Progress Report Machine Learning Project on Flight Route Optimization

Gizem Erol, Selvinaz Zeynep Kıyıkcı, Zeynep Sude Kırmacı

December 2024

1. Introduction

The primary goal of this project is to optimize flight routes using machine learning techniques. By analyzing historical flight data, including delays and weather conditions, the project aims to predict potential delays and determine the shortest, most efficient routes for airlines. This report details the progress made so far, including data preprocessing, model training, and the development of a graph-based routing algorithm.

2. Accomplishments

2.1 Data Collection and Preparation

We utilized the "Flights.csv" dataset, which includes detailed flight information such as airport IDs, distances, flight times, weather conditions, and delays. The following preprocessing steps were implemented:

- **Data Cleaning:**
  - Removed commas from numerical fields like `DistanceKilometers` and converted them to floats.
  - Removed unnecessary symbols (e.g., $ and ,) from ticket prices and distances.
  - Handled missing values in key columns such as `FlightTimeMin` (filled with mean values) and `Weather` fields (filled with 'Unknown').
  - Ensured all string-based fields, such as `Origin` and `Dest`, were standardized to uppercase.

- **Feature Encoding:**
  - Encoded categorical variables like `OriginWeather(Origin Weather, DestWeather(Destination Weather), AvgTicketPrice(Average ticket prices), DistanceKilometers, FlightTimeHour` and `FlightTimeMin` into numeric values using factorization.

- **Target Selection:**
  - Identified `FlightDelayMin` as the target variable for delay prediction and selected features such as `DestWeather`, `OriginWeather`, `FlightTimeMin`, `FlightTimeHour, AvgTicketPrice` and `DistanceKilometers` for model training.

2.2 Graph Construction

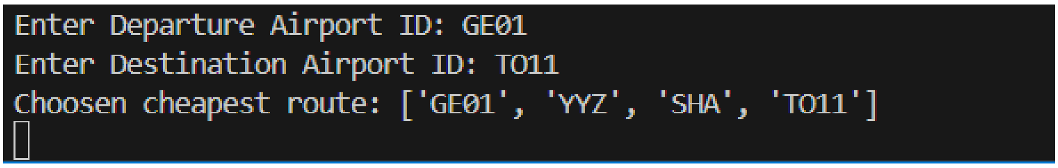
A directed graph was constructed using the NetworkX library to model the flight network:

- Nodes represent airports, and edges represent flight connections with weights corresponding to average flight delays.
- A utility function (`create_graph`) was developed to dynamically create graphs from the dataset.

2.3 Shortest Path Algorithm

- Implemented Dijkstra’s algorithm using NetworkX’s `dijkstra_path` method to find the most efficient flight routes based on delay times.

- Created a function (`find_shortest_path`) that takes origin and destination airports as input and returns the optimal route and its total delay.
- We also tested finding the shortest path based on the best available ticket price.



```
Enter Departure Airport ID: GE01
Enter Destination Airport ID: TO11
Choosen cheapest route: ['GE01', 'YYZ', 'SHA', 'TO11']
```

Figure 1. Output from Shortes Path based on the best available ticket price.

2.4 Visualization

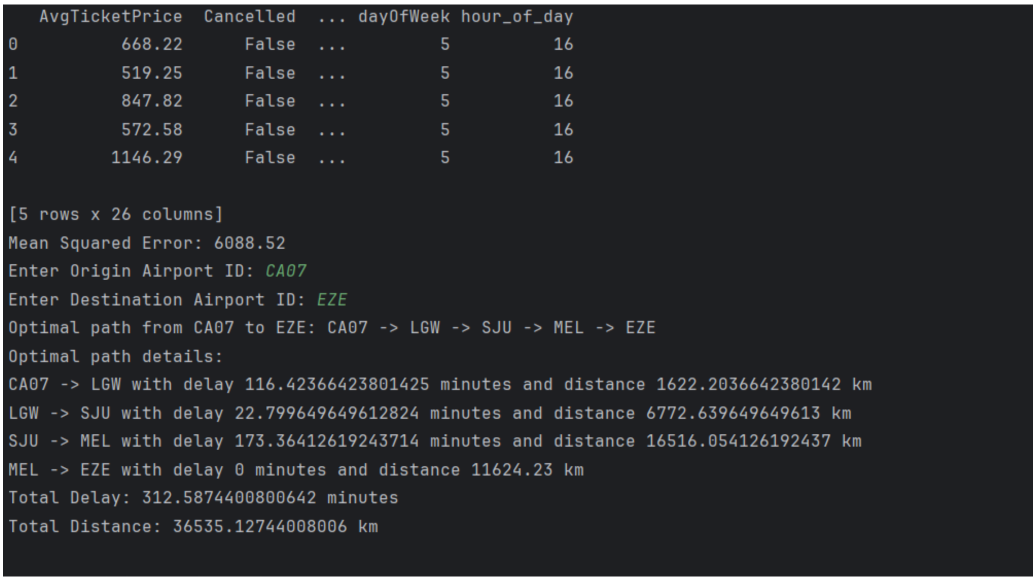
- Visualized the flight network using Matplotlib and NetworkX.
- Highlighted the computed optimal path on the graph to aid understanding and debugging.

2.5 Model Development

The project focuses on predicting flight delays using a supervised learning approach. So far, linear regression model, logistic regression model and decision tree algorithms have been selected. By experimenting with the models, we have focused on finding the training model that gives the most optimal values for our project. Later in the project, we will experiment with different models.

Linear Regression Model:

- **Training and Testing:**
  - Split the data into training (70%) and testing (30%) sets.
  - Trained the linear regression model using the selected features.
  - Evaluated the model's performance using Mean Squared Error (MSE). The MSE on the test set was high, indicating that the model's predictions were far from the actual values. This highlights the need for more robust models and feature engineering.



```
   AvgTicketPrice  Cancelled  ... dayOfWeek hour_of_day
0          668.22      False  ...         5          16
1          519.25      False  ...         5          16
2          847.82      False  ...         5          16
3          572.58      False  ...         5          16
4         1146.29      False  ...         5          16

[5 rows x 26 columns]
Mean Squared Error: 6088.52
Enter Origin Airport ID: CA07
Enter Destination Airport ID: EZE
Optimal path from CA07 to EZE: CA07 -> LGW -> SJU -> MEL -> EZE
Optimal path details:
CA07 -> LGW with delay 116.42366423801425 minutes and distance 1622.2036642380142 km
LGW -> SJU with delay 22.799649649612824 minutes and distance 6772.639649649613 km
SJU -> MEL with delay 173.36412619243714 minutes and distance 16516.054126192437 km
MEL -> EZE with delay 0 minutes and distance 11624.23 km
Total Delay: 312.5874400800642 minutes
Total Distance: 36535.12744008006 km
```

Figure 2. Output of the Linear Regression Model

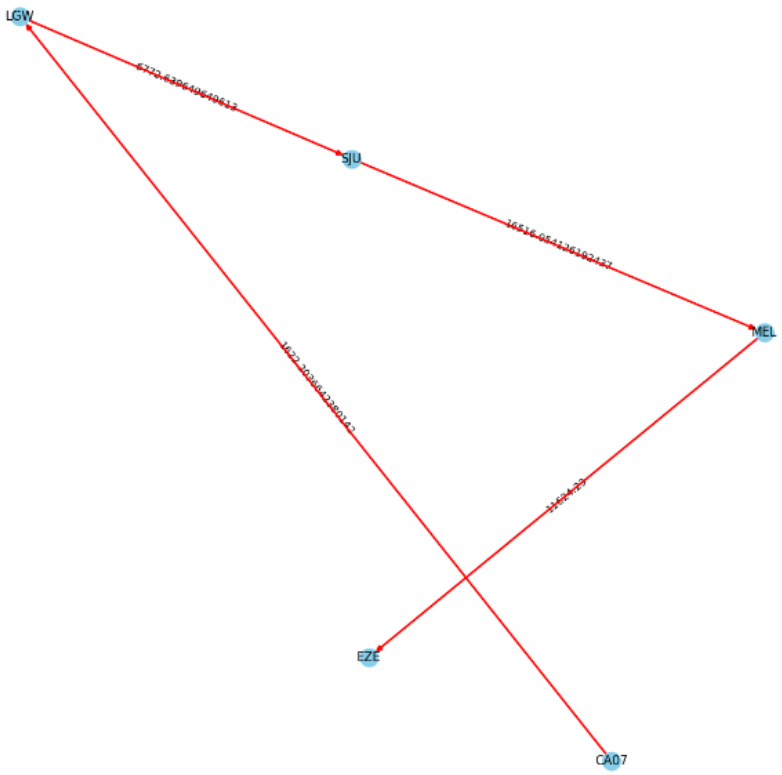
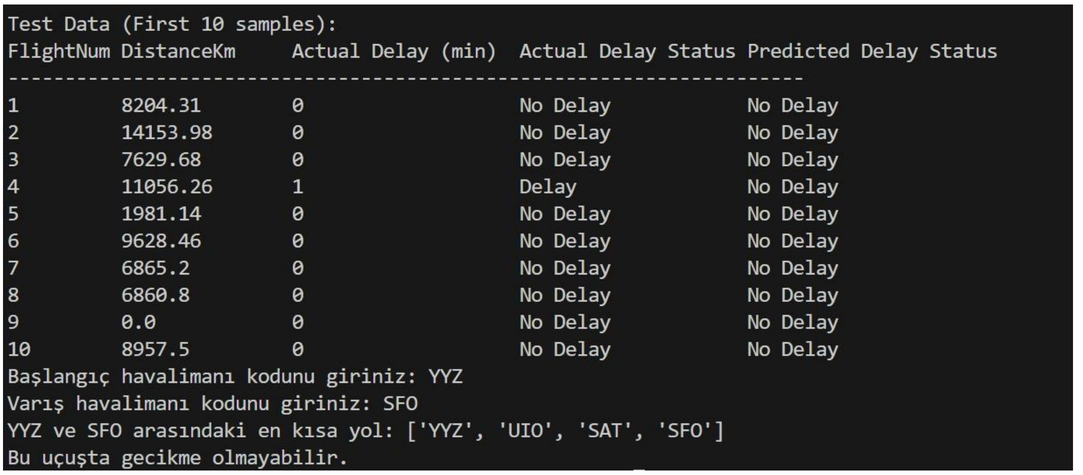


Figure 3. Graph plot output of the Linear Regression Model

- **Results:**
  - Preliminary findings suggest that there is a reasonable capacity for delay estimation that could be improved by adding new features or improving preprocessing procedures. Even if the MSE is high, this first method provides a starting point. To improve the prediction accuracy, more advanced models will be investigated in the future.

Logistic Regression Model:

- **Training and Testing:**
  - Created a new binary target variable, `DelayStatus`, indicating whether a flight was delayed (`FlightDelayMin` > 0).
  - Used `DistanceKilometers` and other features as predictors.
  - Split the data into training (80%) and testing (20%) sets.
  - Standardized the numerical features using `StandardScaler`.
  - Trained a logistic regression model and evaluated it using accuracy and classification reports.
- **Results:**
  - The logistic regression model showed a low accuracy in distinguishing between delayed and non-delayed flights. This suggests the need for further optimization and additional features to improve the model.



```
Test Data (First 10 samples):
FlightNum DistanceKm     Actual Delay (min)  Actual Delay Status Predicted Delay Status
----------------------------------------------------------------------
1         8204.31        0                   No Delay            No Delay
2         14153.98       0                   No Delay            No Delay
3         7629.68        0                   No Delay            No Delay
4         11056.26       1                   Delay               No Delay
5         1981.14        0                   No Delay            No Delay
6         9628.46        0                   No Delay            No Delay
7         6865.2         0                   No Delay            No Delay
8         6860.8         0                   No Delay            No Delay
9         0.0            0                   No Delay            No Delay
10        8957.5         0                   No Delay            No Delay
Başlangıç havalimanı kodunu giriniz: YYZ
Varış havalimanı kodunu giriniz: SFO
YYZ ve SFO arasındaki en kısa yol: ['YYZ', 'UIO', 'SAT', 'SFO']
Bu uçuşta gecikme olmayabilir.
```

Figure 4. Output of the Logistic Regression Model

- One potential reason for the low accuracy is the class imbalance in the dataset (i.e., significantly more non-delayed flights than delayed ones), which affects the model’s ability to generalize.
- Despite these limitations, the logistic regression model provides a baseline for evaluating future models and improvements.

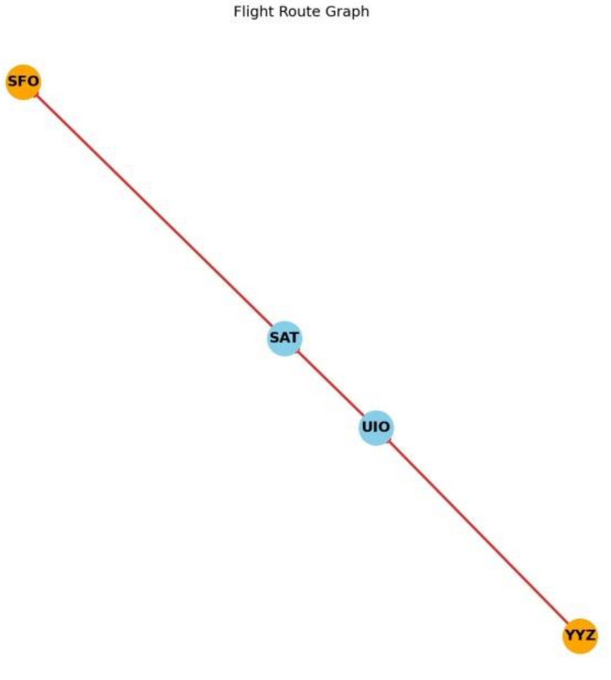


Figure 5. Graph plot output of the Linear Regression Model

Decision Tree Algorithm:

The goal is to predict the delay type based on several features, including weather conditions and previous delay information.

- **Dataset Preparation:**
  - **Target Variable:** The `FlightDelayType` column represents the delay types, which were converted to numeric values for model training:

    No Delay: 0
    Late Aircraft Delay: 1
    Weather Delay: 2
    NAS Delay: 3
    Security Delay: 4

  - **Features:** Extra features used for classification include:
    - `FlightDelay` (Binary: 0 or 1 indicating if there was a delay)
    - `FlightDelayMin` (Flight delay in minutes)

- **Training and Testing:**

  - The dataset was split into training (80%) and testing (20%) sets.
  - A `DecisionTreeClassifier` was trained with the training data. The model was configured with a maximum depth of 5 to prevent overfitting.

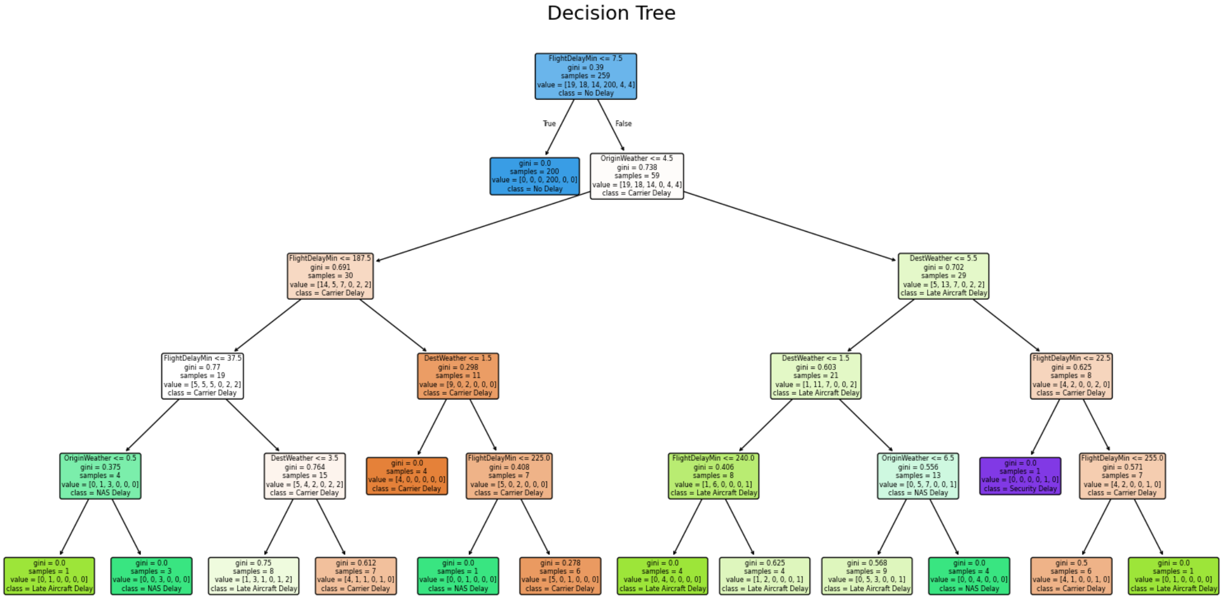
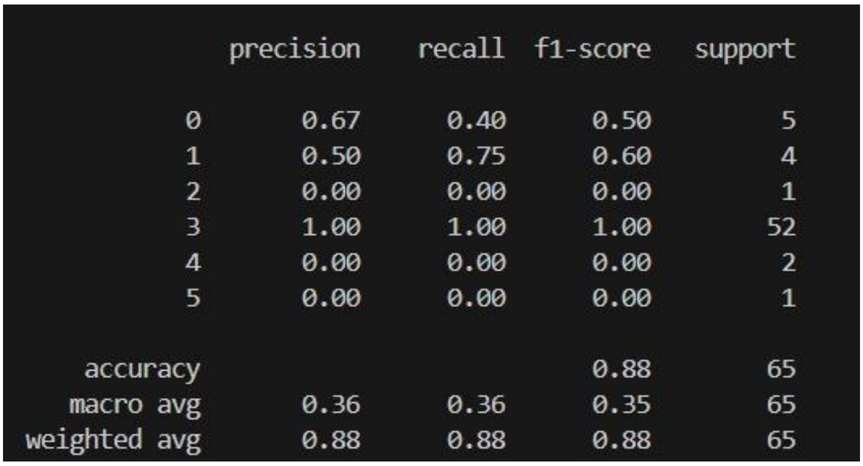

Decision Tree
FlightDelayMin <= 7.5 gini = 0.39 samples = 259 value = [19, 18, 14, 200, 4, 4] class = No Delay
True
False
gini = 0.0 samples = 200 value = [0, 0, 0, 200, 0, 0] class = No Delay
OriginWeather <= 4.5 gini = 0.738 samples = 59 value = [19, 18, 14, 0, 4, 4] class = Carrier Delay
FlightDelayMin <= 187.5 gini = 0.691 samples = 30 value = [14, 5, 7, 0, 2, 2] class = Carrier Delay
DestWeather <= 5.5 gini = 0.702 samples = 29 value = [5, 13, 7, 0, 2, 2] class = Late Aircraft Delay
FlightDelayMin <= 37.5 gini = 0.77 samples = 19 value = [5, 5, 5, 0, 2, 2] class = Carrier Delay
DestWeather <= 1.5 gini = 0.298 samples = 11 value = [9, 0, 2, 0, 0, 0] class = Carrier Delay
DestWeather <= 1.5 gini = 0.603 samples = 21 value = [1, 11, 7, 0, 0, 2] class = Late Aircraft Delay
FlightDelayMin <= 22.5 gini = 0.625 samples = 8 value = [4, 2, 0, 0, 2, 0] class = Carrier Delay
OriginWeather <= 0.5 gini = 0.375 samples = 4 value = [0, 1, 3, 0, 0, 0] class = NAS Delay
DestWeather <= 3.5 gini = 0.764 samples = 15 value = [5, 4, 2, 0, 2, 2] class = Carrier Delay
gini = 0.0 samples = 4 value = [4, 0, 0, 0, 0, 0] class = Carrier Delay
FlightDelayMin <= 225.0 gini = 0.408 samples = 7 value = [5, 0, 2, 0, 0, 0] class = Carrier Delay
FlightDelayMin <= 240.0 gini = 0.406 samples = 8 value = [1, 6, 0, 0, 0, 1] class = Late Aircraft Delay
OriginWeather <= 6.5 gini = 0.556 samples = 13 value = [0, 5, 7, 0, 0, 1] class = NAS Delay
gini = 0.0 samples = 1 value = [0, 0, 0, 0, 1, 0] class = Security Delay
FlightDelayMin <= 255.0 gini = 0.571 samples = 7 value = [4, 2, 0, 0, 1, 0] class = Carrier Delay
gini = 0.0 samples = 1 value = [0, 1, 0, 0, 0, 0] class = Late Aircraft Delay
gini = 0.0 samples = 3 value = [0, 0, 3, 0, 0, 0] class = NAS Delay
gini = 0.75 samples = 8 value = [1, 3, 1, 0, 1, 2] class = Late Aircraft Delay
gini = 0.612 samples = 7 value = [4, 1, 1, 0, 1, 0] class = Carrier Delay
gini = 0.0 samples = 1 value = [0, 0, 1, 0, 0, 0] class = NAS Delay
gini = 0.278 samples = 6 value = [5, 0, 1, 0, 0, 0] class = Carrier Delay
gini = 0.0 samples = 4 value = [0, 4, 0, 0, 0, 0] class = Late Aircraft Delay
gini = 0.625 samples = 4 value = [1, 2, 0, 0, 0, 1] class = Late Aircraft Delay
gini = 0.568 samples = 9 value = [0, 5, 3, 0, 0, 1] class = Late Aircraft Delay
gini = 0.0 samples = 4 value = [0, 0, 4, 0, 0, 0] class = NAS Delay
gini = 0.5 samples = 6 value = [4, 1, 0, 0, 1, 0] class = Carrier Delay
gini = 0.0 samples = 1 value = [0, 1, 0, 0, 0, 0] class = Late Aircraft Delay

Figure 6. Output from Decision Tree Algorithm

- **Results:**

  - **Accuracy:** The model achieved an accuracy of {{accuracy}} on the test set.
  - **Classification Report:** A classification report was generated, which included precision, recall, and F1-scores for each delay type. This report helps in understanding how well the model classifies each type of delay.



```
              precision    recall  f1-score   support

           0       0.67      0.40      0.50         5
           1       0.50      0.75      0.60         4
           2       0.00      0.00      0.00         1
           3       1.00      1.00      1.00        52
           4       0.00      0.00      0.00         2
           5       0.00      0.00      0.00         1

    accuracy                           0.88        65
   macro avg       0.36      0.36      0.35        65
weighted avg       0.88      0.88      0.88        65
```

Figure 7. Output of the Decision Tree Algorithm

- **Model Evaluation:**
  - The model was evaluated using the `accuracy_score` and `classification_report` metrics.

3. Challenges Encountered

Throughout the project, we encountered several challenges. One of the main difficulties was the inconsistent formatting in the dataset. For example, some numerical fields contained commas, which required a lot of time and effort to clean up and make usable. We also faced issues with missing or incomplete data for certain flights, which forced us to use methods like filling missing values with averages or placeholders to maintain the dataset's usability.

Another challenge was encoding the weather conditions, which are categorical values, into numerical representations that the model could understand. This task was harder than expected because it needed careful consideration to ensure the encoded values still represented their original meaning.

Handling the graph model for the flight network was also tricky due to the large size of the dataset. With thousands of connections between airports, building and querying the graph in a computationally efficient way was not easy. Lastly, when we trained the linear regression, logistic regression model and decision tree algorithm, we realized its predictions were not very accurate. This showed us that we needed to explore other machine learning models that could better handle the complexity of the data.

4. Future Plans

Model Improvement:

- As a team, we decided to focus on improving the models we've used so far. One of our main goals is to try out more advanced algorithms. We believe new methods can handle the dataset better and provide more accurate delay predictions.

Advanced Optimization

- We also decided to enhance our graph-based optimization. We want to extend the graph model to handle more complex flight scenarios, such as multi-leg journeys with layovers.

5. Conclusion

The foundation for machine learning and graph theory-based aircraft route optimization has been effectively established by this study. We hope to create a more dependable and effective system for anticipating delays and enhancing airline operations by resolving the issues that have arisen and putting the planned improvements into action.

6. References

- Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- NetworkX Developers. (2023). NetworkX Documentation. Retrieved from https://networkx.org/documentation/stable/
- Flights dataset: Retrieved from https://github.com/mdrilwan/datasets.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

210101112 - Gizem Erol
210101101 - Selvinaz Zeynep Kıyıkcı
200101011 - Zeynep Sude Kırmacı