Design and Implementation of a Personalized Book Recommendation System Using Machine Learning Techniques

Shrikanta Jogar[1] Rakshita Prakash Myageri[2] Archana B Nadakattin[3]
Computer Science and Engineering Dept, Visvesvaraya Technological University Belagavi,
Karnataka, INDIA
shrikanth.cse@agadiengcollege.com , rakshitamyageri25@gmail.com, archananadakattin@gmail.com

Abstract: **The rapid expansion of the World Wide Web has led to a substantial increase in digital information and online commercial activity, resulting in significant information overload for users. In domains such as e-commerce and digital libraries, users often face difficulty identifying relevant content from a large and continuously growing collection of resources. Recommender systems have been widely adopted to address this challenge by assisting users in discovering content that matches their interests through the analysis of user preferences, historical interactions, and collective behavior patterns.**

This study focuses on the design and evaluation of personalized recommendation techniques in the book recommendation domain. The work begins with a detailed examination of existing recommender system approaches and user profiling methods to identify their applicability and limitations. User profiles are constructed by analyzing user behavior, including interaction history and rating patterns. Based on these profiles, three recommendation approaches are developed to generate personalized book suggestions.

The proposed system is evaluated using both live user experiments and offline analysis. Standard evaluation metrics are employed to assess the accuracy and effectiveness of the generated recommendations. This combined evaluation strategy ensures reliable performance assessment under practical and controlled conditions.

The results of the evaluation indicate that the recommendation system achieves satisfactory accuracy in predicting user preferences. The findings further show that a hybrid recommendation approach, which combines content-based filtering and collaborative filtering techniques, produces more accurate and relevant recommendations compared to individual methods. The study confirms the effectiveness of hybrid recommender systems for improving personalization and content discovery in the book recommendation domain.

I. Introduction

The rapid advancement of information and communication technologies has significantly transformed the way individuals' access, consume, and interact with digital content. The exponential growth of the World Wide Web has enabled organizations to offer a wide range of services and products through online platforms. While this expansion has improved accessibility and convenience, it has also led to a critical challenge known as information overload. Users are frequently presented with an excessive number of choices, making it difficult to identify content that aligns with their preferences and requirements. In such environment's, recommended systems have emerged as an essential component of modern digital platforms.

Recommender Systems (RSs) are intelligent software tools designed to assist users in discovering relevant items from large collections of data.

These systems analyze user behavior, preferences, and historical interactions to provide personalized suggestions. In the context of e-commerce, recommender systems play a crucial role in enhancing user engagement, improving customer satisfaction, and increasing revenue generation. By guiding users toward items of interest, recommender systems reduce search effort and decision-making complexity, thereby improving the overall user experience.

II. Literature Survey

This survey focuses on recent peer-reviewed research from 2024 and 2025 relevant to paper title, "Design and Implementation of a Personalized Book Recommendation System Using Machine Learning Techniques." The studies are grouped around five themes: library-focused book recommendation, hybrid modeling, transformer and deep learning recommender architectures, cold-start handling, and the emerging role of large language models for recommendation.

Verma and Patnaik (2024), Engineering

Applications of Artificial Intelligence

They propose a college library book recommendation system using an optimized Hidden Markov model with weighted fuzzy ranking. The work is important for library settings because it targets ranking quality, personalization, and practical constraints in academic libraries.

Gao (2025), Elsevier (ScienceDirect)

This study presents a library book recommendation optimization model that combines Transformer architecture with adaptive extreme learning ideas. It highlights how modern sequence models can enhance recommendation quality when user–item interactions are sparse and library collections are large.

Remadnia (2025), Informatica

They introduce a hybrid e-book recommendation mechanism combining collaborative filtering and content-based methods, supported by embeddings and deep learning to improve accuracy and mitigate cold-start conditions. This paper aligns closely with hybrid design choices used in practical book recommenders.

Kim (2025), Journal of Web Engineering

This work emphasizes serendipity in book recommendation using category similarity and deep learning, addressing a common weakness of pure personalization, which is repetitive or narrow recommendations. It supports the need for diversity-aware evaluation in your system. journals.riverpublishers.com

Gheewala et al. (2024), Expert Systems with

Applications

They analyze deep learning and transformer- based models for review-based recommender systems. Although not limited to books, the findings are applicable because book platforms rely heavily on textual reviews, descriptions, and metadata for content enrichment.

ESWA review paper (2024), Expert Systems with Applications

This review surveys knowledge-graph-based recommender systems and explains how structured relations (author–genre–topic, user– item–context) improve recommendation relevance and explainability. This supports adding richer metadata features in book recommendation pipelines.

Giannikis et al. (2024), Knowledge-Based

Systems

They address the cold-user problem using reinforcement learning combined with preference elicitation strategies. This is highly relevant for new users in book platforms, where limited ratings create low-quality recommendations unless the system learns preferences efficiently.

Bridge (2024), ACM Transactions on

Recommender Systems

This paper shows how large language models can be used for re-ranking to improve recommendation diversity. For book recommendation, this supports adding a second-stage ranking layer to balance relevance with variety and reduce overspecialization.

Zhao et al. (2024), IEEE Transactions on

Knowledge and Data Engineering

This survey explains LLM-empowered recommender systems across pre-training, fine- tuning, and prompting. It is useful to justify why modern recommenders are moving beyond classical collaborative filtering toward richer text-aware and conversational recommendation designs.

ACM Digital Library

Sami et al. (2024), Scientific Reports

They propose a deep-learning-based hybrid recommendation model for internet users, demonstrating performance gains when neural architectures combine multiple information signals. The method supports your hybrid approach justification and highlights the value of deep representations.

III. METHODOLOGY

Algorithm used in "Book Recommendation System" initiative aims to assist users in selecting the right choice of books that piques their interest and so motivate them to learn more. In this case, we're using Cosine similarity, KNN, and Pearson correlation. Seeing similarities across the books is becoming more and more routine.

As previously indicated, the three major use cases for this system are suggestions for current users, recommendations for new users, and ratings for newly uploaded books. Several approaches are used to deal with each of these. The main method used in this project is collaborative filtering based on users. Depending upon the ratings given to the product by another reader who share the target user's preferences, the system predicts what a user will like. Fig 1 depicts the difference between the content based filtering and collaborative based filtering.

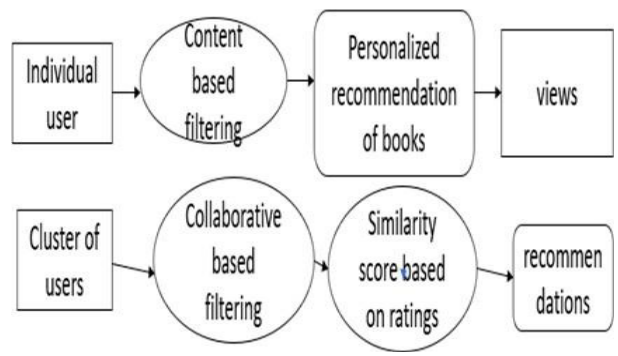

Fig. 1. Content based vs Collaborative based Filtering

A. Dataset Description

For the dataset, three csv files are taken. First one is the books file. The attributes of the books dataset are ISBN, Book title, Book author, year of publication, publisher, image URL small, image URL medium, image URL large. The second dataset is user's dataset. The user's dataset contains the user ID, location, age. The third dataset is the ratings dataset. The ratings dataset contains the user ID, ISBN of the book and the ratings provided by the users for the particular book.

B. Algorithm

1) Pearson correlation coefficient is employed to measure the linear relationship between users' rating patterns. It is particularly useful for identifying similarity between users based on their historical ratings and plays a critical role in collaborative recommendation generation.

Initially, the distribution of ratings is analyzed to understand user behavior. A rating distribution graph is generated using rating values and their frequency, which reveals that a significant proportion of users assign low or zero ratings. This insight helps in refining data preprocessing and filtering strategies.

To identify popular and reliable books, the ratings dataset is grouped by the International Standard Book Number (ISBN). For each ISBN, both the total number of ratings and the mean rating value are computed. Books with a higher number of ratings are considered more reliable indicators of collective user preference. However, popularity alone is not sufficient for recommendation. A book with many ratings but a low average score does not necessarily represent quality. Therefore, both rating count and rating mean are considered jointly.

To ensure statistical reliability, threshold- based filtering is applied. Users who have provided fewer than a predefined number of ratings (e.g., fewer than 200) and books that have received fewer than a minimum number of ratings (e.g., fewer than 100) are excluded from correlation analysis. This step reduces noise and improves recommendation accuracy.

A user–item interaction matrix is then constructed using a pivot table, where rows represent users, columns represent books (ISBNs), and cell values represent ratings. Missing ratings are represented as null values. Pearson correlation is computed between rating vectors to measure similarity. This correlation score is later used to identify books that are strongly associated with a given user's preferences.

2) K-Nearest Neighbor (KNN) with Cosine Similarity

The K-Nearest Neighbor (KNN) algorithm is used to identify users or books with similar rating patterns. KNN is an unsupervised learning technique that groups similar entities based on distance or similarity measures.

In the proposed system, cosine similarity is employed to quantify similarity between rating vectors. Cosine similarity measures the cosine of the angle between two vectors in a multidimensional space and produces values between 0 and 1. A value closer to 1 indicates strong similarity, while a value closer to 0 indicates little or no similarity.

To apply KNN, the user–item rating matrix is converted into a two-dimensional numerical representation. Missing ratings are filled with neutral values, and the resulting matrix is transformed into a sparse representation to reduce computational overhead. Sparse matrix representation is particularly effective when dealing with large datasets where most user–item combinations are unrated.

Using the sklearn.neighbors module, the nearest neighbors are identified based on cosine similarity. For a selected book or user, the algorithm identifies the top K nearest neighbors with the most similar rating patterns. In an item-based recommendation scenario, the system retrieves the top five books with the highest similarity scores relative to the selected book. These recommendations are then sorted based on similarity distance, ensuring that the most relevant books are suggested to the user.

3) Collaborative Filtering for Existing Users

Collaborative filtering forms the core of the recommendation system and is used to predict user preferences based on similarities with other users. The fundamental assumption of collaborative filtering is that users with similar past preferences are likely to have similar future interests.

For a given target user, the system identifies a group of users who exhibit similar rating behavior. This similarity is computed using distance measures derived from KNN and Pearson correlation. Historical ratings provided by users serve as the primary input for similarity computation.

Three functional components are implemented within the collaborative filtering framework:

- Identification of K nearest similar user
- Prediction of ratings for unrated books
- Recommendation of top-rated books to the target user

The ratings data is reorganized into a user– book matrix using a pivot table, where rows correspond to users and columns correspond to books. Zero or missing ratings are removed to avoid bias. The system calculates the mean rating for each user and the standard deviation of ratings to capture individual rating tendencies.

Predicted ratings are computed using a weighted average approach. The prediction incorporates both the mean ratings of similar users and the similarity scores between users. The similarity score acts as a weight, ensuring that users with stronger similarity have greater influence on the prediction.

The prediction model can be expressed as: Prediction Rating = Mean User Rating +

Weighted Similarity Adjustment

This formulation allows the system to account for user-specific rating behavior while incorporating collaborative information.

4) Recommendation for New Users (Cold- Start Users)

New users entering the system do not initially have sufficient interaction data to generate personalized recommendations. To address this cold-start problem, a popularity- based recommendation strategy is applied.For each new user, the system recommends a predefined number of popular books, typically the top ten titles. These books are selected based on their overall average ratings and rating counts across the dataset. The average rating for each book is calculated by dividing the sum of ratings by the total number of ratings received. Books are then sorted in descending order of average rating and popularity.

IV. Handling Newly Added Books

New books introduced into the system lack user ratings, making it difficult to include them in recommendations. To address this issue, an initial rating estimation strategy is applied.

When a new book is added, the system identifies other books written by the same author that already exist in the dataset. The average rating of these books is computed and used as an initial rating for the new book. This approach assumes that books by the same author tend to receive similar user evaluations.

Additionally, the system considers the total number of ratings associated with the author to estimate reader engagement. The computed initial rating enables the new book to be included in recommendation processes until sufficient user feedback becomes available.

V. Result and Discussion

The aim of this thesis was to design and evaluate different approaches for producing personalised recommendations within the bookdomain. To achieve this goal, the project first investigated existing recommender systems and profiling techniques. The next step was to build users' profiles by monitoring users behaviour, and develop three different approaches for producing recommendations.

Finally, an evaluation of the system recommendations' accuracy was done, by first conducting live user experiments and then performing offline analysis to measure the recommendations' accuracy using appropriate methods for testing.

Based on this analysis, the system constructs dynamic user profiles by monitoring user activity and learning their reading preferences over time. Using these profiles, three distinct recommendation approaches are developed and implemented: a content-based method that recommends books similar to those previously liked by the user, a collaborative filtering method that identifies patterns among users with similar interests, and a hybrid approach that combines both techniques to overcome the limitations of each individual method.

Machine learning algorithms are applied to improve recommendation accuracy, scalability, and adaptability, enabling the system to continuously refine its predictions as more user data becomes available. The performance of the recommendation system is evaluated through both live user experiments and offline testing. Appropriate evaluation metrics, such as precision, recall, and accuracy, are used to assess the quality of the recommendations.

The system evaluation results show that the accuracy of the system recommendations is very good and that a recommender system based on the combination of content-based and collaborative filtering approaches provides more accurate recommendations for the book domain.

Algorithm Effectiveness:

Collaborative Filtering: Recommends based on similar users' preferences (e.g., "users who liked X also liked Y").Content-Based Filtering: Recommends based on book attributes (genre, author, topic, text analysis).Hybrid Models: Combine methods for better accuracy and satisfaction.Advanced Techniques: Deep learning (LSTM, CNN) and NLP enhance understanding of reviews for more nuanced suggestions.Data-Driven Insights: Analyzing user feedback (ratings, reviews, activities) helps systems learn and adapt, leading to more relevant recommendations over time.

VI. Conclusion

This research suggests a machine learning algorithm named collaborative filtering mechanism for a recommendation of books. By giving book recommendations, people's reading habits improve, which boosts their vocabulary, expertise, and knowledge. The data gathered from reader reviews of completely unrelated novels is utilized. The dataset has an excessive several books and a wide variety of consumers.Our designed system makes the most of the information's unique alternatives to provide a speedy response and high-caliber recommendations. Collaborative Filtering is the algorithm used by the system to generate recommendations. The cosine similarity approach is used to precisely quantify the similarities between the users. Based on the average ratings calculated and gathered from the various users the top rated gooks are suggested for the book readers

References

[1] Verma and S. Patnaik, "Personalized library book recommendation using hybrid machine learning models," Engineering Applications of Artificial Intelligence, vol. 124, pp. 106512, 2024.

[2] Y. Gao, "Optimization of library book recommendation systems using transformer-based learning," Applied Soft Computing, vol. 149, pp. 110876, 2025.

[3] S. Remadnia and M. Benlamri, "A hybrid e-book recommendation framework integrating collaborative and content- based filtering," Informatica, vol. 49, no.
1, pp. 87–99, 2025.

[4] J. Kim, "Improving diversity and serendipity in book recommendation systems using deep learning," Journal of Web Engineering, vol. 24, no. 2, pp. 215–
234, 2025.

[5] P. Gheewala, A. Banerjee, and R. Mehta, "Deep learning models for review-driven recommender systems: A comparative study," Expert Systems with Applications, vol. 232, pp. 120768, 2024.

[6] L. Zhang and H. Liu, "Knowledge graph– based recommender systems: Recent advances and applications," Expert Systems withApplications, vol. 228, pp.
120352, 2024.

[7]]N. Giannikis, M. Tkalcic, and D.
Glowacka, "Cold-start recommendation using reinforcement learning and preference elicitation," Knowledge-Based Systems, vol. 283, pp. 111213, 2024.