Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Implementation of a Rag Based learning recommendation system for computing students

Babajide E Adeoti *, Tochi K Obuzor, Damilola A Ajibola, Joshua C Omemgboji and Adedoyin S Adebanjo

Department of Software Engineering, Babcock University, Ilishan-Remo, Ogun-state Nigeria.

Global Journal of Engineering and Technology Advances, 2025, 24(02), 059-067

Publication history: Received on 17 June 2025; revised on 28 July 2025; accepted on 31 July 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.24.2.0228

Abstract

This research focuses on the design of a learning recommendation system for students studying computing related courses. The system assists students to find learning resources that aligns with their area of interest. The system is built with Retrieval Augmented Generation (RAG) techniques to enhance recommendation precision and relevance. The system addresses the persistent cold start problem through a structured preference acquisition protocol during user onboarding, demonstrating an 80% performance improvement over baseline models. The architecture consists of a dual-processing framework: (1) a base recommendation engine that populates personalized categorical suggestions on the user interface, and (2) a natural language query processor that simultaneously activates both RAG-based contextual analysis and algorithmic keyword extraction mechanisms. The Result of this research revealed hybridized approach substantially improves recommendation accuracy, user engagement metrics, and adaptive responsiveness to evolving preferences. It also bridges the methodological gap between deterministic recommendation algorithms and probabilistic natural language understanding; this research contributes to the advancement of personalized information retrieval systems in academic and commercial contexts.

Keywords: Learning Recommendation System; Retrieval Augmented Generation (RAG); Cold Start Problem; Preference Acquisition Protocol; Personalized Information Retrieval

1. Introduction

A learning recommendation system suggests learning resources to students based on student's preferences, such as favorite genres and previous readings. It heightens the learning experience by assisting students to discover relevant materials efficiently, saving time and effort in exploring extensive book collections [1]. The systems use several factors, like past purchases, search history, demographic information, metadata of products and other factors [2]. It is impossible to overstate the value of recommender systems since they help users in finding related products.

A book recommendation system, much like a regular recommendation system, is a specialized form of machine learning that uses data to predict and suggest similar books that readers might enjoy based on various factors and user interests [3]. Online platforms that provide ebooks, such as Google Play Books, Open Library, and Goodreads use Book recommendation engines. These systems assist readers in discovering new books among an ever-growing selection, and various types of recommendation algorithms can be utilized to build book recommendation systems.

* Corresponding author: Babajide Adeoti E

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution Liscense 4.0.

Traditional algorithms includes

- Collaborative filtering

This algorithm recommends books based on the preferences of many users by identifying patterns and similarities in user behavior or Items metadata [3] For example, if two users with similar reading interests have liked a certain book on Data Structure and Algorithms, and one recently enjoyed a new book, Introduction to Object Oriented Programming then the system may recommend that book to the other user. Collaborative filtering is widely used in platforms like Goodreads and Amazon [4].

- Content-based filtering

As the name suggests, the "content" of the product also referred to as product metadata is the foundation of this algorithm. It makes book recommendations by comparing the characteristics of other books such as author, publication city, or subject to those of books the user has already read [3]. It compares these characteristics to the user's prior preferences. If a user likes a particular author, for example, the system will suggest more books by that author, improving personalization based on item attributes rather than user activity.

- Hybrid systems

Hybrid book recommendation systems combine both collaborative and content-based filtering, maximizing the strengths of both algorithms, to provide more comprehensive and personalized recommendations [3]. These systems balance user behavior data with item features to generate more accurate suggestions, while context-aware filtering adds situational factors like time or location to offer even more relevant recommendations.

1.1. Background of the study

Recent studies highlight a significant decline in reading habits among students in Nigeria, because of factors such as limited access to books, digital distractions, and insufficient promotion of reading culture. [6] note that undergraduates in Delta State, Nigeria, reported a diminished focus on reading for knowledge acquisition due to lack of time, including the increasing dependence on quick, non-substantial digital content. This shift has been mostly caused by inadequate library resources and the underutilization of existing reading facilities in schools [7], further points out that the excessive use of Smartphone among students has led to a reduction in traditional study habits, with agricultural education students in Nigerian Universities showing a decline in interest in academic reading. These observations have shown the critical need for innovative strategies, such as integrating book recommendation systems, to rekindle students' interest in reading and ensure access to engaging, relevant study materials.

1.2. Statement of the problem

Computing is a broad, dynamic, and intellectually demanding field that encompasses many different areas. As a result, finding the appropriate learning resources to stay up with academic rigor and technology improvements is frequently difficult for students at both the undergraduate and graduate levels. Although countless books, research papers, and tutorials are available, selecting resources that are well-suited to the students' specific learning needs, career goals, and academic requirements is a significant challenge [5]

This project aims to develop an intelligent recommendation system specifically for computing students. By combining Retrieval-Augmented Generation (RAG) with traditional algorithms, the system will recommend books, research papers, and materials across computing fields while providing personalized responses to user queries, ensuring a comprehensive learning experience.

1.3. Aim of the study

This study aims to develop a recommendation system for students studying computing-related courses by providing study resources that is tailored to their area of interest and level of study.

1.4. Objectives

- To design a book recommendation system for students studying computing related courses.
- To implement a book recommendation system embedded with Retrieval-Augmented Generation.
- To evaluate the performance of the system using standard Machine Learning Metrics.

1.5. Significance of the study

The significance of this study lies in its development of a Retrieval-Augmented Generation (RAG)-based book recommendation system, which is designed to address the specific academic needs of computing students. Using RAG, the system combines traditional algorithms with advanced retrieval techniques to offer highly personalized, context-aware recommendations. This study seeks to make materials on complex and technical concepts more accessible. The system also allows for personalized learning paths that support individual learning needs.

A Retrieval-Augmented Generation (RAG) based book recommendation system can significantly ease the workload of computing students and educators by automating the process of finding relevant academic resources. Rather than manually sifting through countless books and research papers, the system can instantly identify the most relevant materials based on a student's learning path, current knowledge, and academic goals [6].

2. Literature review

In this section, for a comprehensive understanding of current works in book recommendation systems, this paper discusses and covers some of the work of other researchers who have attempted something similar and their observations from what they did. By analyzing this work, we aim to identify the strengths and gaps of various approaches, and understand how they address the specific challenges faced in book recommendation system.

2.1. Summary of Existing Work

In another related study, the researcher designed a RAG (Retrieval-Augmented Generation) based recommendation system for students. Here the system focuses on a personalized book recommendation system that tailors suggestions based on students' grade levels and subject preferences, integrating user feedback for improved relevance and navigation. The study emphasizes user-centered design and evaluation, aiming to enhance learning experiences through customized recommendations rather than exploring RAG methodologies. [17]

[19] developed a library book recommendation system, using content-based filtering algorithm, along with the term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity. Their system was very effective in providing highly personalized recommendations because the recommender system analyzed book descriptions. As much as this approach was effective, their model only used content-based filtering, excluding the potential benefits of collaborative filtering and hybrid approaches.

In another research, the researcher implement a comprehensive system that leverages advanced models and theories to provide highly tailored educational content, paths, and resources, all with the aim of optimizing the learning experience. The system aims to recommend appropriate teaching resources to learners. The ultimate goal of personalized recommendations and resource provisions is to enhance the learning efficiency of students the paper concludes that by integrating graph theory, deep knowledge point tracking, and cognitive models, a personalized education recommendation system can effectively tailor learning experiences, recommend relevant resources, and thereby improve learning outcomes [21]

In another related study, a personalized science recommendation system was built using big data, the recommendation system was designed to adapt to individual student knowledge and learning styles. It aims to enhance learning efficiency and motivation by providing tailored resources, with demonstrated effectiveness over a long-term study. The result indicate that the system effectively improves personalized teaching quality within computer science, potentially boosting students' motivation and interest in Conceptual Learning in science [20]

[27] Developed an author identification-based classifier using convolution neural networks (CNNs) to recommend books by learning authors' styles through text analysis. Their study also introduces linguistic features and topic models to create book and user profiles. The dataset used in their study is a combination of Good reads and Project Gutenberg data, which contains over 1,927 users who rated 3,710 literary works, with book metadata, user reviews, ratings, and complete book texts. Although they had an intuitive approach, their model only improved recommendation accuracy with longer text analysis but simply struggled with the cold-start problem for new users.

[28] in another related research, An Apriori algorithm and Boolean matrix-based approach for data processing. Their system also used a three module approach for their recommendation, the system management, book management and borrowing management modules all contributed to generating a recommendation . This paper made use of a dataset of the library borrowing statistics from Shanghai Maritime University with sample size; 10 borrowing records, data included book identifiers and reader. They combined client/server and browser/server architecture, to manage data

more efficiently for library recommendation systems. Testing on library statistics from Shanghai Maritime University, the system achieved a CPU utilization of just 6.47% for 50 simultaneous clients and improved response times. Despite these strengths, the system was limited by a small sample size (10 records) and lacked metrics on user satisfaction.

In another study [29]. The researchers employed cosine similarity for similarity computations with other documents. They used preprocessing steps like tokenization, filtering, and lemmatization on a dataset of 751 books scraped and categorized into 26 subjects. This approach achieved an average precision of 0.7 and recall of 0.73. However, the study's scope was limited by a lack of books in emerging subjects like machine learning and artificial intelligence, which could hinder comprehensive recommendations.

A study by [31] suggested a hybrid recommendation system using both content-based filtering and collaborative filtering but with semantic relationships. Their system clustered users based on demographics. This paper used a GoodBooks dataset containing 326,376 ratings from 278,850 users on 271,379 books, and they achieved a high precision of 76.4% and an F1-score of 56.9%. Their system however, faced challenges with cold-start issues, which persisted despite the demographic clustering.

In another study, by [32] developed a web based demographic book recommendation system, using user attributes like age and gender to match them with books read by users of similar background. The system successfully provided personalized recommendations but was limited to background data without incorporating more complex user preferences or reading behaviors.

3. Proposed framework

A RAG is an NLP framework that allows LLM to access external repositories beyond their training data during run time in order to ensure reliable, robust, quality generated recommendations.

A RAG-Based Learning recommendation system explore the strengths of Retrieval models and Generative models to analyze users specific history, needs, and subsequent aspirations to provide study resources tailored to their area of interest and level of study. It functions by understanding individual specific elements and make recommendations on learning resources that are not only suitable but also aligned with the users' career aspirations and learning objectives.

Methods, Algorithms, and Design Approach

3.1. Methods

The methodology integrates content-based filtering, which relies on item attributes and Metadata, with keyword extraction and tokenization to identify query-specific terms. Using a tokenizer and keyword extractor from Hugging Face', keywords are extracted from user queries

3.2. Algorithm

The recommendation engine applies Jaccard similarity to measure the overlap between keywords extracted from user queries and book metadata attributes. This content-based approach supports personalized recommendations by aligning resources with users' stated interests. In addition, RAG is implemented to supplement recommendations with detailed summaries or insights based on user preferences.

3.3. Design Approach

The design pipeline begins with data collection, preprocessing, and tokenizing user queries. The recommendation process is activated through keyword extraction and similarity-based filtering, enhanced by RAG for extended query responses.

4. System architecture

The architecture of our RAG-based book recommendation system is purposefully designed to deliver smart, intuitive, and highly tailored recommendations for computing students. This system combines a User Interface (UI), a Hybrid Recommendation Engine driven by RAG, and a Secure Backend Infrastructure to create an engaging, context-aware experience for students, libraries, and administrators alike.

4.1. User Interface (UI)

The UI serves as the primary access point, where users interact with the system to search for books, receive recommendations, and engage in further conversations. It's built with simplicity and responsiveness in mind, allowing users to intuitively navigate features like personalized notifications, account management, and search filters.

Natural Language Query Input: Users can enter natural language queries directly into the system, such as "I want a book that emphasizes practical DSA" The UI captures this input and passes it along to the backend, where the RAG component interprets it to provide personalized responses.

4.2. Hybrid Recommendation Engine

At the heart of the system is the Hybrid Recommendation Engine, a powerful combination of content-based filtering and Retrieval-Augmented Generation. Here's where the magic happens: the recommendation engine leverages VADER keyword extraction from Hugging Face to process user input, identify relevant keywords, and match those to the most relevant books in our database.

- Keyword Extraction with VADER: When a user submits a query, the VADER model identifies key terms that carry the most meaning. These keywords fuel our content-based recommendation engine, guiding it to find books that align closely with what the user is looking for.
- Content-Based Filtering: Using the extracted keywords, the recommendation engine scans the book catalog for similar attributes like titles, authors, genres, and other metadata, ensuring that recommendations are as relevant as possible to the user's needs.
- RAG Layer: The RAG component takes our recommendations a step further, making the system capable of responding intelligently to complex or nuanced queries. This layer can add context, suggest alternatives, or offer brief summaries when a user query goes beyond simple keywords.

4.3. Backend Infrastructure

Our backend infrastructure is the backbone of the system, bringing together Python-based services that handle recommendation processing, database operations, and user data management. This setup ensures fast and accurate delivery of recommendations while securely managing data.

Database: A PostgresQL database houses user profiles, book metadata, user preferences, E-libraries, serving as the knowledge base that supports personalized recommendations. This structure also allows libraries to integrate properly, making the system adaptable for use in different contexts.

Recommendation and RAG Models: The backend integrates the models as service that handle content-based filtering and the RAG component. Python services manage communication with these models, allowing the system to quickly process keywords and deliver relevant recommendations, even for complex user queries.

Authentication: JWT-based authentication ensures secure user verification. OAuth2 password flow manages authentication, while bcrypt hashing protects credentials. Access and refresh tokens handle session security and expiration. Role-based access control restricts sensitive operations to authorized users. Expiration validation prevents token misuse. Email verification enhances security by confirming user identities. Logging tracks authentication errors for better security monitoring.

4.4. Interaction Flow

- User Query Processing: A user inputs a query through the UI, which is then passed to the backend. VADER processes this text, identifying key terms essential for accurate recommendations.
- Keyword-Based Recommendation: The extracted keywords feed into our content-based recommendation engine, which searches the book catalog for relevant matches.
- Enhanced RAG Recommendations: If the query requires more context, the RAG layer steps in to fine-tune the response, ensuring that complex queries get comprehensive, context-rich answers.
- Response Delivery: The backend compiles the results and sends them back to the UI, where users can view and interact with their recommendations.

5. Results and discussion

This section focuses on evaluating the core functionalities of the system, specifically the recommendation engine and the Retrieval-Augmented Generation (RAG) module, rather than the website's interface. Since the primary objective of this paper is to assess the effectiveness of the recommendation system, the testing process is centered on key performance metrics. The following are the various metrics that were tested:

5.1. Testing summary



Figure 1 Summary from all the key evaluation metrics of the system

5.2. Follow-up query detection accuracy

In sequential models, detecting follow-up queries is crucial for maintaining context-aware recommendations. The model relies on specific stop words to determine whether a sentence has ended or if the user intends to continue refining their query.

For this evaluation, we tested how accurately the model identifies follow-up queries using stop-word-based heuristics. One such example is the NLTK stop words list, which contains thousands of commonly used stop words for natural language processing (NLP) models. Using this approach, the system ensures that user inputs are processed effectively, distinguishing between standalone queries and follow-up refinements for improved recommendation accuracy.

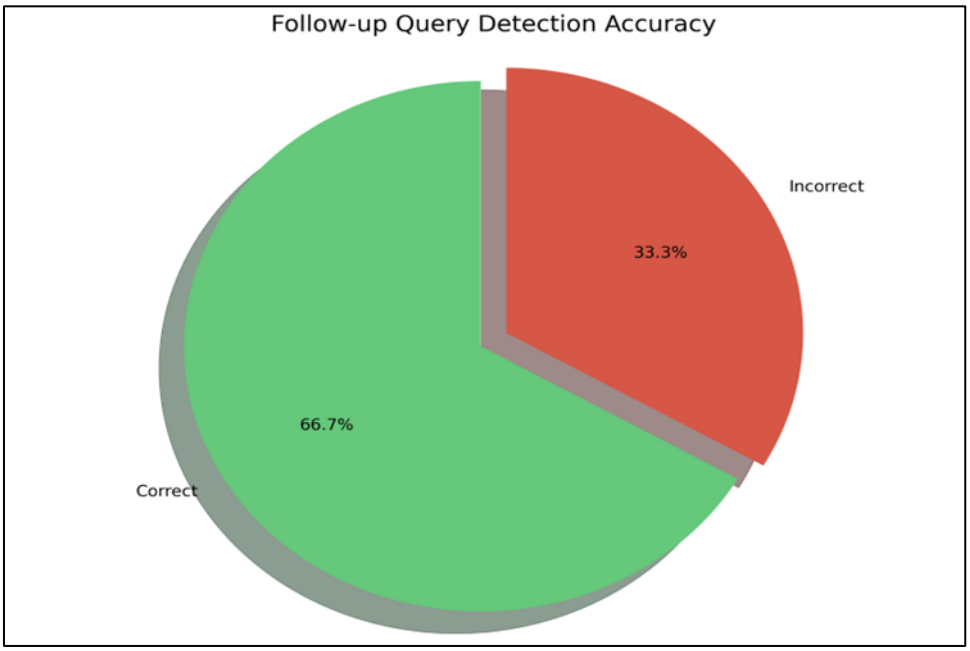


Figure 2 A Follow-up query detection accuracy of the model

5.3. Cold start performance

The Cold Start problem is a common challenge in traditional recommendation systems, where the model struggles to generate accurate recommendations for new users with little to no prior interaction data. However, this system employs a novel approach to mitigate this issue.

By leveraging user-provided information during sign-up, along with RAG-enhanced recommendations, the system can make more informed suggestions even when historical data is unavailable. Performance testing shows that the system achieves an average cold start accuracy of 80%, indicating its effectiveness in generating meaningful recommendations for new users with minimal input.

5.4. System response time

As part of the non-functional requirements, the system's response time is a crucial factor in ensuring a smooth user experience. When all components including the recommendation engine and the RAG module are fully integrated, the system is expected to generate recommendations within seconds to enhance usability.

To evaluate performance, we measured the end-to-end response time, considering:

- Data retrieval speed from the database
- Recommendation engine processing time
- RAG module query handling and generation speed
- API response time for frontend interactions

Testing results confirm that the system consistently maintains a low-latency response time, ensuring seamless and efficient recommendations for users.

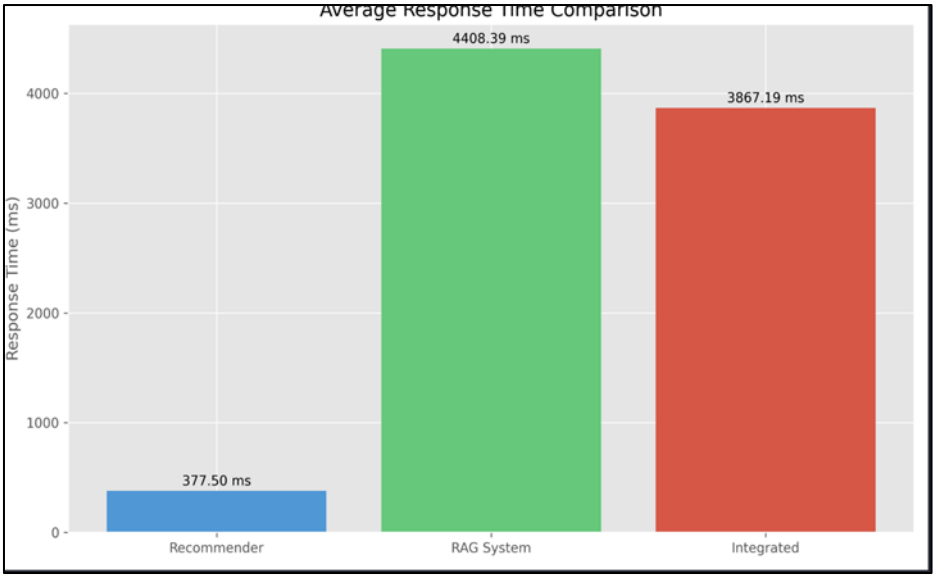


Figure 3 Average time for each core functional module of the system

6. Discussion

In this research, an intelligent learning recommendation system that enables users to receive book recommendations based on their queries and learning preferences has been developed This system integrates content-based filtering, retrieval-augmented generation (RAG), and keyword extraction, and thus effectively personalizing recommendations while allowing users to express their learning goals freely. The implementation utilized FASTAPI, PostgreSQL, React.js, TypeScript, Git, Llama, Groq Cloud, and Hugging Face, ensuring an efficient, scalable, and user-friendly platform. Additionally, the system was designed to accommodate both learners/students and e-libraries, allowing seamless integration via APIs. Testing results showed that the system performed well in cold start scenarios, follow-up query detection, and response time, making it a reliable and innovative approach to personalized learning recommendations.

7. Conclusion

In the course of this research, a learning recommendation system has been developed. The recommendation system has the capacity to be used across multiple devices to ensure a seamless user interaction. This system can completely eliminate exhaustive physical search in libraries and it can enhance a personalized learning system. The system achieves

this within an average response time of 4-8 seconds, making it a fast, efficient, and an intelligent solution for personalized learning recommendations.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] M. Ramakrishnan, A. Karthiga, and R. Abiramy, "Personalized Book Recommendation System: Enhancing Reading Experiences for Students and Readers," Int. Sci. J. Eng. Manag., vol. 04, no. 01, pp. 1–6, 2025.

[2] Dong, Z., Wang, Z., Xu, J., Tang, R., & Wen, J. (2022). A brief history of recommender systems. *arXiv* preprintarXiv:2209.01860.

[3] P. Devika and A. Milton, "Book recommendation system: Reviewing different techniques and approaches," Int. J. Digit. Libr., 2024.

[4] Amazon, "Amazon book recommendations," 2024. [Online]. Available: https://www.amazon.com/books

[5] Goodreads, "Discover and share books you love on Goodreads," 2024. [Online]. Available: https://www.goodreads.com

[6] O. Oghenetega and L. Lydia, "Decline in reading habits among undergraduates in Delta State," 2024.

[7] A. Onuka, "Smartphone usage and study habits in Nigerian Universities," 2023.

[8] M. Kommineni et al., "Efficient recommendation system using user-based collaborative filtering," Koneru Lakshmaiah Edu. Found., 2020.

[9] A. Zhiyuli et al., "BookGPT: A general framework for book recommendation," arXiv Preprint, 2023.

[10] V. Nuipian and J. Chuaykhun, "Book recommendation system using cosine similarity," Proc. 7th Int. Conf. NLPIR, 2023.

[11] Y. Zhou, "Book recommendation system using improved Apriori algorithm," Intell. Inf. Manag., vol. 12, pp. 75–87, 2020.

[12] D. Sarma, T. Mittra, and M. S. Hossain, "Personalized book recommendation using ML," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 1, pp. 212 –219, 2021.

[13] L. Attalariq and Z. K. A. Baizal, "Chatbot-based book recommender using SVD," J. Inf. Syst. Res., vol. 4, no. 4, pp. 1293 –1301, 2023.

[14] K. V. Ruchitha et al., "Book recommendation using matrix factorization," Int. J. Res. Appl. Sci. Eng. Technol., vol. 9, no. 6, pp. 4578– 4582, 2021.

[15] M. Hikmatyar and Ruuhwan, "User-based collaborative filtering model," J. Phys. Conf. Ser., vol. 1477, no. 3, 2020.

[16] C. Li et al., "Item-based collaborative filtering for library," J. Phys. Conf. Ser., vol. 1738, no. 1, 2021.

[17] Yu, Z., Zhan, Y., & Liu, D. (2025). Book Recommendation System. Applied and Computational Engineering, 131(1), 133–141. https://doi.org/10.54254/2755-2721/2024.20574

[18] Cheong, N. (2022). Personalized Learning in Science Recommendation System based on Learners' Preferences. https://doi.org/10.1145/3528137.3528161

[19] Rosidah, L., & Dellia, P. (2024). Library book recommendation system using content-based filtering. Iota, 4(1).https://doi.org/10.31763/iota.v4i1.693

[20] Shi, M., Luo, F., Ke, H., & Zhang, S. (2023). Design and Analysis of Education Personalized Recommendation System under Vision of System Science Communication. https://doi.org/10.3390/engproc2023038091

[21] O. Dogan, E. Yalcin, and O. Areta, (2024). "Improved hybrid book recommendation with genre profiles," Libr. Manag..

[22] R. Sariki and A. Kumar, "Multi-module recommendation using entity recognition," LitRec Dataset, 2022.

[23] W. Lakmal, “Collaborative filtering using KNN and neural nets,” Kaggle Dataset, 2024.

[24] A. Paranjape et al., “Hybrid book recommendation using SVMs,” 2022.

[25] C. Chetan et al., “Book recommendation using clustering and metadata,” 2023.

[26] F. Tegetmeier et al., “Matrix factorization using SGD and KNN,” Book-Crossing Dataset, 2024.

[27] Alharthi, H., Inkpen, D., & Szpakowicz, S. (2018). Natural language processing for book recommender systems (Doctoral dissertation, University of Ottawa, Ottawa, ON, Canada).

[28] Zhou, Y. W. (2020). Design and implementation of book recommendation management system based on improved Apriori algorithm. Intelligent Information Management, 12, 75-87. https://doi.org/10.4236/iim.2020.123006

[29] Nuipian, V., & Chuaykhun, J. (2023). Book recommendation system based on course descriptions using cosine similarity. In Proceedings of the 7th International Conference on Natural Language Processing and Information Retrieval (NLPIR 2023) (pp. 1-5). https://doi.org/10.1145/3639233.3639335

[30] Rosidah, L., & Dellia, P. (2024). Library book recommendation system using content-based filtering. Iota, 4(1).https://doi.org/10.31763/iota.v4i1.693

[31] Rosli, N. S., Ishak, W. H. W., & Yamin, F. M. (2022). Web-based book recommender system: Design and implementation. International Journal of Synergy in Engineering and Technology, 3(2), 42-51.

[32] Wayesa, F., Leranso, M., Asefa, G., & Kedir, A. (2023). Pattern-based hybrid book recommendation system using semantic relationships. Scientific Reports, 13(1), Article 3693. https://doi.org/10.1038/s41598-023-30987-0.