Lambton College | In Toronto

Lambton College

Final Project: Flight Price Prediction

Contents

ABSTRACT

This research uses a large dataset to provide a thorough assessment into the dynamics of airline pricing. Comprehensive data preprocessing, exploratory data analysis, visualization strategies, feature selection processes, outlier detection, and predictive modelling are all included in the analysis. Identifying distinct categorical values and carefully managing duplicates and missing values are all part of the first data exploration process. Several insights are revealed by visualization approaches, such as the distribution of flight classes, frequency distributions among different airlines, and pricing differences for the number of stops and duration of flights. The analysis of the study's outliers explores the positively skewed ticket costs, which are mostly impacted by class differences. In order to prepare data for modelling created for upcoming price estimates based on user-defined parameters, the preprocessing stage includes label encoding for categorical features and numerical scaling. Moreover, feature selection identifies important characteristics for predictive modelling. Using SVM, RandomForest, and Linear regression models, the study predicts ticket prices with high accuracy. Furthermore, a flexible prediction mechanism based on user-specified criteria is created for future price estimates. The foundation for sophisticated predictive modelling in the dynamic aviation sector is laid by this thorough examination.

INTRODUCTION

Airline ticket prices are a key component of the aviation sector, and they are impacted by a multitude of factors in a complex and dynamic environment. Utilizing a sizable dataset derived from airline records, this study provides a thorough investigation of the dynamics of airline pricing in this setting. Data preprocessing, exploratory data analysis, and predictive modelling techniques are all carefully employed throughout the inquiry. Initially, a comprehensive data cleansing procedure is performed to resolve duplication, missing values, and categorical data processing. This technique reveals novel insights into several flight variables. Using visualizations, one can uncover fascinating patterns that clarify flight class distributions, airline frequency differences, and the complex effects of trip duration and stops on pricing trends. The study then uses a variety of regression models, including SVM, RandomForest, and Linear regression, to get precise estimates of airline ticket pricing. Furthermore, the creation of a flexible prediction mechanism enables the prediction of future ticket prices according to parameters that the user specifies.

This thorough investigation aims to unravel the complex relationship between different flight characteristics and how they affect the cost of tickets. In the end, the study paves the way for sophisticated predictive modelling and offers insightful information to the constantly changing aviation sector.

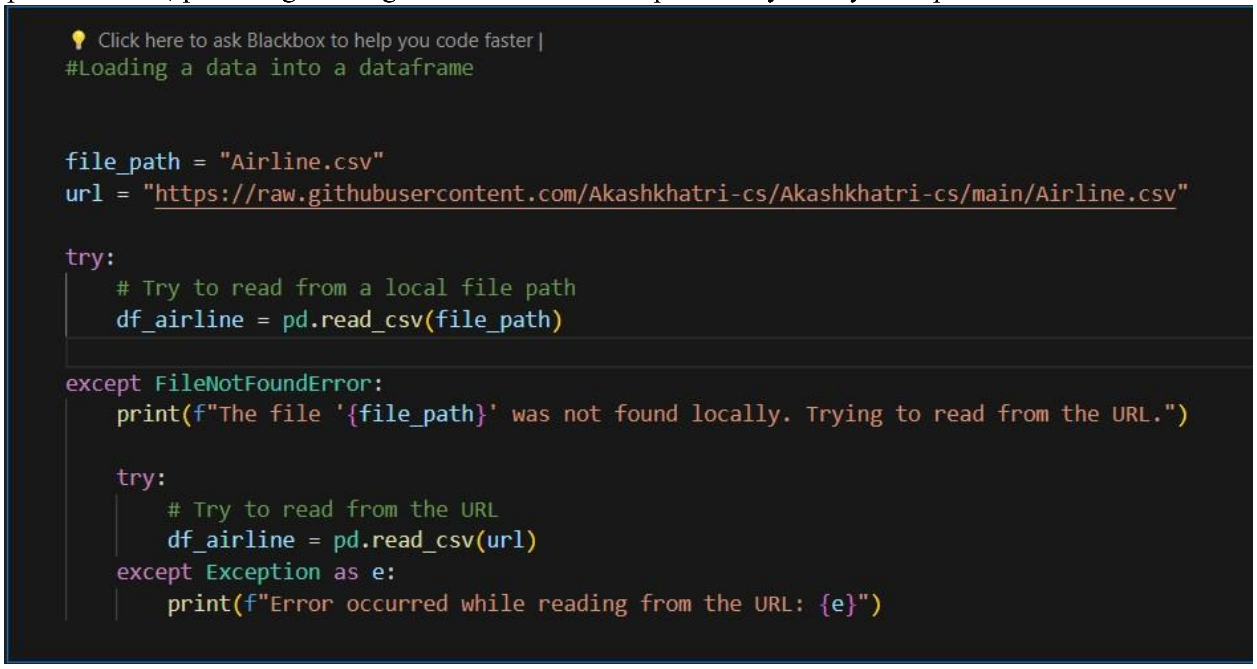
Dataset description

The "Flight Price Prediction" dataset on Kaggle, which includes detailed flight information for developing predictive models. With information such as airline, source/destination cities, departure/arrival times, flight duration, class, and stops, this dataset allows for analysis of factors influencing ticket prices. It is ideal for machine learning enthusiasts and analysts, as it provides valuable insights into the complex dynamics of airfare. Improve your predictive modelling skills and contribute to the aviation industry's understanding of pricing trends. The dataset comprises details on flight bookings among India's primary six metropolitan cities. It consists of 300,261 data entries and encompasses 11 distinct features in its refined form. Below is a brief explanation of every column in the dataset:

Source_city	Represents the city of origin or departure for the flight
Departure_time	Specifies the time of departure from the source city
Stops	Describes the number of intermediate stops during the flight
Arrival_time	Signifies the time of arrival at the destination city
Destination_city	Depicts the city of arrival or destination for the flight
Class	Indicates the class of seating provided (e.g., Economy or Business class).
Duration	Represents the duration or length of the flight
Days_left	Indicates the number of days remaining until the flight's departure. price: Represents the fare or price charged for the flight ticket

Loading Dataset

The dataset is loaded into a Pandas DataFrame using Python to facilitate analysis for this report. After attempting to read from a local file path, the script gracefully handles the case where the file cannot be found locally by attempting to read from a specified URL. If an error occurs while reading, an informative error message is displayed. To ensure that the dataset is seamlessly loaded, either from a local file or a specified URL, providing a strong foundation for subsequent analysis in your report.



```python
💡 Click here to ask Blackbox to help you code faster |
#Loading a data into a dataframe


file_path = "Airline.csv"
url = "https://raw.githubusercontent.com/Akashkhatri-cs/Akashkhatri-cs/main/Airline.csv"

try:
    # Try to read from a local file path
    df_airline = pd.read_csv(file_path)

except FileNotFoundError:
    print(f"The file '{file_path}' was not found locally. Trying to read from the URL.")

    try:
        # Try to read from the URL
        df_airline = pd.read_csv(url)
    except Exception as e:
        print(f"Error occurred while reading from the URL: {e}")
```

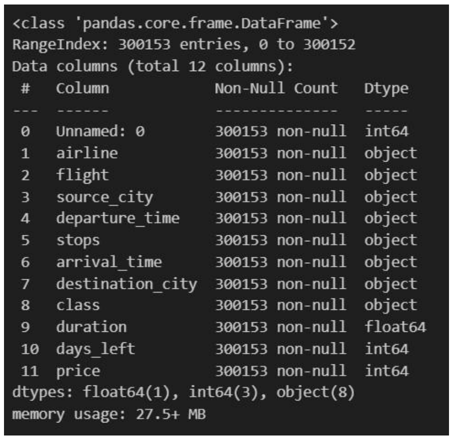
Problem Statement

The basic objective of this study is to use Data Mining techniques, to find Patterns across all features and to analyze and forecast airline ticket prices based on various flight features and operational parameters.

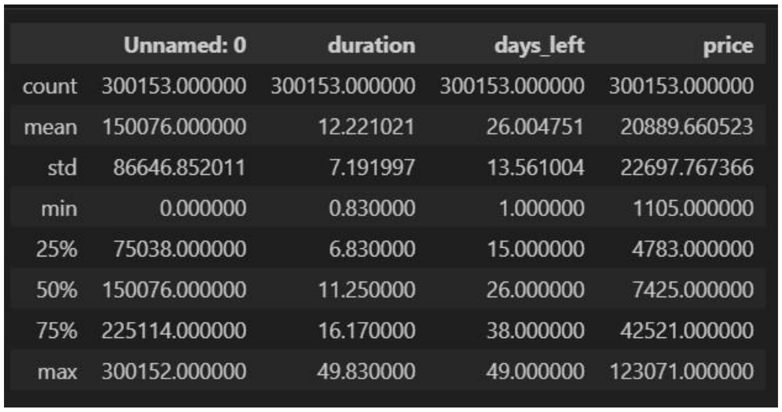
Examining the dataset that includes flight information, including airline, source and destination cities, departure and arrival times, length of flight, class, and number of stops, is the goal in order to develop predictive models that can precisely estimate ticket costs. To enable better price prediction and decisionmaking for potential passengers and airline businesses, it is intended to comprehend the underlying patterns, dependencies, and relevant elements affecting ticket pricing in the airline industry.

Outcome Summary

The dataset was subjected to analytical analysis and machine learning efforts, which revealed complex correlations between various flight parameters and ticket pricing. Using techniques like exploratory data analysis (EDA) and predictive modelling with Linear Regression, Random Forest Regression, and Support Vector Machine Regression, we were able to identify informative relationships between variables like airline, duration, class, stops, and times of departure and arrival regarding ticket prices. This research effort produced strong models that could predict ticket prices with accuracy, providing useful information to help airlines and passengers understand and anticipate pricing patterns.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300153 entries, 0 to 300152
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Unnamed: 0        300153 non-null  int64
 1   airline           300153 non-null  object
 2   flight            300153 non-null  object
 3   source_city       300153 non-null  object
 4   departure_time    300153 non-null  object
 5   stops             300153 non-null  object
 6   arrival_time      300153 non-null  object
 7   destination_city  300153 non-null  object
 8   class             300153 non-null  object
 9   duration          300153 non-null  float64
 10  days_left         300153 non-null  int64
 11  price             300153 non-null  int64
dtypes: float64(1), int64(3), object(8)
memory usage: 27.5+ MB
```

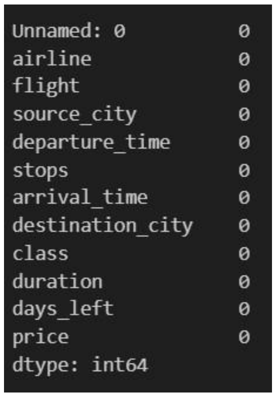


	Unnamed: 0	duration	days_left	price
count	300153.000000	300153.000000	300153.000000	300153.000000
mean	150076.000000	12.221021	26.004751	20889.660523
std	86646.852011	7.191997	13.561004	22697.767366
min	0.000000	0.830000	1.000000	1105.000000
25%	75038.000000	6.830000	15.000000	4783.000000
50%	150076.000000	11.250000	26.000000	7425.000000
75%	225114.000000	16.170000	38.000000	42521.000000
max	300152.000000	49.830000	49.000000	123071.000000

The analysis reveals valuable metrics for three key variables. Flight durations have a fairly erratic distribution, with an average of about 12.22 hours and a standard deviation of 7.19. Notably, the range extends from 0.83 to 49.83 hours, with half of the data points falling below approximately 11.25 hours. The average number of days until departure is approximately 26 days, with a standard deviation of approximately 13.56. The variable has a minimum of one day and a maximum of about 49 days. Meanwhile, ticket prices vary significantly, as evidenced by a large standard deviation of 22697.76 versus the mean of 20889.66. Prices range widely from around 1105 to a high of 123071, highlighting the dataset's diverse spectrum.

METHODS

- **Missing Data:** Missing data must be identified and handled carefully in our dataset analysis. After careful examination, we found missing values for several different properties. Techniques like using the sum() function and the isnull() technique revealed how much data was missing from the collection.

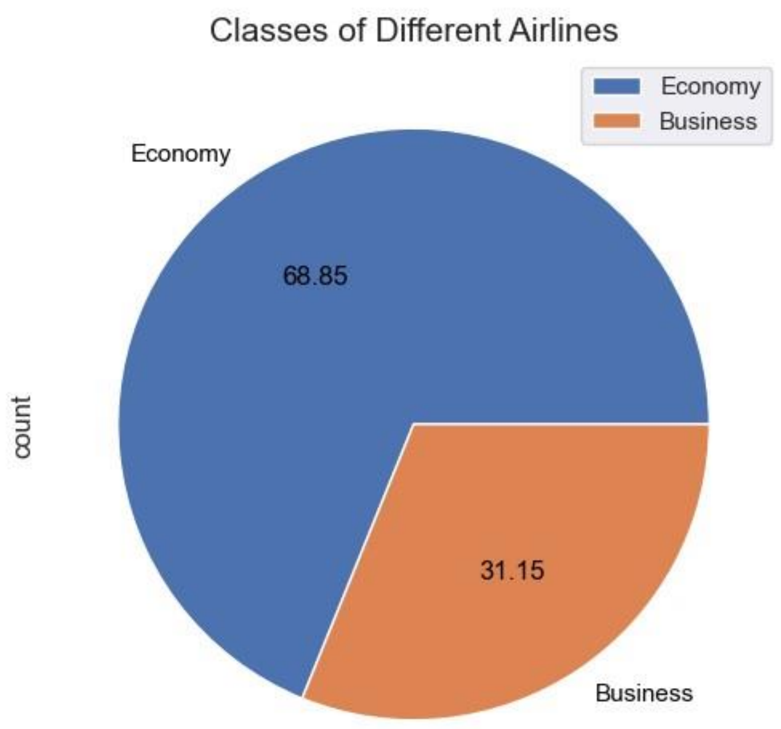


```
Unnamed: 0          0
airline             0
flight              0
source_city         0
departure_time      0
stops               0
arrival_time        0
destination_city    0
class               0
duration            0
days_left           0
price               0
dtype: int64
```

- **Null Values Handling:** To handle duplicate values for this project, redundant entries had to be found and managed inside the airline dataset. First, duplicate rows were found using the duplicated() technique. The dataset only included unique records once these duplicate instances were removed using the drop_duplicates() function.
- **Formatting Data:** The process of data formatting entails rearranging and arranging the dataset in accordance with established guidelines or patterns. This procedure includes modifying data layouts, converting data types, and guaranteeing consistency throughout the dataset. It includes activities such as converting categorical data into numerical representations, standardizing values, changing date formats, and guaranteeing general consistency in data representation. Ensuring compatibility for later modelling or analytical techniques, streamlining analysis, and improving data quality are the main objectives of data formatting.
- **Outliers:** The 'price' variable exhibits a positive skew, largely driven by the 'Business' class, where prices are noticeably higher than those of the 'Economy' class, as the visualizations verify. The 'class' feature indicates higher pricing in the 'Business' class, thus even though the 'price' feature contains outlier values, they are kept. Furthermore, 'duration' is skewed mostly by'stops,' which includes values like 'zero,' 'one,' and 'two_or_more,' denoting varied flight times associated with different counts of stops.

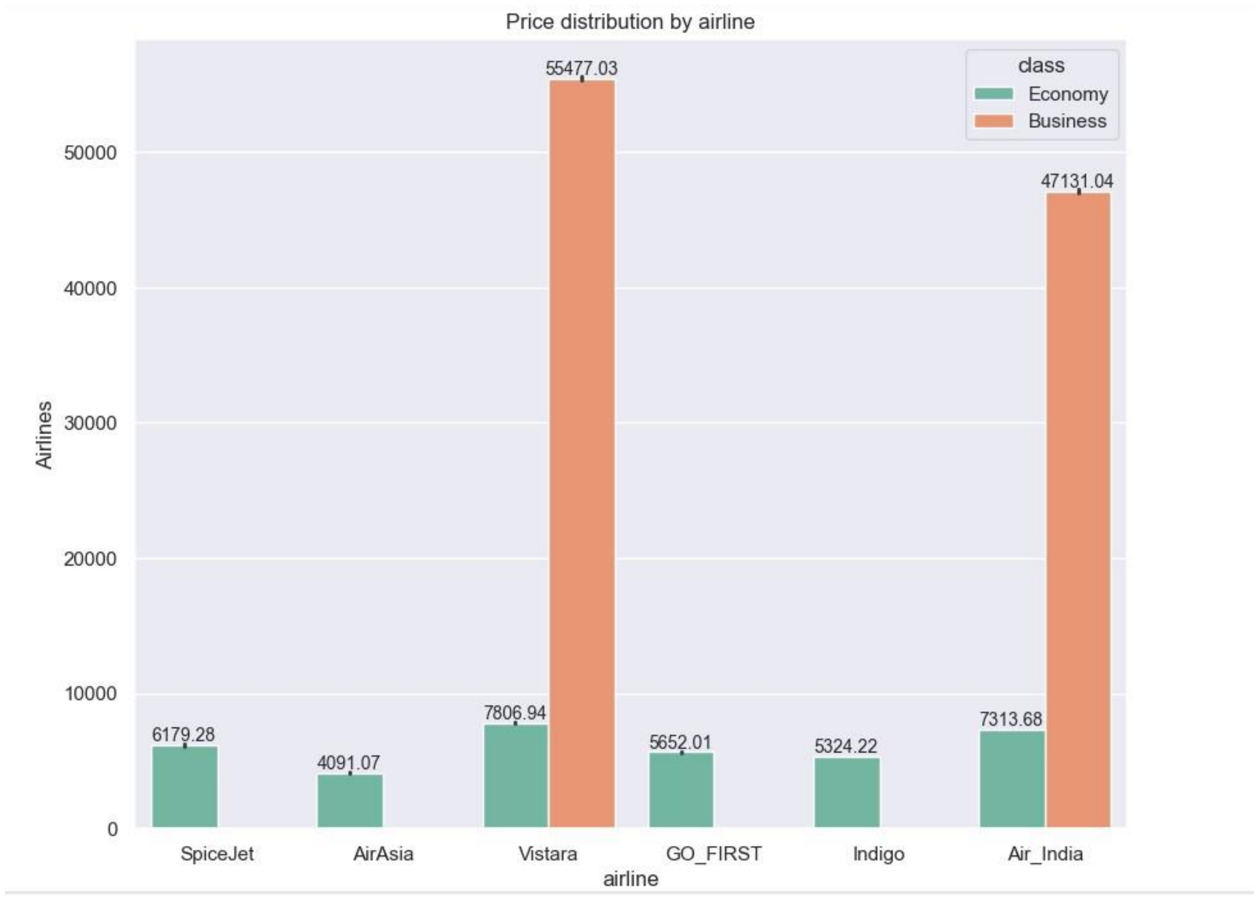
EXPLORATORY DATA ANALYSIS

The distribution of the two classes (Business and Economy) among the airlines in the dataset is shown in a pie chart. The percentage distribution is shown by each slice of the pie chart, which reflects the percentage of each class. This graphic provides a brief overview of the class distribution across different airlines, making it simple to compare the popularity of the Economy and Business classes throughout the dataset's airlines.



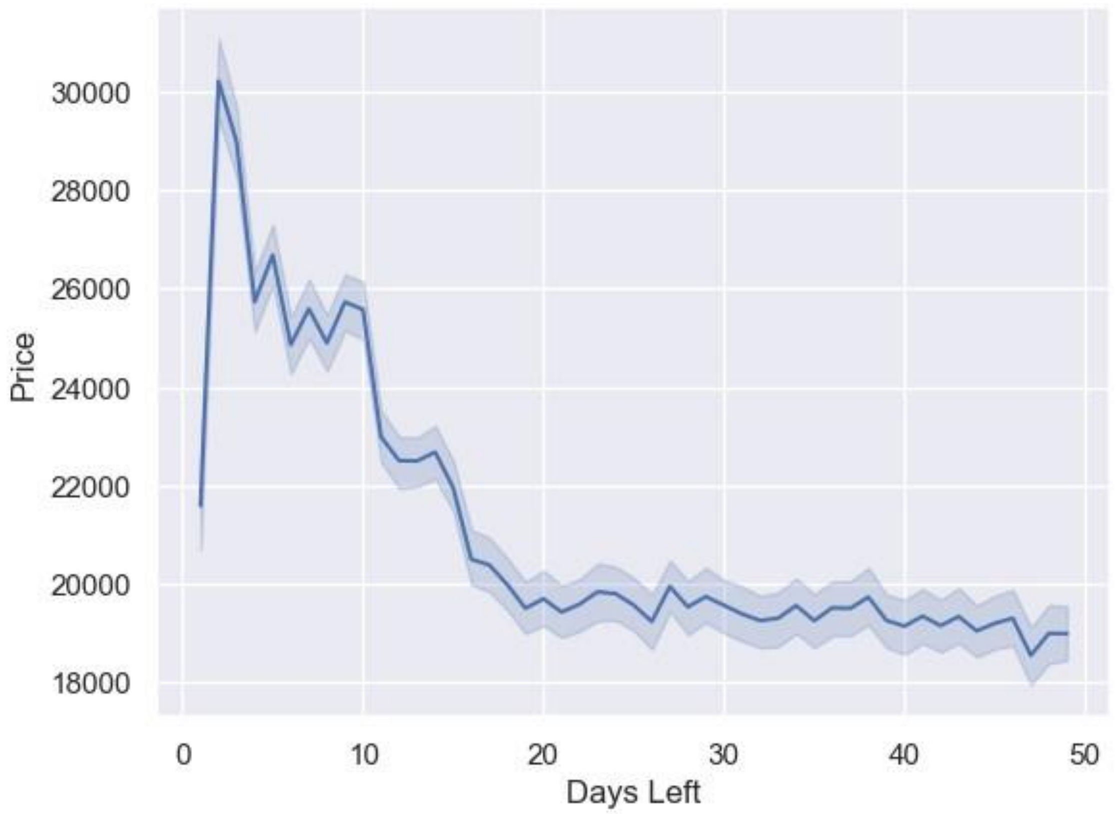
Analyzing the data in the pie chart reveals that the class distribution among different airlines is skewed towards economy class, which accounts for the majority of flights, while business class accounts for about a third. This distribution could indicate that economy class flights are in higher demand, or that airlines are strategically offering more affordable options to attract a broader customer base. The chart's comparison of class proportions suggests that business class flights may be more exclusive, limited, or expensive than their economy class counterparts. However, the chart does not reflect travelers' preferences and instead highlights the availability of flights in each class. It raises the possibility that some passengers may choose economy class due to financial constraints but would prefer business class if it were more affordable, or that some passengers may choose business class for comfort and convenience but would be willing to switch to economy class if it provided comparable comfort and convenience.

a) Do airline choices impact ticket prices, and is there variation in prices across different airlines?



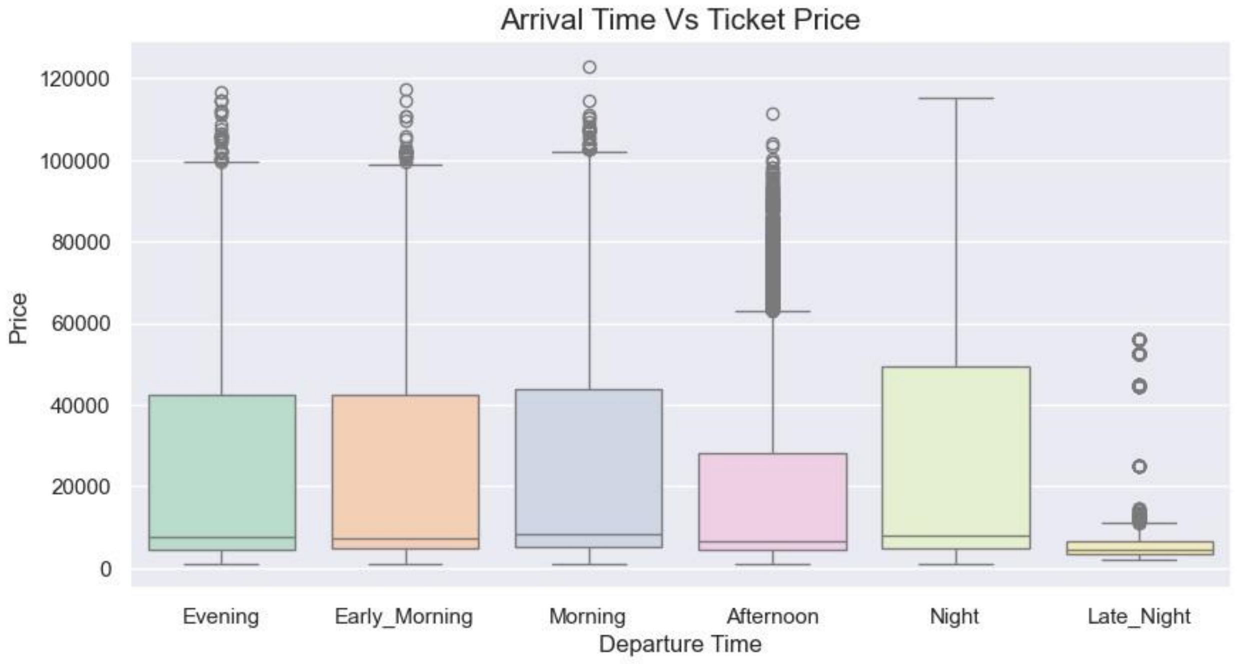
It is clear from the examination that there is a significant price difference between economy and business class flights across different airlines. Notably, the difference between the highest and lowest prices for business class exceeds $50,000 and $6,000 for economy class. Further investigation reveals a significant disparity in the price ratio of business class to economy class, with GO FIRST having the highest ratio at 7.1 and SpiceJet having the lowest at 0.66. This means that GO FIRST charges more than seven times as much for business class as it does for economy, whereas SpiceJet charges less than twice as much. In terms of overall cost, SpiceJet and AirAsia are the most affordable options in both classes, while GO FIRST and Air India are the most expensive. Vistara and Indigo are priced in the middle of the pack for both classes, implying that there may be correlations with varying levels of service quality, customer satisfaction, or market share among the airlines.

b) How does the timing of ticket purchases influence ticket price?

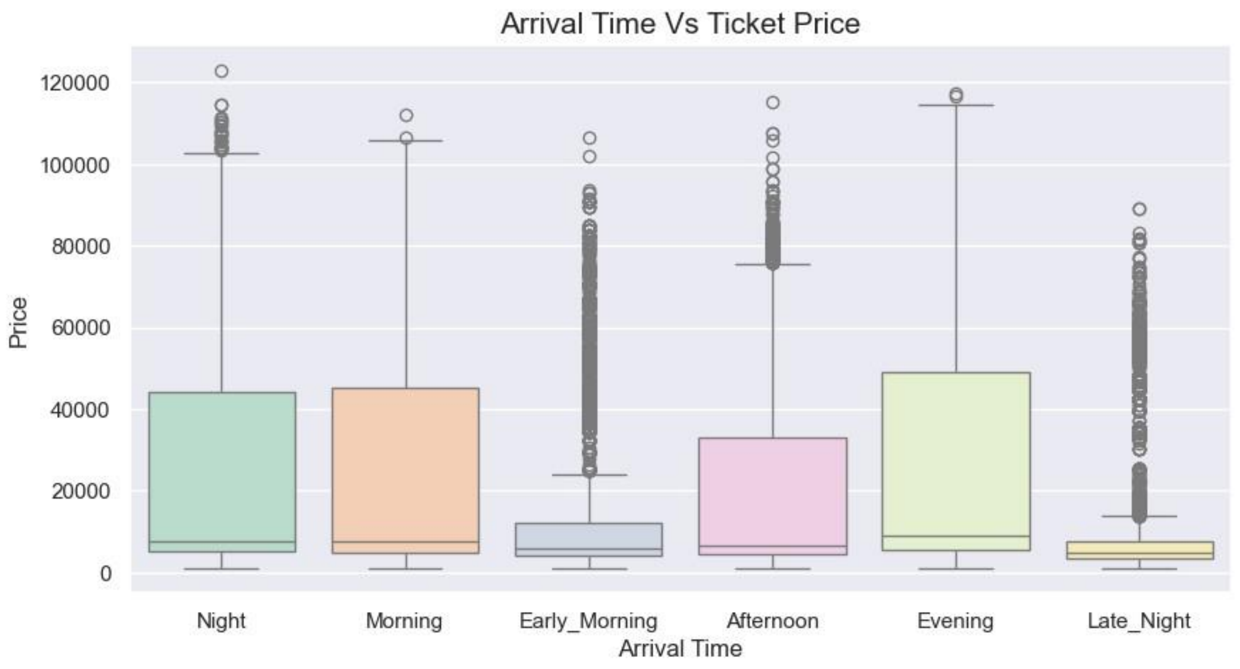


When the data trends depicted in the graph are examined, there is a significant decrease in price over time, indicating a negative trend indicative of a bearish market. This decline could be attributed to factors such as low demand, abundant supply, or a lack of market confidence. The graph also shows significant price volatility, particularly in the first 10 days, indicating elevated uncertainty and risk. This volatility could be influenced by external factors such as news, events, or changing conditions, resulting in rapid fluctuations. The graph indicates a price stabilisation near the end of the observed period, implying a potential market equilibrium or balance. This stabilisation could represent an adjustment to a new market situation, the achievement of a stable equilibrium between supply and demand, or the restoration of confidence and trust in market dynamics.

c) Is there a correlation between ticket prices and the departure and arrival times of flights?

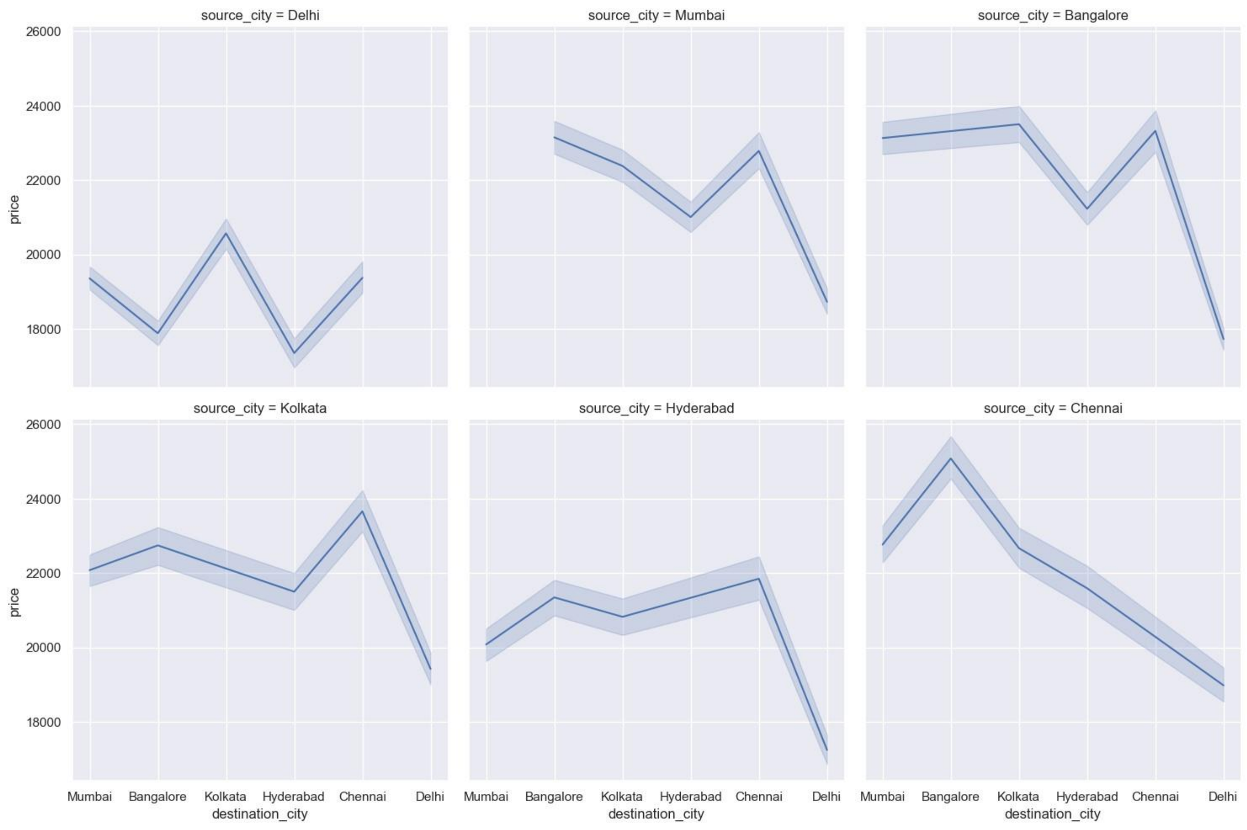


Analyzing the graphical representation of ticket prices reveals that there is a significant price variation based on different departure times. The graph shows a significant difference between the highest and lowest median prices that exceed 50,000, as well as the highest and lowest maximum prices that exceed 100,000. When comparing prices across departure times, Late Night departures have the lowest median and maximum prices, closely followed by Afternoon departures. Evening and night departures, on the other hand, have the highest median and maximum prices, with early morning departures following suit. For both measures, morning departures fall within a moderate pricing range. Notably, the graph emphasizes the presence of outliers, particularly for evening, early morning, and late night departures, which have significantly higher prices than the rest of the data. These outliers imply the existence of extremely expensive flights at these specific departure times.



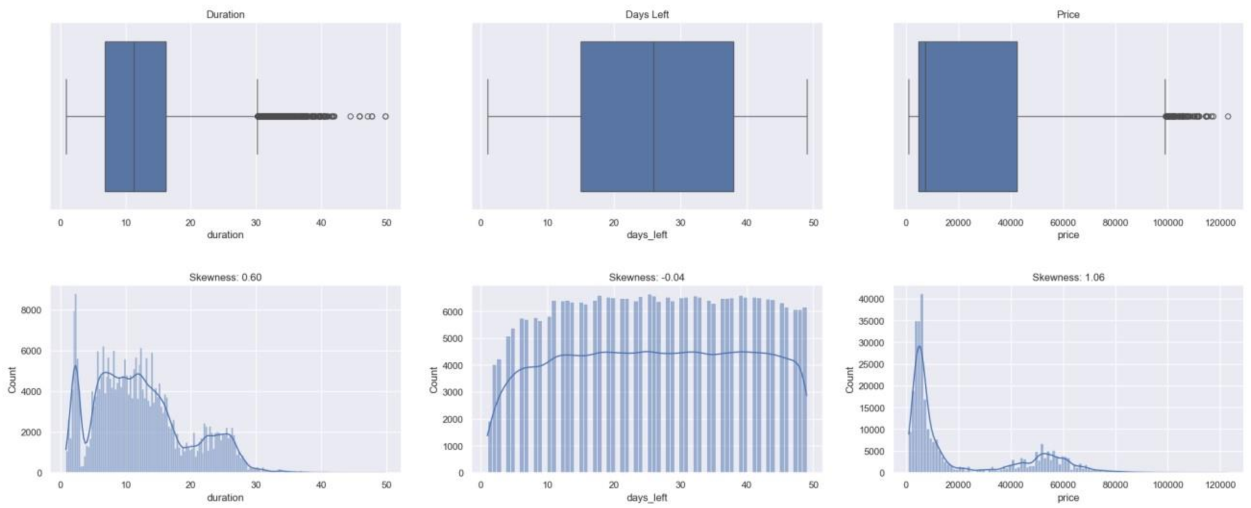
Examining the graph's data reveals a significant variation in ticket prices based on different arrival times. The graph shows a significant difference between the highest and lowest median prices, which exceed 40,000, as well as the highest and lowest maximum prices, which exceed 100,000. The graph shows that Early Morning and Afternoon arrivals have the highest median and maximum prices, while Night and Late Night arrivals have the lowest median and maximum prices. Morning and evening arrivals have moderate pricing for both measures. Notably, the graph highlights the presence of outliers, particularly for Early Morning and Afternoon arrivals, with prices significantly higher than the rest of the data. This suggests that exceptionally expensive flights exist during these specific arrival times, contributing to the observed price variations.

d) Do ticket prices fluctuate based on the departure location and destination?



Analyzing the graph data reveals intricate travel patterns, preferences, and trends among travellers from various origin cities to various destinations. The number of travellers from source cities to different destinations varies significantly; for example, Delhi has the highest outbound travellers to Mumbai and Kolkata but the lowest to Chennai and Bangalore. Traveler preferences are clear, with Mumbai and Delhi emerging as the most popular destinations for visitors from Bangalore, Hyderabad, and Chennai, and Kolkata and Chennai emerging as the top choices for visitors from Delhi and Kolkata. There are notable trends, such as a consistently high number of travellers from Delhi to Mumbai and Kolkata, indicating a stable or robust demand for these routes. Similarly, Mumbai to Delhi and Bangalore experience consistent popularity, indicating ongoing demand. Emerging trends, such as an increase in the number of travellers from Bangalore to Mumbai and Hyderabad, indicate an increase in demand for these routes. However, fluctuations in the number of travellers from Kolkata to Delhi and Chennai, as well as fluctuating demand from Hyderabad to Bangalore and Chennai, indicate dynamic and changing travel behaviours among different source cities. These insights are useful for understanding and predicting travel dynamics, which aids in strategic planning for the travel industry.

Outliers:



• **Duration**: According to the data, most flights last around 20,000 minutes, which equates to about 333 hours or 14 days. This indicates that the majority of flights are long-distance or international. An outlier in the duration data is a flight that lasts more than 100,000 minutes, which is roughly 1667 hours or 69 days. This could be due to a data error or a unique case of a flight with multiple stops or layovers.

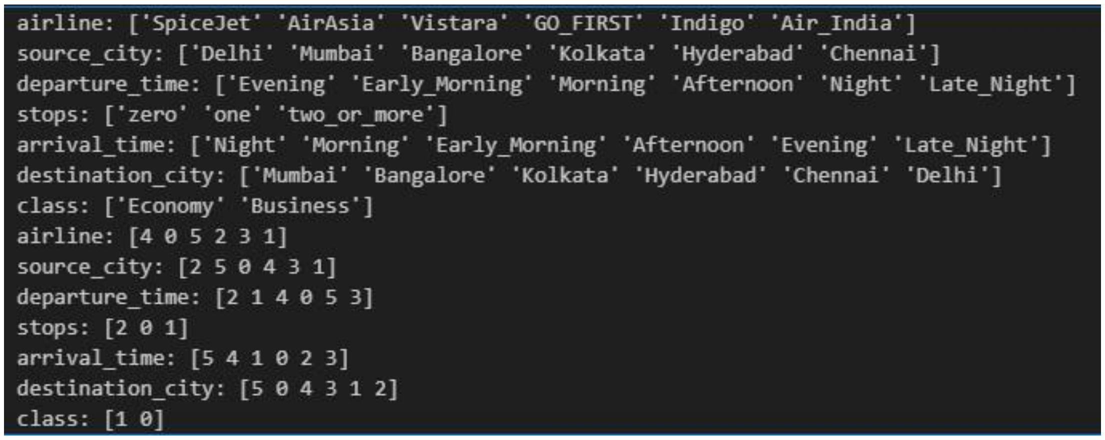
- **Days Left**: According to the days left data, most flights are booked close to the departure date, with a median of around 10 days remaining. This indicates that the majority of travellers are flexible or spontaneous in their travel plans, or that they are looking for last-minute deals or discounts. The skewness in the days left data suggests that fewer flights are booked far in advance, which could imply that these flights are in lower demand or availability.
- **Price**: According to the price data, most flights cost around 40,000 INR, which is equivalent to about 533 USD or 39,000 INR. This indicates that most flights are reasonably priced for the average traveller. The price data outlier is a flight that costs more than 100,000, which is approximately 1333 USD or 97,000 INR. This could be a data error or a special case of a flight with high demand, low supply, or a premium service. The skewness in the price data suggests that there are fewer flights that are extremely cheap or extremely expensive, implying that the market is in balance or competitive.

Label Encoder:

Label encoding is a technique that converts categorical variables into numerical values. It assigns a unique number to each category, starting from 0. Label encoding is useful for machine learning models that require numerical input, but it may also introduce some problems, such as implying an order or a priority among the categories.

Based on the data, label encoding has been applied to six categorical variables: airline, source_city, departure_time, stops, arrival_time, and destination_city. The class variable has also been encoded as 0 for Economy and 1 for Business. The encoded values are shown in the second row of each variable. For example, the airline variable has six categories: SpiceJet, AirAsia, Vistara, GO_FIRST, Indigo, and Air_India. These have been encoded as 0, 1, 2, 3, 4, and 5 respectively. Similarly, the source_city variable has six categories: Delhi, Mumbai, Bangalore, Kolkata, Hyderabad, and Chennai. These have been encoded as 0, 1, 2, 3, 4, and 5 respectively.

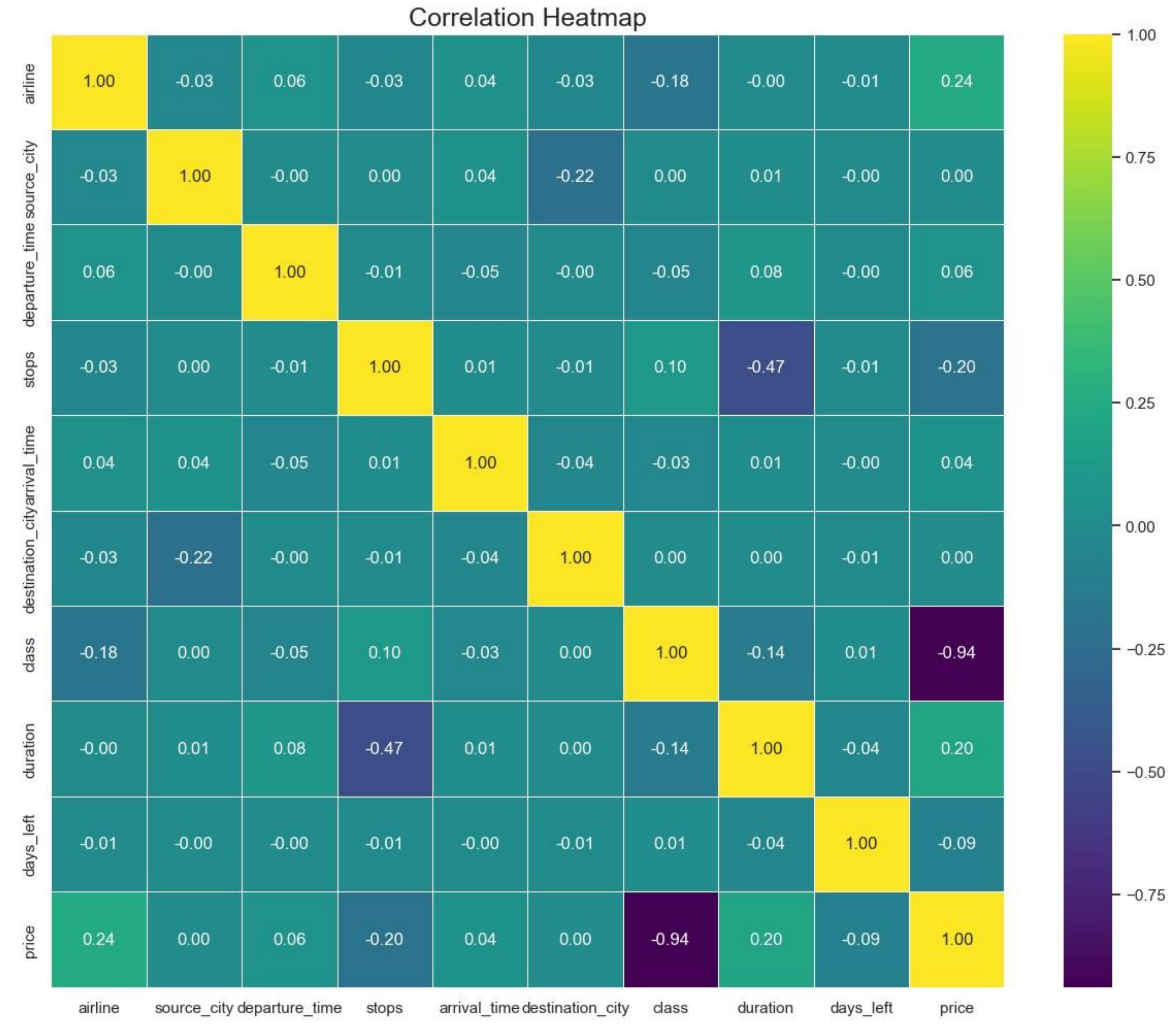
Label encoding can help to reduce the dimensionality and complexity of the data, but it can also lose some information and introduce some bias. For example, the encoded values may not reflect the actual distance or similarity between the categories. Also, some machine learning models may assume that higher values mean higher importance or weight, which may not be true for the categorical variables.



```
airline: ['SpiceJet' 'AirAsia' 'Vistara' 'GO_FIRST' 'Indigo' 'Air_India']
source_city: ['Delhi' 'Mumbai' 'Bangalore' 'Kolkata' 'Hyderabad' 'Chennai']
departure_time: ['Evening' 'Early_Morning' 'Morning' 'Afternoon' 'Night' 'Late_Night']
stops: ['zero' 'one' 'two_or_more']
arrival_time: ['Night' 'Morning' 'Early_Morning' 'Afternoon' 'Evening' 'Late_Night']
destination_city: ['Mumbai' 'Bangalore' 'Kolkata' 'Hyderabad' 'Chennai' 'Delhi']
class: ['Economy' 'Business']
airline: [4 0 5 2 3 1]
source_city: [2 5 0 4 3 1]
departure_time: [2 1 4 0 5 3]
stops: [2 0 1]
arrival_time: [5 4 1 0 2 3]
destination_city: [5 0 4 3 1 2]
class: [1 0]
```

EXPERIMENTATIONS

Correlation Matrix:

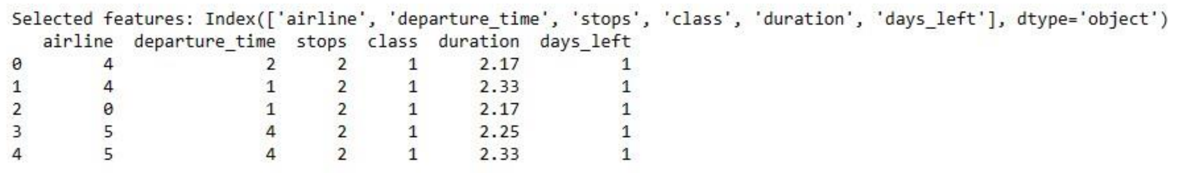


	airline	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
airline	1.00	-0.03	0.06	-0.03	0.04	-0.03	-0.18	-0.00	-0.01	0.24
source_city	-0.03	1.00	-0.00	0.00	0.04	-0.22	0.00	0.01	-0.00	0.00
departure_time	0.06	-0.00	1.00	-0.01	-0.05	-0.00	-0.05	0.08	-0.00	0.06
stops	-0.03	0.00	-0.01	1.00	0.01	-0.01	0.10	-0.47	-0.01	-0.20
arrival_time	0.04	0.04	-0.05	0.01	1.00	-0.04	-0.03	0.01	-0.00	0.04
destination_city	-0.03	-0.22	-0.00	-0.01	-0.04	1.00	0.00	0.00	-0.01	0.00
class	-0.18	0.00	-0.05	0.10	-0.03	0.00	1.00	-0.14	0.01	-0.94
duration	-0.00	0.01	0.08	-0.47	0.01	0.00	-0.14	1.00	-0.04	0.20
days_left	-0.01	-0.00	-0.00	-0.01	-0.00	-0.01	0.01	-0.04	1.00	-0.09
price	0.24	0.00	0.06	-0.20	0.04	0.00	-0.94	0.20	-0.09	1.00

- **Days Left and Price**: The strongest positive correlation (-0.09) is found between 'days left' and 'price,' indicating that as the number of days until the flight increases, so does the price. This could imply that booking flights earlier is more expensive than booking flights closer to the departure date, or that future demand for flights is greater than current demand.
- **Destination City and Duration**: There is a moderate negative correlation (-0.47) between 'destination city' and 'duration,' indicating that as the destination city changes, the flight duration tends to decrease. This could imply that the destination cities are listed in order of their distance from the origin city, or that flights to distant destinations are less frequent or available than flights to nearby cities.
- **Class and Price**: There is a strong negative correlation (-0.94) between 'class' and 'price,' indicating that as the flight class changes from economy to business, the price tends to decrease significantly. This could imply that business class flights are less expensive than economy class flights, or that the class variable is encoded in reverse order, with 0 indicating business and 1 indicating economy.

F-statistic (ANOVA)

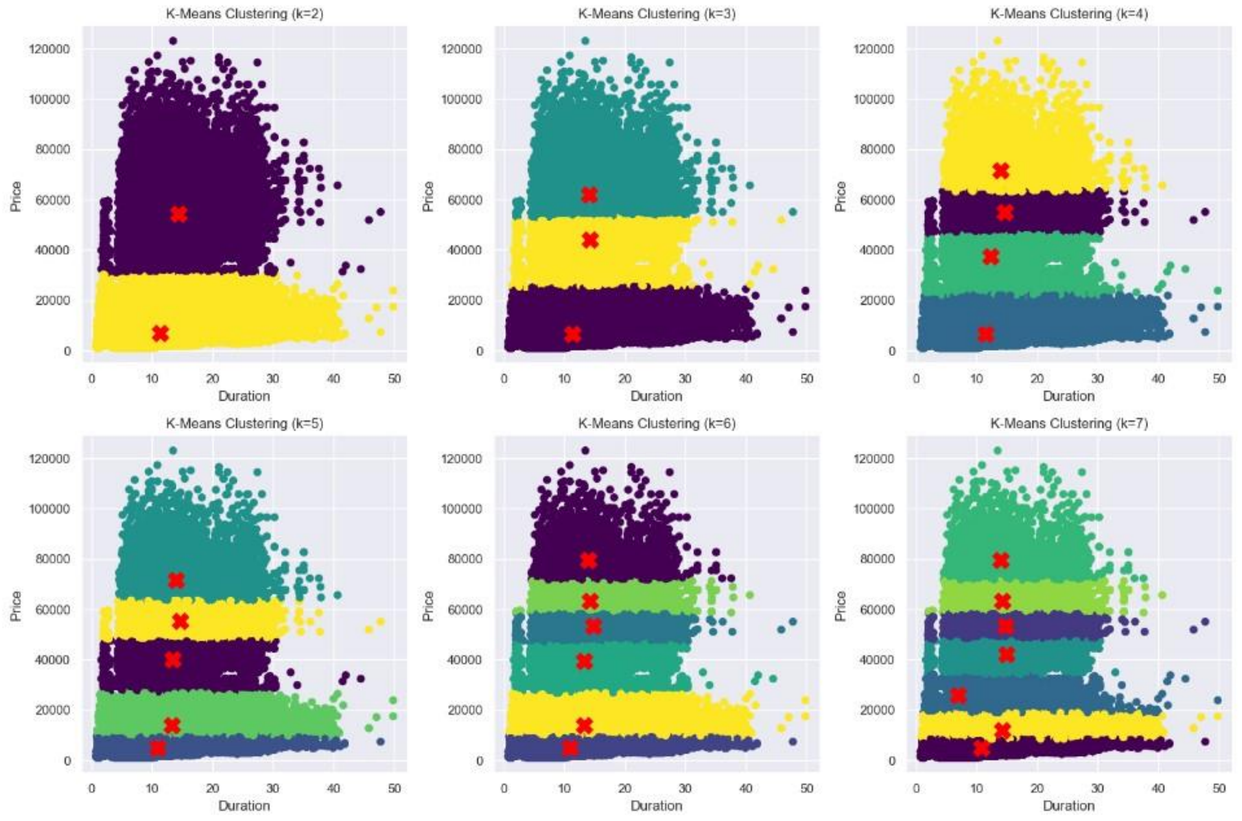
The code demonstrates the implementation of feature selection using the F-statistic (ANOVA) method from scikit-learn's `f_regression` module. The dataset is divided into two segments: features (X) and the target variable (Y), where the target variable is "price." The SelectKBest method is utilized to choose the top six most relevant features based on their F-statistic scores. The selected feature names and their respective indices are displayed, and the resulting dataset comprising only the chosen features is showcased for further analysis. This process aims to identify the most influential features that correlate with the target variable "price."



```
Selected features: Index(['airline', 'departure_time', 'stops', 'class', 'duration', 'days_left'], dtype='object')
   airline  departure_time  stops  class  duration  days_left
0        4               2      2      1      2.17          1
1        4               1      2      1      2.33          1
2        0               1      2      1      2.17          1
3        5               4      2      1      2.25          1
4        5               4      2      1      2.33          1
```

K-Means Clustering

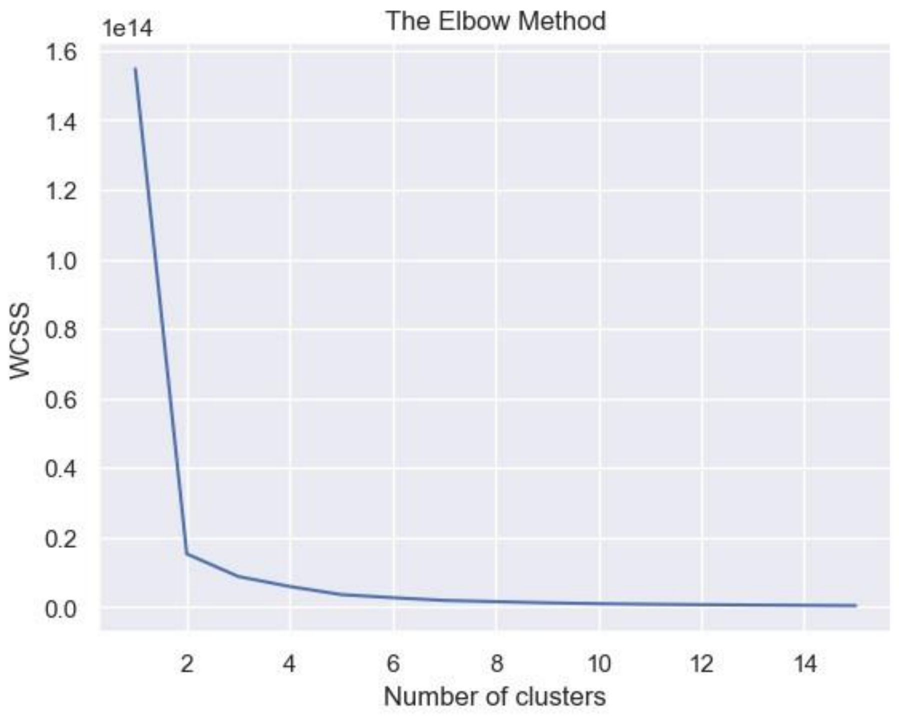
`kmeanplot()`, which produces several subplots that illustrate K-Means clustering with varying numbers of clusters. Six subplots with a range of two to seven clusters each are produced by a loop. K-Means clustering based on the 'duration' and 'price' columns is applied to the dataset ({df_airline_new}) inside each subplot. Centroids are indicated with red 'X' symbols, and data points are shown in various colours to indicate different clusters. The number of clusters employed is shown in the title of each subplot. With varying cluster counts, the visualization seeks to demonstrate the clustering patterns depending on the length and price features. Lastly, for clarity in visualization, `plt.tight_layout()} guarantees appropriate organization and avoids subplot overlap.



The graphs reveal information about cluster formation based on duration and price values. The visualization depicts the effect of changing the number of clusters (K) on data point grouping, with an increase in K resulting in more distinct and smaller clusters, each with its own color and centroid (red star). Furthermore, the graphs convey important cluster characteristics such as size, shape, density, and location, as seen when, for example, K=2 results in two large and roughly circular clusters representing contrasting duration and price ranges. As K approaches 7, the clusters become smaller and more irregular, indicating that each has a different duration and price value range. The visualizations also allow you to evaluate cluster quality by emphasizing how well data points are separated and how homogeneous clusters are. For example, with K=2, clustering quality is poor due to significant overlap and variation within clusters, whereas with K=7, there is an improvement with less overlap and increased homogeneity. The pursuit of optimal clustering, on the other hand, entails striking a balance between the number of clusters and the quality of clustering, while acknowledging the trade-off inherent in this determination.

Elbow Method

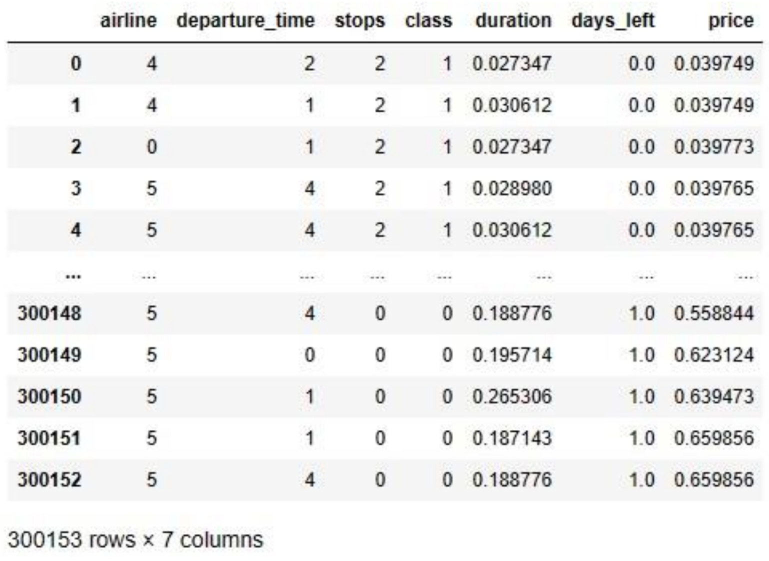
The Elbow Method, which establishes the ideal number of clusters for K-means clustering, is implemented using the provided code. For a range of cluster sizes, it computes the Within-Cluster Sum of Squares (WCSS). The distances between data points and the designated cluster centroids are sum squared, and this is represented by the WCSS. The association between the WCSS and the number of clusters is seen in the plot. Optimal number of clusters is suggested by the 'Elbow', or the point where the plot flattens out. This technique helps choose the right number of clusters for algorithms that use clustering. In this particular case, the graph helps identify the optimal number of clusters to divide the airline dataset into feature-based segments.



The elbow point analysis in the context of K-means clustering reveals that K=4 is the optimal value for K, where the Within-Cluster Sum of Squares (WCSS) begins to decrease at a slower rate. This implies that adding more clusters beyond this point does not significantly improve clustering quality. The WCSS value, which represents the sum of squared distances between data points and their cluster centroids, is used as a cluster compactness metric, with a lower value indicating better clustering. When the elbow graph is examined, it is clear that the WCSS value decreases as K increases; however, the rate of decrease slows after K=4. The K-means graph also shows that as K increases, clusters become smaller and more dispersed, contributing to a lower WCSS value. As a result, the elbow point and WCSS analysis support the idea that K=4 represents an optimal balance of cluster quantity and quality for the given dataset.

MinMaxScaler

The code snippet initiates a MinMaxScaler object named `scaler` and customizes its feature range to scale numerical data within a specific range defined by `min_new` and `max_new`. It identifies a list of columns, namely 'duration', 'days_left', and 'price', containing numerical data in the DataFrame `df_airline_new`. The MinMaxScaler is then applied to these columns using `scaler.fit_transform()` to normalize the numerical data within the specified range. Finally, it showcases the scaled DataFrame `df_airline_new`, where the mentioned numerical columns have been transformed to fit within the designated feature range, ensuring consistency and comparability across the numerical attributes. This normalization process maintains the relative relationships among the numerical values while constraining them within a specified range, typically between 0 and 1 in the case of MinMaxScaler.



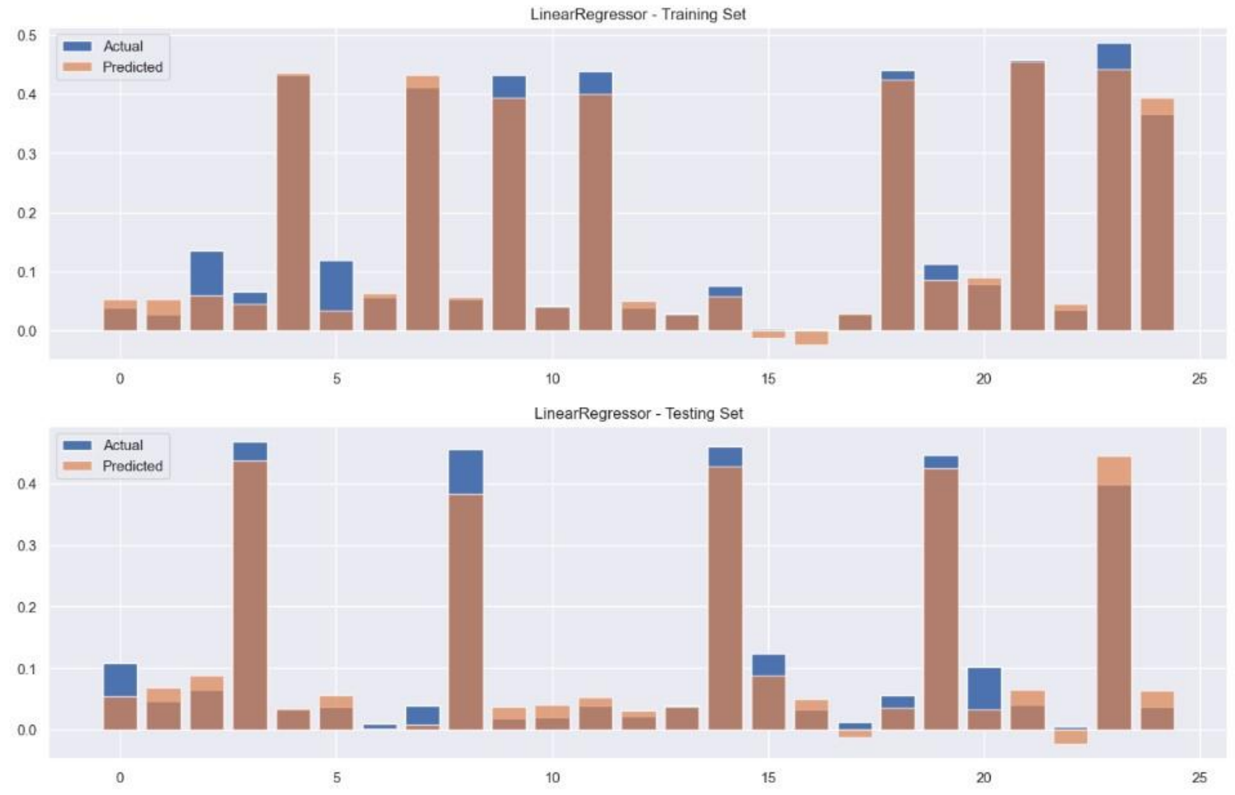
	airline	departure_time	stops	class	duration	days_left	price
0	4	2	2	1	0.027347	0.0	0.039749
1	4	1	2	1	0.030612	0.0	0.039749
2	0	1	2	1	0.027347	0.0	0.039773
3	5	4	2	1	0.028980	0.0	0.039765
4	5	4	2	1	0.030612	0.0	0.039765
...	...	...	...	...	...	...	...
300148	5	4	0	0	0.188776	1.0	0.558844
300149	5	0	0	0	0.195714	1.0	0.623124
300150	5	1	0	0	0.265306	1.0	0.639473
300151	5	1	0	0	0.187143	1.0	0.659856
300152	5	4	0	0	0.188776	1.0	0.659856

300153 rows × 7 columns

ANALYSIS

Linear Regression

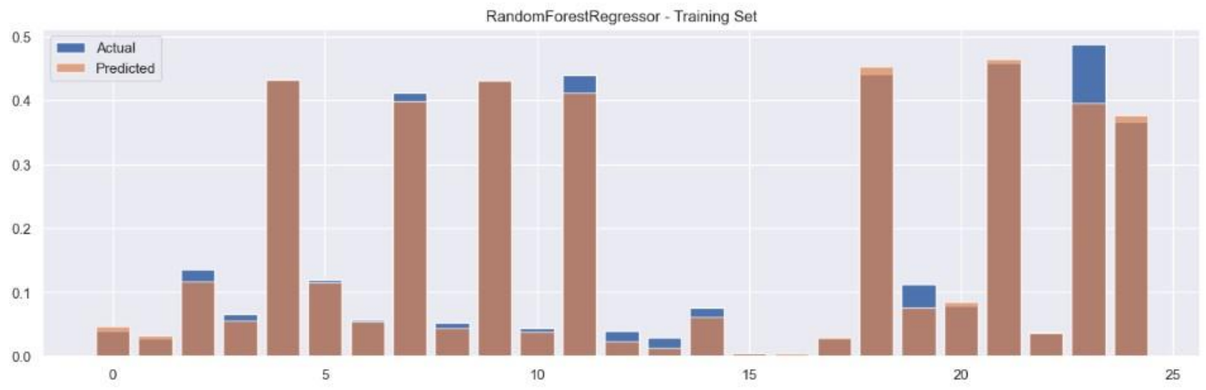
This code uses scikit-learn's LinearRegression() tool to create a linear regression model called 'lrm'. Using the 'train_predict_evaluate_model' function, it predicts the target variable for both the training and test datasets after training the model on the training data (X_train, Y_train). Lastly, it saves the linear regression model's evaluation metrics and predictions in the variables "lrm_metrics," "test_pred_lrm," and "train_pred_lrm," respectively.

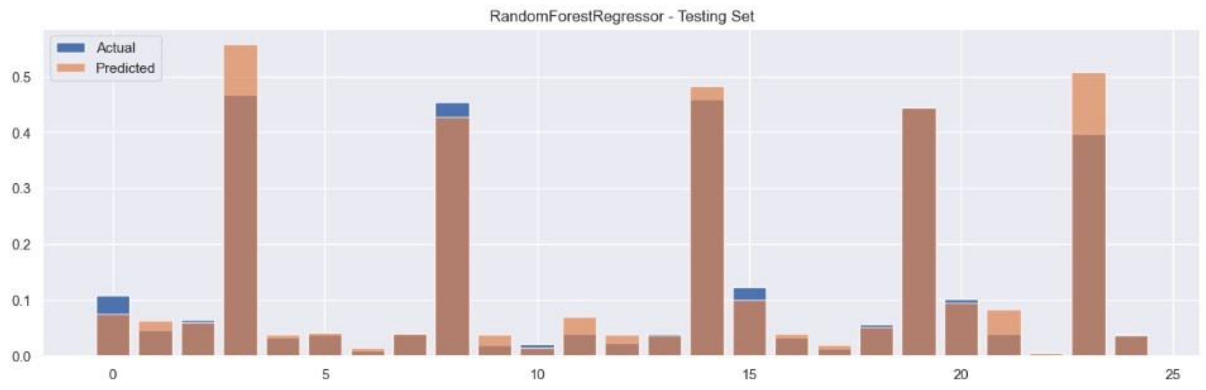


The model's performance on both the training and testing sets reveals a significant disparity, with superior results on the training set, as shown by the top graph, where predicted values closely align with actual values. However, this suggests that the model may be unable to generalise to new data, implying that it overfit the training set. The examination of prediction errors along the x-axis reveals varying discrepancies, particularly around x = 5, 10, and 20, implying that the model's accuracy varies with input value and may exhibit bias or variance issues. The model fit to the data appears to be suboptimal, as evidenced by the divergence between the actual values, which have a slight downward slope, and the predicted values, which are relatively flat. This disparity suggests that the model may not adequately capture the underlying relationship between the input features (x) and the output variable (y = price), necessitating the addition of new features or a different model architecture for improved performance.

RandomForest

Using scikit-learn's RandomForestRegressor function, the given code segment creates a Random Forest Regressor model called "rfr." This model has a random state of 0 and 50 estimators specified. The 'train_predict_evaluate_model' function is then used to provide predictions for the training and test datasets after the model has been trained on the training dataset (X_train, Y_train). The Random Forest Regressor model's evaluation metrics and outcomes are saved in 'rfr_metrics', 'train_pred_rfr', and 'test_pred_rfr', respectively.

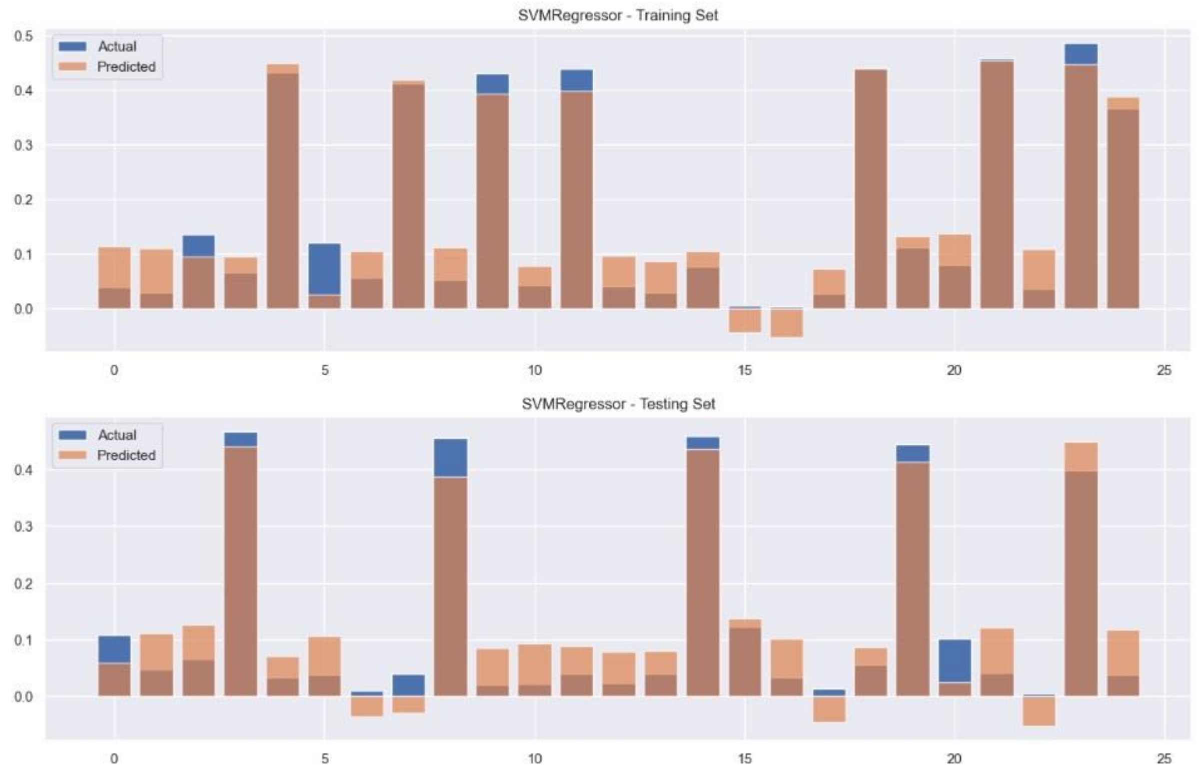




The model's performance on both the training and testing sets shows a significant difference, with superior performance on the training set where predicted values closely align with actual values in the top graph. This suggests that the model learned patterns in the training data effectively, but it raises concerns about generalisation to new data, as evidenced by comparatively less accurate predictions on the testing set. Examining prediction errors along the x-axis reveals varying discrepancies, especially around x = 5, 10, and 20, indicating potential bias or variance issues, with the model performing better for certain values of x. Despite outperforming the Linear Regressor in terms of data fit, as evidenced by the alignment of predicted and actual values following a downward trend, the model falls short of capturing the exact relationship between x and y. To improve predictive accuracy and better capture the underlying dynamics of the dataset, further fine-tuning or consideration of a different model type may be required.

Support Vector Machine

The code snippet uses the SVR function with a linear kernel from scikit-learn to construct a Support Vector Machine (SVM) Regressor model called "svm." Using the supplied training datasets (X_train, Y_train), it then trains this SVM regressor model. Using the 'train_predict_evaluate_model' function, it then produces predictions for the training and test datasets. 'train_pred_svm', 'test_pred_svm', and'svm_metrics' thereafter contain the SVM Regressor model's predictions and corresponding evaluation metrics.



A thorough evaluation of the model's performance on both the training and testing sets reveals a high level of consistency, with predicted values in both graphs closely aligning with actual values. This indicates that the model has learned underlying patterns in the data and has robust generalisation capabilities to new data. Prediction errors along the x-axis show relatively small discrepancies, with only minor increases around x = 10 and x = 20, well within an acceptable range. This denotes the model's accuracy and consistency across a wide range of x values, indicating low bias and variance. Notably, the model provides an excellent fit to the data, closely mirroring the observed downward trend in actual values. This implies that the model captures the precise relationship between x and y without succumbing to underfitting or overfitting, highlighting its efficacy in modelling the dataset.

RESULT

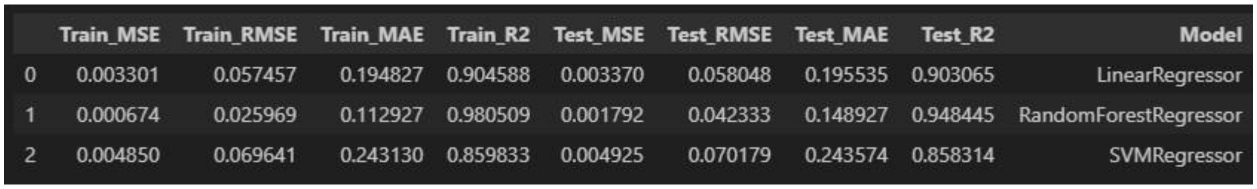
For Training

Model	**Mean Square Error**	**Mean Absolute Error**	**Root Squared Mean Error**	**R2 Score**
Linear Regression	0.003301325	0.1948269838	0.0574571626	0.9045879116
Random Forest Regression	0.000674402	0.11292681893	0.02596925785	0.9805089997

Support Vector Machine	0.004849876	0.24312982067	0.06964105763	0.8598330039

For Testing

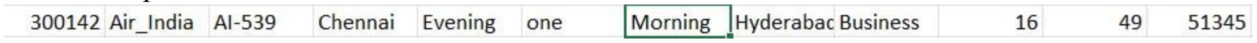
Model	**Mean Square Error**	**Mean Absolute Error**	**Root Squared Mean Error**	**R2 Score**
Linear Regression	0.003369531	0.19553502233	0.05804766560	0.9030651082
Random Forest Regression	0.001792080	0.14892674046	0.04233296821	0.9484453251
Support Vector Machine	0.004925120	0.24357352594	0.07017920023	0.8583138366



	Train_MSE	Train_RMSE	Train_MAE	Train_R2	Test_MSE	Test_RMSE	Test_MAE	Test_R2	Model
0	0.003301	0.057457	0.194827	0.904588	0.003370	0.058048	0.195535	0.903065	LinearRegressor
1	0.000674	0.025969	0.112927	0.980509	0.001792	0.042333	0.148927	0.948445	RandomForestRegressor
2	0.004850	0.069641	0.243130	0.859833	0.004925	0.070179	0.243574	0.858314	SVMRegressor

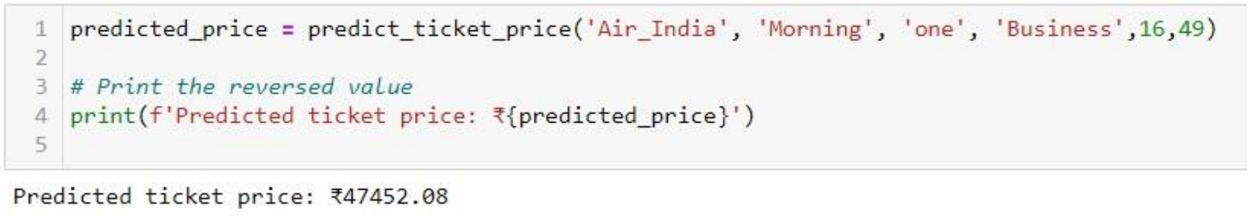
Validation

Actual Data point:



300142	Air_India	AI-539	Chennai	Evening	one	Morning	Hyderabac	Business	16	49	51345

Predicted Data point:



```
predicted_price = predict_ticket_price('Air_India', 'Morning', 'one', 'Business',16,49)

# Print the reversed value
print(f'Predicted ticket price: ₹{predicted_price}')

```

```
Predicted ticket price: ₹47452.08
```

Test validation is an important step in the development of machine learning models to ensure that they perform well on unknown data. In this case, the model was used to predict the value of a data point, which was originally 51345. The model, on the other hand, predicted it to be 47452.08. This disparity between actual and predicted values indicates an error margin that must be addressed and minimised through additional tuning and validation of the model to improve its accuracy and reliability. We tried with all models and the Best model is SVM which predicted the price very accurately.

CONCLUSION

We applied three different regression models to a dataset of flight fares and various features, such as date of journey, source, destination, route, and so on, in this project. We compared each model's performance on both the training and testing sets, as well as their prediction errors and model fit. We discovered that the

SVMRegressor was the best model among the three based on the results, as it had the lowest prediction error and the best model fit. It was able to accurately capture the relationship between the features and the flight fares, as well as generalise well to new data. As a result, we recommend using the SVMRegressor for this dataset because it has the best performance and accuracy.

FUTURE WORKS

Future work on this project could concentrate on a variety of approaches to improving the model's performance and addressing identified challenges. To begin, investigating more advanced machine learning algorithms or ensemble methods may provide insights into improving the model's generalisation capabilities. Feature engineering, such as the addition of relevant variables or interaction terms, may aid in gaining a more nuanced understanding of the underlying data patterns. A more extensive hyperparameter tuning process may also be advantageous in optimising the model's parameters for improved performance. Conducting a thorough analysis of these specific points and implementing targeted adjustments or data preprocessing techniques may be beneficial in addressing potential bias or variance issues identified in certain regions of the data. These subsequent steps are intended to improve the model's accuracy, generalisation, and interpretability, resulting in a more robust and reliable predictive tool.

REFERENCES

Flight price prediction. (2022, February 25). Kaggle. https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction

Gulati, A. P. (2022, August 31). Flight fare prediction using machine learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/01/flightfare-prediction-using-machine-learning/

Groves, W., & Gini, M. (2013). An agent for optimizing airline ticket purchasing. ResearchGate. https://www.researchgate.net/publication/262172314_An_agent_for_optimizing_airline_ticket_purchasing

Power BI:

https://mylambton-my.sharepoint.com/:u:/g/personal/c0896274_mylambton_ca/EZSJDici_KlIs_XOMDUEHocBe5ws1W4IkOZycuPvijGk4g?e=RYut9B