

BOOK RECOMMENDATION SYSTEM X | BOOK RECOMMENDATION SYSTEM X | BOOK RECOMMENDATION SYSTEM X +

127.0.0.1:5500/index.html Summarize Chat

GIS SCIENCE JOURNAL • Volume 25, Issue 4, 2025 • ISSN: 1469-9349 • Page 537

## BOOK RECOMMENDATION SYSTEM USING MACHINE LEARNING

Mr E. Sunkar<sup>1</sup>, G. Karthikayur<sup>2</sup>, S. K. Preehan<sup>1</sup>

<sup>1</sup>Assistant Professor, <sup>2</sup>Student, Department of Computer Science & Engineering, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Jeerapet, Kanchipuram

---

### ABSTRACT

In the digital age, navigating vast book collections can be overwhelming for readers seeking personalized recommendations. A book recommendation system leveraging pre-trained models and item-based collaborative filtering enhances the user experience by providing relevant book suggestions. Developed as a Flask-based web application, the system efficiently suggests books based on user preferences and reading selections.

The recommendation algorithm employs cosine similarity to identify books with similar attributes, ensuring that suggestions align with individual reading interests. The system features three primary functionalities: displaying popular books, generating personalized recommendations, and providing access to the complete book catalog. By utilizing precomputed similarity scores, recommendations are delivered in real time, optimizing both speed and accuracy.

With a scalable and deployable design, the system simplifies book discovery and fosters greater reading engagement by helping users explore relevant literary content effortlessly. Through intelligent filtering and adaptive learning, this solution contributes to the advancement of personalized digital libraries, making book selection more intuitive and enjoyable.

**Index Terms:** Book recommendation system, collaborative filtering, cosine similarity, digital book discovery, Flask web application, machine learning, personalized recommendations, pre-trained models, reading engagement, scalable system, item-based filtering, user preferences.

### INTRODUCTION

With the rapid expansion of digital libraries and online bookstores, finding the right book can be a challenging task for readers. Traditional methods of book discovery, such as browsing through categories or relying on bestseller lists, often fail to cater to individual preferences. To address this issue, personalized recommendation systems have emerged as an effective solution, offering users book suggestions tailored to their interests.

A book recommendation system utilizing pre-trained models and item-based collaborative filtering can generate personalized recommendations. Developed as a Flask-based web application, the system provides users with seamless access to trending books and customized reading suggestions. By leveraging cosine similarity, the algorithm identifies books with similar attributes, ensuring that recommendations align with user preferences.

The core functionality of the system revolves around three key features: Popular Book Display (showcasing books that are trending among users), Personalized Recommendations (suggesting books based on individual user interactions), and Complete Book Catalog (offering a structured collection of available books). To enhance efficiency, the recommendation process is optimized using precomputed similarity scores, enabling real-time suggestion generation and ensuring minimal latency.

By implementing a scalable and deployable architecture, the system significantly improves book discovery, making it easier for users to navigate vast collections. This solution not only enhances reading engagement but also contributes to the growing field of intelligent recommendation systems in digital libraries.

### ARCHITECTURE AND ALGORITHM

Earnings upcoming 21:44 13-02-2026

BOOK RECOMMENDATION SYSTEM X | BOOK RECOMMENDATION SYSTEM X | BOOK RECOMMENDATION SYSTEM X +

127.0.0.1:5500/index.html Summarize Chat

## ARCHITECTURE AND ALGORITHM

The book recommendation system follows a structured approach to analyze historical data, compute book similarities, and generate personalized recommendations. The algorithm consists of the following steps:

**Step 1: Data Collection and Preprocessing**

- Load the book dataset obtained from Kaggle using pandas.
- Clean the data by handling missing values, removing duplicates, and standardizing text format.
- Extract relevant features such as book title, authors, genres, and user ratings to build meaningful recommendations.

**Step 2: Feature Extraction and Similarity Computation**

- Convert book features into numerical representations suitable for similarity calculations.
- Use cosine similarity to measure the relationship between books based on their attributes.
- Construct a similarity matrix, where each book is compared with every other book to determine closeness in terms of content and user preferences.

**Step 3: Model Training and Storage**

- Compute and store precomputed similarity scores to optimize real-time recommendation generation.
- Save the trained similarity model using the pickle library for future use without reprocessing the data.

GDS Science Journal • Volume 25, Issue 4, 2025 • ISSN: 1069-9911 • Page 137

(GDS SCIENCE) JOURNAL • Volume 25, Issue 4, 2025 • ISSN: 1069-9911 • Page 538-539

**Step 4: Recommendation Generation**

- When a user inputs a book title, retrieve its corresponding similarity scores from the stored model.
- Identify the top N books with the highest similarity scores, ensuring recommendations align closely with user interests.
- Return the recommended books in descending order of similarity.

**Step 5: Deployment as a Web Application**

- Use Streamlit to create an intuitive and user-friendly web interface.
- Allow users to enter a book title and receive instant recommendations based on the precomputed similarity model.
- Display book details along with recommendations, enhancing user experience.

**Step 6: Scalability and Optimization**

- Optimize the recommendation process to handle large datasets efficiently.
- Implement caching mechanisms to reduce computational overhead and improve response times.
- Ensure the system is scalable to accommodate more users and books in the future.

## LITERATURE SURVEY

- Wang, J., & Li, X. (2023). Improving book recommendation accuracy through user behavior analysis. *Journal of Intelligent Systems*, 33(4), 567–580.
- Kumar, R., & Singh, P. (2022). Deep learning for personalized book recommendations. *International Journal of Data Science*, 21(2), 210–225.
- Chen, Y., & Zhao, L. (2011). Hybrid recommender systems for book recommendations. *Expert Systems with Applications*, 45(6), 13-2026

Type here to search Earnings upcoming 21:45 ENG 13-02-2026

- 3 Chen, Y., & Zhao, L. (2011). Hybrid recommender systems for book recommendations. *Expert Systems with Applications*, 45(6), 719–803.
- 4 Patel, S., & Gupta, M. (2021). Exploring user preferences in book recommendation systems using content similarity. *Computational Intelligence Review*, 30(3), 415–430.

## RESULTS

The performance of different recommendation techniques is summarized in terms of accuracy, as shown in the table below:

Algorithm	Type	Accuracy (%)
Cosine Similarity (Item-Based CF)	Memory-Based CF	82–85
User-Based Collaborative Filtering	Memory-Based CF	75–82
Content-Based Filtering	Feature-Based	80–88
Hybrid Filtering	Combination	85–90
Matrix Factorization (SVD)	Model-Based CF	83–92
Deep Learning (Neural CF)	Model-Based (Deep)	90–95

## CONCLUSION

This study explores various book recommendation techniques, addressing challenges such as cold start issues, computational complexity, and lack of diversity. The proposed system leverages pre-trained models, item-based collaborative filtering, and cosine similarity to provide accurate and efficient recommendations.

By precomputing similarity scores and deploying a user-friendly web application using Streamlit, the system ensures real-time responses and scalability. Future enhancements can focus on deep learning integration, improving recommendation diversity, and adapting to user preferences, thereby advancing personalized digital book discovery.

## REFERENCES

- 1 Y Wang, Y Zhang, and W Y Wang, "Personalized Recommendation for E-Commerce Using Collaborative Filtering and Deep Learning," *IEEE Access*, vol 9, pp. 45721–45732, 2021.
- 2 Z Liu, Q Liu, and J Zhang, "A Hybrid Recommendation System Using Collaborative Filtering and Content-Based Filtering for e-Commerce," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 1–11, Jan. 2021.
- 3 M S Ghaisas, S X Zhao, and R M Razani, "Exploring Hybrid Models for Recommender Systems: Challenges and Solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 1305–1318, May 2022.
- 4 S M S Islam, M S Amin, and T S A Iqbal, "Cold Start Problem in Collaborative Filtering: Approaches and Research Directions," *IEEE Access*, vol. 9, pp. 17381–17394, 2021.
- 5 H P Tang, X Xin, and C Z Xia, "A Survey on Hybrid Recommender Systems and Challenges," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 10, pp. 4239–4250, Oct. 2022.
- 6 D S N K Murthy, P M S S Reddy, and A K J Dev, "Improved Collaborative Filtering with Weighted Graph-Based Models," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 640–650, Jun. 2021.
- 7 P R Agarwal and A B Chawla, "Collaborative Filtering and Content-Based Hybrid Recommendation Systems for E-Learning Platforms," *IEEE Transactions on Learning Technologies*, vol. 14, no. 2, pp. 212–224, Apr. 2021.

© 2024 ScienceDirect • Volume 25, Issue 4, 5025–5040, pages 5028–5039