# Dynamic Pricing for Urban Parking Lots

This project implements an intelligent **dynamic pricing engine** for urban parking lots, adjusting parking rates in real-time based on demand, queue lengths, traffic conditions, and competitor prices.

The solution uses **real-time data streaming with Pathway** and visualizes price fluctuations using **interactive Bokeh plots**.

# Overview

Urban parking spaces are limited and static pricing often leads to **underutilization or overcrowding**.\ This project solves that problem by:

- Increasing prices during high demand and traffic
- Reducing prices when occupancy is low
- 🚔 Considering **competitor prices and proximity** to suggest rerouting
- \*\* Streaming real-time data to update prices dynamically

We built **three pricing models**, progressively improving intelligence:

- 1. Baseline Linear Model
- 2. Demand-Based Price Function
- 3. Competitive Pricing Model

### **Tech Stack**

Component	<b>Details</b>
Language	Python 3
Data Processing	Pandas, NumPy
Real-Time Streaming	Pathway
Visualization	Bokeh, Panel
Hosting/Sharing	Google Colab, GitHub
Geospatial Analysis	Latitude/Longitude-based proximity logic

## **X**Architecture Diagram

```
flowchart TD
A[CSV Data Source] -->|Replay in Real-Time| B(Pathway Streaming Engine)
B --> C{Feature Engineering}
C --> D[Model 1: Linear Pricing]
C --> E[Model 2: Demand-Based Pricing]
C --> F[Model 3: Competitive Pricing]
D & E & F --> G[Dynamic Price Stream]
G --> H[Bokeh Visualization]
G --> I[Real-Time Recommendations]
```

### TProject Architecture & Workflow

#### **Data Ingestion**

- Input data (dataset.csv) includes occupancy, capacity, traffic conditions, queue lengths, and GPS coordinates for 14 parking lots.
- Simulated real-time ingestion using Pathway.demo.replay\_csv().

#### **Feature Engineering**

- Parse timestamp, calculate **occupancy ratio** and **demand metrics**.
- Extract day-level and lot-level features.

#### **Pricing Models**

• Model 1: Simple linear pricing

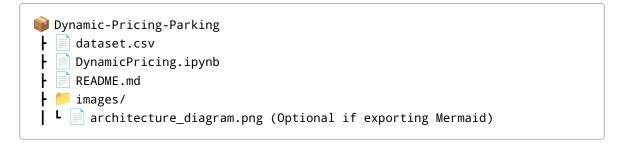
```
price_t+1 = price_t + α * (Occupancy / Capacity)
```

- Model 2: Demand-based pricing using:
- Occupancy Rate
- Queue Length
- Traffic Level
- Special Day Indicator
- Vehicle Type Weightage
- Model 3: Adds competitor prices and rerouting logic.

#### **Real-Time Visualization**

- Plot dynamic prices over time using **Bokeh** interactive charts.
- Simulate price updates and recommendations live.

### **Project Structure**



## Visual Output Examples

(Sample: Real-time price fluctuations for Parking Lot 5)

#### **Documentation**

#### **Features**

- Smooth price transitions (avoids erratic jumps).
- Handles high-demand and low-capacity scenarios gracefully.
- Reroutes vehicles to nearby lots if occupancy is high.

#### **Challenges Solved**

- Simulating real-time streams in Colab.
- Normalizing demand and keeping prices bounded between **0.5x and 2x base price**.
- Integrating geospatial logic with pricing.

# **#**How to Run

- 1. Open in Google Colab:
- 2. Upload dataset.csv
- 3. Run all cells to start real-time simulation.

### **Author**

- · Chirayu Khalwa
- Elkhalwachirayu@gmail.com