FareSight: Leveraging Machine Learning to Forecast Flight Prices

Aarzoo (2022008)
Anushka Srivastava (2022086)
Nandini Jain (2022316)
Suhani Kalyani (2022511)

Final Project Report

Abstract

Flight prices can vary significantly, making it difficult for travelers to plan their budgets effectively. This paper presents the development of a machine learning-based predictive model to forecast flight prices. By utilizing a dataset containing various flight details, we converted different categorical features to numerical using different preprocessing techniques and removed the outliers to ensure reliability. Several regression models were implemented and evaluated based on metrics such as MSE, RMSE, R2 scores and MAE. Through comprehensive analysis, Random Forest and Bagging Regressor were found to give best prediction accuracy. To enhance robustness, Grid Search was performed.

Additionally, a command line interface was developed which enables users to input their travel details and identify the date for which they can get the minimum price. The interface displays the lowest predicted price, best airline and ideal booking date, offering assistance to travelers.

The code for this paper is available at Github link

1. Introduction

Flights have become the preferred choice for long distance travels. However, the cost of flight tickets varies a lot, increasing one day and dropping the very next. This leaves many travelers confused about the factors affecting these variations which is an important financial consideration.

To solve this problem, we are building a predictive model that can help predict the prices of flight tickets and help them identify the day with the most optimal price. This will help travelers make more informed decisions in their travel plans, and allow us to contribute to making air travel more accessible and affordable.

2. Literature Survey

2.1. Dynamic Flight Price Prediction Using Machine Learning Algorithms

The paper describes a machine learning model capable of predicting flight prices. The dataset contains flight prices of various airlines between March and June 2019 across different cities. Three machine learning models are implemented, including Linear Regression, Random Forest, and Decision Trees, achieving an accuracy of 80%. [1]

2.2. Flight Price Prediction for Enhanced Recommendations via Machine Learning Web Application

The paper describes a dependable machine learning model that can accurately estimate flight prices. Six different machine learning algorithms are thoroughly evaluated to ensure the model's accuracy and reliability. Performance metrics such as MSE, RMSE etc. are used for comparison. The Random Forest Regressor outperformed the others, achieving the highest R^2 score. [2]

2.3. A Prediction of Flight Fare Using K-Nearest Neighbors

The study identifies the factors driving airplane price fluctuations, how they influence price changes. The study employed the K-Nearest Neighbors algorithm to model the factors. The model was evaluated using various metrics. The results show that the K-Nearest Neighbors algorithm achieved a solid accuracy of 81.77%. [3]

3. Dataset

We have used the **Flight Prediction Dataset** from Kaggle [4] for this project. The dataset has 300,153 records and 11 columns. From the 11 columns, we have 10 features and 1 label. There are no NULL values in the dataset.

The features `duration` and `days_left` are numeric, while the other features are categorical features. The feature `price` is a continuous target value.

#	Column	Count	Null Values	Dtype
0	airline	300153	0	object
1	flight	300153	0	object
2	source_city	300153	0	object
3	departure_time	300153	0	object
4	stops	300153	0	object
5	arrival_time	300153	0	object
6	destination_city	300153	0	object
7	class	300153	0	object
8	duration	300153	0	float64
9	days_left	300153	0	int64
10	price	300153	0	float64

Table 1. Dataset Column Information

Feature Distribution

1. **Airlines:** There are 6 different airlines in our dataset - Vistara, Air India, Indigo, GO_FIRST, AirAsia, and

SpiceJet. SpiceJet has maximum count of flights (with 42.6% of the entire distribution) while Air_India has minimum count of flights (with 3% of the entire distribution).

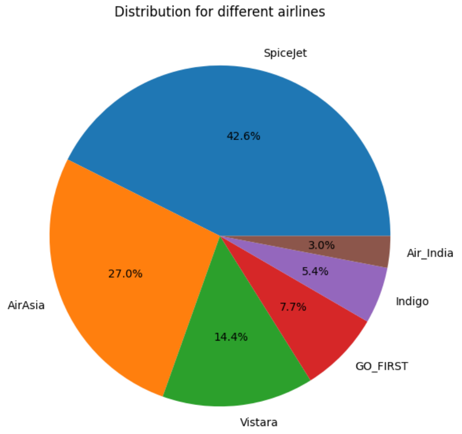


Figure 1. Pie Chart for Airline Distribution

2. **Class:** There are two types of classes - Economy Class and Business Class. The count of Economy class is 206,666 whereas that of Business Class is 93,487.

3. **Source City and Destination City:** The flights travel to and fro from the following cities: Delhi, Mumbai, Bangalore, Kolkata, Hyderabad, and Chennai. There are maximum departing flights from Delhi (61,343), while there are maximum arriving flights for Mumbai (59,097).

4. **Number of stops:** This feature is a discrete value. Most flights have one stop (250,863) followed by zero stops (36,004), and then two or more (1,286).

EDA Analysis

1.

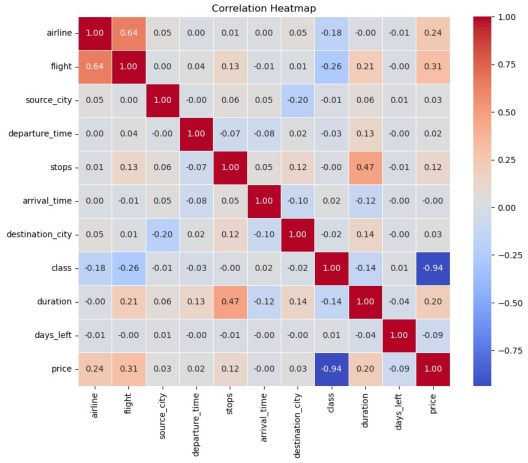


Figure 2. Observation:

The matrix shows that feature `class` and `price` are highly negatively correlated whereas features like `arrival_time`, `departure_time`, `source_city` and `destination_city` do not have significant impact on the flight prices.

2. The range of prices for Business class is significantly broader, with prices ranging from around 12,000 to above 120,000. Economy class has a much more compressed range, with prices mostly under 45,000.

3.

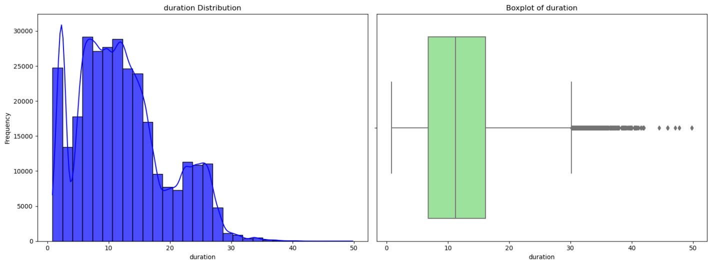


Figure 3. Duration for Economy and Business classes.

The histogram shows a positively skewed distribution of flight durations, with most values concentrated between 5 and 20 units. The boxplot reveals a clear presence of outliers beyond the upper whisker, representing unusually long flight durations. These outliers will be handled later.

4.

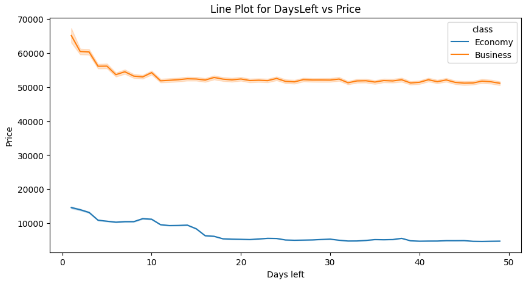


Figure 4. Flight prices vs. days left for departure

The price of both Economy and Business tickets rises significantly as the days left for departure decreases. Business class prices start at a much higher level and spike when the days left are less than 8 whereas Economy class prices start at a lower level and gradually increase as the days left falls below 20.

5.

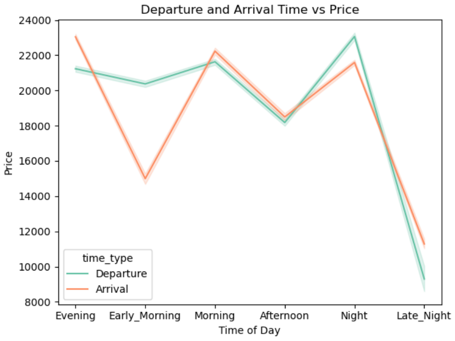


Figure 5. Flight prices based on times.

Flights which arrive in the evening, morning or night have higher prices than those arriving early morning or late night indicating they are more preferred. Similarly, flights which depart in late night are least preferred over others.

Preprocessing

8 out of 10 features are categorical and 2 are numerical.The numerical features are duration and days_left. Box plots and inter-quartile range were used to detect outliers, identifying 2110 outliers, which were dropped.
Next, in the categorical features, stops was mapped to its respective numerical values. Time features : arrival_time and departure_time were mapped to numerical values. Further, label encoding was performed on class and flight features , while airline, source_city and destination_city were one hot encoded to handle their categorical nature as they are nominal data.

Feature selection was performed to identify the most important predictors for the model. Using ExtraTreeRegressor and feature_importance, we checked for the relative importance of features. We then used PCA (Principal Component Analysis) to reduce dimensions of the features. Its results gave us the most optimal 7 features to predict the price.

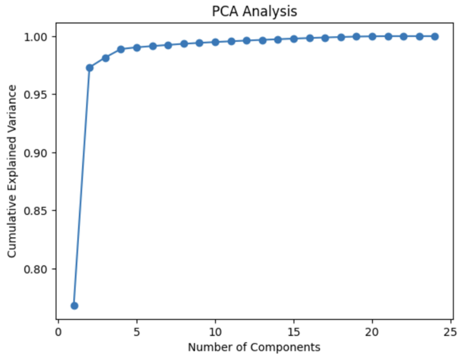


Figure 6. PCA

After final processing of the dataset, we have converted all features to numerical values and handled outliers, preparing the dataset for training and analysis by the models.

4. Methodology

We split the data into a 70:30 train-to-test ratio with 210,107 training samples and 90,046 testing samples.

We use regression models because the task involves predicting a continuous target variable - price. Regression models can capture both linear and non-linear relationships, between the target variable and features.

We train the models for both datasets - with and without feature selection to compare and understand which model works better.

Grid Search has been performed on our best performing model, which is Random Forest. It was performed for 5 folds for 16 candidates, giving a total of 80 fits.

Evaluation Metrics

1. **Mean Squared Error (MSE):** It penalizes the large errors more due to squaring .

2. **Root Mean Squared Error (RMSE):** It expresses the error in the same unit as the target variable, providing easier interpretation.

3. **R-Squared Score:** It helps us understand the proportion of variance in the dependent variable that can be explained by the independent variable.

4. **Adjusted R2 Score:** It adjusts for the inclusion of non-significant predictors, indicating whether adding more predictors enhances the model's performance or not.

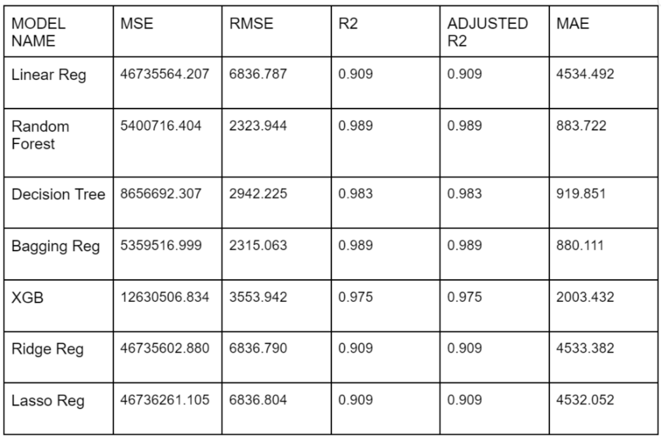
5. **Mean Absolute Error (MAE):** It does not penalize the outliers heavily and gives a balanced overview.

Models Used

We used a variety of models to capture linear and non-linear relationships in our dataset. The random seed is set to 42 if required in the model to ensure reproducibility of results .
We used **Linear Regression** to explore the basic linear relationship in our dataset. Here, we have used this model to observe the changes in the price of the flights (dependent variable) based on the various other features (independent variables) available in the dataset.
To prevent possible overfitting of models, we used **Lasso Regression** to introduce L1 regularization with a learning rate of 1 and balance the bias-variance tradeoff.
Ridge Regression was also implemented to prevent possible overfitting and rectify multicollinearity problems in regression analysis, we introduced L2 regularization with a learning rate of 1.
To capture more complex and non-linear relationship between the data, we used **Decision Tree**. It splits data based on the features of the dataset. However, it is prone to overfitting.
An ensemble model that uses Decision Trees as its base model and is resistant to overfitting is **Random Forest**. Hence, to enhance performance, we used Random Forest, which averages the result for 100 decision trees (estimators).
We also tried to improve performance by learning iteratively by correcting previous errors using **XGBoost Regression**. It uses 100 trees (estimators) with learning rate 0.1. The objective used is squared-error.
Another ensemble method used was **Bagging Regression** to improve accuracy as it reduces variance and prevents overfitting of the data. With decision trees as its base estimator, it uses 300 base estimators .

5. Results and Analysis

1. **Observation:**



MODEL NAME	MSE	RMSE	R2	ADJUSTED R2	MAE
Linear Reg	46735564.207	6836.787	0.909	0.909	4534.492
Random Forest	5400716.404	2323.944	0.989	0.989	883.722
Decision Tree	8656692.307	2942.225	0.983	0.983	919.851
Bagging Reg	5359516.999	2315.063	0.989	0.989	880.111
XGB	12630506.834	3553.942	0.975	0.975	2003.432
Ridge Reg	46735602.880	6836.790	0.909	0.909	4533.382
Lasso Reg	46736261.105	6836.804	0.909	0.909	4532.052

Figure 7. Testing errors without feature selection

- Without feature selection, Random Forest and Bagging Regression have the best performance with the lowest MSE, RMSE, and MAE values and the highest R² values (0.989). Decision Tree also performs well but is slightly less effective than Random Forest and Bagging. Linear, Ridge, and Lasso regressions show the worst performance, with much higher MSE and RMSE and significantly lower MAE compared to tree-based models.

MODEL NAME	MSE	RMSE	R2	ADJUSTED R2	MAE
Linear Reg	231935823.372	15229.439	0.551	0.551	12557.948
Random Forest	6867957.545	2620.678	0.986	0.986	1026.754
Decision Tree	11356508.598	3369.941	0.978	0.978	1115.286
Bagging Reg	6826426.636	2612.743	0.986	0.986	1022.045
XGB	14689042.174	3832.628	0.971	0.971	2230.332
Ridge Reg	231935813.169	15229.439	0.551	0.551	12557.980
Lasso Reg	231935800.438	15229.438	0.551	0.551	12558.164

Figure 8. Testing errors with feature selection

- With feature selection, the performance of regression models significantly deteriorates, demonstrated by high MSE, RMSE, and low R² values. Random Forest and Bagging Regression remain the best performers, maintaining high R² (0.986) with relatively low errors. Decision Tree and XGBoost perform slightly worse compared to Random Forest and Bagging but still significantly outperform linear models.

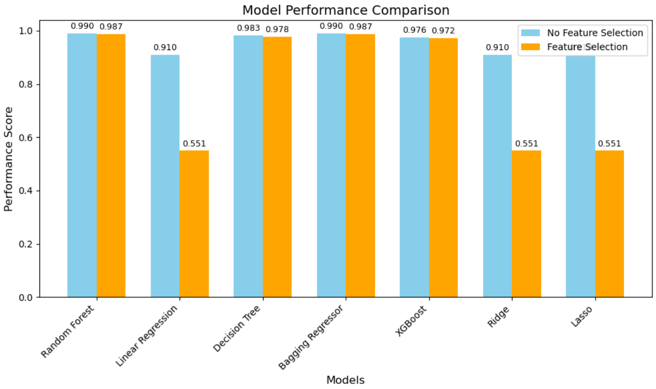
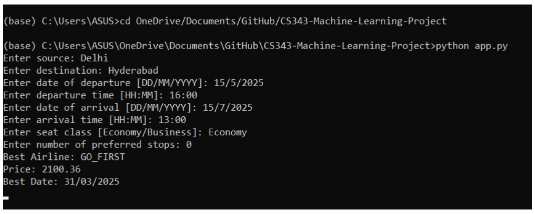


Figure 9. Comparison of model performance across different metrics

- Feature selection negatively impacted Linear, Ridge, and Lasso Regression performance drastically, leading to poor fit and high error values. Tree-based methods remain robust even after feature selection. This indicates that these models are better at handling a reduced feature set without significant performance loss.

- Grid Search performed on Random Forest also gave similar results with its Test R2 score being 0.981. Here, model was trained on 200 estimators with max_depth 20.

- Developed a Command Line Interface where the users can enter their travel details and get the date on which they book the ticket at minimum price.



```
(base) C:\Users\ASUS>cd OneDrive/Documents/GitHub/CS343-Machine-Learning-Project

(base) C:\Users\ASUS\OneDrive\Documents\GitHub\CS343-Machine-Learning-Project>python app.py
Enter source: Delhi
Enter destination: Hyderabad
Enter date of departure [DD/MM/YYYY]: 15/5/2025
Enter departure time [HH:MM]: 16:00
Enter date of arrival [DD/MM/YYYY]: 15/7/2025
Enter arrival time [HH:MM]: 13:00
Enter seat class [Economy/Business]: Economy
Enter number of preferred stops: 0
Best Airline: GO_FIRST
Price: 2100.36
Best Date: 31/03/2025
```

Figure 10. Example for the usage of CLI

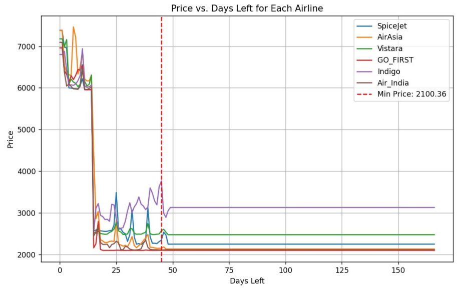


Figure 11. Graph for the corresponding example

6. Conclusion

We developed and evaluated a machine learning-based system for predicting flight prices to assist travelers in making cost-effective travel plans. We identified the most impactful features influencing flight prices by preprocessing a comprehensive dataset of flight details, addressing outliers, and performing PCA. Despite a lower correlation between other features and price, our analysis revealed that price is highly dependent on class. We trained our models on datasets with and without feature selection and observed similar loss and accuracy in both, indicating that decreasing the number of features did not significantly impact model performance. Various regression models were employed, with Random Forest and Bagging Regressor achieving the highest prediction accuracy across multiple evaluation metrics. While Random Forest demonstrated slightly higher variance, it consistently outperformed other models on the testing dataset. However, Decision Tree exhibited a noticeable gap between training and testing errors, suggesting potential overfitting.

To make the model practical and accessible, we designed a command-line interface that allows users to input their travel details and receive recommendations on the optimal booking date, airline, and estimated price. This tool enhances travel planning by offering insights to minimize costs.

Member Contribution

All members contributed equally.Everyone discussed and worked together. The tasks listed below are only a representation of the assigned tasks.

- **Aarzoo** - Preprocessing, Model Training
- **Anushka** - Preprocessing, Model Training
- **Nandini** - EDA, Model Training
- **Suhani** - EDA, Model Training

References

[1] “Dynamic flight price prediction using machine learning algorithms,” IEEE Conference Publication — IEEE Xplore, Dec. 16, 2022. Paper 1

[2] “Flight price prediction for enhanced recommendations via machine learning web application,” IEEE Conference Publication — IEEE Xplore, Jul. 10, 2024. Paper 2

[3] “A prediction of flight fare using K-Nearest neighbors,” IEEE Conference Publication — IEEE Xplore, Apr. 28, 2022. Paper 3

[4] ”Flight Price Prediction,” Kaggle. Dataset

[5] ”Flight Fare Prediction — Time Series ML Project”, Medium. Article