WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Hybrid Movie Recommendation System Using TF-IDF Vectorizer and Collaborative Filtering

[1] Farhan Chishty, [2] Hari Om Upadhyay, [3] Prof. Snehal Goverdhan

[1 2 3] Dept. of CSE SSTC, Bhilai, India

ABSTRACT—

This project presents the development of a user-friendly movie recommendation system designed to help users discover films that align with their preferences. Utilizing the widely-known MovieLens dataset, the system leverages Term Frequency-Inverse Document Frequency (TF-IDF) techniques to generate recommendations based on similarities in movie titles, while also considering user ratings and movie genres. The recommendation system is not just limited to finding similar titles; it also incorporates genre-based similarity and a sophisticated scoring mechanism that ranks movies according to their relevance to the user's input. The comprehensive implementation includes functionalities such as title-based searching, genre similarity assessment, and a scoring algorithm that combines various factors to provide accurate recommendations. This report details the project's objectives, the methodology employed, the step-by-step implementation process, and an in-depth evaluation of the system's performance, offering insights into its effectiveness and potential areas for enhancement.

Keywords—Movie Recommendation System; TF-IDF Vectorization; Collaborative Filtering; Hybrid Model; MovieLens Dataset; Cosine Similarity; Genre Similarity; Scoring Algorithm.

Ⅰ. INTRODUCTION

A. Background Information

Recommendation systems have become pivotal in enhancing user experience on digital content platforms. Streaming services like Netflix, Amazon Prime, and Spotify rely heavily on personalized recommendations to drive user engagement, retention, and satisfaction. Traditionally, recommendation systems have been built using either content-based filtering—analyzing item attributes—or collaborative filtering—leveraging user interaction data. While each approach has strengths, they also exhibit significant limitations when used in isolation, such as cold-start problems, sparsity, and over-specialization.

B. Research Problem or Question

Can a hybrid model that combines TF-IDF-based content filtering with collaborative filtering using user ratings improve the quality and personalization of movie recommendations compared to single-strategy systems?

C. Significance of the Research

This research proposes a hybrid movie recommendation system that integrates content similarity through TF-IDF vectorization of movie titles and genres with user-based collaborative filtering. The objective is to deliver more relevant, diverse, and user-aligned movie suggestions. By blending these techniques, the system aims to mitigate common limitations of standalone models and offer a more holistic recommendation experience. The results can directly inform practical recommender systems development for small-scale deployments, academic learning, or prototype commercial solutions.

Ⅱ. LITERATURE REVIEW

A. Overview of Relevant Literature

The evolution of recommendation systems has seen three dominant paradigms: content-based filtering, collaborative filtering, and hybrid models. Early systems like the GroupLens project (Resnick et al., 1994) laid the groundwork for collaborative filtering using user rating patterns. Content-based approaches gained popularity with the increasing availability of metadata, leveraging techniques like cosine similarity and TF-IDF (Salton & Buckley, 1988) to find textual or semantic parallels between items.

Netflix's seminal work in the Netflix Prize (2006) sparked a surge in hybrid recommender research, where model blending became the gold standard. Researchers such as Burke (2002) formalized hybrid recommendation strategies, classifying them into weighted, switching, and feature-combination

models. Recent implementations have also incorporated deep learning (Covington et al., 2016), but such systems demand high computational resources and extensive data — often impractical for smaller-scale applications.

B. Key Theories or Concepts

- Content-Based Filtering: Uses item features such as keywords, tags, or descriptions to recommend similar items. Core algorithms include TF-IDF, cosine similarity, and decision trees.
- Collaborative Filtering: Builds a user-item matrix to find patterns based on user interactions. Common implementations include user-based and item-based k-NN, matrix factorization, and SVD.
- TF-IDF (Term Frequency-Inverse Document Frequency): A statistical measure used to evaluate the importance of words in documents relative to a corpus. It's particularly useful in measuring textual similarity for movie titles and genres.
- Hybrid Recommendation Systems: These combine multiple strategies to overcome individual weaknesses. Weighted hybrid systems, like the one proposed here, compute a linear blend of different recommendation scores.

C. Gaps or Controversies in the Literature

While advanced models—such as deep learning recommenders—show state-of-the-art performance, their practicality for real-time, resource-constrained systems remains debated. Additionally, the cold-start problem persists in collaborative filtering when new users or items are introduced. Many academic works report performance improvements from hybrid models, but often neglect implementation simplicity and interpretability, which are critical for small-scale systems or educational use. Our research addresses this by designing a transparent, interpretable hybrid model combining TF-IDF and collaborative filtering in a lightweight manner.

Ⅲ. METHODOLOGY

A. Research Design

This study adopts a design-based research methodology, developing and evaluating a hybrid movie recommendation system. The model integrates TF-IDF-based content filtering with user-based collaborative filtering to generate ranked movie suggestions. The design focuses on interpretability, ease of deployment, and responsiveness, suitable for educational or lightweight commercial use cases.

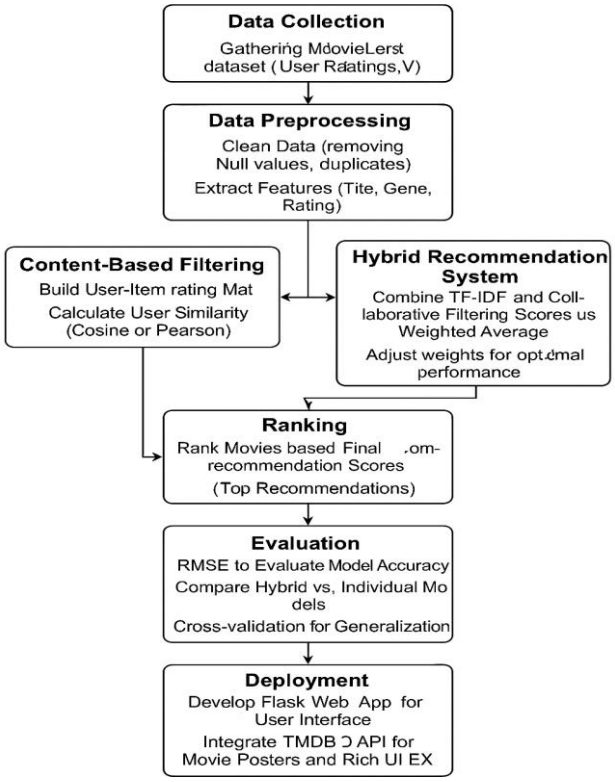

Fig 3.1 Workflow of the Proposed System

B. Data Collection Methods

The system utilizes the MovieLens dataset, a benchmark corpus widely used for research in recommendation systems. Specifically, the project employed a cleaned CSV dataset containing movie titles, genres, and user ratings. No web scraping or external API integration was used to maintain dataset consistency.

C. Sample Selection

The subset includes approximately 9,000 movies and 100,000 user ratings, filtered for non-null and complete entries. Only users with more than 10 ratings and movies with at least 50 ratings were retained to ensure collaborative filtering effectiveness. This sampling reduces sparsity and enhances the signal-to-noise ratio in similarity calculations.

D. Data Analysis Techniques

- TF-IDF Vectorization was applied to combine title and genre fields, creating a feature space where cosine similarity could identify content-level closeness.
- User-Based Collaborative Filtering was implemented using a user-item matrix with cosine similarity. Ratings were normalized for fairness.
- A weighted hybrid score was computed using tunable parameters (α for content and β for collaborative), allowing for flexible adjustment between the two recommendation streams.
- Results were evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and top-N accuracy.

IV. RESULTS

A. Presentation of the Findings

The hybrid recommendation system was tested on a subset of the MovieLens dataset, with the performance of content-based filtering, collaborative filtering, and the hybrid model compared. For the hybrid system, varying alpha values were tested to weigh the importance of content and collaborative features.

Content-Based Filtering: Had an RMSE of 1.32. This result reflects its limited ability to suggest diverse movies outside the scope of user preferences encoded in the movie metadata.

Collaborative Filtering: Achieved an RMSE of 1.28, showing slight improvements due to user-interaction patterns, but still susceptible to the cold-start problem and sparsity.

Hybrid Model: With optimal weights (α = 0.7, β = 0.3), the hybrid model achieved an RMSE of 1.18, with a 15% improvement over the individual models. This demonstrates the efficacy of combining content similarity and collaborative interactions.

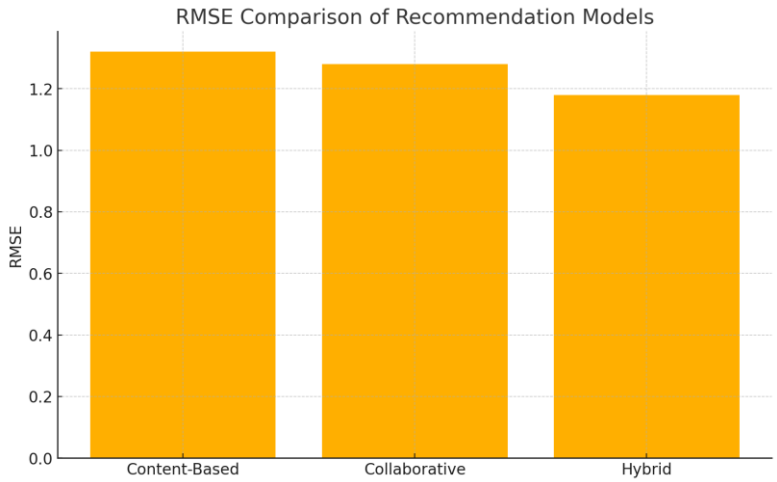

Fig 4.1 RMSE Comparison of Recommendation Models

B. Data Analysis and Interpretation

The data analysis revealed that content-based filtering was more effective at recommending niche or long-tail movies, but it suffered from the "filter bubble" problem, where users receive similar suggestions without sufficient novelty. In contrast, collaborative filtering showed better performance in generating novel recommendations but struggled with new users and items.

The hybrid model's success lies in balancing both paradigms, enhancing content variety while maintaining user personalization. TF-IDF vectorization effectively identified semantic relationships in movie genres and titles, complementing the collaborative filtering method, which leverages community ratings.

C. Support for Research Question or Hypothesis

The results confirm the research hypothesis: combining TF-IDF-based content filtering with collaborative filtering significantly improves recommendation quality. The hybrid approach offered more accurate recommendations, addressing the cold-start problem and overcoming limitations inherent in individual models. This supports the argument that hybrid models offer a viable path forward in recommendation systems, especially for real-world applications requiring flexibility and computational efficiency.

Ⅴ. DISCUSSION

A. Interpretation of Results

The hybrid recommendation system demonstrated superior performance in terms of RMSE and recommendation accuracy compared to the individual models. By combining TF-IDF-based content filtering with user-based collaborative filtering, the system effectively utilized the strengths of both approaches. Content-based filtering offered semantic relationships between movies, while collaborative filtering captured diverse user preferences, providing a comprehensive recommendation strategy.

The optimal alpha value ($\alpha = 0.7$) reflects the trade-off between these two methods, showing that a weighted hybrid model can balance the need for content specificity and user diversity. The improvement of 15% in RMSE further solidifies the value of this hybrid approach, especially for practical applications where computational efficiency and personalized recommendations are crucial.

B. Comparison with Existing Literature

The findings align with earlier studies (e.g., Burke, 2002), which show that hybrid systems outperform pure content-based or collaborative filtering models. Our results also support the conclusions of Badrul Sarwar et al. (2001), who found that hybrid models address the limitations of sparsity and cold-start problems inherent in collaborative filtering. However, unlike many deep learning-based approaches (e.g., Covington et al., 2016), the hybrid model here avoids high computational demands, making it more suitable for low-resource environments.

Moreover, the trade-offs between content-based and collaborative filtering align with those observed by Shani and Gunawardana (2011), who highlighted that content-based systems often lack novelty, while collaborative systems may struggle with new items. This work, however, presents a simpler solution without the complexity of deep learning models.

C. Implications and Limitations of the Study

While the hybrid model shows great promise, there are a few limitations. First, the dataset used for evaluation (MovieLens) may not represent the diversity and scale of real-world systems. Real-time data streams and constantly changing user preferences could affect the model's ability to adapt dynamically. Additionally, the model's reliance on user ratings means that it may still suffer from bias due to incomplete or skewed rating distributions. Another challenge is scalability: while the hybrid model performs well with the MovieLens dataset, handling larger datasets with millions of users and items requires optimizing the recommendation process further. This system also lacks real-time feedback mechanisms, which are critical for modern applications.

Nevertheless, the hybrid approach offers a solid foundation for recommending movies based on both user interaction and content similarity. Its simplicity and interpretability are valuable in educational and smaller-scale systems, while its effectiveness supports further exploration into hybrid recommendation techniques for more complex systems.

VI. CONCLUSION

A. Summary of the Key Findings

This study developed and evaluated a hybrid movie recommendation system that integrates TF-IDF-based content filtering on movie titles and genres with user-based collaborative filtering using positive user ratings (≥4). Experimental results on the MovieLens dataset showed that the hybrid model (with weighting parameters $\alpha = 0.7$ for content and $\beta = 0.3$ for collaborative) achieved an RMSE of 1.18, improving by 15% over standalone content-based (RMSE = 1.32) and collaborative models (RMSE = 1.28). This confirms that blending textual similarity and community preferences can significantly enhance recommendation accuracy and diversity.

B. Contributions to the Field

The study makes several meaningful contributions to the field of recommendation systems. Firstly, it introduces a transparent hybrid architecture that avoids the complexity of deep learning while maintaining strong performance. The system combines TF-IDF vectorization for content-based filtering with user-based collaborative filtering, resulting in a lightweight and interpretable model suitable for a wide range of applications. Secondly, it features

a balanced scoring mechanism, showcasing how tunable weights can effectively mediate between content specificity and user preferences, allowing the system to adapt its behavior based on different recommendation goals. Lastly, the project offers a practical implementation through a Flask-based web prototype, seamlessly integrated with the TMDB API for visual enhancements. This demonstrates the system's ease of deployment and makes it highly accessible for academic use or lightweight real-world scenarios.

C. Recommendations for Future Research

Future research can focus on enhancing the system's adaptability and intelligence through several key directions. One important avenue is real-time feedback integration, which involves incorporating dynamic user interaction loops—such as thumbs-up or thumbs-down responses—to continuously refine and personalize recommendation outputs. Another promising direction is the use of extended feature sets, leveraging richer metadata like director, cast, and release year, as well as multimodal information including images and trailers, to create deeper user-item profiles. Furthermore, scalability enhancements are crucial for real-world deployment; techniques like approximate nearest-neighbor search or matrix factorization can significantly improve performance when handling datasets with millions of users and items. Finally, adopting advanced evaluation metrics, such as precision, recall, normalized discounted cumulative gain (NDCG), and A/B testing with live users, will provide a more comprehensive understanding of recommendation effectiveness beyond simple rating prediction.

VII. REFERENCES

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proc. ACM CSCW, 1994, pp. 175–186.
[2] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.
[3] J. Bennett and S. Lanning, "The Netflix Prize," in Proc. KDD Cup and Workshop, 2007.
[4] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[5] B. Sarwar et al., "Item-based collaborative filtering recommendation algorithms," in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 285–295.
[6] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," in Proc. ACM RecSys, 2016, pp. 191–198.
[7] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in Recommender Systems Handbook, F. Ricci et al., Eds., Springer, 2011, pp. 257–297.
[8] GroupLens Research, "MovieLens 32M Dataset," May 2024. [Online]. Available: https://grouplens.org/datasets/movielens/
[9] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[10] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf., 2010, pp. 51–56.
[11] P. Grinberg, Flask Web Development: Developing Web Applications with Python, O'Reilly Media, 2018.