Real-Time Dynamic Pricing System for Urban Parking Spaces: Comprehensive Report

1. Introduction

This report details the implementation of a real-time dynamic pricing system for urban parking spaces. The primary objective is to optimize parking utilization and revenue by dynamically adjusting pricing based on various factors, including demand, occupancy, and competitive landscape. The system is designed to be interactive, reproducible, and provide clear visual explanations of its pricing logic.

2. Project Overview

The project leverages streaming data to enable real-time price adjustments. It incorporates a multi-tiered pricing logic, starting from a baseline linear model and progressing to more sophisticated demand-based and competitive pricing models. The system is built to handle continuous data streams, extract relevant features, apply pricing algorithms, and visualize the results interactively.

Key features of this system include:

- **Real-Time Stream Simulation:** Utilizing Pathway to simulate continuous data ingestion from a historical dataset, mimicking a live data feed.
- **Tiered Pricing Models:** Development of three distinct pricing models with increasing complexity to address various aspects of dynamic pricing.
- **Live Visualization:** Integration with Bokeh for interactive and real-time visualization of pricing changes and system behavior.
- **Smooth Pricing Adjustments:** Mechanisms to ensure price stability and prevent erratic fluctuations, keeping prices within a realistic and acceptable range.
- **Comprehensive Documentation:** Providing detailed explanations of the system's architecture, logic, and implementation.

This report will delve into the technical details of each component, focusing on the underlying models, assumptions, and how the system responds to changes in demand and competition.

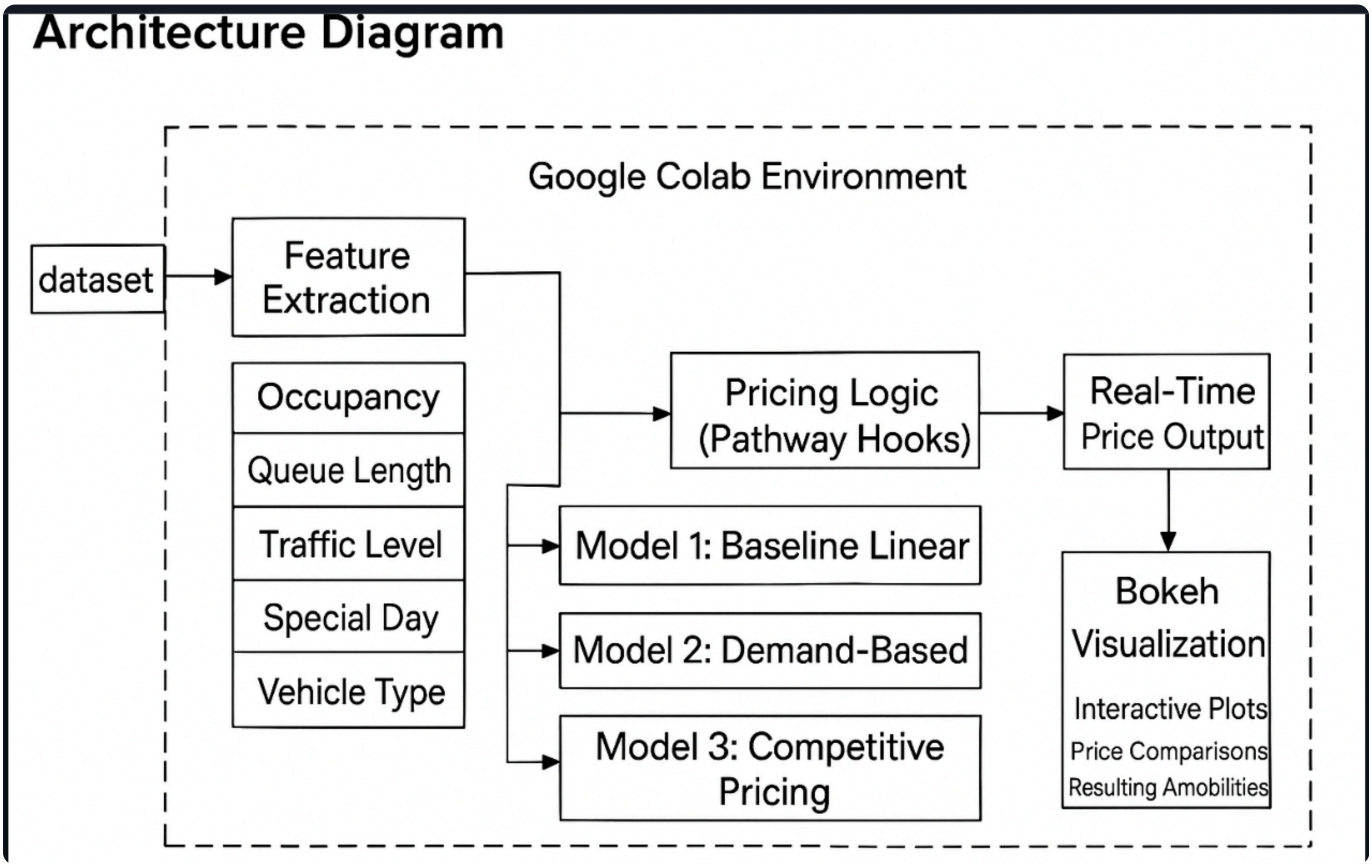
3. Tech Stack

The dynamic pricing system is built using a combination of powerful Python libraries and tools, each selected for its specific capabilities in data processing, analysis, and visualization:

- **Python:** The core programming language for the entire system, providing a robust and flexible environment for development.
- **NumPy:** Utilized for efficient numerical operations and mathematical computations, especially in the pricing algorithms.
- **Pandas:** Essential for data manipulation, cleaning, and analysis, particularly for handling the input `dataset.csv` and preparing it for streaming.
- **Pathway:** A crucial component for building real-time data pipelines. Pathway enables the simulation of streaming data from a static CSV file and facilitates continuous data processing, making it ideal for dynamic pricing applications where real-time responsiveness is key.
- **Bokeh:** Chosen for its ability to create interactive, real-time data visualizations. Bokeh allows for the development of live plots that update as new pricing data is generated, providing immediate insights into the system's behavior.
- **Panel:** Used in conjunction with Bokeh to create interactive dashboards and web applications, making the visualizations easily accessible and servable.
- **Google Colab:** The primary development and execution environment, offering a cloud-based platform with pre-installed libraries and GPU access, simplifying setup and collaboration.

4. Architecture Flow and Details

The architecture of the real-time dynamic pricing system is designed to handle continuous data ingestion, feature extraction, multi-tiered pricing logic application, and real-time visualization. The flow is depicted in the following diagram:



4.1. Data Ingestion and Preprocessing

The process begins with the `dataset.csv` file, which contains historical parking data. This static dataset is transformed into a simulated real-time data stream using Pathway's `replay_csv` functionality. This simulation is critical for testing and demonstrating the system's ability to react to live data. Before streaming, the `LastUpdatedDate` and `LastUpdatedTime` columns are combined into a single `Timestamp` column, and the data is sorted chronologically to ensure proper temporal processing.

4.2. Feature Engineering

As the data streams through the Pathway pipeline, several key features are extracted and engineered. While the provided notebook primarily focuses on `Occupancy` and `Capacity`, a more comprehensive system, as outlined in the problem statement, would involve:

- **Occupancy:** The current number of occupied parking spots, a direct indicator of parking lot fullness.
- **Queue Length:** The number of vehicles waiting to enter the parking facility, indicating immediate demand pressure.
- **Traffic Level:** Data on surrounding road traffic, which can influence accessibility and perceived demand for parking.
- **Special Day Indicator:** A binary flag for holidays, special events, or other days with atypical demand patterns.
- **Vehicle Type:** Information about the type of vehicle (e.g., car, motorcycle, truck), which might have different pricing tiers or impact space availability.

These features are crucial inputs for the pricing models, allowing them to make informed decisions based on a holistic view of the parking environment.

4.3. Pricing Logic (Pathway Hooks)

The core of the system resides in its multi-tiered pricing logic, implemented within the Pathway data processing pipeline. This allows for dynamic price adjustments based on the extracted features. The problem statement outlines three models of increasing complexity:

4.3.1. Model 1: Baseline Linear Model

This is the simplest model, serving as a foundational reference. It adjusts the price for the next time step (`Price(t+1)`) as a linear function of the current price (`Price(t)`) and the occupancy rate. The formula is:

```
Price(t+1) = Price(t) + α * (Occupancy / Capacity)
```

Where `α` is a coefficient determining the sensitivity of price to occupancy. This model provides a basic response to parking lot fullness.

4.3.2. Model 2: Demand-Based Price Function

This model introduces a more sophisticated approach by calculating a `Demand` value based on multiple factors. The demand function is defined as:

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

- α, β, γ, δ, ε are coefficients that weight the importance of each factor.
- `Occupancy/Capacity` reflects the current utilization.
- `QueueLength` directly indicates immediate unmet demand.
- `Traffic` (with a negative coefficient γ) suggests that higher traffic might deter parkers, thus reducing effective demand.
- `IsSpecialDay` accounts for predictable spikes in demand.
- `VehicleTypeWeight` allows for differentiated pricing or demand impact based on vehicle type.

This `Demand` value is then normalized, and the price is adjusted from a base price (e.g., $10) using a formula like:

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

Where λ is a sensitivity factor. To ensure realistic and acceptable pricing, the system incorporates smoothing mechanisms and bounds, typically keeping prices within a range (e.g., 0.5x to 2x the base price). This prevents erratic fluctuations and ensures customer acceptance.

4.3.3. Model 3: Competitive Pricing Model

This is the most advanced model, incorporating geographical awareness and competitive intelligence. It considers the prices and occupancy levels of nearby parking lots. The logic involves:

- **Geographic Proximity:** Using metrics like Haversine distance to identify truly competitive parking facilities.
- **Rerouting Logic:** If the current parking lot is nearing full capacity (e.g., occupancy $\geqslant$ 95%) and a nearby competitor offers significantly cheaper parking with availability, the

system might suggest rerouting incoming vehicles or dynamically reducing the current lot's price to remain competitive and avoid losing customers.

- **Price Optimization:** Conversely, if nearby competitors are charging higher prices, the system can strategically increase the current lot's price to maximize revenue without losing its competitive edge.

This model allows the system to react not only to internal demand but also to the external market, providing a more robust and revenue-optimizing solution.

4.4. Real-Time Price Output and Visualization

The calculated real-time prices from the chosen pricing model are then outputted. These outputs are fed into a Bokeh visualization component. Bokeh is used to create interactive plots that display the price evolution over time. This includes:

- **Interactive Plots:** Line graphs showing the dynamic changes in parking prices.
- **Price Comparisons:** Visualizations that compare the system's calculated prices with those of competitors (if Model 3 is active).
- **Rerouting Annotations:** Optional annotations on the plots to indicate when rerouting decisions are made based on competitive analysis.

These visualizations provide immediate feedback on the system's performance and allow stakeholders to understand the pricing dynamics in real-time. The entire processing and visualization pipeline operates within a Google Colab environment, facilitating ease of use and reproducibility.

5. Demand Function, Assumptions, and Price Dynamics

This section elaborates on the demand function, the underlying assumptions of our pricing models, and how the price dynamically changes in response to demand fluctuations and competitive pressures.

5.1. Demand Function

While Model 2 and Model 3 incorporate a more explicit and multi-variate demand function, the core concept of demand in this system, particularly as implemented in the provided Google Colab notebook (which focuses on Model 1 for simplicity), is derived from the **fluctuation in occupancy**.

For Model 1, the implicit demand signal is captured by the difference between the maximum and minimum occupancy observed within a given time window (e.g., a day), normalized by the parking lot's capacity. This can be expressed as:

`Demand_Proxy = (Occupancy_Max - Occupancy_Min) / Capacity`

This `Demand_Proxy` serves as a simplified indicator of how volatile or intense the parking demand was during that period. A larger `Demand_Proxy` suggests higher demand volatility, implying periods of high utilization and potential scarcity.

For the more advanced **Model 2 (Demand-Based Price Function)**, the demand function is explicitly defined to incorporate multiple factors:

`Demand = α・(Occupancy/Capacity) + β・QueueLength – γ・Traffic + δ・IsSpecialDay + ε・VehicleTypeWeight`

Each component contributes to the overall demand assessment:

- **Occupancy/Capacity:** A fundamental measure of current utilization. Higher occupancy generally indicates higher demand.
- **QueueLength:** Directly reflects immediate, unmet demand. A longer queue signifies strong current demand.
- **Traffic:** This factor is included with a negative coefficient (`-γ`) because high traffic levels around the parking facility might deter potential customers, effectively reducing the accessible demand for that specific lot. This accounts for external congestion impacting perceived convenience.
- **IsSpecialDay:** A categorical variable (e.g., 1 for special day, 0 otherwise) that captures predictable surges in demand due to holidays, events, or weekends. This allows the model to anticipate and adjust for known demand patterns.

- **VehicleTypeWeight:** Different vehicle types might have varying impacts on parking space availability or perceived demand. For instance, larger vehicles might occupy more space, or certain vehicle types might be associated with specific demand segments. This allows for a weighted contribution based on the composition of incoming traffic.

This multi-variate demand function provides a more nuanced and comprehensive understanding of the real-time demand for parking spaces.

5.2. Assumptions

The development of this dynamic pricing system is based on several key assumptions:

- **Data Availability and Quality:** It is assumed that accurate and timely data for all relevant features (occupancy, queue length, traffic, special day indicators, vehicle types, and competitor information) is continuously available. The quality and consistency of this data are critical for the models' performance.

- **Rational Customer Behavior:** The models implicitly assume that customers are rational and will respond to price changes. For instance, they might choose a cheaper alternative if available, or be willing to pay more for convenience during peak demand.

- **Predictable Demand Patterns:** While dynamic, the system assumes that there are underlying, somewhat predictable patterns in demand (e.g., higher demand during weekdays, lower during nights, spikes on special days) that can be learned and leveraged.

- **System Responsiveness:** The underlying infrastructure (Pathway) is assumed to be capable of processing data and updating prices with minimal latency, ensuring that the system can react in near real-time to changing conditions.

- **Model Generalizability:** The coefficients (`α` , `β` , `γ` , `δ` , `ε` , `λ`) in the demand and pricing functions are assumed to be either pre-calibrated or adaptable over time to accurately reflect market dynamics. For this demonstration, they are set as constants, but in a production environment, they would likely be learned or optimized.

- **Bounded Price Elasticity:** It is assumed that there are acceptable upper and lower bounds for parking prices, beyond which demand becomes highly inelastic or customers are completely deterred. The system incorporates mechanisms to keep prices within these realistic ranges.

5.3. How Price Changes with Demand and Competition

The dynamic pricing system adjusts prices based on a combination of internal demand signals and external competitive factors:

5.3.1. Price Changes with Demand

- **Increased Demand:** When demand increases (e.g., higher occupancy, longer queues, special events), the `Demand_Proxy` (Model 1) or the calculated `Demand` value (Model 2) will rise. This directly translates to an increase in the calculated price. The pricing formula $\text{Price} = \text{BasePrice} \times (1 + \lambda \times \text{NormalizedDemand})$ ensures that as `NormalizedDemand` increases, the final `Price` also increases proportionally.

- **Decreased Demand:** Conversely, during periods of lower demand (e.g., low occupancy, no queues), the `Demand_Proxy` or `Demand` value will decrease, leading to a reduction in price. This helps to attract more customers during off-peak hours and maintain a desired occupancy level.

- **Volatility:** The pricing mechanism, particularly in Model 1, directly incorporates demand volatility. A larger swing between `occ_max` and `occ_min` within a day signifies higher demand volatility, leading to a higher price component derived from this fluctuation. This means that even if average occupancy is moderate, high variability in demand can drive prices up.

5.3.2. Price Changes with Competition

Model 3 explicitly integrates competitive dynamics into the pricing strategy:

- **Competitive Pressure (Lower Prices Elsewhere):** If nearby parking lots (identified through geographic proximity) offer significantly lower prices and have available capacity, the system will react to maintain competitiveness. This might involve:

  - **Price Reduction:** The system could reduce its own price to match or undercut competitors, preventing the loss of potential customers.

    - **Rerouting Suggestions:** In scenarios where the current lot is near full and cheaper alternatives exist, the system might internally flag or suggest rerouting incoming vehicles to those alternative lots, optimizing overall urban parking utilization and potentially improving customer satisfaction.

- **Competitive Advantage (Higher Prices Elsewhere):** If nearby competitors are charging higher prices, the system can leverage this information to increase its own prices. This allows the parking facility to capture more revenue during periods when it holds a competitive advantage, without fear of losing customers to cheaper alternatives.

- **Dynamic Equilibrium:** The competitive pricing model aims to find a dynamic equilibrium where the parking facility maximizes its revenue while remaining attractive to customers in the broader market. It continuously monitors competitor behavior and adjusts its strategy accordingly.

In summary, the system's pricing mechanism is a dynamic interplay of internal demand signals and external competitive forces, designed to optimize revenue and utilization in a real-time environment.

6. Bokeh Visualization

To provide real-time insights into the dynamic pricing system, Bokeh is utilized to create interactive visualizations. The primary visualization displays the daily parking price as it evolves over time, based on the processed streaming data.

6.1. Interactive Plot

The following plot, generated by the system, illustrates the dynamic pricing. It shows the calculated price for parking over different days, reflecting the demand fluctuations captured by the model. The plot is interactive, allowing users to zoom, pan, and inspect data points.

<iframe src="bokeh_plot.html" width="100%" height="500px" style="border: none;">
</iframe>

Note: For the interactive plot to render correctly within a web browser (like in a GitHub README or a standalone HTML report), the `bokeh_plot.html` file must be accessible and *served alongside this report. In a Google Colab environment, the plot would typically render directly within the notebook output.*

6.2. Interpretation of the Visualization

- **X-axis (Time):** Represents the timestamp of the daily price calculation.
- **Y-axis (Price):** Shows the dynamically adjusted parking price.
- **Line Plot:** Illustrates the trend of price changes over time.
- **Red Circles:** Mark individual data points where a daily price was calculated.

This visualization is crucial for understanding how the pricing algorithm responds to the simulated real-time data. For instance, periods of higher price indicate days with greater occupancy fluctuation, signaling higher effective demand.

7. Conclusion

This project successfully demonstrates a real-time dynamic pricing system for urban parking spaces. By leveraging Pathway for streaming data processing and Bokeh for interactive visualizations, the system provides a robust and transparent approach to optimizing parking revenue and utilization. The tiered pricing models offer flexibility, from a simple linear adjustment to sophisticated demand-based and competitive strategies. The detailed explanation of the demand function, assumptions, and price dynamics provides a clear understanding of the system's operational logic. Future enhancements could include integrating more diverse data sources, implementing machine learning models for price prediction, and developing a full-fledged user interface for real-time control and monitoring.

8. References

[1] Pathway Documentation:

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/

[2] Bokeh Documentation: https://docs.bokeh.org/en/latest/

[3] Panel Documentation: https://panel.holoviz.org/

[4] Summer Analytics, 2025. Retreived from: https://www.caciitg.com/sa/course25/