Food purchase recommendation system based on frequent pattern mining and collaborative filtering

Group 16: Hui Tao a1878184,
Lanxi Zhang a1883049, Zhige Zhang a1894523

Faculty of Sciences, Engineering and Technology
Master of Computing and Innovation

2024

CONTENTS

LIST OF FIGURES AND TABLES

Figure Lists:

Table Lists:

1 Executive Summary

Traditional recommendation systems struggle to capture nuanced individual preferences, while offline grocery success hinges on proper shelf arrangements and personalized products (Neysiani et al., 2019, pp. 48-55). An excellent system helps us grasp frequently purchased combinations, optimize in-store shelf displays for convenience and repeat purchases, and manage inventory to reduce stockouts and costs (Jannach et al., 2020, pp. 88-98).

To identify the optimal system, we tested three methodologies: association rules from frequent item-sets for recommendation, collaborative filtering deriving recommendations directly from transaction data, and transaction data mining to detect frequent item-sets used for clustering similarities and extracting related user preferences into new item-sets. Combining these new item-sets with collaborative filtering outcomes enables highly relevant suggestions per customer's purchase history.

The recommend system we chose is the association rule based on FP-growth algorithm due to it efficiently identifies and recommends items that customers are most likely to buy based on frequently bought item patterns, achieving a precision of 17.34%, recall of 5.15%, and MAP of 5.15%, and it scales well with reasonable computational costs, making it suitable for large-scale, real-time applications.

2 Introduction

Our aim is to identify the optimal recommendation system that can detect customers' frequently purchased product item-sets, optimize in-store layouts, and provide highly relevant personalized recommendations to enhance the customer experience. This will help stores lower costs and boost sales, achieving business benefits. (Jannach et al., 2020, pp. 88-98).

Current mainstream recommendation systems include collaborative filtering based on users or items. However, collaborative filtering faces challenges like data sparsity, scalability issues, and low recommendation efficiency. Association rule mining and frequent pattern mining also have limitations when handling big data. Additionally, customer purchase habit data collected through loyalty programs may lack new user data and other issues, compounded by the long-tail effect and a high proportion of popular data, further exacerbating data sparsity, leading to inaccurate recommendations. (Muthuperumal, S., Titus, P. & Venkatachalapathy, M, 2020, p. 18690)

To address these problems, we are exploring an improved hybrid method combining collaborative filtering with association rule mining, incorporating neighborhood-based techniques, clustering methods, and association rule mining to enhance collaborative filtering performance. The key objectives are: 1) Applying frequent itemset and association rule mining on transaction data to identify high-frequency product combinations and recommendations; 2) Using standard collaborative filtering methods for recommendations; 3) Generating personalized recommendations through clustering on the collaborative filtering system; 4) Evaluating the effectiveness of various recommendation methods to determine the optimal system.

3 Exploratory Analysis

3.1 Data Analysis

For the loyal user purchase record dataset, we focus on analyzing user purchasing behavior and merchandise sales.

- **Item Set Size Distribution:** demonstrates the typical number of items purchased by users in a single transaction, providing insights for product bundling strategies or cross-selling recommendations.
- **Recency and Frequency Distribution:** shows recent user purchasing habits and peaks and troughs of purchases within specific time periods based on recency and frequency distribution, which is beneficial for targeted promotions or inventory management.
- **Top 10 Most Popular Items:** hot items such as daily necessities (e.g., Whole Milk, Other Vegetables, Soda, etc.) are in high demand and purchased with stability, making special recommendations unnecessary as consumers naturally prioritize these items when shopping.

The charts indicate that users typically buy 2 to 5 items, with stable daily transactions but a decrease at the end of the month. Recent user activity is high, and highlights popular everyday food items such as whole milk, other vegetables, and rolls/buns. (see Figures1, 2, and 3)

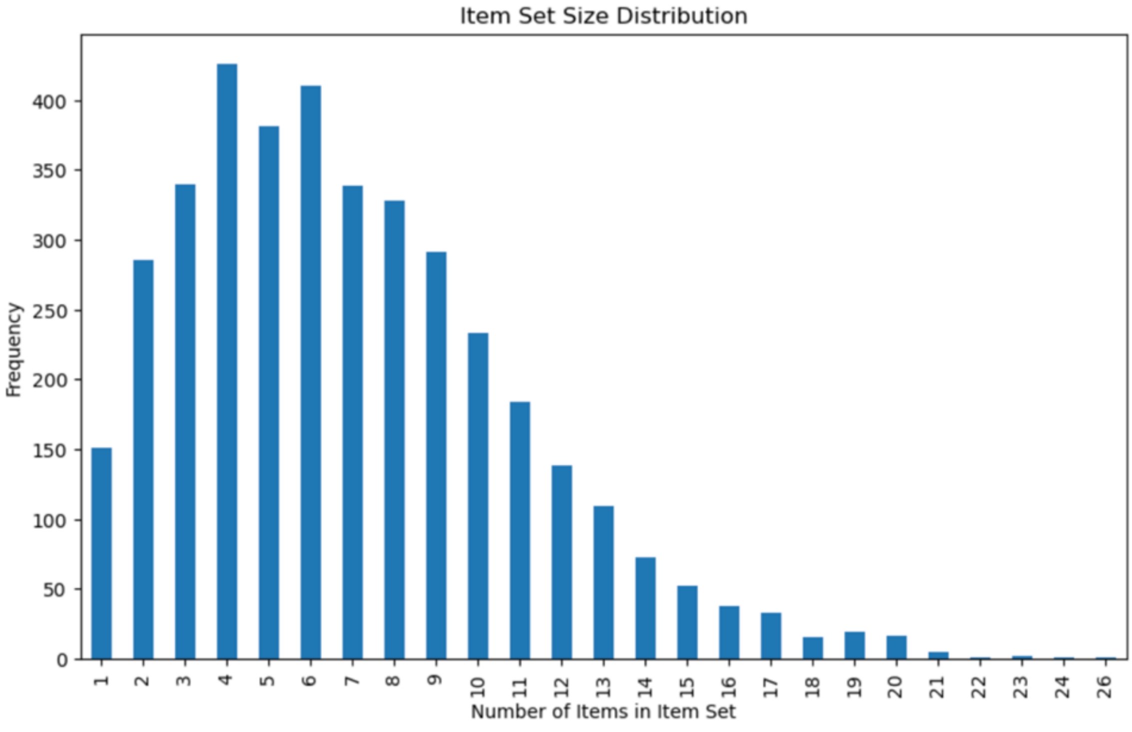

Figure 1: Item sets distribution

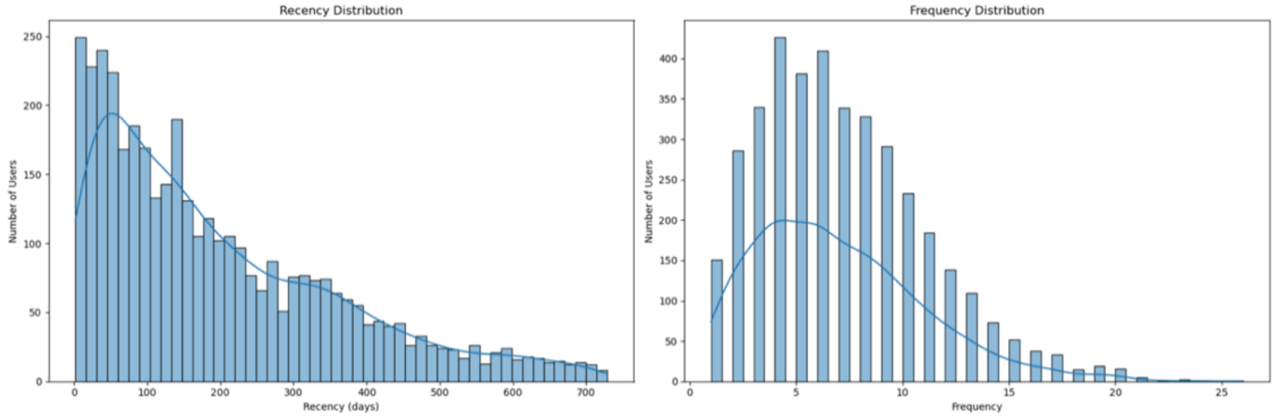

Figure 2 Recency and Frequency Distribution

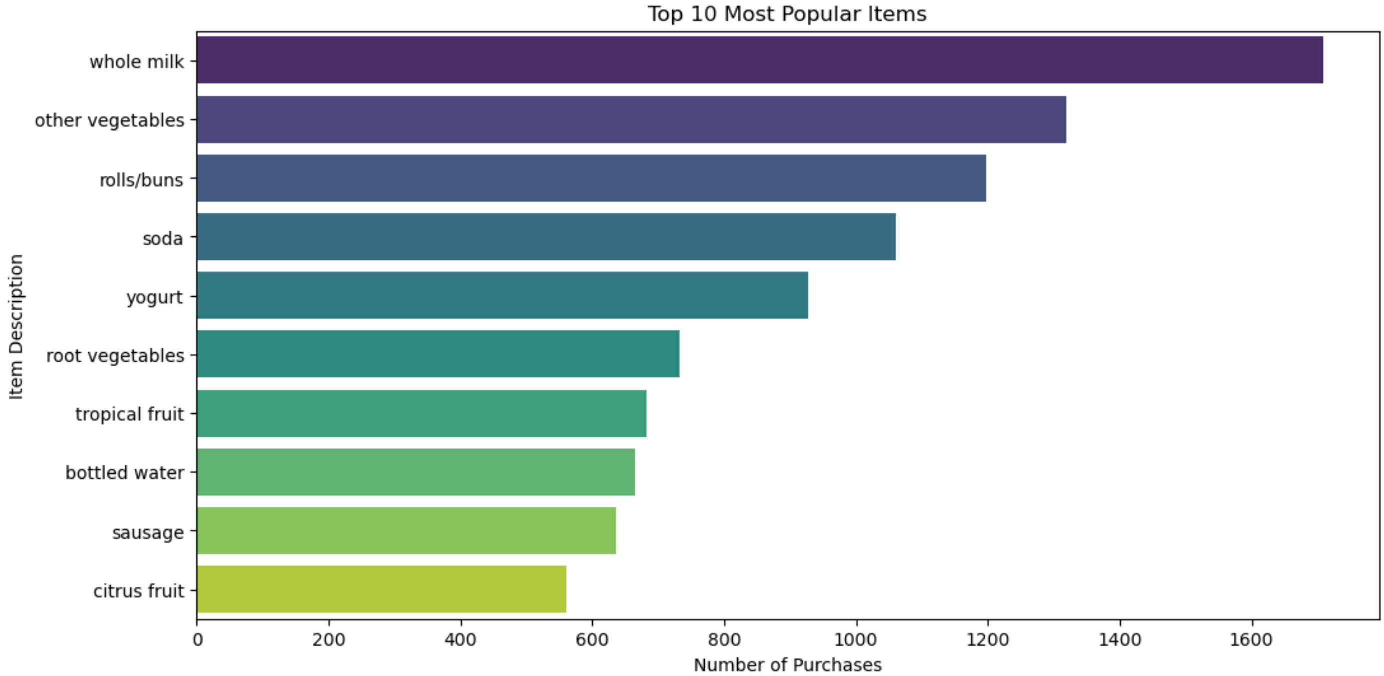

Figure 3 Top 10 Most Popular Items

3.2 Potential Problem and Solution

Based on the analysis, the potential problems include a decline in month-end purchases, a risk of user churn due to recent or low purchase frequency, and sparse data, which affects prediction accuracy. To address these issues, we will preprocess the data by consolidating all purchase records related to each user into transaction data. This data will be processed using frequent itemset and various recommendation algorithms.

3.3 Structure of Proposed System

The proposed system involves data processing (using detailed exploratory data analysis EDA and frequent pattern mining), recommendation analysis (association rules, k-means and cluster, and collaborative filtering), building and refining a recommendation engine.

4 Frequent Pattern Mining

To determine which items customers most often buy together in a basket, we need to perform frequent itemset mining on the data. We pre-process the data by checking for non-zero values, and then aggregate the data based on the frequency of user purchases to convert it into transaction data. Using Apriori and FP-growth algorithms, we generate frequent itemset from transactional data.

4.1 Apriori and FP-growth Algorithm

we present a comparative analysis of two algorithms for generating frequent itemset to determine the suitability for the grocery data set. While both algorithms are based on recursive search, they differ in performance.

For the Apriori algorithm, the pre-processed transactional data can be directly used as input. By setting the minimum support to 0.003, frequent itemset with support above this threshold are identified.
For the FP-Growth algorithm, the transactional data needs to be converted into a Boolean sparse matrix, with user IDs as matrix indices to associate data with corresponding users. The matrix represents whether a user has purchased a specific item, with rows representing users, columns representing items, True indicating a purchase, and False indicating no purchase. Using this input data and a minimum support of 0.005, frequent item purchases are identified.

The minimum support values were not set consistently because of the limitations of the Apriori algorithm. If the support was set to match the FP-Growth algorithm, a significant portion of the data would be filtered out, potentially due to the characteristics of the Apriori algorithm and the sparse distribution of the dataset. This highlights the advantage of the FP-Growth algorithm in handling large-scale and sparse datasets. (Han, J., Pei, J. and Yin, Y., 2000)

4.2 Association Rules

After generating frequent itemset, the itemset need to be utilized to generate association rules, which provide more specific insights into the frequency and relationships between item co-occurrences. The confidence and lift of each rule are calculated. By setting a minimum threshold for confidence, high-quality recommendations can be produced to help companies increase profits. Selecting appropriate support and confidence thresholds is crucial, as they directly impact the quality and quantity of the generated association rules.

However, the selection of support and confidence thresholds is an iterative process, and we conducted multiple experiments to find the optimal values. During the experiments, we found that when the confidence threshold was set to 0.8, both the Apriori and FP-Growth algorithms generated almost no rules, which is unfavorable for data analysis and decision support. After careful consideration, we standardized the confidence threshold to 0.1 for both algorithms to ensure a consistent comparison.

When using the Apriori algorithm, the confidence threshold of 0.1 ensured a reasonable number of high-quality rules were generated without significantly impacting computational efficiency, enabling the discovery of potentially valuable patterns. Similarly, for the FP-Growth algorithm, setting the confidence threshold to 0.1 allowed us to efficiently handle large-scale datasets while ensuring that the generated rules were of high quality. This unified threshold approach facilitated a more straightforward comparison between the two algorithms.

4.3 Evaluation Performance

Upon evaluating the time and space complexity, running time, and performance metrics of the Apriori algorithm and FP-Growth algorithm, we found that: 1. Regarding time and space complexity, the Apriori algorithm relies on recursive search with higher complexity, resulting in reduced efficiency and increased I/O overhead when handling large-scale datasets. (Agrawal, R. and Srikant, R., 1994) Conversely, the FP-Growth algorithm achieves superior efficiency by constructing FP-trees for frequent itemset mining but necessitates greater memory usage. (Han, J., Pei, J. and Yin, Y., 2000) 2. In terms of running time, processing one million transactions took 4867.69 seconds for the FP-Growth algorithm, longer than the 1407.06 seconds required by the Apriori algorithm. 3. From a performance standpoint, the FP-Growth algorithm surpasses the Apriori algorithm in average precision, average recall rate, and mean average precision values.

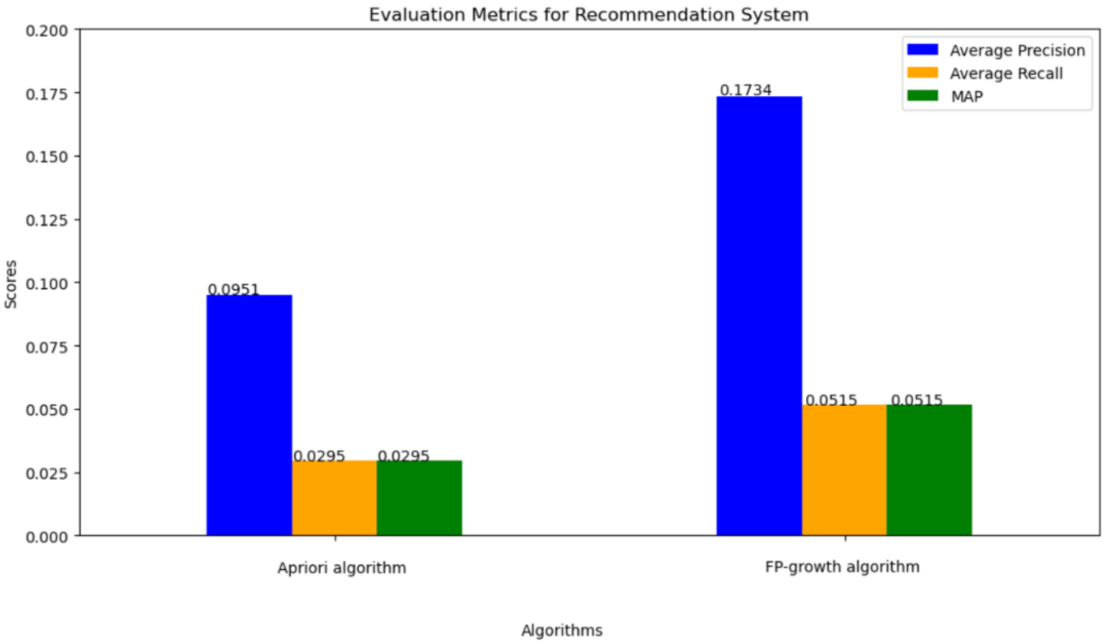

Figure 4 Evaluation Metrics for Association Rules

	Estimated Run Time for One Billion Transactions (seconds)
Apriori	1407.06
FP-growth	4867.69

Table 1 Estimated Run Time for One Billion Transactions (seconds)

In conclusion, for optimizing store benefits within a medical academic context, we recommend adopting the FP-Growth algorithm despite its need for additional hardware investment; its enhanced precision and recall capabilities will result in improved recommendation quality beneficial for augmenting customer satisfaction and sales revenue.

5 Collaborative Filtering

5.1 Method Description

Collaborative filtering is a method used to analyze user behavior by predicting a user's taste based on the preferences of similar users (Parvatikar & Joshi 2015). Collaborative filtering can be divided into two main types: user-user collaborative filtering and item-item collaborative filtering.

This method is scalable to large datasets and involves three main steps. Firstly, to address the sparsity drawback in collaborative filtering, cosine similarity is used to find similar users or similar items, as it effectively handles the sparsity in the user-item matrix by only calculating the similarity of non-zero elements. Secondly, it predicts the target user's rating for items they haven't purchased yet by using the ratings or purchase frequencies of similar users. Finally, by analyzing the purchasing frequencies and ratings from similar users, it recommends the top N items to the current customer.

Collaborative filtering faces two main drawbacks: sparsity, where even active users rate only a few items, leading to a limited number of ratings for popular items, and the cold start problem, where it is challenging to recommend items to new users who haven't rated any items yet. These issues make it difficult to generate reliable recommendations in the absence of sufficient user data (Parvatikar & Joshi 2015).
Based on the visual analysis of the dataset, we identified specific strategies to improve user-user collaborative filtering

- *Active Users Priority:* In collaborative filtering, prioritizing the data of active users in the last 100 days ensures that the purchasing behavior considered is more representative and up-to-date.
- *Segmented Modeling:* User data is processed in segments according to purchase frequency and amount to improve the accuracy and relevance of recommendations.

Therefore, we model high-frequency and low-frequency users separately using data from the last 100 days.

By implementing these strategies, we aim to enhance the performance of our collaborative filtering system, making it more responsive to recent purchasing trends and better tailored to different user segments.

5.2 Evaluation Metrics

In solving the supermarket recommendation problem, we choose MAP as the primary evaluation metric and Precision and Recall as the secondary reference metrics. The advantage of MAP is that it comprehensively evaluates the accuracy and order of the recommendation results and ensures that the relevant items are efficiently recommended to the users, which in turn improves the user satisfaction and actual purchase rate. The combination of precision and recall provides a more comprehensive evaluation of the performance of the recommender system: precision measures the relevance of the recommended items to ensure that the recommended items are indeed of interest to the user; recall measures the number of relevant items that the system can reach to ensure that the user is able to see as many relevant products as possible.

Ideally, a MAP value of 0.02 to 0.1 indicates that the recommendations are accurate and reasonably sorted; precision should be 0.2 to 0.6, indicating that most of the recommended items are relevant; and recall should be 0.2 to 0.5, indicating that the system is able to cover a larger number of relevant items. Through the comprehensive evaluation of these three indicators, we can ensure that the overall performance of the recommender system is excellent and the user experience is good.

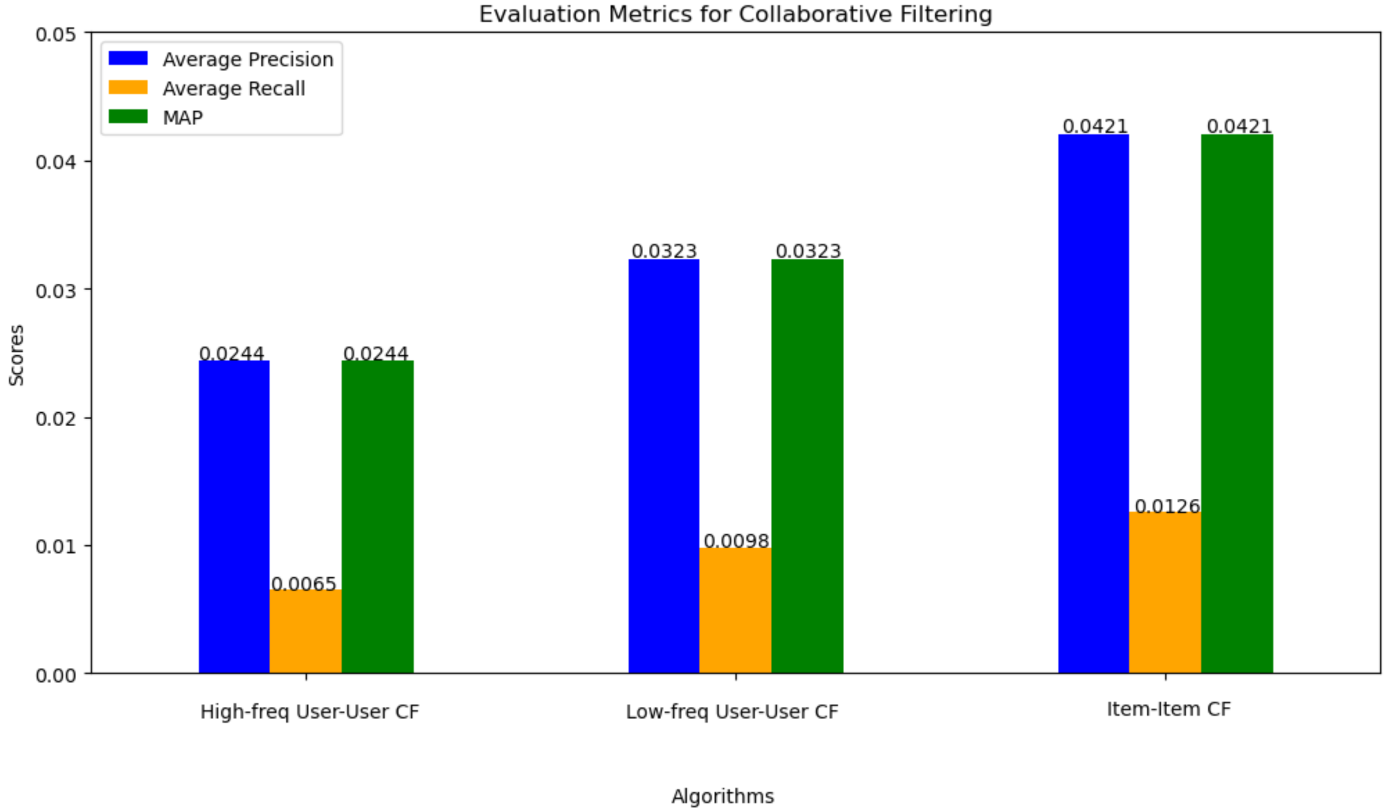

Figure 4 Evaluation Metrics for Collaborative Filtering

From the graphical analysis, we can see that Item-Item Collaborative Filtering is the best recommendation method. It scores the highest in three key aspects: precision (0.0421), recall (0.0126), and MAP (0.0421). This means it excels at recommending relevant items and showing users a variety of products, they might like.

In contrast, the Low-Frequency User-User Collaborative Filtering method, with a precision of 0.0323, recall of 0.0098, and MAP of 0.0323, performs moderately well but shows fewer relevant products. The High-Frequency User-User Collaborative Filtering method performs the worst, with a precision of 0.0244, recall of 0.0065, and MAP of 0.0244. This indicates it is less effective in recommending relevant products and displaying a wide range of items.

Therefore, we recommend using Item-Item Collaborative Filtering as it provides the most accurate and useful recommendations, improves user satisfaction, and potentially boosts sales.

6 Recommendation Methods from Frequent Patterns

6.1 Method Description

The paper explores a hybrid recommendation system - the combination of frequent patterns and collaborative filtering to enhance recommender systems (Yao et al., 2019 pp. 65-66). Frequent patterns mining identifies special relationships between variables in large datasets, providing patterns that highlight frequent co-occurrences. Collaborative filtering recommends items by finding similarities between users based on historical data. The synergy between these two methods aims to leverage the strengths of both to improve recommendation accuracy and relevance.

6.2 Method Execution

The integration of frequent patterns into collaborative filtering involves several key steps:

- *Data Visualization:* Begin by visualizing the frequent itemset data from Task 1 and the collaborative filtering (CF) table from Task 2. This visualization helps in observing the distribution and characteristics of the data, providing insights into underlying patterns.
- *Clustering with K-Means:* frequent item datasets based on Task1 are taken as input, apply clustering techniques, such as k-means, to identify groups of similar users based on their preferences and behaviors. This step helps in segmenting users into clusters with shared characteristics.
- *Combining with Collaborative Filtering:* The clusters are integrated with the CF table from Task 2. This involves analyzing the CF recommendations for users within the same cluster and combining these recommendations based on their ratings. By weighting items according to the ratings, further recommendations are then made for users within the same cluster, ensuring that highly rated items are prioritized for similar users.
- *Generating Recommendations Output:* Utilize the adapted collaborative filtering algorithm to generate recommendations for users. The output is a comprehensive list of recommended items that considers both primary and secondary recommendations:
- *Primary Recommendations:* These are generated through the traditional collaborative filtering approach, tailored to specific users based on their preferences.
- *Secondary Recommendations:* These are enhanced by considering highly rated items from other users within the same cluster, and providing additional relevant suggestions.

Through these steps, this method of generating recommendations can enhance the effectiveness of both frequent patterns and collaborative filtering (CF) approaches.

7 Discussion of Results

We evaluate the performance of the recommendation system using key metrics such as average accuracy (MAP), accuracy, and recall. We first integrated all purchase records into transaction data, and then used Apriori and FP-Growth algorithms for frequent item set mining. Clustering users based on their buying behavior to identify similar groups of users helps customize recommendations. The collaborative filtering method is used to calculate predictive scores and rank recommendations based on these scores.

7.1 Examples and Recommendations of Frequent Patterns

In this study, we categorized users as high-frequency and low-frequency users based on their purchase frequencies. To evaluate the effectiveness of the recommendation system, we randomly selected five high-frequency users as a sample and performed frequent pattern mining on both the training and test sets, recording the support and confidence values. The following table presents the frequent patterns and their corresponding support and confidence values for these five users on the training and test sets.

	User id	Frequent patterns (Train)	Support (Train)	Confidence (Train)	Frequent patterns (Test)	Support (Test)	Confidence (Test)
0	1146	'whole milk', 'bottled water', 'soda', 'berries'	0.0031	0.23	'frozen dessert', 'brown bread'	0.0014	0.13
1	1265	'chicken', 'pip fruit', 'tropical fruit'	0.0041	0.39	'bottled water', 'yogurt', 'butter'	0.0014	0.39
2	1743	'frankfurter', 'fruit/vegetable juice', 'brown...	0.0030	0.31	'bottled water', 'pastry', 'pip fruit'	0.0014	0.27
3	1929	'chicken', 'bottled water', 'butter'	0.0036	0.29	'fruit/vegetable juice', 'yogurt', 'pip fruit'	0.0011	0.44
4	2273	'bottled beer', 'other vegetables', 'rolls/bun...	0.0052	0.15	'finished products', 'butter'	0.0011	0.19

Table 2 Five examples of frequent patterns for recommendation system

Based on the above selected five examples of frequent patterns, we employed collaborative filtering and hybrid recommendation methods to suggest ten items. The results are as follows:

User id	Recommendations (CF)	Recommendations (Hybrid)
1146	dental care, cereals	candy, canned fish
1265	packaged fruit/vegetables, whisky	candy, canned beer
1743	canned beer, pudding powder	candy, whole milk
1929	candy, frozen fish	candy, frozen fish
2273	organic sausage, rice	candy, bottled beer

Table 3 Ten recommendations from different systems

The table shows a significant difference between the items recommended by the two recommendation methods. This is because these five users are grouped into a cluster in the hybrid recommendation system. Users within the same cluster are considered to share similar characteristics, and recommendations are made for these users based on the highest ratings within the cluster.

7.2 Results of Frequent Pattern Mining on Train and Test Datasets

By comparing support, confidence, and lift on both test and train datasets, we evaluated the performance differences between the Apriori and FP-Growth algorithms. The comparison shows that the FP-Growth algorithm has a higher support on the test dataset (0.007151) compared to the train dataset (0.005579). On the other hand, the Apriori algorithm shows a decrease in support from 0.003512 in the train dataset to 0.001150 in the test dataset. However, the confidence and lift for Apriori significantly improve.

This indicates that FP-Growth displays higher support values in the test dataset, suggesting its ability to discover frequent patterns in new data. Although its confidence and lift decrease, the algorithm is beneficial when identifying high-frequency patterns is prioritized over the reliability of these patterns.

Considering the need for high support in recommending frequently co-occurring products, the FP-Growth algorithm is recommended. Its ability to maintain higher support in new data indicates it can effectively identify frequent itemset commonly purchased by users, providing valuable insights for merchants in product decision-making and marketing strategies.

	FP-growth_TrainData	FP-growth_TestData	Apriori_TrainData	Apriori_TestData
support	0.005579	0.007151	0.003512	0.001150
confidence	0.677340	0.277566	0.857878	0.910000
lift	1.949934	1.431846	2.630857	7.237375

Table 4 The difference between FP-growth and Apriori algorithm on train and test datasets

7.3 Results of Recommendations on Training and Test Sets

According to the evaluation metrics in the chart, the FP-growth algorithm performs best with an average precision of 0.17, an average recall of 0.05, and an average accuracy (MAP) of 0.05. The Apriori algorithm follows closely with an average precision of 0.14, an average recall, and an average accuracy of 0.04. Collaborative filtering algorithms and clustering-based recommendation systems show average performance across various metrics, suitable for specific scenarios and needs. User-User CF is ideal for social networks or community platforms, Item-Item CF fits well for e-commerce platforms or large content libraries, while clustering-based recommendation systems are suited for scenarios with new users or cold-start problems, as well as applications with diverse and personalized requirements. Applying these algorithms in their respective scenarios can enhance user satisfaction and platform engagement.

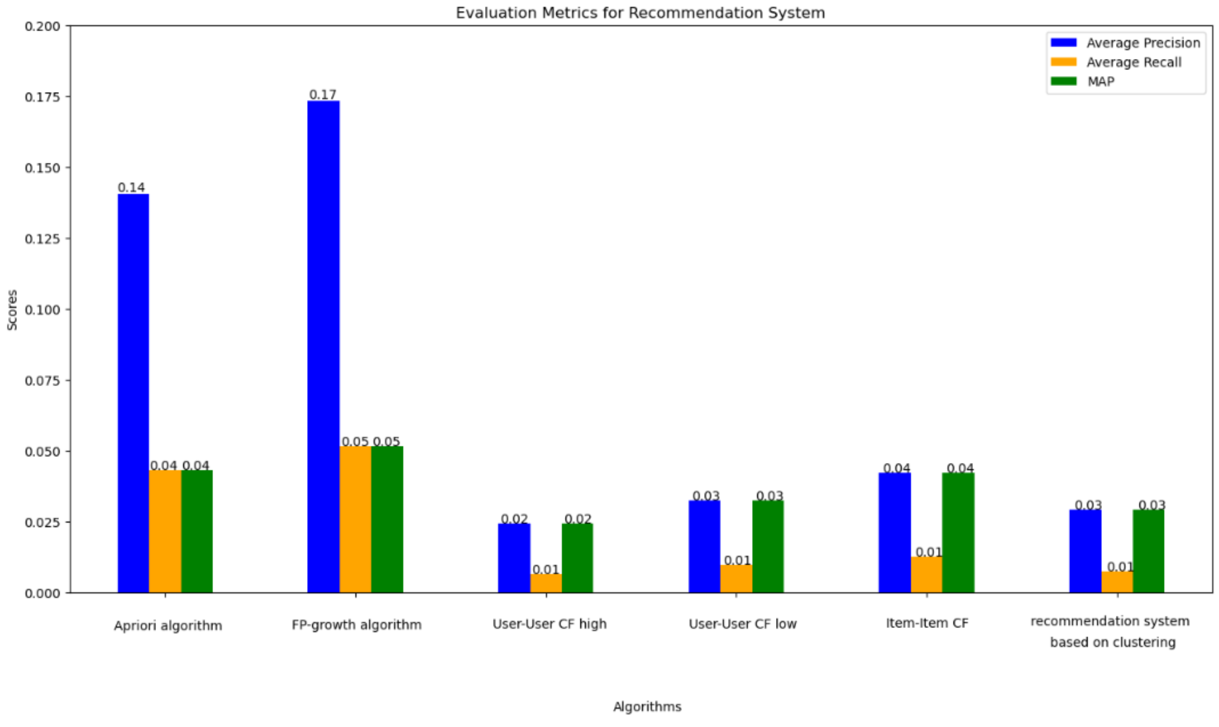

Figure 5 Evaluation Metrics for Recommendation System

7.4 Timing of the Recommendation System (estimate the dataset is scaled up to one million transactions)

7.4.1 Device Specifications

The experiment device is equipped with an Intel(R) Core (TM) i7-10510U CPU running at 1.80GHz with a boost up to 2.30GHz. It has 8.00 GB of installed RAM, of which 7.59 GB is usable. The system operates on a 64-bit operating system and features an x64-based processor.

7.4.2 Evaluation Time Summary

Based on the evaluation times shown in the chart, the time required for each algorithm to evaluate one million transactions on this device is as follows:

- User-User CF: 196,798.23 minutes
- Item-Item CF: 3,004.47 minutes
- Association rule based on Apriori Algorithm: 23.45 minutes
- Association rule based on FP-growth Algorithm: 81.13 minutes
- Clustering-Based Recommendation System: 49.83 minutes

In summary, the Apriori algorithm has the shortest evaluation time on this device, making it the most efficient. The FP-growth algorithm is next, suitable for scenarios requiring efficient processing and high recommendation quality. User-User CF and Item-Item CF algorithms take significantly longer, making them more appropriate for offline processing or small datasets. The clustering-based recommendation system has a moderate evaluation time, suitable for scenarios needing a balance between computational cost and recommendation quality.

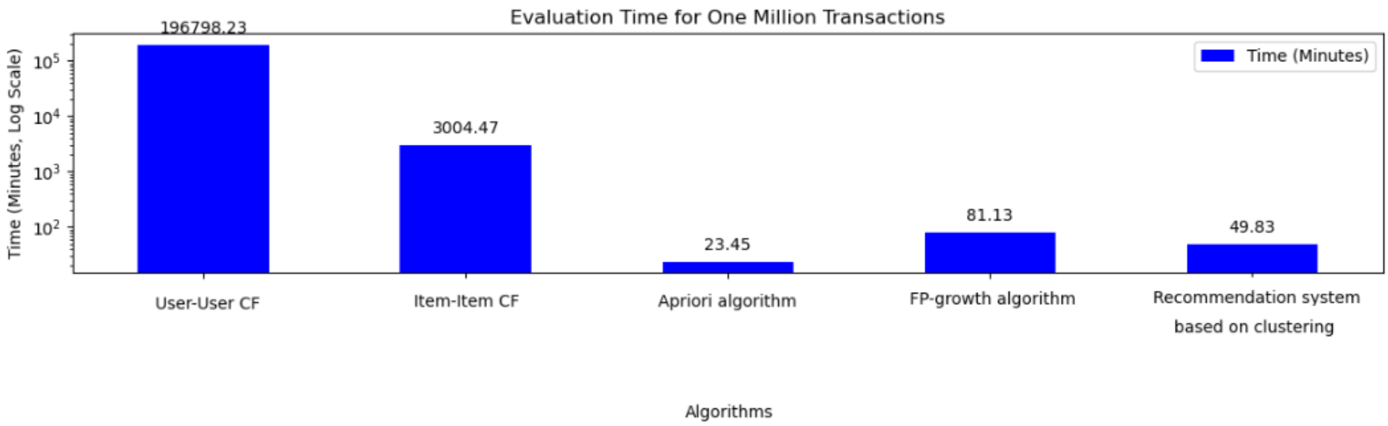

Figure 5 Evaluation Time for One Million Transactions

8 Conclusion and Recommendations

Method Recommendation:

We recommend using the association rule based on FP-growth algorithm for the recommendation system. It has the highest average precision (0.17), average recall (0.05), and average accuracy (0.05) among the evaluated methods, with a reasonable evaluation time (81.13 minutes for one million transactions). The FP-growth algorithm efficiently identifies and recommends items that customers are most likely to buy based on frequently bought item patterns, ensuring that the top recommendations are relevant and beneficial for both in-store displays and online shopping.

Scaling Considerations:

The association rule based on FP-growth algorithm is efficient and scalable, offering high-quality recommendations as the dataset grows without significant computational costs, making it suitable for large-scale and real-time applications. This scalability ensures that as the number of transactions increases, the system can continue to provide accurate and timely recommendations.

Future Improvements:

For future enhancements, consider implementing a hybrid recommendation model that combines FP-growth and collaborative filtering to leverage their strengths. This approach can provide more personalized recommendations, balancing the efficiency of pattern-based recommendations with the adaptability of user behavior analysis. Additionally, gathering more specific customer data will improve personalization, while integrating real-time data processing and advanced machine learning techniques will enhance recommendation accuracy and efficiency.

Benefits for the Company:

A robust recommendation system is crucial for optimizing product displays and online shopping experiences. Accurate and well-ordered recommendations can significantly impact sales by highlighting the most relevant products. For example, recommendations like "candy, canned beer" for user ID 1265 and "candy, frozen fish" for user ID 1929 show that hybrid recommendations can prioritize items that are more likely to be purchased together, enhancing the customer shopping experience both in-store and online.

9 Reflection

In this project, detailed exploratory data analysis (EDA) is first conducted through both user and item perspectives to understand the characteristics and high and low frequency patterns of the data. These insights will guide the data filtering and parameter setting of the recommendation algorithm. To improve the operational efficiency of the collaborative filtering (CF) algorithm, we cope with the time complexity problem by feature selection and reducing the dataset size, and use clustering methods to group similar users to improve the accuracy of recommendations.

The metrics of the recommender system include mean average precision (MAP), accuracy and recall. These metrics will be used to evaluate the performance of the recommendation system and optimize the algorithm based on the results. In terms of project management, we rationalize the order of tasks and unify variables and file names to improve the efficiency of the combined manuscript. In addition, consensus is established through preliminary discussions on data and modelling methods to reduce time wasted due to inconsistent understanding.

In the future, we will establish a complete logical framework from EDA to data processing, model debugging and result presentation, and formulate the best practices for big data mining to improve the overall efficiency and effectiveness of the project.

References

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules. Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), pp. 487-499.

Han, J., Pei, J. and Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 1-12.

Jannach, D., Fankhauser, P. and Zanker, M., 2020. A Survey of Recommender Systems in the Food Domain. Journal of Food Engineering, pp. 88-98.

Kotu, V 2019, Data science: concepts and practice, B Deshpande (ed.), Second edition., Morgan Kaufmann Publishers an imprint of Elsevier, Cambridge, MA, p. 1 online resource (570 pages).

Muthuperumal, S., Titus, P. & Venkatachalapathy, M, 2020. An algorithmic approach to solve unbalanced triangular fuzzy transportation problems, p. 18690.

Neysiani, B.S., Soltani, N., Mofidi, R., and Nadimi-Shahraki, M.H., 2019, Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm. I.J. Information Technology and Computer Science, pp. 48-55.

Parvatikar, S & 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.

Yao, Liming & Xu, Zhongwen & Zhou, Xiaoyang & Lev, Benjamin. (2019). Synergies Between Association Rules and Collaborative Filtering in Recommender System: An Application to Auto Industry. 10.1007/978-3-319-95651-0_5.

Contribution Appendices

Group Number	Contribution	Details
Zhige Zhang	Code Development	Responsible for implementing frequent patterns mining and association rules.
	Report Writing	Contributed to writing the 1th, 2th, 3th, 4th, 7th sections of the report.
	Other	Responsible for outlining the entire structure of report and document formatting.
Tao Hui	Code Development	Responsible for implementing the collaborative filtering algorithm.
	Report Writing	Contributed to writing the 5th, 7th, 8th, 9th sections of the report.
	Other	Responsible for integrating group code and outputting metrics and time results.
Lanxi Zhang	Code Development	Responsible for implementing the hybrid (clustering) algorithm.
	Report Writing	Contributed to writing the 6th, 7th sections of the report.
	Literature Review	Read the provided references and conduct additional research.