Book Recommendation System Based on Cosine Similarity

Your Name

July 15, 2025

Abstract

Contents

1 Introduction

2 Literature Review

3 Data and Feature Engineering

In the words of Yoshua Bengio: Good input features are essential for successful ML. Feature engineering is close to 90

page 6: Features them-selves are not so clear cut, going from raw data to features involves extracting features following a featurization process (Section 1.5.2) on a data pipeline. This process goes hand in hand with data cleaning and enhancement.

4 Data and Feature Engineering

4.1 Dataset Description

The dataset used for this study was sourced from *[insert source, e.g., Kaggle, Goodreads API]*, and contains metadata for approximately **10,000 books**. Each entry includes attributes such as the *book title*, *author name*, *average rating*, *number of ratings*, *user-generated tags*, and a *textual description*.

These diverse fields provide both structured and unstructured information, allowing the construction of a feature-rich content-based recommendation engine. Our system leverages these features to compare and recommend books based on their similarity.

4.2 Preprocessing and Cleaning

Before constructing features, the dataset underwent several preprocessing steps to ensure consistency and quality:

- Removal of duplicate or incomplete entries (e.g., books missing descriptions or titles).
- Standardization of textual data by converting all characters to lowercase.
- Elimination of punctuation, digits, and non-ASCII characters.
- Removal of common English stop words (e.g., “and”, “the”, “is”).
- Optional lemmatization or stemming to reduce word variants to their root forms.

Numerical fields such as average rating and number of reviews were normalized to ensure comparability across different scales if used in later analysis.

4.3 Feature Selection and Construction

The goal of this step was to define a meaningful and discriminative representation of each book. We selected the following features:

- **Title**: The official book title, which can carry thematic cues.
- **Tags**: User-generated tags summarizing book themes (e.g., "fantasy", "science-fiction").
- **Description**: A summary or synopsis of the book content.

These three fields were concatenated into a single text document per book, forming a unified textual representation. This aggregated text served as the input for vector embedding in the subsequent step.

4.4 Feature Representation Overview

The resulting text document for each book was vectorized using the **TF-IDF (Term Frequency-Inverse Document Frequency)** method. This representation captures the relative importance of terms in a given document compared to the entire corpus, helping highlight distinctive keywords.

The final result is a sparse vector for each book, which can be used to compute cosine similarity — a measure of textual closeness — between any pair of books. This process is detailed in Section **??**.

5 Vector Embedding

6 Similarity Scoring (Cosine Similarity Matrix)

7 Visualization and Heatmap

8 Clustering

9 Discussion

10 Conclusion and Future Work

References