

Capstone 8: Book Recommendation System

Program: Machine Learning Models

Project Type: Content-Based Recommender System

Dataset: [Goodbooks-10K](#) and [Book-Crossing Dataset](#)

Objective

The goal of this project was to build a simple **recommendation system** that suggests books to users based on their preferences. The system primarily uses **content-based filtering**, which relies on item similarity using metadata such as tags and genres.

We also added user interaction through a **Streamlit dashboard** that allows users to select a book and get real-time recommendations.

Project Structure

The project follows a clean folder structure:

```
capstone-recommendation-system/  
|  
├── data/           # Raw and processed datasets  
├── app/            # Streamlit app  
├── reports/        # Markdown + summary reports  
├── notebooks/      # EDA and preprocessing notebooks  
├── requirements.txt # Project dependencies  
└── README.md       # Project documentation
```

Dataset Overview

We used two datasets:

- **Goodbooks-10K:** Contains 10,000 popular books and over 6 million ratings.
- **Book-Crossing:** Contains metadata about books, authors, and user preferences.

From these, we extracted:

- Book titles
- Authors

- Ratings
 - Tags/Genres (features for similarity)
-

Data Preparation

We performed the following preprocessing steps:

- Removed null values and duplicate entries
 - Merged rating data with book metadata
 - Filtered out books with very few ratings to reduce noise
 - Generated a new matrix with book features for similarity comparison
-

Recommendation Logic

We implemented **content-based filtering** using the **cosine similarity** method. Here's how it works:

1. For each book, we combined metadata like title, author, and tags into a single string.
 2. We applied **TF-IDF vectorization** to convert the text into numerical features.
 3. Cosine similarity was computed between all books based on these features.
 4. When a user selects a book, the system finds the top 5–10 most similar books and displays them.
-

Exploratory Data Analysis

We explored and visualized:

- **Rating distribution**
- **Number of ratings per user**
- **Books with more than 500 ratings**

These helped us understand user behavior and identify popular books.

[View Histogram Report - \(reports/histogram_report.md\)](#)

Streamlit Dashboard

We built a lightweight user interface using **Streamlit**. The app:

- Lets users select a book from a dropdown

- Instantly shows a list of recommended books
- Is simple and easy to use

You can run it locally with:

```
streamlit run app/app.py
```

Key Results

- Successfully implemented a book recommendation system with clear, ranked suggestions.
 - Interface allows user-friendly interaction and real-time recommendations.
 - Achieved meaningful recommendations using only content-based filtering.
-