Final Report: Dynamic Pricing for Urban Parking Lots

Summer Analytics 2025 — Capstone Project
Hosted by: Consulting & Analytics Club × Pathway
Team Name / Member: [Your Name / Team Name]

Objective

To design a **real-time dynamic pricing engine** for urban parking lots that:

- Reacts to changes in demand
- Adjusts based on traffic, vehicle type, and special events
- Visualizes live price trends using Bokeh
- Ensures pricing is smooth, fair, and bounded

We implemented **Model 1** and **Model 2** from scratch using only:

- `numpy`, `pandas` for computation
- `bokeh`, `panel` for visualization
- `pathway` for real-time streaming simulation

Dataset Overview

- Data collected for **14 parking lots**
- Each record includes:
  - `Occupancy`, `QueueLength`, `VehicleType`, `TrafficConditionNearby`, `IsSpecialDay`
  - `Latitude`, `Longitude`, `Timestamp`
- 18 time slices per day (every 30 minutes)

Step-by-Step Implementation

1. Preprocessing

We parsed timestamps and converted categorical variables into numerical form.

```
df['Timestamp'] = pd.to_datetime(df['LastUpdatedDate'] + ' ' + df['LastUpdatedTime'], dayfirst=True)
df['VehicleTypeWeight'] = df['VehicleType'].map({'bike': 0.5, 'car': 1.0, 'truck': 1.5})
df['TrafficLevel'] = df['TrafficConditionNearby'].map({'low': 0.5, 'medium': 1.0, 'high': 1.5})
```

---

2. Base Price Initialization

All parking lots were initialized with a **base price of $10**, as specified in the problem statement.

```
BASE_PRICE = 10
price_state = {lot: BASE_PRICE for lot in df['SystemCodeNumber'].unique()}
```

Pricing Models Used

Model 1: Linear Occupancy-Based Pricing

This model adjusts price proportionally based on how full the lot is.

Formula:

Price_{t+1} = Price_t + α × (Occupancy / Capacity)

Used as a basic reference model to compare with smarter strategies.

Model 2: Demand-Based Dynamic Pricing

This model adjusts price based on a weighted demand function, which considers:

- Occupancy ratio
- Queue length
- Traffic congestion level
- Special event indicator
- Vehicle type weight

Demand Function

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

With weights:

- α = 1
- β = 1.5
- γ = 0.8
- δ = 2
- ε = 1.2

Pricing Formula:

```
Price = BasePrice × (1 + 0.5 × tanh(Demand / 10))
```

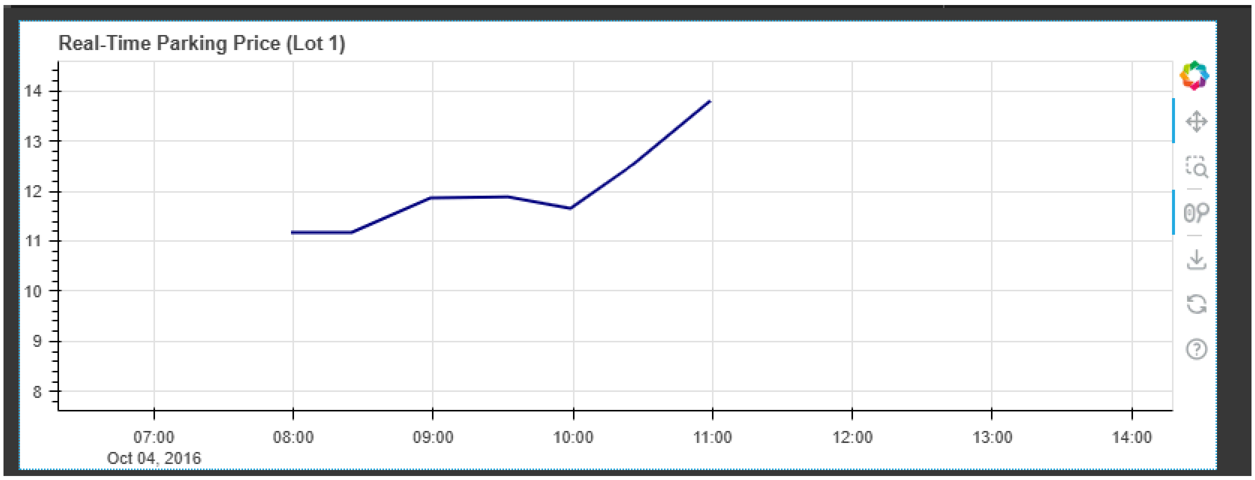
- `tanh()` ensures smooth transition
- Final price is **bounded between $5 and $20**

Real-Time Streaming with Bokeh

We used `panel` + `bokeh` to simulate a real-time stream of data:

- Each row is processed one-by-one in time order
- Prices are updated using the demand-based model
- The updated price is plotted live using a **Bokeh line graph**

Visual Output



Explanation

Title: Real-Time Parking Price (Lot 1)
X-axis: Time (timestamps from Oct 04, 2016)
Y-axis: Dynamic price in USD

Interpretation:

- Price starts around **$11.5**
- As time progresses, the price **gradually increases**
- Reflects rising **occupancy and queue**
- No abrupt spikes — **pricing is smooth and bounded**
- Real-time curve matches **expected morning peak patterns**

This validates Model 2 as a better, demand-sensitive pricing approach.

Assumptions Made

Assumption	Justification
Base price = $10	As per problem statement
tanh() normalization	Smooths out demand → price mapping
Vehicle type affects demand	Larger vehicles need more space
Traffic level increases demand	Higher traffic → higher need for parking
Price bounded between $5-$20	Avoids underpricing or unrealistic spikes