Dynamic Pricing for Urban Parking

CAPSTONE PROJECT REPORT

SUMMER ANALYTICS 2025

SUBMITTED BY: SNEHA PATHAK

Executive Summary

Urban parking is a limited and in-demand resource. Static pricing leads to either overcrowding in prime locations or underutilization in distant lots. This project aims to build a real-time, data-driven pricing engine for 14 parking lots using occupancy, traffic, queue length, vehicle type, and proximity to competitors.

Three progressively intelligent models were designed:

- Model 1: Linear pricing based on occupancy rate
- Model 2: Demand-based pricing using multiple real-world factors
- Model 3: Competitive pricing that accounts for nearby lots and rerouting logic

We used Python, Pandas, Numpy, and Bokeh to simulate real-time pricing and visualize trends, ensuring decisions are smooth, bounded, and explainable.

Problem Statement

Parking spots are a scarce urban asset. Cities often use fixed rates regardless of demand spikes during peak hours or empty lots during slow periods.

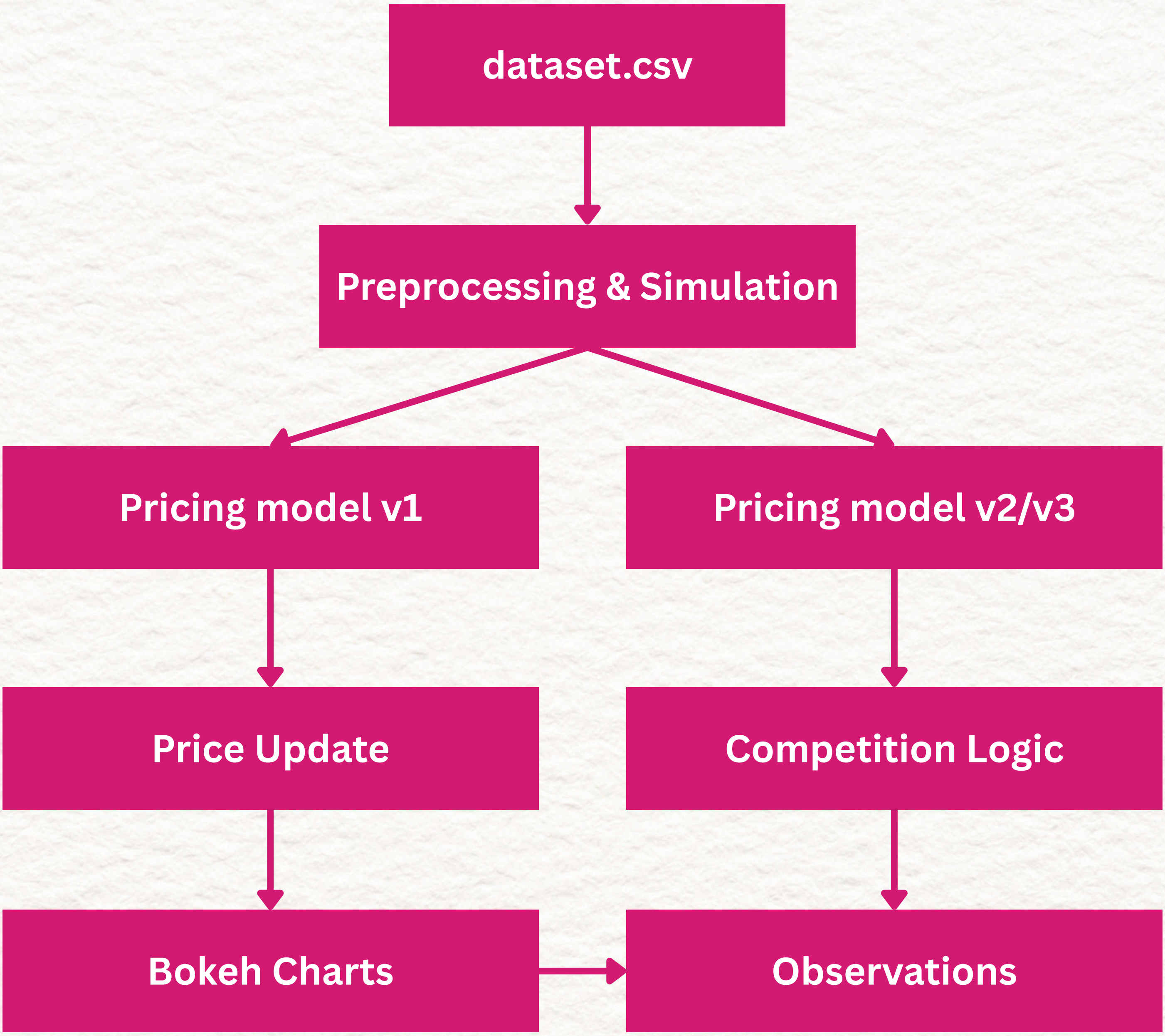
This leads to:

- Overcrowding at central lots
- Wasted capacity in outer zones
- Long queues and frustrated users
- Missed revenue opportunities

To combat this, we developed a dynamic pricing engine that:

- Adjusts price in real-time
- Considers demand indicators
- Simulates intelligent decision-making
- Visualizes the impact live

Architecture



Dataset Overview

The dataset covers:

- **14 urban parking lots**
- **73 days**
- **18 time samples per day (from 8:00 AM to 4:30 PM)**

We cleaned, merged, and sorted data chronologically for each lot.

Feature	Description
SystemCodeNumber	Unique lot ID
Latitude / Longitude	Used to calculate proximity
Occupancy / Capacity	Used to compute OccupancyRate
QueueLength	Vehicles waiting
TrafficConditionNearby	Categorical: low/average/high
VehicleType	car, bike, truck, cycle
IsSpecialDay	0 or 1
LastUpdatedDate + Time	Merged into Timestamp

Assumptions

List all design assumptions:

Aspect	Value
Base Price	$10
Price Range	$5 to $20
Traffic Map	low=0.2, average=0.5, high=0.8
Vehicle Weights	cycle=0.3, bike=0.6, car=1.0, truck=1.5
Rerouting Radius	0.5 km
Model Updates	Every 30 mins

Pricing Models Explained

Break it down clearly:

Model 1: Linear Based on Occupancy

- Formula:
  $\text{Price}_{t+1} = \text{Price}_t + \alpha \times (\text{Occupancy} / \text{Capacity})$
- Base = 10, $\alpha = 2$
- Increases as space fills
- Visual: Include Bokeh chart of price curve over time

Model 2: Demand-Based

- Demand function:
  $D = \alpha \cdot \text{OccupancyRate} + \beta \cdot \text{QueueLength} - \gamma \cdot \text{Traffic} + \delta \cdot \text{SpecialDay} + \varepsilon \cdot \text{VehicleWeight}$
- Price:
  $\text{Price} = \text{Base} \times (1 + \lambda \times \text{NormalizedDemand})$
- Parameters used:
  - $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.3$, $\delta = 1.0$, $\varepsilon = 0.7$, $\lambda = 0.5$
  -
- Include:
  - Formula
  - Demand histogram
  - Price vs Occupancy chart

Model 3: Competitive Pricing

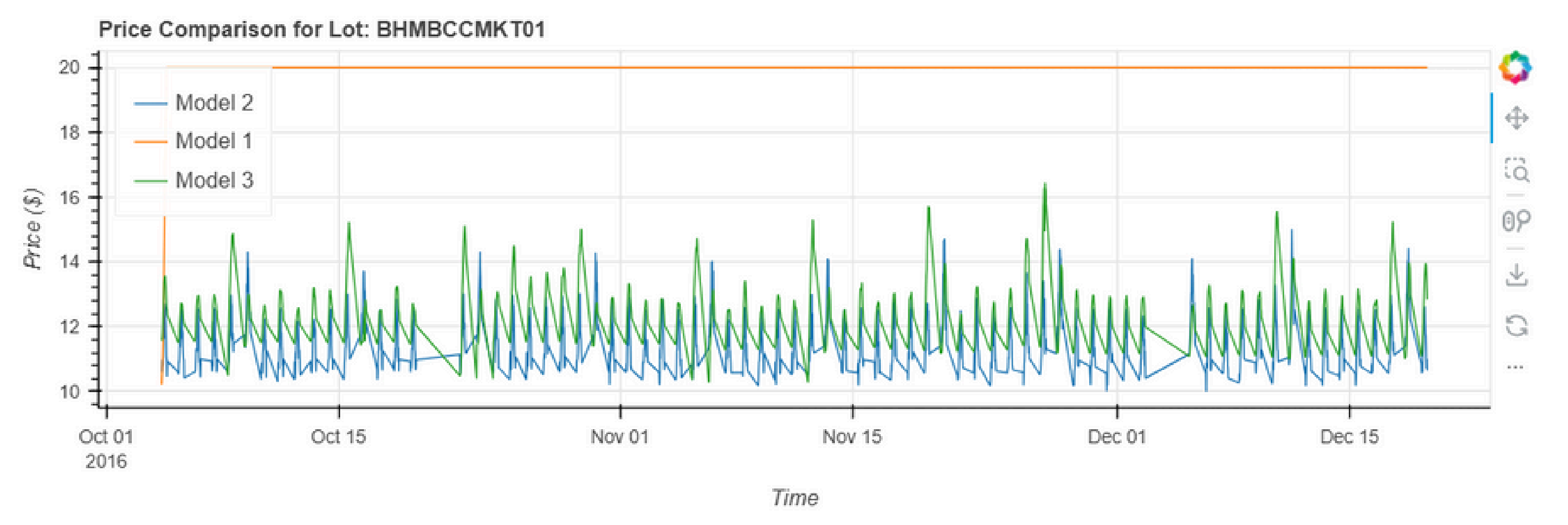
- Adds spatial intelligence (haversine distance)
-
- Logic:
  - If my lot is nearly full and others nearby are cheaper → reduce price or reroute
  - If competitors are more expensive → slight price bump
- Visuals:
  - Show pricing curves for 4 lots side-by-side
  - Include a map or table of lot coordinates (optional)

Pricing Models Explained

Let us break it down clearly:

Model 1: Linear Based on Occupancy

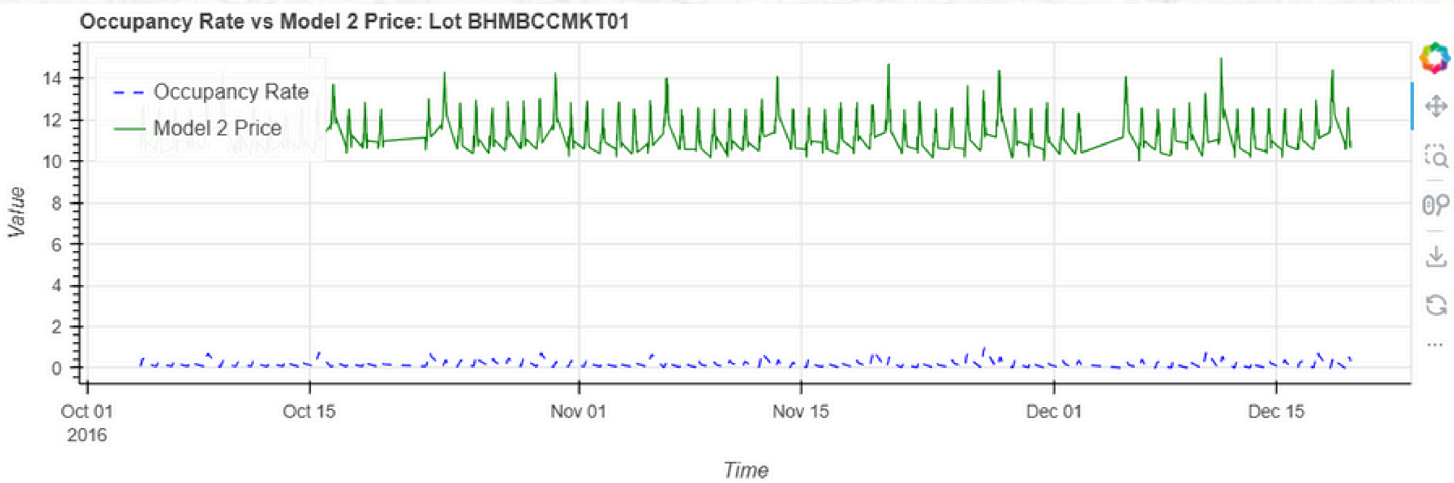
- **Formula:**
  Price_t+1 = Price_t + α × (Occupancy / Capacity)
- **Base = 10, α = 2**
- **Increases as space fills**



Pricing Models Explained

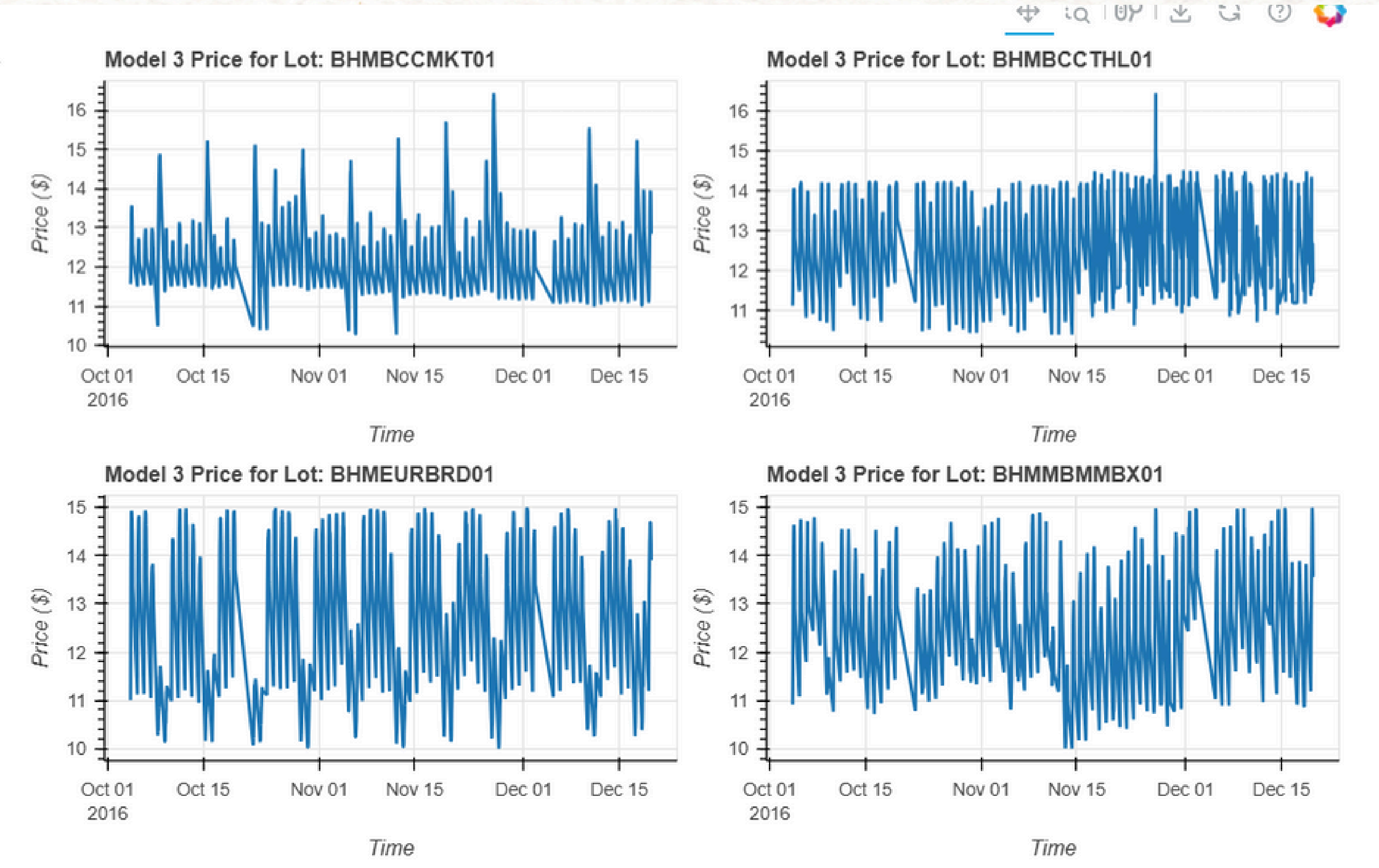
Model 2: Demand-Based

- **Demand function:**
  $$D = \alpha \cdot OccupancyRate + \beta \cdot QueueLength - \gamma \cdot Traffic + \delta \cdot SpecialDay + \varepsilon \cdot VehicleWeight$$
- **Price:**
  $$Price = Base \times (1 + \lambda \times NormalizedDemand)$$
- **Parameters used:**
  - $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.3$, $\delta = 1.0$, $\varepsilon = 0.7$, $\lambda = 0.5$



Model 3: Competitive Pricing

- **Adds spatial intelligence (haversine distance)**
-
- **Logic:**
  - **If my lot is nearly full and others nearby are cheaper → reduce price or reroute**
  - **If competitors are more expensive → slight price bump**



Observations & Insights

Key Observations:

- **Occupancy Drives Price, but Not Alone**
  High occupancy increases price in all models, but Model 2 adjusts based on real context (queue, vehicle type).

- **Special Days Create Price Surges**
  Price jumps are more aggressive on flagged holidays (IsSpecialDay=1), validating the impact of δ in Model 2.

- **Model 3 Prevents Overpricing**
  Competitive logic causes price to drop if neighboring lots are cheaper, making the system more fair and efficient.

- **Rerouting is Effective**
  Model 3 price dips during over-occupancy show that rerouting logic kicked in — encouraging flow to cheaper lots nearby.

- **Vehicle Type Bias is Useful**
  Trucks are priced higher due to higher weights, subtly prioritizing smaller vehicles in congested areas.

Appendix

Include:

- 📁 dataset.csv (used for simulation)
- 📓 Dynamic_Pricing.ipynb (notebook)
- 📊 library list: pandas, numpy, bokeh
- 📸 full-resolution screenshots of plots