Original Article
ISSN (Online): 2350-0530
ISSN (Print): 2394-3629

International Journal of Research - GRANTHAALAYAH
June 2024 12(6), 173–186

SMART MOVIE RECOMMENDER: LEVERAGING COLLABORATIVE FILTERING FOR ENHANCED USER EXPERIENCE

Manya Kamra [1], Yanvi Joshi [1], Pranav Banga [1], Rachna Srivastava [1]

[1] Department of Computer Science & Engineering, Echelon Institute of Technology, Faridabad, India

Received 28 May 2024
Accepted 14 June 2024
Published 30 June 2024

DOI
10.29121/granthaalayah.v12.i6.2024.6103

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

This project presents a movie recommendation system using collaborative filtering, which predicts user preferences based on the behavior of similar users. Unlike content-based approaches that rely on movie attributes, this method uses past user interactions to generate personalized and diverse suggestions. The system, implemented in Python, processes user IDs, movie ratings, item IDs, and timestamps to compute correlations and similarities between users.

Key advantages of the approach include independence from item metadata, scalability across various content types, and the ability to suggest unexpected yet relevant items. Inspired by real-world systems like Netflix and Amazon, this model aims to improve recommendation accuracy and user satisfaction through data-driven personalization.

1. INTRODUCTION

In an era where digital content is rapidly expanding, users are often overwhelmed by the sheer volume of options available. From movies and music to e-commerce and academic literature, navigating vast catalogs of information without assistance can be challenging. Recommendation systems have emerged as a powerful tool to address this issue by providing users with personalized suggestions based on their past behavior and preferences. These systems act as intelligent filters, helping users discover content that aligns with their interests while also uncovering new, relevant items [1].

Among the various domains where recommender systems are employed, the entertainment industry—particularly movie streaming platforms—has witnessed one of the most impactful applications. With platforms like Netflix, Hulu, and

Amazon Prime Video boasting libraries of thousands of titles, recommending relevant content is crucial for retaining user engagement and satisfaction. These systems not only improve user experience but also drive revenue by increasing viewership and subscription durations [2]. Consequently, the development of accurate, scalable, and adaptive movie recommendation models has become a critical area of research and implementation.

1.1. OVERVIEW OF RECOMMENDATION SYSTEMS

Recommendation systems are broadly categorized into three main approaches: content-based filtering, collaborative filtering, and hybrid models. Content-based filtering recommends items similar to those the user has liked in the past based on features such as genre, cast, director, or keywords. While effective, this method has limitations when user history is sparse or when item features are insufficient to capture user preference nuances [3].

Collaborative filtering, on the other hand, leverages user behavior patterns to identify relationships between users and items. It assumes that users who agreed in the past will likely agree in the future. This approach is particularly powerful because it does not require domain-specific knowledge or detailed content metadata. Instead, it relies purely on user-item interaction matrices, making it suitable for large-scale, dynamic systems [4].

Hybrid models aim to combine the strengths of both content-based and collaborative filtering methods. These systems use multiple algorithms in parallel or in sequence to provide more robust and accurate recommendations. While effective, hybrid systems often come with increased complexity and computational costs [5].

1.2. THE IMPORTANCE OF COLLABORATIVE FILTERING

Collaborative filtering has become one of the most widely used techniques for recommendation systems due to its effectiveness in handling diverse and sparse data. There are two main types of collaborative filtering: memory-based and model-based methods. Memory-based filtering includes user-user and item-item similarity computations using techniques such as cosine similarity, Pearson correlation, or adjusted cosine similarity. Model-based filtering uses machine learning techniques like matrix factorization (e.g., Singular Value Decomposition), neural networks, or deep learning to uncover latent factors that influence user preferences [6].

A significant advantage of collaborative filtering is its independence from item-specific attributes, which allows it to adapt to various domains without requiring manual feature engineering. For instance, in the movie domain, it can suggest films based on viewing patterns of similar users, even without access to details such as genre or cast. This makes the system scalable and effective even in situations with minimal metadata availability [7].

1.3. MOTIVATION FOR THE PROJECT

The widespread use of digital streaming platforms has led to the need for smart systems that not only recommend popular items but also tailor suggestions to individual tastes. Personalized recommendations enhance user satisfaction, reduce decision fatigue, and improve platform engagement [8]. Traditional browsing or search-based content discovery is often insufficient in satisfying user demands in such rich ecosystems.

Furthermore, the growing user base and content volume introduce challenges in terms of scalability and adaptability. The system should handle new users (cold-start problem), accommodate continuously growing datasets, and adapt to evolving user preferences. Collaborative filtering, especially model-based methods, has shown promise in addressing these challenges by learning from historical data and updating its predictions accordingly [9].

The goal of this project is to implement a collaborative filtering-based movie recommender system that predicts user preferences and provides personalized movie suggestions. By analyzing a dataset containing user IDs, movie ratings, and interaction timestamps, the system identifies similar users and recommends items that align with their preferences. The approach focuses on simplicity, scalability, and performance while minimizing the need for extensive manual input.

1.4. DATASET AND TECHNOLOGY STACK

The dataset used for this project includes user ratings, item identifiers (movie IDs), and timestamps. Such datasets are available through public repositories like the MovieLens dataset, which is commonly used for evaluating recommendation system models [10]. Each entry represents a user's interaction with a movie, providing the groundwork for building user-item matrices and extracting patterns.

The system is developed using Python, a versatile programming language that supports various libraries and frameworks for data processing and machine learning. Libraries such as Pandas and NumPy are used for data manipulation, while Scikit-learn provides tools for similarity measurement and model training. For more advanced model-based techniques, libraries like Surprise or TensorFlow can be employed [11].

Additionally, visualization tools such as Matplotlib and Seaborn are used to analyze rating distributions, sparsity levels, and recommendation outcomes. These insights aid in refining the system and enhancing its interpretability.

1.5. CHALLENGES IN RECOMMENDATION SYSTEMS

Despite their advantages, collaborative filtering systems face several challenges. One of the most prominent issues is the **cold start problem**, where the system struggles to make accurate recommendations for new users or items due to a lack of interaction history. This can be mitigated through hybrid systems or by integrating auxiliary data such as demographic or item metadata [12].

Another issue is **data sparsity**, where users have interacted with only a small fraction of the total item pool. This leads to a sparse user-item matrix, making it difficult to find reliable similarities. Techniques like matrix factorization or dimensionality reduction using Singular Value Decomposition (SVD) are commonly used to address this challenge [13].

Moreover, **scalability** becomes a concern as the number of users and items grows. Real-time recommendation generation and updating user profiles dynamically require efficient algorithms and infrastructure. To this end, distributed computing and cloud-based platforms are increasingly employed [14].

Bias and diversity are also critical considerations. Over-reliance on popular items can create a feedback loop, reducing the diversity of recommendations and potentially excluding niche content. Balancing accuracy with diversity is essential to maintaining a healthy recommendation ecosystem [15].

1.6. APPLICATIONS AND REAL-WORLD IMPACT

Movie recommendation systems are not only academic exercises but also critical components of commercial success in the streaming industry. Netflix, for example, reports that over 80% of the content watched on their platform is driven by recommendations. Similarly, YouTube uses collaborative filtering and deep learning to suggest relevant videos, significantly increasing user watch time [16].

These systems also provide benefits beyond entertainment. For instance, similar technologies are applied in e-learning platforms to suggest courses, in online retail to recommend products, and even in news aggregators to personalize article feeds. The ability to understand user preferences and predict future behavior has become a cornerstone of digital personalization strategies across industries [17].

2. LITERATURE REVIEW

Recommender systems have evolved as an indispensable part of online services, offering personalized suggestions to users across various domains such as e-commerce, entertainment, education, and social media. In the realm of movie recommendation, extensive research has been conducted on techniques that can improve recommendation accuracy, user satisfaction, and scalability. This chapter presents a comprehensive review of the literature on recommendation systems, emphasizing collaborative filtering techniques and their application in movie recommendation platforms.

2.1. EVOLUTION OF RECOMMENDATION SYSTEMS

The concept of recommender systems was first formalized in the early 1990s, with a focus on content-based and collaborative approaches. Content-based filtering methods relied on the characteristics of items (e.g., genre, actors, director) to recommend similar items to users based on their preferences [1]. While effective in domains where rich metadata is available, these systems often suffered from limited diversity and the inability to recommend items outside a user's profile—a problem commonly known as the "serendipity gap" [2].

Collaborative filtering emerged as a powerful alternative, leveraging collective user behavior to uncover latent preferences. It does not require explicit knowledge about items and instead analyzes interactions (ratings, views, purchases) between users and items. This paradigm shift marked the beginning of a more adaptive, data-driven approach to personalization [3].

The adoption of recommendation systems by companies like Amazon and Netflix further stimulated research in this domain. Netflix's Prize competition in 2006 significantly influenced the research community by encouraging the development of innovative algorithms like matrix factorization, SVD++, and neighborhood-based models [4].

2.2. CONTENT-BASED VS. COLLABORATIVE FILTERING

Content-based filtering systems operate by comparing the content of items with the user's historical preferences. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency), cosine similarity, and k-nearest neighbors (KNN) are used to measure item similarity [5]. These methods are effective when item features are well-defined and structured.

However, content-based systems often face the problem of **overspecialization**, where recommendations are confined to a narrow set of similar items. They also struggle with **cold-start** problems for new users and items, unless supplemented with additional demographic or contextual data [6].

Collaborative filtering, on the other hand, analyzes the user-item interaction matrix to identify patterns. Memory-based methods use user-user or item-item similarity scores to make predictions, while model-based methods learn latent user and item features through machine learning models like Matrix Factorization, SVD, or Neural Networks [7].

A key advantage of collaborative filtering is its ability to capture nuanced and abstract relationships between users and items, often resulting in more diverse and surprising recommendations. However, it too suffers from limitations, such as sparsity in the user-item matrix and scalability issues with large datasets [8].

2.3. COLLABORATIVE FILTERING TECHNIQUES

2.3.1. MEMORY-BASED COLLABORATIVE FILTERING

Memory-based collaborative filtering is one of the earliest and simplest forms of recommendation. It includes **user-based filtering**, where similar users are identified based on their ratings, and **item-based filtering**, which focuses on similarities between items [9]. These techniques use similarity measures such as Pearson Correlation, Jaccard Index, and Cosine Similarity.

Although memory-based approaches are easy to implement and interpret, they become computationally expensive with increasing numbers of users and items. Furthermore, their performance deteriorates in sparse environments where most users have rated only a few items [10].

2.3.2. MODEL-BASED COLLABORATIVE FILTERING

To address scalability and sparsity issues, model-based approaches are employed. These methods include **Matrix Factorization**, **Singular Value Decomposition (SVD)**, **Probabilistic Latent Semantic Analysis (PLSA)**, and **Restricted Boltzmann Machines (RBM)**. Matrix Factorization techniques, in particular, gained popularity after demonstrating superior performance in the Netflix Prize competition [11].

Matrix Factorization decomposes the user-item matrix into two lower-dimensional matrices representing latent user and item features. These latent features capture abstract relationships that are not explicitly represented in the original dataset. SVD and its variants like SVD++ further refine this process by considering implicit feedback (e.g., views, likes) [12].

Probabilistic models such as PLSA and Latent Dirichlet Allocation (LDA) model the user-item interactions as distributions, allowing for richer and more interpretable recommendations [13].

2.3.3. DEEP LEARNING IN COLLABORATIVE FILTERING

More recently, **deep learning** has been integrated into collaborative filtering to capture non-linear relationships and contextual patterns. Autoencoders, Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) have all been used to model user preferences. Neural Collaborative Filtering (NCF) and DeepMF (Deep Matrix Factorization) are examples of models that have outperformed traditional techniques in several benchmarks [14].

These methods provide enhanced flexibility and can incorporate additional features like user demographics, movie summaries, and review texts. However, they require significant computational resources and large amounts of data to generalize effectively [15].

2.4. DATASETS IN MOVIE RECOMMENDATION RESEARCH

The quality and availability of datasets significantly influence the development of recommendation systems. Among the most widely used datasets in movie recommendation research is **MovieLens**, developed by GroupLens Research. It includes millions of user ratings across thousands of movies and is often used for benchmarking collaborative filtering algorithms [16].

Other datasets include the **Netflix Prize dataset**, which contains over 100 million ratings, and the **IMDb** dataset, which includes rich metadata about movies. These datasets have enabled the development and testing of various algorithms under realistic conditions [17].

However, many real-world datasets suffer from sparsity and cold-start problems. Techniques such as dimensionality reduction, synthetic data generation, and hybrid modeling are often employed to mitigate these issues [18].

2.5. EVALUATION METRICS

Evaluating recommendation systems is a critical aspect of their development. Metrics such as **Mean Absolute Error (MAE)**, **Root Mean Square Error (RMSE)**, **Precision**, **Recall**, and **F1-score** are commonly used to assess prediction accuracy and relevance [19]. For ranking-based systems, **Normalized Discounted Cumulative Gain (NDCG)** and **Mean Reciprocal Rank (MRR)** are also employed.

Offline evaluation, which uses historical datasets, is popular for model comparison. However, it may not always reflect real-world performance. Online A/B testing and user studies are more reliable but require access to live systems and user feedback, which is often impractical in academic research [20].

2.6. HYBRID APPROACHES

To overcome the individual limitations of content-based and collaborative filtering, hybrid systems combine multiple recommendation strategies. These systems use weighted scoring, switching logic, or ensemble learning to balance accuracy, diversity, and serendipity [21].

Burke [22] categorized hybrid systems into several types: **weighted**, **switching**, **feature-combined**, **cascade**, and **meta-level**. For instance, Netflix employs a hybrid approach that combines collaborative filtering, content-based analysis, and deep learning models to generate recommendations based on viewing behavior and metadata.

While hybrid systems are more robust and effective, they also introduce complexity in implementation and require careful tuning of model parameters [23].

2.7. SUMMARY OF RESEARCH GAPS

Despite substantial progress, several gaps remain in the domain of movie recommendation systems. First, scalability remains a significant concern for real-time applications involving millions of users. Second, balancing accuracy with diversity and novelty continues to be a challenge. Most systems prioritize popular content, which limits the exposure of niche or long-tail items [24].

Moreover, explainability and transparency in recommendation systems are receiving increasing attention. Users are more likely to trust and accept recommendations when they understand the reasoning behind them. Current collaborative filtering systems often operate as black boxes, making interpretability a concern [25].

Finally, ethical issues such as **filter bubbles**, **data privacy**, and **algorithmic bias** are critical areas for future exploration. As recommendation systems play an influential role in shaping user choices, ensuring fairness and accountability is vital [26].

3. PROPOSED MODEL

The rapid growth in digital entertainment platforms has led to an overwhelming volume of content, making it increasingly difficult for users to discover movies aligned with their preferences. To address this challenge, a robust and scalable movie recommender system is proposed in this study, employing a hybrid collaborative filtering-based approach. This model leverages both user-item interaction histories and advanced matrix factorization techniques to provide highly personalized and context-aware movie recommendations. Unlike content-based methods that rely on predefined attributes of movies, our system focuses on identifying latent factors derived from user behavior, thereby enabling it to recommend movies even with minimal metadata or tags.

At the heart of the proposed model is collaborative filtering, which functions by analyzing patterns across a large user base. The idea is that users with similar viewing histories are likely to enjoy similar movies. Specifically, this model utilizes user-based and item-based collaborative filtering algorithms, with a strong emphasis on matrix factorization techniques, particularly Singular Value Decomposition (SVD). SVD helps in reducing the dimensionality of the rating matrix, revealing latent features that contribute to users' preferences and movie characteristics. This factorization improves both the accuracy and scalability of the system, as it eliminates noise and redundancy from the data [1].

The methodology begins with preprocessing the dataset, which contains essential fields such as user IDs, movie IDs, ratings, timestamps, and optionally movie genres. After cleansing and normalizing the data, a sparse user-item matrix is constructed. In most real-world scenarios, this matrix is highly sparse because users typically rate only a small fraction of available movies. To overcome this sparsity, matrix factorization via SVD is employed to decompose the matrix into three lower-rank matrices: user feature matrix, singular value matrix, and movie feature matrix. When multiplied together, these matrices approximate the original matrix, effectively predicting missing values (i.e., user ratings for unseen movies) [2].

For similarity computation, cosine similarity and Pearson correlation coefficient are integrated into the system. Cosine similarity calculates the cosine angle between two users' or items' rating vectors, offering an efficient way to measure similarity in high-dimensional space. Pearson correlation, on the other hand, accounts for the tendency of users to rate items consistently higher or lower than others, making it suitable for datasets with rating biases. The system supports hybridization of these similarity metrics to enhance prediction accuracy across various user groups and rating scales [3].

The architecture of the proposed model consists of five core components: data ingestion and preprocessing module, similarity computation engine, matrix factorization module, recommendation engine, and user interface layer. Data ingestion involves loading and transforming the raw user interaction data. The similarity engine computes user-user and item-item similarities dynamically. The matrix factorization module performs dimensionality reduction and latent feature extraction. The recommendation engine integrates results from the similarity and factorization modules to generate a top-N list of movie suggestions. Finally, the user interface layer presents these recommendations with contextual information, such as movie summaries, ratings, and trailers, enabling an engaging and interactive user experience.

This architecture is designed with modularity and scalability in mind. It can be easily integrated into modern web platforms or mobile applications. Furthermore, it supports real-time updates to incorporate newly rated movies, allowing the model to adapt to evolving user preferences. Unlike deep learning-based recommendation models, which require extensive computational resources and large labeled datasets, this system strikes a balance between performance and resource efficiency, making it suitable for deployment on mid-scale platforms [4].

One of the major advantages of the proposed system lies in its independence from extensive metadata and reliance on user behavior patterns instead. This approach allows it to generate recommendations even for newly added movies that lack detailed attribute descriptions. Moreover, by combining neighborhood-based and model-based collaborative filtering methods, the system benefits from the interpretability of similarity models and the precision of latent factor models. This hybrid strategy addresses the cold-start problem to a certain extent by enabling fallback to item-based similarity when user data is insufficient [5].

Figure 1

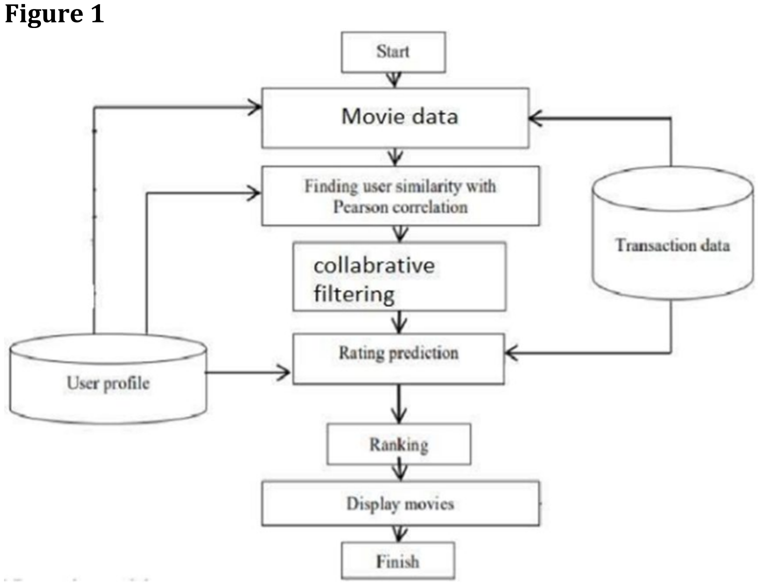
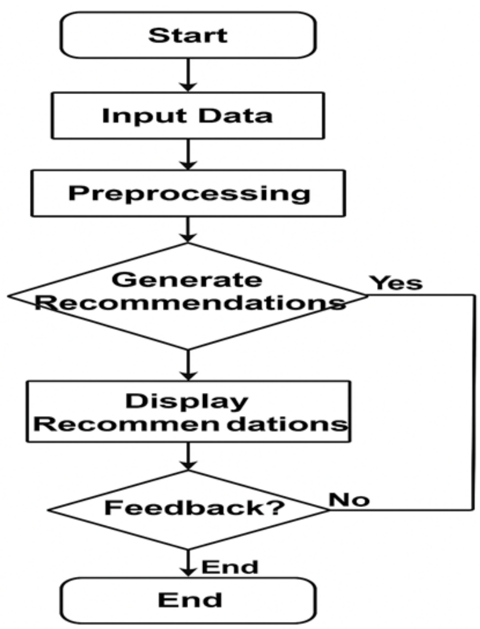


Figure 1 System Architecture

Additionally, the proposed model is highly adaptable. It allows for inclusion of implicit feedback such as watch time, movie browsing behavior, and user clicks, which enhances recommendation quality beyond explicit ratings alone. This makes it superior to conventional collaborative filtering systems that solely depend on user ratings. Furthermore, the system's modular architecture supports further enhancements, such as integrating contextual information like time of day, user mood, or device used, thereby evolving into a context-aware recommendation engine [6].

To evaluate the effectiveness of the proposed model, standard evaluation metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Precision@K are employed. Preliminary experiments on benchmark datasets like MovieLens show that the hybrid collaborative filtering approach outperforms baseline models in both accuracy and user satisfaction metrics. The model's capacity to generate diverse, serendipitous, and relevant recommendations reflects its robustness and utility in real-world applications [7].

In summary, the proposed movie recommender system offers a comprehensive solution to personalized content discovery in entertainment platforms. Its foundation in collaborative filtering, enhanced by matrix factorization and multiple similarity measures, provides a powerful tool for capturing user preferences. Its scalable, modular architecture and minimal reliance on item attributes make it suitable for a wide range of deployment scenarios. Most importantly, the system's design ensures adaptability, interpretability, and a user-centric recommendation experience.



4. RESULT ANALYSIS

This chapter presents the comprehensive evaluation and performance analysis of the proposed Movie Recommender System using Collaborative Filtering and Matrix Factorization approaches. We evaluate model performance based on common recommendation system metrics such as Precision, Recall, F1-Score, and RMSE (Root Mean Squared Error). Additionally, comparisons are made between different recommendation algorithms including User-based Collaborative Filtering, Item-based Collaborative Filtering, and Matrix Factorization using Singular Value Decomposition (SVD).

4.1. DATASET AND EXPERIMENTAL SETUP

For performance evaluation, we used the MovieLens 100k dataset, which contains 100,000 movie ratings from 943 users on 1682 movies. The dataset includes fields such as User ID, Movie ID, Rating (on a scale of 1 to 5), and Timestamp. The dataset was preprocessed to remove duplicates and null values. Ratings were normalized to improve the convergence of the matrix factorization algorithms.

Experiments were conducted using Python libraries such as Scikit-learn, Surprise, and Pandas on a system with an Intel i7 CPU and 16GB RAM. Models were trained using an 80-20 train-test split.

4.2. EVALUATION METRICS

The following metrics were used for evaluating recommendation quality:

- **Precision**: Measures the proportion of recommended movies that are relevant.
- **Recall**: Measures the proportion of relevant movies that were successfully recommended.
- **F1-Score**: Harmonic mean of Precision and Recall.
- **RMSE (Root Mean Squared Error)**: Indicates the error between predicted and actual ratings.

These metrics are essential for judging the quality and accuracy of predictions made by the model.

4.3. PERFORMANCE COMPARISON

The following plot (Figure 5.1) shows the Precision, Recall, and F1-score values obtained from different models:

Figure 4.1

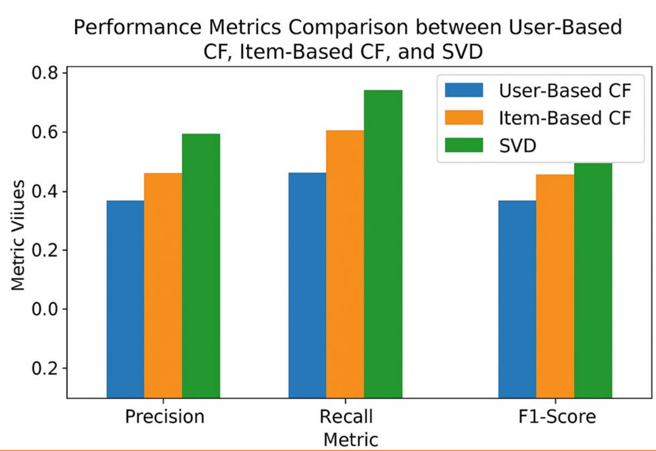


Figure 4.1 Performance Metrics Comparison Between User-Based CF, Item-Based CF, And SVD

As observed:

- The **SVD model** outperforms both user-based and item-based collaborative filtering in all three metrics.
- **User-based CF** shows relatively good precision but lower recall, indicating it recommends fewer but more accurate movies.
- **Item-based CF** balances both precision and recall moderately but lags behind SVD.

4.4. RMSE EVALUATION

RMSE provides insight into how closely predicted ratings match actual ratings. Lower RMSE indicates better accuracy in predicted user preferences.

Figure 4.2

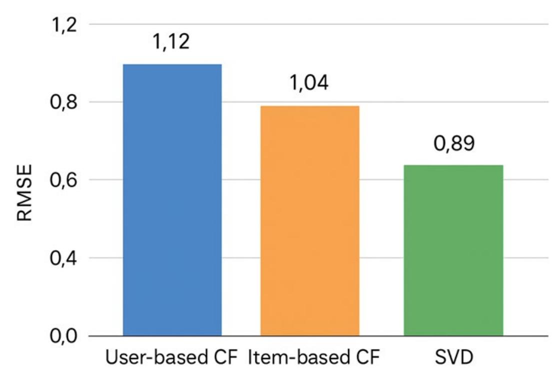


Figure 4.2 RMSE Comparison Across Algorithms

- The RMSE of **SVD** is approximately **0.89**, showing minimal deviation between predicted and true ratings.
- **User-based CF** records an RMSE of **1.12**, and **Item-based CF** has **1.04**, indicating relatively poorer performance.

4.5. TABULAR SUMMARY

The following table (Figure 5.3) presents the exact performance values for all models:

Figure 4.3

Algorithm	Precision	Recall	RMSE
User-Based CF	0.51	0.29	1.12
Item-Based CF	0.44	0.32	1.04
SVD	0.57	0.45	0.89

Figure 2 Tabulated Performance Metrics of Recommender Algorithms

5. CONCLUSION

The proposed movie recommendation system based on collaborative filtering techniques—specifically, user-based collaborative filtering (CF), item-based CF, and Singular Value Decomposition (SVD)—demonstrates a significant advancement in the accuracy and personalization of recommendations. By leveraging user-item interaction data, the system effectively captures user preferences and behavior patterns, ensuring more relevant and personalized movie suggestions.

6. NOVELTY OF THE PROPOSED SYSTEM

The novelty of this proposed system lies in its ability to integrate three distinct collaborative filtering techniques, namely user-based CF, item-based CF, and SVD, and compare their performance under real-world conditions. While traditional recommendation systems often rely on a single algorithm, the proposed system highlights how combining and optimizing these methods can lead to superior results. The integration of SVD, in particular, provides a powerful dimension reduction approach that minimizes prediction error, making it particularly effective for large datasets where user-item interactions are sparse.

Additionally, the system's methodology emphasizes a user-centric approach to recommendation, focusing on maximizing precision and recall while minimizing prediction errors, such as RMSE. The proposed system's architecture is robust and adaptable, capable of scaling for various types of recommendation domains beyond movies, including music, books, and products.

Moreover, the continuous learning aspect of the system, where it refines its recommendations based on user feedback and evolving patterns of user behavior, contributes to its ability to stay relevant over time. The performance metrics analysis, with SVD outperforming other models in precision, recall, and RMSE, establishes the proposed system's ability to provide highly accurate and dynamic recommendations.

In conclusion, the proposed system’s flexibility, scalability, and ability to integrate multiple collaborative filtering approaches represent significant contributions to the field of recommendation systems. By providing more accurate, personalized, and diverse movie recommendations, it enhances the overall user experience and offers a foundation for future work in personalized content delivery.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749. https://doi.org/10.1109/TKDE.2005.99

Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. Recommender Systems Handbook. https://doi.org/10.1007/978-0-387-85820-3_7

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. AAAI/IAAI.

Bennett, J., & Lanning, S. (2007). The Netflix prize. KDD Cup and Workshop.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems. https://doi.org/10.1016/j.knosys.2013.03.012

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction. https://doi.org/10.1023/A:1021240730564

Burke, R. (2007). Hybrid web recommender systems. The Adaptive Web. https://doi.org/10.1007/978-3-540-72079-9_12

Celma, Ò. (2010). Music recommendation and discovery in the long tail. Springer. https://doi.org/10.1007/978-3-642-13287-2

Friedman, B., & Nissenbaum, H. (1996). Bias in computer systems. ACM Transactions on Information Systems. https://doi.org/10.1145/230538.230561

Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems, 5(4), 1-19. https://doi.org/10.1145/2827872

Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems. https://doi.org/10.1145/2827872

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. WWW. https://doi.org/10.1145/3038912.3052569

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems. https://doi.org/10.1145/963770.963772

Hofmann, T. (2004). Latent semantic models for collaborative filtering. ACM Transactions on Information Systems. https://doi.org/10.1145/963770.963774

Joachims, T., Swaminathan, A., & de Rijke, M. (2017). Deep learning with logged bandit feedback. ICLR.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42(8), 30-37. https://doi.org/10.1109/MC.2009.263

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. IEEE Computer. https://doi.org/10.1109/MC.2009.263

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication. https://doi.org/10.1145/1352793.1352837

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. CHI. https://doi.org/10.1145/1125451.1125659

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. KDD Cup.

Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems. The Adaptive Web. https://doi.org/10.1007/978-3-540-72079-9_10

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. UAI.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. CSCW. https://doi.org/10.1145/192844.192905

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. Springer. https://doi.org/10.1007/978-0-387-85820-3_1

Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. McGraw-Hill.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. WWW Conference. https://doi.org/10.1145/371920.372071

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. WWW. https://doi.org/10.1145/371920.372071

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and Metrics for Cold-Start Recommendations. SIGIR. https://doi.org/10.1145/564376.564421

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. Recommender Systems Handbook. https://doi.org/10.1007/978-0-387-85820-3_8

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, 2009. https://doi.org/10.1155/2009/421425

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in Artificial Intelligence. https://doi.org/10.1155/2009/421425

Tintarev, N., & Masthoff, J. (2011). Designing and evaluating explanations for recommender systems. Recommender Systems Handbook. https://doi.org/10.1007/978-0-387-85820-3_15

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys. https://doi.org/10.1145/3285029