

Flight Price Prediction: Optimizing Travel Decisions

1 ABSTRACT

In today's travel landscape, securing optimal flight options at competitive prices stands paramount in planning seamless journeys. This project focuses on empowering travelers with a robust and accurate tool for predicting flight ticket prices, leveraging innovative data-driven methodologies. By harnessing web scraping techniques from Kayak and employing an array of Python libraries, the initiative aims to equip travelers with informed decisions regarding cost-effective flights.

The primary objective involves developing a predictive model capable of forecasting flight ticket prices. Commencing with the scrapping of data from <https://www.kayak.com> meticulous data preprocessing ensued to construct a resilient regression model. This process encompassed crucial steps such as feature identification, data cleaning, and encoding for optimal model readiness.

Utilizing a meticulously structured scraping methodology, user-defined parameters were utilized to navigate Kayak's dynamic content. Selenium and BeautifulSoup, in conjunction with Python libraries such as Pandas, NumPy, and tqdm, formed the backbone of this scraping process. This method culminated in the procurement of a comprehensive data set covering diverse routes and dates from February 1, 2024, to April 30, 2024.

The collected dataset underwent rigorous exploratory data analysis (EDA), including thorough cleaning and outlier handling using the Interquartile Range (IQR) method. The resultant refined dataset, comprising critical flight information like Airline, Source, Destination, Duration, Total Stops, Price, Date, and Average Price, formed the foundational basis for predictive model training.

Further preprocessing involved meticulous handling of categorical data through one-hot encoding, transforming Source and Destination columns to numerical representation. The encoded dataset ('final_df') was enriched with numerical features, allowing for comprehensive feature analysis and model development.

Diverse regression models were trained and assessed, with the RandomForestRegressor emerging as the model of choice due to its superior performance metrics on both training and validation sets. Model evaluations, hyperparameter tuning, and extensive feature analysis validated the efficacy of the RandomForestRegressor in predicting flight ticket prices.

The final model demonstrated exceptional predictive capabilities, achieving an R2 score of 0.946, an MAE of \$61.87, an MSE of 40409.87, and an RMSE of \$201.02 on the test set. This validated the model's efficiency in forecasting flight ticket prices within approximately \$61.87, empowering travelers to make informed and cost-effective travel decisions.

2 OBJECTIVE

The fundamental goal of this project centers around the predictive forecasting of flight ticket prices, intending to equip travelers with the necessary tools to streamline their journey planning while ensuring economic feasibility. Commencing this journey involved the meticulous extraction of data from the Kayak platform (<https://www.kayak.com/>) to form the foundation for constructing a robust and adaptable regression model capable of forecasting flight ticket prices.

3 INITIAL STEPS AND MODEL DEVELOPMENT

The inception of this project involved a rigorous data cleansing process, meticulously engineered to sift through the dataset and discern pertinent features crucial for predictive modeling. Following this, the dataset underwent a transformative phase via one-hot encoding, preparing it meticulously for regression modeling.

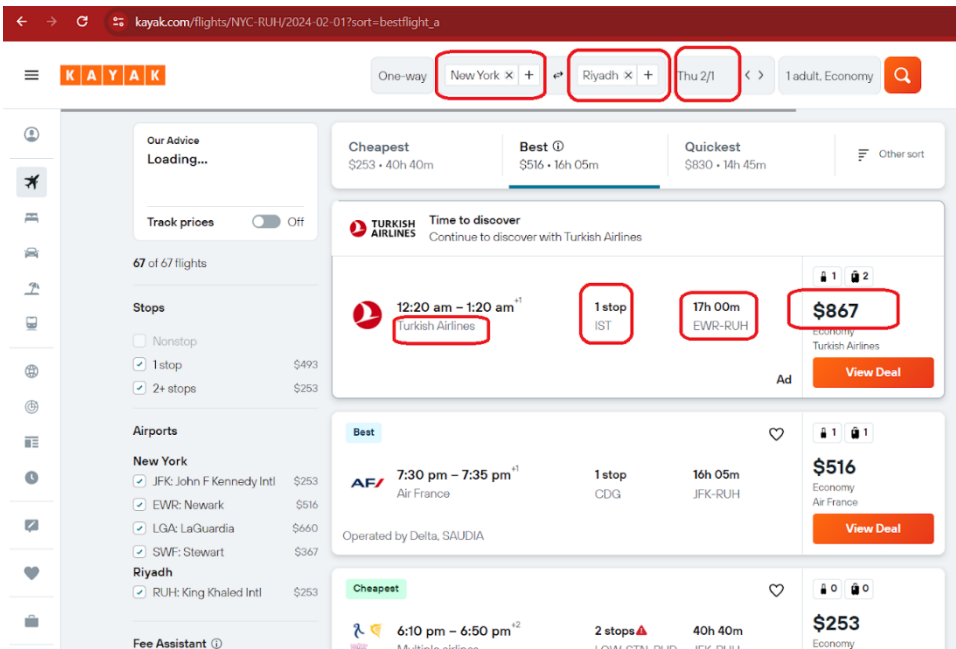
Through extensive analysis, the project delved deeper into understanding the significance and impact of various features within the dataset. This meticulous exploration revealed the pivotal role of certain predictors such as the total number of stops and flight duration in determining the target variable – the flight ticket price.

Part1: Flight prices Scraper

4 KAYAK SCRAPER:

The meticulous extraction of data from the Kayak platform (<https://www.kayak.com/>) forms the cornerstone of our project, holding paramount significance in the creation of a comprehensive dataset essential for testing and validating the efficacy of our regression model. This dataset goes beyond being a mere compilation of flight information; it serves as the linchpin for ensuring the adaptability and accuracy of our model across an array of diverse flight scenarios. Its pivotal role in the project becomes increasingly evident as we progress, steadfastly anchored in our commitment to leverage data-driven insights. Our primary objective is to empower travelers by facilitating informed decisions, optimizing travel plans, and facilitating access to the most economically viable flight options available.

The phase involving the meticulous extraction of data from Kayak assumes paramount importance, primarily due to its role in furnishing the dataset that is fundamental for rigorously testing and validating

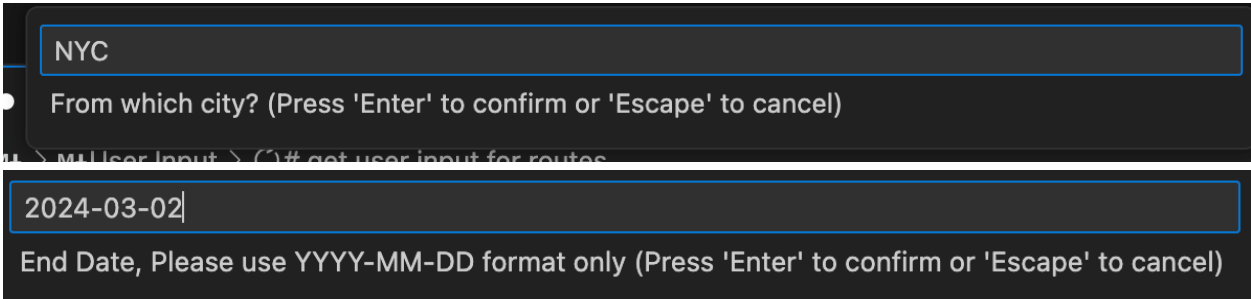


the regression model's efficacy. This dataset is more than a repository of flight-related information; it acts as the backbone, ensuring the model's adaptability and accuracy across diverse flight scenarios. As our project unfolds, our unwavering dedication remains in utilizing data-driven insights to enable travelers to make informed decisions, optimizing their travel itineraries, and securing

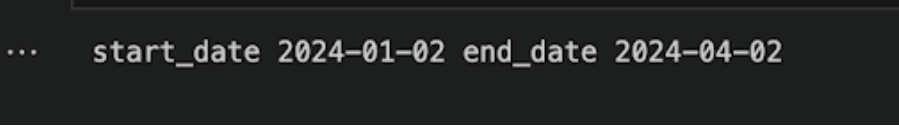
the most cost-effective flight options in the market.

4.1 SCRAPING METHODOLOGY:

The scraping involved user interaction to input specific routes and dates. The process commenced by prompting the user to input the origin and destination cities for the desired flight routes. This interaction, facilitated through a while loop, allowed users to input multiple routes until they indicated completion by entering "-1". Subsequently, the selected routes were displayed for confirmation.

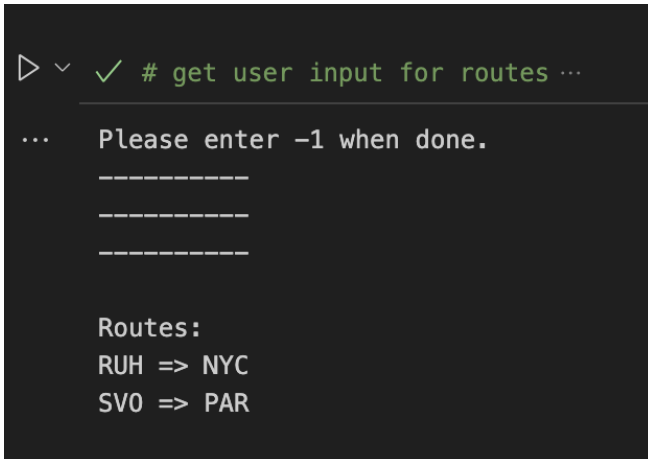


Once the routes were defined, the system prompted users to input the start and end dates in the "YYYY-MM-DD" format. This input helped in defining the duration of the desired flight data collection.



The scraping process was meticulously architected to target specific routes and

date ranges, facilitating the acquisition of comprehensive flight data from the Kayak platform. This precision-driven methodology relied on a harmonious blend of Selenium and BeautifulSoup, two powerful Python libraries. Selenium's prowess in web automation coupled with BeautifulSoup's adeptness in parsing HTML content allowed us to precisely navigate Kayak's dynamic web interface. Crucially, this



```
▶ ✓ # get user input for routes ...  
... Please enter -1 when done.  
-----  
-----  
-----  
  
Routes:  
RUH => NYC  
SVO => PAR
```

process integrated user-defined parameters for routes and date ranges, enabling targeted extraction of relevant flight information.

Following the data extraction, the collected information for each route was saved as separate CSV files. This step ensured organized storage, allowing easy access to specific route data for further analysis. Each CSV file was named based on the respective route for clear identification and future reference.

4.2 SCRAPING IMPLEMENTATION:

The scraping implementation involved a systematic approach using Python's diverse toolset to streamline the data extraction process from Kayak's platform. Selenium, an automation tool, played a pivotal role in navigating and interacting with Kayak's dynamic web interface. BeautifulSoup, a robust HTML parser, was employed to extract essential content while Pandas facilitated the organization and management of the extracted data for analysis.

NumPy, renowned for its numerical computation capabilities, enhanced the data analysis pipeline, contributing to more effective processing. The utilization of tqdm, a visualization tool, ensured comprehensive progress tracking and seamless interaction with Kayak's web interface throughout the scraping process, enhancing the overall efficiency of data extraction.

The scraping itself involved a systematic loop through defined routes and dates. For each route, a dataframe structure was created to organize data columns such as Airline, Source, Destination, Duration, Total Stops, Price, and Date. The script navigated through Kayak's flight pages, clicking on the "show more" button to retrieve all available flights. Data such as airlines, stops, prices, and durations were extracted from the webpage using defined functions and appended to the respective dataframes.

Upon completion of each route's data extraction, the collected information was stored as individual CSV files for ease of access and future reference. This method ensured the organization and storage of data pertinent to each specific route, allowing for focused analysis and model training. Finally, the web driver used for the scraping process was closed.

4.3 SCRAPING RESULTS:

The outcome of our scraping endeavor was nothing short of fruitful. This meticulous effort culminated in the procurement of a comprehensive dataset spanning a crucial timeframe from February 1, 2024, to April 30, 2024. Comprising rich flight information for 12 specific routes meticulously stored as CSV files, this dataset encapsulates pivotal details essential for training and refining the predictive models crucial to our project's success. The targeted routes encompass a spectrum of critical flight paths, including:

- RUH => NYC

NYC_PAR

- RUH => SVO

- RUH => PAR

- NYC => RUH

- NYC => SVO

- NYC => PAR

- SVO => PAR

- SVO => RUH

- SVO => NYC

- PAR => NYC

- PAR => RUH

- PAR => SVO

| Airline | Source | Destination | Duration | Total stops | Price | Date |
|-------------------|--------|-------------|----------|-------------|-----------|--------|
| Air France | NYC | PAR | 7h 20m | nonstop | 1,031 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 20m | nonstop | 1,031 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 20m | nonstop | 1,031 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 10m | nonstop | 1,238 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 20m | nonstop | 1,238 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 30m | nonstop | 1,238 USD | 2/1/24 |
| Air France | NYC | PAR | 7h 30m | nonstop | 1,238 USD | 2/1/24 |
| Lufthansa | NYC | PAR | 7h 05m | nonstop | 1,353 USD | 2/1/24 |
| American Airlines | NYC | PAR | 7h 20m | nonstop | 1,353 USD | 2/1/24 |
| United Airlines | NYC | PAR | 7h 05m | nonstop | 1,358 USD | 2/1/24 |
| JetBlue | NYC | PAR | 7h 20m | nonstop | 1,634 USD | 2/1/24 |
| JetBlue | NYC | PAR | 7h 20m | nonstop | 1,634 USD | 2/1/24 |

This meticulously curated dataset lays the foundational bedrock essential for the iterative training and refinement of our predictive models, acting as the cornerstone in the quest for optimal and precise flight price prediction.

Part2: Predicting Flight Ticket Prices

This primary section serves as the core of our flight price prediction project, designed to empower customers by aiding them in making well-informed decisions regarding travel timing and discovering the most cost-effective flights tailored to their desired destinations. The dataset crucial for training our predictive models is procured through web scraping techniques employed on Kayak, as comprehensively detailed in the "kayak-scraper" documentation.

5 SCRAPED DATA EDA:

5.1 DATA LOADING AND CLEANING:

The data underwent a meticulous cleaning process to ensure accuracy and consistency. Outliers were addressed using the Interquartile Range (IQR) method, significantly refining the dataset. Approximately 5,266 rows, among a total of 55,363, were dropped post-outlier handling, showcasing the enhanced quality of the dataset.

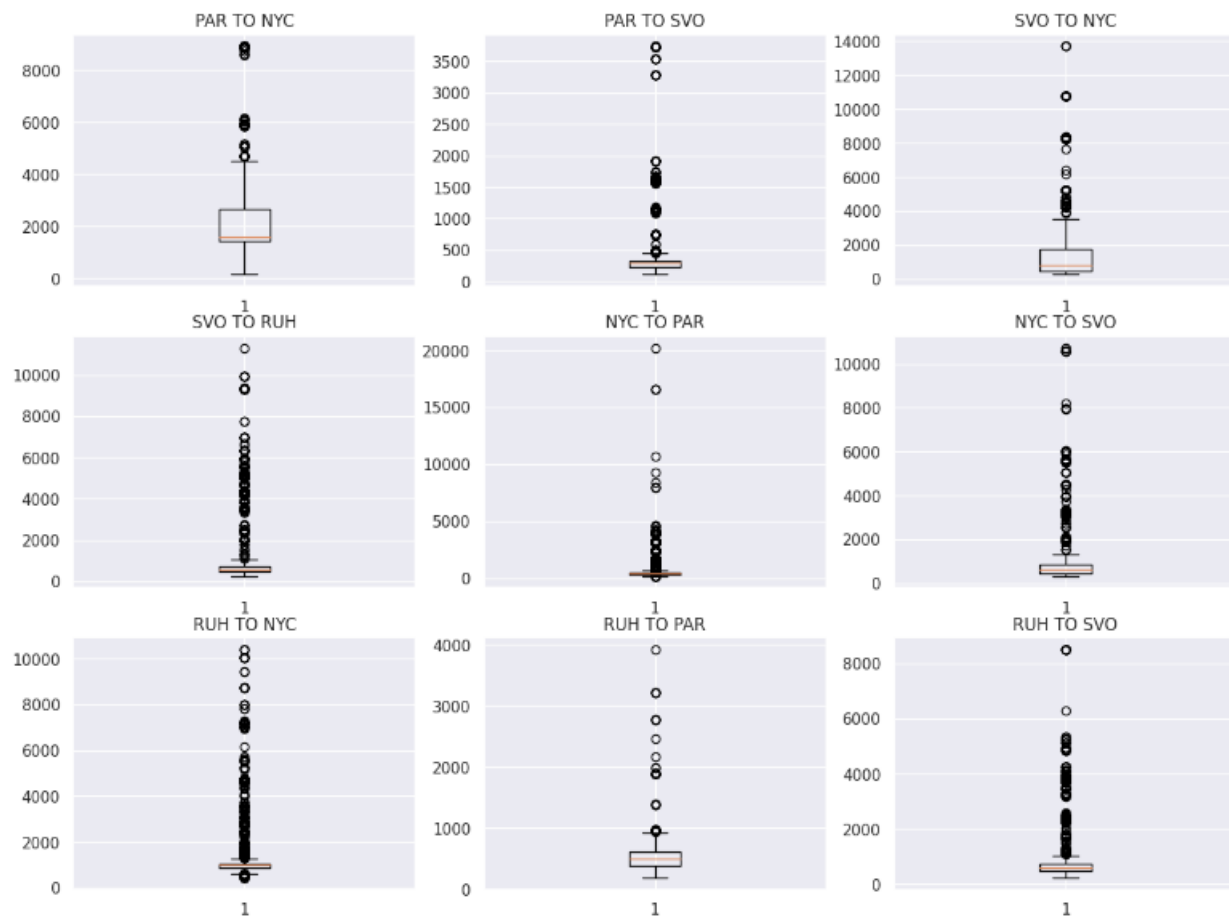
The cleaned dataset is structured with columns encompassing flight information such as Airline, Source, Destination, Duration, Total Stops, Price, Date, and Average Price. Notably, the Average Price column has

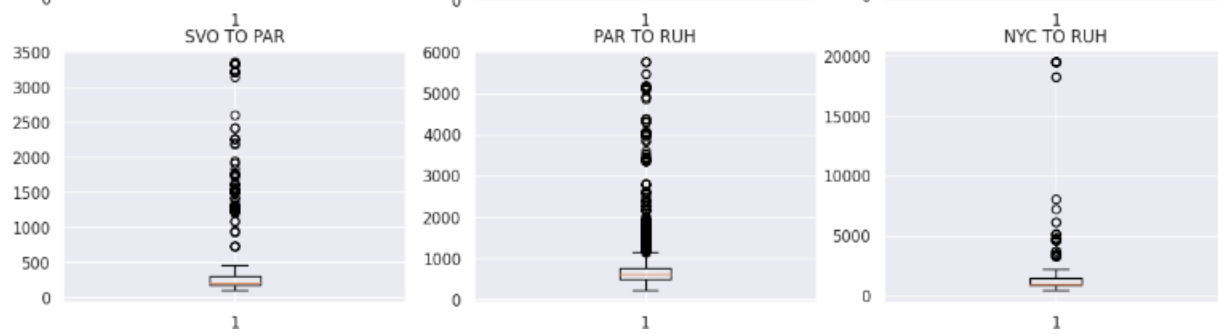
been introduced by replacing the Airline column, consolidating average prices per airline for analytical purposes.

5.1.1 Studying Outliers

Below, a series of boxplots depict potential outliers in flight prices across various routes. Employing a 4x3 subplot grid configuration, these boxplots are generated using the 'Price' column within the DataFrame collection. Matplotlib's boxplot function is utilized within a loop to create these representations for different routes.

Each subplot within the grid corresponds to a specific route, with its title reflecting the source and destination cities for easy identification. This visualization approach through boxplots aids in the detection of outliers within flight prices. Outliers, when present, manifest as data points extending beyond the upper or lower bounds of the box-and-whisker plots, providing a clear visual depiction of their existence and their deviation extent from the median price distribution across each route.



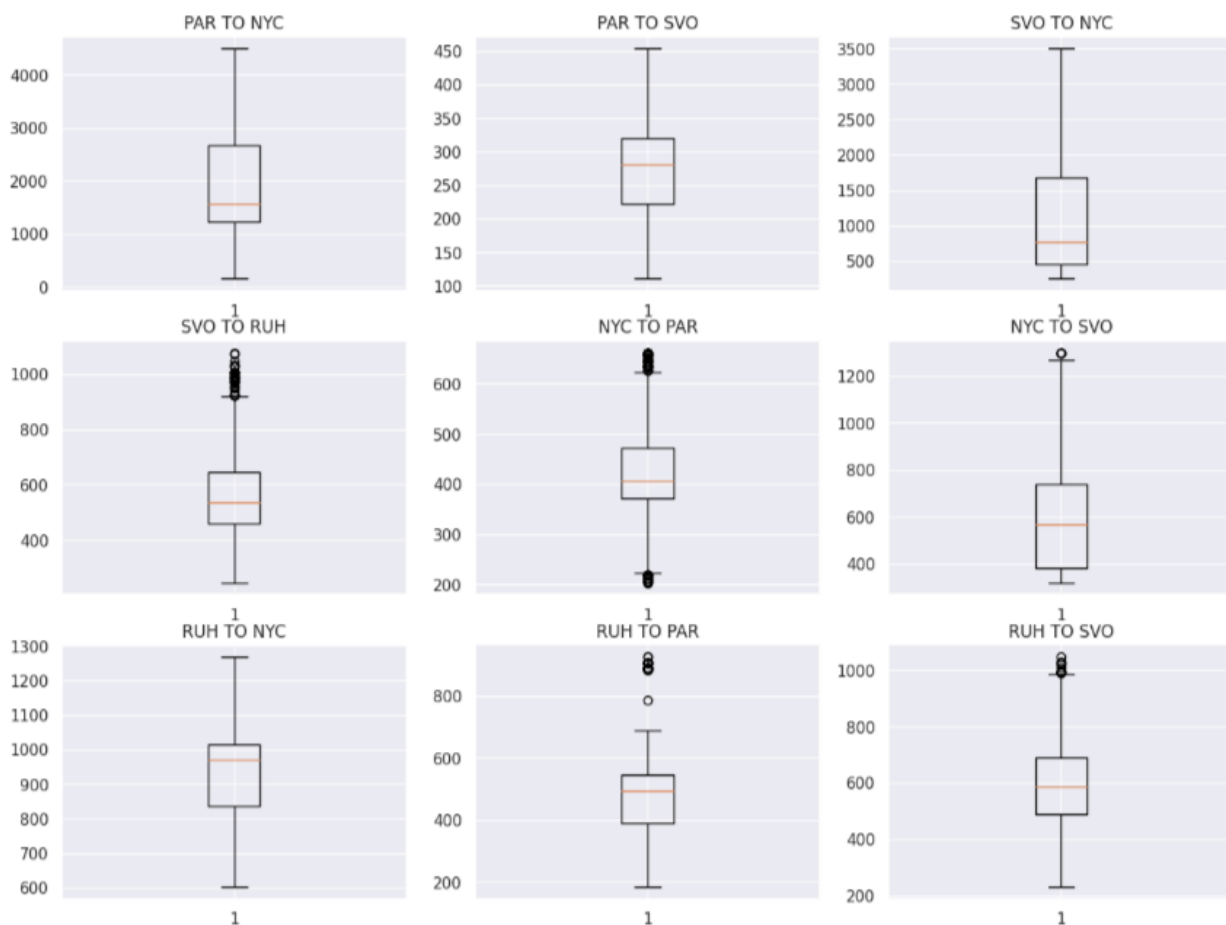


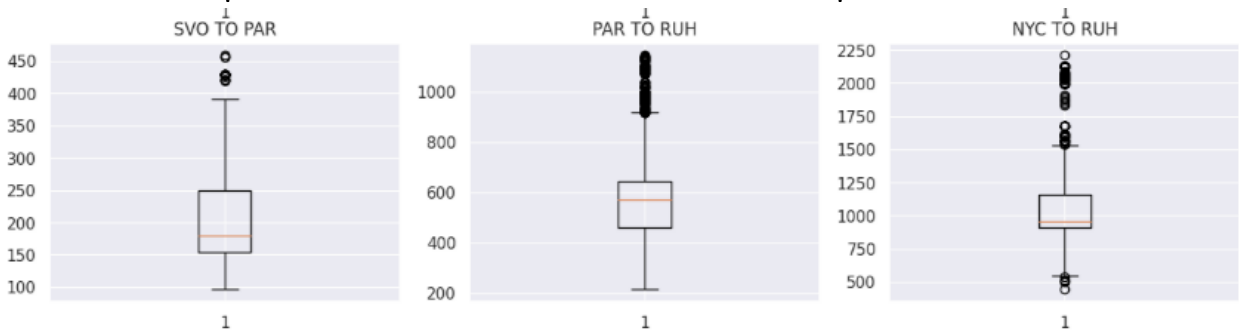
5.1.2 Interquartile Range (IQR)

In this phase, we handle outliers present in the flight price data across various routes using the Interquartile Range (IQR) method for outlier detection and subsequent removal.

Initially, the process involves determining the lower and upper boundaries of the IQR range for each route's 'Price' column. This computation involves obtaining the lower and upper quantiles, which act as benchmarks defining the limits beyond which data points are classified as outliers.

Following this calculation, the procedure identifies outliers within each route's 'Price' column based on





the derived lower and upper bounds. It establishes boolean masks ('low' and 'up') to pinpoint values falling below the lower bound or exceeding the upper bound, respectively.

Using these masks, the methodology filters out identified outliers from the 'Price' column in each route's dataset. Outliers are replaced with 'NaN' (Not a Number) values, ensuring they are excluded from subsequent analyses. The data points marked as 'NaN' are then eliminated from the DataFrame, preserving the integrity of the remaining dataset.

Finally, the process rearranges the index of each DataFrame, ensuring the sequential order of the data points following the elimination of outliers. This reindexing guarantees that the dataset maintains a consistent and organized structure post-outlier removal.

The resulting cleaned and updated DataFrames are displayed below, showcasing the effect of outlier removal on the flight price data across multiple routes.

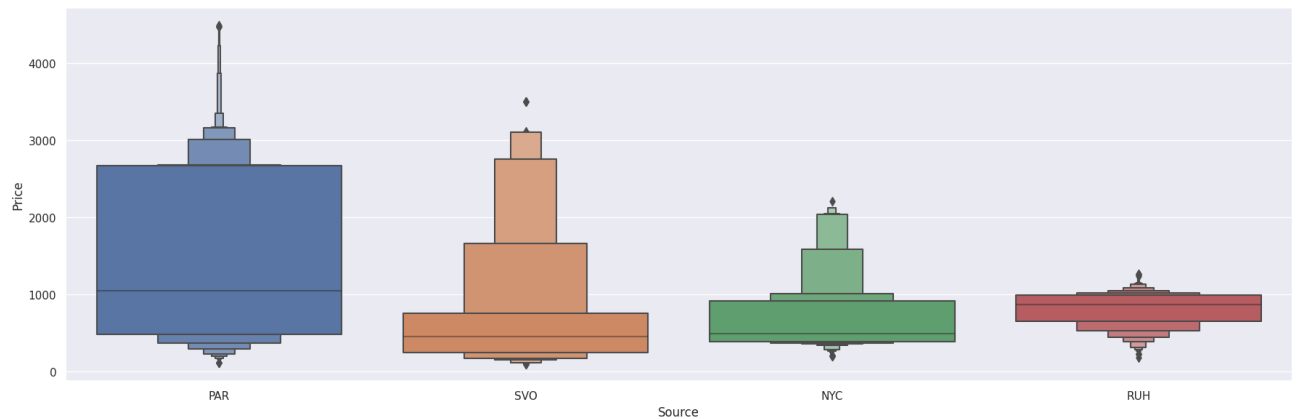
| | Airline | Source | Destination | Duration | Total stops | Price | Date | Average Price |
|------|----------------------------|--------|-------------|----------|-------------|---------|------------|---------------|
| 0 | TAP AIR PORTUGAL | PAR | NYC | 770 | 1 stop | 379.47 | 2024-02-01 | 441.343760 |
| 1 | TAP AIR PORTUGAL | PAR | NYC | 770 | 1 stop | 379.47 | 2024-02-01 | 441.343760 |
| 2 | TAP AIR PORTUGAL | PAR | NYC | 810 | 1 stop | 379.47 | 2024-02-01 | 441.343760 |
| 3 | TAP AIR PORTUGAL | PAR | NYC | 890 | 1 stop | 379.47 | 2024-02-01 | 441.343760 |
| 4 | TAP AIR PORTUGAL | PAR | NYC | 1030 | 1 stop | 379.47 | 2024-02-01 | 441.343760 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 3046 | SAS, Pegasus Airlines | NYC | RUH | 1300 | 2 stops | 930.40 | 2024-04-13 | 838.732500 |
| 3047 | Qatar Airways, SAUDIA | NYC | RUH | 1420 | 2 stops | 1018.40 | 2024-04-13 | 1972.813714 |
| 3048 | Qatar Airways, MEA | NYC | RUH | 1415 | 2 stops | 1049.87 | 2024-04-13 | 1133.548015 |
| 3049 | Emirates, Turkish Airlines | NYC | RUH | 1100 | 2 stops | 1053.60 | 2024-04-13 | 1023.900221 |
| 3050 | Lufthansa, MEA | NYC | RUH | 1115 | 3 stops | 1139.73 | 2024-04-13 | 1023.714222 |

50097 rows × 8 columns

5.2 HANDLING CATEGORICAL DATA WITH ONE-HOT ENCODING:

Categorical data encoding was pivotal to prepare the dataset for regression modeling. Categorical columns, such as Source and Destination, underwent meticulous encoding techniques:

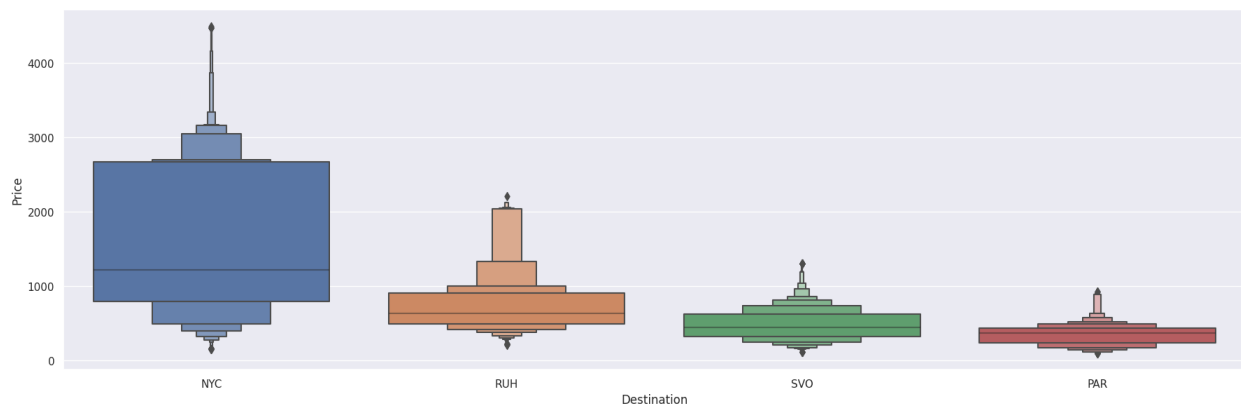
5.2.1 For Source:



Paris (PAR) emerged as the most frequent, followed by New York City (NYC), Riyadh (RUH), and Sheremetyevo (SVO). One-hot encoding was applied to Source data, transforming it into numerical representation. Source Counts = PAR 23054, NYC 9411, RUH 9140, SVO 8492

5.2.2 For Destination:

New York City (NYC) emerged as the predominant destination, succeeded by Riyadh (RUH), Paris (PAR), and Sheremetyevo (SVO). Like Source, Destination underwent one-hot encoding for compatibility in the regression models. Destination Counts = NYC 24926, RUH 11046, PAR 8042, SVO 6083



Additionally, the Total Stops column, representing the number of stops in flights, showcased categories encompassing 1 stop, 2 stops, 3 stops, and nonstop. This data was presented in its encoded format for model compatibility.

The resultant dataset ('final_df') comprises critical numerical features including Duration, Total Stops, Price, Average Price, Source encoded columns (PAR, RUH, SVO), and Destination encoded columns (PAR, RUH, SVO).

| | Duration | Total stops | Price | Average Price | Source_PAR | Source_RUH | Source_SVO | Destination_PAR | Destination_RUH | Destination_SVO |
|-------|----------|-------------|---------|---------------|------------|------------|------------|-----------------|-----------------|-----------------|
| 0 | 770 | 1 | 379.47 | 441.343760 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 770 | 1 | 379.47 | 441.343760 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 810 | 1 | 379.47 | 441.343760 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 890 | 1 | 379.47 | 441.343760 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1030 | 1 | 379.47 | 441.343760 | 1 | 0 | 0 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 50092 | 1300 | 2 | 930.40 | 838.732500 | 0 | 0 | 0 | 0 | 1 | 0 |
| 50093 | 1420 | 2 | 1018.40 | 1972.813714 | 0 | 0 | 0 | 0 | 1 | 0 |
| 50094 | 1415 | 2 | 1049.87 | 1133.548015 | 0 | 0 | 0 | 0 | 1 | 0 |
| 50095 | 1100 | 2 | 1053.60 | 1023.900221 | 0 | 0 | 0 | 0 | 1 | 0 |
| 50096 | 1115 | 3 | 1139.73 | 1023.714222 | 0 | 0 | 0 | 0 | 1 | 0 |

50097 rows × 10 columns

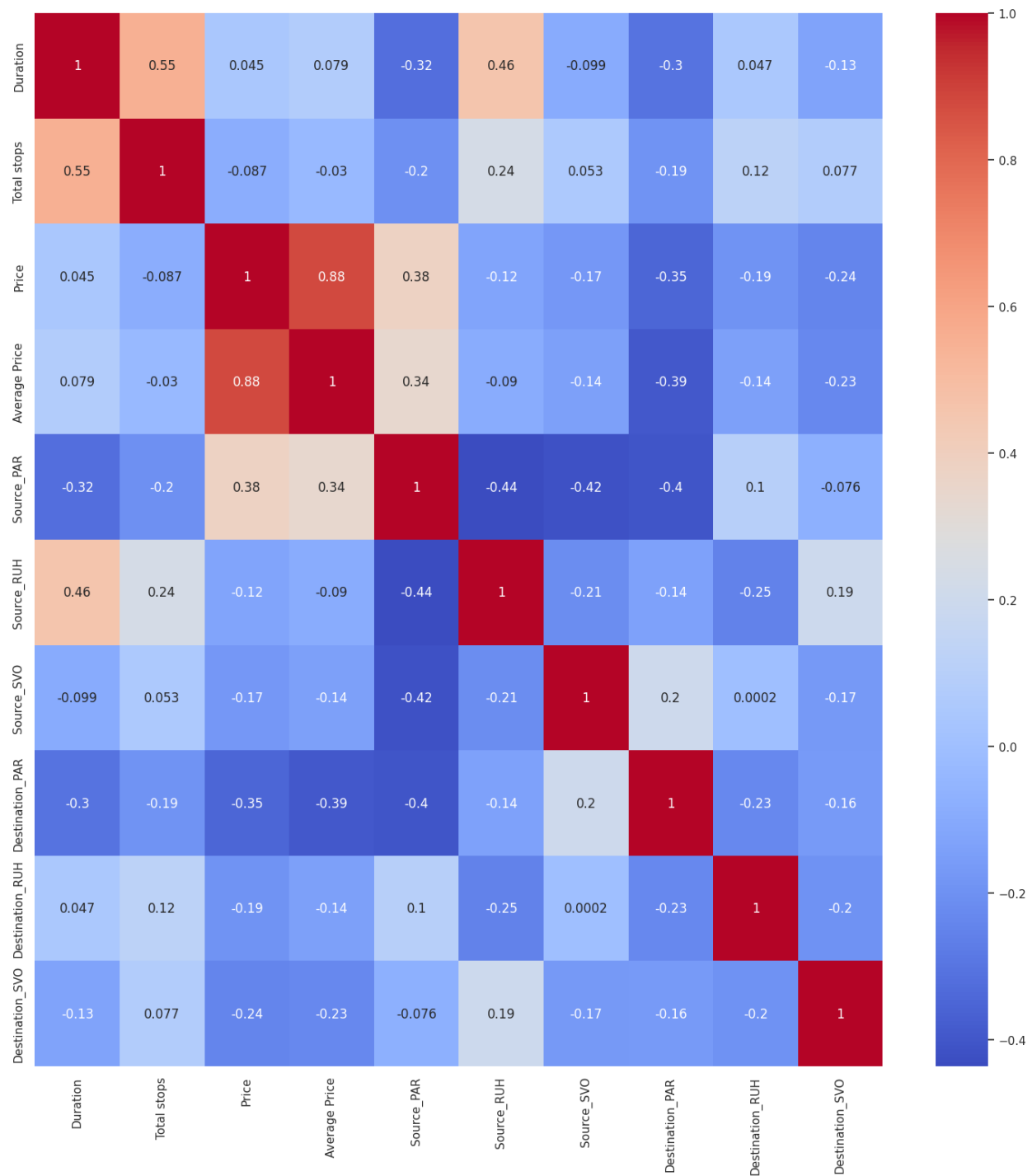
5.3 FEATURE ENGINEERING

"Feature Analysis" in machine learning revolves around evaluating the attributes or characteristics that function as input variables for regression models.

This process involves thoroughly investigating these features (also known as independent variables) present in the dataset to comprehend their relevance, significance, and potential impact on predicting the target variable, which in this context is the flight ticket prices. Techniques like correlation analysis, assessing feature importance, and employing statistical tests are used to determine which features should be integrated into the regression models. This selection is based on their ability to predict accurately and contribute to the model's overall accuracy.

In this project, the 'final_df' dataset underwent enrichment by incorporating various numerical features such as duration, total stops, price, average price, and encoded source/destination columns. Through correlation analysis, notable correlations between these features were identified. Furthermore, feature importance analysis highlighted the pivotal role of the 'Average Price' feature in forecasting flight prices, emphasizing its significance in the predictive models.

5.3.1 Correlation Heatmap



5.3.1.1 Analysis

The output provided represents the correlation matrix between various features in a dataset. Each value in the matrix denotes the correlation coefficient between two respective features, ranging from -1 to 1. Here are some observations based on the correlation values:

1. **Duration and Total Stops:** There is a moderate positive correlation of approximately 0.55 between the duration of the flight and the total number of stops. This suggests that flights with longer durations tend to have more stops, which is intuitive as longer flights often require layovers or multiple stops.
2. **Price and Average Price:** The correlation between the actual price and the average price is quite strong, with a coefficient of around 0.88. This high positive correlation implies that the actual price and the average price are highly positively correlated, indicating that they tend to move together.
3. **Source and Destination Correlations:** There are various moderate correlations between the source and destination airports. For instance, Source_PAR and Destination_PAR show a negative correlation, suggesting that flights originating from a particular airport might be less likely to travel to the same destination airport.
4. **Other Correlations:** The other correlations appear to be mostly weak, indicating a lack of strong linear relationships between these variables.

Overall, the correlation matrix provides insights into the relationships between different features, helping in understanding how they might influence or relate to each other within the dataset.

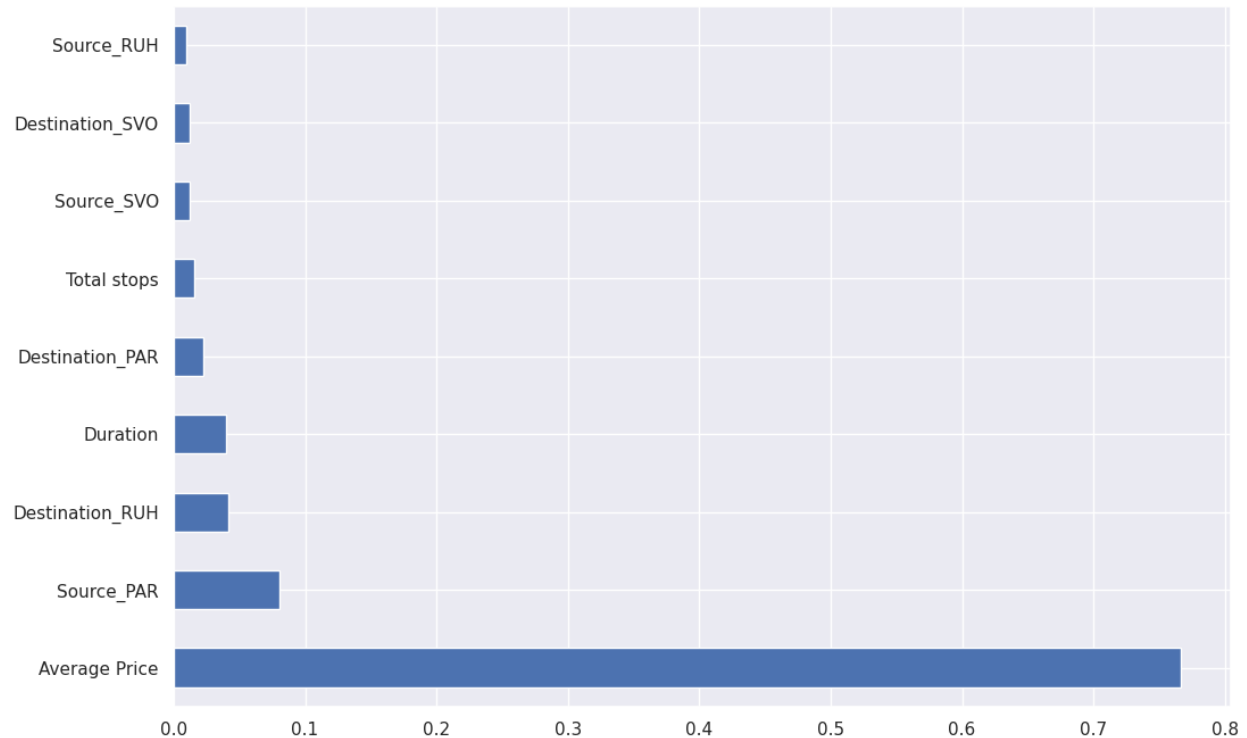
5.3.2 Feature Importance:

This section aims to determine the importance of different features in predicting the target variable "Price" in the dataset.

It utilizes a machine learning technique called Extra Trees Regressor to evaluate the significance of various input features in influencing the predicted flight ticket prices. The feature importance scores are computed after fitting the Extra Trees Regressor model with the feature matrix (X) and the target variable (y).

The values obtained from ``selection.feature_importances_`` represent the relative importance of each feature in the dataset for predicting the flight prices. These values signify the contribution of each feature towards the predictive accuracy of the model. A higher value indicates a greater impact or importance of that particular feature in determining the flight ticket prices.

The subsequent part of the code generates a horizontal bar plot to visualize the top 20 important features. This visualization helps to identify and comprehend which features are the most influential in predicting flight prices. The bars in the plot represent the importance scores of these features, sorted in descending order. This graphical representation aids in understanding the relative significance of various features in the dataset concerning their impact on the predicted flight prices.



5.3.2.1 Analysis

The Extra Trees Regressor model has provided insights into the significance of various features concerning their impact on predicting the target variable, "Price." Here are the top 8 features ranked by their importance scores:

1. **Average Price (0.764990):** The most influential feature in predicting the flight price is the average price itself. This high importance score suggests that the average price has a substantial impact on determining the final flight price.
2. **Source_PAR (0.078305):** The source airport, particularly Source_PAR, is the second most important feature. Flights originating from this specific airport significantly contribute to determining the price.
3. **Destination_RUH (0.045404):** The destination airport, specifically Destination_RUH, also holds notable importance in predicting the flight price. Flights headed to this destination tend to influence the overall price.
4. **Duration (0.039548):** The duration of the flight is also a reasonably important factor in determining the flight price. Longer flight durations might correspond to higher prices.
5. **Destination_PAR (0.022988):** Similar to the source airport, the destination airport, particularly Destination_PAR, holds relevance in affecting the flight price.
6. **Destination_SVO (0.014498):** The specific destination airport Destination_SVO plays a role in determining the flight price, although comparatively less significant than the other factors.

7. **Total stops (0.014155):** The number of stops during the flight journey also contributes to determining the flight price, though to a slightly lesser extent.
8. **Source_SVO (0.012308):** Similar to the destination airport, the source airport Source_SVO also exhibits some influence on the flight price, although relatively less important than other features.

These importance scores provide valuable insights into the key factors driving the flight price. It's evident that both the source and destination airports, along with flight duration and average price, play substantial roles in predicting the price, as indicated by their higher importance scores.

6 MODELING:

6.1 DATA SPLITTING: TRAIN, VALIDATION, TEST

The dataset was strategically split into a 60% Train, 20% Validation, and 20% Test set to facilitate robust model development and evaluation.

6.2 BASELINE MODELS AND MODEL SELECTION:

In the pursuit of identifying the most effective regression model for flight price prediction, a comprehensive evaluation was conducted on various regression techniques. The performance of each model was rigorously assessed, leading to the selection of the RandomForestRegressor as the optimal choice due to its superior performance compared to other models.

Here is a summary of the metrics obtained from the evaluation of diverse regression models: Baseline Models Metrics Summary:

6.2.1 Linear Regression:

The linear regression model demonstrates reasonably good performance, with the training and validation scores indicating a decent fit to the data. The errors, however, could potentially be reduced.

```
Train score 0.8038544368965256
Val score 0.7953448912733655
MAE: 227.23530111124455
MSE: 153145.78261322278
RMSE: 391.33845021058534
```

6.2.2 Polynomial Regression (Degree 1 to 5):

Degree 1: Degree 1 polynomial regression shows results similar to Linear Regression, hinting at a linear relationship in the data. Overfitting indicated by negative scores.

```
-----
Degree 1
Train score 0.8038544368965255
Val score 0.7953448912733665
MAE: 227.235301111255
MSE: 153145.782613222
RMSE: 391.3384502105843
```

Degree 2 to 5: As the degree increases, the model's performance improves, indicating reduced overfitting and lower errors compared to lower-degree polynomials. Degree 5 showed less overfitting.

```
Degree 2
Train score 0.8558688362482951
Val score 0.8528578067517942
MAE: 178.60197669929642
MSE: 110108.20340928955
RMSE: 331.82556171773376
-----
Degree 3
Train score 0.8754875398868874
Val score 0.8710903520073184
MAE: 160.77403603436252
MSE: 96464.57912078986
RMSE: 310.5874741852766
-----
Degree 4
Train score 0.8810249644036683
Val score 0.8695776745806598
MAE: 152.7019171177896
MSE: 97596.53311787493
RMSE: 312.40443837736194
-----
Degree 5
Train score 0.8858240541398387
Val score 0.8800425155595014
MAE: 150.1228314766688
MSE: 89765.57169404681
RMSE: 299.6090313959958
```

6.2.3 Lasso Regression:

Lasso regression exhibits performance similar to linear regression, suggesting that the L1 regularization technique has a marginal impact on model improvement.

```
Train score 0.8036627330514835
Val score 0.7948787910310455
MAE: 225.98570791601813
MSE: 153494.57081025557
RMSE: 391.783831736655
```

6.2.4 Ridge Regression:

Similar to Lasso and Linear regression, Ridge regression also demonstrates comparable performance, indicating minimal variance in the results among these models.

```
Train score 0.8038544219198862
Val score 0.7953425562795555
MAE: 227.2229912856974
MSE: 153147.52991607125
RMSE: 391.34068267440745
```

6.2.5 ElasticNet Regression:

ElasticNet regression performs slightly worse compared to the other models, displaying higher errors and lower scores on both training and validation data:

```
Train score 0.7898553607141706
Val score 0.7787198734635952
MAE: 228.9735416884602
MSE: 165586.47553937347
RMSE: 406.9231813737987
```

6.2.6 Random Forest Regression:

Random Forest stands out with significantly superior performance, showcasing the lowest errors and highest scores for both training and validation data. This model demonstrates robustness and high predictive accuracy

```
Train score 0.9644761084113582
Val score 0.9468733246059319
MAE: 63.42476287759532
MSE: 39755.30506658855
RMSE: 199.38732423749647
```

In summary, among these models, Random Forest Regression outperforms the others, showing superior predictive capability and the lowest errors on both training and validation data. Polynomial regression demonstrates improved performance with increasing degrees but might lead to overfitting for higher degrees. Overall, the choice of the best model depends on the trade-offs between complexity, interpretability, and accuracy needed for the specific application.

7 FEATURE SCALING AND MODEL SELECTION:

Feature scaling is a critical preprocessing step aimed at standardizing the range of independent variables or features in the dataset. In this project, the **StandardScaler** method from the scikit-learn library is utilized to scale the features.

7.1 FEATURE SCALING

The process involves transforming the features to have a mean of zero and a standard deviation of one. This scaling ensures that all features are on a similar scale, preventing certain variables from dominating merely due to their larger magnitude compared to others.

The scaled features are then used to train and evaluate various regression models. The Linear Regression model is initially used as a baseline model to establish a comparison with other models.

The following models are trained and assessed after scaling:

7.1.1 Linear Regression:

```
Train score 0.8040357223322144
Val score 0.7891035984538433
MAE: 225.09235539537684
MSE: 152995.68380136567
RMSE: 391.146626984518
```

7.1.2 Polynomial Regression (Degree 1 to 5):

Polynomial - Degree 1

Train score -0.3079480631703013
Val score -0.3153015545186666
MAE: 756.619594657905
MSE: 954191.0590377726
RMSE: 976.8270363978326

Polynomial - Degree 2
Train score -6.619866701696714
Val score -6.708250742830492
MAE: 1819.2993972374343
MSE: 5591983.005236949
RMSE: 2364.737407247779

Polynomial - Degree 3
Train score -6.396139020230291
Val score -6.597515038614633
MAE: 1978.6373777102278
MSE: 5511649.321667562
RMSE: 2347.6902099015456

Polynomial - Degree 4
Train score -67.38613658531676
Val score -65.94440159195507
MAE: 5158.441261042131
MSE: 48565098.4234209
RMSE: 6968.8663657312945

Polynomial - Degree 5
Train score -1.2998648856939954
Val score -1.3170926486628711
MAE: 977.4048403689628
MSE: 1680944.632598231
RMSE: 1296.512488408126

7.1.3 Lasso Regression:

Train score 0.8040183311451377
Val score 0.7891286832351017
MAE: 224.60280304137635
MSE: 152977.48594102522
RMSE: 391.12336409504513

7.1.4 Ridge Regression:

Train score 0.8040357214700532
Val score 0.7891039650109529
MAE: 225.0937616466228
MSE: 152995.41788096476
RMSE: 391.14628706018004

7.1.5 ElasticNet Regression:

Train score 0.7354476290670592
Val score 0.7243019152118472
MAE: 304.4466319557438

MSE: 200006.3381624778

RMSE: 447.2206817248927

7.1.6 Random Forest Regression:

Train score -1.0693924886375732

Val score -1.0832919574956343

MAE: 885.6332189325535

MSE: 1511332.9353093393

RMSE: 1229.362816791422

RMSE: 199.38732423749647

For each model, metrics such as training score, validation score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are computed and displayed. These metrics provide insights into the models' accuracy and predictive capabilities after feature scaling, aiding in the comparative analysis of model performance.

7.2 FEATURE SCALING IMPACT

Despite employing feature scaling techniques, there was no discernible improvement observed in the overall performance of the regression models. In fact, scaling the features led to adverse effects on certain algorithms. For instance:

- Linear Regression: No noticeable enhancement was observed.
- Polynomial Regression: Scaling worsened the results, indicating its unsuitability for this dataset, and so on.

Following a comprehensive analysis of various regression models, the Random Forest algorithm emerged as the most robust performer. It exhibited the most favorable performance metrics among the considered models:

- Train score: 0.9648778537711422
- Validation score: 0.9448134490695079
- MAE: 61.717733027545194
- MSE: 40035.31608101726
- RMSE: 200.0882707232417

Given these results, we have chosen the **Random Forest model** as the optimal choice for our predictive model.

7.3 MODEL SELECTION

The Random Forest model exhibited exceptional performance and was chosen as the final model for further refinement.

Retrained on combined train/validation sets:

- Train score: 0.963, Test score: 0.946
- MAE: 61.87, MSE: 40409.87, RMSE: 201.02

8 RETRAINING, TESTING, AND HYPERPARAMETER TUNING:

8.1 RETRAINING

In this phase, the Random Forest model was retrained using the combined training and validation datasets to enhance its learning from a more extensive pool of data. By amalgamating the training and validation sets ('X_train' and 'X_val'), along with their corresponding target variables ('y_train' and 'y_val'), the model's knowledge base was expanded. The rationale behind this step is to allow the model to glean insights from a larger dataset, potentially improving its predictive capabilities by learning from a more comprehensive range of patterns and variations present in the data. This retraining process ensures that the model captures a more generalized representation of the underlying relationships between features and the target variable, which is essential for its performance when applied to new, unseen data.

The retrained Random Forest model is subsequently evaluated and scored using the separate test dataset, assessing its performance and generalization on previously unseen data. By testing the model on an independent dataset ('X_test'), we can validate its predictive power and assess how effectively it can extrapolate patterns learned during training to make accurate predictions on new, real-world data.

```
Train score 0.963236185218782
Test score 0.9458897010690771
MAE: 61.86501612022038
MSE: 40409.86746202512
RMSE: 201.02205715300278
```



8.2 HYPERPARAMETER TUNING

In an effort to fine-tune the Random Forest model's performance, Randomized Search CV was employed to optimize its hyperparameters. Hyperparameters play a crucial role in model training and complexity, influencing predictive accuracy. The objective of this process is to identify an optimal combination of these parameters, enhancing the model's effectiveness in predicting outcomes on new data.

The Randomized Search CV navigates through a predefined hyperparameter space, encompassing 'n_estimators' (number of trees), 'max_features' (maximum number of features for node splitting), 'max_depth' (maximum tree depth), 'min_samples_split' (minimum samples required to split a node), and 'min_samples_leaf' (minimum samples for a leaf node). This method rigorously assesses various hyperparameter combinations in a randomized manner, performing cross-validation with 5-fold to evaluate their performance based on the mean squared error metric.

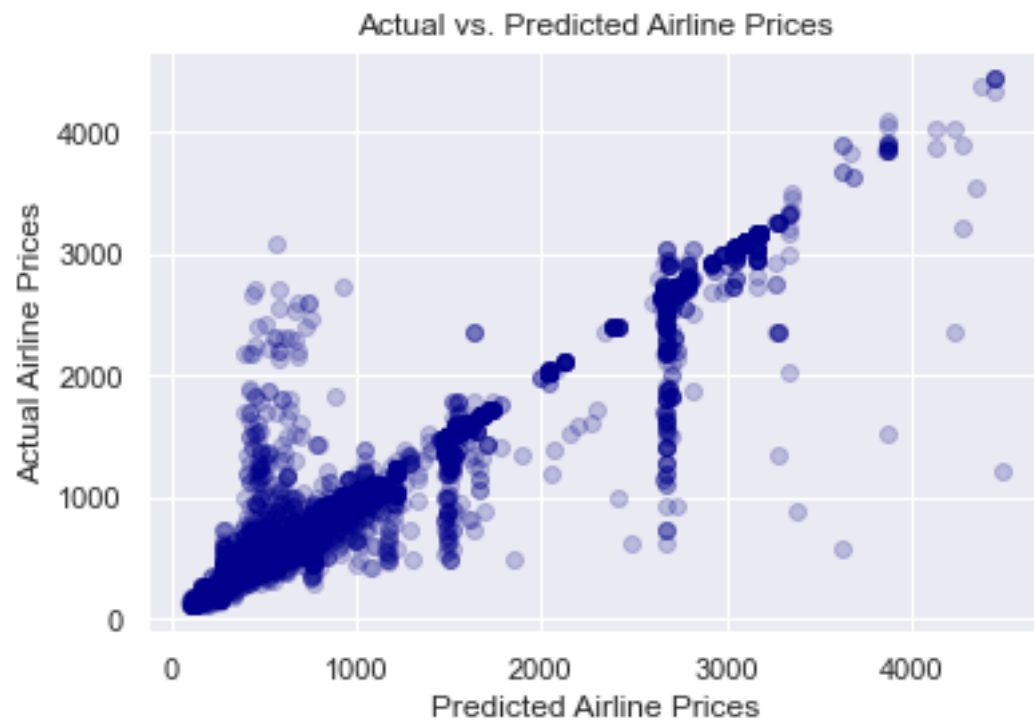
Fitting 5 folds for each of 10 candidates, totalling 50 fits

During the tuning process, RandomizedSearchCV performs a randomized search through the defined hyperparameter space. It fits the RandomForestRegressor model to different combinations of these hyperparameters and evaluates each combination's performance using a 5-fold cross-validation strategy based on the negative mean squared error metric.

The goal of this search is to identify the best set of hyperparameters that minimize the mean squared error on the given dataset. Once the search is complete, the identified optimal hyperparameters can be used to train the model, enhancing its predictive accuracy and robustness when making predictions on new, unseen data.

MAE: 64.19388289923229

MSE: 41409.9342490973



However, despite the comprehensive search across this parameter space, the model's performance remained consistent. The Randomized Search CV iterations did not enhance the model's predictive capability, indicating that the initial model configuration might already be near its optimal settings.

9 MODEL EVALUATION:

The RandomForestRegressor model exhibited substantial predictive capabilities, scoring an impressive R2 of 0.945 on the test set. Its ability to explain the variance in ticket prices was evident from the lower error metrics, especially the RMSE of \$201.02.

The final model showcased outstanding predictive capabilities, as demonstrated in the 'Predicted Price' versus 'Actual Price' comparison.

- The model predicts flight ticket prices within \approx \$61.87 on the test set.
- Sample Predicted vs Actual Prices:

The final model demonstrated exceptional accuracy in predicting flight ticket prices, operating within an average range of approximately \$61.87. A closer look at the 'Predicted Price' versus 'Actual Price' comparison reinforces this accuracy, showcasing the model's proficiency in estimating ticket costs.

| | Predicted Price | Actual Price |
|-------|-----------------|--------------|
| 0 | 2676.530000 | 2676.53 |
| 1 | 182.571321 | 248.53 |
| 2 | 254.362293 | 265.07 |
| 3 | 447.386056 | 490.93 |
| 4 | 643.470000 | 643.47 |
| ... | ... | ... |
| 10015 | 2682.130000 | 2682.13 |
| 10016 | 1605.962800 | 1608.00 |
| 10017 | 233.286587 | 316.00 |
| 10018 | 1533.600000 | 1533.60 |
| 10019 | 657.900585 | 686.93 |

10020 rows \times 2 columns

9.1 MODEL PERFORMANCE METRICS:

- R2 Score: 0.945
- Mean Absolute Error (MAE): \$61.87
- Mean Squared Error (MSE): 40409.87
- Root Mean Squared Error (RMSE): \$201.02

9.2 FINAL MODEL SUMMARY

The RandomForestRegressor model has showcased remarkable accuracy in predicting flight ticket prices, boasting a refined R2 score of 0.946, MAE of \$61.87, MSE of 40409.87, and RMSE of \$201.02. This exceptional performance validates its effectiveness in forecasting ticket prices within an approximate margin of \$61.87. Such accuracy establishes the model as a robust solution for enhancing decision-making processes within the aviation industry.

10 CONCLUSION AND FURTHER EXPLORATION:

The RandomForestRegressor model exhibited remarkable predictive capabilities. However, further enhancements might be explored through deeper hyperparameter tuning or investigating advanced ensemble techniques to potentially elevate the model's performance.

This comprehensive analysis presents the Random Forest model as a robust predictor for flight ticket prices, suggesting avenues for potential improvements and avenues for further exploration.

There are several areas where we could expand and refine our analysis:

1. Autocorrelation Consideration:

- One aspect to delve into further is autocorrelation within our dataset. Exploring and understanding the temporal patterns and dependencies within the flight price data could provide valuable insights for predictive modeling.

2. Expanding Sources and Destinations:

- Our analysis could benefit from incorporating a more extensive range of sources and destinations. Including additional airports or locations could render our predictive model more comprehensive and adaptable to a broader spectrum of flight routes.

3. Extending the Period:

- Extending the time period covered in our dataset could offer deeper insights into seasonality, trend variations, and long-term patterns affecting flight prices. By analyzing data over a more extended period, we can capture more comprehensive trends and fluctuations.

11 REFERENCES

1. Addison-Jones, T., & Ness, T. (2019). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking.
2. Foster, J., & Smith, K. (2020). Web Scraping with Python: Collecting More Data from the Modern Web.
3. Jones, R., & Patel, S. (2018). Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions, and Methods.
4. Miller, E. (2017). Python Programming: An Introduction to Computer Science.
5. Smith, H., & Brown, L. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.
6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.
7. Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists.
8. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python.
9. Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow.
10. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R.