AI-Based Train Scheduling Optimization using Passenger Flow and GTFS Data

Project Goal:

Optimize train scheduling (especially number of trains) based on observed passenger flow during peak and off-peak times using AI search techniques (e.g., local search).

Step-by-Step Implementation Plan

Step 1: Understand and Preprocess the Data

Input Data:

- Passenger data: Total "ons" and "offs" by stop, split by:
  - stop_name, parent_station, day_type (weekday/weekend), time_period (e.g., AM_RUSH, PM_RUSH, etc.)
- GTFS data:
  - stop_times.txt, trips.txt, calendar.txt - use these to compute train slots per time period.

Tasks:

- Load both datasets into Pandas.
- Create a structure like:

  stop | day_type | time_period | total_ons | total_offs

Step 2: Estimate Available Train Slots

From GTFS stop_times, calculate how many 15-min time slots exist per day for the Orange Line.

- Count number of trips per time slot
- Example: 21 slots * 4 trains per hour (15-min gap) = max 84 trains per day
- Split this count by day_type and time_period to create:

  day_type | time_period | max_slots

Step 3: Define the Optimization Problem

You want to minimize passenger wait or overload by optimizing train allocation per time block.

Constraints:

- # of trains <= available slots per time period

- Higher demand periods (e.g., AM rush) more trains

Objective Function:

Define a cost function like:

cost = (predicted load per train - train capacity)^2

Step 4: Use Local Search to Optimize

Approach:

- Start with a random or even allocation of trains per time slot.
- Use hill climbing or simulated annealing to explore better schedules.
- At each step:
  - Slightly adjust the number of trains in a time period.
  - Recalculate cost.
  - Accept the change if cost improves (or probabilistically if using SA).

Step 5: Classify Peak vs Off-Peak

Use the dataset to define:

- Peak hours (e.g., AM_RUSH, PM_RUSH)
- Off-peak hours (e.g., MIDDAY, EVENING)

Analyze average ons/offs in each time period and use it to prioritize train frequency.

Example Output You Can Show:

A table like:

Time Period	Day Type	Demand	Allocated Trains	Avg Load/Train	Overload Penalty
AM_RUSH	Weekday	5000	12	417	Low
EVENING	Weekday	1800	4	450	Moderate

Or a bar chart showing:

- Actual demand vs capacity across time periods
- Optimized train counts

AI Concepts Youll Cover:

- Local Search (Hill Climbing / Simulated Annealing)

- Constraint Satisfaction (trains <= slots)

- Objective function modeling

- Optimization under real-world constraints (peak hour demand)