BookWise: Smart Recommendations for Students

Enrollment. No.(s) - 21103119 , 21103120 , 21103128
Name of Student(s) - Vanshika Jalhotra , Vidhi Rastogi , Himral Garg
Name of Supervisor - Dr. Sherry Garg

Course Name: Major Project 1
Program: B. Tech. CS&E/ B.Tech IT
7th Sem

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY NOIDA

TABLE OF CONTENTS

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: JIIT , Noida
Date:

Signature:
Name:Vanshika Jalhotra , Vidhi Rastogi , Himral Garg
Enrollment No: 21103119,21103120,21103128

CERTIFICATE

This is to certify that the work titled "**BookWise : Smart Recommendations for Students**" submitted by "**Vanshika Jalhotra, Vidhi Rastogi, Himral Garg**." in partial fulfillment for the award of degree of **B. Tech** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……..

Name of the Supervisor : Dr.Sherry Garg

Designation : Assistant Professor

Date :

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to our supervisor, Dr. Sherry Garg, for her invaluable guidance, encouragement, and unwavering support throughout this research journey. Her insightful feedback, thoughtful advice, and genuine enthusiasm for our work have been instrumental in shaping this project and helping us achieve our goals.

We are deeply grateful for her patience, wisdom, and the opportunities she provided to challenge and improve ourselves. Working under her mentorship has been a privilege, and her dedication and expertise will continue to inspire us in our future endeavors.

Signature of the Student ……
Name of Student: Vanshika Jalhotra
Enrollment Number : 21103119
Date ……

Signature of the Student ……
Name of Student: Vidhi Rastogi
Enrollment Number :21103120
Date ……

Signature of the Student ……
Name of Student : Himral Garg
Enrollment Number :21103128
Date ……

SUMMARY

This project focuses on building a personalized book recommendation system aimed at improving the way undergraduate students find academic resources. The system, currently under development, will analyze user input, such as search queries and reading preferences, to provide tailored book suggestions relevant to their studies. By minimizing the time spent searching for materials and offering recommendations that students might not have previously considered, the system is designed to enhance the overall learning experience.

Key features under development include a hybrid recommendation engine that leverages content-based and collaborative filtering techniques to deliver accurate, personalized results. The system will integrate with the Open Library API, offering access to a broad range of book titles. The interface is being designed to be simple and intuitive, ensuring easy navigation and interaction. Once completed, this tool aims to become a vital resource for students, supporting efficient academic research and study planning.

Once fully developed, this book recommendation tool aims to serve as an indispensable resource for undergraduate students, helping them streamline their study planning and improve their academic research efficiency. By catering to their specific interests and study requirements, the system will support more informed and effective learning decisions, contributing to students' academic success.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 General Introduction

Our book recommendation project is designed to make finding your next read easier and more enjoyable. It includes several features to personalize suggestions for users. Firstly, it recommends books that match well with each user's unique preferences. It also suggests books from the same series to help readers continue enjoying stories they love. Apparently, it provides cross-referenced recommendations, connecting users to books related by theme, genre, or author. Finally, it offers content-based recommendations, analyzing book content to find titles with similar themes or styles.

1.2 Problem Statement

Undergraduate students often face challenges when selecting relevant books for their studies, assignments, and research. They spend considerable time searching for appropriate resources, often leading to inefficiency and frustration. A tailored book recommendation system can simplify this process by suggesting books based on their search queries, previous reads, or areas of interest. The goal of this project is to build a recommendation system that responds to student searches and provides them with the most relevant books from a digital library.

1.3 Significance/Novelty of the Problem

With variety and a huge number of books available, sometimes it becomes challenging for students to find new books that accurately suit their tastes and requirements. Traditional recommendation systems suggest only popular or trending titles, which may not incline with individual preferences. Our project addresses this gap by offering a more personalized recommendation experience. By developing multiple recommendation features like best-matched books, similar series suggestions, cross-referenced recommendations, and content-based matching, our project ensures that students are presented with books that are closely aligned to their interests.

The features that make our project different is how it combines these diverse strategies to provide a multi-faceted recommendation experience. Students not only receive suggestions that reflect their personal reading patterns but also get recommendations that connect books by content, themes, or series, allowing them to explore new titles in an engaging, curated way. This novel, layered approach enhances the user experience by delivering more relevant and meaningful book choices, making the search for the next great read both efficient and enjoyable.

1.4 Empirical Study (Field Survey/Existing Tool Survey/Experimental Study)

Empirical Study: The empirical study for this project focused on evaluating existing recommendation systems and conducting experimental studies to identify challenges in accuracy. personalization and the scalability of book recommendations. Field surveys were conducted with users to gather insights about their preferences, pain points, and expectations about book recommendations. Participants provided feedback on the relevance, diversity, and perceived validity of existing recommendation platforms. It emphasizes the need for better personalization and management of fragmented user data…

Review of existing tools and systems: This includes platforms like Goodreads and library-based recommendation engines. To evaluate its features, benefits, and limitations, this analysis reveals gaps in addressing the cold start issue. Lack of context awareness and over-reliance on user ratings.

Experimental studies : Empirical studies have been conducted to verify the performance of different algorithms such as collaborative filtering. Filtering by content and hybrid methods to overcome these challenges, and indicators such as precision, recall, F1 score, and RMSE are used to evaluate the effectiveness of these algorithms. Moreover, integration is explored. Contextual demographics to improve the quality of recommendations and meet specific user needs This empirical approach provides valuable insights into optimizing recommendation techniques for better accuracy and usability.

1.5 Brief Discussion of Solution Approach

The book recommendation project is built on combining multiple recommendation strategies.The approach encompasses a range of features and functionalities aimed to create a comprehensive and user-centric experience.

1. **Best-Matched Book Recommendations:** Our project uses the K-Nearest Neighbors algorithm to provide personalized recommendations.The algorithm maps users to books that are similar to those they've liked, using metrics such as genre, author, or user rating patterns. This method ensures that users see recommendations closely aligned with their particular interests.
2. **Series-Based Recommendations:** For students who enjoy particular series, our project includes a feature to recommend other books within the same series. This is particularly useful for helping users explore further into series they've previously engaged with.
3. **Cross-Referenced Recommendations:** The project cross-references books based on related themes, genres, or authors. It connects users to books that may not be within their immediate preference but are still related, by analyzing book attributes.This approach expands users' exposure to different books that share a similar appeal.
4. **Content-Based Recommendations:** A content-based filtering approach is implemented to suggest books based on specific characteristics within each title, such as keywords, themes, or writing style. This adds a deep layer to the recommendations by focusing on the internal qualities of each book, making it easier for users to find titles that align with their preferred style or themes.
5. **Future Integration of SVD (Singular Value Decomposition):** To further improve the recommendation quality, there is an additional focus on implementing SVD. This method will allow for more complex matrix factorization, improving the system's ability to capture nuanced relationships between users and books based on collaborative filtering. SVD will enhance the system's ability to make accurate predictions even when data is sparse.

1.6 Comparison of Existing Approaches to the Problem Framed

Our project is implemented with given recommendation techniques to provide personalized, and relevant book suggestions, overcoming common limitations and creating an efficient, user-centered system.

Popularity-based recommendation systems often suggest popular or trending books, but these are generic and may not match individual preferences. Our project addresses this problem by focusing on personalized recommendations according to each reader's interests, making it possible for users to find books that genuinely appeal to them.

Collaborative filtering is another common approach that identifies patterns in user preferences but has a cold start problem when we have limited data. To overcome this, we have implemented K-Nearest Neighbors (KNN) to map users with similar books. We also plan to add Singular Value Decomposition (SVD), which will give us more enhanced recommendations even when data is sparse, making the system adaptable for new users and books.

Content-based filtering analyzes details like genre and theme for personalized suggestions but can limit discovery. To address this, we add cross-referenced and series-based recommendations, helping users explore relevant yet new titles.

Many systems combine collaborative and content-based filtering in complex hybrids, which improve accuracy but are resource-intensive. Our project's multi-strategy approach simplifies this by combining KNN, content-based, series-based, and cross-referenced recommendations, achieving accurate results efficiently.

Advanced systems use deep learning or context-based suggestions, which require substantial data and resources. Although our system doesn't use deep learning, it still provides meaningful and varied recommendations by integrating content-driven and series-based suggestions, offering a balanced, practical experience.

CHAPTER 2 : LITERATURE SURVEY

2.1 Research Papers

Title	Year	Author (s)	Dataset Used	Recommendation Technique / Algorithm	Evaluation Metrics	Key Findings	Challenges/Future Work
(Jiao, N., 2020)	2 July 2020	Liu, H.; Jiao, N.	Goodreads, User Profiles	Hybrid (Collaborative Filtering + Context Awareness)	Precision, Recall, F1 Score	Combining context awareness with social networks improves recommendation accuracy.	Scalability and integration with real-world platforms.

(Fayyaz et al., n.d.)	2 November 2020	Zeshan Fayyaz , Mahsa Ebrahimian , Dina Nawara , Ahmed Ibrahim and Rasha Kashef	N/A	Various collaborative filtering algorithms like mean-squared difference, Pearson correlation, cosine similarity, Spearman correlation, and adjusted cosine similarity	Metrics such as recall, precision , accuracy, ROC curve, and F-measures	The paper provides a comprehensive guide to recommendation systems, highlighting different categories, challenges, evaluation metrics, and business adoptions	Addressed challenges include cold-start, data sparsity, scalability, diversity, and the need for more robust recommendation algorithms for wider applications
(Ms. Praveena, 2023)	2023	Ms. Praveena Mathew , Ms. Bincy Kuriakose , Mr.Vinayak Hegde	The paper used dataset generated by scraping library website data	Content-Based Filtering, Collaborative Filtering, Association Rule Mining, Keyword Based Filtering	N/A	The study shows that the system successfully recommends books based on user interests, increasing satisfaction and productivity.	Addressing issues related to online payment systems, customer order tracking, and order management (cancellations and

						User feedback improves recommendation relevance, while a hybrid approach delivers more accurate results compared to single-method systems.	replacements). Enhancing the recommendation system by incorporating additional features and improving the algorithms used for recommendations.
(Kanwal et al., n.d.)	12 February 2021	Safia Kanwal, Sidra Nawaz, Muhammad Kamran Malik, Zubair	The paper discusses several datasets for text-based recommendation systems, including Yelp, Docear, Book-Crossing, Plista and Globo, each serving	Collaborative Filtering (CF) and Content-Based Filtering (CB). Deep Learning models, particularly Recurrent Neural Networks (RNNs) and embeddings. Specific models: CTransR-CF algorithm and	The evaluation metrics mentioned include Precision, Recall, F-measure, Mean Absolute Error, Mean Squared Error, and others.	The study reveals that most research focuses on English textual data, with news recommendation being the most popular domain. Word Embedding is the preferred feature selection technique in recent studies.	Challenges include a limited exploration of non-English datasets and addressing specific issues in recommendation systems. Future work should focus on enhancing hybridization techniques and

			different domains and purposes.	Sem-RevRec model.			expanding dataset diversity.
(Rahman & Nabil, n.d.)	2022	Md. Mijanur Rahman ,Ismat Ara Shama, Siamur Rahman and Rahmat ullah Nabil	MovieLens , Synthetic Data	K-Nearest Neighbors (KNN), Matrix Factorization, Content-based Filtering	Precision , Recall, F1 Score, RMSE	The hybrid approach effectively addresses the cold start problem by combining collaborative filtering with content-based methods.	Further research needed to improve scalability and adapt to diverse user preferences.
(Gogula, et al., n.d.)	5 Januar y 2023	Gogula, S.D.; Rahouti, M. et al.	Cloud-base d User Reviews	Sentiment Analysis + Collaborative Filtering	Accurac y, RMSE	Sentiment-bas ed ratings enhance the personalized recommendati on process.	Handling diverse emotions in reviews and reducing bias.
(Wayesa et al., n.d.)	6 March 2023	Fikadu Wayesa, Mesfin	The research utilizes the	The paper proposes a hybrid	The effective ness of	The proposed hybrid model outperforms	The paper highlights challenges in

		Leranso , Girma Asefa & Abdulje bar Kedir	Good Books dataset from Harper and Konstan (2015)	recommendatio n system that combines Content-based Filtering (CBF) and Collaborative Filtering (CF),Informatio n Extraction (IE)	the proposed model is evaluate d using standard metrics such as precision , recall, and F1-score	existing models, achieving 63.84% precision, 40.42% recall, and a 52.1% F1-score. It effectively addresses challenges in recommending books to new users and items with limited rating history, making it a more robust solution for personalized recommendati ons.	predicting preferences for new users without demographic data and recommendin g items with no ratings. Future work will aim to enhance recommendati on accuracy in these scenarios and explore advanced data mining techniques to improve overall system performance.
(ahmed & letta, n.d.)	11 March 2023	Esmael Ahmed and	5189 records and 76,888 ratings	Algorithms: K-nearest neighbor (KNN),	Root Mean Squared Error	The study reveals that the SVD algorithm	The research faces challenges like the cold

		Adane Letta	from the University of Gondar student information system and online catalog system.	Singular Value Decomposition (SVD).	(RMSE), 5-fold cross-validation.	outperforms KNN in book recommendations, achieving an accuracy score of 85%. Effective data preprocessing significantly enhances the model's performance and recommendation quality .	start problem for new users. Future work should explore integrating collaborative filtering with demographic data and leveraging implicit feedback to enhance recommendation systems.
(raza et al., n.d.)	18 July 2024	Shaina Raza, Mizanur Rahaman, Safiullah Kamawal, Armin Toroghi, Ananya Raval,	N/A	Traditional methods: Collaborative filtering, content-based filtering, and hybrid approaches. Advanced methods: Deep learning techniques including CNNs,	The paper suggests that RS should be evaluated on various criteria, including: Model Accuracy; Diversity and	Highlights the shift from traditional to advanced recommendation system (RS) techniques. Emphasizes the integration of deep learning and graph-based models.	Challenges include data quality, authenticity, privacy issues, and the need for scalable solutions in real-world applications. Future work should focus on addressing

		Farshad Navah		RNNs, GNNs, and reinforcement learning. Specialized systems: Context-aware, review-based, and fairness-aware recommender systems.	Serendipity; Scalability, Interpretability, Computational Efficiency, and Reproducibility.	Deep learning-based RS improves personalization and captures complex user-item interactions.	emerging technological and societal trends, ensuring trustworthy and real-time recommendations.

Table 1 : List of Research Papers Studied

2.2 Integrated survey of Literature studied

This survey of book recommendation systems explores a wide range of approaches, from traditional collaborative filtering and content-based filtering to hybrid methods integrating social networks, context-awareness, sentiment analysis, and deep learning.

Liu and Jiao (2020) enhance recommendations with social and context-based insights.

Fayyaz et al. (2020) review collaborative filtering techniques, addressing challenges like data sparsity and the cold start problem.

Mathew et al. (2023) combine content-based and collaborative filtering to improve user satisfaction.

Kanwal et al. (2021) gives more importance to deep learning models for text-based systems, showcasing the need for non-English dataset diversity.

Gogula and Rahouti (2023) sentiment analysis system, focus on personalized recommendations based on user emotions.

Wayesa et al. (2023) and Ahmed and Letta (2023) solve cold start issues with the help of hybrid models like KNN and SVD.

Raza et al. (2024) review advanced recommender systems, rooting for deep learning and graph-based models to meet changing technological and societal needs.

Together, these studies highlight the trend towards hybrid, scalable, and contextually-aware recommendation systems, advancing personalization and accuracy for diverse user needs.

CHAPTER 3 : REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Overall Description of Project

a) Product Perspective

The book recommendation system is designed to assist users in discovering books based on their interests, preferences, and reading habits. It employs a content-based filtering approach using textual features such as book titles, authors, subjects, and ratings. The system fits into the broader category of recommendation engines, commonly used in e-commerce, media streaming, and online libraries.

b) Product Functions

1. **Best-matched Books:** Books are recommended based on students interest which further depends on their ratings and previous reading choices.
2. **Books from Similar Series:** If a student is enjoying a particular series, it will suggest other books in the same series to keep them engaged.
3. **Cross-referenced Book Recommendations:** Books with similar themes, genres, or authors are recommended to users who have enjoyed certain books. This helps them discover new reads of similar vibe.
4. **Content-based Recommendations:** The project looks at the content of books (such as the genre, author, or themes) and suggests books with similar content to what the user has liked and read in the past.
5. **Search & Recommendations:** students can also search for specific books, and it will suggest similar books based on that search.
6. **Book Recommendations for topics in the course module:** Books are recommended for each topic in the subtitle of the module in the course module pdf uploaded.

c) User Characteristics

The target users of the system include:

1. **Casual Readers:** Individuals looking for personalized book recommendations to discover new reads.
2. **Librarians and Bookstore Managers:** Professionals seeking to enhance user engagement by providing tailored recommendations.
3. **Students and Researchers:** Individuals looking for books on specific topics or authors.

d) Constraints

1. **Data Availability:** The quality and quantity of book metadata directly impact recommendation accuracy.
2. **Computational Resources:** Real-time recommendations for large datasets may require significant memory and processing power.
3. **Algorithmic Simplicity:** The current implementation uses content-based filtering, which may not perform well for users with sparse interaction history.
4. **Language and Cultural Biases:** Recommendations may favor books with rich metadata in certain languages or regions.

e) Assumptions and Dependencies

1. **Quality Metadata:** It is assumed that the dataset contains accurate and complete metadata for books, including title, author, ratings, and subjects.
2. **Static Dataset:** The system is currently designed to work with a preloaded dataset and may not handle dynamic updates in real time.
3. **User Feedback:** Users will provide feedback to refine and improve recommendations, but this assumption may not always hold true.

f) Apportioning of Requirements

- **Core Functionality:** The initial focus will be on content-based recommendations and basic search functionalities.
- **Extended Features:** Advanced features like collaborative filtering, hybrid recommendation systems, and user account management will be postponed for future iterations.
- **Scalability:** Current development will prioritize small to medium datasets, with scalability for larger datasets considered in future upgrades.

Vision:

The vision for our project is to help students find books that make them feel natural and personal.Our project learns user's tastes over time and suggests books based on it, whether it's more from your favorite series,your past favorites, or fresh picks based on themes you care about. The project has features that aim to turn book discovery into an experience that's as engaging and enjoyable as the reading itself.

3.2 Requirement Analysis

Functional Requirements

1. **Data Loading and Preparation :** Import and prepare book datasets with information such as titles, authors, genres, ratings, and languages. Ensure the data is cleaned and ready for processing.
2. **Focused Data Collection :**Use a web scraper to gather only books categorized under the "Computer Science" genre, ensuring data relevance.
3. **Personalized Book Recommendations :** Implement a recommendation engine that uses content-based filtering to suggest books tailored to user preferences or based on selected books.
4. **Interactive User Interface :** Provide a search feature where users can look up books by title, author, or keyword. Allow users to explore recommended books in an intuitive way.

5. **Handling Missing Information :** Detect missing data in the book dataset and address it through techniques like filling in missing values (imputation) or excluding incomplete entries.
6. **Recommendation Presentation :** Display book recommendations clearly and attractively, using tables or visual elements to enhance user experience.
7. **PDF Upload for Course Outcomes :** Enable users to upload PDFs that contain course outcomes. Ensure only valid PDF files are accepted.
8. **Text Extraction from PDFs :** Extract text from uploaded PDFs, focusing on identifying key topics, keywords, or phrases related to course outcomes.
9. **Matching Keywords with Books :** Compare extracted keywords with book metadata (e.g., title, genre, description) to find relevant books.
10. **Topic-Specific Recommendations :** Generate a list of recommended books for specific course topics or modules. Provide filters for users to refine recommendations by genre, publication year, or rating.
11. **Results Display and Download :** Show recommendations on the interface and offer a downloadable report (e.g., in PDF or Excel format).
12. **User Feedback Collection (Future Feature) :** Add a mechanism to gather feedback on recommendations for system enhancements over time.

Non-Functional Requirements

These requirements outline the quality attributes of the system:

1. **Performance :** The system should generate recommendations within a few seconds for datasets with up to 1,000,000 records and ensure efficient computation of similarity metrics using optimized algorithms. The system should process and analyze a PDF file (up to a standard size, e.g., 5 MB) within a few seconds.
2. **Scalability :** The system should scale to handle larger datasets as user adoption grows, possibly integrating distributed computing.The feature should be capable of handling multiple file uploads or larger PDF files in the future.

3. **Accuracy :** Provide meaningful and accurate recommendations by leveraging content similarity metrics like TF-IDF. The system must accurately identify and match keywords from course outcomes with relevant book metadata.
4. **Usability :** The interface must be intuitive for users with minimal technical expertise, including clear instructions and accessible designs. The upload process should be simple and user-friendly, with clear instructions and feedback.
5. **Maintainability :** The system should be modular, enabling updates to algorithms, datasets, or user interfaces with minimal disruption. The codebase for text extraction and matching should be modular and easy to extend for future updates.

3.3 Solution Approach

1. **Data Collection and Preprocessing :**
    - **Data Sources:** To guarantee a complete dataset, the project uses information on books from APIs like Google Books or Goodreads, including titles, authors, genres and ratings.
    - **Data Cleaning:** This phase deals with any missing or noisy data. For example, text data is preprocessed to make it appropriate for similarity calculations, and incomplete records are taken out.
    - **Data Transformation:** To produce an accurate summary of each book, features are taken from ratings, categories, and book descriptions. Since content-based recommendation algorithms are being used sp text data is vectorized.
2. **Recommendation Techniques :**
    - **K-Nearest Neighbour (KNN) Algorithm:** The KNN algorithm finds similar books based on feature vectors that represent book attributes (e.g., genre, author, keywords).
        - **Feature Representation:**

          Each book is represented as a vector in an n-dimensional space.

          We use attributes like title,genre and author_name, a book vector looks like:

          v=[title,genre_list,author similarity]

- **Similarity Measure:**

  The similarity between books is computed using a cosine similarity :

  $$\text{Cosine Similarity (A, B)} = (A \cdot B) / (||A|| * ||B||)$$

- **Algorithm Steps:**

1. Compute the similarity or distance between the input book and all other books.
2. Rank the books based on similarity.
3. Recommend the top k most similar books.

- **Singular Value Decomposition (SVD):** It is applied to a combined feature vector of title, genre, and author name.
  - **Feature Representation:**

    Each book is represented as a vector in an n-dimensional space.

    We use attributes like title,genre and author_name, a book vector looks like:

    v=[title,genre_list,author similarity]
  - **Building the Feature Matrix:**

1. **Matrix Construction:**

   A feature matrix M is constructed where each row represents a book, and columns represent combined features (title, genre, author). For n books and f features, M has dimensions n×f:

$$M = \begin{bmatrix} \mathbf{v}_{\text{book}_1} \\ \mathbf{v}_{\text{book}_2} \\ \vdots \\ \mathbf{v}_{\text{book}_n} \end{bmatrix}$$

2. **Matrix Decomposition:**

   Decompose M using SVD:

$$M \approx U\Sigma V^T$$

- U: Captures book-specific latent factors.
- Σ: Diagonal matrix with singular values indicating the importance of latent features.
- V^T: Captures latent relationships between features (title, genre, author).

- **Content-based Filtering:** The project uses book properties (such as genre, author,and titles) to make content-based suggestions.
  - **Feature Representation:**

    Each book is represented as a vector in an n-dimensional space.

    We use attributes like title,genre and author_name, a book vector looks like:

    v=[title,genre_list,author similarity]
  - **Similarity Measure:**

    Content-based filtering also uses similarity metrics like cosine similarity to compare books.
  - **Algorithm Steps:**
    1. For an input book, find its feature vector.
    2. Compute similarities with all other books.
    3. Rank books based on similarity scores.
    4. Recommend the most similar books.

3. **Recommendation Features :**

   The project's recommendation features focus on providing specific book recommendations using a combination of advanced algorithms and user-centered approaches. Using K-Nearest Neighbors (KNN) and Singular Value Decomposition (SVD), the system adjusts recommendations to each user's reading history and preferences. For users who recognize specific series or genres, the system makes series-based and cross-referenced recommendations, increasing engagement by providing related novels or sequels to recognized topics. Further, when a user searches for a book, our project not only returns that book, if it exists but also recommends similar books for further research and discovery. Moreover one additional feature is also added which recommends books based on the modules in the uploaded course outcome pdf.

4. **Evaluation :**

   Our project implements metrics like recommendation accuracy, precision, and recall to evaluate the algorithms. Adjustments are made to improve the relevance of recommendations based on testing results.

CHAPTER 4 : MODELING AND IMPLEMENTATION DETAILS

4.1 Design Diagrams

4.1.1 Class Diagram

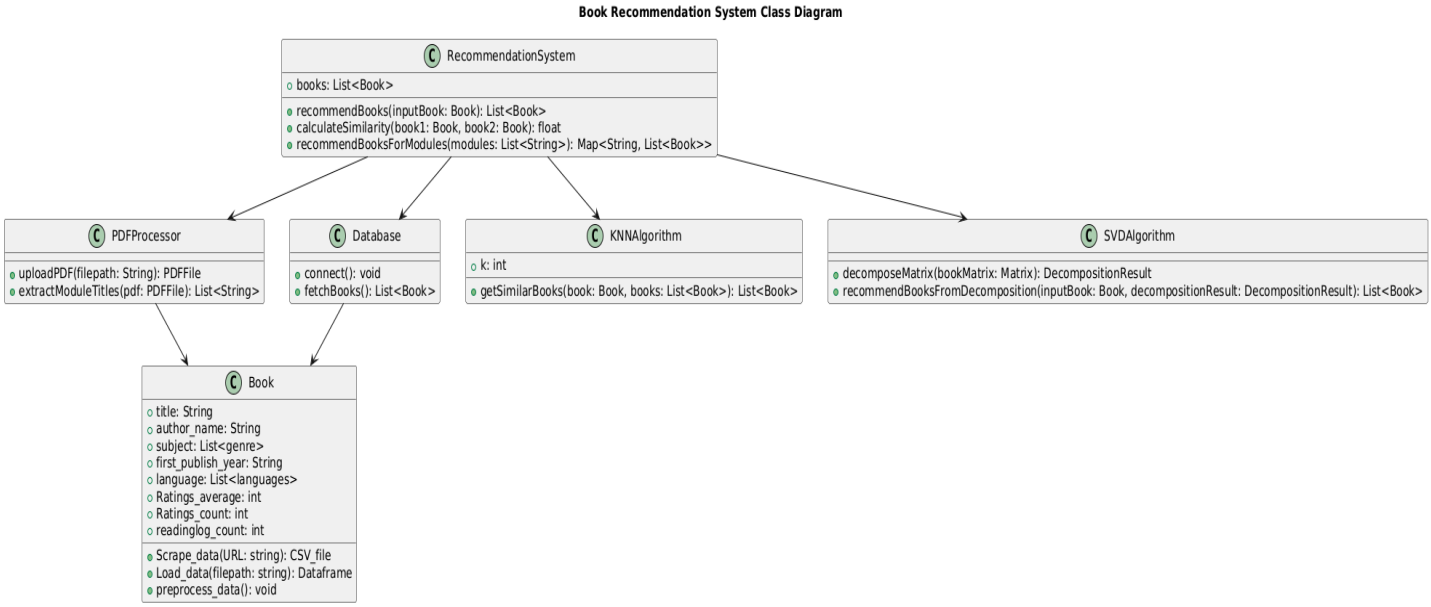

Fig 1 : Class Diagram showcasing the entities and their attributes in the book recommendation system

4.1.2 Use Case Diagram

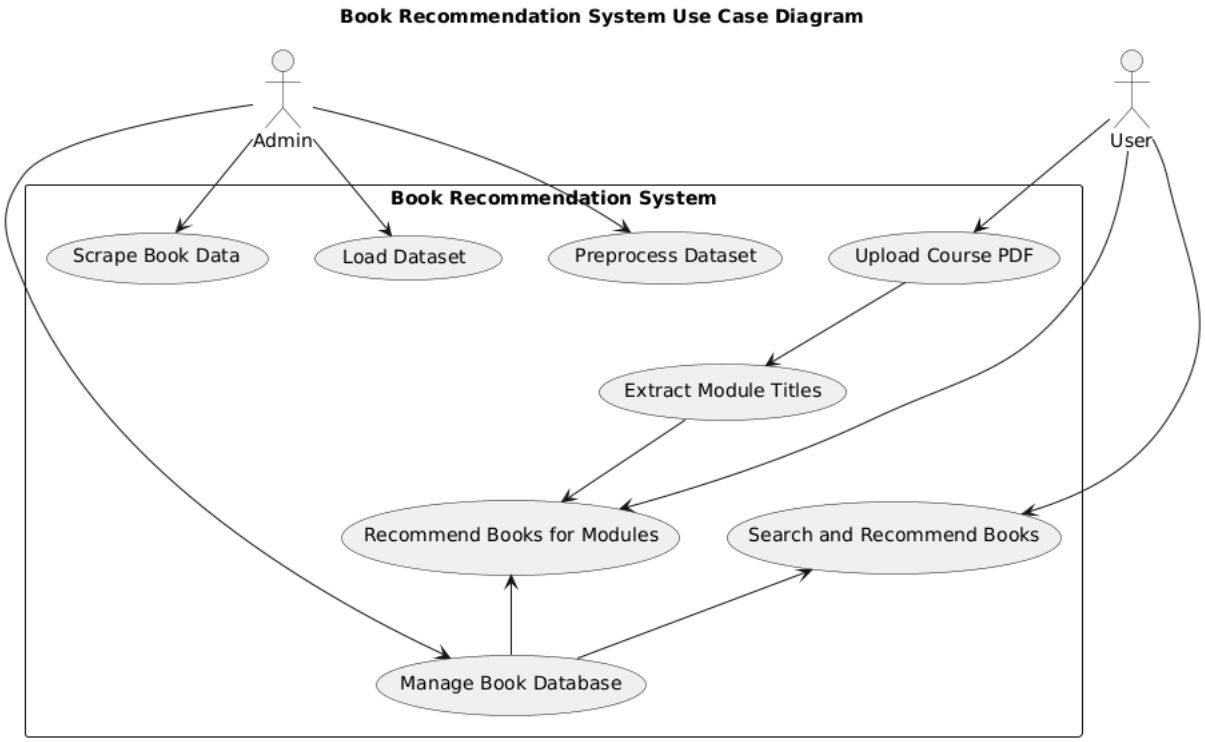

Fig 2 : Use Case Diagrams representing user and admin interactions

4.1.3 Activity Diagram

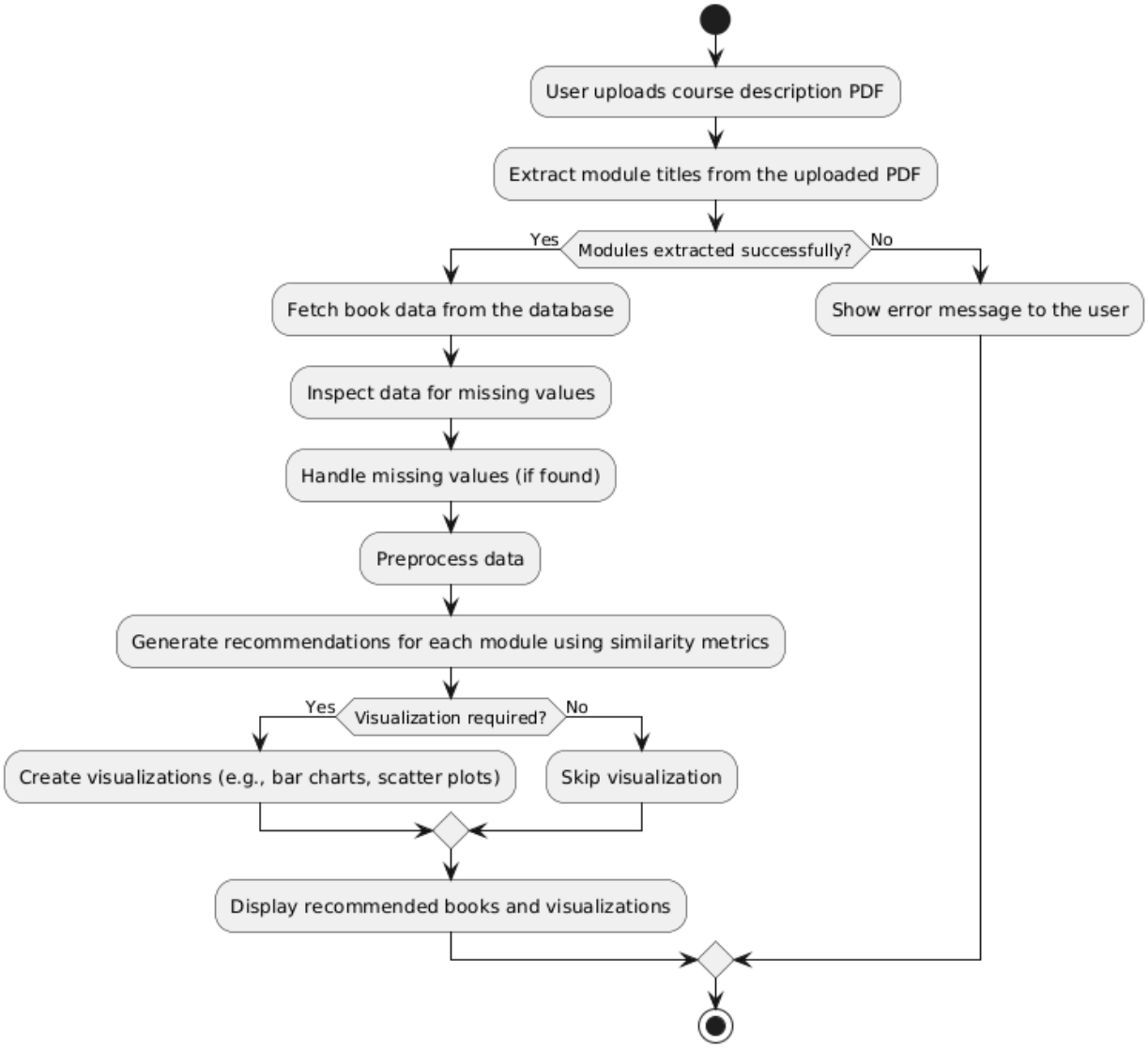

Fig 3 : Activity diagram showing the workflow from User end

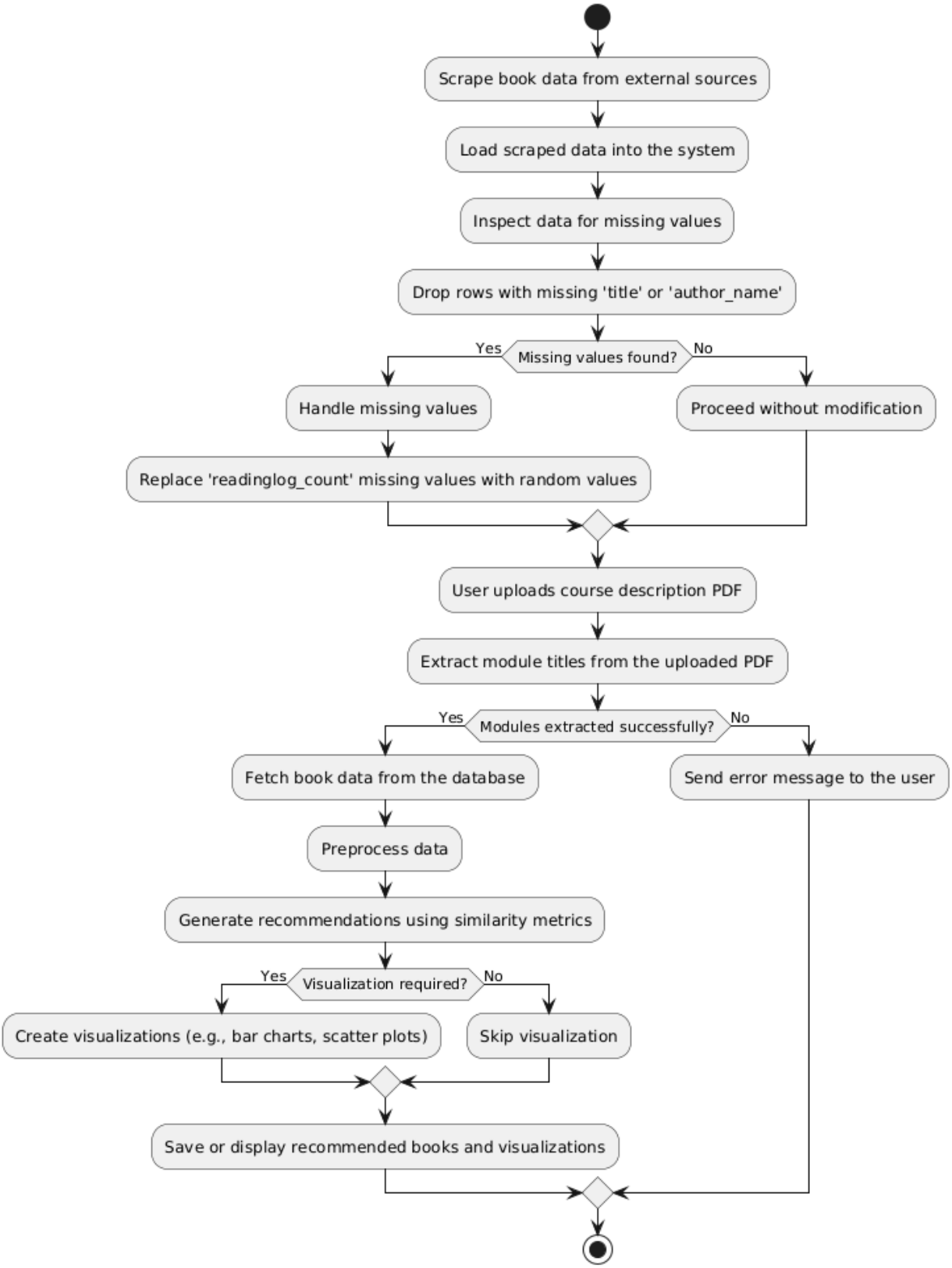

Fig 4 : Activity diagram showing the workflow from system end

4.1.4 Sequence Diagram

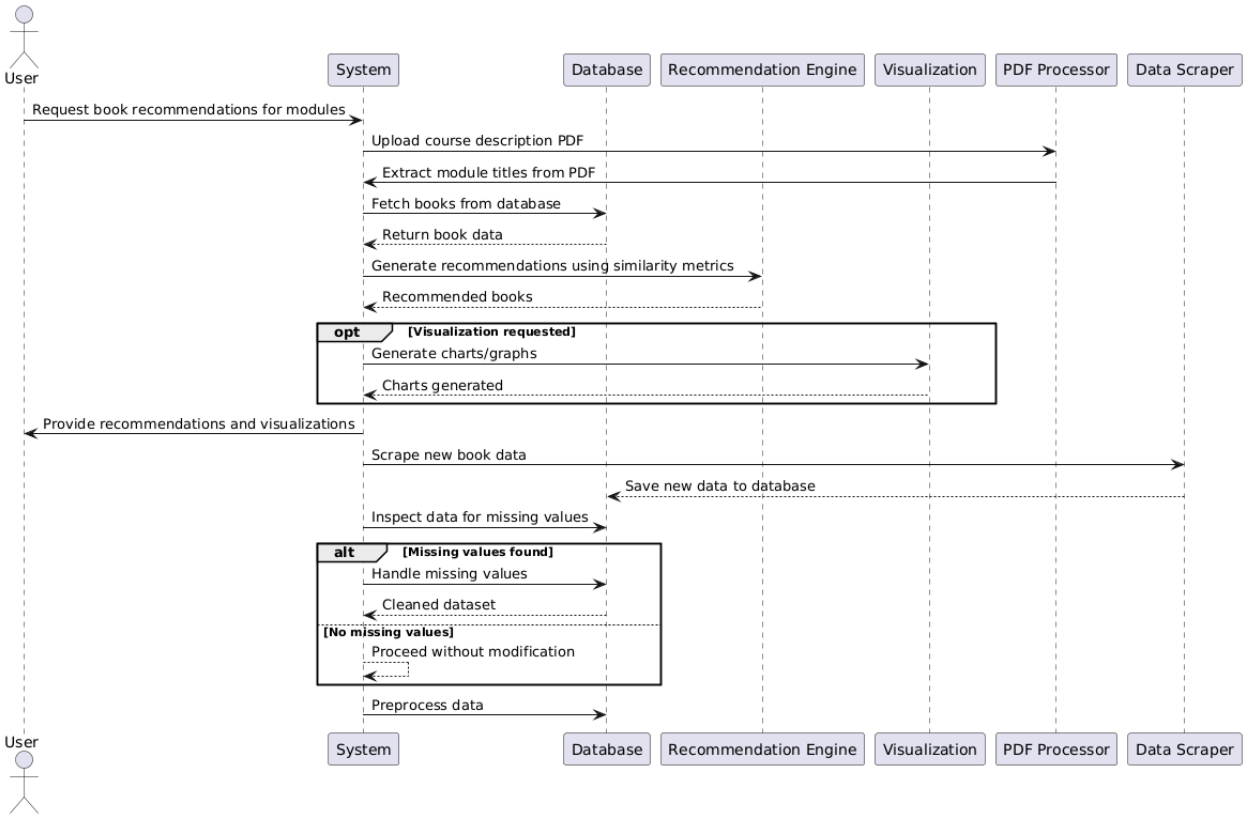

Fig 5 : Sequence diagram showing interactions between users , systems and its components for generating recommendation

4.2 Implementation Details

- **Dataset Overview**: The dataset contains information about 10,000 Computer Science books, scraped using the Open Library API. It includes key details like: **Book Title , Author Name, Genre , Publication Year ,Rating**
- **Preprocessing the Data**: The data was cleaned to ensure consistency and usability. This likely involved steps like:
    - Handling missing or incomplete information.
    - Organizing genres and removing duplicates.
- **Exploring the Data**: Visualizations and summaries helped uncover trends, such as:
    - Popular genres.
    - Publication year patterns.
    - Average ratings for different categories.
- **Building the Recommendation System**: The system includes one or more recommendation techniques:
    - **Content-Based Filtering**: Recommends books based on attributes like genre or author.
    - **KNN**: Finds and recommends books that are most similar to the input book based on attributes.
    - **SVD**: Uses matrix factorization to uncover latent features and recommend books based on genre and title.
- **Recommending Books Based on Module Topics :** The system extends its functionality by recommending books for topics derived from module subtitles. This feature involves:
    - **Substring Extraction:** Breaking down module titles into key phrases and searching for relevant books for each substring.
    - **Aggregating Recommendations:** Combining results from all substrings, ranking books by relevance, and presenting the top 10 recommendations for each module.
    - **Example Output:** For the module "Network Concept," the system recommends books like "Network Security Assessment" and "Cryptography and Network Security" based on computed similarity scores.
- **Measuring Performance**:
    - Metrics like accuracy and precision/recall are used to evaluate how well the recommendations work.
- **Tools Used**:
    - Common Python libraries like pandas (for data manipulation), numpy (for calculations), and visualization tools like matplotlib and seaborn.

4.3 Risk Analysis and Mitigation

Risk ID	Classification (SEI Taxonomy)	Description of Risk	Risk Area	Probability	Impact	RE (P * I)
1.	Development Process	Inadequate familiarity with SVD and KNN models	Model Performanc e	High (5)	High (5)	25
2.	Development System	Scalability issues due to a large dataset (10,000 books)	Performanc e	Medium (3)	High (5)	15
3.	Resources	Limited computation al resources for model training	Program Constraints	High (5)	Mediu m (3)	15
4.	Data Integrity	Incomplete or incorrect metadata from Open Library API	Data Quality	Medium (3)	High (5)	15
5	Work Environment	Collaboratio n issues within the team	Personnel	Medium (3)	Mediu m (3)	9
6.	Development System	Unreliable course description parsing (PDF processing)	Module Matching Quality	Medium (3)	Mediu m (3)	9

Table 2 : Risk's identified in the project

4.3.1 Interrelationship Graph (IG):

Weighted Relationships:

- Model Performance (1) → Data Quality (2) (Weight: 9): Poor data quality significantly impacts recommendation accuracy.
- Performance(3) → Program Constraints (4) (Weight: 3): Scalability issues strain computational resources.
- Personnel(5) → Work Environment (6) (Weight: 9): Team collaboration affects overall productivity.
- Module Matching Quality (7) → Model Performance(1) (Weight: 3): Parsing errors degrade module-topic recommendations.

Graph Representation:

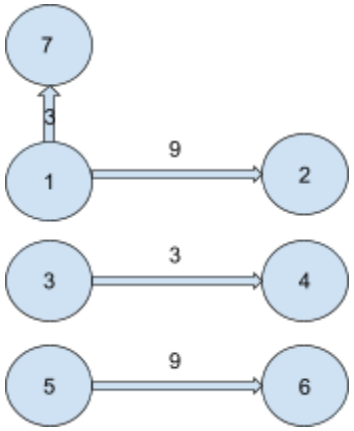

Fig 5 : Interrelationship Graph

4.3.2 Mitigation Plans:

Risk ID	Risk Statement	Mitigation Approach	Resources Needed
1.	Inadequate familiarity with SVD and KNN models	Learnt and discussed ML algorithms.	Online tutorials and articles, research papers.
2.	Scalability issues due to large datasets	Implement dimensionality reduction and parallel computing.	Python libraries.
3.	Limited computational resources	Restricted the size of dataset to efficiently use the resources.	Data preprocessing tools.
4.	Incomplete or incorrect metadata	Validate data integrity during preprocessing.	Data cleaning tools.
5.	Collaboration issues within the team	Use collaborative tools like Git for efficient tracking.	Communication.

Table 3 : Mitigation Approach for the risks identified.

CHAPTER-5 TESTING (FOCUS ON QUALITY OF ROBUSTNESS AND TESTING)

5.1 Testing Plan

Type of Test	**Will Test Be Performed? (Yes/No)**	**Comments/Explanations**	**Software Component**
Requirements Testing	Yes	Ensures all functional requirements (content-based filtering, KNN, SVM, accuracy calculations) and module title extraction from uploaded course description PDFs are met, along with correct handling of Open Library data.	Entire Recommendation System
Unit Testing	Yes	Individual components like the PDF parser, content-based filtering, KNN, and SVM algorithms will be tested separately for correctness.	PDF extraction module, Filtering module, KNN algorithm, SVM model
Integration Testing	Yes	Validates the integration between the content-based module, KNN, and SVM for end-to-end functionality.	Combined recommendation pipeline

Performance Testing	Yes	Measures system performance for large datasets (including 10,000 books from Open Library) to ensure smooth operation.	Recommendation system pipeline
Compliance Testing	No	Not applicable as the project does not target industry regulations.	N/A
Security Testing	No	Not applicable since the project does not handle sensitive or user-specific data.	N/A
Load Testing	Yes	Ensures the system can handle a high number of queries and maintain performance.	Recommendation engine
Volume Testing	Yes	Tests the system's capability to handle a large number of books and attributes and course descriptions.	Dataset handling module

Table 4 : Different Types of Testing Required

Test Team Details

Role	Name	Responsibilities
Tester	Vanshika Jalhotra	Tested recommendations made using SVD-algorithm and the overall recommendation pipeline accuracy.
Tester	Vidhi Rastogi	Tested data scraping, data loading, preprocessing and content-based filtering.
Tester	Himral Garg	Tested recommendations of KNN model and PDF module extraction and topic wise recommendations.

Test Environment

Software Items

- Operating System: Windows and MacOS
- Software: Jupyter Notebook, Python 3.9+, NumPy, Pandas, scikit-learn, Matplotlib, Google Collab
- Migration: Dataset and models will be imported and utilized within the Jupyter Notebook for testing.

Hardware Items

- Processor: Quad-core 2.5 GHz or higher
- RAM: 8 GB or more
- Storage: 50 GB (for dataset and temporary files)

5.2 Component decomposition and type of testing required

S. No.	List of Various Components (Modules)	Type of Testing Required	Technique for Writing Test Cases
1.	Data Loading and Preprocessing	Requirement, Unit, Volume	Black Box: Test for handling missing values, edge cases with empty/duplicate rows, boundary value analysis for column data types.
2.	Similarity Calculation	Unit, Integration, System	Black Box: Validate output similarity scores. Robustness testing for invalid/malformed inputs.
3.	Recommendation Pipeline (End-to-End)	Integration, System, Performance	Black Box: Validate recommendations' accuracy and relevance.
4.	Accuracy Calculation	Unit, Integration	White Box: Test mathematical formula accuracy with different scenarios. Ensure values match manually calculated expectations.
5.	Error Handling and Logging	Unit	Black Box: Ensure meaningful error messages. Test log generation for invalid inputs.

6.	PDF Upload and Module Title Extraction	Requirement, Unit, Integration, System	Black Box: Validate PDF parsing, accurate module title extraction, and recommendation generation. Test robustness with invalid PDFs.

Table 5 : Component Decomposition and types of testing required

5.3 List all test cases

Unit 1: Data Loading and Processing

Test Case ID	Input	Expected Output	Status
TC_1_01	Dataset with all required columns	Successful loading and correct column format	Pass
TC_1_02	Dataset with missing title values	Rows with missing titles dropped	Pass
TC_1_03	Empty dataset	Error logged, no crash	Pass

Unit 2: TF-IDF Vectorization

Test Case ID	Input	Expected Output	Status
TC_2_01	Text with common words	Low scores for common words	Pass

TC_2_02	Empty text	Vectorization skipped, error logged	Pass
TC_2_03	Large text corpus	Successful vectorization, no memory errors	Pass

Unit 3: Similarity Calculation

Test Case ID	Input	Expected Output	Status
TC_3_01	Two similar texts	High similarity score	Pass
TC_3_02	Two unrelated texts	Low similarity score	Failed
TC_3_03	Missing/empty input	Graceful handling, error logged	Pass

Unit 4: Recommendation Pipeline

Test Case ID	Input	Expected Output	Status
TC_4_01	Valid user query	Accurate top 10 recommendations	Pass
TC_4_02	Invalid query (nonexistent book)	Empty recommendations, handled gracefully	Pass
TC_4_03	Large dataset	Timely recommendations without performance lag	Pass

Unit 5: Accuracy Calculation

Test Case ID	Input	Expected Output	Status
TC_5_01	Manually calculated accuracy values	Matching system-calculated values	Pass
TC_5_02	Input with extreme scenarios (e.g., no correct matches)	Correct handling of edge cases	Failed

Unit 6: PDF Upload and Module Title Extraction

Test Case ID	Input	Expected Output	Status
TC_6_01	Valid course description PDF	Accurate extraction of module titles	Pass
TC_6_02	Invalid file format (e.g., .txt, .docx)	Error message displayed	Pass
TC_6_03	PDF with complex formatting	Extraction of meaningful titles or graceful failure	Failed

Table 6 : List of all Test Cases

5.4 Error and Exception Handling

Test Case ID	Test Case for	Debugging Technique	Explanation
TC_3_02	Two Unrelated Texts (Unit 2: TF-IDF vectorization)	Remote Debugging	Similarity score issue due to misalignment in feature vectors. Used remote debugging to step through the process and analyze data.
TC_5_02	Input with Extreme Scenarios (No Matches) (Unit 5: Accuracy Calculation)	Print (or tracing) debugging	No correct matches caused failure. Added debug prints to verify input processing and similarity calculations.
TC_6_03	PDF with Complex Formatting (Unit 6: PDF Upload and Module Title Extraction)	Backtracking	The title extraction failed due to complex formatting. Tracked the issue to incorrect handling of PDF structures.

Table 7 : Error and Exception Handling

5.5 Limitations of the Solution

While the book recommendation system works well, there are a few areas where it could be improved:

1. **Incomplete Data:**

The system depends on data from the Open Library API, which sometimes lacks details like ratings or subjects. This can impact the quality of recommendations, especially for less popular books.

2. **Language Limitation:**

Currently, the system only supports English-language books, so users looking for recommendations in other languages won't benefit from the full experience.

3. **Scaling Issues:**

As the dataset grows, the system might slow down. It's optimized for around 10,000 books, but more data could lead to performance issues.

4. **PDF Parsing Challenges:**

Extracting module titles from course description PDFs works well for simple files, but more complex PDFs with mixed formatting might cause errors or incomplete recommendations.

5. **Text Similarity Limitations:**

The content-based filtering approach works well but can miss deeper connections between books with different wording on similar topics. Advanced methods could improve this but add complexity.

6. **Model Performance:**

The KNN and SVM models could struggle with overfitting or underfitting, affecting the accuracy of recommendations in some cases.

7. **No User Feedback:**

The system doesn't learn from user preferences. Adding feedback would allow it to offer better recommendations over time.

8. **Reliance on Open Library API:**

If the Open Library API experiences downtime or updates, it could affect the data and recommendations, limiting the system's reliability.

CHAPTER 6 : FINDINGS , CONCLUSION AND FUTURE WORK

6.1 Findings

1. **Implementation of a Content-Based Recommender System**

   The system successfully employs a content-based filtering approach using TF-IDF vectorization and cosine similarity. Recommendations are generated based on book metadata like titles, authors, and genres.
2. **Web Scraping for Dataset Generation :** The dataset was created by scraping book information using the Open Library API. Approximately 10,000 books related to Computer Science were collected, focusing on metadata such as book titles, authors, genres, publication years, and average ratings. The scraping process involved:
   - Automated Data Collection: Open Library API was used to extract relevant metadata for each book efficiently.
   - Handling Missing Data: Certain fields like genre and average ratings were occasionally missing in the API responses. These gaps were handled during preprocessing.
   - Data Structuring: The scraped data was organized into a structured dataset, making it compatible with the recommendation system. Fields were standardized to ensure uniformity across all entries.
3. **Dataset Analysis and Preprocessing**

   The dataset contains multiple fields such as title, author, average ratings, and genres, which were preprocessed to remove missing values and standardize formats. However, some columns required more advanced handling due to sparsity or incomplete information.
4. **System Efficiency**
   - The similarity computation using linear_kernel is efficient for small to medium-sized datasets but may face performance challenges with larger datasets.
   - The current implementation does not support real-time updates to the dataset, which could limit scalability.

5. **Integration of Topic-Based Book Recommendations :** The system extends its capabilities by recommending books based on topics extracted from module subtitles. This implementation allows tailored book suggestions for specific academic or thematic modules, enhancing its applicability in educational and research contexts.
    - **Topic Extraction** : Module subtitles are split into meaningful substrings to capture key concepts within the topics. For example, the subtitle "Network Concept" generates substrings like "Network", "Concept", and "Network Concept" to broaden the search scope.
    - **Recommendation Process :** For each substring, the system leverages the get_recommendations function to retrieve books related to the extracted topics. This ensures that all facets of the topic are explored during the recommendation process.
    - **Aggregating and Ranking Recommendations :** Recommendations from all substrings are aggregated, and the books are ranked based on their similarity scores (distances computed using KNN).
    - **Top 10 Recommendations :** Only the top 10 books with the highest relevance to the module topic are displayed, ensuring concise and focused results for users.
6. **Recommendation Accuracy**

   While the content-based approach provides relevant recommendations, it may bias towards popular or frequently rated books, potentially neglecting less-known options.

6.2 Conclusion

The project demonstrates the viability of a content-based book recommendation system that leverages metadata to suggest relevant books. The methodology provides a good foundation for developing more sophisticated recommendation systems, particularly for niche datasets or applications with minimal user interaction history.

However, the system has certain limitations:

- It relies heavily on metadata quality and completeness.
- It cannot handle cold-start problems effectively (e.g., for new users or books without prior data).

- The model lacks diversity in its recommendations due to its content-based nature.

Despite these limitations, the project showcases a solid baseline for creating recommendation systems with room for expansion and enhancement.

6.3 Future Work

1. **Integration of Collaborative Filtering**
   Combine content-based and collaborative filtering to create a hybrid system that leverages both user preferences and book metadata for improved recommendations.
2. **Improved Scalability**
   Implement advanced techniques like Approximate Nearest Neighbors (ANN) for faster similarity calculations on large datasets.
3. **Dynamic Dataset Updates**
   Enable real-time updates to the dataset, allowing for seamless inclusion of newly published books and user-generated data.
4. **Enhanced Diversity in Recommendations**
   Develop mechanisms to introduce more diversity in recommendations, reducing biases towards popular or frequently rated books.
5. **Cold-Start Problem Mitigation**
   Implement strategies like clustering or fallback recommendations for cases with limited data on users or books.
6. **User Personalization and Feedback Integration**
   - Build user profiles based on interaction history, including clicks, ratings, or time spent on recommended books.
   - Incorporate feedback loops to refine recommendations over time.
7. **Visualization and User Interface Improvements**
   Create interactive dashboards or interfaces that allow users to explore recommendations visually, enhancing usability and engagement.

REFERENCES

[1] S. Kanwal, S. Nawaz, M. K. Malik and Z. Nawaz, "A Review of Text-Based Recommendation Systems," in IEEE Access, vol. 9, pp. 31638-31661 [February, 2021], [Online]. Available: https://ieeexplore.ieee.org/document/9354169

[2] Esmael Ahmed and Adane Letta, "Book Recommendation Using Collaborative Filtering Algorithm," Applied Computational Intelligence and Soft Computing, vol. 2023, issue 1, pp. 12, [July, 2023]. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2023/1514801

[3] Shaina Raza, Mizanur Rahaman, Safiullah Kamawal, Armin Toroghi, Ananya Raval, Farshad Navah , "A Comprehensive Review of Recommender Systems: Transitioning from Theory to Practice," arXiv, [July 2024]. [Online]. Available: https://arxiv.org/html/2407.13699v1

[4] "A casual intro to recommendation models," YouTube, uploaded by Tunadorable,[August,2024]. [Online]. Available: https://youtu.be/jXpu13OeClI?si=sM70B0hpNE9ibq5X.

[5] Md. Mijanur Rahman, Ismat Ara Shama, Siamur Rahman and Rahmatullah Nabil " Hybrid Recommendation System to Solve Cold Start Problem" [June 2022] Vol 100 No 11 [Online].Available:https://www.researchgate.net/publication/364357987_HYBRID_RECOMMENDATION_SYSTEM_TO_SOLVE_COLD_START_PROBLEM

[6] Liu, H.; Jiao, N , "A Hybrid Book Recommendation Algorithm Based on Context Awareness and Social Network" [2 July 2020] [Online] Available : https://www.researchgate.net/publication/342652581_A_Hybrid_Book_Recommendation_Algorithm_Based_on_Context_Awareness_and_Social_Network

[7] Zeshan Fayyaz 1,Mahsa Ebrahimian 1,Dina Nawara 1,Ahmed Ibrahim 2 andRasha Kashef , "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities" [2 November 2020] [Online] Available : https://www.mdpi.com/2076-3417/10/21/7748

[8] Mathew, Praveena & Kuriakose, Bincy & Hegde, Vinayak. "Book Recommendation System through content based and collaborative filtering method"[2016] 4[Online] Available : https://www.researchgate.net/publication/311610181_Book_Recommendation_System_through_content_based_and_collaborative_filtering_method

[9]Sandhya Devi Gogula 1,Mohamed Rahouti 2,Suvarna Kumar Gogula 3,Anitha Jalamuri and Senthil Kumar Jagatheesaperumal. "An Emotion-Based Rating System for Books Using Sentiment Analysis and Machine Learning in the Cloud " [5 January 2023] [Online] Available : https://www.mdpi.com/2076-3417/13/2/773

[10]S. Kumar, "PDF Table Processing with Python," *Plain English*, Oct. 23, 2021. [Online]. Available: https://python.plainenglish.io/pdf-table-processing-with-python-5528f6302e28.

[11]Tabula, "Tabula: Extract tables from PDFs," GitHub repository, [Online]. Available: https://github.com/tabulapdf/tabula.

[12]H. E. Guler, "Reading PDF files as pandas DataFrame," *Medium*, Oct. 2, 2019. [Online]. Available: https://medium.com/@hikmetemreguler/reading-pdf-files-as-pandas-dataframe-62f1b7601c6c.

[13]"Recommender Systems using KNN," *GeeksforGeeks*, [Online]. Available: https://www.geeksforgeeks.org/recommender-systems-using-knn/.

[14]"SVD in Recommendation Systems," *GeeksforGeeks*, [Online]. Available: https://www.geeksforgeeks.org/svd-in-recommendation-systems/.

[15]"Open Library API," *Public APIs*, [Online]. Available: https://publicapis.io/open-library-api-api.