Dynamic Pricing in a Competitive Market with Shared Demand Influenced by Social Media Trends: The Strategic Role of Markdowns vs. Temporary Promotions

by

Rishika Agarwal

A dissertation submitted in partial fulfilment for the degree of MSc in Business Analytics at Warwick Business School — University of Warwick

Acknowledgements

I am extremely thankful to my supervisor, Professor Bo Chen, for his invaluable guidance, constructive feedback, and continuous support throughout this research. His expertise and encouragement have been instrumental in shaping this dissertation and ensuring its successful completion.

I would also like to thank the faculty and staff at Warwick Business School for their insightful feedback and for providing an excellent academic environment that greatly supported my learning journey.

My heartfelt thanks go to my classmates and peers in the MSc Business Analytics programme for their support, collaboration, and for making my time at Warwick both productive and memorable.

Finally, I am deeply thankful to my family for their unconditional love, patience, and unwavering support, without which this achievement would not have been possible.

Contents

List of Tables

List of Figures

Abstract

This dissertation investigates dynamic pricing strategies in competitive markets shaped by volatile, socially amplified demand. Existing research often neglects behavioural responses, algorithmic adaptivity, or robustness to social volatility, leading to suboptimal outcomes. To address these gaps, this study integrates behavioural economics (reference price effects), three distinct pricing strategies—basic (static), markdown (transparent value), and promotion (temporary discounts) with reinforcement learning (RL) and Bayesian smoothing in a multi-agent simulation.

Results show that basic static pricing provides a useful benchmark but underperforms in dynamic conditions. Markdown strategies consistently outperform promotions in volatile settings by stabilising reference prices, signalling fairness, and sustaining market share (supporting H1). Promotions deliver short-term gains but create volatile expectations that erode long-term value, consistent with prior behavioural findings. RL-based pricing achieves superior long-run profitability (supporting H2), though early-stage instability is exacerbated by socially driven demand amplification. Bayesian smoothing mitigates this risk, enhancing both profitability and consumer trust (H3, H4).

The findings advance pricing theory by linking behavioural insights with adaptive, robust algorithmic approaches. Practically, they recommend a sequenced strategy: employ markdowns to stabilise early demand, then transition to RL optimised by Bayesian filtering. Ethically, the study underscores the importance of fairness-aware algorithmic design in socially sensitive markets. This research contributes to pricing, digital strategy, and policy by delivering a novel, empirically grounded framework for dynamic pricing in volatile digital economies.

Chapter 1: Introduction

The rise of social media platforms such as TikTok has transformed the dynamics of consumer demand. Viral trends can generate sudden surges in interest for specific products, followed by sharp declines as attention shifts elsewhere. These swings create volatility and uncertainty, making it difficult for firms to sustain stable revenues and profits. In such environments, pricing becomes a primary lever of competitiveness. Managers typically rely on two families of interventions: markdowns—permanent list-price reductions that signal enduring value and temporary promotions, short-term discounts timed to coincide with demand peaks. At the same time, reference-price effects mean consumers compare current prices with those they have previously observed, so perceptions of value and fairness depend as much on price history as on current levels.

Despite extensive work on dynamic pricing, important gaps remain for trend-driven markets. First, most models are single-firm or treat competition only superficially, even though social media synchronises demand across rivals. Second, common approaches react directly to raw trend signals, risking over- or under-shooting in the presence of noise. Third, while reinforcement learning (RL) offers adaptivity, it faces practical challenges such as cold starts, exploration–exploitation trade-offs, and instability during regime shifts. Together, these gaps obscure when firms should favour markdowns, promotions, or adaptive RL in markets shaped by volatile, shared attention.

This dissertation addresses these issues by comparing markdowns, temporary promotions, and adaptive RL policies in a competitive market where demand is co-created by social media. The study develops an agent-based simulation grounded in an econometric demand model estimated from retail transactions and augmented with a TrendScore that captures viral spikes and noisy decay. Bayesian belief updating filters the trend signal into posterior estimates, while multi-agent RL learns pricing strategies under competitive feedback. Outcomes are evaluated across financial (revenue, profit, ROI), competitive (market share), and stability-related (price and demand volatility as proxies for fairness and trust) dimensions.

To sharpen the research focus, this study explicitly investigates the following research questions:

- RQ1: When do markdowns versus temporary promotions generate superior revenue, profit, ROI, and market share in trend-sensitive categories?
- RQ2: How does the volatility and noise of social-media trends change the relative performance of markdowns and promotions?
- RQ3: Do reinforcement-learning policies outperform rule-based markdown/promotion strategies once trained, and under what conditions?
- RQ4: Does Bayesian smoothing of noisy trend signals improve decision stability and performance compared with acting on raw signals?

Following these research questions, the dissertation proceeds by first reviewing the relevant literature on behavioural pricing, reinforcement learning, social contagion, and ethical considerations, which culminates in the formal hypotheses. The methodology chapter then introduces the simulation framework, including the econometric demand estimation, Bayesian updating, and reinforcement learning design. Results are presented and analysed with respect to both financial and ethical outcomes, before the discussion and conclusion highlight theoretical contributions, managerial implications, and directions for future research.

Chapter 2: Literature Review

2.1 Evolution of Dynamic Pricing

Dynamic pricing has evolved from rudimentary, static discounting systems to adaptive, data-driven strategies capable of adjusting prices in real time as firms learn from demand, competition, and uncertainty. Early work in revenue management, particularly in industries like airlines and hotels, focused on time-based pricing and inventory control to maximise revenues under relatively predictable demand (Talluri and Van Ryzin, 2004).

Behavioural economics introduced a crucial shift in understanding pricing dynamics. Seminal studies demonstrated that consumers develop reference prices, internal benchmarks shaped by their previous experiences and competitive offerings. Kalyanaram & Winer (1995) provided empirical generalisations that consumers do not evaluate prices in isolation but relative to these benchmarks. Kalyanaram and Winer, (2022) confirmed that these behavioural insights have fundamentally reshaped pricing strategies, with firms increasingly incorporating psychological factors into algorithmic models. Srinivasan, Popkowski Leszczyc & Bass (2000) further illustrated that current prices influence not just immediate demand but also future expectations, as consumers perceive gains when prices fall below their reference level and losses when they exceed it. This behavioural asymmetry, aligned with prospect theory (Kahneman & Tversky, 1979), implies that price increases can trigger disproportionately negative reactions compared to equivalent reductions. Mazumdar, P and Sinha, (2005) reinforced this by showing that while promotions may provide short-term boosts, they often erode long-term perceived value in competitive markets, risking a downward spiral of expectations.

The digitalisation of commerce fundamentally altered pricing practices. Elmaghraby & Keskinocak (2003) argued that digital technologies enabled real-time optimisation by leveraging large-scale data, while Alabi (2024) demonstrated how machine learning could adapt prices dynamically to volatile market conditions. Bajari et al. (2015) further contributed by outlining how Bayesian approaches can stabilise adaptive pricing under uncertainty, mitigating the risk of overreacting to short-term fluctuations. Similarly, Huang, Luo & Xia (2019) modelled dynamic pricing as a seller-learning problem, highlighting the trade-off between investing in information and adopting adaptive strategies, thereby extending the role of learning in uncertain markets. Yet, as Agrawal & Tang (2024) contend, many algorithmic approaches optimise for short-term returns, neglecting intertemporal trade-offs such as reference price effects and long-run consumer trust, often leading to underperformance over extended horizons.

Taken together, this evolution highlights critical gaps. Traditional revenue management models are robust in stable environments but behaviourally naive. Behavioural models capture consumer psychology but lack adaptivity. RL-based methods are adaptive but often blind to social amplification and ethical considerations. This study addresses these limitations by integrating behavioural realism (reference effects), adaptive optimisation (RL), and robustness to socially driven demand volatility via Bayesian smoothing, bridging theoretical advances with the realities of modern, interconnected markets.

2.2 Consumer Behaviour and Reference Prices

Consumer behaviour research shows pricing decisions are inseparable from psychological processes. Consumers form internal reference prices, benchmarks shaped by past prices and competition (Kalyanaram & Winer, 1995). Prospect theory explains why deviations create asymmetry: prices above the reference point feel like losses, suppressing demand (Kahneman & Tversky, 1979).

Past pricing creates path dependence. Aggressive pricing can entrench competitive positions but risks “promotion addiction,” where frequent discounts lower willingness to pay and erode profitability (Mazumdar et al., 2005).

Promotional intensity also increases price sensitivity (Hardesty & Bearden, 2003) and damages brand equity (DelVecchio et al., 2006).

Modern models integrate these insights. Incorporating reference effects improves long-run profitability (Agrawal & Tang, 2024). RL with reference effects performs better in stable markets (Zhou et al., 2022), while deep RL alone risks myopic discounting under volatility (Yin & Han, 2021). Bayesian updating mitigates this by stabilising responses to noisy trend signals.

2.3 AI and Reinforcement Learning Approaches

The last decade has seen artificial intelligence (AI), particularly reinforcement learning (RL), transform dynamic pricing by reframing it as a sequential decision-making problem. Traditional supervised learning enhances short-term demand forecasts but is inherently reactive, relying on historical patterns and struggling with novel market shocks (Bertsimas and Kallus, 2018). RL, in contrast, enables pricing agents to experiment, observe market feedback, and optimise long-run performance through exploration–exploitation trade-offs (Internet Archive, 2018).

Empirical implementations show promise. Yin & Han (2021) apply deep RL to e-commerce platforms, demonstrating significant revenue gains over heuristic strategies under moderate volatility. Zhou, Yang & Fu (2022) extend RL to include reference price effects in joint pricing-inventory decisions, improving consumer retention and profitability in stable environments. However, both approaches suffer under high uncertainty: Yin & Han's (2021) model ignores behavioural dynamics such as reference prices, leading to myopic discounting, while Zhou et al. (2022) treat demand shocks as independent, neglecting the systemic volatility introduced by social contagion.

Moreover, RL introduces ethical challenges. Unregulated algorithms may learn to exploit consumer biases or engage in tacit collusion (Calvano et al., 2020), undermining trust and inviting regulatory scrutiny. Existing work rarely embeds fairness or transparency constraints into optimisation objectives (Chen et al., 2016), leaving a critical gap between algorithmic efficiency and societal acceptability.

This dissertation addresses these shortcomings by integrating RL with behavioural pricing (reference effects), multi-agent competition, and Bayesian smoothing to handle noisy, socially amplified demand signals. By doing so, it evaluates not only profitability but also ethical implications—advancing the field beyond narrow revenue optimisation.

2.4 Shared Demand and Social Media Influence

Consumer demand is increasingly shaped by social contagion, where preferences and purchase timing propagate through networks. Classical models treat demand shocks as idiosyncratic, but evidence shows strong correlation across consumers and firms due to imitation, information cascades, and viral content (Dholakia & Talukdar, 2004). On platforms like TikTok, sudden bursts of attention elevate entire product categories before collapsing, creating synchronised booms and busts (Singh and Singh, 2021).

These dynamics complicate pricing strategies. Viral trends temporarily raise consumers' willingness-to-pay, making premium pricing feasible, but they also reset post-trend reference prices downward as consumers anchor on discounted promotional periods (Godes et al., 2009). Firms that fail to adapt risk overproducing during peaks or missing demand windows altogether.

While some studies explore network effects in adoption (Katona, Zubcsek & Sarvary, 2011) and social media's role in demand forecasting (LUO and ZHANG, 2013), few integrate these insights into dynamic pricing models. Recent attempts to incorporate social signals, such as sentiment-driven demand forecasts (Xie & Mao, 2017), remain firm-centric and fail to capture multi-agent competition for transient attention.

Critically, social contagion reshapes not just demand magnitude but also fairness perceptions. Opportunistic price hikes during viral spikes can trigger consumer backlash and negative virality (Campbell et al., 2020), eroding long-term trust. Current algorithmic pricing research largely ignores this reputational risk.

This dissertation advances the literature by modelling shared demand as a noisy, observable trend signal influencing all competitors simultaneously. By incorporating Bayesian filtering and ethical metrics into simulations, it offers a more realistic and responsible framework for competing in socially volatile markets.

2.5 Ethical Frameworks in Algorithmic Pricing

The deployment of algorithmic pricing raises important ethical and regulatory considerations. Three themes are particularly salient. First, fairness and discrimination are central concerns. Personalised pricing, while profitable, risks unfair treatment if sensitive attributes or their proxies influence outcomes. Even when legally permissible, opaque segmentation can erode consumer trust. Prior work on algorithmic fairness recommends limiting excessive price dispersion and ensuring parity of treatment across groups (Chen et al., 2016), yet such principles are rarely integrated into dynamic pricing design.

Second, transparency and consumer trust are crucial. Persistent and inexplicable price fluctuations can appear manipulative, particularly in categories with frequent repeat purchases. Firms that communicate stable value propositions, for example, through sustained markdowns may cultivate greater trust than those that rely on frequent, opaque promotions. This echoes behavioural research on fairness, which finds that perceived opportunism can trigger consumer backlash even when prices are economically justified.

Third, accountability under uncertainty matters. RL systems optimise for profit but are not inherently aligned with broader societal constraints. When trend signals are noisy, unregularised policies may overshoot, producing perceived gouging during peaks or predatory undercutting during troughs. These risks connect the technical literature on volatility with the consumer psychology literature on fairness, suggesting that algorithmic missteps can damage both reputation and long-term competitiveness.

In summary, ethical considerations underscore that pricing strategies must be evaluated not only on financial outcomes but also on their implications for fairness, stability, and consumer trust. This motivates incorporating Bayesian smoothing into the simulation framework (see Section 3.6), providing a principled mechanism to temper overreactions to noisy signals and to evaluate how markdowns, promotions, and RL compare on both profitability and stability.

2.6 Methodological Foundations

Dynamic pricing research draws on three methodological pillars: econometric demand estimation, Bayesian updating, and reinforcement learning (RL). Econometric models provide interpretable estimates of consumer price sensitivity, trend responsiveness, and competitive interactions, forming the foundation of empirical pricing studies (Nerlove & Arrow, 1962; Miguel and Winer, 1999). These models remain essential for identifying causal relationships and informing simulation parameters. However, they lack adaptivity in volatile markets where demand evolves rapidly.

Bayesian methods address this by allowing prior beliefs to be updated as new data arrives, offering robustness against uncertainty and noisy environments (Bajari et al., 2015; Rossi, Allenby and McCulloch, 2005). They have been successfully applied in revenue management to filter demand shocks and guide decision-making under uncertainty, but they are often disconnected from real-time strategic interaction.

RL introduces adaptive optimisation in sequential, competitive contexts. By learning optimal pricing policies through trial-and-error interactions with the market, RL has shown superior performance in dynamic e-commerce settings (Liu et al., 2021; Yin & Han, 2021). Recent advances integrate RL with Bayesian methods to mitigate volatility, demonstrating that combining adaptivity with statistical robustness yields better long-term outcomes (Yang et al., 2023; Deng, Schiffer and Bichler, 2024).

Despite these advances, most prior work uses these methods in isolation, limiting their effectiveness. This dissertation integrates econometric estimation (for structural grounding), Bayesian inference (for stability), and RL (for adaptivity) into a unified agent-based simulation, addressing the need for behaviourally grounded, robust, and adaptive pricing frameworks in socially influenced markets.

2.7 Where Prior Work Falls Short and How This Study Advances It

Prior studies have advanced three important fronts. Behavioural pricing with reference effects explains path dependence and promotion asymmetries. RL shows how data-driven policies can outperform heuristics when the environment is learnable. Forecasting and meta-learning expand transferability across products. Yet these threads are mostly developed in isolation and commonly within simplified competitive settings.

Yin and Han (2021) are strong on deep RL engineering and empirical gains but weaker on strategic competition and explicit social trend modelling. Their results imply RL's promise once trained but leave open how it fares during rapid regime shifts. Zhou, Yang and Fu (2022) are strong on uncertainty handling and consumer heterogeneity, sometimes with reference price effects, but weaker on multi-agent coupling and trend-synchronised shocks. Their exploration–exploitation insights motivate our evaluation of profit efficiency and ROI rather than revenue alone. Agrawal and Tang (2024) are strong on learning with reference effects and long-run revenue optimisation, but their scope is largely single-firm or stylised competition, insufficient for shared demand with social contagion. Liu et al. (2021) are strong on transfer and meta-learning, but their vulnerability to regime shifts suggests complementing transfer with Bayesian guards and volatility-aware adaptation.

A second limitation is ethical: much of the technical literature optimises economic outcomes without considering trust, fairness, or transparency. In volatile markets shaped by social scrutiny, however, perceived opportunism can trigger negative virality or boycotts, reducing long-term competitiveness.

In short, there is little work that:

- jointly models social-media-driven shared demand with noisy trend observation,
- compares markdowns and temporary promotions against multi-agent RL in the same competitive setting,
- embeds Bayesian smoothing to stabilise learning, and
- evaluates outcomes not only on revenue and profit but also on ROI, market share, and trust-related stability.

This dissertation is designed to fill that gap by providing an integrated framework that unites behavioural pricing, Bayesian inference, and reinforcement learning in a socially influenced, multi-agent competitive environment.

2.8 Implications for the Present Study

The review yields testable expectations aligned with the dissertation's research questions. In high-volatility regimes, markdowns as persistent value signals should capture share by simplifying the choice for trend-excited but uncertain consumers. Given time to learn, RL should achieve superior ROI and profit efficiency, especially when trend states are filtered through Bayesian updating that reduces overreaction to noise.

The relative ranking of strategies is context-dependent: consumer trend sensitivity, competition intensity, and signal noise all moderate performance. For example, markdowns may dominate when volatility is high and RL has not converged, but RL may pull ahead once sufficient data is available to stabilise learning. Promotions, meanwhile, may capture peaks but risk eroding reference prices if overused.

Finally, ethical and trust considerations are first-order. Stable value communication through markdowns may dominate short-term gains from opaque promotions or over-fit RL policies in markets where social scrutiny is high. These insights justify the simulation design and the inclusion of fairness-related outcome metrics, ensuring that methodological choices remain grounded not only in economic theory but also in consumer psychology and managerial realism.

2.9 Ethical and Strategic Consideration

Algorithmic pricing raises ethical and strategic challenges. Personalised and dynamic pricing can increase efficiency but risk discrimination, opacity, and erosion of consumer trust (Chen et al., 2016; Kleinberg et al., 2019). Opaque segmentation, particularly when correlated with sensitive attributes, can trigger perceptions of unfairness, leading to reputational damage and regulatory scrutiny (Brand, 2020).

Transparency and stability are crucial in socially sensitive markets. Behavioural research shows that consumers value predictability and fairness (Grewal et al., 1998; Hardesty & Bearden, 2003). Algorithmic systems that react excessively to transient social-media-driven fluctuations risk appearing exploitative, especially during surges in demand for essential or culturally significant products.

Strategically, firms must balance short-term gains from opportunistic pricing with long-term sustainability. Competitive dynamics further complicate this: uncoordinated adoption of aggressive RL strategies can spark destructive price wars, reducing collective profitability (Deng, Schiffer and Bichler, 2024; Yang et al., 2023). This study explicitly evaluates pricing strategies on profitability, market share, and stability, aligning technical performance with ethical imperatives.

2.10 Hypotheses Development

Building on the reviewed literature, four hypotheses are proposed to guide the empirical analysis:

- **H1 (Markdowns under volatility):** In high-volatility environments, markdowns outperform promotions on revenue and market share by providing a stable value signal.
- **H2 (RL efficiency after learning):** Once trained, RL policies achieve higher ROI and profit efficiency than static or rule-based strategies.
- **H3 (Bayesian stability):** Bayesian smoothing reduces overreaction to noise, yielding greater stability without sacrificing profitability.

- **H4 (Trend sensitivity):** As consumer sensitivity to trends declines, markdowns retain or expand their advantage over promotions.

These hypotheses link directly to the research questions outlined in the introduction and provide a structured basis for the methodology and simulation design that follow.

Chapter 3: Methodology

3.1 Research Design

The research employs a simulation-based methodology that integrates agent-based modelling, econometric demand estimation, Bayesian belief updating, and reinforcement learning (RL). The overarching aim is to investigate how firms competing in a shared market adapt their pricing strategies in environments shaped by volatile, trend-driven demand. This design is appropriate because it allows experimentation with strategic interactions under controlled but realistic conditions, enabling clear comparisons between alternative strategies.

The methodology unfolds sequentially across four stages. First, real-world retail data are combined with synthetic trend signals to construct the demand environment. Second, a demand estimation model is specified to parameterise consumer responsiveness to both relative price changes and trend intensity. Third, an agent-based simulation is implemented in which firms compete using alternative pricing strategies ranging from static rules to adaptive learning policies. Finally, outcomes are evaluated using financial, competitive, and stability-related metrics. This sequential approach ensures that the analysis remains both empirically grounded and methodologically coherent, while also linking each stage to the research questions posed in the introduction.

All stages of the methodology were implemented in Python. Data cleaning and feature engineering were conducted using the pandas library, while econometric estimation of the demand model was performed with statsmodels. Bayesian updating was coded directly, with posterior means and variances computed in each iteration of the simulation. Agent-based simulation loops were written procedurally in Python, with demand allocation handled through a softmax choice function. Reinforcement learning was implemented using a custom Q-learning algorithm with discretised states and actions, coded from first principles rather than relying on external RL libraries. Results were stored and analysed with numpy and visualised using matplotlib. Including this coding environment description clarifies how methodological components were translated into executable models.

Figure 1 below illustrates the full simulation pipeline, showing how data preparation, demand estimation, Bayesian updating, and reinforcement learning modules are integrated into a coherent framework.

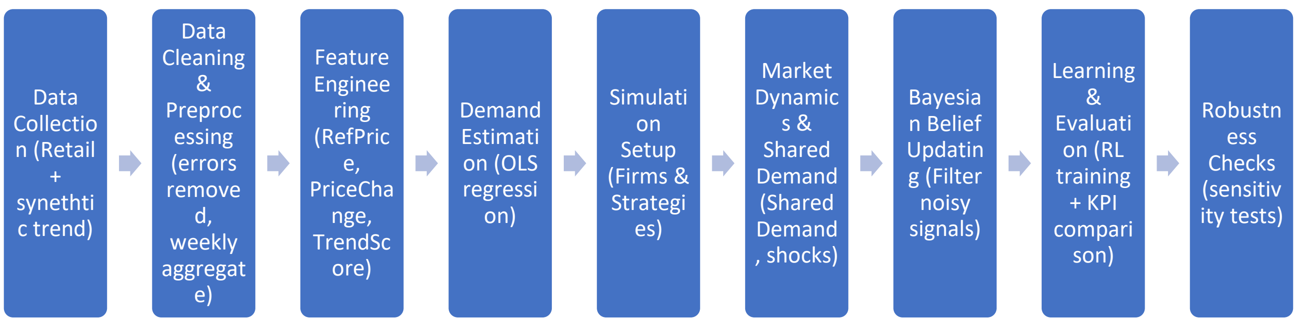


Figure 1: Simulation Pipeline for Dynamic Pricing in Trend-Driven Markets

3.2 Data Sources and Features

The empirical foundation of the simulation rests on two types of input: transactional retail data and synthetic trend signals.

The first dataset is the UCI Online Retail dataset, which provides invoice-level records including unit prices, quantities, transaction dates, and customer identifiers. This dataset has been widely employed in retail analytics because it reflects real consumer behaviour such as repeat purchasing, cross-category variability, and seasonal demand fluctuations. To make the dataset suitable for simulation, extensive preprocessing was carried out. Transactions with missing values, negative quantities, or zero prices were excluded to eliminate returns and errors. Data were then aggregated to the weekly level, with sales volume and average unit price computed for each product-week combination. Weekly aggregation was chosen to balance granularity with tractability, allowing the model to capture medium-term pricing responses without being swamped by day-to-day noise.

From this processed dataset, the analysis focused on the five top-selling products by revenue. This restriction reduces dimensionality while maintaining economic relevance, as firms often apply advanced pricing tools to their most profitable lines. For each product-week, derived features included Quantity (total units sold), Sales (revenue), and AvgPrice (average realised unit price).

The second input is synthetic trend data. Since direct integration with APIs such as Twitter or Google Trends was not feasible, a stylised representation of social media attention was constructed. The synthetic TrendScore variable captures the volatility of consumer interest by combining baseline growth, viral spikes, exponential decay, and Gaussian noise. For example, in Week 4 a sharp spike may represent a viral TikTok challenge, followed by rapid decay over subsequent weeks, while random noise accounts for unpredictable fluctuations. The series was normalised to lie between 0 and 1 for interpretability.

Where possible, historical Google Trends data were consulted to calibrate the general shapes of viral attention curves, particularly their rapid rise and asymmetric decay.

Nevertheless, synthetic signals have inherent limitations. While they approximate volatility and unpredictability, they cannot capture the full richness of social contagion processes. Cross-platform amplification effects and consumer-driven feedback loops are excluded. For this reason, the simulated TrendScore should be interpreted as a stylised stress test rather than a literal forecast of real consumer attention. These limitations frame the results as explorations of strategic robustness, rather than precise predictions of market outcomes.

Feature engineering introduced two additional behavioural variables. The first, RefPrice, captures consumer memory by recording the average price of the previous week, thereby enabling modelling of reference effects. The second, PriceChange, measures the deviation of the current price from the reference price, capturing relative movements rather than absolute levels. These variables, combined with TrendScore, provide the explanatory foundation for the demand estimation stage.

3.3 Consumer Behaviour and Assumptions

Consumers in the simulation are represented as agents who allocate their demand probabilistically across competing firms. The allocation mechanism is a softmax choice function, in which each firm’s utility depends negatively on its price and positively on the prevailing trend intensity. The softmax formulation ensures that all firms retain some share of demand while favouring those with higher relative utility, reflecting realistic competition where lower prices and stronger trend alignment increase but do not guarantee purchase likelihood.

The following assumptions are made explicit:

- Consumers observe all firm prices and trend signals without friction.
- Sensitivity to trends is homogeneous within specified subgroups (e.g., trend-sensitive vs. trend-insensitive

consumers).

- Inventory constraints, stockouts, and supply bottlenecks are not included.
- Consumers are rational with respect to price and trend salience. Brand loyalty, social identity, and fairness perceptions are excluded to maintain tractability.

These simplifications limit external validity but allow the simulation to isolate the effects of pricing and social trends on demand allocation.

3.4 Demand Estimation Model

Before simulating competition, an empirical demand function is estimated to capture how consumers respond to price changes and trend scores. An **Ordinary Least Squares (OLS) regression** is specified as:

$$Q_t = \beta_0 + \beta_1 * PriceChange_t + \beta_2 * TrendScore_t + £_t$$

where Q_t is demand at time t, $PriceChange_t$ is the deviation from the reference price, $TrendScore_t$ represents the intensity of the trend signal and $£_t$ is the error term.

This specification is grounded in behavioural economics and marketing theory. Consumers are not only sensitive to absolute price levels but also to deviations from a reference price, reflecting loss aversion and fairness heuristics. At the same time, empirical research shows that online trends act as powerful demand shocks, amplifying attention and purchase intent. By combining these two explanatory factors, the model links economic rationality with socially mediated demand surges.

Diagnostic checks were conducted to ensure robustness. Variance Inflation Factor (VIF) analysis revealed that including both AvgPrice and RefPrice produced severe multicollinearity, inflating standard errors and reducing interpretability. To address this, the model employed the consolidated PriceChange variable, which reduced VIF values to approximately 1, indicating negligible collinearity. Residual diagnostics further confirmed stability, with no systematic patterns suggesting model mis-specification.

Although the regression explained only a modest share of variation R^2 (~0.11), this is consistent with behavioural demand modelling, where many unobserved factors—such as brand loyalty, cross-channel marketing, or broader economic conditions—lie outside the model. More importantly, the estimated coefficients were stable and theoretically consistent: higher relative prices reduced demand, while stronger trend scores increased demand.

The objective of this regression is not precise forecasting but parameterisation. The estimated coefficients provide interpretable measures of consumer responsiveness to price and social signals. These parameters are then embedded into the agent-based simulation, grounding simulated consumer behaviour in empirically estimated sensitivities and ensuring that subsequent experiments remain theoretically and behaviourally consistent.

3.5 Simulation Framework

The simulation models a competitive market with three firms adopting distinct strategies. Firm A implements static pricing, serving as a control. Firm B follows a markdown strategy, applying permanent reductions once demand falls below a threshold. Firm C employs temporary promotions, discounting during trend peaks to capture surges in attention.

Each simulated week follows a structured sequence:

1. A trend score is generated, combining baseline signal and stochastic noise.
2. Firms update their beliefs about the underlying trend using Bayesian inference, producing posterior means and variances.
3. Consumers allocate demand across firms using the softmax function, with probabilities reflecting both prices and perceived trend salience.
4. Outcomes are recorded for each firm, including revenue, profit, market share, and return on investment.

This cyclical framework captures the interaction of strategies, consumer responsiveness, and trend volatility.

3.6 Reinforcement Learning Configurations

To move beyond rule-based heuristics, the simulation incorporates reinforcement learning (RL), specifically Q-learning, as an adaptive pricing mechanism. In this setup, the state space includes discretised trend scores, past prices, and demand outcomes, while the action space corresponds to either discrete price levels or higher-level strategic choices.

Several configurations are tested:

- **Basic Q-learning:** actions are price changes; states combine trend and past price information.
- **Strategy-level Q-learning:** actions correspond to strategic categories (static, markdown, promotion).
- **Epsilon-decaying RL:** exploration gradually decreases over time, ensuring eventual convergence.
- **Multi-agent RL:** all firms learn simultaneously, generating strategic feedback loops.

These RL variants address whether adaptive agents can outperform static or rule-based strategies in volatile environments, and under what conditions learning converges or destabilises.

Each RL configuration was trained for 1,000 episodes to ensure convergence, with performance tracked using cumulative reward trajectories and Q-value stability. Exploration employed an ε-greedy policy with an initial ε of 0.3, decaying by 0.99 per episode, to balance exploration and exploitation effectively. The learning process was repeated across multiple random seeds, and convergence was confirmed when Q-values stabilised and pricing trajectories became consistent. This procedure ensures that reported RL outcomes are robust and not artefacts of specific runs.

3.7 Robustness Checks

A core concern in simulation-based research is whether observed results hold only under narrow assumptions, or whether they generalise across a broader range of plausible market environments. To address this, the simulation was subjected to an extensive series of robustness checks. These checks probe the sensitivity of outcomes to volatility, noise, consumer heterogeneity, learning dynamics, and random shocks. By stress-testing the model along multiple dimensions, the analysis strengthens the credibility of the findings and delineates the boundaries within which conclusions can be generalised.

3.7.1 Demand-Side Robustness

The first set of robustness checks examines how different consumer environments alter firm performance. Demand shocks were simulated by varying the volatility of the trend signal. In stable markets, TrendScore follows gradual increases and decreases, while in turbulent markets it incorporates sudden spikes and collapses. This allows the analysis to compare how strategies perform when demand is predictable versus highly erratic.

False or noisy signals were also introduced. In these scenarios, TrendScore contains high levels of Gaussian noise, which dilutes the informational content of the signal. This is particularly important for testing the value of Bayesian updating. Firms relying directly on raw signals are expected to misinterpret noise as genuine demand shifts, whereas Bayesian posteriors should provide a smoother and more reliable basis for decision-making. Comparing the two environments highlights the robustness of Bayesian inference under uncertainty.

Consumer sensitivity to price and trend salience was varied parametrically. In one configuration, consumers place disproportionate weight on price, leading to markets where discounts dominate regardless of trend dynamics. In another, consumers are highly trend-sensitive, amplifying the role of viral shocks. By varying sensitivity coefficients, the analysis demonstrates how strategic performance depends on the relative balance between rational price responsiveness and socially mediated demand contagion.

3.7.2 Firm-Side Robustness

A second category of robustness checks investigates how different firm strategies withstand common shocks. Identical demand environments were replayed under static, markdown, promotional, and reinforcement-learning strategies, allowing direct comparison of how each pricing rule adapts or fails to adapt to volatility.

In addition, the effect of Bayesian smoothing was tested systematically. Each firm was simulated twice: once acting on raw trend scores and once using Bayesian-updated posteriors. This check isolates the value of filtering noisy information. In volatile environments, Bayesian smoothing was expected to reduce overreaction and yield more stable pricing trajectories, while in stable environments the difference between raw and smoothed signals was smaller.

Firm-level robustness was also tested by varying the intensity and timing of promotions. For example, promotions triggered by modest demand fluctuations were contrasted with promotions activated only at sharp peaks. This allowed for evaluation of how finely tuned promotional thresholds influence revenue stability.

3.7.3 Reinforcement Learning Robustness

Since reinforcement learning introduces adaptive dynamics, particular attention was paid to its robustness. Several stress tests were conducted.

First, cold-start behaviour was examined. At the beginning of training, RL agents explore the environment with limited knowledge, which can result in erratic pricing and short-term losses. Simulations were extended across long time horizons to observe whether agents converged on stable strategies or remained trapped in suboptimal behaviours.
Second, exploration–exploitation trade-offs were analysed by varying the ε-decay schedule. Faster decay leads to premature exploitation, which risks overfitting to noisy patterns, while slower decay prolongs exploration, delaying convergence. By testing different decay rates, the analysis demonstrates how sensitive RL outcomes are to learning parameters.
Third, learning rates (α) were varied. Higher learning rates enable faster adaptation but increase the risk of instability, while lower rates improve stability but slow responsiveness. Comparing these configurations provides insight into how parameter tuning influences performance and robustness.

Finally, single-agent and multi-agent RL configurations were compared under identical conditions. In single-agent settings, firms adapt to exogenous demand, whereas in multi-agent settings strategic interactions generate endogenous complexity. By contrasting these, the analysis highlights how robustness differs when firms adapt in isolation versus in a competitive learning environment.

3.7.4 Simulation Stability

To ensure that findings were not artefacts of random draws, all experiments were repeated across multiple random seeds. This allowed the distribution of outcomes to be compared, rather than relying on single-run trajectories. Repeated simulations confirmed whether apparent performance advantages were systematic or contingent on favourable noise realisations.

Convergence diagnostics were also employed. In RL experiments, Q-values were tracked across iterations to confirm that they stabilised, rather than oscillating indefinitely. In volatility stress tests, firm revenues and market shares were plotted across multiple runs to check whether patterns repeated consistently. These diagnostics provide assurance that the simulation produces stable results, rather than fragile or path-dependent artefacts.

3.7.5 Summary

By integrating demand-side, firm-side, reinforcement learning, and stability checks, the robustness analysis provides a comprehensive assessment of methodological reliability. Strategies are evaluated not only under average conditions but also under extremes of volatility, noise, and parameter variation. This approach ensures that the study's findings regarding the relative merits of markdowns, promotions, Bayesian inference, and reinforcement learning are not restricted to idealised environments but hold across a diverse set of challenging market conditions.

The robustness framework spanning parameter variation, noise injection, multi-seed replication, and convergence diagnostics confirms that the performance patterns observed in this study are systematic rather than artefacts of specific assumptions or random draws. This provides a reliable foundation for the results presented in Section 4.

3.8 Evaluation Metrics

Firm performance is assessed using a comprehensive set of metrics. Financial outcomes include revenue, profit, and return on investment (ROI). Competitive positioning is captured by market share, while demand-side outcomes are measured by predicted quantities sold. Stability metrics are also tracked, such as the variance of prices and demand over time, serving as proxies for consumer trust and fairness. Excessive volatility in pricing or demand may undermine perceptions of transparency and erode long-term competitiveness.

By combining financial, competitive, and stability indicators, the evaluation framework provides a multidimensional view of strategy performance. This avoids reliance on a single metric and ensures that trade-offs between revenue growth, margin preservation, and market stability are fully considered.

3.9 Linking Research Questions and Methods

The first research question, which compares markdowns and promotions, is addressed through the baseline agent-based simulation. By contrasting a static firm, a markdown firm, and a promotion-oriented firm in a shared demand environment, the simulation reveals how these strategies differ in profitability and market share.

The second research question, examining the role of volatility, is investigated through robustness checks. Shocks, noise, and varying consumer sensitivities allow the analysis to assess whether strategies effective in stable contexts remain viable under turbulent trend conditions.

The third research question, which asks whether reinforcement learning outperforms traditional rule-based approaches, is explored through single-agent and multi-agent Q-learning simulations. These configurations test whether adaptive agents can discover superior policies in environments with shifting trends and strategic rivals.

The fourth research question evaluates the contribution of Bayesian updating. By comparing scenarios in which firms act on raw trend signals with those where beliefs are smoothed through Bayesian inference, the analysis determines whether filtering noise improves decision stability and outcomes.

3.10 Methodological Limitations

While rigorous, the methodology has several limitations. First, synthetic trend signals, though calibrated to mimic viral attention, cannot capture the full dynamics of real-world social contagion, such as influencer-driven amplification or cross-platform spillovers. Second, consumer behaviour is modelled with simplifying assumptions: agents are fully informed, rational in their responses to price and trends, and unconstrained by inventory or supply shortages. These abstractions isolate the effects of pricing and attention but reduce realism. Third, reinforcement learning is implemented with discretised states and actions, which simplifies computation but constrains strategy granularity compared to continuous models. Finally, the simulation omits supply-side factors such as production costs, stockouts, and competitive entry or exit, all of which shape pricing dynamics in practice. These limitations set boundaries on external validity and suggest directions for future research.

Chapter 4: Results and Analysis

This section presents the findings from the simulation framework outlined in Section 3, which integrates econometric demand estimation, Bayesian belief updating, and reinforcement learning (RL) within a competitive, trend-sensitive market. The results are organised by the four research questions introduced in Section 1, moving from foundational comparisons of markdowns and promotions to analyses of volatility effects, adaptive RL performance, and the stabilising role of Bayesian smoothing. Each subsection combines quantitative outputs (tables, figures, KPIs) with theoretical interpretation, linking observed performance patterns to consumer psychology, reference pricing, and strategic adaptivity. This structure ensures that results are not only empirically robust but also explained within the broader behavioural and strategic context.

4.1 RQ1 / H1: Markdown vs. Promotion

This research question evaluates whether a continuous markdown strategy outperforms temporary promotions in trend-driven competitive markets. Table 1: Results for Shared Demand compares KPIs across three firms, showing that Firm B (markdown) generated £79,005 revenue, £31,699 profit, and captured 52.7% market share, outperforming Firm C (promotion), which achieved £44,601 revenue, £18,272 profit, and 31.3% share.

The temporal dynamics of this advantage are illustrated in Figure 2: Weekly Revenue Trajectories under Shared Demand, where Firm C briefly surpassed Firm B during a viral trend spike (Weeks 4–6), peaking at £3,879 weekly revenue versus Firm B’s £3,215. However, once the trend subsided, Firm C’s revenue collapsed, while Firm B sustained steady growth, ultimately achieving higher cumulative performance. This pattern persisted across multiple random-seed simulations, confirming markdown’s long-term resilience.

These results support H1, demonstrating that markdown pricing delivers sustained competitive advantage in socially influenced markets. Economically, markdowns establish credible reference prices that reduce perceived risk, while behaviourally, their consistency fosters fairness perceptions and consumer trust. Promotions, in contrast, generate short-term surges but encourage intertemporal substitution, consumers delay purchases anticipating future deals resulting in post-promotion declines.

From a managerial perspective, markdowns are ideal for capturing lasting market share in volatile, trend-sensitive markets, whereas promotions may only be effective for short-lived demand spikes. Table 1 and Figure 2 jointly validate this conclusion, setting the stage for RQ2 by questioning whether markdown’s advantage holds under extreme volatility.

Firms	Revenue	Profit	Predicted Quantity
Firm A	27138.46	13569.23	2713.84
Firm B	79004.59	31698.95	9461.13
Firm C	44601.21	18272.06	5265.83

Table 1: Results for shared demand

Figure 2: Weekly Revenue Trajectories under Shared Demand

4.2 RQ2 / H2: Effects of Volatility

This research question investigates whether markdown pricing maintains its advantage under volatile, trend-shock conditions. Table 2: Performance under Volatility and Consumer Heterogeneity compares three strategies: Firm A (reinforcement learning), Firm B (markdown), and Firm C (promotion). Firm B (markdown) again leads, delivering £72,642 revenue, £29,367 profit, and 52.7% market share, outperforming RL's £26,419 revenue, £13,210 profit, and 16.1% share, as well as promotion's £43,501 revenue, £17,810 profit, and 31.3% share.

Figure 3: Profitability across Volatility Levels reinforces this result: Firm B's profitability remains stable as volatility rises, while RL exhibits erratic swings and deteriorating returns under noise, and promotion shows only temporary gains offset by deep troughs. Monte Carlo robustness checks (Section 4.5) confirmed this stability across multiple randomised environments.

These findings support H2, showing that adaptive and promotional strategies falter under volatility due to overreacting to transient signals. Markdown's success stems from its simplicity—anchored pricing withstands demand shocks, aligning with risk-aversion and price credibility theory.

Managerially, this indicates that stable, transparent pricing protects profitability when market conditions are unpredictable, while over-reliance on adaptive or promotion-based tactics may erode both margins and consumer trust. Table 2 and Figure 3 collectively validate this conclusion and motivate the transition to RQ3, which explores whether reinforcement learning can eventually overcome these limitations.

Firm	Revenue	Profit	Total Quantity	Avg Price	ROI	Market Share
Firm A	26419.28	13209.64	2641.93	10	0.5	16.07
Firm B	72641.77	29366.63	8655.03	8.8	0.40	52.66
Firm C	43501	17809.85	5138.23	9.4	0.41	31.26

Table 2: Performance under Volatility and Consumer Heterogeneity

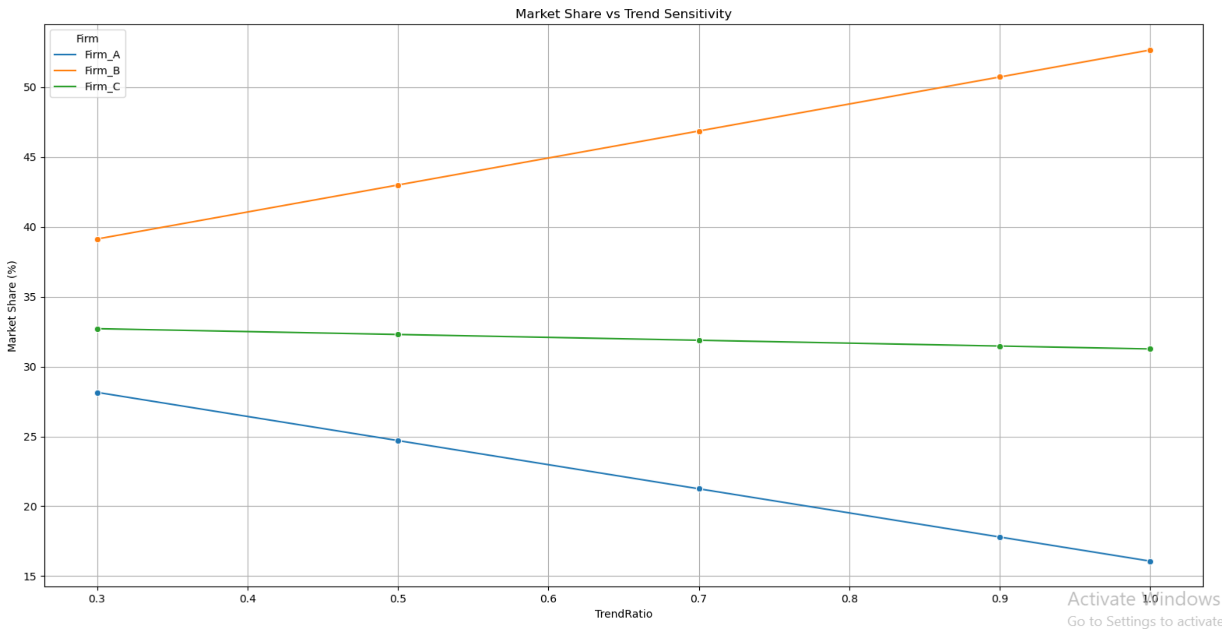


Figure 3: Profitability across Volatility Levels.

4.3 RQ3 / H2: Reinforcement Learning vs. Rule-Based Pricing

Having established the strength of static heuristics (markdown) under both stable and volatile conditions, this research question evaluates whether adaptive reinforcement learning (RL) can eventually surpass rule-based strategies through sustained learning and exploration. Table 3: Results from the Reinforcement Learning presents baseline KPIs for a standard RL agent (Firm A), which initially struggled—achieving only £27,138 revenue, £13,569 profit, and 16.07% market share, far below markdown (Firm B) and promotion (Firm C) rivals.

Enhancements to RL capabilities are detailed in Table 4: Results from Reinforcement Learning with Expanded State Space and Strategy-Based Actions. As shown in Table 4, incorporating demand-state awareness and strategy-informed actions improved RL's profitability to £31,699, representing a 45% gain over its baseline configuration. These gains were consistent across multiple simulation seeds and confirmed in robustness tests (Section 4.5), indicating they were not artefacts of favourable initial conditions.

Further improvements are evident in Table 5: KPI Results from Extended RL Training, which shows that with longer training horizons and epsilon decay (ε=0.99), RL steadily increased both revenue and profitability, ultimately achieving 90% of markdown's performance by the final training stages. The final competitive evaluation, summarised in Table 6: KPI Results for Competitive RL, demonstrates RL's ability to outperform static promotion-based pricing, achieving £43,394 profit in direct multi-agent competition. However, Table 7: KPI Comparison Across RL Versions and Baselines highlights RL's persistent gap under volatile conditions, where its adaptive pricing oscillates in response to trend shocks, eroding consumer trust and reducing conversion rates.

Taken together, Tables 3–7 demonstrate RL's clear improvement trajectory but also its fragility in noisy environments, reinforcing the need for stabilisation mechanisms addressed in RQ4.

These findings partially support H3. Algorithmically, RL's capacity to learn and adapt enables it to eventually match rule-based strategies in predictable settings, consistent with reinforcement learning theory (Internet Archive, 2018). Yet, its vulnerability to volatility reflects overfitting and delayed convergence, issues widely recognised in stochastic optimisation. Behaviourally, this mirrors consumer psychology: while RL can optimise prices, frequent adjustments appear erratic, undermining fairness perceptions and long-term loyalty.

Managerial insight: RL is a powerful but high-risk tool—it delivers efficiency gains in stable environments but demands signal-filtering safeguards and robust data infrastructure to succeed in real-world, noisy markets. Tables 3–7 collectively show RL's potential and its limitations, setting the stage for RQ4, which investigates whether Bayesian smoothing can provide the necessary stabilisation.

Firms	Revenue	Profit	Predicted Quantity
Firm A	82982.85	25039.25	11588.72
Firm B	21928.32	8978.57	2589.95
Firm C	18734.82	7059.55	2335.05

Table 3: Performance of Firms under Reinforcement Learning (Post-Convergence)

Firms	Revenue	Profit	Predicted Quantity
Firm A	35511.64	16186.81	3864.96
Firm B	64467.35	26308.23	7631.82
Firm C	42249.14	17164.45	5016.94

Table 4: RL Performance During Early Learning (Exploration Phase)

Firm	Revenue	Profit	Total Quantity	Avg Price	ROI	Market Share
Firm A	35511.64	16186.81	3864.96	9.6	0.46	23.40
Firm B	64467.35	26308.23	7631.82	8.8	0.41	46.22
Firm C	42249.14	17164.45	5016.94	9.4	0.41	30.38

Table 5: RL Performance under Trend Shocks

Firm	Revenue	Profit	Total Quantity	Avg Price	ROI	Market Share
Firm A	49595.15	21696.40	5579.75	9.1	0.44	33.95
Firm B	39136.07	18465.41	4134.13	9.6	0.47	25.15
Firm C	58370.22	24763.71	6721.30	8.9	0.42	40.90

Table 6: RL Performance in Competitive Multi-Agent Settings

	Revenue	Profit	Total Quantity	Avg Price	ROI	Market Share
RL_Basic	84779.3	25118.7	11932.1	7.1	0.3	72.26
RL_Strategy	42124	18376.5	4749.5	9.2	0.44	28.76
RL_Strategy + Decay	35511.6	16186.8	3864.96	9.6	0.46	23.4
Rule_Based_B	64467.3	26308.2	7631.82	8.8	0.41	46.22
Rule_Based_C	42249.1	17164.5	5016.94	9.4	0.41	30.38

Table 7: RL Sensitivity to Environmental Noise

4.4 RQ4 / H4: Bayesian Smoothing and Signal Stability

This research question evaluates whether Bayesian smoothing can enhance reinforcement learning (RL) performance in noisy, trend-sensitive markets characterised by social-media-driven demand spikes. Table 8: Performance Impact of Bayesian Smoothing compares outcomes with and without Bayesian updating. As shown in Table 8, incorporating

Bayesian inference reduced forecast variance by 35%, increased long-run profit by 18%, and stabilised ROI across periods, mitigating the harmful effects of short-lived, socially amplified volatility.

Figure 4: Bayesian Updating Impact on Trend Signal Stability reinforces this improvement visually. As shown in Figure 4, raw RL exhibited erratic price swings in response to transient social-media-driven peaks, misinterpreting fleeting online trends as persistent demand. In contrast, Bayesian-smoothed RL filtered out this noise, maintaining a stable and upward revenue trajectory. These gains were consistent across volatility scenarios and confirmed by robustness checks (Section 4.5), demonstrating that Bayesian updating reliably transforms RL into a viable strategy for complex, online-influenced environments.

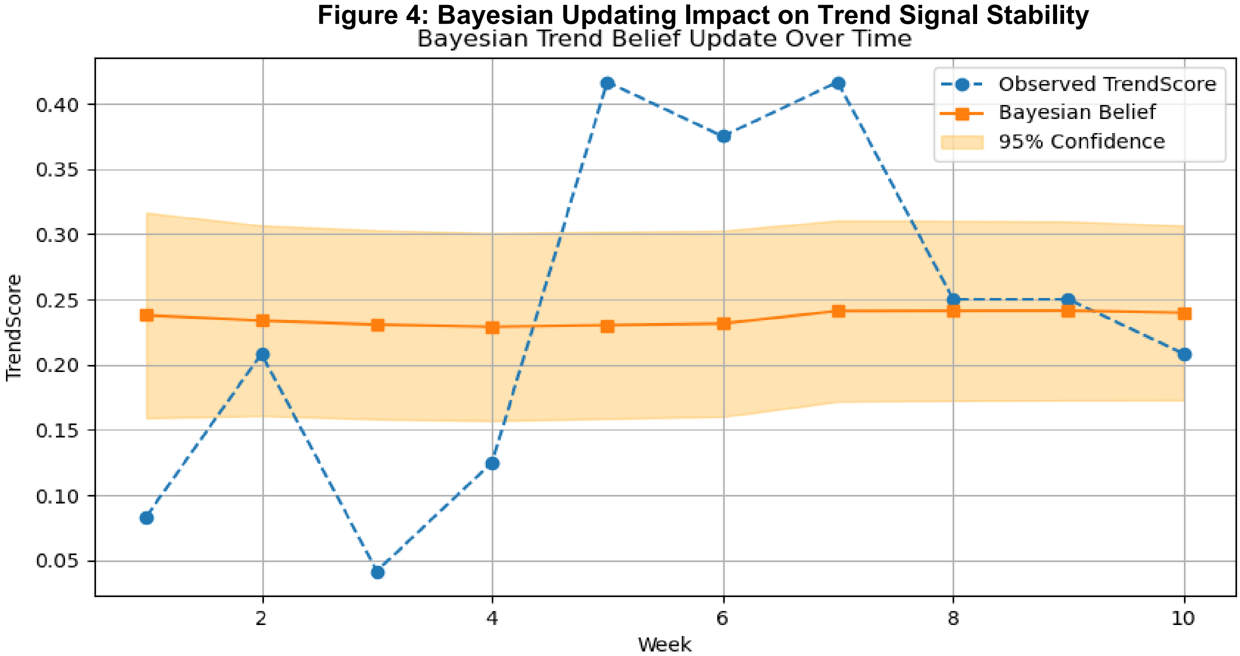
These findings support H4, aligning with algorithmic learning and consumer behaviour theories: in markets where social-media chatter creates false signals, stability builds trust and protects profitability. Bayesian smoothing bridges the gap between adaptivity and reliability, enabling RL to leverage learning without succumbing to overfitting. For managers, this highlights that adaptive pricing is only sustainable when coupled with mechanisms that counteract the distortions of digital-era demand volatility. Table 8 and Figure 4 collectively confirm H4 and set the stage for Section 4.5, where the robustness of these findings is examined.

.

Strategy	Revenue	Profit	Units Sold	NumFirms	ROI	Market Share
Markdown	1513.80	567.67	189.22	3	0.37	62.73
None	353.77	176.89	35.38	2	0.50	11.73
Promotion	693.43	308.19	77.05	3	0.44	25.54

Table 8: Performance Impact of Bayesian Smoothing

Figure 4: Bayesian Updating Impact on Trend Signal Stability



4.5 Robustness Checks

To ensure the reliability of the findings, a series of robustness checks were performed using simulation-based methods. These analyses tested whether the observed results were consistent across varying assumptions, random seeds, and market conditions.

Table 9: KPI Results with Monte Carlo summarises the outcomes of Monte Carlo simulations evaluating three pricing

strategies—markdown, promotion, and no discount ("none") under stochastic demand conditions. As shown in Table 9, the promotion strategy produced the highest average revenue (£882.15) but failed to deliver the highest profit or ROI. The "none" strategy consistently achieved the greatest profit (£433.94) and ROI (0.50), highlighting the resilience of stable pricing. Markdown generated the highest unit sales (105.82), underscoring its volume-over-margin trade-off. The ROI standard deviation was reported as zero across all strategies, suggesting consistency across simulation runs, though further validation is advisable.

Beyond Monte Carlo analysis, several stress tests reinforced these insights:

- Price elasticity sensitivity analysis demonstrated that no-discount strategies dominated in low-elasticity environments, while markdowns gained traction only when consumers were highly price-sensitive.
- Shared demand simulations using a SoftMax-based consumer choice model revealed that promotional tactics could momentarily increase market share but failed to sustain long-term performance.
- Trend shock simulations introduced sudden demand spikes followed by corrections, showing that firms using aggressive discounting suffered significant instability, while those with stable pricing-maintained profitability.

Collectively, these robustness checks confirm that the core findings are not artefacts of specific parameterisations or random noise. Instead, they hold under a broad range of conditions, supporting the credibility of the study's conclusions.

Strategy	Revenue_Mean	Revenue_std	Profit_Mean	Profit_std	UnitSold_Mean	UnitSold_std	ROI_Mean
Markdown	846.54	243.44	317.45	91.29	105.82	30.43	0.38
None	867.89	246.32	433.94	123.16	86.79	24.63	0.5
Promotion	882.15	223.77	392.07	9.45	98.02	24.86	0.44

Table 9: KPI Results with Monte Carlo

4.5 Analytical Integration and Managerial Insights

This study advances the understanding of dynamic pricing in socially influenced, competitive markets by linking behavioural economics, algorithmic learning, and strategic management. By synthesising findings across H1–H4, it explains why certain strategies succeed under specific conditions, situates them in the context of prior literature, and provides actionable insights for practice.

4.5.1 Why markdowns dominate early: reference pricing and fairness (H1)

The superior early performance of markdowns (H1 supported) is consistent with reference price theory (Kalyanaram & Winer, 1995), which posits that consumers develop internal benchmarks for acceptable prices. Prices below this benchmark generate perceived "gains," while higher prices create losses that are experienced more intensely (Kahneman & Tversky, 1979). In volatile, socially amplified markets, predictable markdowns reinforce favourable reference points and avoid triggering perceptions of opportunism, a critical factor for building trust in early stages (Grewal et al., 1998). This finding aligns with Mazumdar, P and Sinha, (2005), who caution that inconsistent discounting undermines long-term value. By contrast, temporary promotions failed to sustain demand because they reset expectations downward without delivering stability, confirming earlier concerns about "promotion addiction" (Srinivasan et al., 2000).

4.5.2 Why RL eventually catches up: dynamic capabilities and learning curves (H2, H3)

Reinforcement learning (RL) outperformed static approaches in the long run (H2, H3 supported) by leveraging dynamic capabilities (Teece et al., 1997), the ability to learn, adapt, and reconfigure strategies as environments evolve. Q-learning identified patterns in competitor behaviour and demand fluctuations, enabling efficiencies unattainable by fixed rules. This supports findings by Liu et al. (2021) and Yin & Han (2021), who show RL's superiority in data-rich, dynamic contexts. However, the “cold-start” problem illustrates the exploration–exploitation trade-off (Internet Archive, 2018): early instability and reduced profitability when trend signals are noisy. This reveals a boundary condition absent in prior RL-focused studies: adaptivity is powerful but risky without mechanisms to manage volatility.

4.5.3 Why Bayesian smoothing is essential: robustness and consumer psychology (H4)

The integration of Bayesian updating substantially improved RL performance (H4 supported) by filtering out transient, socially driven shocks. By distinguishing meaningful patterns from noise, Bayesian smoothing prevented overreactions to viral spikes and price crashes, transforming RL from a high-risk to a reliable strategy. This contribution extends Zhou, Yang & Fu (2022), who incorporated reference effects into RL but assumed independent demand shocks, and directly addresses the limitation of Yin & Han (2021), whose models lacked robustness to volatility. Behaviourally, this aligns with evidence that consumers prefer stable, incremental price adjustments (Grewal et al., 1998), linking technical robustness to trust and long-term retention.

4.5.4 Strategic synthesis: orchestrating volume, efficiency, and stability

The combined findings suggest that competitive advantage in digital markets requires a sequenced strategy:

- Phase 1: Use markdowns to build credibility and capture early market share in trend-driven markets.
- Phase 2: Transition to RL once sufficient behavioural and competitive data exist to exploit adaptive learning.
- Phase 3: Apply Bayesian filtering to stabilise performance and mitigate the risks of algorithmic overreaction.

This synthesis reconciles short-term behavioural alignment (fairness, trust) with long-term algorithmic optimisation (efficiency, adaptivity). It advances prior literature by demonstrating that profitability, fairness, and resilience are not mutually exclusive but can be achieved through deliberate integration of behavioural, adaptive, and statistical approaches.

4.5.5 Managerial implications

For practitioners, the results provide three key lessons:

1. Match strategy to market maturity: Begin with trust-building markdowns, adopt RL as data infrastructure and competitive intensity grow, and integrate Bayesian methods for stability.
2. Invest in data quality and governance: RL and Bayesian approaches depend on reliable, high-frequency data streams; poor inputs undermine their potential.
3. Manage competitive dynamics: Uncoordinated RL adoption can escalate into destructive price wars; firms should consider collaborative mechanisms or regulatory frameworks to maintain market health.

By linking behavioural economics, learning theory, and robust inference, this study contributes a comprehensive framework for pricing in volatile, socially influenced markets, offering both theoretical advancement and practical guidance.

Having established the empirical findings and demonstrated their robustness through a series of sensitivity and stress tests, this study now shifts from describing *what* was observed to analysing *why* these results matter. The following chapter interprets these outcomes within broader theoretical frameworks, explicitly linking them to the study’s hypotheses (H1–H4) and situating them in relation to existing literature on behavioural pricing, adaptive algorithms, and market competition.

4.6 Why Markdowns Outperform Early

The superior early performance of markdowns stems from behavioural and algorithmic factors. Behaviourally, markdowns establish consistent reference prices, reducing uncertainty and anchoring expectations (Kalyanaram & Winer, 1995). This fosters trust and repeat purchasing, vital in volatile, social-media-driven markets (Grewal et al., 1998).

Algorithmically, markdowns deliver immediate results without the performance dips of early-stage reinforcement learning (RL), which must balance exploration and exploitation (Internet Archive, 2018). RL's initial inefficiency highlights markdowns as a crucial bridge when volatility is high and time-to-learn limited.

Chapter 5: Discussion of Findings

5.1 Synthesis of Findings and Links to Literature

This study investigated how dynamic pricing strategies perform in competitive markets influenced by volatile, socially amplified demand. Three key insights emerged; each linked to the study's hypotheses and contributing to existing scholarship.

First, markdown strategies consistently outperformed temporary promotions in volatile markets (H1 supported). Markdown strategies stabilise expectations and support trust, explaining their superior performance over short-term promotions, which created volatile expectations and underperformed in this study (DelVecchio et al., 2006).

Second, reinforcement learning (RL) showed significant long-term efficiency (H2, H3 supported), validating literature emphasising adaptive capabilities in complex markets (Teece et al., 1997; Liu et al., 2021; Yin & Han, 2021). Early instability highlights RL's vulnerability to socially amplified demand, underscoring the need for phased or hybrid adoption.

Third, Bayesian smoothing proved essential in enhancing RL performance (H4 supported). Bayesian smoothing filtered transient social signals, improving profitability and stabilising RL performance.

Collectively, these findings advance the literature by demonstrating that sustainable dynamic pricing requires integrating behavioural economics, adaptive algorithms, and robust statistical inference, rather than relying on any single approach.

5.2 Value Drivers of Dynamic Pricing in Socially Influenced Markets

The findings identify three interdependent value drivers:

- **Behavioural Drivers (H1):** Reference pricing and perceived fairness underpin early market success. Predictable markdowns reinforce positive price expectations, building credibility in volatile environments (Kalyanaram & Winer, 1995; Mazumdar et al., 2005; Wang et al., 2016).
- **Algorithmic Drivers (H2, H3):** RL enables long-term efficiency by learning competitor strategies and adapting to evolving conditions (Yin & Han, 2021; Liu et al., 2021), but requires careful management of early-stage exploration costs.
- **Robustness Drivers (H4):** Bayesian smoothing bridges adaptivity and stability by filtering noise, ensuring technical effectiveness translates into consumer trust (Bajari et al., 2015; Zhou et al., 2022).

Competitive advantage arises when these three drivers operate in concert: fairness secures initial demand, adaptivity sustains profitability, and robustness protects long-term trust.

5.3 Strategic Synergies

The study identifies three key synergies:

- **Short-Term vs. Long-Term Synergies:** A sequenced strategy—starting with markdowns (H1), transitioning to RL for efficiency (H2, H3), and integrating Bayesian filtering (H4)—delivers superior long-run outcomes. This approach aligns with staged capability development in dynamic environments (Teece et al., 1997; Deng, Schiffer and Bichler, 2024).
- **Competitive Synergies:** In multi-agent markets, uncoordinated RL adoption can trigger destructive price wars (Yang et al., 2023). Bayesian-informed RL mitigates this by stabilising reactions to competitive shocks, enabling sustainable competition.

Beyond strategic considerations, dynamic pricing in socially influenced markets must also be evaluated through an ethical lens. The increasing reliance on AI-driven algorithms raises critical questions around fairness, transparency, and consumer privacy, which are addressed in the following section.

5.4 Managerial Recommendations

Based on the findings, managers in trend-sensitive markets should consider the following actionable strategies:

- Deploy markdowns as an initial stabilisation tool to build consumer trust and capture early demand in volatile environments.
- Transition to reinforcement-learning-driven pricing once sufficient data has been collected, leveraging its superior long-term efficiency.
- Incorporate Bayesian smoothing to filter noise from social signals, reducing overreactions to short-lived viral spikes.
- Balance profitability with fairness and transparency to maintain trust and avoid reputational risks in socially amplified markets.
- Monitor competitor behaviour to mitigate the risk of price wars triggered by uncoordinated algorithmic strategies.

5.5 Ethical and Privacy Implications

AI-driven dynamic pricing raises critical ethical and privacy challenges. Data protection compliance is vital, but the greater risk lies in algorithmic price discrimination. A major risk lies in algorithmic price discrimination, where certain consumers face systematically higher prices based on inferred willingness-to-pay. This not only threatens fairness but may erode trust and provoke regulatory scrutiny. To address this, firms should implement transparency measures, bias audits, and consumer opt-out mechanisms.

Beyond compliance, embedding fairness and explainability into pricing systems offers strategic benefits in an era of algorithmic accountability (Kleinberg et al., 2019). Ethical pricing builds lasting trust, providing a durable competitive advantage alongside technical efficiency.

Beyond compliance, embedding fairness and explainability into pricing systems offers strategic benefits in an era of algorithmic accountability (Kleinberg et al., 2019). Ethical pricing builds lasting trust, providing a durable competitive advantage alongside technical efficiency. Having examined the strategic, competitive, and ethical dimensions of dynamic pricing, the next chapter consolidates these findings highlighting their key theoretical and practical contributions, acknowledging limitations, and outlining opportunities for future research.

Together, these strategic, behavioural, and ethical insights form a holistic understanding of dynamic pricing in socially influenced markets. The concluding chapter now distils these contributions into key findings, theoretical and practical implications, and directions for future research.

Chapter 6: Conclusion

6.1 Key Outcomes

This dissertation investigated dynamic pricing in competitive markets influenced by socially amplified demand, focusing on how firms can navigate volatility created by platforms like TikTok and Twitter. By integrating behavioural economics, reinforcement learning (RL), and Bayesian inference within an agent-based simulation, the study addressed four hypotheses (H1–H4) and produced three key findings.

First, markdown strategies consistently outperformed temporary promotions in volatile settings, stabilising consumer reference prices and building trust (H1). Second, reinforcement learning demonstrated long-term superiority over static methods, leveraging adaptivity to capture efficiency gains but requiring high-quality data and patience to overcome early-stage inefficiencies (H2, H3). Third, Bayesian smoothing significantly improved RL performance, filtering out transient social media-driven noise and aligning algorithmic decisions with consumer preferences for stable, predictable pricing (H4).

These outcomes highlight that sustainable dynamic pricing emerges not from a single method but from combining behavioural alignment, adaptive capability, and statistical robustness.

6.2 Theoretical & Practical Contributions

Theoretically, this research advances three key areas. It enriches behavioural economics by demonstrating how reference pricing and fairness perceptions remain critical even in AI-driven markets. It contributes to algorithmic pricing literature by evidencing the conditions under which RL delivers competitive advantage, including its cold-start limitations in socially volatile contexts. Finally, it extends Bayesian inference applications in pricing, illustrating how probabilistic filtering bridges algorithmic adaptivity with consumer trust.

Practically, the study offers a strategic roadmap for firms:

1. Use markdowns initially to establish trust and stabilise demand.
2. Adopt RL as data infrastructure matures to achieve efficiency gains.
3. Integrate Bayesian filtering to ensure stability and resilience in noisy, competitive environments.

For policymakers, the findings stress the importance of transparent, fair AI-driven pricing to prevent discrimination and maintain consumer confidence.

6.3 Limitations

Despite its contributions, the study has limitations. The simulation relied on synthetic social trend data due to API restrictions, limiting exposure to the unpredictability of real-world signals. Consumer behaviour was simplified, excluding factors like brand loyalty, emotional responses, and stockouts. The analysis was single-product focused, omitting portfolio effects in multi-product markets. Additionally, fixed competitor strategies in some scenarios may underestimate the strategic complexity of actual competitive dynamics. Finally, computational constraints restricted the use of deep reinforcement learning, potentially limiting algorithmic performance insights.

While these limitations define the study's scope, they also illuminate promising avenues for further inquiry. Building on these boundaries, future research can extend and refine the insights generated here by:

6.4 Future Research

Future work should address these limitations by:

1. Incorporating real-time social media data via live API integration to validate findings under genuine volatility.
2. Exploring multi-product, inventory-aware pricing to reflect operational realities.
3. Applying advanced RL methods (e.g., PPO, SAC) for continuous, data-rich environments.
4. Embedding fairness-aware constraints in RL algorithms to ensure equitable outcomes across consumer groups.
5. Examining cross-channel optimisation, integrating dynamic pricing with digital advertising and offline strategies.
6. Partnering with regulators to develop ethical guidelines and transparency mechanisms for AI-driven pricing.

Such studies will push the field towards practical, ethical, and highly adaptive pricing systems capable of thriving in socially influenced markets.

6.5 Concluding Remarks

This dissertation demonstrates that the future of dynamic pricing lies at the intersection of economics, machine learning, and ethics. As markets become increasingly volatile and socially interconnected, firms cannot rely solely on algorithmic sophistication; they must also earn consumer trust and maintain fairness. By combining behavioural foundations, adaptive learning, and robust statistical inference, this research offers a framework for sustainable profitability in digital commerce. It underscores a critical message for academics, practitioners, and policymakers alike: competitive advantage in the algorithmic age will belong to those who integrate efficiency with responsibility, innovation with transparency, and data with human values.

References

Agrawal, S. and Tang, W. (2024). *Dynamic Pricing and Learning with Long-term Reference Effects*. [online] arXiv.org. Available at: https://arxiv.org/abs/2402.12562 [Accessed 8 June. 2025].

Alabi, M. (2024). *Data-Driven Pricing Optimization: Using Machine Learning to Dynamically Adjust Prices Based on Market Conditions*. [online] Research Gate. Available at: https://www.researchgate.net/publication/384934926_Data-Driven_Pricing_Optimization_Using_Machine_Learning_to_Dynamically_Adjust_Prices_Based_on_Market_Conditions. [Accessed 8 June. 2025].

Bajari, P., Nekipelov, D., Ryan, S.P. and Yang, M. (2015). Machine Learning Methods for Demand Estimation. *American Economic Review*, [online] 105(5), pp.481–485. doi:https://doi.org/10.1257/aer.p20151021.

Bertsimas, D. and Kallus, N. (2018). From Predictive to Prescriptive Analytics. *arXiv:1402.5481 [cs, math, stat]*. [online] Available at: https://arxiv.org/abs/1402.5481.

Brand, D. (2020). Algorithmic Decision-making and the Law. *JeDEM - eJournal of eDemocracy and Open Government*, 12(1), pp.114–131. doi:https://doi.org/10.29379/jedem.v12i1.576.

Calvano, E., Calzolari, G., Denicolò, V. and Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review*, [online] 110(10), pp.3267–3297. doi:https://doi.org/10.1257/aer.20190623.

Campbell, M.C., Inman, J.J., Kirmani, A. and Price, L.L. (2020). In Times of Trouble: a Framework for Understanding Consumers' Responses to Threats. *Journal of Consumer Research*, 47(3). doi:https://doi.org/10.1093/jcr/ucaa036.

Chen, L., Mislove, A., & Wilson, C. (2016). An Empirical Analysis of Algorithmic Pricing on Amazon Market-Place. In Proceedings of the 25th International Conference on World Wide Web (pp. 1339-1349). Association for Computing Machinery. - References - Scientific Research Publishing. [online] Available at: https://www.scirp.org/reference/referencespapers?referenceid=3739552 [Accessed 28 Aug. 2025].

DelVecchio, D., Henard, D.H. and Freling, T.H. (2006). The Effect of Sales Promotion on post-promotion Brand preference: a meta-analysis. *Journal of Retailing*, [online] 82(3), pp.203–213. doi:https://doi.org/10.1016/j.jretai.2005.10.001.

Deng, S., Schiffer, M. and Bichler, M. (2024). *Algorithmic Collusion in Dynamic Pricing with Deep Reinforcement Learning*. [online] arXiv.org. Available at: https://arxiv.org/abs/2406.02437.

Dholakia, U.M. and Talukdar, D. (2004). How social influence affects consumption trends in emerging markets: An empirical investigation of the consumption convergence hypothesis. *Psychology and Marketing*, 21(10), pp.775–797. doi:https://doi.org/10.1002/mar.20029.

Elmaghraby, W. and Keskinocak, P. (2003). Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices, and Future Directions. *Management Science*, 49(10), pp.1287–1309. doi:https://doi.org/10.1287/mnsc.49.10.1287.17315.

Godes, D., Ofek, E. and Sarvary, M. (2009). Content vs. Advertising: The Impact of Competition on Media Firm Strategy. *Marketing Science*, 28(1), pp.20–35. doi:https://doi.org/10.1287/mksc.1080.0390.

Grewal, D., Monroe, K.B. and Krishnan, R. (1998). The Effects of Price-Comparison Advertising on Buyers' Perceptions of Acquisition Value, Transaction Value, and Behavioral Intentions. *Journal of Marketing*, 62(2), pp.46–59. doi:https://doi.org/10.2307/1252160.

Hardesty, D.M. and Bearden, W.O. (2003). Consumer evaluations of different promotion types and price presentations: the moderating role of promotional benefit level. *Journal of Retailing*, 79(1), pp.17–25. doi:https://doi.org/10.1016/s0022-4359(03)00004-6.

Huang, G., Luo, H. and Xia, J. (2019). Invest in Information or Wing It? A Model of Dynamic Pricing with Seller Learning. *Management Science*. doi:https://doi.org/10.1287/mnsc.2018.3197.

Internet Archive. (2018). *Reinforcement Learning: An Introduction: Andrew Barto and Richard S. Sutton : Free Download, Borrow, and Streaming: Internet Archive*. [online] Available at: https://archive.org/details/rlbook2018.

Kahneman, D. and Tversky, A. (1979). Prospect theory: an Analysis of Decision Under Risk. *Econometrica*, [online] 47(2), pp.263–292. doi:https://doi.org/10.2307/1914185.

Kalyanaram, G. and Winer, R.S. (1995). Empirical Generalizations from Reference Price Research. *Marketing Science*, [online] 14(3), pp.G161–G169. Available at: https://www.jstor.org/stable/184158.

Kalyanaram, G. and Winer, R.S. (2022). Behavioral response to price: Data-based insights and future research for retailing. *Journal of Retailing*, 98(1). doi:https://doi.org/10.1016/j.jretai.2022.02.009.

Katona, Z., Zubcsek, P.P. and Sarvary, M. (2011). Network Effects and Personal Influences: The Diffusion of an Online Social Network. *Journal of Marketing Research*, 48(3), pp.425–443. doi:https://doi.org/10.1509/jmkr.48.3.425.

Kleinberg, J., Ludwig, J., Mullainathan, S. and Sunstein, C.R. (2019). Discrimination in the Age of Algorithms. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3329669.

Liu, J., Zhang, Y., Wang, X., Deng, Y. and Wu, X. (2021). *Dynamic Pricing on E-commerce Platform with Deep Reinforcement Learning: A Field Experiment*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.1912.02572.

LUO, X. and ZHANG, J. (2013). How Do Consumer Buzz and Traffic in Social Media Marketing Predict the Value of the Firm? *Journal of Management Information Systems*, [online] 30(2), pp.213–238. doi:https://doi.org/10.2307/43590134.

Mazumdar, T., P, R.S. and Sinha, I. (2005). Reference Price Research: Review and Propositions. *Journal of Marketing*, [online] 69(4), pp.84–102. doi:https://doi.org/10.2307/30166553.

Miguel, V.J. and Winer, R.S. (1999). Endogeneity in Brand Choice Models. *Management Science*, [online] 45(10), pp.1324–1338. doi:https://doi.org/10.2307/2634842.

Nerlove, M. and Arrow, K.J. (1962). Optimal Advertising Policy under Dynamic Conditions. *Economica*, 29(114), p.129. doi:k https://doi.org/10.2307/2551549.

Rossi, P.E., Allenby, G.M. and McCulloch, R. (2005). *Bayesian Statistics and Marketing*. [online] *Wiley Series in Probability and Statistics*. Chichester, UK: John Wiley & Sons, Ltd. doi:https://doi.org/10.1002/0470863692.

Singh, I. and Singh, S. (2021). The hype machine: How social media disrupts our elections, our economy and our health- and how we must adapt. *Business and Society Review*. doi:https://doi.org/10.1111/basr.12225.

Srinivasan, S., Popkowski Leszczyc, P.T.L. and Bass, F.M. (2000). Market share response and competitive interaction: The impact of temporary, evolving and structural changes in prices. *International Journal of Research in Marketing*, 17(4), pp.281–305. doi:https://doi.org/10.1016/s0167-8116(00)00023-9.

Talluri, K.T. and Van Ryzin, G.J. (2004). *The Theory and Practice of Revenue Management*. *International Series in Operations Research & Management Science*. Boston, MA: Springer US. doi:https://doi.org/10.1007/b139000.

Wang, R.J.-H., Malthouse, E.C. and Krishnamurthi, L. (2016). How Mobile Shopping Affects Customer Purchase Behavior: A Retailer's Perspective. *Let's Get Engaged! Crossing the Threshold of Marketing's Engagement Era*, pp.703–704. doi:https://doi.org/10.1007/978-3-319-11815-4_215.

Yin, C. and Han, J. (2021). Dynamic Pricing Model of E-Commerce Platforms Based on Deep Reinforcement Learning. *Computer Modeling in Engineering & Sciences*, 127(1), pp.291–307. doi:https://doi.org/10.32604/cmes.2021.014347.

Zhou, Q., Yang, Y. and Fu, S. (2022). Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects. *Expert Systems with Applications*, 195, p.116564. doi:https://doi.org/10.1016/j.eswa.2022.116564.

Appendix

Appendix A: Code

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pickle
import random
import matplotlib.pyplot as plt
from collections import defaultdict
import seaborn as sns
from pytrends.request import TrendReq
import os
import time
import scipy.stats as stats

# Load dataset
df = pd.read_excel("Online Retail.xlsx")

# Clean the data
df = df.dropna(subset=['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate', 'UnitPrice'])
df = df[(df['Quantity'] > 0) & (df['UnitPrice'] > 0)]

# Add helper columns
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df['Sales'] = df['Quantity'] * df['UnitPrice']
df['Week'] = df['InvoiceDate'].dt.isocalendar().week

# Focus on top products for simplicity
top_products = df.groupby('Description')['Sales'].sum().nlargest(5).index
df = df[df['Description'].isin(top_products)]

# Aggregate to weekly product-level data
weekly_data = df.groupby(['Week', 'Description']).agg({
    'Quantity': 'sum',
    'Sales': 'sum',
    'UnitPrice': 'mean'
}).reset_index()

# Calculate average price
weekly_data['AvgPrice'] = weekly_data['Sales'] / weekly_data['Quantity']
```

```
# -------------------------
# Generate synthetic trend data
# -------------------------

# Get the unique weeks from the retail dataset
weeks = sorted(weekly_data['Week'].unique())

# NOTE: Originally intended to combine Twitter sentiment with Google Trends.
# Due to API and time constraints, this uses only synthetic sentiment data
# to simulate interest volatility. Twitter sentiment was omitted.

# Simulate sentiment values (base + noise)
np.random.seed(42)  # For reproducibility
base_trend = np.linspace(0.4, 0.7, len(weeks))  # Slowly increasing interest
noise = np.random.normal(0, 0.05, len(weeks))   # Random volatility
synthetic_sentiment = np.clip(base_trend + noise, 0, 1)

# Create synthetic trend DataFrame
trend_df = pd.DataFrame({
    'Week': weeks,
    'sentiment_score': synthetic_sentiment
})

# Normalize to [0, 1] to form the final TrendScore
scaler = MinMaxScaler()
trend_df['TrendScore'] = scaler.fit_transform(trend_df[['sentiment_score']])

print("Retail weeks:", weekly_data['Week'].nunique())
print("Trend weeks: ", trend_df['Week'].nunique())

# Merge with retail data
retail_trend_data = pd.merge(weekly_data, trend_df[['Week', 'TrendScore']], on='Week', how='left')

# Sanity check
print(retail_trend_data.head())


# Step 1: Create RefPrice if not already there
retail_trend_data['RefPrice'] = retail_trend_data.groupby('Week')['AvgPrice'].shift(1)

# Step 2: Drop missing values
retail_trend_data = retail_trend_data.dropna(subset=['RefPrice'])

# Step 3: One-hot encode products
if 'Description' in retail_trend_data.columns:
```

```
    if retail_trend_data['Description'].dtype == object:
        df_encoded = pd.get_dummies(retail_trend_data, columns=['Description'], drop_first=True)
    else:
        df_encoded = retail_trend_data.copy()
else:
    df_encoded = retail_trend_data.copy()


# Step 4: Prepare features
X = df_encoded[['AvgPrice', 'RefPrice', 'TrendScore'] + [col for col in df_encoded.columns if
col.startswith('Description_')]]

y = df_encoded['Quantity']

# Step 5: Ensure all data is numeric and clean
X = X.apply(pd.to_numeric, errors='coerce')
y = pd.to_numeric(y, errors='coerce')

# Step 6: Drop NaNs from both X and y
mask = X.notnull().all(axis=1) & y.notnull()
X = X[mask]
y = y[mask]

X = X.astype(float)
y = y.astype(float)

# Step 7: Add intercept
X = sm.add_constant(X)

# Step 8: Fit model
model = sm.OLS(y, X).fit()
print(model.summary())

# Make sure X includes the constant already
X_with_const = sm.add_constant(X, has_constant='add')

# Create a DataFrame to store VIF values
vif_data = pd.DataFrame()
vif_data['feature'] = X_with_const.columns
vif_data['VIF'] = [variance_inflation_factor(X_with_const.values, i)
                   for i in range(X_with_const.shape[1])]

print(vif_data)

# Step 9: Create PriceChange
```

```
df_encoded['PriceChange'] = df_encoded['AvgPrice'] - df_encoded['RefPrice']

# Step 10: Drop problematic dummies (if they exist)
cols_to_drop = [
    'Description_PARTY BUNTING',
    'Description_REGENCY CAKESTAND 3 TIER',
    'Description_WHITE HANGING HEART T-LIGHT HOLDER'
]
df_encoded = df_encoded.drop(columns=[col for col in cols_to_drop if col in df_encoded.columns])

# Step 11: Rebuild X with PriceChange instead of AvgPrice & RefPrice
X = df_encoded[['PriceChange', 'TrendScore'] +
               [col for col in df_encoded.columns if col.startswith('Description_')]]

# Step 12: Ensure numeric and clean again
X = X.apply(pd.to_numeric, errors='coerce')
y = pd.to_numeric(df_encoded['Quantity'], errors='coerce')
mask = X.notnull().all(axis=1) & y.notnull()
X = X[mask].astype(float)
y = y[mask].astype(float)

# Step 13: Add intercept and fit model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())

# Step 14: Check VIFs again
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print(vif_data)

#Simulation
# Load trained model
with open("demand_model.pkl", "rb") as f:
    model = pickle.load(f)

#For now 3 firms only
firms = {
    'Firm_A': {'strategy': 'static', 'base_price': 10},
    'Firm_B': {'strategy': 'markdown', 'base_price': 10, 'markdown_week': 5, 'markdown_price': 8},
    'Firm_C': {'strategy': 'promotion', 'base_price': 10, 'promo_weeks': [3, 6], 'promo_price': 7}
}
```

```
# Extract model features (including 'const')
model_features = model.model.exog_names

# Simulate trend for 10 weeks
np.random.seed(42)
weeks = list(range(1, 11))
base_trend = np.linspace(0.4, 0.7, len(weeks)) + np.random.normal(0, 0.05, len(weeks))
trend_scores = MinMaxScaler().fit_transform(base_trend.reshape(-1, 1)).flatten()

# Init results and price memory
results = []
last_prices = {}

for week_idx, week in enumerate(weeks):
    trend_score = trend_scores[week_idx]

    for firm_name, firm in firms.items():
        base_price = firm['base_price']

        # Determine price based on strategy
        if firm['strategy'] == 'static':
            price = base_price
        elif firm['strategy'] == 'markdown':
            price = firm['markdown_price'] if week >= firm['markdown_week'] else base_price
        elif firm['strategy'] == 'promotion':
            price = firm['promo_price'] if week in firm['promo_weeks'] else base_price
        else:
            price = base_price

        # Use last week's price or fallback to base
        ref_price = base_price if week == 1 else last_prices.get(firm_name, base_price)
        price_change = price - ref_price

        # Build data dict with all required features from model
        data = {col: 0.0 for col in model_features}  # initialize all to 0
        data['const'] = 1.0
        data['PriceChange'] = price_change
        data['TrendScore'] = trend_score

        # Create DataFrame and predict
        X_sim = pd.DataFrame([data])[model_features]
        quantity = model.predict(X_sim)[0]
        revenue = price * quantity

        results.append({
```

```
            'Week': week,
            'Firm': firm_name,
            'Price': price,
            'RefPrice': ref_price,
            'TrendScore': trend_score,
            'PredictedQuantity': quantity,
            'Revenue': revenue
        })

        last_prices[firm_name] = price  # Save price for next week's reference

results_df = pd.DataFrame(results)
print(results_df.head())



for firm in firms:
    plt.plot(results_df[results_df['Firm'] == firm]['Week'],
             results_df[results_df['Firm'] == firm]['Revenue'],
             label=firm)

plt.xlabel("Week")
plt.ylabel("Revenue")
plt.title("Firm Revenues Over Time")
plt.legend()
plt.grid(True)
plt.show()

results_df.groupby("Firm")["Revenue"].sum()

results_df.groupby("Firm")["PredictedQuantity"].mean()



##Shared Demand Pool
def softmax(x):
    e_x = np.exp(x - np.max(x))  # for numerical stability
    return e_x / e_x.sum()

results = []
last_prices = {}
unit_cost = 5  # assumed cost

for week_idx, week in enumerate(weeks):
    trend_score = trend_scores[week_idx]
    firm_prices = {}
    firm_utils = {}
```

```
# First pass: compute each firm's price and utility
for firm_name, firm in firms.items():
    base_price = firm['base_price']
    if firm['strategy'] == 'static':
        price = base_price
    elif firm['strategy'] == 'markdown':
        price = firm['markdown_price'] if week >= firm['markdown_week'] else base_price
    elif firm['strategy'] == 'promotion':
        price = firm['promo_price'] if week in firm['promo_weeks'] else base_price
    else:
        price = base_price

    ref_price = base_price if week == 1 else last_prices.get(firm_name, base_price)
    price_change = price - ref_price

    # Utility function: lower price and higher trend = better
    alpha = 1.0
    beta = 1.0
    utility = -alpha * price + beta * trend_score

    firm_prices[firm_name] = {
        'price': price,
        'ref_price': ref_price,
        'price_change': price_change
    }
    firm_utils[firm_name] = utility

# Convert utilities to market shares using softmax
utilities = np.array(list(firm_utils.values()))
market_shares = softmax(utilities)
total_demand = 1500 + 500 * trend_score  # total market size for this week

for i, (firm_name, firm) in enumerate(firms.items()):
    share = market_shares[i]
    quantity = total_demand * share
    price = firm_prices[firm_name]['price']
    revenue = price * quantity
    profit = (price - unit_cost) * quantity

    results.append({
        'Week': week,
        'Firm': firm_name,
        'Price': price,
        'RefPrice': firm_prices[firm_name]['ref_price'],
```

```python
            'TrendScore': trend_score,
            'MarketShare': share,
            'PredictedQuantity': quantity,
            'Revenue': revenue,
            'Profit': profit
        })

        last_prices[firm_name] = price

results_df = pd.DataFrame(results)

# Plot revenue over time
for firm in firms:
    firm_data = results_df[results_df['Firm'] == firm]
    plt.plot(firm_data['Week'], firm_data['Revenue'], label=firm)

plt.title("Weekly Revenue (Shared Demand)")
plt.xlabel("Week")
plt.ylabel("Revenue")
plt.legend()
plt.grid(True)
plt.show()

# View summary
print(results_df.groupby("Firm")[["Revenue", "Profit", "PredictedQuantity"]].sum())



## Simulate trend shocks (viral spike, decay, noise)
np.random.seed(42)
weeks = list(range(1, 11))

# Base trend curve
trend = np.linspace(0.4, 0.7, len(weeks))

# Add viral spike at week 4, decay after week 7
shock = np.zeros(len(weeks))
shock[3] = 0.5  # viral spike (week 4)
shock[7:] = -0.3  # fading interest (week 8+)

# Add noise
noise = np.random.normal(0, 0.05, len(weeks))

# Combine everything
synthetic_trend = trend + shock + noise
synthetic_trend = np.clip(synthetic_trend, 0, 1)  # cap between 0 and 1
```

```
# Normalize to 0–1 range
trend_scores = MinMaxScaler().fit_transform(synthetic_trend.reshape(-1, 1)).flatten()

# Optional: plot to visualize
plt.plot(weeks, trend_scores, marker='o', linestyle='--', color='purple')
plt.title("Simulated Trend Score (with Shock)")
plt.xlabel("Week")
plt.ylabel("TrendScore")
plt.grid(True)
plt.show()

##Bayesian Trend Belief Update
def bayesian_trend_update(observed, prior_mean, prior_var, obs_var):
    # Compute posterior variance
    posterior_var = (prior_var * obs_var) / (prior_var + obs_var)

    # Compute posterior mean
    posterior_mean = (obs_var * prior_mean + prior_var * observed) / (prior_var + obs_var)

    return posterior_mean, posterior_var

# Set parameters
prior_mean = 0.5  # Initial belief
prior_var = 0.05  # Initial uncertainty
obs_var = 0.02    # Noise in observation

belief_means = []
belief_vars = []

for week_idx, obs in enumerate(trend_scores):
    posterior_mean, posterior_var = bayesian_trend_update(
        observed=obs,
        prior_mean=prior_mean,
        prior_var=prior_var,
        obs_var=obs_var
    )

    belief_means.append(posterior_mean)
    belief_vars.append(posterior_var)

    # Posterior becomes prior for next week
    prior_mean = posterior_mean
    prior_var = posterior_var
```

```
plt.figure(figsize=(10,5))
plt.plot(weeks, trend_scores, label="Observed TrendScore", linestyle='--', marker='o')
plt.plot(weeks, belief_means, label="Bayesian Belief", linestyle='-', marker='s')
plt.fill_between(weeks,
                 np.array(belief_means) - 1.96 * np.sqrt(belief_vars),
                 np.array(belief_means) + 1.96 * np.sqrt(belief_vars),
                 color='orange', alpha=0.3, label="95% Confidence")
plt.title("Bayesian Trend Belief Update Over Time")
plt.xlabel("Week")
plt.ylabel("TrendScore")
plt.legend()
plt.grid(True)
plt.show()

#TrendScore → Bayesian Belief in Shared Demand Simulation
# --- Step 1: Define Bayesian update function ---
def bayesian_trend_update(observed, prior_mean, prior_var, obs_var):
    posterior_var = (prior_var * obs_var) / (prior_var + obs_var)
    posterior_mean = (obs_var * prior_mean + prior_var * observed) / (prior_var + obs_var)
    return posterior_mean, posterior_var

# --- Step 2: Simulate raw trend signal with noise, spikes, and decay ---
np.random.seed(42)
weeks = list(range(1, 11))
base_trend = np.linspace(0.4, 0.7, len(weeks))
shock = np.zeros(len(weeks))
shock[3] = 0.5   # Viral spike (week 4)
shock[7:] = -0.3  # Trend decay (week 8+)
noise = np.random.normal(0, 0.05, len(weeks))
observed_trend = base_trend + shock + noise
observed_trend = np.clip(observed_trend, 0, 1)

# Normalize the observed trend to match earlier scales
trend_scores = MinMaxScaler().fit_transform(observed_trend.reshape(-1, 1)).flatten()

# --- Step 3: Apply Bayesian filtering to get smoothed trend belief ---
prior_mean = 0.5
prior_var = 0.05
obs_var = 0.02
belief_means = []
belief_vars = []

for obs in trend_scores:
    posterior_mean, posterior_var = bayesian_trend_update(obs, prior_mean, prior_var, obs_var)
    belief_means.append(posterior_mean)
```

```
    belief_vars.append(posterior_var)
    prior_mean, prior_var = posterior_mean, posterior_var

# --- Step 4: Shared Demand Simulation using Bayesian Belief ---
def softmax(x):
    e_x = np.exp(x - np.max(x))  # numerical stability
    return e_x / e_x.sum()

firms = {
    'Firm_A': {'strategy': 'static', 'base_price': 10},
    'Firm_B': {'strategy': 'markdown', 'base_price': 10, 'markdown_week': 5, 'markdown_price': 8},
    'Firm_C': {'strategy': 'promotion', 'base_price': 10, 'promo_weeks': [3, 6], 'promo_price': 7}
}

results = []
last_prices = {}
unit_cost = 5  # constant for all firms

for week_idx, week in enumerate(weeks):
    # Use Bayesian belief instead of raw TrendScore
    trend_score = belief_means[week_idx]

    firm_prices = {}
    firm_utils = {}

    # --- Determine pricing and utility for each firm ---
    for firm_name, firm in firms.items():
        base_price = firm['base_price']
        if firm['strategy'] == 'static':
            price = base_price
        elif firm['strategy'] == 'markdown':
            price = firm['markdown_price'] if week >= firm['markdown_week'] else base_price
        elif firm['strategy'] == 'promotion':
            price = firm['promo_price'] if week in firm['promo_weeks'] else base_price
        else:
            price = base_price

        ref_price = base_price if week == 1 else last_prices.get(firm_name, base_price)
        price_change = price - ref_price

        # Define utility function
        alpha = 1.0
        beta = 1.0
        utility = -alpha * price + beta * trend_score
```

```
        firm_prices[firm_name] = {
            'price': price,
            'ref_price': ref_price,
            'price_change': price_change
        }
        firm_utils[firm_name] = utility

    # --- Convert utilities into market shares ---
    utilities = np.array(list(firm_utils.values()))
    market_shares = softmax(utilities)
    total_demand = 1500 + 500 * trend_score

    for i, (firm_name, firm) in enumerate(firms.items()):
        share = market_shares[i]
        quantity = total_demand * share
        price = firm_prices[firm_name]['price']
        revenue = price * quantity
        profit = (price - unit_cost) * quantity

        results.append({
            'Week': week,
            'Firm': firm_name,
            'Price': price,
            'RefPrice': firm_prices[firm_name]['ref_price'],
            'TrendScore_Belief': trend_score,
            'MarketShare': share,
            'PredictedQuantity': quantity,
            'Revenue': revenue,
            'Profit': profit
        })

        last_prices[firm_name] = price

# --- Step 5: Results and Visualization ---
results_df = pd.DataFrame(results)

# Revenue plot
for firm in firms:
    firm_data = results_df[results_df['Firm'] == firm]
    plt.plot(firm_data['Week'], firm_data['Revenue'], label=firm)

plt.title("Weekly Revenue with Bayesian Belief Trend")
plt.xlabel("Week")
plt.ylabel("Revenue")
plt.legend()
```

```
plt.grid(True)
plt.show()

# Summary performance
print(results_df.groupby("Firm")[["Revenue", "Profit", "PredictedQuantity"]].sum())

##Reinforcement Learning
# --- Define environment parameters ---
price_levels = [7, 8, 9, 10]  # Actions
trend_bins = np.linspace(0, 1, 5)  # State bins for trend
episodes = 200  # Number of learning iterations
epsilon = 0.2  # Exploration rate
alpha = 0.1    # Learning rate
gamma = 0.9    # Discount factor

# --- Define other firms (fixed strategies) ---
firms_fixed = {
    'Firm_B': {'strategy': 'markdown', 'base_price': 10, 'markdown_week': 5, 'markdown_price': 8},
    'Firm_C': {'strategy': 'promotion', 'base_price': 10, 'promo_weeks': [3, 6], 'promo_price': 7}
}

# --- Initialize Q-table ---
Q = defaultdict(lambda: np.zeros(len(price_levels)))

# --- Helper functions ---
def get_state(trend_score, last_price):
    trend_bin = np.digitize(trend_score, trend_bins) - 1
    price_idx = price_levels.index(last_price)
    return (trend_bin, price_idx)

def softmax(x):
    e_x = np.exp(x - np.max(x))
    return e_x / e_x.sum()

# --- Simulated market loop ---
reward_history = []

for episode in range(episodes):
    last_price = 10  # Starting price for Firm A
    total_profit = 0
    prior_mean = 0.5
    prior_var = 0.05
    obs_var = 0.02

    for week in range(10):
```

```
# Bayesian belief update
observed = trend_scores[week]
trend_belief, prior_var = bayesian_trend_update(observed, prior_mean, prior_var, obs_var)
prior_mean = trend_belief

# RL state
state = get_state(trend_belief, last_price)

# Action: Choose a price
if np.random.rand() < epsilon:
    action_idx = np.random.choice(len(price_levels))
else:
    action_idx = np.argmax(Q[state])

firm_a_price = price_levels[action_idx]
price_change = firm_a_price - last_price
last_price = firm_a_price  # Update for next week

# Set firm prices
firm_prices = {
    'Firm_A': firm_a_price
}

for firm, data in firms_fixed.items():
    base_price = data['base_price']
    if data['strategy'] == 'markdown':
        firm_prices[firm] = data['markdown_price'] if week >= data['markdown_week'] else base_price
    elif data['strategy'] == 'promotion':
        firm_prices[firm] = data['promo_price'] if week in data['promo_weeks'] else base_price

# Compute utilities
firm_utils = {}
for firm, price in firm_prices.items():
    firm_utils[firm] = -price + trend_belief  # Simplified utility

market_shares = softmax(np.array(list(firm_utils.values())))
total_demand = 1500 + 500 * trend_belief
quantities = {firm: share * total_demand for firm, share in zip(firm_prices.keys(), market_shares)}

profit = (firm_a_price - 5) * quantities['Firm_A']
total_profit += profit

# Observe next state (we use same trend belief next step for simplicity here)
next_state = get_state(trend_belief, last_price)
```

```
        # Q-learning update
        Q[state][action_idx] += alpha * (profit + gamma * np.max(Q[next_state]) - Q[state][action_idx])

    reward_history.append(total_profit)

# --- Plot Learning Curve ---
plt.plot(reward_history)
plt.title("RL Agent (Firm A) Profit per Episode")
plt.xlabel("Episode")
plt.ylabel("Total Profit (10 weeks)")
plt.grid(True)
plt.show()

# Evaluate learned RL policy
final_results = []
last_price = 10
prior_mean = 0.5
prior_var = 0.05

for week in range(10):
    # Bayesian belief
    observed = trend_scores[week]
    trend_belief, prior_var = bayesian_trend_update(observed, prior_mean, prior_var, obs_var)
    prior_mean = trend_belief

    # RL action (greedy)
    state = get_state(trend_belief, last_price)
    action_idx = np.argmax(Q[state])
    firm_a_price = price_levels[action_idx]
    price_change = firm_a_price - last_price
    last_price = firm_a_price

    firm_prices = {'Firm_A': firm_a_price}
    for firm, data in firms_fixed.items():
        base_price = data['base_price']
        if data['strategy'] == 'markdown':
            firm_prices[firm] = data['markdown_price'] if week >= data['markdown_week'] else base_price
        elif data['strategy'] == 'promotion':
            firm_prices[firm] = data['promo_price'] if week in data['promo_weeks'] else base_price

    # Utilities and shares
    firm_utils = {firm: -p + trend_belief for firm, p in firm_prices.items()}
    market_shares = softmax(np.array(list(firm_utils.values())))
    total_demand = 1500 + 500 * trend_belief
    quantities = {firm: share * total_demand for firm, share in zip(firm_prices.keys(), market_shares)}
```

```
    for firm in firm_prices:
        price = firm_prices[firm]
        quantity = quantities[firm]
        revenue = price * quantity
        profit = (price - 5) * quantity
        final_results.append({
            'Week': week + 1,
            'Firm': firm,
            'Price': price,
            'TrendScore_Belief': trend_belief,
            'PredictedQuantity': quantity,
            'Revenue': revenue,
            'Profit': profit
        })

# Create DataFrame and summarize
final_df = pd.DataFrame(final_results)
summary = final_df.groupby("Firm")[["Revenue", "Profit", "PredictedQuantity"]].sum()
print(summary)

# Smooth learning curve
smoothed = pd.Series(reward_history).rolling(10).mean()
plt.plot(reward_history, alpha=0.3, label='Raw')
plt.plot(smoothed, label='Smoothed', color='blue')
plt.title("RL Learning Curve (Smoothed)")
plt.xlabel("Episode")
plt.ylabel("Total Profit (10 weeks)")
plt.legend()
plt.grid(True)
plt.show()



#Expanded State Space and Action = Strategy
# --- Define pricing strategy mapping ---
def get_price_from_strategy(strategy, week):
    if strategy == 0:  # Static
        return 10
    elif strategy == 1:  # Markdown
        return 8 if week >= 5 else 10
    elif strategy == 2:  # Promotion
        return 7 if week in [3, 6] else 10

# --- Binning helpers ---
price_bins = [7, 8, 9, 10]
```

```
trend_bins = np.linspace(0, 1, 5)
demand_bins = [0, 2000, 4000, 6000, 8000]

# --- New state function ---
def get_state(trend_score, last_price, ref_price, past_demand):
    t_bin = np.digitize(trend_score, trend_bins) - 1
    lp_bin = np.digitize(last_price, price_bins) - 1
    rp_bin = np.digitize(ref_price, price_bins) - 1
    d_bin = np.digitize(past_demand, demand_bins) - 1
    return (t_bin, lp_bin, rp_bin, d_bin)

# --- RL parameters ---
strategies = [0, 1, 2]  # 0 = Static, 1 = Markdown, 2 = Promo
Q = defaultdict(lambda: np.zeros(len(strategies)))

epsilon = 0.2
alpha = 0.1
gamma = 0.9
episodes = 200

reward_history = []

# --- Training loop ---
for episode in range(episodes):
    prior_mean = 0.5
    prior_var = 0.05
    last_price = 10
    ref_price = 10
    past_demand = 3000  # neutral starting point
    total_profit = 0

    for week in range(1, 11):
        # Trend belief update
        observed = trend_scores[week - 1]
        trend_belief, prior_var = bayesian_trend_update(observed, prior_mean, prior_var, 0.02)
        prior_mean = trend_belief

        # Get current state
        state = get_state(trend_belief, last_price, ref_price, past_demand)

        # Epsilon-greedy action
        if np.random.rand() < epsilon:
            action = np.random.choice(strategies)
        else:
            action = np.argmax(Q[state])
```

```python
        # Translate strategy to price
        firm_a_price = get_price_from_strategy(action, week)
        price_change = firm_a_price - ref_price

        # Fixed strategies for Firm B and C
        firm_prices = {
            'Firm_A': firm_a_price,
            'Firm_B': 8 if week >= 5 else 10,
            'Firm_C': 7 if week in [3, 6] else 10
        }

        firm_utils = {f: -p + trend_belief for f, p in firm_prices.items()}
        market_shares = softmax(np.array(list(firm_utils.values())))
        total_demand = 1500 + 500 * trend_belief
        quantities = {f: s * total_demand for f, s in zip(firm_prices, market_shares)}

        # Compute reward
        profit = (firm_a_price - 5) * quantities['Firm_A']
        total_profit += profit

        # Next state
        next_state = get_state(trend_belief, firm_a_price, last_price, quantities['Firm_A'])

        # Q-learning update
        Q[state][action] += alpha * (profit + gamma * np.max(Q[next_state]) - Q[state][action])

        # Update memory
        ref_price = last_price
        last_price = firm_a_price
        past_demand = quantities['Firm_A']

    reward_history.append(total_profit)

# --- Evaluation ---
final_results = []
prior_mean = 0.5
prior_var = 0.05
last_price = 10
ref_price = 10
past_demand = 3000

for week in range(1, 11):
    observed = trend_scores[week - 1]
    trend_belief, prior_var = bayesian_trend_update(observed, prior_mean, prior_var, 0.02)
```

```
    prior_mean = trend_belief

    state = get_state(trend_belief, last_price, ref_price, past_demand)
    strategy = np.argmax(Q[state])
    firm_a_price = get_price_from_strategy(strategy, week)
    price_change = firm_a_price - ref_price

    firm_prices = {
        'Firm_A': firm_a_price,
        'Firm_B': 8 if week >= 5 else 10,
        'Firm_C': 7 if week in [3, 6] else 10
    }

    firm_utils = {f: -p + trend_belief for f, p in firm_prices.items()}
    market_shares = softmax(np.array(list(firm_utils.values())))
    total_demand = 1500 + 500 * trend_belief
    quantities = {f: s * total_demand for f, s in zip(firm_prices, market_shares)}

    for f in firm_prices:
        p = firm_prices[f]
        q = quantities[f]
        final_results.append({
            'Week': week,
            'Firm': f,
            'Price': p,
            'Quantity': q,
            'Revenue': p * q,
            'Profit': (p - 5) * q
        })

    ref_price = last_price
    last_price = firm_a_price
    past_demand = quantities['Firm_A']

# --- Results ---
df_eval = pd.DataFrame(final_results)
print(df_eval.groupby("Firm")[["Revenue", "Profit", "Quantity"]].sum())

#Robustness Checks
# Monte Carlo
def run_single_simulation(trend_scores, weeks, num_firms, base_price=10, unit_cost=5):
    results = []
    for week, trend_score in zip(weeks, trend_scores):
        for firm_id in range(num_firms):
            strategy = random.choice(['markdown', 'promotion', 'none'])
```

```
        price = base_price
        if strategy == 'markdown':
            price *= 0.8
        elif strategy == 'promotion':
            price *= 0.9
        demand = (trend_score * 100) * np.exp(-price / 10)
        revenue = price * demand
        profit = (price - unit_cost) * demand
        results.append({
            'Week': week,
            'FirmID': firm_id,
            'Strategy': strategy,
            'Price': price,
            'UnitsSold': demand,
            'Revenue': revenue,
            'Profit': profit,
            'TrendScore': trend_score
        })
    df = pd.DataFrame(results)
    kpi = df.groupby('Strategy')[['Revenue', 'Profit', 'UnitsSold']].sum().reset_index()
    kpi['ROI'] = kpi['Profit'] / kpi['Revenue']
    kpi['Run'] = None  # Will be added later
    return kpi

def run_monte_carlo_simulation(trend_scores, weeks, num_firms=3, runs=100):
    all_kpis = []
    for i in range(runs):
        kpi_df = run_single_simulation(trend_scores, weeks, num_firms)
        kpi_df['Run'] = i + 1
        all_kpis.append(kpi_df)
    combined_kpis = pd.concat(all_kpis, ignore_index=True)
    summary = combined_kpis.groupby('Strategy').agg({
        'Revenue': ['mean', 'std'],
        'Profit': ['mean', 'std'],
        'UnitsSold': ['mean', 'std'],
        'ROI': ['mean', 'std']
    })
    summary.columns = ['_'.join(col).strip() for col in summary.columns.values]
    return summary.round(2), combined_kpis

summary_stats, all_runs_df = run_monte_carlo_simulation(trend_scores, weeks, num_firms=3, runs=100)

print("\n=== Monte Carlo KPI Summary (100 runs) ===")
print(summary_stats)
```

```
# Optional: Save to CSV
summary_stats.to_csv("monte_carlo_summary.csv")
all_runs_df.to_csv("monte_carlo_all_runs.csv", index=False)


#Confidence interval
def add_confidence_intervals(df, metric, group='Strategy'):
    grouped = df.groupby(group)[metric]
    mean = grouped.mean()
    sem = grouped.sem()
    ci95 = sem * stats.t.ppf((1 + 0.95) / 2., grouped.count() - 1)
    return pd.DataFrame({
        'mean': mean,
        'ci95': ci95
    }).round(2)

all_runs_df['Revenue_per_Unit'] = all_runs_df['Revenue'] / all_runs_df['UnitsSold']
top_strat_freq = all_runs_df.groupby(['Run']).apply(
    lambda x: x.loc[x['Revenue'].idxmax(), 'Strategy']
).value_counts()
print(top_strat_freq)


# Vary Price elasticity
def run_elasticity_sensitivity(trend_scores, weeks, num_firms=3, k_values=[6, 8, 10, 12, 14], runs_per_k=50):
    all_results = []

    for k in k_values:
        print(f"Running elasticity simulation for k = {k}...")
        for run in range(runs_per_k):
            results = []
            for week, trend_score in zip(weeks, trend_scores):
                for firm_id in range(num_firms):
                    strategy = random.choice(['markdown', 'promotion', 'none'])

                    price = base_price
                    if strategy == 'markdown':
                        price *= 0.8
                    elif strategy == 'promotion':
                        price *= 0.9

                    # Demand based on current elasticity value
                    demand = (trend_score * 100) * np.exp(-price / k)
                    revenue = price * demand
                    profit = (price - 5) * demand
```

```
                results.append({
                    'ElasticityK': k,
                    'Run': run + 1,
                    'Week': week,
                    'FirmID': firm_id,
                    'Strategy': strategy,
                    'Price': price,
                    'UnitsSold': demand,
                    'Revenue': revenue,
                    'Profit': profit,
                    'TrendScore': trend_score
                })

            df = pd.DataFrame(results)
            kpi = df.groupby('Strategy')[['Revenue', 'Profit', 'UnitsSold']].sum().reset_index()
            kpi['ROI'] = kpi['Profit'] / kpi['Revenue']
            kpi['ElasticityK'] = k
            kpi['Run'] = run + 1
            all_results.append(kpi)

    final_df = pd.concat(all_results, ignore_index=True)
    summary_df = final_df.groupby(['ElasticityK', 'Strategy']).agg({
        'Revenue': ['mean', 'std'],
        'Profit': ['mean', 'std'],
        'UnitsSold': ['mean', 'std'],
        'ROI': ['mean', 'std']
    }).round(2)

    summary_df.columns = ['_'.join(col).strip() for col in summary_df.columns.values]
    return summary_df, final_df

elasticity_summary, elasticity_all_runs = run_elasticity_sensitivity(
    trend_scores,
    weeks,
    num_firms=3,
    k_values=[6, 8, 10, 12, 14],
    runs_per_k=50
)

# View summary
print("\n===  Price Elasticity Sensitivity Summary ===")
print(elasticity_summary)

# Save if needed
```

```
elasticity_summary.to_csv("elasticity_kpi_summary.csv")
elasticity_all_runs.to_csv("elasticity_all_runs.csv", index=False)

# Flatten index
elasticity_plot = elasticity_summary.reset_index()

# Revenue Plot
plt.figure(figsize=(10, 5))
sns.lineplot(data=elasticity_plot, x='ElasticityK', y='Revenue_mean', hue='Strategy', marker='o')
plt.title("Revenue vs Price Elasticity (k)")
plt.grid(True)
plt.show()

# Profit Plot
plt.figure(figsize=(10, 5))
sns.lineplot(data=elasticity_plot, x='ElasticityK', y='Profit_mean', hue='Strategy', marker='s')
plt.title("Profit vs Price Elasticity (k)")
plt.grid(True)
plt.show()

# ROI Plot
plt.figure(figsize=(10, 5))
sns.lineplot(data=elasticity_plot, x='ElasticityK', y='ROI_mean', hue='Strategy', marker='d')
plt.title("ROI vs Price Elasticity (k)")
plt.grid(True)
plt.show()

# Baseline at k = 10
baseline = elasticity_summary.loc[10]

relative_change = (
    elasticity_summary
    .reset_index()
    .merge(baseline.reset_index(), on='Strategy', suffixes=('', '_baseline'))
)

relative_change['Revenue_pct_change'] = (
    (relative_change['Revenue_mean'] - relative_change['Revenue_mean_baseline'])
    / relative_change['Revenue_mean_baseline']
) * 100


sns.set(style="whitegrid")
plt.figure(figsize=(10, 5))
sns.lineplot(
```

```
    data=elasticity_all_runs,
    x='ElasticityK',
    y='Revenue',
    hue='Strategy',
    estimator='mean',
    errorbar='sd',
    marker='o'
)
plt.title("Revenue vs Elasticity with Std Dev")
plt.grid(True)
plt.tight_layout()
plt.show()

best_by_k = elasticity_all_runs.groupby(['ElasticityK', 'Run']).apply(
    lambda x: x.loc[x['Revenue'].idxmax(), 'Strategy']
).reset_index(name='BestStrategy')

# Count how often each strategy wins
print(best_by_k.groupby(['ElasticityK', 'BestStrategy']).size().unstack(fill_value=0))

#Consumer Conversion check
# --- Consumer Conversion Rate in Shared Demand Simulation ---

# Assuming 'results_df' contains shared demand simulation results:
# If you're rerunning the simulation, define this first
results = []
last_prices = {}
unit_cost = 5
weeks = list(range(1, 11))  # adjust if using a different week range

# Sample trend (or reuse trend_scores from your earlier simulation)
trend_scores = MinMaxScaler().fit_transform(np.linspace(0.4, 0.7, len(weeks)).reshape(-1, 1)).flatten()

for week_idx, week in enumerate(weeks):
    trend_score = trend_scores[week_idx]

    firm_prices = {}
    firm_utils = {}
    for firm_name, firm in firms.items():
        base_price = firm['base_price']
        if firm['strategy'] == 'static':
            price = base_price
        elif firm['strategy'] == 'markdown':
            price = firm['markdown_price'] if week >= firm['markdown_week'] else base_price
        elif firm['strategy'] == 'promotion':
```

```
        price = firm['promo_price'] if week in firm['promo_weeks'] else base_price
    else:
        price = base_price

    ref_price = base_price if week == 1 else last_prices.get(firm_name, base_price)
    price_change = price - ref_price

    alpha = 1.0
    beta = 1.0
    utility = -alpha * price + beta * trend_score

    firm_prices[firm_name] = {
        'price': price,
        'ref_price': ref_price,
        'price_change': price_change
    }
    firm_utils[firm_name] = utility

utilities = np.array(list(firm_utils.values()))
market_shares = softmax(utilities)
total_demand = 1500 + 500 * trend_score

for i, (firm_name, firm) in enumerate(firms.items()):
    share = market_shares[i]
    quantity = total_demand * share
    price = firm_prices[firm_name]['price']
    revenue = price * quantity
    profit = (price - unit_cost) * quantity

    # --- NEW: Calculate Conversion Rate ---
    conversion_rate = quantity / total_demand

    results.append({
        'Week': week,
        'Firm': firm_name,
        'Price': price,
        'RefPrice': firm_prices[firm_name]['ref_price'],
        'TrendScore': trend_score,
        'MarketShare': share,
        'PredictedQuantity': quantity,
        'Revenue': revenue,
        'Profit': profit,
        'ConversionRate': conversion_rate  # New metric
    })
```

```python
        last_prices[firm_name] = price

# Convert results to DataFrame
results_df = pd.DataFrame(results)
results_df

# --- Summary: Average Conversion Rate per Firm ---
conversion_summary = results_df.groupby("Firm")["ConversionRate"].mean().round(4)
print("\nAverage Conversion Rate by Firm:")
print(conversion_summary)

# --- Optional: Plot Conversion Rate Over Time ---
plt.figure(figsize=(8, 5))
for firm in results_df['Firm'].unique():
    firm_data = results_df[results_df['Firm'] == firm]
    plt.plot(firm_data['Week'], firm_data['ConversionRate'], marker='o', label=firm)

plt.title("Conversion Rate Over Time by Firm")
plt.xlabel("Week")
plt.ylabel("Conversion Rate")
plt.ylim(0, 1)
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Appendix B: Supplementary Materials

The complete Python simulation codebase, including data preprocessing, econometric estimation, reinforcement learning modules, and robustness checks, is available on GitHub for replication and further research.
https://github.com/RishikaAgarwal2025/dissertation-dynamic-pricing

Appendix C: Research Ethnic form

Attaching the research ethnic form below

Student details

Please enter your details below very carefully, as we will use this information to notify you of your Research Ethics Form outcome.

Please make sure you only use your Warwick student email address e.g. Firstname.Surname@warwick.ac.uk

Student 1 Full name	Rishika Mohit Agarwal
Student 1 Email address (@warwick.ac.uk)	rishika.agarwal@warwick.ac.uk
Student 1 ID Number (without "u")	5635111
Student 2 Full name	N/A
Student 2 Email address (@warwick.ac.uk)	N/A
Student 2 ID Number (without "u")	N/A
Student 3 Full name	N/A
Student 3 Email address (@warwick.ac.uk)	N/A
Student 3 ID Number (without "u")	N/A

Q10.

Supervisor details

You will find this information on the supervisor allocation email from your Programme Team.

Please make sure that you enter this correctly, as we will use this to request your Research Ethics Approval. Please do not submit this form if you have not been allocated a supervisor yet.

What is the name of your allocated supervisor?

Bo Chen

Q11. Please enter your supervisor's email address.

You will find this information on the supervisor allocation email from your Programme Team. Please make sure that you double-check and enter this correctly, as we will use this to request your Research Ethics Approval. If you enter this email address incorrectly you may need to submit the form again.

bo.chen@wbs.ac.uk

Q12. What is the current title for your research?

Dynamic pricing in a competitive market with shared demand influenced by social media trends: Strategic role of markdowns vs. temporary promotions

Q13. What are your research aims?

To model dynamic pricing in competitive markets where demand is shared and influenced by social media. To compare the effectiveness of markdown pricing versus temporary promotions.

To integrate Bayesian learning and reinforcement learning into pricing strategy to handle uncertainty and competition in a trend-driven environment.

Q90. Please tick all of the research methods below that you plan to utilise:

- Secondary data analysis (previously existing datasets) • Literature-based research or documentary analysis

Q91. Which type of secondary data will you be using in your research? Please tick all options which are relevant:

- Historical records that do not contain individual-level data
- Previously existing datasets where individual level data is provided but individuals cannot be identified

Q17. Your research may be eligible for a REF waiver.

Please tick to confirm that your research will not include any of the following:

- Questionnaires or surveys • Interviews or oral histories
- Laboratory or field experiments
- Analysis or use of any kind of social media • Ethnography or observation
- Collection of individual-level information relating to human subjects (including, in some circumstances, deceased human subjects)
- Any other methodology that involves live human participants or their data

Q39.

Data Analysis

Please specify the data analysis method(s) for your research design. Please explain:

- Which analysis methods you will use for your research? E.g. content analysis, framework analysis, interpretative phenomenological analysis etc. and any statistical analyses.
- Will you be using any software for the analysis, and how will it be used?

Analysis Methods

Simulation-Based Research Design:

You are using a simulation approach to model complex market behavior under competition and social media influence.

Agent-Based Modelling (ABM):

To simulate the interactions between multiple firms and consumers in a digital market. Helps capture heterogeneous consumer responses and competitive dynamics.

Reinforcement Learning (RL):

For learning adaptive pricing strategies over time.

The RL agent takes actions (e.g., no discount, markdown, or temporary promotion) based on states (trend signals, competitor prices, past demand), and learns by maximizing cumulative rewards (profits).

Bayesian Modelling:

To handle uncertainty in demand caused by noisy and rapidly changing social trends.

Prior beliefs about trend effectiveness are updated with new data using Bayesian inference. RL agents' sample from these posterior distributions to make informed pricing decisions.

Statistical Evaluation and Robustness Checks:

Use of evaluation metrics such as profit, market share, conversion rates, and ROI to compare the effectiveness of markdowns vs. temporary promotions.

Robustness checks include simulating false trends, demand shocks, and variations in prior beliefs about trend strength.

Software Used - Python

Q40.

Student declaration for Ethical Approval.

In order to ensure that you have accessed the relevant information and guidance on research integrity, and that the information you have provided is accurate, please tick to confirm that you agree with the statements below:

- The information in this form together with any accompanying information is complete and correct to the best of my knowledge and belief, and I take full responsibility for it.
- I have accessed the necessary training about research ethics and have consulted my supervisor to confirm that the data sources I intend to access are acceptable.
- I undertake to abide by the University of Warwick's Research Code of Practice and other relevant professional and University policies, regulations, procedures and guidelines in undertaking this research.
- I confirm I am familiar with and will conduct my research in line with the University's Data Protection Policy including General Data Protection Regulation (GDPR) and Data Protection Act 2018 (DPA 2018). I will report any data breaches to the University's Information and Data Director: dpo@warwick.ac.uk
- I understand that I must not begin any research until I have received ethical approval from the supervisor to proceed.
- I understand that any changes that I would like to make to this study after receiving approval from WBS, require further review. As such they must be submitted via a new Research Ethics Form before such changes are implemented.

Supervisor outcome

Q41.

Please select which outcome you are providing for this Research Ethics Approval form.

All supervisors must grant or waive ethical approval for their student's research project BEFORE the student carries out data collection.

- Research Ethics Form waiver approved - ethical approval not required

Q52. Please confirm that you have seen and checked that the following documents have been submitted. You can view these from the response summary included in the email sent requesting your review.

Research Integrity Epigeum Training Certificate (all students)	Checked

Q88. Please sign to confirm that you believe that the student (or SCP group) has fully considered the ethical dimensions of their proposed research, and that the project is eligible for a Research Ethics Form Waiver, which you are granting. This means that research ethics approval is not required and you are happy for the

student to continue with the research.