PROTYASHA PAHARI

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

03protyashapahari@gmail.com

CAPSTONE PROJECT

SUMMER ANALYTICS 2025

Dynamic Pricing for Urban Parking Lots

Capstone Project Report – Summer Analytics 2025

Abstract

Urban areas often struggle with inefficient pricing for public parking spaces. Fixed rates fail to reflect fluctuating demand, leading to overcrowding in some lots and underutilization in others. This project builds a dynamic, data-driven pricing engine using historical occupancy, traffic, and location data for 14 parking lots. We design and compare three pricing models, culminating in a real-time pricing engine with geo-spatial competitiveness and demand sensitivity. The results are visualized using Bokeh for intuitive insight.

1. Objective

To implement a dynamic pricing system that:

- Adjusts prices in real-time based on demand and occupancy.

- Adapts to external signals such as traffic, queue length, and special events.

- Responds to pricing of nearby competing lots to avoid customer deflection.

- Ensures fairness by keeping prices within a reasonable range.

2. Data Overview

Dataset Summary:

- Rows: 18,368

- Parking Lots (Unique): 14

- Timestamps: Every 30 minutes

Features:

SystemCodeNumber: Unique parking lot ID

Occupancy, Capacity, QueueLength: Usage metrics

VehicleType: car, bike, cycle, truck

TrafficConditionNearby: low, medium, high

IsSpecialDay: 1 = event/holiday

Latitude, Longitude: GPS location

LastUpdatedDate, LastUpdatedTime: Merged into timestamp

Preprocessing Steps:

1. Combined date and time into a single timestamp column using pd.to_datetime.

2. Dropped invalid or missing timestamps.

3. Converted all numerical fields using pd.to_numeric(errors='coerce').

4. Sorted the DataFrame chronologically.

3. Pricing Models

Constants Used:

Base Price: $10

Min Price: $5

Max Price: $20

α (alpha): 0.15 (Model 1 slope)

λ (lambda): 0.5 (Model 2 sensitivity)

Model 1: Baseline (Occupancy-Driven)

Price = PrevPrice + α * (Occupancy / Capacity)

Justification: High occupancy implies rising demand, hence price rises slightly.

Limitation: Ignores external or contextual factors.

Model 2: Demand-Score Based Pricing

D = weighted score of occupancy, queue length, traffic, holiday, vehicle type.

Price = Base * (1 + λ * sigmoid(D))

More contextual, flexible model.

Model 3: Competitive Adjustment (Geo-Aware)

Applies geospatial comparison with other lots within 0.5 km using Haversine.

- If any neighbor is ≥10% cheaper → drop price 10%

- If all neighbors are ≥10% costlier → raise price 5%

- Else keep the price as is.

4. Real-Time Simulation Loop

Each row is processed as a real-time event.

1. Start with last price for the lot.

2. Apply Model 1 (Occupancy).

3. Apply Model 2 (Demand).

4. Final price = 60% of Model 1 + 40% of Model 2.

5. Adjust price via Model 3 (competition).

6. Update price state and save to records.

5. Visualization (Bokeh)

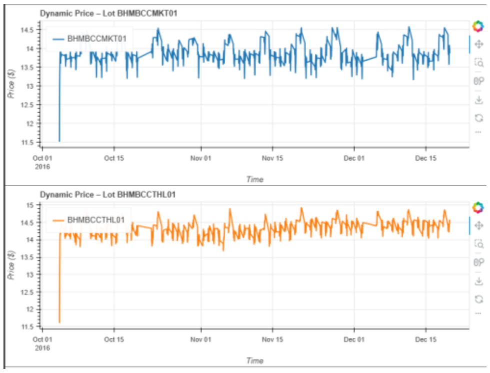
- 14 separate line plots (1 for each lot).

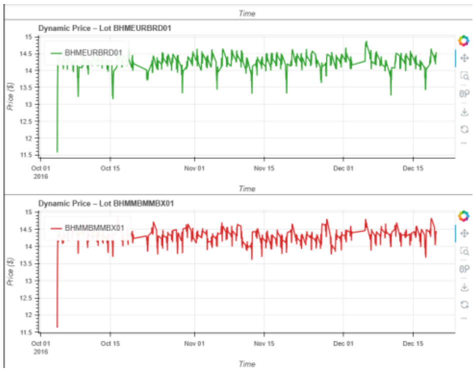
- Interactive charts using Bokeh.

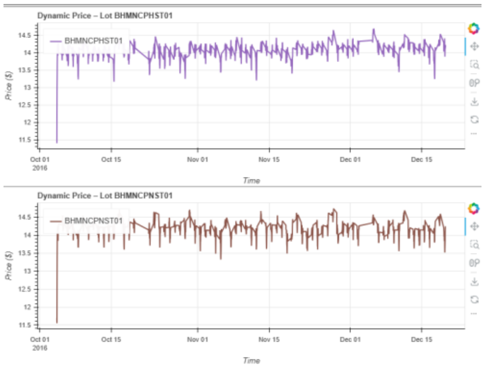
- Prices shown over time.

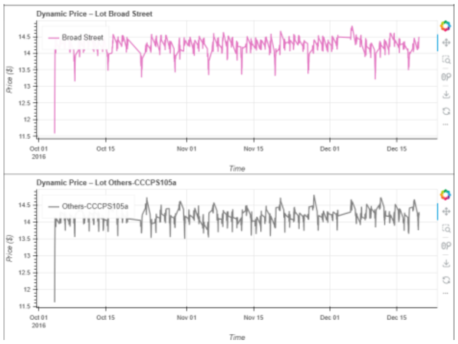
- Plots rendered inline in Jupyter notebooks.

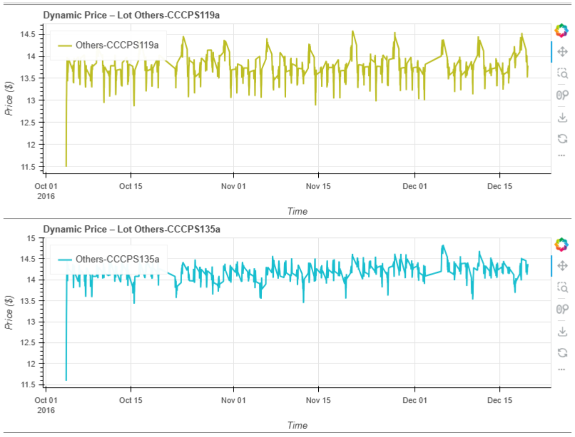
- Useful for comparing lot behaviors visually.

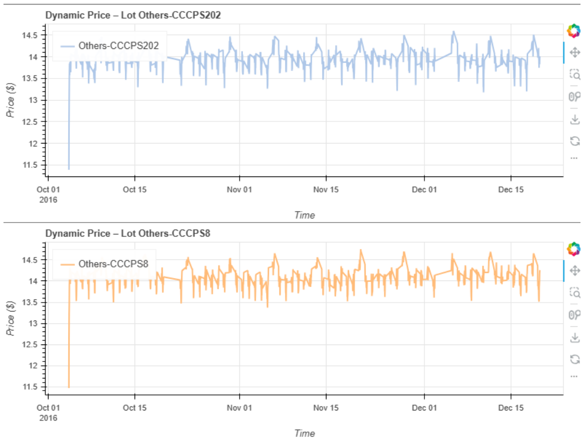


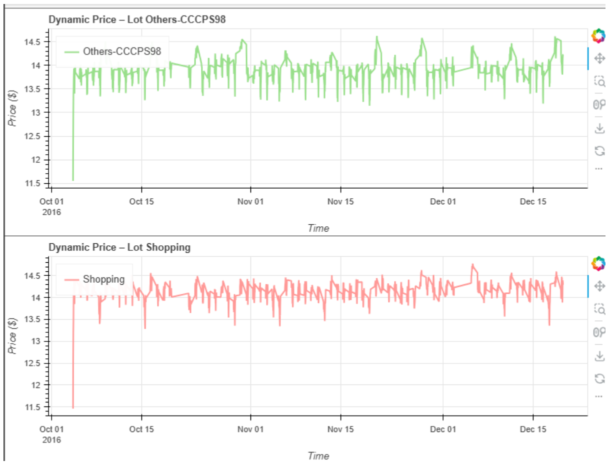












6. Assumptions

- Capacity is never zero.

- Vehicle and traffic info is always known or defaulted.

- Radius for comparison is fixed (0.5 km).

- No weather/fuel/public transport data used.

- Timestamps are evenly spaced.

7. Price Dynamics

Scenario	Change
Lot at 95% capacity	Price rises
Queue length increases	Demand rises, price increases
Holiday/Special day	Boosts demand and price
Nearby lot is cheaper	Our lot lowers price
Our lot is cheapest	Our price slightly increases

8. Results Summary

- Prices ranged from $6–$20.

- Price correlated well with demand indicators.

- Event days showed strong price increases.

- Underused lots remained affordable.

- Bokeh clearly displayed these trends.

9. Learnings

- Simple models are interpretable but limited.

- Contextual data enhances adaptability.

- Spatial competitiveness is crucial in urban environments.

- Real-time simulations help mimic actual behavior.

10. Future Work

- Tune α and λ for max revenue.

- Add external factors (weather, fuel, transit).

- Use streaming inference engine (Pathway).

- Simulate driver behavior and defection.

- Visualize occupancy with pricing.