CSE 250 Final Project Report

Group Members

Vaibhav Sharma (A69042944), Chhavi Kirtani (A69041408), Tarika Gupta (A69041489), Ayushi Mishra (A69042734)

Problem Description

Dynamic pricing involves adjusting product prices over time in order to optimize revenue under uncertain, fluctuating, and often partially observable demand conditions [**Dan**]. The central challenge lies in the fact that consumer demand is not static: it evolves with factors such as seasonality, competitor behavior, macroeconomic conditions, and even short-term trends such as promotions or stock availability. When prices are not aligned with true market conditions, whether due to incorrect assumptions, outdated expectations, or failure to incorporate new information, a vendor may undervalue its products and forgo substantial profit or overprice them and drive customers toward competitors.

Since both scenarios negatively impact revenue for a given vendor, efficient price adjustment is essential for maintaining competitiveness and ensuring stable supply–demand alignment within a dynamic market. In modern e-commerce settings, where pricing decisions must often be made at daily or even hourly granularity, automated and data-driven techniques become not just beneficial but necessary for a business's operations. We chose this problem since it is inherently tied to probabilistic reasoning and learning because pricing decisions must be made in the face of uncertainty about customer behavior. Demand at any given price is not completely observable. Instead, it can be represented as a distribution driven by multiple latent factors. Conceptually, a dynamic pricing model aims to update a vendor's beliefs about how likely a customer is to purchase a product at a given price, and how those beliefs evolve over time as new transactional evidence is observed.

From this perspective, the pricing system functions like a belief network or Bayesian agent that must continually update its understanding of demand elasticity as it receives feedback in the form of realized sales quantities. This framing naturally motivates us to use probabilistic models and learning-based approaches because they provide formal mechanisms for representing uncertainty, updating beliefs, and making optimal decisions under those uncertainties.

Data Sourcing and Preprocessing

Data Sourcing

The dataset used in this project is the Online Retail II Dataset from UCI Machine Learning Repository. This dataset contains transactional logs from a UK-based online retail store between 2009–2011, with features as below:

- **Invoice Number:** A 6-digit distinct code for each transaction.
- **Stock Code:** A 5-digit distinct code for each product.
- **Description:** Name of product.
- **Quantity:** How many products are sold per transaction.
- **Unit Price:** The price of a single product in pound sterling, official currency of the UK.
- **Invoice Date:** Time and date when a particular transaction has been executed.
- **Customer ID:** A unique 5-digit number assigned to each customer.
- **Country:** The residential country of a particular customer.

This dataset is widely used in the fields of demand forecasting, recommender systems, and market basket analysis, due to its transactional granularity and real-world retail context. Its combination of time-stamped purchase events, customer identifiers, and product-level details makes it valuable for studying repeat purchasing patterns, seasonality, product association mining, and churn prediction. Additionally, the presence of returns and cancellations introduces realistic complexity often encountered in retail analytics, making it suitable for probabilistic reasoning models.

Data Pre-Processing

The dataset contains 525,461 transaction-level observations spanning December 2009 to December 2010, and an initial schema validation confirmed appropriate data types for all fields, enabling accurate grouping and temporal aggregation. To prepare the Online Retail II dataset for analysis, we implemented several preprocessing steps to ensure data quality, reliability, and suitability for demand modeling and dynamic pricing.

Our first step was to determine how to handle missing data. We primarily found missing values in the CustomerID field (20 percent), which is expected in anonymous or guest transactions; these omissions were retained because they do not affect product-level demand modeling. Missing item descriptions were also negligible, as stock codes uniquely identify each product. A more significant issue we found was the presence of negative quantities representing returns or canceled orders, which spanned 12,326 rows. We removed these rows since return behavior does not necessarily reflect forward demand and could bias a pricing model or reinforcement-learning environment designed to simulate purchasing responses.

We implemented outlier analysis, which revealed substantial irregularities in both price and quantity, including negative or zero prices, extreme price spikes, and unusually large order quantities indicative of wholesale activity or administrative adjustments. We removed rows involving invalid prices, excessively high quantities, or non-product stock codes to eliminate noise unrelated to consumer purchasing behavior. We also conducted additional filtering of stock codes since some codes represented financial entries or sparsely sold items, while others exhibited no historical price variability, making them unsuitable for estimating price elasticity.

Country-level analysis showed that approximately 92 percent of transactions originated from the United Kingdom. Therefore, only UK data was retained to avoid mixing heterogeneous markets and to maintain consistent seasonality patterns and simplify the target consumer market we are examining. Temporal data helped us see clear weekday effects and strong seasonal spikes during holiday periods, justifying the inclusion of time-based features in downstream models. Finally, inspection of the raw price–quantity relationship confirmed that transaction-level data is highly noisy due to promotions, wholesale orders, and variability in order size. To mitigate this, we used daily SKU-level aggregation to compute total daily quantity sold and daily average price to reveal stable, meaningful elasticity patterns.

Collectively, these preprocessing steps, such as cleaning invalid entries, removing returns, filtering unsuitable SKUs, restricting to a homogenous market, and aggregating to daily granularity, were essential for constructing a reliable dataset that accurately represents true consumer purchasing behavior. This cleaned dataset provides the foundation for demand modeling and forms the state dynamics of the reinforcement-learning environment, ensuring that simulated pricing decisions generate realistic and interpretable demand responses.

Modeling Overview

Since pricing decisions influence future customer behavior, we formulate the problem as a reinforcement learning (RL) task or Markov Decision Model (MDP). In this type of model, the parameters include a series of states, actions taken at each state, transition probabilities to the next state based on the current state and action, rewards, and the policy that maps actions to states.

In the case of dynamic pricing, the states represent the series of prices across the given timescale for a specified product. The action space corresponds to the price chosen for the product at the next time step. Because price is a continuous variable in principle, we discretize the price range into a feasible set of pricing actions based on historical price distributions and reasonable retail margins. The reward function is defined as the revenue generated from the chosen price and the resulting demand, using the standard formulation: Reward = Price × Quantity Sold.

The transition model, $P(s'|s, a)$, captures how the environment evolves after a pricing action is taken. In an MDP, we assume the Markov property, meaning that the next state depends only on the current state and the chosen action.

For dynamic pricing, this assumption implies that the effects of all past pricing decisions are sufficiently summarized by the current state variables (such as current observed demand level and price trajectory). Formally, we assume: $P(S_t|S_{t-1}, S_{t-2}, \cdots) = P(S_t|S_{t-1})$. Additionally, we adopt a stationary assumption that the time does not affect the policy decision that maps actions to states.

Before we implement the RL model, we need to create an adequate environment that simulates consumer behavior. Testing the model in real-time would be risky as a vendor could experience profit losses from either under or over-pricing. Therefore, we are in the process of creating a demand model to predict how customers react to a given price given probability of purchase, expected quantity sold, expected revenue, and effect of seasonality, time of day, inventory, and competition. In effect, the demand model acts as a digital model of a real work consumer environment. We cannot use historical pricing alone to feed into the learning process as as there are other factors in play as previously mentioned.

Designing the RL environment for dynamic pricing requires selecting state variables that sufficiently summarize all information that affects future demand. Typical state components include the current price and recent price history (capturing reference-price or fairness effects), observed demand signals, inventory levels, temporal factors such as seasonality or day-of-week, and competitive or contextual variables. We therefore define each state s_t as a vector that captures relevant demand and supply parameters, specifically current price p_t, demand from previous period q_{t-1}, remaining inventory I_t, and what the current seeling period is (i.e. what day and hour). This is expressed as follows:

$$s_t = \{p_t, q_{t-1}, I_t, q_t\}$$

Theoretically, price can be selected by a specified vendor across a continuous price interval: $[p_{min}, p_{max}]$. However, since the policy our RL pricing model involves updating the price, a continuous price interval would result in a continuous action space. In order to simplify the implementation, we decided to discretize the action space into a finite set of feasible price points we discretize the feasible price range into a grid of K price points, where K= 40 [**Bitran**]. Thus, the action set for any given product is defined as follows:

$$A = \{p^{(1)}, p^{(2)}, \cdots, p^{(40)}\}$$

We also have the Bellman value function, which is defined as follows:

$$V_{k+1}(s) = \max_{p \in p_1 \cdots p_K} [R(s,p) + \gamma \cdot \sum_{s'} P(s'|s,p) \cdot V_k(s')]$$

The reward at time t is defined as realized revenue:

$$R_t = p_t \cdot q_t$$

Furthermore, we simplify our representation of inventory by modeling it infinitely available to meet demand. We also model state transitions by implementing a parametric demand function based on historical data. Demand is modeled using a constant-elasticity specification as follows:

$$q_t = A_t \cdot p_t^{-\epsilon} + \epsilon_t$$

In this function, ϵ_t captures random fluctuations in consumer demand such as day-to-day variations in consumer behavior. On the other hand, ϵ represents price elasticity, which is predetermined to a specified value and determines how sensitive demand is to price changes.We use $\epsilon = 1.3$ in line with standard retail practice.

Given this structured environment, we apply model-based RL using value iteration, through which transition probabilities $P(s'|s,a)$ are estimated empirically by repeatedly simulating demand realizations under the calibrated demand model, enabling the value iteration algorithm to converge to a fixed point that approximates the optimal pricing policy.

Results and Discussion

To evaluate the performance of the reinforcement learning pricing model, we simulated revenue over a horizon matching each SKU's historical sales window. Table 1 presents the historical revenue, RL-simulated revenue, and percentage lift for ten representative SKUs that span a range of price variability and sales volume.

Across these products, the RL model generally achieves higher simulated revenue than the historical pricing strategy, often substantially so. For example, SKUs 21212, 84077, 84991, 84270, 21977, and 21213 all exhibit revenue lifts exceeding 100%. These improvements arise because the RL policy explicitly optimizes over the demand model, adjusting prices dynamically in response to simulated demand realizations rather than relying on static or heuristically chosen prices.

However, not all SKUs benefit equally. Products 84879 and 21232 exhibit negative lift, where RL-based pricing yields lower revenue than historical. These cases generally correspond to SKUs with highly volatile demand, limited historical price variability, or data conditions under which the constant-elasticity model may be misspecified. Such discrepancies highlight the sensitivity of value iteration to the accuracy of the underlying demand model and reinforce the importance of robust elasticity estimation.

Table 1: Comparison of Historical Revenue vs. RL-Simulated Revenue for Selected SKUs

StockCode	**Historical Revenue**	**RL Simulated Revenue**	**Lift %**
21212	28,815.61	58,878.93	104.33
84077	6,983.92	51,486.49	637.21
84991	16,360.01	34,566.86	111.29
84270	4,237.31	32,695.87	671.62
84879	57,857.96	31,710.88	-45.19
21977	14,894.89	30,519.25	104.90
21232	36,201.11	25,564.23	-29.38
21213	12,280.98	25,494.84	107.60
84755	14,837.99	22,322.18	50.44
22197	19,798.21	21,485.35	8.52

Visual Comparison of Historical vs. RL Pricing Performance

To better illustrate these results, Figure 1 plots historical and RL-simulated revenue side-by-side for all ten SKUs. The RL agent outperforms historical pricing in most cases, though certain SKUs reveal the limitations of the constant-elasticity model.

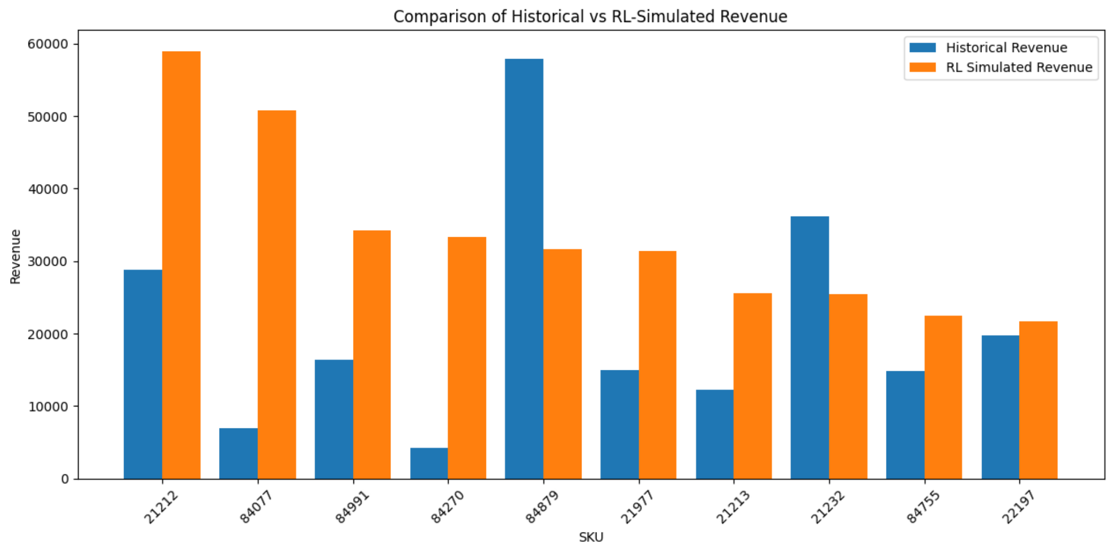

Figure 1: Comparison of Historical vs. RL-Simulated Revenue Across SKUs. RL pricing generally increases revenue, though performance varies across products.

Training Dynamics Across SKUs

To assess the learning behavior of the value iteration procedure, we compute the average value function across all candidate SKUs at each iteration. Figure 2 demonstrates rapid convergence within the first 50–75 iterations, with values stabilizing thereafter. This confirms that our discretized pricing space and assumed demand structure allow the Bellman operator to converge efficiently.

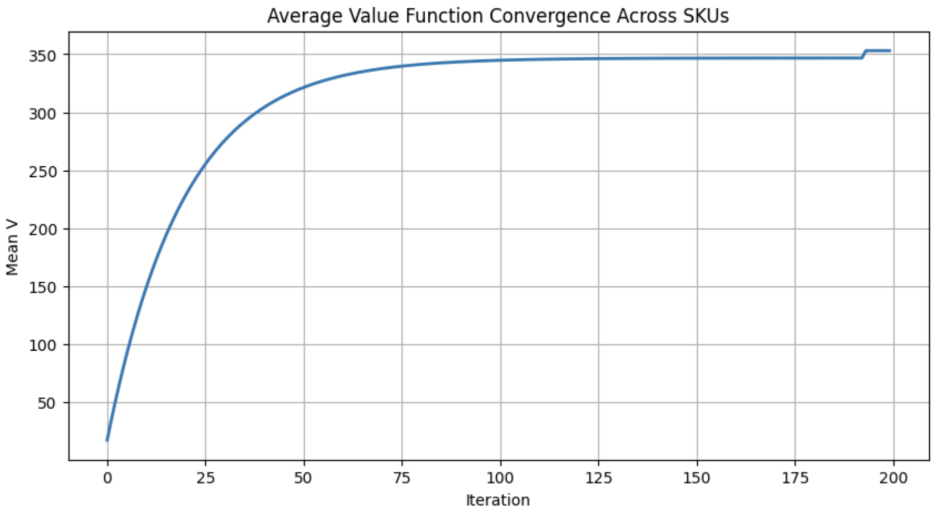

Figure 2: Average Value Function Convergence Across SKUs. The RL value function converges smoothly, indicating stable learning behavior.

Baseline Comparisons

To contextualize RL performance, we compare it against three benchmark pricing strategies:

- **Fixed Price Baseline**: Always price at the SKU's historical median.
- **Elasticity Baseline**: Uses analytically optimal pricing under the assumed demand model.
- **Random Baseline**: Draws prices uniformly at random from the action grid.

Table 2: Baseline Comparison Across SKUs with RL Gain Relative to Best Baseline

SKU	**Historical**	**RL**	**Fixed Price**	**Elasticity**	**Random**	**Best Baseline**	**RL Gain (%**
21212	28,815.61	58,878.93	50,949.19	58,499.87	46,314.01	58,499.87	0.65
84077	6,983.92	51,486.49	48,874.45	52,267.82	43,813.40	52,267.82	-1.49
84991	16,360.01	34,566.86	32,196.37	34,116.05	28,388.34	34,116.05	1.32
84270	4,237.31	32,695.87	29,942.69	32,508.06	27,814.89	32,508.06	0.58
84879	57,857.96	31,710.88	30,412.77	31,816.38	29,017.48	31,816.38	-0.33
21977	14,894.89	30,519.25	28,359.99	30,292.76	23,980.58	30,292.76	0.75
21232	36,201.11	25,564.23	23,070.78	25,528.92	20,955.56	25,528.92	0.14
21213	12,280.98	25,494.84	23,113.99	25,710.05	18,806.89	25,710.05	-0.84
84755	14,837.99	22,322.18	21,485.37	22,491.66	14,503.10	22,491.66	-0.75
22197	19,798.21	21,485.35	20,957.77	21,826.80	18,209.55	21,826.80	-1.56

Across the three baseline methods, the elasticity-based strategy consistently produced the highest revenues, reflecting its alignment with the same constant-elasticity demand structure used in the RL environment. When comparing the RL policy directly against the best-performing baseline for each SKU, we find that the RL strategy achieves higher revenue for 5 out of the 10 SKUs evaluated. These gains, though sometimes modest, highlight the value of dynamic price adaptation in settings where demand evolves over time rather than remaining stationary at a single analytically optimal point.

To visualize revenue distribution across SKUs, Figure 3 presents boxplots for RL and baseline methods. RL achieves the highest median revenue, while the elasticity baseline performs competitively, expected since both rely on the same structural demand model. The fixed-price and random strategies substantially underperform, reinforcing the benefits of dynamic price adaptation.

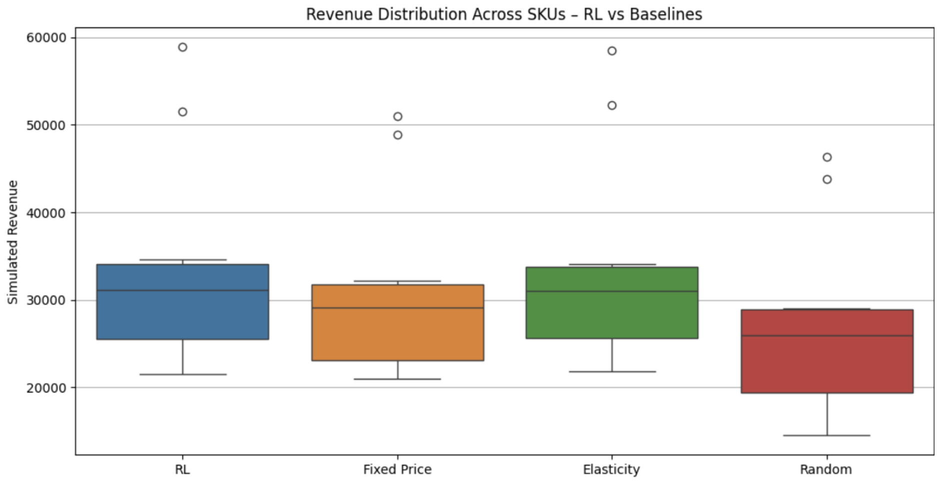

Figure 3: Revenue Distribution Across SKUs for RL and Baseline Pricing Strategies

Overall, these results demonstrate that the reinforcement learning–based pricing strategy delivers strong performance relative to historical pricing and simple baselines, particularly for SKUs exhibiting stable elasticity patterns and sufficient price variation. Cases of under-performance reveal opportunities for future improvement, including richer demand models, hierarchical elasticity estimation, and more expressive belief-updating mechanisms.

Conclusion and Reflection

This project provided a great opportunity for us to integrate probabilistic reasoning, reinforcement learning, and real-world retail data to address a critical problem of dynamic pricing. One key insight we realized is that the quality of the final model is fundamentally constrained by the quality of the underlying data pipeline. Pre-processing and ensuring quality data is essential for ensuring reliable results later on during the modeling phase. For future students pursuing similar work, they should understand that dynamic pricing

References

[SS73] Richard D. Smallwood and Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. INFORMS, 1973.

[BC03] Gabriel Bitran and Rene Caldentey. An Overview of Pricing Models for Revenue Management. INFORMS, 2003.

[DA25] Bennet Dan and Kolade Ajeigbe. Dynamic Pricing Strategies Using Deep Reinforcement Learning. N/A, 2025.