**Dynamic Pricing for Urban Parking Lots**

**Project by:** S. Premanandh

**Date of Submission:** 10 June 2025

**Background and Motivation**

Urban parking spaces are a limited and highly demanded resource, yet they are often managed with static pricing. Fixed rates that remain unchanged throughout the day can lead to significant inefficiencies – parking lots may become overcrowded during peak times or sit underutilized during off-peak periods . This capstone project addresses that problem by exploring dynamic pricing strategies to optimize the use of urban parking lots. The motivation is to improve parking space utilization and driver satisfaction by adjusting prices in real-time based on demand and other relevant factors . In essence, the project simulates an intelligent, data-driven parking fee system that responds to changing conditions, rather than a one-size-fits-all pricing scheme .

**Pricing Models Implemented**

To achieve dynamic pricing, we implemented and evaluated two models of increasing complexity. Both models were developed from scratch using Python (with libraries like NumPy and Pandas) and integrated into a streaming context using the Pathway API for real-time operation.

**Model 1: Baseline Linear Pricing.** The first model serves as a simple baseline. It assumes that price should increase linearly as the lot fills up. In this approach, each time step updates the parking price by a factor proportional to the current occupancy level relative to capacity . For example, if occupancy is rising, the price will increment slightly (from a base price) to temper demand, whereas if occupancy is low, the price remains closer to the base. This linear model uses the previous time step’s price as a reference and adjusts it by a constant α times the occupancy ratio (Occupancy/Capacity). It provides a straightforward, interpretable reference point for dynamic pricing, ensuring that higher utilization directly translates to higher pricing in a controlled manner .

**Model 2: Demand-Based Pricing.** The second model is more advanced and accounts for multiple real-time factors to determine price, reflecting a more holistic demand-based strategy. Instead of using occupancy alone, Model 2 constructs a demand score from various features:

* **Occupancy rate:** how full the parking lot is (vehicles parked as a fraction of capacity).
* **Queue length:** number of vehicles waiting to enter the lot.
* **Traffic level:** the congestion in nearby roads.
* **Special events:** whether the day has a holiday or event that spikes demand.
* **Vehicle type:** category of incoming vehicle (for instance, different weights for car vs. truck).

These factors are combined into a mathematical demand function that we designed, drawing inspiration from basic economic theory . For instance, occupancy and queue length drive the demand up, while heavy traffic (which might discourage driving) could drive demand down, and special events add a positive demand shock. An example formulation (as suggested in the project guidelines) could be a weighted linear combination of these features . Once the demand score is calculated at a given time, the price is updated as a percentage adjustment on the base price according to that normalized demand. In practice, our implementation computes the price at time *t* using the formula: Pricet = BasePrice × (1 + λ × NormalizedDemand) . We also enforced practical bounds on pricing so that it remains within reasonable limits (for example, capping the price between 50% and 200% of the base price) to keep fluctuations smooth and avoid erratic jumps . Model 2 thus responds to real-time conditions more sensitively than Model 1, but still ensures pricing changes are gradual and explainable .

*(Note: A more complex Model 3 involving competitive pricing was outlined in the project brief, integrating factors like nearby lot prices and rerouting suggestions . However, our capstone primarily focuses on developing and validating Model 1 and Model 2 as described above.)*

**Real-Time Data Processing with Pathway**

A key requirement of the project was to simulate a **real-time streaming environment** for the pricing engine. We addressed this by leveraging the Pathway API, a Python framework for stream data processing. Using Pathway, we ingested the historical parking dataset as if it were a live feed, introducing data records with timestamp delays to mimic real-world streaming input . The dataset consisted of time-indexed records for 14 parking lots over 73 days, with updates every 30 minutes from morning to evening . Pathway allowed us to preserve timestamp order and process each update sequentially, ensuring that our pricing models reacted to events in the correct chronological sequence .

Within the Pathway pipeline, we performed on-the-fly feature engineering. For example, at each time step, the occupancy rate had to be calculated (vehicles occupied vs. capacity) and fed into the models. We also combined raw inputs (queue length, traffic indicators, etc.) to compute the demand score in real time for Model 2. One practical challenge we overcame was **timestamp parsing and synchronization**: the raw data timestamps needed to be correctly parsed and aligned so that Pathway’s event streamer would issue updates in true temporal order. We utilized Pathway’s utilities to replay the data with real-time delays based on actual timestamps, which required careful formatting of datetime fields and handling of any irregularities in the data. By addressing these issues, we ensured that the system would not only run without errors but would also accurately reflect the timing of real parking lot dynamics.

Another challenge involved ensuring that our real-time calculations remained efficient. Because Pathway continuously processes incoming data, any heavy computation in the demand function or feature engineering could slow down the streaming loop. We mitigated this by vectorizing computations with NumPy and precomputing any static information (like capacity or vehicle type weights) outside the real-time loop. As a result, the pricing engine maintained real-time responsiveness, demonstrating that dynamic pricing calculations can keep up with live data streams.

**Visualization and Interpretation**

*Real-time pricing visualization (Model 1 – Linear Pricing) for sample parking lots. Each line chart (one per lot) shows how the price (y-axis) evolves over the course of a day (x-axis, time) in response to occupancy changes. Visual tools like these Bokeh plots were used to monitor and explain the dynamic pricing behavior.*

To make sense of the models’ output and to justify their behavior to stakeholders, we incorporated real-time visualization using Bokeh in Google Colab. As the Pathway stream processed data and updated prices, those prices were simultaneously fed into live-updating line plots. We created interactive charts for each parking lot showing the price over time, alongside indicators of lot status (such as occupancy level or a marker for special events). These visualizations were crucial for interpreting the model outcomes: for example, one can observe that when occupancy or queue length spikes, the corresponding lot’s price curve slopes upward in real time, indicating a price increase, and when demand falls, the price stabilizes or dips. By examining the graphs, we verified that the pricing changes were smooth and correlated logically with the underlying factors, as intended .

The Bokeh dashboard not only served as a verification tool for us as developers but also as a communication tool. In a practical deployment, parking lot managers or city officials could watch these charts to understand how and why prices are changing. For instance, if a special event caused a surge in price for a downtown lot, the visualization would clearly link that event (perhaps indicated with a visual flag) to the price surge, making the model’s decision-making transparent. This real-time feedback loop helped ensure that our dynamic pricing system remained **explainable** and aligned with common sense expectations, which is critical for gaining user trust and regulatory acceptance.

**Challenges and Practical Considerations**

Implementing dynamic pricing in real time came with several challenges beyond just algorithm design. One major practical hurdle was **data handling in a streaming context**. Unlike a traditional static analysis, we had to ensure that every piece of incoming data was correctly processed in sequence and that the system could handle continuous updates. Debugging in this streaming environment required careful logging and testing, as errors could accumulate over time or only manifest after many iterations. We addressed this by initially simulating the stream at a slower pace and verifying outputs (such as the generated model2\_stream.csv log of price updates) against expected patterns.

Another challenge was synchronizing the output between the Pathway stream and the Bokeh visualization. We needed the plotting to update without significant lag as new data arrived. In a Colab environment, this meant using Bokeh’s server or periodic callback features. We ended up streaming the price data to a CSV (continuously appended) and reading it into Bokeh plots on a schedule. This intermediate output (model2\_stream.csv) was essentially a real-time trace of Model 2’s pricing decisions, and ensuring its integrity (especially the time stamps and ordering) was essential. We also produced a daily summary output (demand\_model\_daily\_output.jsonl) to aggregate results for offline analysis, which required collecting and resetting daily stats in the streaming job.

Crucially, throughout these technical efforts, we kept the **human element** in focus. The modeling choices were driven by interpretability and practical reasoning. For example, we intentionally chose a linear model and a transparent demand formula (as opposed to a “black-box” machine learning model) so that each price change can be justified by a clear factor (occupancy went up, so price went up, etc.). Tuning parameters like α or λ was done with domain intuition—ensuring, for instance, that a full lot might raise the price by a noticeable but not outrageous amount. We also had to consider edge cases (like what if the occupancy drops to zero suddenly, or an event ends abruptly) and make decisions on how the pricing algorithm should gracefully handle such scenarios (e.g., by setting a floor for price drops to avoid whiplash effects).

**Interpretability and Practical Application**

A key outcome of this project is the demonstration that dynamic pricing for parking is not only technically feasible but also **practically actionable**. We emphasized human interpretability at every step so that the end solution is something that city administrators or parking lot operators could confidently use. Every component of the system was designed to be understandable: from the linear relationship in Model 1 to the multi-factor demand in Model 2, the logic aligns with intuitive decision-making processes (for instance, “charge more when you’re almost full” or “offer discounts when traffic is low”). This decision-driven modeling approach means that stakeholders can tweak the pricing rules or weights based on policy or preference, and they will immediately grasp the effect of those changes.

Moreover, the project underscored how real-time data can empower smarter urban management. By processing live information about occupancy and traffic, the system can make minute-by-minute pricing adjustments that would be impossible to manage manually. Yet, because our model’s decisions are explainable, these automatic adjustments remain under control and can be reviewed or adjusted by humans if needed. In practical application, such a system could reduce overcrowding (by nudging drivers to cheaper, emptier lots), increase revenue during high-demand periods, and improve overall user satisfaction by ensuring that parking availability and pricing are more closely aligned.

**Conclusion**

In summary, *Dynamic Pricing for Urban Parking Lots* was a capstone project that combined data-driven modeling with real-time system implementation to address a common urban challenge. Completed by S. Premanandh on June 10, 2025, the project delivered two dynamic pricing models and a live demonstration of their effectiveness. Through the use of Pathway for streaming data ingestion and Bokeh for live visualization, we showed how parking prices can be continually optimized based on demand indicators, all in an interpretable and controlled manner. The successful execution of this project suggests that cities could adopt similar approaches to make parking management more efficient and responsive. Ultimately, the work bridges the gap between theoretical dynamic pricing models and their deployment in a realistic setting, highlighting the value of both technology and thoughtful human-centric design in solving real-world problems.