

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

Palak Sharma

23FS20MCA00021



**MANIPAL UNIVERSITY
JAIPUR**

Under the guidance of

Dr. Pradeep Kumar

Department of Computer Applications
School of AIML, IoT&IS, CCE, DS and Computer Applications
Faculty of Science, Technology and Architecture
Manipal University Jaipur
Jaipur, Rajasthan
2025

INTRODUCTION :

The Book Recommendation System is an online tool that suggests books to users according to their interests, reading habits, and preferences. The system uses a Python (Jupyter Notebook) implementation of a machine learning-powered recommendation algorithm that analyzes user data and makes book recommendations. HTML is used in the frontend design to provide an interactive experience that allows users to view and choose books. The goal of this project is to make book discovery easier so that users may identify books that fit their reading preferences without having to sift through lengthy lists.

Important Features:

- Personalized book suggestions according to user tastes.
- An HTML-based dynamic user interface that allows users to communicate with the system.
- A machine learning model for book recommendations, such as content-based filtering or collaborative filtering.
- Python-powered real-time book recommendations and effective data processing.

By expediting book searches and offering recommendations based on personal preferences, this technology improves the user experience.

II. Motivation

Because of the size of the book industry and the sheer volume of books available, readers frequently have difficulty finding novels that pique their interest. Many businesses, including media and e-commerce, employ recommendation systems extensively to assist customers in finding the goods or information that best fit their tastes. This book recommendation system was developed with the goal of automating the book discovery process by using machine learning techniques to provide users with tailored recommendations.

In order to make the book selection process fun and efficient, the objective is to assist users in quickly finding new books based on their reading interests or past reading history. This recommendation system offers a crucial service to improve user engagement and

boost book sales or library checkouts in the context of online retailers, libraries, and reading platforms.

III. Problem Statement

- For Users: o The abundance of selections makes it difficult to identify books that are relevant.
- A tedious book search experience due to a lack of tailored recommendations.

A system that can recommend books based on individual reading interests or histories is required.

- For libraries and book sellers: o Tools that enhance consumer interaction by providing tailored book recommendations are required.

It is necessary to offer users pertinent book recommendations in order to improve client retention and happiness.

IV. Methodology/Planning of Work

1: Initial Development & Project Setup

- Complete the tech stack, which includes HTML, Jupyter Notebook, and Python.
- Configure the environment and tools for HTML (for frontend development) and Python (for data processing).
- Compile the first dataset of books with characteristics such as author, genre, title, and rating.
- Arrange the algorithms for book recommendations (content-based, collaborative filtering, or hybrid).

2: Gathering and Preparing Data

- Compile a book and user rating dataset (from an API or an already-existing dataset such as Goodreads or BookCrossing).
- Handle missing numbers, normalize ratings or other attributes, and clean and preprocess the data.
- Divide the dataset into sections for testing and training.

3: Development of Recommendation Algorithms

Using user-item ratings or content-based filtering based on book qualities like genre, author, and keywords, apply a collaborative filtering algorithm.

- Use the dataset to train the machine learning model, then adjust it for increased accuracy.
- Provide a feature that allows users to receive recommendations based on their input or previous reading choices.

4: HTML Frontend Development

- Create a clear, uncomplicated HTML interface that allows users to enter their reading history or preferences.
- Show book recommendations in real time based on suggestions from the backend model.
- Provide an easy-to-use search interface so that consumers may ask questions about books and receive suggestions.

5: Testing and Integration

- Connect the HTML frontend to the Python recommendation system.

To make sure the suggestions are correct and pertinent, test them.

- Compile user input to enhance the system's functionality and precision of recommendations.

6: Final Testing & Optimization

- To improve performance, optimize the recommendation algorithm.
- Test the system using a bigger dataset and deal with edge cases, like the issue of new users not starting up.

Perform performance tests to check the scalability and speed of the system.

- Set up the system for last-minute testing and demonstration.

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/>