Introduction

This project tackles the challenge of static pricing in urban parking systems. Using real-time data and basic ML modeling with Pandas and Numpy, we design a dynamic pricing engine for 14 parking spaces to improve utilization.

Objective

To build a real-time dynamic pricing model that adjusts based on:

- Occupancy levels
- Queue lengths
- Nearby traffic conditions
- Special days
- Vehicle types
- Competitor parking prices

Data Overview

- 73 days of data
- 14 parking lots
- 18 time steps/day (every 30 mins from 8:00 AM to 4:30 PM)
- Features: Occupancy, Capacity, QueueLength, VehicleType, TrafficConditionNearby, IsSpecialDay, GPS coordinates

Preprocessing Steps

- Fixed malformed date/time entries
- Combined date & time into a proper timestamp
- Filled missing values with logical defaults
- Standardized vehicle and traffic types
- Computed occupancy ratio
- Grouped data by parking lot and timestamp

Models Implemented

Model 1: Baseline Linear Pricing

Formula:

Price_{t+1} = Price_t + alpha * (Occupancy / Capacity)

- Smooth and intuitive
- Clipped between $5 and $20
- Used as the reference model

Model 2: Demand-Based Pricing

Demand Function:

Demand = A*(OccupancyRatio) + B*(QueueLength) - C*(TrafficScore) + D*(IsSpecialDay) + E*(VehicleWeight)

- Normalized demand per lot
- Final price:

Price = BasePrice * (1 + lambda * NormalizedDemand)

- Parameters: A=4, B=1, C=1.5, D=2, E=1, lambda=0.5
- Constrained between $5 and $20

Model 3: Competitive Pricing Model

- Identified nearby lots (within 0.5 km using Haversine distance)
- Adjusted price based on competitor pricing:
    - If full and neighbors cheaper: reduce price
    - If neighbors more expensive: slight increase

Real-Time Simulation (Pathway)

- Implemented schema and data pipeline
- Streaming setup with timestamp-based updates
- Connected pricing logic to live data

Visualizations (Bokeh)

- Real-time price updates for each parking lot
- Comparison of pricing models
- Insights into demand-driven adjustments

Assumptions

- Base price = $10
- Minimum price = $5, Maximum = $20

- Smooth variations preferred over abrupt changes
- Vehicle & traffic weights fixed based on heuristics

Conclusion

This system uses historical and real-time data to intelligently update parking prices. It demonstrates how simple economic and competitive strategies can be implemented in real-world applications using basic data tools.