Dang Thien Nguyen

Dxn2100021

Huy Tran

hxt200010

CS 4372.501 – Computational Methods for Data Scientists

FLIGHT PRICE DATA ANALYSIS AND PREDICTION



# Background

The airline industry generates over $800 billion in annual revenues worldwide. Ticket fares represent most of these revenues and are a crucial driver of profitability. Airlines aim to price tickets competitively to maximize yields and stay profitable on each route.g

However, forecasting flight demand and optimizing ticket prices remains a challenge. Prices fluctuate frequently due to competitor pricing, supply/demand dynamics, operating costs, and other factors. Finding the right price to sell each seat is crucial.

In today's data-driven world, airlines are increasingly looking to leverage historical booking data and machine learning to build predictive models. These models can forecast expected demand and recommend optimal prices for each route, flight, and booking time.

This helps airlines dynamically adjust fares based on complex variables like seasons, competitions, time left to departure, etc. Access to accurate price predictions allows airlines to improve revenues and stay profitable.

# Data collection

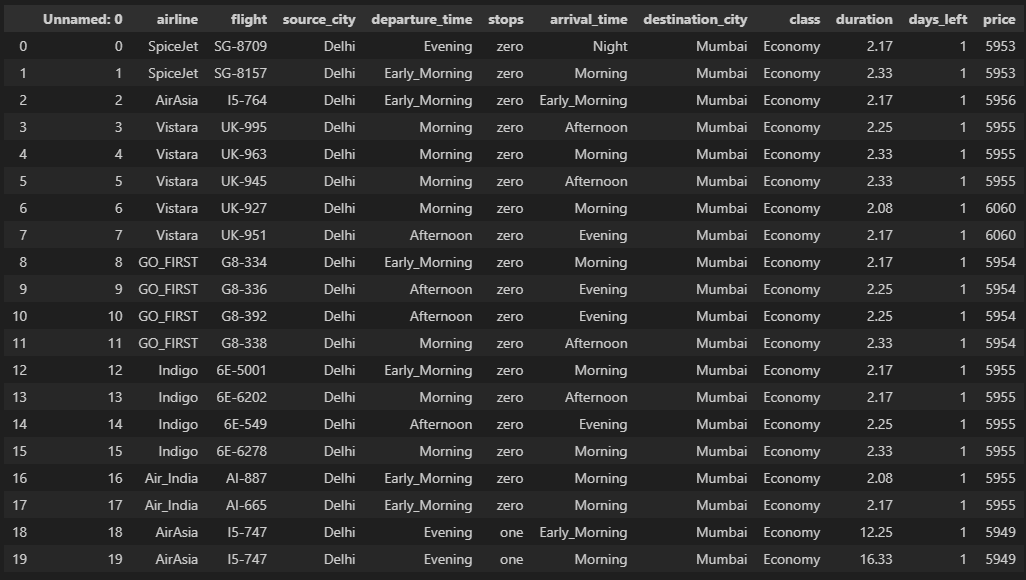
The flight booking data in this study was obtained by scraping the EaseMyTrip website using the Octoparse scraping software. Data was gathered in two separate extracts - one for economy class tickets and one for business class tickets. The dataset contains information about flight booking between India's top 6 metro cities. There are 300,261 data points, and 11 features were extracted from the site over 50 days from February 11th to March 31st, 2022.

The flight booking data used in this project was obtained from Kaggle: [Flight Price Prediction (kaggle.com)](https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction/data)

The data was available as 3 CSV files: economy.csv, business.csv, and the combination of 2 previous files: Clean\_Dataset.CSV.

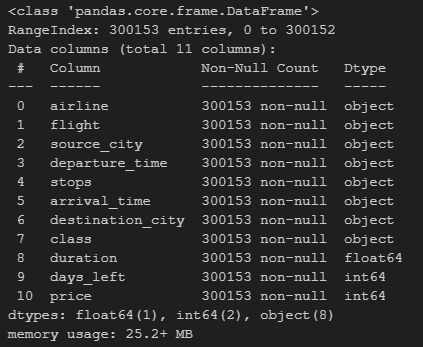
The dataset contains 11 attributes that capture vital details about each flight reservation:

* Airline - Categorical variable indicating the airline carrier, with 6 different airlines represented.
* Flight - Categorical code uniquely identifying each flight number.
* Source City - Categorical departure airport for the flight, with 6 distinct cities.
* Departure Time - Categorical feature created by binning flight departure times into 6 time period labels.
* Stops - Categorical attribute with 3 values indicating the number of stops on flight.
* Arrival Time - Categorical feature generated by grouping arrival times into 6 time period bins.
* Destination City - Categorical arrival airport with 6 unique destination cities.
* Class - Categorical variable for ticket class, either Business or Economy.
* Duration - Continuous measure of flight duration in hours.
* Days Left - Derived variable calculating days between booking and departure.
* Price - Continuous target variable representing the ticket price.



# Exploratory Data Analysis

## Checking Null and Missing data

The data consisted of 300,153 rows and 11 columns.

The flight prediction data was checked for null and N/A values using df.isnull().sum(). This identified no null values, and the data appeared to have no null value or any missing data.

A screen shot of a computer

Description automatically generated A screen shot of a computer

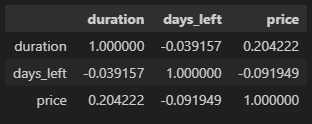
Description automatically generated

## Let's do some statistic

A screen shot of a black screen

Description automatically generated

* Airline, source/destination cities have 6 unique values indicating most major Indian cities are represented.
* Stops show 250863 one-stop flights vs 4290 non-stop, so most flights have 1 stop.
* Class is imbalanced with 206666 economy tickets vs 79487 business class.
* Flight durations vary widely from 0.83 to 49.83 hours, with a mean of 12.22 hours.
* Days left between booking and travel range from 1 to 49, with a mean of 26 days.
* Prices range from 1105 to 123071 INR, right skewed with mean 20890 > median 7425.
* Departure and arrival times binned into 6 periods show Morning/Night as most expected.

Using the correlation matrix, we have:

* Duration has a moderate positive correlation of 0.20 with price. This aligns with the expectation that longer flights are more expensive.
* Days\_left has a weak negative correlation of -0.09 with price. Booking further in advance is associated with slightly lower fares.
* The correlation between duration and days\_left is negligible at -0.04, indicating they are independent variables.
* Of the two features, duration has a stronger relationship with the target price variable based on the higher absolute correlation value.

## Data Visualization

### Price Distribution

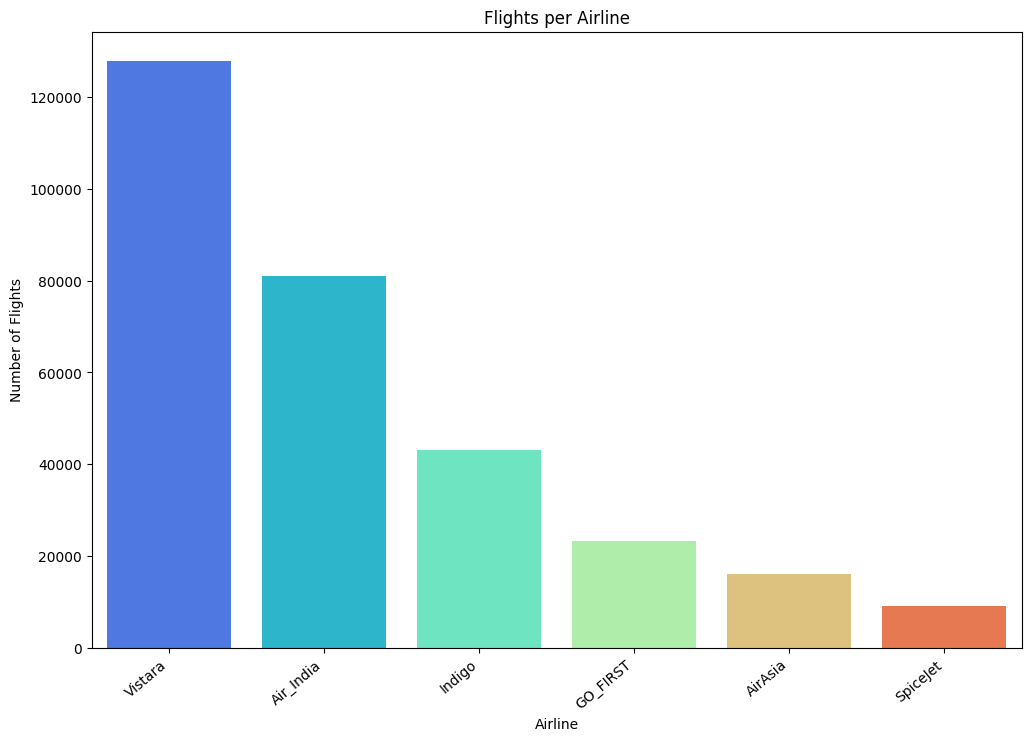
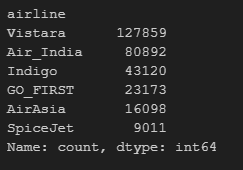
A graph of a graph of a graph

Description automatically generated with medium confidence

* The histogram shows a right-skewed, unimodal distribution. This indicates that most prices are concentrated on the lower end, with a long tail to higher values.
* Fitting a kernel density estimate (KDE) shows the distribution is approximately normal but shifted higher than a centered Gaussian.
* The boxplot also displays the right skew, with the median around 7,500 INR lower than the mean indicated by the dot at ~20,000 INR.
* According to the boxplot, several potential outliers exist at the high end beyond 100,000 INR.
* IQR range is ~5,500 to ~35,000, indicating that the middle 50% of prices lie within that spread.

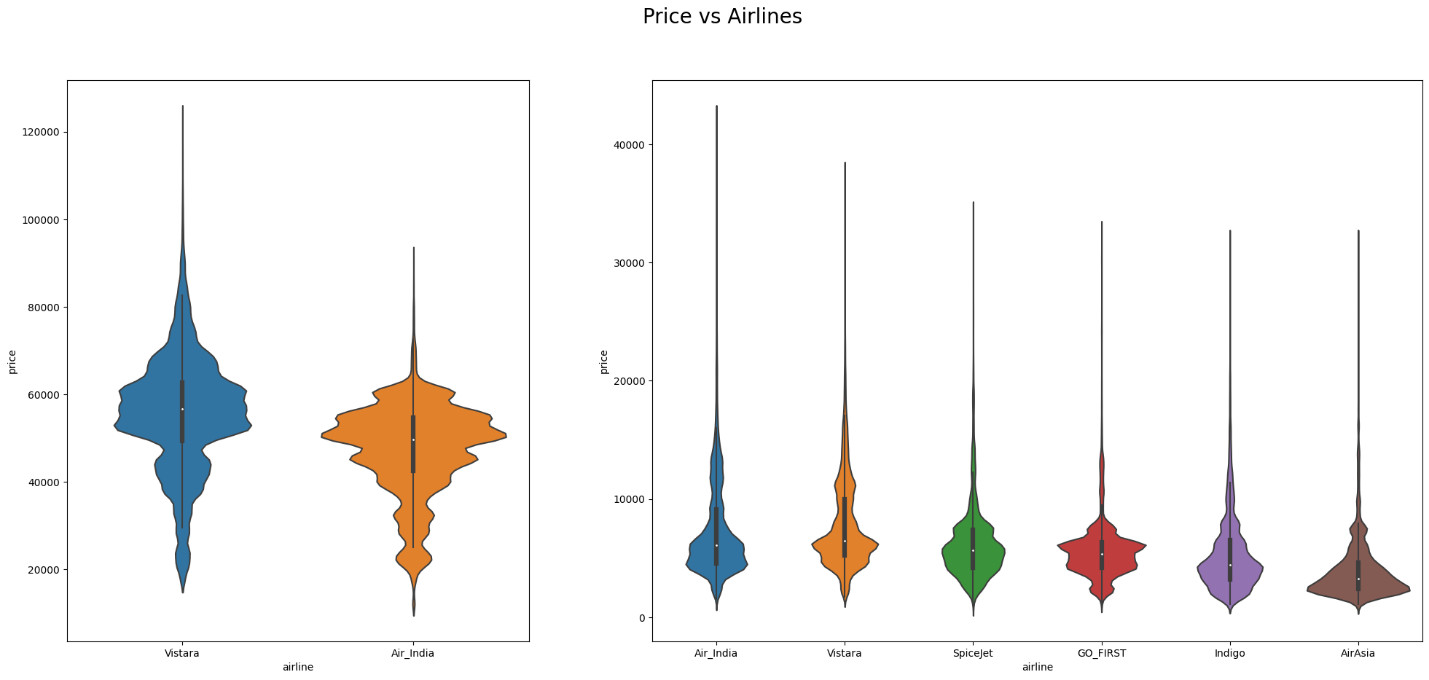
### Number of flights per Airline

Analyzing the pie chart showing the distribution of ticket classes:

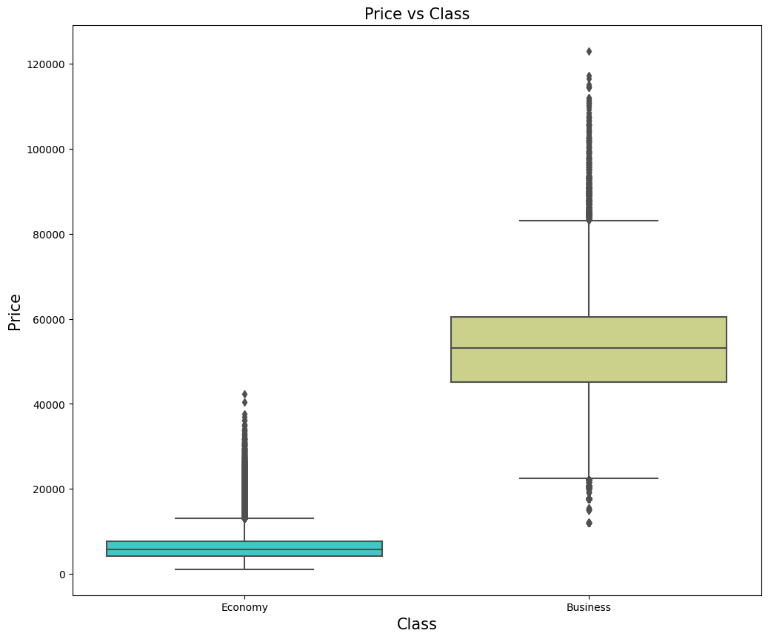
* The majority of tickets (around 68%) are economy class.
* Business class accounts for the remaining 32% of tickets.
* There is a clear imbalance between the two classes, with the economy being twice as prevalent.
* The size of each slice's size accurately reflects each ticket type's relative proportions.
* Economy class dominates because it generally has lower fares and higher demand.
* The visualized imbalance should be considered during analysis as economy seats will likely drive overall pricing trends.

### A blue and orange pie chart Description automatically generatedPrice distribution by classA black background with white text Description automatically generated

* The data shows 206,666 Economy class passengers and 93,487 Business class passengers.
* The Economy class has a significantly higher count than the Business class.
* This suggests that most passengers in this dataset traveled in Economy class.
* The difference in count between the two classes could indicate a difference in pricing or availability of seats.



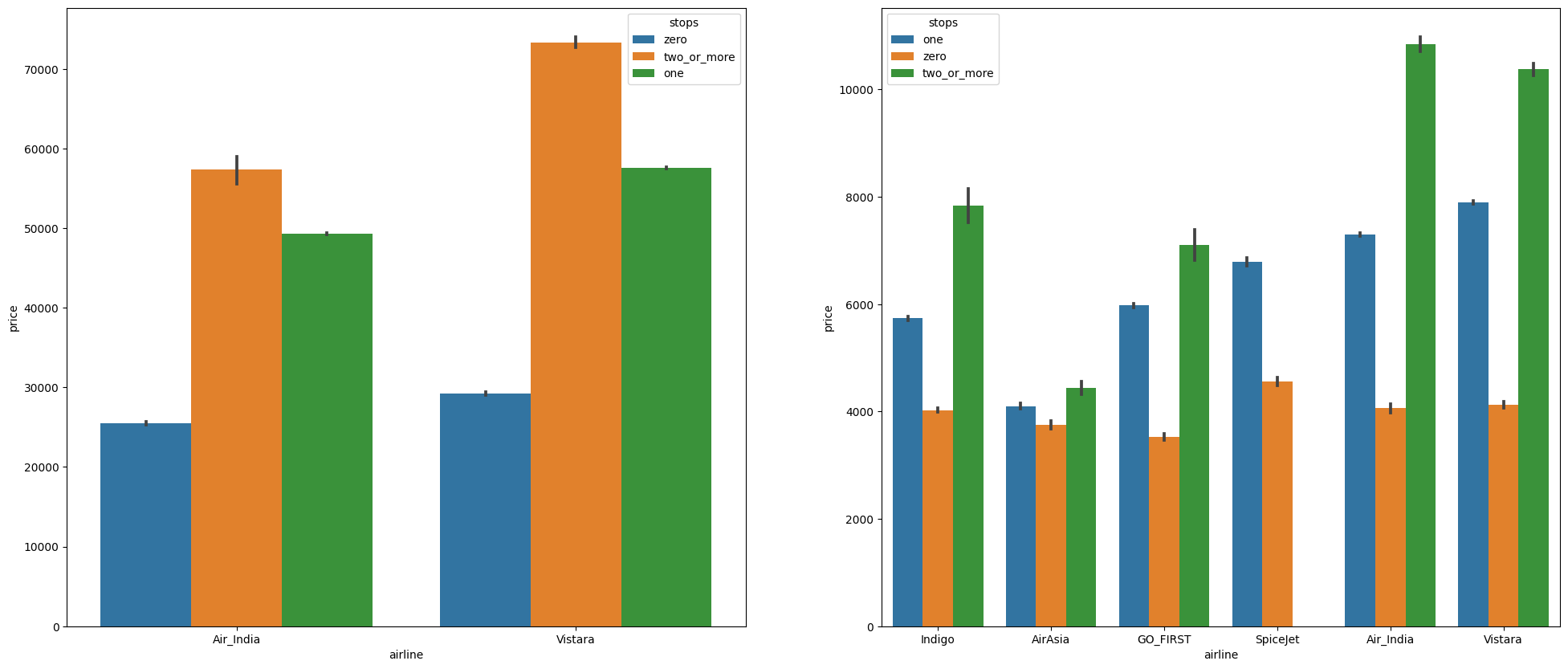
The graph shows slight variations in flight prices between the three airlines. AirAsia offers the most affordable fares, while Air India and Vistara are more costly. However, Vistara's business class tickets appear slightly pricier than Air India's business class fares.



* Prices for business class are substantially higher overall than the economy.
* The business class distribution has a higher median and broader spread of values.
* Several potential business class outliers exist at very high prices over 100k.
* Economy class prices have a lower median of around 7-8k and tighter IQR between 5-15k.
* The whiskers indicate similar minimum prices, but the maximum is much higher for business.
* Notches between the boxes do not overlap, indicating statistically significant differences in medians.

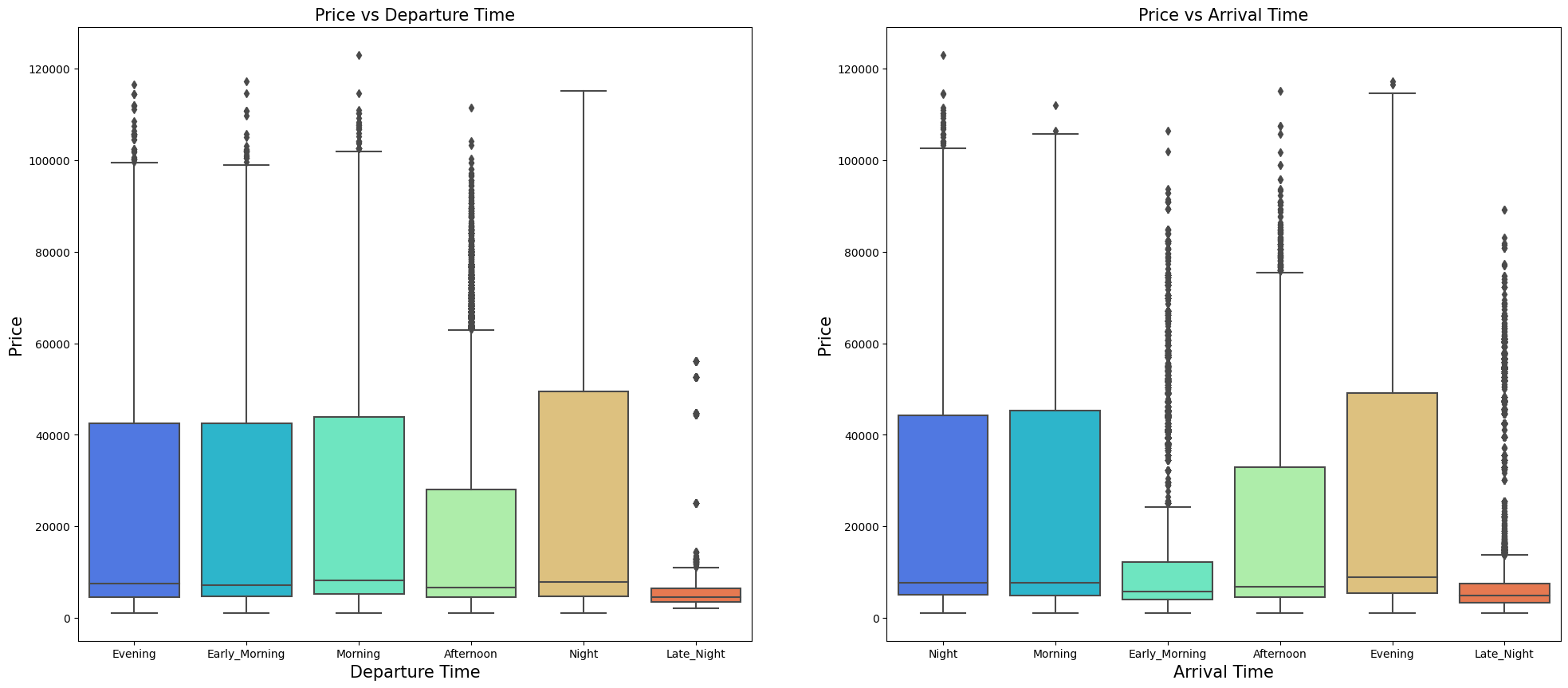
### A chart with different colored squares Description automatically generatedPrice distribution based on the number of stops

* While Economy class prices are primarily concentrated in the lower range, some exceptions are priced higher, likely due to factors like time of booking, flight duration, or exceptional amenities.
* Business class ticket prices have a broader range, suggesting there might be different levels or tiers within the Business class itself or varying amenities and services offered at different prices.



* Non-stop flights are generally cheaper for airlines like AirAsia and Indigo.
* Air\_India tends to be more expensive, especially for multiple-stop flights.
* Vistara has a higher price point for direct and one-stop flights than other airlines but is competitively priced for multiple-stop flights.

### Price distribution based on departure time and arrival time



Departure Time:

* Tickets for flights departing in the Afternoon seem to generally cost more, while late-night departures are cheaper, though exceptions exist in the form of outliers.
* Each departure time has its range of outliers, with the Afternoon and Late Night slots showing a significant number.

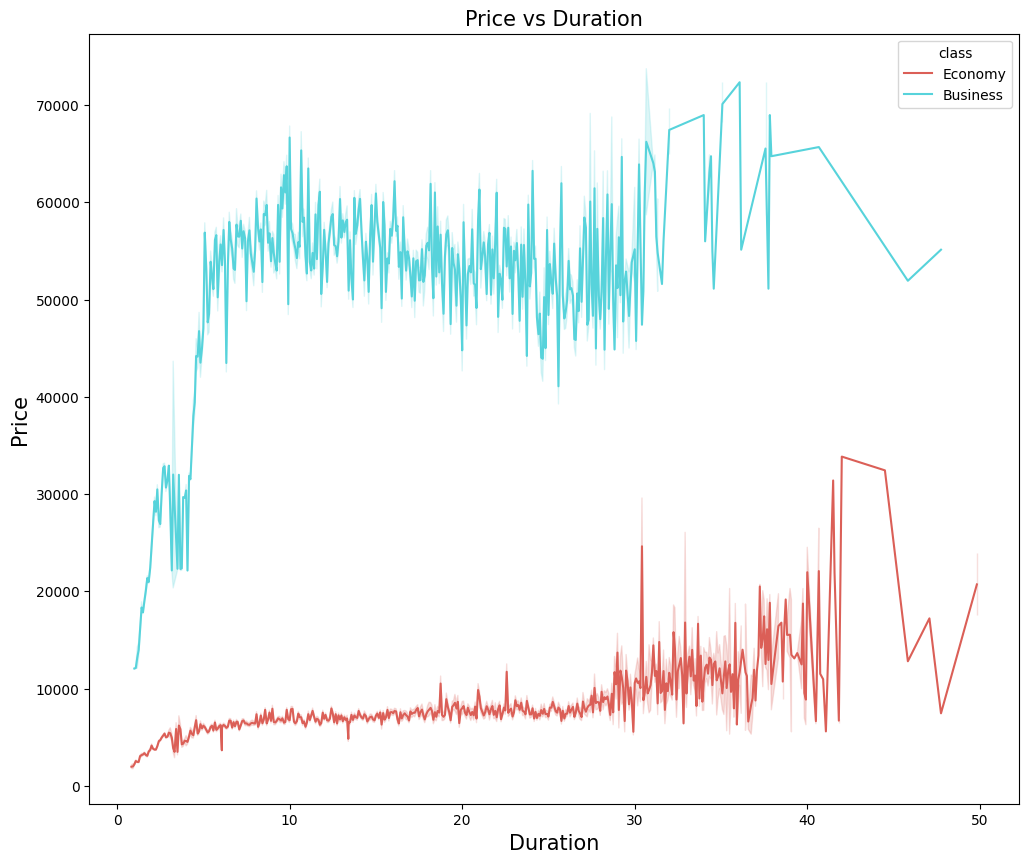
Arrival Time:

* Flights arriving in the Evening and Afternoon tend to have higher median prices, while Early Morning arrivals are generally cheaper.
* Similar to the departure plot, each arrival time has outliers, but the Late Night and Evening slots have the most noticeable outliers.

### Price distribution based on Departure city and Arrival city



* Delhi as Source City:
  + Flights from Delhi to Mumbai, Bangalore, and Hyderabad seem to have similar average prices.
  + Flights to Kolkata and Chennai have a slight dip in prices, making them a bit more affordable from Delhi.
* Mumbai as Source City: Flights from Mumbai to other cities seem to experience a price increase, with the highest being to Bangalore, followed by a significant drop towards Chennai.
* Bangalore as Source City: Flights originating from Bangalore see a steady rise in prices towards Kolkata and then a gradual drop towards Chennai.
* Kolkata as Source City: Flights from Kolkata to Mumbai appear the most expensive, with prices decreasing gradually as they head towards southern cities like Hyderabad and Chennai.
* Hyderabad as Source City: Prices of flights from Hyderabad show a noticeable drop when heading to Bangalore and Kolkata, with an increase towards Chennai.
* Chennai as Source City: Flights from Chennai consistently trend toward increasing prices towards Mumbai and decreasing towards Delhi.



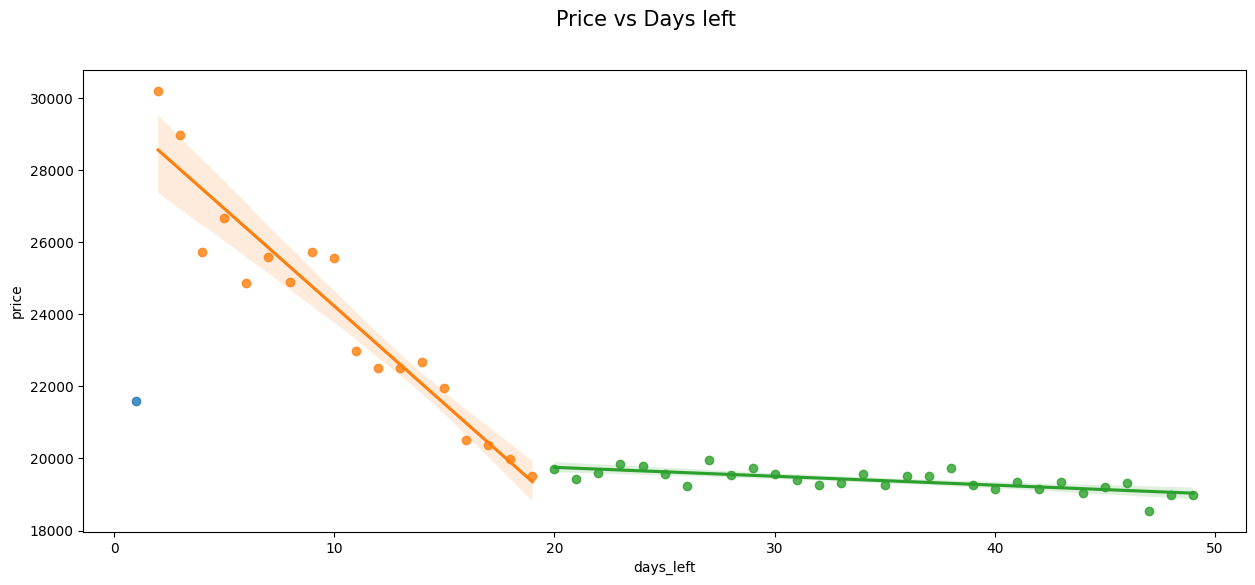
1. Economy Class (Teal Line)

* For shorter durations (around 0 to 10 units, possibly hours), there's a sharp
* increase in price. This suggests that shorter economic class flights might have some premium pricing due to high demand or convenience.
* Post the initial surge, the price fluctuates but generally stays within 50,000 to 60,000 units for most flight durations. There's no clear linear relationship between flight duration and price in this range.
* As the duration approaches around 40 units, a price dip is noticeable, followed by an increase. This could indicate unique routes or less popular long-duration flights priced lower.

1. Business Class (Red Line)

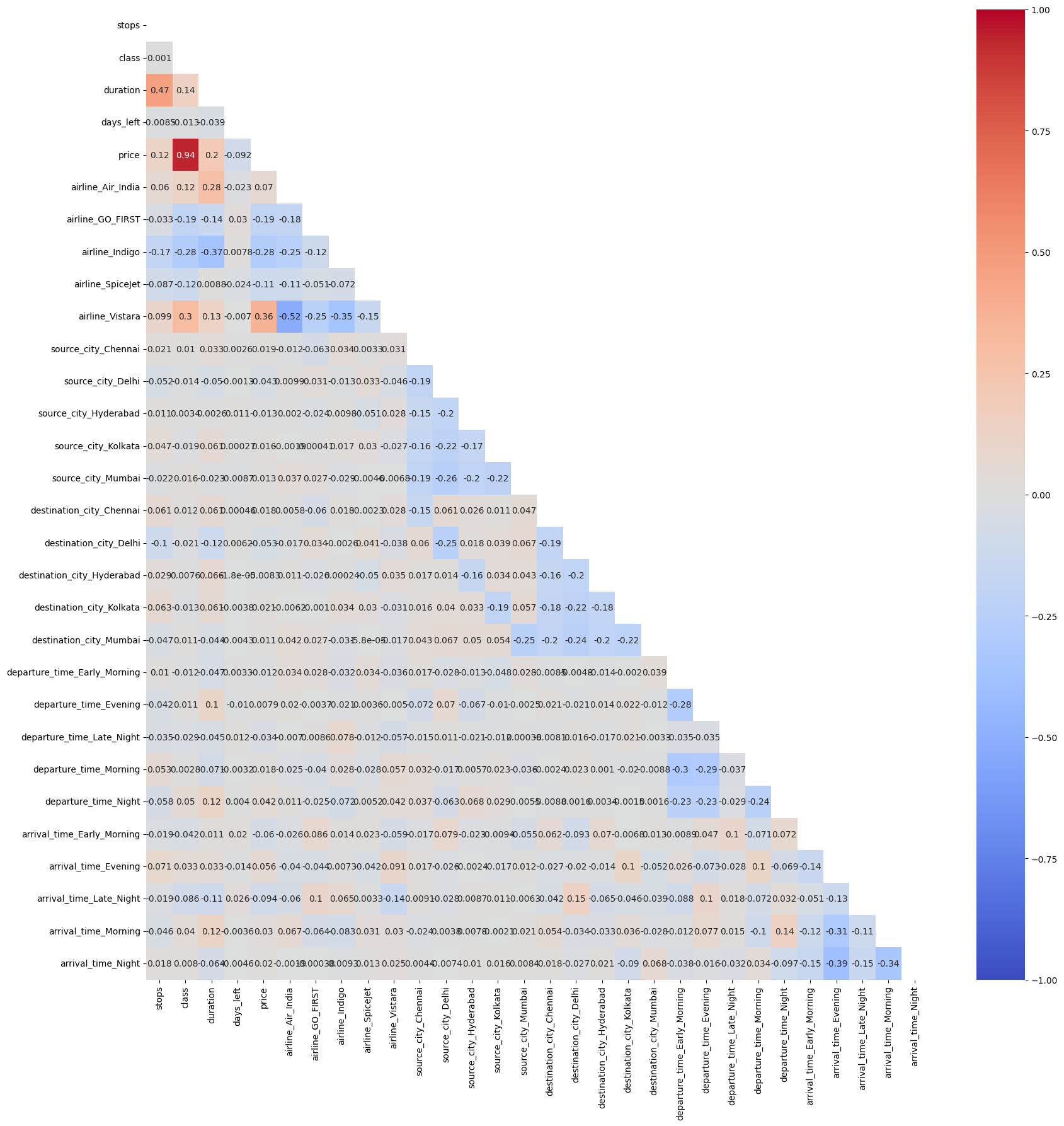
* The business class tickets are priced significantly lower for very short durations but see a steep increase early on.
* Between durations of 10 to 30 units, the price of business class flights stays relatively stable, hovering slightly above the 10,000 mark. This is interesting because it means that business class can be pretty affordable for a significant range of flight durations, at least when compared to the peaks of economy class pricing.
* There's a sharp rise in price around the duration of 30 units, peaking near 40 units. Post this peak, the price sees a decline. This could signify premium long-haul business flights that command higher prices.

### Price distribution based on the number of days purchased before the flight



* Prices are highest when purchasing 1-2 days before departure, likely due to last-minute travel needs and seat scarcity, allowing airlines to charge more.
* There is a negative correlation between price and days left in the 3-20 day range. As advance purchase increases and average prices decrease.
* The downward price trend levels off beyond 20 days before departure. Prices remain relatively stable and don't drop much further.
* R^2 values are low for the linear fit lines, so days left alone do not fully explain price variance. Other factors are at play.
* The "sweet spot" for cheaper fares appears 3-4 weeks in advance. Last-minute purchases are the most expensive.

## Correlation

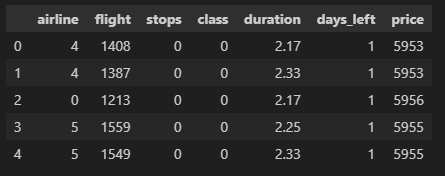


* Price and Duration: A coefficient of 0.92 suggests a very strong positive correlation between the price of a flight and its duration. As the duration increases, the price tends to increase as well.
* Departure and Arrival Times: The matrix shows various times of day for departure and arrival (e.g., departure\_time\_Early\_Morning, arrival\_time\_Late\_Night). Some combinations, like 'departure\_time\_Evening' and 'arrival\_time\_Night,' have a positive correlation of 0.28. This suggests that flights departing in the evening will likely arrive at night.
* Airline and Source/Destination City: Different airlines seem to have some correlation with source or destination cities. For instance, 'airline\_Spicejet' and 'source\_city\_Mumbai' have a negative correlation of -0.22, indicating that as the frequency of flights by Spicejet increases, the flights originating from Mumbai tend to decrease (or vice versa).
* Stops and Duration: These have a positive correlation of 0.47, which suggests flights with more stops tend to have longer durations.
* Self-correlation: Diagonal elements (from the top-left to bottom-right) all show a correlation of 1. This is expected because any variable has a perfect positive correlation with itself.
* Airline Affinities: Certain airlines have noticeable correlations or anti-correlations with others. For instance, 'airline\_Air\_India' and 'airline\_IndiGo' have a negative correlation of -0.19, suggesting that when one airline operates more, the other might use less on specific routes or timings.

## Data Preparation

First, a subset of flight prediction data is created, keeping only 'airline,' 'flight,' 'stops,' 'class,' 'duration,' 'days\_left,' and 'price.'

Next, we will use sci-kit-learn's LabelEncoder to encode any object columns to numeric values for modeling.



The `price` column is separated into a target variable, y. The remaining columns are assigned to the feature matrix X

In the end, sci-kit-learn's train\_test\_split() function is used to split the feature and target arrays into randomized train and test sets with a test size of 30%. A fixed random state ensures consistent splits.

The training and testing feature/target shapes are printed, confirming a `70%/30%` split of the data.



## Modeling

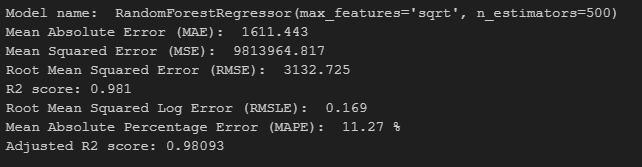
### Training 4 models with parameter tunning

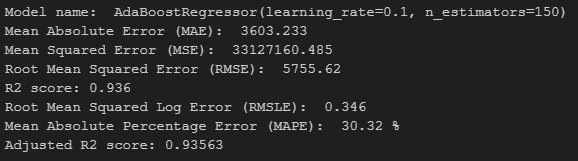
The pipeline follows standard machine learning steps - data preprocessing, feature engineering, model building, hyperparameter tuning, and evaluation. Models tested include linear regression, decision trees, random forest, and XGBoost. Grid search is used for hyperparameter optimization. Evaluation metrics are RMSE, R2 score, and MAPE.

After tuning and training, here is the result of 4 models:

A computer screen with numbers and symbols

Description automatically generated





A screenshot of a computer error

Description automatically generated

* Mean Absolute Error (MAE):

Lowest: XGBRegressor (1497.2): This indicates that on average, the XGBRegressor model predictions are off by 1497.2 units from the actual values, which is the smallest error among the models.

* Mean Squared Error (MSE):

Lowest: XGBRegressor (7795825.793): This metric penalizes large errors more, and the XGBRegressor model has the lowest MSE.

* Root Mean Squared Error (RMSE):

Lowest: XGBRegressor (2792.101): This is the square root of MSE, and again, the XGBRegressor model outperforms the others.

* R2 Score:

Highest: XGBRegressor (0.985): The R2 score (coefficient of determination) indicates the proportion of the variance for the dependent variable explained by the independent variables. A higher R2 score is generally better.

* Root Mean Squared Log Error (RMSLE):

Lowest: RandomForestRegressor (0.169): RMSLE is useful when penalizing underestimations more than overestimations. Here, the RandomForestRegressor has the lowest error.

* Mean Absolute Percentage Error (MAPE):

Lowest: RandomForestRegressor (11.27%): This indicates the error as a percentage, and the RandomForestRegressor has the lowest percentage error.

* Adjusted R2 Score:

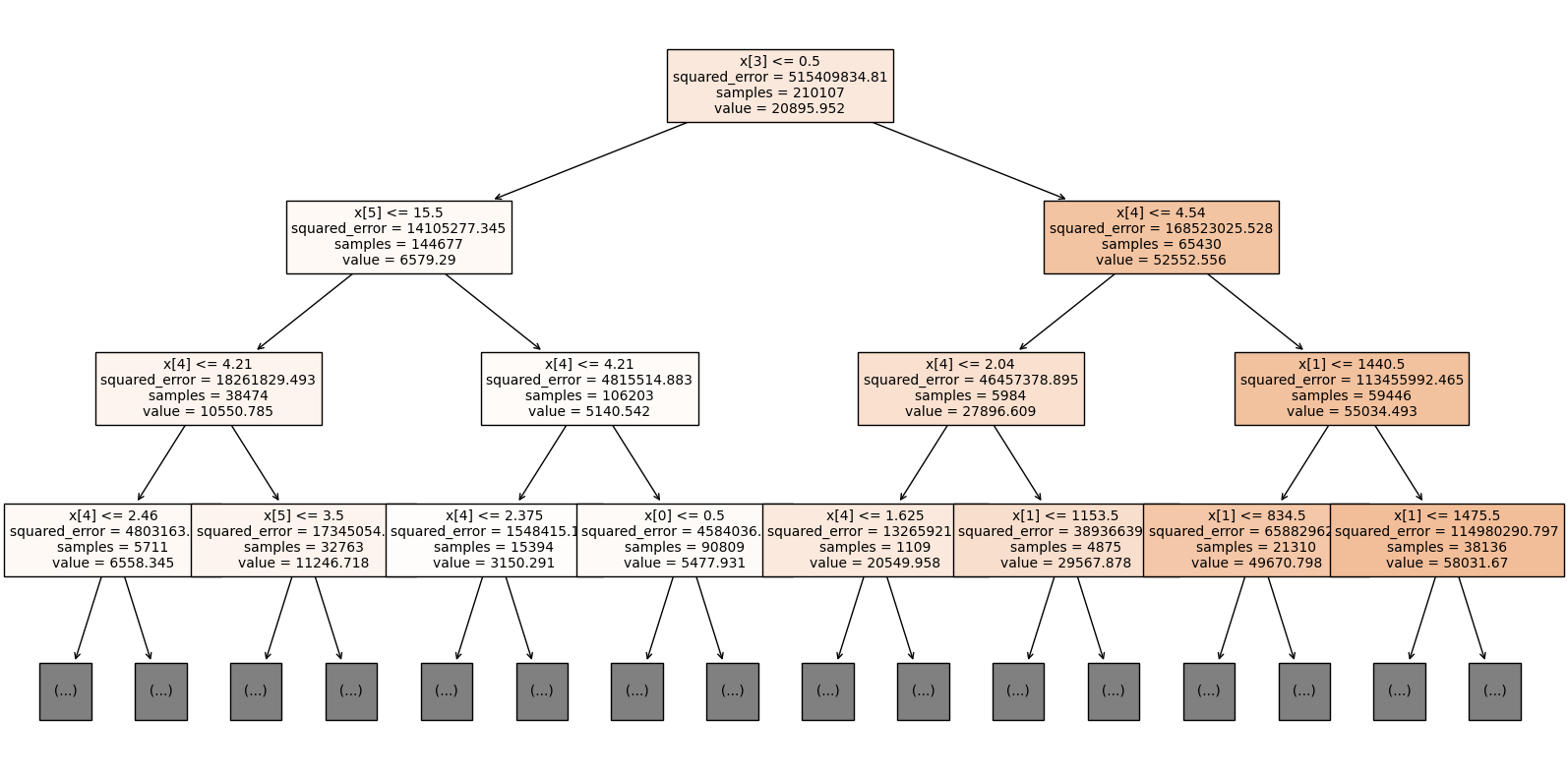
Highest: XGBRegressor (0.98485): Adjusted R2 score accounts for the number of predictors in the model. The XGBRegressor again tops this metric.

Summary:

* XGBRegressor excels in most metrics, especially MAE, MSE, RMSE, R2 score, and Adjusted R2 score. It indicates that this model, on average, makes the most accurate predictions among the four models.
* RandomForestRegressor has the lowest RMSLE and MAPE. This means it's less likely to underestimate, and its predictions as a percentage of actual values are the closest among the models.
* AdaBoostRegressor performs the poorest in every metric compared to the other models.
* DecisionTreeRegressor is a middle-performing model here. Both XGBRegressor and RandomForestRegressor outperform it on almost all metrics.

### Tree Visualization

### Decision Tree Regressor



### Random forest Regressor

A diagram of a network

Description automatically generated

### AdaBoost Regressor

A diagram of a diagram

Description automatically generated

### XGBoost Regressor

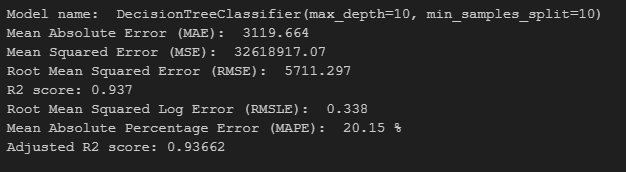
A white background with red and blue dots

Description automatically generated

### Result Analysis

To be able to calculate precision, recall, and F-statistic and plo tROC and Precision-Recall curve, we have to use the Classifier model. However, due to the limit resources, I will use DecisionTreeClassifier.

First, we will train the model again with tuning by GribSearchCV.



Then we will use precision\_score, recall\_score, and f1\_score from sci-ki-learn to calculate:

A black screen with white text

Description automatically generated

* The model seems to fit the data well in explaining the variance (R2 of 0.937).
* However, the classification metrics (precision, recall, and F1-score) are pretty low. This indicates that the model might be struggling to classify the data correctly. The low precision suggests many false positives, while the low recall suggests many positive cases are missed.
* The model's predictions, on average, deviate by a magnitude of 3119.664 units, which may or may not be acceptable depending on the problem's context.
* A MAPE of 20.15% indicates the model's predictions, in relative terms, are off by roughly one-fifth.

A graph of a function

Description automatically generatedAt the end is the ROC and Precision-Recall Curves:

A graph of a graph with a line

Description automatically generated with medium confidence

* The ROC curve suggests the model has a strong ability to distinguish between positive and negative classes.
* The Precision-Recall curve also indicates good performance, especially given an average of 0.90. However, the rapid drop in precision after a certain recall point suggests there may be challenges with simultaneously achieving high precision and recall.
* In real-world scenarios, the choice between precision and recall often depends on the specific business or application needs. The threshold can be adjusted depending on what's more important (precision or recall).
* Given the high AUC in the ROC curve and the high average in the precision-recall curve, the model seems to perform well overall, but the exact threshold should be selected based on the desired trade-off between precision and recall.