



INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON
Flight Ticket Cost Prediction

About me

- Vijay sada
- B-tech in Electronics and Communications Engineering.
- Fresher
- LinkedIn - <https://www.linkedin.com/in/vijay-sada/>
- GitHub - <https://github.com/vijaysada>

Introduction

The concept of flight price prediction involves analyzing historical data and various factors that influence the cost of airline tickets, such as the airline, departure date, booking class, and other relevant features. The goal is to predict the fare of a flight ticket for customers, enabling them to make informed decisions and potentially save money on their travel expenses.

The airline industry is highly competitive and dynamic, with ticket prices fluctuating based on demand, seasonality, fuel prices, and other factors. By leveraging predictive models, airlines and travel agencies can optimize pricing strategies, enhance customer satisfaction, and increase revenue.

In this context, flight price prediction serves multiple purposes:

- **Customer Decision-Making:** Helping customers find the best deals and plan their travel budget effectively.
- **Revenue Management:** Assisting airlines in setting competitive prices while maximizing revenue.
- **Market Analysis:** Providing insights into market trends and customer preferences.

Data Overview

The dataset was obtained from Kaggle.

The dataset includes the following features:

- **Airline:** The name of the airline company. It is a categorical feature having 8 different airlines.
- **Flight:** Information regarding the plane's flight code. It is a categorical feature.
- **Source City:** The city from which the flight takes off. It is a categorical feature having 6 unique cities.
- **Departure Time:** A derived categorical feature created by grouping time periods into bins. It stores information about the departure time and has 4 unique time labels.
- **Stops:** A categorical feature with 3 distinct values that stores the number of stops between the source and destination cities.
- **Arrival Time:** A derived categorical feature created by grouping time intervals into bins. It has 4 distinct time labels and keeps information about the arrival time.
- **Destination City:** The city where the flight will land. It is a categorical feature having 6 unique cities.
- **Class:** A categorical feature that contains information on seat class; it has two distinct values: Business and Economy.
- **Duration:** A continuous feature that displays the overall amount of time it takes to travel between cities in hours.
- **Days Left:** A derived characteristic calculated by subtracting the trip date from the booking date.
- **Price:** The target variable that stores information about the ticket price.

Our target column is 'Price'

| S No | Feature Name | Data Type |
|------|----------------|-----------|
| 1 | Date | Object |
| 2 | Airline | Object |
| 3 | Flight | Object |
| 4 | From | Object |
| 5 | Departure time | Object |
| 6 | Stops | Object |
| 7 | Destination | Object |
| 8 | Arrival time | Object |
| 9 | Class | Object |
| 10 | Duration | Float64 |
| 11 | Days left | Int64 |
| 12 | Price | Float64 |

Shape : 300261 Rows x 12 Columns



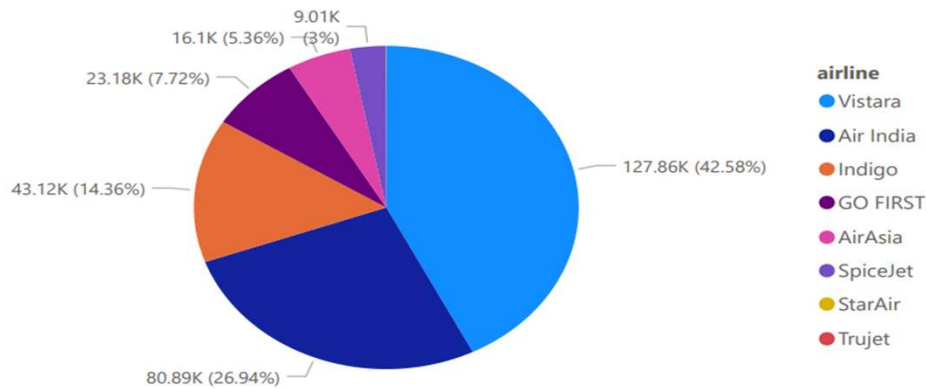
Pre-Processing

For the project, the following preprocessing steps were performed:

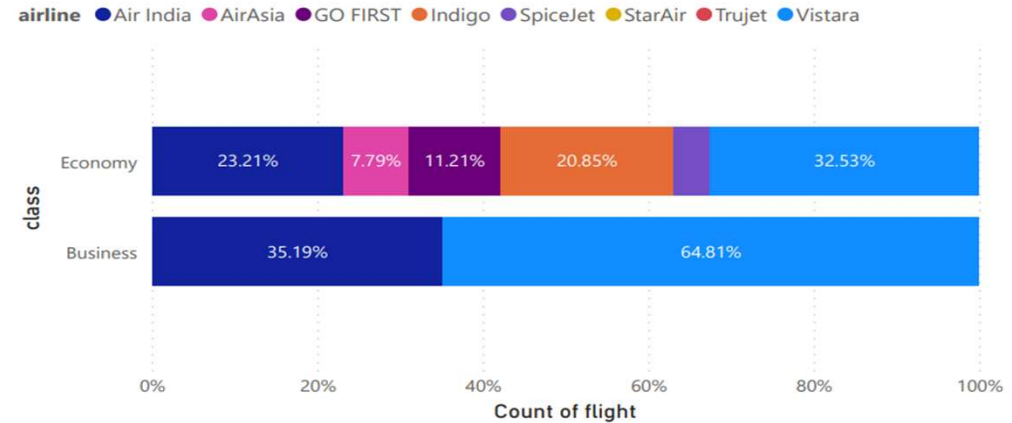
- **Data Cleaning:** Identified and handled inconsistencies like unwanted tags, wrong data types, datetime columns, duplicate values, and outliers were found in the 'duration' column the dataset.
- **Handling Duplicate Values:** Addressed the 3195 duplicate datapoints by removing to maintain the data consistency.
- **Robust Scaling:** Rescaled the numerical columns (Duration, Days left) using Robust Scaling to ensure features are on a similar scale and address outliers in the duration column.
- **Handling Categorical Variables:** Used Binary Encoding since the data in these categorical columns are nominal with a high cardinality ranging from 4 to 8.

Exploratory Data Analysis (EDA)

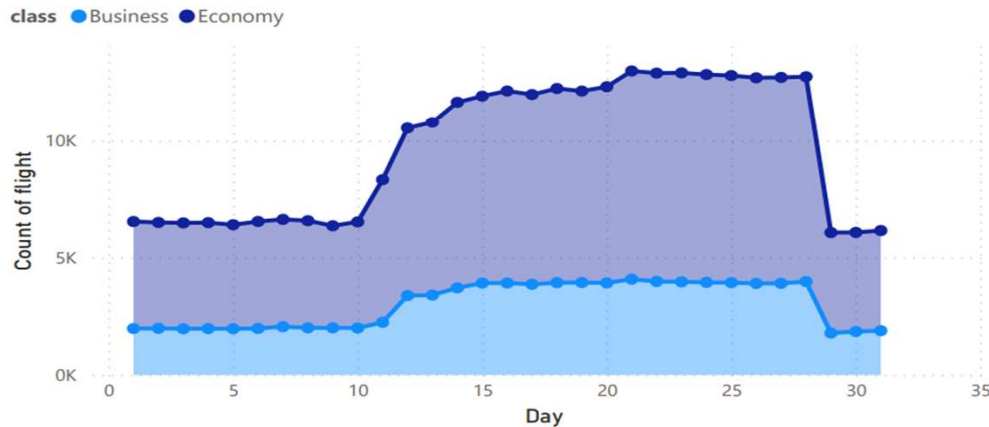
Distribution of airlines



Distribution of airlines by class



Count of flights available by Day and class

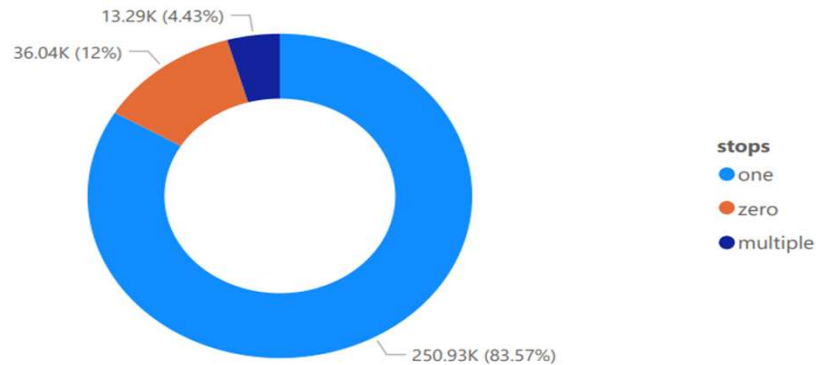


Observations:

- Vistara is the leading airline, making up 42.58% of the total flights, with a particularly strong presence in the business class at 64.81%. It is followed by Air India in both the economy and business class flights.
- We can observe that only two airlines (Vistara and Air India) are available with the business class flights operating between our destinations.
- Economy class flights are more frequent than business class flights across all airlines and days, indicating a higher demand for economy travel.
- Flights activity shows a generally stable pattern from 21st day with a significant peak around the 28th day, followed by a sharp decline.
- We can observe that there are a lot of flights available during the middle of the month, than that at the end/ start of the month.
- The airline SpiceJet has an very low presence in the airlines operating between our destinations.

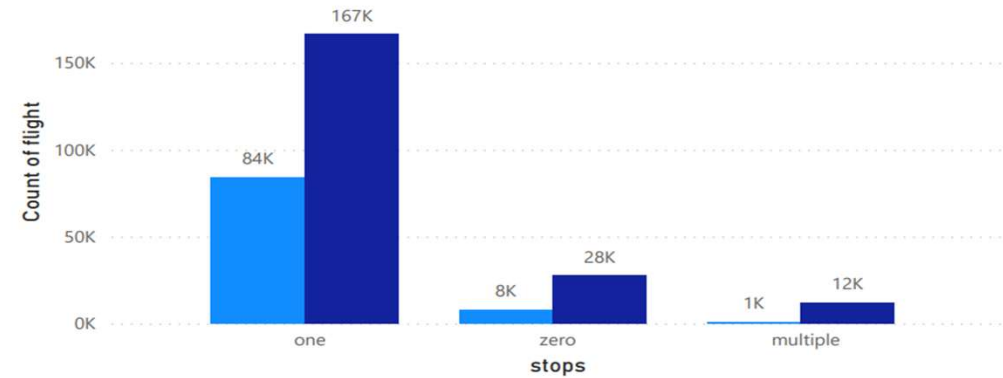
EDA (Cont.):

Distribution of stops



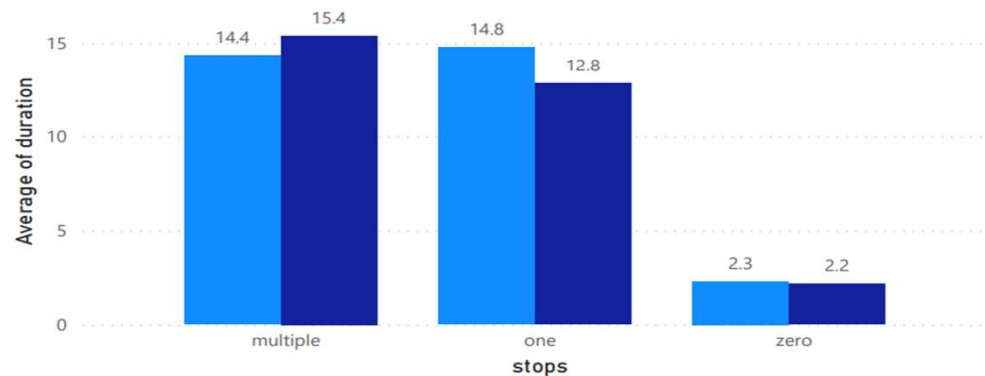
number of flights by stops and class

class ● Business ● Economy



Average of duration by stops and class

class ● Business ● Economy



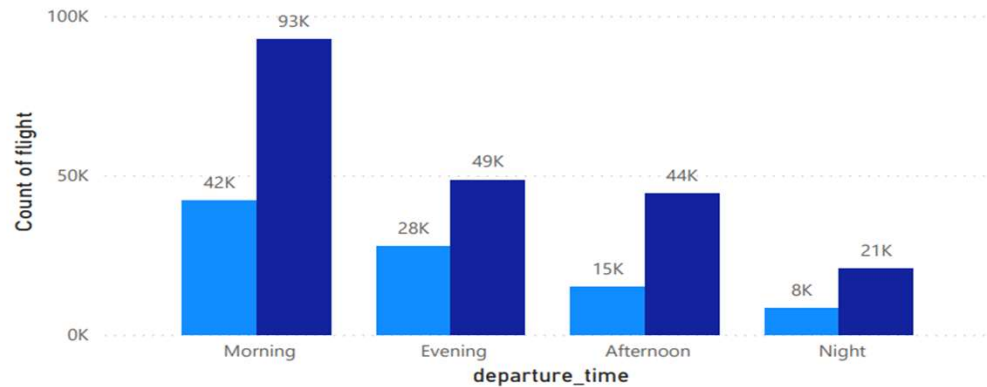
Observations:

- We can observe that majority of the flights with one stop are the most common, making up 83.57% of the total.
- Direct (zero stop) flights account for 12% of the total, while multiple-stop flights are rare, comprising only 4.43% of the total.
- Flights with multiple stops have the longest average duration for economy class, while flights with single stop have the longest average duration for the business class.
- Direct flights have the shortest duration, with similar durations for both business and economy classes.

EDA (Cont.):

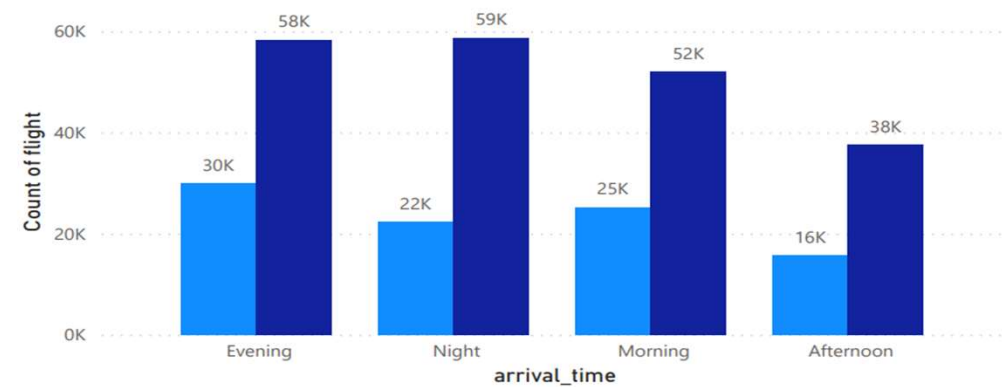
Count of flight by departure_time and class

class ● Business ● Economy

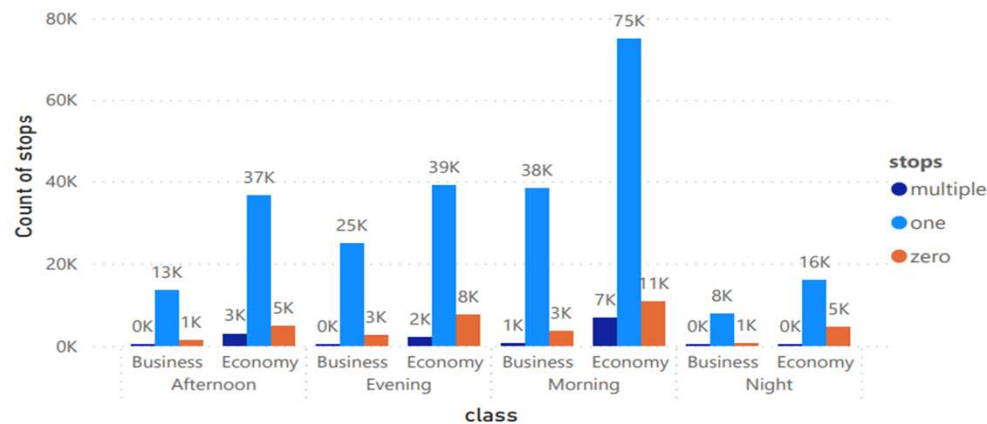


Count of flight by arrival_time and class

class ● Business ● Economy



Count of stops by class and departure_time



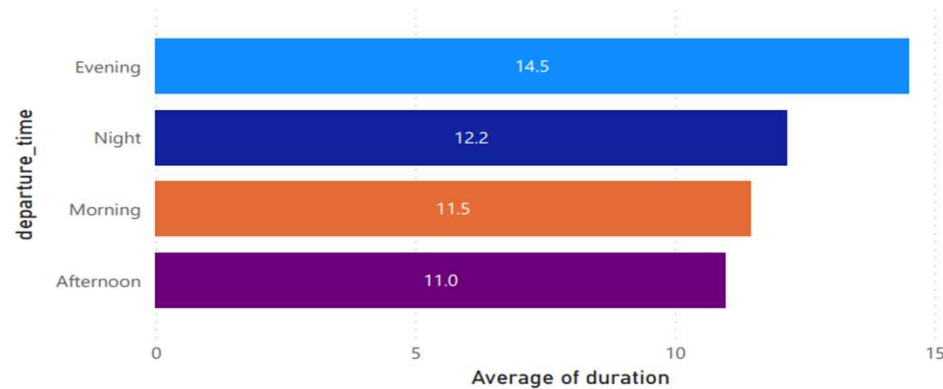
Observations:

- The highest no. of flights by departure time and class is for economy class in the morning, with 93K flights. This significantly outnumbers any other category.
- The lowest no. of flights by departure time and class is for business class in the night, with 8K flights.
- The evening arrival time sees the highest count of flights across both classes, with a total of 58K flights for economy and 30K flights for business, totaling 88K flights.
- Business class flights are most frequent in the evening in terms of arrivals. While the economy class flights are almost equal during evening and nights in terms of arrival.
- The economy class flights have a lot of stops which depart in the morning and the least during the night.

EDA (Cont.):

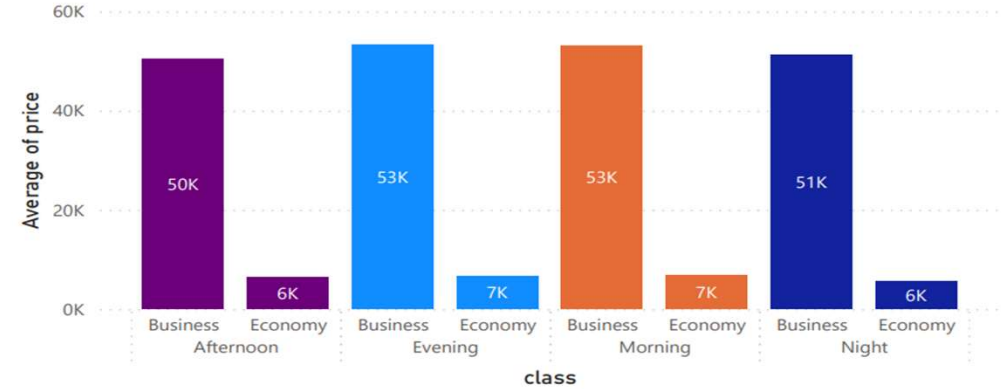
Average of duration by departure_time

departure_time ● Evening ● Night ● Morning ● Afternoon



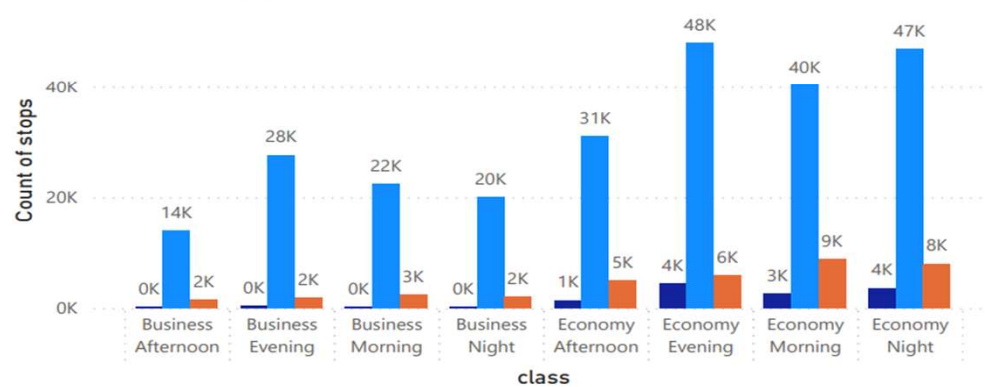
Average of price by departure time and class

departure_time ● Afternoon ● Evening ● Morning ● Night



Count of stops by arrival time and class

stops ● multiple ● one ● zero



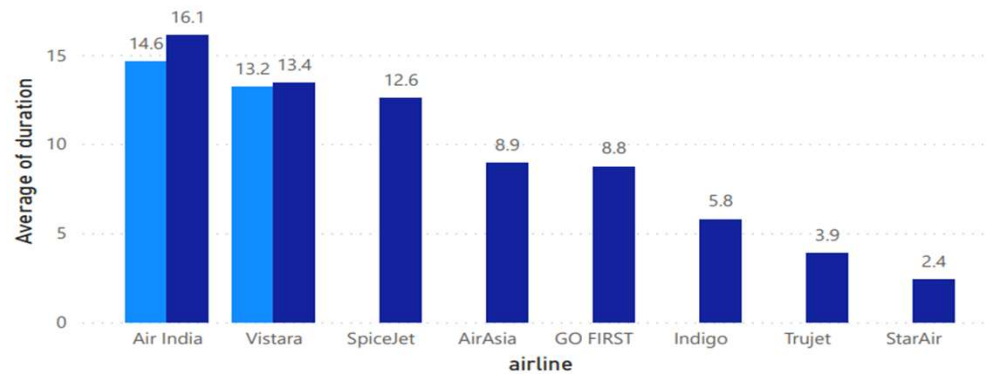
Observations:

- Flights departing in the evening have the highest average duration of 14.5 hours, significantly longer than flights at other times of the day.
- Flights departing in the morning and afternoon have the shortest average durations, at 11.5 hours and 11.0 hours respectively.
- The average price for business class flights is fairly consistent across departure times, ranging from 50K to 53K. In contrast, economy class flights are consistently cheaper, averaging between 6K and 7K across all times.
- Economy class flights in the evening and night have a high count of stops, with evening flights having 48K stops and night flights having 47K stops. In contrast, business class flights have fewer stops, with lowest no. of multiple-stop flights recorded.

EDA (Cont.):

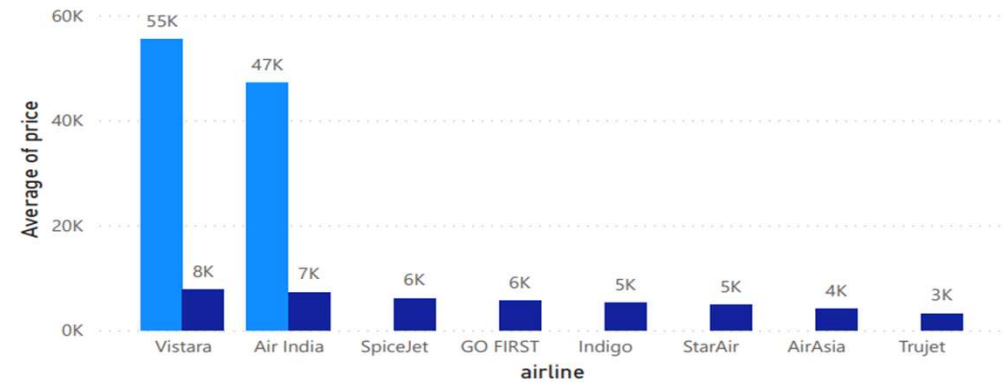
Average of duration by airline and class

class ● Business ● Economy



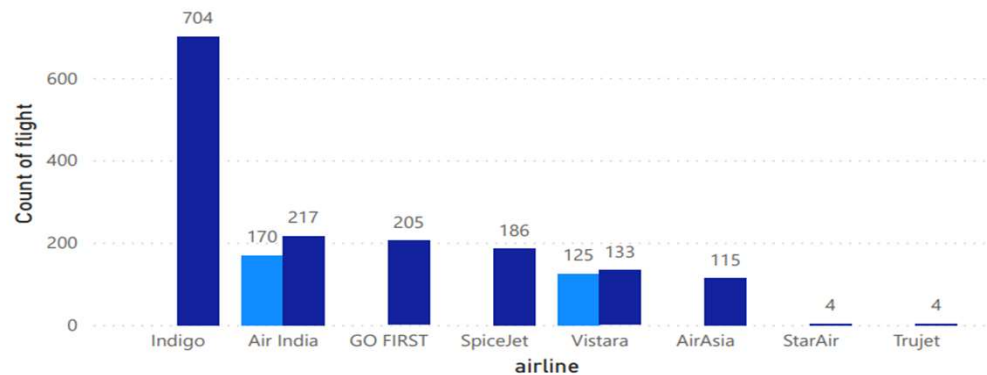
Average of price by airline and class

class ● Business ● Economy



Count of flight by airline and class

class ● Business ● Economy



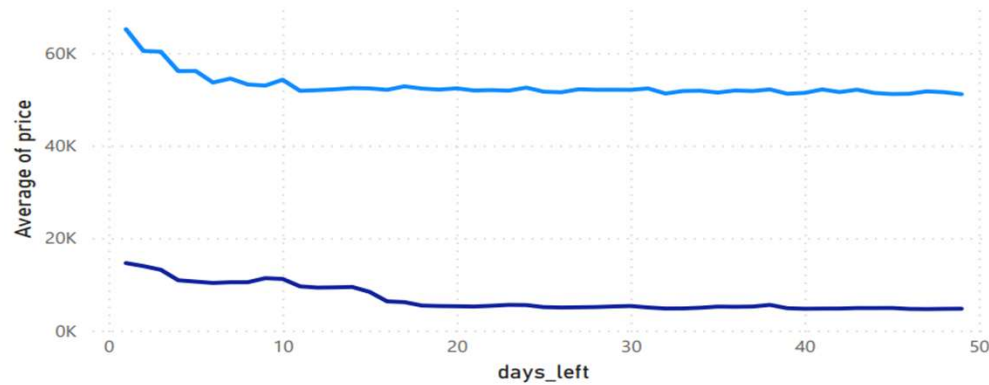
Observations:

- Vistara has the highest average duration for economy class flights, with an average of 16.1 hours. This is higher than any other airline for either class.
- The average price for business class flights on Vistara is the highest, at 55K. This is significantly higher compared to other airlines, with the next highest being Air India at 47K. While they're at a similar range in economy class flights.
- StarAir and Trujet have the shortest average durations for economy class flights, with StarAir at 2.4 hours and Trujet at 3.9 hours.
- Across all airlines, economy class flights have significantly lower average prices compared to business class flights. The average price for economy class flights remains below 8K for all airlines, whereas business class prices are much higher in Vistara and Air India.
- The StarAir and Trujet have the lowest count of flights with different travel routes. The Indigo airlines have the highest no. of flight routes among all the airlines. It's followed by Air India with the 2nd highest highest no. of travel routes available.

EDA (Cont.):

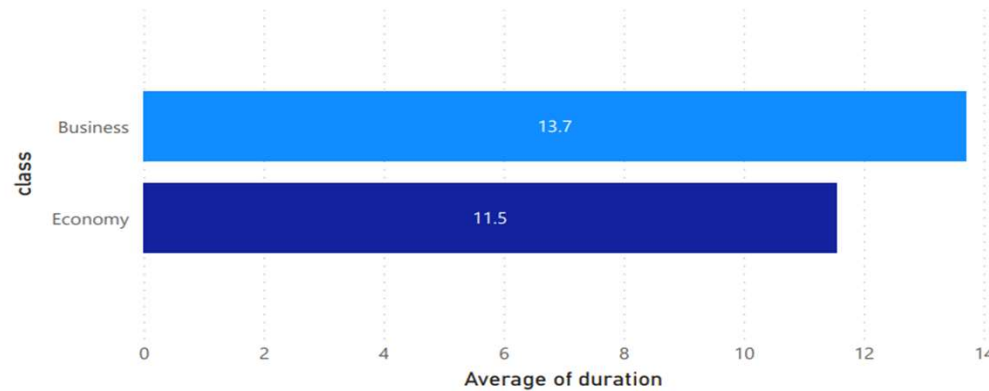
Average of price by days_left and class

class ● Business ● Economy

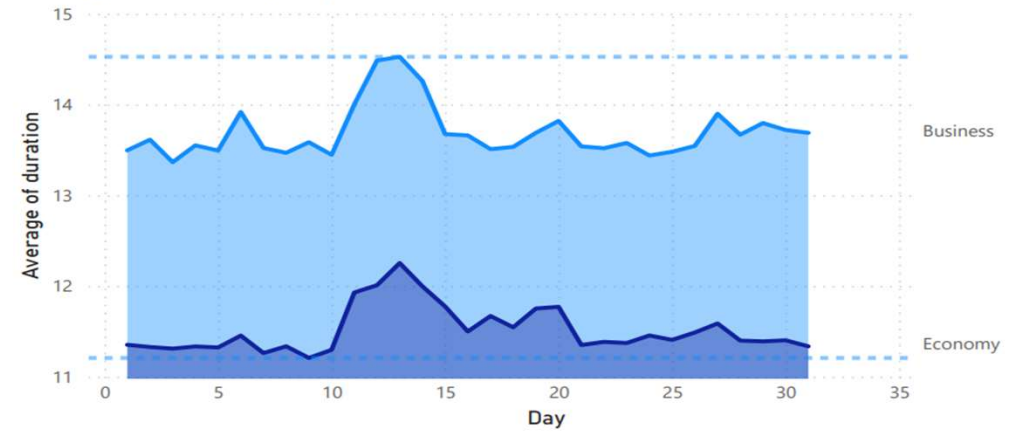


Average of duration by class and class

class ● Business ● Economy



Average of duration by Day and class



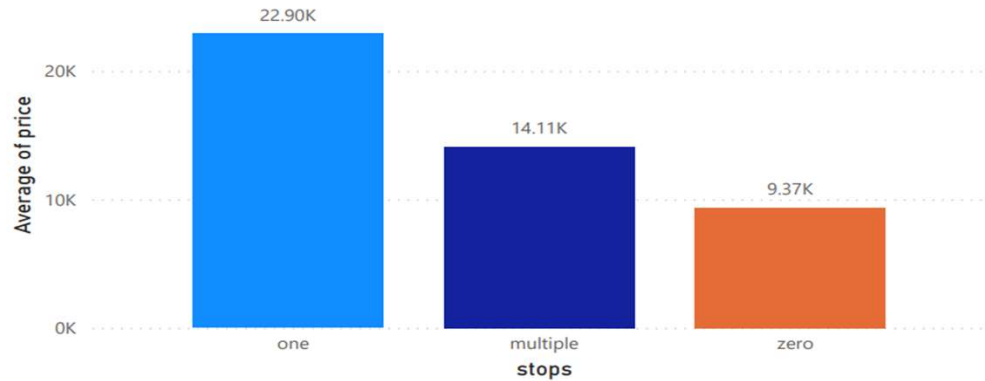
Observations:

- The average price of flights decreases as the days left until departure increase. This trend is evident for both business and economy classes, with a steeper decline for business class prices.
- Business class flights consistently have higher average prices compared to economy class across all days left until departure.
- On average, business class flights have a longer duration (13.7 hours) compared to economy class flights (11.5 hours). This might be due to the no. of stops for the respective flights.
- The average duration of flights varies significantly day-to-day for both business and economy classes. We can observe that it gets longer in the middle of the month around 10th to 16th.
- While both classes show variations in flight duration over days, economy class durations are more consistent and show fewer fluctuations compared to business class durations.

EDA (Cont.):

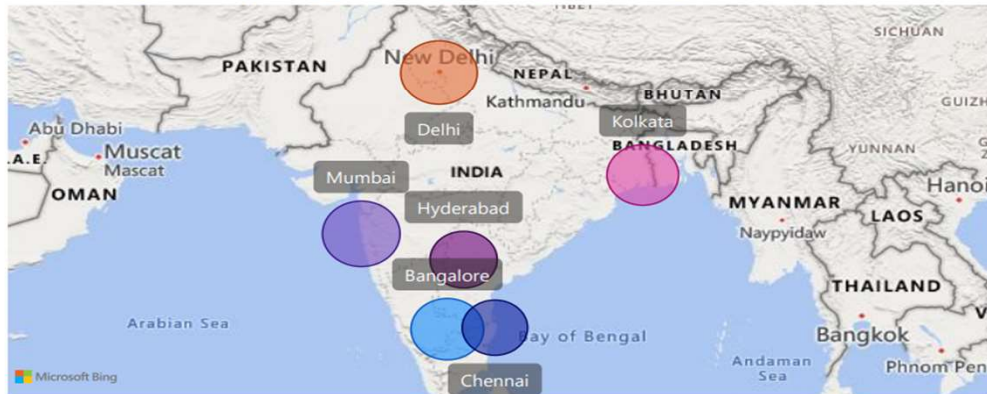
Average of price by stops

stops ● one ● multiple ● zero



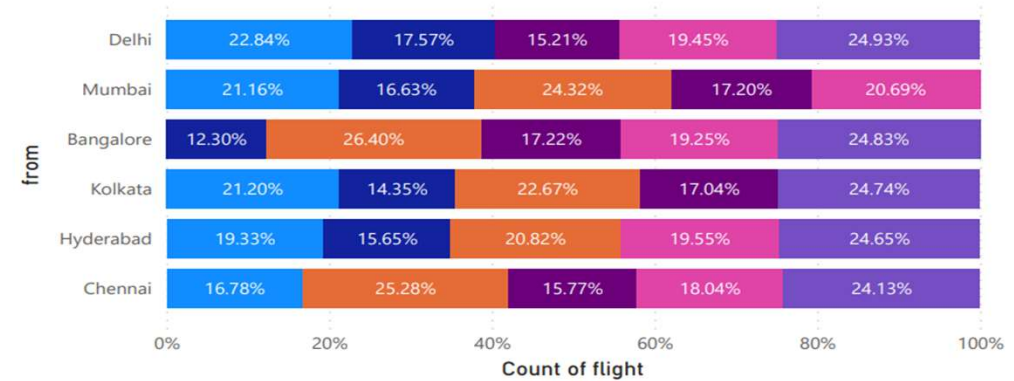
Count of flight by destination

destination ● Mumbai ● Delhi ● Bangalore ● Kolkata ● Hyderabad ● Chennai



Count of flight by from and destination

destination ● Bangalore ● Chennai ● Delhi ● Hyderabad ● Kolkata ● Mumbai



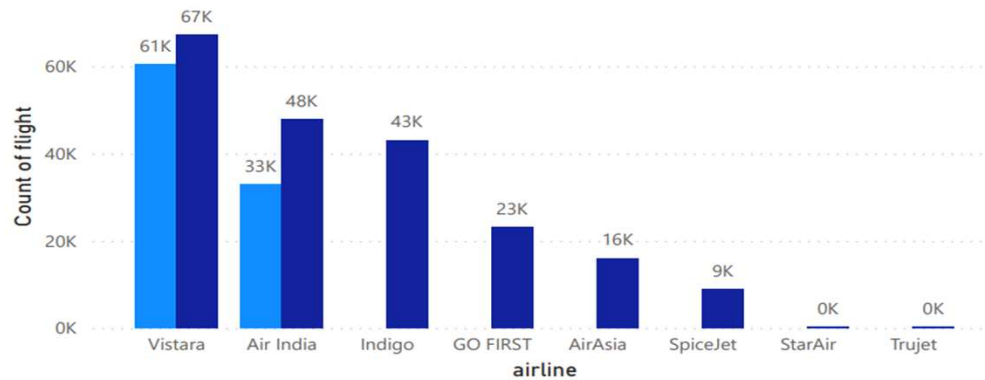
Observations:

- Flights with one stop have the highest average price, at 22.9K. This is significantly higher compared to flights with multiple stops (14.11K) and non-stop flights (9.37K).
- A notable percentage of flights from Bangalore are destined for Delhi, with 26.40% of flights heading there. This is the highest percentage for any destination from Bangalore.
- A notable percentage of flights from Bangalore are destined for Chennai, with 12.30% of flights heading there. This is the lowest percentage for any destination from Bangalore.
- Delhi receives a high proportion of flights from multiple origins, with notable percentages from Bangalore (26.40%), Chennai (25.28%), and Mumbai (24.32%).
- The geographical distribution plot shows that Mumbai and Delhi are major hubs with significant flight activity. They have a substantial number of flights to and from various destinations, indicating their importance in the flight network.
- The flights to Mumbai from different locations have an almost same distribution maintaining consistency.

EDA (Cont.):

Count of flight by airline and class

class ● Business ● Economy

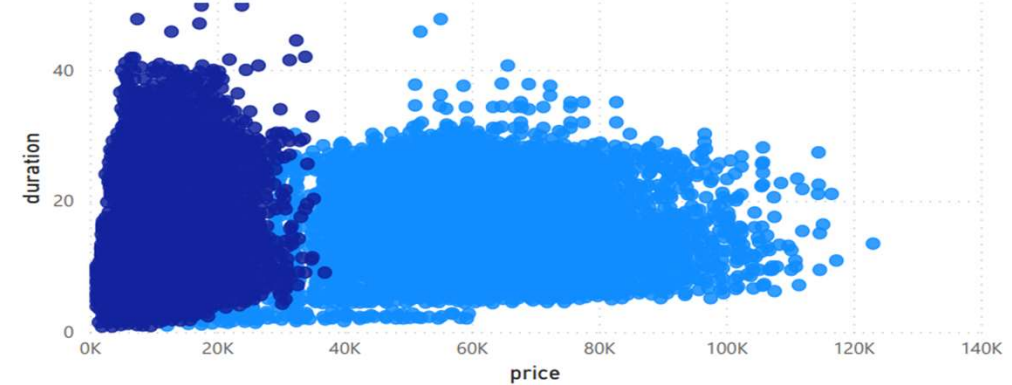


descriptive statistics table on price based on the departure location to destination

| from | No. of Flights | Average price | Max | Median | Min | S.D of price |
|--------------|----------------|-----------------|---------------|-------------|-------------|-----------------|
| Mumbai | 60903 | 21481.71 | 114523 | 7413 | 1890 | 23393.42 |
| Kolkata | 46347 | 21746.24 | 123071 | 7958 | 2436 | 23439.72 |
| Hyderabad | 40860 | 20133.24 | 115211 | 6799 | 1543 | 21714.79 |
| Delhi | 61345 | 18950.98 | 117307 | 6840 | 1998 | 20920.00 |
| Chennai | 38700 | 21995.34 | 114704 | 7846 | 1105 | 23526.92 |
| Bangalore | 52106 | 21455.88 | 111883 | 7488 | 1603 | 23165.99 |
| Total | 300261 | 20883.72 | 123071 | 7425 | 1105 | 22695.87 |

price vs duration by class

class ● Business ● Economy



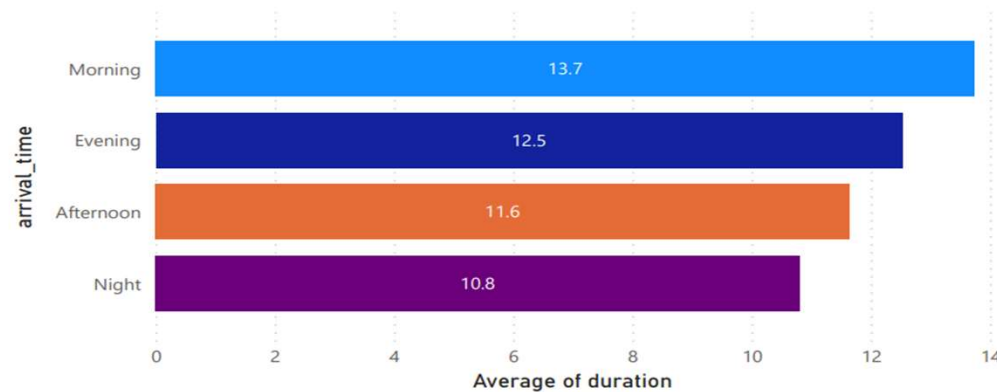
Observations:

- Vistara operates the highest number of flights overall, with 67K economy flights and 61K business flights, making it the leading airline in terms of flight count.
- Air India and Indigo also have a substantial number of flights, with Air India operating 48K economy and 33K business flights, and Indigo operating 43K economy flights. This positions them as significant players in the airline market.
- There is a visible clustering of flight durations around shorter times, with economy flights generally priced lower than business flights. However, some business flights have much higher prices, indicating a broader price range for this class.
- The table reveals significant variability in flight prices across different cities. Chennai has the highest average price at 21,995.34, followed by Kolkata and Mumbai. Delhi has the lowest average price at 18,950.98 among the major cities listed.
- The standard deviation of prices is relatively high for all cities, indicating considerable price variability. Kolkata has the highest standard deviation (23,439.72), followed closely by Chennai and Mumbai, suggesting a wide range of pricing options within these cities.

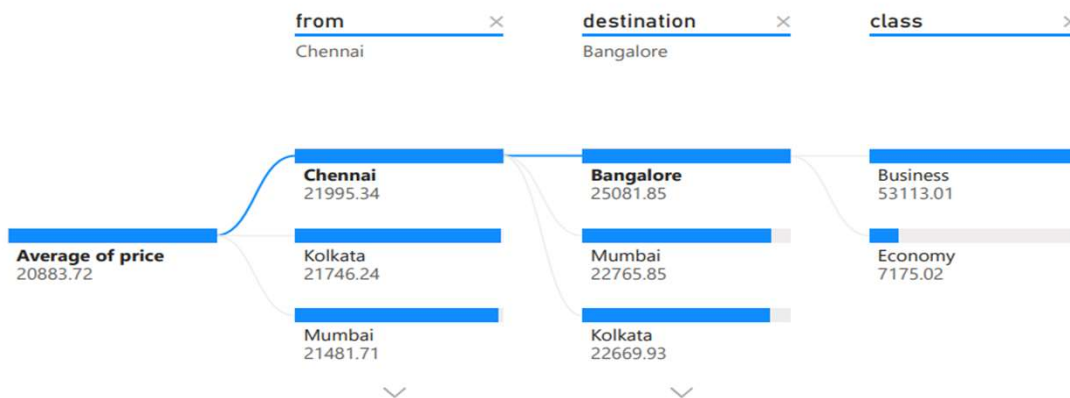
EDA (Cont.):

Average of duration by arrival_time

arrival_time ● Morning ● Evening ● Afternoon ● Night



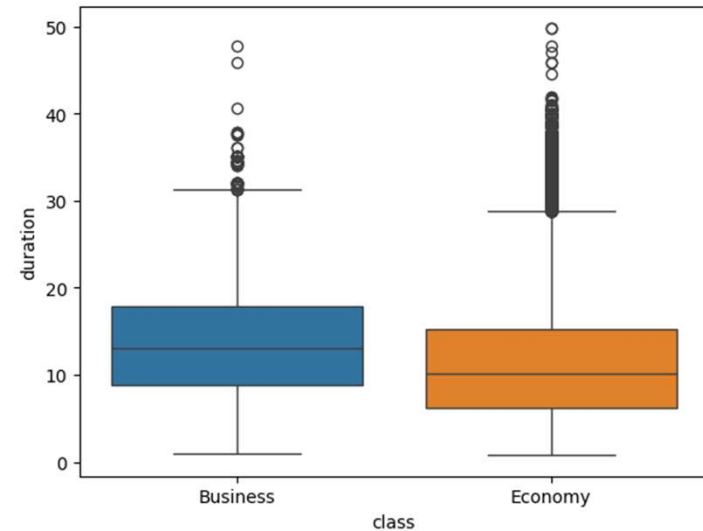
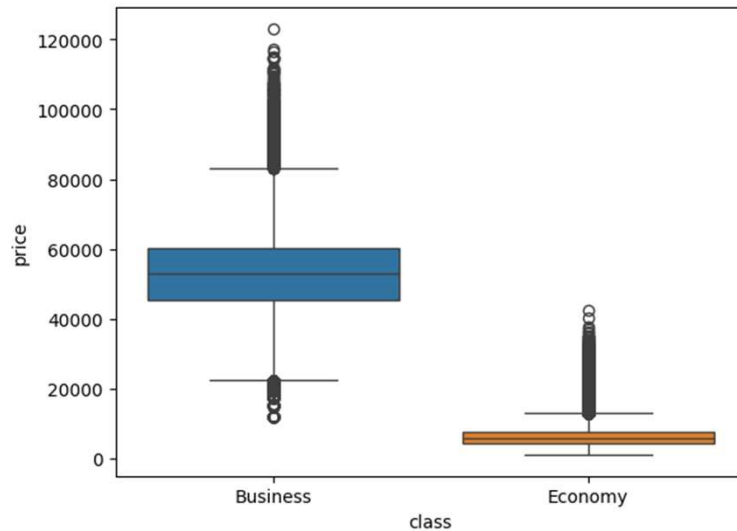
Average of price by class



Observations:

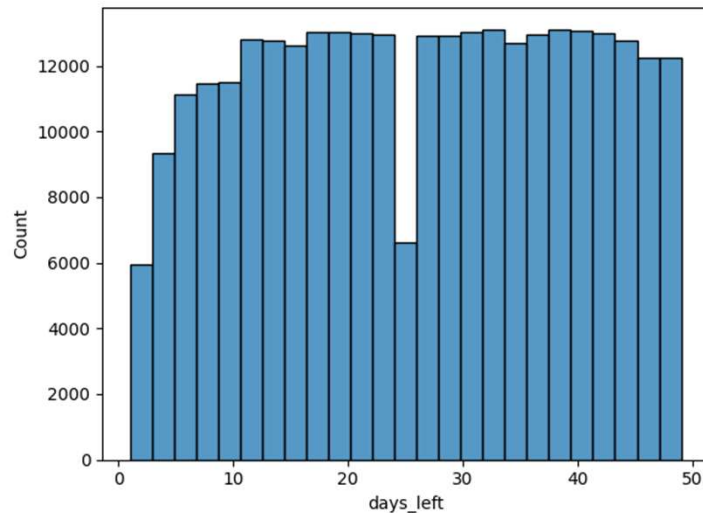
- From the bar graph we can observe that on a most of the flights arrive either in the morning or the evening, meaning it's the most crowded times in the airport.
- The business class flights cost around 8 times more than economy class flights on an average. It can be a major factor to be considered based on the urgency of the person.
- The tree diagram shows the avg. costs of flights along with their duration when hovered from one place to another with in detail of travelling class division too.

EDA (Cont.):



- The outliers in the price column are present primarily because all identified high-priced flights are in the Business class, have either one or multiple stops, are operated exclusively by Vistara, majority are departing in the morning or evening and some are booked close to the departure date, all of which contribute to significantly higher prices. So we can't exclude the outliers as they are true outliers.
- From the above we can observe that the flight durations contains outliers and these outliers in the flight duration column are valid and likely caused by real-world variations in flight operations, such as the number of stops, specific flight routes, layover times, and the types of aircraft used. So we can't exclude the outliers as they are true outliers.

EDA (Cont.):



| destination | Bangalore | Chennai | Delhi | Hyderabad | Kolkata | Mumbai |
|-------------|-----------|---------|-------|-----------|---------|--------|
| from | | | | | | |
| Bangalore | 0 | 6410 | 13756 | 8971 | 10029 | 12940 |
| Chennai | 6493 | 0 | 9783 | 6103 | 6983 | 9338 |
| Delhi | 14012 | 10780 | 0 | 9328 | 11934 | 15291 |
| Hyderabad | 7898 | 6395 | 8507 | 0 | 7987 | 10073 |
| Kolkata | 9824 | 6653 | 10506 | 7897 | 0 | 11467 |
| Mumbai | 12885 | 10130 | 14809 | 10477 | 12602 | 0 |

- The histogram of the days_left variable from the flight dataset reveals a bimodal distribution with peaks around 20 and 30 days before departure, and a noticeable dip around the 25-day mark, indicating fewer flights at that point.
- The highest no. of flights are flying from Delhi to Mumbai, this might be because they're quite the famous cities also known as capital and financial capital of India.
- The lowest count of flights are from Chennai to Hyderabad, this might be because of them being close to each other, leading to the availability of other kinds of transportations.

EDA (Cont.):

| | duration | days_left | price |
|-----------|-----------|-----------|-----------|
| duration | 1.000000 | -0.039105 | 0.204473 |
| days_left | -0.039105 | 1.000000 | -0.091917 |
| price | 0.204473 | -0.091917 | 1.000000 |

- Above table shows the correlation of the different numerical columns in our dataset. We can observe that price and duration have a weak positive relation, also that price and days_left have a weak negative correlation.
- From the right table we can observe the flight tickets cost with respect to each of the base city to the destination city. It contains the values like "min", "max", "mean", "median" and "Standard deviation" of the price based on the different locations the flight travels between.

| | | min | max | mean | median | std |
|-----------|-----------|--------|----------|--------------|---------|--------------|
| Bangalore | Chennai | 1603.0 | 90720.0 | 23321.850078 | 9241.0 | 22573.185689 |
| | Delhi | 2723.0 | 111883.0 | 17723.313972 | 7164.0 | 19746.484106 |
| | Hyderabad | 1694.0 | 83239.0 | 21152.051053 | 7813.0 | 21861.177859 |
| | Kolkata | 3026.0 | 105168.0 | 23498.234221 | 8112.0 | 24630.560155 |
| | Mumbai | 2150.0 | 103819.0 | 23127.231376 | 7113.0 | 25887.165127 |
| Chennai | Bangalore | 1443.0 | 107597.0 | 25081.850454 | 10469.0 | 23405.422526 |
| | Delhi | 2051.0 | 103683.0 | 18981.863948 | 7352.0 | 21946.879653 |
| | Hyderabad | 1105.0 | 92752.0 | 21591.345404 | 7373.0 | 22866.927328 |
| | Kolkata | 2359.0 | 104624.0 | 22669.932407 | 8394.0 | 23667.149966 |
| | Mumbai | 1830.0 | 114704.0 | 22765.849647 | 8233.0 | 25118.401202 |
| Delhi | Bangalore | 3090.0 | 85353.0 | 17880.216315 | 6642.0 | 19904.508234 |
| | Chennai | 1998.0 | 104466.0 | 19369.881354 | 7425.0 | 22127.553940 |
| | Hyderabad | 2022.0 | 114507.0 | 17347.288379 | 6109.0 | 18768.239479 |
| | Kolkata | 2480.0 | 117307.0 | 20566.409418 | 7084.0 | 23655.844456 |
| | Mumbai | 2281.0 | 95657.0 | 19354.405336 | 7262.0 | 19776.397176 |
| Hyderabad | Bangalore | 1755.0 | 97767.0 | 21245.945429 | 6855.0 | 22174.741408 |
| | Chennai | 1543.0 | 95208.0 | 21848.065989 | 7702.0 | 22527.946093 |
| | Delhi | 2200.0 | 86203.0 | 17242.639473 | 6138.0 | 18547.945651 |
| | Kolkata | 2056.0 | 97381.0 | 20823.893201 | 7767.0 | 22237.613504 |
| | Mumbai | 2250.0 | 115211.0 | 20065.715179 | 6633.0 | 22633.659515 |
| Kolkata | Bangalore | 3465.0 | 105638.0 | 22744.808428 | 8111.0 | 24130.762785 |
| | Chennai | 2966.0 | 95183.0 | 23660.361040 | 8589.0 | 23371.419897 |
| | Delhi | 2994.0 | 123071.0 | 19422.354559 | 6723.0 | 22693.238883 |
| | Hyderabad | 2436.0 | 114705.0 | 21500.011397 | 8467.0 | 22690.671624 |
| | Mumbai | 3379.0 | 110936.0 | 22078.883579 | 7958.0 | 23887.604966 |
| Mumbai | Bangalore | 2074.0 | 114523.0 | 23147.873807 | 7192.0 | 25900.493645 |
| | Chennai | 1890.0 | 111964.0 | 22781.899112 | 8148.0 | 24690.486578 |
| | Delhi | 2336.0 | 111437.0 | 18725.320008 | 6300.0 | 19493.523862 |
| | Hyderabad | 2105.0 | 99677.0 | 20992.128567 | 7584.0 | 22807.139498 |
| | Kolkata | 2835.0 | 100909.0 | 22379.146723 | 7518.0 | 23998.184785 |

EDA (Cont.):

Recommendations:

1. Book Economy Class for Cost Savings:

- Economy class consistently has lower prices compared to business class across all airlines and times of day. If budget is a priority, opt for economy tickets.

2. Choosing Airlines:

- Vistara and Air India offer the highest number of flights and a good balance of price and duration. For a balance between frequency and cost, these airlines are recommended.
- For the lowest prices, consider airlines like GO FIRST, Indigo, and AirAsia, but be aware that these may also have longer average durations and fewer flight options.

3. Direct vs. Stopover Flights:

- Direct flights (zero stops) are generally cheaper than flights with one or multiple stops. If cost is a major factor, prioritize booking direct flights.
- Multiple stops significantly increase the duration and sometimes the cost of the flight. If possible, avoid flights with multiple stops.

4. Leverage Price and Duration Insights:

- For business travelers prioritizing time, choosing business class might be more beneficial despite the higher cost, especially during morning or evening times.
- For personal travel, where budget might be more critical, evening flights in economy class can offer a good balance between cost and convenience.

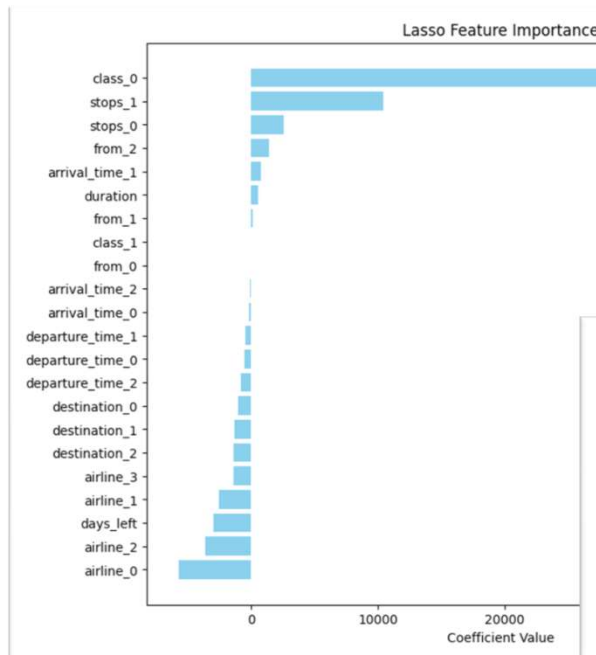
5. Book Early for Peak Times:

- During peak travel seasons or times (e.g., holidays, weekends), flights, especially in the morning and evening, fill up quickly. Booking early can help you secure better prices and preferred times.
- Booking at-least 10 days early can help you cut down the costs for any pre-planned occasions. Booking the flights at the middle of the month can be budget friendly as more flights are available at that time.

Feature Importance – Feature Selection

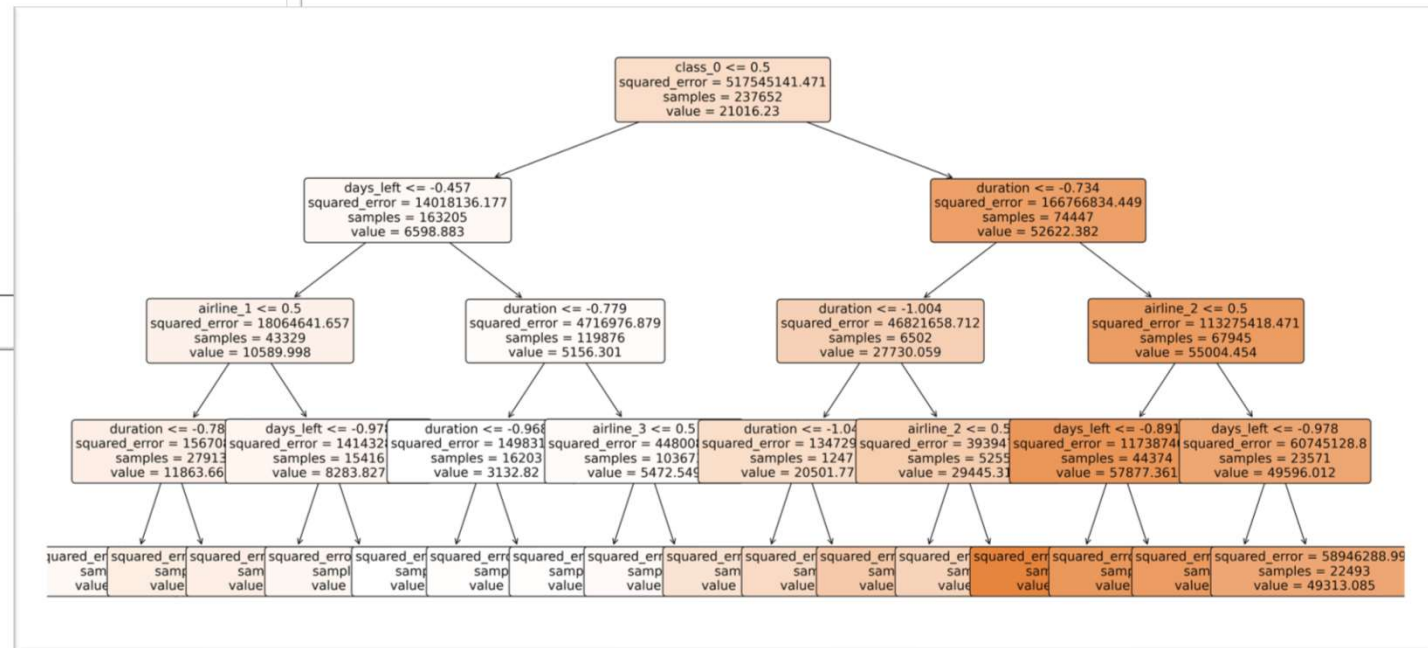
| | Removed Feature Name | Reason | Test Performed |
|---|----------------------|--|---|
| 1 | Date | Dropped this column since we've created a replacement for this column named 'days left' | NA |
| 2 | Flight | Dropped this column since it has high cardinality making it similar to IDs type of data. | NA |
| 3 | From | Dropped because of low feature importance in all observations. | Lasso, DecisionTreeRegressor, RandomForestRegressor |
| 4 | Destination | Dropped because of low feature importance in all observations. | Lasso, DecisionTreeRegressor, RandomForestRegressor |
| 5 | Departure time | Dropped because of low feature importance in all observations. | Lasso, DecisionTreeRegressor, RandomForestRegressor |
| 6 | Arrival time | Dropped because of low feature importance in all observations. | Lasso, DecisionTreeRegressor, RandomForestRegressor |

Feature Importance – Feature Selection



← Lasso

Decision Tree
Regressor →



Model Building

For the regression task of flight ticket cost prediction, the below machine learning models were used:

- 1. KNeighbors Regressor:** K-nearest neighbors (KNN) regression predicts the target variable by averaging the values of its k-nearest neighbors in the feature space. It assumes that similar data points have similar target values, making it suitable for locally smooth relationships between features and the target.
- 2. Decision Tree Regressor:** Decision tree regression builds a model that predicts the target variable by partitioning the data into subsets based on the values of input features. It recursively splits the data based on feature thresholds, aiming to minimize the variance of the target variable within each subset.
- 3. Linear Regression:** Linear regression models the relationship between the dependent variable and one or more independent variables by fitting a linear equation. It assumes a linear relationship between the variables and is widely used for predicting continuous outcomes.
- 4. RANSAC Regressor:** RANSAC (RANdom SAmple Consensus) regression fits a regression model to a subset of data points (inliers) while ignoring outliers. It iteratively refits the model to improve accuracy by minimizing the impact of outliers on the model coefficients.

Model Building (Cont.)

5. **Theil-Sen Regressor:** Theil-Sen regression estimates the slope of the relationship between variables using the median of slopes between all pairs of sample points. It is robust to outliers and works well in the presence of noise and heteroscedasticity (unequal variance across data).
6. **Huber Regressor:** Huber regression combines the best properties of least squares and least absolute deviation methods. It minimizes the sum of squared errors for samples close to the regression line (like least squares) and absolute error for samples far from it (like least absolute deviation).
7. **Lasso Regression:** Lasso (Least Absolute Shrinkage and Selection Operator) regression adds a penalty to the sum of absolute values of the regression coefficients, promoting sparsity and feature selection by shrinking some coefficients to zero.
8. **Ridge Regression:** Ridge regression adds a penalty to the sum of squared coefficients (L2 regularization), reducing the effect of multicollinearity and shrinking the coefficients towards zero, but rarely to zero.
9. **ElasticNet Regression:** ElasticNet regression combines penalties from both Lasso and Ridge, using a convex combination of L1 and L2 regularization terms. It balances between feature selection (like Lasso) and handling multicollinearity (like Ridge).

Model Building (Cont.)

- 10. Random Forest Regressor:** Random forest builds multiple decision trees during training and outputs the average prediction of the individual trees. It reduces overfitting compared to a single decision tree and provides high accuracy.
- 11. Gradient Boosting Regressor:** Gradient boosting builds an ensemble of trees sequentially, where each tree corrects the errors of the previous one. It combines the predictions of multiple weak learners (decision trees) to produce a strong prediction model.
- 12. XGBoost Regressor:** XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library designed for efficient computation. It improves upon traditional gradient boosting with system optimizations and algorithmic enhancements.
- 13. AdaBoost Regressor:** AdaBoost (Adaptive Boosting) combines multiple weak learners (typically decision trees) to create a strong predictor. It assigns higher weights to incorrectly predicted instances, focusing subsequent learners on harder cases.

Model Training

The model training process involved:

- Splitting the dataset into training and testing sets (75:25 Ratio) to train the models on a subset of the data and evaluate their performance on unseen data.
- Training each model using the training set, where the model learns patterns and relationships between features and the target variable (charges).

| S No | Type of Problem | Approach | Algorithm Name |
|------|-----------------|----------------------------------|---------------------------|
| 1 | Regression | Distance-Based | KNeighborsRegressor |
| 2 | Regression | Decision Tree | DecisionTreeRegressor |
| 3 | Regression | Linear Model | LinearRegression |
| 4 | Regression | Robust Linear Model | RANSACRegressor |
| 5 | Regression | Robust Linear Model | TheilSenRegressor |
| 6 | Regression | Robust Linear Model | HuberRegressor |
| 7 | Regression | Linear Model with Regularization | Lasso |
| 8 | Regression | Linear Model with Regularization | Ridge |
| 9 | Regression | Linear Model with Regularization | ElasticNet |
| 10 | Regression | Ensemble - Bagging | RandomForestRegressor |
| 11 | Regression | Ensemble - Boosting | GradientBoostingRegressor |
| 12 | Regression | Ensemble - Boosting | XGBRegressor |
| 13 | Regression | Ensemble - Boosting | AdaBoostRegressor |

Model Evaluation Metrics

Evaluation metrics employed to assess model effectiveness are:

- **Mean Absolute Error (MAE):** MAE measures the average absolute difference between the predicted and actual values, providing a straightforward indication of the model's performance in terms of prediction accuracy without considering the direction of errors. Lower MAE values indicate better predictive accuracy, with each absolute difference equally weighted in the calculation.
- **Mean Squared Error (MSE):** MSE calculates the average of the squares of the differences between predicted and actual values. By squaring the differences, MSE penalizes larger errors more than smaller ones, making it sensitive to outliers. Lower MSE values indicate better model performance, reflecting smaller prediction errors overall.
- **Root Mean Squared Error (RMSE):** RMSE is the square root of the MSE. RMSE provides a measure of prediction accuracy that maintains the same units as the target variable, making it more interpretable. Like MSE, lower RMSE values indicate better model performance, with a higher penalty for larger errors.
- **R-squared (R²):** R² represents the proportion of variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, where a value closer to 1 indicates that the model explains a large portion of the variance in the target variable, signifying better model fit.
- **Adjusted R-squared (Adj R²):** Adj R² adjusts the R-squared value for the number of predictors in the model, providing a more accurate measure of model performance, especially when comparing models with different numbers of independent variables. Unlike R², Adj R² can decrease if the added predictors do not improve the model, ensuring a more reliable comparison across models.

Model Evaluation on base models:

| Model | Train MAE | Train MSE | Train RMSE | Train R2 | Train Adj R2 | Test MAE | Test MSE | Test RMSE | Test R2 | Test Adj R2 |
|------------------|-----------|-----------|------------|----------|--------------|----------|----------|-----------|---------|-------------|
| KNeighbours | 2791.82 | 2.28E+07 | 4777.5 | 0.96 | 0.96 | 3427.56 | 3.49E+07 | 5904.78 | 0.93 | 0.93 |
| DecisionTree | 1033.68 | 7.52E+06 | 2743.07 | 0.99 | 0.99 | 3874.28 | 4.92E+07 | 7013.73 | 0.91 | 0.91 |
| LinearRegression | 4592.36 | 4.84E+07 | 6957.66 | 0.91 | 0.91 | 4594.61 | 4.94E+07 | 7030.13 | 0.91 | 0.91 |
| RANSAC | 4544.57 | 6.18E+07 | 7861.98 | 0.88 | 0.88 | 4561.98 | 6.27E+07 | 7916.18 | 0.88 | 0.88 |
| TheilSen | 4477.81 | 4.91E+07 | 7005.85 | 0.9 | 0.9 | 4491.58 | 5.02E+07 | 7084.13 | 0.9 | 0.9 |
| HuberRegressor | 4270.72 | 5.15E+07 | 7173.77 | 0.9 | 0.9 | 4288.27 | 5.27E+07 | 7259.65 | 0.9 | 0.9 |
| Lasso | 4593.04 | 4.84E+07 | 6957.96 | 0.91 | 0.91 | 4596.02 | 4.94E+07 | 7030.97 | 0.91 | 0.91 |
| Ridge | 4593.41 | 4.84E+07 | 6957.65 | 0.91 | 0.91 | 4595.61 | 4.94E+07 | 7030.34 | 0.91 | 0.91 |
| Elastic Net | 10524.03 | 1.73E+08 | 13140.6 | 0.66 | 0.66 | 10685.08 | 1.78E+08 | 13353 | 0.66 | 0.66 |
| RandomForest | 1679.38 | 9.89E+06 | 3144.75 | 0.98 | 0.98 | 3341.12 | 3.49E+07 | 5906.82 | 0.93 | 0.93 |
| GradientBoosting | 3175.06 | 2.91E+07 | 5393.16 | 0.94 | 0.94 | 3196 | 2.95E+07 | 5434.52 | 0.94 | 0.94 |
| XGBoost | 2779.56 | 2.25E+07 | 4742.14 | 0.96 | 0.96 | 3059.9 | 2.80E+07 | 5294.74 | 0.95 | 0.95 |
| AdaBoost | 3726.34 | 3.48E+07 | 5901.17 | 0.93 | 0.93 | 3737.32 | 3.54E+07 | 5946.64 | 0.93 | 0.93 |

Hyper-Parameter Tuning

Hyperparameter Tuning

Hyperparameter tuning is the process of optimizing the hyperparameters of a machine learning model to improve its performance. Unlike model parameters, which are learned during the training process, hyperparameters are set before training and control the behavior and efficiency of the model.

RandomizedSearchCV

RandomizedSearchCV is a hyperparameter optimization technique used to improve the performance of a machine learning model by searching for the best combination of hyperparameters. Unlike GridSearchCV, which evaluates all possible combinations in a specified grid, RandomizedSearchCV randomly samples a specified number of combinations from the hyperparameter space. This approach is more efficient when dealing with a large number of hyperparameters or when computational resources are limited.

| S No | Algorithm Name | Hyper-parameter tuning | Metric used for Evaluation |
|------|---------------------|---|--|
| 1 | KNN | n_neighbors: 2-30, weights: uniform, distance | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 2 | Decision Tree | max_depth: 1-45 (step 5) | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 3 | Linear Regression | None | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 4 | RANSAC Regression | None | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 5 | TheilSen Regression | None | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 6 | Huber Regression | None | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 7 | Lasso Regression | alpha: 0.01, 0.1, 1, 10, 100 | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 8 | Ridge Regression | alpha: 0.01, 0.1, 1, 10, 100 | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 9 | Random Forest | n_estimators: 50-199, max_depth: None, 10, 20, 30, 40, 50 | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 10 | Gradient Boosting | n_estimators: 50-199, max_depth: 3, 4, 5, 6, None | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 11 | XGBoost | n_estimators: 50-199, max_depth: 3, 4, 5, 6, 8, 10, learning_rate: 0.01, 0.1, 0.15, 0.3 | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |
| 12 | AdaBoost | n_estimators: 50-199, learning_rate: 0.01, 0.1, 0.15, 0.3 | Mean Squared Error (MSE), R-squared, MAE, RMSE, ADJ_R2 |

Model Evaluation on multiple models:

Models trained using the best hyper-parameters obtained after using RandomizedSearchCV:

| S No | Model | Training Time | Testing Time | R2 Score | Adj Score | R2 | Hyperparameter |
|------|------------------|---------------|--------------|----------|-----------|----|---|
| 1 | KNeighbours | 0.07761 | 1.048659 | 0.94 | 0.94 | | n_neighbors=29 |
| 2 | DecisionTree | 0.072904 | 0.004241 | 0.94 | 0.94 | | max_depth=11 |
| 3 | LinearRegression | 0.01687 | 0.004502 | 0.91 | 0.91 | | None |
| 4 | RANSAC | 0.321958 | 0.002916 | 0.87 | 0.87 | | None |
| 5 | TheilSen | 12.76159 | 0.004496 | 0.90 | 0.90 | | None |
| 6 | HuberRegressor | 0.501576 | 0.002852 | 0.90 | 0.90 | | None |
| 7 | Lasso | 0.033868 | 0.002928 | 0.91 | 0.91 | | alpha=0.01 |
| 8 | Ridge | 0.00771 | 0.002706 | 0.91 | 0.91 | | alpha=0.1 |
| 9 | RandomForest | 3.781979 | 0.123334 | 0.95 | 0.95 | | max_depth=10, n_estimators=79 |
| 10 | GradientBoosting | 8.884084 | 0.071349 | 0.95 | 0.95 | | max_depth=6, n_estimators=160 |
| 11 | XGBoost | 0.24346 | 0.021409 | 0.95 | 0.95 | | max_depth=5, n_estimators=74, learning_rate=0.3 |
| 12 | AdaBoost | 3.170216 | 0.094205 | 0.93 | 0.93 | | learning_rate=0.1, n_estimators=79 |

Model Evaluation on basic and multiple models:

Observations on basic models:

- From above we can observe that Elastic Net has quite the poor performance of all the models and XGBoost has the best overall R2 scores on both train and test data.

Observations on tuned models:

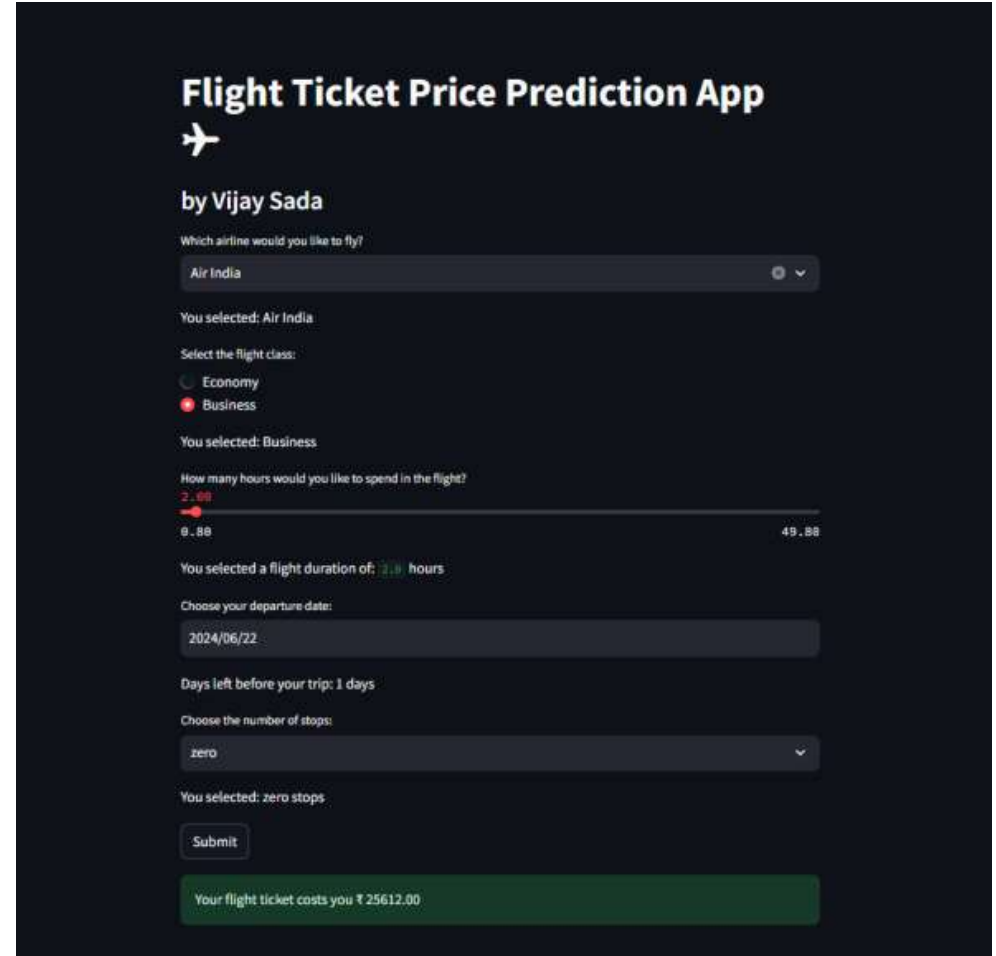
- RandomForest, GradientBoosting, and XGBoost models achieve the highest R2 and Adjusted R2 Scores of 0.95.
- The TheilSen model has an exceptionally high training time (12.761589 seconds), significantly longer than any other model.
- The KNeighbours model has the highest testing time (1.048659 seconds), which is much longer compared to other models.
- Ridge has the shortest training time (0.007710 seconds) and very short testing time (0.002706 seconds).
- XGBoost balances well with a relatively short training time (0.243460 seconds) and testing time (0.021409 seconds).
- Linear Regression, Ridge, and Lasso models all have identical R2 and Adjusted R2 Scores of 0.91, indicating similar predictive performance.

Model Deployment Using Streamlit:

The flight ticket price prediction model has been deployed via a Streamlit web application on a local server environment. This deployment aims to provide users with a convenient interface for predicting flight ticket prices based on various parameters.

Functionality:

The deployed app enables users to input flight details such as airline, class, date, and other relevant parameters. It then provides an estimated price for the flight based on historical data and machine learning predictions.



The screenshot displays the 'Flight Ticket Price Prediction App' interface. At the top, it says 'by Vijay Sada' with an airplane icon. The form includes several input fields: a dropdown for 'Which airline would you like to fly?' (selected: Air India), radio buttons for 'Select the flight class:' (selected: Business), a slider for 'How many hours would you like to spend in the flight?' (selected: 2.00), a date input for 'Choose your departure date:' (selected: 2024/06/22), and a dropdown for 'Choose the number of stops:' (selected: zero). A 'Submit' button is located below the form. At the bottom, a green box displays the result: 'Your flight ticket costs you ₹ 25612.00'.

Challenges Faced:

- While cleaning the dataset, some tags which we're causing problems for changing the wrong data type to correct like “stops” and “time taken”.
- Also, while dealing with the datetime format columns.
- While doing the hyper-parameter tuning and selections of parameters.

Conclusion

The best models are the following:

- Decision Tree can be considered the best model with quite the fast prediction time (0.004241 seconds) and with a slight trade-off in the R2 Score (0.94).
- XGBoost has the best overall performance due to its perfect balance of high R2 Score (0.95) and a bit slow prediction time (0.071349 seconds).

Thus, Decision Tree can be considered the best choice for scenarios where computational efficiency is prioritized and a slight reduction in predictive performance (from 0.95 to 0.94) is acceptable.

THANK
YOU

