CMPE 255 – Fall 2024

Project Report: Predicting Flight Prices

Team Name: Data Mining Developers
Team Members: Vivek Ponnala, Maggie Qin, Hoai An Nguyen
Project Title: Predicting Flight Prices: An Approach Using Random Forest

---

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.

Other options, like Support Vector Machines, were considered but ultimately not selected due to their higher computational cost on large datasets.

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

A data flow diagram or architecture diagram would go here to show the flow from data loading to model training and evaluation.

---

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.

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

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.

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**: 60.23%
- **Precision**: 64.47%
- **Recall**: 60.23%
- **F1 Score**: 60.93%

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. `isNonStop` and `isWeekend` also contribute meaningfully to the model's decisions.

Include a feature importance plot here to visually demonstrate the weight of each feature.

Confusion Matrix Analysis

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.

Include a confusion matrix here to give a detailed view of classification accuracy.

---

6. Discussion

Key Insights and Decisions

- **Algorithm Selection**: The decision to use Random Forest was justified, as it balanced accuracy with interpretability.
- **Feature Engineering**: Including a weekend indicator proved beneficial for capturing demand trends.

Challenges Faced

- **Data Imbalance**: Initially, there were more entries in the low and medium categories, which required careful balancing.
- **Feature Selection**: Selecting features with the highest predictive power required iterative testing and adjustment.

Observations on Model Performance

The model’s accuracy could be further improved by integrating additional features, such as holiday periods, or by experimenting with more complex ensemble methods.

---

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

- **Vivek Ponnala**: Data cleaning, feature engineering, and data visualization.
- **Maggie Qin**: Model training, hyperparameter tuning, and results interpretation.
- **Hoai An Nguyen**: Evaluation metrics, report documentation, and presentation preparation.

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

---

9. References

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

GitHub Repository:

https://github.com/vipvivek15/CMPE_255_Final_Project