Dynamic Pricing for Urban Parking Lots

Capstone Project – Summer Analytics 2025

Debasmita Saha
sahadebasmita2002@gmail.com
+91 9330386551

1. Introduction

Urban parking lots experience fluctuations in demand based on time of day, traffic, vehicle type, and special events. Static pricing leads to inefficiencies like underutilization or over-congestion. This project implements a real-time dynamic pricing engine for 14 urban parking lots using only `NumPy`, `Pandas`, and `Pathway` for real-time streaming.

2. Objectives

- Implement a dynamic, explainable, and smooth pricing model based on:
  - Occupancy rate
  - Queue length
  - Nearby traffic congestion
  - Special events or holidays
  - Vehicle type
  - Competitor pricing
- Build and evaluate three progressively complex pricing models:
  - **Model 1:** Baseline linear model using occupancy
  - **Model 2:** Demand-based function incorporating multiple features
  - **Model 3:** Competitive pricing model using location intelligence
- Simulate real-time pricing predictions using the Pathway framework
- Visualize pricing and demand trends over time with interactive Bokeh charts
- Ensure smooth, interpretable price variations (e.g., bounded between 0.5x and 2x base price)
- Optional: Implement rerouting logic for vehicles when parking lots are full

3. Dataset Overview

- **Time Span:** 73 days
- **Time Resolution:** 30-minute intervals, 18 time steps per day (from 8:00 AM to 4:30 PM)
- **Number of Parking Spaces:** 14
- **Collected Features:**
  - **Parking Lot Characteristics:** Occupancy, capacity, queue length
  - **Vehicle Information:** Type of vehicle (car, bike, truck)
  - **Environmental Factors:** Nearby traffic congestion, special day indicators
  - **Location Data:** Latitude and longitude of each lot (used for computing proximity and competitive pricing)

Each record corresponds to a snapshot in time for each parking lot. Demand and congestion fluctuate throughout the day, reflecting real-world urban patterns.

4. Model Descriptions

This section outlines three pricing models of increasing complexity. Each model is built from scratch using basic economic logic, mathematical formulations, and smooth bounded pricing behavior.

Model 1: Baseline Linear Model

Objective: Create a reference model where price increases linearly with occupancy.

Formula:

$$Price_{t+1} = Price_t + \alpha \cdot \left(\frac{Occupancy}{Capacity}\right)$$

Notes:

- Starts from a base price of \$10.
- The parameter α controls the sensitivity of pricing to occupancy (e.g., $\alpha = 2$).
- Linear increase ensures price reflects load but may lack responsiveness to external events.
- This model serves as a benchmark for evaluating more advanced approaches.

Model 2: Demand-Based Pricing Model

Objective: Incorporate multiple real-world features into a normalized demand function to adjust prices more intelligently.

Demand Function:

$$Demand = \alpha \cdot \left(\frac{Occupancy}{Capacity}\right) + \beta \cdot QueueLength - \gamma \cdot Traffic + \delta \cdot IsSpecialDay + \epsilon \cdot VehicleTypeWeight$$

Pricing Rule:

$$Price = BasePrice \cdot (1 + \lambda \cdot NormalizedDemand)$$

Implementation Highlights:

- Demand is computed at every time step and normalized to avoid volatility.
- Prices are constrained between 0.5x and 2x the base price.
- Vehicle types are weighted: for example, trucks have higher weights than bikes.
- More responsive and context-aware than Model 1.

Model 3: Competitive Pricing Model

Objective: Simulate market-driven behavior by incorporating prices and load of nearby parking lots.

Approach:

- Compute distance between lots using the Haversine formula.
- Identify nearby lots within a 0.5 km radius.
- Monitor competitor pricing and occupancy to adapt own strategy.

Example Logic:

```
if nearby_price_avg < current_price and occupancy == capacity:
    price = nearby_price_avg * 0.95
elif nearby_price_avg > current_price:
    price += price * 0.05
```

Benefits:

- Enables underutilized lots to attract demand through price competition.
- Prevents overcrowding by suggesting rerouting if nearby lots have lower prices.
- Simulates real-world economic behavior in urban contexts.

5. Real-Time Simulation (Pathway)

Objective: Implement real-time pricing behavior using a simulated streaming environment.

Implementation:

- **Streaming Engine:** Pathway framework used for real-time ingestion, transformation, and emission of pricing data.
- **Data Flow:**
  - Ingest rows sequentially by timestamp using delayed simulation.
  - Apply transformations: feature engineering, demand computation, and pricing updates.
  - Emit new prices continuously via a user-defined pricing function.
- **Key Function:** `apply_model()` used as a hook to insert pricing logic within Pathway's stream.

Outcome: Allows tracking of how pricing decisions evolve over time and in response to dynamic demand signals.

6. Visualization (Bokeh)

Objective: Present intuitive, real-time visualizations of pricing behavior.
Tools Used: Bokeh (Python-based interactive visualization library)
Key Visuals:

- **Line Charts:** Price over time for each parking space
- **Competitor Comparison:** Overlay of own price vs average nearby lot prices
- **Dynamic Refresh:** Periodic updates as new data is ingested

Implementation:

- Real-time updates managed via `ColumnDataSource`, `curdoc()`, and periodic callbacks.
- Plots include legends, axes labels, and time-stamped x-axes for interpretation.

Outcome: Enhances interpretability of the pricing engine and provides business-level insight into real-time dynamics.

7. Key Assumptions

- **Base Price:** All models initialize at a base price of $10.
- **Price Bounds:** Dynamic prices are constrained within the range of $5 to $20 (0.5x to 2x base price).

- **Vehicle Type Weights:**
  - Truck = 1.5
  - Car = 1.0
  - Bike = 0.75
- **Proximity Threshold:** A lot is considered a competitor if it lies within 0.5 km (as per the Haversine distance).
- **Smooth Variation:** Pricing functions are constrained to avoid erratic jumps across time steps.
- **Demand Normalization:** All demand values are normalized before scaling prices to avoid over-amplification.
- **Traffic and Events:** External conditions (congestion and special days) directly influence demand and pricing.

8. Results & Insights

Model-wise Observations:

- **Model 1 (Linear):**
  - Simple and explainable
  - Works well in isolated lots without external competition
  - Lacks responsiveness to nuanced demand fluctuations
- **Model 2 (Demand-Based):**
  - Captures demand dynamics from multiple input features
  - Better pricing differentiation across time periods and vehicle types
  - Normalized demand leads to stable price changes
- **Model 3 (Competitive):**
  - Introduces market-based feedback and competition simulation
  - Reduces overburdening of popular lots via rerouting
  - Promotes smarter pricing when neighboring lots are full or expensive

General Insights:

- Dynamic pricing improves space utilization and prevents saturation
- Competition-aware strategies yield better balance across locations
- Visualization helps validate model smoothness and real-world interpretability

9. Future Improvements

- **Machine Learning-Based Pricing:** Use regression or classification models to predict demand and set optimal prices.
- **Clustering of Parking Lots:** Group lots by behavior profiles (e.g., business districts vs residential zones) to fine-tune models.
- **Vehicle Rerouting Recommendation Engine:** Offer alternate parking suggestions in real-time to balance load across locations.
- **Reinforcement Learning:** Optimize long-term revenue and occupancy patterns using state-action-reward formulations.
- **Dynamic Congestion Penalties:** Adjust prices more aggressively during peak traffic to discourage further load.
- **Pathway Deployment:** Scale the simulation to real-time APIs with cloud-based deployment of Pathway models.

10. Conclusion

This project successfully demonstrates a real-time dynamic pricing system for urban parking lots based on historical occupancy, real-time traffic, vehicle types, and competitive conditions.

By building three models of increasing sophistication—from simple linear logic to competition-aware pricing—we created a smooth, explainable, and adaptive pricing engine. Integration with Pathway enabled real-time data flow and pricing computation, while Bokeh provided interpretable visual feedback.

The work sets the foundation for intelligent urban mobility pricing solutions, scalable to smart city infrastructures.