# AI-Powered Library Management and Book Recommendation System

Aditi Rathore[1], Adarsh Singh[2⋆], Tanmay Dubey[3], and Nishad G. Deshpande[4]

[1] Electronics and Communication Engineering,
Indian Institute of Information Technology, Surat, India
ui22ec03@iiitsurat.ac.in
[2] Electronics and Communication Engineering,
Indian Institute of Information Technology, Surat, India
ui22ec02@iiitsurat.ac.in
[3] Electronics and Communication Engineering,
Indian Institute of Information Technology, Surat, India
dubey.tanmay@iiitsurat.ac.in
[4] Applied Science and Humanities,
Indian Institute of Information Technology, Surat, India
nishad.deshpande@iiitsurat.ac.in

**Abstract.** This paper presents BookMitra, an intelligent library management system integrated with a hybrid book recommendation framework designed for academic environments. The system combines content-based filtering using Term Frequency–Inverse Document Frequency (TF–IDF) and cosine similarity with collaborative filtering based on matrix factorization to deliver personalized and context-aware book recommendations. The hybrid approach effectively addresses challenges such as data sparsity and cold-start scenarios while making updates with user evolving behaviour. Alongside recommendation intelligence, BookMitra also automates library operations such as catalog management, transactions, and role-based access control. Experimental evaluation demonstrates improved recommendation relevance highlighting the effectiveness of embedding hybrid recommender systems within modern digital library platforms.

**Keywords:** Library Management System · Hybrid Recommendation System · Collaborative Filtering · Content-Based Filtering · Book Recommendation

## 1 Introduction

Academic libraries play a key role in learning support, intellectual growth, and knowledge exchange within educational institutions. As academic content grows, we see from us that libraries must present as information centers which also offer smart support in the search for relevant material. At the same time many

⋆ Corresponding author


S. Sharma et al. (eds.), *Proceedings of the International Conference on Recent Advances in Intelligent and Sustainable Technologies (RAIST 2026)*, Atlantis Highlights in Intelligent Systems 17,
https://doi.org/10.2991/978-94-6239-707-1_27

of our library settings are still in the past they are stuck in old or only semi automated systems which mainly focus on manual book issue, static catalog search, and low level user interaction. Also without real time inventory tracking and personal recommendation features the libraries do a poor job of putting available resources to use which in turn leaves the users down.

Early library management systems (LMS) were mainly used to digitize administrative operations like cataloging, circulation, and member data maintenance. Some popular systems like Koha and Evergreen [1] significantly improved operational efficiency by automating repetitive administrative tasks. Although they were robust but these systems primarily followed transactional and rule-based paradigms that gave every user the same interaction experiences with no personalization, irrespective of their reading preferences or habits. As a result, user interaction remained largely passive and content exploration was restricted to keyword-based search techniques.

Research efforts have increasingly concentrated on incorporating recommendation systems into library platforms in order to improve user experience. By finding commonalities among user behavior patterns, collaborative filtering techniques uses user preferences. In early surveys we identified the advantages and disadvantages of item-based and user-based collaborative filtering strategies [2], which established the groundwork for recommender systems [3]. By capturing latent user preferences, matrix factorization models like Singular Value Decomposition enhanced this field [4].

At the same time, book metadata such as author, genre, and description is used in conjunction with content-based filtering techniques to recommend similar items to what the user has already accessed. Lops et al.'s state-of-the-art study provides a comprehensive overview of content-based recommender systems [5]. Modeling of semantic relations has been reported to improve accuracy of recommendations in recent studies of hybrid book recommendation systems [6].

The existing works, while progress has been made, continues to be centered on algorithms and evaluated on benchmark datasets, as opposed to being embedded in working library systems. Studies have identified a lack of real-world implementations and gaps in research where problems remain unsolved [7]. To counter such a phenomenon, hybrid systems combining collaborative and content-based techniques have been proposed [8] with more recent studies explicitly focusing on digital library environments and real-world deployment scenarios [13]. However, a good number of learning management systems (LMS) solutions prioritize simple web-based automation as opposed to sophisticated intelligent analytics [9].

Moreover, scalability, cloud-readiness, and secure user management are sometimes overlooked by traditional systems. Role-Based Access Control (RBAC) [10] needs to be incorporated in contemporary web applications to ensure safe and distinct system usage. Semantic-aware content-based recommender systems significantly enhanced personalization by taking into account a broader contextual understanding of the items [11]. Neural collaborative filtering models have also been explored to understand intricate patterns of user-item interactions [12].

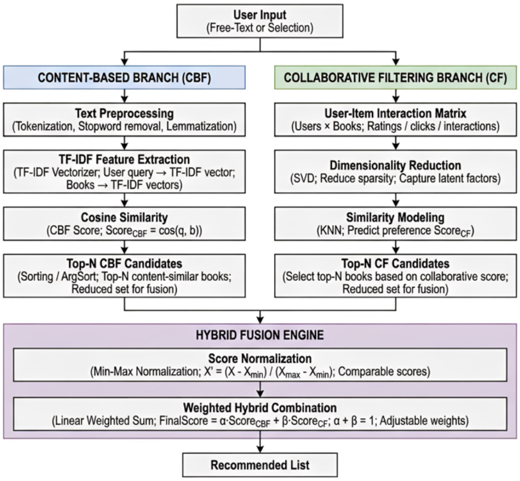


**Fig. 1.** Architectural workflow of the proposed hybrid recommendation model.

Each of these flaws highlights a crucial problem: customized material retrieval and innovative user engagement are frequently overlooked by conventional library management systems. Despite having robust cataloging and circulation management capabilities, systems like Koha[1] mostly adhere to transactional paradigms and lack adaptive, data-driven recommendation intelligence. In order to overcome cold-start, sparsity, and personalization issues in academic and digital library settings, recent surveys show a growing trend toward hybrid and learning-based book recommendation systems.

In this work, we introduce *BookMitra*, a cloud-ready library management system with a hybrid recommendation engine that creates customized recommendations based on user searches, borrowings, and returns.

## 2 System Methodology

The system is designed to support routine library operations while simultaneously enabling personalized book discovery based on user behavior and book content. The recommendation structure is very much a part of the library management process and which also grows from user interaction data is what this technique is all about. The system adopts a modular design which has a library management layer that takes care of data integrity and transactional consistency and a recommendation layer which is more of an analytics play that we designed to get insights out of the action data the system puts out.

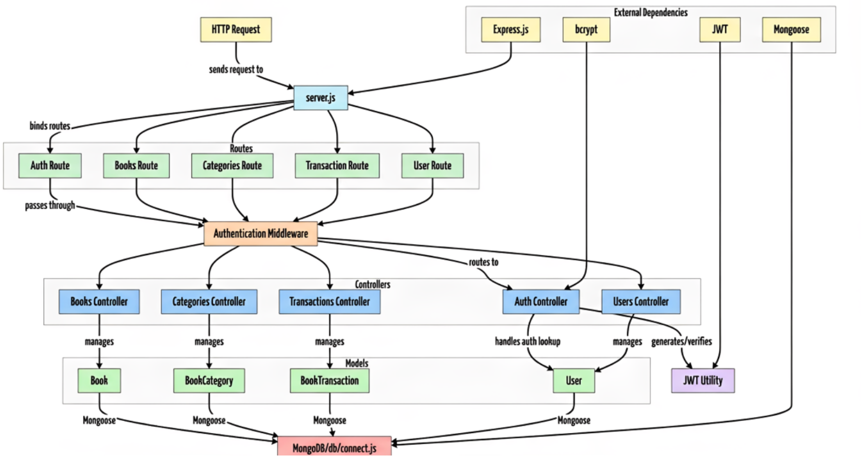


**Fig. 2.** Backend architecture of library management system.

## 2.1 Library Management System Architecture

The central component of the BookMitra platform is the Library Management System (LMS), which oversees user accounts, book catalog upkeep, and transaction processing, including issuance, return, renewal, and reservations. *server.js* is the foundation of the backend architecture, as shown in Fig. 2, and *Express.js* directs HTTP requests to useful modules like Books, Categories, Transactions, and Users. While book records hold structured metadata such as title, author, genre, ISBN, and descriptions, user profiles keep track of borrowing histories and activity logs.

Using *Mongoose*, all system interactions are permanently stored in a MongoDB database, providing a trustworthy source of behavioral data. After passing through an authentication middleware, requests reach the controller layer, where secure access is guaranteed by *bcrypt* and JWT-based authentication. The logged interaction data supports administrative operations and serves as input for collaborative recommendation modeling, while tight database integration ensures recommendations remain consistent with inventory status and evolving user behavior.

## 2.2 Content-Based Recommendation Model

To produce book recommendations, the proposed system makes use of a hybrid design with two main branches. The process starts with the Content-Based Branch (CBF), which uses book metadata kept by the library system to calculate semantic similarity between user preferences and available books, as shown in the system flow chart (Fig. 1).

Each book is illustrated as a text file containing a title, author, genre, and description. User preference predictions are based on search queries in the system interface or aggregated data from books that users have checked out. Preprocessing of user input begins with tokenization, stopword removal, and lemmatization. After that, text descriptions are converted into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF–IDF) model. The TF–IDF weight of a term $t$ in document $d$ is calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right) \tag{1}$$

where $N$ is the total number of documents in the corpus, $\text{DF}(t)$ indicates the number of documents containing term $t$, and $\text{TF}(t, d)$ indicates the frequency of term $t$ in document $d$. Each book is represented as a high-dimensional sparse vector, and the user query is projected into the same vector space using the learnt vocabulary, as shown in Fig. 1's TF-IDF Feature Extraction stage.

Cosine similarity is used to measure the degree of relevance between the user query and specific books. The content-based similarity score for a book $i$ is calculated as follows using the similarity modeling phase depicted in Fig. 1:

$$\text{Score}_{CBF}(i) = \cos(\mathbf{q}, \mathbf{b}_i) = \frac{\mathbf{q} \cdot \mathbf{b}_i}{\|\mathbf{q}\|\|\mathbf{b}_i\|} \tag{2}$$

We use a similarity score to rank books, which is then used to pick the top items, dubbed Top-N CBF Candidates, as seen in Fig. 1, when trying to find the best matches. Following the ranking process, this Top-N CBF component gives us a smaller, yet more targeted list of content-related books that are perfect for the final fusion in the Hybrid Fusion Engine. It is also helpful when we have very little user interaction history to go by.

### 2.3 Collaborative Recommendation Model

The collaborative recommendation component models user preferences based on past interactions captured by the library system, as seen in Fig. 1's Collaborative Filtering Branch (CF). Ratings, clicks, and other user-book interactions are arranged in a *User–Item Interaction Matrix*:

$$R \in \mathbb{R}^{U \times I} \tag{3}$$

where $I$ represents the number of books and $U$ represents the number of users. The interaction strength between user $u$ and book $i$ is represented by each item $R_{u,i}$. Direct similarity computation is frequently inadequate since library interaction data is inherently sparse.

Singular Value Decomposition (SVD) is used in *Dimensionality Reduction* to overcome this constraint and capture hidden factors. Users and books are projected onto a lower-dimensional latent factor space by decomposing the matrix:

$$R \approx U \Sigma V^T \tag{4}$$

Thematic interests and reading style are examples of latent preference variables that are captured by this decomposition but are not clearly visible from metadata alone. The system enhances generalization by decreasing sparsity. After this reduction, the algorithm predicts preference scores using similarity modeling, such as K-Nearest Neighbors. The preference score for an unseen book $i$ is predicted for a target user $u$ as follows:

$$\text{Score}_{CF}(i) = \hat{R}_{u,i} \tag{5}$$

where $\hat{R}_{u,i}$ denotes the estimated affinity derived from the latent space. This process yields the Top-N CF Candidates, which are then passed to the hybrid fusion engine for final ranking.

### 2.4 Hybrid Score-Level Fusion

The relevance scores produced by the collaborative and content-based components come from several statistical distributions. Min–max normalization is applied separately to both score sets to generate similar values in order to facilitate meaningful integration within the Hybrid Fusion Engine:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{6}$$

This transformation adjusts all scores to a common range $[0, 1]$, as seen in Fig. 1's *Score Normalization* stage. A Weighted Hybrid Combination approach is then used to determine the final recommendation score for each book $i$:

$$\text{FinalScore}(i) = \alpha \cdot \text{Score}_{CBF}(i) + \beta \cdot \text{Score}_{CF}(i) \tag{7}$$

This linear weighted sum is subject to the following constraints for adjustable weights:

$$\alpha + \beta = 1, \quad \text{where } \alpha, \beta \geq 0 \tag{8}$$

The system effectively balances collaborative preference signals from the CF branch and semantic relevance from the CBF branch by employing this hybrid formulation. This lessens the drawbacks of individual filtering techniques by enabling the final Recommended List to dynamically adjust as user interaction history changes.

### 2.5 Algorithmic Workflow of the Proposed System

**Input:** Book metadata $B$, user interaction logs $U$, fusion weights $(\alpha, \beta)$ **Output:** Ranked Top-$N$ book recommendations

1. Extract textual metadata from $B$.
2. Apply text preprocessing (tokenization, stopword removal, lemmatization).
3. Generate TF–IDF vectors for books and user queries.

4. Compute cosine similarity to obtain content-based scores.
5. Construct user–item interaction matrix from $U$.
6. Apply SVD to obtain latent user and item factors.
7. Predict collaborative filtering scores.
8. Normalize CBF and CF scores using min–max scaling.
9. Compute final scores using weighted fusion.
10. Rank items and return Top-$N$ recommendations.

## 3 Experimental Setup

To assess the effectiveness of the integrated library management system and hybrid recommendation system, we needed a real-world situation. Because it captures user-item interactions similar to borrowing events and maps rich book metadata to catalog records, the Amazon Books dataset was chosen as an appropriate proxy for large-scale academic libraries. The dataset has a skewed rating distribution, with most feedback falling into the 4–5 star range, as seen in Fig. 3. Sparse negative feedback and long-tail user activity are examples of natural outliers brought about by this imbalance, also known as positivity bias. In order to evaluate recommendation behavior when there are few negative signals available, the suggested system is built to function robustly in such circumstances rather than filtering these outliers.

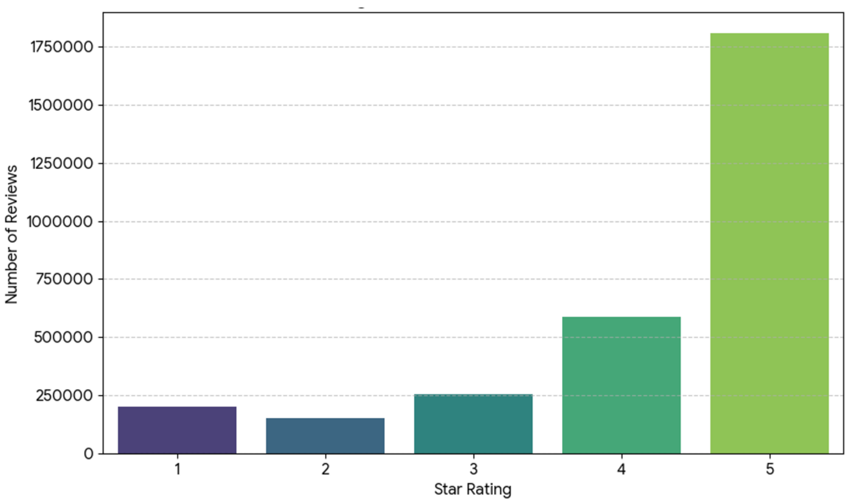


**Fig. 3.** Rating frequency distribution for Amazon dataset.

The collaborative filtering component uses latent factor modeling via Singular Value Decomposition to reduce noise from rare or extreme interaction patterns,

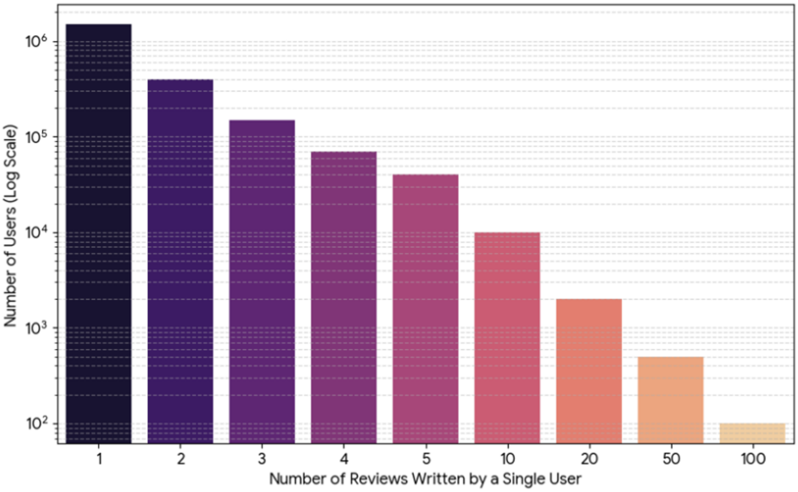


**Fig. 4.** Graphical representation of number of reviews written by a single user.

thereby mitigating the impact of outliers resulting from uneven user interactions. In order to avoid disproportionately high similarity scores controlling the final recommendations, min–max normalization is also used during score-level fusion. The data is divided into training and test sets using the user-level hold-out method, and content-based and collaborative models are trained separately. Throughout the analysis, hybrid weighting parameters are changed to look at each component's relative contribution.

## 4 Results and Discussion

The performance of BookMitra under actual library usage and recommendation scenarios is assessed in this section. Secure role-based access control and real-time data synchronization, features that are usually lacking in standalone content-based or offline hybrid recommendation models, are two ways that the system closely integrates with the Library Management System. BookMitra integrates adaptive recommendation intelligence and traditional library management features into a single workflow, as shown in Table 1. Functional testing verified the dependable performance of essential functions like transactions, availability updates, and catalog access under concurrent usage.

The hybrid architecture showed consistent behavior from a recommendation standpoint for users with different levels of interaction. The long-tail interaction pattern shown in Fig. 4 was addressed by content-based filtering, which used semantic metadata to produce insightful recommendations for new or infrequent users. Collaborative filtering complemented the content-based branch by capturing implicit user preferences as interaction data grew. These signals

**Table 1.** Comparison of BookMitra with Existing Library Recommendation Systems

| Parameter | Koha LMS[1] | CBF Models [5] | Hybrid Models[8][9] | BookMitra |
|---|---|---|---|---|
| Recommendation Approach | None | Content-Based | Hybrid (Offline) | Hybrid (CBF + CF) |
| Cold-Start Handling | Not Supported | Supported | Partially Supported | Effectively Supported |
| Integration with LMS Workflow | Strong | None | Limited | Tight Integration |
| Role-Based Access Control | Yes | No | No | Yes |

were balanced by the score-level fusion mechanism, which also lessened the over-specialization that single-strategy recommenders frequently exhibit.The suggested books were generally suitable for academic use, according to qualitative feedback gathered during trial deployment. Regarding the skewed rating distribution in Fig. 3, which exhibits a limited amount of negative feedback and positivity bias, consistent recommendation quality across users with varying activity levels can be seen. These findings imply that even in situations with sparse and unbalanced feedback, the hybrid approach continues to work.

Real-time usability was supported by backend evaluation, which revealed average API response times under concurrent access of less than 400 ms. But, especially for specialized content, batch-based updates, poor mobile responsiveness, and sparse data can lead to recommendation errors and popularity bias.

## 5 Conclusion and Future work

This paper introduces BookMitra, an intelligent and scalable library management system that incorporates tailored book recommendations into essential library operations, going beyond traditional automation. The system overcomes the enduring drawbacks of conventional library platforms, such as their limited user engagement, manual record-keeping, and lack of adaptive recommendation systems. BookMitra makes academic book discovery more dynamic and user-focused by integrating hybrid recommendation intelligence into standard library operations. Modern web technologies combined with machine learning enhance user experience and operational efficiency. The library management layer consistently manages catalog management, role-based access control, transactions, and stock tracking, while the recommendation layer keeps learning from user interactions to provide tailored recommendations.The hybrid combination of collaborative preference modeling and content-based semantic similarity allows the system to remain effective across cold-start and sparse-interaction scenarios.

According to experimental evaluation and pilot user feedback, BookMitra provides consistent system performance, accurate transaction handling, and meaningful recommendation outputs under realistic usage conditions. Overall usability was improved by automated alerts and real-time availability updates, and users generally thought the recommended books were pertinent for academic use. Recommendation results can be understood in terms of false positives and false negatives from an error-analysis standpoint. False negatives occur when relevant but specialized or recently added books are not recommended because of sparse interaction histories or limited semantic overlap, whereas false positives are suggested books that are not in line with a user's academic intent and are frequently impacted by popularity bias in collaborative signals. These effects are a reflection of inherent trade-offs in recommendation systems that operate with sparse and unbalanced data. Future work will focus on adaptive user-aware weighting strategies, incorporation of temporal dynamics, and richer feedback signals to further enhance recommendation accuracy and personalization.

## 6 Acknowledgments

The authors thank the Indian Institute of Information Technology, Surat, for providing the necessary facilities and resources to conduct this research.

## 7 Declarations

**Conflict of Interest** The authors declare that they have no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

## References

1. Koha Community, *Koha Open Source Library System*, 2024. Available: https://koha-community.org/
2. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *Item-based collaborative filtering recommendation algorithms*, In Proceedings of the 10th International Conference on World Wide Web, 2001.
3. X. Su and T. M. Khoshgoftaar, *A survey of collaborative filtering techniques*, Advances in Artificial Intelligence, vol. 2009, pp. 1–19, 2009.
4. Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.
5. P. Lops, M. de Gemmis, and G. Semeraro, *Content-based recommender systems: State of the art and trends*, In Recommender Systems Handbook, Springer, 2011, pp. 73–105.
6. G. S. Mahalle, S. V. Patil, and A. M. Deshmukh, *Pattern-based hybrid book recommendation system using semantic relationships*, Scientific Reports, Nature, 2023.
7. A. K. Singh et al., *Book Recommendation System: A Systematic Review and Research Issues*, International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, 2021.
8. S. Rautray and R. Balabantaray, *A hybrid book recommendation system using content-based filtering and collaborative filtering*, In IEEE International Advance Computing Conference (IACC), 2014.
9. S. Siddiqui et al., *Designing Web-based Library Management System*, 2018. [Online]. Available: Semantic Scholar.

10. D. M. Ignatovich, *Implementing Role-Based Access Control (RBAC) in Node.js and React*, Medium, 2023.
11. C. Musto, G. Semeraro, and P. Lops, *Semantics-aware content-based recommender systems*, Decision Support Systems, vol. 86, pp. 45–57, 2016.
12. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, *Neural collaborative filtering*, In Proceedings of the 26th International Conference on World Wide Web (WWW), 2017, pp. 173–182.
13. P. Jomsri, *Hybrid recommender system model for digital library from multiple online publishers*, F1000Research, vol. 12, 2024.
14. W. Gao, *Research on optimization of library book recommendation system based on Transformer architecture and adaptive extreme learning machine*, Decision Analytics Journal, 2025.