Dynamic Pricing for Urban Parking Spaces with Real-Time Streaming using Pathway & Bokeh

---

Project Report

1. Problem Statement

Urban areas are experiencing exponential growth in vehicle ownership, causing traffic congestion, inefficient parking, and underutilization of spaces. Traditional fixed-rate pricing models fail to adapt to real-time demand and do not reflect the dynamic nature of parking availability. The aim of this project is to create a dynamic pricing system for urban parking spaces using real-time streaming, machine learning, and clustering techniques.

---

2. Project Overview

This solution dynamically calculates parking prices using three models:

- **Model 1**: Linear pricing based on occupancy rate.
- **Model 2**: Demand-based pricing using a weighted ML formula.
- **Model 3**: Competitive pricing adjusted for local cluster averages.

Real-time visualization is powered by **Bokeh** and **Panel**, while data streaming is simulated using **Pathway**.

---

3. Technologies & Libraries Used

Python Libraries:

- **pandas**: For data handling
- **numpy**: For numerical operations
- **sklearn**:
    - `LabelEncoder` (categorical encoding)
    - `MinMaxScaler` (normalization)
    - `KMeans` (clustering)
- **matplotlib**: Basic plotting (used optionally)
- **bokeh**: Interactive plotting
- **panel**: Bokeh dashboard hosting
- **threading** & **time**: Simulated real-time data updates

- **pathway**: For schema definition and time-windowed streaming pipeline

---

4. Feature Engineering

The dataset contains:

- Occupancy & Capacity
- Vehicle Type
- Traffic Conditions
- Queue Length
- Special Day Indicator
- Latitude & Longitude

Engineered Features:

- `OccupancyRate = Occupancy / Capacity`
- `VehicleWeight` (mapped from type)
- `RawDemand` using:
  - OccupancyRate
  - QueueLength
  - TrafficEncoded
  - IsSpecialDay
  - VehicleWeight
- `NormalizedDemand` scaled to [0, 1]

---

5. Pricing Models Explained

Model 1: Linear Pricing

Simple baseline model:

```
Price = BasePrice + alpha * OccupancyRate
```

- Alpha is a sensitivity factor (e.g., 2)
- Encourages higher prices for fuller lots

Model 2: Demand-Based Pricing

Combines multiple real-time features:

```
RawDemand = 1.5*OccupancyRate + 1.2*QueueLength - 0.8*Traffic +
1.0*IsSpecialDay + 0.7*VehicleWeight
```

- Scaled between 0 and 1

- Used to calculate price: `Price = 10 * (1 + 0.5 * NormalizedDemand)`
- Clipped between 5 and 20 for fairness

Model 3: Competitive Pricing

Applies local adjustments using clustering:

- Clusters created using Latitude & Longitude (via KMeans)
- Compares price to cluster average
- Adjusts downward if OccupancyRate > 0.9
- Adjusts upward if cluster mean > current price
- Final price clipped to [5, 25]

---

6. Pathway Streaming Pipeline

- **Schema Defined**: `ParkingSchema` with Timestamp, Occupancy, Capacity, Model Prices
- **Replay CSV**: Data is streamed at 100 records/sec using Pathway's `replay_csv`
- **Tumbling Window**: Aggregates prices daily
- **Reducers**: Mean prices and min/max occupancy captured

---

7. Real-Time Dashboard

Using `Bokeh` + `Panel`:

- Daily average prices for each model are plotted
- Interactive tooltips show all prices per date
- Streaming is simulated using `threading` and `sleep`

---

8. Scalability Discussion

This model is highly extensible:

- Can ingest real-time IoT sensor data
- Scales to multiple cities via clustering
- Easy to integrate with APIs for payment & parking systems

---

9. Limitations & Future Work

- Currently uses simulated streaming
- Demand weights are hand-tuned

- No true real-time IoT or camera feed integration
- Limited to 3 pricing models; could expand to reinforcement learning or RL agents

Next Steps:

- Integrate with live city parking APIs
- Add forecasting for peak times
- Introduce anomaly detection in price jumps