Book Recommendation System Using Similarity-Based Collaborative Filtering Approach

Uswatun Hasanah[1], Arief Arfriandi[2], Rahmadani Nur Permanawati[3], Citra Nur Aini[4], Arif Adi Wibowo[5], Berliandi Kurniawan Prasetya[6], Ani Ariani[7]
[1,2]Department of Computer Engineering, Faculty of Engineering, Universitas Negeri Semarang
[3]Department of Management, Faculty of Economics and Business, Universitas Negeri Semarang
[4,5,6]Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Semarang
[7]Internal Audit Unit, Universitas Negeri Semarang
Semarang, Indonesia
uswatun_hasanah@mail.unnes.ac.id

Submitted: October 30, 2025; accepted: May 5, 2026

Abstract

This study implements and compares the performance of three primary approaches to book recommendation systems: rank-based methods, similarity-based collaborative filtering (both user-user and item-item), and matrix factorization-based collaborative filtering. The dataset comprises 433,671 user ratings from 78,805 users on 185,973 books, enriched with book metadata such as title, author, publication year, and publisher. The recommendation systems were developed using the Surprise library, with hyperparameter optimization performed via grid search cross-validation. Model performance was evaluated using precision@k, recall@k, and F1-score metrics, as well as RMSE for prediction accuracy. Results indicate that the user-user similarity-based collaborative filtering model achieved the best performance in terms of relevance, attaining an F1-score of 0.86. This model effectively identifies users with similar preferences and recommends books based on collective behavior patterns. Meanwhile, the matrix factorization approach yielded the lowest RMSE value of 1.50, highlighting its strength in capturing latent factors that influence user preferences. The item-item similarity model also showed reasonable performance but did not surpass the other approaches, possibly due to homogeneity in item rating patterns across users. Overall, the study confirms that user-user similarity is highly effective for datasets exhibiting consistent user behavior, while matrix factorization excels in minimizing prediction error by leveraging latent feature structures. These findings offer valuable insights for developing adaptive recommendation systems in book-centric literacy platforms and content-driven e-commerce applications.

Keywords: *collaborative filtering, item-based, user-based, book recommendation, recommendation systems.*

1. Introduction

Leading global e-commerce platforms employ personalized recommendation systems as a strategic means to sustain their competitive advantage. A study by [1] designed and evaluated multiple recommendation models, assessing their effectiveness in delivering both accurate and diverse suggestions. Book recommendation algorithms are now widely used on popular reading platforms and online bookstores. As the number of available books continues to rise, research and innovation in this field are becoming increasingly essential. Accurate and effective book recommendations provide significant benefits. They help readers discover new authors and genres they might not have considered, which in turn boosts reader engagement and satisfaction [2]. For bookstores and publishers, well-designed recommendation systems can increase sales and customer loyalty [3]. One particularly powerful approach in this area is collaborative filtering [4], which learns from readers' past behavior and preferences to continually refine its suggestions, ensuring that recommendations remain relevant and engaging.

Collaborative filtering is generally categorized into two primary methodologies: item-based and user-based approaches [5]. The item-based technique evaluates the similarities among items to predict a user's potential preferences. In contrast, the user-based method identifies individuals with analogous consumption behaviors and recommends content favored by these similar users [6].

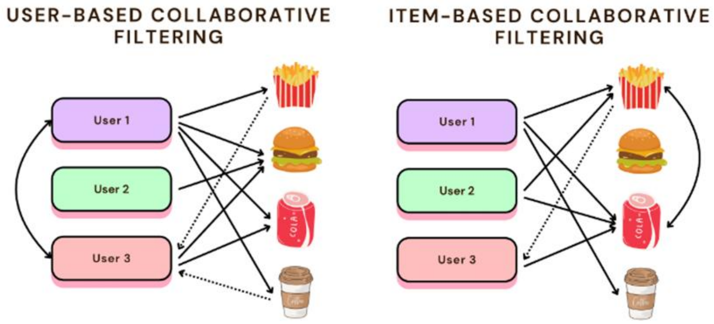

Fig. 1 The distinct mechanisms employed by user-based and item-based collaborative filtering

Fig. 1 illustrates the conceptual differences between user-based collaborative filtering and item-based collaborative filtering mechanisms in recommendation systems. On the left side, the user-based collaborative filtering model identifies and leverages the preferences of users who exhibit similar behavior. For instance, if User 1 and User 2 have demonstrated comparable preferences in the past, the system assumes they will have similar tastes in the future. Therefore, items liked by User 1 (e.g., burger, soda, fries) are recommended to User 2, even if User 2 has not interacted with those specific items before. The key principle is that recommendations are derived by analyzing user-user similarities based on historical interactions. On the right side, the item-based collaborative filtering model operates differently by focusing on item-item similarities. In this approach, the system identifies relationships among items based on co-occurrence patterns across multiple users. For example, suppose the hamburger and fries are frequently liked or purchased together. In that case, the system will recommend fries to users who have shown interest in the hamburger, regardless of other users' preferences. Thus, recommendations are generated by examining the correlation between items rather than between users. While user-based filtering emphasizes shared preferences among users to suggest new items, item-based filtering prioritizes the similarity among items themselves to offer more context-aware and consistent recommendations, particularly effective in large-scale systems with sparse user data. The rapid expansion of recommendation systems has led to the use of increasingly large datasets. However, because most users interact with only a limited number of available items, the resulting user-item matrices are often sparse [7]. This lack of sufficient interactions makes it difficult for algorithms to learn accurate user preferences, particularly during cold-start situations when little data is available.

Collaborative filtering is a popular method in recommendation system design that relies solely on user-item interaction data, without requiring any explicit information about the users or the items themselves [8]. This interaction data may include ratings provided by users on books, movies, or other products, likes on social media platforms, purchases made on e-commerce websites, or reading history on blogs and news platforms. The key principle behind collaborative filtering is that users with similar preferences in the past are likely to share similar interests in the future [9]. Similarly, items that receive similar feedback from users are likely to appeal to similar audiences.

Hikmatyar et al. [10] proposed a collaborative filtering model based on Cosine Similarity to recommend books using the Book-Crossing dataset. The study adopted a memory-based approach and compared prediction accuracy using MAE and RMSE. The authors showed that Cosine Similarity is effective in determining user or item proximity, particularly in large-scale datasets. However, they also noted the limitations in handling cold-start users and sparse rating matrices, suggesting potential enhancements through hybrid methods or the inclusion of auxiliary data.

Ahmed et al. [8] conducted a study using the Book-Crossing dataset to implement collaborative filtering-based recommendation systems. The study explored both user-based and item-based collaborative filtering methods and evaluated their performance using standard metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The authors found that the item-based approach yielded more accurate results, as it mitigated sparsity and better captured item correlations. The work emphasized the importance of similarity measures and quantitative evaluation in assessing the recommendation quality of collaborative models.

Patro et al. [11] proposed a hybrid recommendation approach called Hybrid Action-Related K-Nearest Neighbor (HAR-KNN), which combines user behavior modeling with KNN-based filtering. Unlike traditional similarity-based methods, HAR-KNN incorporates "action weightage" by assigning different weights to user interactions such as clicks, purchases, and views. This behavioral enrichment enables the system to deliver more personalized and context-aware recommendations. The hybrid approach demonstrated improvements in relevance and reduced bias, making it suitable for real-world applications with diverse interaction types.

Singh et al. [12] conducted a hybrid recommendation system that integrates content-based filtering using TF-IDF and cosine similarity with collaborative filtering and SVD to overcome traditional limitations. The model is further enhanced with machine learning algorithms such as decision trees, random forests, and support vector regression (SVR), alongside boosting techniques like CatBoost and XGBoost. Experimental results on the MovieLens 1M dataset show that the hybrid model achieves superior accuracy, with SVR emerging as the best-performing method, demonstrating the potential of combining collaborative and content-based approaches with advanced machine learning.

There are two main types of collaborative filtering: similarity (neighborhood)-based and model-based. In this study, we implement a similarity-based collaborative filtering approach [6] using cosine similarity to compute how closely aligned the preferences of different users are. Cosine similarity is a widely used metric for comparing sparse, high-dimensional vectors such as rating profiles, as it captures the direction of user preferences regardless of magnitude [13]. This similarity is then used to identify users with behavior patterns most similar to the target user.

To operationalize this approach, we use the K-Nearest Neighbors (KNN) algorithm [14] to find the most similar users (or "neighbors") based on their past ratings or interactions. The preferences of these neighbors form the basis for generating personalized recommendations for the target user, suggesting items the user has not yet rated but which were highly rated by similar users. This method allows the system to make intelligent predictions even when metadata is missing or sparse. The goal of this model is to assess the effectiveness of user-user and item-item collaborative filtering, leveraging KNN and cosine similarity to build adaptive recommendation systems for applications such as online bookstores, educational platforms, or e-commerce environments.

2. Method

This study is conducted through computational experiments using Python-based data analysis tools on local computing resources.

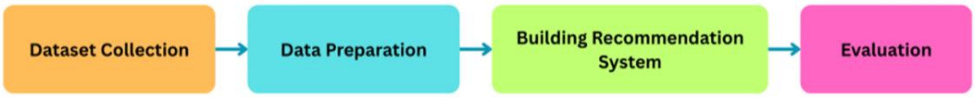

Fig. 2 Research flow

The research flow in this study, which is shown in Fig. 2, can be described as follows:

1. **Dataset Collection**

   This study uses a dataset downloaded from https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset. The dataset comprises three essential components: (1) user data containing anonymized user IDs, (2) book metadata including title, author, year of publication, publisher, and book cover images, and (3) explicit user ratings on a 0–10 scale. This comprehensive structure facilitates diverse experimental setups across various algorithmic recommendation paradigms.

   The rating data contains 1,149,780 observations spread across seven columns, with the most relevant being user_id, book_id, and rating. The user_id and rating fields are stored in numeric format, which is directly compatible with most modeling frameworks. Meanwhile, the book_id column is stored as an object type, necessitating its conversion to string format to ensure seamless downstream processing, particularly in collaborative filtering algorithms where this identifier serves as a primary reference.

2. **Data Preparation**

   To enhance computational efficiency and ensure the robustness of the developed recommendation models, a rigorous data preprocessing phase was carried out. Given the scale of the dataset, the raw interaction matrix exhibits high sparsity, which could potentially degrade model performance and increase computational cost. To mitigate this, a filtering strategy was applied to the user–item interaction matrix. This strategy involved setting minimum threshold values for both users and items: users who provided fewer than a specified number of ratings, and books that received too few ratings, were excluded from the final dataset. This approach ensured that only users with meaningful engagement and books with sufficient feedback were retained, leading to a denser and more informative interaction matrix.

The goal of this step was to reduce the influence of cold-start users and underrepresented items, which often introduce noise and unpredictability into collaborative filtering models [15]. By enforcing interaction thresholds, the preprocessing step not only stabilized the similarity computations (especially in KNN-based models using cosine similarity) but also contributed to faster training convergence and more reliable recommendation quality across the evaluated systems.

3. **Building Recommendation System**

After the data preprocessing stage, four distinct recommendation models were developed to explore various algorithmic approaches for generating personalized book suggestions. These models include User-Based and Item-Based Collaborative Filtering Recommendation Systems. These models rely solely on user–item interaction data, such as rating history, without requiring explicit information about the users or content metadata. In the user-based approach, recommendations are generated by identifying users who exhibit similar rating behaviors (using cosine similarity) and predicting items favored by those similar users. Conversely, the item-based model recommends books based on the similarity between items, where similarity is computed from user rating patterns. Books that have been rated similarly by users are considered related and thus likely to be co-preferred.

4. **Evaluation**

In evaluating the performance of a recommendation system, it is essential to define key terminologies that underpin the assessment of model effectiveness, particularly in distinguishing between relevant and non-relevant recommendations [16]. In the context of this study, a relevant item refers to a book whose actual user rating exceeds a predefined threshold; in this case, a rating above 7. Conversely, items rated below this threshold are considered non-relevant. A recommended item is one for which the system's predicted rating surpasses the same threshold. If the predicted rating falls below 7, the item is not included in the recommendation list for the user. Two critical types of errors can arise in the recommendation process. A false negative (FN) occurs when an item that is genuinely relevant is not recommended to the user. This represents a missed opportunity, as the user is likely to appreciate the item, and its exclusion may lead to user dissatisfaction or lost sales. On the other hand, a false positive (FP) describes a scenario in which an item is recommended but ultimately deemed non-relevant by the user. This results in inefficient utilization of recommendation bandwidth and can diminish user trust in the system.

To quantify the model's ability to generate accurate and useful recommendations, two widely adopted performance metrics are employed: precision and recall. Recall is defined as the proportion of actually relevant items that are successfully recommended to the user. For instance, if 6 out of 10 relevant items are recommended, the recall value is 0.60. In contrast, precision denotes the proportion of recommended items that are truly relevant. If 6 out of 10 recommended items are found to be relevant, the precision is likewise 0.60. Both metrics are critical for assessing classification-based systems, particularly in the domain of personalized recommendations.

3. Result and Discussion

3.1. Exploring Dataset

This study utilizes three distinct datasets: books, ratings, and users. All datasets are provided in CSV format and imported via Google Colab for analysis. The book dataset contains eight attributes, namely: ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, and Image-URL-L. The ratings dataset includes three fields: User-ID, ISBN, and Book-Rating. Meanwhile, the user dataset comprises the following variables: User-ID, Location, and Age. However, user demographic data was not utilized in this study, as it was deemed irrelevant to the scope of the case study. Only the User-ID from the ratings dataset was retained for the purpose of building the recommendation models. Hereafter, several features were considered non-essential for the development of the recommendation system and were therefore removed, including Image-URL-S, Image-URL-M, and Image-URL-L. The ISBN, which serves as a unique identifier for each book, was used as the primary key to merge the ratings and book datasets into a single, comprehensive dataset. For consistency and ease of reference throughout the study, we renamed the variables User-ID, ISBN, and Book-Rating to user_id, book_id, and rating, respectively. The fixed dataset fragments can be seen in Fig. 3.

	user_id	book_id	rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
1	276726	0155061224	5	Rites of Passage	Judith Rae	2001	Heinle
2	276727	0446520802	0	The Notebook	Nicholas Sparks	1996	Warner Books
3	276729	052165615X	3	Help!: Level 1	Philip Prowse	1999	Cambridge University Press
4	276729	0521795028	6	The Amsterdam Connection : Level 4 (Cambridge ...	Sue Leather	2001	Cambridge University Press

Fig. 3 The fixed dataset fragments

The ratings dataset comprises 1,149,780 observations. Fig. 4 depicts the distribution of ratings from the dataset.

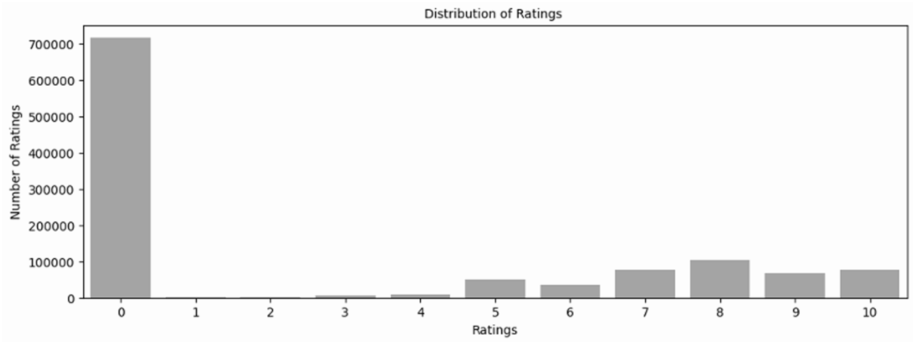

Fig 4. Distribution of ratings

Based on the histogram analysis, ratings with a value of 0 represent the highest frequency, accounting for approximately 700,000 entries and constituting most of the dataset. Given that the valid rating scale ranges from 1 to 10, a rating of 0 can reasonably be interpreted as a missing or invalid value. To eliminate potential bias toward this single class and to improve data quality, all entries with a rating of 0 were excluded from the dataset. After this filtering step, the most frequent valid ratings were 8 and 10, with approximately 100,000 and 80,000 observations, respectively. After removing all entries with a rating of 0, the dataset was reduced to a total of 433,671 records. The distribution of the remaining ratings is illustrated in Fig. 5.

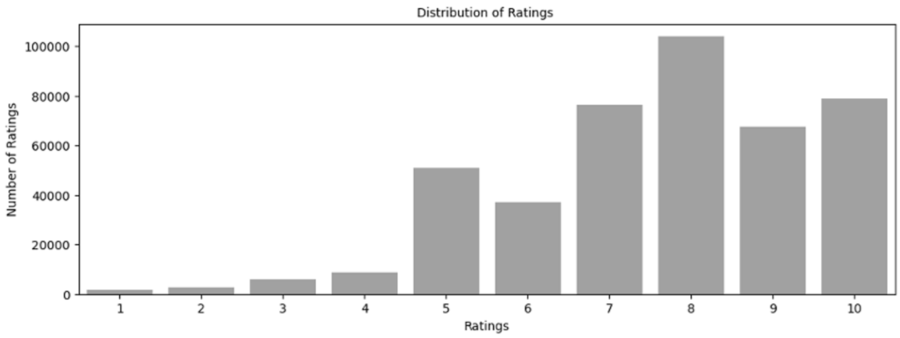

Fig. 5 Distribution of ratings after removing invalid values

The dataset contains a total of 185,973 unique books. Given the presence of 77,805 unique users and 185,973 books, the maximum possible number of user-book interactions would be 14,469,629,265. However, only 433,671 actual ratings are available, indicating that the user-item matrix is extremely sparse. It means that most users have not rated the majority of books. This sparsity presents a suitable opportunity to develop a recommendation system aimed at suggesting books that users have not previously interacted with. The book with book_id: 0316666343 has received the highest number of user interactions, totaling 707 ratings. However, considering that there are 77,805 unique users in the dataset, this still leaves 77,098 potential interactions unobserved for this particular book. A recommendation system can therefore be employed to identify which of these remaining users are most likely to engage with the book. Furthermore, it is essential to analyze the distribution of ratings among the 707 interactions to determine whether the users predominantly liked or disliked the book. This analysis provides insight into the book's overall reception and helps inform the recommendation strategy.

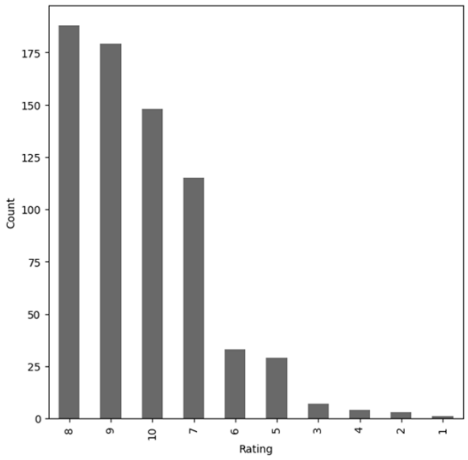

Fig. 6 The ratings

Based on Fig. 6, the rating distribution for this book reveals that most users assigned a rating of 8, followed by ratings of 9, 10, and 7. In contrast, the number of low ratings, specifically 1 through 4, is significantly smaller. This pattern indicates that the book is generally well-received and favored by most users, suggesting high user satisfaction. The user with user_id: 11676 has the highest activity level, having interacted with 8,524 books. However, given that the dataset contains 185,973 unique books, 177,449 potential book interactions remain that have not been recorded for this user. This gap underscores the relevance of a recommendation system in identifying additional books that may align with the user's preferences.

Although the dataset consists of 433,671 observations, its size and sparsity make it computationally inefficient for direct model training. A substantial number of users have rated only a few books, and very few users have rated many books. We applied a filtering strategy based on logical assumptions to address this issue to retain only the most informative data. Specifically, we included only users who have provided at least 50 ratings and books with a minimum of 10 ratings. This approach aligns with typical user behavior in online retail environments, where consumers are more inclined to consider products with a reasonable volume of reviews.

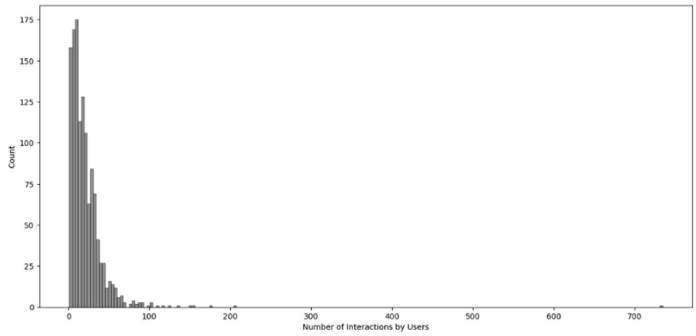

Fig. 7 The distribution of the number of interactions by users

The histogram presented in Fig. 7 illustrates the distribution of the number of interactions (ratings) per user within the dataset. The plot reveals a highly right-skewed distribution, indicative of substantial disparity in user engagement levels. A majority of users have provided relatively few ratings, with the frequency of users rapidly declining as the number of interactions increases. This long-tailed behavior is a characteristic phenomenon in user-generated datasets, especially in recommendation systems, where a small fraction of highly active users contribute disproportionately to the overall interaction volume.

The skewness observed has important implications for model development. Users with minimal interaction history (cold-start users) present a challenge in learning reliable preference patterns, potentially impacting the accuracy of personalized recommendations. Conversely, the presence of a few users with extensive activity can dominate the similarity computations in collaborative filtering models, introducing bias toward frequently rated items.

3.2. Building Models

To prepare the data for collaborative filtering using the Surprise library, the dataset is first converted into a format compatible with Surprise's Dataset module. The transformed dataset is then partitioned into training and testing subsets, with 30% of the data allocated for testing and the remaining 70% used for training. This split allows for effective model evaluation while ensuring sufficient data for learning user-item interaction patterns.

3.2.1. User-Based Collaborative Filtering Recommendation System

In this section, the first baseline similarity-based recommendation system using cosine similarity and KNN was built, and the result is shown in Table 2. The baseline similarity-based collaborative filtering model achieved a Root Mean Square Error (RMSE) of approximately 1.84 on the test dataset. The model also demonstrated a balanced performance in terms of relevance, with a precision of 0.816 and a recall of 0.812. This implies that 81.6% of the books recommended by the system were relevant (precision), and 81.2% of all relevant books were successfully recommended (recall). The resulting F1 score of 0.814 reflects a strong harmonic mean between precision and recall, indicating that the model was effective in both recommending relevant books and capturing a significant portion of all relevant titles. To further enhance the model's performance, hyperparameter tuning using techniques such as GridSearchCV can be employed in subsequent iterations.

Table 2. Baseline model of a user-based collaborative filtering approach

Metric	Value
RMSE	1.8455
Precision	0.816
Recall	0.812
F1 Score	0.814

In the next stage, we perform hyperparameter tuning for the KNNBasic algorithm to optimize its performance. The algorithm provides several key hyperparameters that can be adjusted. The parameter k defines the maximum number of nearest neighbors considered during rating prediction, with a default value of 40. The min_k parameter specifies the minimum number of neighbors required to generate a valid prediction; if the number of neighbors falls below this threshold, the system defaults to predicting the global mean rating. By default, min_k is set to 1. Additionally, sim_options is a dictionary used to configure the similarity computation. The Surprise library supports four types of similarity measures: cosine, MSD (mean squared difference, used by default), Pearson, and Pearson_baseline. These hyperparameters allow fine-grained control over how user or item similarities are computed and how neighbor contributions are aggregated during prediction.

Next, we build the final model by using tuned values of the hyperparameters, which we received by using grid search cross-validation, as shown in Table 3.

Table 3. Hyperparameter tuning of the user-based collaborative filtering approach

Metric	Value
RMSE	1.6866
Precision	0.834
Recall	0.891
F1 Score	0.862

Following the hyperparameter tuning process, the Root Mean Square Error (RMSE) on the test set decreased from 1.84 to 1.68, indicating a notable improvement in predictive accuracy. Additionally, the F1 score increased from 0.814 in the baseline model to 0.862 in the tuned model. This enhancement demonstrates that the optimized model is more effective in recommending relevant books while also capturing a greater proportion of relevant items. Thus, it can be concluded that hyperparameter tuning has significantly contributed to improving the overall performance of the recommendation system.

3.2.2. Item-based Collaborative Filtering Recommendation System

In this method, recommendations are generated by identifying items similar to those previously interacted with by the user, based on their rating patterns across the user base. The similarity metric employed is cosine similarity, which effectively captures the angular distance between item vectors in the user-item interaction matrix. Initial evaluation, as shown in Table 4, yields a Root Mean Square Error (RMSE) of 1.6210, indicating the average magnitude of prediction error. In terms of classification-based performance metrics, the model achieves a precision of 0.802, a recall of 0.800, and an F1-score of 0.801 on the test set. These results suggest a balanced performance in both identifying relevant items and avoiding false recommendations.

Table 4. Baseline model of the item-based collaborative filtering approach

Metric	Value
RMSE	1.6210
Precision	0.802
Recall	0.80
F1 Score	0.801

To further optimize the model's performance, hyperparameter tuning is proposed using GridSearchCV, which systematically explores combinations of key algorithm parameters. This process aims to reduce prediction error and improve relevance metrics, thereby enhancing the overall recommendation quality of the item-item similarity-based model.

Subsequent to the baseline evaluation, hyperparameter tuning was performed using grid search cross-validation to identify the optimal configuration for the item-item similarity-based collaborative filtering model. This technique systematically explored a predefined set of hyperparameter values, aiming to minimize prediction error and enhance recommendation accuracy. Upon training the final model with the optimal hyperparameter settings derived from the grid search, the performance metrics on the test set improved moderately. The model, as shown in Table 5, achieved a Root Mean Square Error (RMSE) of 1.5882, representing a reduction from the baseline RMSE of 1.6210. In terms of relevance, the tuned model yielded a precision of 0.818, a recall of 0.836, and an F1-score of 0.827, all slightly outperforming the baseline configuration. Fig. 8 shows the comparison of all the methods used in this study.

Table 5. Hyperparameter tuning of the item-based collaborative filtering approach

Metric	Value
RMSE	1.5882
Precision	0.818
Recall	0.836
F1 Score	0.827

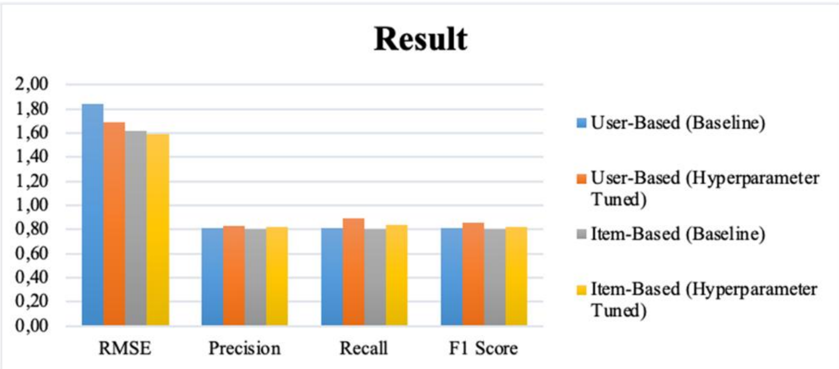

Fig. 8 Experiment’s Result

Fig. 8 presents the comparative performance results between user-based and item-based collaborative filtering models, both in their baseline and hyperparameter-tuned configurations. Four key evaluation metrics were employed: Root Mean Square Error (RMSE), Precision, Recall, and F1 Score, to assess prediction accuracy and recommendation quality.

Overall, Fig. 8 indicates that hyperparameter tuning significantly improves model performance, particularly by reducing RMSE values across both user-based and item-based approaches. The User-Based (Hyperparameter Tuned) model shows a noticeable decrease in RMSE compared to the baseline, implying better prediction accuracy in estimating user ratings. Similarly, the Item-Based (Hyperparameter Tuned) model achieves the lowest RMSE, suggesting enhanced stability and consistency in capturing item–item relationships. In terms of Precision, Recall, and F1 Score, both tuned models demonstrate marginal yet consistent improvements over their baseline counterparts. The User-Based (Tuned) model slightly outperforms others in Recall and F1 Score, indicating better sensitivity to relevant recommendations. Meanwhile, the Item-Based (Tuned) model maintains balanced results across all three classification metrics.

In summary, the findings show that hyperparameter optimization contributes positively to both user-based and item-based collaborative filtering, with notable gains in error reduction and modest improvements in precision-oriented measures. These results validate the importance of parameter tuning in enhancing recommendation accuracy and overall model robustness.

4. Conclusion

This study demonstrates that the user-user similarity-based collaborative filtering algorithm is the most effective approach for generating relevant and diverse book recommendations, achieving an F1-score of approximately 0.86. This method successfully identifies users with similar preference patterns and leverages that information to provide personalized suggestions. Although item-item similarity-based collaborative filtering also delivers promising results, it falls slightly behind in comparison. Given these findings, future research is encouraged to explore hybrid recommendation systems that combine similarity-based models for improved relevance and accuracy. Addressing the cold-start problem, particularly for new users or items lacking sufficient interaction data, remains crucial, potentially through content-based or demographic-based solutions. Furthermore, deploying and evaluating models on large-scale real-world e-commerce datasets is recommended to assess scalability and robustness. Enhancing performance through advanced hyperparameter tuning methods such as Bayesian Optimization or AutoML, along with enriching the dataset with additional contextual features like user reviews, timestamps, or explicit preferences, may further improve the effectiveness of recommendation systems.

Reference

[1] J. Kim, I. Choi, and Q. Li, "Customer Satisfaction of Recommender System: Examining Accuracy and Diversity in Several Types of Recommendation Approaches," *Sustainability*, vol. 13, no. 11, p. 6165, May 2021, doi: 10.3390/su13116165.

[2] O. Dogan, E. Yalcin, and O. Areta Hiziroglu, "Digitalization for enhancing reading habits: the improved hybrid book recommendation system with genre-oriented profiles," *LM*, vol. 45, no. 8/9, pp. 489–505, Nov. 2024, doi: 10.1108/LM-03-2024-0030.

[3] B. Dhanush Kumar, D. Nirmala, A. Akash, R. Sridhar, and S. Buvanesh Kumar, "Online Bookstore and Management System," *Int. J. Sci. R. Tech.*, vol. 2, no. 4, pp. 250–255, 2025.

[4] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowl Inf Syst*, vol. 64, no. 1, pp. 35–74, Jan. 2022, doi: 10.1007/s10115-021-01628-7.

[5] M. G. Silva, S. C. Madeira, and R. Henriques, "A Comprehensive Survey on Biclustering-based Collaborative Filtering," *ACM Comput. Surv.*, vol. 56, no. 12, pp. 1–32, Dec. 2024, doi: 10.1145/3674723.

[6] Peng Yu, "Collaborative filtering recommendation algorithm based on both user and item," in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, Harbin, China: IEEE, Dec. 2015, pp. 239–243. doi: 10.1109/ICCSNT.2015.7490744.

[7] Z. Xia, A. Sun, J. Xu, Y. Peng, R. Ma, and M. Cheng, "Contemporary Recommendation Systems on Big Data and Their Applications: A Survey," *IEEE Access*, vol. 12, pp. 196914–196928, 2024, doi: 10.1109/ACCESS.2024.3517492.

[8] E. Ahmed and A. Letta, "Book Recommendation Using Collaborative Filtering Algorithm," *Applied Computational Intelligence and Soft Computing*, vol. 2023, pp. 1–12, Mar. 2023, doi: 10.1155/2023/1514801.

[9] H. I. Abdalla, A. A. Amer, Y. A. Amer, L. Nguyen, and B. Al-Maqaleh, "Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures," *Int J Comput Intell Syst*, vol. 16, no. 1, p. 123, Jul. 2023, doi: 10.1007/s44196-023-00299-2.

[10] M. Hikmatyar and Ruuhwan, "Book Recommendation System Development Using User-Based Collaborative Filtering," *J. Phys.: Conf. Ser.*, vol. 1477, no. 3, p. 032024, Mar. 2020, doi: 10.1088/1742-6596/1477/3/032024.

[11] S. G. K. Patro *et al.*, "A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems," *IEEE Access*, vol. 8, pp. 90978–90991, 2020, doi: 10.1109/ACCESS.2020.2994056.
[12] Kulvinder Singh, S. Dhawan, and N. Bali, "An Ensemble Learning Hybrid Recommendation System Using Content-Based, Collaborative Filtering, Supervised Learning and Boosting Algorithms," *Aut. Control Comp. Sci.*, vol. 58, no. 5, pp. 491–505, Oct. 2024, doi: 10.3103/S0146411624700615.
[13] S. C. Mana and T. Sasipraba, "Research on Cosine Similarity and Pearson Correlation Based Recommendation Models," *J. Phys.: Conf. Ser.*, vol. 1770, no. 1, p. 012014, Mar. 2021, doi: 10.1088/1742-6596/1770/1/012014.
[14] T. Anwar, V. Uma, Md. I. Hussain, and M. Pantula, "Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis," *Multimed Tools Appl*, vol. 81, no. 25, pp. 35693–35711, Oct. 2022, doi: 10.1007/s11042-021-11883-z.
[15] J. Cao, J. Sheng, X. Cong, T. Liu, and B. Wang, "Cross-Domain Recommendation to Cold-Start Users via Variational Information Bottleneck," Mar. 31, 2022, *arXiv*: arXiv:2203.16863. doi: 10.48550/arXiv.2203.16863.
[16] P. Chowdhury and B. B. Sinha, "Evaluating the Effectiveness of Collaborative Filtering Similarity Measures: A Comprehensive Review," *Procedia Computer Science*, vol. 235, pp. 2641–2650, 2024, doi: 10.1016/j.procs.2024.04.249.