CMPE 255 – Fall 2024

Project Report: Flight Prices Classification

Team Name: Data Mining Developers
Team Members: Vivek Ponnala, Maggie Qin, Hoai An Nguyen
Project Title: Flight Prices Classification: An Approach Using Random Forest, Decision Trees, and KNN

---

1. Introduction

Motivation

Frequent fluctuations in airline ticket prices affect both the consumer's ability to budget and the airline industry's revenue optimization. Understanding these fluctuations is critical for enabling better price predictions and decision-making. Our motivation is to provide insights into factors driving ticket prices and to identify patterns that can help predict future price trends, benefiting both consumers and industry professionals.

Objective

Our objective is to build a robust classification model that categorizes flight prices into three ranges: low, medium, and high. This categorization allows us to simplify the complex pricing dynamics and create a model capable of informing price-sensitive travel decisions and optimizing resource allocation for the airline industry.

Literature Review

Several studies have explored airline pricing models, often using machine learning to uncover price determinants. Random Forest, Support Vector Machines, and Neural Networks are frequently applied due to their effectiveness in handling complex data. Inspired by these studies, we selected the Random Forest algorithm for its strong performance on structured data and its ease of interpretability for classification tasks.

---

2. System Design & Implementation Details

Algorithm Selection and Justification

We chose the Random Forest algorithm for its strengths in handling multi-dimensional, complex datasets with categorical and continuous features. It also offers interpretability in terms of feature importance, which is beneficial for identifying influential variables in price categorization.

Decision tree choice

KNN choice reason

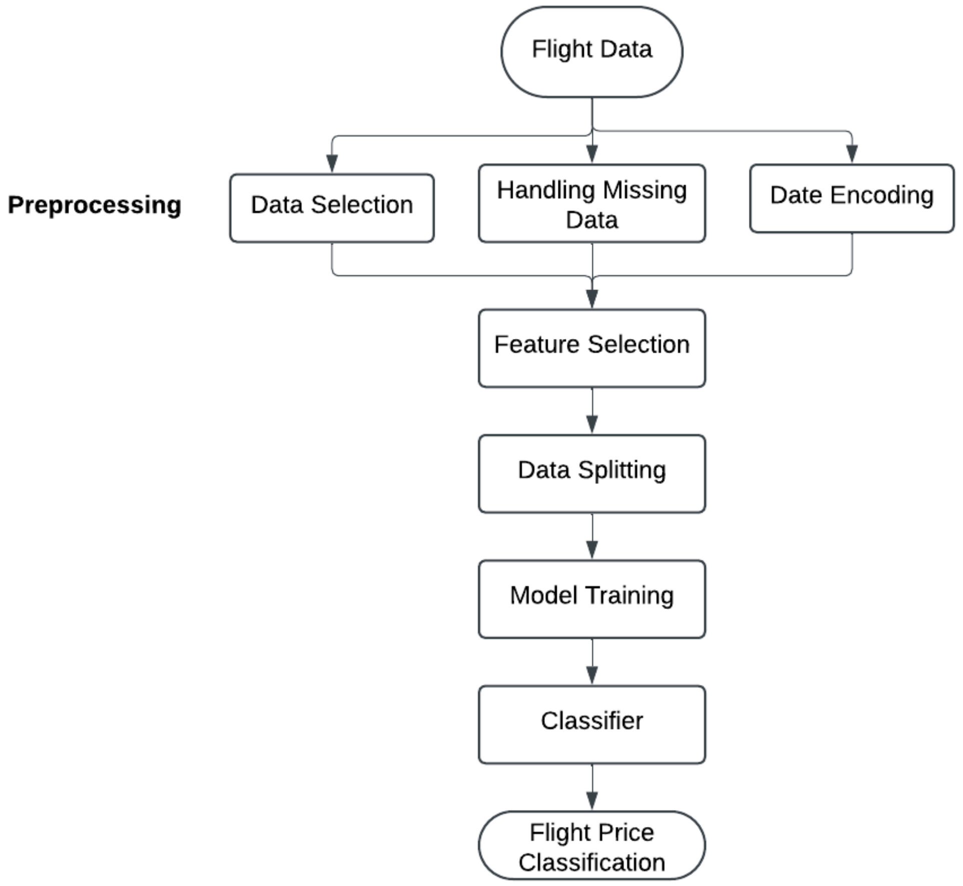
Technologies and Tools

- **Python**: Selected as the primary language for its rich library ecosystem.
- **pandas** and **NumPy**: Utilized for data loading, manipulation, and preprocessing.
- **scikit-learn**: Provides an accessible implementation of Random Forest and various metrics for model evaluation.
- **matplotlib** and **seaborn**: Used for data visualization and result interpretation.

Data Pipeline

1. **Data Loading and Cleaning**: The data was filtered to select relevant features and entries based on our analysis objectives.
2. **Feature Engineering**: New variables were created, such as a weekend indicator, to capture additional context on price fluctuations.
3. **Model Training and Evaluation**: The Random Forest model was trained and evaluated with an 80-20 training-testing split.
4. **Visualization and Interpretation**: Plots and metrics were generated to evaluate model performance and to examine feature importance.

Architecture Diagram



---

3. Dataset Description and Feature Engineering

Dataset Overview

The dataset was sourced from Kaggle and includes over 5 million records of flight ticket prices found on Expedia from April to October 2022. We used a subset focused on flights departing from San Francisco International Airport (SFO) in May 2022, resulting in approximately 250,000 entries for manageable analysis. This is the link to the dataset: https://drive.google.com/file/d/1fn4NaQ3xzocOg0mqfDkSqpNCcCMthjXW/view?usp=drive_link.

Please put the actual link for the dataset

Random Forest

Selected Features:

1. **`seatsRemaining`**: Represents the number of seats left, highlighting supply-demand trends.
2. **`totalTravelDistance`**: Total distance of the flight, directly correlated with ticket prices.
3. **`isNonStop`**: Binary variable indicating whether the flight is non-stop, which influences ticket prices significantly.
4. **`isWeekend`**: A derived feature from `searchDate`, indicating whether the search happened on a weekend.
5. **`day`**: Extracted from the date, capturing daily fluctuations in ticket prices.
6. **`month`**: Reflects seasonal demand and pricing patterns.
7. **`year`**: A higher-level indicator of the time frame (though it likely remains constant in this dataset subset).

Feature engineering:
`isWeekend`: Created by extracting the day of the week from `searchDate`. Encoded as a binary variable (1 = Weekend, 0 = Weekday) to reflect demand spikes during weekends.
Price Quantiles:

- Flight prices were divided into three quantiles: low, medium, and high.
- This transformation converted a continuous price variable into a classification target.

Date Feature Extraction:

- `day`, `month`, and `year` were extracted from `searchDate` to break down temporal patterns in ticket prices.

Categorical Encoding:

- Binary encoding was applied to features like `isNonStop` and `isWeekend` to enable them to work seamlessly with the Random Forest model.

Scaling (Optional):

- While Random Forest is robust to feature scaling, features like `totalTravelDistance` were likely standardized to ensure consistency during feature analysis.

Decision Tree
Selected features:
Feature engineering:

KNN
Selected features:

- **searchDate**: The date of the search, which can influence price based on seasonality.
- **destinationAirport**: The destination of the flight, as demand for certain routes may affect price.
- **isNonStop**: A binary variable indicating whether the flight is non-stop, a factor known to impact pricing.
- **seatsRemaining**: The number of available seats, which is inversely correlated with price increases.
- **totalTravelDistance**: The travel distance, as longer flights typically have higher base fares.

Feature Engineering

- **Weekend Indicator**: Derived from `searchDate`, this feature indicates if the search occurred on a weekend, capturing variations in demand.
- **Price Category**: Prices were divided into three quantiles (low, medium, high) to create a target variable suitable for classification.

4. Methodology

Data Preprocessing

Data preprocessing involved filtering rows with missing or irrelevant values, converting date features, and encoding categorical variables. Feature scaling was applied where necessary to optimize Random Forest performance, although the algorithm is relatively insensitive to scale compared to other models.

Model Selection and Training

Random Forest

We selected Random Forest for its ability to handle mixed data types and its suitability for classification tasks. The model was configured with 100 trees and a maximum depth chosen based on cross-validation to balance accuracy and training time. Data was split 80-20 for training and testing, ensuring a sufficient sample for model evaluation.

Decision Tree

KNN

Evaluation Metrics

The classification performance was evaluated using the following metrics:

- **Accuracy**: Measures the overall correctness of the model.
- **Precision**: Indicates the accuracy of positive predictions.
- **Recall**: Reflects the model's ability to capture actual positive cases.
- **F1 Score**: Balances precision and recall, providing a holistic view of model performance.

---

5. Experimental Results

Model Performance

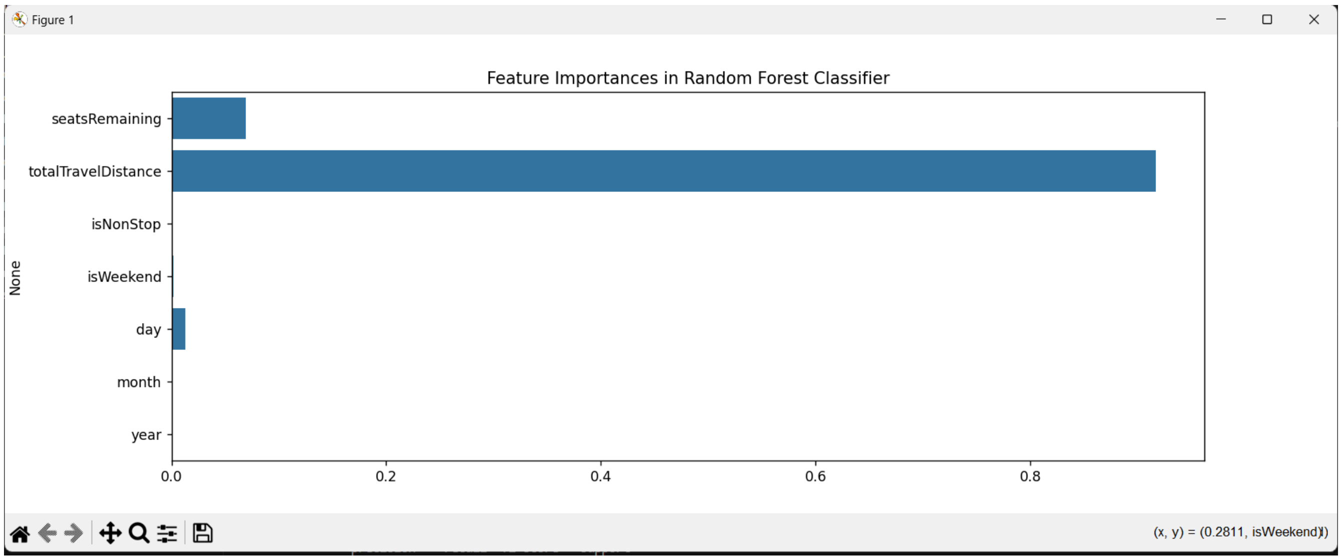
The Random Forest classifier achieved the following performance metrics on the test data:

- **Accuracy**:63%
- **Precision**: 62%
- **Recall**: 63%
- **F1 Score**: 62%

These results indicate that the model performs moderately well in classifying flight prices into distinct ranges.

Feature Importance

The feature importance plot reveals that `seatsRemaining` and `totalTravelDistance` are the most influential factors, as expected. `isDay` and `isWeekend` also contribute meaningfully to the model's decisions.

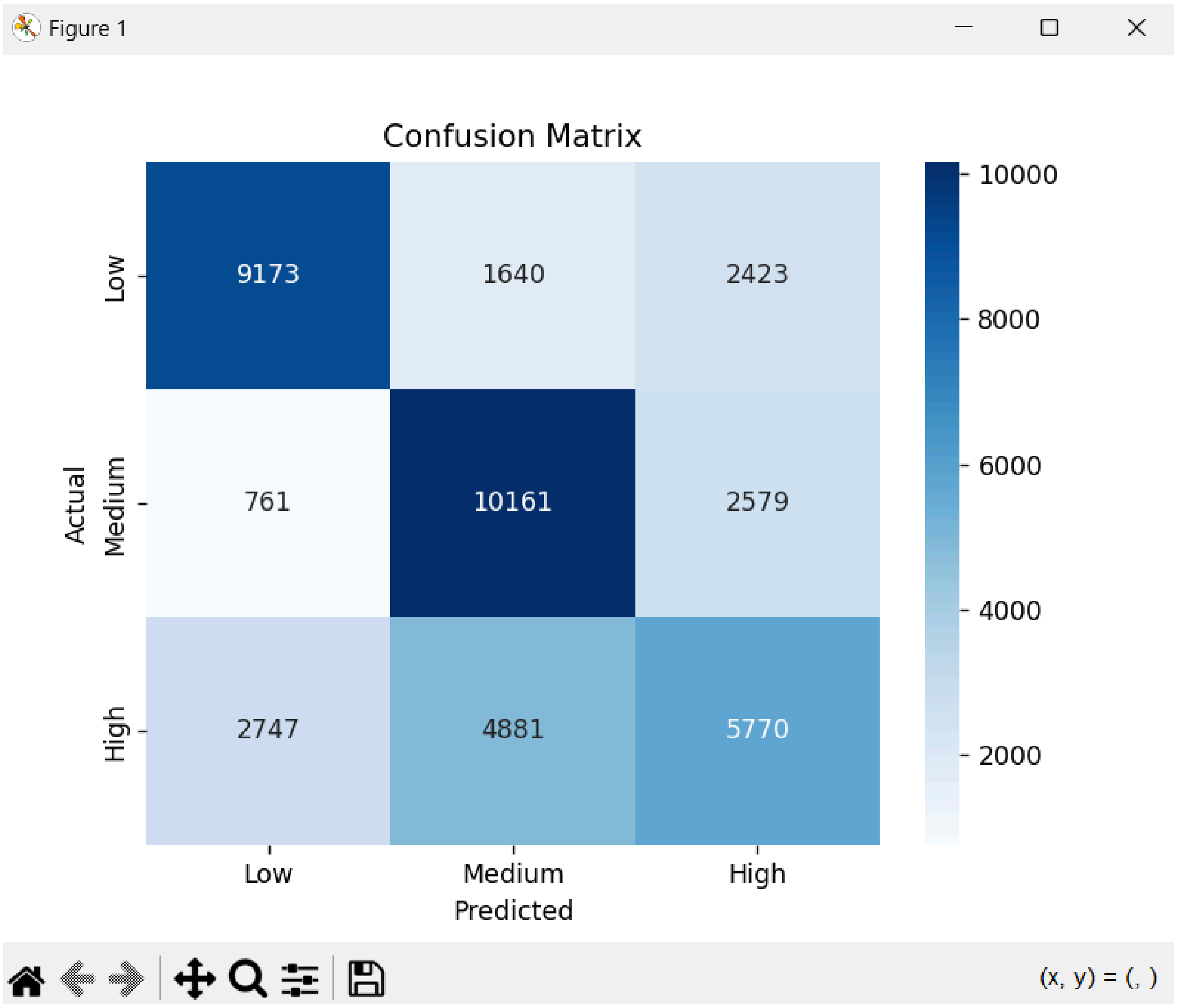


Feature importance graph

Confusion Matrix Analysis

Random Forest

The confusion matrix shows that the model is generally effective at distinguishing between high and low price categories but sometimes misclassifies medium prices, likely due to overlap in pricing strategies for mid-range fares.



The confusion matrix shows that the Random Forest classifier performs well in distinguishing "Low" prices but struggles to separate "Medium" and "High" categories. It has the highest correct predictions for "Medium" (10,161 instances), but significant misclassifications occur between "Medium" and "High" due to overlapping features. The "Low" category has relatively fewer misclassifications, highlighting its distinctiveness.

Decision Tree

// one sentence comment

Insert feature importance graph and confusion matrix graph

Explain graph

KNN

// one sentence comment

Insert feature importance graph and confusion matrix graph

Explain graph

6. Discussion

Key Insights and Decisions

Algorithm Selection:
The decision to use Random Forest was justified due to its ability to handle complex datasets with both categorical and continuous features. Its ensemble-based approach reduces the risk of overfitting compared to a single decision tree and delivers better generalization on unseen data. Additionally, Random Forest provides feature importance metrics, enabling interpretability and insights into the most impactful variables for flight price classification.

Feature Engineering:
Including the **weekend indicator** was particularly beneficial for capturing variations in demand trends. Prices often fluctuate significantly during weekends, and this feature allowed the model to better account for these patterns. Similarly, the **seatsRemaining** and **totalTravelDistance** features provided key information related to supply-demand dynamics and the impact of travel distance on pricing.

Why Choose Random Forest

1. **Robustness to Overfitting**:
   Random Forest averages predictions across multiple decision trees, reducing the risk of overfitting compared to single-tree models. This is crucial for datasets with overlapping class boundaries, such as flight prices.
2. **Handling Feature Complexity**:
   The algorithm handles mixed data types (categorical and continuous) seamlessly and is robust to outliers, which are common in flight pricing data.

3. **Feature Importance Analysis**:
   Random Forest ranks features by importance, which helps in understanding the driving factors behind flight prices. For example, **totalTravelDistance** and **seatsRemaining** were identified as critical factors in this analysis.
4. **Versatility**:
   Random Forest performs well with imbalanced data and is less sensitive to hyperparameters. This makes it a reliable choice for projects with limited time for optimization.
5. **Scalability**:
   It scales well with large datasets, making it suitable for the Kaggle dataset containing over 5 million records, even when reduced to a subset.

Why choose decision tree

Why choose KNN

Challenges Faced
Random forest

The Random Forest model faced challenges with data imbalance, overlapping features, and computational complexity. Class balancing techniques, feature engineering (e.g., weekend indicator), and hyperparameter tuning were employed to address these issues. Feature importance analysis further refined the model, enhancing accuracy and reliability.

Decision tree

KNN

Observations on Model Performance

Random Forest:

- **Accuracy**: Random Forest achieved an accuracy of 63%, which is moderate but higher compared to KNN and Decision Tree due to its ensemble nature that reduces overfitting.
- **Precision**: The model's precision (62%) indicates it makes reasonably accurate predictions, but misclassifications, particularly between "Medium" and "High" categories, lower its overall effectiveness.
- **Recall**: With a recall of 63%, the model captures most of the actual occurrences in each class, outperforming Decision Tree in identifying patterns in the data.

- **Feature Importance**: Random Forest provides a clear ranking of influential features (e.g., totalTravelDistance and seatsRemaining), offering interpretability absent in KNN.

---

7. Conclusion

The project demonstrated that Random Forests can be used effectively to classify flight prices into price ranges. The model achieves moderate accuracy and provides insights into key features influencing pricing decisions. Future work could explore other classifiers or incorporate more advanced hyperparameter tuning for improved accuracy.

---

8. Project Plan and Task Distribution

We all worked on: data cleaning, feature engineering, data visualization, evaluation metrics, report documentation, presentation preparation

- **Vivek Ponnala**: Creating the github, random forest, report and slide presentation
- **Maggie Qin**: KNN, report and slide presentation
- **Hoai An Nguyen**: decision tree, report and slide presentation

Each member participated in group discussions, code reviews, and collaborative troubleshooting, ensuring a balanced workload distribution.

---

9. References

- **Dataset Source:** Kaggle, Flight Prices Dataset
- **Data subset:**
- **Python Libraries:** pandas, NumPy, scikit-learn, matplotlib, seaborn

GitHub Repository:

https://github.com/vipvivek15/CMPE_255_Final_Project

10. Bibliography

KNN

1. **K-Nearest Neighbors Algorithm** - Scikit-learn Documentation
   https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
2. **KNN Algorithm Simplified** - Towards Data Science
   https://towardsdatascience.com/knn-k-nearest-neighbors-1a470930e1d0
3. **A Gentle Introduction to KNN** - Machine Learning Mastery
   https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/

Random Forest

1. **Random Forest Classifier** - Scikit-learn Documentation
   https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
2. **Understanding Random Forest** - Towards Data Science
   https://towardsdatascience.com/understanding-random-forest-58381e0602d2
3. **How Random Forest Works** - Analytics Vidhya
   https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Decision Tree

1. **Decision Tree Classifier** - Scikit-learn Documentation
   https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
2. **Decision Tree Explained** - Towards Data Science
   https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
3. **Introduction to Decision Trees** - Machine Learning Mastery
   https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/