Team Name:

PJN Programmers

Team Members:

- Parsh Patel
- Noah Lunberry
- Joel Samuel

Project Title:

Parking Ticket Prediction Tool

Problem:

What problem are we trying to solve? *We are trying to solve the problem of predicting the likelihood of receiving a parking ticket based on historical parking violation data.*

Motivation:

Why is this a problem? *When people result to illegally parking somewhere, they don't know the exact risk they are taking. If people knew the exact probability of getting a ticket when parking in a certain location, they would make better decisions about where to park.*

Features:

When do we know that we have solved the problem?

- *Users can input a location and receive a probability of getting a parking ticket there.*
- *The app provides a comparison of execution times between two different algorithms (BFS and Dijkstra's) for processing the data.*
- *The interface displays the probability, the computation time, and a visual representation of nearby violations.*

Data:

Public data set we will be using and the link to the public data set or **Schema of randomly generated data** *Parking Violations Issued in Washington DC in 2022. Link: https://opendata.dc.gov/datasets/DCGIS::parking-violations-issued-in-september-2022/about*

Tools:

Programming languages or any tools/frameworks we will be using

- ***Programming Languages****: Python*
- ***Tools/Frameworks****: Pandas, Matplotlib, Geopy, Geocodio API, Time module*

Visuals:

Wireframes/Sketches of the interface or the menu-driven program

Input Screen*: Fields for entering the user's location, and a choice to use either Djikstras or BFS.*

Result Screen*: Displays the probability of receiving a ticket, the time taken by each algorithm, and a map with marked violation spots.*

Strategy:

Preliminary algorithms or data structures you may want to implement and how would you represent the data

We will turn the data from the CSV file into a graph structure where each unique parking violation location is a node in our graph. For BFS the edges will represent simple connections between locations. For Dijkstra's Algorithm, edges will be weighted based on the geographical distances between locations. BFS will help explore all reachable violation locations and then Dijkstra's Algorithm will find the shortest path from the user's location to different violation hotspots, letting us compare the efficiency and effectiveness of both algorithms in predicting parking violations.

Distribution of Responsibility and Roles:

Who is responsible for what?

- *Parsh - Implementing the BFS algorithm into our data.*
- *Joel - Implementing the Dijkstra's algorithm into our data.*
- *Noah - General functionality of the app: front end, data cleaning and preprocessing, etc.*

References:

- *Course materials and textbooks.*
- *Documentation fromPandas, Matplotlib, Geopy, Geocodio API.*
- *Websites with relevant datasets: Data.gov, Corgis Datasets Project.*