Documentation: Demand-Based Dynamic Pricing Model

Project Title:

Real-Time Dynamic Pricing System Using Demand Estimation with Adaptive Lambda

---

Problem Overview:

This model addresses **real-time dynamic pricing** using demand-related features such as occupancy, queue length, traffic, special days, and vehicle types. The price must respond **proportionally and immediately** to fluctuations in demand — increasing during high demand and relaxing when demand drops.

Objective:

To compute **price** based on a continuously updated demand function using:

- Feature-weighted linear demand estimation.
- Real-time adjustment of the price sensitivity parameter ().
- Adaptive learning of feature weights using online regression.

Input Features:

Feature	Description
`Occupancy / Capacity`	Demand ratio
`QueueLength`	Current queue of waiting vehicles
`TrafficConditionNearby`	Encoded as a score (low=0.2, high=0.8, etc.)
`IsSpecialDay`	Boolean (1 if special day, else 0)
`VehicleTypeWeight`	Type-based weight (car=1.0, truck=2.0, etc.)

Model Components:

Demand Function:

$$\text{Demand} = \alpha \cdot \frac{\text{Occupancy}}{\text{Capacity}} \cdot \text{VehicleWeight} + \beta \cdot \text{QueueLength} + \gamma \cdot \text{TrafficScore} + \delta \cdot \text{IsSpecialDay} + \epsilon$$

Price Update Equation:

$$\text{Price}_t = \text{BasePrice} \cdot (1 + \lambda_t \cdot \text{NormalizedDemand})$$

Lambda Tuning Logic:

- () is not fixed.
- Updated **dynamically** based on the **difference** between current demand and a `minimum demand`
- Learning Rate controls how fast () responds to changes.

```
error = norm_demand - self.d_min
lambda += learning_rate * error
lambda = clip(lambda, -0.5, 1.0)
```

This ensures price adjustments remain between **0.5× to 2×** of base price.

Online Weight Learning:

Instead of training offline, weights (, , , ,) are learned using an **online regression** logic:

```
X_list → stores feature vectors
y_list → stores observed price deltas

Every 20 steps:
    weights ← least squares solution from X and y
```

This keeps the model **self-improving**.

Visualization:

- Live graph generated using **Bokeh**
- Plots **price vs. timestamp** in real-time
- Useful to observe:
    - Price fluctuations
    - Daily or event-driven spikes
    - Overfitting or smoothing behavior

Key Strengths:

- Lightweight, no ML libraries needed

- Adaptable and interpretable
- Fully real-time and responsive
- Modular design (can plug in other demand models)

📅 Deliverables:

- Real-time CSV streaming parser
- Adaptive price engine class
- Lambda control mechanism
- Feature encoder maps
- Live price plot