**FLIGHT PRICE PREDICTION**

**TEAM: 16**

Mary Pranavi, Allam – 1002077034

Sri Lasya, Kommareddy – 1002082513

Rajashekhar Reddy, Mucharla – 1002027966

# **TOPIC INTRODUCTION**

Traveling by airline is common nowadays. The flight ticket price is a significant factor in traveling as price changes due to various factors like timing of the flights, destination, duration of flights, and various occasions such as vacations or festive season. It is easy for people who regularly travel to find the best price, but for the one who travels occasionally, it is difficult to find the best price, so they end up paying high prices.

In this project, a predictive model will be created using machine learning algorithms to predict the flight prices for various flights. We will be analyzing the flight fare prediction using machine learning dataset using essential exploratory data analysis techniques then will draw some predictions about the price of the flight based on some features.

# **DATASET**

Data was used from Kaggle.

Source: <https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh>

The dataset contains 11 input features. The output column “price” should be predicted on this set. We use a regression technique here because the predicted output will be continuous. The features available in the dataset are Airline, Date\_of\_Journey, Source, Destination, Route, Dep\_Time, Arrival\_Time, Duration, Total\_Stops, Additional\_Info, and Price

We are using jupyter notebook to run Flight Price Prediction task.

* **Airline:** So this column will have all the types of airlines like Indigo, Jet Airways, Air India, and many more.
* **Date\_of\_Journey:** This column will let us know about the date on which the passenger’s journey will start.
* **Source:** This column holds the name of the place from where the passenger’s journey will start.
* **Destination:** This column holds the name of the place to where passengers wanted to travel.
* **Route:** Here we can know about that what is the route through which passengers have opted to travel from his/her source to their destination.
* **Dep\_Time:** Departure time is when the passenger will depart from his/her source.
* **Arrival\_Time:** Arrival time is when the passenger will reach his/her destination.
* **Duration:**Duration is the whole period that a flight will take to complete its journey from source to destination.
* **Total\_Stops:** This will let us know in how many places flights will stop there for the flight in the whole journey.
* **Additional\_Info:** In this column, we will get information about food, kind of food, and other amenities.
* **Price:** Price of the flight for a complete journey including all the expenses before onboarding.

# **METHOD**

Our approach involves a comprehensive methodology, encompassing Data analysis, Exploratory Data Analysis (EDA), data visualization, and feature engineering. Each step is meticulously crafted to extract meaningful patterns from the dataset and enhance the predictive power of our machine learning models.

* Data Analysis: Extracting valuable information using mathematical techniques.
* Exploratory Data Analysis (EDA): Uncovering insights from the dataset.
* Data Visualization: Creating visual representations for better understanding.
* Feature Engineering: Enhancing model performance through innovative feature creation.
* Machine Learning Models: Deploying regression models for accurate flight price predictions.

# **CODE SNIPPETS**

* **STEP 1 : Importing Libraries**

A screenshot of a computer

Description automatically generated

* **STEP 2 : Reading the training data of our dataset**





* **STEP 3 : Data Analysis**



After running the above code you will get a report as shown in the below figure. This report contains various sections or tabs.  ‘Overview’ section of this report provides us with all the basic information of the data we are using. For the current data we are using we got the following information:

A screenshot of a computer

Description automatically generated

Let’s explore other sections of the report one by one.

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A graph and a chart

Description automatically generated with medium confidence

Missing Values : This section has multiple ways using which we can analyze missing values in variables. We will discuss three mostly used methods, bar-chart, spectrum, and Heat Map.

* The bar chart method shows the ‘number of missing and present values’ in each variable.
* The spectrum method shows the percentage of missing values in each variable.
* The heat Map method shows variables having missing values in terms of correlation. Since ‘Route’ and ‘Total\_Stops’ both are highly correlated, they both have missing values.

A graph with a red square

Description automatically generated

As we can observe ‘Route’ and ‘Total\_Stops’ variables have missing values. Since we did not find any missing values information from Bar-Chart and Spectrum method but we found missing value variables using the Heat Map method. Combining both information, we can say that the ‘Route’ and ‘Total\_Stops’ variables have missing values but are very low.

* **STEP 4 : Exploratory Data Analysis (EDA)**

Now here we will be looking at the kind of columns our dataset has.

A screenshot of a computer

Description automatically generated

Now while using the IsNull function we will gonna see the number of null values in our dataset and using the IsNull function and sum function we will gonna see the number of null values in our dataset

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedDropping NAN values and Duplicate values

A screenshot of a computer

Description automatically generated

Here we will be removing those repeated values from the dataset and keeping the in-place attribute to be true so that there will be no changes.

A screenshot of a computer

Description automatically generated

* **STEP 5 : Data Visualization**

Plotting Price vs Airline plot

A screen shot of a graph

Description automatically generated

Plotting Price vs Source plot

**A screenshot of a graph

Description automatically generated**

Plotting Price vs Destination plot

A screenshot of a graph

Description automatically generated

* **STEP 6 : Data Extraction**

**A screenshot of a computer code

Description automatically generated**

We are dividing the features and labels. We have duration information on the ‘Duration’ variable. This variable contains both duration hours and minutes information combined. We are converting the hours in minutes.

We can extract ‘Duration\_hours’ and ‘Duration\_minutes’ separately from the ‘Duration’ variable.

We have ‘Date\_of\_Journey’, a ‘date type variable and ‘Dep\_Time’, ‘Arrival\_Time’ that captures time information.

Here we are organizing the format of the date of journey in our dataset for better preprocessing in the model stage.

We can extract ‘Journey\_day’ and ‘Journey\_Month’ from the ‘Date\_of\_Journey’ variable. ‘Journey day’ shows the day of the month on which the journey was started.

Similarly, we can extract ‘Departure\_Hour’ and ‘Departure\_Minute’ as well as ‘Arrival\_Hour and ‘Arrival\_Minute’ from ‘Dep\_Time’ and ‘Arrival\_Time’ variables respectively.

After final preprocessing let’s see our dataset

A screenshot of a computer

Description automatically generated

Just checking the correlation between different features of training data.

We can see that Total\_stops is highly correlated with Duration\_hours which is very obvious. If the no. of stops would increase, the duration hours of the flight will also increase.

Also, price is highly correlated with total stops because if stops would increase that would also require a high quantity of fuel, and that would increase the price.

A screenshot of a screen

Description automatically generated

* **STEP 7: Feature Engineering**

Airline, Source, Destination, Route, Total\_Stops, Additional\_info are the categorical variables we have in our data. Let’s handle each one by one.

As the name of the airline matters. ‘JetAirways Business’ has the highest price range. Other airlines price also varies.

Since the Airline variable is Nominal Categorical Data (There is no order of any kind in airline names) we will use one-hot encoding to handle this variable.

Again ‘Source’ and ‘Destination’ variables are Nominal Categorical Data. We will use One-Hot encoding again to handle these two variables.

Route variable represents the path of the journey. Since the ‘Total\_Stops’ variable captures the information if the flight is direct or connected so I have decided to drop this variable.

Here in Total stop Variable, non-stop means 0 stops which means direct flight. Similarly meaning other values is obvious. We can see it is an Ordinal Categorical Data so we will use Label Encoder here to handle this variable.

Now this variable is also Nominal Categorical Data. Let’s use One-Hot Encoding to handle this variable.

Now we will create the final dataframe by concatenating all the One-hot and Label-encoded features to the original dataframe. We will also remove original variables using which we have prepared new encoded variables.

So, we have 29 variables in the final dataframe including the dependent variable ‘Price’. There are only 28 variables for training. Dropping the Price column as it is of no use.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

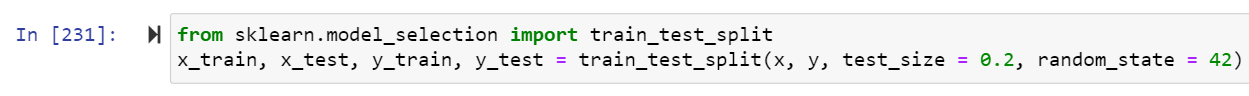
Description automatically generated

* **STEP 8 : Model Building**

For model building, we will use regression algorithms of supervised learning as it is a problem of predicting a value. We will see them in the next section.

# **EXPERIMENTS AND RESULTS**

Firstly, we have to split data into a training set and a test set before model validation, so that we would be able to develop and optimize the model (using the training data) before testing its performance on previously unseen data (the test data).



* RandomForestRegressor

RandomForestRegressor is a strong choice for accurate flight price prediction, with potential for further optimization and validation.

A screenshot of a computer program

Description automatically generatedA graph with blue and orange lines

Description automatically generated

* XGBRegressor

XGBRegressor stands out as a top-performing model, offering high accuracy and generalization. Further fine-tuning and validation are recommended for sustained effectiveness.

A screenshot of a computer program

Description automatically generated A graph with blue and orange lines

Description automatically generated

* KNeighborsRegressor

KNeighborsRegressor is considered, but its performance may benefit from further exploration and optimization.

A screenshot of a computer program

Description automatically generated A graph of blue and orange lines

Description automatically generated

* DecisionTreeRegressor

DecisionTreeRegressor is effective but requires careful tuning to balance model complexity and prevent overfitting. Further optimization may enhance its performance.

A screenshot of a computer program

Description automatically generatedA graph with blue and orange lines

Description automatically generated

* Comparing Regression Models

We could see here that which Regressor has high accuracy for training and testing data.

A screenshot of a computer

Description automatically generated

# **DISCUSSION**

In this analysis, we evaluated the performance of various regression models for predicting flight prices based on provided features. RandomForestRegressor and XGBRegressor emerged as standout performers, showcasing high training and test scores. Their ensemble and gradient boosting nature, respectively, allowed them to effectively capture complex patterns in the data. In contrast, KNeighborsRegressor displayed moderate predictive capabilities, while DecisionTreeRegressor exhibited signs of overfitting.

The visual representation of predicted versus actual flight prices for a subset of observations reinforced the superiority of RandomForestRegressor and XGBRegressor in aligning with the true values. These models not only demonstrated accuracy in learning from the training data but also generalization to the test set.

While RandomForestRegressor and XGBRegressor are recommended for their robust performance, it's essential to acknowledge that further fine-tuning, hyperparameter optimization, and ongoing monitoring are necessary to ensure sustained effectiveness. Additionally, considerations for model interpretability and computational efficiency may influence the choice between these two top-performing models.