target variable. There are various feature selection methods that were implemented:

- *heatmap*
- *feature_importance_*
- *SelectKBest*

Implementing the Machine Learning Models:

1) *Random Forest Regressor:* Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Simple Random Forest Regressor gave *R2 score* of *0.76*. So, done the hyperparameter tuning using *RandomizedSearchCV*.

2) *Decision Tree Regressor:* Decision trees are predictive models that use a set of binary rules to calculate a target value. Each individual tree is a fairly simple model that has branches, nodes and leaves.Decision trees regression use mean squared error (MSE) to decide to split a node in two or more sub-nodes.
3) *Gradient Boosting Regressor:* Gradient Boosting Regressor gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.
4) *Naive Bayes - Gaussian:* Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.
5) *XGBoost Regressor:* XGBoost stands for Extreme Gradient Boosting. It uses more accurate approximations to find the best tree model. Implemented XGBoost Regressor along with the hyperparameter tuning using *RandomizedSearchCV*.
6) *Linear Regression:* Linear Regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables)
7) *Support Vector Regressor:* Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.
8) *K Nearest Neighbours:* The k-nearest neighbors algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

V. Evaluation of Models

The models implemented were evaluated using techniques like *r2 score*, *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE*) and *Mean Absolute Error (MAE)*

TABLE I: Evaluation of Models

Model	r2 Score	MAE	MSE	RMSE
XG Boost Regressor	0.83	1224.78	3791100.56	1947.07
Random Forest Regressor	0.78	1186.04	4758302.85	2181.35
K Nearest Neighbours	0.77	1389.15	5048097.63	2246.8
Gradient Boosting Regressor	0.77	1290.97	5214309.93	2283.48
Gaussian NB	0.68	1503.8	7114578.74	2667.32
Decision Tree Regressor	0.59	1434.82	9011197.55	3027.60
Linear Regression	0.58	1979.95	9321759.74	3053.15
Support Vector Regression	0.05	3441.04	20895022.89	4571.1

VI. Conclusion

The table shows the evaluation of all the models implemented. XGBoost Regressor gave the best result.

References

[1] Pattern Classification - Book by David G. Stork, Peter E. Hart, and Richard O. Duda

[2] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[3] https://scikit-learn.org/stable/index.html