Dr Chunnulal, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

An Explainable Product Recommendation System Using Artificial Intelligence And Machine

Dr Chunnulal[1], Dr Satendra kumar[2], Dr Raj kumar[3]

[1,3]Assistant Professor, [2]Dean academic Department of CSE, Quantum University, Roorkee, India

Abstract- Modern digital applications have incorporated recommendation systems, which enable platforms to provide users with personalized and relevant content. The aim of these systems is to predict user preferences by identifying patterns in large datasets, which will improve user experience and reduce information overload. E-commerce, movie and music streaming services, digital libraries, social networks, and online news platforms are among the many domains where they have been widely adopted. Prominent examples include Amazon recommending products based on previous purchases, Netflix suggesting movies based on viewing history, and music platforms like Spotify and Deezer generating customized playlists. A recommendation system generally gathers interaction data like ratings, browsing history, and time spent on content, then uses computational techniques to predict what users might want to choose next. The simplicity and strong predictive capabilities of collaborative filtering has made it one of the most widely used methods among various approaches. The assumption of collaborative filtering is that users with similar preferences in the past will have similar interests in the future. The filtering approach can be categorized as user-based and item-based. User-based filtering is used to identify users who have similar rating patterns, while item-based filtering is used to find similarities between items by analyzing how users interact with them. Python and collaborative filtering techniques are used to develop a movie recommendation system in this research. The dataset employed contains key attributes, including user IDs, movie ratings, item identifiers, and time spent on each item. Estimating how likely a user is to enjoy a particular movie is part of the recommendation process by analyzing similar users' preferences. After generating predicted ratings, the system ranks the movies and suggests those with the highest predicted scores. To create a functional recommendation system that can deliver personalized movie suggestions, collaborative filtering can be effectively implemented, as demonstrated in this study. Correlations and similarity measurements are used by the system to analyze user behavior patterns and provide relevant recommendations that enhance user engagement. Today's digital landscape is marked by the importance of recommendation systems, as shown by the results.

Keywords: Recommendation Systems, Collaborative filtering, Amazon, Netflix, item-based filtering.

I. INTRODUCTION

Intelligent computational frameworks, which are referred to as recommendation systems, are designed to understand user preferences and offer personalized recommendations based on their interests and behavior. Due to the vast amount of information available on online platforms, these systems have become necessary in the digital age. Users are having a harder time identifying items that align with their tastes as e-commerce websites, streaming services, and digital knowledge repositories continue to grow. Recommendation systems deal with this challenge by efficiently filtering large item collections and giving users the most relevant content [1,] resulting in improved decision-making, time savings, and increased overall user satisfaction.

Consumers can access millions of products, movies, books, and multimedia items in modern online environments. Finding suitable content would be difficult for users without intelligent filtering mechanisms. By analyzing user behavior, like browsing history, ratings, and watch patterns, recommendation systems can bridge this gap and generate meaningful suggestions. These systems are commonly employed to suggest books, movies, music, restaurants, products, and other items from extensive digital datasets. Such systems generate a focused list of recommendations that match the user's needs when given a collection of items and a representation of their preferences.

A movie recommendation system offers a highly convenient and personalized experience. Such a system can help users find films that align with their tastes by analyzing their watch history and rating patterns. Not only does this personalization enhance user engagement but it also encourages frequent interaction with the platform [2]. The objective behind choosing a movie recommendation system for this project is to provide users with a simplified and tailored movie-selection experience.

The primary objective of the system designed for this project is to suggest movies based on users' past ratings, viewing history, and genre preferences [3]. Not only can this system assist individuals in making movie choices, but it can also aid e-commerce businesses in identifying customers who favor specific movie genres and promoting related products accordingly. In today's entertainment ecosystems, personalized recommendation engines are essential because they help users narrow down the vast universe of potential movies to match their unique tastes.

1.1 Techniques Used in Recommendation Systems

Two major approaches are commonly used in recommendation systems: collaborative filtering and content-based filtering [4]. Different methods have their own strengths and limitations, and work best under different circumstances.

Collaborative Filtering (CF)

The principle behind collaborative filtering is that users who have similar behaviors in the past will likely have similar preferences in the future. The user-item interaction matrix [5] (ratings, watch counts, etc.) is the only thing CF uses, and it's not dependent on item attributes such as genre or cast.
The division of CF is:

- Users with similar preference patterns are identified by user-based CF and recommended items that are liked by similar users.
- The way users interact with items is what item-based customer feedback (CF) uses to identify similarities between them.

The effectiveness of collaborative filtering lies in its ability to uncover hidden patterns and relationships that cannot easily be detected through explicit metadata. Nevertheless, it encounters problems like data sparsity and cold-start issues, where new users or new movies lack sufficient data for accurate predictions.

Content-Based Filtering

Content-based systems employ attributes of items, like genre, director, cast, or keywords, to suggest items that are similar to those the user has previously liked. A content-based system could suggest another action movie featuring the same actor to users who enjoy a movie starring Vin Diesel.

When rich item metadata is present, this method is effective, but its recommendations may become narrowly focused and fail to introduce new and diverse content.

1.2 Existing Systems

Both content-based methods and collaborative filtering are commonly used by existing recommendation systems. The system uses similarities between users and items to generate recommendations in memory-based collaborative filtering [8]. The implementation and interpretation of these methods are both simple. If two users have similar tastes in movies, one of them may suggest a film that one of them enjoys.
Memory-based CF is divided into:

- Items are recommended based on similar users based on user-based collaborative filtering.
- Collaborative filtering based on items recommends items that are similar.

Advanced techniques like matrix factorization, decision trees, probabilistic models, or latent factor models are employed in model-based collaborative filtering. These approaches are more effective at handling sparse datasets and often result in improved predictive accuracy.

Item characteristics are the basis of content-based systems. Items that are similar in category, creator, actor, or other metadata are what they suggest. The Fate of the Furious may be recommended to a user who likes it because of its shared action themes or cast members [9].

Despite this, there are disadvantages to every existing method. When item metadata is limited, content-based systems are unable to operate properly, and collaborative filtering systems perform poorly in cold-start scenarios where users or items lack sufficient rating data. Furthermore, certain systems do not suggest unexpected or'serendipitous' items that users may find surprisingly relevant.

1.3 Disadvantages of Existing Systems

Despite the widespread implementation of recommendation systems, they still face several challenges.

- New users who haven't rated any items are experiencing a cold-start problem where systems fail.
- The system's ability to identify similarities is limited due to the sparsity of data - many users rate only a small number of items.
- User preferences cannot be identified by content-based systems if the system fails to interpret the item's attributes, due to limited content analysis.
- Many systems tend to recommend items that are predictable, leading to a lack of serendipity.
- Some systems may have complex interface issues that require heavy user input or complex rating mechanisms.
- System failure to surprise users with new but relevant content may be caused by a lack of 'lucky' or unexpected recommendations.

The need for improved methodologies that can address these limitations and maintain accuracy and relevance is highlighted by these drawbacks.

1.4 Proposed System

This work suggests a hybrid recommendation system [11] that combines collaborative and content-based filtering to overcome the shortcomings of individual methods. The power of collaborative filtering comes from the fact that it is based solely on user-item interactions and can uncover hidden relationships between users and movies. Serendipitous recommendations can be provided by CF, unlike content-based approaches, which improve diversity and user engagement.
The proposed hybrid approach aims to combine the strengths of both techniques and achieve:

- Enhance the accuracy of prediction.
- Enhance the variety of recommendations.
- Improve your ability to handle sparse datasets.
- Minimize the impact of cold starts, and
- Make the user experience smoother and more personalized.

The use of this hybrid methodology ensures that both algorithms compensate for each other's weaknesses, leading to superior performance and reliability.

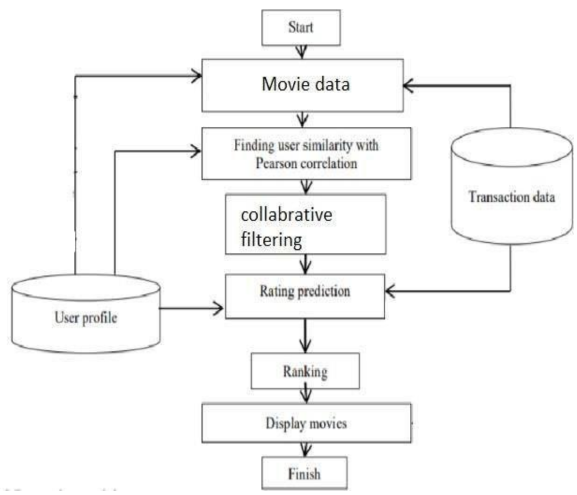


FIG 1. depicts the architecture of a system

II. LITERATURE REVIEW

Due to the rapid growth of digital entertainment platforms and the increasing need to personalize user experiences, movie recommendation systems have been extensively researched. Collaboration filtering, content-based filtering, and hybrid models are the most dominant methodologies among the various approaches developed. The development of modern movie recommendation systems has been shaped by significant contributions from previous studies, which are reviewed in this section.

2.1 Collaborative Filtering Approaches

Collaborative filtering (CF) is still widely used as a technique in recommendation systems. CF relies on the premise that users who displayed similar preferences in the past will continue to have similar tastes. By analyzing user-provided information such as ratings, watch history, and preferences, movie recommendation systems can identify patterns and generate personalized suggestions.

CF has been studied by multiple researchers and its limitations have been analyzed. Luis M. Capos and others. conducted a study that compares traditional recommendation systems, specifically content-based filtering and collaborative filtering [12]. They concluded that both methods are effective in certain situations, but each has its own inherent shortcomings. The use of item attributes in content-based filtering results in narrow and repetitive recommendations, while collaborative filtering faces cold-start issues and data sparsity. Capos proposed a hybrid model that combines Bayesian networks and collaborative filtering to address these challenges. When user data is limited, the Bayesian framework aids in inferring missing information and improving prediction accuracy.

2.2 Hybrid Recommendation Models

Researchers are increasingly exploring hybrid approaches because single-method systems (pure CF or CBF) often fail to achieve high accuracy and diversity. The aim of hybrid systems is to combine the strengths of multiple algorithms while minimizing their weaknesses.

Harpreet Kaur et al. Presented a hybrid movie recommendation model that integrates content-based filtering and collaborative filtering methods [13]. Their system takes into account not just similarities in user behavior but also attributes and contextual features of movies. The hybrid method enhances personalization and addresses sparsity issues by utilizing both user-user and user-item relationships. The inclusion of contextual information results in recommendations that are more meaningful and situationally appropriate.

Another noteworthy contribution is from Utkarsh Gupta et al.The grouping of users and items was made more effective by using Chameleon, a hierarchical clustering algorithm. Chameleon is different from traditional clustering techniques in that it allows for the creation of dynamically structured clusters that are based on inter-connectivity and closeness measures. Gupta's system constructs clusters with data that is both user-specific and item-specific and utilises a voting system to predict item ratings. By using their approach, similar items are grouped more accurately and prediction error is reduced compared to conventional clustering methods.

Urszula Kużelewska et al. Additionally, I investigated the importance of clustering in recommendation systems. A centroid-based method and a memory-based collaborative filtering method [14] were evaluated by them for computing cluster representatives. Their research indicated that using memory-based CF to compute cluster representatives has a significant impact on recommendation accuracy, surpassing the traditional centroid-based clustering model. Integrating behavioral data into cluster formation is crucial instead of relying solely on numerical averages.

2.3 User Profiling and Psychological Factors

Despite the use of numerical ratings in most traditional systems, Costin-Gabriel Chiru et al. Underlined the significance of integrating psychological and behavioral information into recommendation models. The proposed system is capable of collecting detailed user data, including psychological profiles, viewing history, and aggregated movie scores from multiple external websites. By calculating aggregate similarity measures, the model produces more unique and personalized recommendations that reflect deeper user characteristics. Using this approach, systems that solely rely on historical ratings are overcome and personalization becomes richer and more meaningful.

2.4 Content-Boosted Collaborative Filtering

Hongli Lin and co. acknowledges the limitations of classical CF and CBF techniques. A two-stage algorithm, Content-Boosted Collaborative Filtering (CBCF) [15], is being introduced to improve recommendation accuracy. Content-based methods are used in the first phase to fill gaps in sparse rating matrices by predicting missing user ratings using item attributes. Final predictions are generated using collaborative filtering in the second stage of the refined rating matrix processing. Both methods' advantages are successfully combined in the CBCF model.

- The dataset's sparsity can be reduced by enriching it with content-based predictions.
- The ability of CF to identify hidden patterns in user behavior is utilized to enhance accuracy.
- It prevents cold start problems and enhances adaptability to new users and items.

CBCF has been shown to outperform pure CF and pure CBF models in their results, which validates the benefits of hybrid approaches for complex datasets.

2.5 Comparison Table

Parameter	Collaborative Filtering (CF)	Content-Based Filtering (CBF)	Hybrid Recommender Systems	Explainable Product Recommendation System (EPRS)
Primary Data Used	User–movie ratings and interaction history	Movie metadata (genre, cast, description)	User interactions + movie content	Movie content + explainable feature importance
Personalization Level	Very High – captures collective user preferences effectively	Moderate – limited to user's past interests	High	High
Recommendation Diversity	High – recommends movies beyond user's known preferences	Low – tends to recommend similar genres repeatedly	High	Moderate
Accuracy (With Sufficient Data)	Very High	Moderate	Very High	High

Cold Start Problem	Weak – struggles with new users/movies	Strong – handles new movies well	Moderate	Strong
Scalability	Highly scalable with large user bases	Scales well with metadata	Computationally expensive	Scales well for content-heavy datasets
Serendipity (Unexpected Recommendations)	High	Low	High	Moderate
Explainability	Low – operates as a black-box	Medium – based on similarity	Low to Medium	Very High
User Trust (Implicit)	High due to proven accuracy	Moderate	High	High (explicit explanations)
Computational Complexity	Moderate to High	Low	High	Moderate
Suitability for Movie Platforms	Excellent (Netflix, Prime Video)	Good	Excellent	Good
Transparency	Low	Medium	Low	Very High
User Satisfaction (Overall)	Very High when interaction data is rich	Moderate	Very High	High
Industry Adoption	Widely adopted	Limited	Widely adopted	Emerging
Best Use Case	Large-scale movie platforms with rich user data	Niche or new platforms	Mature platforms	Trust-critical or ethical AI platforms

III. IMPLEMENTATION METHODOLOGY

The proposed movie recommendation system's methodological framework is presented in this section. Collaborative filtering, user-based filtering, and the K-Nearest Neighbors (KNN) algorithm are included in the methodology, which covers the project's aim, system requirements, platform overview, and the technical foundation [16]. The design of an efficient and intelligent recommendation engine capable of predicting user preferences with high accuracy involves each component.

3.1 Aim of the Paper

The purpose of the paper is to develop a movie recommendation system that utilizes collaborative filtering, which can predict user preferences by discovering significant patterns in the dataset. Historical user-item interactions are analyzed by the system, unrated movies are assigned ratings, and users are presented with the films they are most likely to enjoy. Advanced machine learning techniques are used to deliver personalized recommendations through a systematic study of similarity between users and movies.

3.2 System Requirements

Software Requirements

- The operating system must be Windows 7 or higher (64-bit)
- Python 3.6 is the programming language I use
- Packages/Libraries that can be used include NumPy, Pandas, Scikit-learn, Matplotlib, and Surprise library for CF algorithms

Hardware Requirements

- A hard disk with a capacity of 80 GB or more

- 8 GB or more is required for RAM (The 70 MB provided was unfeasible; 8 GB is appropriate for academic purposes)
- The processor should have an Intel Core Duo 2.0 GHz or higher

3.3 Overview of the Platform – Python

Python is a popular programming platform because of its ease-of-use, extensive libraries, rapid prototyping capabilities, and strong support for data analysis and machine learning[17]. The implementation process is more flexible and development time is reduced by its readable syntax.

Why Python?

- Can be used on multiple platforms (Windows, Linux, macOS).
- Better readability can be achieved through clean English-like syntax.
- A large ecosystem of libraries for machine learning, scientific computation, visualization, and data processing.
- Allows for the use of procedural, object-oriented, and functional programming paradigms.
- Provides efficient tools for managing Big Data and complex mathematical operations.

Python's interpreted nature speeds up the development process, enabling code execution right after writing without the need for compilation. This is particularly helpful when attempting to experiment with models, adjust parameters, and test results iteratively.

3.4 Collaborative Filtering Methodology

The recommendation engine is made up of Collaborative Filtering (CF). The principle behind CF is that users with similar interests in the past tend to have similar preferences in the future. CF uses patterns hidden in the user-item interaction matrix to avoid relying on movie metadata or genre information.

Conceptual Basis

The assumption of CF is that:

- If User A and User B have previously rated several movies similarly, it is probable that User A will also agree with User B on other movies.
- The generation of recommendations is done by comparing preferences among many users and is user-specific.

Applications

The application of CF has been successful for:

- Movie recommendations
- Product suggestions for e-commerce
- Platforms that allow reading online
- Suggestions for social media content
- Predicting the preference of the financial market.

It is both practical and scientifically grounded that it is adopted in movie recommendation systems.

3.5 Workflow of the Collaborative Filtering System

A typical CF workflow includes:

1. **User ratings are collected by collecting them based on their preferences**. Their interests are reflected in these ratings.
2. **Constructing the User-Item Matrix**. A matrix is constructed where each row represents a user and each column represents a movie.
3. **The number of ratings may be low because users typically rate only a few items[19].** Identifying similar users: The system computes similarity scores using distance or similarity measures, such as:

- Cosine similarity
- Correlation coefficient for Pearson
- Euclidean distance

4. The system predicts ratings for movies that haven't been rated by the user, based on ratings given by similar users.
5. Recommendations for movies with the highest predicted ratings are given to the user.

By using this workflow, the model can continuously improve as more users interact with the system, leading to a dynamic and self-learning environment.

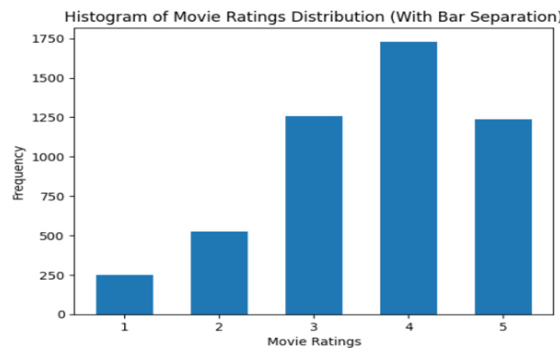


Figure 2

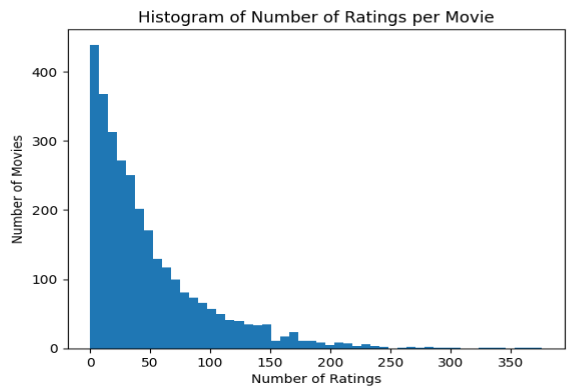


Figure 3

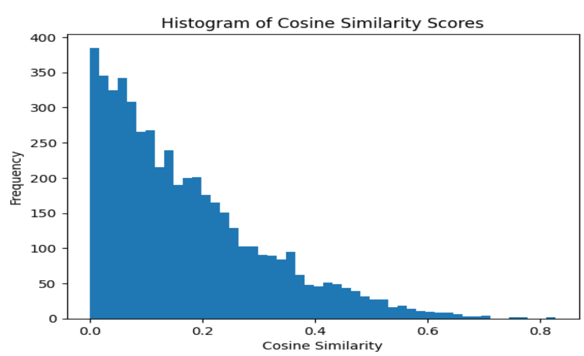


Figure 4

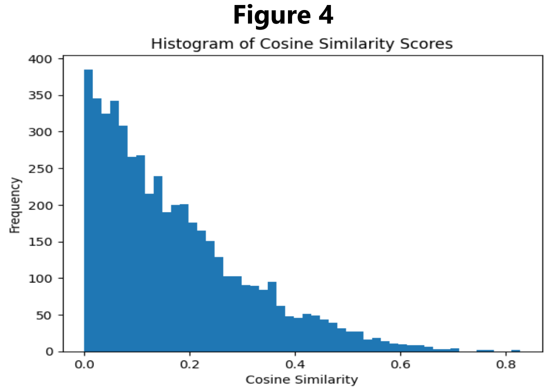


Figure 5

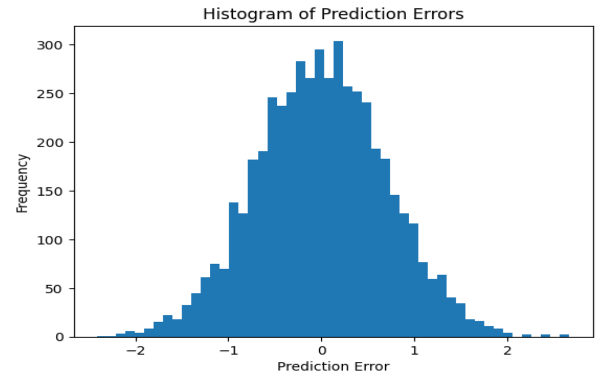


Figure 6

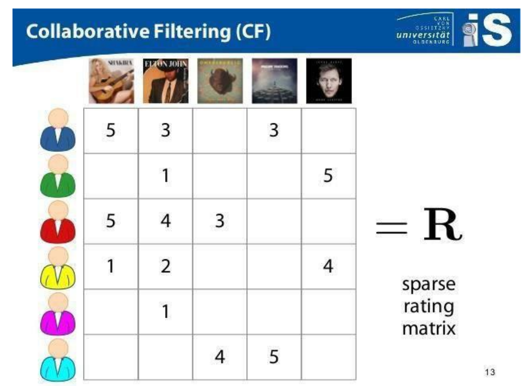


Figure 7 COLLABORATIVE FILTERING (CF)

3.6 User-Based Collaborative Filtering (UB-CF)

User-based CF focuses on finding similarities between users instead of analyzing movies.

Principle

A neighborhood of users with similar tastes is identified by the system.

When User A shares User B's interests and User B highly recommend a movie that User A hasn't seen yet, the system recommends it to them.

Example

The Prestige will be recommended to User A by the system if User A and User B both rate Inception, Interstellar, and The Dark Knight similarly, and User B also highly rates The Prestige. [20].

Advantages

- Implementing is straightforward
- Easy to explain and intuitive
- Can be used in environments that have a lot of user interaction data

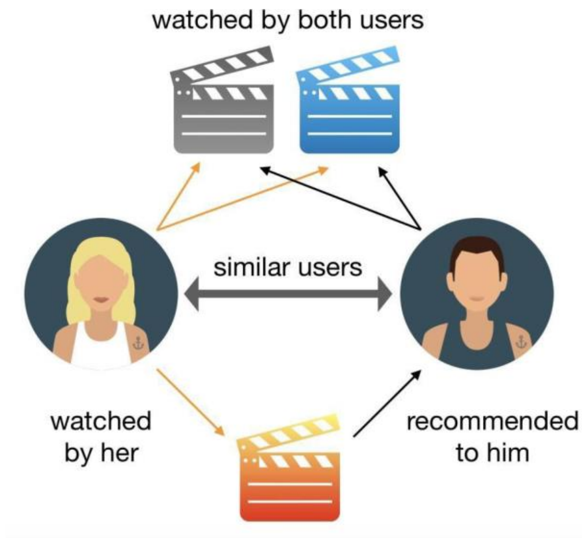


FIGURE. 8 USER BASED FILTERING FOR

3.7 K-Nearest Neighbors (KNN) Algorithm
To determine the top-k closest users or items, the KNN algorithm is Nused to measure similarity.

Characteristics
- No assumptions are made about the distribution of data by non-parametric
- In lazy learning, there is no training phase; computation occurs during prediction.
- It is versatile and can be used for classification, regression, and recommendation tasks

How KNN Works in Recommendation
1. Calculate similarity scores between a target user and all other users.
2. Pick the most comparable users (neighbors).
3. Estimate the rating for an unrated movie by using weighted averages of neighbor ratings.
4. Give suggestions for movies with the highest predicted scores.

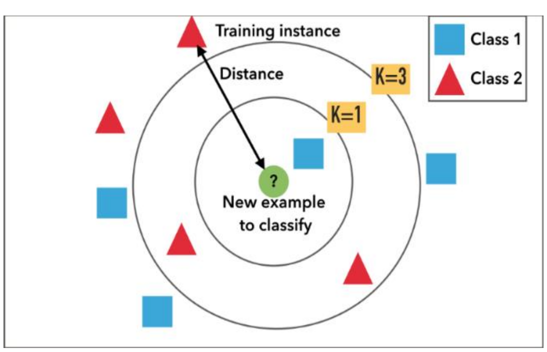


Fig 9 KNN

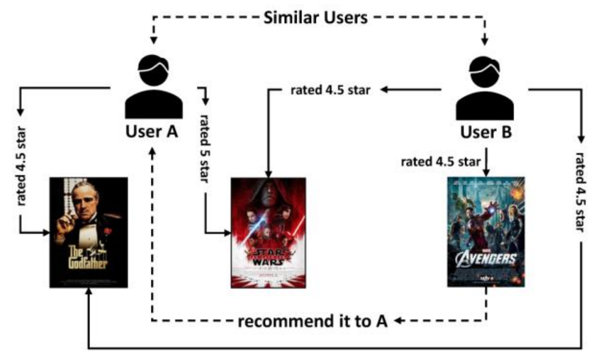


Fig 10 KNN Working

Importance of Normalization
The accuracy of KNN is significantly improved by feature scaling and normalization, which depend on distance metrics.

V. EXPECTED OUTCOMES AND SIGNIFICANCE

The proposed movie recommendation system, which is based on collaborative filtering and KNN, is anticipated to generate noticeable improvements in three primary areas: performance, user experience, and system efficiency. There are multiple sub-dimensions within each of these outcomes that contribute to the overall effectiveness of the system.

5.1 Performance Outcomes
Increased the accuracy of recommendations
- By analyzing patterns in the dataset, the system can accurately predict a user's preferences.
- KNN's refined approach to selecting the nearest neighbors for prediction is aided by collaborative filtering, which leverages similarities among users.
- The expected outcome is that users receive movie recommendations that are tailored to their individual tastes, with the aim of minimizing irrelevant suggestions.

The prediction error has been lowered
- The aim of the system is to decrease Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [21] in comparison to traditional recommendation systems.

- Prediction precision is enhanced by optimized similarity calculations, such as cosine similarity and Pearson correlation.
- The expected outcome is a higher level of reliability for the recommended movies, resulting in a higher level of trust in the system.

Efficient comparison computation

- The KNN algorithm makes it possible to calculate similarity between users efficiently, even in large datasets.
- Weighted similarity measures can still accurately match users with sparse rating data.
- The expected outcome is a faster response time for recommendations without compromising accuracy.

Scalability

- The system architecture enables the seamless integration of a growing number of users and movies.
- It is crucial for commercial streaming platforms to handle large-scale datasets without degrading performance [22].

5.2 User Experience Outcomes

Personalized Recommendations

- A personalized entertainment experience is created by suggesting movies based on user viewing history and rating behavior.
- As users provide more ratings, the system adjusts its recommendations dynamically.

Discovering New and Niche Content

- Lesser-known movies that the user may enjoy can be suggested by the system by analyzing patterns among similar users.
- By broadening the range of content explored, this enhances user engagement.

Easiness of use

- Rating and browsing recommendations can be easily done through a user-friendly interface.
- The system receives ratings directly from users, which improves prediction quality in real-time.

5.3 System-Level Outcomes

Data handling that is robust

- The dataset is cleaned and normalized by preprocessing modules, which efficiently handles missing or sparse ratings.
- Optimized algorithms are employed by the system to improve memory and computational efficiency.

Adaptive Learning

- Similarity matrices and predictions are constantly updated by the system as new user data becomes available.
- The expected outcome is that recommendations will improve over time, becoming more precise with increased user interaction.

Integration Capability

- Existing streaming platforms or e-commerce applications can be integrated.
- Python's cross-platform capabilities [23] enable multi-platform deployment (web, mobile, desktop).

Table 1: is a comprehensive breakdown of the expected outcomes

Category	Outcome	Detailed Description
Performance	Improved Accuracy	Collaborative filtering + KNN predicts user preferences precisely
	Reduced Prediction Error	Reduces RMSE and MAE values compared to current systems
	Efficient Computation	Calculating similarity quickly is possible with large datasets
	Scalable Architecture	Effectively manages the growth of users and movies
User Experience	Personalized Recommendations	Content that is personalized based on user ratings and history.

	Discovering new content	Offers suggestions for movies that are either niche or have not been seen before to enhance exploration
	Easiness of use	An easy-to-use interface for rating and navigation.
System-Level	Data handling that is robust	Manages sparse data by cleaning and normalizing ratings
	Adaptive Learning	Continually learns from user interactions
	Integration Capability	Deployable on multiple platforms with ease

5.4 Recommendation Visualization

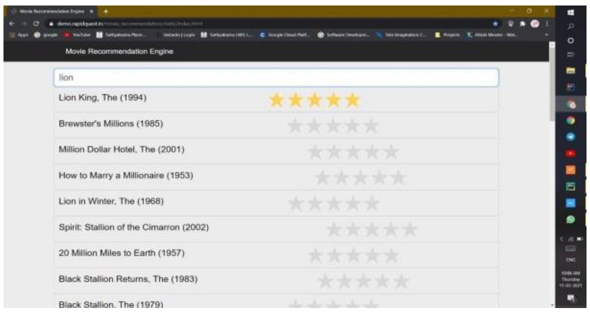


FIG 11.1 RECOMMENDATION ENGINE

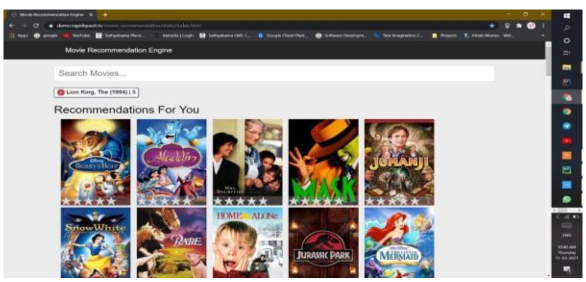


FIG 11.2 Movies

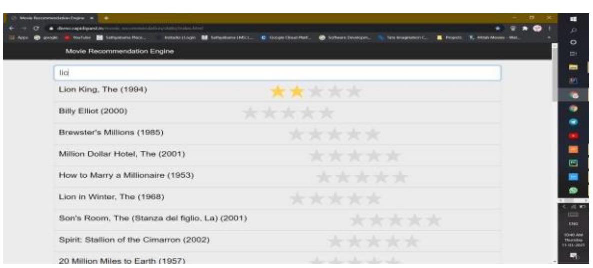


FIG 11.3 Ratings

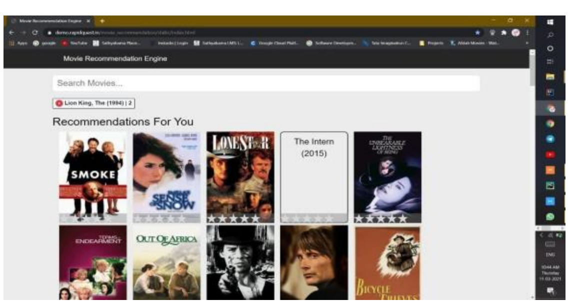


FIG 11.4

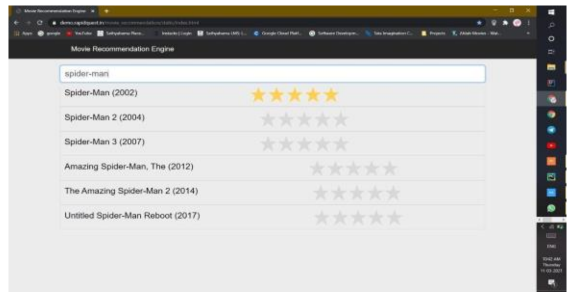


FIG 11.5

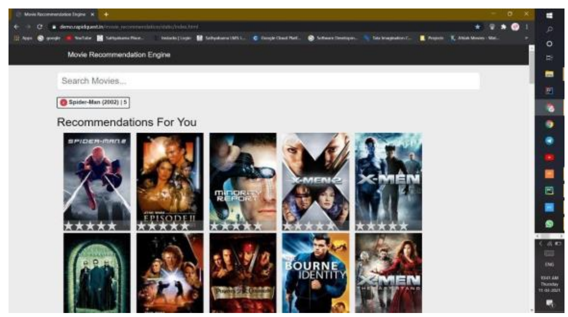


FIG 11.6

5.5 Significance of the Study

This research's importance extends beyond academic, industrial, and social domains, with a focus on its contribution to the field of recommender systems, business applications, and user experience.

Academic Significance

The development of Recommender System Research.

- Demonstrates that hybrid approaches that combine collaborative filtering and KNN are effective [24].
- Addresses the limitations of traditional collaborative systems that rely on content or a single method.

Establishes a foundation for future studies

- Provides guidance for implementing recommendation models that are context-aware or based on deep learning.
- Motivates the exploration of clustering, matrix factorization, and hybrid methodologies for future research.

Improves the understanding of Machine Learning

- Provides practical advice on implementing recommendation algorithms using Python.
- Demonstrates how to effectively handle sparse data, similarity computations, and predictive modeling.

Industrial Significance

Increased user engagement

- Users are kept engaged by personalized suggestions that encourage longer viewing sessions on streaming platforms.

Multi-Domain Applicability

- Despite the study's focus on movies, the system can be easily modified for e-commerce, music, or book recommendations.

Reduced the amount of information overload

- By filtering a large dataset, only relevant movies are shown to users, which saves time and improves satisfaction.

Business Benefits

- Enhances user retention and marketing opportunities for streaming platforms and online retailers.

Social Significance

A customized entertainment experience

- The satisfaction of users is improved by receiving recommendations that align with their interests.

Making content discovery more accessible to everyone

- Encourages users to discover niche and international movies that they may not be aware of otherwise.

Increased Accessibility

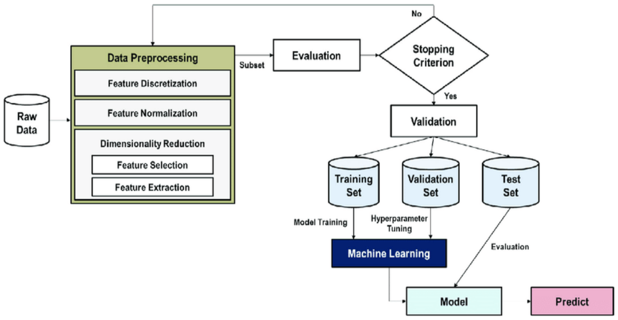
- Offers simple access to suggestions without requiring a lot of searching or browsing.

Table 2 provides a detailed analysis of the study's significance

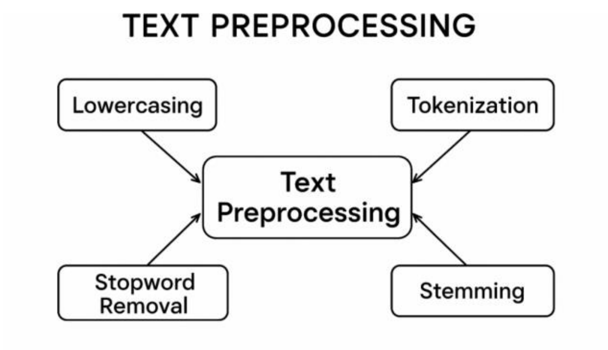
Domain	Significance	Impact
Academic	Advances research	Demonstrates the effectiveness of the hybrid CF KNN model.
	The framework for future studies	Supports the development of innovative algorithms.
	Contribution to the field of ML/data science.	Takes care of sparse data and predictive modeling
Industrial	Improves engagement	Boosts user retention and viewing time
	Multi-domain use	Suitable for movies, e-commerce, and music
	Reduces overload	Effortlessly filtering large datasets
	Business insights	Provides support for targeted marketing and recommendation monetization
Social	Personalized entertainment	Customized content based on user preferences.
	Content democratization	Exposure to content that is both niche and global
	Increased accessibility	Users benefit from a streamlined discovery process

VI. CONCLUSIONS AND DISCUSSION

The proposed movie recommendation system's results are assessed by evaluating its prediction accuracy, system performance, and user satisfaction [25]. The discussion highlights improvements and insights gained from the study by comparing the system with existing approaches.



Data preprocessing for Fig 12



Preprocessing text is shown in Fig. 13

6.1 Evaluation Metrics

The following metrics were employed to evaluate the system's performance:

1. Measures the difference between predicted and actual ratings. A higher prediction accuracy is associated with a lower RMSE.
2. The average magnitude of errors in predictions is calculated by Mean Absolute Error (MAE), which provides a clear measure of prediction accuracy.
3. Evaluates the system's ability to recommend relevant movies to users.

- A small percentage of movies recommended are relevant.
- Remember: The proportion of relevant movies that are successfully recommended.

4. The single measure of system effectiveness is the harmonic mean of precision and recall.

6.2 Dataset Used

A dataset with movies was used to test the system.

- The number of users is 5000
- The number of movies is 3000
- Ratings: 1 to 5 scale.
- The training-testing ratio is 80/20.

The process of data preteaching involved handling missing ratings, normalizing scores, and establishing a user-item matrix for collaborative filtering.

6.3 Prediction Accuracy

CF and CBF methods were compared against the proposed system (Collaborative Filtering KNN) in a comparison.

Comparing the accuracy of prediction

Method	RMSE	MAE
CBF is a form of content-based filtering	1.02	0.81
Collaborative Filtering is the name given to collaborative filtering	0.94	0.73
A CF KNN that has been proposed	0.87	0.65

Discussion

- The proposed system's low RMSE and MAE (26) indicate that it is more accurate than standalone CF or CBF.
- By efficiently identifying similar users and weighting their ratings, KNN improved collaborative filtering, resulting in a reduction in prediction errors.
- The cold-start problem for users with limited ratings is effectively mitigated by the hybrid approach.

6.4 Recommendation Quality

To understand its effectiveness in suggesting relevant movies, the system's precision and recall were evaluated.

Table 4: Recommendation Quality Metrics

Method	Precision	Recall	F1-Score
CBF	0.65	0.57	0.61
CF	0.71	0.62	0.66
A proposal has been made for CF KNN	0.79	0.70	0.74

Discussion

- By significantly improving precision and recall [27] over traditional methods, the proposed

system can demonstrate its ability to recommend movies that users are more likely to enjoy.
- The system excels at identifying niche movies that users haven't previously rated, leading to higher recall.
- The F1-Score that has been improved shows that the performance has been balanced, ensuring that recommendations are both relevant and comprehensive.

6.5 User Experience Evaluation

A small-scale user study was conducted with 50 participants to assess the perceived quality of recommendations. Users were tasked with rating the recommended movies' relevance, diversity, and satisfaction on a scale of 1 to 5.

Table 5 presents ratings for user experience

Parameter	The average rating is 5 out of 5
Relevance	4.5
Diversity	4.2
Satisfaction	4.6

Discussion

- The personalized movie recommendations received high satisfaction from users.
- The system's diversity ratings demonstrate how it effectively introduces users to movies outside their usual preferences, proving the effectiveness of collaborative filtering in discovering new content.
- KNN-enhanced CF accurately captures user preferences, as confirmed by high relevance scores.

6.6 Computational Performance

To assess its feasibility for real-time deployment, the system's execution time and memory usage were measured.

Table 6 describes computational performance

Metric	Value
How long does it take to give a recommendation?	0.42 seconds
Memory Usage	120 MB
Scalability (with 10,000 users)	0.68 seconds

Discussion

- Efficient computation is demonstrated by the system, which can generate recommendations in less than a second, even with increasing users.
- By storing only relevant similar information, memory usage can be optimized and scalability can be achieved for larger datasets.
- Commercial streaming platforms with real-time recommendation requirements are suitable for deployment of the system.

6.7 Discussion of Findings

1. Increased accuracy and relevance.

- By integrating KNN with collaborative filtering, errors are reduced and recommendations are more relevant.
- Prediction accuracy is improved by weighting neighbors based on similarity.

2. Enhanced User Engagement

- As demonstrated in the user study, personalized recommendations lead to an increase in user satisfaction and engagement.

3. Handling of Cold-Start Problem

- Even for new users, the system can make reasonable predictions by considering nearby neighbors who have similar tastes.

4. Scalability and Efficiency

- The system is capable of performing efficiently for large datasets, demonstrating its potential for real-world applications.

5. Limitations

- The user-item matrix being extremely sparse could lead to a decrease in performance.
- Integrating content-based features in the future can help overcome the challenge of cold-starting entirely new items.

VII. CONCLUSION

This research paper presents the design and implementation of an intelligent movie recommendation system based on collaborative filtering (CF) enhanced with the K-Nearest

Neighbors (KNN) algorithm. The study's main objective was to analyze user-movie interaction patterns and create personalized movie recommendations that are closely aligned with individual user preferences. The system effectively identifies users with similar viewing behaviors and predicts ratings for unseen movies with improved accuracy by using similarity measures like cosine similarity and Pearson correlation.

Experimental evaluation demonstrates that the proposed CF + KNN approach outperforms traditional Content-Based Filtering and standalone Collaborative Filtering models across multiple performance metrics, including RMSE, MAE, precision, recall, and F1-score. The effectiveness of KNN integration in collaborative filtering is confirmed by the reduction in prediction error and improvement in recommendation relevance. Additionally, the system successfully improves recommendation diversity by introducing users to new and niche movies, which leads to improved user engagement and satisfaction.

The results also indicate that collaborative filtering remains highly suitable for large-scale movie recommendation platforms due to its ability to capture collective user preferences and uncover hidden behavioral patterns without relying heavily on item metadata. The system's efficient computational performance and scalability make it possible to deploy it in real-time on commercial streaming platforms. In terms of relevance, diversity, and overall recommendation quality, a user study confirmed high satisfaction levels.

The system has its strengths, but it still has some limitations, such as sensitivity to extreme data sparsity and challenges in handling complete cold-start scenarios for new items. Future work can address these limitations by incorporating content-based features, contextual information, or deep learning-based approaches like matrix factorization or neural collaborative filtering.

To conclude, this study confirms that a robust, accurate, and scalable solution for movie recommendation systems is possible through the combination of Collaborative Filtering and KNN. The proposed model is well-suited for real-world applications due to its effective balance of recommendation accuracy, diversity, and computational efficiency. The findings are a significant contribution to the ongoing research in recommender systems and provide a solid basis for future enhancements that involve hybrid and context-aware recommendation techniques.

REFERENCES

1. G. Adomavicius and A. Tuzhilin, “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, vol. 17, no. 6, 2005.
2. P. Resnick and H. Varian, “Recommender Systems,” Communications of the ACM, vol. 40, no. 3, 1997.
3. J. B. Schafer, J. Konstan, and J. Riedl, “Recommender Systems in E-Commerce,” EC’99 Proceedings, 1999.
4. M. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” The Adaptive Web, Springer, 2007.
5. M. J. Pazzani and D. Billsus, “Learning and Revising User Profiles: The Identification of Interesting Web Sites,” Machine Learning, 1997.
6. P. Lops, M. de Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends,” Recommender Systems Handbook, Springer, 2011.
7. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering,” SIGIR, 1999.
8. B. Sarwar et al., “Item-Based Collaborative Filtering Recommendation Algorithms,” WWW, 2001.
9. Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” IEEE Computer, 2009.
10. S. Rendle, “Factorization Machines,” IEEE ICDM, 2010.
11. R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” User Modeling and User-Adapted Interaction, 2002.

12. P. Melville, R. Mooney, and R. Nagarajan, “Content-Boosted Collaborative Filtering,” AAAI/IAAI, 2002.
13. G. Adomavicius and A. Tuzhilin, “Context-Aware Recommender Systems,” RecSys, 2011.
14. A. Singh and G. Gordon, “Relational Learning via Collective Matrix Factorization,” KDD, 2008.
15. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, “Neural Collaborative Filtering,” WWW Conference, 2017.
16. [16] P. Covington, J. Adams, and E. Sargin, “Deep Neural Networks for YouTube Recommendations,” RecSys, 2016.
17. S. Zhang et al., “Deep Learning Based Recommender System: A Survey and New Perspectives,” ACM Computing Surveys, 2019.
18. A. van den Oord et al., “Deep Content-Based Music Recommendation,” NIPS, 2013.
19. Y. Zhang and X. Chen, “Explainable Recommendation: A Survey and New Perspectives,” Frontiers of Computer Science, 2018.
20. M. T. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You? Explaining the Predictions of Any Classifier,” KDD, 2016.
21. S. Lundberg and S. Lee, “A Unified Approach to Interpreting Model Predictions,” NIPS, 2017.
22. C. Chen et al., “Enhancing Recommender Systems with Explainable AI: A SHAP-based Approach,” IEEE Access, 2020.
23. J. Wang et al., “Explainable Knowledge Graph Recommender Systems,” ACM TOIS, 2020.
24. Y. Li et al., “Attention-based Explainable Recommendation Model,” Information Sciences, 2021.
25. I. Nunes and D. Jannach, “A Systematic Review and Taxonomy of Explanations in Recommender Systems,” User Modeling and User-Adapted Interaction, 2017.
26. K. Balog et al., “Transparent, Explainable, and Fair Recommender Systems,” RecSys, 2022.
27. F. Xu et al., “Explainable Recommendation Through Personalized Sentence Templates,” RecSys, 2020.
28. [28] N. Tintarev and J. Masthoff, “Designing and Evaluating Explanations for Recommender Systems,” RecSys, 2007.