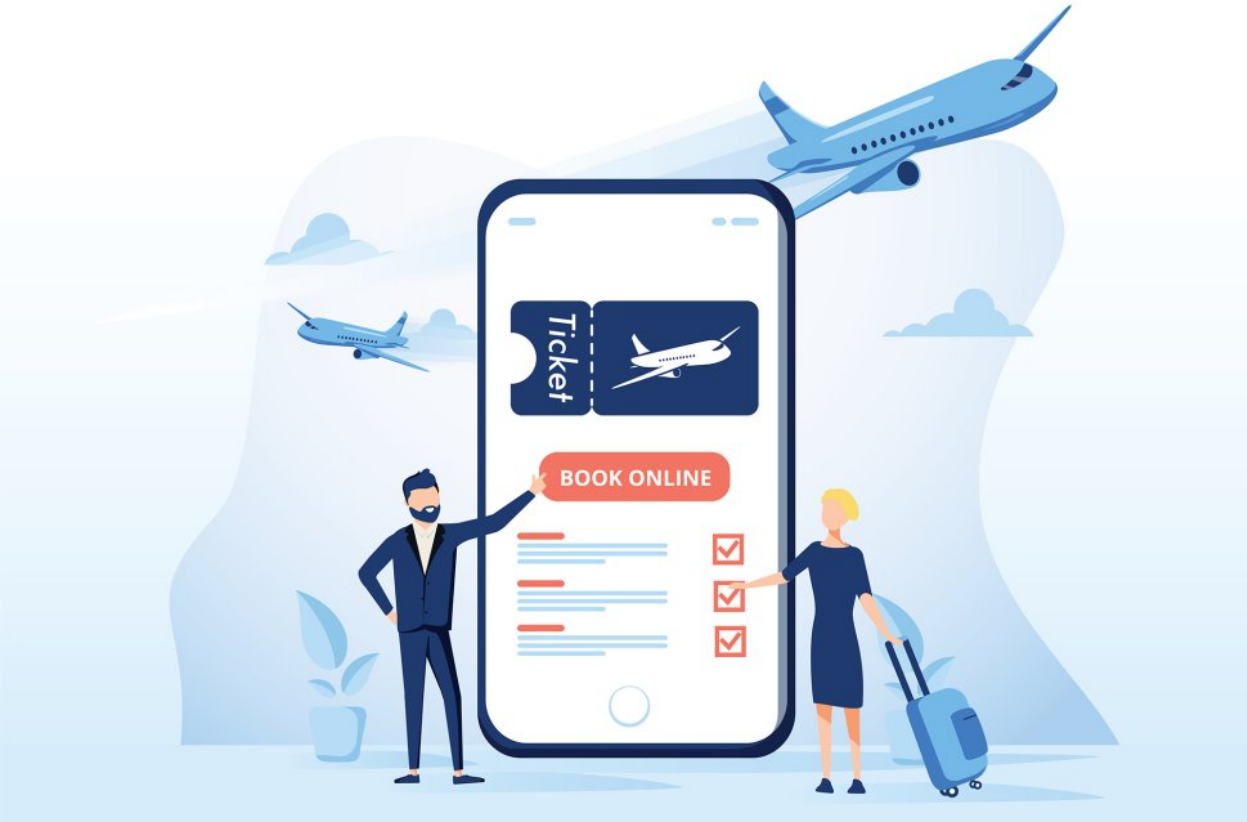
Flight Price Prediction using Machine Learning Techniques



1. Introduction

Now a days, a lot of people are using flights to commute to various places. The flights may be booked for holidays or for a business trip but the demand for air travel is increasing day by day. Airline companies use complex algorithms to calculate flight prices based on the current situations at that time. They analyze various social, financial and market conditions to offer flight price and maintain flight price but the prices change dynamically due to various conditions. Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and 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 in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

So, in this project we collected data of flight fares with other features and worked to make a model to predict fares of flights. In this article we will talk about how to use machine learning to predict the flight prices based on previous booking preferences of the customers. Various data processing techniques and exploratory data analysis have been covered in this article. The performance of models were compared based of different parameters.

2. Data Collection

The data is extracted form makemytrip.com the details of the extracted data are as follows

Size of data set: **97759** records

FEATURES:

Airline: The name of the airline.

Date_enq: The date of the journey

Date_journey: The date of the journey

Source: The source from which the service begins.

Departure: The time when the journey starts from the source

Destination: The destination where the service ends.

Arrival: Time of arrival at the destination.

Duration: Total duration of the flight.

Stops: Total stops between the source and destination.

Route: The route taken by the flight to reach the destination.

LABEL

Price: The price of the ticket

3. Data Preprocessing

The first step in building a model is to understand the dataset and prepare it for efficient prediction. In this step we use various cleaning techniques to produce a normally distributed dataset.

Initially we will import all the necessary libraries which we will use for our analysis.

The next step is to load the csv file which contains our dataset. After loading the first thing which we do is check the dataset and understand the features available in the dataset. We will also check the size of the dataset, number of entries and number of columns. There are two sets of data, one is training set and other is test set. We will fit the training set into different models and check their performances. After the model is finalized we will predict the flight prices for the test set.

We see that there 97759 entries in training set and each of the entries have 10 features and 1 label. Our features are a mixture of categorical features and numerical features. We will identify the categorical features and numerical features and analyze them separately. Let us look at the data description to have an initial look at the variation of data.

3 features popped up using describe() method.

a. Mean flight duration is 4.15 hours, minimum is 1 hour and maximum is 56 hours.

b. 50 % of data has no stops, 25 % have 1 stops and the rest have 2 or 3 stops.

c. Mean price is 4883, minimum is 1443 and maximum is 30202.

This shows that there are some outliers in the dataset. We will remove these outliers. We checked the outliers using distribution plot and the values which have Z score > 2.

After removal of outlier we have 90517 records.

The data which was collected has more than 2 stops. But, only we have information of only 552(547+5) flights which have more than one stops. In order to have a more organised data set. I have considered filtered the data for non-stop and just one stop flights.

After applying the filter we have 89965 records.

From the first look analysis of all the columns it is clear that we need to do feature engineering to find out intermediate stops and day of the week. We will also two features for departure hour and arrival hour of the day.

Lets find out how many missing values are there in the dataset. The features which have missing values are Route. Route is the feature which gives information about the stops. Hence, the values which are missing are for the direct flights. We will feature engineer and remove the missing values.

Let us check the count of different airlines available for booking. We we have flights with multiple stops and multiple carrier. I have separated the flights into Airline 1 and Airline 2.

4. Exploratory Data Analysis

1. Airline 1 vs Price

Direct Flight: Vistara has highest price followed by Spicejet and Indigo Star Air has least price

1 Stop: Star Air has highest price followed by Vistara. Trujet has least price

2. Count of Flights

Direct Flight: There are more number of direct flights from New Delhi followed by Mumbai and Hyderabad
1 Stop: There are more number of connecting flights from Kolkata followed by Ahmedabad and Goa

3. Source vs Price

Direct Flight: The flight price is highest from Kolkata followed by Goa and New Delhi

1 Stop: The flight price is highest from Kolkata followed by Mumbai and Goa

4. Day of Week vs Price

The feature ‘Date of Journey’ contains date. For better analysis let us extract the day of the month for the journey and month of the journey. We did feature engineering and created two new features ‘Journey_day’ and ‘Journey_month’. From the date the day of the week when the flight departs is also added as separate column to analyse flight price for different day of the week. It was found that flight price is highest on Sundays and least on Mondays.

5. Month vs Price

The price for October and November are high as compared to December and January. It is because there are less number of days between the date of enquiry and date of journey. Hence, it appears that the flight prices prices are slashed down or December and January.

6. Departure Period of the day vs Price

The price is higher for flights departing at forenoon followed by the flights which depart past midnight

7. Arrival Period of the day vs Price

The flight prices are highest when the arrival time is past midnight followed by afternoon

5. Model Building

With this we are done with EDA and feature engineering. At this stage we need to encode all the columns which have categorical values. The model requires all input and output variables to be numeric, so we need to convert all categorical features into numerical values using proper encoding techniques. The categories can be encoded using binary encoder as this encodes the given information into a compact form. Binary encoder converts text attributes into numerical values for further processing.

After encoding we have 89965 entries and 34 features in our dataset. The next step in our analysis is to find out the features which are strongly correlated using correlation matrix and heatmap. This will help us to understand the relation between feature vs feature and in feature selection. We will drop the features which are highly correlated.

The feature Duration and Stops are strongly corelated which is kind of obvious as duration will be more for flights which are not non-stop

The following models were tested to predict the approval of loan:

1. Linear Regression
2. Lasso
3. Ridge
4. KNN Regressor
5. Decision Tree Regressor
6. Random Forest Regressor
7. Ada Boost Regressor
8. Gradient Boost Regressor

The following table shows the MSE for all models fitted with our dataset without tuning hyper parameters.

S. No.	Model Name	Mean Squared Error
1.	Linear Regression	1615
2.	Lasso	1615
3.	Ridge	1615
4.	KNN Regressor	973
5.	Decision Tree Regressor	704
6.	Random Forest Regressor	538
7.	Ada Boost Regressor	1390
8.	Gradient Boost Regressor	988

- Decision Tree, Random Forest and Gradient Boost have performed better than other models. Lets tune the hyper parameters and evaluate their performance.

The hyper parameters for the models were tuned using Randomized Search CV technique wherever applicable. The following table summarizes the performance of model:

S. No.	Model Name	Score
1.	Decision Tree Regressor	0.883
2.	Random Forest Regressor	0.889
3	Gradient Boost Regressor	0.83

The performance of Random Forest and Decision Tree is almost same. We will select Random Forest as our final model as it is combination of multiple decision trees.

The hyper tuned parameters for the Random Forest Regressor are:

```
{'n_estimators': 500,
 'min_samples_split': 2,
 'max_features': 'sqrt',
 'max_depth': 20}
```

The RMSE value of our final model is 627 and MAE is 388.

5. Concluding Remarks

We cleaned the data, performed EDA and successfully trained a model and tuned the hyper parameters to predict the flight price for our test set. The Random Forest Regressor proved to be the best model in prediction as the model had the highest score and RMSE was minimum.