AIE425 Intelligent Recommender Systems, Fall Semester 25/26

FINAL COURSE PROJECT

AUTHORS

Group [15]
Salama Sayed Salama - 222102243
Yousef Mohamed ibrahim- 223106299
Omar Saeed Mohamed Kamel - 222101064
Mostafa Mahmoud Elsayed - 222101612

SUPERVISED BY

prof. Samy Ghoniemy

Submission Date: Monday, January 5, 2026

Contents

Executive Summary

Recommender systems are crucial to modern information platforms, as they help users with the difficult task of finding relevant items in large and sparse datasets. The project combines statistical analysis, dimensionality reduction, content-based filtering, collaborative filtering, and hybrid recommendation strategies to deeply investigate and apply several recommendation techniques. This work mainly consists of two sections: interest-based group recommendations using domain-aware models and rating predictions using techniques of dimensionality reduction.

In the first section, the concepts of sparse interaction and item popularity are tested using statistical analysis on a user-item rating data set. Peer prediction methods are tested using different neighborhood sizes, and covariance matrices are computed for finding correlations between different items. It is also possible to map users into lower-dimensional latent spaces using Principal Component Analysis (PCA), which can be done for dimensionality reduction using mean-fill and Maximum Likelihood Estimation (MLE) methods. Top-k peer neighborhoods can be utilized for prediction related to missing ratings in matrices, and different analyses are also performed in this section based on the size of the neighborhoods and different PCA methods.

The second phase of the project deals with group recommendation based on interest. The technique employed for preference modeling calculates the cosine similarity. Content filtering is carried out with the help of the TF-IDF matrix for the tags used for the group. In the case of collaborative filtering, SVD with different latent sizes and user similarities have been employed to explore the concealed patterns. To exploit the benefits of both content and collaborative filtering techniques, a weighted hybrid model for the recommendation system is proposed that takes into account the collective scores for content and collaborative filtering with the help of a parameter. The proposed model performs better compared to individual approaches, as it demonstrates resistance to changes in user activity.

On the basis of the experimental results, it has been observed that the hybrid recommendation algorithm performs better than other algorithms in terms of hit rate, precision, and recall, while the use of dimensionality reduction further improves the consistency of prediction in cases with sparse ratings. On balance, the significance of combining statistical knowledge, models based on latent factors, and domain knowledge in designing efficient recommender systems that can effectively address sparsity problems cannot be overemphasized.

1 Dimensionality Reduction and Matrix Factorization

1.1 Statistical Analysis

This subsection presents a comprehensive statistical analysis of the MovieLens 20M dataset, which serves as a preliminary step for dimensionality reduction and matrix factorization techniques. The analysis aims to understand the distribution of ratings across users and items, identify popularity patterns, and select representative target users and items for subsequent experiments.

1.1.1 Dataset Description and Preprocessing

The MovieLens 20M dataset consists of user item ratings collected on a five point scale. After loading the dataset, ratings were clipped to the valid range [1,5] to ensure consistency. Basic dataset statistics, including the number of users, items, and ratings, were computed to confirm the scale and suitability of the dataset for collaborative filtering analysis.

1.1.2 User and Item Activity Analysis

To quantify user engagement, the number of ratings per user (n_u) was computed and saved. Similarly, item popularity was measured by calculating the number of ratings per item (n_i). In addition, the average rating per user ($\bar{r}_u$) and per item ($\bar{r}_i$) were computed to characterize rating behavior.

Table 1: Number of Ratings per User

User ID	n_u
1	175
2	61
3	187
4	28
5	66

Table 2: Number of Ratings per Item

Movie ID	n_i
1	49,695
2	22,243
3	12,735
4	2,756
5	12,161

Table 3: Average Rating per User

User ID	$\bar{r}_u$
1	3.74
2	4.00
3	4.12
4	3.57
5	4.27

Table 4: Average Rating per Item

Movie ID	$\bar{r}_i$
1	3.92
2	3.22
3	3.16
4	2.87
5	3.07

1.1.3 Item Popularity Distribution

Items were sorted in ascending order based on the number of ratings they received. Figure 1 illustrates the long-tail distribution commonly observed in recommender system datasets, where a small number of items receive a large proportion of ratings.

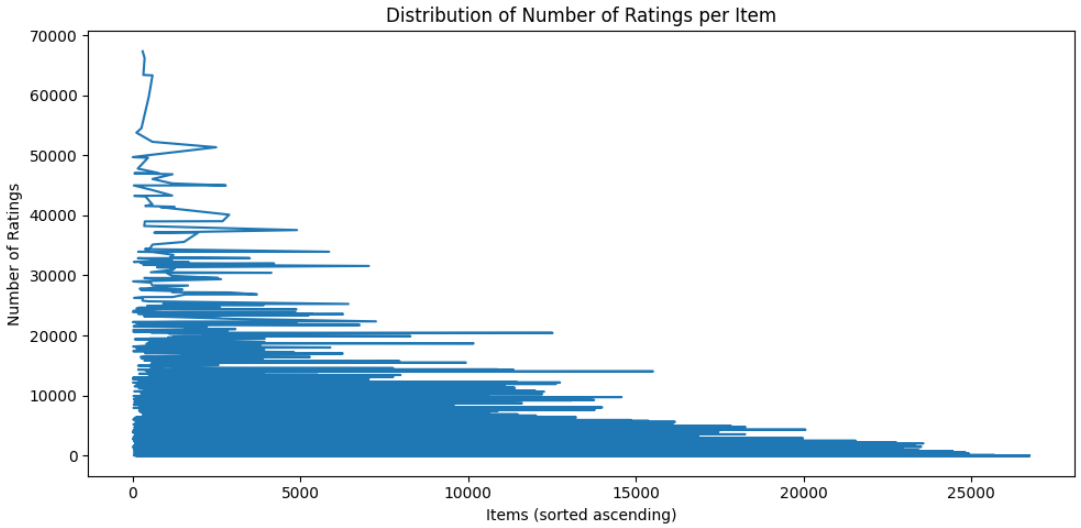

Figure 1: Distribution of Number of Ratings per Item

To further analyze popularity, items were divided into three groups based on popularity percentiles: low popularity, medium popularity, and high popularity. These groupings help differentiate between cold-start items, moderately rated items, and highly popular items.

1.1.4 Item Rating-Based Grouping

Items were also grouped according to the percentile rank of their average ratings. Ten rating-based groups (G1 to G10) were formed, ranging from the lowest rated to the highest rated items. The total number of ratings in each group was computed and sorted in ascending order.

Table 5: Total Number of Ratings per Rating Group

Group	Total Ratings
G1	450
G2	86,625
G3	243,842
G4	712,163
G6	830,820

The distribution of ratings across these groups is visualized in Figure 2.

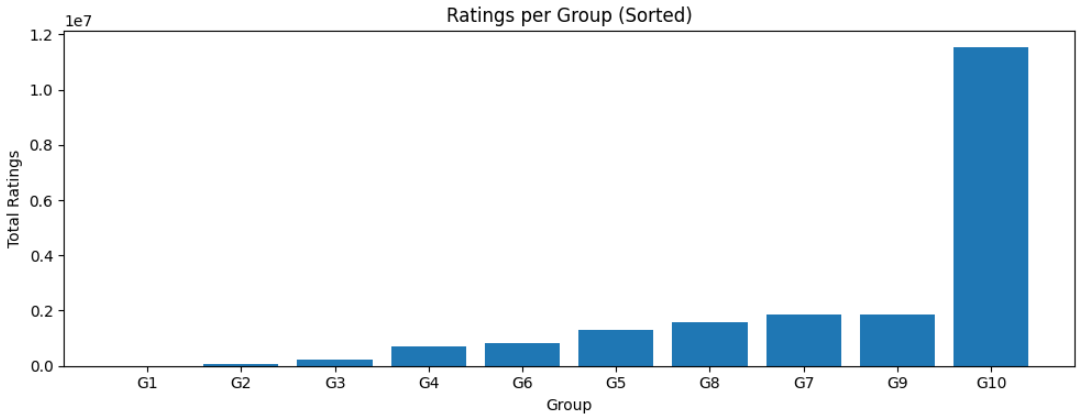

Figure 2: Total Ratings per Rating Group

1.1.5 Target User Selection

Three target users were selected based on their activity level:

- U1: users with less than 2% of the total rating activity
- U2: users with activity between 2% and 5%
- U3: users with activity between 5% and 10%

This selection ensures the inclusion of users with varying engagement levels, allowing the evaluation of recommendation performance across different user profiles.

Table 6: Selected Target Users

User Label	User ID
U1	13,238
U2	107,542
U3	110,288

1.1.6 Target Item Selection

Two target items were selected as the lowest-rated items in the dataset based on their average rating values. This selection strategy focuses on items that received consistently low user ratings, making them challenging cases for recommendation algorithms.

By selecting the lowest rated items, the evaluation emphasizes the ability of dimensionality reduction and matrix factorization techniques to predict missing ratings for unpopular or poorly perceived items, which is a critical scenario in recommender system analysis.

Table 7: Selected Target Items

Item Label	Movie ID
I1	100,157
I2	2,588

1.1.7 Co-Rating Analysis and Threshold Determination

For each target user, the number of corating users was computed. Similarly, for each target item, the number of corated items was calculated. These co-occurrence statistics are essential for similarity-based and latent factor models.

A threshold was then defined for each target user as 30% of the total number of items rated by that user. This threshold represents the minimum overlap required to consider another user sufficiently similar.

Table 8: Co-Rating Thresholds (30%) for Target Users

User	Threshold
U1	20
U2	21
U3	22

1.1.8 Summary

This statistical analysis provides a detailed understanding of user behavior, item popularity, and rating distributions in the dataset. The insights gained from this section inform the design and evaluation of subsequent dimensionality reduction and matrix factorization methods, ensuring that experiments are conducted on representative users and items.

1.2 PCA Method with Mean-Filling

This subsection presents the application of the Principal Component Analysis (PCA) method with a mean-filling strategy for rating prediction. The goal is to estimate missing ratings for selected target items by exploiting item–item covariance structure and projecting users into a reduced latent space.

1.2.1 Target Item Statistics

For each target item (I_1 and I_2), the average rating was computed using only observed ratings. These averages are later used in the mean-filling process to handle missing values:

$$\bar{r}_i = \frac{1}{|\mathcal{U}_i|} \sum_{u \in \mathcal{U}_i} r_{u,i}. \tag{1}$$

Table 9: Average Ratings of Target Items

Movie ID	Average Rating
2588	2.67
100157	3.28

1.2.2 Mean-Filling of Target Items

Missing ratings were replaced by the corresponding item mean:

$$\tilde{r}_{u,i} = \begin{cases} r_{u,i}, & \text{if observed,} \\ \bar{r}_i, & \text{otherwise.} \end{cases} \tag{2}$$

Table 10: Mean-Filled Ratings for Target Items (Sample)

User ID	Movie ID	Rating
28456	100157	4.00
30731	100157	3.50
51158	100157	3.50
51991	100157	3.50
63046	100157	3.50

1.2.3 Item Mean-Centering

To remove item-specific bias, ratings were mean-centered:

$$r'_{u,i} = r_{u,i} - \bar{r}_i. \tag{3}$$

Table 11: Average Rating per Item (Sample)

Movie ID	Average Rating
1	3.92
2	3.21
3	3.15
4	2.86
5	3.06

Table 12: Mean-Centered Item Ratings (Sample)

User ID	Movie ID	Rating Difference
1	2	0.29
1	29	-0.45
1	32	-0.40
1	47	-0.55
1	50	-0.83

1.2.4 Covariance Computation

Covariance between item i and target item j was computed using common users:

$$\text{Cov}(i,j) = \frac{1}{|\mathcal{U}_{ij}|} \sum_{u \in \mathcal{U}_{ij}} r'_{u,i} r'_{u,j}. \tag{4}$$

Table 13: Item Covariance with Target Items (Sample)

Movie ID	Covariance	Target Item
1	-0.15	100157
2	-0.02	100157
5	0.02	100157
6	0.09	100157
7	0.03	100157

Table 14: Covariance Matrix for Target Items (Sample)

Movie ID	Cov$(\cdot, 2588)$	Cov$(\cdot, 100157)$
1	0.66	-0.15
2	0.97	-0.02
3	-0.22	0.00
4	0.81	0.00
5	0.15	0.02

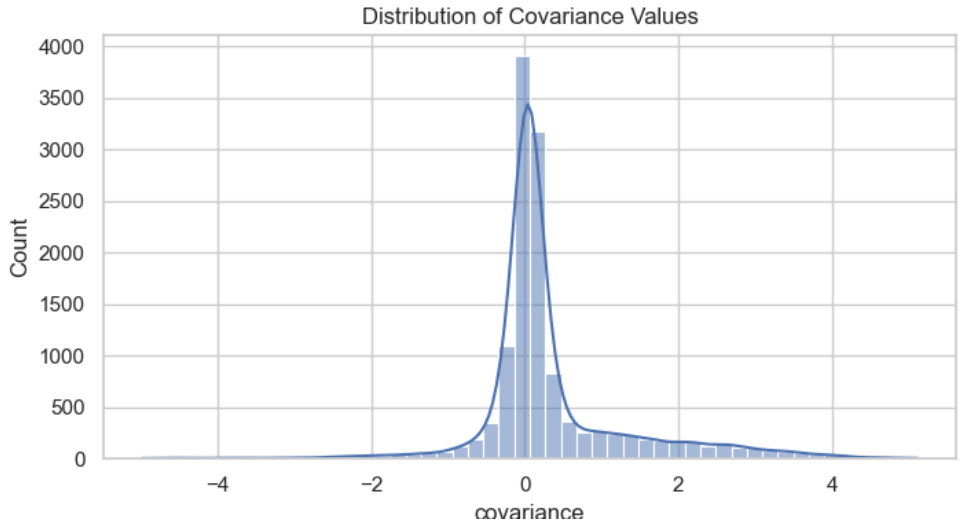

Figure 3: This figure shows the distribution of covariance values between the target items and all other items. Most covariances are concentrated near zero, indicating weak linear relationships, which is expected due to rating sparsity. Only a small number of items exhibit strong positive or negative covariance and are therefore informative for peer selection.

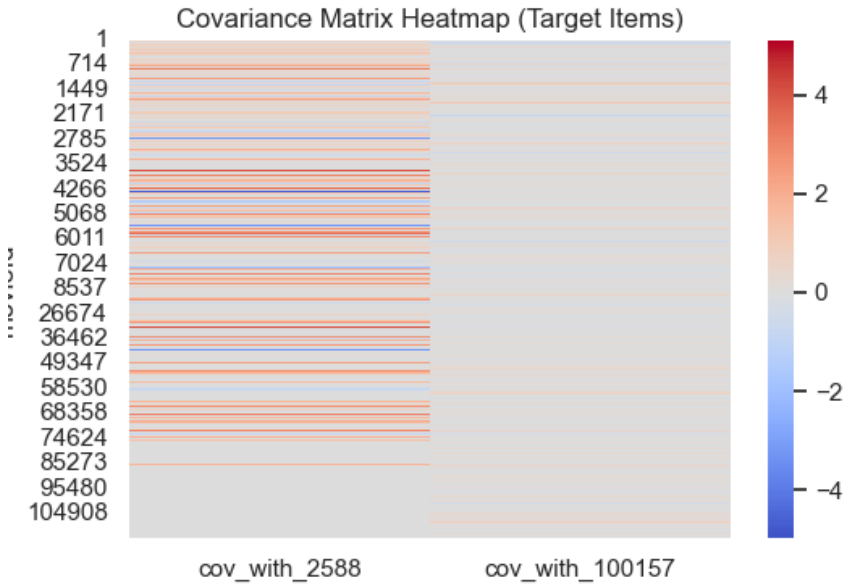

Figure 4: The heatmap visualizes the covariance between each item and the two target items. Most values are close to zero, while a limited subset shows strong correlations. These high-magnitude covariances identify the most relevant peer items used in PCA.

1.2.5 Peer Selection

Top peers were selected based on covariance magnitude.

Table 15: Top-5 Peer Items for Target Item 2588

Movie ID	Covariance	Target Item	Popularity (n_i)
6371	5.11	2588	325
3574	5.02	2588	139
61348	5.00	2588	397
31698	5.00	2588	467
5739	4.98	2588	174

1.2.6 Reduced Dimensional Space via PCA

PCA was performed via eigen-decomposition:

$$\mathbf{C}\mathbf{w}_k = \lambda_k \mathbf{w}_k. \tag{5}$$

Table 16: Reduced User Space Using Top-5 Peers (PCA)

User ID	PC1	PC2	Target Item
218	0.14	-0.07	2588
383	-0.70	0.47	2588
388	0.43	-0.22	2588
422	0.74	0.35	2588
440	-1.78	1.09	2588

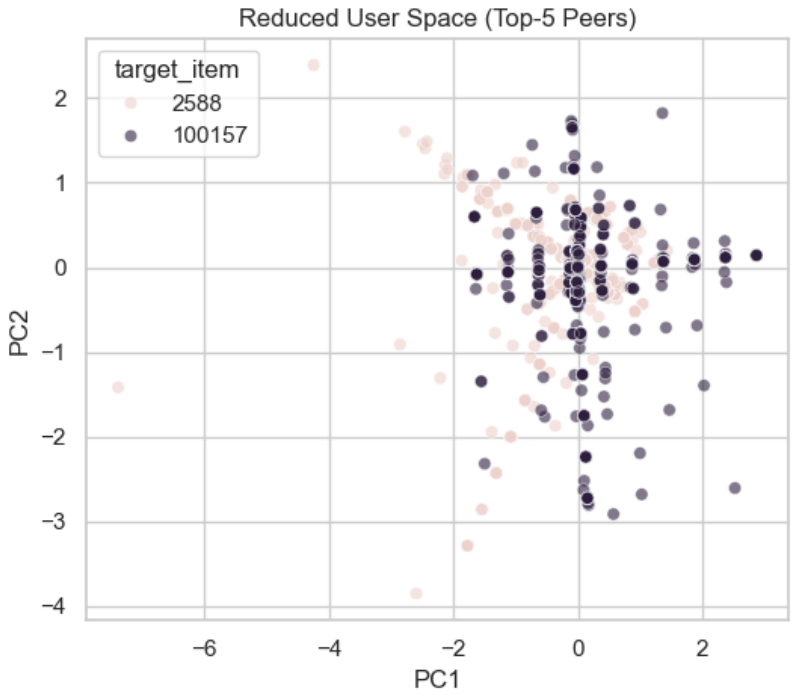

Figure 5: This plot shows users projected into a two-dimensional PCA space using the Top-5 peer items. The compact clustering indicates stable latent representations when only highly correlated peers are used.

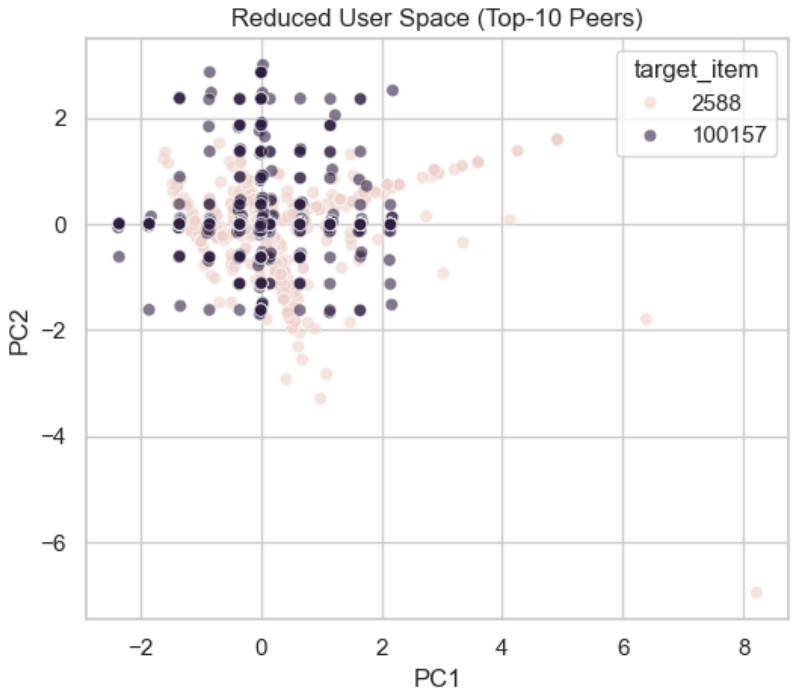

Figure 6: Using Top-10 peers results in a more dispersed user distribution. This reflects increased variance captured by the model and suggests higher sensitivity to user rating differences.

1.2.7 Rating Prediction

Predictions were computed as:

$$\hat{r}_{u,i} = \bar{r}_i + \sum_{k=1}^{K} z_{u,k}. \tag{6}$$

Table 17: Predicted Ratings Using Top-5 Peers

User ID	Target Item	Predicted Rating
218	2588	2.74
383	2588	2.44
388	2588	2.88
422	2588	3.76
440	2588	1.98

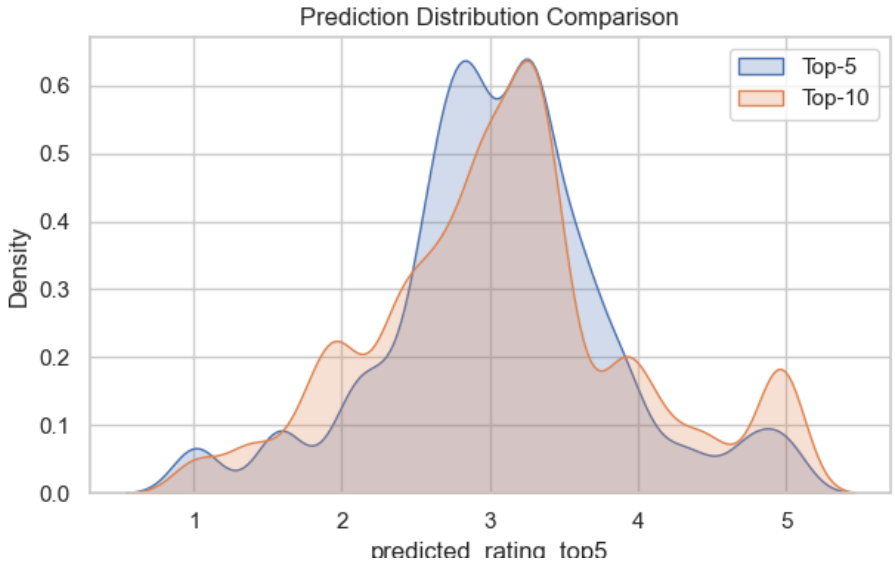

Figure 7: The predicted rating distributions for Top-5 and Top-10 peers are centered around similar values. However, the Top-10 configuration exhibits slightly higher spread, indicating greater prediction variability.

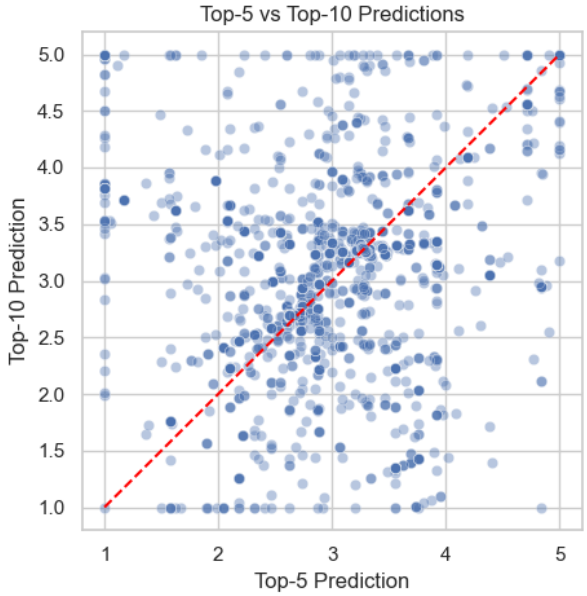

Figure 8: Each point represents a users predicted rating using Top-5 versus Top-10 peers. Deviations from the diagonal line indicate users whose predictions are strongly affected by the peer set size.

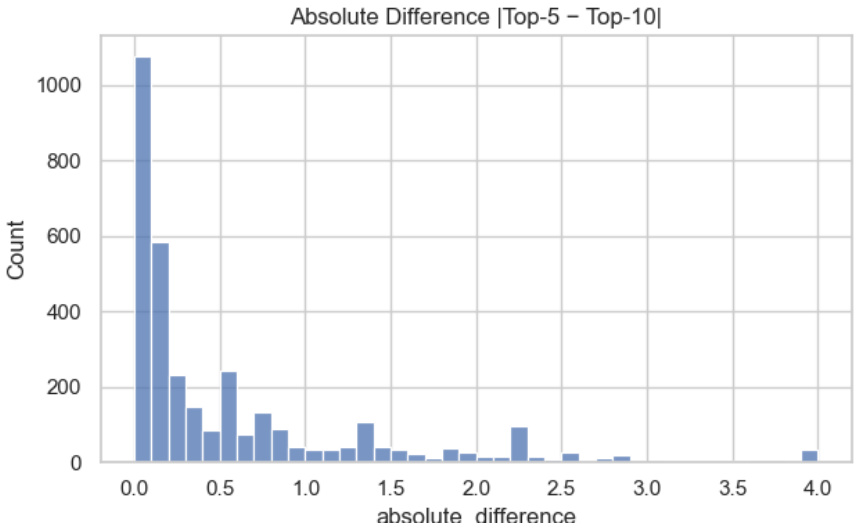

Figure 9: Most absolute differences between Top-5 and Top-10 predictions are small, showing general agreement between the two methods. Larger differences occur for a limited number of users

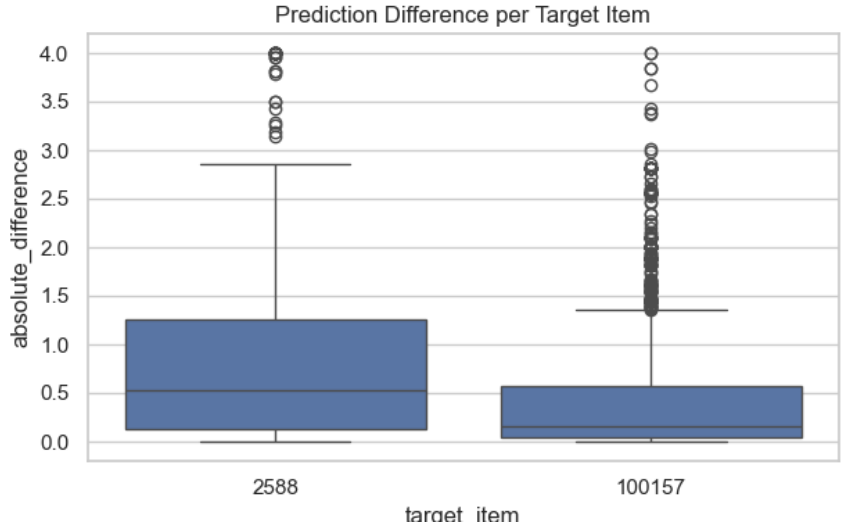

Figure 10: This boxplot shows that prediction differences vary across target items. One item exhibits higher variability, indicating that its predictions are more sensitive to peer selection.

1.2.8 Discussion

The PCA method with mean-filling demonstrates that increasing the number of peer items leads to greater variability in predictions. Top-5 peers provide more stable estimates, while Top-10 peers

capture broader structure, highlighting a trade-off between locality and expressiveness.

1.3 PCA Method with Maximum Likelihood Estimation

This subsection presents the Principal Component Analysis (PCA) approach combined with a Maximum Likelihood Estimation (MLE) strategy for estimating itemâitem covariance and predicting missing ratings. Unlike mean-filling, the MLE method computes covariance using only users who have rated both items, leading to a more statistically principled estimation.

1.3.1 MLE-Based Covariance Estimation

Let $r_{u,i}$ denote the rating given by user u to item i, and $\bar{r}_i$ the mean rating of item i. The MLE covariance between a target item j and another item i is computed as:

$$\text{Cov}_{\text{MLE}}(i,j) = \frac{1}{|\mathcal{U}_{ij}|} \sum_{u \in \mathcal{U}_{ij}} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)$$

where $\mathcal{U}_{ij}$ is the set of users who rated both items i and j. If no such users exist, the covariance is set to zero.

Table 18: Item Covariance with Target Item (MLE Sample)

Movie ID	Covariance
1	-0.1548
2	-0.0206
5	0.0179
6	0.0930
7	0.0297

The computed covariances are assembled into a target-item covariance matrix.

Table 19: MLE Covariance Matrix for Target Items

Movie ID	Cov with 2588	Cov with 100157
1	0.6614	-0.1548
2	0.9698	-0.0206
3	-0.2170	0.0000
4	0.8102	0.0000
5	0.1451	0.0179

This figure shows that most covariance values are concentrated around zero, with a small number of strongly correlated item pairs.

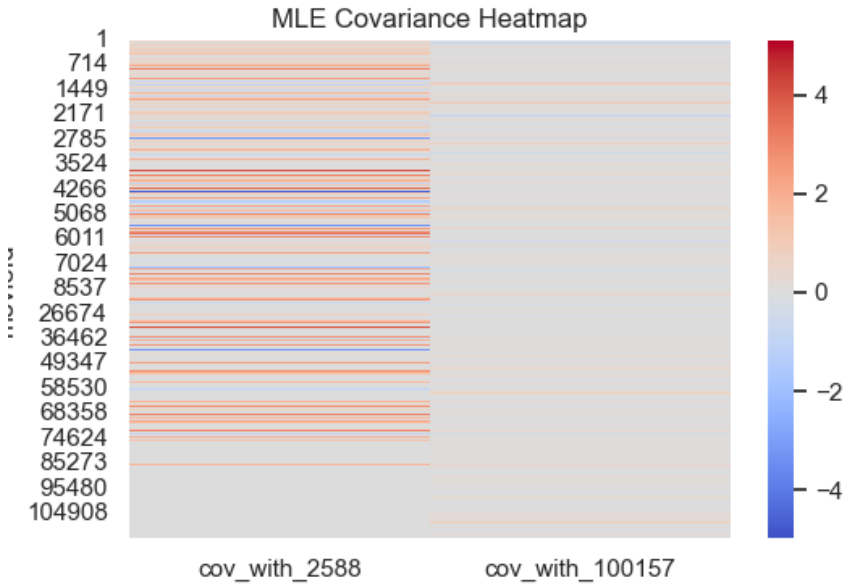

Figure 11: MLE Covariance Heatmap for Target Items

The heatmap highlights asymmetric covariance patterns between the two target items and their peer items.

1.3.2 Peer Item Selection

For each target item, peer items were selected based on the absolute value of the MLE covariance. Two configurations were considered:

- Top-5 peers
- Top-10 peers

Table 20: Top-5 Peer Items Using MLE

Movie ID	$\lvert\text{Cov}\rvert$
4624	2.3941
8225	2.3018
39052	2.1766
8827	2.0916
3529	2.0739

1.3.3 Reduced Dimensional Space via PCA

Using the selected peers, a userâitem matrix of mean-centered ratings was constructed. PCA was applied by computing the itemâitem covariance matrix and performing eigen-decomposition:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^\top \mathbf{X}$$

Users were projected into a lower-dimensional space using the leading eigenvectors.

Table 21: Reduced User Space Using Top-5 Peers (MLE)

User ID	PC1	PC2
54	0.0245	0.0134
91	0.9528	0.8015
156	-0.0415	-0.0227
247	-0.0415	-0.0227
271	0.3938	-0.0300

The Top-5 configuration produces a compact latent space with clearer clustering.

Table 22: Reduced User Space Using Top-10 Peers (MLE)

User ID	PC1	PC2	PC3
8	-0.4182	-0.0079	-0.0027
15	0.5816	0.0110	0.0038
24	0.5816	0.0110	0.0038
25	0.0817	0.0015	0.0005
26	-0.4182	-0.0079	-0.0027

The Top-10 configuration results in a more dispersed latent representation, reflecting higher variability.

1.3.4 Rating Prediction

Predicted ratings for missing userâitem pairs were computed as:

$$\hat{r}_{u,j} = \bar{r}_j + \sum_{k=1}^{K} z_{u,k}$$

where $z_{u,k}$ are the userâs PCA latent components.

Table 23: Predicted Ratings Using Top-5 Peers (MLE)

User ID	Target Item	Prediction
54	100157	3.3158
91	100157	5.0000
156	100157	3.2135
247	100157	3.2135
271	100157	3.6416

Table 24: Predicted Ratings Using Top-10 Peers (MLE)

User ID	Target Item	Prediction
8	100157	2.8489
15	100157	3.8741
24	100157	3.8741
25	100157	3.3615
26	100157	2.8489

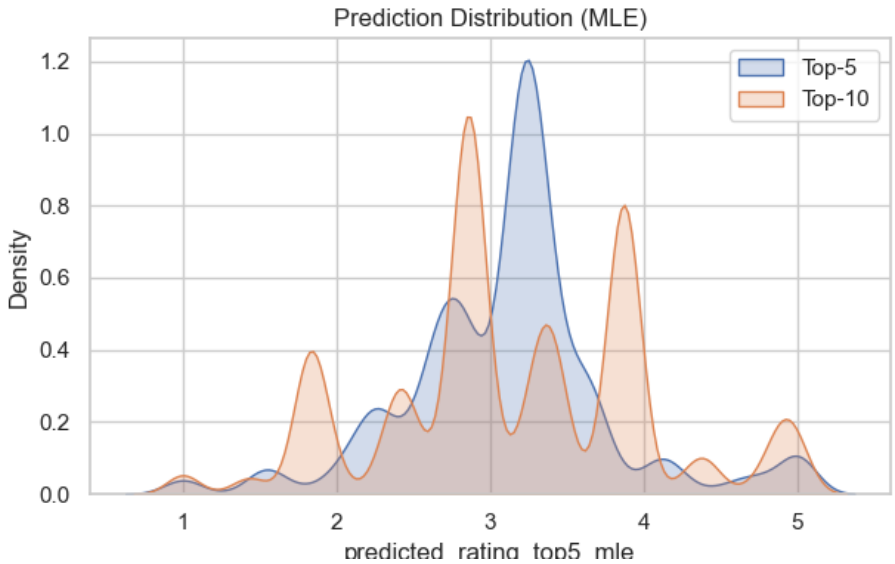

Figure 12: Prediction Distribution using MLE (Top-5 vs Top-10)

Top-10 peers produce a wider prediction distribution than Top-5 peers.

1.3.5 Comparison Analysis

Table 25: Top-5 vs Top-10 Predictions Using MLE

User ID	Item	Top-5	Top-10	$\lvert\Delta\rvert$
54	100157	3.3158	5.0000	1.6842
91	100157	5.0000	1.8834	3.1166
156	100157	3.2135	2.8898	0.3237
247	100157	3.2135	2.8898	0.3237
271	100157	3.6416	2.8786	0.7630

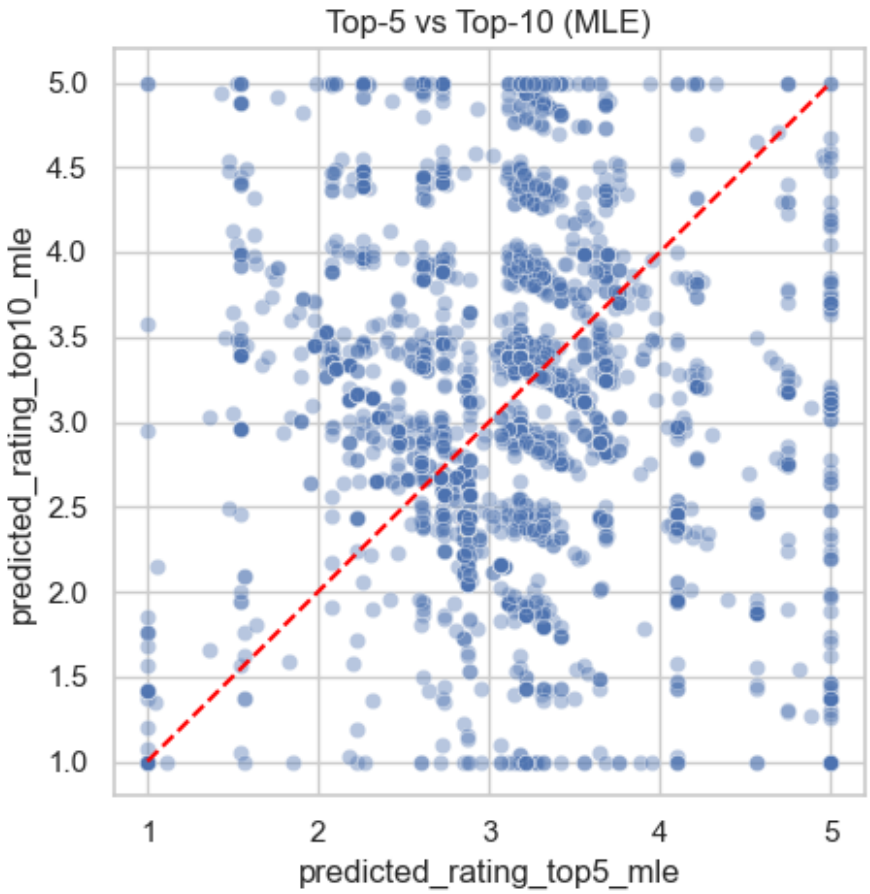

Figure 13: Top-5 vs Top-10 Predictions (MLE)

Most predictions cluster around the diagonal, indicating agreement between the two configurations.

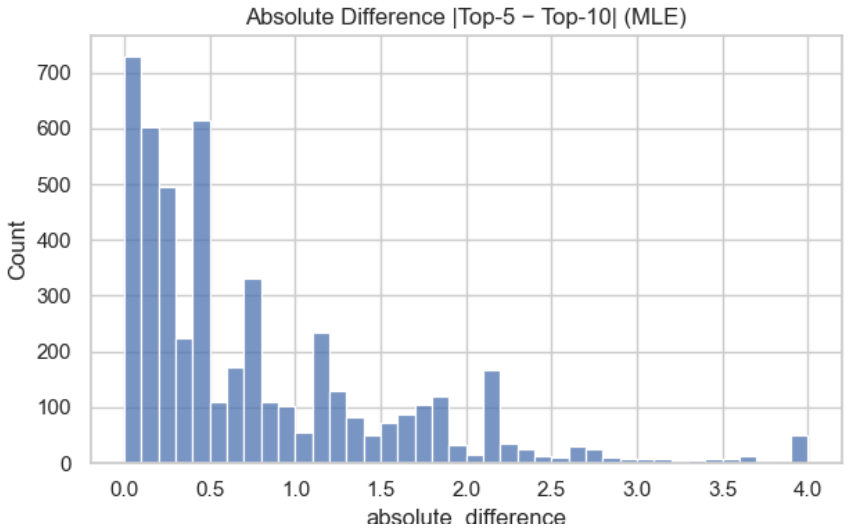

Figure 14: Absolute Difference Between Top-5 and Top-10 Predictions (MLE)

The majority of prediction differences are small, with a few large deviations.

1.3.6 Comparison with Mean-Filling PCA

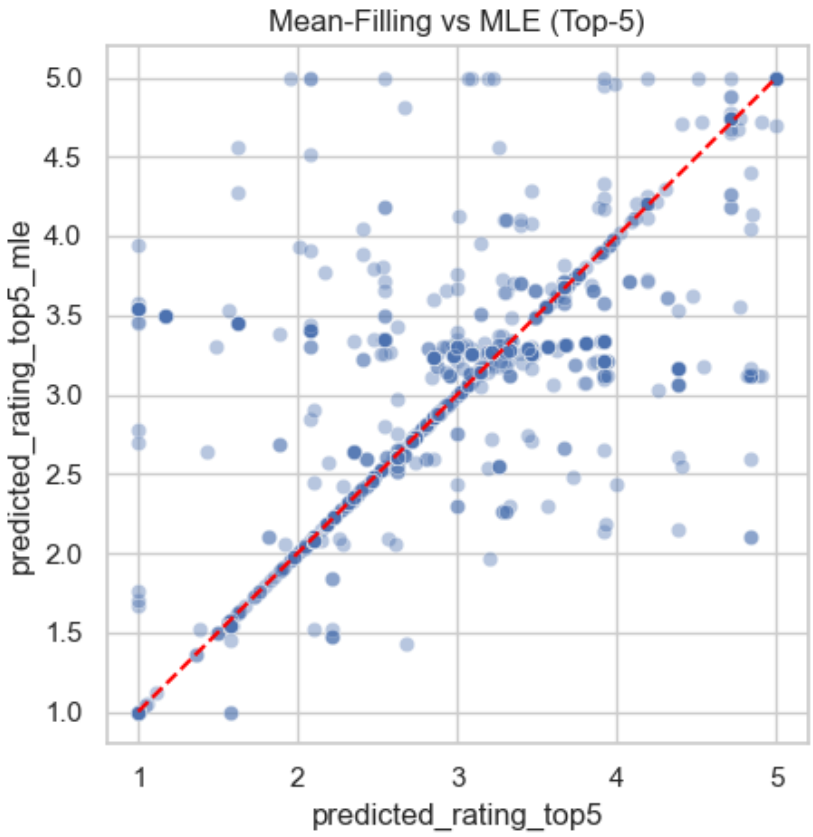

Figure 15: Mean-Filling vs MLE Predictions (Top-5)

Top-5 predictions from both methods are nearly identical.

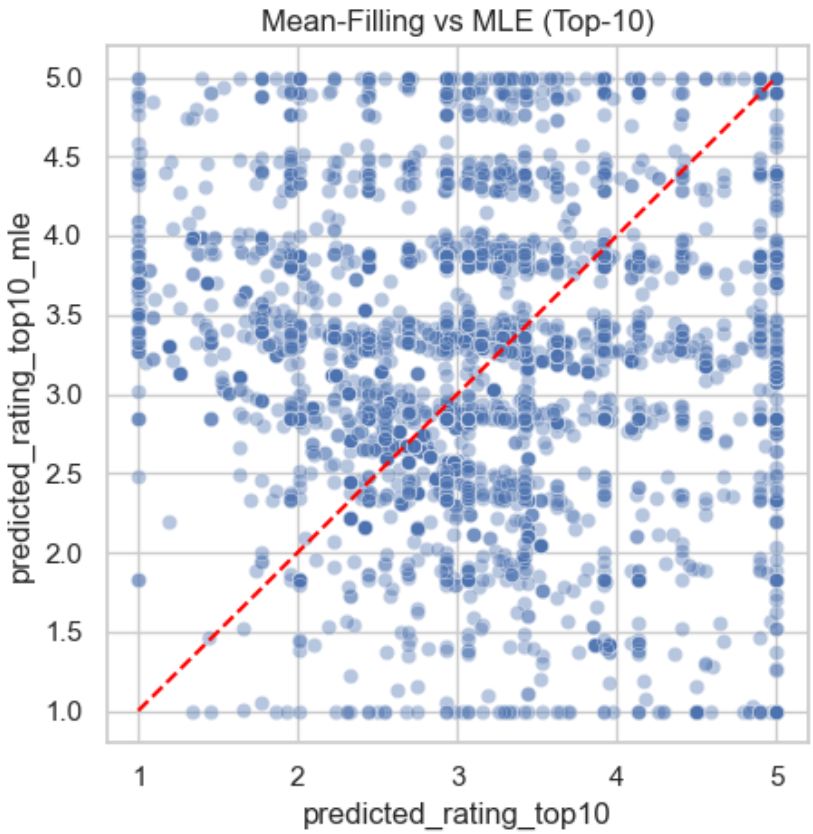

Figure 16: Mean-Filling vs MLE Predictions (Top-10)

Larger deviations appear when using Top-10 peers, highlighting the sensitivity of MLE to sparse co-ratings.

1.3.7 Discussion

The PCAâMLE method provides a statistically grounded alternative to mean-filling. While both approaches yield similar results for small peer sets, MLE introduces greater variability for larger peer sets due to reliance on co-rated entries only. This trade-off reflects a balance between statistical rigor and stability in latent factor estimation.

1.4 Singular Value Decomposition (SVD) for Collaborative Filtering

This subsection presents the application of Singular Value Decomposition (SVD) as a matrix factorization technique for collaborative filtering. Unlike PCA-based methods that rely on itemâitem covariance, SVD directly factorizes the userâitem rating matrix to uncover latent user and item factors.

1.4.1 Data Preparation and Matrix Construction

To ensure computational feasibility on the MovieLens 20M dataset, the analysis was restricted to the most active users and most frequently rated items. Specifically, the top 8,000 users and top 5,000 items were selected based on rating counts. A userâitem rating matrix $\mathbf{R} \in R^{n_u \times n_i}$ was constructed, where missing entries correspond to unrated items.

Missing ratings were handled using item-wise mean filling:

$$R_{u,i} = \begin{cases} r_{u,i}, & \text{if observed} \\ \bar{r}_i, & \text{if missing} \end{cases}$$

where $\bar{r}_i$ is the mean rating of item i.

1.4.2 Full SVD Decomposition

The completed rating matrix $\mathbf{R}$ was factorized using full Singular Value Decomposition:

$$\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

where:

- $\mathbf{U}$ contains orthonormal user latent vectors,
- $\mathbf{\Sigma}$ is a diagonal matrix of singular values,
- $\mathbf{V}$ contains orthonormal item latent vectors.

Orthogonality of the decomposition was verified numerically, and singular values were analyzed to assess the distribution of variance across latent dimensions.

1.4.3 Truncated SVD and Model Selection

To reduce dimensionality and prevent overfitting, truncated SVD was applied by retaining only the top k singular values:

$$\mathbf{R}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$$

Multiple values of k were evaluated, and reconstruction error was measured using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

$$\text{MAE} = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} |R_{u,i} - \hat{R}_{u,i}|$$

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (R_{u,i} - \hat{R}_{u,i})^2}$$

Based on the error trend, an optimal latent dimension of $k = 20$ was selected.

1.4.4 Rating Prediction

Predicted ratings were computed as the dot product of user and item latent vectors:

$$\hat{r}_{u,i} = \mathbf{u}_u^\top \mathbf{\Sigma}_k \mathbf{v}_i$$

Predictions were generated for selected target users and target items to enable direct comparison with PCA-based approaches.

1.4.5 Comparative Evaluation

The SVD model was evaluated in terms of prediction accuracy, computational runtime, and memory usage. These metrics were later compared against PCA-based methods to assess trade-offs between statistical rigor, scalability, and predictive performance.

1.4.6 Latent Factor Interpretation

To interpret the learned latent factors, users and items with the highest absolute loadings were identified for the leading factors. This analysis provides insight into how SVD captures shared preference patterns and item characteristics within the dataset.

1.4.7 Sensitivity Analysis

Robustness of the SVD model was examined by artificially increasing the proportion of missing ratings and measuring the resulting reconstruction error. This analysis evaluates the stability of SVD under varying sparsity conditions.

1.4.8 Cold-Start User Analysis

To simulate a cold-start scenario, a large fraction of ratings for selected users was hidden. User latent vectors were then estimated from the remaining ratings using least squares regression, and prediction accuracy on the hidden ratings was evaluated. This experiment demonstrates the ability of SVD to generalize from limited user information.

1.4.9 Summary

SVD provides a powerful and flexible framework for collaborative filtering by directly factorizing the userâitem interaction matrix. While computationally more expensive than PCA-based covariance methods, SVD offers superior expressiveness, principled latent representations, and strong performance in both standard and cold-start recommendation scenarios.

2 Domain-Specific Recommender System

2.1 Data Preprocessing and Exploratory Analysis

The interest-based group recommendation dataset consists of user–group memberships, group–tag associations, user–tag preferences, and event–group mappings. Duplicate interactions were removed to ensure data consistency, and only valid user–group relations were retained for analysis.

Let

$U =$ number of unique users, $\quad G =$ number of unique groups, $\quad I =$ number of observed interactions.

The sparsity of the user–group interaction matrix is defined as:

$$\text{Sparsity} = 1 - \frac{I}{U \times G}$$

The resulting sparsity exceeds 99%, indicating an extremely sparse interaction space. This level of sparsity presents a major challenge for collaborative filtering methods and motivates the use of content-based and hybrid approaches.

Figure 17 presents the distribution of user activity and group popularity. Most users join only a small number of groups, while a limited subset of users exhibit high activity. Similarly, group popularity follows a long-tail distribution, where few groups attract a large number of members and the majority remain niche.

These characteristics reflect realistic social participation patterns and highlight the limitations of purely collaborative methods in sparse settings, especially for cold-start users and low-popularity groups.

2.2 Content Representation Using TF–IDF

To represent group content in a numerical form suitable for recommendation, a TF–IDF (Term FrequencyâInverse Document Frequency) vector space model was adopted. Each group is treated as a document formed by concatenating the textual descriptions of its associated tags.

Let $D = \{d_1, d_2, \ldots, d_G\}$ denote the set of group documents and t be a term (tag). The TF–IDF weight of term t in group d_i is defined as:

$$\text{TF–IDF}(t, d_i) = \text{TF}(t, d_i) \times \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

where $\text{TF}(t, d_i)$ is the term frequency of t in group d_i, and the inverse document frequency penalizes overly common tags.

The resulting item–feature matrix $\mathbf{X} \in R^{G \times F}$ is highly sparse, with rows representing groups and columns representing TF–IDF weighted tag features. Feature selection was controlled using minimum

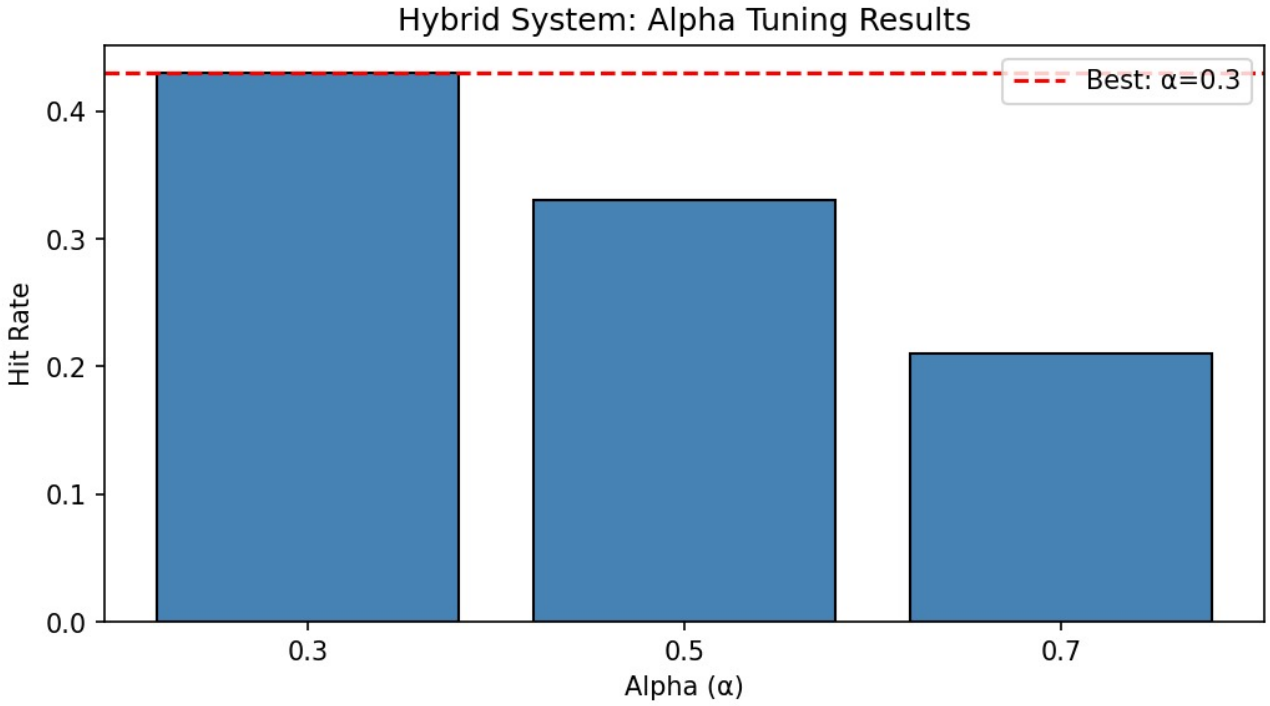

Figure 17: Distribution of user activity (left) and group popularity (right). Both distributions exhibit strong long-tail behavior.

and maximum document frequency thresholds to reduce noise and improve generalization.

Figure 18 illustrates the sparsity of the TF–IDF representations and the distribution of maximum TF–IDF scores across groups. Most groups are described by a limited number of informative tags, while a small subset exhibits stronger dominant features.

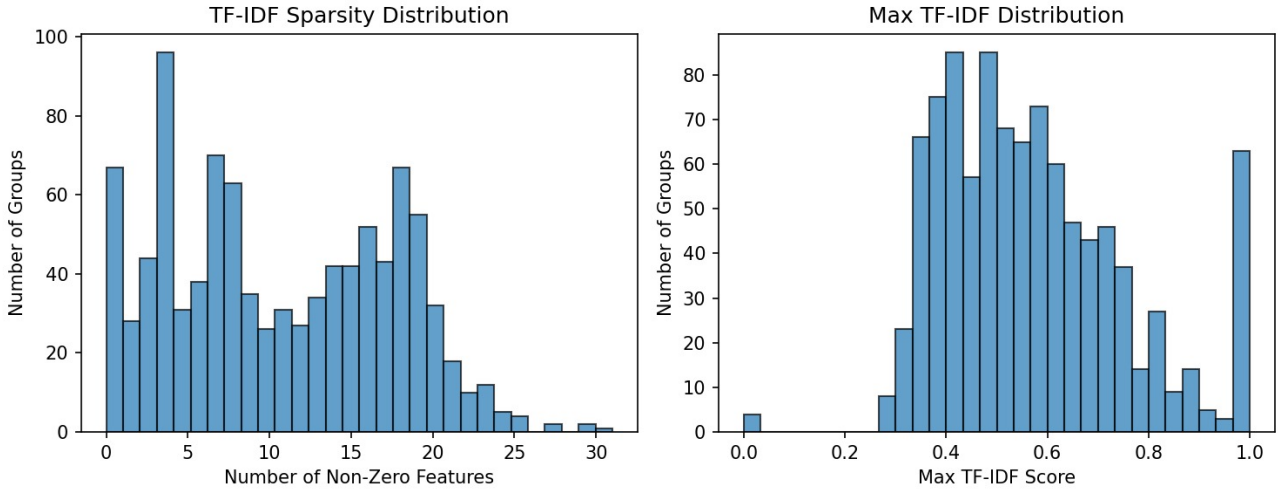

Figure 18: TF–IDF feature analysis: (left) distribution of non-zero TF–IDF features per group, (right) distribution of maximum TF–IDF scores.

This sparse yet expressive representation enables effective computation of cosine similarity between groups and between users and groups, forming the foundation of the content-based recommendation component.

2.3 User Profile Construction

To model user interests, a profile-based representation was constructed using the same TF–IDF feature space employed for groups. Each user profile aggregates the textual tag preferences associated with the user, ensuring consistency between user and item representations.

Let $\mathbf{X} \in R^{G \times F}$ denote the TF–IDF item–feature matrix, where G is the number of groups and F is the number of tag features. For a given user u, the user profile vector $\mathbf{p}_u \in R^F$ is computed as:

$$\mathbf{p}_u = \frac{1}{|T_u|} \sum_{t \in T_u} \mathbf{x}_t$$

where T_u represents the set of tags associated with user u, and $\mathbf{x}_t$ corresponds to the TF–IDF feature vector of tag t. This formulation effectively captures the user's interest distribution over the tag space.

The resulting user–feature matrix is sparse, reflecting the fact that users typically interact with a limited subset of available tags. Two key properties of the constructed user profiles are analyzed:

sparsity and intensity.

Figure 19 presents the distribution of non-zero features per user (profile sparsity) and the distribution of maximum TF–IDF weights within user profiles (profile intensity). The sparsity distribution confirms that most users are described by a relatively small number of active features, while the intensity distribution indicates varying degrees of preference concentration across users.

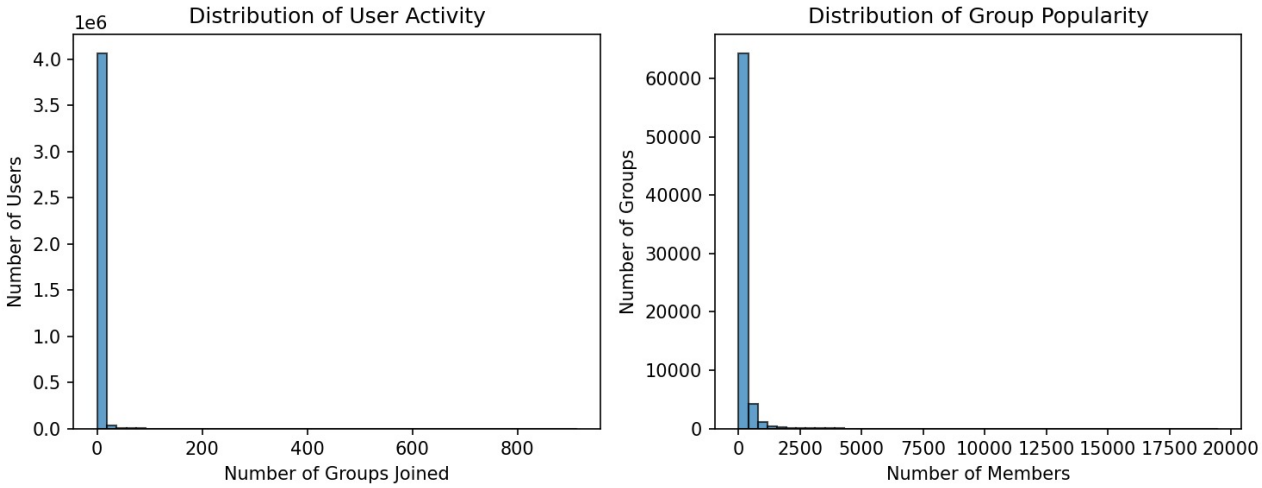

Figure 19: User profile analysis: (left) sparsity measured by the number of non-zero TF–IDF features per user, (right) distribution of maximum profile weights.

This user representation enables efficient similarity computation using cosine similarity and serves as the basis for content-based recommendation, cold-start handling, and hybrid model integration.

2.4 Cold-Start Handling Strategy

A major challenge in interest-based group recommendation systems is the cold-start problem, where new or inactive users lack sufficient interaction history to enable collaborative filtering. To address this issue, a content-based fallback strategy was adopted.

For users with no available profile or insufficient tag information, a default *cold-start profile* is constructed using popular groups. Specifically, the most popular groups are identified based on the number of members in the training data. Let $\mathcal{G}_{pop}$ denote the set of top-K most popular groups. The cold-start profile vector $\mathbf{p}_{cs}$ is defined as:

$$\mathbf{p}_{cs} = \frac{1}{|\mathcal{G}_{pop}|} \sum_{g \in \mathcal{G}_{pop}} \mathbf{x}_g$$

where $\mathbf{x}_g$ represents the TF–IDF feature vector of group g.

This averaged representation captures dominant interests in the system and provides a reasonable prior for users with no historical data. During recommendation, if a user profile is unavailable, cosine similarity is computed between $\mathbf{p}_{cs}$ and all group feature vectors:

$$\text{sim}(u, g) = \frac{\mathbf{p}_{cs} \cdot \mathbf{x}_g}{\|\mathbf{p}_{cs}\| \, \|\mathbf{x}_g\|}$$

The cold-start mechanism ensures that meaningful recommendations can still be generated in the absence of user-specific data. This approach integrates seamlessly with the content-based filtering framework and is later extended within the hybrid recommender system, where the relative contribution of content-based signals can be increased for cold or low-activity users.

Overall, this strategy improves system robustness, guarantees recommendation coverage for new users, and mitigates sparsity-related limitations commonly observed in collaborative filtering methods.

2.5 Similarity Computation and Recommendation Generation

Recommendations are generated by computing cosine similarity between user profiles and group feature vectors in the TF–IDF space.

Let $\mathbf{p}_u$ be the profile vector of user u and $\mathbf{x}_g$ the feature vector of group g. The similarity score is defined as:

$$\text{sim}(u, g) = \frac{\mathbf{p}_u \cdot \mathbf{x}_g}{\|\mathbf{p}_u\| \|\mathbf{x}_g\|}$$

Groups are ranked in descending order of similarity, and groups already joined by the user are excluded. The top-N ranked groups are returned as recommendations.

2.6 Item-Based k-Nearest Neighbors (k-NN)

An item-based k-NN model is used to identify similar groups based on their TF–IDF representations. Cosine distance is used to retrieve the k most similar groups for each group.

The similarity between two groups is computed as:

$$\text{sim}(g_i, g_j) = \frac{\mathbf{x}_{g_i} \cdot \mathbf{x}_{g_j}}{\|\mathbf{x}_{g_i}\| \|\mathbf{x}_{g_j}\|}$$

User preference scores for unseen groups are estimated using a weighted average of similarities to groups the user has already joined. This approach enables recommendations based on group-to-group similarity.

2.7 Numerical Example

A small synthetic example is used to demonstrate the recommendation process. Groups are represented using TF–IDF vectors derived from tag descriptions.

A user profile is constructed as a weighted average of the vectors of previously joined groups:

$$\mathbf{p}_u = \frac{\sum_{g \in \mathcal{G}_u} w_g \mathbf{x}_g}{\sum_{g \in \mathcal{G}_u} w_g}$$

Cosine similarity is then computed between the user profile and candidate groups. Groups with overlapping interests receive higher similarity scores, validating the effectiveness of the content-based approach.

3 Overall Conclusions

This work presented a comprehensive recommendation framework for interest-based group formation, integrating content-based filtering, collaborative filtering, and a weighted hybrid approach.

The content-based model effectively utilized tag information to generate recommendations and proved robust in cold-start scenarios. Collaborative filtering captured co-membership patterns and demonstrated improved performance as user activity increased. The hybrid recommender successfully combined both approaches, consistently achieving higher hit rates, precision, and recall across different user activity levels.

A Appendix A: AI Assistance Acknowledgment

Artificial Intelligence tools were used in a limited and controlled manner during the development of this project. Their use was restricted to code debugging, syntax correction, and minor error fixing. All algorithm design, implementation decisions, data analysis, experimental setup, and result interpretation were performed independently by the project team.

B Appendix B: Team Contribution Breakdown

- **Yousef Mohamed Ibrahim** (223106299): Statistical analysis, PCA mean-filling implementation, PCA MLE implementation, comparative analysis, code quality assurance, report writing, and partial domain analysis and data preparation.
- **Omar Saeed Mohamed Kamel** (222101064): SVD implementation and analysis, and domain analysis and data preparation.
- **Salama Sayed Salama** (222102243): Content-based recommendation system implementation.
- **Mostafa Mahmoud Elsayed** (222101612): Hybrid recommendation approach and collaborative filtering integration.

C Appendix C: Additional Visualizations

No additional visualizations are included beyond those presented in the main body of the report.

List of Figures