Project Report: ECON AIR

Simple Solution for Flight Selection

Nethmee Perera
CSC 4520
Dec 4, 2023

Introduction

This project aims to provide a simple and effective solution for finding the most economically efficient flight for travel. The primary challenge addressed is the difficulty users face when trying to select the best flight based on various factors such as distance, ticket price, and duration.

Objective

The goal is to implement a solution using Python that uses a greedy algorithm and a basic sorting algorithm (Bubble Sort) to identify the economically best flight option.

Algorithm Overview

Greedy Algorithm

The greedy algorithm focuses to make locally optimal choice at each step for finding a global optimum.

In flight selection, the algorithm selects flights with the lowest cost-to-distance ratio until a specified total distance constraint is met.

```python
def greedy_flight(flights):
    total_distance = 0
    total_cost = 0

    for flight in flights:
        if total_distance + flight.distance <= 2000:
            total_distance += flight.distance
            total_cost += flight.ticket_price

    #  last selected flight == the best
    return flights[-1]
```

Bubble Sort

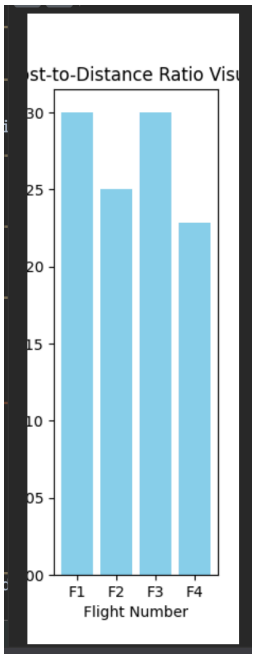
Bubble Sort is used to sort flights based on the cost-to-distance ratio.

```
def bubble_sort(flights):
    n = len(flights)

    for i in range(n - 1):
        for j in range(0, n - i - 1):
            ratio_j = flights[j].ticket_price / flights[j].distance
            ratio_j1 = flights[j + 1].ticket_price / flights[j + 1].distance

            if ratio_j > ratio_j1:
                flights[j], flights[j + 1] = flights[j + 1], flights[j]
```

The sorting is done in ascending order, making sure that the flight with the lowest ratio comes first.

Implementation

The implementation has done by creating Flight class to show flight information, using Bubble Sort to sort flight ratios, and implementing a greedy algorithm for flight selection. The matplotlib library is used for data visualization.



Code Structure

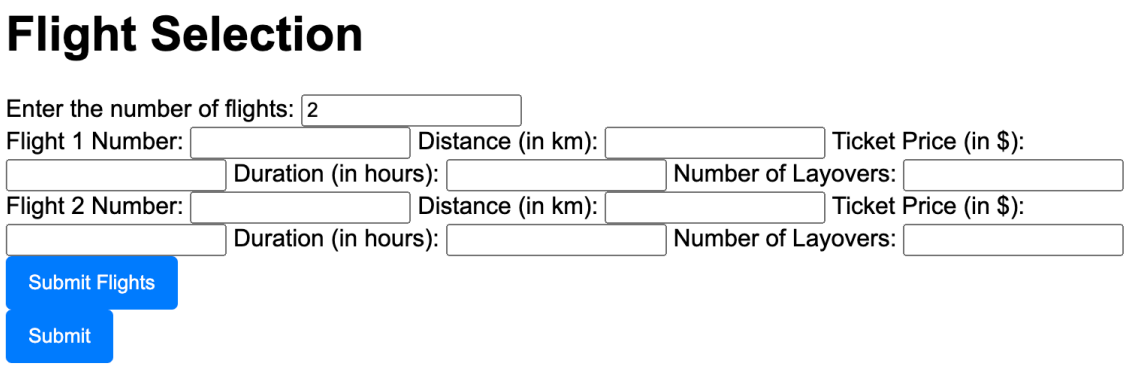
The project code is made in a simple way with short and clear variable names to improve readability.

Functions include bubble_sort for sorting, greedy_flight for the greedy algorithm, and visualize_data for data visualization.

User Interaction

The program prompts the user to input flight details, including flight number, distance, ticket price, and duration.

The user specifies the total distance constraint.



Results

The sorted data is displayed, showing flights in ascending order of cost-to-distance ratios.

The final selection of flights based on the greedy algorithm is presented, considering the total distance constraint.

Selected Flight:

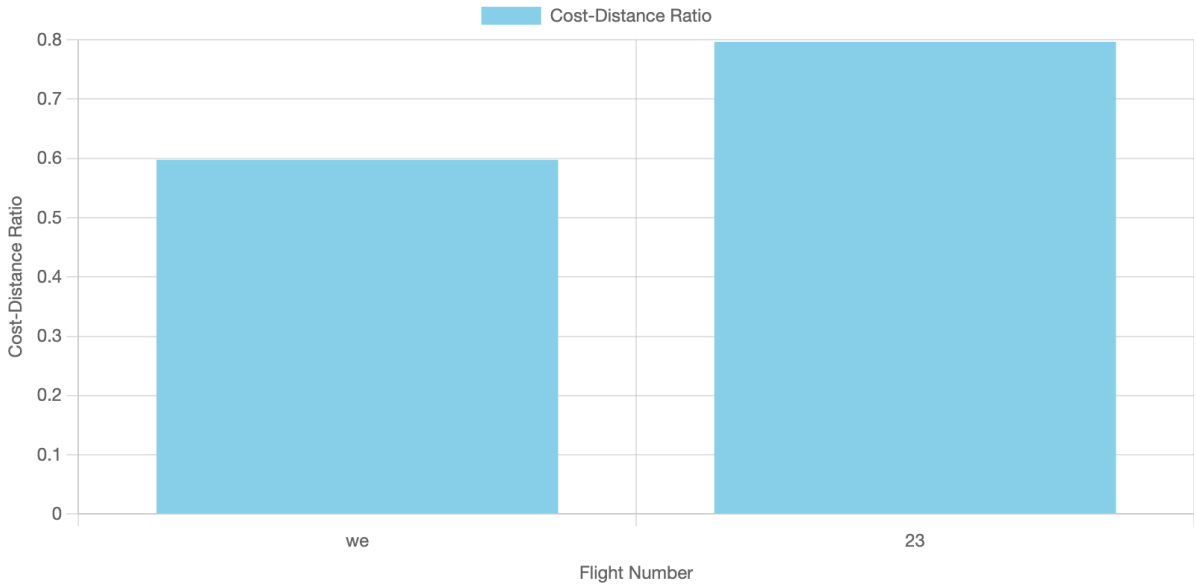
Flight Number: 23

Distance: 432 km

Ticket Price: $344

Duration: 32 hours

Layovers: 342




```
Sorted Data (Lowest Cost-to-Distance Ratio):
Flight us890: 0.1000
Flight us786: 0.1000

Selected Flight:
Flight Number: us786
Distance: 510 km
Ticket Price: $51.0
Duration: 2.0 hours
Layovers: 0
```

Conclusion

This project offers a user-friendly solution to the problem of finding the economically best flight. By using a greedy algorithm and a basic sorting technique, the code simplifies the decision-making process for travelers.

Future Improvements

Future improvements could include more factors such as layover duration and airline preferences. Think of turning it into a website using Flask and a Chrome extension. Then user does not have to enter information manually. Also there will be some features to add ratings, reviews, and airline options for an even smoother experience.

The algorithm and code could be optimized for larger datasets.

Acknowledgments

This project was completed as part of CSC 4520 - Design Analysis and Algorithm.

Special thanks to Prof. Shiraj for guidance and support.

Citations

Bubble Sort and Greedy Algorithm from a YouTube video lecture by Abdul Bari https://youtu.be/ARvQcqJ_-NY?si=He_VJsXhZWsK4t2J

Lecture notes

GitHub link : https://github.com/dudum98/ECON_AIR