Graduate Certificate in Artificial Intelligence with Machine Learning
AIGC 5002 - Machine Learning and Deep Learning
Fall 2023

FLIGHT PRICE PREDICTION

Due: Dec. 12, 2023

Team Members:
Nevin Toms (N01630442)

Submitted To:
Dr Ibrahim Tamim

AI Tools usage:
Chatgpt: Used to get the code syntaxes.
Quillbot: Used to paraphrase the sentences in a professional way.

Introduction

Background

Due to the erratic nature of ticket rates, travellers frequently face difficulties when booking flights in the modern airline sector. Numerous elements, such as the time of departure, the number of stops, seasonal fluctuations, and other pertinent details, affect these costs. Setting competitive initial fares is another challenge faced by recently established airlines. It is difficult for both customers and airlines to make well-informed judgements due to the volatile nature of flight prices.

Problem Statement

1. New airlines are having trouble setting competitive introductory prices.
2. It's challenging for customers to determine whether the price they see is elevated or not.

Objectives

The objective is to develop a predictive model integrating features like departure location, arrival time and many other features for predicting the flight price, based on historical data, and optimise accuracy through algorithm selection and fine tuning.

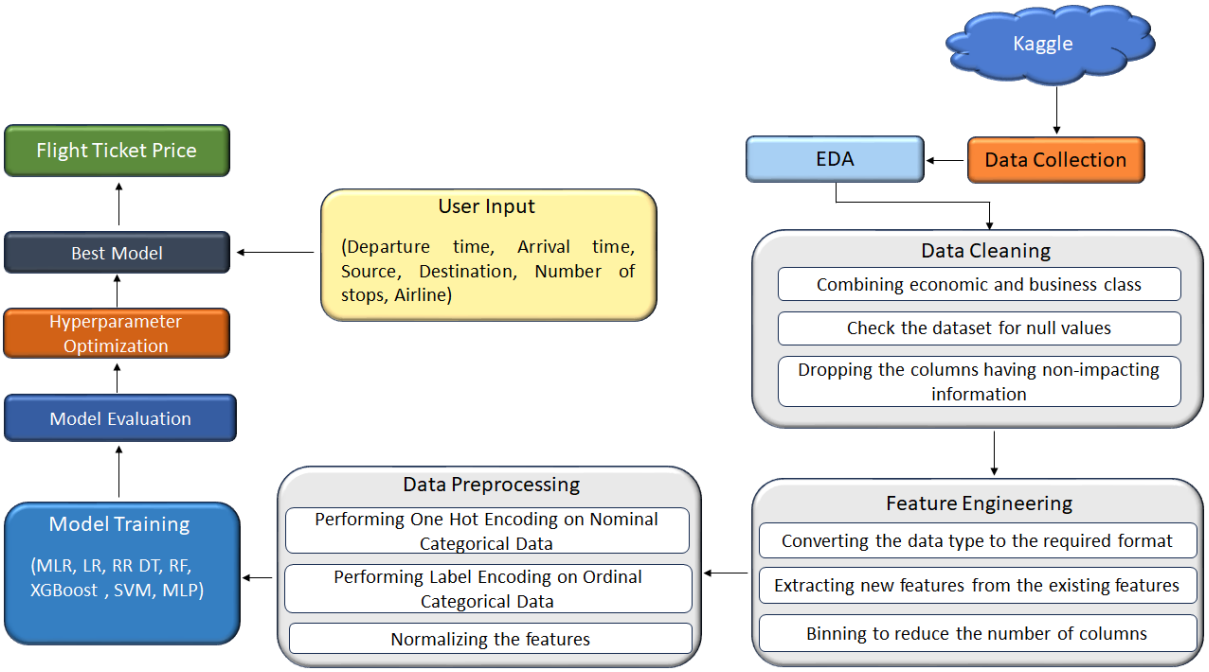
Significance

If the model could accurately forecast prices, it can be utilised to bring up competitive pricing, drawing in customers and ultimately growing market share.

Impact

The flight price prediction model offers significant value by empowering airlines to optimise pricing strategies, resulting in increased revenue and enhanced customer satisfaction. This also provides a competitive edge by enabling airlines to offer attractive prices, improving market share.

Methodology



Data Source

The dataset used in this project is from Kaggle. The dataset in Kaggle was collected in two parts: one for economy class tickets and another for business class tickets Data was collected for 50 days, from February 11th to March 31st, 2022. The business class dataset consists of 93,484 records and 11 features and the economy class dataset consists of 206,774 records and 11 features.

Data Preprocessing

- Airline tickets from business class were renamed to “airline_name” + “Business”, this was done because all the other features will remain the same and renaming this can help the model to understand this was a business class ticket.
- Features “date”, “dep_time”, “arr_time” were in string and were converted to datetime columns. This will help me to extract day, month, year, hour, minute from them.
- Feature “time_taken” was in string format and needed to be converted into integer format (for eg: 2h 30min should be converted to 150)
- Unwanted characters were eliminated from feature "stop" in order to preserve column-wide value consistency. Similarly target column “price” had “object” as dtype which was converted to integer after cleaning.

- The columns "Ch_code" and "num_code" were eliminated because they didn't provide good information for modelling.
- In order to reduce the number of columns during one hot encoding and to capture the variance in flight prices, "Dep_time" and "arr_time" were binned to "Early Morning," "Morning," "Noon," and "Night." and the column was named "time_of_day"
- The columns "airline", "from", "to", "time_of_day" were one hot encoded and "stop" was label encoded.

Model Selection

I have tried 9 different ML models(Multi Linear Regression, Lasso Regression, Ridge Regression, Decision Tree, Random Forest, XGBoost, Bagging Regressor, Support Vector Machine, Multilayer Perceptron), out of which MLP performed the best with MAE of 2315 followed by Random Forest with MAE of 2318.

Training Process

The data was divided into 80% for training and 20% for testing. Under the specified conditions, MLP performed the best; however, I was unable to execute the hyperparameter tweaking for MLP because of time constraints and model complexity, so I instead performed it for Random Forest which was the second best performing model.

Implementation

Code Structure

Extracting hour and minute out of the time columns and day and month out of the date columns.

Feature Creation

In [113]:
```
# Convert date to pandas date format and extracting day and month
df["date"] = pd.to_datetime(df["date"], dayfirst=True)
df["travel_day"] = df["date"].dt.day
df["travel_month"] = df["date"].dt.month
```

In [114]:
```
# Convert time to pandas time format and extracting hour and minute
df["dep_time"] = pd.to_datetime(df["dep_time"])
df["departure_hour"] = df["dep_time"].dt.hour
df["departure_minute"] = df["dep_time"].dt.minute
```

In [115]:
```
# Convert time to pandas time format and extracting hour and minute
df["arr_time"] = pd.to_datetime(df["arr_time"])
df["arrival_hour"] = df["arr_time"].dt.hour
df["arrival_minute"] = df["arr_time"].dt.minute
```

In [116]:
```
# Below function is to correct the values in column "time_taken" which had a different format

def convert_time_to_same_format(x):
    time_taken_split_arr = x.split(".")
    if len(time_taken_split_arr) > 1:
        string = time_taken_split_arr[0] + "h " +  time_taken_split_arr[1] + "m"
        return string
    return x

df["time_taken"] = df["time_taken"].apply(lambda x: convert_time_to_same_format(x))

df["duration_in_mins"] = df["time_taken"].apply(lambda x: int(x.split(" ")[0][:-1])*60 + int(x.split(" ")[1][:-1]))
```

Extracting the weekend feature to improve the model's clarity by saying to raise the price slightly if it is a weekend.

In [28]:
```
df["Weekend"] = (df["date"].dt.dayofweek > 4).astype(int)
```

In [29]:
```
def get_time_of_day(x):
    if ((x >= 0) & (x < 6)):
        return "Early Morning"
    elif ((x >= 6) & (x < 12)):
        return "Morning"
    elif ((x >= 12) & (x < 18)):
        return "Noon"
    else:
        return "Night"

df["departure_time_of_day"] = df["departure_hour"].apply(lambda x: get_time_of_day
df["arrival_time_of_day"] = df["arrival_hour"].apply(lambda x: get_time_of_day(x))
```

Converting categorical columns to numerical columns using One-Hot Encoding

In [166]:
```
df_encoded = pd.get_dummies(df, columns=['airline', 'from', 'to', 'departure_time_of_day', 'arrival_time_of_day', 'travel_month',
                                         'travel_day'], prefix=['airline', 'source', 'destination', 'departure_time_of_day',
                                                                'arrival_time_of_day', 'travel_month', 'travel_day'])
```

Splitting of the data

Splitting the data into training and testing

In [170]:
```
X = df_encoded.drop("price", axis=1)
y = df_encoded["price"]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

Feature Scaling

Normalising the features using MinMaxScaler.

In [171]:
```
# Normalisation using MinMax Scaler

scaler = MinMaxScaler()

scaled_data = scaler.fit_transform(X_train)
df_train_scaled_normalisation = pd.DataFrame(scaled_data, columns=X_train.columns)

scaled_data = scaler.transform(X_test)
df_test_scaled_normalisation = pd.DataFrame(scaled_data, columns=X_test.columns)
```

MLP

In training, MLP gave the best performance

In [130]:
```
%%time

from sklearn.neural_network import MLPRegressor

model_mlp = MLPRegressor(hidden_layer_sizes=(128, 64, 64), max_iter=1000, random_state=42)
model_mlp.fit(df_train_scaled_normalisation, y_train)

# Make predictions on the test set
y_pred_mlp = model_mlp.predict(df_test_scaled_normalisation)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred_mlp)
print(f'Mean absolute Error: {mae:.2f}')
```

```
Mean absolute Error: 2315.63
```

Results and Discussions

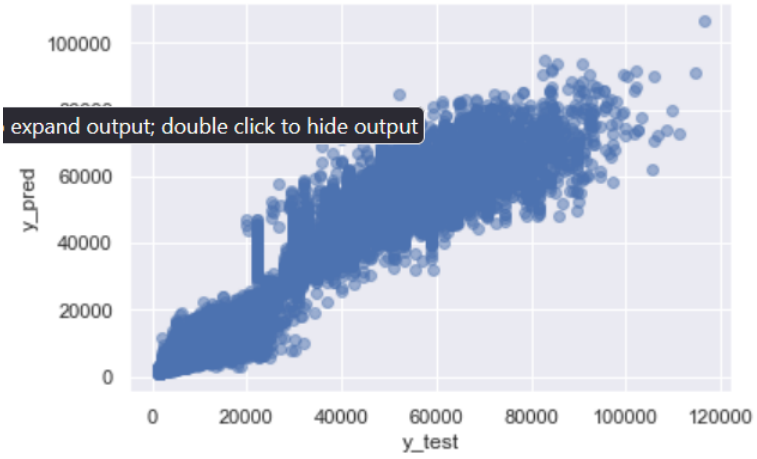
Model Performance

I have tried 8 different algorithms as shown in the figure. Only by looking at the MAE I could say that MLP performed the best followed by Random Forest.

	R2 SCORE	MAE	TIME	
Multi Linear Regression	0.91	4266	473ms	
Lasso Regression	0.91	4265	14s	
Ridge Regression	0.91	4265	170ms	
Decision Tree	0.95	2392	3s	
Random Forest	0.96	2318	3min	2
XGBoost	0.96	2354	2s	3
Support Vector Machine	0.83	5562	1h 19min	
Multi Layer Perceptron	0.97	2315	39min	1

Visualisations

The graph between y_test and y_pred looks like this. Since the model was not entirely accurate in predicting every point, some points are spread. If the model had anticipated every point, all of the points would have been on the diagonal line.



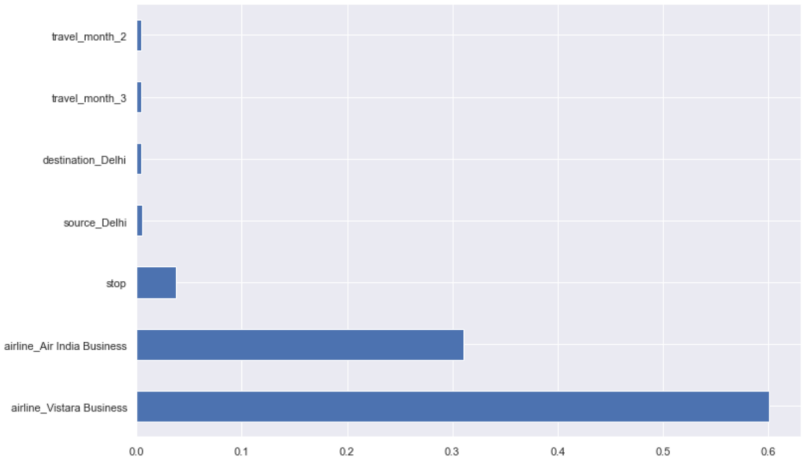
Interpretation

1. Model Performance:

- MAE was the metric used to evaluate the performance of each model.
- Multi-Layer Perceptron (MLP) demonstrated the best performance, followed by the Random Forest model.
- Provide specific values of the metrics to quantify the performance differences.
- MAE Values
    - MAE(MLP) - 2315
    - MAE(RF) - 2318

2. Feature Importance:

Airline_vistara_business and airline_air_india_business were two most important features for modelling



Ethical Concerns

Developing a flight price prediction project raises ethical concerns, particularly regarding transparency and consent. Full awareness of users about data usage for price prediction is crucial, requiring explicit consent. Clear communication ensures users can make informed decisions about participating in the predictive model.

Conclusion

Summary of Findings and limitations:

Our flight price prediction project, utilising MLP and Random Forest models, provides valuable insights for refining airline pricing strategies. Despite achieving commendable accuracy with factors like departure location and arrival time, the model's success is influenced by industry volatility and reliance on historical data. Acknowledging limitations, such as challenges in adapting to market shifts and potential biases in historical data, the model prioritises ethical and equitable performance through continuous bias monitoring.

Future Work:

In future research, exploring advanced models, ensemble methods, and real-time data feeds aims to enhance prediction accuracy. Crucially, addressing data scarcity for less-travelled routes using imputation or transfer learning is essential. Ethical considerations will guide work, emphasising responsible AI frameworks, transparency, and user consent. Collaborating with regulatory bodies and industry stakeholders is key to aligning predictive models with evolving ethical standards for a positive impact on airlines and travellers.