Dynamic Pricing for Urban Parking Lots

Project Overview

Urban parking is a constrained resource often plagued by inefficiencies due to static pricing.
To tackle this, the project implements a real-time dynamic pricing engine for 14 parking lots using data such as occupancy,
queue length, traffic, and competition. Models are built from scratch using numpy, pandas, and Pathway to simulate intelligent,
adaptive pricing systems that optimize parking space utilization.

Tech Stack Used

- Python: Core logic
- NumPy: Vectorized computations
- Pandas: Data processing
- Pathway: Real-time data ingestion and transformation
- Bokeh: Visualization of price over time
- Google Colab: Development environment
- Mermaid: Architecture diagram

Model Summary

Model 1 (Baseline Linear Model):
  - P_t+1 = P_t + alpha * (Occupancy / Capacity)

Model 2 (Demand-Based Model):
  - Demand = alpha * (Occupancy/Capacity) + beta * Queue - gamma * Traffic + delta * SpecialDay + epsilon * VehicleTypeWeight
  - P_t = BasePrice * (1 + lambda * NormalizedDemand)

Model 3 (Competitive Model):
  - Factors in prices of nearby lots and location
  - May reroute vehicles or reduce price to stay competitive

Dynamic Pricing for Urban Parking Lots

Architecture & Workflow

1. Data Ingestion:
   - Uses Pathway to simulate real-time ingestion of .csv files.

2. Feature Engineering:
   - Extract features: occupancy, queue, vehicle type, etc.

3. Model Selection:
   - Apply Model 1, 2 or 3 depending on design.

4. Price Prediction:
   - Real-time pricing output is produced and bounded.

5. Visualization:
   - Interactive line plots using Bokeh to show price evolution.

6. Optional Rerouting:
   - Suggests nearby lots if current is full and cheaper options are available.