Airline Tickets: Observations and Predicting fares of the flight

UTHMAN BELLO MAIGARI

220055347

Uthman-bello.maigari@city.ac.uk

ABSTRACT

Due to the increasing popularity and availability of different air ticket booking channels online these days, passengers are interested in understanding how airline companies set their ticket prices over time. You can do so using a wide range of methods that provide the opportunity to do so. It was proposed that customers buy airline tickets based on estimated costs. These strategies usually involve sophisticated algorithms called Machine Learning (ML) Prediction Models. In this paper, the results of a study are presented as an attempt to provide data-driven insight into intelligent models.

INTRODUCTION

Air travel within the country is becoming increasingly common these days. With multiple online air travel booking outlets, travelers are learning how airline companies decide how much to charge for tickets over time. The process of searching websites for deals and offers is time-consuming for a passenger. Therefore, the cost is determined by a variety of factors. Machine learning models are used in this venture to estimate the cost of flight tickets after some time.

Airlines aim to maximize profits while customers seek the lowest prices. It's a common misconception that consumers should book tickets well in advance to avoid airfare increases as their departure date approaches. However, that's not really the case. Putting in more than they need for a seat allows the customer to end up paying more than they need. We will ultimately be predicting the price of flight from my dataset.

1 ANALYTICAL QUESTION AND DATA

1.1 Analytical Questions

The only major factor I am considering in answering this section is price (to come up with my questions), as that is what I will be predicting in my models. As a result, we have the following analytical questions:

1. What is the daily trend of flights? The analysis requires that I determine how many flights are available in the morning, afternoon, evening, and night.
2. Where do flights have the most departures? It will be necessary for me to know which cities have the highest number of flights in order to conduct this analysis.
3. Do prolonged flights have an impact on the price? As a result, I should investigate whether the price changes with the duration of a flight.
4. Where are the most final destinations concentrated? Consequently, I have to determine which cities generate the most flights.
5. Are there any trends between airlines and prices? As part of my analysis, I will have to identify whether or not there is any correlation between the two. It may be the case that some airlines charge higher or lower rates than others.

1.2 Data

I have sourced my data from Kaggle (which is an online community platform that allows users to find datasets). My data are flight tickets which consist of the following variables: Airline, Date of journey, source, Destination, Route, Departure time, Arrival time, Duration, Total stops, and Price. The shape of the data consists of 10,683 columns and 11 rows. In light of all my analytical questions, I am confident that the data is sufficient for my analysis. Using the Jupyter notebook.

2 ANALYSIS

2.1 Data Preparation

Among the various analytical steps I took in data preparation, data cleansing was the most significant. It is necessary for me to determine if there are any missing values in the data that are null. A number of features in the data had missing values, including Total Stop and Route. Once the rows have been identified, I impute the missing values and update the data frame.

2.2 Data Preprocessing

As part of data preprocessing, a model needs to be able to understand the contents of the data, such as the day, month, years, or hours. Among the various analytical steps I took in data preprocessing, performing outlier detection was the most significant. The detection and removal of outliers in a dataset are fundamental preprocessing steps. The analysis of data can be misleading if outliers are not identified and removed. Due to outliers in the price feature, statistical imputation was necessary.

2.3 Data Derivation and Encoding

The derived data contains supplementary information that is not present in the original data. There are several ways to derive data, including:

1. Extracting data
2. Restructuring data
3. Augmenting data

My derived data has 13 additional features ('journey_day', 'journey_month', 'Dep_Time_hour', 'Dep_Time_minute', 'Arrival_Time_hour', 'Arrival_Time_minute', 'Duration_hours', 'Duration_mins', 'Source_Banglore', 'Source_Kolkata', 'Source_Delhi', 'Source_Chennai', 'Source_Mumbai') that will aid my machine learning model.

As machine learning is primarily about maths and can only work with numbers, vectors, or matrices, I extracted the features from string values. As a result, any feature with string values will have to support numbers at all times which means transforming columns into numerical values to make them easier to work with.

There is no single effective method for encoding categorical values, so I must experiment to find the most appropriate one. Therefore, we encode our data in two different ways: One-Hot Encoding and Label Encoding. If data belongs to Nominal data (i.e. data is not in any order), OneHotEncoder is used in this case. if data belongs to Ordinal data (ie data is in order), LabelEncoder is used in this case

2.4 Construction Of Models

Data is needed for building a basic ML model, which will then be divided into training and testing data. In order for ML to understand the relationships within the data, training data is essential. In order to evaluate the effectiveness of this ML model, we need testing data. There are two models that I have constructed: a Random Forest and a Linear Regression Model. Among the various analytical steps I took in constructing my models, training the model and hypertuning the model were the most significant.

During the machine learning process, training is the most crucial step. When I train my machine learning model, I pass the prepared data to help find patterns and predict things. In this way, the model learns from the data in order to accomplish the task set. Training improves the model's prediction abilities over time.

As I train the model, we have an independent feature and a dependent feature. I consider the price column to be my dependent feature since it depends on other features. Thus, the rest of the columns are independent because they do not depend on anything else. The independent feature will be stored in 'X' and the dependent feature will be stored in 'y'. Hence we have the X_train, X_test, y_train, y_test.

Following this step is hyperparameter optimization, which involves finding every parameter's optimal value. To accomplish this, I will create a dictionary ('dic1') in which I will store the parameters, then initialise RandomiseCV. RandomiseCV generates a randomised search over parameters by sampling each parameter setting from a distribution of possible values. The typical parameters for my random forest model are: *n_estimators': 1120, 'min_samples_split': 15, 'max_features': 'auto', 'max_depth': 21.*

2.5 Validation Of Results

In this section using a validation technique will allow you to see how your machine learning model responds to unknown data. Despite similarities, all validation methods use the train-test split, with slight variations between them. Metrics used for evaluating machine learning models and comparing accuracy are called performance metrics. Based on regression metrics, the sklearn.metrics module implements functions to track the errors in each model. Every model is checked for error using the following metrics:

1. **MAE** (Mean Absolute Error): This is derived by summing the average difference between the estimated value and the actual value, which is known as Mean Absolute Error. MAE is a measure of how well your model performs. The lower it is, the better it will perform.
2. **MSE** (Mean Square Error): By using Mean Square Error, instead of using absolute values, the difference between actual and predicted output values is squared before summing all. A decrease in MSE improves model performance.

3. **RMSE** (Mean Absolute Percentage Error): By taking the square root of the average squared difference between the prediction and the actual value, RMSE can be estimated. A model's performance is better when its RMSE is lower than its MAE. RMSE is more significant than MAE.
4. **MAPE** (Mean Absolute Percentage Error): The MAPE is calculated by dividing the absolute errors (each period separately) by the demand. Percentage errors are averaged. In this case, a lower value is better.
5. **R2 Score**: In statistics, R squared measures how well changes in the independent variables can be predicted by changes in the dependent variables. In order to have a more precise regression, one that has a fairly high R squared (close to 1) is best.

The below shows my models and their Training accuracy, error metrics:

RANDOM FOREST

Name	Accuracy/Metric Error
Training Accuracy	0.9519794499765041
r2_score	0.8061699847279256
MSE	3773405.140460956
MAE	1186.5471133711021
RMSE	1942.5254542633297
MAPE	13.286563186944422

Table 1: Random Forest Performance metric.

LINEAR REGRESSION

Name	Accuracy/Metric Error
Training Accuracy	0.5794483128817276
r2_score	0.555713367195872
MSE	2061.784107595206
MAE	8649194.304132724
RMSE	2940.951258374189
MAPE	24.94861105877553

Table 2: Linear Regression Performance metric.

As shown in the tables the training accuracy for random forest is far superior to the linear regression. We can conclude from all the performance metrics that Random Forest will provide us with accurate and better results. As a result, we can deploy Random Forest models.

The below shows the performance metric after hyper-tuning the random forest model:

HYPER-TUNED RANDOM FOREST

Name	Accuracy/Metric Error
Training Accuracy	0.9516275830922576
r2_score	0.8283998807085746
MSE	3340642.4249055255
MAE	1152.027121877014
RMSE	1827.7424394332822
MAPE	12.909939018229052

Table 3: Hyper-tuned RF Model.

As shown in *table 3* the performance of this model has improved significantly after hyper parameter optimization.

3 Findings

3.1 Trends of Flights in a day.

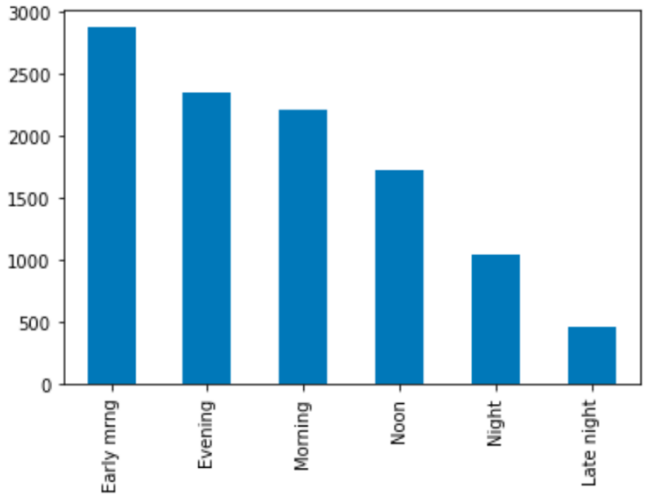

Figure 1: visualisation of the number of flights in each category

This figure shows a visual representation of flights within a day. As a result, we can determine whether flights leave early in the morning, late in the evening, or late at night. We can see that most flights occurred early in the morning with 2,880 flights; the least flights happened late at night with 465 flights.

3.2 Airports with the most number of departures.

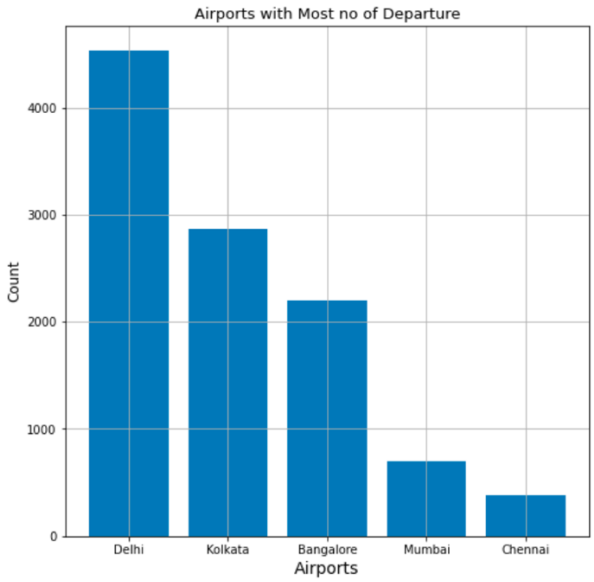

Figure 2: visualisation of the cities with majority of departure time

According to the graph above, Delhi has the most departures (making it the busiest) with 4536. In the city of Cheenai, there are only 381 departures, perhaps because Delhi is the capital of India.

3.3 Impact on price due to duration of flight

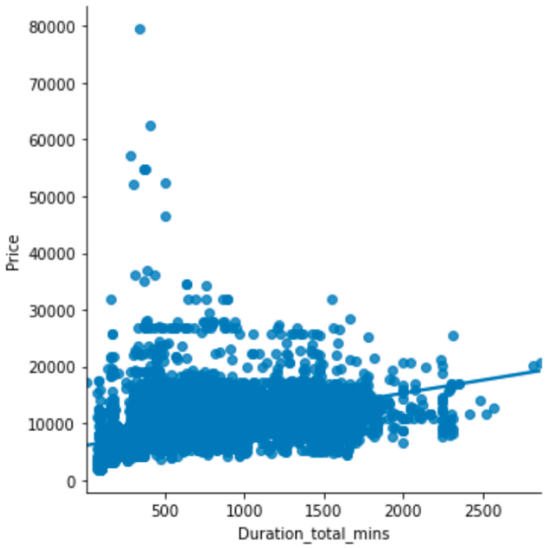

Figure 3: visualisation shows a plot data/regression of price and duration

There is a clear correlation between the price of flights and the duration of minutes, as the graph shows. With the increase in minutes, there is also an increase in the price of flights.

3.4 Cities with the highest number of flights.

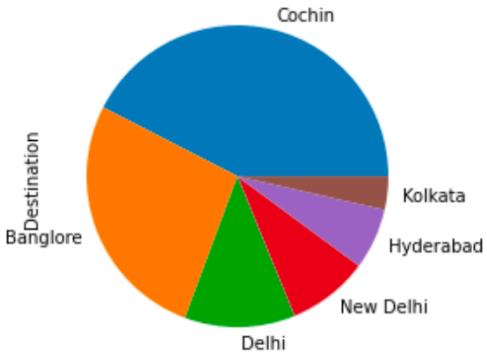

Figure 4: visualisation in pie chart showing each destination

The city with the highest flight is Cochin (4536 flights) and the city with the least flight is Kolkata (381 flights)

3.5 Comparison between airline and price

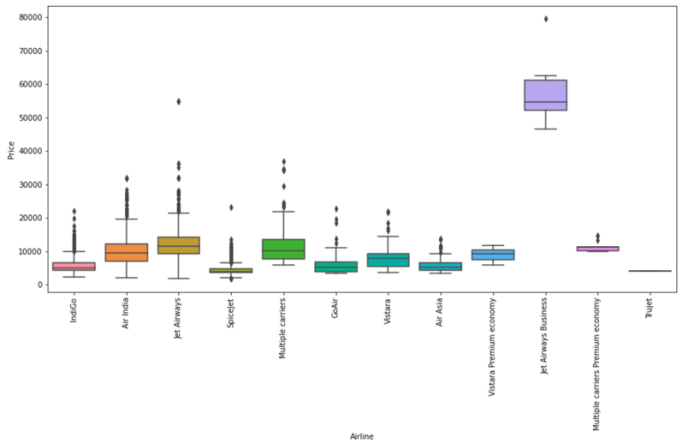

Figure 5: visualisation in distribution plot which shows airline and price.

According to the correlation between airline and price data, most airlines charge the same prices with the exception of Jet Airways Business Class, which tends to be higher than other airlines.

4 REFLECTION AND FUTURE WORK

This project might save money for less experienced travellers as it gives them information regarding trends in flight prices as well as a predicted value of the price. This is something they can use to decide whether to buy a ticket now or in the future. During the process of testing different models, the Random Forest algorithm was found to give the highest accuracy in predicting output as compared to other models.

However, due to the nature of the datasets, there are a number of limitations that should be taken into account when using these techniques. The datasets do not include any detailed information related to ticket sales, such as the time and day of departure and arrival. In addition, they do not include any additional details related to ticket sales.

Air ticket transaction data can eventually be added to the framework. This could provide more details about an itinerary, such as departure and arrival times, seat locations, ancillary products covered, etc. Such data can be combined with existing macroeconomic and market segment information to develop a daily or even hourly airfare price prediction model.

5 REFERENCES

[1] Craig Stedman. Article on data preparation (published feb 2022) https://www.techtarget.com/searchbusinessanalytics/definition/data-preparation

[2] Michael Halarnyk. Article on Train Test Split (punblished jul 28 2022) https://builtin.com/data-science/train-test-split

[3] Mayank Banoula. Article on Machine learning (published Nov 15 2022) https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

[4] Samira Pouyanfar, Tianyi wang. Paper on 'Framework for airline price pridiction"

[5] Zach. Journal on MSE and RMSE (published sep 30 2021) https://www.statology.org/mse-vs-rmse/

[6] Kartikrath, Anubhav Kumar. Paper on airline price prediction (published may 2022) https://www.irjet.net/archives/V9/i5/IRJET-V9I5571.pdf