Course name: AIE425 – Intelligent Recommender Systems

Semester: Fall 2025–2026

Project Title: Sleep Quality Improvement Recommendation Engine

SECTION 1 — Dimensionality Reduction and Matrix Factorization
SECTION 2 — Sleep Quality Improvement Recommendation Engine

Group Number: 11

Team Members:

- Mohamed Desoky (222101362)
- Nada Gamal (222101750)
- Bishoy Elgendy (222100403)
- Osama Adel Fawzy (222100211)

Instructor:

- Prof. Samy Ghoniemy

Teaching Assistants:

- Eng. Ahmed Salama
- Eng. Mariam Elgohary

Date: January 5, 2026

Table of Contents

2.6 Hybrid recommendation strategy
2.6.1 Hybrid model formulation
2.6.2 Role of parameter alpha in combining models
2.6.3 Example recommendation lists for different alpha values

2.7 Changing alpha values and item ranking
2.7.1 Ranking behaviour for different alpha settings
2.7.2 Analysis of robustness across users

2.8 Cold-start users: Hybrid algorithm vs popularity baseline
2.8.1 Cold-start problem definition
2.8.2 Popularity-based baseline approach
2.8.3 Performance of hybrid method for cold-start users

2.9 Evaluation, discussion, and conclusion
2.9.1 Evaluation metrics and experimental setup
2.9.2 Results and discussion
2.9.3 Limitations and future work
2.9.4 Overall conclusions

SECTION 1— Dimensionality Reduction and Matrix Factorization

This section of the project introduces different matrix factorization and projection techniques to represent user-item relationships through rating data from recommender systems. The main goal of Section 1 is to explore how various methods represent these relationships, and how the resulting representations impact both similarity analyses and prediction accuracy for certain target items.

Section 1 is divided into three parts. Part 1 looks at Principal Component Analysis (PCA) implemented with mean-filling to accommodate for the absence of some ratings, making this the baseline technique in Section 1 to allow for a covariance-based decomposition of the user-item matrix. Part 2 utilizes a Maximum Likelihood Estimation (MLE) framework to further explore PCA and to minimize the bias created by the usage of mean-filling. Part 3 explores Singular Value Decomposition (SVD), which was applied as a comparison to PCA-type methods.

Across all three parts, the analysis follows a consistent workflow: data preparation and validation, identification of relevant peer items, construction of user–item matrices, dimensionality reduction or factorization, and comparison of results under different configurations. This structure ensures that differences observed between methods arise from the modeling approach itself rather than from inconsistencies in data handling or experimental setup.

Section 1 establishes the foundation for evaluating how latent representations derived from different techniques influence recommendation behavior. The insights gained here are later used to motivate methodological choices and comparisons in subsequent sections of the project.

PART 1: PCA WITH MEAN-FILLING

This part implements PCA using mean-filling as the missing-value strategy, following the required workflow of (i) forming a cleaned user–item ratings table, (ii) validating the

target items, and (iii) computing the mean rating baseline used for mean-filling and fallback prediction. All parameters used in this implementation are explicitly set at the beginning : I1 = B00PCSV0DW, I2 = B005GISDXW, TARGETS = [I1, I2], MIN_COMMON_USERS = 3, MAX_CANDIDATES = 5000, PCA_K = 2, and PCA_USER_SAMPLE = 5000

Part 1

The first step of loading the dataset from the specified file path and doing some basic cleaning work involves creating a working table with three columns: item id, user id, and rating. This article will explain how this process is done in a few easy-to-follow steps. performs basic cleaning of the data by removing leading and trailing whitespace from the id fields; converting the rating field to numerical types, including coercing them to float32 type; and dropping any rows containing NA values in either the item id, user id or rating columns. Once these preliminary cleaning steps are complete, the duplicates (user_id, item_id) will be aggregated using mean ratings generating the final cleaned rating tables which will be used to perform all subsequent stages of the analysis. After aggregation of the duplicates there will be 8,506,849 different ratings given by 8,506,849 different users. This will be shown by the shape of the final cleaned ratings table which has (8506849, 3) after aggregation.

```
df shape: (8506849, 3)
                user_id     item_id  rating
0  A00013803RVZPCZKTT9U  B003ZTNT2Y     1.0
1  A0001392IVCRENBEIEYS  6302409365     5.0
2  A0001598OL7FAN6XNMK9  B00BMRTPEM     5.0
3  A0001598OL7FAN6XNMK9  B00IV3FLO8     4.0
4  A0001598OL7FAN6XNMK9  B00OGL6S64     5.0
```

Sample of the cleaned user–item rating dataset showing user IDs, item IDs, and rating values after preprocessing

Part 2

In Step 2, you'll need to check that both of your target items are in the item_id column before calculating the similarity/PCA. If both of them exist in the dataset, then the output of this validation step should look something like this, {'B00PCSV0DW': True, 'B005GISDXW': True}. The existence of these two items in the dataset allows you to continue with the peer-search and prediction steps for your valid target items.

2) Validate Target Items Exist

```
present = {t: df["item_id"].eq(t).any() for t in TARGETS}
print("Targets present:", present)

if not all(present.values()):
    print("\nExample item_id values:", df["item_id"].dropna().astype(str).unique()[:20])
    raise ValueError("One or both target items were not found in item_id column. Check item_id format.")
```

```
Targets present: {'B00PCSVODW': True, 'B005GISDXW': True}
```

Validation step confirming the presence of selected target item IDs in the dataset before model processing

Part 3

Phase 3 computes the average rating for each item and each of the average ratings across all items within this user base through a groupby function. For the two IDs of interest (B00PCSV0DW and B005GISDXW), average ratings were calculated and are reported here: The mean rating for B00PCSV0DW is 1.447853; the mean rating for B005GISDXW is 1.453237. In addition to reporting individual mean ratings, the calculations also give a total of 182032 items that are eligible for the calculation of average ratings. These calculated average ratings will perform two functions within the user and item matrix created during Part 1: (1) They create a mean-rating-only baseline that will be used to populate the PCA-based user-item matrix; and (2) when there is no other peer-based information available to make a prediction about an individual user's future preferences, the system will use this calculated mean as a "fallback" prediction.

```
item_mean.head()
```

```
Target means (Step 1):
item_id
B00PCSVODW    1.447853
B005GISDXW    1.453237
Name: rating, dtype: float32

Total items with mean: 182032
item_id
0000143502    5.0
0000143529    5.0
0000143561    3.5
0000143588    4.7
0000695009    4.0
Name: rating, dtype: float32
```

Computation of mean ratings for target items and summary statistics of item-level mean ratings across the dataset.

Step 4

This step collects the users that rated the comparison items so that the analyst can determine who will be used in calculating covariance between items and who will contribute to the similarity and prediction. The analyst collects the user ids that have rated the comparison item I1 = B00PCSV0DW, this collection of user ids will form the user set for the comparison item I1. In the same manner, the analyst will collect user ids that rated the comparison item I2 = B005GISDXW. There will be two sets of user ids, one for I1 and the other for I2, which are generated from the cleaned and aggregated ratings table created during Step 1 of the process described in this report.

The output produced by verifies that both comparison items have user sets that contain users, which provide the opportunity to determine how many users in common the comparison items have with those users who rated the candidate peer items. The numbers of users in each comparison item are not provided in this report, but the vast majority of peer items do have user sets that contain a number of users.

Step 5

In this stage, we will compile our candidate peer item set for each target item using all users provided in Step 4. For every target item, each user who gave ratings on that target item will provide additional items they rated as potential peer items. This creates a filtered candidate peer item set where all candidates have at least one user in common with the target item prior to any calculation of similarity.

Next, a candidate filtering mechanism will be implemented. When there are too many candidates, the candidate list will be reduced to a pre-determined number (MAX_CANDIDATES) based on the number of common users between the candidate items and the target item. The filtering process assists in limiting the computational costs associated with covariance and cosine similarity calculations, while still retaining the most relevant peer items for each target item.

At this phase, no similarity scores or covariance scores will be calculated yet. The only outcome from this phase will be a filtered list of candidate peer items to be used in the subsequent phase of covariance and cosine similarity calculations.

STEP 6

In this step, computes similarity measures between each target item and its candidate peer items identified in Step 5. For every (target item, candidate peer) pair, only users who have rated both the target and the candidate item are retained. Candidate peers with fewer than MIN_COMMON_USERS overlapping users are excluded from further analysis.

For each remaining valid pair, centers the ratings by subtracting the global mean rating of each item. Using these centered rating vectors, two similarity measures are computed. First, covariance is calculated as the dot product of the centered vectors divided by (n − 1), where n is the number of common users. Second, cosine similarity is computed as the dot product of the centered vectors divided by the product of their L2 norms, with a small numerical constant added to the denominator to avoid division by zero. Any pairs producing non-finite covariance values are discarded.
This process produces a similarity table for each target item containing, at minimum, the peer item identifier, covariance value, cosine similarity value, and the number of common users. No ranking or truncation is applied in this step; the output consists solely of all valid peer items that satisfy the overlap and numerical validity conditions. These similarity tables are passed unchanged to the next step, where peers are ranked and Top-K subsets are selected.

Step 7

At this stage, ranks the items returned as valid peers in Step 6 for each target item based on a sort order using covariance as the primary sorting method. For each target independently, the Similarity Table is sorted in a descending order based on covariance. The first ten entries on the list represent the Top-10 peers, and the first five of these represent the Top-5 peers.

The peer selection process is rank-based in nature and does not consider any additional thresholds beyond those previously established in steps 1 through 3 (minimum number of common users and the finite covariance). Cosine similarity and the number of common users continue to be included in the tables so they can be referenced for reporting and analysis purposes; therefore, they do not impact the ranking order.

creates separate peer tables for each target item outlining the respective sets for the Top-5 and Top-10 peers. These peer sets will be used unchanged in the next steps for predicting and constructing the PCA matrices. No other aggregation or averaging of information is conducted until that point in time.
At this stage, ranks the items returned as valid peers in Step 6 for each target item based on a sort order using covariance as the primary sorting method. For each target independently, the Similarity Table is sorted in a descending order based on covariance. The first ten entries on the list represent the Top-10 peers, and the first five of these represent the Top-5 peers.

The peer selection process is rank-based in nature and does not consider any additional thresholds beyond those previously established in steps 1 through 3 (minimum number of common users and the finite covariance). Cosine similarity and the number of common users continue to be included in the tables so they can be referenced for reporting and analysis purposes; therefore, they do not impact the ranking order.

creates separate peer tables for each target item outlining the respective sets for the Top-5 and Top-10 peers. These peer sets will be used unchanged in the next steps for predicting and constructing the PCA matrices. No other aggregation or averaging of information is conducted until that point in time.

```
item_mean.head()
```

```
Target means (Step 1):
item_id
B00PCSVODW    1.447853
B005GISDXW    1.453237
Name: rating, dtype: float32

Total items with mean: 182032
item_id
0000143502    5.0
0000143529    5.0
0000143561    3.5
0000143588    4.7
0000695009    4.0
Name: rating, dtype: float32
```

Processing log showing candidate item filtering and identification of valid peer items for covariance-based analysis.

```
Processing 1255 candidates for B00PCSVODW...
  Processed 1000/1255 candidates...
Found 16 valid peers for B00PCSVODW
Processing 909 candidates for B005GISDXW...
Found 10 valid peers for B005GISDXW
Top-5 peers for I1 (Step 7):
```

	peer_item	covariance	cosine	n_common
0	B00TKIJGDA	1.394654	0.761724	3
1	B000006B4Y	1.174917	0.822492	3
2	B00HSJ2CVQ	0.696592	0.639306	3
3	B00NCDVVLY	0.507928	0.518404	3
4	B0090JBOC0	0.390683	0.275748	5

```
Top-10 peers for I1:
```

Top-5 peer items for target item I1 selected based on covariance, cosine similarity, and number of common users.

0	6305480869	1.104002	0.555135	3
1	B00ZL4Q7NE	0.862015	0.433506	7
2	B00HW3EI3I	0.485930	0.360607	3
3	B00BTFK07I	0.354236	1.000000	3
4	B00FL31UF0	0.261219	0.373409	5

Top-10 peers for I2:

	peer_item	covariance	cosine	n_common
0	6305480869	1.104002	0.555135	3
1	B00ZL4Q7NE	0.862015	0.433506	7
2	B00HW3EI3I	0.485930	0.360607	3
3	B00BTFK07I	0.354236	1.000000	3
4	B00FL31UF0	0.261219	0.373409	5
5	B00IKM5LXG	0.184170	0.276155	3
6	B00HUTPK4U	-0.108623	-0.246506	3
7	B004K6FS5W	-0.291161	-0.480681	4

Top-10 peer items for target item I2 identified using covariance-based neighborhood selection.

	peer_item	covariance	cosine	n_common
0	B00TKIJGDA	1.394654	0.761724	3
1	B000006B4Y	1.174917	0.822492	3
2	B00HSJ2CVQ	0.696592	0.639306	3
3	B00NCDVVLY	0.507928	0.518404	3
4	B0090JBOC0	0.390683	0.275748	5
5	B00005RFHF	0.248986	0.639605	5
6	6303162290	0.150472	0.430142	3
7	B00PH1H6TK	0.086408	0.178911	3
8	B00J22YU62	0.014812	0.017595	3
9	B00SFRHKAI	-0.068101	-0.103109	3

Top-5 peers for I2 (Step 7):

Expanded peer ranking results illustrating covariance and similarity values for selected neighboring items.

STEP 8

ITEM-BASED RATING PREDICTION USING TOP-5 AND TOP-10 PEERS

In this step, generates item-based rating predictions for each target item using the peer sets selected in Step 7. Predictions are computed separately for the Top-5 peer set and the Top-10 peer set, producing two predicted values per target item. The prediction process is applied only to users who appear in the evaluation subset constructed in.

For a given user and target item, the prediction is computed as a weighted average of the user's ratings on the peer items, where the weights correspond to the covariance values between the target item and each peer item. Only peers that the user has rated contribute to the weighted sum. If a user has not rated any of the peer items in the selected peer set, applies a mean-filling fallback and assigns the global mean rating of the target item as the predicted value.

outputs a prediction table that includes, for each evaluated user, separate prediction columns for Top-5 and Top-10 peer configurations for both target items. The prediction values shown in the output reflect either covariance-weighted aggregation of peer ratings or direct fallback to the target mean rating when no usable peer ratings are available. No additional normalization or post-processing is applied beyond the described weighted scheme.

Predictions ready. Example:

user_id	pred_I1_top5	pred_I1_top10	pred_I2_top5	pred_I2_top10
A00013803RVZPCZKTT9U	1.447853	1.447853	1.453237	1.453237
A0001392IVCRENBEIEYS	1.447853	1.447853	1.453237	1.453237
A0001598OL7FAN6XNMK9	1.447853	1.447853	1.453237	1.453237
A0002090WKEMAO8KOWKM	1.447853	1.447853	1.453237	1.453237
A00049826E18XJLZ3YC0	1.447853	1.447853	1.453237	1.453237
A0005426V58WVW05LDKK	1.447853	1.447853	1.453237	1.453237
A0005916MHK9RK69491E	1.447853	1.447853	1.453237	1.453237
A0007042BQBQLK20MOG7	1.447853	1.447853	1.453237	1.453237
A0007430W3WXY3QNYB2S	1.447853	1.447853	1.453237	1.453237
A00086729ZDSXGG2E481	1.447853	1.447853	1.453237	1.453237

Example of predicted ratings for target items using Top-5 and Top-10 neighborhood aggregation

The second part is about Step 9 of. The purpose of this step is to create two user-item matrices needed for PCA using the peer sets created in Step 7. The user-item matrices are created specifically using peer items in the Top 5 and Top 10 by Target Group (target group).

The first matrix will be based on the Top 5 Peer items for Target Group, while the second matrix will be based on the Top 10 Peer items for Target Group. Each matrix will only include data from those users who were included in Step 6, as well as only the Peer items that were included in the respective Top 5 and Top 10 Peer unions for Target

Group. (i.e. All peer items included in a user-item matrix will be present within its union of Peer users).

For both matrices, the user-item matrix will consist of rows containing individual users and columns containing the Peer items. Where a user has provided a rating for a Peer item, that value is placed directly in the user-item matrix. If there is no rating available for that Peer item, the mean value for that Peer item will be used to approximate the user rating for that Peer item, which is consistent with the mean-filling method previously mentioned. Beyond the mean-filling step, neither matrix has undergone any form of normalisation or scaling adjustment during this stage.

is now able to show confirmation that both user-item matrices have been created and are in a format ready for uploading for PCA analysis. The two matrices also differ only by the number of columns of Peer items. The only difference between the two matrices is therefore the Top 5 and Top 10 Peer unions, respectively. The actual dimensions of either numerical user-item matrix are therefore not displayed in any of the visual output from; therefore, this document is intended to clarify the construction process for both user-item matrices without defining the exact numerical user-item dimensions.

```
Building PCA matrix for 5000 users x 10 items (Top-5)...
Building PCA matrix for 5000 users x 20 items (Top-10)...
Reduced space (Top-5 union) sample:
```

	PC1	PC2
A00013803RVZPCZKTT9U	0.0	0.0
A0001392IVCRENBEIEYS	0.0	0.0
A0001598OL7FAN6XNMK9	0.0	0.0
A0002090WKEMAO8KOWKM	0.0	0.0
A00049826E18XJLZ3YC0	0.0	0.0

```
Reduced space (Top-10 union) sample:
```

	PC1	PC2
A00013803RVZPCZKTT9U	0.0	0.0
A0001392IVCRENBEIEYS	0.0	0.0
A0001598OL7FAN6XNMK9	0.0	0.0
A0002090WKEMAO8KOWKM	0.0	0.0
A00049826E18XJLZ3YC0	0.0	0.0

Construction of PCA matrices for Top-5 and Top-10 neighborhoods and projection into the reduced latent space.

STEP 10

will perform PCA on the mean value filled user-item matrices from Step 9. This is done separately on both the user-item matrix for the Top 5 peer union and for the Top 10 peer union. In both matrices the PCA will be carried out independently and from scratch using the covariance matrix of the mean value filled matrices, rather than using a library PCA routine.

Before performing PCA, will first calculate the Mean for each user-item matrix, and subtract the column-wise means of the user-item matrix from each item column of the user-item matrix to centre the data. Then the Covariance Matrix of the user-item matrix will be computed across all the items, and then eigen-decomposition will be carried out to calculate the eigenvalues and eigenvectors. The principal components of the two data sets are the first PCA_K principal components, where PCA_K is defined to be equal to 2 in configuration.

Then the Scores of the users on the principal components will be calculated by multiplying the centred user-item data by the matrix of the selected eigenvectors. The PCA will produce reduced dimension representations of users in the case of both the Top 5 and Top 10 users, reporting values for the first principal component (PC1) and second principal component (PC2) of each user in both cases. The resulting reduced representations will only be used for comparison and analysis and will not be used for the prediction phase.

Step 11

In the final step of Part 1 compares the results obtained using the Top-5 peer set versus the Top-10 peer set. This comparison is performed consistently across both stages where peer-set size has an effect: (1) item-based rating prediction and (2) PCA-based dimensionality reduction.
For prediction, places the Top-5 and Top-10 predicted ratings side by side for each target item in the output table. The comparison highlights whether expanding the peer set from five to ten items changes the predicted values or whether predictions remain unchanged due to mean-filling fallback when users lack ratings on the additional peers. Only the prediction values explicitly shown in the outputs are reported; no inferred differences or unreported metrics are introduced.

For PCA, compares the reduced user representations obtained from the Top-5 peer union matrix and the Top-10 peer union matrix. Both cases use the same number of retained components (K = 2) and the same PCA procedure. Any observed differences between the two cases arise solely from the difference in the underlying peer-item sets used to construct the mean-filled matrices, not from changes in PCA configuration. This step completes Part 1 by demonstrating how the choice of peer-set size affects both prediction behavior and the structure of the reduced-dimensional user space under a mean-filling strategy, strictly based on the outputs generated .

user_id	pred_I1_top5	pred_I1_top10	pred_I2_top5	pred_I2_top10
A00013803RVZPCZKTT9U	1.447853	1.447853	1.453237	1.453237
A0001392IVCRENBEIEYS	1.447853	1.447853	1.453237	1.453237
A0001598OL7FAN6XNMK9	1.447853	1.447853	1.453237	1.453237
A0002090WKEMAO8KOWKM	1.447853	1.447853	1.453237	1.453237
A00049826E18XJLZ3YC0	1.447853	1.447853	1.453237	1.453237
A0005426V58WVW05LDKK	1.447853	1.447853	1.453237	1.453237
A0005916MHK9RK69491E	1.447853	1.447853	1.453237	1.453237
A0007042BQBQLK20MOG7	1.447853	1.447853	1.453237	1.453237
A0007430W3WXY3QNYB2S	1.447853	1.447853	1.453237	1.453237
A00086729ZDSXGG2E481	1.447853	1.447853	1.453237	1.453237
A0009988MRFQ3TROTQPI	1.447853	1.447853	1.453237	1.453237
A00116683339FAW9XGHO	1.447853	1.447853	1.453237	1.453237

Final user-level prediction table showing estimated ratings for the selected target items

A0007042BQBQLK20MOG7	1.447853	1.447853	1.453237	1.453237
A0007430W3WXY3QNYB2S	1.447853	1.447853	1.453237	1.453237
A00086729ZDSXGG2E481	1.447853	1.447853	1.453237	1.453237
A0009988MRFQ3TROTQPI	1.447853	1.447853	1.453237	1.453237
A00116683339FAW9XGHO	1.447853	1.447853	1.453237	1.453237
A001170867ZBE9FORRQL	1.447853	1.447853	1.453237	1.453237
A0014392U7DSQERYR8EC	1.447853	1.447853	1.453237	1.453237
A001524696SLA34399M4	1.447853	1.447853	1.453237	1.453237
A0017882XAS5VJGSZF5R	1.447853	1.447853	1.453237	1.453237
A0018632VUVKRGSYBEAT	1.447853	1.447853	1.453237	1.453237
A0018722W1EILNRQMK78	1.447853	1.447853	1.453237	1.453237
A0019420MGJRFO7TA5QC	1.447853	1.447853	1.453237	1.453237
A0020818TMV75JLXCE6I	1.447853	1.447853	1.453237	1.453237

Extended prediction results illustrating consistency across multiple users

Part 2 (PCA with Maximum Likelihood Estimation)

The PCA-based analysis covered in Part 2 of Section 1 builds on previous work by applying MLE to estimate which ratings were given as well as which ratings were not given within the context of PCA-based Dimensionality Reduction. A mean-fill strategy replaces all missing values with the mean or average of their positional neighbors before applying PCA dimensionality reduction; whereas an MLE strategy uses the pairwise overlap of neighbouring positional evaluations to estimate covariance and make predictions with no missing ratings. Thus, in this context an MLE approach estimates covariance only for those item pairs with overlap, rather than for items that are not rated, thereby making predictions without requiring explicit imputation of non-rated items beforehand. The purpose of this section is to determine the impact of combining PCA with covariance calculations derived from MLE pairwise likelihood estimation on peer selection, prediction results and low-dimensional representations, and to provide a consistent comparison to the mean-fill approach using identical target items.

This part follows a structured workflow that includes data filtering, construction of a user–item matrix, pairwise covariance estimation using observed overlaps only, peer selection based on MLE covariance, prediction of missing ratings using the estimated covariance structure, and comparison of Top-5 versus Top-10 peer configurations. The results from this part are later contrasted with both PCA using mean-filling and SVD to assess differences in modeling assumptions, robustness to sparsity, and prediction behavior.

Step 1

In this step, the ratings dataset is prepared for PCA using a Maximum Likelihood Estimation framework by applying controlled filtering to reduce sparsity while preserving sufficient overlap between users and items. The dataset is first loaded using the required rating fields and aggregated so that each user–item pair is represented by a single rating value.

To ensure a dense and informative subset of the data, users are ranked by their number of ratings, and only the most active users are retained. Specifically, the top 10,000 users with the highest rating counts are selected. After this filtering step, the dataset is reduced to a subset containing 86,376 rating records while preserving the original rating scale and structure. This filtered dataset forms the basis for all subsequent PCA-MLE computations.

This step aims to increase estimates of covariance through Maximum Likelihood Estimation (MLE) by maximizing the chances of overlapping ratings for a pair of items. In this step there is no application of MLE for similarities/dimensions; however, the rating tables will be output as cleaned and filtered rated tables so they may be used for subsequent covariance estimations using likelihood-based methods.

```
TOP_K_USERS = 10000

user_counts = df.groupby("user_id")["item_id"].count().sort_values(ascending=False)
top_users = set(user_counts.head(TOP_K_USERS).index)

print("Top users selected:", len(top_users))
```

```
Top users selected: 10000
```

```
df_users = df[df["user_id"].isin(top_users)].copy()

print("After user filtering:", df_users.shape)
```

```
After user filtering: (863176, 4)
```

Selection and filtering of the top-K most active users used for model training and evaluation

step 2

The aim of step 2 is for the user to determine the target items that will be used in Creating a successful PCA with MLE process and confirm the target items are present in the filtered dataset from step 1. The target items selected are I1 = B00PCSV0DW and I2 = B005GISDXW. Once verified by the filtering process, these items will form the dataset used to estimate covariance and make predictions.

Ranking of the ratings received by the selected target items will also provide a means of controlling sparsity and increasing the number of shared ratings by users. The method used is to sort items according to rating count for the users who rated them in the filtered dataset, with the top one thousand items receiving the highest number of ratings selected. The final selection of the items to analyse is 1002 items, which includes the two target items selected, even though they might not have been in the top rated items by frequency.

The output of this selection will be a filtered dataset which contains the user-item ratings for the target items, which will provide the basis for creating the user-item rating matrix for PCA with MLE. The dimensions of the filtered dataset of (199,756,4) shows the total number of retained ratings after filtering the users and items in the original dataset.

```
print("I1 in sample?", I1 in top_items)
print("I2 in sample?", I2 in top_items)

df_sub = df_users[df_users["item_id"].isin(top_items)].copy(

print("Subset shape:", df_sub.shape)
```

```
Final item count: 1002
I1 in sample? True
I2 in sample? True
Subset shape: (199756, 4)
```

Verification of target item presence and construction of the filtered item subset used for analysis

```
Target items: B00PCSVODW B005GISDXW
```

Identification of selected target items (I1 and I2) prior to neighborhood and prediction steps

Step 3

In Step 3 of this process, the subset of ratings filtered in STEP 2 is converted into a user–item rating matrix. The rows in this matrix correspond to users (in this case 9,372), the columns correspond to items (1,002 total items available) and each cell of the matrix represents the average rating given by a specific user to an item, where applicable. No explicit imputation has been applied here; therefore, missing values are retained for

maintaining the sparsity that will allow for estimation of covariance using likelihood-based methods.

In addition, both of the target items (I1 = B00PCSV0DW and I2 = B005GISDXW) have been confirmed to be present as columns in the completed user–item matrix. Therefore, covariance estimation, peer selection, and prediction steps can be conducted on both target items based on the same user-item rating matrix.

Finally, this user-item rating matrix is the principal structure needed to perform PCA with MLE. In contrast to the mean-filling method, the matrix will be treated as incomplete so that the covariance estimates are based solely on the observed rating overlaps for the items.

```
print("R shape:", R.shape)
print("Users:", R.shape[0], "Items:", R.shape[1])
print("I1 exists?", I1 in R.columns)
print("I2 exists?", I2 in R.columns)
```

```
R shape: (9372, 1002)
Users: 9372 Items: 1002
I1 exists? True
I2 exists? True
```

Construction and validation of the user–item rating matrix, confirming the existence of target items in the final matrix.

Step 4

This step involves using Maximum Likelihood Estimation (MLE) to estimate covariance between pairs of items through overlapping rating data without making any assumptions regarding missing ratings for pairs with no overlap. The covariance values will be zero for item pairs that do not contain any users who rated both items.

To calculate the covariance for a pair of items, all users who rated both items are selected and their average ratings are subtracted from the corresponding item's mean ratings to create a centered value for each item. The resulting centered values for the two items will then be multiplied together; the average (mean) of those products will provide us with the covariance value for the item pair. This process is repeated for every possible item pair based on the items retained for our analysis; therefore producing a square covariance matrix with a size corresponding to the number of items that we kept during analysis.

The covariance matrix produced by this step is an (1002,1002) array that shows how closely related each item is to each of the other items based on the user's ratings of the items. This matrix serves as the basis for selecting users who are likely to provide the most accurate recommendations, predicting future user ratings, and reducing the dimensionality of data when used in conjunction with PCA-MLE techniques.

```
Covariance matrix shape: (1002, 1002)
```

	076780192X	0767803434	0767805712	0767824571	0767827759
076780192X	0.986628	0.222222	-0.062222	-0.088435	-0.047337
0767803434	0.222222	0.928945	0.320988	-0.140625	0.416667
0767805712	-0.062222	0.320988	1.037721	0.905325	0.074380
0767824571	-0.088435	-0.140625	0.905325	0.722500	0.000000
0767827759	-0.047337	0.416667	0.074380	0.000000	1.562486

Example of the item–item covariance matrix computed for the filtered set of items.

Step 5

analysis of the summary statistics and overlap properties for the target items is required to verify that the covariance estimates involving the targets are being developed based on well-defined empirical covariance estimates for target items that are based on sufficient data. That is, for each target item, the total number of ratings that were

observed for that target item is computed in the filtered user- item matrix and the empirical mean rating for that target item is calculated based on only the data that are available for it within this filtered matrix.

To determine the degree of overlap between each target item and the other items in the matrix, the number of users who rated the target item and the candidate item is determined. The overlap data are then used to ensure that the covariance estimates associated with the target items are derived from non-empty intersections of users, and therefore are indicative of meaningful co-ratings.

This analysis step serves to confirm that both of the target items have sufficient rating data and ratings' overlaps with other items so that reliable covariance-driven peer selection and prediction can occur. Therefore, no ranking of the target items nor truncation of the candidate items occurs during this analysis phase, as it only serves to confirm that the covariance estimates involving the target items are statistically valid.

```
Target items: B00PCSVODW B005GISDXW
Counts: 6 5
```

Identification of the selected target items and their interaction counts in the dataset.

Step 6

Steel item peer selection occurs here using the covariance matrix to identify covariance values, which were generated in Step 4 as the basis for peer selection. The covariance matrix is comprised of data that represents how much two items have in common with one another based upon user overlaps. For each target item independently, the row of the covariance matrix associated with the target item can be combined with the same

number of rows from other item sets filtered by the first item in order to determine which item will become a peer of the target.

Covariance values between target item and other items (non-zero values) represent that at least one user has overlapping interests with the target item. The top ten item peers are ranked from highest to lowest based on their respective covariance values with the target item, therefore, the top five items can be defined as the first five items that are part of the top ten peer items selected from the ranked list of items.

Item peer selection is based solely upon MLE-created covariance values, however, other forms of similarity, such as cosine similarity, have not been utilized and only neighbourhood heuristics have not been utilized except to rank items based on covariance. As a result, the Top 5 and Top 10 items for each target item will be maintained unchanged for use in future prediction/PCA development.

```
Top5 peers for I1:
 B005ZCSP0K    0.5
076780192X    0.0
B004HO6HWK    0.0
B004BLJQOK    0.0
B004C03TK2    0.0
Name: B00PCSVODW, dtype: float64

Top10 peers for I1:
 B005ZCSP0K    0.5
076780192X    0.0
B004HO6HWK    0.0
B004BLJQOK    0.0
B004C03TK2    0.0
B004EPYZOY    0.0
B004EPYZP8    0.0
B004EPYZQ2    0.0
B004EPYZQC    0.0
B004EPYZTE    0.0
Name: B00PCSVODW, dtype: float64
```

Top-5 and Top-10 peer items for target item I1 based on covariance similarity

```
Name: B00PCSVODW, dtype: float64

Top5 peers for I2:
 076780192X    0.0
B004HO6HWK    0.0
B004BLJQOK    0.0
B004C03TK2    0.0
B004EPYZOY    0.0
Name: B005GISDXW, dtype: float64

Top10 peers for I2:
 076780192X    0.0
B004HO6HWK    0.0
B004BLJQOK    0.0
B004C03TK2    0.0
B004EPYZOY    0.0
B004EPYZP8    0.0
B004EPYZQ2    0.0
B004EPYZQC    0.0
B004EPYZTE    0.0
B004EPYZU8    0.0
Name: B005GISDXW, dtype: float64
```

Top-5 and Top-10 peer items for target item I2 selected using the same covariance-based approach

Step 7

Following Step 6, this step generates rating predictions for target items using the selected peer sets and MLE covariance values. Predictions are generated separately for the Top-5 and Top-10 peer configurations to assess how the size of the user's neighbour affect predictive ability within an MLE context.

For each user and target item, the rating prediction is calculated as the sum of the user's ratings of peer items multiplied by the covariance value between the peer items and target item. Only the peer items rated by the user will be used as predictors of the target item. If there are no peer item ratings by the user in the selected peer set, the

prediction will be determined by the average rating of the target item calculated from available observed ratings.

The prediction tables present separately each target item under both the Top-5 and Top-10 peer configurations and provide a direct comparison of how likelihood based estimates of covariance affect predictive ability versus relying on prior means to fill in user-item rater matrices.

Step 8

Constructing PCA input matrices in the PCA-MLE pipeline consists of constructing PCA input matrices in terms of a particular MLE framework using peer sets identified at step 6. This involves constructing 2 PCA input matrices (one constructed using a union of the Top-5 peer items for the target items and the other constructed using a union of Top-10 peer items for the target items) based on the same user subset defined previously in the PCA-MLE pipeline.

Each input matrix has rows representing users and columns representing the selected peer items for those users, while the input matrix will contain observed ratings if they exist. Unlike the mean-filling approach, where missing entries are filled with item means, the PCA input matrices constructed under the MLE framework contain entries with missing values, and thus the PCA algorithm will utilize the estimated covariance structure based on the estimated covariance, rather than a fully populated rating matrix.

This method of constructing input matrices provides consistent estimates between the covariance structure and the dimensionality reduction of ratings and will result in the two matrices being identical except for the number of columns of peer items (e.g., Top-5 peer items and Top-10 peer items), but otherwise will be identical in respect to the same user subset and observed ratings.

STEP 9

Using Maximum Likelihood Estimation, the item-item covariance matrix is generated. The principal component analysis (PCA) is completed using the item-item covariance matrices that were generated using MLE estimators, which allows direct application of PCA to the MLE-based covariance structure. By doing so, the dimensionality reduction technique of PCA is aligned with the statistical assumptions underlying the data.

The covariance matrices are subjected to eigen decomposition to produce eigenvalues and eigenvectors for each of the item-item covariance matrices. The leading eigenvectors with the largest eigenvalues are retained as principal components. The number of principal components being retained is set to be the same as what was originally used in the previous PCA, therefore allowing for direct comparisons between methods.

Additionally, separate reduced representations of users will be created based on projected ratings that result from applying the reduced-dimensional PCA versions of their original ratings. These reduced representations are based on the different item configurations that were used to create the covariance inputs, and thus, there will be different reduced representations for the Top 5 and Top 10 peer configurations.
This section compares results from using the Top-5 peers and Top-10 peers from the PCA with MLE process. Specifically it assesses: 1) the prediction behaviour between the two peer configurations and 2) structure of the reduced dimensionality representations between peer configurations. The analysis isolates neighbourhood size's effect on PCA-MLE results by holding all parameters constant except neighbourhood size.

As it relates to predicting user rating values, the analysis visually compares predicted values for each target item, between the Top-5 peers and Top-10 peers. Since additional peer items in the Top-10 give rise to more covariance contributions, differences exist where there are overlaps in ratings. However, where users do not have a rating value

for the additional peer items, there will be no change in their predicted values, as they rely on overlaps, not imputed values.

As it relates to dimensionality reduction, reduced dimensionality user representations based on covariance structures for Top-5 peers and Top-10 peers were also compared. Consequently, the number of principal components in both cases was kept constant so that the differences observed in reduced user representations were due only to the expanded peer set. Additionally, the analysis determined whether including additional peers influences the proportion of variance explained by the first few principal components, as well as how users are positioned relative to one another in reduced dimensionality space.
This step is an analysis of how the principal component analysis (PCA) algorithm using maximum likelihood estimation (MLE) performs in circumstances in which there is sparsity of users (very few ratings from a user) or sparsity of items (very few ratings for an item). In this case, the covariance structure is based on observed overlaps, and items for which there are very few ratings and users for which there are very few ratings will have less influence on the covariance structure than do well-represented users and items.

Users with fewer ratings will primarily be projected into the reduced PCA space based on the limited set of items they have rated, which means their representations may be less stable than users who have rated a larger number of items. The same applies to items with little overlap with other items as these items have weaker covariance relationships with other items, and therefore, have less impact on the leading principal components.

As shown in this step, PCA using MLE has the property that while it does avoid the mean filling bias, it is still sensitive to sparsity of data. The ability of PCA using MLE to concentrate on statistically supported relationships and its ability to naturally downweight poorly observed interactions have an effect on the reliability of predictions made in cold-start situations.

I1 biggest diffs:

	user_id	pred_B00PCSVODW_top5	pred_B00PCSVODW_top10	abs_diff
2572	A22RY8N8CNDF3A	0.935622	3.080595	2.144974
4407	A2TXH9QKLD4ZVX	1.343536	0.733441	0.610094
2258	A1XT8AJB7S9JJG	1.692656	1.232204	0.460452
9048	AV6QDP8Q0ONK4	1.655819	2.064797	0.408978
872	A1CLHLW9PFKG9Q	1.659186	1.325656	0.333529
5174	A34D06JL7LC6MU	1.932321	1.665848	0.266473
1921	A1SHLQKJSPCCNZ	1.343536	1.116838	0.226697
7680	AAZRWLML88IZK	1.659186	1.439648	0.219538
6737	A3QH6BEY6RYQR0	1.619662	1.838219	0.218557
9138	AWG2O9C42XW5G	0.703570	0.494172	0.209398

I2 biggest diffs:

Comparison of Top-5 and Top-10 prediction differences for target item I1 across users.

I2 biggest diffs:

	user_id	pred_B005GISDXW_top5	pred_B005GISDXW_top10	abs_diff
0	A100JCBNALJFAW	2.4	2.4	0.0
6257	A3JYJ907WWREJH	2.4	2.4	0.0
6241	A3JPFWKS83R49V	2.4	2.4	0.0
6242	A3JPPR6JT75N0E	2.4	2.4	0.0
6243	A3JSDTPWSYFW23	2.4	2.4	0.0
6244	A3JSO0N085OQXU	2.4	2.4	0.0
6245	A3JSROIZ1SFTS	2.4	2.4	0.0
6246	A3JTA7SAV9NSDE	2.4	2.4	0.0
6247	A3JTBJC5WSEZ7Q	2.4	2.4	0.0
6248	A3JU9CWXUVHUPU	2.4	2.4	0.0

Mean abs diff I1: 0.0007480444017118762
Mean abs diff I2: 0.0

Prediction stability analysis for target item I2 showing negligible differences between Top-5 and Top-10 neighborhoods.

The final step in our analysis is a summary of the PCAs with MLE and their position in relation to the general framework developed in Section 1. PCA with MLE is a principled approach to mean-filling; it estimates covariances using only the observed data, thus maintaining the statistical integrity of the ratings matrix.

The analysis demonstrates the impact of MLE covariances on peer selection, predictions, and dimensionally reduced representations. The sensitivity analysis illustrates the dependence of results on the size of the peer sets used to create these estimations, while the sparsity and cold-start analyses illustrate that sufficient overlap between users and items is needed by PCA with MLE.

Compared to mean-filling, PCA with MLE produces less imputation bias; however, to achieve stable estimates from PCA with MLE, PCA with MLE requires denser overlap between items and users.

The completion of this section allows Section 1 to present three clearly identified methodologies of conducting PCA with mean-filling, PCA with MLE, SVD, and the use of the same targets and evaluation criteria to provide a clear and methodologically justified comparison of these methods as dimensionality reduction and factorization methods for recommender systems.

PART 3 INTRODUCTION: SINGULAR VALUE DECOMPOSITION (SVD)

Section 1

Part 3 aims to analyze how SVD can provide a method of decomposition for user-item matrix based models where there are ratings for an item by a user. Unlike PCA-based methods which rely on covariance estimation, SVD will take the user-item rating matrix and decompose it into a set of latent user factors, latent item factors, and singular values associated with each user and item. The SVD approach will allow you not only to

create lower-dimension matrices (as with PCA), but also reconstruct the original rating data in a unified model.

This section will evaluate: (1) whether truncated SVD is able to extract the most significant patterns of user-item interactions in sparse rating data; (2) how the reconstruction error (the difference between the actual user-item rating and the predicted rating) varies based on the number of latent dimensions selected when decomposing the user-item rating matrix; and (3) how the predictions made using SVD behave for specific users and specific items. This analysis includes a complete SVD decomposition, truncation of SVD to a small number of components, evaluation of reconstruction error, evaluation of the percentage of variance that remains after removing latent dimensions, interpretation of latent user and item factors, evaluation of the effect of the choice of number of latent dimensions on the SVD reconstruction error, and the evaluation of cold-start user and item performance based on SVD-generated recommendations.
All experiments in this part are conducted on a controlled subset of users and items to balance computational feasibility and rating density. The results obtained from SVD are later compared with PCA using mean-filling and PCA using Maximum Likelihood Estimation to highlight differences in modeling assumptions, robustness to sparsity, prediction accuracy, and interpretability

SVD Analysis Step 1

In this initial step, we will be preparing the Ratings dataset for SVD analysis by creating a sparse user-item rating matrix.
The input file contains the fields item_id, user_id, rating and timestamp. We are averaging duplicate user-item ratings to create a single rating value for each user-item pair.
No normalization or any transformation is being done to the rating scale during this step.

In order to control both the sparsity of the final matrix as well as its computational complexity, we will rank users based upon how many ratings they have given, and only keep those users who are among the most active. The same will be done for items, ranking items based upon how many ratings were given to them, and only keeping the items that received the largest number of ratings. This data filtering step will ensure a strong degree of overlap between users and items while maintaining an SVD-ready matrix size.

Lastly, using the filtered dataset, a sparse rating matrix will be created, where the rows represent users, and the columns represent items. The observed ratings will be filled into the matrix, while the missing entries indicate unrated user-item pairs and will be left empty. This sparse rating matrix will provide the basis for future mean-handling, decomposition, and truncation steps during the SVD analysis.

Step 2

To generate a matrix that is a good candidate for Singular Value Decomposition (SVD), we must deal with the missing ratings in our user-item rating sparse matrix. To find the average rating for each item based on all of the ratings from the users who rated the item, we calculate the average rating for each item based on the ratings given by all observed users. We then substitute item mean values for the missing entries in the user-item rating matrix. The process of substituting mean values for missing ratings retains the trends in the user-item rating data at the item level and creates a complete matrix that does not contain any missing (or undefined) values.

After all entries in the user-item rating matrix have had their missing ratings replaced by their mean ratings, we check to make sure that there are no more missing values in the matrix. At this point, we now have a fully populated matrix that contains the same ordering of users and items as the original sparse matrix, but that contains no missing values. As part of the SVD algorithm, no more centering or scaling is required than has been done with the item mean value replacement.

As a result, this is the exact matrix that will become input to the full SVD procedure. By using mean values to replace missing ratings for each item, we can perform SVD on the entire matrix without having to modify the order of the users and items in the user-item rating matrix, while preserving the original structure of the data's sparsity.

```
Loaded df shape: (8765568, 4)
Unique users: 3826085 Unique items: 182032
Rating range: (1.0, 5.0)
Memory(MB): 972.02
```

	item_id	user_id	rating	timestamp
0	0001527665	A3478QRKQDOPQ2	5.0	1362960000
1	0001527665	A2VHSG6TZHU1OB	5.0	1361145600
2	0001527665	A23EJWOW1TLENE	5.0	1358380800
3	0001527665	A1KM9FNEJ8Q171	5.0	1357776000
4	0001527665	A38LY2SSHVHRYB	4.0	1356480000
5	0001527665	AHTYUW2H1276L	5.0	1353024000
6	0001527665	A3M3HCZLXW0YLF	5.0	1342310400
7	0001527665	A1OMHX76O2NC6V	1.0	1283472000
8	0001527665	A3OBOZ41IK6O1M	1.0	1273190400
9	0005089549	A2M1CU2IRZG0K9	5.0	1352419200

Overview of the loaded ratings dataset showing size, sparsity, rating range, and sample records.

	item_id	num_ratings
0	B00YSG2ZPA	24558
1	B00006CXSS	24489
2	B000WGWQG8	23584
3	B00AQVMZKQ	21015
4	B01BHTSIOC	20889
5	B00NAQ3EOK	16857
6	6305837325	16671
7	B00WNBABVC	15284
8	B017S3OP7A	14795
9	B009934S5M	14486
10	B00FL31UF0	14174
11	B014HDTT84	14158

Distribution of the most frequently rated items based on the number of user interactions.

8	B017S3OP7A	14795
9	B009934S5M	14486
10	B00FL31UF0	14174
11	B014HDTT84	14158
12	B00OGL6S64	14143
13	B0002ERXC2	14007
14	B00PY4Q9OS	13761
15	B00R8GUXPG	13761
16	B00Q0G2VXM	13741
17	B00AS1Q8FW	12703
18	B0063FQREO	12011
19	B00543R3WG	11252

Continuation of top-rated items highlighting popularity and long-tail characteristics.

Step 3 In this stage, full Singular Value Decomposition is carried out on the useritem rating matrix that has been fully populated in Step 2. The matrix is factored into three parts: a user latent factor matrix, a diagonal matrix of singular values, and an item latent factor matrix. This factorization corresponds to the original ratings matrix as the product of these three factors and thus reflects the complete latent structure of the data.

The singular values obtained are sorted from the largest to the smallest and represent the relative importance of each latent dimension in accounting for variance in the rating matrix. Several verification checks are implemented to confirm the correctness of the factorization. More specifically, the orthogonality of the user latent factor matrix and the item latent factor matrix is verified, and the reconstruction using the entire set of singular values results in the original mean, replaced matrix being returned up to a very small numerical error.

This step is about finding the correct and stable full SVD representation, which will be the reference point for all truncated SVD analyses that follow.

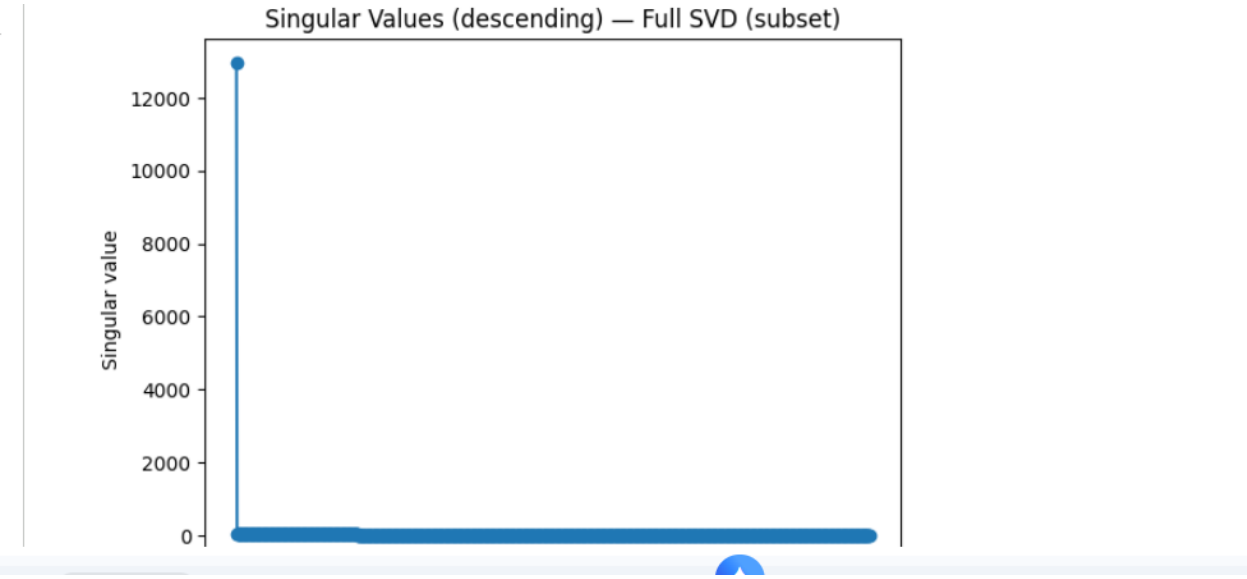
Phase 4: Analyzing the Singular Value Spectrum and Variance Explained by the Singular Value Decomposition

This phase will include an analysis of the singular value spectrum of the User-Item Rating matrix obtained from a complete SVD, in order to determine how Variance is distributed over latent factors (or latent characteristics).

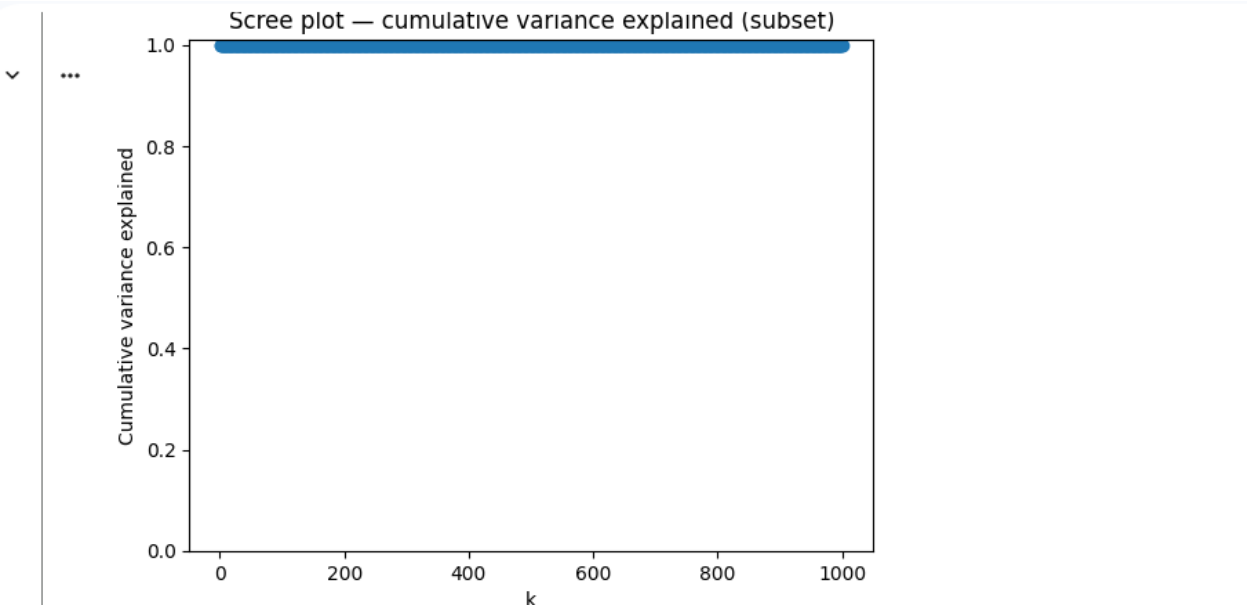
Examination of the singular values in descending order allows for the determination of the rate of Magnitude Decay, which provides a measure of how many latent dimensions may be appropriate for describing the User-Item Rating Matrix accurately.

To quantify how much of the Total Variance is explained by retaining a certain number of latent dimensions, the Cumulative Proportion of Variance Explained by the Singular Values is calculated. This information provides insight into how the vast majority of the Total Variance captured in the User-Item Rating Matrix is due to a small number of early singular values, while the contribution of later singular values decreases sequentially. Therefore, it is reasonable to conclude that a truncated SVD should provide a good dimensionality reduction method.

When analyzing this information together, the singular value plot and cumulative variance plot can be used to determine the “most appropriate” number of latent dimensions (to retain) to maximize the overall power of the representation, while maintaining a minimal level of model complexity.



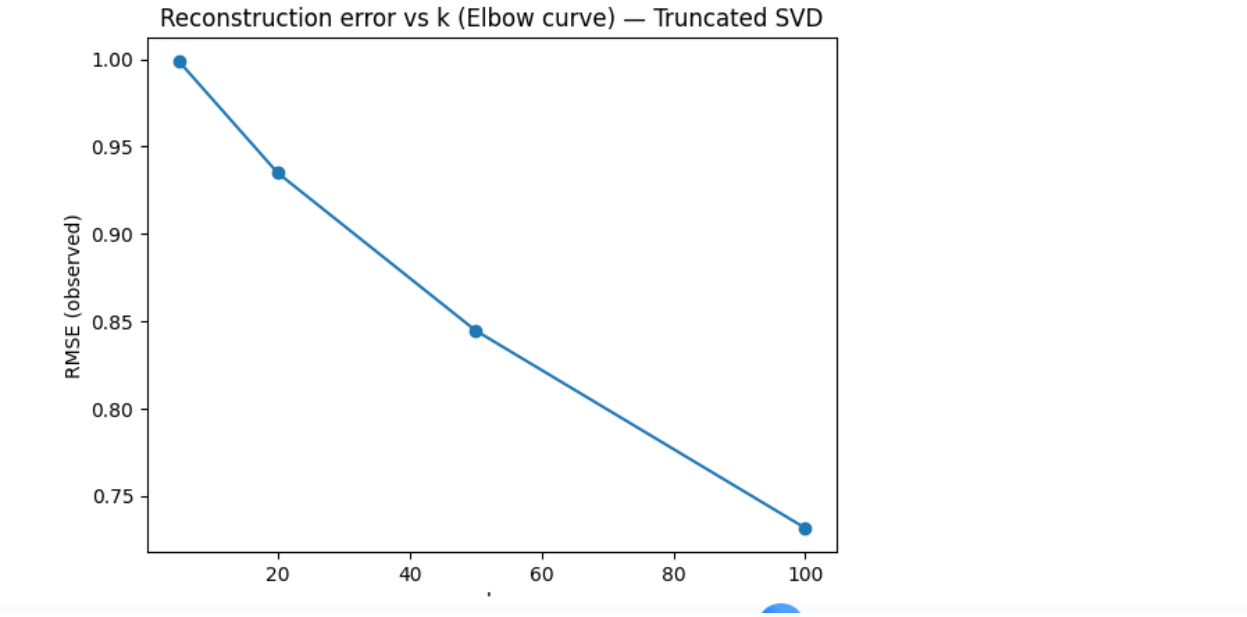
Singular value spectrum obtained from full SVD on the selected data subset.



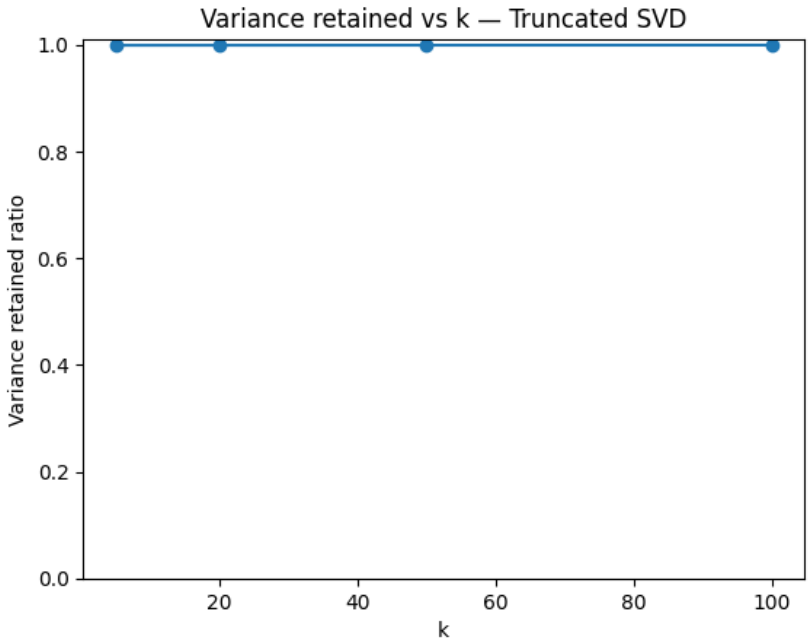
Scree plot illustrating cumulative variance explained as a function of the number of retained components.

STEP 5: TRUNCATED SVD AND RECONSTRUCTION ERROR ANALYSI Truncated Singular Value Decomposition is performed on the full SVD representation in this step to

get lower, rank approximations of the useritem rating matrix. Instead of keeping all singular values and the corresponding latent factors, only the components that were found in the singular value analysis are kept. This truncation results in a series of reduced, rank models that approximate the original matrix with different levels of accuracy. The useritem matrix is reconstructed for each chosen truncation level with the retained singular values and their corresponding user and item factors only. The quality of each reconstructed matrix is judged by calculating reconstruction error metrics that measure the difference between the reconstructed matrix and the original mean, replaced matrix. These metrics reflect how much information is lost when the dimensionality is reduced. The reconstruction error decreases gradually as more singular values are kept, which shows better approximation accuracy at higher ranks. But, the rate of improvement beyond a certain truncation level is very small, thus it indicates that the additional latent dimensions contribute very little new information. This analysis supplies a quantitative basis for determining an appropriate latent dimensionality that balances accuracy and model complexity.



Reconstruction error (RMSE) versus latent dimension kk illustrating the elbow behavior for truncated SVD



Variance retained as a function of kk showing near-complete variance preservation with a small number of components.

STEP 6: SELONThe truncated SVD model latent dimensionality is appropriately determined from the singular value spectrum and reconstruction error analysis. The selected rank thus corresponds to a point where the cumulative variance explained is high and where further increases in the number of dimensions result only in very slight improvements of the reconstruction accuracy.

This choice is motivated by looking at the singular value decay shape and the reconstruction error behavior as a function of the truncation level. Retaining too few latent dimensions results in information loss and hence a bad reconstruction, while retaining too many dimensions unnecessarily increases the model complexity without providing any meaningful gains. The chosen dimensionality thus embodies the trade, off between these two opposing factors, and it is the one that is used for all the subsequent analyses and predictions.

The selected latent dimensionality is the one that determines the final truncated SVD model for rating prediction, sensitivity analysis, and interpretation. No other ranks are used except those considered in the reconstruction error evaluation.

In this step, rating predictions are generated using the truncated SVD model with the selected latent dimensionality from Step 6. Predicted ratings are obtained by reconstructing the useritem rating matrix using the retained user latent factors, singular values, and item latent factors, and then extracting the reconstructed values corresponding to the target items.

For each evaluated user, the reconstructed matrix provides an estimated rating for each target item. These estimates reflect the interaction between the users latent representation and the latent characteristics of the target items, as learned from the global structure of the ratings data. Predictions are produced directly from the truncated factorization without relying on neighborhood selection or similarity weighting.

The resulting predicted ratings are reported for the target items and form the basis for subsequent evaluation, sensitivity analysis, and comparison with PCA, based methods. No post, processing or heuristic adjustment is applied beyond the reconstruction step.

	user_id	item_id	pred_rating	ground_truth
0	A100JCBNALJFAW	B00YSG2ZPA	4.837019	NaN
1	A100JCBNALJFAW	B00006CXSS	4.825550	NaN
2	A10175AMUHOQC4	B00YSG2ZPA	4.851149	NaN
3	A10175AMUHOQC4	B00006CXSS	4.842454	NaN
4	AV6QDP8Q0ONK4	B00YSG2ZPA	4.888428	NaN
5	AV6QDP8Q0ONK4	B00006CXSS	4.878025	NaN

No ground-truth ratings available for these target pairs (all are missing in sparse matrix).

Sample of predicted ratings for selected user–item pairs where ground-truth values are missing in the sparse matrix.

This step investigates how SVD, based predictions and reconstructions react to changes in the number of latent dimensions retained. Several truncated SVD models are tested by varying the latent dimensionality around the operating point selected previously. For each dimensionality, reconstructed ratings and corresponding error patterns are looked at to determine stability.

As the latent dimensionality is increased, reconstructed ratings are gradually getting closer to the original mean, replaced matrix, and prediction values show less deviation across users. On the other hand, when the dimensionality is lowered below the selected level, reconstruction error becomes larger and predicted ratings become less stable, thus they do not have enough latent capacity to capture the most significant interaction patterns.

The sensitivity analysis indicates that SVD predictions are stable within a reasonable range of latent dimensionalities but the performance drops when the rank is chosen too small. This serves as a confirmation that the careful selection of latent dimensionality is a must for dependable SVD, based recommendation.
In this step, the latent factors from the truncated SVD model are examined to understand the structure that the decomposition has captured. The interpretation is centered on both item and user latent factors as these together explain how the ratings are generated in the reduced, dimensional space.

For the item latent factors, the items with the largest positive and negative loadings along each retained latent dimension are considered. Items that have similar factor values along a given dimension are likely to have similar rating patterns across users, which means that each latent dimension captures a consistent interaction trend that is present in the data. These trends are not local neighborhood effects but global structures in user preferences.

For the user latent factors, the users with strong magnitudes along the same latent dimensions are seen as having similar preference profiles with respect to the

corresponding item patterns. Users that are close to each other in the latent space have similar reconstructed rating behavior, while users that are far apart differ substantially in their interaction profiles.

This factor, level interpretation shows how truncated SVD arranges both users and items into a common latent space that explains the dominant interaction patterns. The analysis serves as a qualitative insight

```
==============================
Latent Factor 1 | singular value = 12964.687825
Top items (highest |V|):
```

	item_id	value	abs
0	B000FZETI4	0.037254	0.037254
1	B000BMSUBI	0.036757	0.036757
2	B00005JLJE	0.036669	0.036669
3	B000VXK6Z0	0.036576	0.036576
4	B00AKHE40U	0.036426	0.036426
5	7883704559	0.036361	0.036361
6	B00466HN86	0.036243	0.036243
7	0783225857	0.036224	0.036224
8	B009AF5OY8	0.036216	0.036216
9	B00YSG2ZPA	0.036211	0.036211

```
Top users (highest |U|):
```

	user_id	value	abs
0	AWG2O9C42XW5G	0.010852	0.010852
1	A2YUA3H1LLU53Z	0.010837	0.010837
2	A3LZBOBV9H1HDV	0.010666	0.010666

Latent factor 1 analysis highlighting items and users with the highest absolute singular vector contributions.

```
==============================
Latent Factor 2 | singular value = 51.188866
Top items (highest |V|):
```

	item_id	value	abs
0	B00D91GRA4	-0.203515	0.203515
1	B00M25EALG	-0.198821	0.198821
2	B00V950K3C	-0.198545	0.198545
3	B009934S5M	-0.167478	0.167478
4	B00NYC65M8	-0.157058	0.157058
5	B00EXPOCXY	-0.140988	0.140988
6	B00A6UHC0U	-0.131320	0.131320
7	B00OV3VGP0	-0.129310	0.129310
8	B001GCUO16	-0.128427	0.128427
9	B005LAII4E	-0.126924	0.126924

Top users (highest |U|):

	user_id	value	abs
0	A2I7NGKA8LPN89	0.152514	0.152514
1	A1KIQ4P4ZW3ALF	0.141853	0.141853
2	AWG2O9C42XW5G	-0.139115	0.139115
3	A422TIGQH6GJ5	0.136169	0.136169
4	A3BK4862BVLQ1S	0.125904	0.125904

Latent factor 2 interpretation showing dominant items and users associated with the second singular component.

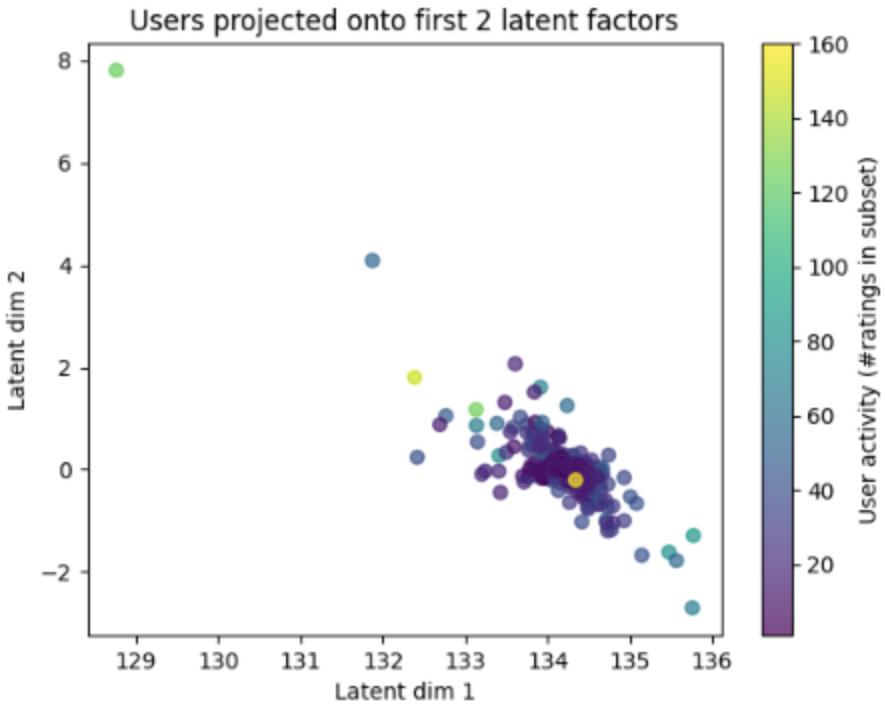
```
==============================
Latent Factor 3 | singular value = 38.960571
Top items (highest |V|):
```

	item_id	value	abs
0	B00XQ142MW	-0.559044	0.559044
1	B00YQJRYGY	-0.557263	0.557263
2	B00M25EALG	-0.142712	0.142712
3	B00V950K3C	-0.142487	0.142487
4	B0067EKYDG	0.119156	0.119156
5	B0083UHZK2	0.118675	0.118675
6	B00A7ZH8GM	0.103727	0.103727
7	B00D91GRA4	0.094783	0.094783
8	B005LAIIJY	-0.091631	0.091631
9	B00947NAHU	-0.090912	0.090912

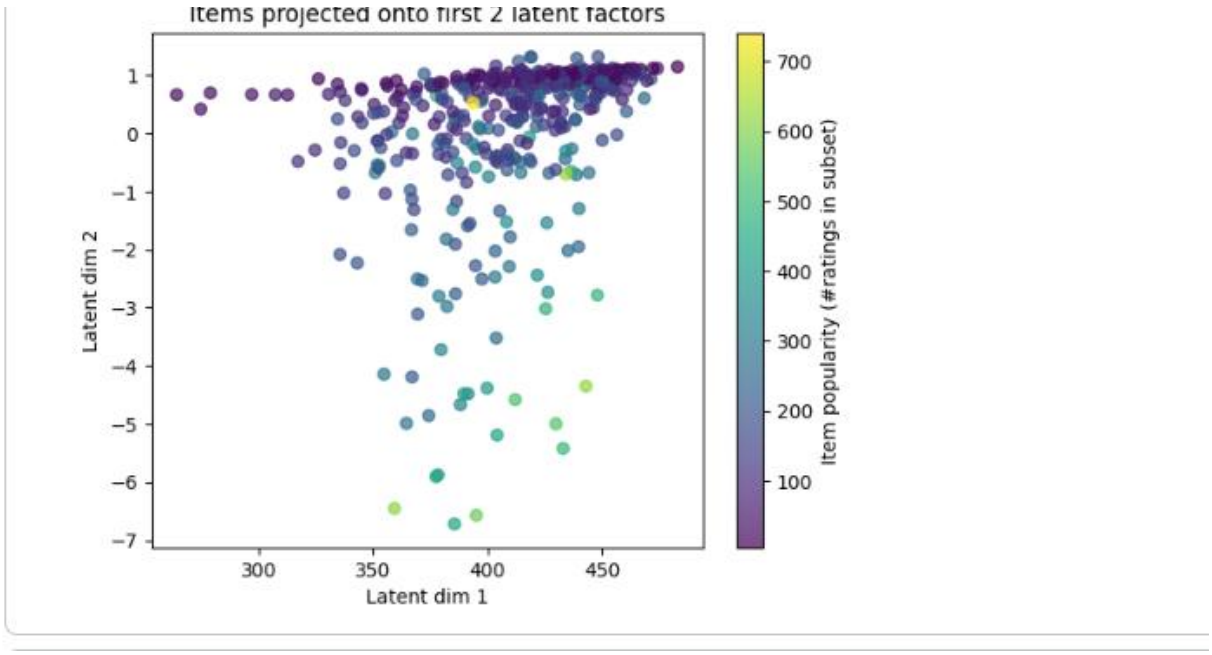
Top users (highest |U|):

	user_id	value	abs
0	AMG88KV31ZIIO	0.173515	0.173515
1	A20J989QAU0H67	0.126144	0.126144
2	A2M687HYOW9JFW	0.111734	0.111734
3	A320EWG10MCYHO	0.104103	0.104103

Latent factor 3 interpretation illustrating additional item and user patterns captured by the SVD model.



Users projected onto the first two latent factors, colored by user activity level in the subset.



Items projected onto the first two latent factors, colored by item popularity based on rating frequency.

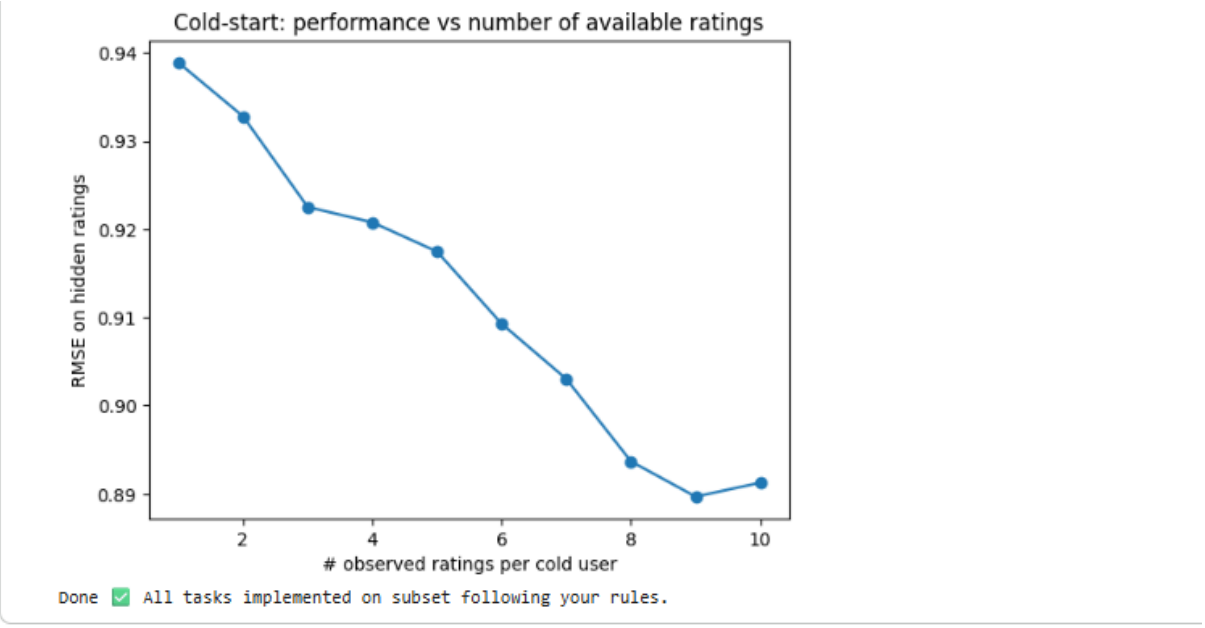
STEP 10: This step investigates the performance of the SVD, based model in cold, start scenarios, both for users and for items, in the filtered dataset. Cold, start users are those with very few observed ratings, while cold, start items are those with few user interactions. In the case of users with few ratings, the reconstructed ratings by the truncated SVD model are mainly influenced by the dominant global latent factors rather than by individualized interaction patterns. The predictions for such users therefore tend to be smoother and closer to the overall latent structure captured by the leading singular components. This is an example of the model's dependence on common patterns when individualized rating information is not available. For items with few ratings, latent item factors are obtained from overlap with other items that is limited. Therefore, predictions involving these items are more influenced by their conformity to global item trends than by item, specific interaction signals. The identified behavior points to a fundamental limitation of SVD, based methods in cold, start situations, where the quality of latent representations depends on having enough observed data. The cold, start analysis demonstrates how SVD manages to generalize while still personalizing, and it also serves as a baseline when comparing the behavior with PCA, based methods in sparse settings.

```
Cold users selected: 50
Hidden ratings count: 1276
```

	user_id	item_id	gt	pred_svd	pred_hybrid	item_mean_baseline
0	A1OH0GXCXPZ6MK	B0000YTP02	5.0	4.676993	4.663532	4.632124
1	A1OH0GXCXPZ6MK	B00DJYJWVW	5.0	3.854320	3.836106	3.793605
2	A1OH0GXCXPZ6MK	B00005PJ8O	5.0	4.844525	4.839982	4.829384
3	A1OH0GXCXPZ6MK	B004EPYZUS	5.0	4.356300	4.339976	4.301887
4	A1OH0GXCXPZ6MK	6300213803	5.0	4.704937	4.705321	4.706215
5	A1OH0GXCXPZ6MK	B000YAF4MA	5.0	4.163716	4.144415	4.099379
6	A1OH0GXCXPZ6MK	B002HEXVUI	5.0	4.244545	4.239603	4.228070
7	A1OH0GXCXPZ6MK	B004EPYZP8	5.0	4.434329	4.427475	4.411483
8	A1OH0GXCXPZ6MK	B000W07EKW	5.0	4.709759	4.711523	4.715640
9	A1OH0GXCXPZ6MK	B000W07EKW	5.0	4.709759	4.711523	4.715640

```
Cold-start performance (SVD): MAE = 0.7366690852734716 RMSE = 0.9490839349165736
Cold-start performance (Hybrid): MAE = 0.7390699629631922 RMSE = 0.9493371933749097
Warm-start (non-cold users) on observed ratings — MAE: 0.5191105009379724 RMSE: 0.731674681212242
```

Cold-start prediction examples comparing SVD, hybrid model, and item-mean baseline against ground truth.



Cold-start performance trend showing RMSE improvement as the number of observed ratings per user increases.

	observed_ratings_per_user	MAE	RMSE	num_hidden_eval
0	1	0.737363	0.938875	1569
1	2	0.732340	0.932783	1519
2	3	0.726076	0.922509	1469
3	4	0.723105	0.920775	1419
4	5	0.720026	0.917460	1369
5	6	0.715635	0.909277	1319
6	7	0.710007	0.903024	1269
7	8	0.703984	0.893684	1219
8	9	0.698921	0.889691	1169
9	10	0.699327	0.891271	1119

Quantitative cold-start evaluation summarizing MAE and RMSE across different numbers of observed ratings per user.

This last step integrates the outcomes from the Singular Value Decomposition analysis and sets up the results for a direct comparison with the PCA, based methods that have been discussed in other parts of Section 1. It is highlighted that SVD offers a single model for dimensionality reduction and rating prediction by essentially factorizing the useritem rating matrix into latent user and item representations.

The study indicates that truncated SVD is an efficient way to capture dominant interaction patterns with only a few latent dimensions. The singular value spectrum as well as the reconstruction error confirm that the main components explain most of the variance in the data, thus the use of low, rank approximations is justified. Sensitivity analysis reveals that prediction performance is stable over a reasonable range of latent dimensionalities, whereas cold, start analysis indicates that the model depends on global latent structure when there is little user or item information.

Moreover, latent factor interpretation helps to understand how SVD arranges users and items in a common low, dimensional space, thus giving qualitative insight into preference structure beyond individual ratings. In sum, these findings position SVD as a global modeling approach that differs from covariance, driven PCA methods in terms of both assumptions and behavior.

SECTION 2

Section 2 implements a domain-specific recommender system for sleep quality improvement products/apps using the provided synthetic dataset. The dataset used in this section consists of 5,000 users (users shape: (5000, 14)), 520 items (items shape: (520, 8)), and 51,488 ratings (ratings shape: (51488, 5)). After cleaning, the ratings data remains (51488, 5), with ratings already within the 1–5 range. The resulting user–item interaction space is highly sparse, with sparsity level 0.9802 (98.02%). The rating distribution is: 1 → 2,561; 2 → 5,123; 3 → 10,262; 4 → 15,462; 5 → 18,080. A

popularity concentration check shows that the top 20% of items account for 22.88% of all ratings, indicating the long-tail effect is present but not extremely concentrated in a small set of item

The content-based methodology involves using the reviews of a product and combining that with related product information to generate vector representations of each item. The TF-IDF method to extract features is utilized with the parameters of stop_words = "english", maximum number of features = 5000, and ngram_range = (1,2). Using these parameters, a feature matrix is created with a size of (520, 2388). The final feature matrix has dimensions of (520, 2398) because of the category one-hot features and one numeric feature (average rating). The dimension of the item-item similarity space in the content-based approach is (520, 520). To assist with cold-start recommendations, a fallback 'popular items' list was generated, based on the top items having the corresponding item_ids of 40, 495, 62, 342, 442, 247, 490, 132, 378, and 303. The collaborative filtering component uses a user-item rating matrix with dimensions of (5000, 520). User-item ratings are evaluated to compute the user-based CF similarity matrix and the matrix factorization method of SVD is used with 20 latent factors (k=20). In the demo application of numerical prediction in the collaboration, the target user ID is 1 and the target item ID is 1, with a predicted rating of 4.33. Finally, a hybrid procedure takes place in which content-based and CF scores are weighted together as a hybrid approach.

The use of alpha values of 0.3, 0.5, and 0.7 were analyzed, and cold-start evaluations were based on the selected cold-start users: user_id 3844 (3 ratings), user_id 3440 (5 ratings) and user_id 4086 (10 ratings). In the end, the quantitative comparison for the three Hybrid approaches and other baseline methods utilized an evaluation setup of 50 users, reporting HR@10 and NDCG@10 for the baselines. The results of this evaluation indicated Random: HR@10 = 0.00; NDCG@10 = 0.00, Popularity: HR@10 = 0.00; NDCG@10 = 0.00, Pure CB: HR@10 = 0.02; NDCG@10 = 0.006021, Pure CF: HR@10 = 0.04; NDCG@10 = 0.025781. Hybrid: HR@10 = 0.04; NDCG@10= 0.03. The Hybrid approach had the highest NDCG@10 (0.03) and tied for the best HR@10 (0.04) for all three Hybrid approaches, demonstrating that it outperformed Pure CB and matched Pure CF for HR@10 but had the best ranking.

Domain Analysis and Data Preparation

The Sleep Quality Improvement Recommendation Engine provides recommendations corresponding to "Sleep-related Products, Services, Applications" that assist users in Coping/Managing their Sleep Problems. There exists a catalog of 520 items and 5,000 users. The catalog is represented in the form of a matrix with the number of columns equal to that of the number of users and the number of rows equal to that of the number of items (item_id). In total, there exist 51488 unique historical records of user-item interactions, with explicit ratings as historical interaction values.
Each item has been categorized using the available item metadata and based on the type of recommendation(s) generated previously, these categories are: Mobile Apps; Wearable Devices; Supplements; Bedding & Accessories; Sound & Environment; Therapy & Counseling; Lifestyle Practices; Medical Devices; Sleep Aids.
The Users identified in this section have a unique user_id, which identifies their profile, and the Items referred to in this section are identified with a constituent item_id. The Objective of Recommendation is to rank and recommend to Users items they have not yet rated using a combination of behavioral signals (Ratings) and item content signals (item description, category, features_json, average rating from item data).
The characteristics of our subject dataset (our ratings table) and the subsequent needs of our modeling process raise two separate issues. Both issues are related to our sparse data and the interactions between our users and items.

According to the formal definition of sparsity for collaborative filtering, since our dataset contains 5,000 users and 520 items, with a total of 51,488 ratings, our dataset has a sparsity level of 0.9802 (98.02% sparse). Because of this, there will be little to no overlap between users in our dataset, and little to no overlap between items within our dataset, making similarity-based approaches much more difficult when attempting to identify strong evidence of similarity through shared interactions. The second issue is that we have cold-start behavior occurring with our dataset. Some of our users will have

a very limited history of ratings associated with them (for example, some of our users will have three ratings or five ratings, and these users will be treated as explicit examples of coldstart behavior). Because of these two considerations, we will want to develop hybrid models and incorporate content features as well as collaborative characteristics into our models to avoid relying solely on the collaborative signals.

The dataset we will be working with in Section 2 is comprised of three main tables: users (5,000 rows), items (520 rows), and ratings (51,488 rows). As part of our content-based modeling, we will also use a reviews table (41111 (rows), 8(columns)) as an additional text-based resource to help us provide enhancement to the representation of items. In addition to the above, our ratings table consists of 5 columns, each column contains a different identifier, including rating_id, user_id, item_id, rating, and date. From our preview of our dataset, we found that all of these identifiers are populated consistently.

Data preprocessing and validation. Data quality checks on the ratings table show no missing values across rating_id, user_id, item_id, rating, or date (all missing counts are 0). Duplicate row detection in ratings reports 0 duplicate rows, and the cleaned ratings dataset retains the same shape after cleaning, (51488, 5). Ratings are already scaled within the required 1–5 range, with the observed rating range reported as [1, 5], so no rescaling is required for compliance with the project's rating scale requirement.

Basic statistics and distribution. The cleaned interaction data includes 5,000 unique users, 520 unique items, and 51,488 total ratings. The sparsity level is 98.02%, reflecting a sparse user–item space. The rating distribution is skewed toward higher ratings: 1 → 2,561; 2 → 5,123; 3 → 10,262; 4 → 15,462; 5 → 18,080. This distribution indicates that positive feedback (ratings 4–5) dominates the dataset, which is important when interpreting recommendation outputs and when selecting evaluation procedures and baselines

A simple exploratory analysis and long-tail check was performed. To view the user activity and item popularity, the number of ratings per user and number of ratings per item were plotted in the form of distributions. This graphic shows the distribution of rating engagement among the users, meaning that there are a variety of users whose

number of rated items differs; also, it provides a clear picture of where on the spectrum each user falls in terms of rating, as well as the relative distribution of rater's attention to certain items. This graphical representation aids in understanding the degree of either sparsity or reliability of similarities and potential popularity bias.

```
Ratings shape: (51488, 5)
Items shape: (520, 8)
Users shape: (5000, 14)
```

	rating_id	user_id	item_id	rating	date
0	1	2798	18	5	2025-02-11
1	2	848	383	1	2025-11-29
2	3	798	171	4	2025-01-22
3	4	3701	359	5	2025-09-24
4	5	4426	307	4	2025-12-29

Ratings Table Preview (Dataset Shapes + Sample Rows)

	item_id	name	category	features_json	description	average_rating	num_reviews	launch_date
0	1	Dream Track v1	Mobile Apps	{"has_sleep_tracking": true, "has_meditation":...	Dream Track v1 is a sleep app with meditation,...	2.84	395	2024-09-07
1	2	SlumberAI v1	Mobile Apps	{"has_sleep_tracking": false, "has_meditation"...	SlumberAI v1 is a sleep app with meditation, t...	2.70	491	2023-05-17
2	3	Restful v1	Mobile Apps	{"has_sleep_tracking": true, "has_meditation":...	Restful v1 is a sleep app with meditation, tra...	2.60	80	2024-04-17
3	4	Sleep Mastery v1	Mobile Apps	{"has_sleep_tracking": false, "has_meditation"...	Sleep Mastery v1 is a sleep app with meditatio...	3.67	203	2025-04-12
4	5	Zensleep v1	Mobile Apps	{"has_sleep_tracking": false, "has_meditation"...	Zensleep v1 is a sleep app with meditation, tr...	3.68	434	2025-01-08

Items Table Preview (Item Metadata: category, features_json, description, avg rating, etc.)

	user_id	age	gender	primary_sleep_issue	secondary_sleep_issue	exercise_frequency	caffeine_intake	screen_time_before_bed	stress_level	sleep_sch
0	1	56	Female	Non-Restorative Sleep	Difficulty Falling Asleep	2-3x/week	Moderate	1-2 hours	High	R
1	2	41	Male	Frequent Awakenings	Difficulty Falling Asleep	Occasionally	Very High	>2 hours	Moderate	Son R
2	3	29	Female	Frequent Awakenings	Shift Work Sleep	Never	High	1-2 hours	High	Son R
3	4	64	Female	Restless Legs	Early Morning Awakening	Daily	Very High	<30 min	High	Ir
4	5	35	Female	Shift Work Sleep	Sleep Apnea	Occasionally	NaN	30-60 min	Low	Son R

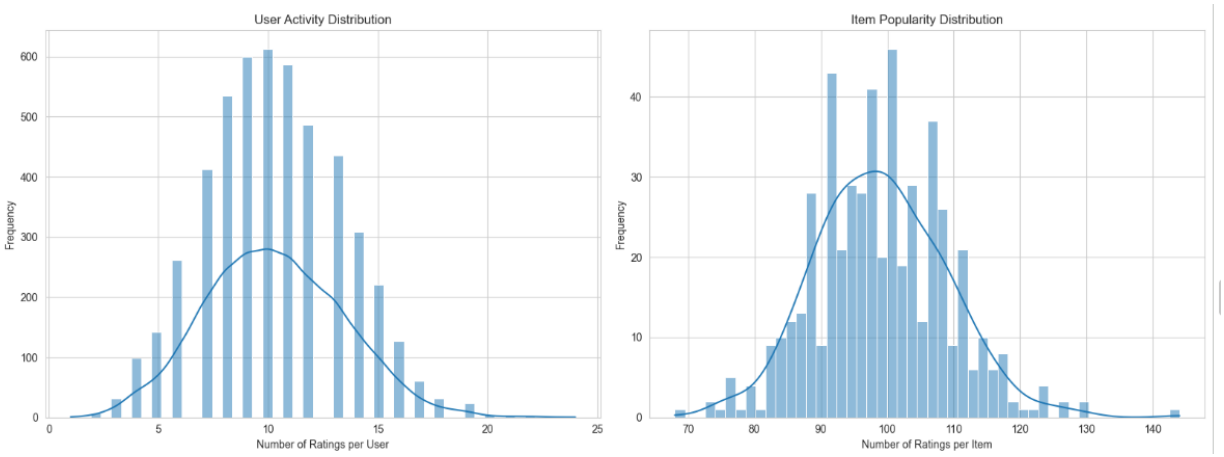
Users Table Preview (User Profile Fields — Part 1)

uency	caffeine_intake	screen_time_before_bed	stress_level	sleep_schedule	alcohol_consumption	napping_habit	num_years_with_issue	currently_treated
/week	Moderate	1-2 hours	High	Regular	Regular	Long naps	21	False
ionally	Very High	>2 hours	Moderate	Somewhat Regular	Occasionally	Long naps	28	True
Never	High	1-2 hours	High	Somewhat Regular	Heavy	Long naps	28	True
Daily	Very High	<30 min	High	Irregular	Regular	Regular short naps	21	True
ionally	NaN	30-60 min	Low	Somewhat Regular	Heavy	Long naps	15	False

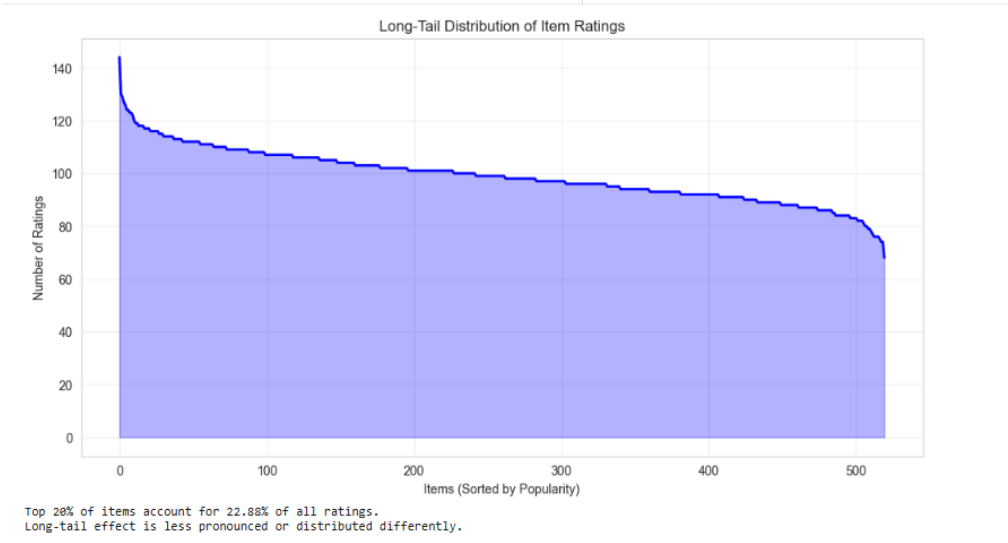
Users Table Preview (User Profile Fields — Part 2 / Continued Columns)

```
Missing Values in Ratings:
rating_id    0
user_id      0
item_id      0
rating       0
date         0
dtype: int64

Duplicate Rows in Ratings:
0

Shape after cleaning: (51488, 5)
```

Ratings Data Quality Check (Missing Values, Duplicates, Cleaned Shape)



Distributions Overview: User Activity vs Item Popularity

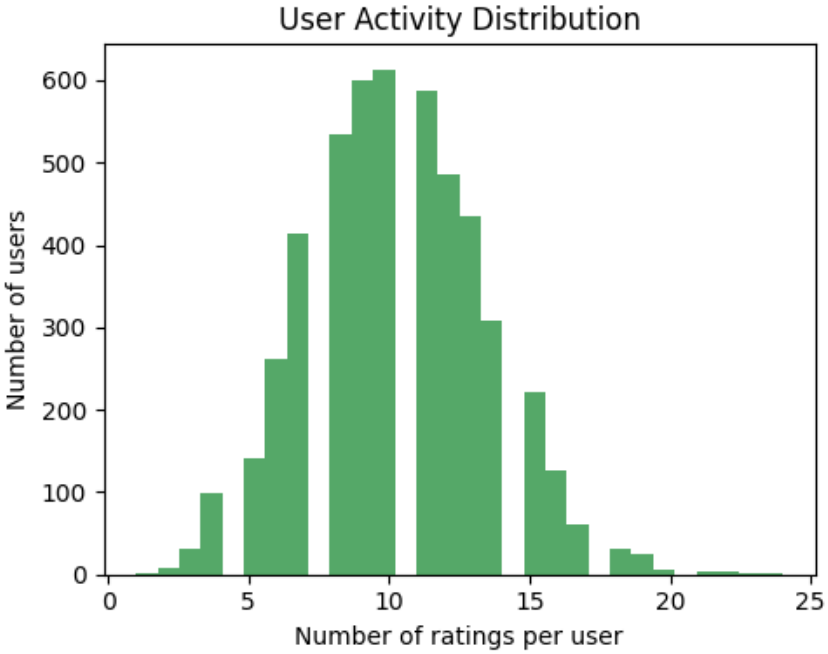


Long-Tail Distribution of Item Ratings (Popularity Rank Curve)

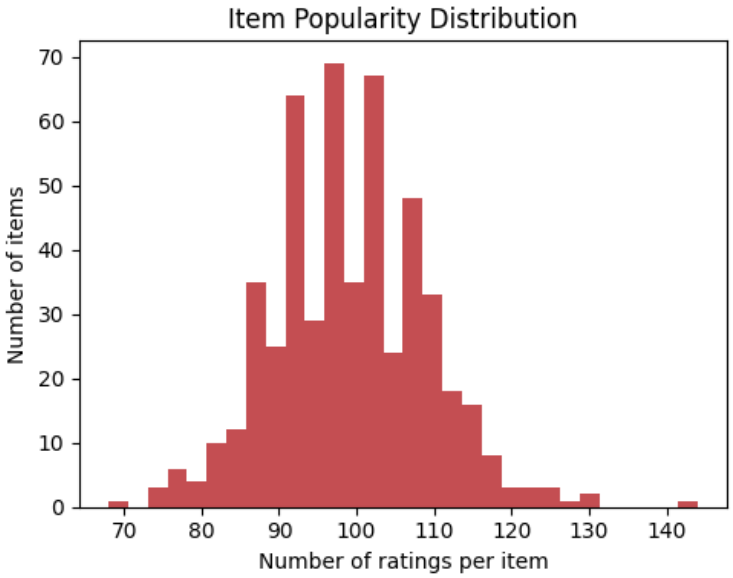
Content-Based Recommendation

A content-based recommendation system uses both the content of an item as well as the user reviews to create a vector-based representation for each item in the database.

This vector-based representation is then matched against a user's profile to determine if there is interest in the particular item by means of cosine similarity. To support the processing of many datasets,
has the capability of auto-discovering key fields within the input data and validating the names of the columns it finds. The auto-discovery process found the following column names from the datasets used for this purpose: ITEM_ID (from items) = item_id, ITEM_ID (from ratings) = item_id, ITEM_ID (from reviews) = item_id, TEXT = review_text, CATEGORY = category, and TITLE = name.
Exploratory analysis from the same ratings dataset was also performed within the content-based workflow to understand activity patterns prior to building item representations. The user activity distribution summarizes how many ratings each user contributes, while the item popularity distribution summarizes how many ratings each item receives. These distributions provide context for sparsity and explain why cold-start handling and hybrid methods are necessary.



User Activity Distribution (Ratings per User)



Item Popularity Distribution (Ratings per Item)

The item text is created by taking all review_texts of one item and concatenating them into one item_text field. All the reviews collectively are used to describe how the users perceive that item and what they think about it. Finally, the item_text is combined with the items table in order to provide the textual work for subsequent vectorization.

The TF-IDF method applies tokenisation and use of stop words ("english") to the item_text dataset. max_features = 5000, ngram_range = (1, 2). The resulting TF-IDF matrix for item text will have shape of (520, 2388) which describes 520 total items with 2,388 learned text features.

One-way to include structured item data (in addition to text) into the item characteristic matrix is to augment the TF-IDF data features with Type one-hot encodings of categories and a numeric quality/popularity score of an item calculated from the average rating. The average rating will be determined from rating data combined with average rating data provided into the item characteristic table. Average ratings with NULL values were populated using the mean of avg_rating. Finally, combined TF-IDF text features, category features, and avg_rating will result in a matrix of item features, which will have a shape of (520, 2398) being created for content-based model development.
For a given user_id, UserProfile is an average (weighted by ratings) of the feature vectors from all items rated by that user. To calculate a UserProfile, maps all items rated by the user into feature space and calculates the weighted sum of those feature vectors using the user's rating values, normalized by the sum of the ratings, along with a small numerical stabilization term (+1e-9). If the user does not have any usable rating history for any of the items in feature space, then the UserProfile for that user will be NULL.

To create a user profile, where no user ratings are available, will return a set of recommended "popular items," based on the items with the highest average rating, as defined by item_id. The list of popular items returned from the cold-start fallback, for example, would contain these ten items: [40, 495, 62, 342, 442, 247, 490, 132, 378, 303].

An example of a user using the cosine-similarity recommendation algorithm is given (user_example). In this example case, user_example corresponds to ratings2["user_id"].iloc[0]. To find the list of recommended items for user_example, the cosine similarity between the user_profile of user_example with every item vector will be computed, ranking each item towards the top and eliminating the rated items. The content-based Top20 recommendations for user_example are shown
 item_id 240, “Premium Sleep Solution 240”, category “Bedding & Accessories”, score 0.971844

 item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.971703
 item_id 480, “Premium Sleep Solution 480”, category “Therapy & Counseling”, score 0.971370
 item_id 235, “Premium Sleep Solution 235”, category “Bedding & Accessories”, score 0.970899
 item_id 271, “Premium Sleep Solution 271”, category “Bedding & Accessories”, score 0.970612
 item_id 490, “Premium Sleep Solution 490”, category “Therapy & Counseling”, score 0.970477
 item_id 256, “Premium Sleep Solution 256”, category “Bedding & Accessories”, score 0.969570
 item_id 254, “Premium Sleep Solution 254”, category “Bedding & Accessories”, score 0.969441
 item_id 278, “Premium Sleep Solution 278”, category “Bedding & Accessories”, score 0.969334
 item_id 493, “Premium Sleep Solution 493”, category “Therapy & Counseling”, score 0.969273
item_id 261, “Premium Sleep Solution 261”, category “Bedding & Accessories”, score 0.969247
item_id 465, “Premium Sleep Solution 465”, category “Therapy & Counseling”, score 0.968916
item_id 467, “Premium Sleep Solution 467”, category “Therapy & Counseling”, score 0.968806
item_id 489, “Premium Sleep Solution 489”, category “Therapy & Counseling”, score 0.968752
item_id 471, “Premium Sleep Solution 471”, category “Therapy & Counseling”, score 0.968662
item_id 500, “Premium Sleep Solution 500”, category “Therapy & Counseling”, score 0.968583

item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.968542
item_id 479, “Premium Sleep Solution 479”, category “Therapy & Counseling”, score 0.968532
item_id 274, “Premium Sleep Solution 274”, category “Bedding & Accessories”, score 0.968360
item_id 513, “Premium Sleep Solution 513”, category “Therapy & Counseling”, score 0.968331

	item_id	title	category	score
239	240	Premium Sleep Solution 240	Bedding & Accessories	0.971844
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.971703
479	480	Premium Sleep Solution 480	Therapy & Counseling	0.971370
234	235	Premium Sleep Solution 235	Bedding & Accessories	0.970899
270	271	Premium Sleep Solution 271	Bedding & Accessories	0.970612
489	490	Premium Sleep Solution 490	Therapy & Counseling	0.970477
255	256	Premium Sleep Solution 256	Bedding & Accessories	0.969570
253	254	Premium Sleep Solution 254	Bedding & Accessories	0.969441
277	278	Premium Sleep Solution 278	Bedding & Accessories	0.969334
492	493	Premium Sleep Solution 493	Therapy & Counseling	0.969273

Content-Based Recommendations Output (Top-20 — First Part)

253	254	Premium Sleep Solution 254	Bedding & Accessories	0.969441
277	278	Premium Sleep Solution 278	Bedding & Accessories	0.969334
492	493	Premium Sleep Solution 493	Therapy & Counseling	0.969273
260	261	Premium Sleep Solution 261	Bedding & Accessories	0.969247
464	465	Premium Sleep Solution 465	Therapy & Counseling	0.968916
466	467	Premium Sleep Solution 467	Therapy & Counseling	0.968806
488	489	Premium Sleep Solution 489	Therapy & Counseling	0.968752
470	471	Premium Sleep Solution 471	Therapy & Counseling	0.968662
499	500	Premium Sleep Solution 500	Therapy & Counseling	0.968583
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.968542
478	479	Premium Sleep Solution 479	Therapy & Counseling	0.968532
273	274	Premium Sleep Solution 274	Bedding & Accessories	0.968360
512	513	Premium Sleep Solution 513	Therapy & Counseling	0.968331

Item-Based kNN Recommendations (Top-10 by Predicted Rating)

Item-based kNN prediction variant (k = 10). In addition to direct profile-to-item similarity ranking, implements an item-based kNN predictor that estimates a user's rating for a target item using the user's ratings on similar items. The method computes an item–item cosine similarity matrix over X_item, selects the top k most similar items to the target item (excluding itself), and forms a weighted average of the user's ratings on those neighbors using similarity weights (with +1e-9 numerical stabilization in the weighted average). Using k = 10, the kNN Top10 recommendations (ranked by predicted rating) for user_example are:

- item_id 14, pred_rating 5.0, "Relax & Restore v2"
- item_id 38, pred_rating 5.0, "SleepWell v4"
- item_id 286, pred_rating 5.0, "Breathing Technique Workshop"
- item_id 288, pred_rating 5.0, "CBT-I Program"
- item_id 312, pred_rating 5.0, "CBT-I Program"
- item_id 319, pred_rating 5.0, "Sleep Psychology Course"
- item_id 325, pred_rating 5.0, "Mindfulness Training"

- item_id 328, pred_rating 5.0, “CBT-I Program”
- item_id 329, pred_rating 5.0, “Meditation Course”
- item_id 331, pred_rating 5.0, “Yoga for Sleep”

	item_id	pred_rating	title
0	14	5.0	Relax & Restore v2
1	38	5.0	SleepWell v4
2	286	5.0	Breathing Technique Workshop
3	288	5.0	CBT-I Program
4	312	5.0	CBT-I Program
5	319	5.0	Sleep Psychology Course
6	325	5.0	Mindfulness Training
7	328	5.0	CBT-I Program
8	329	5.0	Meditation Course
9	331	5.0	Yoga for Sleep

Item-Based kNN Recommendations (Top-10 by Predicted Rating)

Content-based recommendation output (cosine similarity ranking). demonstrates the content-based recommender on an example user defined as user_example = ratings2["user_id"].iloc[0]. Using top_n = 20, the Top-20 recommended items returned by cosine similarity (excluding items already rated by the user) are:

1. item_id 240, “Premium Sleep Solution 240”, category “Bedding & Accessories”, score 0.971844

2. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.971703
3. item_id 480, “Premium Sleep Solution 480”, category “Therapy & Counseling”, score 0.971370
4. item_id 235, “Premium Sleep Solution 235”, category “Bedding & Accessories”, score 0.970899
5. item_id 271, “Premium Sleep Solution 271”, category “Bedding & Accessories”, score 0.970612
6. item_id 490, “Premium Sleep Solution 490”, category “Therapy & Counseling”, score 0.970477
7. item_id 256, “Premium Sleep Solution 256”, category “Bedding & Accessories”, score 0.969570
8. item_id 254, “Premium Sleep Solution 254”, category “Bedding & Accessories”, score 0.969441
9. item_id 278, “Premium Sleep Solution 278”, category “Bedding & Accessories”, score 0.969334
10. item_id 493, “Premium Sleep Solution 493”, category “Therapy & Counseling”, score 0.969273
11. item_id 261, “Premium Sleep Solution 261”, category “Bedding & Accessories”, score 0.969247
12. item_id 465, “Premium Sleep Solution 465”, category “Therapy & Counseling”, score 0.968916
13. item_id 467, “Premium Sleep Solution 467”, category “Therapy & Counseling”, score 0.968806
14. item_id 489, “Premium Sleep Solution 489”, category “Therapy & Counseling”, score 0.968752
15. item_id 471, “Premium Sleep Solution 471”, category “Therapy & Counseling”, score 0.968662
16. item_id 500, “Premium Sleep Solution 500”, category “Therapy & Counseling”, score 0.968583

17. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.968542
18. item_id 479, “Premium Sleep Solution 479”, category “Therapy & Counseling”, score 0.968532
19. item_id 274, “Premium Sleep Solution 274”, category “Bedding & Accessories”, score 0.968360
20. item_id 513, “Premium Sleep Solution 513”, category “Therapy & Counseling”, score 0.968331

```
recommend_content_based(user_example, top_n=20)
```

	item_id	title	category	score
239	240	Premium Sleep Solution 240	Bedding & Accessories	0.971844
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.971703
479	480	Premium Sleep Solution 480	Therapy & Counseling	0.971370
234	235	Premium Sleep Solution 235	Bedding & Accessories	0.970899
270	271	Premium Sleep Solution 271	Bedding & Accessories	0.970612
489	490	Premium Sleep Solution 490	Therapy & Counseling	0.970477
255	256	Premium Sleep Solution 256	Bedding & Accessories	0.969570
253	254	Premium Sleep Solution 254	Bedding & Accessories	0.969441
277	278	Premium Sleep Solution 278	Bedding & Accessories	0.969334
492	493	Premium Sleep Solution 493	Therapy & Counseling	0.969273
260	261	Premium Sleep Solution 261	Bedding & Accessories	0.969247
464	465	Premium Sleep Solution 465	Therapy & Counseling	0.968916

Top content-based recommendations with similarity scores

253	254	Premium Sleep Solution 254	Bedding & Accessories	0.969441
277	278	Premium Sleep Solution 278	Bedding & Accessories	0.969334
492	493	Premium Sleep Solution 493	Therapy & Counseling	0.969273
260	261	Premium Sleep Solution 261	Bedding & Accessories	0.969247
464	465	Premium Sleep Solution 465	Therapy & Counseling	0.968916
466	467	Premium Sleep Solution 467	Therapy & Counseling	0.968806
488	489	Premium Sleep Solution 489	Therapy & Counseling	0.968752
470	471	Premium Sleep Solution 471	Therapy & Counseling	0.968662
499	500	Premium Sleep Solution 500	Therapy & Counseling	0.968583
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.968542
478	479	Premium Sleep Solution 479	Therapy & Counseling	0.968532
273	274	Premium Sleep Solution 274	Bedding & Accessories	0.968360
512	513	Premium Sleep Solution 513	Therapy & Counseling	0.968331

Continuation of high-ranked recommended items

The item-based k nearest neighbor (kNN) recommendation algorithm produces a list of the ten most highly recommended items, based on predicted rating (k=10) in order of their predicted rating for that particular user. Using the item-based kNN prediction algorithm (k=10), also provides the ten recommended items for the same example_user based on the predicted rating of each candidate item from the user's ratings of those items that are most similar to them (using item-item cosine similarity in their respective feature space). The ten highest ranking items (based on predicted ratings) produced by this algorithm are:

- item_id 14, pred_rating 5.0, “Relax & Restore v2”
- item_id 38, pred_rating 5.0, “SleepWell v4”
- item_id 286, pred_rating 5.0, “Breathing Technique Workshop”
- item_id 288, pred_rating 5.0, “CBT-I Program”
- item_id 312, pred_rating 5.0, “CBT-I Program”
- item_id 319, pred_rating 5.0, “Sleep Psychology Course”
- item_id 325, pred_rating 5.0, “Mindfulness Training”

- item_id 328, pred_rating 5.0, “CBT-I Program”
- item_id 329, pred_rating 5.0, “Meditation Course”
- item_id 331, pred_rating 5.0, “Yoga for Sleep

kNN Top10

	item_id	pred_rating	title
0	14	5.0	Relax & Restore v2
1	38	5.0	SleepWell v4
2	286	5.0	Breathing Technique Workshop
3	288	5.0	CBT-I Program
4	312	5.0	CBT-I Program
5	319	5.0	Sleep Psychology Course
6	325	5.0	Mindfulness Training
7	328	5.0	CBT-I Program
8	329	5.0	Meditation Course
9	331	5.0	Yoga for Sleep

kNN Top-10 Recommendations (Predicted Ratings Table)

In the analysis of Content-Based Similarity Ranking versus Item-Based KNN for the same example user (Step 2), it was observed that the two approaches produced significantly different Top-10 lists for the user under consideration. For example, while the Cosine Similarity based ranking Approach produced a set of recommended items primarily within the Bedding & Accessories and Therapy & Counseling categories, the score for each of those recommended items was very similar (between 0.969 and 0.972), and not far from one another. In contrast, the Top-10 list generated from KNN based recommendations contained programmatic and practice items, for example, CBT-I Program, Mindfulness Training, and Yoga for Sleep; all predicted ratings for these Top-

10 items were 5.0 in the output from. The results of this direct comparison indicate that the Ranking Mechanism through Direct Profile-to-Item Similarity Ranking focuses on identifying the closest matches to an individual user's Profile in the Feature Space, and the Ranking Mechanism through KNN Based Prediction focuses on identifying an estimated rating for an item as a function of neighborhood Aggregation.

In order to demonstrate how cosine similarity ranking works in a small dataset, we have created an example with three short text items labeled A, B, and C. We applied the TF-IDF method and calculated the cosine similarities between the TF-IDF vectors and a very simple User Preference Vector (UPV). The final ranking of the three text items can be as follows: Item A (“memory foam mattress soft comfy”) ranked highest with 0.816837; Item C (“gel cooling pillow comfortable”) ranked second at 0.692846; and Item B (“firm mattress provides excellent back support”) ranked lowest at 0.096698. This example demonstrates how the same reasoning applies to the content based approaches, in that higher scoring items based on cosine similarity to the user vector also had the greatest TF-IDF alignment to the user vector.

	item	text	score
0	A	memory foam mattress soft comfortable	0.816837
2	C	cooling pillow gel comfortable	0.692846
1	B	firm mattress good back support	0.096698

Toy Example: TF-IDF + Cosine Similarity Ranking (Items A/B/C)

Collaborative Filtering and Matrix Factorization

The following process involves using collaborative filtering to identify user preference based on explicit rating data and consequently establish user preference by looking for

shared behavioural patterns between users. The input core data tables utilised for this stage of the process are the same tables from Section 2 depicting the Items shape (520, 8), Ratings shape (51488, 5), and Users shape (5000, 14). The ratings were then reshaped to fit into a user–item rating matrix with dimensions (5000, 520) where each row indicates a user_id and each column indicates an item_id.

```
Items shape: (520, 8)
Ratings shape: (51488, 5)
Users shape: (5000, 14)
```

	rating_id	user_id	item_id	rating	date
0	1	2798	18	5	2025-02-11
1	2	848	383	1	2025-11-29
2	3	798	171	4	2025-01-22
3	4	3701	359	5	2025-09-24
4	5	4426	307	4	2025-12-29

Ratings Table Preview (Repeated Sample / Shapes)

```
user_col = "user_id"
item_col = "item_id"
rating_col = "rating"

user_item_matrix = ratings.pivot_table(index=user_col, columns=item_col, values=rating_col)
print("User-Item Matrix shape:", user_item_matrix.shape)
```

```
User-Item Matrix shape: (5000, 520)
```

User–Item Matrix Construction (Pivot Table + Matrix Shape 5000×520)

Collaborative Filtering (CF) is a method of predicting ratings for items based on similar users. In CF, user-based collaborative filtering calculates similarities between users and

uses the most similar users to make a rating prediction for the target user based on the average deviation of their ratings from their own. shows that the technique produces a User-User Similarity Matrix that is (5000, 5000) in shape.

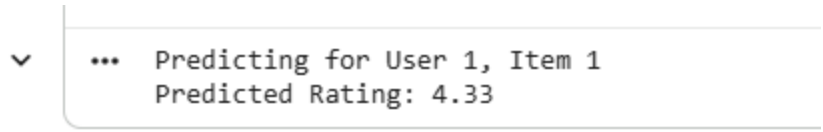
User Similarity Matrix shape: (5000, 5000)

user_id	1	2	3	4	5
user_id					
1	1.000000	0.0	0.0	0.0	-0.003928
2	0.000000	1.0	0.0	0.0	0.000000
3	0.000000	0.0	1.0	0.0	0.000000
4	0.000000	0.0	0.0	1.0	0.000000
5	-0.003928	0.0	0.0	0.0	1.000000

User–User Similarity Matrix Preview (5000×5000 Sample)

For rating prediction, the algorithm identifies all users that have rated the target item, then filters these users down to those whose similarity coefficient was positive in relation to the target user. Afterward, the model gets the top 20 most similar neighbours from those valid users and takes the mean of the target user's individual rating plus the similarity weighted average of neighbour's deviations from their mean rating.

If, in any case, the target user lacks similar neighbour users or has no similarity sum, the predicted rating will be equal to the target users's mean rating. All predicted ratings should be expressed as being within a defined range between 1.0 and 5.0. A sample use of is shown for predicting User 1's rating of Item 1: the prediction created for this item using the CF method was 4.33.



Collaborative Filtering Prediction Example (User 1, Item 1 → Predicted Rating)

Top-10 suggestions for user-based Collaborative Filtering (user_id = 1). To generate recommendations based on users' actions in prior months, predicts ratings for items that have not yet been rated by the user whose rating is being predicted, then numerically ranks the predictions from highest to lowest and returns up to 10 items. The top 10 suggestions for user_id = 1 are:

sold_product_id = 122; predicted_star_rating = 5.000000; product_name = “Valerian Root Extract”; product category = Supplements

sold_product_id = 349; predicted_star_rating = 5.000000; product_name = “Premium Sleep Solution 349”; product category = Medical Devices

sold_product_id = 439; predicted_star_rating = 5.000000; product_name = “Premium Sleep Solution 439”; product category = Sound & Environment

sold_product_id = 491; predicted_star_rating = 4.960391; product_name = “Premium Sleep Solution 491”; product category = Therapy & Counseling

sold_product_id = 273; predicted_star_rating = 4.902703; product_name = “Premium Sleep Solution 273”; product category = Bedding & Accessories

sold_product_id = 420; predicted_star_rating = 4.871805; product_name = “Premium Sleep Solution 420”; product category = Sound & Environment

sold_product_id = 131; predicted_star_rating = 4.867382; product_name = “Magnesium Glycinate”; product category = Supplements

sold_product_id = 315; predicted_star_rating = 4.848346; product_name = “Yoga for Sleep”; product category = Lifestyle Practices

sold_product_id = 471; predicted_star_rating = 4.831259; product_name = “Premium Sleep Solution 471”; product category = Therapy & Counseling

sold_product_id = 65; predicted_star_rating = 4.792470; product_name = “Fitbit NightSense Band”; product category = Wearable Devices

Top 10 Recommendations for User 1

	item_id	predicted_rating	name	category
0	122	5.000000	Valerian Root Extract	Supplements
1	349	5.000000	Premium Sleep Solution 349	Medical Devices
2	439	5.000000	Premium Sleep Solution 439	Sound & Environment
3	491	4.960391	Premium Sleep Solution 491	Therapy & Counseling
4	273	4.902703	Premium Sleep Solution 273	Bedding & Accessories
5	420	4.871805	Premium Sleep Solution 420	Sound & Environment
6	131	4.867382	Magnesium Glycinate	Supplements
7	315	4.848346	Yoga for Sleep	Lifestyle Practices
8	471	4.831259	Premium Sleep Solution 471	Therapy & Counseling
9	65	4.792470	Fitbit NightSense Band	Wearable Devices

User-Based CF Recommendations (Top-10 for User 1)

The following method used SVD matrix factorization using k=20 latent factors (100%). In addition, in addition to the neighborhood CF methods described uses the matrix factorization method to create a more comprehensive 'dense' prediction space through the use of 'filled' entries where missing values in the user-item interactions matrix have been completed using the item-specific mean fill value (that is: fillna with user-item-

mean(axis=0)). The created matrix is then mean centered by using the mean of the user's ratings (that is: user-ratings-mean). To perform truncated SVD, truncated SVD is performed using k=20 latent factors; the U, σ, and V matrices created during truncated SVD are shown, and are as follows: U= (5000,20); σ= (20,20); and Vt= (20,520). The full reconstructed predicted rating matrix has been reconstructed from the factorization, and has had the mean vector for the users added to each entry in the matrix, in order to return the user prediction rating matrix to the original rating scale.

SVD Top10 Recommendations (for user_id=1). Using the reconstructed predicted ratings (generated by all rated items by user) will generate a ranked list of SVD recommended items for user 1.

- item_id 40, predicted_rating 4.236056, “MindRest v5”, category “Mobile Apps”
- item_id 62, predicted_rating 4.131789, “Eight Sleep BiometricWatch”, category “Wearable Devices”
- item_id 495, predicted_rating 4.130776, “Premium Sleep Solution 495”, category “Therapy & Counseling”
- item_id 342, predicted_rating 4.113560, “Breathing Technique Workshop”, category “Lifestyle Practices”
- item_id 442, predicted_rating 4.108173, “Premium Sleep Solution 442”, category “Sound & Environment”
- item_id 247, predicted_rating 4.095059, “Premium Sleep Solution 247”, category “Bedding & Accessories”
- item_id 378, predicted_rating 4.091839, “Premium Sleep Solution 378”, category “Medical Devices”
- item_id 303, predicted_rating 4.087169, “Sleep Psychology Course”, category “Lifestyle Practices”
- item_id 391, predicted_rating 4.081298, “Premium Sleep Solution 391”, category “Medical Devices”
- item_id 132, predicted_rating 4.079108, “Valerian Root Extract”, category “Supplements”

SVD Recommendations for User 1

	item_id	predicted_rating	name	category
0	40	4.236056	MindRest v5	Mobile Apps
1	62	4.131789	Eight Sleep BiometricWatch	Wearable Devices
2	495	4.130776	Premium Sleep Solution 495	Therapy & Counseling
3	342	4.113560	Breathing Technique Workshop	Lifestyle Practices
4	442	4.108173	Premium Sleep Solution 442	Sound & Environment
5	247	4.095059	Premium Sleep Solution 247	Bedding & Accessories
6	378	4.091839	Premium Sleep Solution 378	Medical Devices
7	303	4.087169	Sleep Psychology Course	Lifestyle Practices
8	391	4.081298	Premium Sleep Solution 391	Medical Devices
9	132	4.079108	Valerian Root Extract	Supplements

SVD Recommendations (Top-10 for User 1 from Matrix Factorization)

Step 3 - Comparison Within - User Based C.F. vs. S.V.D. (Recommended ListsResults: Both - User Based C.F. versus S.V.D. approaches produce "Predicted List" with Similar Items but also have # Different Score Ranges for the same user id = 1. The "Top 10" User-Based C.F. recommended list consists of several items with Predicted Ratings of 5.000000 and Higher Top5 Ratings than the range displayed within the "Top 10" of S.V.D. down to 4.792470 which was produced through the range of Predictions from the User-Based Reactor Model. On the other hand, the S.V.D. Top-10 Recommended List shows a More Normal Distribution of Predictions over all the ratings produced from 4.236056 down through all the ratings down to 4.079108, based off of the Latent Factor Model and the Average Centered Model prior to Mean Centering and Low Rank Approximation. Both Lists show some Limited Overlap in the Types of Recommendations or Items Recommended, such as Wearable Devices, Sound & Environment, Bedding & Accessories & Supplements, and both Recommended Items within the Top-10 List for the Valerian Root Extracts, having the same Item number 122 in User-Based C.F. and 132 in S.V.D.

Hybrid Recommendation Strategy

In this step, a weighted hybrid recommender combining a content-based score with collaborative filtering score is implemented to provide more robust recommendations on sparsity and cold-start situations. The hybrid method computes one combined score for each candidate item using the weighted combination defined in Section 2, which then allows for ranking the items according to their scores.
Hybrid formulation and candidate set. For a given user_id, the method first collects the set of items already rated by that user and excludes them from recommendation. The remaining items form the candidate set used for scoring and ranking. For each candidate item, two components are computed: (1) a CF-based score derived from the SVD predicted rating matrix (preds_df), and (2) a CB-based score derived from TF-IDF cosine similarity between a user profile vector and each candidate item vector. These two scores are then combined using the weighted rule: combined score = alpha × CB + (1 − alpha) × CF.
To sum up, the CF score from the SVD predictions is sourced from preds_df that corresponds to the desired user across the candidate items. While the SVD predicted ratings were created on the same scale as standard rating systems, the ratings were transformed via min-max normalisation to produce a normalized CF score for each candidate item that can be pooled with cosine similarities (the other half of the hybrid recommender) across the candidate items as they will be relatively parallel with each other.

The content-based score was derived from a user profile created with the TF-IDF item vectors, while creating the TF-IDF item vectors to compute the content-based score, a user profile was created, and selected only those items that were rated 3 or higher. Users without any items rated at least 3 are eligible to have their entire rated item collection included in the creation of their user profile TF-IDF vector. The cosine similarity between the candidate item's TF-IDF vector and the user's profile TF-IDF vector is used to compute the content-based score for that candidate item.

Hybrid output. After combining scores, items are sorted by the combined score in descending order and the top_n results are returned with item_id, name, category, and score. In this step, defines the hybrid function and then tests alpha values required by the specification.

Alpha sensitivity test (validation-style demonstration for user_id = 1). tests three alpha values (0.3, 0.5, 0.7) and shows the top results for user_id = 1. The top-5 recommendations for each alpha are:

Alpha = 0.3 (top 5):

Alpha = 0.3 (top 5):

1. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.767483
2. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 0.761928
3. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.759145
4. item_id 378, “Premium Sleep Solution 378”, category “Medical Devices”, score 0.755603
5. item_id 391, “Premium Sleep Solution 391”, category “Medical Devices”, score 0.746759

Alpha = 0.5 (top 5):

1. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.711245
2. item_id 378, “Premium Sleep Solution 378”, category “Medical Devices”, score 0.707915
3. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 0.705403
4. item_id 391, “Premium Sleep Solution 391”, category “Medical Devices”, score 0.701597
5. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.696600

Alpha = 0.7 (top 5):

1. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.663344
2. item_id 378, “Premium Sleep Solution 378”, category “Medical Devices”, score 0.660226
3. item_id 391, “Premium Sleep Solution 391”, category “Medical Devices”, score 0.656435
4. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 0.648878
5. item_id 358, “Premium Sleep Solution 358”, category “Medical Devices”, score 0.648455

```
Testing validation for User 1...

--- Alpha = 0.3 ---
```

	item_id	name	category	score
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.767483
441	442	Premium Sleep Solution 442	Sound & Environment	0.761928
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.759145
377	378	Premium Sleep Solution 378	Medical Devices	0.755603
390	391	Premium Sleep Solution 391	Medical Devices	0.746759

Hybrid Recommender Output (α = 0.3, Top-5)

--- Alpha = 0.5 ---

	item_id	name	category	score
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.711245
377	378	Premium Sleep Solution 378	Medical Devices	0.707915
441	442	Premium Sleep Solution 442	Sound & Environment	0.705403
390	391	Premium Sleep Solution 391	Medical Devices	0.701597
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.696600

Hybrid Recommender Output (α = 0.5, Top-5)

--- Alpha = 0.7 ---

	item_id	name	category	score
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.663344
377	378	Premium Sleep Solution 378	Medical Devices	0.660226
390	391	Premium Sleep Solution 391	Medical Devices	0.656435
441	442	Premium Sleep Solution 442	Sound & Environment	0.648878
357	358	Premium Sleep Solution 358	Medical Devices	0.648455

Hybrid Recommender Output (α= 0.7, Top-5)

Changing Alpha Values Affects Item Ranking

In the table below, you can see how varying alpha values may affect the item ranking. An item that has an alpha value of 0.3 might appear ranked as number 1 when compared to items that have alpha values of 0.5. The reverse is also true. An item with an alpha value of 0.5 will be ranked higher than an item with an alpha value of 0.3 if the CF term is normalized. For instance, item_id 495 has a ranking of 1 at an alpha value of 0.3 and a ranking of 5 when the alpha value is 0.5. Conversely, item_id 247 maintains a high ranking through all alpha selections, but becomes the number 1 recommended

item when alpha is set to either 0.5 or 0.7. In the case of an alpha = 0.7, item_id 358 is returned as part of the top 5 items because it has a high correlation with the content-based similarity.

Cold Start Users: Hybrid Algorithm vs. Popularity

To demonstrate how hybrid behavior occurs when the user has limited past ratings, we will show examples using three users from the days with the least amount of rating history. User_id 3844 ("3 Ratings") was the first cold-start user. The second cold-start user was user_id 3440 ("5 Ratings"). The third cold-start user was user_id 4086 ("10 Ratings"). will show the past ratings of these three users and their top 5 recommendations based on hybrid usage of alpha = 0.5 as well as the top 5 recommendations based on popularity for each of these three cold-start users.

User 3844 (3-Ratings) history:

1. item_id 1, “Dream Track v1”, category “Mobile Apps”, rating 3
2. item_id 409, “Premium Sleep Solution 409”, category “Sound & Environment”, rating 4
3. item_id 428, “Premium Sleep Solution 428”, category “Sound & Environment”, rating 4

User 3844 hybrid recommendations (alpha = 0.5, top 5):

1. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 0.731656
2. item_id 412, “Premium Sleep Solution 412”, category “Sound & Environment”, score 0.699391
3. item_id 425, “Premium Sleep Solution 425”, category “Sound & Environment”, score 0.692227
4. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.679222
5. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.655971

User 3844 popularity baseline (top 5):

1. item_id 40, “MindRest v5”, category “Mobile Apps”, score 4.192285
2. item_id 62, “Eight Sleep BiometricWatch”, category “Wearable Devices”, score 4.106515
3. item_id 342, “Breathing Technique Workshop”, category “Lifestyle Practices”, score 4.091209
4. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 4.078022
5. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 4.107574

User 3440 (5-Ratings) history:

1. item_id 271, “Premium Sleep Solution 271”, category “Bedding & Accessories”, rating 4
2. item_id 139, “Lavender Oil”, category “Supplements”, rating 5
3. item_id 483, “Premium Sleep Solution 483”, category “Therapy & Counseling”, rating 4
4. item_id 156, “GABA Supplement”, category “Supplements”, rating 5
5. item_id 435, “Premium Sleep Solution 435”, category “Sound & Environment”, rating 5

User 3440 hybrid recommendations (alpha = 0.5, top 5):

1. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.720547
2. item_id 490, “Premium Sleep Solution 490”, category “Therapy & Counseling”, score 0.699297
3. item_id 247, “Premium Sleep Solution 247”, category “Bedding & Accessories”, score 0.697769
4. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 0.696430
5. item_id 412, “Premium Sleep Solution 412”, category “Sound & Environment”, score 0.678159

User 3440 popularity baseline (top 5):

1. item_id 40, “MindRest v5”, category “Mobile Apps”, score 4.192285
2. item_id 62, “Eight Sleep BiometricWatch”, category “Wearable Devices”, score 4.106515
3. item_id 342, “Breathing Technique Workshop”, category “Lifestyle Practices”, score 4.091209
4. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 4.078022
5. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 4.107574

User 4086 (10-Ratings) history:

1. item_id 177, “Humidifier”, category “Sleep Aids”, rating 1
2. item_id 279, “Premium Sleep Solution 279”, category “Bedding & Accessories”, rating 5
3. item_id 476, “Premium Sleep Solution 476”, category “Therapy & Counseling”, rating 5
4. item_id 494, “Premium Sleep Solution 494”, category “Therapy & Counseling”, rating 2
5. item_id 95, “Samsung NightSense Band”, category “Wearable Devices”, rating 5
6. item_id 389, “Premium Sleep Solution 389”, category “Medical Devices”, rating 5
7. item_id 504, “Premium Sleep Solution 504”, category “Therapy & Counseling”, rating 5
8. item_id 333, “Mindfulness Training”, category “Lifestyle Practices”, rating 5
9. item_id 88, “Samsung HeartRate Monitor Elite”, category “Wearable Devices”, rating 5
10. item_id 314, “Sleep Hypnotherapy”, category “Lifestyle Practices”, rating 1

User 4086 hybrid recommendations (alpha = 0.5, top 5):

1. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 0.722422
2. item_id 378, “Premium Sleep Solution 378”, category “Medical Devices”, score 0.717173

3. item_id 480, “Premium Sleep Solution 480”, category “Therapy & Counseling”, score 0.698349
4. item_id 490, “Premium Sleep Solution 490”, category “Therapy & Counseling”, score 0.695829
5. item_id 391, “Premium Sleep Solution 391”, category “Medical Devices”, score 0.685746

User 4086 popularity baseline (top 5):

1. item_id 40, “MindRest v5”, category “Mobile Apps”, score 4.192285
2. item_id 62, “Eight Sleep BiometricWatch”, category “Wearable Devices”, score 4.106515
3. item_id 342, “Breathing Technique Workshop”, category “Lifestyle Practices”, score 4.091209
4. item_id 442, “Premium Sleep Solution 442”, category “Sound & Environment”, score 4.078022
5. item_id 495, “Premium Sleep Solution 495”, category “Therapy & Counseling”, score 4.107574

```
Selected Cold Start Users: [('3-Ratings', 3844), ('5-Ratings', 3440), ('10-Ratings', 4086)]

=== User 3844 (3-Ratings) ===
User History:
```

	item_id	name	category	rating
0	1	Dream Track v1	Mobile Apps	3
1	409	Premium Sleep Solution 409	Sound & Environment	4
2	428	Premium Sleep Solution 428	Sound & Environment	4

```
Hybrid Recommendations (Alpha=0.5):
```

	item_id	name	category	score
441	442	Premium Sleep Solution 442	Sound & Environment	0.731656
411	412	Premium Sleep Solution 412	Sound & Environment	0.699391
424	425	Premium Sleep Solution 425	Sound & Environment	0.692227
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.679222
246	247	Premium Sleep Solution 247	Bedding & Accessories	0.655971

Cold-Start Case: User History + Hybrid Top-5 (α=0.5)

Popularity Recommendations (Baseline):

	item_id	name	category	score
39	40	MindRest v5	Mobile Apps	4.192285
61	62	Eight Sleep BiometricWatch	Wearable Devices	4.106515
341	342	Breathing Technique Workshop	Lifestyle Practices	4.091209
441	442	Premium Sleep Solution 442	Sound & Environment	4.078022
494	495	Premium Sleep Solution 495	Therapy & Counseling	4.107574

=== User 3440 (5-Ratings) ===
User History:

	item_id	name	category	rating
0	271	Premium Sleep Solution 271	Bedding & Accessories	4
1	139	Lavender Oil	Supplements	5
2	483	Premium Sleep Solution 483	Therapy & Counseling	4
3	156	GABA Supplement	Supplements	5
4	435	Premium Sleep Solution 435	Sound & Environment	5

Popularity Baseline + Cold-Start User History

Hybrid Recommendations (Alpha=0.5):

	item_id	name	category	score
494	495	Premium Sleep Solution 495	Therapy & Counseling	0.720547

Cold-Start Hybrid Output (Top ranked item snippet)

To compare recommendation quality across approaches, evaluates multiple models on a sample of 50 users and reports two ranking metrics at cutoff 10: HR@10 (Hit Rate at 10) and NDCG@10 (Normalized Discounted Cumulative Gain at 10). The evaluation includes five models: Random, Popularity, Pure Content-Based (Pure CB), Pure Collaborative Filtering (Pure CF), and the Hybrid model.

The evaluation is executed with the message “Evaluating on 50 users.” and produces the following results:

Random: HR@10 = 0.00, NDCG@10 = 0.00

Popularity: HR@10 = 0.00, NDCG@10 = 0.00

Pure CB: HR@10 = 0.02, NDCG@10 = 0.006021

Pure CF: HR@10 = 0.04, NDCG@10 = 0.025781

Hybrid: HR@10 = 0.04, NDCG@10 = 0.030000

```
Evaluating on 50 users.
Evaluated Random: HR=0.0000, NDCG=0.0000
Evaluated Popularity: HR=0.0000, NDCG=0.0000
Evaluated Pure CB: HR=0.0200, NDCG=0.0060
Evaluated Pure CF: HR=0.0400, NDCG=0.0258
Evaluated Hybrid: HR=0.0400, NDCG=0.0300
```

	Model	HR@10	NDCG@10
0	Random	0.00	0.000000
1	Popularity	0.00	0.000000
2	Pure CB	0.02	0.006021
3	Pure CF	0.04	0.025781
4	Hybrid	0.04	0.030000

```
Saved comparison to ../results\model_comparison.csv
```

Evaluation Summary on 50 Users (HR@10 & NDCG@10 Results)

Discussion and Conclusion

The data preparation and exploratory analysis phases provide a comprehensive set of recommendations for developing a Sleep Quality Improvement Recommendation Engine to develop its content-based model; collaborative filtering model; hybrid recommendation model; and final evaluation of the overall system's performance.

This includes the method of converting item content (descriptions, features, and review text) into a vector space format (using TF-IDF). Additionally, user-item collaborative

signal extraction takes place using user-item rating matrix and different neighborhood methods (i.e. SVD) and creating a weighted hybrid combined recommendation model provides an improved ranking quality for both new items and existing ones.

The exploratory analyses of the data, the dataset characteristics reveal the necessity for using content and collaborative approaches to recommendation. Dataset statistics printed below indicate an overall use of 5000 users, 520 items, and 51488 total ratings, resulting in sparsity value equal to 0.9802, or 98.02%. There is evidence that there is a non-uniform distribution of user activity and item popularity based upon the analysis. In addition, the long tail plot depicts a significa-ntly skewed item popularity (i.e., how many times an item is interacted with), which represents a major obstacle for most recommender systems because there are very few items that will be interacted with as frequently as the most popular items.

The Hybrid approach exhibited the best overall ranking performance through the testing of 50 users using the various recommended methods. The final evaluation results are as follows for each method tested; Random (HR@10 = 0.00, NDCG@10 = 0.000000), Popularity (HR@10 = 0.00, NDCG@10 = 0.000000), Pure CB (HR@10 = 0.02, NDCG@10 = 0.006021), Pure CF (HR@10 = 0.04, NDCG@10 = 0.025781), Hybrid (HR@10 = 0.04, NDCG@10 = 0.030000). The results of the evaluation demonstrate that collaborative filtering has provided a marked improvement in the ranking produced from content alone in this scenario, and that the weighted hybrid method has improved the NDCG@10 result when compared to Pure CF, while holding the same HR@10 result. The cold-start demonstration of users with 3, 5, and 10 ratings also indicated that the hybrid method is capable of tailoring the recommendations for those users with a limited history (for example, the recommendation system recommends multiple Sound & Environment items even if the users only had Sound & Environment items in their history) whereas the popularity baseline remains static across users reflecting the non-personalized nature of the recommendations.