Flight Price Prediction Project

Kagan Heper

December 2024

Overview

This is the second assignment in the 'Data Science: Capstone' course (PH125.9x) given by edX HarvardX. The goal of this project is to use a publicly available dataset to apply machine learning approaches that go beyond simple linear regression while also clearly communicating the process and insights acquired from the study.

Introduction

In this project, I aim to predict flight prices using a variety of machine learning methods. The dataset utilised, **Flight Price Prediction** by Shubham Bathwal from **Kaggle**, contains detailed flight information such as departure_time, arrival_time, airline, and other pertinent characteristics. The purpose is to extensively investigate the dataset, gather insights from exploratory data analysis, and use supervised machine learning models to estimate flight costs based on the provided attributes. This study focusses not only on developing prediction models, but also on understanding the connections between attributes and how they affect travel pricing. This study demonstrates the complete data science pipeline, from preprocessing and visualisation to model construction and assessment.

Data Preprocessing

This part focusses on getting the dataset ready for analysis and modelling. Data preparation stages include replacing commas in the price column and converting it to a numeric format to ensure numerical consistency. Furthermore, categorical variables such as airline, source_city, and departure_time are turned into factors for more effective analysis and modelling. The dataset is then cleaned and formatted for further studies, ensuring that any missing or incorrect values are handled correctly. These pretreatment techniques are necessary to guarantee that the dataset is appropriate for both exploratory data analysis and machine learning activities.

Loading Dataset

The dataset is put into the R environment and examined to determine its structure and content. This entails reading the **Clean_Dataset.csv** file and presenting the first few rows of data to ensure that it was imported properly. The dataset contains a variety of attributes that capture flight facts, including the airline, source and destination cities, class, stops, travel duration, and days till departure. This stage establishes the groundwork for further investigation and analysis.

```
# URL of the dataset
url <- "https://raw.githubusercontent.com/QuantumForgeX/HarvardX_Data_Science_Professional_Certificate/

# Load the dataset directly into R
dataset <- read.csv(url, stringsAsFactors = FALSE)

# Replace commas in 'price' and convert to numeric
```

```r
dataset$price <- as.numeric(gsub(",", "", dataset$price))

# Convert categorical variables to factors
categorical_cols <- c("airline", "source_city", "destination_city",
                      "departure_time", "arrival_time", "class", "stops")
dataset[categorical_cols] <- lapply(dataset[categorical_cols], as.factor)

head(dataset)
```

```
##   serial  airline  flight source_city departure_time stops  arrival_time
## 1      0 SpiceJet SG-8709       Delhi        Evening  zero         Night
## 2      1 SpiceJet SG-8157       Delhi  Early_Morning  zero       Morning
## 3      2  AirAsia  I5-764       Delhi  Early_Morning  zero Early_Morning
## 4      3  Vistara  UK-995       Delhi        Morning  zero     Afternoon
## 5      4  Vistara  UK-963       Delhi        Morning  zero       Morning
## 6      5  Vistara  UK-945       Delhi        Morning  zero     Afternoon
##   destination_city   class duration days_left price
## 1           Mumbai Economy     2.17         1  5953
## 2           Mumbai Economy     2.33         1  5953
## 3           Mumbai Economy     2.17         1  5956
## 4           Mumbai Economy     2.25         1  5955
## 5           Mumbai Economy     2.33         1  5955
## 6           Mumbai Economy     2.33         1  5955
```

Exploratory Data Analysis

The exploratory data analysis (EDA) part gives a thorough comprehension of the dataset via statistical summaries and visualisations. It starts with an overview of the dataset, emphasising the distribution of major variables. Multiple visualisations are used to investigate various elements of the data, including the distribution of flight costs, average prices by airline, and the link between flight pricing and categorical variables such as class and stop. This section also dives into numerical factors such as flight duration distribution and the effect of remaining days on ticket pricing. EDA is critical in influencing feature selection and modelling decisions because it identifies patterns and linkages in data.

```r
# Summary of the dataset
summary(dataset)
```

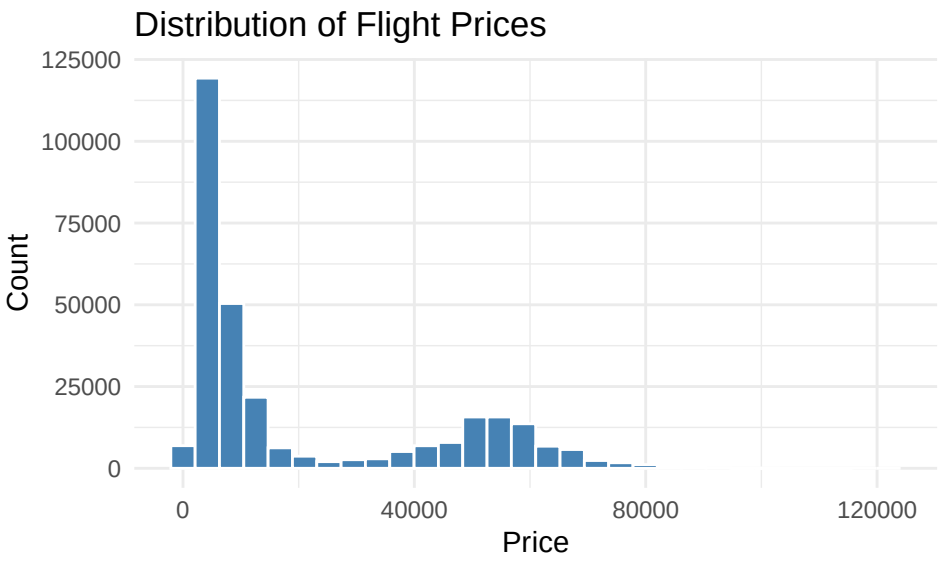
```
##      serial            airline          flight             source_city
##  Min.   :     0   Air_India: 80892   Length:300153      Bangalore:52061
##  1st Qu.: 75038   AirAsia  : 16098   Class :character   Chennai  :38700
##  Median :150076   GO_FIRST : 23173   Mode  :character   Delhi    :61343
##  Mean   :150076   Indigo   : 43120                      Hyderabad:40806
##  3rd Qu.:225114   SpiceJet :  9011                      Kolkata  :46347
##  Max.   :300152   Vistara  :127859                      Mumbai   :60896
##        departure_time          stops              arrival_time
##  Afternoon    :47794   one        :250863   Afternoon    :38139
##  Early_Morning:66790   two_or_more: 13286   Early_Morning:15417
##  Evening      :65102   zero       : 36004   Evening      :78323
##  Late_Night   : 1306                        Late_Night   :14001
##  Morning      :71146                        Morning      :62735
##  Night        :48015                        Night        :91538
##   destination_city      class           duration       days_left
##  Bangalore:51068   Business: 93487   Min.   : 0.83   Min.   : 1
##  Chennai  :40368   Economy :206666   1st Qu.: 6.83   1st Qu.:15
```

```
##  Delhi    :57360                    Median :11.25   Median :26
##  Hyderabad:42726                    Mean   :12.22   Mean   :26
##  Kolkata  :49534                    3rd Qu.:16.17   3rd Qu.:38
##  Mumbai   :59097                    Max.   :49.83   Max.   :49
##      price
##  Min.   :  1105
##  1st Qu.:  4783
##  Median :  7425
##  Mean   : 20890
##  3rd Qu.: 42521
##  Max.   :123071
```

Distribution of Flight Prices

This part investigates the distribution of flight fares to better understand their variability and range. A histogram shows how flight fares are distributed over the dataset, indicating whether they are concentrated in a single range or widely dispersed. Understanding the distribution allows you to discover probable outliers or skewness in the data, which can affect modelling performance.

```r
# Distribution of Flight Prices
ggplot(dataset, aes(x = price)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  theme_minimal() +
  labs(title = "Distribution of Flight Prices", x = "Price", y = "Count")
```

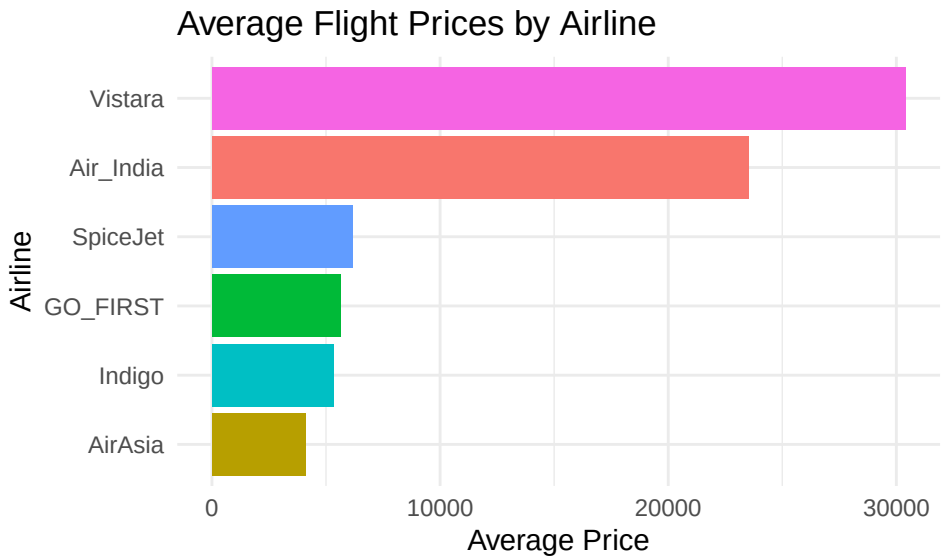


Average Prices by Airline

This section examines the variations in average flight fares between carriers. It uses a bar chart to visualise the average pricing for each airline, offering insights into how airline selection affects trip expenses. Such study is useful for evaluating pricing patterns and determining an airline's cost competitiveness.

```r
# Average Prices by Airline
ggplot(dataset %>% group_by(airline) %>% summarise(avg_price = mean(price, na.rm = TRUE)),
       aes(x = reorder(airline, avg_price), y = avg_price, fill = airline)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  theme_minimal() +
  labs(title = "Average Flight Prices by Airline", x = "Airline", y = "Average Price") +
```

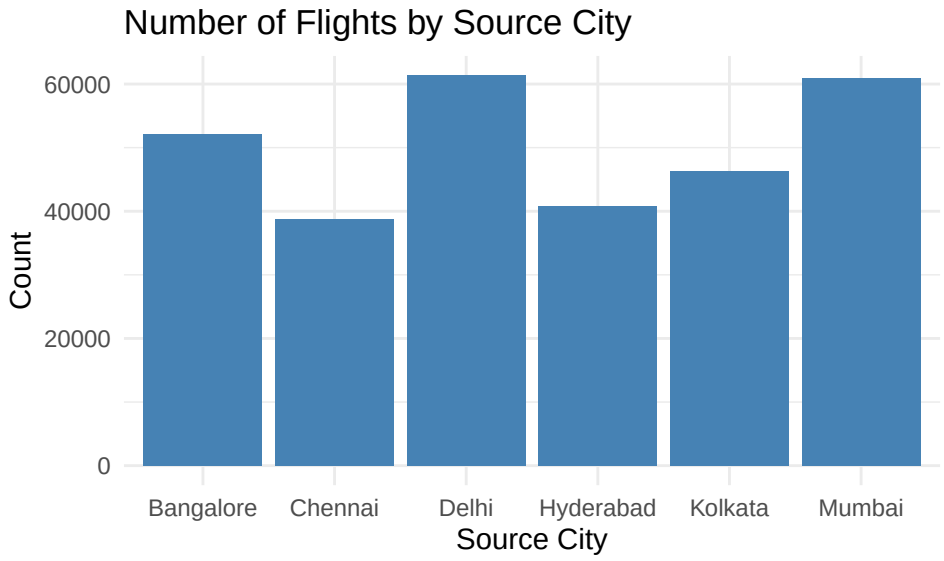
```
  coord_flip()
```



Number of Flights by Source City

This section examines the distribution of flights leaving from various source cities. A bar chart is used to represent the number of flights for each city, allowing you to determine the most popular departure points. This research sheds information on the concentration of flights across cities and their possible influence on flight prices.

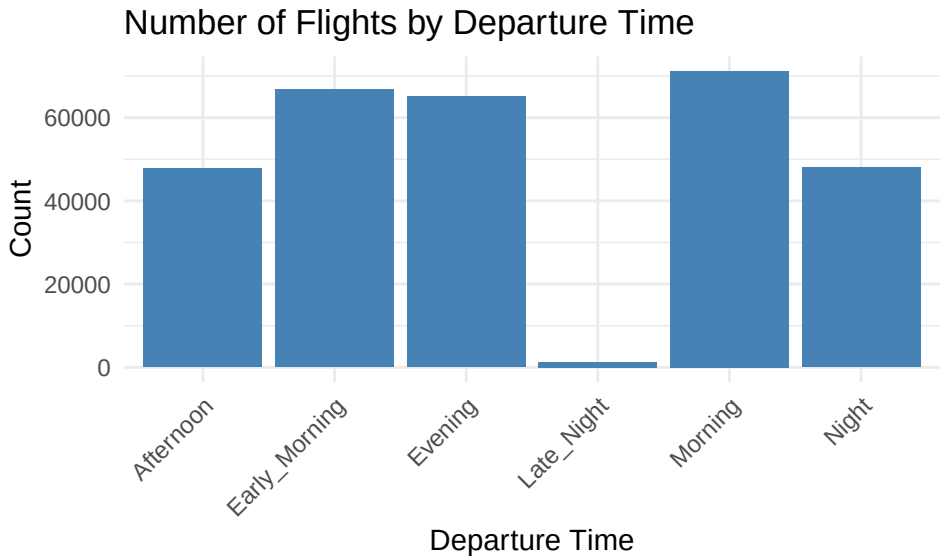
```
# Number of Flights by Source City
ggplot(dataset, aes(x = source_city)) +
  geom_bar(fill = "steelblue") +
  theme_minimal() +
  labs(title = "Number of Flights by Source City", x = "Source City", y = "Count")
```



Number of Flights by Departure Time

This section investigates the frequency of flights according to their departure periods, which are divided into intervals such as morning, afternoon, and evening. A bar chart depicts the data, emphasising the most frequent departure times. Such patterns are valuable for determining temporal trends in flight availability.

```
# Number of Flights by Departure Time
ggplot(dataset, aes(x = departure_time)) +
  geom_bar(fill = "steelblue") +
  theme_minimal() +
  labs(title = "Number of Flights by Departure Time", x = "Departure Time", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Flight Prices by Class

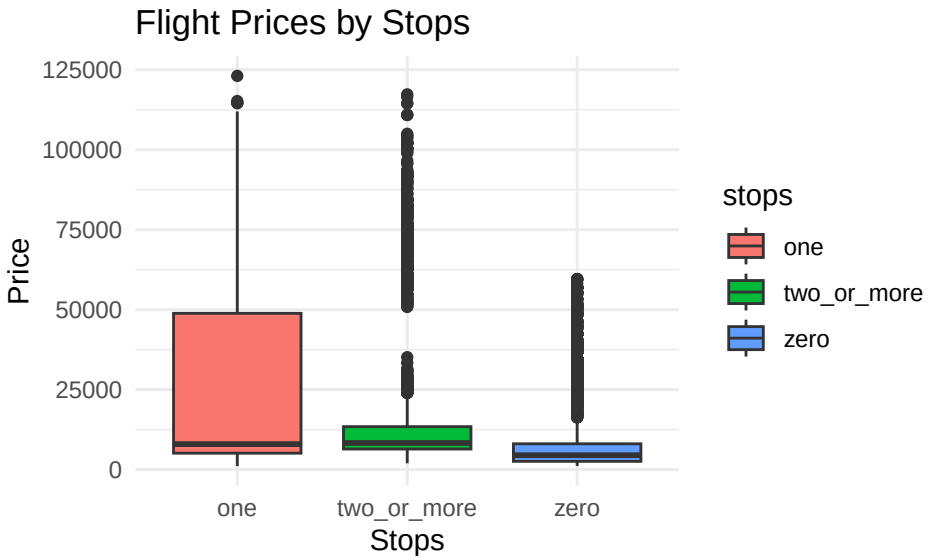
This part examines the link between flight fares and ticket class (e.g., Economy, Business). A boxplot is used to illustrate the price variance among classes, demonstrating how premium ticket alternatives often result in higher pricing. This visualisation shows a clear comparison of price ranges for various classes.

```
# Flight Prices by Class
ggplot(dataset, aes(x = class, y = price, fill = class)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Flight Prices by Class", x = "Class", y = "Price")
```

Flight Prices by Stops

This section looks at how the number of stops influences flight pricing. It uses a boxplot to show the price distribution for flights with varied number of stops (e.g., nonstop, one stop). The methodology helps establish if layovers have a substantial influence on ticket cost.

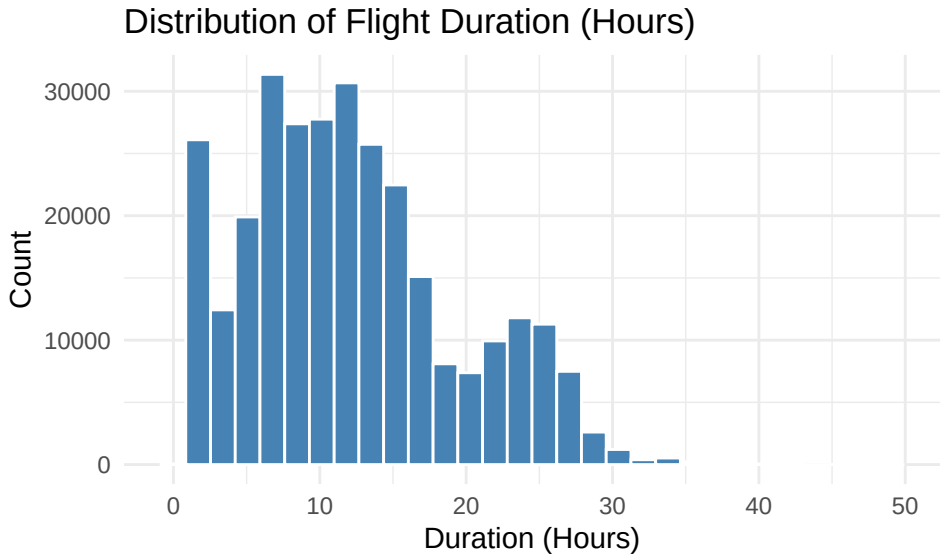
```
# Flight Prices by Stops
ggplot(dataset, aes(x = stops, y = price, fill = stops)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Flight Prices by Stops", x = "Stops", y = "Price")
```



Distribution of Flight Duration

This part investigates the distribution of flight lengths in hours. A histogram visualises flight length, indicating whether the majority of flights are short-haul or long-haul. This research gives background for determining how flight time affects pricing.

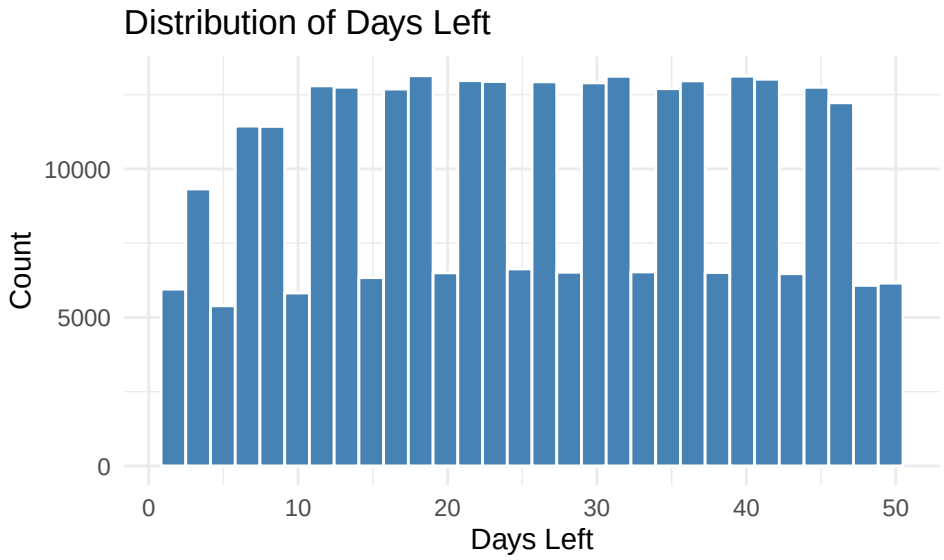
```r
# Distribution of Flight Duration
ggplot(dataset, aes(x = duration)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  theme_minimal() +
  labs(title = "Distribution of Flight Duration (Hours)", x = "Duration (Hours)", y = "Count")
```



Distribution of Days Left

This section looks at how the number of days till departure is spread across the dataset. A histogram depicts this aspect, demonstrating whether most bookings are made far in advance or closer to the travel date. This technique is critical for determining the time sensitivity of flight reservations and price.

```r
# Distribution of Days Left
ggplot(dataset, aes(x = days_left)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  theme_minimal() +
  labs(title = "Distribution of Days Left", x = "Days Left", y = "Count")
```



Flight Prices by Source City

This part investigates the difference in flight prices across various source cities. A boxplot depicts the price range for each city, making it easier to notice regional pricing patterns or discrepancies. This data sheds light on how origin cities affect ticket prices.

```
# Flight Prices by Source City
ggplot(dataset, aes(x = source_city, y = price, fill = source_city)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Flight Prices by Source City", x = "Source City", y = "Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Flight Prices by Destination City

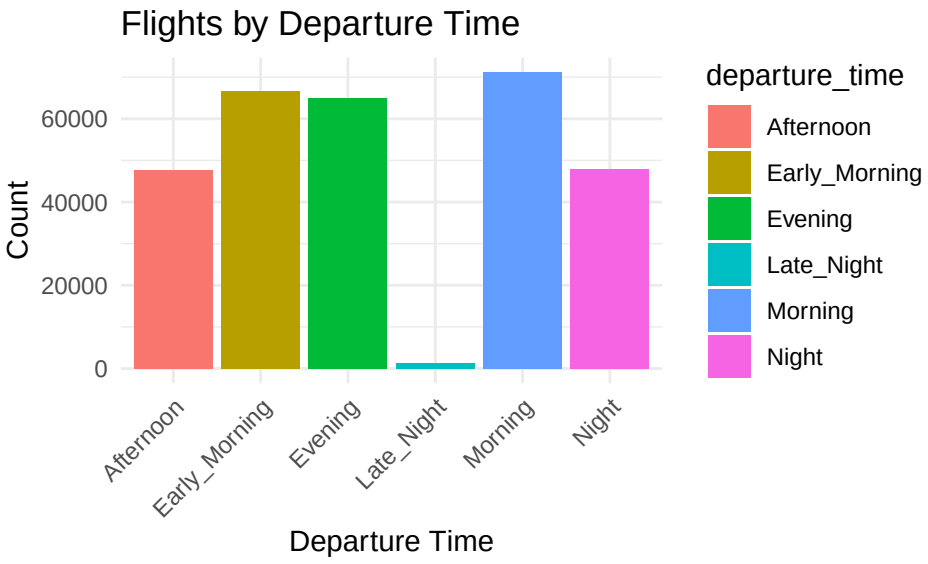
As with the preceding section, this research focusses on flight pricing across destination cities. A boxplot is used to show the price range for each city, emphasising how destination choice influences pricing patterns.

```
# Flight Prices by Destination City
ggplot(dataset, aes(x = destination_city, y = price, fill = destination_city)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Flight Prices by Destination City", x = "Destination City", y = "Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Flights by Departure Time

This section examines the frequency of planes departing at various times of day. A bar chart visualises the data by categorising flights into morning, afternoon, and evening periods. This research supplements the price-related findings by highlighting temporal flight tendencies.

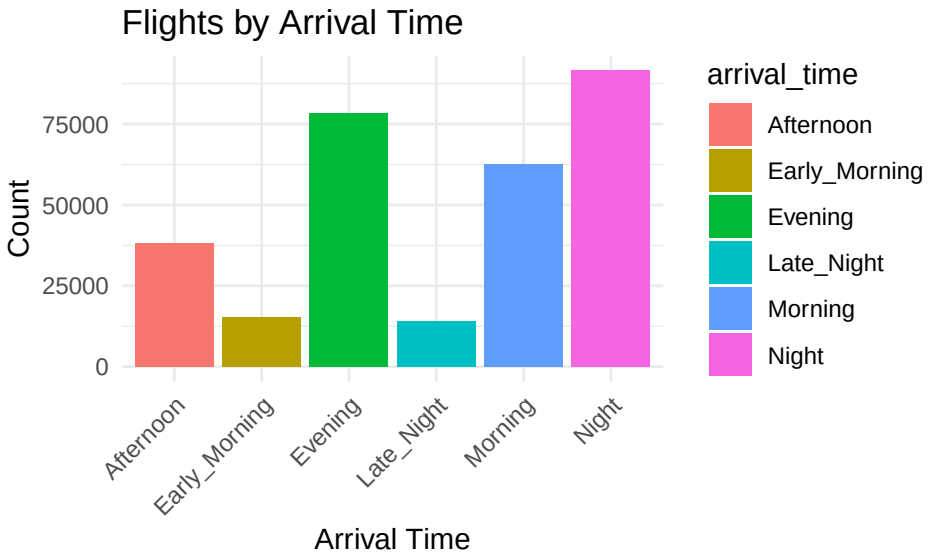
```
# Flights by Departure Time
ggplot(dataset, aes(x = departure_time, fill = departure_time)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Flights by Departure Time", x = "Departure Time", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Flights by Arrival Time

This part investigates the frequency of planes arriving at various times of day. Using a bar chart, it offers a temporal picture of flight arrivals, which may be compared against departure trends to better understand scheduling.

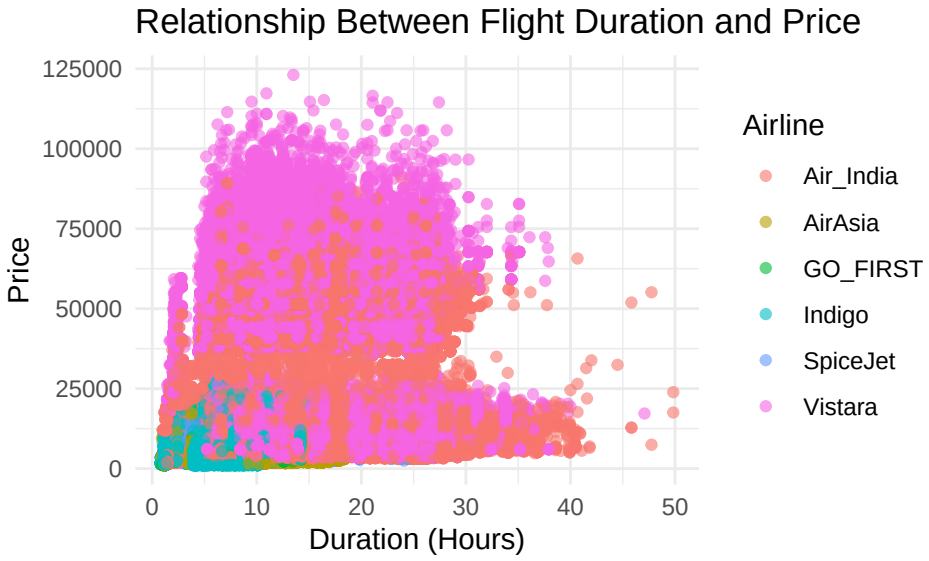
```r
# Flights by Arrival Time
ggplot(dataset, aes(x = arrival_time, fill = arrival_time)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Flights by Arrival Time", x = "Arrival Time", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Relationship Between Flight Duration and Price

This section uses a scatter plot to investigate the relationship between flight length and price. It displays if lengthier flights are more expensive or if other variables affect pricing. By analysing this connection, we may learn about how length influences cost.

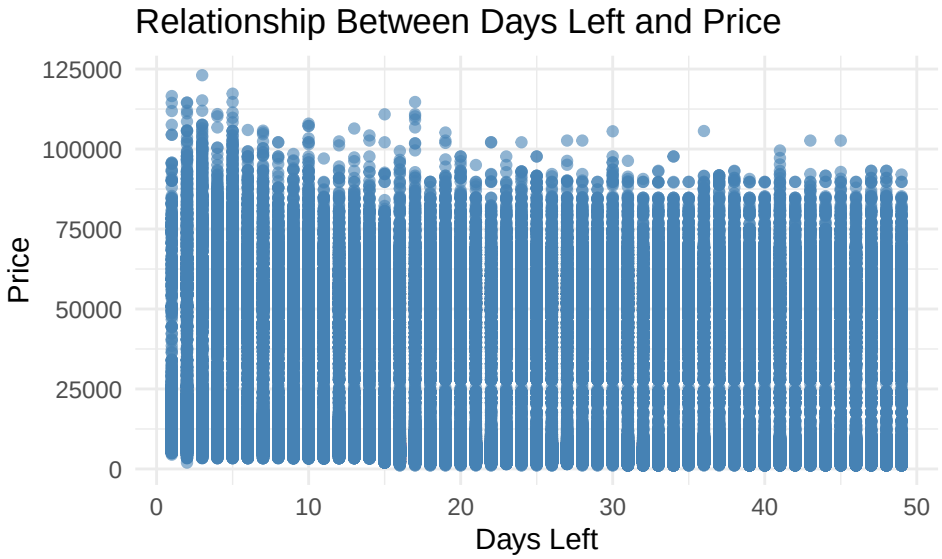
```r
# Relationship Between Flight Duration and Price
ggplot(dataset, aes(x = duration, y = price, color = airline)) +
  geom_point(alpha = 0.6) +
  theme_minimal() +
  labs(title = "Relationship Between Flight Duration and Price", x = "Duration (Hours)",
       y = "Price", color = "Airline")
```



Relationship Between Days Left and Price

This investigation looks at how the amount of days until travel impacts flight pricing. A scatter plot depicts the trend, usually indicating that prices rise as the departure date approaches. This information is useful for analysing booking trends and their impact on pricing strategies.

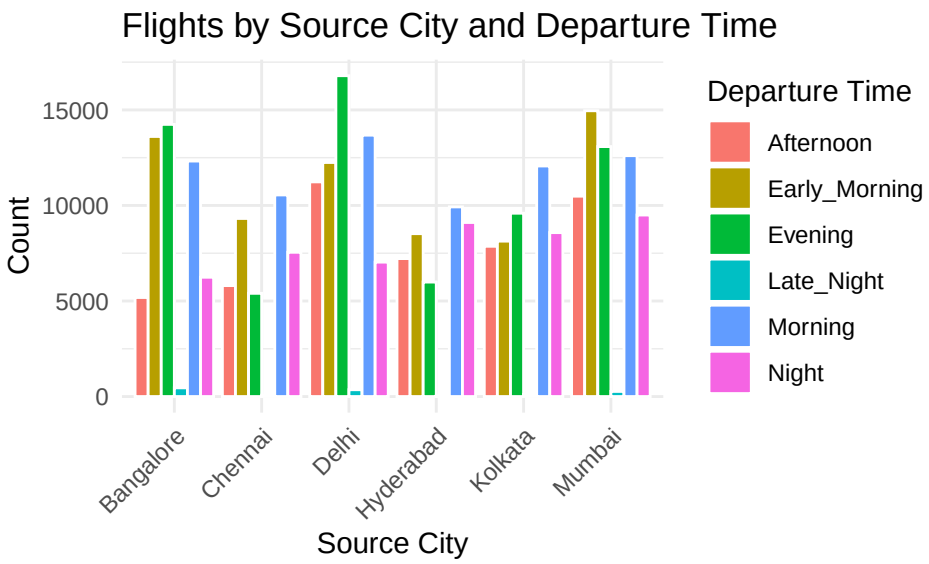
```
# Relationship Between Days Left and Price
ggplot(dataset, aes(x = days_left, y = price)) +
  geom_point(alpha = 0.6, color = "steelblue") +
  theme_minimal() +
  labs(title = "Relationship Between Days Left and Price", x = "Days Left", y = "Price")
```



Flights by Source City and Departure Time

This subchapter analyses the interaction of two categorical variables, source_city and departure_time. A grouped bar chart depicts the number of flights from each source city at various departure times, offering insights into temporal trends for certain cities.

```r
# Flights by Source City and Departure Time
ggplot(dataset, aes(x = source_city, fill = departure_time)) +
  geom_bar(position = "dodge", color = "white") +
  theme_minimal() +
  labs(title = "Flights by Source City and Departure Time", x = "Source City", y = "Count",
       fill = "Departure Time") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Model Building

This section focusses on developing and assessing machine learning models. Three models are used:

1) Linear Regression: Sets a baseline for price prediction by combining individual characteristics and a multivariate method.

2) Random Forest: Detects nonlinear links and interactions between features.

3) XGBoost: A cutting-edge algorithm for regression problems, recognised for its efficiency and accuracy.

Splitting Data

To guarantee that machine learning models are evaluated fairly, the dataset is separated into training and test sets. A part of the data (80%) is utilised for training, with the remaining 20% saved for testing. This divide is essential for determining the models' generalisation capabilities.

```r
set.seed(123)
train_index <- createDataPartition(dataset$price, p = 0.8, list = FALSE)
train_set <- dataset[train_index, ]
test_set <- dataset[-train_index, ]
```

Linear Regression

```r
# Individual models are fitted
fit_source_city <- lm(price ~ source_city, data = train_set)
fit_departure_time <- lm(price ~ departure_time, data = train_set)
fit_stops <- lm(price ~ stops, data = train_set)
```

```r
fit_arrival_time <- lm(price ~ arrival_time, data = train_set)
fit_destination_city <- lm(price ~ destination_city, data = train_set)
fit_class <- lm(price ~ class, data = train_set)
fit_duration <- lm(price ~ duration, data = train_set)
fit_days_left <- lm(price ~ days_left, data = train_set)

# Multivariate Linear Regression
# Fit multivariate linear regression
fit_multivariate <- lm(price ~ source_city + departure_time + stops + arrival_time +
                         destination_city + class + duration + days_left, data = train_set)
y_hat_multivariate <- predict(fit_multivariate, test_set)

# Evaluate performance
mult_rmse <- RMSE(y_hat_multivariate, test_set$price)
mult_mae <- MAE(y_hat_multivariate, test_set$price)
mult_r2 <- R2(y_hat_multivariate, test_set$price)
```

Random Forest

```r
# Random Forest Model
rf_model <- randomForest(price ~ source_city + departure_time + stops + arrival_time
                         + destination_city + class + duration + days_left, data = train_set,
                         ntree = 50, maxnodes = 20)
rf_pred <- predict(rf_model, test_set)

# Evaluate performance
rf_rmse <- RMSE(rf_pred, test_set$price)
rf_mae <- MAE(rf_pred, test_set$price)
rf_r2 <- R2(rf_pred, test_set$price)
```

XGBoost

```r
# Prepare data for XGBoost
xgb_train <- xgb.DMatrix(data = model.matrix(~ source_city + departure_time + stops +
                                               arrival_time + destination_city + class +
                                               duration + days_left - 1, data = train_set),
                         label = train_set$price)
xgb_test <- xgb.DMatrix(data = model.matrix(~ source_city + departure_time + stops +
                                              arrival_time + destination_city + class +
                                              duration + days_left - 1, data = test_set))

# Train XGBoost Model
xgb_model <- xgboost(data = xgb_train, max_depth = 4, eta = 0.1, nrounds = 50,
                     objective = "reg:squarederror", verbose = 0)
xgb_pred <- predict(xgb_model, xgb_test)

# Evaluate performance
xgb_rmse <- RMSE(xgb_pred, test_set$price)
xgb_mae <- MAE(xgb_pred, test_set$price)
xgb_r2 <- R2(xgb_pred, test_set$price)
```

Results Comparison

This section evaluates the three models' performance using measures including RMSE, MAE, and R^2. A results table presents a succinct comparison, emphasising the best-performing model (XGBoost) and its greater accuracy at forecasting flight costs.

```
# Compile results
results <- data.frame(
  Model = c("Multivariate Linear Regression", "Random Forest", "XGBoost"),
  RMSE = c(mult_rmse, rf_rmse, xgb_rmse),
  MAE = c(mult_mae, rf_mae, xgb_mae),
  R2 = c(mult_r2, rf_r2, xgb_r2)
)

# Display results
knitr::kable(results, caption = "Performance Comparison of Models")
```

Table 1: Performance Comparison of Models

Model	RMSE	MAE	R2
Multivariate Linear Regression	6964.262	4656.313	0.9059019
Random Forest	9824.884	7481.765	0.9131757
XGBoost	5350.473	3183.110	0.9446479

Conclusion

The performance comparison findings show that several machine learning models are successful at forecasting flight prices. XGBoost is the best-performing algorithm examined, with the lowest RMSE (5350.473) and MAE (3183.110), as well as the greatest R^2 value (0.9446). These measurements show that XGBoost not only predicts flight fares more accurately, but also explains a considerable percentage of the volatility in the target variable, making it an extremely dependable choice for this job.

The Multivariate Linear Regression model, with an RMSE of 6964.262 and MAE of 4656.313, serves as a solid foundation for comparison. The R^2 value of 0.9059 indicates that it captures a significant percentage of the variation in the data. However, it lacks the flexibility to model complicated, non-linear correlations between attributes and prices. This restriction is shown in the greater prediction errors compared to XGBoost.

The Random Forest model has a somewhat higher R^2 (0.9132) than linear regression, but with greater prediction errors (RMSE of 9824.884 and MAE of 7481.765). This shows that, while Random Forest can capture certain feature relationships, it may not generalise as well on this dataset. Factors such as the number of trees and depth limitations might have contributed to the poor performance.

Overall, the findings highlight the necessity of using sophisticated algorithms such as XGBoost for applications requiring complicated datasets with nonlinear interactions. While linear regression is an effective baseline, and Random Forest gives substantial gains, XGBoost's higher performance indicates that it is the best model for forecasting flight prices in this context. Future research might focus on fine-tuning XGBoost hyperparameters or introducing new characteristics to improve prediction accuracy.

References

Bathwal, S. (n.d.). Flight Price Prediction [Data set]. Kaggle. Retrieved December, 2024, from https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction