

FareSight: Leveraging ML to forecast flight prices

Aarzoo (2022008)

Anushka Srivastava (2022086)

Nandini Jain (2022316)

Suhani Kalyani (2022511)

Group Number 14



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI



- **Understanding Flight Price Volatility**

The aviation industry often sees significant fluctuations in flight ticket prices, often leaving travelers confused about the factors affecting these variations and the right time to book the tickets.

- **The Financial Impact on Travelers**

Price variations can affect the travel plans and budgets of millions of people. The unpredictability creates uncertainty, impacting financial planning for vacations or business trips.

- **The Need for Predictive Tools**

Identifying key factors driving price changes and developing predictive tools can empower travelers to make well informed decisions and can save money. Thus making air travel more accessible.



Dynamic Flight Price Prediction Using Machine Learning Algorithms (Gupta et al. 2022)

- ❑ The paper describes a machine learning model capable of predicting flight prices.
- ❑ The study reveals that most airline ticket prices fluctuate daily.
- ❑ The dataset contains flight prices from various airlines between March and June 2019, across multiple cities, with relevant attributes.
- ❑ Three machine learning models are implemented: Linear Regression, Random Forest, and Decision Trees.
- ❑ Accuracy achieved: 80%

Flight Price Prediction for Enhanced Recommendations via Machine Learning Web Application(Chavan et al. 2024)

- ❑ Six different machine learning algorithms were evaluated for accuracy and reliability.
- ❑ Performance Metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), other relevant metrics
- ❑ Random Forest Regressor achieved the highest R^2 score, outperforming the other algorithms.

A Prediction of Flight Fare Using K-Nearest Neighbors(Prasath et al. 2022)

- ❑ The paper identifies factors driving airplane price fluctuations and their influence on price changes.
- ❑ K-Nearest Neighbors (KNN) algorithm was used to model these factors.
- ❑ Data Visualization was used to determine the most impactful factors on airplane prices.
- ❑ K-Nearest Neighbors algorithm achieved an accuracy of 81.77%.

Dataset Description

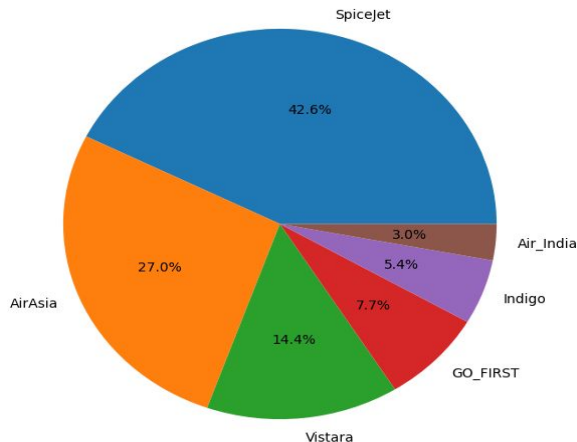


- We have used the Flight Price Prediction dataset from Kaggle ([Flight Fare Dataset](#)).
- The dataset has 3,00,153 records and 11 columns which consists of 10 features and 1 label.
- There are no NULL values in the dataset.
- The feature duration and days_left are numerical.
- Other features are categorical features.
- The price is a continuous target value.

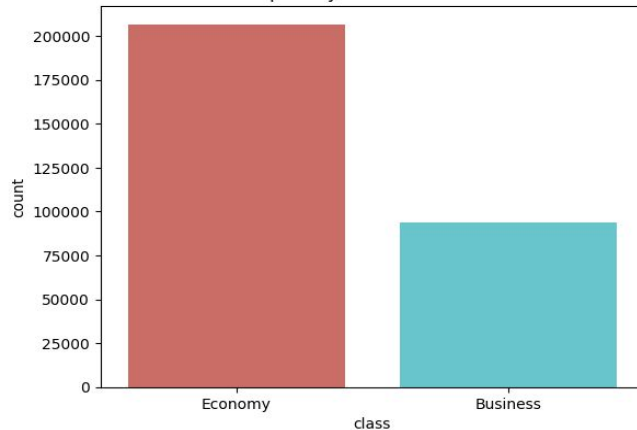
#	Column	Count	Null Values	Dtype
0	airline	300153	0	object
1	flight	300153	0	object
2	source_city	300153	0	object
3	departure_time	300153	0	object
4	stops	300153	0	object
5	arrival_time	300153	0	object
6	destination_city	300153	0	object
7	class	300153	0	object
8	duration	300153	0	float64
9	days_left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

Table 1. Dataset Column Information

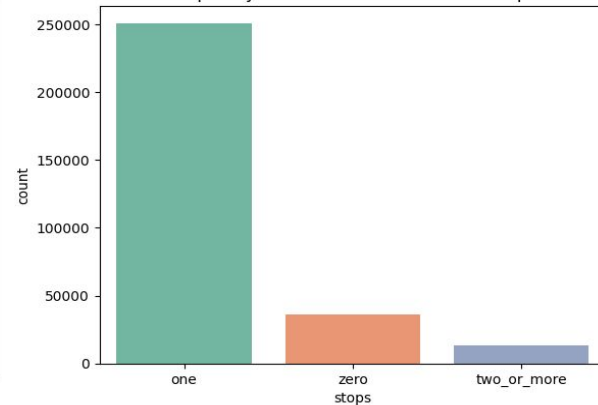
Distribution for different airlines



Countplot by distribution for class



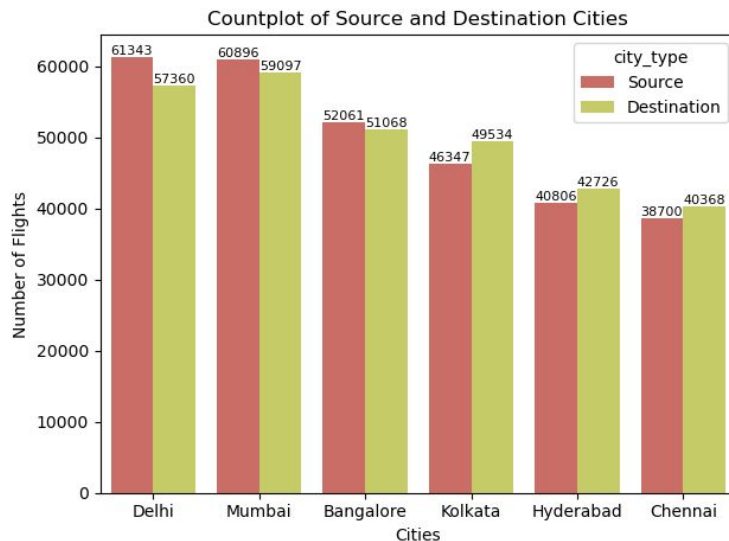
Barplot by Distribution for Number of Stops



→ There are 6 different airlines in our dataset - Vistara, Air India, Indigo, GO_FIRST, AirAsia, and SpiceJet. SpiceJet has maximum count while Air India has minimum count.

→ There are two types of classes - Economy Class and Business Class. The count of Economy class is almost twice as more than Business Class.

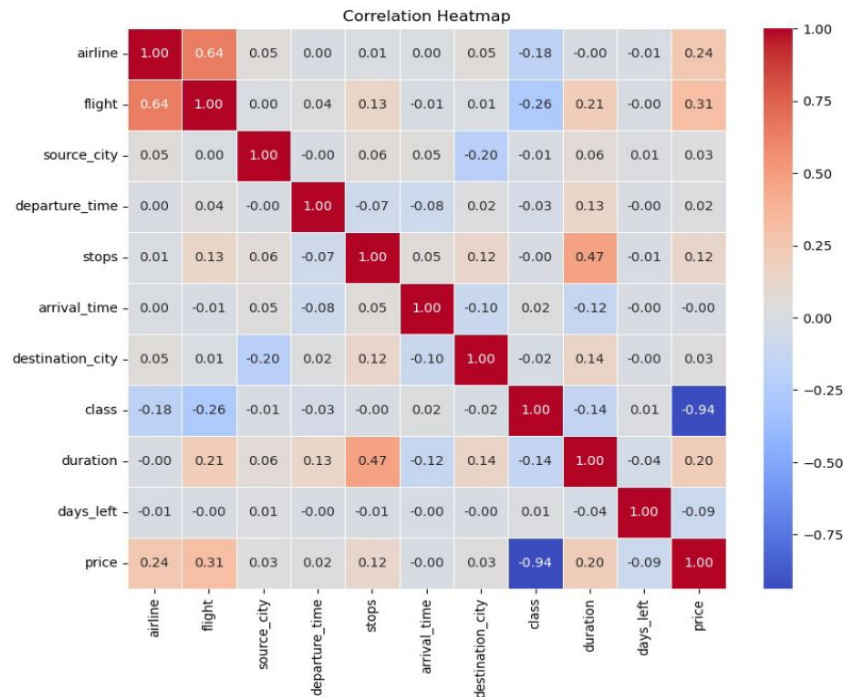
→ This feature is a discrete value. Most flights have one stop.



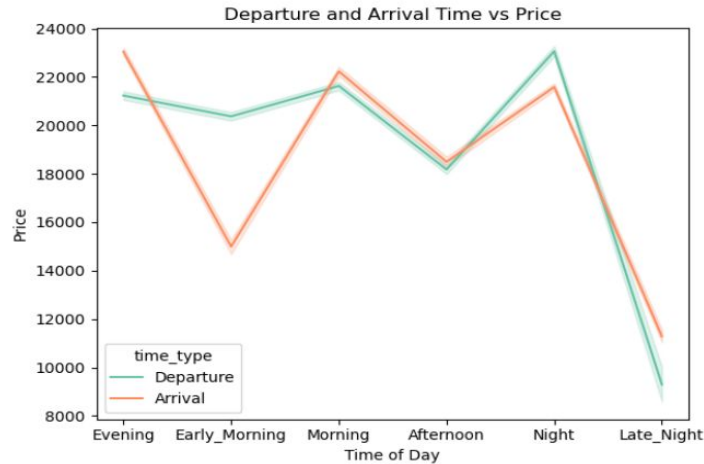
- The flights travel to and fro from the following cities: Delhi, Mumbai, Bangalore, Kolkata, Hyderabad, and Chennai.
- There are maximum outgoing flights from Delhi, while there are maximum arriving flights for Mumbai.

Correlation Matrix

- Feature class and the label price are highly negatively correlated.
- Features like arrival time, departure time, source city and destination city do not have significant impact on the flight prices.

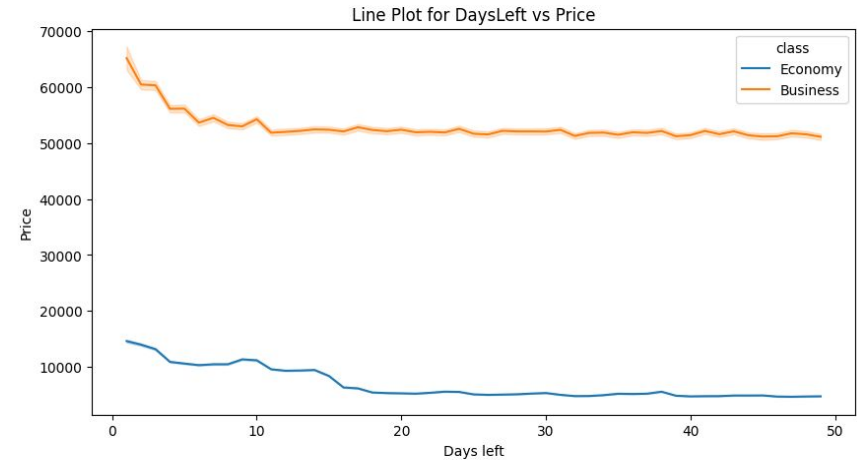


EDA Analysis



**Line plot for Departure and Arrival
Time VS Price**

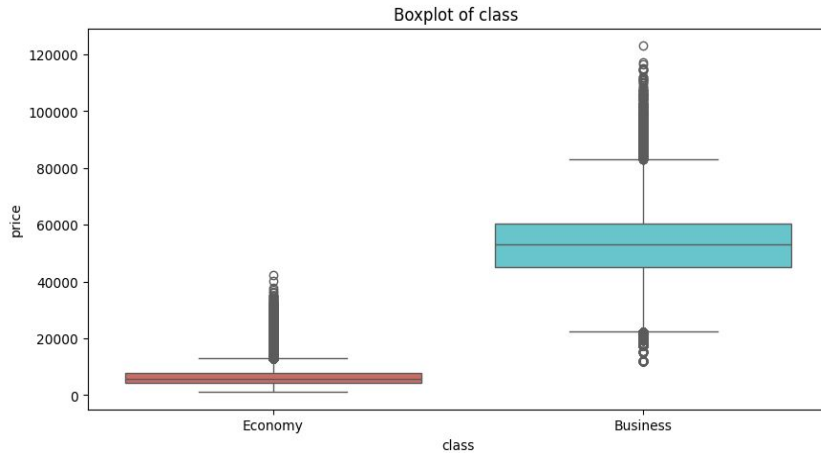
- Flights which arrive in the evening, morning or night have higher prices than those arriving early morning or late night .
- Flights which depart in late night are cheapest.



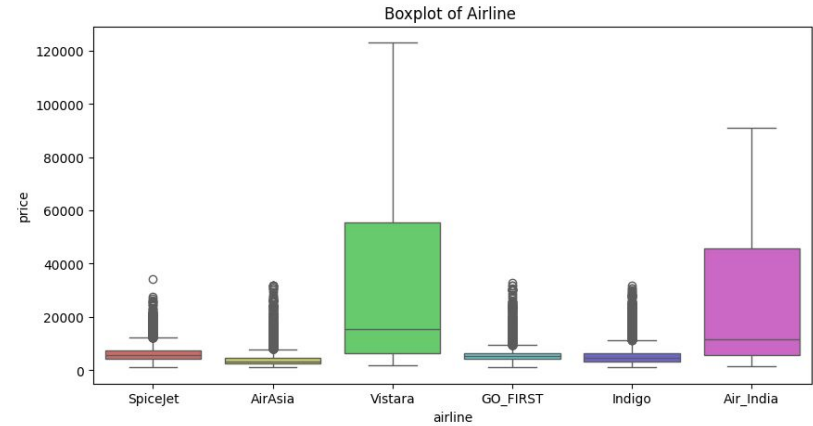
**Line plot for Flight prices vs. days left for
departure**

- 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.
- Economy class prices start at a lower level and gradually increase as the days left falls below 20.

EDA Analysis



- 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.



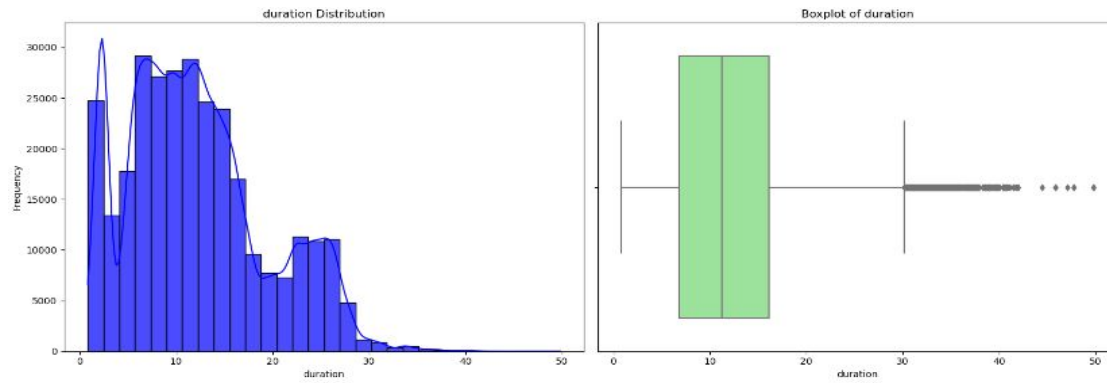
- Vistara has the highest overall price range of the flights
- Lower-cost airlines like SpiceJet, AirAsia, GO FIRST and Indigo exhibit lower price ranges.
- There are many outliers present in these lower-cost airlines.

Data Preprocessing

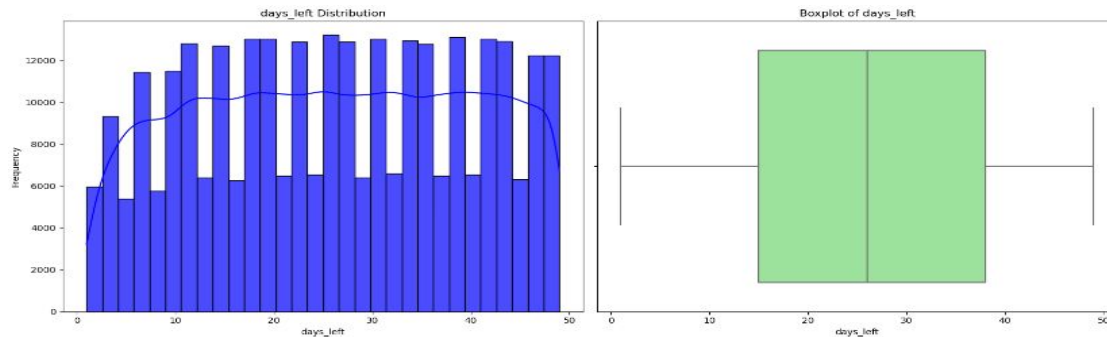


Outlier Detection and Removal

→ Numerical Features :
duration , days_left



→ There were total 2110 outliers for the duration feature and None were found for days_left



Data Preprocessing

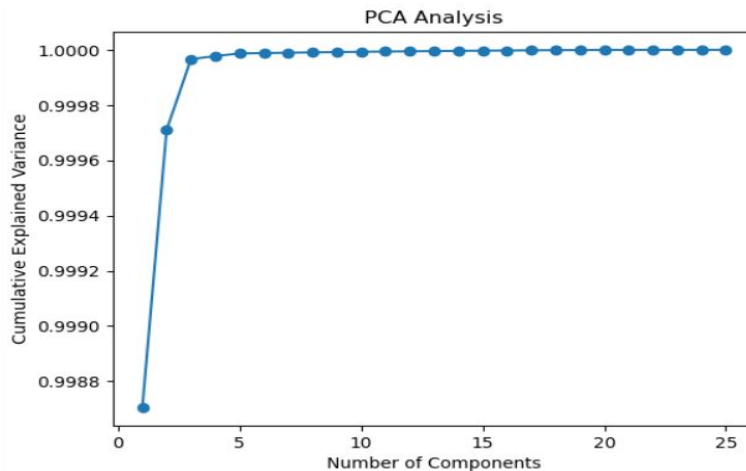
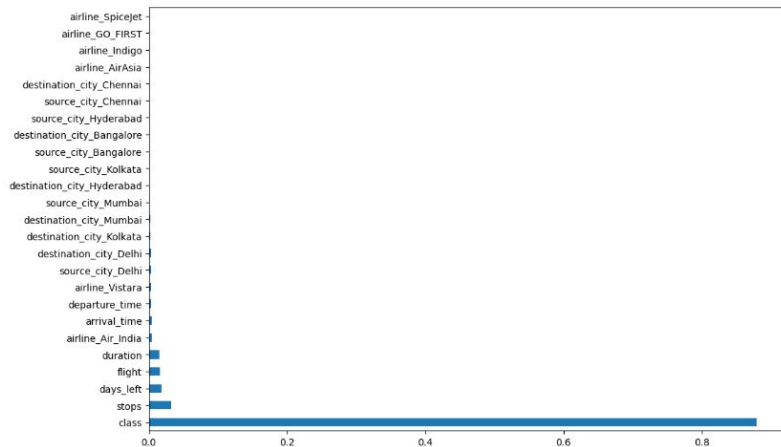


- 8 out of 10 features are categorical, so we encode them.
- **Ordinal Features:** stops, arrival time, departure time, so we use Label Encoding with the help of manually defined maps to map them to integers
- **Nominal Features:** airline, source_city, destination city so we used One-Hot Encoding for them.
- Additionally, we also use integer mapping for features: flight and class.

After final processing of the dataset, we have converted all features to numerical values.

```
# Column Non-Null Count Dtype
---
0 flight 298043 non-null int32
1 departure_time 298043 non-null int64
2 stops 298043 non-null int64
3 arrival_time 298043 non-null int64
4 class 298043 non-null int32
5 duration 298043 non-null float64
6 days_left 298043 non-null int64
7 price 298043 non-null int64
8 airline_AirAsia 298043 non-null float64
9 airline_Air_India 298043 non-null float64
10 airline_GO_FIRST 298043 non-null float64
11 airline_Indigo 298043 non-null float64
12 airline_SpiceJet 298043 non-null float64
13 airline_Vistara 298043 non-null float64
14 source_city_Bangalore 298043 non-null float64
15 source_city_Chennai 298043 non-null float64
16 source_city_Delhi 298043 non-null float64
17 source_city_Hyderabad 298043 non-null float64
18 source_city_Kolkata 298043 non-null float64
19 source_city_Mumbai 298043 non-null float64
20 destination_city_Bangalore 298043 non-null float64
21 destination_city_Chennai 298043 non-null float64
22 destination_city_Delhi 298043 non-null float64
23 destination_city_Hyderabad 298043 non-null float64
24 destination_city_Kolkata 298043 non-null float64
25 destination_city_Mumbai 298043 non-null float64
dtypes: float64(19), int32(2), int64(5)
```


Data Preprocessing



- Dataset have encoded consisted of 25 features
- Performed ExtraTreesRegressor followed by PCA (Principal Component Analysis) to reduce the dimensionality of our large dataset with many irrelevant features.
- Optimal Number of Components from PCA were found to be 7.

- The dataset contains 298,043 rows .
- Train-test split - 70:30
 - Training - 208,630 samples
 - Testing - 89,413 samples
- We used **Regression models** because the task involves predicting a continuous target variable- 'price'.

Evaluation Metrics

- Mean Squared Error (MSE)
 - Root Mean Squared Error (RMSE)
 - R-squared (R^2) Score
 - Adjusted R^2 Score
 - Mean Absolute Error (MAE)
- 

Models Used



- **Linear Models**
 - Linear Regression
 - Lasso Regression ($\text{lr} = 1$)
 - Ridge Regression ($\text{lr} = 1$)
- **Non-linear Models**
 - Decision Tree Regression - prone to overfit
- **Ensemble Models**
 - Random Forest Regression (100 estimators)
 - XGBoost Regression (100 estimators and $\text{lr} = 1$)
 - Bagging Regression (300 DTs)

Grid Search was also performed on our best performing model - Random Forest, giving best parameters as 200 `n_estimators` and `max-depth` as 20.

Testing Error Without Feature Extraction



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

- **Random Forest and Bagging Regression performs the best**
They give low MSE and high R^2 compared to other models showing comparatively better generalization. However, a significant gap between training and testing error compared to other models.
- **Comparable performance of Linear, Ridge, Lasso Regression and XGBoost**
Give similar results in both training and testing errors. Relatively higher errors. XGBoost performs better than these 3 but worse than Random Forest.
- **Possible overfitting in Decision Tree**
Huge gap between training and testing errors.

Testing Error With Feature Extraction

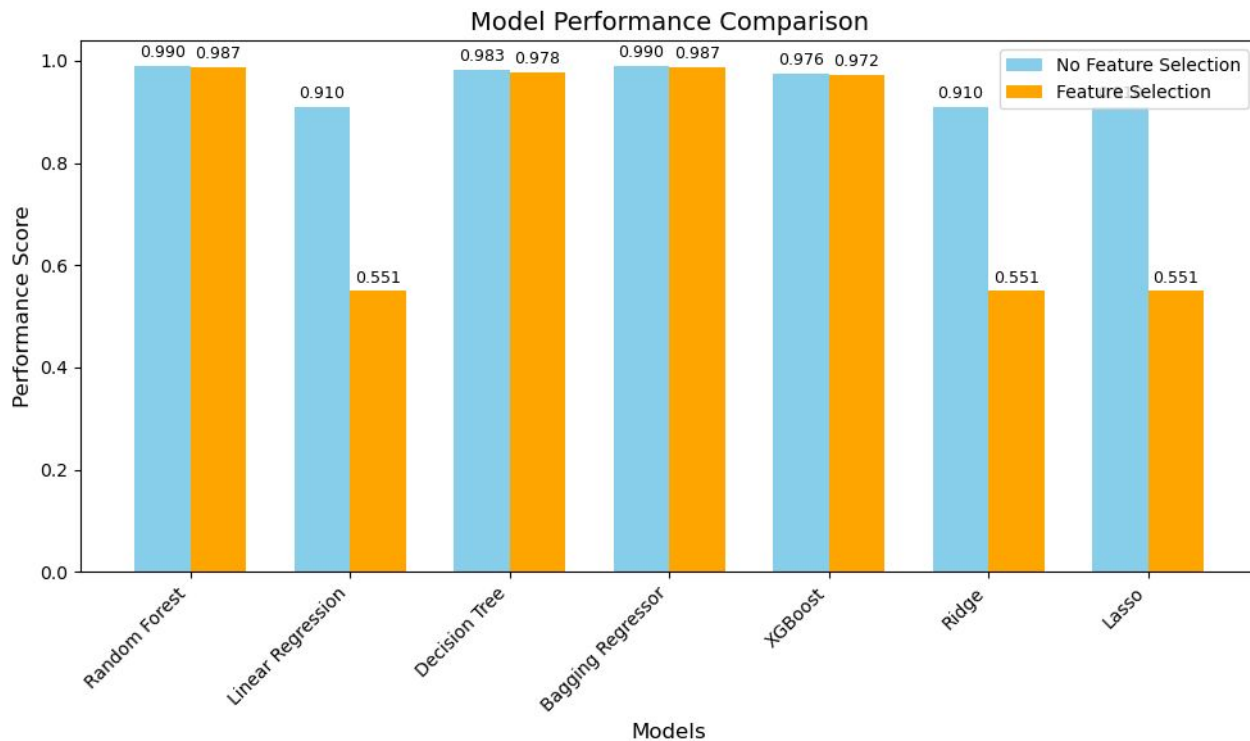


MODEL NAME	MSE	RMSE	R2	ADJUSTED R2	MAE
Linear Reg	231935823.372	15229.439	0.551	0.551	12557.948
Random Forest	6867957.545	2620.678	0.986	0.986	1026.754
Decision Tree	11356508.598	3369.941	0.978	0.978	1115.286
Bagging Reg	6826426.636	2612.743	0.986	0.986	1022.045
XGB	14689042.174	3832.628	0.971	0.971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

- **Random Forest, Decision Tree, Bagging Regression still performs the best**
Among all the models trained with feature extraction, they still give low MSE and high R^2 compared to other models showing comparatively better generalization.
- **XGBoost and Decision Tree also maintains performance**
Decision Tree and XGBoost perform slightly worse compared to Random Forest and Bagging but still significantly outperform linear models.
- **Linear Models show huge errors**
As compared to all the models trained with feature extraction, linear models like ridge, lasso and linear regression have comparable performance but relatively higher errors.



Analysing Feature Extraction



Analysing Feature Extraction



- **No significant difference in performance of tree-based models**

The performance of Random Forest, Decision Tree, XGBoost and Bagging Regressor on the dataset with feature extraction is largely similar to the dataset without feature extraction. This indicates that these models are better at handling a reduced feature set without significant performance loss.

- **Significant deterioration in the performance of linear models**

The performance of Linear, Ridge and Lasso Regression on the dataset with feature extraction falls down significantly as compared to their performance on the dataset without feature extraction.



Analysing GridSearch



- Applied on our best performing model Random Forest.
- 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.



Now solving our initial problem...

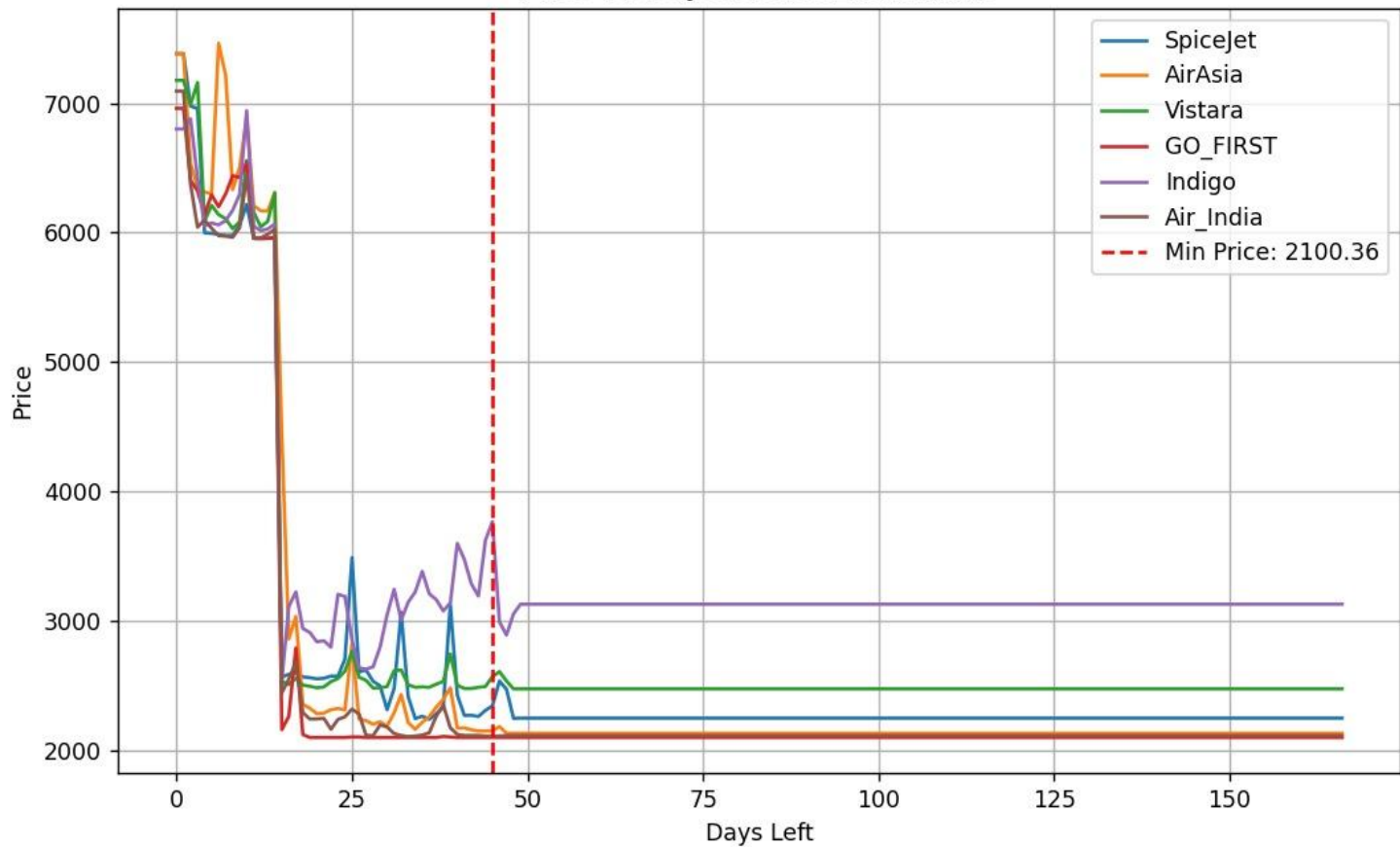


- We aim to predict the minimum price, the best airline to choose and the best date to book tickets on given the travellers preferences.
- We developed a CLI to help the user interact with our model.
- We take the source city, destination city, the date of departure, the date of arrival, preferred departure time, preferred arrival time, preferred number of stops and the preferred seat class as input.
- We create a feature vector using this and encode the categorical data according to the encodings used while training the model.
- We then analyse all the combinations of airlines with the days left for departure wrt current date and display the best airline, best price and best date to book the ticket. We also display the result in graphical form.

```
(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
```


Price vs. Days Left for Each Airline



Conclusion



- High correlation between **price** and **class** indicating high dependency between the features.
- Despite a lower correlation between other features and price, we observed similar loss and accuracy in models trained with and without feature extraction for tree-based models, indicating that these models are more robust.
- However, performance of linear models fell down significantly during feature extraction.
- Best performing models - **Random Forest** and **Bagging Regressor** despite comparatively higher variance as they perform relatively better on testing dataset compared to other models.

Timeline and Contributions



We were able to follow our proposed timeline.

All members contributed equally to the project, discussing the code and helping each other with writing the report and analysis. The individual contributions listed below reflect only the task assigned to each.

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



Thank You