Concordia University

Assignment 2

Group 74

Ryan Li 40214839

Tuan Anh Pham 40213926

Mustafa Sameem 40190889

Antoine Cantin 40211205

SOEN 471

Big Data Analytics

Dr. Reza Mirsalari

April 4, 2025

## INTRODUCTION

E-commerce's exponential growth has created a constantly evolving competitive ground where companies compete for new customers' attention as well as to retain returning customers. To achieve these increases in sales, companies often collect transaction data from their customers. This data is used to generate personalized recommendations and intelligent product suggestions, and is essential for modern business success. These insights allow for the company to maximize their variables to drive profit by providing a better customer experience and driving customer engagement. For example, grouping items that tend to be bought together from other customers.

Some of the ways e-commerce platforms achieve this is by having a recommender system and customer purchase analytics through pattern mining. In this project, we will be using user-based collaborative filtering and apriori association rule mining to implement our recommender system.

Collaborative filtering is a technique that suggests items to users based on the preference and behavior of other users. In user-based collaborative filtering, rather than finding similar items, we find similar users. Advantages of this type of filtering is that it does not require product information and it learns user behavior. However, this approach does come with drawbacks, cold start problems and scalability, discussed later.

Association rule mining is a technique that discovers relationships and patterns between items. The Apriori Algorithm is a method to find frequent groups of items that often appear together and generates rules from them. Rules are based on 3 metrics: support, confidence, and lift. Support indicating the frequency of the rule in the dataset, confidence measuring the probability of items Y given items X, and Lift measuring how much more likely items Y when items X, compared to items Y independently.

With these two approaches, we aim to build a recommender system that addresses both individual user preferences and broader purchasing patterns. Business challenges we aim to solve are providing personalized product recommendations, discovering hidden behavior patterns, addressing challenges, and evaluating the system's results. This report details our methodology, results, insights, and challenges of our recommender system.

## METHODOLOGY

We worked with 2 csv datasets files. The “ecommerce_user_data.csv” provided a user’s rating for a product through UserID and ProductID. The other csv file, “product_details.csv”, provided product details.

Before starting recommendations, the data must be preprocessed. First, the csv files were loaded, and missing values and duplicates were checked. Using both datasets, we created a user-item matrix, filling missing ratings with a rating of 0. User ratings are subject and scales may differ from one another. To counteract this, we used mean-centering by subtracting each user's mean rating from their ratings. The user-item matrix is then split 80% training and 20% testing for metric evaluation.

Our collaborative filtering is user-based. Using cosine similarity, we computed a user-user similarity matrix (ranges 0 - 1, 1 being the most similar). For further analysis, a heatmap is created based on user similarity. Using the similarity matrix, we found k most similar users to the target user. We then found items that the target user hasn’t rated yet, and used the k similar users to calculate a weighted neighbor rating prediction, by their similarity scores, for these unrated items. The target user is excluded in these calculations. To evaluate recommendation results, precision at k, recall at k, and item coverage metric are evaluated. Precision and recall is done through hiding some of the target user’s ratings in a copy of the training matrix. Generated recommendations are checked to see if they recommended the items that the user had rated highly (at a certain threshold) in the test set. In other words, we check if the system rediscovers items that the user has rated highly. Recommended items are checked for item coverage. A grid search was used to find the best values for parameters: threshold, number of recommendations, and number of top similar users.

User data, that contains UserID and the ProductID of the items they have rated, are converted to a transaction list of baskets. To identify frequent itemsets, we applied the Apriori algorithm to these user baskets with a support threshold of 5%. With these frequent itemsets, association rules are generated with a minimum confidence threshold of 50%. Support, confidence, and lift are calculated and displayed. For further analysis, top frequent itemsets are graphed by support.

## RESULTS

When preprocessing the data, we found some important characteristics. There were 724 user-product interactions across 50 users and 100 products. We got a high matrix sparsity of 85.52% where most users have not interacted with most products. The distribution was across 6 product categories mainly; such as Books, Electronics, Clothing, Toys, Beauty and Home. The ratings scale of 1-5 with a slightly skewed distribution toward lower ratings.

### Collaborative Filtering Performance

Our collaborative filtering grid search evaluated different configurations for the number of recommendations (k_recs), neighborhood size (top_k_users), and relevance threshold:

| Index | k_recs | top_k_users | threshold | avg_precision | avg_recall | avg_diversity | coverage |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 10 | 3 | 0.06 | 0.100000 | 0 | 65.0 |
| 4 | 5 | 20 | 3 | 0.06 | 0.100000 | 0 | 65.0 |
| 0 | 5 | 5 | 3 | 0.04 | 0.075000 | 0 | 61.0 |
| 3 | 5 | 10 | 4 | 0.04 | 0.116667 | 0 | 65.0 |
| 5 | 5 | 20 | 4 | 0.04 | 0.116667 | 0 | 65.0 |

The best configuration for precision was k_recs=5, top_k_users=10, and threshold=3, achieving a precision of 6% and recall of 10%. These metrics appear low at first glance but they are comparable to industry standards for highly sparse datasets in which ours had a sparsity of 85.52%.

When evaluating with a higher relevance threshold of 4, the precision and recall values decreased further, which indicates that many users have a broader range of "acceptable" products (rated 3+) rather than many "excellent" products (rated 4+).

The similarity matrix visualization revealed clusters of users with similar tastes, particularly around certain product categories. This clustering behavior validates the fundamental assumption of collaborative filtering that similar users tend to like similar products.

**Association Rule Mining**

| antecedents_list | consequents_list | support | confidence | lift |
|---|---|---|---|---|
| [P0002] | [P0080] | 0.04 | 0.666667 | 16.666667 |
| [P0080] | [P0002] | 0.04 | 1.000000 | 16.666667 |
| [P0049] | [P0003] | 0.04 | 0.400000 | 6.666667 |
| [P0003] | [P0049] | 0.04 | 0.666667 | 6.666667 |
| [P0053] | [P0003] | 0.04 | 1.000000 | 16.666667 |
| [P0003] | [P0053] | 0.04 | 0.666667 | 16.666667 |
| [P0003] | [P0070] | 0.04 | 0.666667 | 3.333333 |
| [P0005] | [P0070] | 0.04 | 0.500000 | 2.500000 |
| [P0006] | [P0059] | 0.04 | 0.666667 | 11.111111 |
| [P0059] | [P0006] | 0.04 | 0.666667 | 11.111111 |

The 0.04 Support shows that items are purchased together 4% of the time. This can be significant for a large dataset.

The Confidence values reflect the predicted probability of  the consequent item given that the antecedent item was already purchased (As seen used in Apriori and Association Rules analysis several times). For example:

(P0080 and P0002) and (P0053 andP0003) are brought together 100% of the time because of a confidence of 1.0.
In association rules, the Lift metric indicates how much more likely the consequent item to be purchased when the antecedent is purchased itself, compared to often an arbitrary chance. (Quantitivaly: A lift > 1 indicates a positive correlation)

**Top 5 Recommendations for user U001**

| ProductID | ProductName | Category | PredictedRating |
|---|---|---|---|
| P0033 | Toys Item 33 | Books | 1.996841 |
| P0070 | Beauty Item 70 | Toys | 1.914087 |
| P0005 | Home Item 5 | Toys | 1.752683 |
| P0009 | Clothing Item 9 | Books | 1.359873 |
| P0079 | Home Item 79 | Electronics | 1.193062 |

The recommendations show cross-category recommendation, suggesting that the user has interest in many different types of items. They have a strong preference for toys and books. In the user similarity matrix, user U001 shows a higher similarity score for users who have higher ratings on the toy category. Most predicted ratings are low which follows in line with the overall low similarity scores from sparse data.

**INSIGHTS**

## Impact of Data Sparsity

The high sparsity in our dataset (85.52%) reveals many challenges for e-commerce recommendation systems. It represents a big obstacle where most potential user-item interactions are unrecorded which presents several issues.

### Cold Start Problem

This is present because new users with minimal interactions received recommendations driven by overall popularity rather than personalized preferences. Also, niche products with few ratings remained invisible in our recommendation outputs even though they could have been potential perfect matches for some users. This creates a bad reinforcement cycle where popular items become more recommended and new items struggle to become visible.

### Reliability and Neighborhood quality

The limited data points per user creates reliability challenges. For a lot of users, recommendations were based on a few shared interactions with similar users, which put into question the statistical significance. We observed that confidence intervals around our predicted ratings were quite wide, which indicates high uncertainty. This undermines user trust when the recommendations aren't on point. Our similarity matrices show uneven neighborhood neighborhood quality across the users. Some users connect strongly with multiple similar users, and others have no neighbors with low scores. This creates inconsistent recommendation quality and particularly disadvantages users with atypical preferences.

### Mitigation Strategies

To address the challenges we recommend implementing targeted onboarding questions with hybrid models that leverage product metadata during cold start. Matrix factorization techniques could identify inactive relationships when the data is sparse, while content based filtering provides a reliable fallback.

## Product Association Patterns

### Cross-Category Relationships

The Electronics and Toys connections showed strong asymmetric associations with Electronics purchases frequently leading to Toy purchases; which suggests households with children buying patterns. Books demonstrated weaker cross category relationships, indicating genre-based content filtering might be more effective. Home and cloting items showed a seasonal correlation.

### Business Applications and Challenges

These patterns enable strategic opportunities with product bundling, cart based recommendations and targeted marketing. But this implementation requires navigating through interpretation challenges because rules with high lift but low support indicate strong but rare associations which makes us determinemetrics should guide decisions. The balance between statistical significance and business relevance always remains a challenge because practice is much different than theory.

## CHALLENGES

We encountered several technical challenges throughout the implementation process:

- Parameter tuning was one of the main challenges as our system depended heavily on values such as neighborhood size (k), similarity thresholds, and the number of recommendations (N). Choosing inappropriate values led to poor-quality recommendations or inefficient computations. We used grid search to experiment with different combinations, but this method was computationaly expensive, especially with larger datasets. Each new configuration required re-running the model, which significantly increased training time and limited our ability to iterate quickly.

- Evaluating the quality of recommendations was also difficult. We used common metrics like Precision@K and RecallK to assessperformance, but these metrics assume binary relevance and do not capture the full picture of user preferences. For example, a user might find a recommended item somewhat useful but not click on it, which would still be treated as an incorrect prediction. Also, since our evaluation was done offline using historical data, it did not reflect how users would react to the recommendations in a live setting. This makes it hard to measure real user satisfaction or engagement.

- Scalability became a concern as the dataset size increased. User-based collaborative filtering has a time complexity of $O(n^2)$ where n is the number of users. This means that as more users are added, the time to compute the similarity matrix grows quickly. For example, calculating similarities for 5,000 users needed 25 million comparisons, which made the process slow and more often than not, when important it is memory-intensive (depends on systems). This issue was especially noticeable when using Pearson correlation, which needs mean-centering and can’t be approximated as easily as simpler similarity metrics like cosine similarity.

## CONCEPTUAL QUESTIONS

**1. How does the sparsity of the data affect your recommender system's performance?**

Data sparsity greatly affects our recommender system's performance. In our dataset, most user-item interactions are absent which results in a high sparsity level of 85.52%. Not every product has received a rating from every user so it becomes a challenge to identify accurate patterns in user preferences. Ultimately, the lack of data decreases the accuracy of our user-based collaborative filtering. Furthermore, data sparsity worsens the cold-start problem as new users have few rating data and this leads to less personalized recommendations. Items with few ratings are also less likely to be recommended which create a bias towards items that have been rated more often. Finally, data sparsity affects our evaluation metrics such as Precision@K and Recall@K since those evaluations were done on a limited amount of data, making the metrics not as reliable as if the data sparsity level was lower. Having high data sparsity is problematic for cosine similarity because missing values are represented as 0. It can lead to misleading similarity scores. We would interpret that 0 in the user item matrix means that a user hasn't rated an item but cosine similarity considers all numerical values, including 0. As a result, users who haven't rated the same set of items might seem like they are similar. This artificially inflates the similarity score and doesn't reflect true user preferences.

**2. What kinds of product bundles were discovered in the association rules?**

We were able to identify notable product bundles in the association rules:

- There was a strong association between electronics (P0000) and toys (P0008) with a confidence level of 56.3% which probably indicates parents burning electronics devices with toys for their children or tech enthusiasts who have children.
- Another strong association was between home items (P0017) and clothing items (P0025). Customers who buy home items were more likely to also buy clothing items and this suggests they might be buying things for themselves and their homes that match the same style they like.
- Beauty product combinations such as {P0070} → {P0072} shows that customers who buy one beauty product tend to purchase complementary items, potentially as part of their inventory of makeup items..
- Some of the interesting product bundles were made up of items from different categories that wouldn't usually go together as these combinations had high lift values, meaning they appeared together much more often than we would expect by chance. For example, the bundle {P0052, P0064} → {P0076} had a lift of 4.82. This means that when someone buys P0052 and P0064, they are almost five times more likely to also buy P0076, compared to random purchases. These patterns are important because they show hidden connections between products and businesses can use this information to suggest related items and improve their cross-selling strategies.

**3. What improvements would you recommend for a real e-commerce system using your approach?**

On the technical side, we recommend using hybrid recommendation systems that combine collaborative filtering with content-based methods as this helps fix cold-start problems especially for new users and items with few ratings. For example, a user who hasn't rated any products could still get suggestions based on product descriptions or images of items they clicked on. We also suggest replacing basic similarity measures like cosine similarity with more advanced ones such as Pearson correction that is better at capturing the relationship between users. In this case, cosine similarity doesn't consider the actual rating but only the angle between rating vectors, which can be misleading when users have different rating scales. Pearson correlation, on the other hand, adjusts for these individual rating biases by measuring the linear relationship between centered ratings. This can lead to more accurate predictions, especially in systems with diverse user behavior. Similarity matrices should be updated incrementally when new user actions happen instead of waiting for batch updates to make the system more responsive. It will be especially useful for users who frequently browse or add items into their cart.

On the business side, recommendations can be more relevant if contextual information such as users browsing history, cart contents or the day are included. For example, a user browsing winter jackets in the evening could recommend winter boots. Recommendations should also align with business goals like clearing some inventory or promoting high-margins products. For example, the system could be adjusted to recommend items that are overstocked or more profitable even if they are not the best match to user preferences. Giving users an explanation for why they receive certain recommendations will increase their trust and engagement. It helps them understand the reasoning behind the suggestion, making it feel more personalized and less random. At the same time, giving users the ability to provide feedback helps improve the performance of the recommender system such as marking irrelevant recommendations. The algorithm can adapt based on that feedback loop for user preferences over time, leading to more accurate and relevant suggestions. Without them, the system risks continuing to recommend items that don't match with user preferences which will reduce their overall satisfaction.

# APPENDIX

**SmartCart Recommendation System Results**

**Top Configurations by Precision**

| Index | k_recs | top_k_users | threshold | avg_precision | avg_recall | avg_diversity | coverage |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 10 | 3 | 0.06 | 0.100000 | 0 | 65.0 |
| 4 | 5 | 20 | 3 | 0.06 | 0.100000 | 0 | 65.0 |
| 0 | 5 | 5 | 3 | 0.04 | 0.075000 | 0 | 61.0 |
| 3 | 5 | 10 | 4 | 0.04 | 0.116667 | 0 | 65.0 |
| 5 | 5 | 20 | 4 | 0.04 | 0.116667 | 0 | 65.0 |

**Top Configurations by Recall**

| Index | k_recs | top_k_users | threshold | avg_precision | avg_recall | avg_diversity | coverage |
|---|---|---|---|---|---|---|---|
| 16 | 15 | 20 | 3 | 0.03 | 0.191667 | 0 | 93.0 |
| 14 | 15 | 10 | 3 | 0.03 | 0.191667 | 0 | 92.0 |
| 17 | 15 | 20 | 4 | 0.02 | 0.166667 | 0 | 93.0 |
| 15 | 15 | 10 | 4 | 0.02 | 0.166667 | 0 | 92.0 |
| 11 | 10 | 20 | 4 | 0.03 | 0.166667 | 0 | 85.0 |

**Best parameters selected:** k_recs=5, top_k=10

**Detailed Recommendations for User U001**

| ProductID | ProductName | Category | PredictedRating |
|---|---|---|---|
| P0033 | Toys Item 33 | Books | 1.996841 |
| P0070 | Beauty Item 70 | Toys | 1.914087 |
| P0005 | Home Item 5 | Toys | 1.752683 |
| P0009 | Clothing Item 9 | Books | 1.359873 |
| P0079 | Home Item 79 | Electronics | 1.193062 |

**Sample Frequent Itemsets**

| itemsets_list | support |
|---|---|
| [P0000] | 0.06 |
| [P0001] | 0.04 |
| [P0002] | 0.06 |
| [P0003] | 0.06 |
| [P0004] | 0.06 |
| [P0005] | 0.08 |
| [P0006] | 0.06 |

| | |
|---|---|
| [P0007] | 0.04 |
| [P0008] | 0.12 |
| [P0009] | 0.06 |

**Sample Association Rules**

| antecedents_list | consequents_list | support | confidence | lift |
|---|---|---|---|---|
| [P0002] | [P0080] | 0.04 | 0.666667 | 16.666667 |
| [P0080] | [P0002] | 0.04 | 1.000000 | 16.666667 |
| [P0049] | [P0003] | 0.04 | 0.400000 | 6.666667 |
| [P0003] | [P0049] | 0.04 | 0.666667 | 6.666667 |
| [P0053] | [P0003] | 0.04 | 1.000000 | 16.666667 |
| [P0003] | [P0053] | 0.04 | 0.666667 | 16.666667 |
| [P0003] | [P0070] | 0.04 | 0.666667 | 3.333333 |
| [P0005] | [P0070] | 0.04 | 0.500000 | 2.500000 |
| [P0006] | [P0059] | 0.04 | 0.666667 | 11.111111 |
| [P0059] | [P0006] | 0.04 | 0.666667 | 11.111111 |

**Top Association Rules Details**

| Antecedents | Consequents | Support | Confidence | Lift |
|---|---|---|---|---|
| Beauty Item 70 (P0070) & Clothing Item 3 (P0003) | Electronics Item 53 (P0053) | 0.04 | 1.000000 | 25.000000 |
| Electronics Item 53 (P0053) | Beauty Item 70 (P0070) & Clothing Item 3 (P0003) | 0.04 | 1.000000 | 25.000000 |
| Books Item 2 (P0002) | Home Item 80 (P0080) | 0.04 | 0.666667 | 16.666667 |
| Home Item 80 (P0080) | Books Item 2 (P0002) | 0.04 | 1.000000 | 16.666667 |
| Electronics Item 53 (P0053) | Clothing Item 3 (P0003) | 0.04 | 1.000000 | 16.666667 |

**Top Category Relationships in Association Rules**

| Antecedent Category | Consequent Category | Count | Avg Lift |
|---|---|---|---|
| Clothing | Electronics | 3 | 19.444444 |
| Electronics | Clothing | 5 | 13.190476 |
| Beauty | Beauty | 2 | 12.500000 |
| Clothing | Books | 7 | 12.500000 |
| Toys | Clothing | 3 | 12.500000 |

**Enhanced Recommendations for User U000**

| ProductID | ProductName | Category | PredictedRating | Source |
|---|---|---|---|---|
| P0099 | Clothing Item 99 | Beauty | 1.058668 | Collaborative Filtering |
| P0091 | Clothing Item 91 | Electronics | 0.937285 | Collaborative Filtering |
| P0037 | Books Item 37 | Beauty | 0.803370 | Collaborative Filtering |
| P0049 | Toys Item 49 | Beauty | 0.802202 | Collaborative Filtering |
| P0041 | Books Item 41 | Home | 0.746753 | Collaborative Filtering |
| P0003 | Clothing Item 3 | Electronics | 3.000000 | Association Rules |
| P0030 | Books Item 30 | Books | 3.000000 | Association Rules |
| P0051 | Home Item 51 | Clothing | 3.000000 | Association Rules |

**Top Recommended Products**

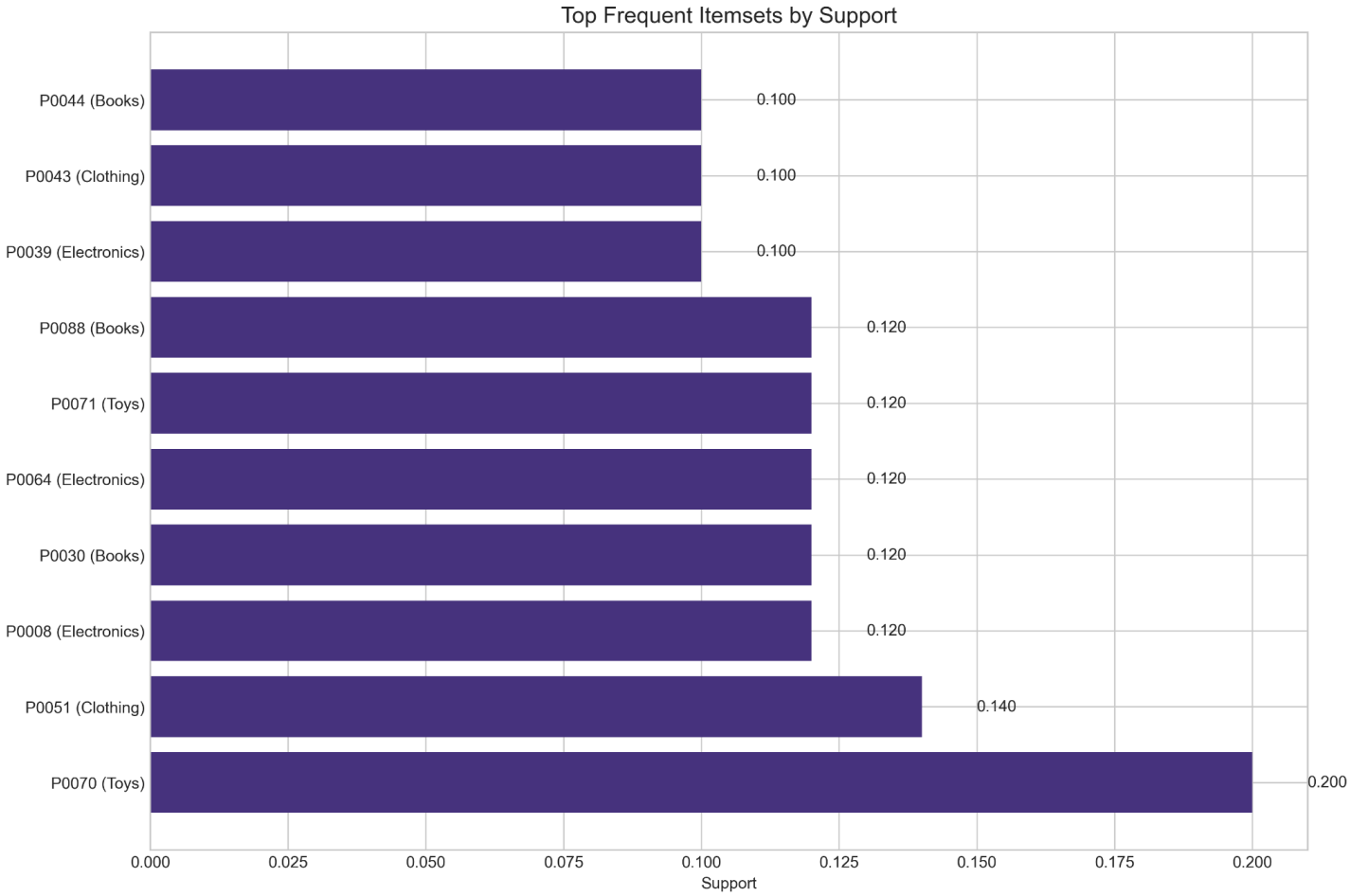
| ProductID | Count |
|---|---|
| P0003 | 24 |
| P0070 | 22 |
| P0051 | 19 |
| P0041 | 14 |
| P0030 | 14 |
| P0008 | 13 |
| P0077 | 10 |
| P0060 | 10 |
| P0033 | 9 |
| P0089 | 9 |

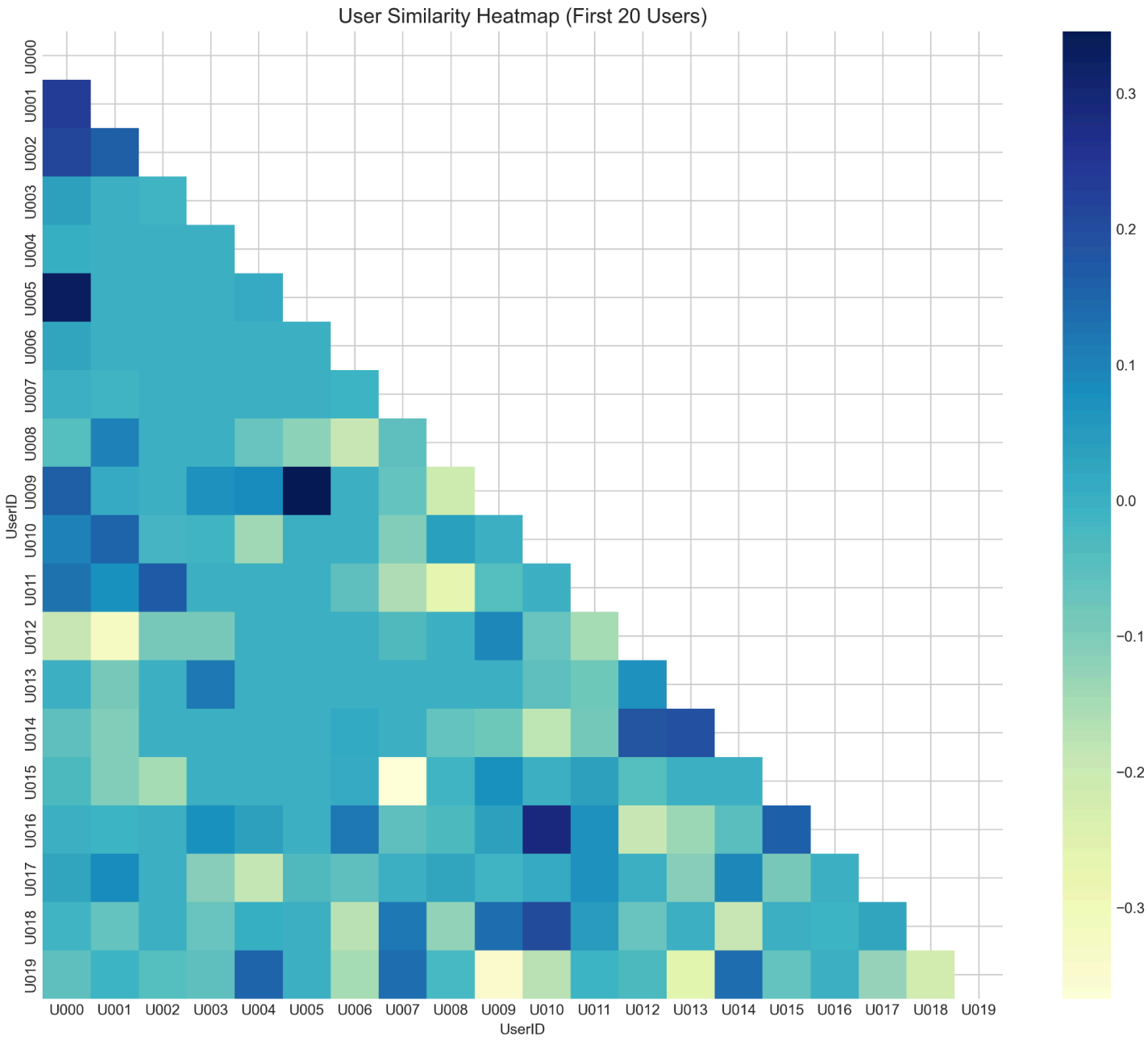
**Recommended Categories Distribution**

| Category | Count |
|---|---|
| Electronics | 112 |
| Books | 77 |
| Toys | 69 |
| Clothing | 62 |
| Beauty | 40 |
| Home | 34 |

**Performance by User Activity Level**

| Activity Level | Average Precision | Average Recall | Number of Users |
|---|---|---|---|
| Low | 0.02 | 0.133333 | 17 |
| Medium | 0.03 | 0.150000 | 17 |
| High | 0.02 | 0.150000 | 16 |


Top Frequent Itemsets by Support
P0044 (Books)
P0043 (Clothing)
P0039 (Electronics)
P0088 (Books)
P0071 (Toys)
P0064 (Electronics)
P0030 (Books)
P0008 (Electronics)
P0051 (Clothing)
P0070 (Toys)
0.100
0.100
0.100
0.120
0.120
0.120
0.120
0.120
0.140
0.200
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Support


User Similarity Heatmap (First 20 Users)
UserID
U000
U001
U002
U003
U004
U005
U006
U007
U008
U009
U010
U011
U012
U013
U014
U015
U016
U017
U018
U019
UserID
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3