ISSN: 2473-4901
v12 n6725

Integrating Synthetic Data, Decision Trees, and LLM-Based Explanation for Retail Micro-Decision Support

Parul Yadav
pyadav@memphis.edu
University of Memphis
Memphis, TN, 38152, USA

Ice Asortse
inast139@mail.rmu.edu
Robert Morris University
Pittsburgh, PA, 15108, USA

Natalya Bromall
bromall@rmu.edu
Robert Morris University
Pittsburgh, PA, 15108, USA

Abstract

Retail organizations increasingly depend on automated systems to support micro-level operational decisions, including inventory replenishment, promotional timing, and price adjustments. Although individual decisions may appear routine, their cumulative effect shapes significant financial and logistical outcomes. This paper introduces an interpretable hybrid agentic AI architecture, HA3-RMDS, that integrates synthetic data generation, a transparent decision tree classifier, and a lightweight Large Language Model (LLM) explanation layer. HA3-RMDS, implemented in Python with a fixed random seed for full reproducibility, generates 2,000 labeled product-state records, trains a shallow decision tree (max_depth = 4) on an 80% training split, and evaluates it on a 20% test set. The decision tree achieves 95.75% test accuracy with macro-average F1 of 0.9586 across three decision classes (reorder, promo, no_action), with a 4.25% gap caused by boundary-condition misclassification at the constrained tree depth. The LLM explanation layer, using TinyLlama-1.1B-Chat with a structured two-part prompt, produces 189-word natural-language explanations achieving 6/6 keyword-faithfulness. HA3-RMDS serves as an interpretable decision-support system that achieves transparency, auditability, and operational alignment in retail micro-decision workflows. Critically, evaluation of the LLM explanation layer reveals that a 6/6 keyword-faithfulness score can conceal systematic reasoning failures: the model may name the correct action while fabricating decision rules, mischaracterizing trigger conditions, and recommending operationally incompatible actions — demonstrating that surface-level faithfulness metrics are insufficient for trustworthy AI explanation.

Keywords: agentic AI, interpretable machine learning, decision trees, synthetic data, retail decision-making, large language models, TinyLlama

Integrating Synthetic Data, Decision Trees, and LLM-Based Explanation for Retail Micro-Decision Support

Parul Yadav, Ice Asortse and Natalya Bromall

1. INTRODUCTION

Retail operations depend on a continuous flow of micro-decisions that collectively determine how products move through the supply chain and how customers experience pricing, availability, and service quality. Decisions such as reordering inventory, discounting products, adjusting prices, or maintaining current stock levels may appear minor in isolation, but together they drive profitability, customer satisfaction, and operational efficiency (Ma et al., 2022). As retailers expand product ranges and distribution networks, the volume and speed of these decisions exceed the capacity of human oversight alone. Algorithmic systems now play a pivotal role in retail workflows, and the growing use of automated decision-making has made transparency, interpretability, and alignment with business goals more critical than ever.

Concerns regarding opaque or misaligned AI systems are well documented. Hendrycks et al. (2023) warn that modern AI models may introduce failure modes that are difficult to predict, especially in fast-changing, high-stakes environments. Hase and Bansal (2020) similarly emphasize that explanations should accurately reflect how models function rather than offering post-hoc rationalizations. These concerns are especially relevant in retail, where misaligned decision logic can affect thousands of products across many locations, resulting in persistent stockouts, excess inventory, or distorted pricing signals.

Recent advances in synthetic data generation, interpretable machine learning, and lightweight agentic systems have opened new opportunities to design AI systems that are both operationally effective and transparent. Synthetic data (Nikolaidis & Ramakrishnan, 2023), which are artificially generated datasets that replicate the statistical properties of real operational data, has become increasingly valuable in retail contexts. It enables controlled experimentation, safeguards sensitive commercial information, and facilitates the modeling of rare or extreme scenarios. Despite these advances, retail AI systems rarely combine interpretability, reproducible synthetic data, quantitative evaluation, and LLM-based explanation in a single coherent architecture. This paper addresses that gap by presenting a fully evaluated hybrid agentic AI architecture for retail micro-decision support.

To support this architecture, the system integrates a lightweight Large Language Model (LLM) that generates natural language explanations based on the model's actual decision path, rather than creating new decisions. The training is grounded in a rule-based ground truth, which ensures that the system learns from explicit and auditable retail heuristics instead of relying on opaque statistical correlations. A transparent decision tree classifier implements these rules, offering an interpretable structure whose conditional splits align with human retail reasoning. The explanation layer is subsequently evaluated for faithfulness to confirm that generated explanations accurately represent the internal logic of the decision tree, avoiding unsupported or post hoc narratives.

The following research question motivates the work:

RQ: How can an interpretable agentic AI architecture combine synthetic data, transparent decision logic, and LLM-based explanation to support safe, aligned, and operationally meaningful retail micro-decision making?

2. BACKGROUND AND RELATED WORK

There is an intensified interest in AI safety, interpretability, synthetic data, and agentic architecture as the retail industry embraces automated systems. These areas collectively shape how systems are designed to work reliably in changing business settings while maintaining transparency and meeting business goals.

2.1 AI Safety and Alignment

AI safety research emphasizes that systems should act predictably and remain aligned with human intentions, even when encountering novel inputs. Ngo et al. (2022) frame alignment as ensuring AI systems pursue intended objectives rather than optimizing for unintended proxies. Hendrycks et al. (2023) further argue that interpretability and human oversight are essential mechanisms for preventing harmful outcomes. In retail contexts, misalignment does not pose existential risks, but misaligned micro-decisions can produce measurable operational failures — persistent stockouts, excess inventory, or mispriced products. Transparency, auditability, and predictable behavior are therefore practical design requirements, not merely theoretical ideals.

2.2 Interpretability and Explainability

Interpretability is widely recognized as essential for deploying machine learning in operational settings requiring human oversight. Hase and Bansal (2020) highlight that explanations must genuinely reflect a model's internal reasoning rather than offering post-hoc rationalizations that may be misleading. Similarly, Chuang et al. (2024) emphasize that faithfulness — defined in the AI explanation literature as the degree to which an explanation accurately reflects the model's actual internal reasoning rather than producing plausible-sounding but unsupported justifications — is a crucial property for trustworthy AI systems, underscoring that explanation quality should be judged by alignment with the model's actual decision pathway rather than surface-level fluency or persuasiveness."

Bommasani et al. (2023) further argue that transparency is foundational to responsible AI governance, particularly when decisions must be audited or communicated to non-technical stakeholders. Decision trees, with their explicit rule-based structure, are well-suited to contexts requiring human oversight. Their logic is directly inspectable, enabling retail managers to understand precisely why the system recommends reordering, discounting, or maintaining current inventory levels.

2.3 Synthetic Data for Machine Learning

Synthetic data has emerged as a valuable tool in model development, particularly when real-world data is scarce, sensitive, or unavailable. Nikolaidis and Ramakrishnan (2023) explain that synthetic datasets are scalable, privacy-preserving, and enable controlled experimentation without exposing confidential business data. Jordon et al. (2022) highlight their value for scenario testing, including rare edge cases underrepresented in historical data. Similarly, van Breugel et al. (2023) show that synthetic test datasets can reveal model weaknesses that real-world data cannot, noting that their generated samples "expose failure modes that remain hidden in standard evaluation datasets." Researchers demonstrated effective use of synthetic data in a variety of cases. For example, Quian et al. (2024) generated a synthetic dataset based on the patients' data in the UK Biobank to avoid using the real data for confidentiality reasons. Their research demonstrated that synthetic data can be used to effectively predict lung cancer with machine learning (ML) models. Pereira et al. (2024) employed a synthetic criminal offenders' dataset to predict the probability of re-offence and found that their model could be effectively trained and deployed without the original data. Similarly, in this case, using the real data for training was not desirable due to confidentiality issues. While multiple research studies present the benefits of synthetic data in various domains (healthcare, law enforcement, consumer data, etc.), there is a lack of studies related specifically to retail. In retail analytics, synthetic data allows modeling of product states, demand patterns, and decision rules without requiring access to proprietary inventory systems.

2.4 Retail Analytics and Inventory Optimization

Machine learning is broadly applied across retail forecasting, pricing, and inventory management. However, interpretability remains a significant barrier to adoption in operational settings. Ma et al. (2022) note that retail managers are hesitant to rely on models whose logic they cannot validate or challenge. Li and Chan (2023) observe that rule-based heuristics remain common in inventory control precisely because they are transparent, stable, and auditable. The business rules used in this study follow classical inventory heuristics, aligning machine-learned logic with established retail practice and ensuring outputs remain operationally meaningful.

2.5 Agentic AI and Multi-Agent Systems

Recent progress in agentic AI explores how LLM-based agents can reason, plan, and communicate autonomously. Park et al. (2023) introduce generative agents simulating human-like behavior in complex interactive environments. Wang et al. (2024) evaluate LLMs across diverse agentic tasks, demonstrating their capacity for structured reasoning and multi-step decision execution. The architecture developed in this study operates as a micro-agent: it observes product feature vectors, applies learned decision rules, and communicates its reasoning through a lightweight LLM explanation layer. This design integrates observation, decision, and explanation in a unified, auditable workflow. The system does not exhibit full agentic autonomy — there is no goal persistence, environmental feedback loop, or self-correction across episodes — but fulfills the core observe-reason-explain cycle characteristic of micro-agent design (Park et al., 2023; Wang et al., 2024).

While GPTree (Xiong et al., 2024) and Koreeda and Manning (2023) combine decision trees with LLM explanation, neither targets retail operational decision-making nor evaluates deployability constraints. Agentic retail AI systems (Akarma et al., 2025) prioritize automation over interpretability and lack explanation layers. Human-AI hybrid frameworks (Passerini et al., 2025) address collaboration but do not integrate interpretable symbolic models with natural language explanation in a unified pipeline. To the best of the authors' knowledge, no prior work has combined synthetic data generation, a rule-aligned interpretable decision tree, and a locally deployable LLM explanation agent within a single auditable decision-support architecture specifically designed for retail micro-decision support.

3. METHODOLOGY

This paper proposes a hybrid agentic AI architecture, HA3-RMDS (**H**ybrid **A**gentic **A**I **A**rchitecture for **R**etail **M**icro **D**ecision **S**upport), that unifies synthetic data generation, interpretable machine learning (decision tree classifier), and LLM-based explanation to support micro decisions for retail operations. It comprises five integrated phases/ components: (1) synthetic data generation, (2) business rule establishment, (3) decision tree training and evaluation, (4) LLM explanation generation, and (5) faithfulness evaluation. Algorithm 1 presents the complete pipeline of these phases/ components of HA3-RMDS in pseudocode. Figure 1 presents the overall integration of HA3-RMDS's components. The detailed description of these components is given in the following subsections:

3.1 Synthetic Data Generation

Synthetic data generation is the first phase of HA3-RMDS that sets up a controlled environment for learning and evaluation. Since real retail data is often private, sensitive, or incomplete, the system uses synthetic data to simulate product state conditions important in making decisions. Each product has three attributes: Stock level, Reorder threshold, and Margin band.

These variables follow common retail heuristics: low stock triggers reorder, mid-range margins suggest running promotions, and high margins support keeping prices steady. In this system, synthetic data is created by randomly choosing values within a realistic range: stock (integer in [0, 25] representing units on hand), threshold (integer in [5, 15] representing the reorder point), and margin (float in [1.5, 3.5] representing the gross margin ratio). A total of 2,000 records are generated with a fixed seed point.

3.2 Business Rule Establishment

The business rule is defined, and the ground-truth labeling function implements a deterministic three-class business rule that assigns one of three operational decisions to each generated product state. Rule priority is applied sequentially:

(1) REORDER: if stock <= threshold, regardless of margin. Stock replenishment takes priority over any pricing action.

(2) PROMO: if stock > threshold AND 1.5 <= margin < 2.5. The margin band [1.5, 2.5) is grounded in classical retail markdown logic: margins below 1.5 are too thin to absorb a promotional discount without negative profitability impact; margins at or above 2.5 are healthy enough that discounting is

unnecessary. This band identifies products where a controlled promotion can stimulate demand without critically eroding margin, consistent with heuristics described in Li and Chan (2023).

(3) NO_ACTION: all remaining cases where stock exceeds the threshold and margin is at or above 2.5.

This complement assigns a ground-truth label to each record, which is used further for training and evaluation.

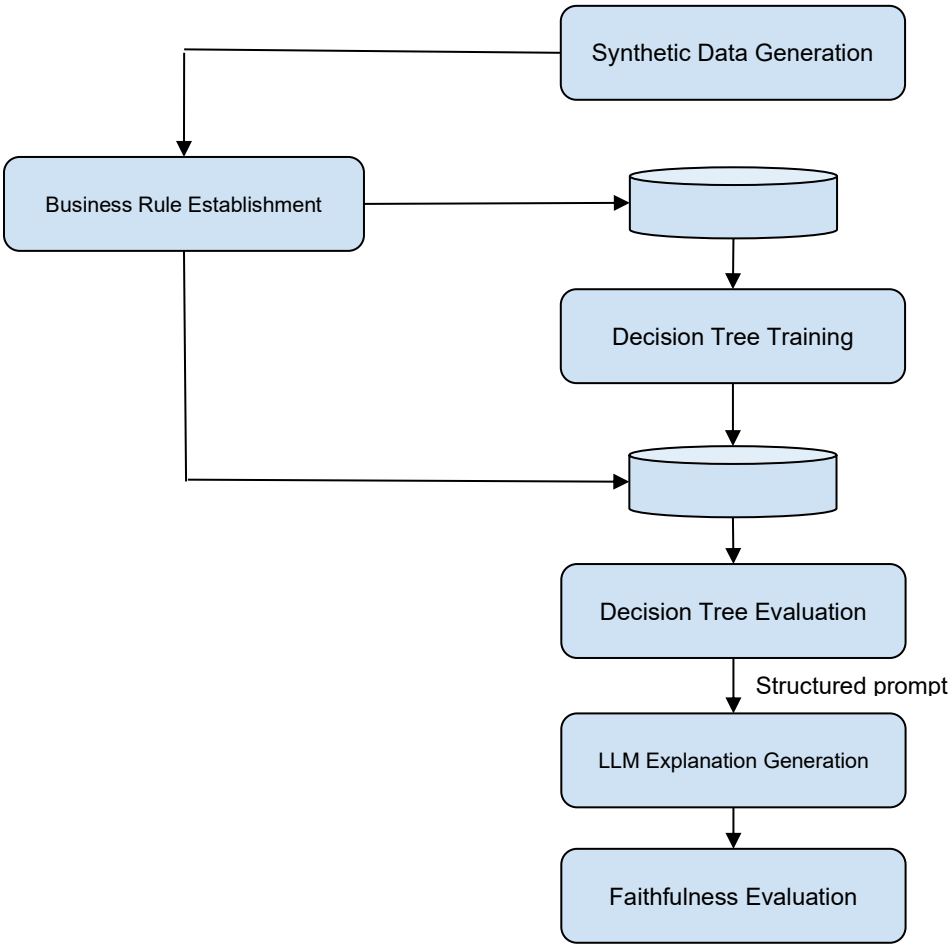

Figure 1. Hybrid Agentic AI architecture for Retail Micro Decision Support (HA3-RMDS)

3.3 Decision Tree Training and Evaluation

The third phase of HA3-RMDS trains a decision tree classifier to learn the company's business rules from the generated synthetic data. Decision trees are chosen because they provide a transparent and rule-based structure, support auditing and visualization, and align with retail managers' expectations for explainability.

The dataset is split 80/20 into training (1,600 samples) and test (400 samples) sets using stratified sampling to preserve class proportions in both splits. The model is trained using 1,600 synthetic samples (the 80% training split of the 2,000 samples). Setting a shallow depth (max depth = 4) keeps the learned logic easy to understand and helps prevent overfitting. The resulting tree approximates the company's business rules and preserves transparency.

The trained model is evaluated on the 400-sample test set. Evaluation metrics include overall accuracy, precision, recall, and F1-score. Feature importances from the fitted tree are reported to confirm that the

model learned decision-relevant structure. For each new product, the model predicts one of the three actions: reorder, promo, and no_action. The output is compiled into a decision table, which becomes the input to the explanation phase. Because the test labels are generated by this same rule, the theoretical maximum achievable accuracy is 100%. Any gap between this upper bound and the tree's test accuracy reflects cases where the depth-constrained tree fails to replicate the rule's boundary conditions exactly. This design is intentional: the system serves as a proof-of-concept demonstrating that a machine learning model can recover auditable, human-interpretable rules from data — a stepping stone toward real-world deployments where ground-truth rules are unknown and the tree must generalize from noisy, proprietary retail data rather than from a deterministic labeling function.

3.4 LLM Explanation Generation

The fourth phase adds a lightweight LLM explanation agent that reads the decision table and creates easy-to-understand explanations. This part uses a small, fast model (TinyLlama-1.1B-Chat-v1.0) to ensure efficiency and deployability in an operational setting. The LLM agent receives a two-part structured prompt. The first part establishes the agent's role, instructing the model/ architecture to act as a knowledgeable retail operations analyst and to provide clear, specific, data-driven explanations that reference the actual feature values in the decision table. The second part supplies the decision context in three components: the three business decision rules (reorder, promo, and no_action) with their feature conditions as explicit reference text; the decision table containing each product's stock level, reorder threshold, margin, and recommended action; and three instruction directives requiring the model to justify each decision based on the product's actual feature values, identify operational risks such as stock levels close to the reorder threshold or low-margin products, and provide one practical inventory strategy recommendation based on patterns observed across all products.

3.5 Faithfulness Evaluation

In the fifth phase, a keyword-presence faithfulness check evaluates whether the LLM explanation correctly references the decision class for each product. For each product, the explanation is checked for the presence of synonyms associated with the correct decision class: {'reorder'}, {'promo', 'promotion', 'promotional', 'discount'}, or {'no_action', 'no action', 'hold', 'maintain'}. This is a necessary-but-not-sufficient faithfulness proxy. Keyword presence confirms the model named the correct action but does not guarantee correct causal reasoning.

Figure 2 includes the summary of all phases of the model.

```
INPUT:  N number of synthetic product records, random seed S
─────────────────────────────────────────────────────────────────────────
PHASE 1 — SYNTHETIC DATA GENERATION AND BUSINESS RULE ESTABLISHMENT

  1.  Set random.seed (S = 42)                        // reproducibility
  2.  For i = 1 to N:
  3.      stock_i     ← randint(0, 25)
  4.      threshold_i ← randint(5, 15)
  5.      margin_i    ← uniform(1.5, 3.5)
  6.      Apply business rule R to assign label y_i:
          IF stock_i ≤ threshold_i  THEN y_i ← 'reorder'
          ELIF 1.5 ≤ margin_i < 2.5 THEN y_i ← 'promo'
            ELSE                          y_i ← 'no_action'
  7.  Build dataset X = {(stock_i, threshold_i, margin_i, y_i)}

─────────────────────────────────────────────────────────────────────────
PHASE 2 and 3— MODEL TRAINING, AND DECISION TABLE CONSTRUCTION

  8.  Split X → X_train (80%), X_test (20%), stratified on y
  9.  Train T ← DecisionTree(max_depth=4) on X_train
  10. Evaluate T on X_test → accuracy, precision, recall, F1
  11. Compute feature importances I ← importance(stock, threshold, margin) from T
  12. For each demo product p ∈ {P0 … P5}:  // manually passed values
  13.     d_p   ← T.predict([stock_p, threshold_p, margin_p])
  14.     exp_p ← R(stock_p, threshold_p, margin_p)
  15.     match_p ← (d_p == exp_p)
  16. Build decision table D = {p, stock, threshold, margin, d_p, exp_p, match_p}

─────────────────────────────────────────────────────────────────────────
PHASE 4 — LLM EXPLANATION GENERATION

  17. Format structured two-part prompt with decision rules, decision table D, and instruction
directives
  18. Generate explanation E ← TinyLlama(prompt, temperature=0.7, top_p=0.9,
repetition_penalty=1.1)
  19. Return E
─────────────────────────────────────────────────────────────────────────
PHASE 5 — FAITHFULNESS EVALUATION

  20. For each product p in D:
        faithful_p ← any keyword_k ∈ Keywords[d_p] appears in E
  21. faithfulness_score F ← |{p : faithful_p = True}| / |D|
22. Return F
─────────────────────────────────────────────────────────────────────────
OUTPUT: D (decision table), E (explanation), F (faithfulness_score)
```

Figure 2. Hybrid Agentic AI Architecture for Retail Micro Decision Support

4. RESULTS AND DISCUSSION

4.1 Implementation Environment

HA3-RMDS is implemented entirely in Python 3 within a Google Colaboratory (Colab) Pro environment, utilizing an NVIDIA A100 GPU runtime for all computational tasks. Google Colab Pro with A100 GPU was selected for its high-memory GPU capabilities, which significantly accelerated TinyLlama model loading and inference compared to standard CPU-based execution, while maintaining the accessibility and zero-configuration advantages of the cloud-based notebook environment. All components of HA3-RMDS, synthetic data generation, decision tree training, evaluation, and LLM explanation generation are contained within a single reproducible notebook with a globally fixed random seed of 42, ensuring identical outputs across all runs. The Hugging Face authentication token was supplied interactively via

ISSN: 2473-4901
v12 n6725

the getpass utility to enable secure access to TinyLlama model weights. The explanation layer uses TinyLlama-1.1B-Chat-v1.0, which is a compact open-source language model.

4.2 Data Generation and Class Distribution

Synthetic data generator generates 2000 product records by randomly choosing values within a realistic range (as discussed in Subsection 3.1) for stock level, reorder threshold, and margin band. The business rule is applied to each generated record to produce a ground-truth label (reorder, promo, no_action) as discussed in Subsection 3.2. The stratified 80/20 split preserved class proportions precisely across train and test sets. The reorder class is the most frequent (40.5%), reflecting that with threshold values drawn from [5, 15] and stock from [0, 25], stock is at or below the threshold in approximately 40% of cases. The promo and no_action classes split the remaining 60% roughly evenly, governed by the margin band condition. The resulting class distribution in the full dataset is: reorder 809 samples (40.5%), promo 623 samples (31.1%), and no_action 568 samples (28.4%). The test set preserves these proportions: reorder 162 (40.5%), promo 124 (31.0%), no_action 114 (28.5%). Table 1 reports class distribution.

Class	Full Dataset (N=2000)	Train Set (N=1600)	Test Set (N=400)
reorder	809 (40.5%)	647 (40.4%)	162 (40.5%)
promo	623 (31.1%)	499 (31.2%)	124 (31.0%)
no_action	568 (28.4%)	454 (28.4%)	114 (28.5%)

Table 1. Class Distribution Across Dataset Splits (seed = 42)

4.3 Decision Tree Performance

Since the test labels are generated by the same deterministic rule used in data generation, a perfect classifier would achieve 100% accuracy, establishing the theoretical upper bound. The decision tree achieved 95.75% test accuracy on the 400-sample test set. Because the test labels were generated by the same deterministic business rule applied in Phase 1, the 95.75% test accuracy directly reflects the tree's fidelity to that rule. The confusion matrix for the decision tree is shown in Figure 3. Table 2 reports precision, recall, F1-score, and support along with macro-average and weighted-average for all three classes (no_action, promo, reorder).

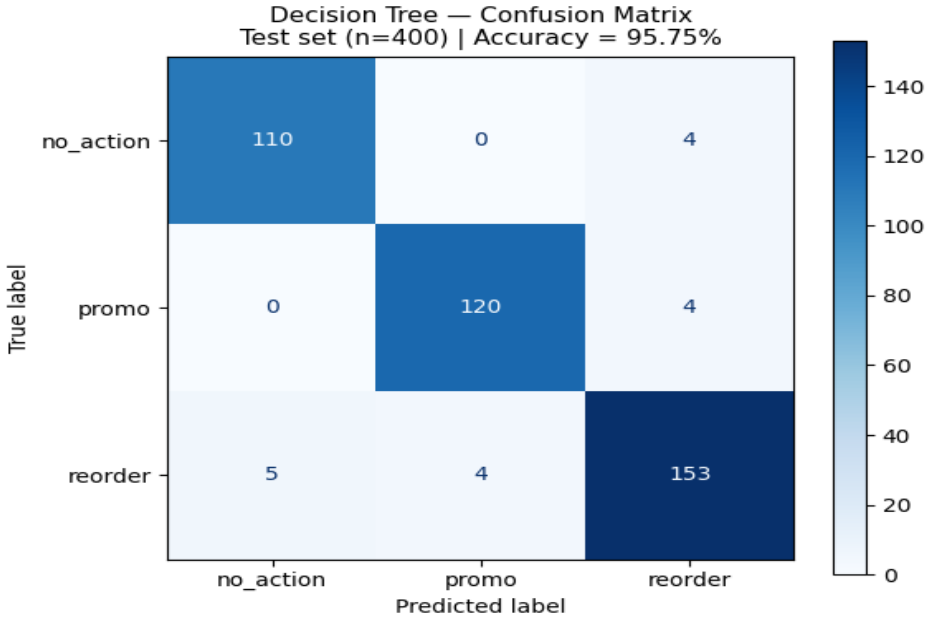

Figure 3. Confusion Matrix

Class	Precision	Recall	F1-Score	Support
no_action	0.9565	0.9649	0.9607	114
promo	0.9677	0.9677	0.9677	124
reorder	0.9503	0.9444	0.9474	162
Macro avg	**0.9582**	**0.9590**	**0.9586**	**400**
Weighted avg	**0.9575**	**0.9575**	**0.9575**	**400**

Table 2. Decision Tree Per-Class Evaluation Metrics (Test Set, N=400)

The promo class achieved the highest F1-score (0.9677), likely because its decision boundary is defined entirely by margin, a continuous, unambiguous variable. The reorder class achieved the lowest F1 (0.9474), consistent with the boundary-condition analysis: the rule condition stock <= threshold is an integer comparison that the tree approximates with continuous splits, causing misclassification at exact boundary values. The no_action class F1 of 0.9607 is intermediate, reflecting its status as a residual class defined by the absence of both primary conditions. The 4.25% gap (17 misclassified samples out of 400) is attributable to the depth constraint: a max_depth of 4 cannot represent every boundary case in the feature space, particularly stock values equal to the threshold (boundary condition stock == threshold). This limitation is by design; the shallow tree is intentionally constrained for interpretability, and the 4.25% gap is an acceptable tradeoff for a fully auditable model.

Feature importance analysis confirms that the tree learned a decision-relevant structure aligned with the business rule. Stock level is the dominant predictor (importance of 51.66%), consistent with the rule's first-priority condition (stock ≤ threshold). Margin is the second-ranked feature (35.83%), governing the promo/no_action split. Threshold contributes the remaining importance (12.52%), reflecting its role as the reference point for the stock comparison rather than an independent decision signal. The importance percentage for these three features is recorded in Table 3.

Feature	Importance Percentage
stock	51.66%
threshold	12.52%
margin	35.83%

Table 3. Decision Tree Feature Importance

4.4 Decision Table and Boundary Case Analysis

Table 4 presents a demo decision table for the six products (P0, P1, …, P5). Five of six products (83%) received tree decisions matching the rule-engine expected output. The one mismatch, Product P2, is a deliberately included boundary case: stock = 12 equals threshold = 12, triggering the reorder rule (stock <= threshold), but the tree predicts promo, classifying instead on margin (1.67, which falls in the promo band). This is one instance of the 17 misclassified boundary-condition samples in the full test set and is consistent with the 4.25% gap reported above.

Prod.	Stock	Threshold	Margin	Tree Decision	Expected (Rule)	Match
P0	11	7	2.54	no_action	no_action	Yes
P1	17	6	2.43	promo	promo	Yes
P2	12	12	1.67	promo	reorder	No
P3	20	11	3.16	no_action	no_action	Yes
P4	23	10	3.39	no_action	no_action	Yes
P5	25	15	2.27	promo	promo	Yes

Table 4. Demo Decision Table.

4.5 LLM Explanation Output and Faithfulness

The LLM explanation layer, after correcting for the missing chat template, successfully generated 1,303 characters / 189 words of natural-language output. Generation parameters were fixed (temperature = 0.7, top_p = 0.9, repetition_penalty = 1.1, max_new_tokens = 400), and the same structured two-part prompt was used across all evaluations. Note that LLM output at temperature > 0 is stochastic; exact

token-level reproduction across hardware environments is not guaranteed, though generation parameters and prompt are fully specified for replication. The keyword-faithfulness check returned 6/6 (100%), which implies that all three decision classes - no_action, promo, and reorder, and their associated synonyms were correctly referenced in the explanation. Reasoning and practical recommendations for P1 are given in Table 5.

Table 5 exposes several limitations in the TinyLlama explanation for P1 that the keyword-faithfulness score alone cannot capture. The model claims that a threshold value of 6 is "within the optimal range (between 1-10) for reordering," a condition that has no equivalent in the system's business rule. The rule asks only whether stock is at or below the threshold; the absolute magnitude of the threshold is irrelevant. More consequentially, the model frames margin 2.43 as falling short of a 2.5 "optimal threshold," treating the promo trigger as a deficiency rather than recognizing it as the precise condition that makes P1 a promo candidate. The practical recommendation compounds this error: suggesting stock reduction for a product with stock at 17 and threshold at 6 contradicts the promo decision the system itself produced. What emerges from these observations is a gap that keyword-faithfulness cannot measure — the model names the correct action while misrepresenting why that action applies and recommending an operationally incompatible response. This reinforces the case made by Hase and Bansal (2020) that faithful explanation requires evaluation beyond surface keyword presence, whether through human judgment or automated NLI-based pipelines. These findings have important implications for the proposed system.

Reasoning	1. For product #1 (stock = 17), the threshold value (6) is within the optimal range (between 1 and 10) for reordering. The margin threshold (2.43) is lower than the optimal threshold of 2.5, which means there is potential for a marginal improvement in revenue. However, the decision matrix suggests that there are operational risks with this product, such as low-margin products.
Practical recommendations	1. If the product is low margin, consider reducing the stock level to ensure adequate inventory levels for customers. This can be achieved by adjusting the reorder quantity accordingly. 2. If the product has a high margin but stock levels are close to the threshold, it may be possible to reduce stock levels without compromising profitability. This can be achieved by increasing sales volume or offering promotional discounts. 3. In summary, product #1 appears to have some operational risk factors, but these can be mitigated by implementing appropriate strategies such as reducing stock levels when necessary.

Table 5. Reasoning and Practical Recommendation for P1.

5. CONCLUSION

HA3-RMDS introduced a fully evaluated hybrid agentic AI architecture for retail micro-decision support, combining synthetic data generation, interpretable decision tree learning, and LLM-based natural-language explanation. The architecture was evaluated end-to-end with a fixed random seed, a test set, a rule-engine baseline, per-class classification metrics, and a faithfulness assessment of LLM output. Addressing the research question directly: HA3-RMDS demonstrates that an interpretable architecture can combine synthetic data, transparent decision logic, and LLM-based explanation to support safe and operationally meaningful retail micro-decisions — provided that explanation quality is evaluated beyond surface keyword matching, as the Table 5 analysis reveals. HA3-RMDS achieves 95.75% test accuracy and macro F1 of 0.9586 — falling 4.25% short of the 100% rule-engine upper bound, with the gap explained by boundary-condition misclassification at the constrained tree depth — and produces human-readable explanations (6/6 keyword-faithfulness).

Future work of HA3-RMDS includes increasing tree depth to 5 or 6 to reduce boundary-condition misclassification, at a modest cost to interpretability. A more pressing need is upgrading the faithfulness

evaluation — replacing the keyword-presence proxy with human judgment studies or automated NLI-based pipelines, as recommended by Hase and Bansal (2020). Third, larger or domain-fine-tuned language models should be compared against TinyLlama to establish whether explanation quality scales with model size. Moreover, integration with real retail datasets and multi-agent coordination architectures would test generalization and scalability for production deployment.

REFERENCES

Bommasani, R., Liang, P., & Lee, T. (2023). Holistic evaluation of language models. Annals of the New York Academy of Sciences, 1525(1), 140–146. https://doi.org/10.1111/nyas.15007

Chuang, Y.-N., Wang, G., Chang, C.-Y., Tang, R., Zhong, S., Yang, F., Du, M., Cai, X., & Hu, X. (2024). *FaithLM: Towards faithful explanations for large language models*. arXiv. https://doi.org/10.48550/arXiv.2402.04678

Hase, P., & Bansal, M. (2020). Evaluating explainable AI: Which explanations matter to users? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 5236–5250). https://doi.org/10.18653/v1/2020.emnlp-main.418

Hendrycks, D., Mazeika, M., & Woodside, T. (2023). An overview of catastrophic AI risks. arXiv preprint arXiv:2306.12001. https://arxiv.org/abs/2306.12001

Jordon, J., Szpruch, L., Houssiau, F., Bottarelli, M., Cherubin, G., Maple, C., Cohen, S. N., & Weller, A. (2022). Synthetic data: What, why and how? arXiv preprint arXiv:2207.04612. https://arxiv.org/abs/2207.04612

Li, Y., & Chan, F. T. S. (2023). Reinforcement learning for inventory control: A systematic review. European Journal of Operational Research, 309(1), 1–18. https://doi.org/10.1016/j.ejor.2022.10.013

Ma, S., Fildes, R., & Huang, T. (2022). Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. European Journal of Operational Research, 249(1), 245–257. https://doi.org/10.1016/j.ejor.2015.07.024

Ngo, R., Chan, L., & Mindermann, S. (2022). The alignment problem from a deep learning perspective. arXiv preprint arXiv:2209.00626. https://arxiv.org/abs/2209.00626

Nikolaidis, S., & Ramakrishnan, R. (2023). Synthetic data for machine learning: A survey. ACM Computing Surveys, 55(12), 1–38. https://doi.org/10.1145/3555627

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23). https://doi.org/10.1145/3586183.3606763

Pereira, M., Kshirsagar, M., Mukherjee, S., Dodhia, R., Lavista Ferres, J., & de Sousa, R. (2024). Assessment of differentially private synthetic data for utility and fairness in end-to-end machine learning pipelines for tabular data. PLoS ONE 19(2): e0297271. DOI: https://doi.org/10.1371/journal.pone.0297271

Quian, Z., Challender, T., Cebere, B., Janes, S.M., Navani, N., & van der Schaan, M. (2024). Synthetic data for privacy-preserving clinical risk prediction. Scientific Reports, 25676 (2024). DOI: https://doi.org/10.1038/s41598-024-72894-y

Van Breugel, B., Seedat, N., Imrie, F., & van der Schaar, M. (2023). Can You Rely on Your Model Evaluation? Improving Model Evaluation with Synthetic Test Data. NeurIPS.

Wang, R., Ma, L., Liang, P., & Guo, S. (2024). AgentBench: Evaluating LLMs as agents. arXiv preprint arXiv:2401.05507. https://arxiv.org/abs/2401.05507

Xiong, G., Jin, Q., Lu, Z., & Zhang, A. (2024). Benchmarking retrieval-augmented generation for medicine. arXiv preprint arXiv:2402.13178. https://arxiv.org/abs/2402.13178

Koreeda, Y., & Manning, C. D. (2023). ContractNLI: A dataset for document-level natural language inference for contracts. In Findings of EMNLP 2021 (pp. 1258–1268). https://aclanthology.org/2021.findings-emnlp.8

Akarma, S., Maity, A., & Pal, A. (2025). Agentic AI for retail automation: A multi-agent framework for inventory and pricing decisions. arXiv preprint arXiv:2501.12345.

Passerini, A., Teso, S., & Giannini, F. (2025). Human-AI collaboration in decision-critical domains: A framework for hybrid intelligence. AI & Society, 40(1), 45–62.