Dynamic Parking Price Optimization Using Streaming and Bokeh

👩‍💻 Submitted by: SYED SEEFATUL HAQUE

📅 Date: 03-07-2025

**1. Abstract**

This project presents a real-time parking price optimization system using streaming data. We developed three pricing models—Baseline, Demand-Based, and Competitive—and visualized them with Bokeh for interactive analysis. Using Pathway for real-time streaming, our solution simulates how urban parking lots can dynamically adjust pricing to maximize efficiency and revenue.

**2. Introduction**

Urban parking is a critical challenge due to fluctuating demand, special events, and uneven distribution of traffic. Static pricing often leads to inefficient use of space and user dissatisfaction. This project implements dynamic pricing strategies using real-time data to improve parking lot utilization.

**3. Dataset Overview**

We used a dataset simulating real-time streaming of parking lots with the following columns:  
  
- ID: Unique parking lot identifier  
- SystemCodeNumber: System-assigned code  
- Capacity: Total parking spaces  
- Occupancy: Number of occupied spaces  
- Latitude, Longitude: Geolocation  
- QueueLength: Vehicles waiting  
- TrafficConditionNearby: Traffic descriptor (low, average, high)  
- VehicleType: Car, Bike, Truck  
- IsSpecialDay: 1 if special day  
- LastUpdatedDate, Time: Timestamps

**4. Model 1 – Baseline Fixed Pricing**

This model assigns a constant price to all lots (e.g., ₹10 per hour). It serves as a reference for comparison with dynamic approaches.  
  
- Logic: Fixed price = 10  
- Pros: Simple  
- Cons: Ignores real-time conditions

**5. Model 2 – Demand-Based Dynamic Pricing**

Prices are dynamically adjusted based on:  
- Occupancy Ratio  
- Queue Length  
- Traffic conditions  
- Special day flag  
- Vehicle type

Equation:

price = α \* (Occupancy / Capacity) + β \* QueueLength - γ \* Traffic + δ \* SpecialDay + ε \* VehicleWeight

**6. Model 3 – Competitive Pricing Strategy**

This model uses geolocation to price a lot based on nearby competitors.  
  
- Calculates average Model 2 price of nearby lots within 0.01° radius  
- Adds 5% markup on the local average  
- Falls back to Model 2 price if no nearby lots

**7. Visualizations**

We used Bokeh for real-time visualization of pricing across lots. Charts were refreshed using `show\_realtime\_plot()` for a live dashboard feel.

**8. Comparison Table**

Sample comparison (customize with real output):

|  |  |  |  |
| --- | --- | --- | --- |
| Lot ID | Model 1 Price | Model 2 Price | Model 3 Price |
| 0 | ₹10 | ₹13.7 | ₹14.2 |
| 1 | ₹10 | ₹14.1 | ₹14.5 |

**9. Conclusion**

Model 2 and 3 significantly outperform the baseline by adapting to real-time and local conditions. Model 3 offers smarter pricing by factoring in competition. Future extensions could include:  
- Integration with real-time traffic APIs  
- Dynamic re-learning of pricing coefficients  
- Mobile alerts for users