*Nanotechnology Perceptions*
*ISSN 1660-6795*
*E-ISSN:2235-2074*
www.nano-*ntp.com*

# "An Empirical Analysis OF Clustering-Based Hybrid Recommender Systems Using Rmse"

**[1,*]Prakash Kumar Lange, [2]Dr. Balendra Kumar Garg**

*[1]Research Scholar, MATS University Raipur, C.G., India,*
*[2]Assistant Professor, School of Information Technology, MATS University, Raipur, Chhattisgarh*
**prakashkumarlange@gmail.com*

Recommender systems are essential for delivering personalized content in multiple fields including e-commerce, entertainment, and education. The research article introduces a hybrid recommendation framework that merges clustering methods with collaborative filtering techniques to improve the precision of recommendations. Users are segmented through their interaction patterns using three clustering methods: KMeans, Agglomerative Clustering, and DBSCAN. The Singular Value Decomposition method serves collaborative filtering purposes while root mean square error functions as the performance metric for various clustering models. Test results show that adding clustering techniques effectively reduces data sparsity while enhancing prediction accuracy.

**Keywords:** Hybrid Recommendation System, Collaborative Filtering, Content-Based Filtering, K-Means Clustering, SVD, Cosine Similarity.

## 1. Introduction

Digital platform users rely on recommender systems to manage the massive amount of information available online. By utilizing algorithms to assess user preferences these systems produce personalized recommendations which improve user satisfaction and engagement. The main traditional recommendation approaches divide into content-based filtering methods as well as collaborative filtering techniques along with hybrid models [3,4]. Despite their effectiveness collaborative filtering faces challenges of data sparsity and cold-start problems when new users and items lack enough historical data to generate accurate recommendations [4, 5].

Large-scale recommender systems frequently encounter difficulties due to data sparsity [4]. A significant number of entries in a standard user-item interaction matrix are unobserved which results in challenges to extract meaningful patterns [5-7]. Traditional collaborative filtering methods fail to produce optimal recommendations since they depend on user-item interactions [6]. The recommendation process becomes more robust when clustering techniques are used to group users according to their behavior patterns. User clustering based on shared

preferences improves prediction reliability because recommendations are generated from collective patterns within each cluster instead of individual user histories [6].

A hybrid approach emerges from combining clustering techniques with collaborative filtering as it merges the advantages of both methods [9]. KMeans, Agglomerative Clustering, and DBSCAN each offer specific advantages for structuring user interactions [10, 11]. The KMeans algorithm assigns users to predetermined clusters and minimizes each cluster's internal variance. Agglomerative Clustering creates user clusters by iteratively merging similar entities and preserving a natural tree structure. The DBSCAN algorithm clusters users based on dense interaction regions by effectively identifying outliers and noise in the dataset. The performance of hybrid recommendation systems depends heavily on choosing the right clustering method.

Mathematically define U as the set of users and I as the set of items with R representing the user-item rating matrix which contains $r_{ui}$ as the rating given by user u for item i. The clustering function $C: U \rightarrow \{C_1, C_2, C_3, \ldots C_n, \}$ assigns each user to one of k clusters based on similarity measures, such as Euclidean distance or cosine similarity [16]. Collaborative filtering techniques become applicable within each cluster after completing the clustering which leads to improved prediction accuracy [17]. SVD is utilized to decompose the rating matrix into three matrices of lower dimensions:

$$R \approx U \sum V^T$$

where U and V are orthogonal matrices capturing user and item latent factors, respectively, and $\sum$ is a diagonal matrix containing singular values representing the importance of each latent feature.

The effectiveness of this hybrid approach is evaluated using root mean square error (RMSE), a widely used metric for measuring prediction accuracy in recommender systems. RMSE is computed as [18]:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)}$$

where $y_i$ represents the actual ratings, $\hat{y}_i$ are the predicted ratings, and N is the total number of predictions. Lower RMSE values indicate higher accuracy, and comparative analysis among clustering techniques provides insights into their impact on recommendation performance.

The contributions of this paper are threefold: (i) integrating clustering techniques with collaborative filtering to reduce data sparsity and improve recommendation quality, (ii) evaluating the performance of different clustering models using RMSE, and (iii) demonstrating the efficacy of the hybrid approach through empirical experiments on the MovieLens dataset. The subsequent sections of this paper delve into related work, methodology, experimental results, and conclusions, providing a comprehensive analysis of the proposed system.

## 2. Related Work

Research literature contains extensive examination of recommendation systems with special emphasis on collaborative filtering (CF), content-based filtering (CB), and hybrid models. User-based and item-based collaborative filtering methods represent traditional CF approaches that utilize user-item interaction matrices to produce recommendation outputs. Despite their utility these methods face significant challenges from sparsity problems and cold-start issues that reduce recommendation precision [5]. Collaborative filtering has shown significant enhancements through matrix factorization methods like Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) which extract latent features from user-item matrices [3-7]. These methods reduce sparsity problems partially through the identification of underlying user preference trends. Digital platform users rely on recommender systems to manage the massive amount of information available online. By utilizing algorithms to assess user preferences these systems produce personalized recommendations which improve user satisfaction and engagement. The main traditional recommendation approaches divide into content-based filtering methods as well as collaborative filtering techniques along with hybrid models [3,4]. Despite their effectiveness collaborative filtering faces challenges of data sparsity and cold-start problems when new users and items lack enough historical data to generate accurate recommendations [4, 5].

Large-scale recommender systems frequently encounter difficulties due to data sparsity [4]. A significant number of entries in a standard user-item interaction matrix are unobserved which results in challenges to extract meaningful patterns [5-7]. Traditional collaborative filtering methods fail to produce optimal recommendations since they depend on user-item interactions [6]. The recommendation process becomes more robust when clustering techniques are used to group users according to their behavior patterns. User clustering based on shared preferences improves prediction reliability because recommendations are generated from collective patterns within each cluster instead of individual user histories [6][20].

A hybrid approach emerges from combining clustering techniques with collaborative filtering as it merges the advantages of both methods [9]. KMeans, Agglomerative Clustering, and DBSCAN each offer specific advantages for structuring user interactions [10, 11]. The KMeans algorithm assigns users to predetermined clusters and minimizes each cluster's internal variance. Agglomerative Clustering creates user clusters by iteratively merging similar entities and preserving a natural tree structure. The DBSCAN algorithm clusters users based on dense interaction regions by effectively identifying outliers and noise in the dataset. The performance of hybrid recommendation systems depends heavily on choosing the right clustering method.

## .3. Methodology

The methodology incorporates multiple essential stages which work together to improve the accuracy of recommendations. The next subsections provide information about the dataset as well as preprocessing steps and explain clustering techniques alongside content-based similarity computation methods and collaborative filtering before evaluating performance.

### 3.1 Dataset

This research employs the MovieLens dataset which serves as a standard benchmark for studying recommendation systems according to [21]. Personalized recommendation models benefit from the dataset which contains user-generated movie ratings. Two primary files are employed:

- ratings.csv: Contains user-item interaction data with attributes (userId, movieId, rating, timestamp).
- movies.csv: Provides metadata for movies, including (movieId, title, genres).

Since user preferences are inherently sparse due to the vast number of movies and limited individual interactions, appropriate preprocessing and clustering techniques are necessary to enhance the quality of recommendations.

### 3.2 Preprocessing

Structured data representation is enabled by merging the ratings dataset with movie metadata to align user interactions to respective movie attributes [4]. This process results in the creation of a user-item matrix R sized m×n where m stands for users and n stands for movies. The algorithm initializes missing matrix values that represent unrated movies to zero to ensure computational feasibility [7]. To reduce rating biases between users and achieve better model stability rating normalization is implemented. The process of mean-centering user ratings represents a widely-used normalization method:

$$\tilde{r}_{ui} = r_{ui} - \bar{r}_u$$

Where $r_{ui}$ represents the rating given by user $u$ to movie $i$ and $\bar{r}_u$ is the mean rating of user $u$. This transformation ensures that user-specific rating tendencies do not disproportionately affect the clustering process

### 3.3 Clustering Users

To address data sparsity and improve collaborative filtering performance, users are segmented into clusters based on their rating behavior. Three clustering techniques are explored:

a) K-Means Clustering:

K-Means partitions users into k clusters by minimizing intra-cluster variance [5]. The objective function is expressed as:

$$J = \sum_{x=1}^{k} \sum_{x_j \in C_i} \|x_j - u_i\|^2$$

where $C_i$ represents the $i^{th}$ cluster, $u_i$ denotes the centroid, and $x_j$ is a user's rating vector. K-Means is computationally efficient but assumes spherical clusters, limiting its applicability to non-Euclidean user preferences.

b) Agglomerative Clustering:

A hierarchical clustering approach that iteratively merges users into clusters based on a predefined linkage criterion, such as Ward's method [8]:

$$d(C_i, C_j) = \|\mu_i - \mu_j\|^2$$

where $d(C_i, C_j)$ denotes the inter-cluster distance between clusters $C_i$ and $C_j$. Unlike K-Means, Agglomerative Clustering does not require prior specification of k, offering flexibility in user segmentation.

c) DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

A density-based clustering technique that detects user communities with similar rating behaviors [12]. A point p is classified as a core point if:

$$|N_\epsilon(p)| \geq \text{minPts}$$

where $N_\epsilon(p)$ is the set of neighboring points within radius $\epsilon$. DBSCAN is robust to noise and does not assume cluster convexity, making it suitable for non-linear user preferences.

Following clustering, each user is assigned a cluster label, allowing collaborative filtering models to be trained within user subgroups rather than the entire dataset, thereby improving recommendation accuracy.

### 3.4 Content-Based Filtering

To incorporate content information, one-hot encoding is applied to movie genres, creating a binary feature matrix $G$ of dimension $n \times d$ where $d$ is the number of unique genres. The similarity between movies is then computed using cosine similarity [17-19]:

$$S_{ij} = \frac{G_i \cdot G_j}{\|G_i\| \cdot \|G_j\|}$$

Where $S_{ij}$ represents the similarity scores between i and j. This similarity matrix enhances recommendation diversity by enabling the retrieval of contentually related items.

### 3.5 Collaborative Filtering with SVD

To leverage implicit user preferences, Singular Value Decomposition (SVD) is employed on the user-item matrix $R$, decomposing it as follows [14]:

$$R \approx U \sum V^T$$

Where

- $U \epsilon \mathbb{R}^{m \times k}$, Captures user latent features
- $\sum \epsilon \mathbb{R}^{m \times k}$, contains singular values, and

- $V \epsilon \mathbb{R}^{m \times k}$, represents movie latent features.

SVD enables dimensionality reduction by retaining only the top $k$ singular values, preserving essential preference information while discarding noise [13]. The dataset is split into training (80%) and testing (20%) sets, and the SVD model is trained to predict missing ratings:

$$\hat{r}_{ui} = U_u \sum V_i^T$$

where $\hat{r}_{ui}$ is the predicted rating of user $u$ and movie $i$. The model is optimized using Stochastic Gradient Descent (SGD) to minimize the error between actual and predicted ratings.

## 4. Results and Discussion

To evaluate the ability of model to predict and recommend relevant items quantitative measures like Root Mean Squared Error (RMSE) and Precision at K are used as primary assessment metrics. Different clustering techniques and dimensionality reduction methods are used to validate the effectiveness of the proposed methodology through further analysis of the findings.

### 4.1 Impact of User Clustering on Collaborative Filtering Performance

Implementing user clustering before collaborative filtering creates a notable impact on model predictive accuracy. The analysis of K-Means, Agglomerative Clustering, and DBSCAN shows that each algorithm delivers different performance characteristics. The application of K-Means clustering with an optimal cluster count determined through the Elbow Method produces the lowest RMSE values which confirms its ability to effectively segment users based on similar preferences. Agglomerative Clustering shows slightly worse error rates because its hierarchical nature limits the adaptability of the clusters. DBSCAN demonstrates strong performance in finding dense user clusters but experiences accuracy loss from noise during sparse user-item interaction analysis. Despite different results among clustering methods it becomes clear that clustering strengthens collaborative filtering effectiveness through sparsity reduction and improved user representation. The consistent decrease in RMSE values across different clustering approaches highlights the advantages of segmenting users before implementing recommendation systems.

### 4.2 Performance of Content-Based Filtering and Hybridization

The use of cosine similarity on one-hot encoded genres enables content-based filtering to enhance personalized recommendations through item attribute analysis. The content-based model shows effectiveness in suggesting movies with similar themes when evaluated independently yet encounters limitations from the cold-start problem when users have sparse interaction histories.

The combination of content-based filtering with collaborative filtering enhances recommendation precision significantly. The recommendation model achieves a superior Precision@K score for top-N suggestions through its effective merging of explicit preferences

and implicit content relationships. The results support the theory that combining filtering approaches produces more diverse recommendations without sacrificing relevance.

### 4.3 Effectiveness of SVD-Based Collaborative Filtering

Singular Value Decomposition (SVD) proves beneficial in collaborative filtering by effectively extracting the hidden elements that define user preferences. Multiple experimental runs show lower RMSE values for dimensionality reduction via SVD which enhances model generalization. The analysis of training and testing data splits demonstrates that dividing data with an 80-20 ratio produces the most consistent predictive accuracy while maintaining a proper balance between training data sufficiency and test validation.

The prediction accuracy improves because iterative optimization of the SVD model is achieved through Stochastic Gradient Descent (SGD). The consistent reduction of RMSE across multiple iterations demonstrates the stable nature of the optimization procedure. The number of latent factors needs precise empirical tuning to achieve the lowest RMSE which underscores the significance of hyperparameter selection in matrix factorization methods.

### 4.4 Comparative Evaluation of Different Model Configurations

The results show that a hybrid model delivers better performance than individual collaborative filtering (CF), clustering-enhanced CF, and content-based filtering (CBF) techniques. The integrated CF-CBF approach achieves both higher accuracy and improved user satisfaction through multiple recommendation strategies.

The experimental results confirm the effectiveness of the proposed methodology through demonstrated enhancements in predictive accuracy along with recommendation diversity and user satisfaction. The research supports the view that combining clustering methods with content-based filtering and SVD-based collaborative filtering creates a stronger recommendation system compared to traditional single-method approaches.

**Table 1: Comparative Performance of Different Recommendation Approaches**

| Model Configuration | RMSE (↓ Lower is Better) | Key Observations |
|---|---|---|
| **Baseline Collaborative Filtering (SVD Only)** | 0.945 | Standard SVD-based collaborative filtering without clustering. |
| **K-Means + Collaborative Filtering** | **0.892** | Lowest RMSE, indicating optimal user segmentation. |
| **Agglomerative Clustering +** | 0.915 | Slightly higher RMSE due to rigid hierarchical clusters. |

| | | |
|---|---|---|
| **Collaborative Filtering** | | |
| **DBSCAN + Collaborative Filtering** | 0.931 | Higher RMSE due to noise sensitivity and handling of sparse data. |
| **Content-Based Filtering Only** | 1.103 | High RMSE due to lack of collaborative information. |
| **Hybrid Model (CBF + CF with K-Means)** | **0.873** | Best performance; benefits from both user preferences and item attributes. |

## Conclusion

This study presented and tested a hybrid recommendation system which combines user clustering with both content-based filtering and collaborative filtering through Singular Value Decomposition (SVD). K-Means clustering before collaborative filtering results in better prediction accuracy with an RMSE of 0.892 whereas Agglomerative Clustering and DBSCAN achieve RMSEs of 0.915 and 0.931 respectively. The greatest prediction error occurred when using content-based filtering by itself, as demonstrated by an RMSE of 1.103. The hybrid model combining content-based and collaborative filtering with K-Means clustering produced the smallest RMSE score of 0.873 demonstrating its superior capability to improve recommendation precision. The application of user segmentation through clustering techniques reduces data sparsity which results in improved collaborative filtering performance. K-Means clustering emerged as the leading technique for accurately identifying user preference similarities. Content-based filtering integration enhances recommendations through item attribute analysis which leads to better personalization. The research demonstrates that hybrid models provide essential value to recommender systems when dealing with sparse user-item interaction domains. Subsequent research should investigate the combination of deep learning methods with reinforcement learning to enhance both recommendation precision and flexibility.

## Acknowledgement

The authors would like to thank the institution's administrative body for providing all the necessary resources to conduct the study.

## References

1. Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, pp. 68–81, Jan. 2016.
2. X. Su, H. Zhang, Y. Yu, and Y. Zhang, "A Hybrid Collaborative Filtering Algorithm Based on Clustering and Regression," in Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 2017, pp. 123–128.

3. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender Systems Survey," Knowledge-Based Systems, vol. 46, pp. 109–132, Jul. 2013.
4. S. Li, J. Kawale, and Y. Fu, "Deep Collaborative Filtering via Marginalized Denoising Autoencoder," in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM), Melbourne, Australia, 2015, pp. 811–820.
5. H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Sydney, Australia, 2015, pp. 1235–1244.
6. X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast Matrix Factorization for Online Recommendation with Implicit Feedback," in Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Pisa, Italy, 2016, pp. 549–558.
7. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web (WWW), Perth, Australia, 2017, pp. 173–182.
8. S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural Collaborative Filtering vs. Matrix Factorization Revisited," in Proceedings of the 14th ACM Conference on Recommender Systems (RecSys), Virtual Event, Brazil, 2020, pp. 240–248.
9. M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Performance Comparison of Neural and Non-Neural Approaches to Session-Based Recommendation," in Proceedings of the 13th ACM Conference on Recommender Systems (RecSys), Copenhagen, Denmark, 2019, pp. 462–466.
10. D. Agarwal and B.-C. Chen, "fLDA: Matrix Factorization through Latent Dirichlet Allocation," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Paris, France, 2009, pp. 91–99.
11. D. Jannach, L. Lerche, F. Gedikli, and G. Bonnin, "What Recommenders Recommend – An Analysis of Accuracy, Popularity, and Sales Diversity Effects," in User Modeling, Adaptation, and Personalization, Rome, Italy, 2013, pp. 25–37.
12. X. Bi, A. Qu, J. Wang, and X. Shen, "A Group-Specific Recommender System," Journal of the American Statistical Association, vol. 112, no. 519, pp. 1344–1353, 2017.
13. Y. Zhu, X. Shen, and C. Ye, "Personalized Prediction and Sparsity Pursuit in Latent Factor Models," Journal of the American Statistical Association, vol. 107, no. 499, pp. 1146–1156, 2012.
14. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering," in Proceedings of KDD Cup and Workshop, San Jose, CA, USA, 2007, pp. 5–8.
15. M. Jahrer, A. Töscher, and R. Legenstein, "Combining Predictions for Accurate Recommender Systems," in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Washington, DC, USA, 2010, pp. 693–702.
16. R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in Advances in Neural Information Processing Systems 20 (NIPS 2007), Vancouver, Canada, 2008, pp. 1257–1264.

17. G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Investigation of Various Matrix Factorization Methods for Large Recommender Systems," in Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys), Lausanne, Switzerland, 2008, pp. 245–248.
18. Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.
19. S. Funk, "Netflix Update: Try This at Home," 2006. [Online]. Available: https://sifter.org/~simon/journal/20061211.html
20. H.-H. Chen and P. Chen, "Differentiating Regularization Weights – A Simple Mechanism to Alleviate Cold Start in Recommender Systems," ACM Transactions on Knowledge Discovery.
21. https://movielens.org/