Summer Analytics Programme- 2025

Name: - Janvi Balani

Project name: - Dynamic Pricing for Urban Parking Lots

GitHub Username: - JB09-ux

Dynamic Pricing for Urban Parking Lots - Project Report

1. Introduction

This report details the development of a dynamic pricing system for 14 urban parking lots, a capstone project for Summer Analytics 2025, hosted by the Consulting & Analytics Club in collaboration with Pathway. The project leverages real-time data processing and visualization to optimize parking lot utilization based on demand, traffic conditions, and competition. The solution is implemented in Google Colab using Python, Pandas, NumPy, Pathway, and Bokeh, with three pricing models of increasing complexity.

1. Project Implementation Steps

i) Environment Setup and Library Installation

- Installed required libraries (`pathway`, `bokeh`) using `!pip install pathway bokeh --quiet` to ensure compatibility.

- Imported libraries including `pathway`, `pandas`, `numpy`, `bokeh`, and `panel` for data processing and visualization.

- Justification: These libraries provide robust tools for real-time data streaming (Pathway), data manipulation (Pandas, NumPy), and interactive plotting (Bokeh, Panel).

ii) Data Preprocessing

- Loaded the dataset (`dataset.csv`) and combined `LastUpdatedDate` and `LastUpdatedTime` into a `Timestamp` column using `pd.to\_datetime`.

- Mapped `TrafficConditionNearby` to numerical values (`low: 0`, `average: 1`, `high: 2`) for Model 2.

- Selected relevant columns (`Timestamp`, `SystemCodeNumber`, `Occupancy`, `Capacity`, etc.) and saved to `parking\_stream.csv`.

- Computed nearby parking lot pairs within 1 km using the Haversine formula and saved to `nearby\_pairs.csv`.

- Justification: Preprocessing ensures data is in a suitable format for Pathway streaming and enables spatial analysis for Model 3.

iii) Schema Definition and Data Streaming

- Defined a `Parking` schema using `@pw.decl.schema` with fields like `Timestamp: pw.timestamp`, `Occupancy: int`, etc.

- Tested schema definition with error handling to ensure compatibility.

- Read data into Pathway streams (`parking\_stream`, `nearby\_data`) using `pw.io.csv.read`.

- Justification: Schema definition ensures type safety, and streaming enables real-time processing, aligning with project requirements.

iv) Model Development

a) Model 1 (Baseline Linear Model): `price = 10 + 10 \* (Occupancy / Capacity)`.

- Calculates a base price of $10, increasing linearly with occupancy rate.

b) Model 2 (Additional Factors): `price = 10 + 10 \* (Occupancy / Capacity) + 2 \* traffic\_factor + 0.5 \* QueueLength + 5 \* IsSpecialDay`.

- Adds adjustments for traffic, queue length, and special days.

c) Model 3 (Competitor Prices): `price = base\_price - 0.5 \* avg\_competitor\_price`, where `base\_price` is from Model 1, and `avg\_competitor\_price` is the average price of nearby lots.

- Uses joins to incorporate competitor data within 1 km.

- Justification: These models progress from simple to complex, meeting the project’s requirement for three pricing strategies.

v) Pipeline Execution

- Defined outputs using `pw.io.output` for each model.

- Ran the Pathway pipeline with `pw.run(outputs={...})` to compute prices.

- Collected results into Pandas DataFrames (`model1\_data`, `model2\_data`, `model3\_data`) using `to\_pandas()`.

- Justification: This ensures real-time computation and compatibility with Bokeh visualization.

vi) Visualization

- Created tabbed visualizations for each model using Bokeh and Panel.

- Added scatter points and custom date ticks ("Oct 1", "Oct 15", etc.) on the x-axis.

- Integrated `HoverTool` to display `Date` and `Price ($)` on hover.

- Justification: Interactive visualizations enhance usability, and custom ticks align with the dataset’s date range (October-December 2016).

1. Demand Function, Assumptions, and Price Dynamics

i) Demand Function

The demand function is implicitly modeled through occupancy as a proxy for demand. The base price starts at $10, with adjustments based on:

* Model 1: Linear increase with `Occupancy / Capacity`, reflecting higher demand with more occupied spaces.
* Model 2: Additional increases with `traffic\_factor`, `QueueLength`, and `IsSpecialDay`, capturing external demand pressures.
* Model 3: Adjustment based on `avg\_competitor\_price`, suggesting demand sensitivity to competitive pricing.

ii) Assumptions

- Occupancy accurately reflects demand, with higher occupancy indicating higher willingness to pay.

- Traffic conditions, queue length, and special days proportionally affect demand and are quantifiable with fixed coefficients.

- Nearby lots within 1 km are direct competitors, and their average price influences local demand.

- The dataset (October-December 2016) is representative of typical parking behaviour.

- Justification: These assumptions simplify the model while aligning with the problem statement’s focus on real-time data and basic economic theory.

iii) Price Changes with Demand and Competition

* Demand: In all models, price increases with higher `Occupancy / Capacity`. Model 2 further raises prices with high traffic, long queues, or special days, reflecting increased demand pressure.
* Competition: Model 3 lowers prices if nearby lots have higher average prices, ensuring competitiveness. For example, if a lot’s base price is $15 and the average competitor price is $20, the adjusted price becomes $15 - 0.5 \* $20 = $5, incentivizing usage.
* Justification: These dynamics optimize utilization by balancing demand-driven increases with competitive reductions.

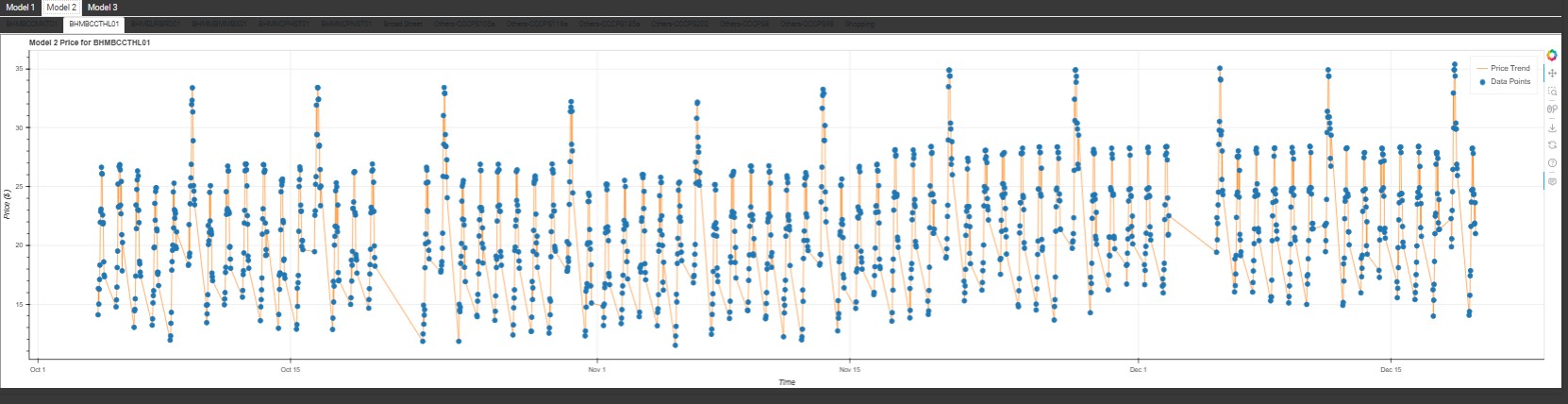
4. Visualizations

The Bokeh visualizations are embedded in the Colab notebook and included here as screenshots:

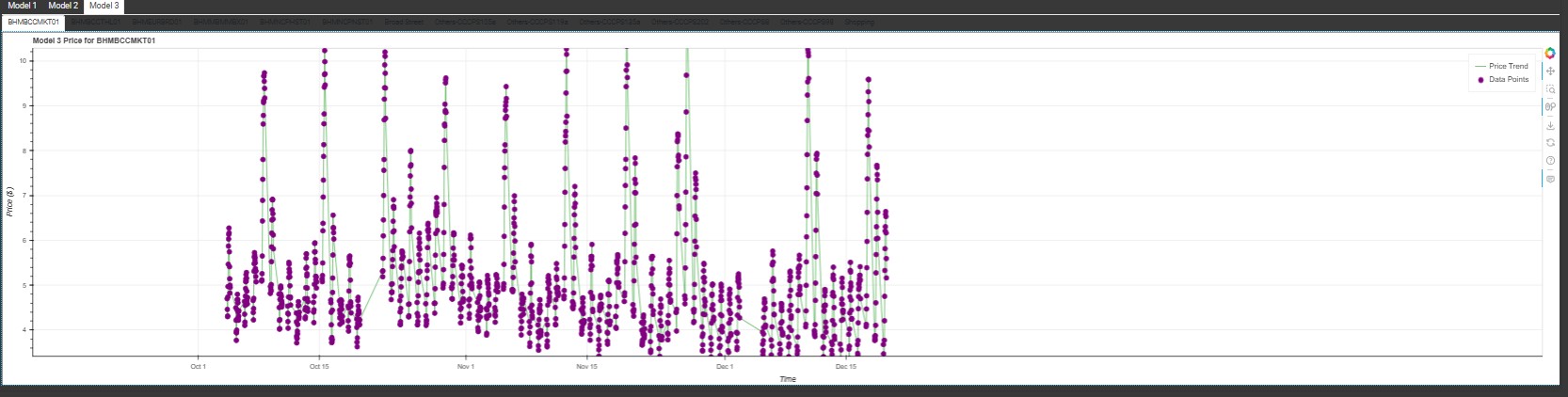
i) Model 1: Shows a linear price trend with dots, custom date ticks, and hover tooltips.



ii) Model 2: Includes additional factors, with distinct color coding and hover details.



iii) Model 3: Reflects competitor adjustments, with hover showing date and adjusted price.



5. Conclusion

This project successfully implements a dynamic pricing system with three models, leveraging Pathway for real-time processing and Bokeh for interactive visualization. The approach meets the capstone requirements, providing a foundation for further optimization with machine learning or additional data sources.