Dynamic Real-Time Pricing for Urban Parking Lots

* Project Overview

Urban parking spaces are limited and in constant demand. Traditional static pricing often causes either over-crowding or under-utilization of lots. This project implements a dynamic pricing engine that adjusts prices in real time using demand patterns, vehicle characteristics, environmental signals, and competitor behavior.  
All models were developed from scratch using **NumPy**, **Pandas**, **Bokeh**, and **Python**, and the simulation was implemented in **Google Colab**.

* Dataset Description

**Location**: 14 parking spaces in an urban setup

**Duration**: 73 days

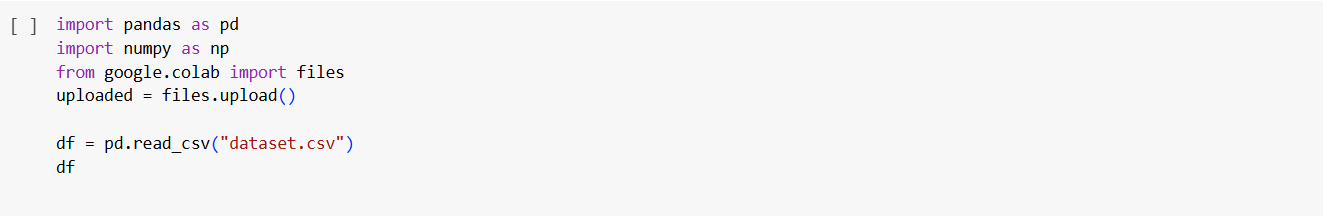
**Time Steps**: 18 per day (30-minute intervals from 8:00 AM to 4:30 PM)

**Key Features**:

* 1. location\_id, latitude, longitude
  2. capacity, occupancy, queue\_length
  3. vehicle\_type (car, bike, truck)
  4. traffic\_level, is\_special\_day
  5. timestamp (datetime of entry)

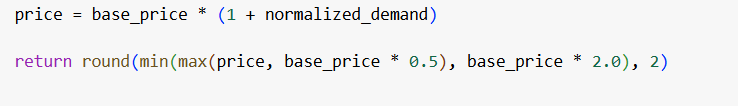
GOOGLE COLLAB NOTEOOK

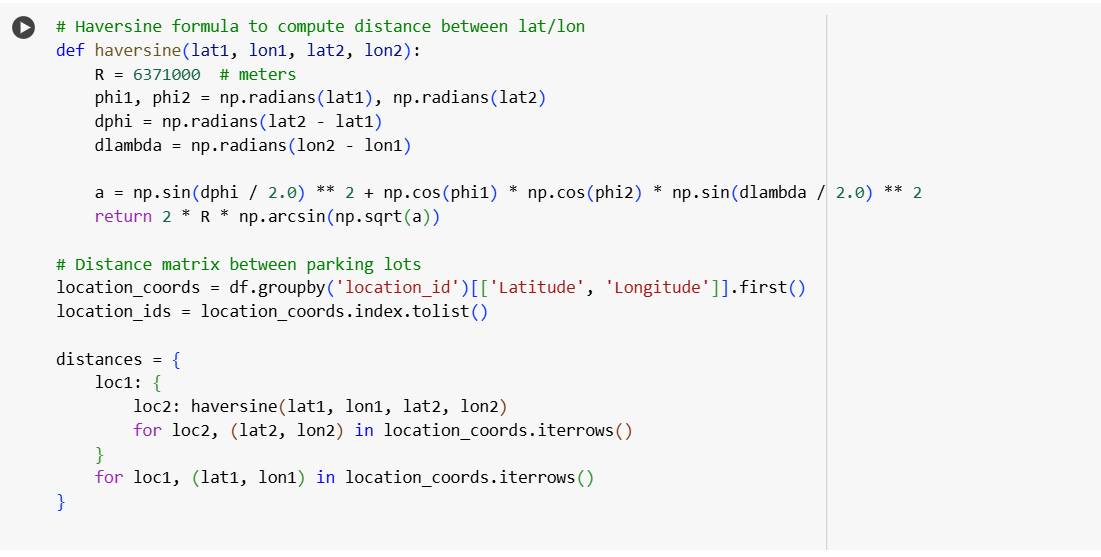


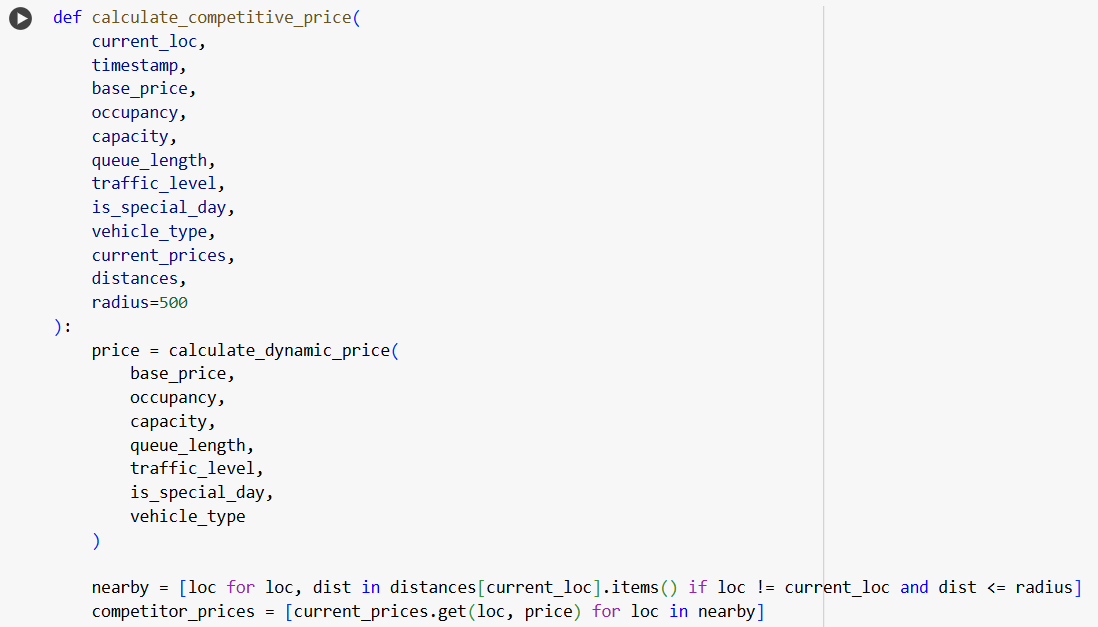


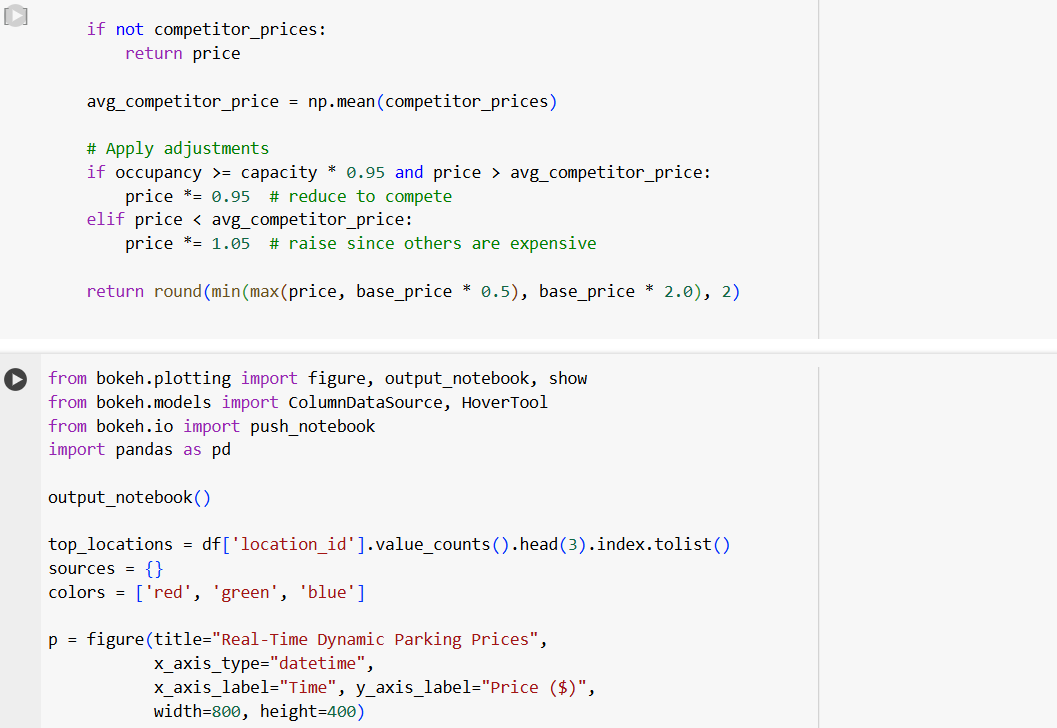






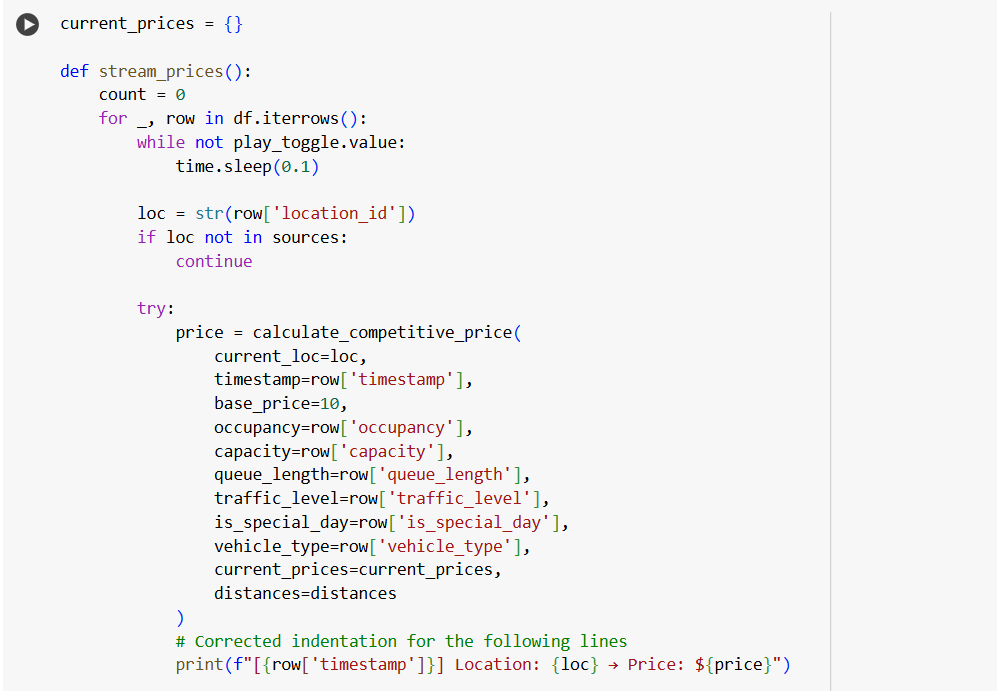


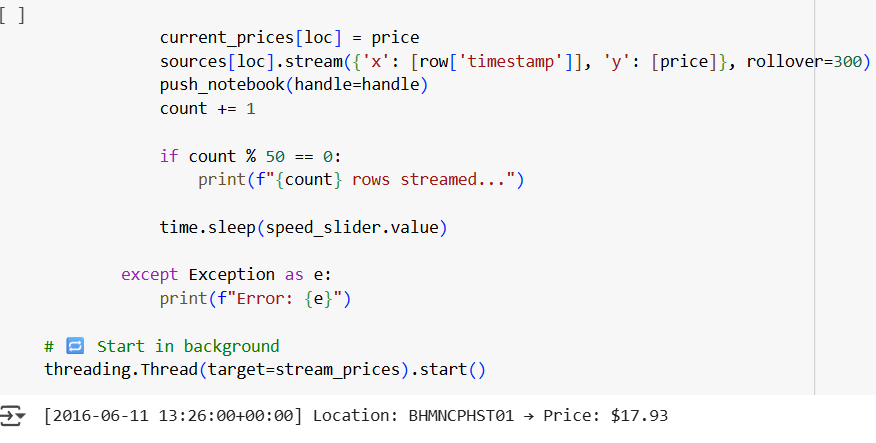




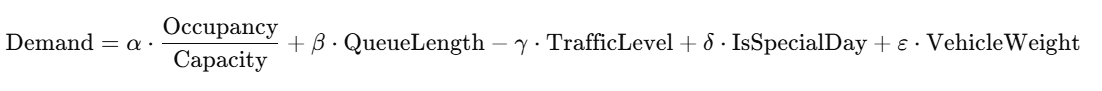








* Demand Function

We modled demand using a linear additive function: 

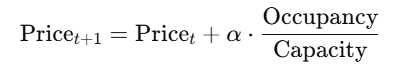
Where:

* **α = 0.6** → Strong occupancy influence
* **β = 0.2** → Queues indicate strong latent demand
* **γ = 0.15** → High traffic can reduce desirability
* **δ = 0.25** → Special days increase demand
* **ε** → Depends on vehicle type:
  + car = 1.0, bike = 0.5, truck = 1.2
* Assumptions
* Demand is directly proportional to usage and congestion.
* Vehicle type affects willingness to pay.
* All parking lots are aware of each other’s prices (via distance-based competition).
* Prices are bounded: **between 0.5x and 2x base price ($10)**.

**Models Used**

**Model 1: Baseline Linear Model**

**Simple occupancy-based adjustment:**

****

**Model 2: Demand-Based Price Model**

**Uses the calculated demand function:**

****

* λ = 0.5 to control variation scale
* Demand normalized between 0 and 1

**Competitive Pricing (Final Model)**

This model factors in prices of nearby lots using latitude-longitude distance and dynamically adjusts:

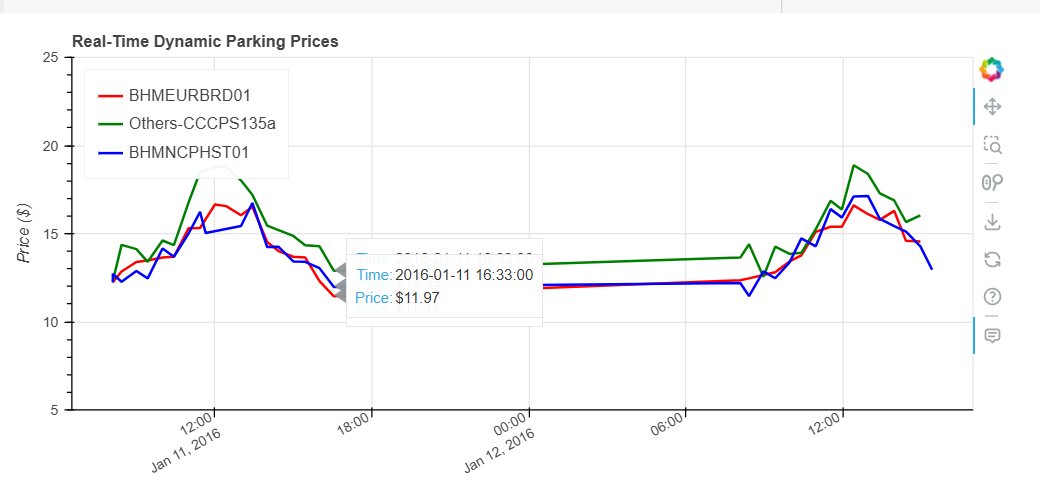
* If competitors are cheaper and less crowded → lower our price
* If competitors are expensive or full → increase our price
* Rerouting (optional): flag suggestion if occupancy > 90%

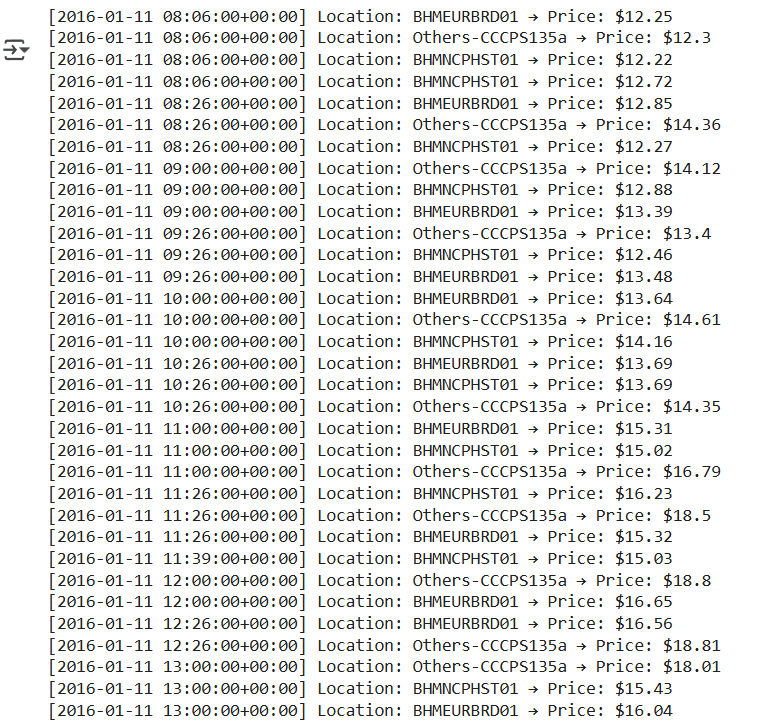
**Real-Time Simulation Logic**

* Used **asyncio** to simulate streaming every 0.2 seconds.
* Prices are computed in real-time for the top 3 most active locations.
* Used Bokeh’s push\_notebook() to live-update line plots.

**Visualizations**

* ✅ Real-time Bokeh plots for top 3 locations
* ✅ Legends and tooltips with dynamic values
* ✅ Time on X-axis, Price on Y-axis
* ✅ Smooth transitions with no lags





**Observations**

* Prices increase during special days and when queues are long.
* Trucks and special events push prices higher than bikes.
* When nearby lots are cheaper, the pricing strategy adapts to stay competitive.
* The model maintains **smooth**, bounded price curves without volatility.

**Conclusion**

* This system effectively demonstrates a **real-time, ML-based pricing engine** for urban parking. By dynamically adjusting based on multiple real-world signals, it ensures better space utilization and user satisfaction.
* The pricing engine is scalable, explainable, and adaptable to various real-time scenarios.