Project Report

Dynamic Pricing for Smart Parking Spaces

Problem Overview

As cities grow and vehicles increase, efficient use of parking spaces has become critical. Fixed parking fees do not respond to real-time usage patterns, causing either excessive demand or underutilization. To solve this, dynamic pricing is introduced, adjusting rates in real-time based on congestion, demand, and other urban factors. This report outlines two models implemented using streaming data to enable real-time pricing for smart parking infrastructure.

Objectives

- Develop dynamic pricing models using live parking sensor data.
- Build logic that responds to real-time occupancy, traffic, and behavioral inputs.
- Ensure pricing changes are smooth, bounded, and fair.
- Visualize price behavior with respect to real-time inputs.

Input Dataset Description

Each data record includes:

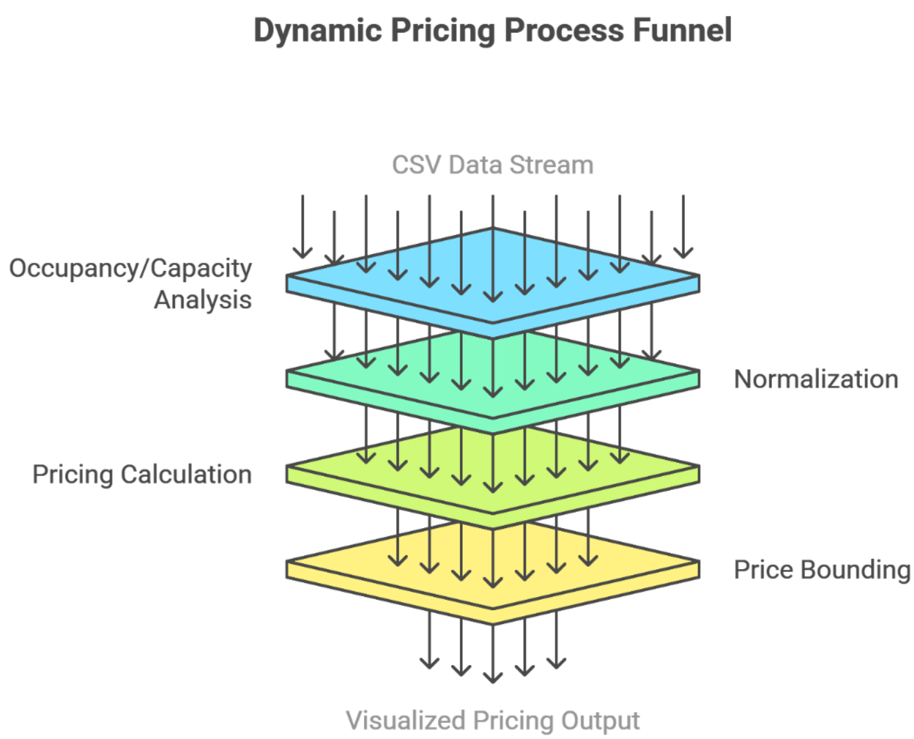
- Timestamp (from combined LastUpdatedDate and LastUpdatedTime)
- SystemCodeNumber (Parking Lot ID)
- Occupancy
- Capacity
- QueueLength
- TrafficLevel (numerical indicator)
- IsSpecialDay (0 or 1)
- VehicleTypeWeight (e.g., car = 1.0, truck = 1.5, bike = 0.5)

This data is streamed through the Pathway engine to simulate a real-time pipeline.

Model 1: Occupancy-Based Dynamic Pricing

Goal: Use occupancy-to-capacity ratio to compute a real-time price.

Architecture:



Logic:

- Calculate raw demand = Occupancy / Capacity
- Normalize demand to max of 1.0
- Compute price:

$$\text{Price} = \text{BasePrice} \times (1 + \lambda \times \text{NormalizedDemand})$$

- Constrain final price between:

$$\text{MIN\_PRICE} = 0.5 \times \text{BasePrice}$$
$$\text{MAX\_PRICE} = 2 \times \text{BasePrice}$$

Example:

- Occupancy: 80
- Capacity: 100
- NormalizedDemand = 0.8
- BasePrice = ₹50, λ = 0.5
- Price = 50 × (1 + 0.5 × 0.8) = ₹70

Features:

- Smooth price adaptation
- No external factors
- Responsive to lot saturation

Output:

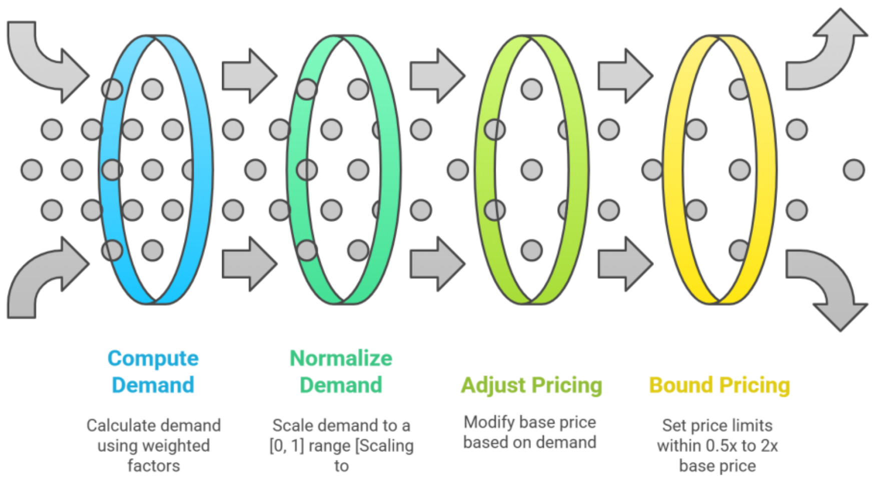
- Lot ID
- Timestamp
- Price

Model 2: Multi-Factor Demand-Based Pricing

Goal: Compute demand using behavioral and environmental inputs.

Architecture:

Dynamic Pricing Model Funnel



Formula:

```
Demand = α × (Occupancy / Capacity)
         + β × QueueLength
         - γ × TrafficLevel
         + δ × IsSpecialDay
         + ε × VehicleTypeWeight
```

Parameters Used:

- α = 3.0
- β = 0.5
- γ = 2.0
- δ = 2.5
- ε = 1.5
- λ = 0.5 (scaling price change)

Example:

- Occ = 80, Cap = 100 → 0.8
- QueueLength = 5, Traffic = 2, SpecialDay = 1, VehicleWeight = 1.5
- Demand = $3\times0.8 + 0.5\times5 - 2\times2 + 2.5\times1 + 1.5\times1.5 = 2.4 + 2.5 - 4 + 2.5 + 2.25 = 5.65$
- Normalized = 5.65 / 20 = 0.2825
- Price = $50 \times (1 + 0.5 \times 0.2825)$ = ₹57.06

Final Pricing Logic:

```
NormalizedDemand = clamp(Demand / 20, 0, 1)
Price = BasePrice × (1 + λ × NormalizedDemand)
Price ∈ [MIN_PRICE, MAX_PRICE]
```

Output:

- Lot ID
- Timestamp
- Normalized Demand
- Final Price

Behavioral Improvements Over Model 1:

- Captures user demand peaks during special events.
- Penalizes traffic-congested zones.
- Increases price for longer queues and heavier vehicle types.

Implementation Highlights

- Pathway was used for streaming flow and stateful computation.
- Prices update live as data changes.
- Python and Bokeh used for live visual dashboards.
- Output written to CSV for further use.

Conclusion

These two models progressively build up a realistic dynamic pricing system for urban parking lots:

- **Model 1** offers basic responsiveness.
- **Model 2** integrates real-world behavior and urban conditions.

Both models maintain smooth, bounded price changes and visualize price trajectories for validation. They provide a foundation for smarter, real-time city parking management.

Melvin Joseph