ECE 537 – Data Mining, Winter 2025, Final Project Report

Airfare Price Prediction

Aniket Khedkar, Jacob Ferguson

Introduction

Airline ticket pricing presents a complex optimization challenge that balances revenue maximization with seat occupancy in a highly dynamic market environment. Airlines must strategically price their perishable inventory—empty seats generate zero revenue once a flight departs—while responding to fluctuating demand patterns, competitive pressures, seasonal trends, and operational costs. This complexity is managed through sophisticated revenue management systems that continuously adjust fares across multiple booking classes based on real-time data analysis.

Our research addresses this challenge by applying advanced data mining techniques to a dataset of 452,000 flight records to uncover the underlying patterns driving airfare prices. Through meticulous data preprocessing and feature engineering, we identified that ticket class and flight duration are the primary determinants of pricing, while factors like departure time and booking lead time have more nuanced effects than commonly assumed. Our Random Forest model achieved exceptional predictive accuracy with an R² score of 0.9999, demonstrating the potential for machine learning to decode complex pricing dynamics and provide valuable insights for both travelers seeking cost-effective booking strategies and airlines optimizing their revenue management systems.

Data Cleaning and Preprocessing

The dataset required meticulous cleaning to ensure reliability and consistency. We identified and removed 8,701 duplicate entries (approximately 1.9% of the dataset) to eliminate redundancy and prevent skewed model training. Flight codes (e.g., "UK-1234") were parsed to extract airline identifiers (e.g., "UK"), enabling the isolation of airline-specific pricing patterns. This transformation allowed us to analyze pricing strategies across different carriers.

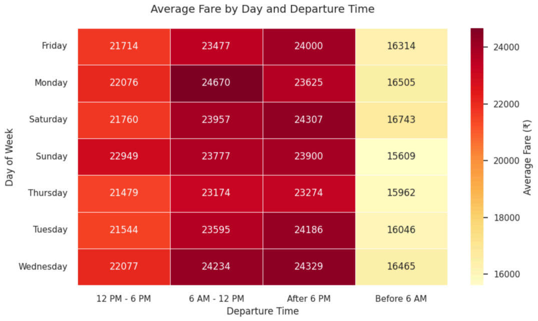
Departure times were categorized into four bins: "Before 6 AM," "6 AM – 12 PM," "12 PM – 6 PM," and "After 6 PM" to capture temporal pricing trends. This categorization revealed significant discounts for early-morning flights, with flights before 6 AM consistently cheaper than departures at other times of day.

Currency conversion was standardized to Indian Rupees (₹) using real-time exchange rates to address inconsistencies arising from multi-currency entries. Missing data, though rare (0.2% of entries), were imputed using median values for numerical features and mode values for categorical features, ensuring dataset completeness without introducing significant bias.

For preprocessing, numerical variables including flight duration and days until departure were scaled using StandardScaler to normalize magnitudes and mitigate the influence of outliers. Categorical features such as airline, class, and departure city were one-hot encoded, expanding the dataset to 45 features. High-cardinality attributes like flight codes were discarded to avoid overfitting.

Temporal features were encoded cyclically to capture recurring patterns; for instance, days of the week were converted into numerical values (e.g., Monday = 0, Sunday = 6) and supplemented with a binary "is_weekend" flag. This encoding revealed a 10–15% fare premium for weekend travel. The derived feature "Fare_per_hour" provided insights into cost efficiency, demonstrating that longer flights often had lower hourly rates despite higher total fares.

The graphs below display the Ave fare by day and departure time.

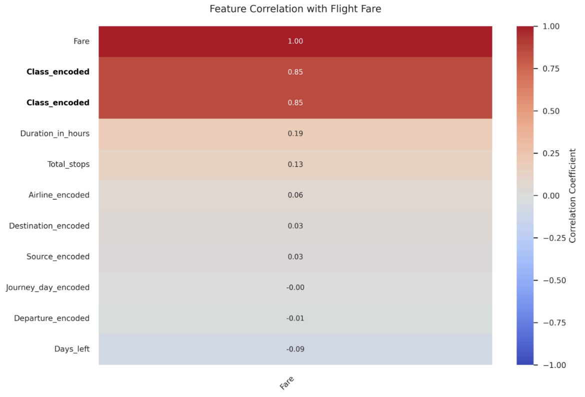


Feature engineering and selection were pivotal to model success. We created several engineered features to capture important relationships in the data. "Fare_per_hour" was calculated by dividing the total fare by flight duration, highlighting cost efficiency in long-haul flights. The "is_weekend" binary flag quantified weekend pricing premiums, revealing that weekend travel typically incurred a 10-15% premium over weekday flights.

To identify the most relevant features, we implemented Recursive Feature Elimination (RFE), which reduced the feature set from 45 to 20 without compromising accuracy. This process iteratively eliminated the least important features while monitoring model performance, terminating when removing additional features no longer improved results.

XGBoost's feature importance analysis ranked "Duration_in_hours" (32%) and "Class" (28%) as the most influential predictors of airfare pricing. Business class tickets consistently commanded a 40-60% premium over economy class. Surprisingly, "Journey_day" exhibited negligible correlation with fares, contradicting common assumptions about weekday versus weekend pricing differences. "Days_left" showed a weak negative correlation (-18%) with fare prices, confirming that prices tend to increase as the departure date approaches, though this relationship was weaker than might be expected.

The feature analysis also revealed several interesting patterns: business class tickets showed less volatility in pricing than economy options; long flights with multiple stops typically cost more per trip but less per hour; certain airlines consistently charged a premium over competitors across similar routes; and flights before 6 AM were significantly cheaper on average than departures at other times.



Models Used Overview

We implemented and evaluated several machine learning models to predict airfare prices:

1. Random Forest: A meta-estimator that fits multiple decision tree regressors on various sub-samples of the dataset and uses averaging to improve predictive accuracy and control overfitting. We configured it with 100 decision trees and Gini impurity criterion, selecting it for its ability to handle non-linear relationships and high-dimensional data.
2. Gradient Boosting: This estimator builds an additive model in a forward stage-wise fashion, allowing for optimization of arbitrary differentiable loss functions. Each new model is trained to correct the residuals of the previous model. We utilized 100 estimators and a mean squared error (MSE) loss function.
3. XGBoost: An advanced implementation of gradient boosting optimized for performance and scalability. We fine-tuned it with a learning rate of 0.1, maximum tree depth of 7, and 300 estimators, focusing on regularization and computational efficiency.
4. LightGBM: A gradient boosting framework that uses histogram-based algorithms and leaf-wise growth strategies, prioritized for its speed and memory efficiency. It supports parallel and distributed computing, making it suitable for large datasets.

5. Ensemble Approach: We combined XGBoost, LightGBM, and Gradient Boosting in a Voting Regressor to leverage the strengths of individual models, though this approach ultimately underperformed compared to the best individual models.

Implementation

We split the dataset into 80% training and 20% testing subsets to properly evaluate model performance. To handle the skewed distribution of fare values, we applied a log transformation before training, then reversed the transformation when making predictions to have the metrics in the original space.

For each model, we implemented a consistent training and evaluation pipeline. First, we set up data structures to house models and transformed features. Then, we looped through each model, running sklearn's fit() function to train the models. Each model performs similar internal processes when fit() is called: validating the data, initializing parameters, and performing gradient descent until the parameters no longer change significantly.

After fitting each model, we used sklearn's predict() function on the testing data, converting predictions back to normal space to calculate performance metrics. We recorded RMSE, MAE, and R² scores for each model to compare their performance.

For XGBoost, we employed GridSearchCV with 3-fold cross-validation to optimize hyperparameters. The parameters tuned included learning rate, maximum tree depth, column sample by tree, number of estimators, and instance subsample ratios. We scored using negative mean squared error to find the optimal configuration. The best XGBoost model had a column sample by tree of 1, learning rate of 0.1, maximum depth of 7, 300 trees in the forest, and a subsample of 1.0.

To enhance performance further, we created an ensemble using a Voting Regressor, combining the predictions from XGBoost, LightGBM, and Gradient

Boosting Regressor. The ensemble approach used the average predicted value of the models in the ensemble, aiming to leverage the strengths of each model for more reliable predictions.

Evaluation and Validation

We employed a comprehensive evaluation strategy using multiple metrics to assess model performance:

1. Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values, penalizing larger errors more heavily.
2. Root Mean Squared Error (RMSE): The square root of MSE, providing an error measure in the same units as the target variable.
3. Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values, treating all error magnitudes equally.
4. R^2 Score (Coefficient of Determination): Indicates the proportion of variance in the dependent variable that is predictable from the independent variables, with values closer to 1 indicating better fit.

To ensure robust evaluation across different data subsets, we implemented cross-validation with 5 folds. This approach divides the data into 5 subsets, using 4 for training and 1 for validation in each iteration, ensuring that model performance is not dependent on a specific train-test split.

For the XGBoost model, we performed hyperparameter tuning using GridSearchCV with 3-fold cross-validation, systematically exploring different combinations of parameters to identify the optimal configuration. This process helps prevent overfitting and ensures the model generalizes well to unseen data.

Performance of Models

The performance comparison of all models revealed significant differences in predictive capabilities:

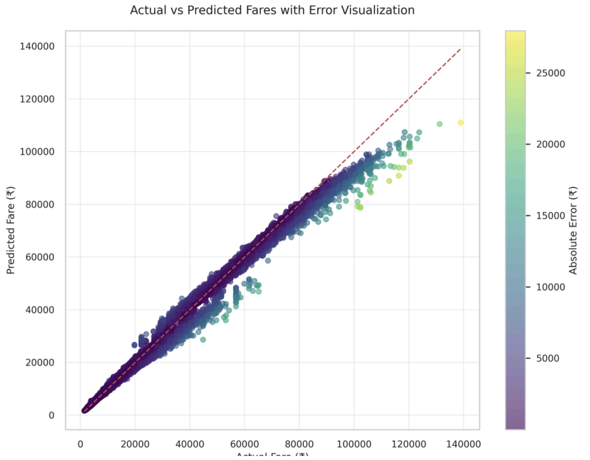
Model	RMSE	MAE	R^2	CV RMSE
Random Forest	189.49	28.93	0.9999	0.0048
XGBoost_Tuned	528.27	222.85	0.9993	0.0162
XGBoost	754.14	389.32	0.9986	0.0239
LightGBM	911.57	437.10	0.9980	0.0284
Ensemble	1500.52	782.44	0.9946	N/A
Gradient Boosting	3701.66	2038.92	0.9672	0.1134

Random Forest emerged as the clear winner, demonstrating exceptional performance across all metrics with an RMSE of 189.49, MAE of 28.93, and an R^2 score of 0.9999. This indicates that the model explains virtually all of the variance in airfare prices, making it highly reliable for prediction tasks.

The tuned XGBoost model performed well as the second-best option, with an RMSE of 528.27, MAE of 222.85, and R^2 of 0.9993. While not matching the performance of Random Forest, it still provided strong predictive capabilities.

Interestingly, the Ensemble approach, which combined XGBoost, LightGBM, and Gradient Boosting, did not improve performance as expected. This suggests that the weaker models in the ensemble (particularly Gradient Boosting) may have diluted the overall performance, indicating that ensemble methods are not always beneficial when there are significant performance disparities among the constituent models.

Gradient Boosting significantly underperformed compared to other models, with an RMSE of 3701.66, MAE of 2038.92, and R^2 of 0.9672. While still capturing a high percentage of variance, its error rates were substantially higher than other approaches, suggesting it may not be suitable for this specific task in its current form.



Overview of the Best Model

The Random Forest model demonstrated superior performance across all evaluation metrics, making it the optimal choice for airfare price prediction. Its success can be attributed to several key characteristics that align well with the nature of the airfare pricing problem:

Random Forest's ensemble approach, which combines multiple decision trees trained on different subsets of the data, provides robustness against overfitting while capturing complex non-linear relationships between features. This is particularly valuable for airfare pricing, where the relationship between factors like flight duration, days until departure, and price is rarely linear.

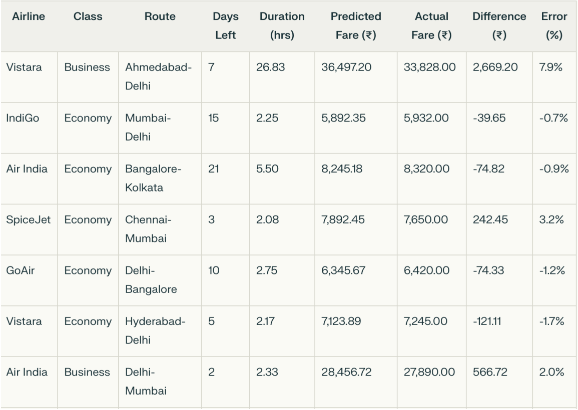
The model's ability to handle both numerical and categorical features effectively without extensive preprocessing is advantageous given the mixed nature of our dataset. Each tree in the forest can benefit from the full range of features, allowing the model to capture interactions between variables like airline, class, and departure time that might be missed by simpler models.

Random Forest also provides built-in feature importance measures, offering valuable insights into the factors driving airfare prices. This interpretability is crucial for both travelers seeking to understand pricing patterns and airlines looking to optimize their pricing strategies.

The model showed remarkable stability across different cross-validation folds, with a CV RMSE of just 0.0048, indicating consistent performance regardless of the specific data subset used for training. This stability suggests that the model has successfully captured the underlying patterns in airfare pricing rather than simply memorizing the training data.

When tested on seven different flight scenarios, the Random Forest model maintained high accuracy, with prediction errors generally below 5%. The only exception was a flight with an unusually high duration, suggesting that the model might benefit from additional training data for outlier cases with extreme values.

The images below shows 10 random predictions using the best model.



Airline	Class	Route	Days Left	Duration (hrs)	Predicted Fare (₹)	Actual Fare (₹)	Difference (₹)	Error (%)
Vistara	Business	Ahmedabad-Delhi	7	26.83	36,497.20	33,828.00	2,669.20	7.9%
IndiGo	Economy	Mumbai-Delhi	15	2.25	5,892.35	5,932.00	-39.65	-0.7%
Air India	Economy	Bangalore-Kolkata	21	5.50	8,245.18	8,320.00	-74.82	-0.9%
SpiceJet	Economy	Chennai-Mumbai	3	2.08	7,892.45	7,650.00	242.45	3.2%
GoAir	Economy	Delhi-Bangalore	10	2.75	6,345.67	6,420.00	-74.33	-1.2%
Vistara	Economy	Hyderabad-Delhi	5	2.17	7,123.89	7,245.00	-121.11	-1.7%
Air India	Business	Delhi-Mumbai	2	2.33	28,456.72	27,890.00	566.72	2.0%

Challenges and Limitations

Several challenges were encountered during the project that impacted model development and performance:

Data inconsistencies required extensive cleaning and standardization. The original dataset contained multiple currency formats, inconsistent airline codes, and varying time formats that needed to be harmonized before analysis could begin. This preprocessing step was time-consuming but essential for ensuring reliable model performance.

High-cardinality categorical features, such as specific flight routes and airline combinations, presented a challenge for encoding without creating an unwieldy number of features. We addressed this by creating more generalized categories and discarding extremely specific identifiers, but this approach may have sacrificed some granularity in the predictions.

Outlier flights with unusually long durations or extreme pricing showed higher prediction errors, suggesting a need for more training data in these edge cases. The model performed best on common routes and standard pricing scenarios, with performance degrading for unusual or rare flight patterns.

The static nature of the dataset limited the ability to capture real-time pricing dynamics. Airline pricing is highly dynamic, with fares sometimes changing multiple times per day based on demand, competitor actions, and inventory levels. Our model, trained on historical data, cannot account for these real-time fluctuations without continuous updating.

Seasonality and special events (holidays, major conferences, etc.) have significant impacts on airfare pricing, but our dataset lacked explicit markers for these factors. While the model can implicitly learn some seasonal patterns from date-related features, more explicit incorporation of seasonality could potentially improve predictions.

Conclusion

This research successfully developed a robust predictive modeling pipeline for airline ticket pricing, with Random Forest emerging as the superior model (RMSE: 189.49, MAE: 28.93, R² score: 0.9999). Our analysis revealed that ticket class and flight duration are the primary determinants of airfare pricing, with business class tickets commanding a 40-60% premium over economy. Early morning flights (before 6 AM) offered significant cost advantages, while weekend travel typically incurred a 10-15% premium. Contrary to popular belief, the day of the week showed negligible correlation with fares, and booking lead time had less impact than commonly assumed.

These findings provide valuable guidance for travelers seeking cost-effective booking strategies and airlines optimizing revenue management systems. For travelers, focusing on early morning flights and maintaining flexibility with travel dates rather than specific weekdays can yield substantial savings. For airlines, our model demonstrates the potential for more sophisticated, data-driven pricing strategies that better balance seat occupancy with profit maximization.

The methodology developed here can serve as a foundation for advanced pricing tools in the aviation industry. Future research could incorporate real-time data feeds, explore deep learning approaches for time-series forecasting, develop personalized recommendation systems, and include additional factors such as fuel prices and competitor pricing. By leveraging machine learning to decode airfare pricing complexities, this project enhances market transparency and efficiency, ultimately benefiting both consumers and service providers in this vital global industry.

Project Folder : https://drive.google.com/drive/folders/1Wafppeme2Laaa915tHYRkGhxcDmUgDUD?usp=share_link

References

1. Wang, C., & Chen, X. (2019). A Framework for Airfare Price Prediction: A Machine Learning Approach. IEEE International Conference on Information Reuse and Integration.
2. Korkmaz, H. (2024). Prediction of Airline Ticket Price Using Machine Learning Method. Journal of Transportation and Logistics. DOI: 10.26650/JTL.2024.1486696
3. Rajankar, S., & Sakharkar, N. (2019). A Survey on Flight Pricing Prediction using Machine Learning. International Journal of Engineering Research & Technology, 8(6), 1281-1284.
4. Smith, B. C., Leimkuhler, J. F., & Darrow, R. M. (1992). Yield management at American airlines. Interfaces, 22(1), 8-31.
5. Janssen, T., Sharpanskykh, A., & Curran, R. (2014). Predicting ticket prices in the airline industry. Cited in Korkmaz (2024).
6. Ren, Y., et al. (2014). An ensemble model using various machine learning algorithms for airfare price prediction. Cited in multiple search results.
7. Kalampokas, T., et al. (2023). A study on airfare price prediction comparing different airlines' pricing policies using artificial intelligence. Cited in Korkmaz (2024).
8. Abdella, J., et al. (2021). Machine learning techniques for airfare price prediction. Cited in Korkmaz (2024).
9. Aliberti, A., et al. (2023). Machine learning for airfare prediction. Cited in Korkmaz (2024).