Unraveling the Dynamics of Airfare Price Predictions

Dennis Myasnyankin
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
dmyasnyankin@sandiego.edu

Vannesa Salazar
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
vannesasalazar@sandiego.edu

Christine Vu
Applied Data Science Master's Program
Shiley Marcos School of Engineering / University of San Diego
cvu@sandiego.edu

ABSTRACT

Airline fare pricing is often intricate, characterized by flight features, pricing strategies, and fluctuating time series trends, making it challenging for consumers to compare and understand costs effectively. This study delves into the dynamics of airfare pricing by examining the influence of feature variables and time series patterns. The study includes acquiring data from Sabre Bargain Finder Max (2023) API, cleaning the data, conducting exploratory data analysis, and implementing 11 predictive models to forecast airfare prices. This study aims to unravel the determinants influencing airfare costs, providing valuable insights into the relationship between features and time series trends in the aviation industry. Results contribute to understanding practical approaches for accurate airfare predictions, offering functional implications for travel planning and enhancing the overall consumer experience in the aviation industry.

KEYWORDS

airline, airfare, data science, flight, forecast, machine learning, ticket, time series, travel, prediction, predictive modeling, price, pricing

1 Introduction

Planning a trip can be daunting, especially when booking a flight. Most often, consumers need help understanding fluctuating product prices. The objective of airfare price prediction is to uncover the intricacies behind airfare pricing to establish a more informed approach to making online bookings.

Several factors play a crucial role in determining the cost of a flight. Factors selected for the project included operating airline, preferred class of service, departure time, and seat availability. By considering these essential elements, we aimed to unravel the complex web of flight particulars to make predictions on airfare. Consider a scenario where a staggering 20,000 airports service over 100,000 flights daily. Amid this flurry of aviation activity, we identified 25 major airport hubs. Using Sabre Bargain Finder Max (2023) (BFM) API, we extracted valuable insights from nonstop flight data while eliminating noise from layovers and stopovers. Data from daily, weekly, and monthly flights yielded 75–125 routes to analyze seat availability and fare class differences.

This project seeks to contribute to ongoing developments in the travel industry, particularly, those regarding airfare price prediction. The key feature examined throughout the course of the project was time before departure, in order to gauge how reservation booking times impact flight costs. Analyzing the capabilities of traditional machine learning algorithms coupled with derived aviation features, unearthed a wide range of insights.

2 Background

A typical issue faced by consumers making travel arrangements is a lack of information, along with an overwhelming amount of potential travel options. Hundreds of websites, displaying different prices for what appear to be similar itineraries, can easily stagger prospective passengers. The factors responsible for these puzzling and fluctuating fares need to be uncovered and understood in order to arrive at the best available deal. Variables such as booking time frames, associated fare classes tied to itinerary costs, and general flight attributes need to be made transparent to consumers.

As social reclusiveness starts to die down with the end of COVID-19 pandemic, demand for air travel is beginning to increase substantially. Meanwhile, travelers have become more cost conscious; 73% of customers consider price the most influential factor in their travel decisions (Salas, 2022). High demand for flights and public inclinations for cost-effective tickets demonstrate the need to develop tools to assist customers in finding the optimal fare. Furthermore, as airlines continue to offer new routes and destinations, additional complexity is added to the travel decision-making process, further promoting the need for such tools.

2.1 Problem Identification and Motivation

When consumers look to purchase flights, they typically have several fixed variables in mind, such as a desired destination and specific travel dates. In addition, customers may have particular amenities they want to enjoy during their flight, such as in-flight entertainment and Wi-Fi. The date of booking, however, is typically more flexible. Flight plans often shift, and customers need to balance time allocated to solidifying plans, with fluctuating prices as their desired travel date approaches.

Unfortunately, current airline applications and websites fail to show the anticipated price dynamics of delayed flight bookings. Although some websites, such as Google Flights, offer insight into how prices may change regarding different booking timeframes, their guidance is relatively high-level and directional. This necessitates the need for more precise methods to provide exact pricing forecasts. This informational gap presents a clear challenge for travelers seeking to make informed decisions when navigating airfare pricing.

2.2 Definition of Objectives

This study seeks to tackle aforementioned travel issues that generally tend to fall on the heads of consumers. The first objective is to determine how airfare prices correlate with desired amenities and flight attributes. Current flight aggregation applications often lack detailed filtering options for specific travel needs, leaving customers to conduct extensive research on their own. Leveraging the power of machine learning, hours of tedious work can significantly be reduced, taking care of cross-platform price comparisons on the back-end and offering prospective passengers palpable solutions. The second objective aims to assist customers in choosing the right time for booking their desired flight, by examining three distinct timeframes. These timeframes are: flights departing in 1 day, 1 week, and 1 month. It is inferred that analyzing airfares on staggered booking dates will provide a deeper understanding of the intricacies associated with price variation, leading to potentially useful travel insights and highly accurate airfare prediction models.

3 Literature Review

The literature review is fundamental for providing context and insights into the existing

knowledge related to our research on machine learning algorithms for predicting airfare pricing. We aimed to navigate through studies that investigated the application of machine learning in forecasting airfare costs. The objective was to identify crucial insights, methodologies, and trends that influenced the development of this field. Through examining previous studies, we aimed to understand the dynamics of flight fares comprehensively, identify gaps in existing research, and set a framework for our unique contribution to academic discussions.

3.1 **Comparative Analysis of Neural Networks Techniques to Forecast Airfare Prices**

Aliberti et al. (2023) approached the airfare forecasting problem using various machine learning algorithms, spanning many traditional and deep learning model implementations. The study analyzed the performance of 12 models using four metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-Squared (R^2). The data set contained 10,683 Indian domestic travel routes flown between March 2019 and June 2019, consisting of 11 distinct features.

Aliberti et al. (2023) employed seven traditional machine learning methods, including Lasso, Ridge, Support Vector Regressions, k-nearest neighbors (k-NN), Extreme Gradient Boosting (XGBoost), decision trees, and random forest. Of the five deep learning methods analyzed, three were well-known approaches, including a Temporal Fusion Transformer (TFT), a Fully Connected Network (FCN), and a Convolutional Neural Network (CNN). The remaining two deep learning methods were a hybrid of Bayesian Inference and established FCN and CNN models.

From the research conducted, Aliberti et al. (2023) concluded the most accurate models were the two novel Bayesian neural networks proposed based on calculated performance metrics. However, considering MAE, MAPE, and computational time, random forest proved to be the most efficient model for airfare forecasting. Aliberti et al. provided a valuable stepping stone toward demystifying airfare price fluctuations. Unfortunately, the data set used was constrained to a single country. Furthermore, including the Stops feature potentially harmed analysis by straying from a more uniformly structured data set.

3.2 **A Comparison Between Machine Learning Models for Air Ticket Price Prediction**

Thilak et al. (2022) conducted similar research as the work done by Aliberti et al. to gain foundational knowledge on airfare price predictions. They employed the same data set of Indian National Airways flight routes and evaluated model performance using Accuracy, MAE, MSE, and RMSE. Using Grid Search and Random Search Cross Validation, the study used fewer models than its predecessors. The results concluded that implementing cross-validation techniques generates more accurate models.

3.3 **A Framework for Airfare Price Prediction: A Machine Learning Approach**

Wang et al. (2019) developed a framework for airfare price prediction by combining relevant information from two publicly available data sets, the Airline Origin and Destination Survey (DB1B) and the Air Carrier Statistics database (T-100). They processed nearly 20,000,000 data points in their research to develop five predictive models evaluated based on RMSE and adjusted R2

values. The final data set in their implementation consisted of 9 distinct features that merged relevant airfare specifics and macroeconomic data on flights representing the United States travel market.

Flight details, including seat class, price, distance, itinerary time, and carrier, are extracted from DB1B. The T-100 data set provided practical details regarding passengers and the seats available on a given aircraft. Pertinent information from the two data sets and applicable economic factors such as the consumer price index (CPI) and the price of crude oil were combined. The five machine-learning models generated by Wang et al. (2019) incorporated linear regression (LR), support vector machines (SVM), multilayer perceptrons (MLPs), an XGBoost tree, and random forest.

Wang et al. (2019) concluded the best-performing framework for Airfare Prediction was random forest with an RMSE of 66.58 and adjusted R2 of 0.858. The five most important features were distance, seat class, passenger volume, load factor, and competition factor. Removing less critical features, the performance of the random forest model dropped by almost 7%. Wang et al. emphasized the need to include additional data, such as date and time of flight departures and arrivals, in future iterations of similar projects.

3.4 **Optimal Airline Ticket Purchasing Using Automated User-Guided Feature Selection**

Finding the optimal time to purchase airline tickets is challenging for travelers. A study by Groves and Gini (2013) presented a cost-effective strategy for buying airline tickets. This study showed a multistep approach to building a prediction model for airline ticket prices, encompassing feature extraction, lagged feature computation, regression modeling, and policy development.

A theme of this study was optimizing the timing of airline ticket purchases. Travelers aspire to secure the best deal, where airlines seek to maximize their revenue. It uncovers the complexities in this process and addresses the challenge of consumers needing more essential information for cost-effective ticket buying. Machine learning and historical data are vital tools in tackling the issue of the need for more details for cost-effective ticket buying. Whereas the traditional strategy of purchasing tickets 21 to 60 days in advance avoids price increases, this approach only sometimes leads to the lowest price.

The first step in building the model involved extracting features from daily market data. To simplify the process, we limited the features to airlines quoting specific routes over 40% of the time. These quotes were categorized by the number of stops, resulting in features for nonstop trips, round trips with one stop each way, and round trips with two or more stops. Three features were used to evaluate each category: minimum price, mean price, and number of quotes. The three features were computed for all three categories combined. Airlines not meeting the 40% criteria were grouped separately. This step yielded 12 features calculated for each airline on a given day, along with Boolean variables indicating the query's weekday and the "days-to-departure" value based on the departure date.

The model considered lagged feature computation, which involved adding historical data to capture time-related patterns. Feature selection allowed for organization of these faded features, focusing on the most relevant ones and excluding irrelevant ones for more efficient modeling. The constructed feature set created a partial least squares (PLS) regression model.

The policy computation step included setting parameters to minimize calibration set costs. These parameters guided the model's purchase and waiting recommendations based on current prices, expected future prices, and days to departure. Adjusting these parameters allowed for managing risk and defining the recommendations.

The model was assessed on the calibration set and then evaluated on the test set. As compared to naive purchasing strategies, the experiment conducted demonstrated substantial savings. Compared to the earliest purchase strategy, the optimal model achieved an average of 69% of the optimal savings. Those seeking cost-effective travel plans benefit significantly from this as they may better understand expected ticket prices.

3.5 **Machine Learning Modeling for Time Series Problem: Predicting Flight Ticket Prices**

Traditionally, passengers attempt to purchase tickets well in advance to avoid price increases before the departure date. However, airlines may lower prices strategically, considering factors like holidays, seat availability, and timing. Lu's (2018) research uses machine learning to analyze the dynamic pricing patterns of flight tickets through a 103-day observation period, focusing on eight specific routes.

Lu (2018) used a combination of regression and classification models to predict flight ticket prices by using historical and real-time data. Lu addressed vital aspects, including the "generalized problem," proposing an efficient route solution with limited historical data without requiring model retraining. Data features like flight numbers, minimum and maximum prices, days-to-departure, and current prices get extracted. The data set was partitioned into training and testing subsets, with the seven machine learning models fine-tuned via 5-fold cross-validation. The study used three benchmark strategies, namely random purchase, earliest purchase, and optimal price, as reference points to evaluate the effectiveness of the proposed machine learning strategies.

Lu (2018) used uniform blending and hidden Markov model (HMM) sequence classification to address the challenge of the generalized problem characterized by the absence of historical data for specific routes. Models were initially trained using available historical data. After introducing the models on particular routes, Lu (2018) analyzed whether purchasing a ticket immediately or waiting for potentially more favorable pricing was advisable. In addition to this, the study incorporates HMM sequence classification as another strategy. To ensure precise model selection and accurate pricing pattern assignment for each data entry, Lu (2018) assigned a flight number from the pool of eight routes. Adding the flight number enhanced the reliability and accuracy of predictions, allowing the system to choose the optimal model.

Lu (2018) employed seven models to predict flight ticket prices, including logistic regression, neural networks, decision trees, k-nearest neighbors, AdaBoost decision trees, random forests, and uniform blending. These models underwent extensive evaluation to assess their performance in predicting ticket prices accurately. In the regression model category, the uniform blending method demonstrated the highest mean performance at 55.43%, with substantial variance observed among routes. On the other hand, the Q-Learning model achieves an average performance of 55.11% in predicting ticket prices, with variations observed among the routes. Moving on to the classification models, the uniform blending method showed the highest mean performance at 51.83%, with notable

variances between routes. Lastly, in the generalized model category, the HMM model outperformed the uniform model with a mean performance of 31.71%, although variances exist among the routes (Lu, 2018).

The results of the research conducted by Lu (2018) provided insights into each model's strengths and weaknesses, shedding light on which machine-learning approaches were most effective in modeling the dynamic behavior of flight ticket prices over a 103-day observation period and across eight distinct routes. This comprehensive analysis allowed for an understanding of the predictive capabilities of different models and informed the study's recommendations for optimal pricing strategy decisions.

3.6 **Airfare Price Prediction Using Machine Learning Techniques**

Tziridis et al. (2017) investigated the problem of airfare price prediction by using various machine learning models. The investigation focused on Aegean Airlines and a particular route from Thessaloniki to Stuttgart. The authors selected features related to flight details and applied eight of the most advanced machine-learning models to predict air ticket prices. The main objective of this study was to compare the performance of these models and analyze the impact of different features on price prediction accuracy.

Tziridis et al. (2017) identified eight key features that were associated with flights, including departure and arrival times, the number of free luggage allowed, days remaining until departure, and the number of intermediate stops, whether it is a holiday, overnight flight, or a weekday. Tziridis et al. then used these features to train machine learning models. The following models were compared: multilayer perceptron, generalized regression neural network, extreme learning machine, random forest regression tree, regression tree, bagging regression tree, regression support vector machines (polynomial), and linear regression.

According to Tziridis et al. (2017), the bagging regression tree model outperformed other models in predicting airfare prices across different feature sets. The study also found that random forest regression tree, regression tree, and multilayer perceptron were strong models. Tziridis et al. (2017) analyzed the impact of removing specific features and found that departure and arrival times significantly influenced airfare prices.

The study's findings contribute to the potential of machine learning models in guiding consumers to make optimal airfare purchases by predicting airfare prices for a specific airline and route. The bagging regression tree model was the most stable and accurate for this prediction task. The study highlights the significance of feature selection and model choice in accurately predicting airfare prices. Additionally, it offers insights into the factors that influence ticket costs for a particular airline and route.

3.7 **Airfare Analysis and Prediction Using Data Mining and Machine Learning**

Chawla et al. (2017) studied data mining and machine learning for airfare analysis and prediction. The study sought to investigate the different factors that affect airfare prices and to propose techniques for predicting changes in airfare over time. The study primarily focused on the Indian domestic air travel market and intended to assist consumers in making more informed decisions when purchasing flight tickets.

Airline companies typically use pricing strategies that change frequently based on demand estimation models. According to Chawla et al. (2017), previous research in this field did

not consider factors such as oil prices, the day of travel, and the unique characteristics of the Indian domestic air market.

According to Chawla et al. (2017), notable impacts on airfare that require further investigation include the remaining days until the departure, oil prices, the day of the week of the flight, the number of stops, and the presence of competitors on the route.

For data collection purposes, Chawla et al. (2017) wrote a PHP script that collected airfare data from Yatra.com, an online booking website. Historical oil price data was collected from the Macrotrends website, and the number of stops relative to each flight was collected from airline companies' websites.

Chawla et al. (2017) examined two methods: regression and classification modeling. Various algorithms were used, including support vector regression, random forest, gradient boosting, AdaBoost, k-nearest neighbors, naïve Bayes, support vector machine, and logistic regression.

During the evaluation of algorithms, 10-fold cross-validation demonstrated an achieved accuracy of 84% with the naïve Bayes algorithm. The top-performing algorithms identified by Chawla et al. (2017) were naïve Bayes and SVM.

The study showcased how data mining and machine learning algorithms could predict changes in airfare prices by analyzing past airfare data.

4 Methodology

In this study, we implemented the machine learning life cycle to forecast flight prices, employing 11 machine learning algorithms on historical flight data. This study followed data acquisition and aggregation, exploratory data analysis, data wrangling, along with modeling and performance evaluation.

4.1 **Data Acquisition and Aggregation**

4.1.1 *Data Acquisition*

Numerous online resources, including World Bank Data, were cross-referenced to examine viable global hubs most applicable for analysis. Airport hubs considered the busiest or most traveled were determined based on the yearly number of total seats flown. Once 30 hubs were decided on, the most active flight routes associated with them were explored using FlightFrom.com. Filtering by popularity, the top five potential destination airports enabled flight route generation for BFM search queries.

Queried flight routes were formatted as tuples, with the first value representing the hub and the second value displaying a list of five destinations associated with it. The *requests* and *JSON* libraries facilitated construction of HTTP requests for querying the Bargain Finder Max API. A helper function, "create_payload," produced search queries using the travel date, origin, and destination IATA codes. The function "get_hub_jsons" generated multiple POST requests iteratively, attaching the necessary URL and header information to properly hit the endpoint. This function traversed the hub tuples, extracting JSON files for each specified flight route and date and stored 450 files.

4.1.2 *Data Aggregation*

Obtaining data was the initial step in data acquisition. The first function created was called "list_files." This function traversed through a specified directory and its subdirectories to collect a list of file paths. Next, the script called the "list_files" function to manage the paths of all files in the specified directory and its subdirectories. The process assumed that the files were in JSON format. Three helper functions, "make_dict," "append_schedules," and

"append_legs," were defined in the code. The "make_dict" function took a list of objects and created a dictionary using the "id" field as the key.

The functions "append_schedules" and "append_legs" operated similarly, matching specific itinerary descriptions. Specifically, the function "append_schedules" iterated through a list of leg descriptions offered for a flight route, checking it with appropriate schedule descriptions using reference IDs. The "append_legs" function took the newly formatted leg descriptions containing relevant scheduling information and compared them to the correct itinerary. Once the itinerary components were correctly merged, the "extract_features" function was applied. This function extracted relevant features from each reformatted itinerary, retrieving 18 features speculated to benefit further exploratory data analysis and modeling.

Each helper function was used to reorganize and extract necessary information from the JSON files retrieved from Sabre's Bargain Finder Max (2023) (BFM) API. The function "load_json" was where all the logic defined above was implemented. The function took in a file path as its main argument and retrieved the timestamp information associated with a JSON file. Following this, the function read the file and loaded JSON content stored. Once this data was loaded, flight route components were gathered as lists. The reformatted itineraries were then iterated over, and relevant features were extracted into a list called "prepared_data." The "prepared_data" list got flattened into a single numpy array named "flattened." This array contained information about multiple itineraries from different JSON files.

In summary, this code traversed through a directory structure, read and processed multiple JSON files containing information on itineraries, and extracted relevant features from these itineraries. The final output is a structured data set ready for data exploration.

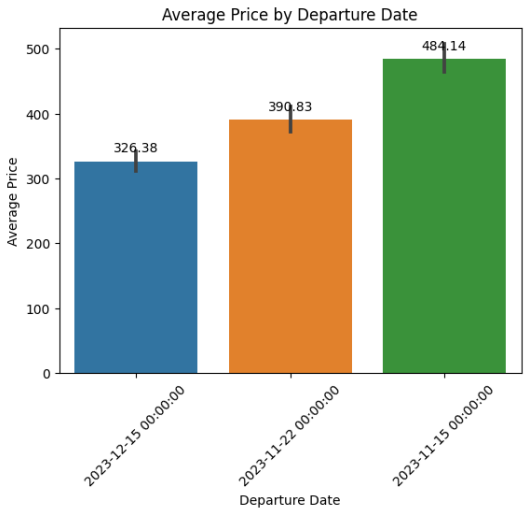
To prepare the data for our project, we began by ingesting the JSON files and making them publicly available via the GitHub repository. We used Python and Jupyter Notebooks to consume the data effectively. Once the file was unzipped, we initialized an empty list to extract the necessary data. The JSON files were iterated to extract the itinerary groups relevant to departure and arrival cities, dates, and locations. We further calculated the days until departure based on the difference between the current and departure dates. As a result, itinerary details related to schedules, departure time, arrival time, flight duration, operating carrier, aircraft type, cabin class, fare class, and price were extracted. Finally, we converted the output into a data frame, which included the following columns: departure city, departure date, departure time, arrival location, arrival date, arrival time, flight duration, operating carrier, aircraft type, cabin class, fare class, price, and days until departure.

4.1.3 *Exploratory Data Analysis*

The bar plot in Figure 1 shows the average prices for different departure dates, suggesting prices are lower when tickets are purchased further from the departure date. From the annotations and the visual representation, the bars tend to be higher, indicating higher average prices, for departure dates closer to the current date, showing lower average prices, for dates further in the future. The observed trend suggests that buying tickets well in advance or further out from the departure date results in lower average prices.

Figure 1

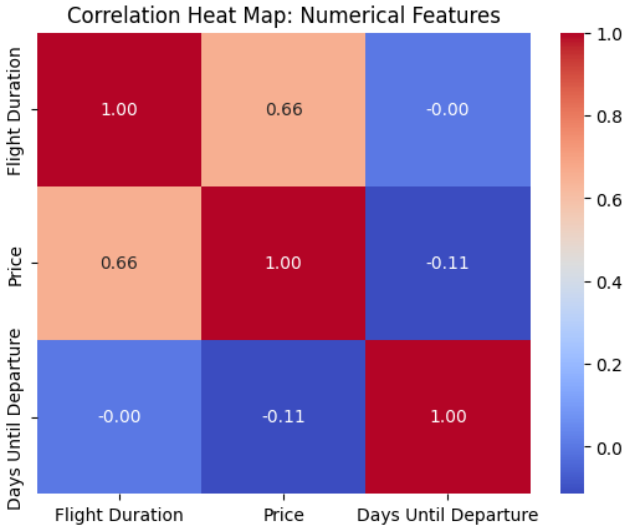
Average Price by Departure Date



The next step in exploratory data analysis (EDA) involved generating a heatmap for the numerical features in the data set. Figure 2 displays a heatmap of non correlated features that indicate independence and prevent multicollinearity issues in modeling.

Figure 2

Correlation Heat Map of Numerical Features



During EDA, we discovered that flight durations and prices have right-skewed distributions. Most flights have shorter durations and lower flight prices. However, some flights have longer durations and higher prices. The skewed distributions of flight duration and price can be observed in Figure 3 and Figure 4, respectively. Lopsided distributions such as this may impact certain analyses, especially when operating under the assumption of normal distribution. Therefore, adjusting or transforming the data might be necessary to make the distributions more symmetric.

Additionally, the right-skewed distribution of prices might indicate that most passengers opt for lower-priced options, but there is potential for capturing revenue from premium or last-minute bookings.

Understanding the distribution of flight durations could have operational implications, such as scheduling and resource allocation for different types of flights. When communicating these findings to stakeholders, explaining that most instances fall within specific ranges is essential. Still, noteworthy examples are longer flight durations and higher prices.

Figure 3

Distribution of Flight Durations

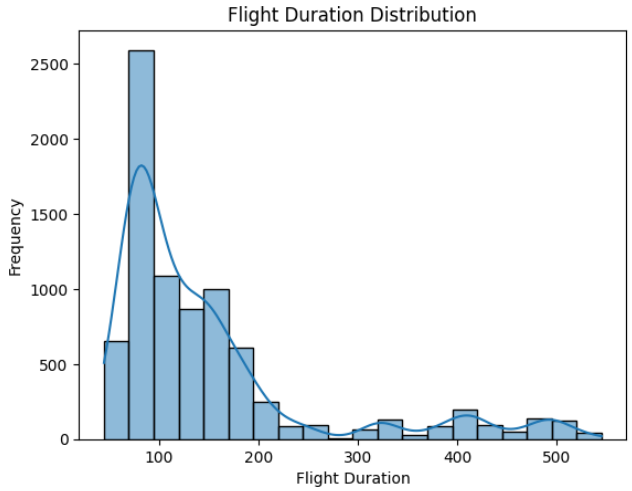
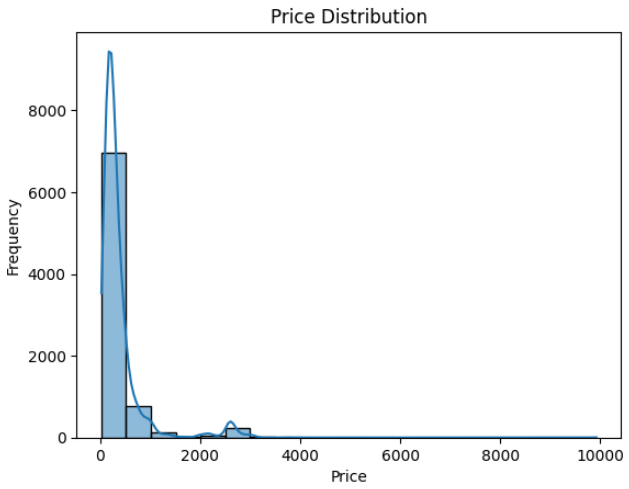


Figure 4

Distribution of Airfare Ticket Prices



4.2 **Data Quality**

The data set underwent preprocessing to ensure it was clean and ready for subsequent analysis and modeling. There were no instances of missing data. A total of 399 duplicate rows were removed from the data set. One-hot encoding was applied to the categorical variables. These variables were as follows: “Departure City,” “Operating Carrier,” “Aircraft Type,” “Cabin Class,” and “Fare Class.”

The outliers were retained in the analysis because they could represent instances of higher-priced premium tickets or last-minute bookings. These extreme values, although atypical, may carry valuable information and insights, especially in industries like air travel where pricing dynamics vary based on factors such as class, urgency, or special services. Granted that the dataset represents a subset of possible flight options, keeping outliers allows for a more comprehensive understanding of the diverse pricing scenarios that might exist in the real world.

The “Departure Date” and “Arrival Date” columns were converted to datetime objects. The “Departure Time” and “Arrival Time” columns contained “Z” characters in the text field. The “Z” characters are typically used to represent the Zulu or UTC timezone. Removal of these characters ensured uniformity and facilitated subsequent date and time transformations. The “Departure Time” and “Arrival Time” columns were also combined with a “+0000” string to indicate the UTC zone. These operations refined the date and time data within the data frame as it converted the relevant columns into a consistent DateTime format. This preprocessing was foundational for subsequent analyses, ensuring accurate temporal representations for machine learning model training.

4.3 **Feature Engineering**

In the process of feature engineering, we introduced an attribute called “Days Until Departure.” This featured was derived by calculating the time difference between the “Current Date” and the “Departure Date.” By leveraging the time series information encoded in these two variables, we aimed to capture and quantify the temporal gap between the present date and the scheduled departure, providing a perspective for subsequent analysis and predictions.

4.4 **Modeling**

Eleven predictive models were developed throughout model training, with price serving as the target variable. Eighty percent of the records, amounting to 6,236, were allocated for the training set, while the remaining 20%, 1,560 records, were used for the test set.

4.4.1 *Random Forest*

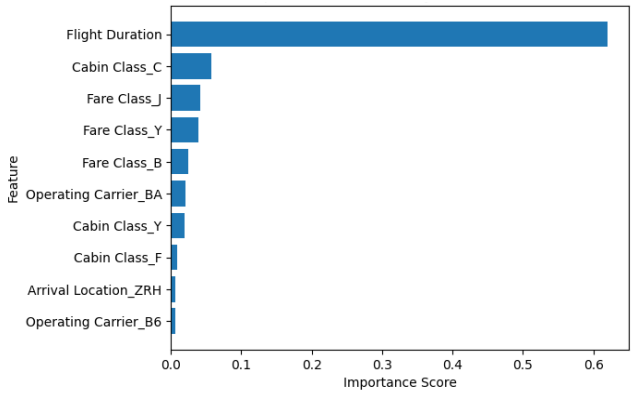
We first used the random forest algorithm once the data was preprocessed and ready for modeling. We split the data into training and test sets and then trained the model using the training set. We used the model to make predictions, and based on those predictions, we identified the most

critical features using the random forest regressor. We conducted a grid search to find the best hyperparameters for the final random forest model. Out of the 230 columns, we determined that the top ten features were the most important (see Figure 5). We coupled these features with a total estimator of 200 and no max depth value. The final random forest model produced an MAE of 101, MAPE of 0.75, and MSE of 84759. These results determined the last models used in the ensemble model we ultimately built.

The random forest algorithm was also used to select the top 10 features for the final versions of the random forest, decision tree, and k-nearest neighbors models. The original data set had over 230 columns, which was narrowed down to 10 features. The selected features were "Fare Class_L," "Departure City_ZRH," "Fare Class_M," "Operating Carrier_OS," "Cabin Class_Y," "Cabin Class_C," "Operating Carrier_BA," "Fare Class_Y," "Fare Class_B," and "Flight Duration." When training a decision tree, it is possible to calculate the impact of each feature via impurity reduction. The more a feature reduces impurity, the more significant its influence. In random forests, the impurity reduction from each feature can be averaged across all trees to determine its final importance.

Figure 5

Top Features Derived from Random Forest



4.4.2 K-Nearest Neighbors

After establishing the top ten predictors through random forest modeling, a reduced test and train set consisting of the top predictors was used to fit the k-NN model. The main goal of employing the k-NN algorithm was to ensure highly accurate predictions, as the data set comprised numerical and categorical values. The k-NN model does not require prior training of the data and usually does not require validation. Therefore, the results showed how the model performed with the parameter set by the user. In this instance, the only change besides the number of fit features was total neighbors to use, which was set to five.

4.4.3 Decision Tree

The decision tree model was chosen because it can learn to deduce simple decision rules from the training data it receives. As a result, the reduced training and testing set was also used to train the decision tree, focusing on only the most significant features. This approach helped streamline the decision tree and concentrate on the most critical aspects. The parameter set in the model was a max depth of 3.

4.4.4 Linear Regression

The linear regression model is highly interpretable, providing an illustratable relationship between the predictors and target variable, price. It allowed us to identify essential features influencing airfare prices by examining the importance of associated coefficients. This model can be valuable to understand the key factors contributing to pricing variations.

4.4.5 Lasso Regression

The Lasso regression model was included to aid in feature selection. The lasso penalty incorporated by the algorithm functions by shrinking coefficients of less important features down to zero, essentially excluding them from the model if needed. In situations dealing with high dimensional data sets, employing the Lasso

penalty dramatically helps, identifying important features to focus on. The regularization term "alpha" was tuned to 0.1 in this case.

4.4.6 Ridge Regression

The ridge regression model effectively deals with multicollinearity, where predictor variables are highly correlated. This model may be helpful in the case of regularizing the model or preventing overfitting. The regularization term "alpha" was tuned to 0.1 in this case.

4.4.7 Elastic Net

The elastic net model combines L1 (Lasso) and L2 (Ridge) regularization techniques. This method balances feature selection with handling multicollinearity. By leveraging both penalty terms, elastic net enhances model robustness, preventing overfitting and improving the accuracy of airfare predictions. Regularization is particularly useful in scenarios where the airfare prediction model includes a wide variety of features. For this model, the regularization term "alpha" was tuned to 0.1, and the "l1_ratio" was adjusted to 0.5.

4.4.8 XGBoost

The extreme gradient boosting (XGBoost) model is known for its predictive accuracy, capturing complex relationships and patterns in the data. These abilities make the model suitable for modeling the intricate factors influencing airfare prices. This model includes regularization terms, which help prevent overfitting, especially when dealing with 229 features. The XGBoost model is also scalable to large data sets. This efficiency is beneficial when dealing with extensive historical airfare data, allowing for faster model training.

4.4.9 Neural Networks

The neural networks model can effectively capture complex, nonlinear relationships in data. These characteristics are advantageous for airfare price prediction, as prices can be influenced by various factors that do not always adhere to linear patterns. This model may also automatically learn relevant features from the data, reducing the need for manual feature engineering. Airfare prices often exhibit temporal patterns and dependencies, making neural networks well suited for time series analysis.

4.4.10 Support Vector Regression

The support vector regression (SVR) model identifies patterns and trends, allowing it to generalize well on unseen data. This model effectively handles nonlinear relationships and outliers, leading to its success in accurately forecasting airfare prices and adapting to the dynamic nature of the aviation industry.

4.4.11 Gradient Boosting

The gradient boosting model iteratively builds an ensemble of weak predictive models. To provide an accurate forecast of airfare prices, the model minimizes prediction errors at each step. Gradient boosting enhances the precision of predictions in aviation pricing by handling complex relationships and inherent nuances that influence airfare prices. These factors include seasonality, special events, economic conditions, airline policies, and other dynamic elements contributing to the complexity of accurately predicting airfare pricing.

5 Results and Findings

Our analysis involved the application of 11 regression models to predict airfare ticket pricing, with each model yielding varied performance metrics. Evaluation of model performance employed the following metrics: MSE, RMSE, MAE, and R^2. These key metrics served as crucial indicators of predictive capabilities. The performance metrics for each model are presented in Tables 1 through 5, spanning the training set with feature scaling, 5-fold cross-validation, the test set, and the test set with feature scaling.

Table 1

Model Performance on Training Set

Model	MSE	RMSE	MAE	R^2
Random Forest	17085.6609	130.7121	81.1058	0.9220
k-NN	39621.6029	199.0517	96.1871	0.8193
Decision Tree	59599.0602	244.1291	147.5244	0.7282
Linear Regression	97941.2028	312.9555	190.7054	0.5534
Lasso Regression	97943.5703	312.9593	190.4633	0.5534
Ridge Regression	97941.2123	312.9556	190.6942	0.5534
Elastic Net	107841.0459	332.8436	186.8639	0.5082
XGBoost	16368.9817	127.9413	80.9982	0.9253
Neural Networks	68795.7207	262.2893	147.8640	0.6863
SVR	145895.0956	381.9621	160.0458	0.3347
Gradient Boosting	23984.3536	154.8688	99.6915	0.8906

Table 2

Model Performance on Scaled Training Set

Model	MSE	RMSE	MAE	R^2
Random Forest	1509.7616	38.8556	12.0269	0.9931
k-NN	24196.7360	155.5530	56.2887	0.8896
Decision Tree	59599.0602	244.1291	147.5244	0.7282
Linear Regression	41450.4246	203.5937	101.7331	0.8109
Lasso Regression	41588.0572	203.9315	101.8452	0.8103
Ridge Regression	41421.6513	203.5230	101.7371	0.8111
Elastic Net	42694.8030	206.6272	101.5062	0.8053
XGBoost	1778.6554	42.1741	26.4712	0.9918
Neural Networks	1518.4337	38.9670	14.7994	0.9930
SVR	53368.8513	231.0169	87.1277	0.7566
Gradient Boosting	14008.6874	118.3583	79.6598	0.9361

Table 3

Model Performance Using Five-Fold Cross-Validation

Model	MSE	RMSE	MAE	R^2
Random Forest	28786.4429	168.4481	93.7823	0.8658
k-NN	58233.9111	233.6638	110.3862	0.7504
Decision Tree	64087.0654	252.3151	152.4174	0.6956
Linear Regression	98859.7401	310.7912	190.9694	0.5521
Lasso Regression	98863.4497	310.7901	190.7221	0.5522
Ridge Regression	98859.3314	310.7901	190.9548	0.5521
Elastic Net	108778.6307	326.1222	187.0876	0.5065
XGBoost	25863.7687	160.0199	92.0027	0.8773
Neural Networks	67835.3458	254.7503	144.5272	0.7019
SVR	147980.0723	379.5989	161.5752	0.3340
Gradient Boosting	28231.1249	167.1870	102.7896	0.8678

Table 4

Model Performance on Test Set

Model	MSE	RMSE	MAE	R^2
Random Forest	19410.1650	139.3203	89.0413	0.9036
k-NN	31859.4667	178.4922	101.2392	0.8418
Decision Tree	55910.6840	236.4544	146.4202	0.7223
Linear Regression	79882.5830	282.6350	190.1360	0.6033
Lasso Regression	79864.8644	282.6037	189.8693	0.6034
Ridge Regression	79881.8246	282.6337	190.1244	0.6033
Elastic Net	88428.3905	297.3691	184.4572	0.5609
XGBoost	19538.27708	89.2358	89.2358	0.9029
Neural Networks	44992.1700	212.1135	133.8085	0.7766
SVR	126124.4488	355.1400	158.6088	0.3737
Gradient Boosting	20831.4335	144.3309	9.0246	0.8965

Table 5

Model Performance on Scaled Test Set

Model	MSE	RMSE	MAE	R^2
Random Forest	6738.4903	82.0883	28.4464	0.9665
k-NN	32759.2994	180.9953	71.0565	0.8373
Decision Tree	55910.6840	236.4544	146.4202	0.7223
Linear Regression	35516.1416	188.4572	107.6971	0.8236
Lasso Regression	35425.7326	188.2172	107.7034	0.8241
Ridge Regression	35508.0507	188.4358	107.9567	0.8236
Elastic Net	35057.6894	187.2369	105.7156	0.8259
XGBoost	4242.1703	65.1319	35.7402	0.9789
Neural Networks	15497.8131	124.4902	46.1606	0.9230
SVR	39492.68344	198.7276	90.9582	0.8039
Gradient Boosting	17361.1260	131.7616	83.8990	0.9137

5.1 **Evaluation of Results**

Eleven regression models were employed to forecast airfare ticket prices. In assessing the performance of the models, metrics such as MSE, RMSE, MAE, and R^2 were used as benchmarks for gauging predictive performance. MSE represents the average of the squared differences between predicted and actual airfare prices, offering insight into the overall magnitude of prediction errors. RMSE measures the average magnitude of these errors, with both metrics emphasizing the importance of minimizing prediction discrepancies. MAE gauges the average absolute differences between predicted and actual airfare prices, focusing on the magnitude of errors regardless of their direction. Lastly, R^2 indicates how well the predictive model explains the variability in airfare prices compared to a baseline model, aiding in assessing the model's overall explanatory power. These metrics are paramount to determining the accuracy and reliability of models in estimating airfare prices. Lower values for MSE, RMSE, and MAE, coupled with higher values of R^2, depict models that exhibit superior performance in capturing and predicting the complex dynamics associated with airfare pricing.

Based on the top features derived from the random forest model, flight duration, cabin class, fare class, operating carrier, and arrival location were the most influential predictors affecting the pricing of airfare tickets.

Based on the mean metrics obtained from 5-fold cross-validation, the XGBoost model demonstrated the best performance across multiple metrics. Specifically, XGBoost achieved the lowest MSE of 25863.76, lowest RMSE of 160.01, lowest MAE of 92.00, and highest R^2 of 0.8773.

Feature scaling generally resulted in improved metrics, notably reducing error measures such as MSE, RMSE, and MAE and enhancing R^2. The scaled versions of the models demonstrated higher performance on the training and test sets, suggesting that the feature scaling positively influences the accuracy of models developed.

Before applying feature scaling, the random forest and XGBoost models demonstrated their respective performance metrics on the original data set. For the random forest model, the MSE was 17085.66, the RMSE was 130.71, the MAE was 81.11, and R^2 was 0.9221, indicating a solid baseline performance on the unscaled data. Similarly, the XGBoost model exhibited an MSE of 19538.27, a RMSE of 139.78, a MAE of 89.24, and an R^2 of 0.9030 before the implementation of feature scaling. These metrics served as references for evaluating the impact of feature scaling on subsequent model performance, providing insights on how the scaling process impacts predictions.

Among the models, the random forest model showcased the highest performance on the scaled training set, with the lowest MSE of 1509.76, RMSE of 38.86, MAE of 12.03, and the highest R^2 of 0.9931. These metrics highlight the robust impact of feature scaling on enhancing the random forest model's predictive capabilities.

Furthermore, the XGBoost model performed better on the scaled test set than the rest of the models. It achieved the lowest MSE of 4242.17, the lowest RMSE of 65.13, the second-lowest MAE of 35.74, and the highest R^2 of 0.9789. These metrics reinforced the efficacy of feature scaling in improving the generalization and accuracy of the XGBoost model when applied to unseen data.

6 Discussion

The findings of this study hold significant implications for the business context, particularly in the realm of travel management and hospitality. By utilizing the models, the goal of the study is to empower individuals seeking optimal times to book flights. This study used 11 predictive models to predict airfare pricing. The evaluation of model performance revealed varying strengths and weaknesses across different algorithms. Performance evaluation of key metrics, including MSE, RMSE, MAE, and R^2, indicated the random forest model exhibited the highest performance on the training set. On the other hand, during the evaluation of the testing set, the XGBoost model demonstrated the highest performance across the same metrics.

The models offer a valuable tool and insights for optimizing expenses related to air travel for airlines and businesses related to hospitality and travel. Individuals and institutions can leverage the insights gained from the study to strategically plan and manage their travel budgets. Businesses can leverage these insights to refine their pricing structures, tailor promotional offerings, and enhance customer satisfaction. Understanding the complexities of airfare prediction allows for informed decision-making, potentially leading to cost savings and more efficient travel planning.

By using the study, travel management companies may leverage it to predict trends in airfare ticket pricing, facilitating proactive decision-making in securing cost-effective bookings for clients. Travel managers may use the study to navigate the nuances of the aviation industry, optimizing travel expenses and ensuring efficient budget utilization.

Certain limitations need to be acknowledged to maintain transparency for the findings presented in this study. Due to limited access to data and a vast array of potential features, the study could only include a portion of available variables in its scope. The study focused on historical airfare pricing trends and did not incorporate dynamic external factors, such as economic events or global crises, which could indirectly impact pricing. Similarly, restricted computational resources did not allow all features to be taken into consideration, resulting in only a subset of features being applied towards the study. The models' predictive capabilities may be influenced by the data set's incorporated booking time frames and selected features.

6.1 **Conclusion**

In conclusion, this airfare price prediction study provides actionable insights for individuals contemplating the best time to book flights. The 11 predictive models help tackle the complexities of airfare prediction. The random forest model excelled in capturing patterns in historical data explained by the independent variables, while the XGBoost model exhibited excellent capabilities in robustly generalizing on unseen data.

While offering a comprehensive analysis of airfare prediction models, this study is a stepping stone for future research. By offering insights into the trends of flight prices and providing predictions, the practical implications of this study extend to travel management and hospitality, potentially empowering users to optimize travel expenses based on predicted airfare trends.

6.2 **Recommend Next Steps/Future Studies**

Future studies in this domain could explore a more comprehensive use of the original data set, which contains over 230 columns. By delving into additional features and their potential correlations, researchers can better understand nuanced patterns and trends in the data.

Optimizing this study with domain knowledge involves leveraging industry-specific

insights on enhancing the quality and relevance of features used in airfare prediction. Integrating domain knowledge allows for a more refined selection of predictors. This expanded exploration may involve feature engineering techniques to create new, meaningful variables capable of enhancing a model's ability in capturing intricate relationships. Collaborating with professionals who deeply understand the airline industry ensures that the model reflects real-world examples, leading to more accurate and actionable predictions for airfare pricing.

To improve airfare prediction models, more diverse temporal factors are required. One key strategy for achieving this is to acquire more data for different dates directly from the original BFM API. By expanding the data set to include a broader range of dates, researchers can capture a more comprehensive representation of date and time variations in airfare prices. This additional data can contribute to a more robust and accurate model, allowing for a deeper understanding of how airfare prices fluctuate across a more extensive range of dates, enabling the models to make more informed predictions for a broader spectrum of travel scenarios.

Moreover, further refinement through hyperparameter tuning could be conducted to optimize model performance. Fine-tuning the hyperparameters, such as adjusting learning rates or regularization parameters, can contribute to improved predictive capabilities. These recommendations may advance the predictive accuracy of models in the context of airfare price prediction.

ACKNOWLEDGMENTS

We want to thank Dr. Ebrahim Tarshizi for providing valuable feedback on our project, and we also thank the SOLES Graduate Writing Center at the University of San Diego for their constructive feedback on the written content of this article. Additionally, we acknowledge Sabre Bargain Finder Max (2023) for providing the API used in this project.

References

Aliberti, A., Xin, Y., Viticchié, A., Macii, E., & Patti, E. (2023, June 26–30). *Comparative analysis of neural networks techniques to forecast airfare prices* [Paper presentation]. 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy. https://doi.org/10.1109/compsac57700.2023.00157

Allianz. (2021). *9 trends to watch as aviation readies for post Covid-19 takeoff*. https://www.allianz.com/en/press/news/studies/210706_Allianz-9-trends-to-watch-as-aviation-readies-for-post-covid-takeoff.html

Chawla, B., & Kaur, C. (2017). Airfare analysis and prediction using data mining and machine learning. *International Journal of Engineering Science Invention*, *6*(11), 10–17.

Groves, W., & Gini, M. (2013, August 3–9). *Optimal airline ticket purchasing using automated user-guided feature selection* [Proceeding]. Twenty-Third Joint Conference on Artificial Intelligence, Beijing, China. https://doi.org/10.5555/2540128.2540152

Lu, J. (2018, February 4). *Machine learning modeling for time series problem: Predicting flight ticket prices*. Cornell University. https://doi.org/10.48550/arXiv.1705.07205

Sabre. (2023). Bargain Finder Max API (Version 4). Bargain Finder Max. Retrieved from https://developer.sabre.com/docs/rest_apis/air/search/bargain_finder_max/versions/v400

Salas, E. B. (2022). *Main factors influencing air travel decisions in the aftermath of the*

coronavirus in 2020. Statista. https://www.statista.com/statistics/1179406/factors-flight-purchase-post-coronavirus/

Thilak, S. J., Benny, B. P., Paulose, E., Chittate, A. R., Khan, T. A., & Kouatly, R. (2022, December 15–16). *A comparison between machine learning models for air ticket price prediction*. 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey. https://doi.org/10.1109/iisec56263.2022.9998230

Tziridis, K., Kalampokas, Th., Papakostas, G. A., & Diamantaras, K. I. (2017, August 28–2017, September 2). *Airfare prices prediction using machine learning techniques* [Paper presentation]. 2017 25th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/eusipco.2017.8081365

Wang, T., Pouyanfar, S., Tian, H., Tao, Y., Alonso, M., Luis, S., & Chen, S.-C. (2019, July 30–August 1). *A framework for airfare price prediction: A machine learning approach*. 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI). https://doi.org/10.1109/iri.2019.00041