Report: Dynamic Pricing for Urban Parking Lots

Capstone Project | Summer Analytics 2025

Consulting & Analytics Club | IIT Guwahati

1. Abstract

This project explores dynamic pricing for urban parking lots using real time data streams and machine learning inspired logic. Static pricing often leads to underutilization or overcrowding. Our objective was to design a multi stage pricing engine for 14 parking spaces using features like occupancy, queue length, traffic, special days, and vehicle types. We built two models from scratch using only pandas and numpy and visualized pricing behaviors using Bokeh. This report details the data pipeline, demand based pricing logic, assumptions and insights derived from visualization.

2. Dataset Overview

The dataset contains usage data from 14 parking lots over 73 days, sampled every 30 minutes (18 time intervals per day).

Key Features:

- SystemCodeNumber: Unique ID for each parking lot
- Latitude / Longitude: Location coordinates
- Capacity: Maximum vehicle slots
- Occupancy: Number of vehicles currently parked
- QueueLength: Vehicles waiting to enter
- VehicleType: car, bike, truck, or cycle
- TrafficConditionNearby: low / average / high

- IsSpecialDay: Indicator for holidays or events

---

3. Data Preprocessing

- Merged LastUpdatedDate and LastUpdatedTime into a single Timestamp
- Converted categorical variables to numerical scales:
    - TrafficConditionNearby: low → 1, average → 2, high → 3
    - VehicleType mapped to weight: car = 1.0, bike = 0.6, truck = 1.5, cycle = 0.3
- Created OccupancyRate = Occupancy / Capacity for normalized demand computation
- Sorted the dataset by SystemCodeNumber and Timestamp for real time simulation

---

4. Model 1: Linear Occupancy-Based Pricing

Formula:

P(t+1)=Pt +α□(Occupancy / Capacity)

- Base price: $10
- Coefficient \alpha = 2.0
- Price increases smoothly as occupancy rises
- Per-lot sequential simulation over time
- Used as a baseline model

---

5. Model 2: Demand-Based Pricing

Demand Function

Demand=α□(CapacityOccupancy)+β□QueueLength−γ□TrafficLevel+ e□IsSpecialDay+ e□VehicleTypeWeight

Coefficients:

Parameter	Value
a (occupancy)	2.0
B (queue)	0.5
V (traffic)	1.0
§ (special day)	2.0
€ (vehicle)	1.0
^ (demand → price)	0.8

Pricing Formula:

Pricing Formula:

Price t =BasePrice□(1+λ□NormalizedDemand)

Price Bounds:

- Minimum: $5.00 (0.5x base)
- Maximum: $20.00 (2x base)

---

7. Assumptions

- Queue length and special events directly increase demand
- Traffic congestion penalizes demand
- Trucks (1.5x) generate more pressure on pricing than bikes or cycles
- All features affect price only through demand, not directly
- Real-time response assumes perfect data ingestion without API delays

---

8. Visualizations

Interactive Bokeh dashboards were created for:

- Model 1 Pricing over time (based on occupancy)
- Model 2 Pricing over time (based on normalized demand)
- Users can select parking lots via dropdown to explore pricing behavior

These visualizations clearly justify:

- The linearity of Model 1
- The intelligent adjustments of Model 2 during special events, queues, and peak hours

---

9. Conclusion

This project demonstrates how intelligent, real time pricing can be built from scratch using simple Python tools. The demand based pricing model responds flexibly to evolving conditions, improving parking lot efficiency. Future extensions could include reinforcement learning, location-based competition and dynamic rerouting suggestions.