# Capstone Project Report: Dynamic Pricing Based on Traffic and Demand

Harshita Garg

## Overview

This project implements a dynamic pricing system that adapts to real-time traffic conditions and demand using Pathway and visualizes the data using Bokeh. We explore and compare two separate pricing models, both aiming to reflect traffic intensity and user demand, and generate real-time price suggestions.

## Data Flow & Setup

* **Streamed Data Source**: Data is ingested live using Pathway.
* **Key Columns**:
  + vehicle\_type: Encoded as numeric (0.7 = Bike, 1 = Car, 1.5 = Truck).
  + cap: Maximum capacity for a lane.
  + current\_occ: Current occupancy in the lane.
  + QueueLength: Proxy for congestion.
  + traffic\_condition: External traffic level input.
  + delta: Captures volatility or change in occupancy.
  + is\_special\_day: Boolean indicator for surge-pricing days.
  + vehicle\_weight: Represents per-vehicle weight in pricing.
  + timestam: Timestamp of the data point.

## Vehicle Type Encoding

Vehicle types are encoded numerically for processing:

* 0.7 → **Bike**
* 1.0 → **Car**
* 1.5 → **Truck**
* 0.5 → **Cycle**

These were later mapped to readable labels for plotting and interpretation.

## Model 1: Base Demand Function

### Stepwise Logic:

1. Normalize occupancies
2. Multiply by a sensitivity coefficients
3. Add to base price

### delta\_occ = (avg\_occ\_now - avg\_occ\_prev) / cap

### load\_factor = current\_occ / cap

### price = base\_price + α \* delta\_occ + β \* load\_factor

### Assumptions

* Queue length reliably reflects real-time demand.
* All vehicles are affected equally by congestion.

## Model 2: Composite Demand Function with Log, Sigmoid, and Weighted Coefficients

This model refines demand estimation using multiple traffic indicators, sigmoid normalization, and weighted traffic contributions.

### Stepwise Logic:

1. **Normalize occupancy using a sigmoid function**:

f\_occ = 1 / (1 + exp(-10 \* (occupancy\_rate - 1)))

1. **Log-normalize queue length**:

f\_queue\_norm = log(QueueLength + 1) / log(max\_queue + 1)

1. **Normalize traffic intensity**:

f\_traffic = log(traffic\_condition + 1) / log(max\_traffic + 1)

1. **Compute demand factor**:

demand\_factor = (3 \* f\_occ + 1.5 \* f\_queue\_norm + 1.2 \* f\_traffic + 1.3 \* delta) \* (is\_special\_day + 1) \* vehicle\_weight

### Explanation of Terms:

* **Sigmoid of Occupancy Rate** (3× weight): Saturates at high occupancy and remains sensitive around threshold 1.
* **Normalized Queue** (1.5× weight): Smooths queue spikes while maintaining comparability.
* **Log Traffic Condition** (1.2× weight): Reflects ambient road traffic conditions.
* **Delta** (1.3× weight): Captures sudden jumps in occupancy.
* **Special Day Multiplier**: Doubles pricing signal during special days (is\_special\_day + 1 gives either 1 or 2).
* **Vehicle Weight**: Encodes per-vehicle type impact on pricing.

### Pricing Rule

price = base\_price \* (1 + 0.6 \* demand\_factor)

### Assumptions

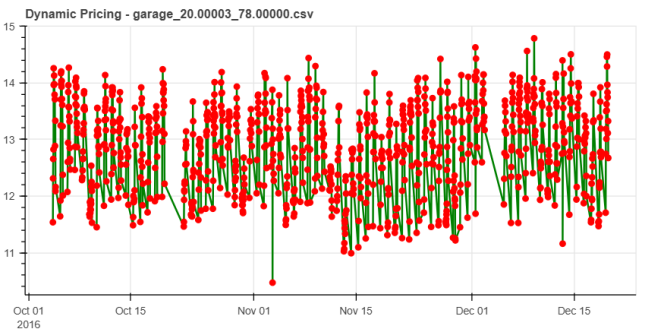
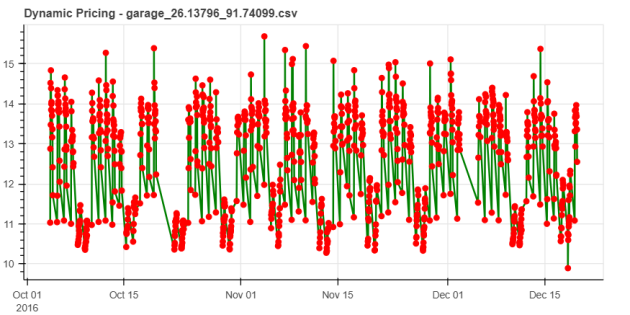
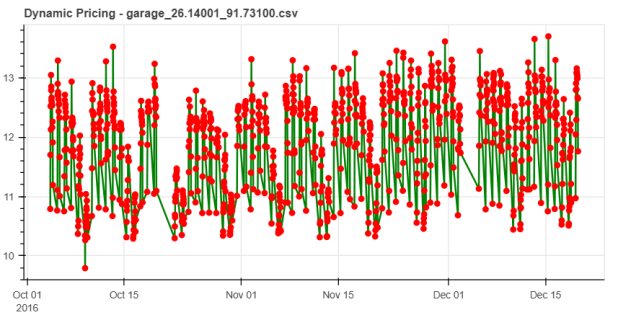
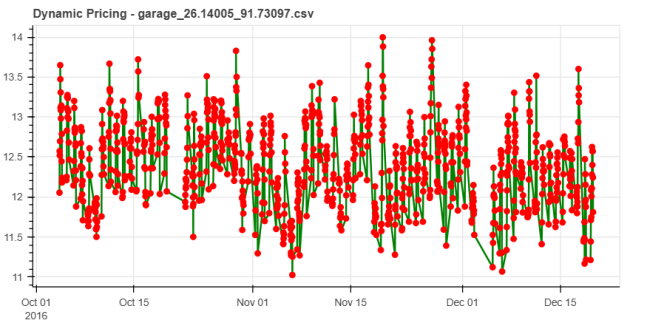
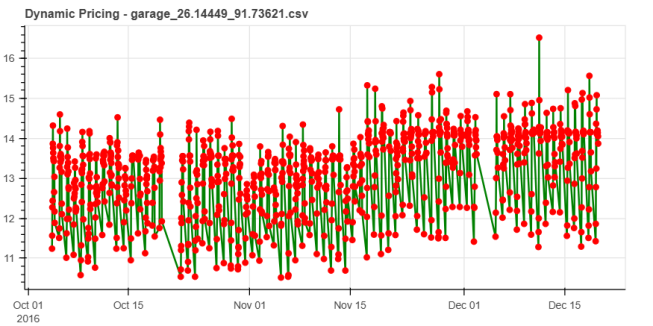
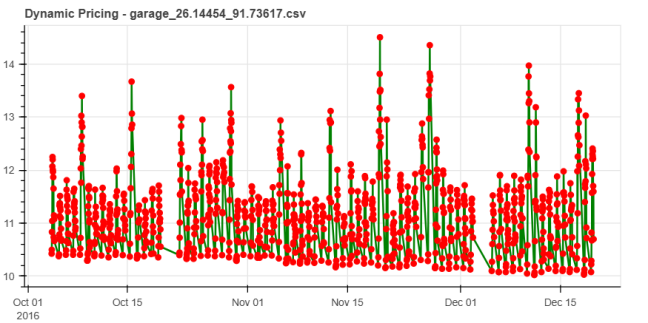
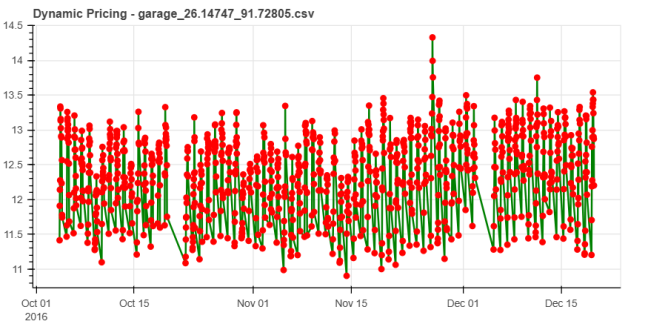
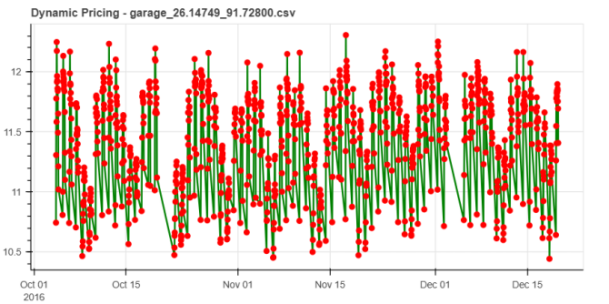
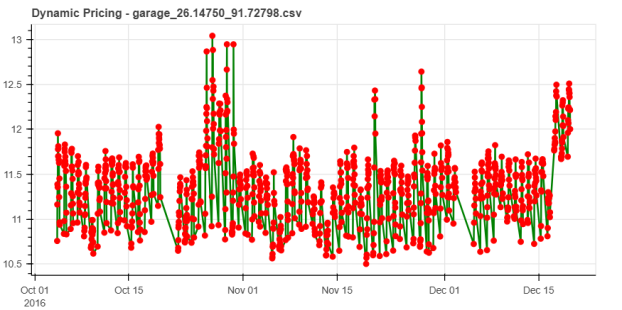
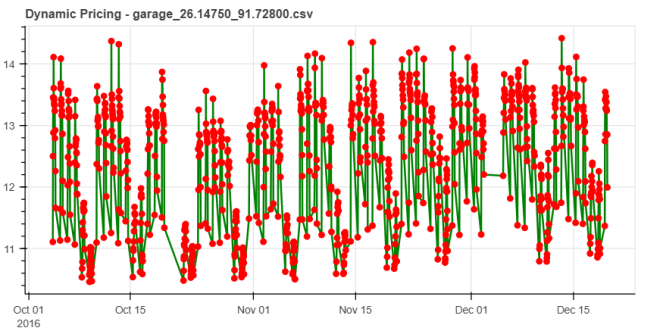
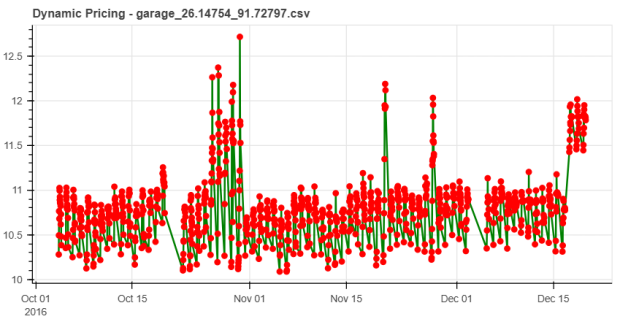
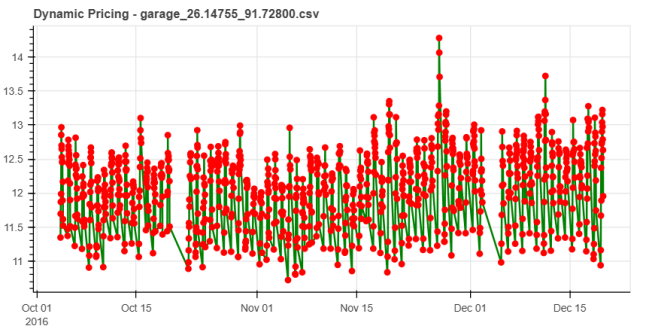
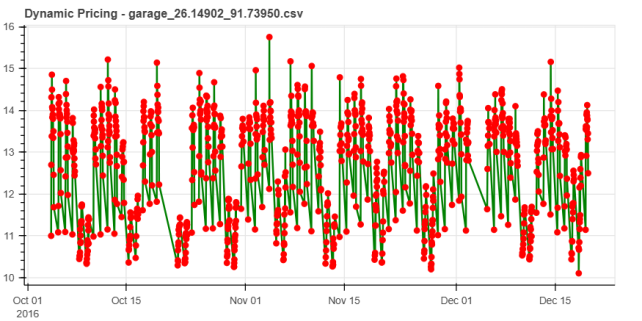
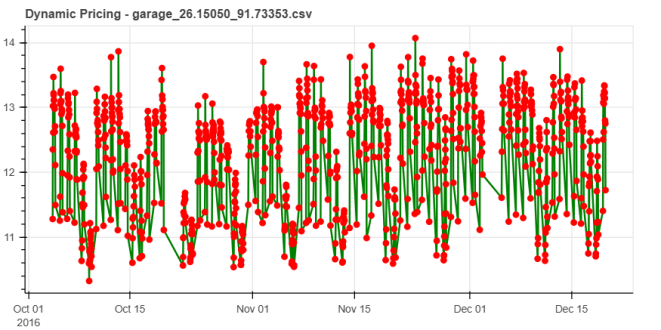
* Logarithmic and sigmoid normalization improves stability.
* Multiplicative structure increases flexibility.
* Coefficients are manually tuned.
* No explicit branching; model reacts smoothly to inputs.

## Visualizations

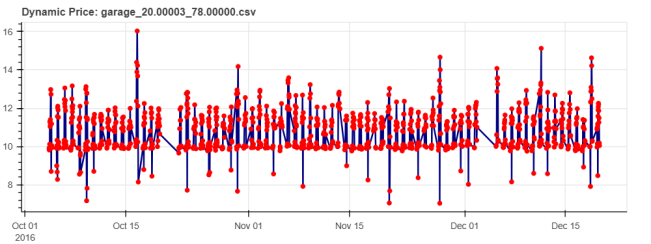
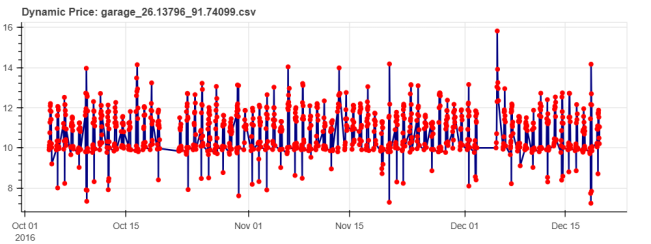
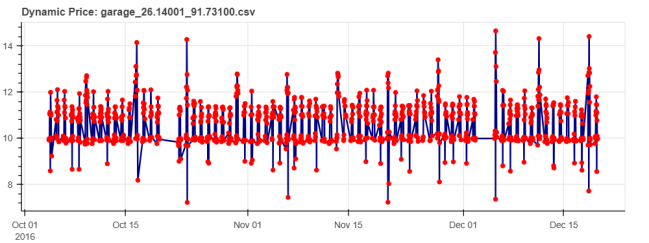
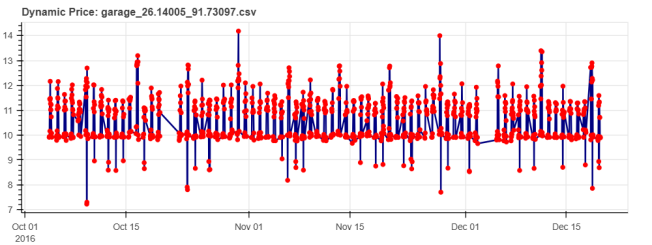
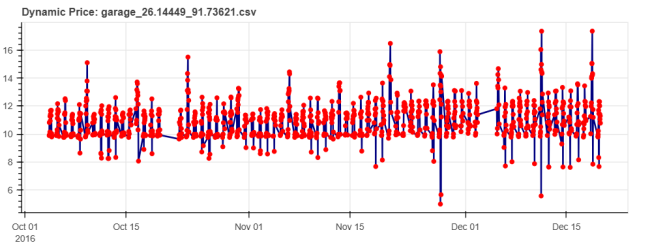
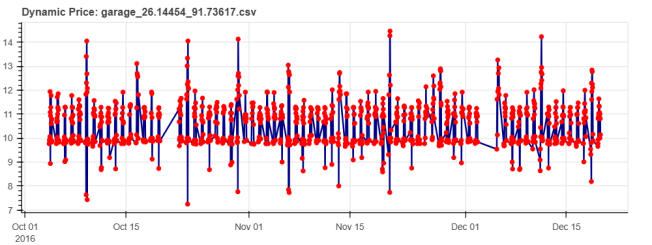
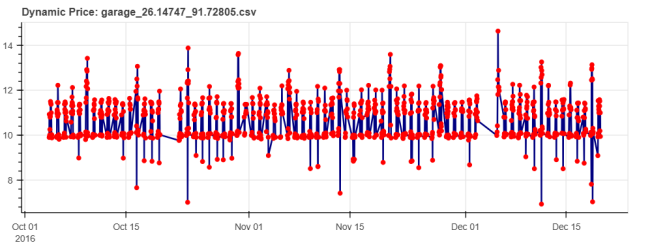
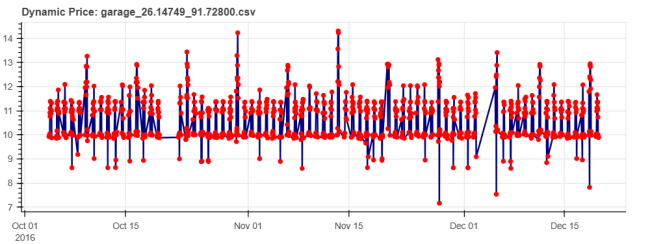
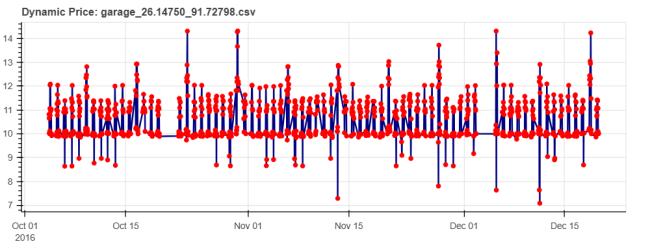
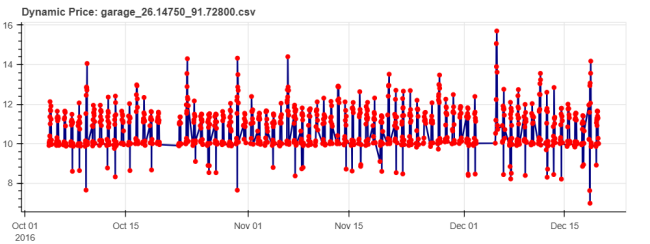
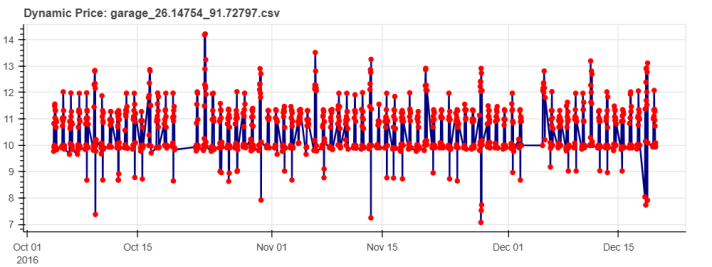
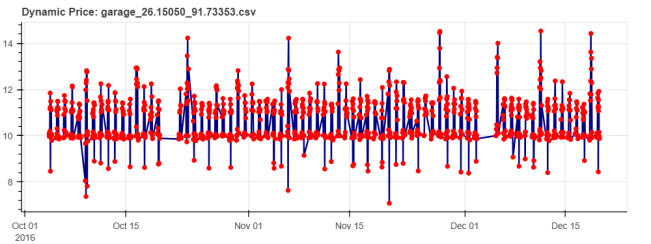
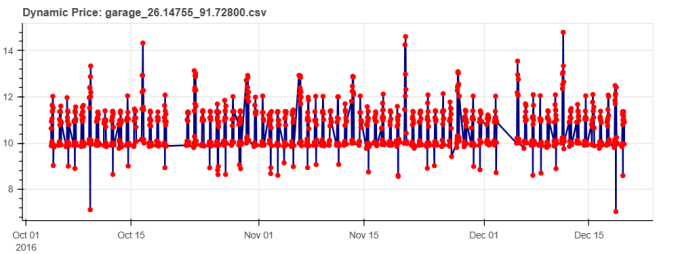
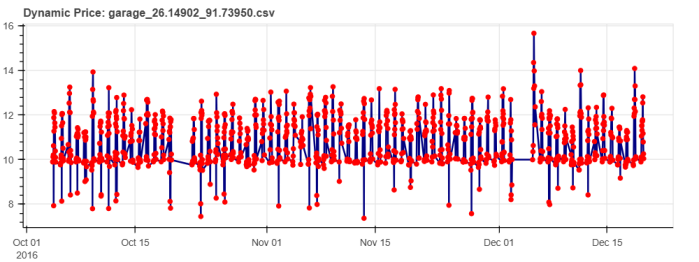
### Bokeh (Interactive)

* Used for plotting price vs. time for each vehicle type.
* Legend based on decoded vehicle labels.
* Output embedded directly in the Colab notebook.

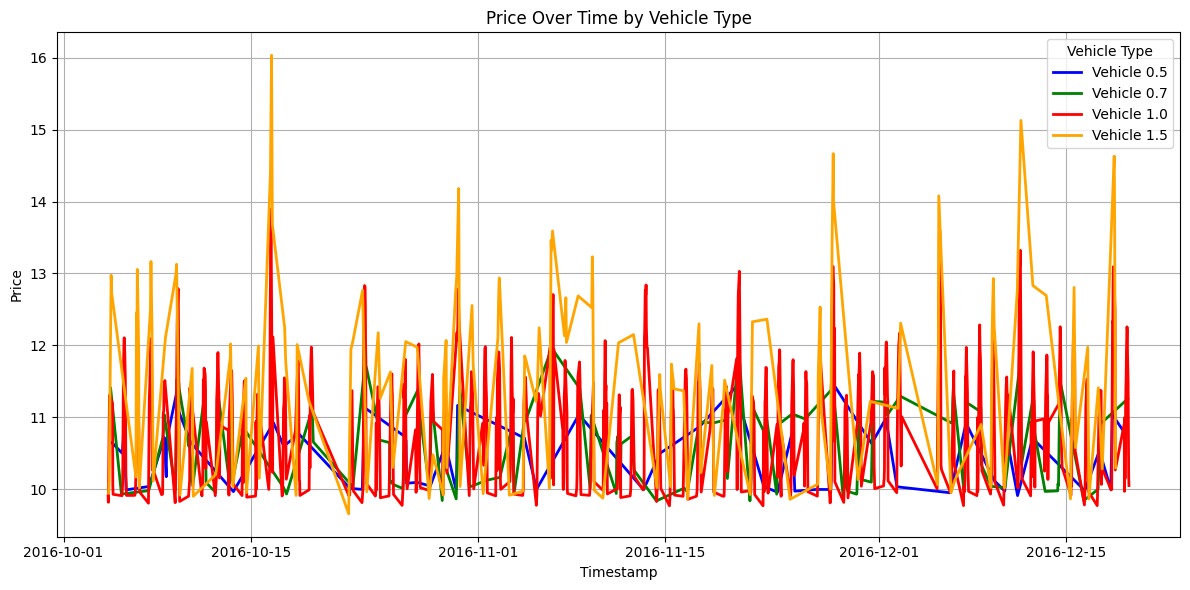
Model 1



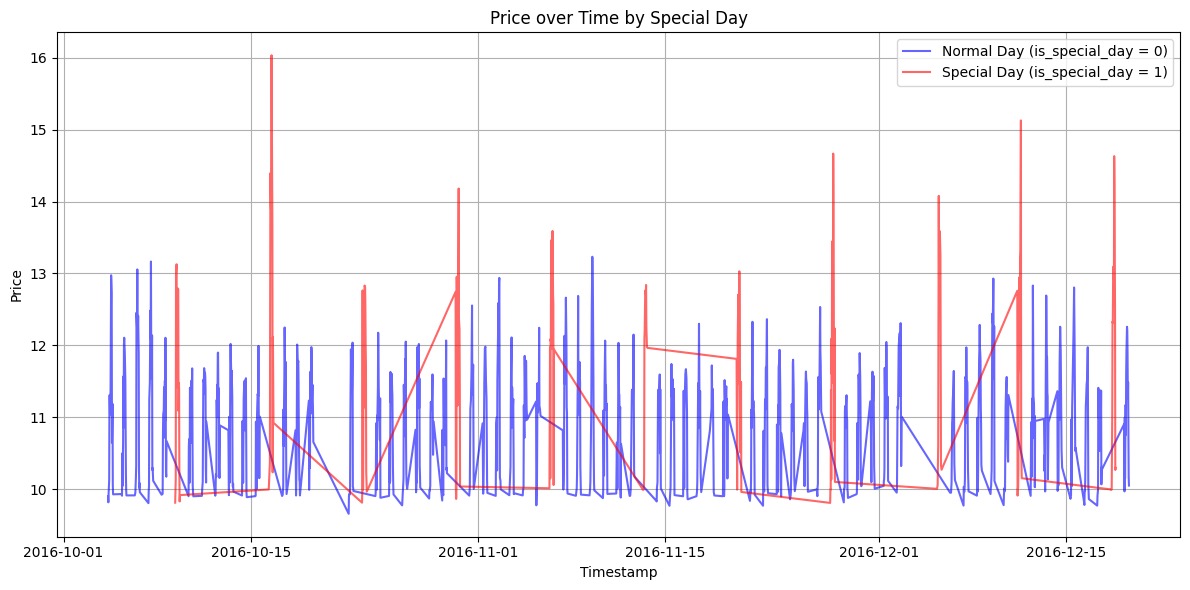
MODEL 2

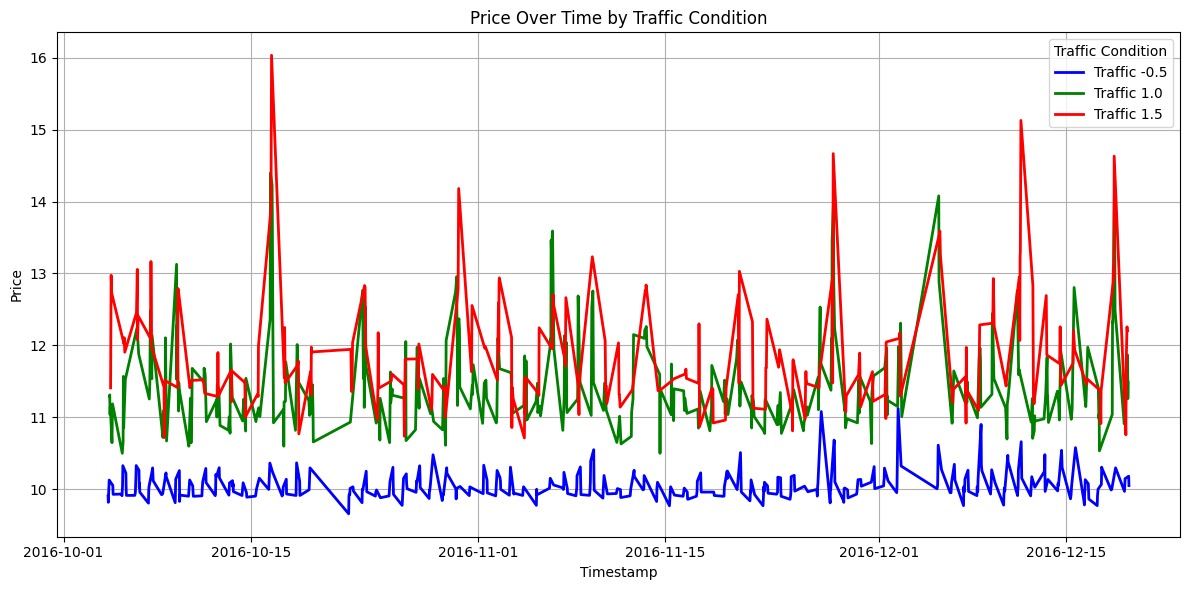


Vehicle Variation in price -



Is Special Day

 Traffic Condition



## Limitations

* Queue length might be affected by sensor noise.
* Encoding assumptions (e.g., 0.7 = bike) need universal consistency.
* Pricing functions are manually tuned; not yet optimized through feedback.

## Conclusion

Two pricing models were developed:

1. **Model 1**: Based on a very simple linear relation.
2. **Model 2**: Uses a six-factor demand function including sigmoid, logarithmic, delta, and traffic weighting.

Visualizations validate the model output, showing rising prices with growing congestion. Further improvements could include:

* Feedback loop using historical behavior.
* Time-of-day adjustments.
* Differential pricing per vehicle type.