Dynamic Pricing for Urban Parking Lots

Capstone Project – Summer Analytics 2025
Consulting & Analytics Club × Pathway

1. Introduction

Urban parking is a scarce and valuable resource. Static pricing leads to inefficiency—either overcrowding or underutilization. This project implements a real-time, data-driven dynamic pricing engine for 14 urban parking spaces, using only Python, Pandas, Numpy, Pathway, and Bokeh for visualization, as per competition guidelines.

2. Data Preparation

The provided dataset contains:

- Parking lot metadata: SystemCodeNumber, Latitude, Longitude, Capacity
- Real-time state: Occupancy, QueueLength, VehicleType, TrafficConditionNearby, IsSpecialDay
- Timestamps split as LastUpdatedDate and LastUpdatedTime

Preprocessing steps:

- Combined date and time into a single ISO 8601 UTC timestamp for Pathway compatibility.
- Mapped TrafficConditionNearby (low, average, high) to numeric values (1, 2, 3) for use in mathematical models.
- Saved the cleaned data as dataset_processed.csv.

3. Pricing Models

Model 1: Baseline Linear Model

The price increases linearly with occupancy:

$$Price_{t+1} = Price_t + \alpha \cdot \frac{Occupancy}{Capacity}$$

where $\alpha = 2.0$.

Model 2: Demand-Based Model

Price adjusts based on a composite demand function:

$$Demand = \alpha \cdot \frac{Occupancy}{Capacity} + \beta \cdot QueueLength - \gamma \cdot TrafficLevel + \delta \cdot IsSpecialDay + \epsilon \cdot VehicleTypeWeight$$

- Normalized demand is used to compute a price factor, bounded between 0.5x and 2x the base price.
- Vehicle type weights: car=1.0, bike=0.5, truck=1.5.

Model 3: Competitive Pricing Model (Optional)

- Intended to adjust prices based on prices at nearby lots using latitude and longitude.
- Due to time and complexity, the submitted notebook implements up to Model 2, but the code structure allows for Model 3 extension.

4. Real-Time Simulation Pipeline

- Used Pathway to ingest the processed CSV as a real-time stream.
- Applied the pricing logic row-wise using pw.apply in Pathway, outputting a stream of predicted prices for each lot and time.
- Output written to output_priced.csv.

5. Visualization

- Used Bokeh to plot real-time price lines for each parking lot.
- Each lot's price evolution is shown over time, allowing visual comparison and validation of the pricing logic.
- The plots justify that prices are responsive to demand features and remain within smooth, explainable bounds.

6. Assumptions

- All categorical features (traffic, vehicle type) are mapped to numeric values for modeling.
- The demand function coefficients are chosen for demonstration; further tuning could be done for production.
- Real-time simulation is approximated using static mode in Pathway for reproducibility.

7. Results

- The output CSV (output_priced.csv) contains timestamped price predictions for each lot, confirming the pipeline's functionality2.
- Visualizations show price increases with occupancy, queue, and special events, and decrease with higher traffic, as designed.

8. Limitations and Future Work

- Model 3 (competition-aware pricing) can be implemented by adding spatial proximity logic and competitor price comparison.
- Further refinement of the demand function and parameter tuning could improve real-world applicability.
- Integration of rerouting suggestions for overburdened lots is possible as an extension.

9. Conclusion

This project demonstrates a robust, hackathon-compliant dynamic pricing system for urban parking, leveraging real-time data streams, basic economic theory, and modern data engineering tools. The code and report meet all specified requirements, and the approach can be extended for more advanced business logic as needed.

Attachments:

- Colab notebook (sub_path_VR_py.ipynb)
- Output pricing CSV (output_priced.csv)
- This report