Egadarshan S
Aravinthan S
III yr AI-DS “B”

School Bus Routing Optimization

1. Project Overview

This project optimizes school bus routes to reduce transportation costs and improve student safety. The web app is built using **Streamlit**, and it leverages **Google's OR-Tools** for route optimization and **Optuna** for hyperparameter tuning.
The optimization is based on minimizing total travel distance while respecting constraints like the number of buses and student capacity per bus.

---

2. Technologies Used

- **Streamlit** (for Web UI)
- **OR-Tools** (for solving Vehicle Routing Problem)
- **Optuna** (for parameter optimization)
- **Python**, **NumPy**, **FPDF**

---

3. User Inputs

Users control the optimization using these adjustable settings in the sidebar:

- **Min Buses**: Minimum number of buses to consider for optimization
- **Max Buses**: Maximum number of buses to consider
- **Min Capacity**: Minimum bus capacity (e.g., 40 students)
- **Max Capacity**: Maximum bus capacity (e.g., 60 students)
- **Optimization Trials**: Number of Optuna iterations to try different combinations

---

4. Sample Locations (Chennai, India)

The app uses sample coordinates for major areas in Chennai to simulate pickup points:

- T. Nagar (Depot)
- Adyar
- Anna Nagar
- Guindy
- Mylapore
- Kilpauk

---

5. Optimization Logic

- A distance matrix is created using the coordinates of all locations.
- **OR-Tools** solves the **Vehicle Routing Problem (VRP)** with:
    - A depot (start and end location for all buses)
    - Student demand per stop
    - Capacity limits for each bus
- **Optuna** is used to try different bus counts and capacities and select the best performing configuration.

---

6. Optimization Techniques Used

🔧 **Google OR-Tools:**

- Solves the VRP using **local search** methods:
    - **Tabu Search**
    - **Guided Local Search**
- Ensures every location is visited once, without exceeding bus capacity.

🎯 **Optuna:**

- Uses **Bayesian Optimization** via the **Tree-structured Parzen Estimator (TPE)**.
- Automatically tries different combinations (bus count, capacity).
- Evaluates the **total distance** and finds the configuration that minimizes it.

---

7. Detailed Optimization Workflow

1. User sets slider ranges for buses and capacities (e.g., 2–5 buses, 40–60 capacity).
2. Optuna runs multiple trials (e.g., 20), selecting random values within these limits.
3. For each trial:
    - A set number of buses and capacity are passed to OR-Tools.
    - OR-Tools generates optimized routes using local search algorithms.
4. Total distance is calculated for the trial.
5. The best result (minimal distance) is displayed to the user.

---

8. Output

- 🚍 **Best total distance** to transport all students
- 🚌 **Optimized number of buses** and **capacity**
- 🗺️ **Route map** showing all locations