**🚗 Dynamic Parking Pricing System - Comprehensive Project Report**

**Summer Analytics 2025 | Akash K**

**🌟 Objective**

This project aims to design a real-time pricing engine for urban parking lots. The objective is to adjust parking prices dynamically based on historical and real-time demand using different pricing strategies. By integrating stream-based processing with visualization tools, we created an end-to-end pipeline that simulates smart city parking logic.

**📦 Technologies Used**

* **Python + Pandas**: Data preprocessing and transformation
* **Pathway**: Real-time stream processing and time-window aggregation
* **Bokeh**: Interactive and comparative visualizations
* **Google Colab**: Development environment for simulation and visualization

**📊 Dataset and Preprocessing**

**Dataset**

* Contains columns: LastUpdatedDate, LastUpdatedTime, Occupancy, Capacity
* Data is timestamped every few minutes

**Preprocessing Steps**

1. **Timestamp Formation**:
   * Combined date and time into one Timestamp
2. **Sorting and Cleaning**:
   * Sorted data chronologically
   * Filtered to a single simulation day: 2024-01-01
3. **Streaming Preparation**:
   * Saved selected columns: Timestamp, Occupancy, Capacity
   * Used Pathway to simulate real-time stream using this file

**🧠 Model Architecture**

**Real-time Windowing**

* All models use **tumbling window aggregation**
* Instance = each day; Window = 1 day
* Behavior = exactly\_once\_behavior to avoid duplicates

**📈 Demand Function (Used in Model 2)**

The demand score is calculated as:

Demand =

1.2 \* (Occupancy / Capacity) # occupancy ratio

+ 0.8 \* QueueLength # set to 3

- 0.5 \* TrafficLevel # "low" = 0.5

+ 0.9 \* IsSpecialDay # 0 (non-holiday)

+ 0.7 \* VehicleType # "car" = 1.0

This simulates demand pressure from both parking lot load and external influences.

**Price Mapping**

normalized\_demand = demand / 10

price = 10 \* (1 + 0.5 \* normalized\_demand)

* **Clamp rule**: price is constrained between $5 and $20

**🛠️ Assumptions**

* **Queue length**, **vehicle type**, and **traffic** are hardcoded for simulation
* **Competitor pricing** is not real — mocked using a basic formula
* **One-day window** simulates temporal pricing update
* **Occupancy data** is assumed reliable and fresh

**🤖 Models Used**

**✅ Model 1: Baseline Linear Pricing**

price = 10 + 5 \* (occupancy / capacity)

* Static linear logic
* Doesn’t respond to external factors
* Good as a reference model

**✅ Model 2: Demand-Based Pricing (Primary)**

* Incorporates multiple features: occupancy, vehicle type, traffic, queue
* Normalized demand mapped to dynamic price
* Most realistic of the three

**✅ Model 3: Competitive Pricing Prototype**

price = (occupancy / capacity) \* 10

* Mimics how a parking system might adjust based on surrounding lots
* In real systems, this would come from GPS + pricing API integration

**📉 Pricing Behavior**

**Price vs. Demand**

* **Model 1**: Price increases linearly with usage
* **Model 2**: Price reacts sharply to total weighted demand
* **Model 3**: Designed to reflect pseudo-competitive pricing

**📈 Visualization Results**

The three models were plotted together using **Bokeh**:

* 🟩 Model 1 (Green)
* 🔵 Model 2 (Blue)
* 🔴 Model 3 (Red)

**📊 Interpretation:**

* Model 1 showed smooth, predictable increases
* Model 2 had fluctuations reflecting demand changes
* Model 3 showed more aggressive pricing due to its multiplier

**📥 Files:**

* bokeh\_output.html: Visualization chart file
* Dynamic\_Parking\_Pricing\_Final.ipynb: Final working notebook with all models

**✅ Conclusion**

* **Model 2** is best suited for a real-world deployment
* **Model 3** is promising if enhanced with external price feeds
* The framework is modular and expandable

**— Report by Akash K | Summer Analytics 2025**