

MLPOWERED BOOK RECOMMENDATION SYSTEM

¹Mohit Sharma, ²Mohit Pradhan, ³Harsh Gupta, ⁴Abhishek Pandey, ⁵Dinesh Kumar, ⁶Ajay Jatav

¹Professor, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India.

^{2,3,4,5,6} UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India.

Article Information

Received : 28 March 2026
Revised : 29 March 2026
Accepted : 31 March 2026
Published : 1 March 2026

Corresponding Author:

Mohit Pradhan
Email: mohitog07@gmail.com

Abstract— The **ML-Powered Book Recommendation System** provides personalized book suggestions by analyzing user preferences and past interactions using content-based filtering. It evaluates features such as genre, author, keywords, and descriptions to identify patterns and recommend books aligned with user interests. This approach enhances the reading experience by simplifying the discovery of relevant content and encouraging exploration of new topics. The system reduces the effort of browsing large collections by automatically filtering suitable options. It incorporates key processes like data preprocessing, feature extraction, vectorization, and similarity measurement to generate accurate recommendations. Designed for simplicity and efficiency, the system demonstrates how machine learning can deliver meaningful, context-aware suggestions for modern digital reading platforms.

Keywords: *Book Recommendation System, Content-Based Filtering, Personalized Recommendations, Feature Extraction, Similarity Measurement.*

Copyright © 2026: Mohit Sharma, Mohit Pradhan, Harsh Gupta, Abhishek Pandey, Dinesh Kumar, Ajay Jatav, .This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

Citation: Mohit Sharma, Mohit Pradhan, Harsh Gupta, Abhishek Pandey, Dinesh Kumar, Ajay Jatav “ML POWERED BOOK RECOMMENDATION SYSTEM”, Journal of Science, Computing and Engineering Research, April 2026.

I INTRODUCTION

The rapid growth of digital reading platforms has transformed how readers discover and engage with books. With millions of titles available across genres, languages, and formats, traditional methods of selecting books have become overwhelming and inefficient. Readers often struggle to find content that matches their interests, leading to increased demand for intelligent systems that can analyze user preferences and deliver personalized recommendations. As digital libraries and online bookstores continue to expand, automated solutions are essential for improving content discovery and enhancing the overall reading experience.

The **ML-Powered Book Recommendation System** addresses this challenge by using machine learning techniques to suggest books based on user interests and reading history. It analyzes features such as genres, authors, descriptions, and keywords to identify patterns and recommend relevant content. By leveraging Natural Language Processing (NLP), the system gains deeper insights into textual data, enabling more accurate and meaningful suggestions. This reduces search time, improves recommendation quality, and helps users explore new themes and authors aligned with their preferences.

The system uses content-based filtering, ensuring recommendations are generated from book attributes rather

than relying solely on user data, making it effective even for new users. Techniques like TF-IDF vectorization and cosine similarity enhance accuracy and relevance. Designed with a flexible and scalable architecture, the system can be integrated into various platforms such as digital libraries and e-book applications. Overall, it provides an efficient, user-centric solution that improves content discovery, increases engagement, and makes reading more personalized and accessible.

II. PROBLEM STATEMENT

The rapid expansion of digital reading platforms has led to an overwhelming availability of books, making it difficult for readers to discover content that matches their preferences. Traditional methods such as manual browsing, bestseller lists, and word-of-mouth recommendations are no longer effective in handling large-scale digital libraries. This results in information overload, reduced user engagement, and difficulty in identifying relevant books.

A. Existing Problems in Digital Reading Platforms

Many existing platforms lack intelligent content discovery and rely heavily on manual search processes, which are time-consuming and inefficient. They often fail to utilize user

interaction data such as reading history, preferences, and behavior, leading to generic and repetitive recommendations. Additionally, traditional systems struggle with semantic understanding, resulting in poor matching between books and user interests, especially for niche or unique reading preferences.

B. Core Problem Statement

Moreover, current recommendation systems face challenges such as limited personalization, cold-start problems, low accuracy due to insufficient data, and poor scalability. These limitations highlight the need for an intelligent, machine learning-based solution that can analyze user preferences and book features effectively. Such a system is essential to provide accurate, personalized, and scalable recommendations, ultimately improving content discovery, user engagement, and the overall reading experience.

III. PROPOSED MODEL

The proposed work model outlines the complete workflow and architecture of the ML-powered book recommendation system, from user entry to personalized suggestions. It ensures a seamless and efficient experience through structured interaction between the frontend, backend, and machine learning modules. Users can securely log in, browse, or search for books, triggering backend processes that analyze data and generate recommendations. The system uses content-based filtering, applying TF-IDF vectorization and cosine similarity to evaluate book metadata such as titles, genres, and descriptions. This approach enables accurate, scalable, and context-aware recommendations, enhancing user experience by delivering relevant and meaningful book suggestions based on individual preferences.

A. User Opens Application:

This is the initial step where the user enters the book recommendation platform through a web interface. The homepage loads essential UI components such as login/register options, search bars, and recommended sections. This step marks the start of the user's interaction with the system.

B. User Selection (Login / Register):

Users can either log in (if existing users) or register as new users. The registration form captures essential details such as name, email, and password. This step ensures that user interactions can be tracked and personalized.

C. Select Option:

After authentication, users are presented with two main options: Browse Books and Get Recommendations, which define their path within the system. In the Browse Books option, users can explore the complete catalog without providing any input. They can view book titles, read descriptions, explore genres, and check author details and

metadata. This option supports manual discovery and allows users to explore content freely.

In the Get Recommendations option, the system shifts to an ML-driven pipeline where users enter a book title as input. This input acts as a query for the content-based filtering model. The system retrieves relevant metadata and preprocesses the data by removing stop-words, cleaning text, normalizing characters, and extracting meaningful features. This step ensures accurate feature extraction and prepares the data for further analysis.

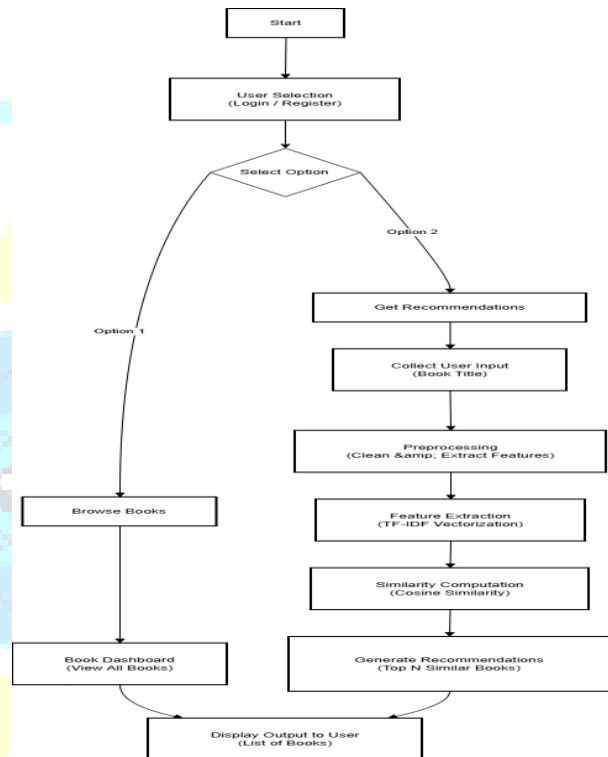


Figure 1: Proposed Work Model

The processed data is then converted into numerical vectors using TF-IDF vectorization, which highlights important terms in the book content. Cosine similarity is applied to compare the input book with others in the dataset and calculate similarity scores. Based on these scores, the system generates a list of top similar books. Finally, the recommendations are displayed on the interface with details such as title, author, genre, description, and relevance, completing the personalized recommendation process.

IV. TECH STACK

The proposed ML-based book recommendation system uses a modern, scalable technology stack designed for efficient performance, secure data handling, and seamless frontend-backend communication. It integrates machine learning with a responsive UI and robust backend to ensure accurate recommendations and real-time content-based filtering, making the system reliable, user-friendly, and efficient.

A. Frontend Technologies

React.js is the primary frontend framework used to build the user interface, offering a component-based architecture for reusable UI elements and efficient code management. Its virtual DOM improves performance by updating only necessary parts of the UI. In this project, React.js handles pages like Home, Search, and Top Books, while managing dynamic states such as user inputs and recommendations.

HTML5, CSS3, and JavaScript ensure responsive design, cross-browser compatibility, and interactive behavior. CSS along with Bootstrap or TailwindCSS is used for styling, providing responsive layouts, utility classes, and a clean, mobile-friendly interface. Additionally, Axios or Fetch API enables communication with the Flask backend by handling HTTP requests for searching books, sending queries, fetching top books, and retrieving recommendations, ensuring smooth data exchange.

B. Backend Technologies

The backend serves as the core engine of the system, handling processing tasks, API calls, computations, and overall workflow management. Built using Flask, a lightweight Python framework, it manages tasks such as dataset loading, preprocessing, TF-IDF vectorization, cosine similarity computation, and generating recommendations, while sending responses to the frontend. The system uses Python ML libraries like Scikit-learn, NumPy, Pandas, Pickle, and NLTK for efficient data processing and content-based filtering. Additionally, JWT (JSON Web Tokens) ensures secure, stateless authentication by enabling safe API access, role-based authorization, and smooth session management without server-side storage.

C. Database Technologies

MongoDB is used as the primary database for storing metadata due to its flexible, document-oriented structure, enabling efficient storage of user profiles, course data, modules, assignments, submissions, and enrollment records while supporting fast read-write operations and scalability. Additionally, the recommendation dataset is stored in CSV format, which is lightweight and easily processed by the Flask backend for tasks such as data cleaning, vectorization, and feature extraction. The dataset includes book titles, authors, genres, descriptions, and ratings, making it suitable for building and executing the content-based recommendation engine.

D. Additional Technologies

Additional technologies include Axios or Fetch API for seamless communication between frontend and backend services, enabling efficient data exchange. Mongoose is used as an ODM tool to interact with MongoDB through structured schemas and models, simplifying database operations. Additionally, trained ML models such as TF-IDF vectorizers and similarity matrices are stored locally in serialized format, ensuring faster loading and avoiding repeated computations, which improves overall system performance and efficiency.

Git is used for version control and GitHub for hosting the project repository, enabling collaboration, code backup, and version tracking. Postman or VS Code's Thunder Client is used for API testing to verify endpoints like /recommend/{id}, /search, and /top books before frontend integration. Visual Studio Code serves as the main development environment, offering useful extensions for Python, React, and MongoDB, along with a built-in terminal and Git integration for efficient development.

V. RESULT SCREENSHOTS

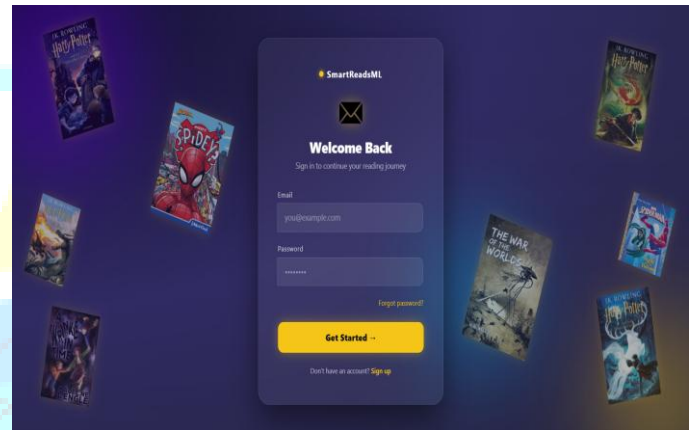


Figure 2: LOGIN PAGE

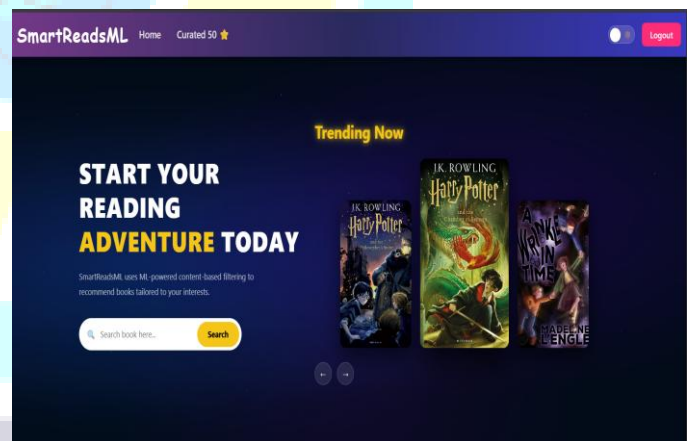


Figure 3: HOME PAGE

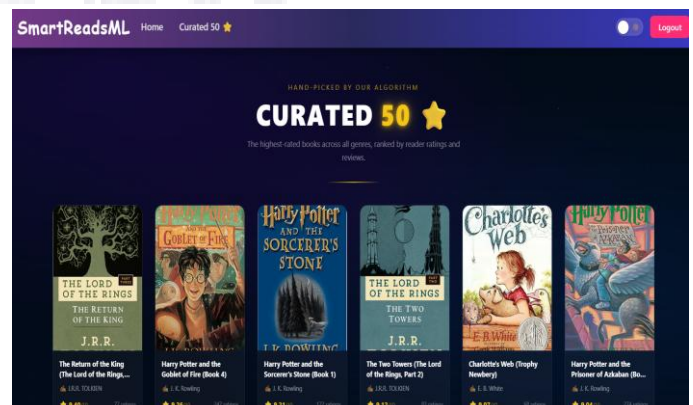


Figure 4: RANKINGS PAGE

VI. CONCLUSION

The ML-Powered Book Recommendation System successfully addresses the growing challenge of information overload in digital reading platforms by providing personalized and intelligent book suggestions. By utilizing **machine learning** techniques, particularly **content-based filtering**, the system analyzes book metadata such as titles, genres, authors, and descriptions to generate relevant recommendations. This approach enhances user experience by reducing search time and improving content discovery. The system demonstrates how data-driven solutions can transform traditional browsing methods into efficient and meaningful recommendation processes, making it highly suitable for modern digital libraries and reading platforms.

The system is designed with a scalable and modular architecture that integrates frontend, backend, and machine learning components effectively. Technologies such as React.js, Flask, and MongoDB ensure smooth communication, efficient data handling, and responsive user interaction. The implementation of TF-IDF vectorization and cosine similarity enables accurate similarity computation, ensuring high-quality recommendations. Additionally, preprocessing techniques improve data quality and model performance. The structured workflow—from user input to recommendation generation—ensures reliability, efficiency, and consistency, making the system adaptable for real-time applications and diverse user needs.

Furthermore, the project highlights the potential for future enhancements, including hybrid recommendation models, expanded datasets, and improved user personalization. Features such as user authentication, saved preferences, and integration with external APIs can further enhance system functionality and user engagement. The inclusion of advanced technologies like NLP and deep learning can improve semantic understanding and recommendation accuracy. Overall, this project demonstrates a practical and effective application of machine learning in solving real-world problems, offering a robust foundation for developing intelligent, scalable, and user-centric **recommendation systems** in the evolving digital ecosystem.

REFERENCES

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web (WWW '17), ACM Press, pp. 173–182.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, pp. 4171–4186.

[3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases

and their compositionality. Advances in Neural Information Processing Systems (NIPS), 26, pp. 3111–3119.

[4] • Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523. Elsevier.

[5] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1), 5–53.

[6] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. Proceedings of the 25th Annual International ACM SIGIR Conference, ACM Press, pp. 253–260.

[7] Wan, M., & McAuley, J. (2018). Item recommendation on monotonic behavior chains. Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18), ACM Press, pp. 86–94.

[8] Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. Proceedings of the 14th International Conference on World Wide Web (WWW '05), ACM Press, pp. 22–32.

[9] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 1532–1543.

[10] Wang, X., He, X., Cao, Y., Liu, M., & Chua, T. S. (2019). KGAT: Knowledge graph attention network for recommendation. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp. 950–958.