IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

Item-Based Collaborative Filtering with Cosine Similarity for Personalized Movie Recommendation: A Comparative Study on the Movielens 100K Dataset

Mr. Siddharth Sanjay Bade[1] and Prof. J. R. Mahajan[2]
[1]Student, Department of Artificial Intelligence and Data Science
[2]Guide and Head of Department, Department of Artificial Intelligence and Data Science
Adsul's Technical Campus, Chas, Ahilyanagar, India

Abstract: *Recommendation systems help users find useful items in very large catalogues. Collaborative filtering (CF) is one of the most common approaches, but its results depend on the similarity measure, the neighbourhood size and the way scores are turned into a ranked list. This paper compares user-based and item-based CF with cosine similarity on the MovieLens 100K dataset (943 users, 1,682 movies, 100,000 ratings), using the five official train/test splits. We also include simple baselines and a biased matrix factorization (MF) model. Two tasks are studied: rating prediction (RMSE, MAE) and top-10 recommendation (Precision@10, Recall@10). Neighbourhood size and a shrinkage term were chosen on validation data taken from the training set. MF gave the lowest rating error (RMSE 0.926). User-based CF (0.942) and item-based CF with adjusted cosine (0.947) were close, and their difference is smaller than the fold-to-fold variation. For top-10 recommendation, ranking by a sum of similarities gave a Precision@10 of 0.267 (item-based) and 0.269 (user-based), against 0.157 for a popularity baseline, while ranking neighbourhood methods by predicted rating gave a Precision@10 near zero. On this small dataset we found no evidence that item-based CF is more accurate than user-based CF.*

Keywords: Recommender Systems, Collaborative Filtering, Cosine Similarity, Matrix Factorization, MovieLens, Offline Evaluation

I. INTRODUCTION

Online platforms offer thousands of movies, songs and products, and users cannot look through all of them. Recommendation systems solve this problem by suggesting items that a user is likely to like, using past behaviour such as ratings, clicks and purchases. Large services such as Amazon and Netflix use recommendations as a core part of their product [2], [3].

Collaborative filtering (CF) predicts a user's interest from the behaviour of similar users or similar items. It does not need item descriptions, which makes it easy to apply to many domains. Its main weakness is data sparsity: most users rate only a small part of the catalogue. Recent work tries to reduce this problem with hybrid models [5], [6], and to go beyond it with large language models [4], [10], [11].

Although newer models exist, neighbourhood-based CF is still a common baseline and is often the first method students and practitioners implement. Its results, however, depend on choices that are often not reported clearly: the similarity measure, the number of neighbours, and how a ranked list is produced. This paper studies these choices on a standard public dataset. Our contributions are:

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40432

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

a side-by-side comparison of user-based CF, item-based CF (raw and adjusted cosine) and biased matrix factorization on MovieLens 100K, using all five official splits and reporting mean and standard deviation;
a separate evaluation of rating prediction and top-10 recommendation, which shows that a method can be good at one and poor at the other;
an honest report of where our results differ from commonly repeated claims, in particular that item-based CF is not clearly better than user-based CF on this dataset.

II. RELATED WORK

Sarwar *et al.* [1] introduced item-based CF and reported that it gave better quality and faster online performance than user-based CF. Linden *et al.* [2] described an item-to-item approach for a very large catalogue at Amazon. Deshpande and Karypis [7] studied item-based methods specifically for top-N recommendation, where items are ranked by combining similarities to items the user already likes. Koren *et al.* [3] showed that matrix factorization models gave strong accuracy in the Netflix Prize. A broader survey of recommender system types and their limits, such as cold start, is given in [8].

Recent studies from 2024 and 2025 follow several directions. Zhao *et al.* [4] review recommender systems that use large language models, organised around pre-training, fine-tuning and prompting. Shehmir and Kashef [10] survey the approaches, applications and challenges of bringing large language models into recommender systems. Vinayababu and Raju [11] give a survey of modern recommender systems that covers collaborative, content-based, hybrid and deep-learning approaches and the challenges they face, such as sparse data. Huda *et al.* [5] propose a hybrid technique with content-boosted collaborative filtering for smart tourism recommendation. Lv *et al.* [6] address the sparsity problem of CF with a hybrid e-commerce algorithm based on a user nearest-neighbour model, in which the user–item matrix is first improved and then used as input to a CF model; they report that the hybrid beats stand-alone CF.

TABLE I
RECENT STUDIES AND THEIR RELEVANCE TO THIS WORK

Study	Year	Focus	Relevance to this work
Zhao *et al.* [4]	2024	Review of language-model-based recommenders	Shows the move towards language models; classical CF is still the usual baseline.
Huda *et al.* [5]	2024	Hybrid, content-boosted CF for tourism	Shows how CF is combined with content data.
Lv *et al.* [6]	2024	Hybrid method based on user nearest neighbours	Targets the sparsity problem of CF.
Shehmir and Kashef [10]	2025	Survey of LLMs in recommender systems	Gives approaches, applications and challenges of newer models.
Vinayababu and Raju [11]	2025	Survey of modern recommender systems	Covers CF, content-based, hybrid and deep-learning methods.
Present study	**2026**	**Controlled comparison of classical CF**	**Fixes the data splits and tuning, and compares rating prediction with top-10 ranking.**

These papers mostly use more complex models. In this work we go back to the basic neighbourhood methods and ask how they behave on a standard benchmark when the neighbourhood size and the ranking rule are chosen carefully and the results are reported over several splits.

III. DATASET AND PROBLEM SETTING

We use the MovieLens 100K dataset [9]. Table II gives its main statistics. We use the five official 80%/20% splits (u1–u5), whose test sets do not overlap. Two tasks are considered. In *rating prediction*, the model predicts the rating a user

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

would give to a movie in the test set. In *top-N recommendation*, the model ranks movies the user has not rated in the training set, and we check whether the top 10 contain movies the user later rated 4 or 5 (called relevant items).

TABLE II: MOVIELENS 100K STATISTICS

Property	Value
Users	943
Movies	1,682
Ratings	100,000
Rating scale	1–5 (integers)
Minimum ratings per user	20
Average ratings per user	106.0
Sparsity of the matrix	93.7%
Train / test per split	80,000 / 20,000

IV. METHODS

A. Item-Based CF

Let r_{ui} be the rating of user u for item i, and r_i the vector of ratings of item i over all users, with missing ratings set to zero. Item similarity is the cosine similarity:

$$\text{sim}(i, j) = (r_i \cdot r_j) / (\|r_i\| \, \|r_j\|) \quad (1)$$

We also test adjusted cosine, where each rating is first reduced by the user's mean rating μ_u, so that users who rate everything high do not distort the similarity:

$$\text{sim}(i, j) = \Sigma_u (r_{ui} - \mu_u)(r_{uj} - \mu_u) / (\text{product of the two norms}) \quad (2)$$

For each item, only the k most similar items with positive similarity are kept. The rating of user u for item i is a weighted average over the kept neighbours that u has rated, with a shrinkage term λ that pulls the prediction towards the item's mean rating μ_i when only a few neighbours are available:

$$\hat{r}_{ui} = (\Sigma_j \text{sim}(i, j) \, r_{uj} + \lambda \, \mu_i) / (\Sigma_j \text{sim}(i, j) + \lambda) \quad (3)$$

B. User-Based CF

User similarity is the cosine similarity between mean-centred rating vectors, which is similar to a Pearson correlation, except that missing ratings are treated as zero after centring. The prediction uses the k most similar users who rated item i:

$$\hat{r}_{ui} = \mu_u + \Sigma_v \text{sim}(u, v)(r_{vi} - \mu_v) / (\Sigma_v |\text{sim}(u, v)| + \lambda) \quad (4)$$

C. Top-N Ranking by Similarity Sum

Following [7], for top-N recommendation we score an unrated item by adding its similarities to the items the user liked (rated 4 or 5 in training). Similarity is computed on the binary rated/not-rated matrix:

$$\text{score}(u, i) = \Sigma_{j \in L_\{u\}} \text{sim}(i, j) \quad (5)$$

The user-based version adds the similarities of u to users who liked item i. This ranking rule is compared with ranking by the predicted rating from (3) and (4).

D. Baselines and Matrix Factorization

The baselines are the global mean rating, the item mean rating, and item popularity (number of training ratings) for ranking. The matrix factorization model is the biased form [3]:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u \cdot q_i \quad (6)$$

It uses 20 latent factors and is trained by stochastic gradient descent (25 epochs, learning rate 0.006, regularization 0.04). These values were set once and not tuned.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

V. EXPERIMENTAL SETUP

All methods were implemented in Python with NumPy. Predicted ratings are limited to the range 1–5. RMSE and MAE are computed on all test ratings. For top-10 recommendation, candidates are all movies not rated by the user in the training set, and Precision@10 and Recall@10 are averaged over users who have at least one relevant test item. Results are the mean and the (population) standard deviation over the five splits.

Hyperparameters were selected on a validation set made of 10% of the training ratings of split u1. For rating prediction we searched $k \in \{30, 50, 100, 200\}$ and $\lambda \in \{0, 5, 10, 25\}$ and chose the lowest RMSE; this gave $k = 200$, $\lambda = 0$ for user-based CF and adjusted-cosine item-based CF, and $k = 50$, $\lambda = 5$ for raw-cosine item-based CF. For similarity-sum ranking we searched $k \in \{10, 20, 30, 50, 100, 200\}$ and chose the best Precision@10: $k = 100$ (item-based) and $k = 50$ (user-based). The same settings were reused for all five splits.

VI. EVALUATION METRICS

Rating prediction is measured with the root mean squared error (RMSE), which is the square root of the average squared difference between predicted and actual ratings, and the mean absolute error (MAE), which is the average absolute difference. Lower values are better. Top-10 recommendation is measured with Precision@10, the share of the 10 recommended movies that the user later rated 4 or 5, and Recall@10, the share of the user's relevant movies that appear in the top 10. Higher values are better.

VII. RESEARCH GAP AND CONTRIBUTION

Recent recommender-system research increasingly uses deep learning and large language models [4], [10], [11]. These models can use richer information, but studies often change the dataset, the evaluation task and the tuning procedure at the same time, so the performance of the basic methods is hard to compare. For neighbourhood-based CF, two points are often left unclear: how the neighbourhood size and the shrinkage value were chosen, and how the ranked list was produced from the scores.

The contribution of this study is a controlled comparison of classical methods on one public dataset, with the splits, similarity measures, tuning protocol and metrics stated explicitly, and with rating prediction and top-10 recommendation evaluated separately. It also reports where our results differ from the common claim that item-based CF is more accurate than user-based CF.

VIII. RESULTS AND DISCUSSION

A. Rating Prediction

Table III shows the rating prediction results. Matrix factorization has the lowest error (RMSE 0.926), about 1.7% lower than user-based CF (0.942). User-based CF and item-based CF with adjusted cosine are very close (0.942 and 0.947); the gap is smaller than the fold-to-fold standard deviation of the item-based result (0.011), so we cannot say that one is better. Raw-cosine item-based CF (0.955) is slightly worse. All CF methods clearly beat the item-mean baseline (1.025).

TABLE III: RATING PREDICTION ON MOVIELENS 100K (5 SPLITS)

Method	RMSE	MAE
Global mean	1.126 ± 0.016	0.945
Item mean	1.025 ± 0.006	0.817
User-based CF (k = 200)	0.942 ± 0.006	0.736
Item-based CF, cosine (k = 50, λ = 5)	0.955 ± 0.006	0.756
Item-based CF, adj. cosine (k = 200)	0.947 ± 0.011	0.738
Biased MF (20 factors)	**0.926 ± 0.007**	**0.730**

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

If the neighbourhood size is too small, the results are poor. With adjusted-cosine item-based CF, RMSE falls from 1.057 at $k = 5$ to 0.947 at $k = 200$ (Fig. 1). The best value in our grid is at the largest k we tried, so a larger k might help further.

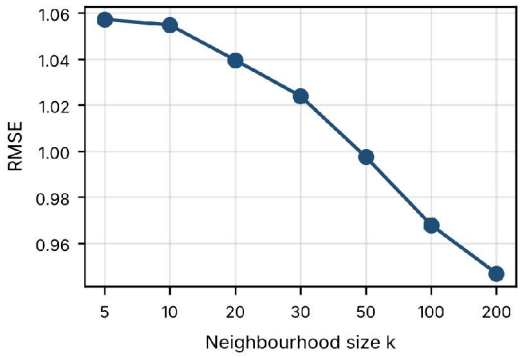

Fig. 1. RMSE of item-based CF (adjusted cosine, $\lambda = 0$) for different neighbourhood sizes k.

B. Top-10 Recommendation

Table IV shows the top-10 results. Ranking by similarity sum works well: item-based CF reaches a Precision@10 of 0.267 and user-based CF 0.269, about 70% higher than the popularity baseline (0.157). The two are practically the same given the standard deviation of about 0.05 across splits. Fig. 2 shows the same comparison as a chart.

The picture changes if the same neighbourhood models are ranked by their predicted rating. Precision@10 then falls to 0.010 for item-based CF and 0.001 for user-based CF, far below the popularity baseline. For comparison, using plain raw-cosine item-based CF without shrinkage ($\lambda = 0$, $k = 30$) gave an RMSE of 0.999 but a Precision@10 of only 0.007. A likely reason is that items with very few neighbours get extreme predicted scores, so rarely rated movies fill the top of the list. The MF model, trained to reduce rating error, also ranks poorly (0.090) compared with popularity.

TABLE IV: TOP-10 RECOMMENDATION (RELEVANT = RATING ≥ 4)

Method	Precision@10	R@10
Popularity	0.157 ± 0.033	0.136
Item-based CF, similarity sum	0.267 ± 0.046	0.237
User-based CF, similarity sum	**0.269 ± 0.048**	**0.247**
Item-based CF (adj. cosine), predicted rating	0.010 ± 0.002	0.005
User-based CF, predicted rating	0.001 ± 0.001	0.001
Biased MF, predicted rating	0.090 ± 0.020	0.052

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

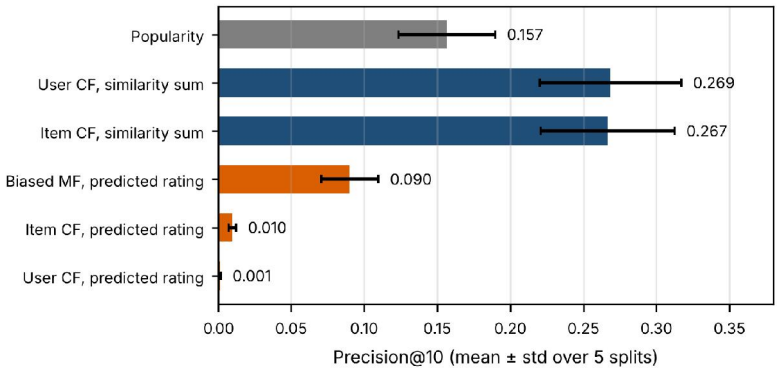

Fig. 2. Precision@10 of the compared methods. Bars show the mean over five splits and the lines show one standard deviation. Dark blue: ranking by similarity sum; orange: ranking by predicted rating; grey: popularity.

C. Discussion

Sarwar *et al.* [1] found item-based CF better than user-based CF. We did not see this. A possible explanation is the shape of the data: MovieLens 100K has fewer users (943) than items (1,682), while the main advantage of item-based CF is scalability when users greatly outnumber items. We have not tested this explanation.

IX. LIMITATIONS

The results should be read with care. (i) Only one small dataset was used. (ii) No statistical significance test was run; we only compare means with the standard deviation over five splits. (iii) The hyperparameters were chosen once, on the training data of split u1, and reused; because the u1 training set contains the test ratings of the other splits, a stricter protocol would tune inside each split. (iv) The best k was at the edge of the search grid. (v) The MF model was not tuned. (vi) Only offline metrics were measured, and cold-start users and items, diversity and fairness were not studied.

X. FUTURE SCOPE

Future work can repeat the comparison on larger datasets such as MovieLens 1M and on data from other domains such as products, music and courses. Each split can be tuned separately and the differences checked with a significance test. The classical methods can also be compared with hybrid methods such as [5] and [6], and with language-model-based recommenders [4], [10], and the study can be extended to cold-start users and to measures of diversity and fairness.

XI. CONCLUSION

We compared user-based CF, item-based CF and matrix factorization on MovieLens 100K. Matrix factorization gave the best rating accuracy (RMSE 0.926), while the two neighbourhood methods were close to each other (0.942 and 0.947) and did not show a clear winner. For top-10 recommendation, the ranking rule mattered more than the choice between users and items: ranking by a sum of similarities gave a Precision@10 of about 0.27 and beat the popularity baseline (0.157), while ranking by predicted rating did not. These results give a transparent baseline for later work with hybrid and language-model-based recommenders.

XII. REPRODUCIBILITY AND RESEARCH INTEGRITY

All reported numbers were produced by running the described pipeline on the MovieLens 100K dataset, and no value was copied from another paper. The splits, similarity measures, neighbourhood sizes, shrinkage values, training settings and metrics are stated, so the experiment can be repeated. The figures were drawn from the measured results. The cited works and the dataset are credited in the references. The use of an AI assistant is stated in the acknowledgment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, October 2026
Impact Factor: 8.2

ACKNOWLEDGMENT

An AI assistant (Claude, by Anthropic) was used to help write the experiment code and to draft and edit the text. The authors are responsible for the content, and the reported numbers come from the code that was run on the MovieLens 100K data.

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW '01)*, 2001, pp. 285–295.

[2] G. Linden, B. Smith and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[3] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[4] Z. Zhao *et al.*, "Recommender systems in the era of large language models (LLMs)," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6889–6907, 2024, doi: 10.1109/TKDE.2024.3392335.

[5] C. Huda, Y. Heryadi, L. Lukas and W. Budiharto, "Smart tourism recommender system modeling based on hybrid technique and content boosted collaborative filtering," *IEEE Access*, vol. 12, pp. 131794–131808, 2024, doi: 10.1109/ACCESS.2024.3450882.

[6] S. Lv, J. Wang, F. Deng and P. Yan, "A hybrid recommendation algorithm based on user nearest neighbor model," *Sci. Rep.*, vol. 14, Art. no. 17119, 2024, doi: 10.1038/s41598-024-66393-3.

[7] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[9] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Art. no. 19, 2016.

[10] S. Shehmir and R. Kashef, "LLM4Rec: A comprehensive survey on the integration of large language models in recommender systems—Approaches, applications and challenges," *Future Internet*, vol. 17, no. 6, Art. no. 252, 2025, doi: 10.3390/fi17060252.

[11] P. R. Vinayababu and P. S. Krishna Raju, "Revolutionizing recommendations a survey: A comprehensive exploration of modern recommender systems," *IAES Int. J. Artif. Intell.*, vol. 14, no. 4, pp. 2579–2589, 2025, doi: 10.11591/ijai.v14.i4.pp2579-2589.