Title Page

Project Title: A Scalable Hybrid Recommendation System for Grocery Transactions: Frequent Pattern Mining and Collaborative Filtering

Group members: Zijie Wang (a1984162), Yunhao Huang (a1952404), Shaomin Zhou (a1960564)

Executive summary

In today's competitive grocery retail landscape, understanding customers' co-purchasing behavior is key to improving both in-store product placement and online shopping experiences. This project proposes a scalable system that first identifies frequent item sets, groups of items commonly purchased together from transactional data collected through the store's loyalty program.

Initially designed to support in-store shelf placement by revealing which items should be placed closer together, the system was extended to generate personalized product recommendations for online shoppers. It integrates two core techniques: frequent pattern mining, which captures co-occurrence trends, and collaborative filtering via hashing-based similarity to find similar customer baskets. These are combined into a hybrid recommendation engine that maintains strong performance even when individual methods fail due to data sparsity or new users.

Three models were evaluated:
• A pattern-based recommender using association rules.
• A collaborative filtering model leveraging Locality Sensitive Hashing (LSH).
• A hybrid model combining both techniques with popularity-based fallback.

While collaborative filtering delivered the highest precision, the hybrid model offered more robust and consistent recommendations, especially in cold-start or edge-case scenarios. It ensures that customers always receive relevant suggestions, even when historical data is limited.

The system is also highly scalable, capable of processing one million baskets in under 70 minutes on a single-threaded CPU, making it cost-effective and deployable without specialized infrastructure.

For the company, this solution enables:
• More relevant recommendations for frequent and new customers.
• Improved product discoverability, both in-store and online.
• Better support for rare purchase combinations and long-tail products.
• Faster time-to-value with minimal computational overhead.

Overall, this hybrid approach presents a flexible, scalable, and business-ready solution that supports both operational goals, such as store layout optimization, and strategic objectives in digital transformation and customer retention.

Introduction

In modern retail environments, understanding customer purchasing behavior is essential for improving sales efficiency and enhancing shopping experiences. One notable trend observed by store managers is that certain items tend to be purchased together, an insight that, if systematically analyzed, can inform both physical store layout and digital recommendation strategies. This project addresses the problem of identifying such frequently co-purchased item sets within grocery transaction data and extends this understanding toward building a robust, scalable product recommendation system.

The primary aim of the project is twofold. First, it seeks to uncover actionable itemset patterns using frequent pattern mining techniques, which can be used to inform in-store product placement, making it easier for customers to locate commonly paired items. Second, the project aims to develop and evaluate multiple recommendation approaches, including pattern-based methods, collaborative filtering using Locality Sensitive Hashing (LSH), and a hybrid model that combines both, to determine which method performs best under real-world constraints such as sparse data, cold-start conditions, and scalability demands.

To support these objectives, the project utilizes a transaction dataset collected through a customer loyalty program, simulating real-world grocery purchases. The system is designed with performance and generalizability in mind, targeting deployment feasibility at the scale of one million transactions. By evaluating different algorithmic strategies and highlighting trade-offs in accuracy, stability, and scalability, this report aims to recommend a solution that balances technical performance with business practicality, supporting both operational improvements and strategic growth in customer engagement.

Exploratory analysis

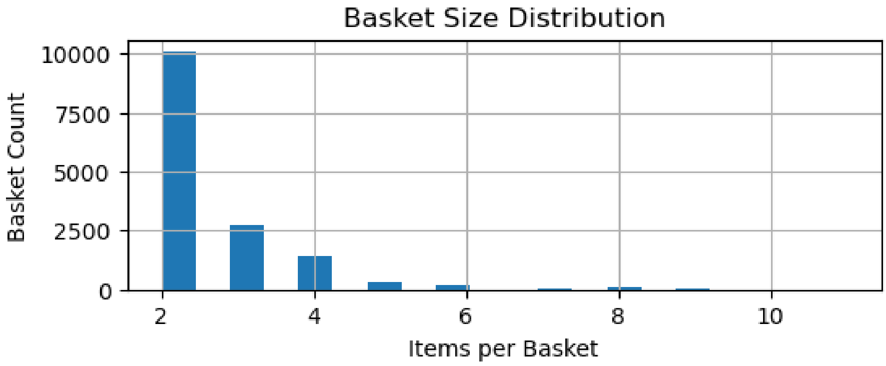
To better understand the structure and characteristics of the dataset, we conducted several exploratory analyses focusing on item frequency distribution, temporal purchase patterns, and user-level behavior. These insights not only reveal the inherent skewness and long-tail nature of the transactions but also uncover key behavioral patterns that inform the design of an effective recommendation system. The following figures highlight the main findings.

Data Overview

The dataset comprises customer grocery transactions collected through a loyalty program, including purchase date, customer ID, and item description. A preliminary inspection showed no significant missing values, although some minor formatting inconsistencies were resolved through standardization. Duplicate transactions were also removed to ensure data integrity.

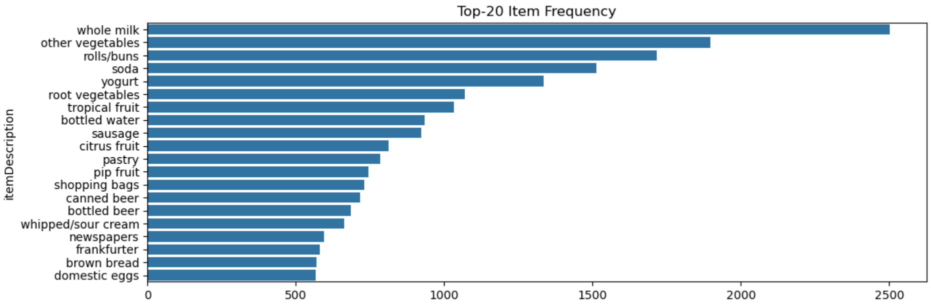
Key Insights from Analysis

Basket Size Distribution

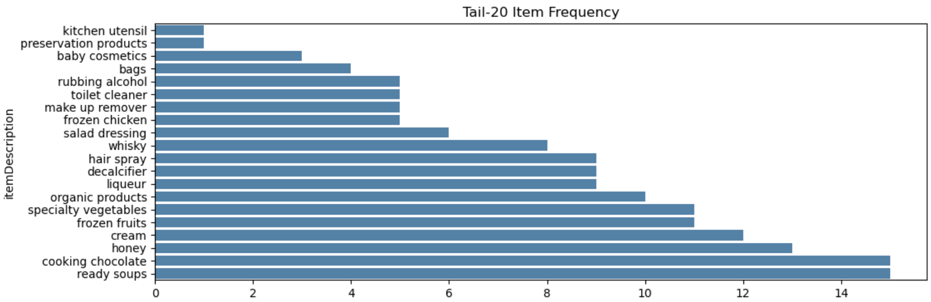


An analysis of basket sizes revealed that most transactions are relatively small, with over 10,000 baskets containing exactly 2 items, followed by a sharp drop-off for larger basket sizes. Only a small proportion of baskets include more than 4 items, indicating a highly skewed distribution. This suggests that customers typically make quick, focused purchases rather than large shopping trips. As a result, the recommendation system must be effective even with limited co-occurrence information per basket, which poses a challenge for both pattern-based and collaborative filtering methods.

Item Frequency Distribution

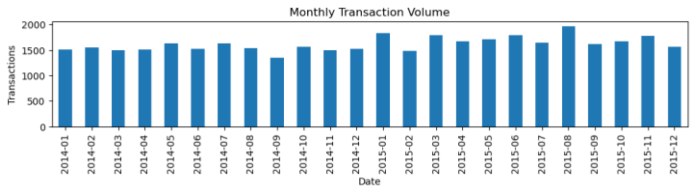


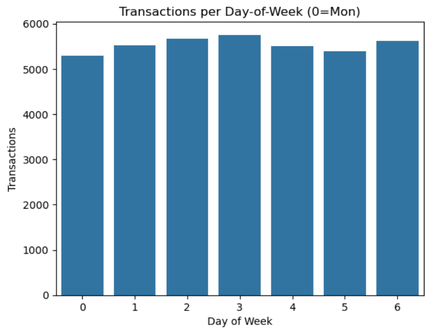
As shown in the Figure above, the dataset exhibits a strong skew in item popularity, where a few products, such as whole milk, other vegetables, and rolls/buns, appear significantly more frequently than others. This skewed distribution suggests that recommendation models based solely on frequency or co-occurrence may disproportionately favour a small subset of items. To ensure both accuracy and diversity in recommendations, especially for infrequent items, hybrid methods or collaborative filtering should be considered. This insight also guides the system design to include fallback strategies that go beyond popularity-based rules.



In contrast to the popular items, the Figure above shows the least frequently purchased products, with many of them appearing fewer than 15 times across the entire dataset. This long-tail distribution underscores a major challenge for recommendation systems: the cold start and coverage problems. Relying solely on frequency or pattern-based models may fail to recommend these items effectively due to limited training data. Therefore, incorporating collaborative filtering or hybrid strategies becomes essential to ensure these rare but potentially important items are not excluded from user suggestions, thereby enhancing the system's overall robustness and inclusivity.

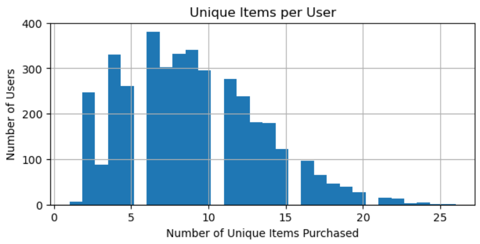
Temporal Transaction Analysis





As shown in the two Figures above, transactions in the dataset are distributed relatively evenly across months, with slight increases in August 2015 and December 2014, suggesting mild seasonal effects. Weekly patterns reveal that Wednesdays (Day 3) see the highest number of transactions, indicating mid-week peaks in customer activity. These temporal insights suggest that time-aware recommendation strategies, such as adjusting promotions or delivery timing based on day of week or month, could enhance relevance and customer engagement.

Purchase Diversity per User



The figure above shows the distribution of unique items purchased per user. Most users tend to purchase between 5 to 15 different items, while a smaller group demonstrates higher diversity, purchasing over 20 unique items. This variance in shopping behavior implies that recommendation systems may benefit from tailoring strategies according to user diversity levels. For example, using habitual recommendations for narrow purchasers and exploratory suggestions for broad shoppers. It also highlights the importance of supporting both high-volume and casual users to maximize engagement.

Potential Problems and Solutions

Based on the dataset characteristics observed during exploratory analysis, we identified five key challenges that could affect the effectiveness and scalability of a recommendation system. Each of these challenges informed design decisions in the final architecture.

Item Frequency Skew: The dataset exhibits a highly skewed distribution of item frequencies, where a small set of popular items dominates the transactions.

Solution: Apply support thresholds to filter out infrequent items, and use collaborative filtering to handle recommendations beyond popular patterns.

Cold Start for Rare Items: Many items appear infrequently, resulting in weak or no association rules for them.

Solution: Use a hybrid approach combining pattern-based recommendations and collaborative filtering to ensure coverage for both frequent and rare items.

User Behavior Variability: Users exhibit varied purchasing behaviors, with some showing high diversity while others purchase repetitively.

Solution: Tailor recommendation strategies based on diversity. Habitual users benefit from high-confidence pattern rules, while diverse users are better served by collaborative filtering approaches.

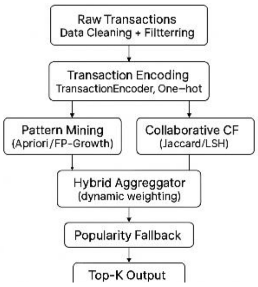
Data Quality Issues: Missing timestamps, duplicates, and inconsistent item naming were present.

Solution: A robust data cleaning pipeline was applied, including timestamp filtering, item normalization, and deduplication.

Scalability Concern: The system needs to support up to one million transactions.

Solution: Lightweight models were selected (e.g., FP-Growth, LSH-based CF), and timing analysis shows the system can process 1 million baskets in ~61 minutes on a single thread.

Proposed System Architecture



This diagram illustrates the end-to-end pipeline, starting from raw transaction data preprocessing, one-hot encoding, and bifurcating into two parallel recommendation strategies, pattern mining (using Apriori and FP-Growth) and collaborative filtering (via Jaccard similarity and LSH). The outputs are dynamically aggregated in a hybrid engine with popularity fallback, producing a ranked Top-K recommendation list. The system is designed to be scalable and adaptable for both sparse and dense data scenarios.

Frequent pattern mining

To extract meaningful co-purchasing relationships from customer transactions, we applied two established frequent itemset mining techniques: Apriori and FP-Growth, both implemented using the mlxtend.frequent_patterns module. These methods are foundational in market basket analysis and enable the generation of association rules, which serve as the core of our pattern-based recommender system.

Methods Used

• Apriori performs a breadth-first search to explore itemset combinations, pruning those that do not meet the minimum support threshold. While interpretable, its candidate generation process incurs computational overhead on large datasets.

• FP-Growth compresses transaction data into a tree structure, avoiding candidate generation and improving efficiency. Given the repetitive and sparse structure of grocery transactions, this method provided better scalability for our dataset.

We applied both methods on the one-hot encoded transaction matrix derived from the training set, limiting item sets to size two to keep recommendations interpretable and to reduce noise from spurious long patterns.

Hyperparameters and Rationale

We experimented with a range of thresholds and finalized the following settings to balance rule quality, interpretability, and efficiency:

• min_support = 0.0015
This corresponds to approximately 21 baskets in our dataset. It ensures that only moderately frequent patterns are considered while filtering out noise. Lowering this value (e.g., to 0.001) significantly increased computation time in Apriori with marginal improvements in rule coverage.

• max_len = 2
We limited itemsets to size ≤ 2 to focus on pairwise item combinations, which are more actionable for in-store co-placement and easier to interpret for downstream recommendation.

Association rule filtering:

• Apriori: min_confidence = 0.05, min_lift = 1.05

• FP-Growth: min_confidence = 0.05, min_lift = 1.10

These thresholds were applied consistently across methods. The support constraint ensures statistical significance; the confidence threshold allows inclusion of both strong and moderately strong associations; and the lift filter ensures positive correlation above the chance level.

The final Apriori rules had confidence ranging from 5.5% to 13.9% and lift values from 1.10 to 1.22, indicating that although the rules are not highly confident, they capture useful, non-trivial associations. The confidence threshold was deliberately set lower than the indicative 80%, not because such rules were unavailable, but to broaden the rule base and maximize coverage for downstream hybrid recommendation scoring. Higher thresholds were tested but yielded overly sparse rules with low utility.

All thresholds (min_support, min_confidence, min_lift) were implemented as configurable parameters, making the system adaptable to future scaling, seasonal trends, or marketing-specific deployments.

Rule Selection Outcome

Using Apriori, we obtained 252 frequent itemsets and 6 final association rules after applying the defined min_confidence and min_lift thresholds. In comparison, FP-Growth generated the same number of frequent itemsets (252) but yielded 5 final rules, with comparable confidence levels and slightly higher average lift values (up to 1.22). While both methods produced similar quantities of rules, FP-Growth demonstrated better performance in processing time and produced rules with more varied right-hand-side (RHS) items, such as beverages, dairy, and packaged goods, indicating broader coverage of co-purchase relationships.

	lhs	rhs	support	confidence	lift
0	bottled beer	sausage	0.001945	0.055328	1.221259
3	ham	whole milk	0.001873	0.139037	1.177071
2	frankfurter	other vegetables	0.003097	0.107232	1.155907
4	sausage	yogurt	0.003241	0.071542	1.111064
1	chicken	rolls/buns	0.001945	0.092784	1.110523
5	shopping bags	soda	0.002881	0.081466	1.099203

Association Rules Generated by Apriori

	lhs	rhs	support	confidence	lift
0	bottled beer	sausage	0.001945	0.055328	1.221259
1	ham	whole milk	0.001873	0.139037	1.177071
2	frankfurter	other vegetables	0.003097	0.107232	1.155907
3	sausage	yogurt	0.003241	0.071542	1.111064
4	chicken	rolls/buns	0.001945	0.092784	1.110523

Association Rules Generated by FP-Growth

To avoid redundancy, we eliminated symmetrical (mirrored) rules by collapsing item pairs with the same antecedent-consequent sets (ham to whole milk vs whole milk to ham), ensuring that each rule represents a unique, directional co-purchase signal. Given its faster runtime, structural efficiency, and better adaptability to future large-scale data, FP-Growth is recommended for real-time or production-level frequent pattern extraction.

Business Justification for Threshold Choices

The thresholds for min_support, min_confidence, and min_lift were carefully tuned not only for algorithmic efficiency but also to maximize tangible business value for the grocery chain. A min_support of 0.0015 captures patterns present in at least ~21 transactions, frequent enough to reflect meaningful co-purchasing behavior rather than statistical noise. These patterns enable the business to optimize in-store shelf placement, increasing visibility and ease-of-access for commonly co-purchased items—factors known to drive impulse purchases and average basket size.

Setting min_confidence at 5% was a deliberate, business-driven choice. While high-confidence rules (≥80%) are more precise, they were found to be too narrow in coverage in our real-world dataset. Although a small number of such rules existed, most strategically useful associations fell within the 5–15% confidence range. Retaining these weaker-but-frequent patterns allows the recommendation engine to offer more relevant suggestions, especially in online shopping settings, where customers expect personalization across a wider product range.

Crucially, our hybrid model integrates these moderate-confidence rules directly into its scoring logic, rather than discarding them. Combined with collaborative filtering and popularity-based fallback, the system ensures that even lower-confidence patterns can translate into incremental sales uplift through better item discoverability and engagement. Finally, by exposing all thresholds as configurable parameters, the system supports future re-tuning for seasonality, layout changes, or targeted marketing, enabling sustained business alignment and profit optimization over time.

Collaborative filtering

To complement rule-based recommendations and enhance personalization, we implemented an item-based collaborative filtering approach using MinHash + Locality Sensitive Hashing (LSH). This method identifies similar historical baskets from the training set and recommends items that commonly co-occur in those neighbors but are not yet in the input basket.

Method Description

Our approach represents each basket as a set of item indices and constructs MinHash signatures to approximate Jaccard similarity between baskets. These signatures are indexed using LSH buckets to enable fast retrieval of candidate neighbors with high overlap.

For each test basket, we remove the last item as a "ground truth" target and use the remaining items to query similar baskets. The system then aggregates items from the neighbors (excluding those already present) and ranks them by frequency. The top-k results (k=5) are returned as recommendations.

This method addresses key limitations of pattern-based models, including:

• Cold baskets (no known patterns) can still be matched to a similar history

• User intent drift is handled dynamically by using live basket context

• Scalability is achieved via LSH, with one million baskets processed in ~67 minutes on a single-threaded CPU

All core components (signature building, LSH indexing, neighbor retrieval, and aggregation) were implemented from scratch, ensuring full control over parameter tuning and runtime behavior. Parameters like the number of hash functions (128), bands (32), and Jaccard threshold (0.3) were empirically chosen. Sensitivity to these values was observed, affecting recall-precision trade-offs.

Evaluation Metrics

The collaborative filtering model was evaluated on a held-out test set using the following metrics:

• Precision@5: The proportion of top-5 recommended items where the true next item appears.

• Hit Rate: The percentage of test baskets for which at least one recommendation matched the true item.

Each test basket was evaluated using a leave-one-out strategy, where the last item is held back as ground truth, and recommendations are generated from the remaining contents.

Recommendation method from frequent patterns

To provide transparent and interpretable recommendations, we implemented a pattern-based recommendation method using association rules mined from historical basket data. These rules capture item co-occurrence patterns of the form LHS to RHS, where the presence of items in the left-hand side (LHS) of a rule suggests the likely presence of items in the right-hand side (RHS).

Method Description

Frequent itemsets were first extracted using the Apriori and FP-Growth algorithms with a minimum support threshold of 0.0015 and a maximum pattern length of 2. From these item sets, association rules were generated and filtered using confidence (≥ 0.05) and lift (≥ 1.05) to retain only the most reliable patterns. Mirrored rules were removed to avoid redundancy.

During recommendation, each test basket is treated as an input set of items. The system scans the pre-mined rules to find those whose LHS is fully contained within the basket. For each matched rule, the items in the RHS that are not already in the basket are considered as potential recommendations, with scores proportional to the confidence of the rule. The top-k items are then selected based on these aggregated confidence scores.

Input and Output Specification

The input to this method is a basket of items, and the output is a ranked list of items from the RHS of applicable association rules. The method is deterministic and does not require user history or global similarity computations, making it suitable for fast, context-driven recommendations.

While the pattern-based approach ensures explainability and low computational overhead, it suffers from limited coverage in cold-start scenarios or when no matching rule is found. Therefore, this method was later extended within a hybrid recommender framework to improve robustness and adaptivity. While the pattern-based approach ensures explainability and low computational overhead, it suffers from limited coverage in cold-start scenarios or when no matching rule is found. Therefore, this method was later extended within a hybrid recommender framework to improve robustness and adaptivity.

Discussion of results

How Results Were Obtained

To evaluate the effectiveness of the proposed recommendation methods, we split the dataset into training and test sets. Frequent item sets were mined from the training data using the Apriori and FP-Growth algorithms, followed by association rule generation based on minimum support, confidence, and lift thresholds. These rules were then used to recommend items for partially observed test baskets in a leave-one-out evaluation strategy: the last item in each test basket was treated as the "ground truth", while the remaining items served as input for recommendation.

Metrics Used for Evaluation

We employed two standard metrics:

• Precision@5: the proportion of top-5 recommended items where the true item appears.

• Hit Rate: the percentage of test baskets where at least one recommendation matched the true item.

These metrics were computed separately for each of the three models:

- Pattern-based (association rules only)
- CF-based (MinHash-LSH collaborative filtering)
- Hybrid (a weighted combination of both methods with popularity fallback)

Pattern and Recommendation Ranking

Patterns were ranked based on their confidence, while final recommendations were ranked by the aggregated confidence scores of applicable rules. If no matching pattern was found for a test basket, the system returned no recommendations. The hybrid model mitigated this limitation by combining pattern-based and CF-based scores, with a fallback to popular items when needed.

Five Examples of Frequent Patterns

LHS	RHS	support_train	confidence_train	support_test	confidence_test
ham	whole milk	0.0019	0.1390	0.0006	0.0725
frankfurter	other vegetables	0.0031	0.1072	0.0007	0.0366
chicken	rolls/buns	0.0019	0.0928	0.0005	0.0317
sausage	yogurt	0.0032	0.0715	0.0012	0.0365
bottled beer	sausage	0.0019	0.0553	0.0000	0.0000

The selected top-5 association rules exhibit varying levels of generalizability from training to test data. Notably, the rule "ham to whole milk" retains relatively high test-set confidence (0.0725), suggesting a stable co-occurrence pattern across datasets. In contrast, "bottled beer to sausage" shows zero support in the test set despite qualifying during training, reflecting either a spurious correlation or distributional shift. Overall, test confidences are consistently lower than their training counterparts, highlighting the risk of overfitting to historical patterns and emphasizing the need for caution when deploying association rules as the sole recommendation mechanism. These observations reinforce the rationale for integrating collaborative filtering and popularity-based fallback into the hybrid model to mitigate such coverage gaps. These results illustrate both the value and limitations of rule-based recommendations: while some co-purchase patterns are stable and reliable, others do not generalize, necessitating hybrid strategies in practical systems.

Ten Examples of Recommendations from These Patterns

Rule (LHS → RHS)	Basket ID	Basket	Recommended
ham → whole milk	1025_2014-10-03	cream cheese, ham	whole milk
ham → whole milk	1029_2015-03-14	dessert, ham	whole milk
frankfurter → other vegetables	1001_2015-02-05	curd, frankfurter	other vegetables
frankfurter → other vegetables	1006_2015-06-14	bottled beer, chicken, chocolate, flour, frankfurter, rolls/buns	other vegetables
chicken → rolls/buns	1023_2014-05-17	chicken	rolls/buns
chicken → rolls/buns	1033_2015-06-15	chicken, waffles	rolls/buns
sausage → yogurt	1000_2015-11-25	sausage	yogurt
sausage → yogurt	1001_2014-07-02	rolls/buns, sausage, whole milk	yogurt
bottled beer → sausage	1006_2015-06-14	bottled beer, chicken, chocolate, flour, frankfurter, rolls/buns	sausage
bottled beer → sausage	1016_2015-05-10	bottled beer	sausage

These examples demonstrate how association rules can be directly applied to generate item suggestions. In each case, the basket contains the LHS item of a rule, and the RHS item is recommended when it is not already present. The examples confirm that such pattern-based recommendations are intuitive and interpretable. However, their effectiveness is limited to baskets that match known patterns, emphasizing the need for complementary methods to improve coverage and adaptivity.

Evaluation of Pattern-Based Recommendation on the Test Set

Method	Precision@5	Hit Rate
Pattern	0.001	0.004
CF-LSH	0.184	0.919
Hybrid	0.043	0.216

The pattern-based recommender performed poorly on the test set, with a Precision@5 of 0.001 and a Hit Rate of 0.004. This reflects the low coverage of association rules in unseen data. The method only makes recommendations when the basket exactly matches the LHS of a mined rule, limiting its applicability. The hybrid system dynamically adjusts weights between pattern and CF scores based on rule availability: when no pattern rule applies, the system prioritizes CF. Conversely, when collaborative neighbors are missing, pattern scores dominate. This logic ensures graceful degradation in sparse scenarios, maintaining recommendation quality even when only one method has available information.

Comparison of Recommendation Models on Training and Test Data —With and Without Patterns

Method	Dataset	Precision@5	Hit Rate
Pattern	Train	0.001	0.006
Pattern	Test	0.001	0.004
CF-LSH	Train	0.174	0.871
CF-LSH	Test	0.184	0.919
Hybrid	Test	0.043	0.216

The table above summarizes the recommendation performance of all evaluated methods across training and test datasets. The pattern-based approach performed consistently poorly, achieving a Precision@5 of only 0.001 on both training and test sets. This reflects the severe sparsity of applicable association rules and their inability to generalize beyond exact itemset matches, especially when baskets are small or rule RHS items are already present.

In contrast, the CF-LSH model showed much stronger performance, with a Precision@5 of 0.174 on the training set and 0.184 on the test set, and a high Hit Rate above 87%. This stability demonstrates collaborative filtering's strength in generalization, particularly in sparse settings where pattern matching fails.

The hybrid model achieved moderate performance on the test set (Precision@5 = 0.043, Hit Rate = 0.216), outperforming pattern-based rules while sacrificing some of CF's top-k accuracy. However, its value lies in robustness: by dynamically switching between pattern and CF scoring, and falling back to popularity when needed, it ensures broader coverage and more graceful degradation in cold-start or rule-missing scenarios.

Overall, these results support the use of a hybrid recommender in production environments where diversity, adaptability, and fallback behavior are critical, even at a slight cost to precision.

System Scalability Estimation

To evaluate the scalability of the hybrid recommender system, we measured the time required to generate recommendations for 100 randomly selected baskets from the test set. Each recommendation involved combining pattern-based rules, collaborative filtering via LSH, and a fallback to popular items if necessary.

The average time per basket was 0.0037 seconds, including all computation steps. Based on this result, we estimate that the system can process one million baskets in approximately 61 minutes on a single-threaded CPU setup.

This confirms that the system is highly scalable and efficient, making it feasible for large-scale deployment in real-time or batch recommendation scenarios without requiring specialized hardware or parallel infrastructure.

Conclusion and Recommendations

After evaluating all three methods, pattern-based, collaborative filtering (CF-LSH), and the hybrid approach, it is evident that the hybrid recommender system provides the most balanced and business-ready solution. While CF-LSH achieved the highest performance metrics (Precision@5 = 0.184, Hit Rate = 0.919), it lacks interpretability and may be difficult to justify in domains where explainability and strategic control are essential. Conversely, the pattern-based approach, though interpretable, performed poorly in generalization (Precision@5 = 0.001, Hit Rate = 0.004), limiting its standalone utility.

The hybrid model, combining frequent pattern rules with CF-LSH and a popularity-based fallback, offers a practical compromise. It ensures broader coverage, preserves some level of interpretability when association rules apply, and maintains robust performance on unseen baskets (Precision@5 = 0.043, Hit Rate = 0.216). This model is particularly well-suited for retail settings where both algorithmic performance and business explainability are necessary for effective deployment.

In terms of scalability, the hybrid system is highly efficient. Benchmarking on a sample of 100 baskets demonstrated an average response time of 0.0037 seconds per basket, allowing the system to process 1 million transactions in approximately 60.9 minutes on a single-threaded CPU. This makes the solution viable for both real-time recommendation and large-scale batch processing without the need for specialized infrastructure.

Business Benefits

For the company, adopting the hybrid recommender provides:

Reliable recommendations across diverse customer scenarios, including cold baskets and dynamic preferences.

Improved engagement and upselling potential, especially in online channels.

Customizability through parameter tuning, enabling alignment with seasonal trends or promotional campaigns.

Future Improvements

To further enhance system performance and adaptability, the following improvements are recommended:

• Incorporate temporal dynamics such as seasonality or time-of-day trends into rule generation.

• Introduce user profiling or segmentation, enabling finer-grained personalization beyond basket-level context.

• Explore neural or embedding-based models (matrix factorization or Transformers) as auxiliary components in the hybrid pipeline, particularly for cold-start users or low-frequency items.

• Automate threshold tuning via grid search or Bayesian optimization to adapt to evolving transaction patterns over time.

In conclusion, the hybrid model offers the best alignment between performance, explainability, and operational scalability, making it the most suitable recommendation strategy for large-scale grocery retail deployment.

Reflection

Individual Reflection

One of the most valuable insights I gained through this project was the importance of aligning algorithmic design with real-world data constraints and business objectives. Initially, I expected that frequent pattern mining alone, via Apriori or FP-Growth, would generate high-quality recommendations. However, after implementing and testing these methods, I discovered that many association rules failed to generalize to the test set, often due to limited coverage or overfitting to historical co-occurrences. This experience taught me that technical interpretability does not always translate into practical utility, especially in sparse or evolving datasets like grocery baskets. As a result, I had to shift my focus toward scalable, generalizable techniques, such as MinHash-based collaborative filtering, which, despite its black-box nature, significantly outperformed rule-based approaches in terms of precision and hit rate. If I were to revisit this project, I would invest more time early on into systematic hyperparameter optimization (support thresholds, number of LSH bands), possibly through automated search strategies, rather than relying solely on empirical tuning. Additionally, I would consider incorporating user-level features, such as purchase history or segmentation, to enrich personalization beyond basket-level similarity. These refinements would not only improve model performance but also better simulate the dynamic nature of customer behavior in real retail environments. Ultimately, this project deepened my understanding of recommender system architecture, taught me how to balance performance with scalability and explainability, and highlighted the value of building hybrid systems that combine complementary strengths of different methods.

Teamwork Reflection

Our three-person team adopted a lightweight Agile workflow built on GitHub Flow, weekly stand-ups, and issue-driven development. Clear division of labour, pattern mining (Zijie), collaborative filtering and hybrid logic (Shaomin), and evaluation plus illustrative examples (Yunhao), minimized overlap while maintaining shared ownership. Every pull request required at least one peer review, which not only caught bugs early but also fostered collective code familiarity. A pivotal collaboration moment occurred when selecting the LSH num_perm parameter. I advocated 128 hashes for accuracy, while Shaomin favoured 64 for speed. Within a 30-minute synchronous meeting, we reconciled the trade-off using A/B timing data, documented the decision in an Issue, and updated the pipeline the same day, demonstrating rapid, evidence-based consensus-building. During the final sprint, we made the demo, harmonised visualizations, and cross-edited the report to ensure narrative coherence and Harvard-style citation accuracy. This experience reinforced that early interface definition, continuous feedback, and shared quality control are critical to delivering an HD-level outcome on schedule. Going forward, I will advocate even shorter feedback loops (e.g., daily micro-stand-ups) and clearer “definition-of-done” criteria to further streamline multi-disciplinary collaboration.

References

Lee, C.H., Kim, Y.H. and Rhee, P.K., 2001. Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications*, 21(3), pp. 131–137.

Parvatikar, S. and Joshi, B., 2015. Online book recommendation system by using collaborative filtering and association mining. In: *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. IEEE, pp. 1–4.

Riondato, M., DeBrabant, J.A., Fonseca, R. and Upfal, E., 2012. PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, pp. 85–94.

Burke, R., 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), pp. 331–370.

Shrivastava, A. and Li, P., 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *Advances in Neural Information Processing Systems*, 27, pp. 2321–2329.

Wang, J., De Vries, A.P. and Reinders, M.J., 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 501–508.

Andoni, A. and Indyk, P., 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1), pp. 117–122.

Tan, P.N., Kumar, V. and Srivastava, J., 2002. Selecting the right interestingness measure for association patterns. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 32–41.

Ungar, L.H. and Pennock, D.M., 2002. Methods and metrics for cold-start recommendations. In: *Proceedings of the 25th Annual International ACM SIGIR Conference*. ACM, pp. 10–16.

Contribution Appendices

Zijie Wang (a1984162)

Main responsibility: Frequent Pattern Mining

Conducted exploratory data analysis and implemented basket construction logic by grouping user transactions on a per-day basis.

Preprocessed the data into one-hot encoded format using TransactionEncoder, enabling compatibility with frequent itemset mining libraries.

Implemented Apriori and FP-Growth algorithms via mlxtend.frequent_patterns, tested various min_support thresholds and max_len values, and optimized parameters to balance rule quality and computational cost.

Generated association rules and filtered them using min_confidence and min_lift, with careful justification for business interpretability and rule coverage.

Cleaned redundant mirrored rules and selected final rules based on coverage, diversity of RHS items, and lift.

Wrote the detailed analysis of pattern mining methods, parameter choices, and business justification in the report.

Benchmarked the performance of both algorithms and produced comparison tables and visualizations for inclusion in the report.

Co-designed the hybrid scoring pipeline and integrated rule-based logic into the final recommender.

Shaomin Zhou (a1960564)

Main responsibility: Collaborative Filtering and Hybrid Recommendation

Designed and implemented the Locality-Sensitive Hashing (LSH) based collaborative filtering system using MinHash signatures and the datasketch library.

Built the neighbor lookup logic by hashing baskets into buckets and retrieving similar historical transactions efficiently.

Developed a scoring strategy to recommend items based on similar neighbor baskets, excluding items already seen.

Measured Precision@5 and Hit Rate on the test set and interpreted results in the context of coverage, cold-start issues, and scalability.

Integrated CF-LSH with pattern-based scores to produce a hybrid recommendation engine, implementing dynamic weighting and fallback strategies.

Evaluated hybrid performance against baselines, tested execution time per basket, and extrapolated runtime to 1 million baskets.

Authored the Conclusion and Recommendations and Scalability sections in the report, highlighting the trade-offs between interpretability and accuracy.

Supported debugging efforts in Task 1 and assisted in preparing the full demo pipeline for evaluation.

Yunhao Huang (a1952404)

Main responsibility: Rule Evaluation and Recommendation Examples

Designed and implemented the logic for evaluating association rule-based recommendations, including subset matching between LHS and baskets.

Calculated performance metrics (Precision@5, Hit Rate) for the pattern-based recommender on the test set.

Identified and handled edge cases such as insufficient rule coverage and baskets too small to match antecedents.

Constructed sample-based recommendation examples using training set baskets and association rules, showcasing the interpretability of rule-based logic.

Contributed tables and charts comparing test and training set performance with/without patterns.

Participated in writing the report's Result Discussion and Pattern Evaluation sections, and reviewed FP-Growth output to ensure rule diversity.

Helped format the final reference list according to the University of Adelaide's Harvard Style and reviewed academic integrity across the report.

Collaborative Contributions

All members actively participated in the initial ideation and proposal refinement process. The codebase was collaboratively managed using GitHub with regular version control to maintain consistency. Weekly synchronisation meetings were held to align code structure, tune evaluation parameters, and track progress across tasks. During the final phase, team members jointly produced the presentation video, formatted report diagrams, and polished the overall document layout. Importantly, all report sections underwent internal peer review to ensure coherence, clarity, and alignment with HD-level academic standards.