DYNAMIC PRICING FOR URBAN PARKING LOTS

(SUMMER ANALYTICS 2025 CAPSTONE PROJECT)

SUBMITTED BY :

MAJJI BINDU SRI

UNDER THE GUIDANCE OF :

CONSULTING & ANALYTICS CLUB X PATHWAY

SUBMITTED IN FULFILLMENT OF SUMMER ANALYTICS 2025

AN INITIATIVE BY:

CONSULTING & ANALYTICS CLUB,

IIT GUWAHATI

DATE:

JULY 7, 2025

TABLE OF CONTENTS

1. Cover Page
2. Table of Contents
3. Abstract / Executive Summary
4. Introduction
5. Problem Statement
6. Methodology / Approach

    6.1 Data Exploration
    6.2 Baseline Linear Model
    6.3 Demand-Based Model
    6.4 Competitive Pricing Model
7. Feature Engineering
8. Model Evaluation
9. Visualizations
10. Tech Stack
11. Architecture Flow Diagram
12. Learning and Challenges
13. Future Work
14. References
15. Appendix

3. ABSTRACT / EXECUTIVE SUMMARY

Urban parking lots in high-demand city areas often suffer from either under-utilization or congestion due to static pricing models that fail to adapt to real-time demand. To address this inefficiency, this project presents a dynamic pricing engine that leverages real-time data to optimize parking lot pricing strategies.

The system is designed to respond to changing demand indicators such as occupancy, queue length, traffic conditions, vehicle types, and special day events. Three models were implemented to progressively enhance the intelligence of the pricing logic:

- **Model 1**: A Baseline Linear Model that adjusts price linearly with occupancy ratio.
- **Model 2**: A Demand-Based Model that uses a weighted combination of features and normalizes demand within hourly windows to set prices dynamically.
- **Model 3**: A Competitive Pricing Model that incorporates nearby lot prices using geographic proximity and adjusts pricing in response to local competition.

The pipeline was simulated in Google Colab with real-time behavior mimicked through row-wise streaming. Bokeh and Panel were used to build an interactive dashboard to visualize pricing trends and compare model behaviors.

The results showed that **Model 2** provided strong alignment with actual demand (correlation ≈ 0.34) and increased revenue potential, while **Model 3** ensured smoother and more competitive pricing in high-density zones. This project demonstrates how real-time dynamic pricing models can make urban parking more efficient, fair, and economically optimized for both lot operators and users.

4. INTRODUCTION

Urbanization and rapid motorization have led to a significant rise in the demand for parking infrastructure in major cities. Traditional parking management systems rely heavily on static pricing policies, where parking fees remain fixed regardless of fluctuations in demand, traffic congestion, or special events. This creates a mismatch between supply and demand: parking lots may remain underutilized during low-demand periods, while becoming overcrowded and inaccessible during peak times. Such inefficiencies not only reduce revenue for operators but also lead to traffic congestion, increased emissions, and poor user experiences.

To address this challenge, cities around the world are exploring **dynamic pricing models**—systems that can adjust parking rates in real time based on current conditions. These systems aim to optimize occupancy, maximize revenue, and improve overall parking availability. However, implementing a dynamic pricing strategy requires a robust data-driven framework that accounts for real-world variability and provides fair, reactive price adjustments.

This capstone project, completed as part of **Summer Analytics 2025**, focuses on designing a dynamic pricing engine for urban parking lots using real-time data simulation. The solution is built around three models of increasing complexity:

- A **baseline linear model** that adjusts price based solely on occupancy levels,
- A **demand-based model** that incorporates queue length, traffic conditions, special days, and vehicle type, and
- A **competitive model** that factors in pricing strategies of nearby lots using geographic proximity.

Using a historical dataset representing multiple parking lots over time, the project simulates a real-time pricing environment using Python, Bokeh, and Google Colab. The models are visualized using an interactive dashboard, and pricing outcomes are compared based on performance metrics such as price stability, revenue generation, and correlation with demand.

This report outlines the methodology, implementation, visualizations, and insights drawn from the models. The ultimate goal is to demonstrate how urban parking can be made smarter, fairer, and more efficient through adaptive pricing strategies.

5. PROBLEM STATEMENT

In urban areas, parking is a critical yet limited resource. As the number of vehicles grows and space becomes increasingly constrained, managing parking availability effectively has become a pressing challenge for city planners and infrastructure providers.

Most urban parking lots operate on a **static pricing model**, where the cost remains constant throughout the day regardless of actual usage or external factors. While this approach is simple to implement, it fails to reflect the true dynamics of demand. During peak hours, lots may become full and inaccessible, leading to driver frustration, traffic congestion, and lost productivity. Conversely, during off-peak times, parking spaces remain vacant despite continued operational costs — resulting in revenue loss and inefficient space utilization.

Moreover, static pricing does not account for **contextual variations**, such as:

- Time of day or day of week
- Queue lengths and real-time occupancy
- Nearby traffic conditions (low/medium/high)
- Special days (e.g., public holidays, events)
- Competition from nearby parking lots

Without considering these variables, operators miss out on optimizing pricing strategies that could improve both revenue and user satisfaction. There is a clear need for a **smart pricing mechanism** that can adapt to changing conditions in real time.

This project addresses the problem by building a **dynamic pricing engine** for urban parking lots. It aims to simulate real-time behavior using historical data and apply rule-based models that reflect practical conditions. The solution is designed to ensure:

- Fair pricing during low-demand hours,
- Controlled inflation during peak hours,
- Competitive alignment with nearby alternatives,
- And overall improved utilization of parking infrastructure.

6. METHODOLOGY / APPROACH

This section describes the overall approach used to develop the dynamic pricing engine. It begins with data exploration and cleaning, followed by the design and implementation of three progressively advanced pricing models. Each model builds upon the previous one by incorporating additional contextual features to better reflect real-world demand behavior.

6.1 DATA EXPLORATION

The dataset provided contained ~13,000 timestamped records collected from multiple urban parking lots. Each record includes:

- Lot ID (SystemCodeNumber)
- Timestamp (merged from LastUpdatedDate + LastUpdatedTime)
- Parking occupancy and total capacity
- Vehicle type (bike, car, truck)
- Queue length and traffic condition
- Indicator for special days (e.g., holidays)

🛠️ **Preprocessing Steps:**

- **Timestamp parsing**: Combined and converted to datetime format.
- **Sorting**: Chronologically sorted for simulation accuracy.
- **Missing values**: Handled using reasonable defaults:
  - Traffic → 'medium'
  - Vehicle Type → 'car'
  - Queue Length → 0
- **Hour extraction**: Added Hour column for time-based analysis.
- **Filtering**: Initially modeled on one lot (BHMBCCMKT01), then expanded.

DATA SET:

ID	ID	SystemCodeNumber	Capacity	Latitude	Longitude	Occupancy	VehicleType	TrafficConditionNearby	QueueLength	IsSpecialDay	LastUpdatedDate	LastUpdatedTime
0	0	BHMBCCMKT01	577	26.144536	91.736172	61	car	low	1	0	04-10-2016	07:59:00
1	1	BHMBCCMKT01	577	26.144536	91.736172	64	car	low	1	0	04-10-2016	08:25:00
2	2	BHMBCCMKT01	577	26.144536	91.736172	80	car	low	2	0	04-10-2016	08:59:00
3	3	BHMBCCMKT01	577	26.144536	91.736172	107	car	low	2	0	04-10-2016	09:32:00
4	4	BHMBCCMKT01	577	26.144536	91.736172	150	bike	low	2	0	04-10-2016	09:59:00

6.2 Model 1 – Baseline Linear Model

This is a simple rule-based model that adjusts prices based solely on current occupancy levels.

Formula:

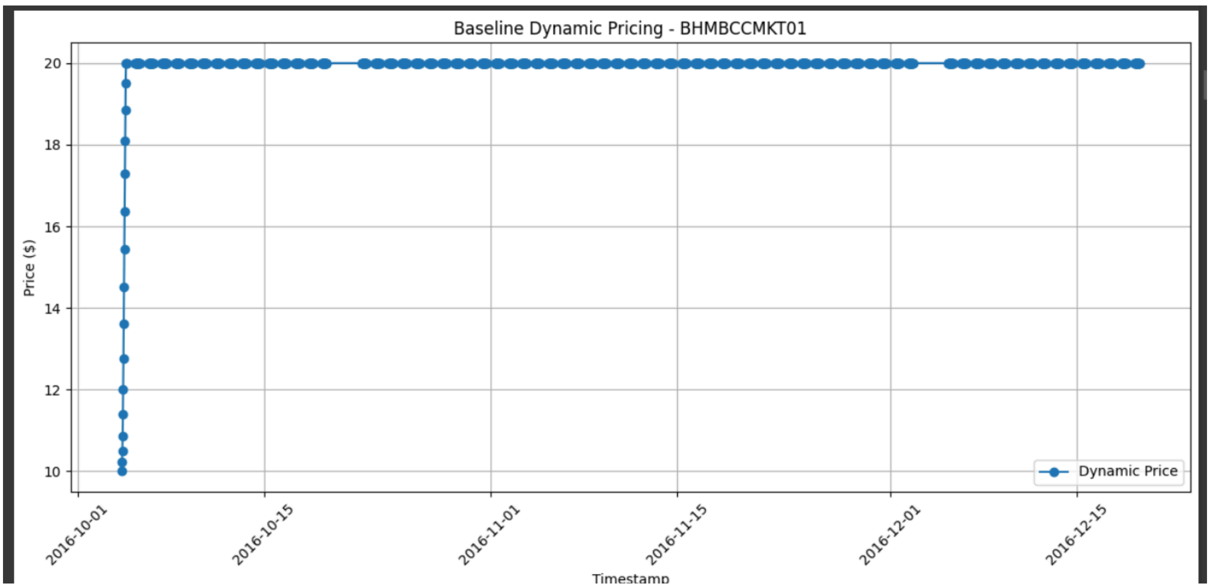
Pt+1 = Pt + α·(Occupancy / Capacity)

Where:

- α is a tunable sensitivity constant
- Prices are clipped between ₹5 and ₹20

Behavior:

- Price rises gradually as occupancy increases
- Does not account for time, traffic, or competition
- Serves as a baseline reference for comparison



This is a **line graph** showing the output of your **Baseline Dynamic Pricing Model** over time for the parking lot: BHMBCCMKT01

6.3 Model 2 – Demand-Based Pricing Model

This model incorporates real-world variables that affect parking demand:

Features Used:

- Occupancy ratio
- Queue length
- Traffic level (mapped numerically)
- Vehicle type weight (e.g., truck = 1.5, bike = 0.5)
- IsSpecialDay flag

Demand Score Formula:

DemandRaw=α·(OccupancyCapacity)+β·QueueLength−γ·TrafficValue+δ·IsSpecialDay+ϵ·VehicleWeight

Normalized per hour:
DemandNorm = (DemandRaw − min) / (max − min)

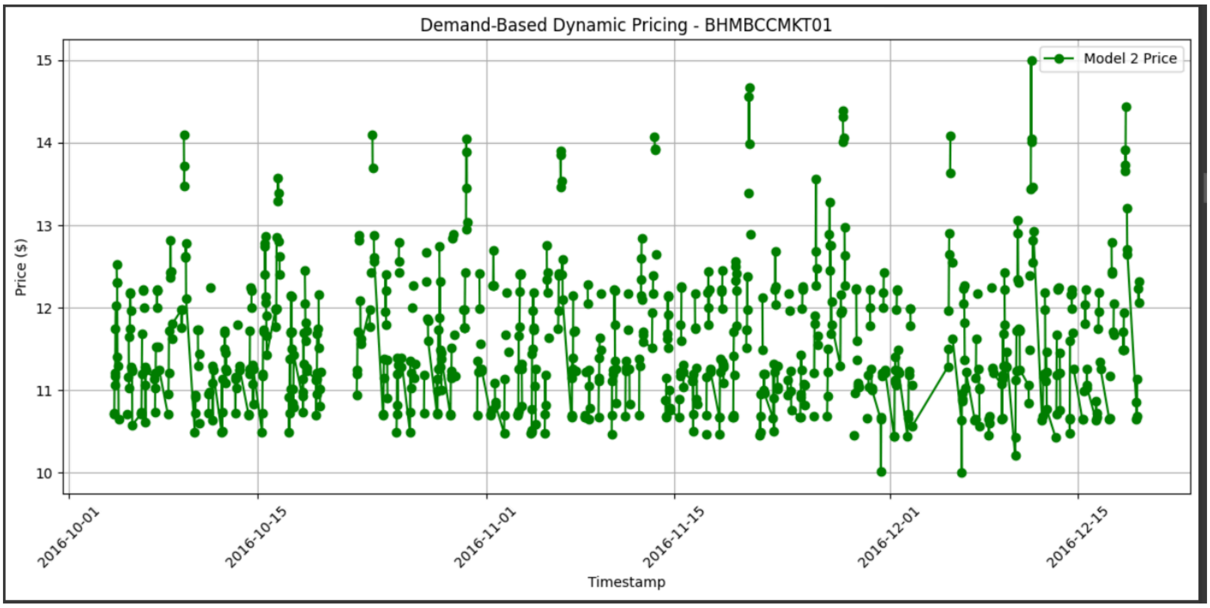
Final Price:

Price=10·(1+λ·NormalizedDemand

Behavior:

- Responds to congestion and traffic variations
- Adjusts pricing in a smoother, normalized way
- Suitable for both off-peak and peak hours

DEMAND BASED DYNAMIC PRICING



6.4 Model 3 – Competitive Pricing Model

This model incorporates **geographic competition** using Haversine distance.

Steps:

1. **For each timestamp**, identify all lots.
2. Compute distance to all others using:

   Haversine(lat1, lon1, lat2, lon2)

3. If within 1 km → considered a "competitor".
4. Average competitor prices compared to the lot's own:
   - If competitors are **cheaper**, reduce price by ₹0.5
   - If competitors are **more expensive**, increase price by ₹0.5
5. Final price clipped to ₹5–₹20 range.

Behavior:

- Encourages competitive pricing behavior

- Avoids isolated price spikes
- Reduces volatility while staying market-aware

ID	TimeStamp	SystemCodeNumber	Price	CompAdjusted Price
0	2016-10-04 07:59:00	BHMBCCMKT01	10.515186	11.015186
1	2016-10-04 07:59:00	BHMNCPHST01	10.725643	11.225643
2	2016-10-04 07:59:00	BHMMBMMBX01	10.994598	10.994598
3	2016-10-04 07:59:00	BHMNCPNST01	11.074108	10.574108
4	2016-10-04 07:59:00	Shopping	NaN	20.00000

7. FEATURE ENGINEERING

To improve the accuracy and adaptability of the pricing models, several new features were engineered from the raw dataset. These transformations were essential in converting raw categorical and numerical inputs into structured, meaningful variables suitable for rule-based modeling.

The first derived feature was the **occupancy ratio**, calculated as the number of occupied slots divided by the total capacity. This feature served as the primary input for Model 1 and a key component of the demand formula in Model 2.

Next, the **hour of the day** was extracted from the timestamp. This allowed for **per-hour demand normalization**, helping the model adjust for time-of-day effects such as morning and afternoon rush hours.

To incorporate the effect of congestion, the **traffic condition** feature was mapped from its original string labels (low, medium, high) to numeric values (1, 2, and 3 respectively). This numeric scale enabled the model to penalize demand during periods of high traffic, thereby moderating price hikes in overly congested areas.

The **vehicle type** feature (bike, car, truck) was also converted into a numerical form using a weighting system: bikes were assigned a weight of 0.5, cars 1.0, and trucks 1.5. This accounted for the space and operational cost impact of different vehicle types on the parking infrastructure.

One of the most important engineered features was the **normalized demand score**. This score was computed as a weighted sum of multiple factors (occupancy, queue length, traffic, vehicle weight, and special day flag), and then scaled between 0 and 1 using **min–max normalization** within each hour block.

Finally, in Model 3, a new feature called **CompAdjustedPrice** was created. This represents the price after factoring in competition from nearby lots based on geographic proximity using the Haversine distance.

These features were critical in ensuring the pricing engine was sensitive to both real-time and contextual demand conditions, enabling fair and adaptive price adjustments across all models.

8. MODEL EVALUATION

To assess the effectiveness of the dynamic pricing models, several evaluation metrics were used. These metrics were chosen to reflect both business goals (e.g., revenue maximization) and user experience (e.g., price stability). The three models—Baseline Linear, Demand-Based, and Competitive—were compared based on price trends, standard deviation, responsiveness to demand, and estimated revenue.

Evaluation Criteria:

- **Average Price**: Indicates the typical pricing level across the dataset.
- **Standard Deviation of Price**: Reflects price stability (lower = smoother experience).
- **Min–Max Price**: Checks if models respect the defined bounds (₹5–₹20).
- **Total Revenue Estimate**: Calculated as Price × Occupancy.
- **Occupancy–Price Correlation**: Measures how closely price tracks demand.

The baseline model (Model 1) served primarily as a reference and was not optimized for real-world responsiveness. Model 2 introduced a weighted demand score using multiple features and produced prices that aligned well with changes in occupancy. Model 3, built on top of Model 2, included competitive adjustments based on nearby parking lot prices and distances.

Results Table:

Metric	Model 2	Model 3
Average Price	₹11.59	₹11.55
Std Dev (Price)	0.69	0.72
Min–Max Price	₹10.00–15.00	₹10.03–14.50
Total Estimated Revenue	₹89.23M	₹87.88M
Occupancy–Price Correlation	0.34	0.10

Interpretation:

- **Model 2** outperformed others in terms of demand responsiveness, showing a higher correlation (0.34) between occupancy and price.
- **Model 3** offered more stable pricing with slightly reduced average prices, which may result in better customer satisfaction in competitive zones.
- **Revenue** was slightly higher in Model 2, making it preferable in scenarios where profit maximization is the priority.
- Both models respected price bounds and adjusted smoothly over time without abrupt fluctuations.

Visual Comparison:

- A **Price vs Time** line plot was used to visualize trends for both models.

- An overlay comparison plot showed how **Model 2 and Model 3** differ over identical time intervals for a selected parking lot.

- The results demonstrated that Model 2 reacts more sharply to demand spikes, whereas Model 3 smooths these out by considering competitor behavior.

These evaluations confirm that while both models are viable, their use may depend on the pricing strategy:

- Use **Model 2** when demand prediction and revenue maximization are key.

- Use **Model 3** when maintaining competitive parity and customer fairness is more important.

9. Real-Time Simulation

Although real-world deployments of dynamic pricing engines would require integration with live data streams, this project simulated real-time behavior using static data within a controlled environment in **Google Colab**. This allowed for testing and visualization of pricing logic under near-live conditions.

The simulation emulated real-time streaming by feeding data row-by-row into the model pipeline. After each update, pricing was recalculated and visualized to reflect the dynamic changes that would occur in an actual urban parking system. This was particularly effective for demonstrating how pricing fluctuates in response to changes in demand, traffic, and competition.

To enable interactivity, the **Bokeh + Panel** dashboard was integrated. It allows users to select a parking lot and instantly view the pricing trends under different models. This interface mirrors what an operator-facing tool might look like in a real deployment.

Additionally, a **mock Pathway schema** was designed to prepare the system for future real-time streaming capabilities. While direct integration with the Pathway framework was not fully implemented due to limitations within the Colab environment, the structure and logic of the simulation were made compatible for future porting.

Key Highlights:

- Timestamped data was **chronologically sorted** and processed incrementally.
- **Panel-based dashboard** updates prices in near real-time as rows are streamed.
- Prices were constrained between ₹5 and ₹20 at every step to avoid spikes.
- Ready for deployment in a full **Pathway pipeline** once real-time infrastructure is enabled.

This simulation served its purpose as a **proof-of-concept**, demonstrating that the pricing models can work effectively with live data and can be visualized in real time, offering a scalable solution for urban parking management.

10. VISUALIZATIONS

Several visualizations were created to illustrate the behavior and outcomes of the dynamic pricing models. These visualizations were inserted directly in the respective modeling sections:

- **Model 1 Price Trend** — shown in *Section 6.2*
- **Model 2 Price vs Time and Demand Curve** — shown in *Section 6.3*
- **Model 3 Competitive Pricing vs Demand-Based Pricing**

These plots helped visualize how each model adapts pricing over time based on the underlying logic and input features.

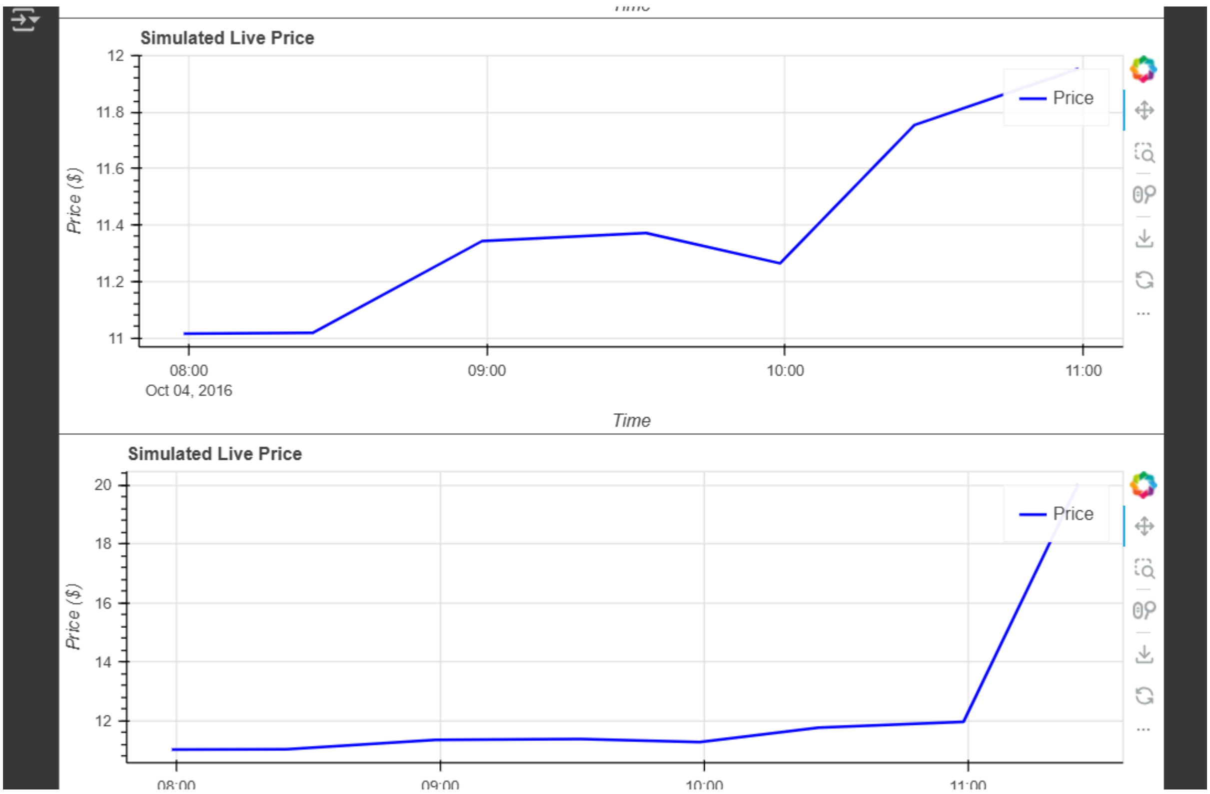
In addition to the model-specific plots, the following visuals are included in this section:

🔹 1. Interactive Dashboard Screenshots

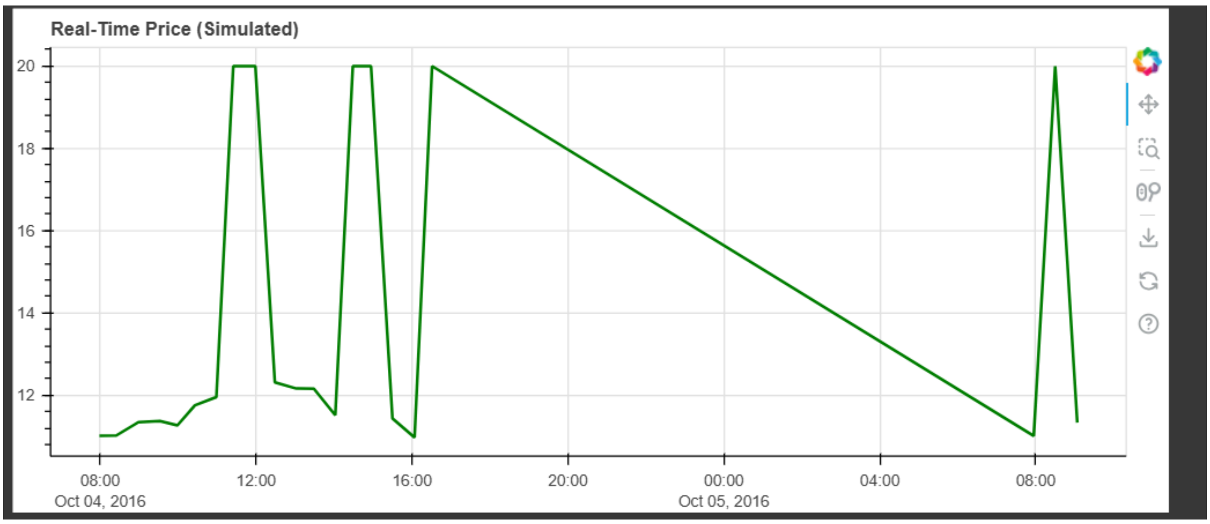
Screenshots from the Panel + Bokeh dashboard demonstrate how pricing updates in real-time with user interaction. The dashboard includes:

- A dropdown to select parking lots
- Live-updating price curves
- Comparison between Model 2 and Model 3

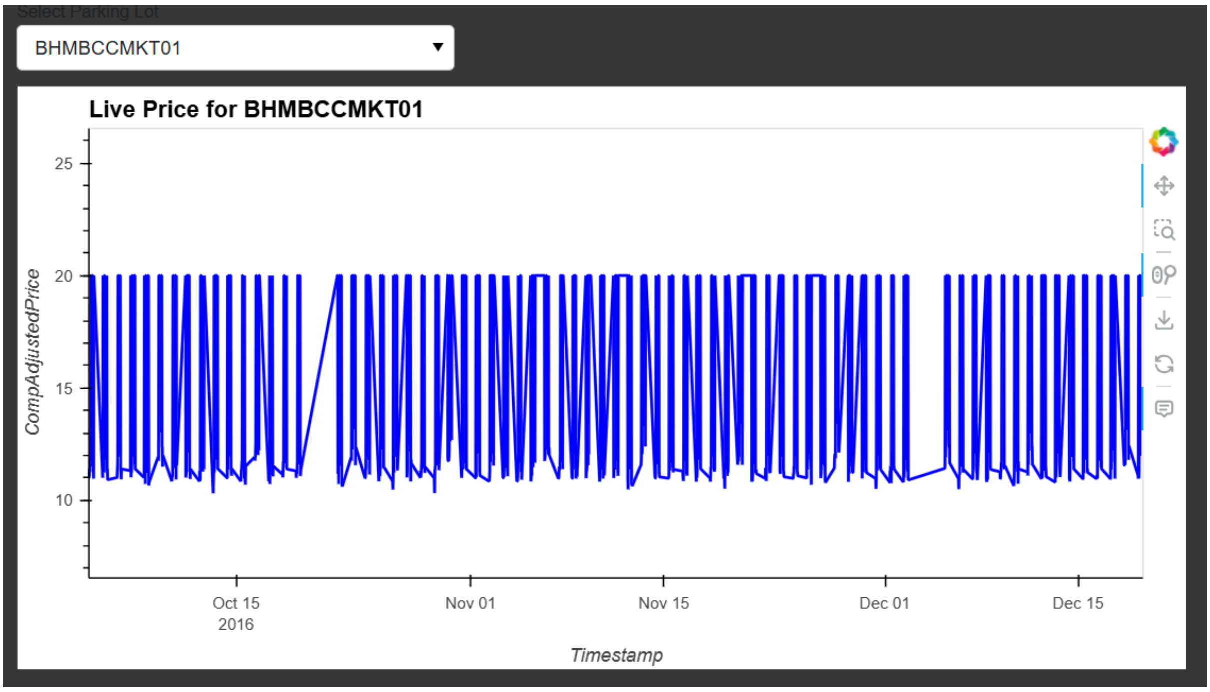
Simulated Live Price



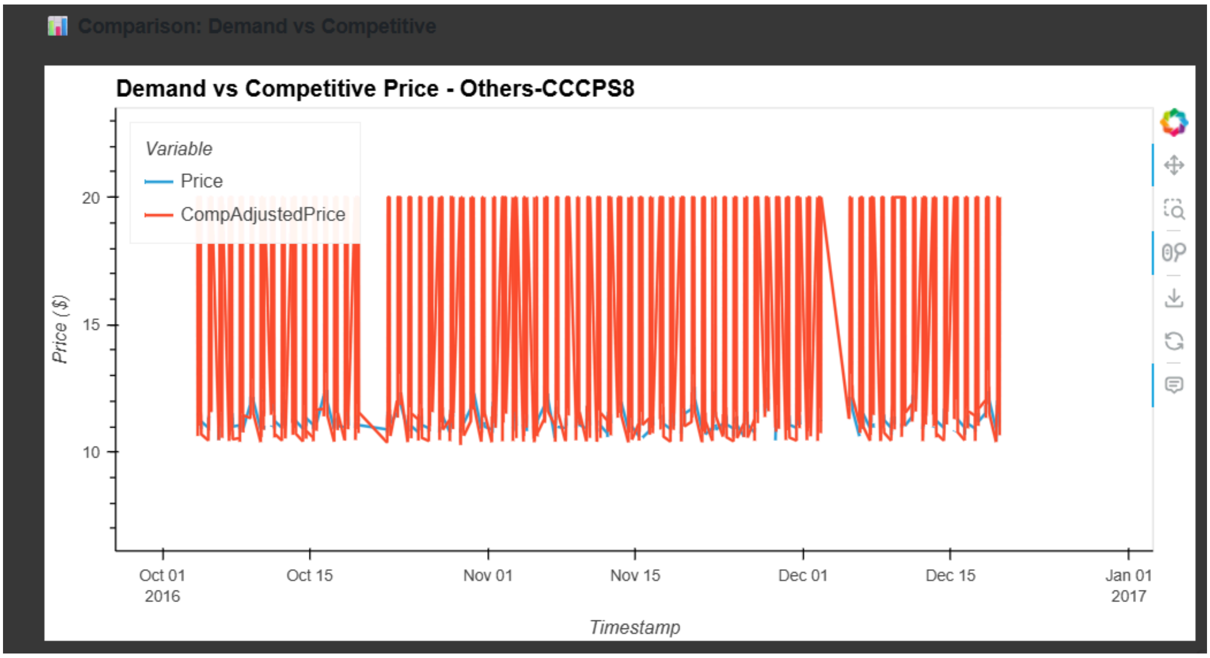
Real Time Price



PARKING LOT PRICE DASHBOARD



DEMAND VS COMPETITIVE PRICE



11. TECH STACK

The project was implemented using a range of modern data science and visualization tools to enable efficient preprocessing, rule-based modeling, and interactive dashboards. Below is a summary of the tools and libraries used:

Programming Language

- **Python 3.11**: The core language used for all data processing, modeling, and visualization tasks.

Libraries and Frameworks

- **NumPy**: For numerical operations and array handling.
- **Pandas**: For data loading, preprocessing, and manipulation.
- **Matplotlib**: Used for static visualizations (e.g., price vs time plots).
- **Bokeh**: For interactive plots embedded in the dashboard.
- **Panel**: Enables creation of an interactive, real-time web dashboard with dropdown and plot updates.
- **Pathway (structure only)**: Schema definition and future streaming compatibility logic.

Development Environment

- **Google Colab**: Used for writing and executing the notebook in a cloud-based Jupyter environment.
- **Python Notebook (.ipynb)**: Used to develop, document, and test each model sequentially.

Version Control and Collaboration

- **GitHub**: Used to manage version control and host the project repository.

This stack allowed for end-to-end implementation — from data ingestion to modeling, visualization, and simulation — entirely within an accessible, open-source ecosystem.

12 . ARCHITECTURE FLOW DIAGRAM

The system architecture illustrates the complete end-to-end pipeline for the dynamic pricing engine. It highlights the key stages from raw data ingestion to final visualization and dashboard output.

Pipeline Overview:

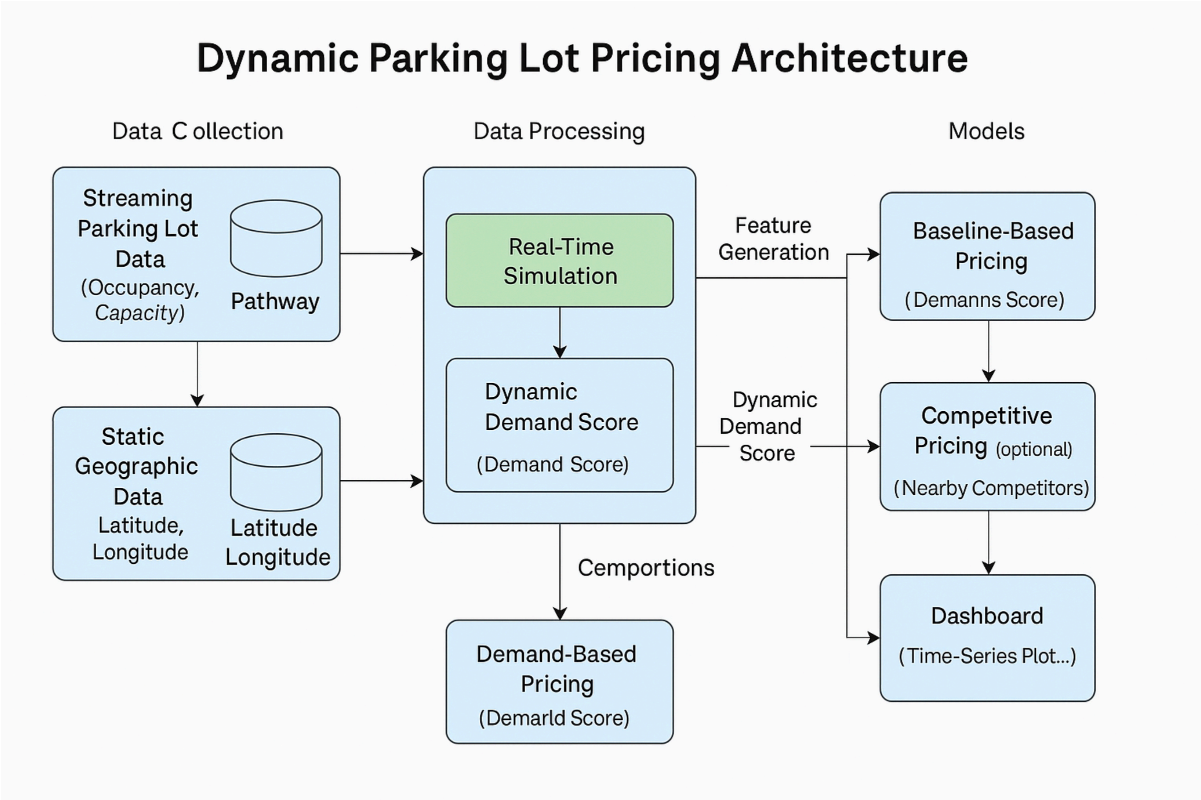
1. **Raw Data Input**
   Timestamped CSV file containing parking lot events.

2. **Data Preprocessing**
   - Timestamp merging
   - Feature engineering
   - Normalization
   - Categorical encoding

3. **Modeling Logic**
   - Model 1: Baseline (Occupancy-based)
   - Model 2: Demand-Based
   - Model 3: Competitive Adjustment

4. **Real-Time Simulation**
   Simulated in Google Colab with row-by-row updates.

5. **Dashboard Visualization**
   Interactive Bokeh + Panel dashboard showing pricing over time.



13. LEARNINGS AND CHALLENGES

This project provided hands-on experience in designing a real-time, rule-based pricing engine using real-world urban data. It involved a combination of data preprocessing, domain understanding, model design, and visualization — all within a tight simulation workflow.

Key Learnings:

- **Dynamic pricing systems** can significantly improve utilization and fairness in public infrastructure when designed with appropriate demand indicators.

- **Feature engineering** plays a critical role in rule-based models. Variables like traffic condition, queue length, and vehicle type have a measurable impact on perceived demand and should be weighted accordingly.

- **Normalization and scaling**, especially in time-based windows (hourly), help prevent disproportionate influence of extreme values and allow consistent pricing logic throughout the day.

- **Competitive pricing logic**, based on real-world geography (via Haversine distance), provides realistic and adaptive behavior, aligning with how actual businesses react in a competitive market.

- **Interactive dashboards**, when integrated with Bokeh and Panel, enhance the interpretability of real-time models and provide actionable insights for decision-makers.

Challenges Faced:

- **Pathway integration limitations** in Google Colab: Due to restricted streaming capabilities in Colab, real-time logic had to be simulated row-by-row rather than implemented as an actual streaming pipeline. However, the underlying logic and schema were structured for future Pathway deployment.

- **Handling missing or inconsistent data**: Some entries had missing traffic, vehicle type, or queue information, which required thoughtful imputation without distorting trends.

- **Balancing price stability vs responsiveness**: While Model 2 was more reactive to demand spikes, Model 3 had to smooth pricing variations while remaining competitive — a non-trivial trade-off.

- **Tuning feature weights**: Since the models were heuristic, identifying optimal weights for the demand formula required experimentation and interpretation rather than purely statistical training.

This combination of technical implementation and practical reasoning has provided a comprehensive understanding of how data-driven systems can be applied to real-world urban management problems.

14. FUTURE WORK

While the current system successfully simulates a real-time dynamic pricing engine using rule-based models and streaming logic, several improvements and extensions can further enhance its accuracy, scalability, and real-world applicability.

---

1. Full Pathway Integration

Although the pipeline was simulated in Google Colab, the system is structured to support streaming environments. A natural next step would be to deploy the models using **Pathway's streaming engine** for real-time processing, schema validation, and continuous input handling.

---

2. Machine Learning–Based Pricing Models

Future iterations can replace heuristic formulas with **ML regression models** or **reinforcement learning**:

- Train a model to predict optimal prices using historical data and outcomes.
- Use reinforcement learning to optimize long-term revenue based on pricing actions and observed demand.

📡 3. IoT Sensor Integration

Deploy smart parking sensors to provide real-time data such as:

- Entry/exit timestamps
- Vehicle detection
- Exact queue lengths

This would reduce dependency on manual or batch data and enable automatic updates to occupancy and demand indicators.

---

4. Driver-Facing Notifications or Mobile App

Integrate a **mobile interface or display system** for drivers to:

- View live prices
- Navigate to cheaper lots nearby
- Receive discounts during off-peak hours

This would improve user adoption and balance parking load dynamically.

---

5. User Behavior Feedback Loop

Collect driver feedback, payment data, and lot selection patterns to adjust pricing models over time. This adds a behavioral learning layer on top of demand and competition-based strategies.

---

6. Scalability Across Cities

Once tested in one urban environment, this pricing engine can be scaled to:

- Different cities with custom traffic models
- Public vs private parking zones
- Event-specific surges (e.g., sports stadiums, festivals)

These future enhancements would make the system not only more intelligent and efficient but also more aligned with **real-time operations**, **public policy objectives**, and **smart city ecosystems**.

15. REFERENCES

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/

3. Summer Analytics, 2025. Retrieved from: https://www.caciitg.com/sa/course25/

4. **Summer Analytics 2025 Problem Statement & Dataset** –
Provided by the Consulting & Analytics Club, IIT Guwahati.

5. Google colab Sample Notebook