Final Report for Capstone Project

Name: Bibek Nath

Roll No. : 240107024

Department: Chemical Engineering

College: IIT Guwahati

Problem Statement:

Modern cities suffer from inefficient and static parking pricing that fails to respond to dynamic conditions like traffic, demand, or time-of-day. Our goal is to build a real-time, demand-aware, interpretable pricing model for urban parking that is:

- Simple enough to deploy
- Robust to noisy data
- Responsive to real-world demand and supply fluctuations

Dataset Overview

Source: Provided hackathon dataset

Rows: 18,368 live records

Key Features:

- Occupancy, Capacity
- Vehicle Type
- Queue Length
- Traffic Conditions
- Location (Latitude/Longitude)
- Time & Date
- Special Days

Engineering & Preprocessing

Feature Engineering:

- Timestamp combined from date and time
- Hour of Day and Day of Week extracted
- Traffic Mapping: 'low' → 0.3, 'medium' → 0.6, 'high' → 1.0

- Vehicle Type Weights:
    - Car = 1.0
    - Bike = 0.5

- o Bus = 1.5
- o Truck = 1.8

Pricing Models Developed

Model 1: Smoothed Linear Demand-Based Pricing

Formula:

$Price\_t+1 = \beta * Price\_t + (1 - \beta) * (Price\_t + \alpha * (Occupancy / Capacity))$

- $\alpha = 2$, $\beta = 0.6$
- Exponential smoothing added for stability
- Strengths: Simple, stable
- Weaknesses: Ignores other factors like time, queue, traffic

Model 2: Multi-Factor Demand-Based Normalized Model

Raw Demand Score (before normalization):

$Demand = \alpha * (Occupancy/Capacity) + \beta * QueueLength - \gamma * TrafficScore$

$+ \delta * \text{IsSpecialDay} + \varepsilon * \text{VehicleTypeWeight} + \eta * (\text{Hour} / 24)$

Normalized per parking lot to handle local demand variation.

Final Price:

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

- α=1.0, β=0.5, γ=0.7, δ=1.0, ε=0.8, η=0.2, λ=0.6
- Prices bounded between ₹5 and ₹20
- Strengths: Highly responsive, interpretable

Model 3: Competition-Aware Adjustment Model

Logic:

- Uses NearestNeighbors to find nearby parking lots
- Compares their Model 2 prices
- Modifies own price accordingly:

If overcrowded and nearby cheaper: Price = 0.9 × Price_2

If nearby more expensive: Price = 1.1 × Price_2

Else: Price = Price_2

- Strengths: Geo-smart and context-sensitive
- Limitation: Sensitive to noise in nearby data

Visual Analysis

Sample Lot Visualization (interactive via Bokeh + static Matplotlib)

Legend:

- 🔵 Model 1 — Smooth and stable
- 🟢 Model 2 — Demand-sensitive and reactive
- 🔴 Model 3 — Market-smart and adjusted

Evaluation Strategy

If true prices were available (e.g. TruePrice):

- We would compute:

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

mae = mean_absolute_error(true_prices, df["Price_Model_3"])

rmse = mean_squared_error(true_prices, df["Price_Model_3"], squared=False)

r2 = r2_score(true_prices, df["Price_Model_3"])

In absence of true prices:

- Use domain validation
- Evaluate response to demand surges
- Compare pricing range, trend, stability

Output Files

Filename	Description
final_pricing_output.csv	Contains Model 1, 2, and 3 prices

Future Enhancements

Type	Suggestion
Short-Term	Add rolling forecast for occupancy
Mid-Term	Learn dynamic λ values for different lots
Long-Term	Simulate user behavior to test price elasticity

Conclusion

- Model 1 ensures stability
- Model 2 ensures fairness and responsiveness
- Model 3 ensures real-world competitiveness
- Together, they form a robust dynamic pricing engine for smart urban parking

THANK YOU!