Dynamic Pricing for Urban Parking Lots

Capstone Project – Summer Analytics 2025
Team: Akshit Gupta (Solo Submission)
Host: Consulting & Analytics Club × Pathway

Problem Statement

Urban parking spaces are limited and highly dynamic in terms of demand. Static pricing often leads to either overcrowding or underutilization. The objective is to create a real-time, intelligent pricing engine that adapts dynamically based on demand, competition, traffic conditions, and vehicle types using only NumPy, Pandas, and Pathway.

Project Goals

- Simulate real-time pricing for 14 parking spaces using data streams
- Create 3 pricing models of increasing complexity
- Visualize results in real-time using Bokeh plots
- Ensure smooth and explainable price adjustments

Models Implemented

Model 1: Baseline Linear Pricing

Formula:

Pricet+1=10+α · (OccupancyCapacity)\text{Price}_{t+1} = 10 + \alpha \cdot \left(\frac{\text{Occupancy}}{\text{Capacity}} \right)

- Directly scales price with occupancy
- Simple and intuitive
- Acts as a benchmark

Model 2: Demand-Based Pricing

Demand Function:

Demand=α · (OccupancyCapacity)+β · QueueLength−γ · Traffic+δ · SpecialDay+ε · VehicleTypeWeight\text{Demand} = \alpha \cdot \left(\frac{\text{Occupancy}}{\text{Capacity}} \right) + \beta \cdot \text{QueueLength} - \gamma \cdot \text{Traffic} + \delta \cdot \text{SpecialDay} + \varepsilon \cdot \text{VehicleTypeWeight}

Price:

Price=10 · (1+λ · NormalizedDemand)\text{Price} = 10 \cdot (1 + \lambda \cdot \text{NormalizedDemand})

- Captures real-world conditions like traffic, queues, and holidays
- Smooth bounded pricing (between $5 and $20)
- Vehicle-specific weighting

Model 3: Competitive Pricing (Location-Based)

- Uses haversine distance to calculate proximity between lots
- Adjusts price based on average nearby prices within 1km
- Logic:
  - If nearby lots are cheaper → reduce price or reroute
  - If nearby lots are expensive → increase price

Technologies Used

- Python
- Pandas, NumPy
- Pathway (Real-time stream processing)
- Bokeh (for visualizations)
- Jupyter / Google Colab (Development)

Data Pipeline (Using Pathway)

1. Load cleaned parking_stream.csv
2. Stream it in real-time (1000 rows/sec)
3. Process & enrich with timestamp, vehicle, and geo data
4. Apply pricing models in separate flows
5. Visualize each model's output with Bokeh

Visual Output

The final output includes:

- Daily average prices for each model
- 3 interactive Bokeh plots (one per model)
- Time-series price changes per day
- Real-time responsiveness to input changes

Deliverables

- dynamic_pricing.ipynb: notebook with all 3 models
- parking_stream.csv: processed dataset used for streaming
- README.md: full documentation
- report.pdf: (this file)
- GitHub Repository: ✅ Public and complete

Key Takeaways

- Dynamic pricing requires balancing fairness with profitability
- Location-aware strategies outperform static ones in real-time systems

Real-time processing using Pathway allows smooth simulation of streaming scenarios

---

Conclusion

This project demonstrated how economic logic, real-time processing, and simple ML-inspired models can optimize parking systems at scale. The implemented system is modular, efficient, and can be extended to live city-level deployments.
