

DATA ANALYTICS BOOTCAMP

Book Recommendation System

By Luis Pablo Aiello
Data Analyst Student



Project Overview

This project builds an end-to-end pipeline

Scraping → Cleaning → Feature Engineering → Clustering → Content-Based Recommendation → Streamlit Web App



01

Scraped data from Goodreads and Open Library to gather information.



02

Cleaned and merge data to create a unified dataset.



03

We **engineered features** from genres and text for the recommendation system.



04

Apply **unsupervised ML** concepts



05

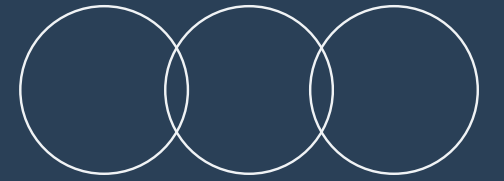
Build **content-based** recommender



06

Deploy a **Streamlit** web app

Data Sources



Goodreads

A popular platform for discovering and reviewing books globally.

- Loop over the Best Books Ever list pages.
- Extract:
 - Book title & URL
 - Author name & URL
 - Average rating
 - Number of ratings
 - Score & votes
 - List rank
- For each book page:
 - Extract Genres (new layout tags).
 - Extract First published year from book details.



Open Library

An expansive online library offering free access to countless book resources.

- Call search.json with the trending query:
 - `trending_score_hourly_sum:[1 TO *] -subject:"content_warning:cover" language:eng -subject:"content_warning:cover" -subject:"content_warning:cover"`
- Extract:
 - Title, author(s)
 - Ratings average & count (if available)
 - trending_score_hourly_sum
 - Work key → build book URL.
- For each work URL:
 - Parse the Subjects block as genres.

Scraping & API Collection



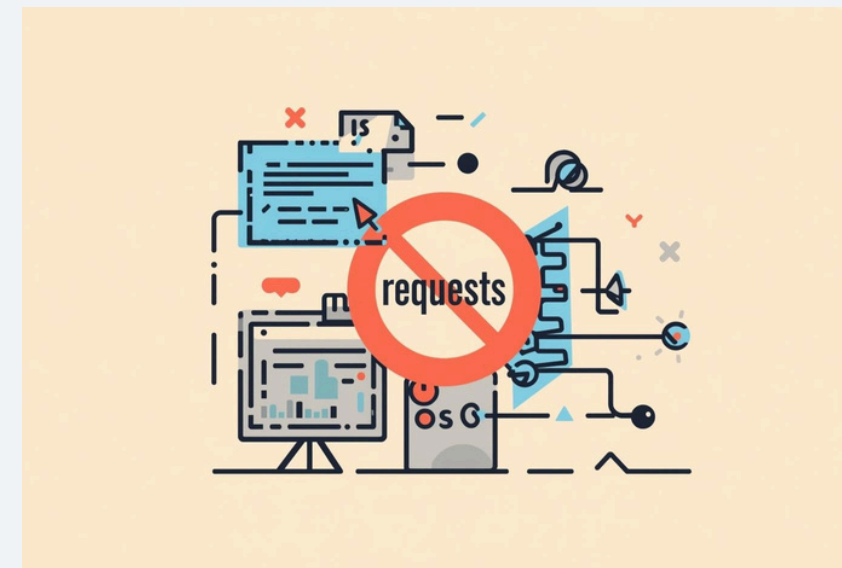
Scraping Tools

Using requests and BeautifulSoup for efficient data extraction.



Open Library API

Collecting data on trending books through an accessible API.



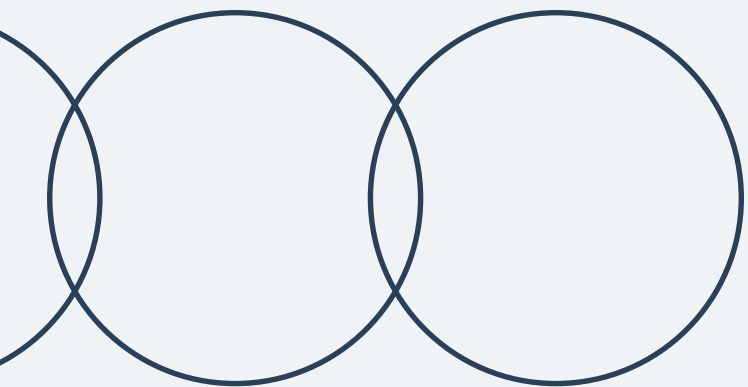
Goodreads Pages

Extracting essential book details from list and book pages.



Respectful Practices

Implementing pauses and User-Agent for ethical scraping.



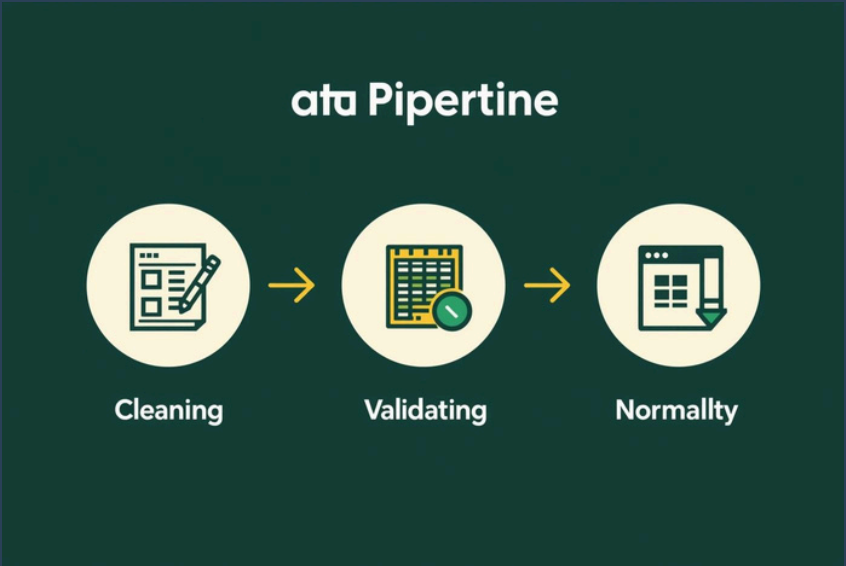
Data Cleaning Steps

“Clean inputs are crucial for good recommendations.”



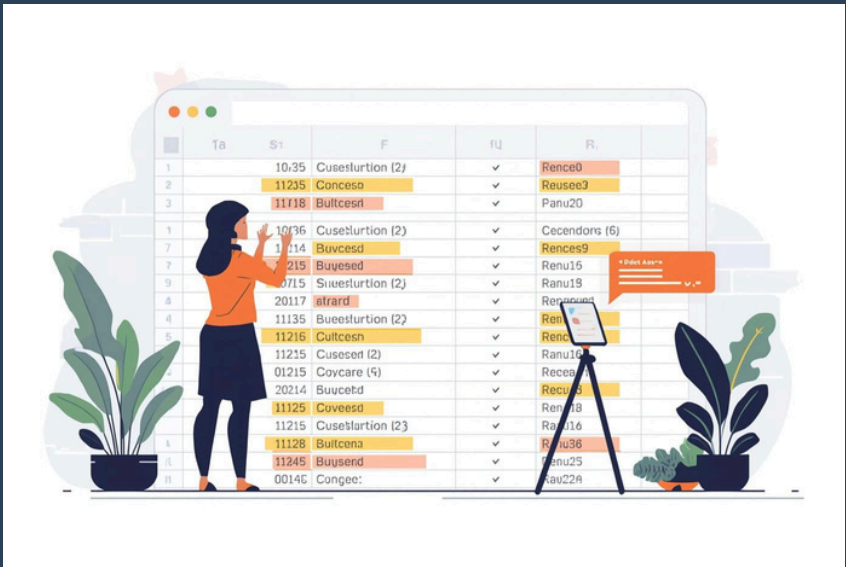
Normalize Titles

Clean titles and authors for consistent formatting and accuracy.



Drop Missing Rows

Remove entries lacking essential cleaned title or author data.



Normalize Ratings

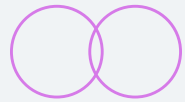
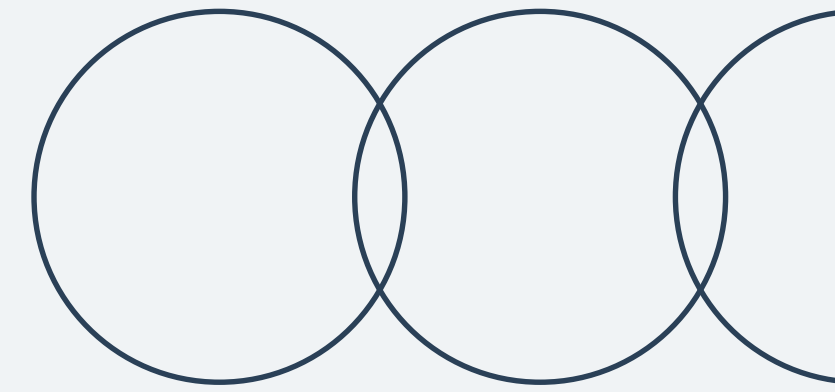
Convert all ratings to a consistent numeric scale for analysis.



Clean Ratings Field

Ensure the ratings field is free of inconsistencies and errors.

Genre Normalization Process



01

Lowercasing ensures consistency across genre entries for better matching.



02

Replacing delimiters with separators simplifies the genre categorization process.



03

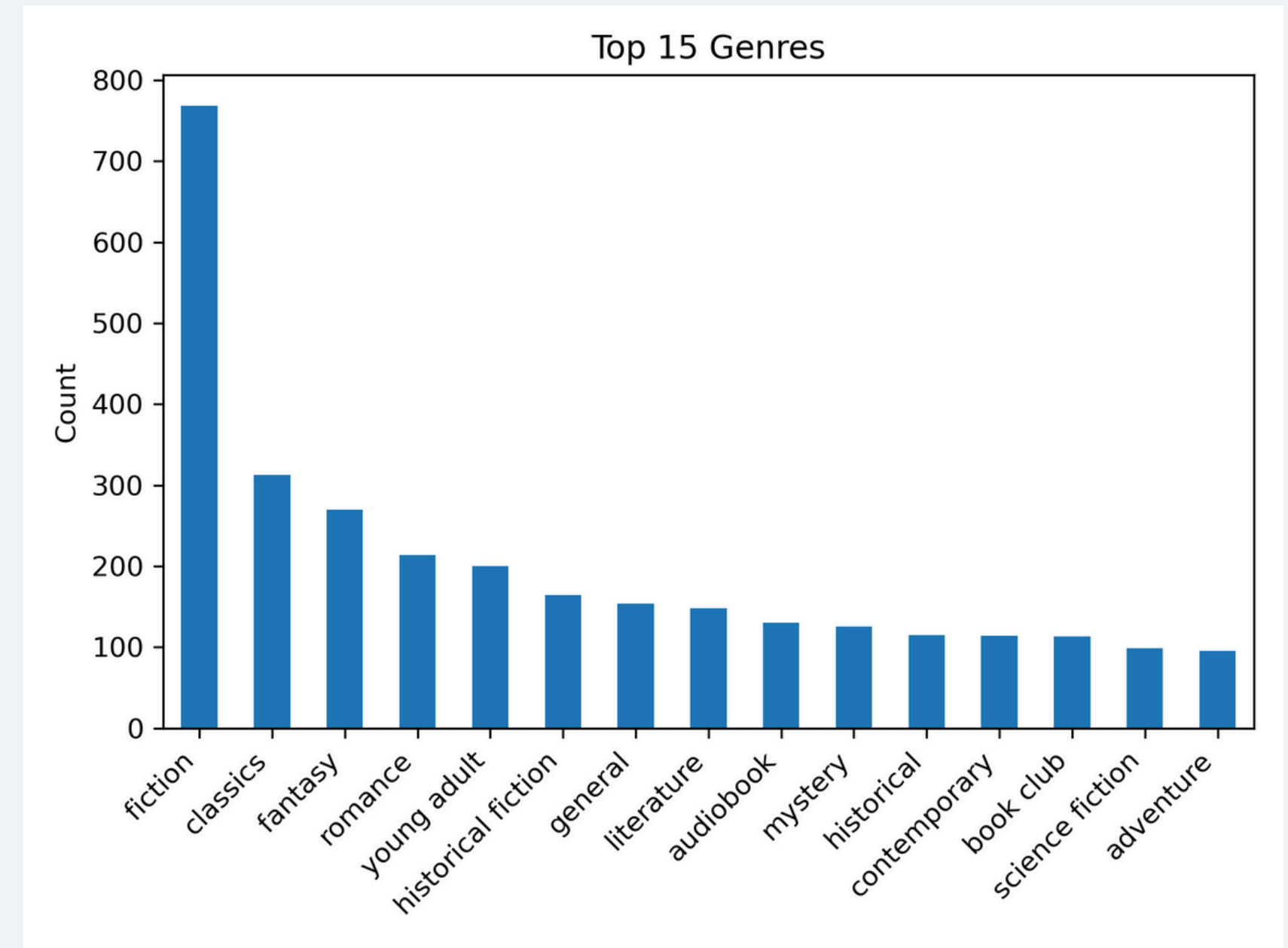
Removing duplicates per book enhances the accuracy of recommendations.



04

Output:

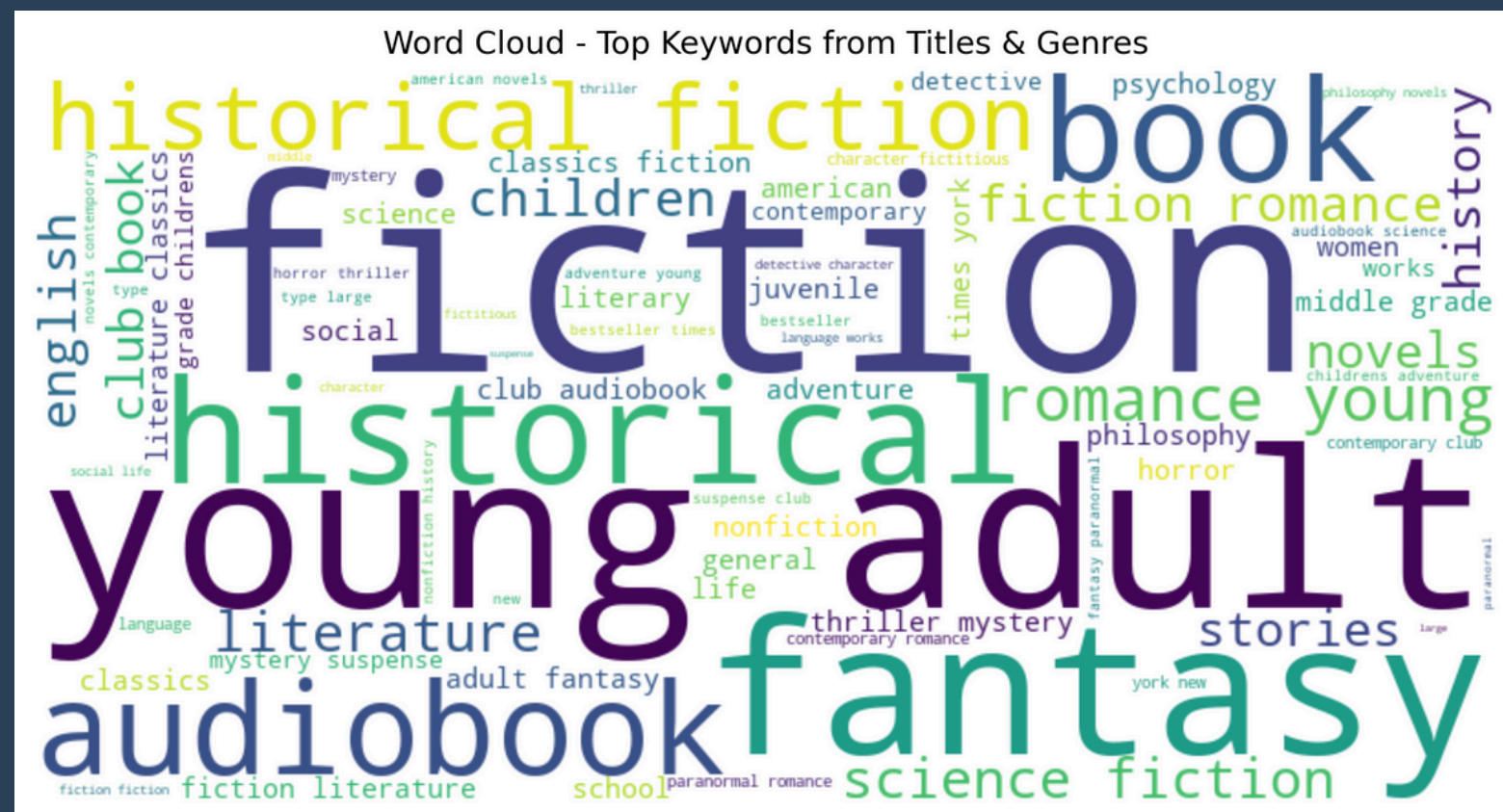
- `genres_list` (clean list)
- `genres_clean` (readable string)



Feature Engineering

Genre Lists & Features

Extracting features is essential for building effective recommendation systems.



TF (Term Frequency):

Words that appear more often in a book's text are more important for that book.

IDF (Inverse Document Frequency):

Words that appear in many books (like “book”, “novel”) are less special.

TF-IDF is high when:

- A word is frequent in one book's text.
- But not so common across all books.

Example:

```
title: "The Hunger Games"
```

genres: "young adult, dystopia, fiction"

→ "the hunger games (young adult, dystopia, fiction)"

We store this in a column like `text_for_keywords`.

Why this is useful:

- This gives us one compact “summary text” per book.
- We can feed this into TF-IDF to find meaningful words.

Clustering Process

Unsupervised ML

This technique effectively groups similar books by identifying patterns within features.

k = 3 -> inertia: 17420.98

k = 4 -> inertia: 15686.79

k = 5 -> inertia: 14677.88

k = 6 -> inertia: 13710.35

k = 7 -> inertia: 12888.65

k = 8 -> inertia: 11987.54

cluster_kmeans

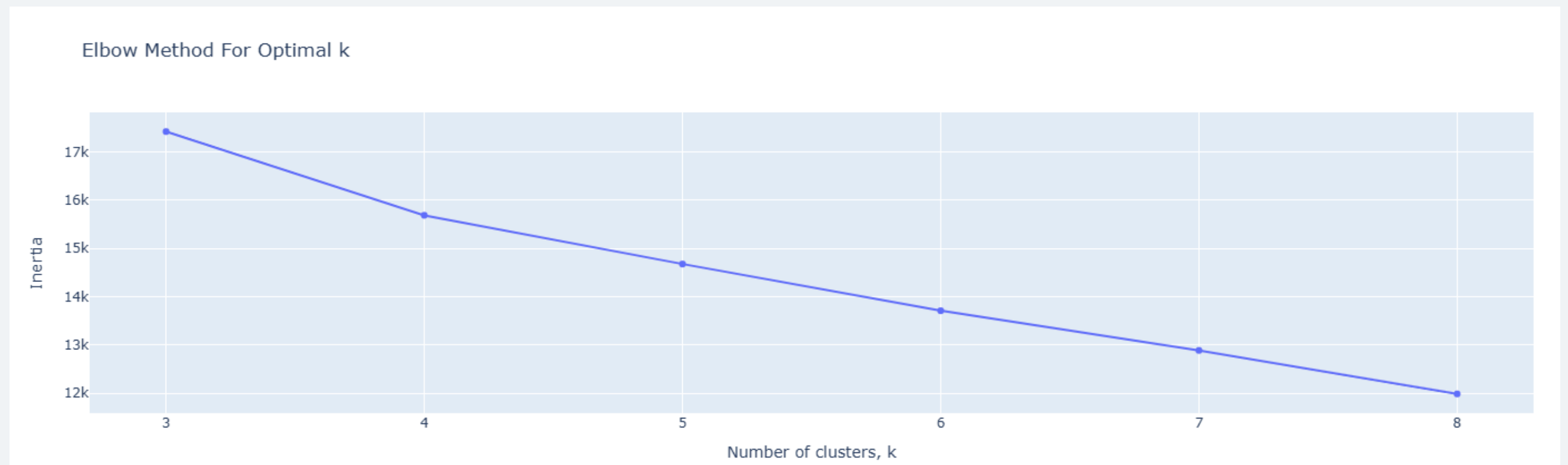
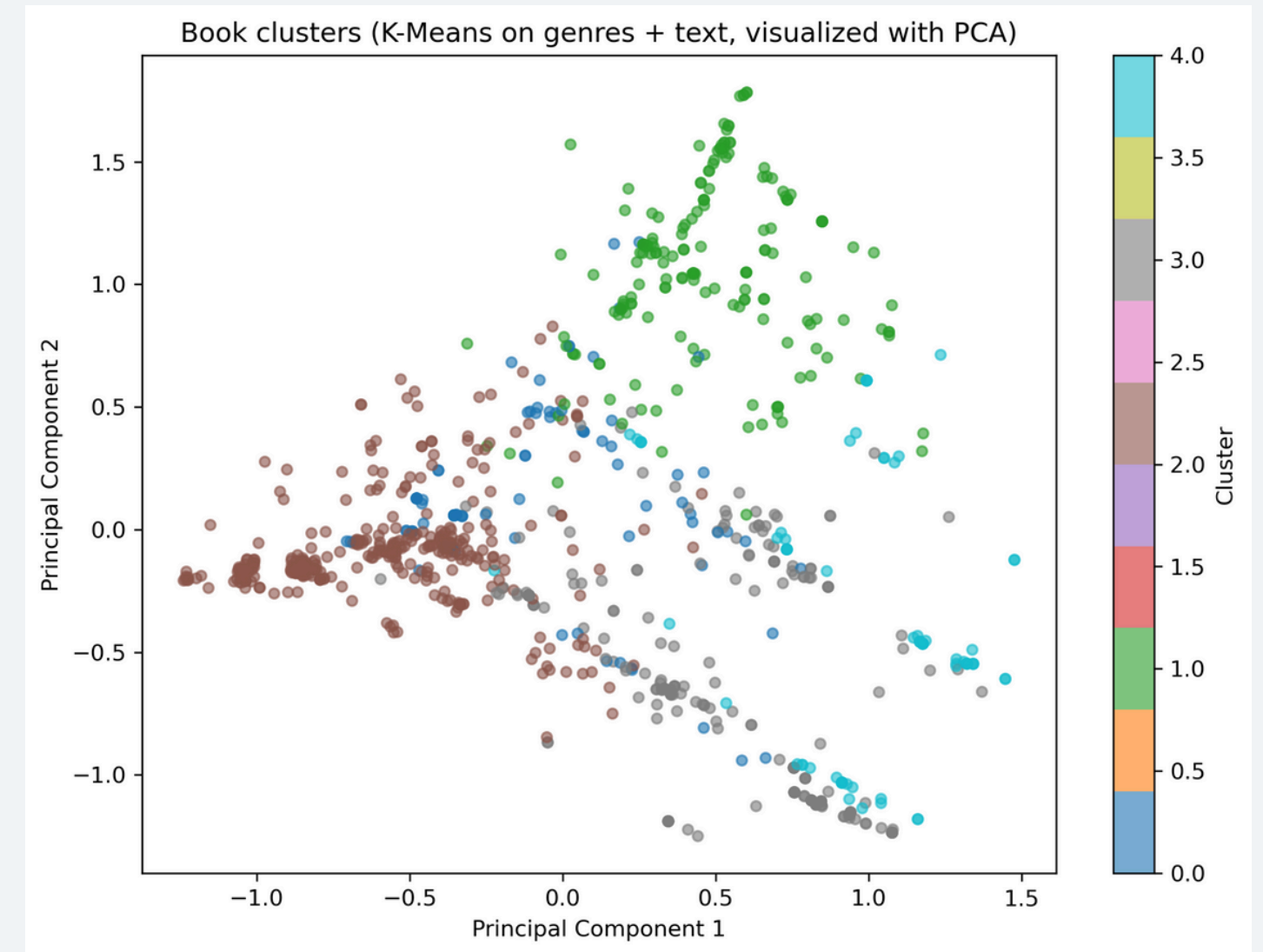
0 98

1 184

2 432

3 221

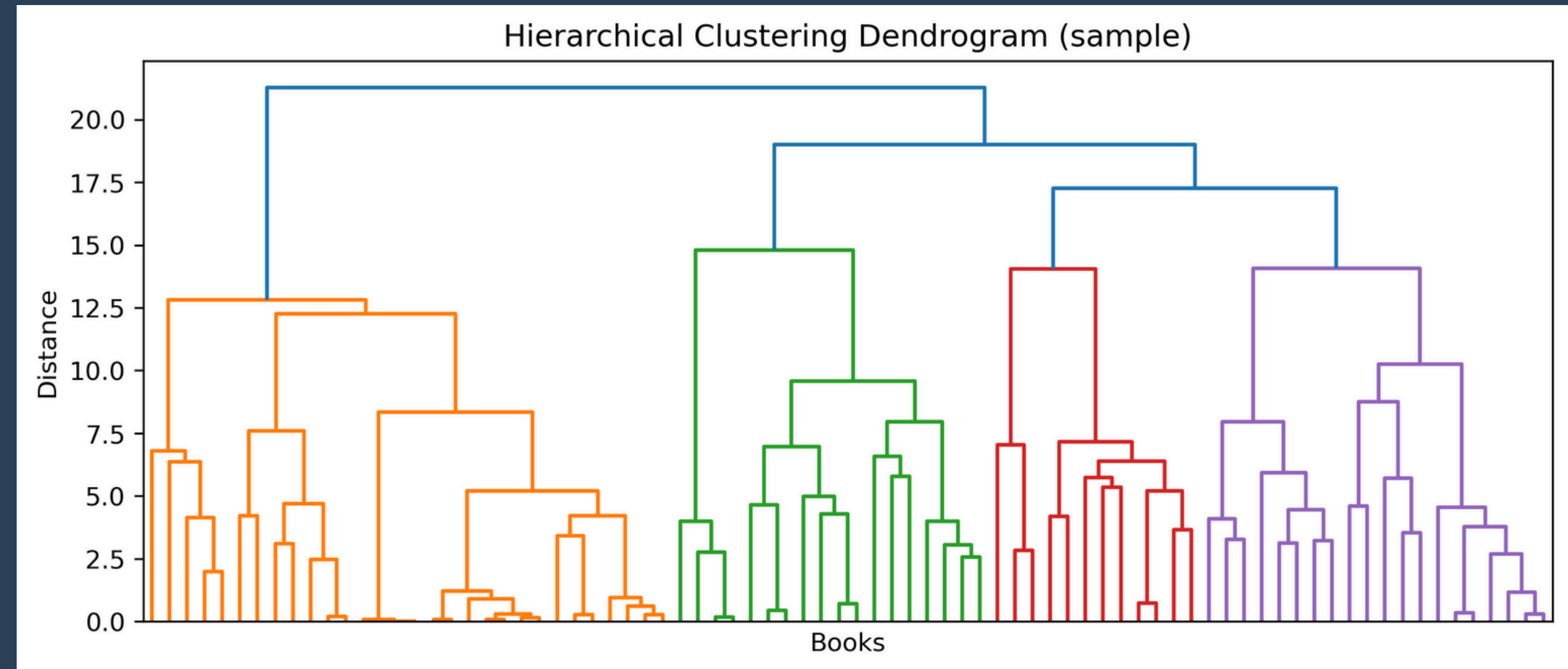
4 75



Content-Based Recommender Logic

01 

Content-based recommendations leverage book features and clustering information.



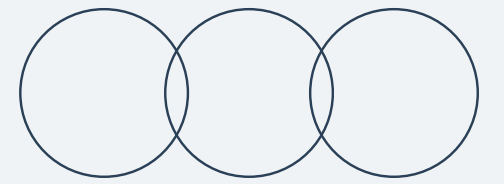
02 

Similarity between books is computed using feature vectors and metrics.

03 

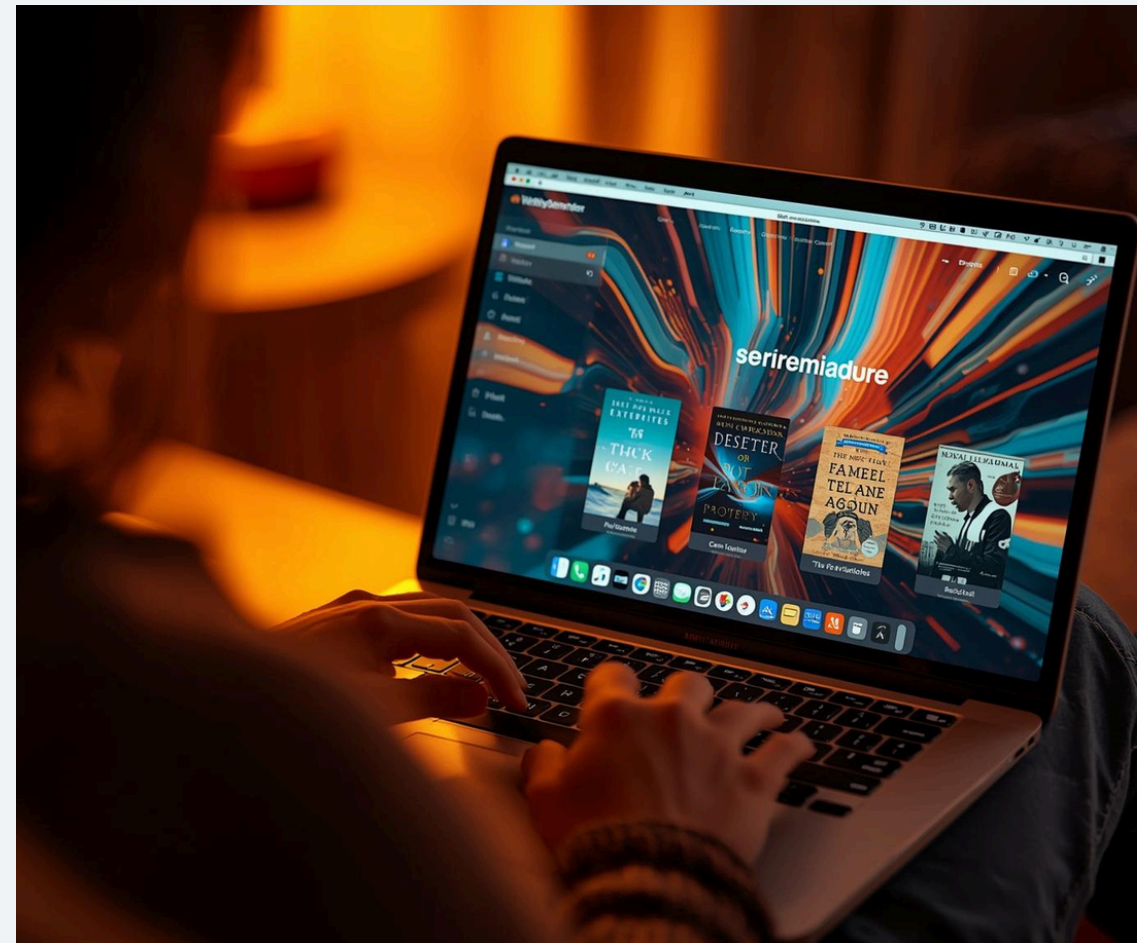
Use cosine similarity:
For a chosen book, find nearest neighbors
Top N most similar = recommendation

Streamlit Web App Features



Interactive Web App

Streamlit provides an **intuitive platform** for creating interactive applications.



User-Friendly Features

Users can search, filter, and find recommendations based on genres.



Deployment Options

Deploy your app on any hosting platform or **run it locally**.



Data Cleaning

Feature Engineering

Transforming raw data into usable formats for recommendations.

Thank You for Your Attention!

Email

hello@reallygreatsite.com

Social Media

[@reallygreatsite](#)

Phone

123-456-7890

