



MANIPAL UNIVERSITY

JAIPUR

A Major Project Synopsis on

Book Recommendation System

Submitted to Manipal University, Jaipur

Towards the partial fulfilment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

2023-2025

By

Yashashvi Shekhawat

23FS20MCA00045

Under the guidance of

Dr. Devershi Pallavi Bhatt

Department of Computer Applications

School of AIML, IoT&IS, CCE, DS and Computer Applications

Faculty of Science, Technology and Architecture

Manipal University Jaipur

2025

I. Introduction

The **Book Recommendation System** is a web-based platform designed to recommend books to users based on their preferences, reading history, and interests. The system utilizes a recommendation algorithm powered by machine learning techniques, implemented in **Python (Jupyter Notebook)**, which processes user data and predicts book recommendations. The frontend is designed using **HTML** to provide an interactive interface for users to view and select books. This project aims to simplify the book discovery process, allowing users to find books that suit their reading habits without manually searching through large lists.

Key Features:

- Personalized book recommendations based on user preferences.
- A dynamic interface built with **HTML** for users to interact with the system.
- Machine learning model (like **Collaborative Filtering** or **Content-Based Filtering**) for suggesting books.
- Efficient data processing and real-time book suggestions powered by **Python**.

This system enhances the user experience by streamlining book searches and providing recommendations tailored to each individual's taste.

II. Motivation

The book industry is vast, and with the sheer number of books available, readers often struggle to find books that meet their interests. Recommendation systems are widely used in various industries, such as e-commerce and media, to help users discover products or content that best suit their preferences. The motivation behind creating this **Book Recommendation System** is to automate the process of book discovery by utilizing machine learning techniques to offer personalized suggestions to users.

The goal is to help users easily discover new books based on their past reading history or preferences, making the book selection process enjoyable and time-efficient. In the context of online bookstores, libraries, and reading platforms, this recommendation system provides an essential service to enhance user engagement and increase book sales or library checkouts.

III. Problem Statement

- **For Users:**
 - Difficulty in finding relevant books due to overwhelming options.
 - Lack of personalized recommendations leading to a frustrating book search experience.

- Need for a system that can suggest books based on personal reading history or preferences.
- **For Book Sellers and Libraries:**
 - Need for tools that improve customer engagement by offering personalized book suggestions.
 - Need to provide users with relevant book recommendations, thus increasing customer satisfaction and retention.

IV. Methodology/Planning of Work

Week 1: Project Setup & Initial Development

- Finalize the tech stack: Python, Jupyter Notebook, HTML.
- Set up the environment and tools for Python (for data processing) and HTML (for frontend development).
- Collect initial dataset of books with attributes like title, author, genre, and rating.
- Plan the book recommendation algorithms (Collaborative Filtering, Content-Based, or Hybrid).

Week 2: Data Collection & Preprocessing

- Gather a dataset of books and user ratings (either from an API or a pre-existing dataset like **BookCrossing** or **Goodreads**).
- Clean and preprocess the data, handling missing values and normalizing ratings or other features.
- Split the dataset for training and testing purposes.

Week 3: Recommendation Algorithm Development

- Implement a **Collaborative Filtering** algorithm using user-item ratings or **Content-Based Filtering** based on book attributes like genre, author, and keywords.
- Train the machine learning model on the dataset and fine-tune it for better accuracy.
- Create a function to generate recommendations for users based on their input or past reading history.

Week 4: Frontend Development (HTML)

- Design a simple and clean **HTML** interface for users to input their preferences or reading history.

- Display book recommendations dynamically based on the backend model's suggestions.
- Implement a user-friendly search interface for users to query books and get recommendations.

Week 5: Integration & Testing

- Integrate the Python recommendation system with the HTML frontend.
- Conduct testing to ensure the recommendations are accurate and relevant.
- Gather user feedback to improve the system's performance and recommendation accuracy.

Week 6: Optimization & Final Testing

- Optimize the recommendation algorithm for better performance.
- Test the system with a larger dataset and handle edge cases (e.g., cold-start problem for new users).
- Conduct performance testing for system speed and scalability.
- Deploy the system for final testing and presentation.

V. Requirements for Proposed Work

5.1 Software Requirements:

- **Operating System:** Windows 10/11, Linux
- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Python, Jupyter Notebook
- **Libraries:** pandas, numpy, scikit-learn, Flask (for backend API), HTML/CSS for frontend
- **Database:** SQLite or any lightweight database to store user data and book information

5.2 Hardware Requirements:

- **RAM:** 8GB (expandable)
- **Storage:** 256GB SSD

VI. Bibliography/References

- **Python Documentation:** <https://python.org>
- **Scikit-learn Documentation:** <https://scikit-learn.org/>

- **Collaborative Filtering Algorithm:** <https://towardsdatascience.com/a-complete-guide-to-building-a-recommendation-system-from-scratch-10895ab3cf79>
- **Book Recommendation Datasets:** <https://grouplens.org/datasets/book-crossing/>

VII. Conclusion

The **Book Recommendation System** project offers a solution to one of the major challenges faced by book readers and book sellers: efficiently discovering and recommending books based on user preferences. By leveraging Python and machine learning algorithms, the system personalizes book suggestions, allowing users to find new books aligned with their interests, and improving the user experience for book marketplaces or libraries. The system's integration with a simple HTML interface ensures accessibility for all users, while the Python-powered backend efficiently handles book recommendations.

With continuous improvements and algorithm fine-tuning, this Book Recommendation System can be enhanced further, incorporating more advanced techniques, larger datasets, and additional features such as user reviews or sentiment analysis for better accuracy and recommendation quality.
