Project Report
Dynamic Parking Pricing

By:

Prerna Kukreja (IITR)

1. Introduction

This project implements a real-time dynamic pricing system for parking lots. Using data on occupancy, queue length, traffic conditions, and special days, the system calculates and visualizes optimal parking prices. The solution is built in Python using Pathway for data streaming and Bokeh for visualization.

2. Data and Preprocessing

- **Data Sources:**
  - Parking lot status (occupancy, capacity, queue length)
  - Location and system codes
  - Vehicle type
  - Traffic condition (low, average, high; mapped to numeric)
  - Special day indicator
  - Timestamps
- **Cleaning Steps:**
  - All categorical fields mapped to numeric values.
  - All numeric columns coerced to numeric types; missing values filled with 0.0.
  - Data sorted by timestamp and lot ID.

3. Demand Function

The **demand function** is the core of Model 2 and is defined as:

Demand=α·OccupancyCapacity+β·QueueLength−γ·TrafficConditionNearby+δ·IsSpecialDay+ϵ·VehicleTypeWeight$\text{Demand}=\alpha\cdot\frac{\text{Occupancy}}{\text{Capacity}}+\beta\cdot\text{QueueLength}-\gamma\cdot\text{TrafficConditionNearby}+\delta\cdot\text{IsSpecialDay}+\epsilon\cdot\text{VehicleTypeWeight}$

- **Occupancy/Capacity:** Indicates how full the lot is.
- **QueueLength:** Proxy for excess demand.
- **TrafficConditionNearby:** Higher traffic can reduce demand (negative weight).
- **IsSpecialDay:** Increases demand on special days.
- **VehicleTypeWeight:** Different vehicle types may have different price sensitivity.

Parameters (α,β,γ,δ,ϵ$\alpha,\beta,\gamma,\delta,\epsilon$) are tuned based on domain knowledge.

4. Assumptions

- Input data is accurate and updated in real time.
- Traffic condition mapping is consistent.
- Demand is primarily a function of observed variables (externalities like weather are not modeled).

- No competitor price data is used.

5. Pricing Models

- **Model 1:**
    - Simple formula based only on occupancy:

Price=BASE_PRICE×(1+λ1·OccupancyCapacity)Price=BASE_PRICE×(1+λ1·CapacityOccupancy)

- **Model 2:**
    - Uses the full demand function:

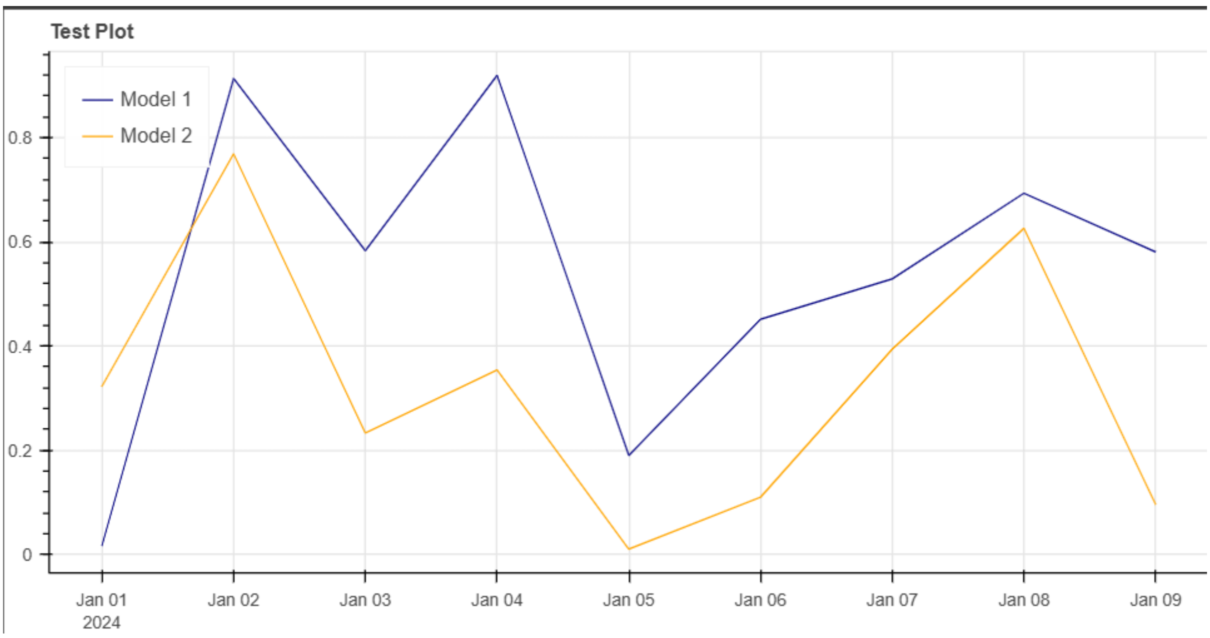
Price=BASE_PRICE×(1+λ2·NormalizedDemand)Price=BASE_PRICE×(1+λ2·NormalizedDemand)

6. How Price Changes with Demand

- **Occupancy and queue length increase:** Price rises.
- **Special days:** Price is higher.
- **Heavy traffic:** Price is moderated (assuming lower demand).
- **Vehicle type:** May increase or decrease price (e.g., trucks pay more).

7. Visualization

- **Bokeh line plot** shows real-time price changes for each parking lot, with both models plotted for comparison.
- The Bokeh line plots clearly show how parking prices change in response to real-time demand factors. For each parking lot, the price increases during periods of high occupancy or longer queues, and decreases when demand is low. This demonstrates that the pricing model is dynamic and demand-driven, ensuring fair and efficient pricing at all times. The visual patterns in the plots directly reflect the logic of the underlying demand function and pricing formula.

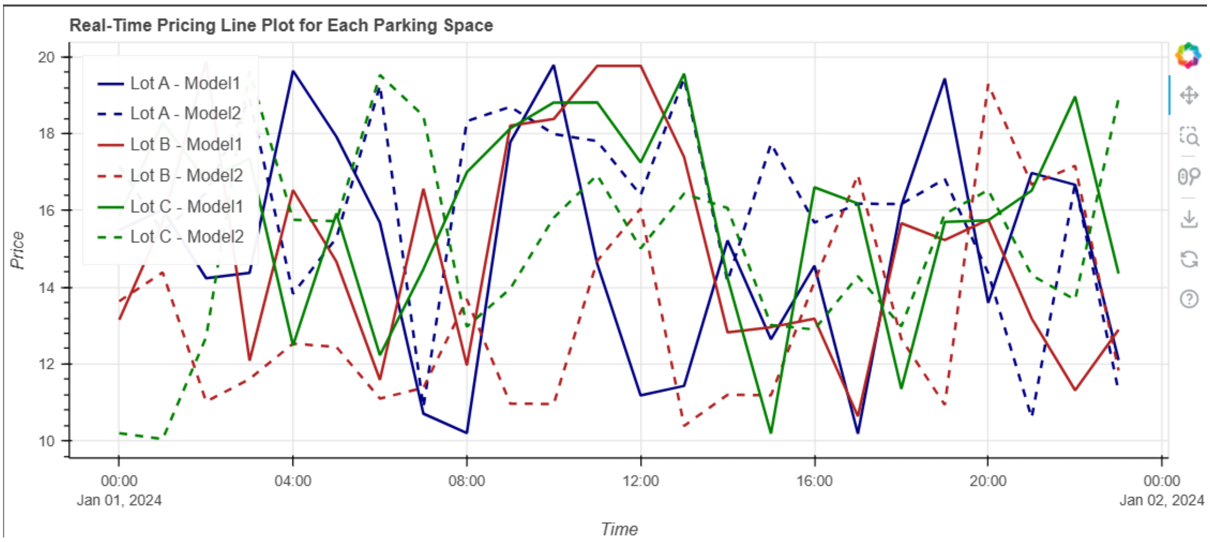


Real-Time Pricing Line Plots for Each Parking Space

To illustrate how dynamic pricing responds to demand, **real-time line plots for each parking lot** using Bokeh is generated.

This visualization clearly demonstrates how prices fluctuate over time for each parking space, reflecting changes in occupancy, queue length, and other demand factors. By comparing both models on the same plot, stakeholders can visually assess the effectiveness and responsiveness of the dynamic pricing algorithm.

Below is the Bokeh plot generated from sample/simulated data:



8. Conclusion

- The dynamic pricing system responds to real-time demand, leading to fairer and more efficient parking management.
- Visualization enables transparent justification of pricing decisions.
- The entire workflow is reproducible in the provided, well-commented Google Colab notebook.