



# Content-Based Book Recommendation System

## Introduction

This project presents a simple yet effective **Book Recommendation System** that leverages **Natural Language Processing (NLP)** techniques to generate personalized book suggestions. The system uses **text embeddings** to represent book titles in a numerical form, enabling similarity-based recommendations using the **FAISS** library.

Additionally, the system allows users to **filter recommendations by genre**, adding an extra layer of personalization.

This system runs in a simple command-line interface (CLI), providing an interactive experience without a graphical user interface.

---

## Project Workflow

Below is the step-by-step process followed in building the recommendation system:

### 1. Data Loading

The book dataset is loaded into a pandas DataFrame for processing.

### 2. Data Cleaning & Preprocessing

Unnecessary columns are removed and missing data is handled to prepare for embedding generation.

### 3. Embedding Generation

A pre-trained Transformer model ( `all-MiniLM-L6-v2` ) is used to create dense vector representations (embeddings) of book titles.

### 4. Vector Indexing with FAISS

The generated embeddings are stored in a FAISS index to enable fast similarity search.

## 5. User Query Embedding

When a user inputs a search query, it's also embedded using the same model.

## 6. Similarity Search

The system retrieves the top N most similar books based on cosine similarity in the vector space.

## 7. Genre Filtering (Optional)

Users can optionally filter results by genre or select top-rated books for more customized recommendations.

## 8. Display Recommendations

The most relevant books are returned as personalized recommendations.

---

## Objectives

- Build a content-based book recommendation system.
  - Utilize NLP and text embeddings to understand book similarity.
  - Enable genre-based filtering for more personalized suggestions.
  - Offer an interactive experience via command-line interface for easy experimentation.
- 

## Tools & Libraries Used

- **Pandas:** Data manipulation and analysis.
  - **NumPy:** Numerical operations.
  - **SentenceTransformers:** For generating sentence-level embeddings using pre-trained models.
  - **FAISS (Facebook AI Similarity Search):** Efficient similarity search on vector embeddings.
  - **Scikit-learn:** Used for computing cosine similarity.
- 

## Features

- Direct user interaction.
- Three different recommendation methods.
- Powered by AI techniques (embeddings + similarity search)
- Supports smart filtering by genre and predicted rating