Dynamic Pricing for Urban Parking Lots

Capstone Project of **Summer Analytics 2025**
hosted by **Consulting & Analytics Club × Pathway**

Background and Motivation

Urban parking spaces are a limited and highly demanded resource. Prices that remain static throughout the day can lead to inefficiencies — either overcrowding or underutilization. To improve utilization, **dynamic pricing** based on demand, competition, and real-time conditions is crucial.

This project simulates such a system: participants will create an intelligent, data-driven pricing engine for 14 parking spaces using real-time data streams, basic economic theory, and ML models **built from scratch**, using only `numpy, pandas` libraries.

Data Description

You are given data collected from **14 urban parking spaces** over **73 days**, sampled at **18 time points per day with 30 minutes of time difference** (from **8:00 AM to 4:30 PM** the same day).

Each record includes the following:

Location Information:

- **Latitude** and **Longitude** of each parking space (to calculate proximity to competitors).

Parking Lot Features:

- **Capacity** (maximum number of vehicles that can be parked)
- **Occupancy** (current number of parked vehicles)
- **Queue length** (vehicles waiting for entry)

Vehicle Information:

- **Type of incoming vehicle**: `car`, `bike`, or `truck`

Environmental Conditions:

- **Nearby traffic congestion level**
- **Special day indicator** (e.g., holidays, events)

Each time step reflects the state of each parking lot, and demand will fluctuate throughout the day based on these features.

Project Objective

Your goal is to build a dynamic pricing model for each parking space such that:

- The price is **realistically updated in real-time** based on:
  - Historical occupancy patterns
  - Queue length
  - Nearby traffic
  - Special events
  - Vehicle type
  - Competitor parking prices
- It starts from a **base price of $10**
- The price variation is **smooth and explainable**, not erratic
- **Optional:** The system **suggests rerouting vehicles** to nearby lots if the current lot is overburdened.

Core Requirements

Pricing Logic Implementation

- You must implement all pricing models **from scratch (only Python, Pandas, Numpy, Pathway)**.

Pricing Model Stages

You will build three models, increasing in complexity and intelligence.

Model 1: Baseline Linear Model

A simple model where the **next price** is a function of the **previous price** and **current occupancy**:

- Linear price increase as occupancy increases
- Acts as a reference point

Example:

$$\text{Price}_{t+1} = \text{Price}_t + \alpha \cdot \left(\frac{\text{Occupancy}}{\text{Capacity}} \right)$$

Model 2: Demand-Based Price Function

A more advanced model where you:

- Construct a **mathematical demand function** using key features:
  - Occupancy rate
  - Queue length
  - Traffic level
  - Special day
  - Vehicle type

Example demand function:

$$\text{Demand} = \alpha \cdot \left(\frac{\text{Occupancy}}{\text{Capacity}} \right) + \beta \cdot \text{QueueLength} - \gamma \cdot \text{Traffic} + \delta \cdot \text{IsSpecialDay} + \varepsilon \cdot \text{VehicleTypeWeight}$$

Use this demand value to adjust prices:

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

- Ensure demand is **normalized** and price variations are **smooth and bounded** (e.g., not more than 2x or less than 0.5x base).

Note: The example pricing functions provided in the models above are linear and serve as simple baselines. Participants are encouraged to design and implement more effective and sophisticated pricing strategies.

Model 3 (Optional): Competitive Pricing Model

This model adds **location intelligence** and simulates real-world competition:

- Calculate **geographic proximity** of nearby parking spaces using lat-long.
- Determine **competitor prices** and factor them into your own pricing.

Competitive logic:

- If your lot is full and nearby lots are cheaper → suggest rerouting or reduce price
- If nearby lots are expensive → your price can increase while still being attractive

This model encourages **creativity and business thinking** along with technical skills.

Real-Time Simulation with Pathway

This project **must simulate real-time data ingestion**.

You will use **Pathway** for:

- Ingesting data **streamed with delay**, preserving time-stamp order
- Processing features **in real time**
- Emitting pricing predictions continuously

Sample notebook will be provided with:

- Data simulation code using `Pathway` and the `dataset.csv`.
- Integration with Pathway's streaming features
- Hooks to insert your pricing logic

Execution Environment

- **All code must be written and run in Google Colab.**

Visualization Requirements

You must provide real-time visualizations using **Bokeh**.

Suggested plots:

- **Real-time pricing line plots** for each parking space
- **Comparison with competitor prices**

The final output should **visually justify** pricing behavior.

Submission Guidelines

- Submit a **well-commented Google Colab notebook** and a **report** explaining all your steps, justifying them and include all the models you used.
- Include a section explaining:
  - Your demand function
  - Any assumptions
  - How your price changes with demand and competition
- Include **visualizations and graphs** from Bokeh in the notebook and report.
- **Evaluation:** Since this is a project, there is no objective metric for evaluation like in a hackathon.

Resources for Real Time Applications

1. Pathway. *From Jupyter to Deploy*. Retrieved from: https://pathway.com/developers/user-guide/deployment/from-jupyter-to-deploy/
2. Pathway. *First Real-Time App with Pathway*. Retrieved from: https://pathway.com/developers/user-guide/introduction/first_realtime_app_with_pathway/
3. Summer Analytics, 2025. Retreived from: https://www.caciitg.com/sa/course25/