

Assignment 3: Report

Sajja Patel - 2021101107, Bhumika Joshi - 2022121006

27 September 2024

1 Overview

2 Data Processing

The first step in the assignment involved preprocessing the dataset to ensure that only relevant data was used for further analysis.

2.1 Transactional Data Formation

To begin, the dataset containing user ratings of movies was filtered to create a transactional dataset. This dataset consisted of entries in the form $\langle \text{user id, movies rated above 2} \rangle$, which included only those users who rated movies higher than 2. The rationale for this filtering step was to focus on movies that users showed a relatively positive interest in, thereby excluding lower ratings that might introduce noise into the analysis. Additionally, users who had rated fewer than 10 movies were excluded from the dataset. This was done to ensure that each user had sufficient data to contribute meaningfully to the analysis, providing a more robust foundation for training and testing.

2.2 Splitting the Data

Once the filtered dataset was prepared, it was divided into two subsets: a training set and a test set. The dataset was split using an 80-20 ratio, where 80% of the data was used for training and 20% for testing. From the test set, 20% of the movies watched by each user were removed, and these removed entries formed the final test set. This method simulated real-world scenarios where a portion of the user's preferences would remain unknown, allowing for an accurate evaluation of the model's ability to predict unseen ratings. This preprocessing ensured that the dataset was clean, balanced, and ready for model training and testing, laying a solid groundwork for the subsequent analysis.

3 Association Rule Mining

After preprocessing the data, the next step was to perform association rule mining to uncover relationships between different movies based on user ratings.

3.1 Building Train and Test Sets

The first part of this phase involved building the training and test sets as dictionaries, where each key represented a user and the corresponding value was a list of movies they had rated. This dictionary structure allowed for efficient lookup and manipulation of user-movie interactions.

In the training set, the goal was to extract a unique list of all movies rated by users, forming the basis for frequent itemset generation. This list of unique movies was crucial for generating the candidates used in association rule mining.

3.2 Frequent Itemset Generation

Frequent itemsets were generated by calculating the support for sets of movies, which is the proportion of users who rated a particular combination of movies. Initially, frequent itemsets of size 2 were created by pairing all unique movies from the training set. For larger itemsets, candidate sets were formed by merging smaller frequent itemsets. The minimum support threshold (*minsup*) was applied to filter out infrequent itemsets, ensuring that only combinations of movies with sufficient user interest were considered for further analysis.

This process allowed us to systematically explore various combinations of movies to find those that frequently appeared together in users' watchlists, which are the core of association rule mining.

3.3 Association Rule Generation

Once the frequent itemsets were generated, association rules were derived from them. An association rule is of the form "if a user watches movie A, they are likely to watch movie B." For each frequent itemset, the confidence of a rule was calculated by dividing the support of the entire itemset by the support of its antecedent (a subset of the itemset). Only rules that met the minimum confidence threshold (*minconf*) were considered valid and included in the final set of rules.

These rules provide valuable insights into the relationships between movies and enable predictive modeling, such as recommending movies to users based on their prior ratings.

By following this systematic approach, we were able to mine meaningful associations from the data, providing a foundation for building recommendation systems or further analyzing user preferences.

4 Association Rule Extraction

In this phase, the objective was to extract association rules from the training set in the form $X \rightarrow Y$, where X contains a single movie, and Y represents a set of other movies that are likely to be watched by users who have rated X . This was achieved by employing a frequent pattern mining approach, specifically using frequent itemset generation followed by association rule creation.

4.1 Setting Minimum Support and Confidence

To ensure that only significant patterns were mined, we defined minimum thresholds for support (*minsup*) and confidence (*minconf*). In this case, the minimum support was set to 9%, meaning that a particular set of movies needed to be rated by at least 9% of users to be considered frequent. The minimum confidence threshold was set to 10%, meaning that the rule $X \rightarrow Y$ was only included if at least 10% of users who watched X also watched Y .

4.2 Generating Frequent Itemsets

The process began by identifying frequent itemsets of size 1, which consisted of individual movies. This was followed by generating itemsets of size 2 and 3 using the Apriori approach. At each step, the support for each itemset was calculated, and only those itemsets that met the minimum support threshold were retained. This ensured that only the most relevant combinations of movies were used for rule generation.

4.3 Extracting Association Rules

Using the frequent itemsets, association rules were generated by calculating the confidence for each rule. For each frequent itemset, possible rules were extracted by taking subsets of the itemset as the antecedent X , while the remaining elements formed the consequent Y . Confidence for each rule was calculated as the ratio of the support of the full itemset to the support of the antecedent. Only those rules that met the minimum confidence threshold were considered valid.

In summary, this step involved systematically mining frequent patterns from the dataset and transforming them into association rules that reflect meaningful relationships between movies. These rules could then be used for recommendation purposes, identifying which movies are likely to be watched together by users.

5 Recommendation: Ranking Association Rules

The final phase of the analysis involved ranking the generated association rules according to two key metrics: support and confidence. This process aimed to identify the most significant rules based on these criteria and analyze any overlap between the top rules in both categories.

5.1 Sorting Rules by Support

The first task was to create a list of the top 100 association rules ranked by their support values. Support indicates the proportion of users for whom the rule applies. Sorting the rules by support allowed us to identify the most commonly occurring relationships between movies in the dataset. The top 100 rules were selected and saved for further comparison.

5.2 Sorting Rules by Confidence

Next, the same set of association rules was sorted by confidence. Confidence reflects the likelihood that users who watched the antecedent X also watched the consequent Y . Sorting by confidence provided insight into the strength of these relationships. Again, the top 100 rules were selected based on confidence and saved separately.


5.3 Identifying Common Rules

To gain deeper insight, the two lists of top 100 rules—one ranked by support and the other by confidence—were compared. The goal was to identify rules that appeared in both lists. These common rules represent strong associations that are both frequent in the dataset (high support) and reliable in terms of prediction (high confidence).

5.4 Ranking Common Rules by Confidence

Finally, the common rules identified in both lists were further ranked according to their confidence scores. This provided a refined list of the most reliable and frequent association rules. These rules could be particularly useful in building recommendation systems, as they capture both the popularity and predictive strength of movie pairings.

By performing this two-pronged analysis, we were able to highlight not only the most frequent patterns in the data but also the most predictive ones, thereby enhancing the value of the association rules for recommendation purposes.




29_top100RulesByConf.txt

an hour ago

File	Edit	View	Language
1	[4226, 7153], [4993], 0.09719934102141681, 0.8939393939393939		
2	[5952, 6539], [7153], 0.09555189456342669, 0.8923076923076924		
3	[296, 1210], [260], 0.09884678747940692, 0.8823529411764706		
4	[47, 1196], [260], 0.10378912685337727, 0.875		
5	[1210, 3578], [2571], 0.09060955518945635, 0.873015873015873		
6	[4993, 6539], [7153], 0.1070840197693575, 0.8666666666666667		
7	[1210, 1270], [260], 0.10378912685337727, 0.863013698630137		
8	[3578, 7153], [4993], 0.09060955518945635, 0.859375		
9	[1196, 1270], [260], 0.10049423393739704, 0.8591549295774649		
10	[1210, 1265], [260], 0.09555189456342669, 0.8529411764705883		
11	[589, 1240], [2571], 0.09060955518945635, 0.8461538461538461		
12	[356, 4993], [7153], 0.11696869851729819, 0.8452380952380952		
13	[1196, 2959], [2571], 0.1070840197693575, 0.8441558441558442		
14	[1196, 1265], [260], 0.09555189456342669, 0.8405797101449276		
15	[356, 5952], [7153], 0.10049423393739704, 0.8356164383561644		
16	[5952, 7153], [4993], 0.12520593080724876, 0.8351648351648352		
17	[1, 1196], [260], 0.09060955518945635, 0.8333333333333334		
18	[6539, 7153], [4993], 0.1070840197693575, 0.8333333333333334		
19	[480, 1196], [1210], 0.10543657331136738, 0.8311688311688312		
20	[1210, 4993], [2571], 0.09719934102141681, 0.8309859154929577		
21	[2571, 5952], [4993], 0.10378912685337727, 0.8289473684210527		
22	[296, 1210], [1196], 0.09225700164744646, 0.823529411764706		
23	[1196, 1210], [260], 0.11367380560131796, 0.8214285714285715		
24	[356, 1196], [260], 0.12026359143327842, 0.8202247191011236		
25	[480, 1210], [260], 0.11202635914332784, 0.819277108433735		
26	[480, 1196], [260], 0.10378912685337727, 0.8181818181818182		
27	[1210, 4993], [260], 0.09555189456342669, 0.8169014084507042		
28	[2762, 2959], [2571], 0.09555189456342669, 0.8169014084507042		
29	[1196, 2858], [260], 0.10214168039538715, 0.8157894736842106		
30	[1196, 2028], [260], 0.10049423393739704, 0.8133333333333334		
31	[1196, 1210], [260], 0.13509060955518945, 0.8118811881188119		
32	[296, 1196], [260], 0.11367380560131796, 0.8117647058823529		
33	[2959, 4993], [7153], 0.11367380560131796, 0.8117647058823529		
34	[1, 1210], [260], 0.09225700164744646, 0.8115942028985508		
35	[593, 7153], [4993], 0.09225700164744646, 0.8115942028985508		
36	[1210, 1265], [2571], 0.09060955518945635, 0.8088235294117647		
37	[589, 1196], [2571], 0.10214168039538715, 0.8051948051948052		
38	[2762, 2959], [356], 0.0939044810543658, 0.8028169014084507		
39	[318, 7153], [4993], 0.1070840197693575, 0.8024691358024691		
40	[260, 480], [1210], 0.11202635914332784, 0.8		

Figure 1: top100RulesByConf.txt



29_top100RulesBySup.txt

an hour ago

	File	Edit	View	Language
1	[356], [318], 0.2355848434925865, 0.562992125984252			
2	[318], [356], 0.2355848434925865, 0.5836734693877551			
3	[318], [296], 0.21252059308072488, 0.5265306122448979			
4	[296], [318], 0.21252059308072488, 0.5584415584415585			
5	[356], [296], 0.21252059308072488, 0.5078740157480315			
6	[296], [356], 0.21252059308072488, 0.5584415584415585			
7	[593], [356], 0.2042833607907743, 0.5876777251184835			
8	[356], [593], 0.2042833607907743, 0.4881889763779528			
9	[593], [296], 0.20098846787479407, 0.5781990521327014			
10	[296], [593], 0.20098846787479407, 0.5281385281385281			
11	[480], [356], 0.19769357495881384, 0.6557377049180327			
12	[356], [480], 0.19769357495881384, 0.47244094488188976			
13	[2571], [356], 0.19769357495881384, 0.5555555555555556			
14	[356], [2571], 0.19769357495881384, 0.47244094488188976			
15	[1196], [260], 0.19604612850082373, 0.7300613496932515			
16	[260], [1196], 0.19604612850082373, 0.6230366492146597			
17	[593], [318], 0.19439868204283361, 0.5592417061611374			
18	[318], [593], 0.19439868204283361, 0.48163265306122455			
19	[2571], [260], 0.1927512355848435, 0.5416666666666666			
20	[260], [2571], 0.1927512355848435, 0.612565445026178			
21	[2571], [318], 0.18780889621087316, 0.5277777777777778			
22	[318], [2571], 0.18780889621087316, 0.4653061224489796			
23	[1210], [260], 0.18616144975288304, 0.7337662337662338			
24	[260], [1210], 0.18616144975288304, 0.5916230366492147			
25	[2959], [2571], 0.18616144975288304, 0.6726190476190477			
26	[2571], [2959], 0.18616144975288304, 0.5231481481481481			
27	[7153], [4993], 0.18616144975288304, 0.7635135135135136			
28	[4993], [7153], 0.18616144975288304, 0.7290322580645161			
29	[356], [110], 0.18451400329489293, 0.4409448818897638			
30	[110], [356], 0.18451400329489293, 0.5989304812834225			
31	[2571], [296], 0.1812191103789127, 0.5092592592592593			
32	[296], [2571], 0.1812191103789127, 0.4761904761904762			
33	[296], [47], 0.17957166392092258, 0.4718614718614719			
34	[47], [296], 0.17957166392092258, 0.6855345911949685			
35	[2571], [1196], 0.17957166392092258, 0.5046296296296297			
36	[1196], [2571], 0.17957166392092258, 0.6687116564417178			
37	[318], [110], 0.17462932454695224, 0.4326530612244898			
38	[110], [318], 0.17462932454695224, 0.5668449179860963			
39	[527], [318], 0.17298187808896212, 0.6441717791411044			
40	[318], [527], 0.17298187808896212, 0.4285714285714286			

Figure 2: top100RulesBySup.txt

6 Evaluation of Recommendations

In this phase, the effectiveness of the generated association rules as a recommendation system was evaluated. The focus was on calculating the average precision and recall for the recommendations provided to users in the test set.

6.1 Precision and Recall Evaluation

Precision and recall are fundamental metrics in evaluating the performance of recommendation systems. Precision measures the proportion of recommended items that are relevant, while recall indicates the proportion of relevant items that were recommended.

To evaluate these metrics, the following steps were undertaken:

1. **User Iteration:** For each user in the test set, the association rules were examined to identify relevant recommendations. Specifically, for each movie X that a user rated in the training set, the corresponding movies Y recommended by the rules were gathered.
2. **Recommendations:** For each user, the top k recommendations (where k varied from 1 to 10) were selected based on the rules associated with the movies they had watched.
3. **Hits Calculation:** The recommended movies were compared against the user's actual ratings in the test set to determine hits—movies that were both recommended and rated by the user.
4. **Precision and Recall Calculation:**

- **Recall** was calculated as the ratio of hits to the total number of movies the user had rated in the test set.
 - **Precision** was determined as the ratio of hits to the total number of recommended movies.
5. **Averaging:** The precision and recall were averaged across all users in the test set to obtain overall metrics for each value of k .

6.2 Plotting Precision and Recall

To visualize the performance of the recommendation system, precision and recall scores were plotted against the varying values of k . The plots provided a clear representation of how the number of recommendations impacted the effectiveness of the system. This visual aid is crucial for understanding the trade-offs between precision and recall as the number of recommendations increases.

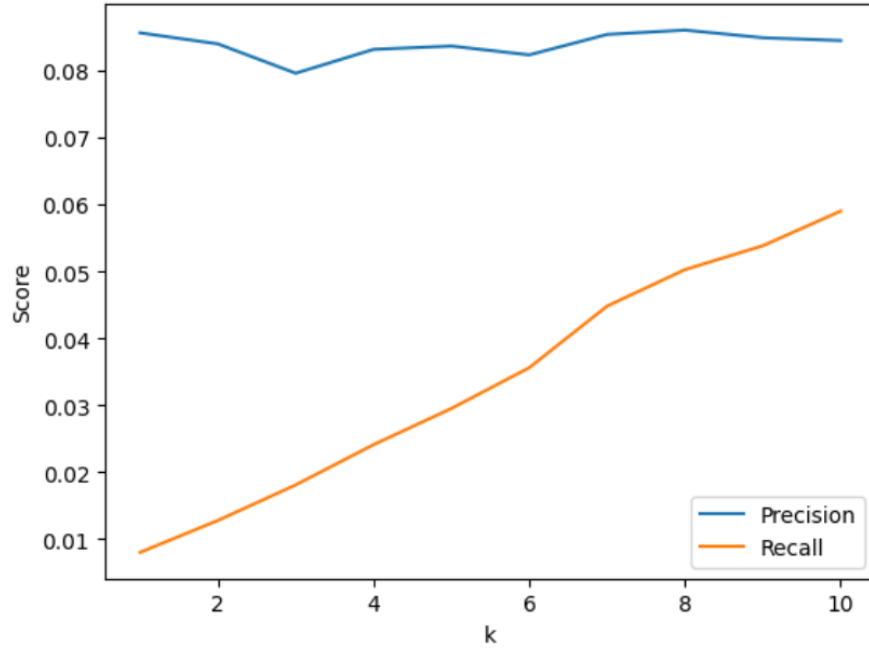


Figure 3: precision and recall vs k

6.3 Observations on Precision and Recall vs. k

1. Trend Analysis:

- **Precision** appears to be relatively stable, with minor fluctuations, suggesting that the quality of recommended items does not drastically change as more recommendations are provided.
- **Recall** shows an upward trend as k increases, indicating that as more recommendations are made, the likelihood of including relevant items (that users rated positively) also increases.

2. Performance Evaluation:

- The higher recall values with increasing k indicate that the recommendation system is effectively capturing more relevant items as it offers more recommendations, which is a desirable characteristic.
- The relatively constant precision suggests that while more relevant items are being recommended, the proportion of relevant items among the recommendations does not significantly improve.

3. Trade-off Insight:

- There appears to be a trade-off between precision and recall. While recall improves, precision does not significantly decrease, which is often a desirable outcome in recommendation systems. This implies that the system can provide more recommendations without diluting the relevance of the suggestions too much.

7 User-Specific Precision and Recall Analysis

In this section, we focused on evaluating the precision and recall metrics for a sample of users in the test set. By analyzing individual user ratings and recommendations, we aimed to provide a more granular understanding of how well the recommendation system performs across different user profiles. To conduct this analysis, a random sample of 10 users was selected from the test set.

7.1 Methodology for Precision and Recall Calculation

For each user, precision and recall were calculated for varying numbers of recommendations (from 1 to 10). This was achieved by utilizing the previously defined evaluation function, which computes these metrics based on the recommendations generated from the association rules. By assessing how many recommended movies aligned with the user's actual ratings, we could derive meaningful insights into the recommendation system's accuracy.

7.2 Visualization of Results

The results were then visualized by plotting the precision and recall scores against the number of recommendations for each user. This visualization allowed us to observe how recommendation performance fluctuated with different values of k . By representing precision and recall separately for each user, we could identify trends, such as whether the recommendation system consistently provided relevant suggestions or if performance varied significantly among users.

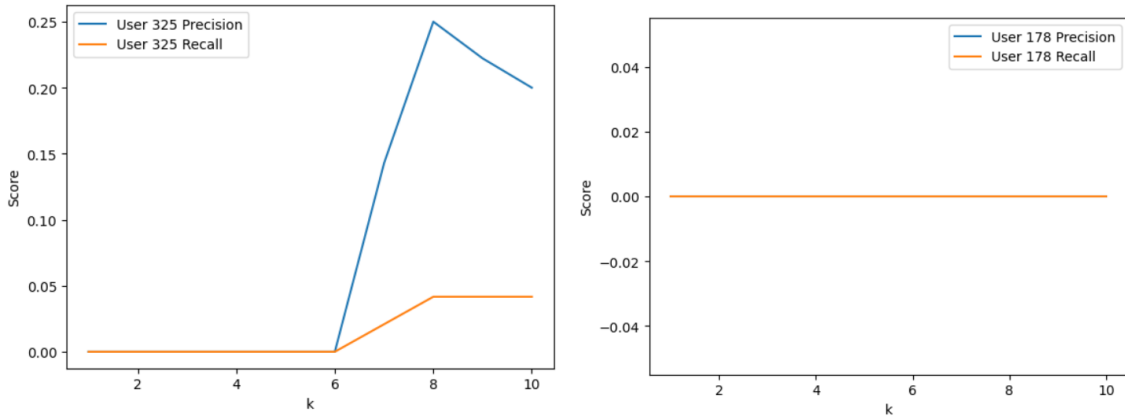


Figure 4: Collection of Images (Second Set)

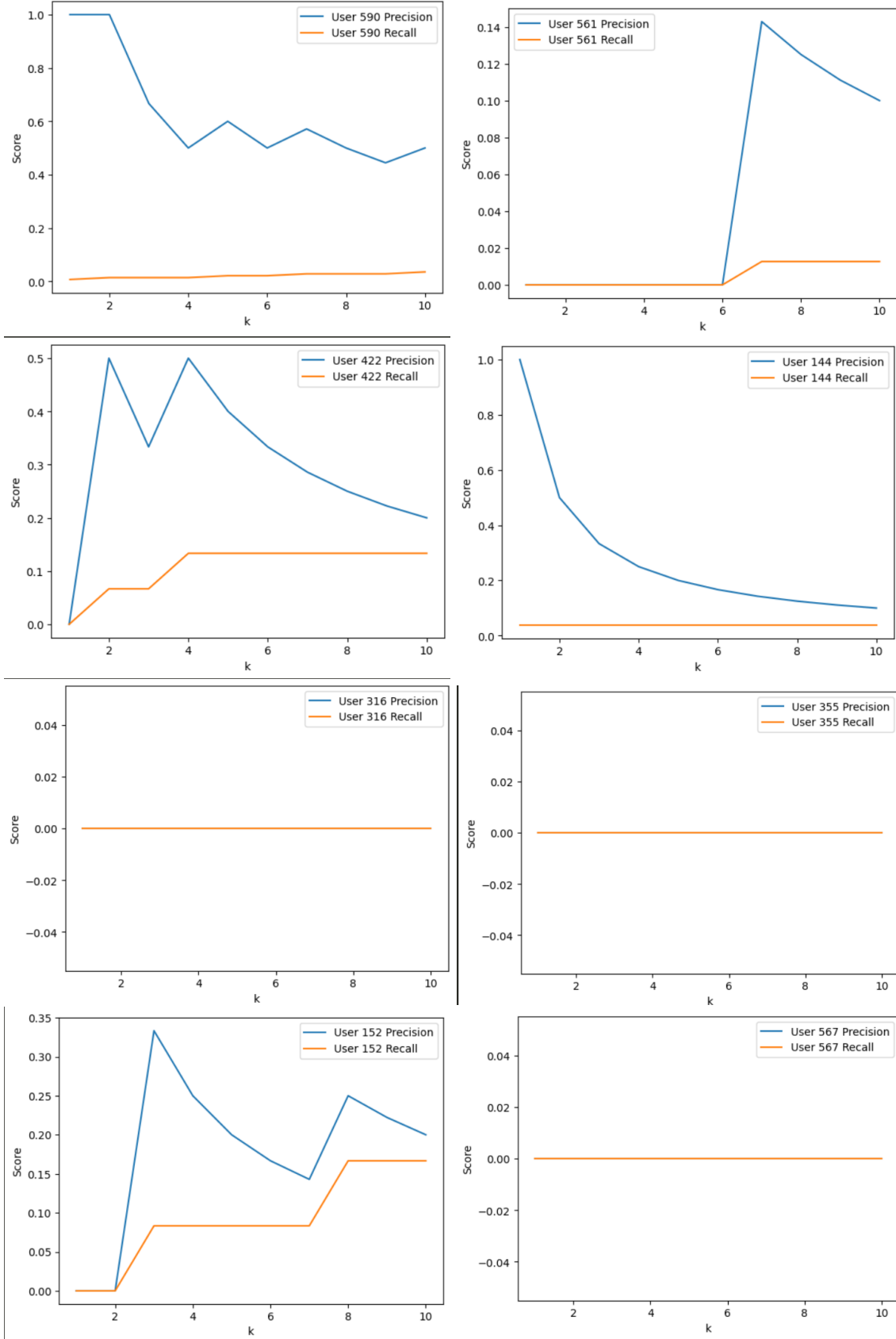


Figure 5: Collection of Images (First Set)