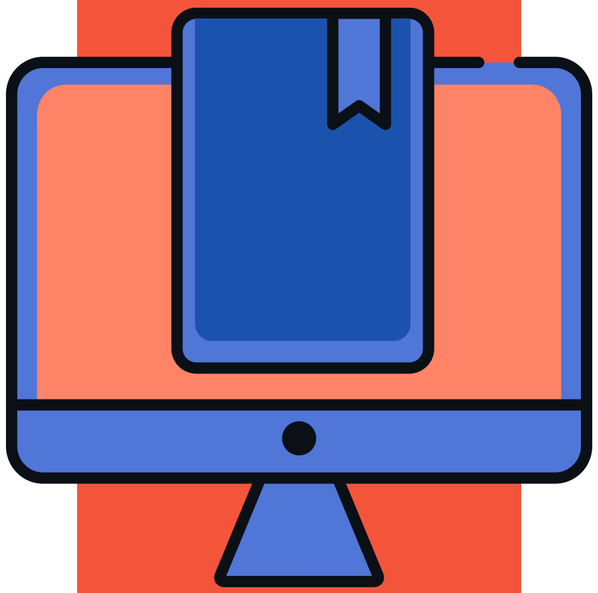


BookRec

Mike Pearce

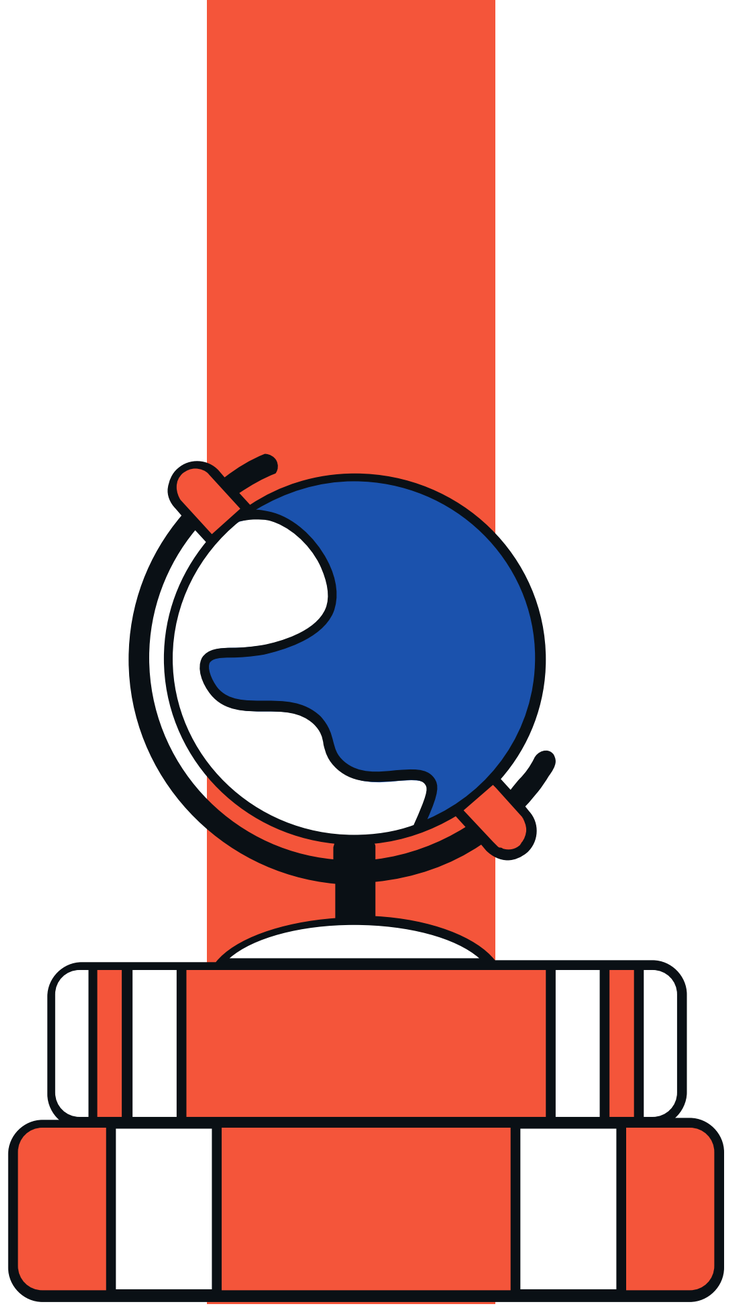
“There is no friend as loyal as a book.”

— Ernest Hemingway

“Write drunk. Edit sober.”

— Also (maybe) Hemingway

Table Of Contents



Project Goals

📖 **Recommend books based on user preferences (age, favorite titles, etc.)**

🌎 **Highlight globally trending books — not just bestsellers**

🧠 **Suggest thematically similar books beyond genre tags**

📊 **Build a system using book metadata, themes, and clustering**

Approach

Use APIs & web scraping to gather:

- Titles, authors, genres, ratings, reviews
- Similar books & writing styles

Data Sources

- Google Books API
- GoodReads.com

1K+	400+	30+	100M+
Books Catalog	Authors	Genres	User Reviews

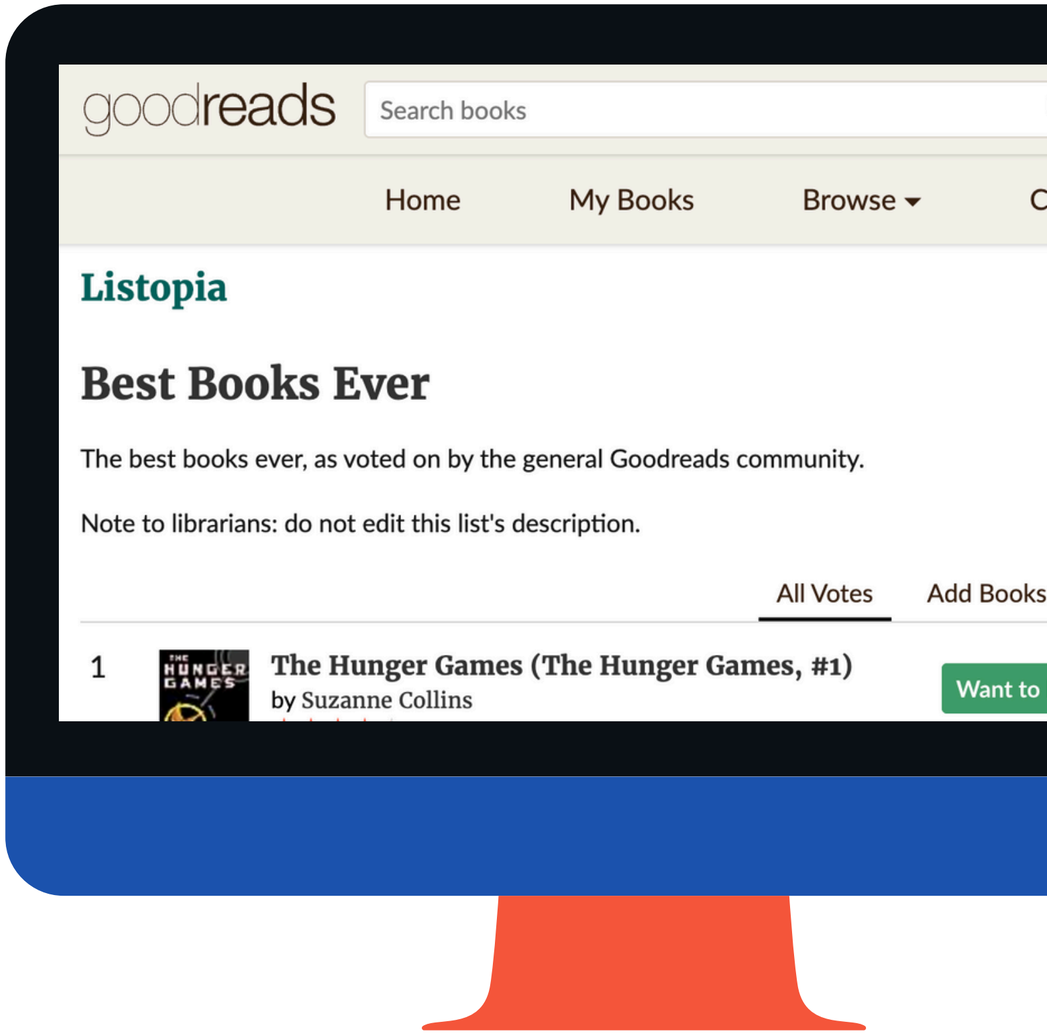
GoodReads.com

Strengths:

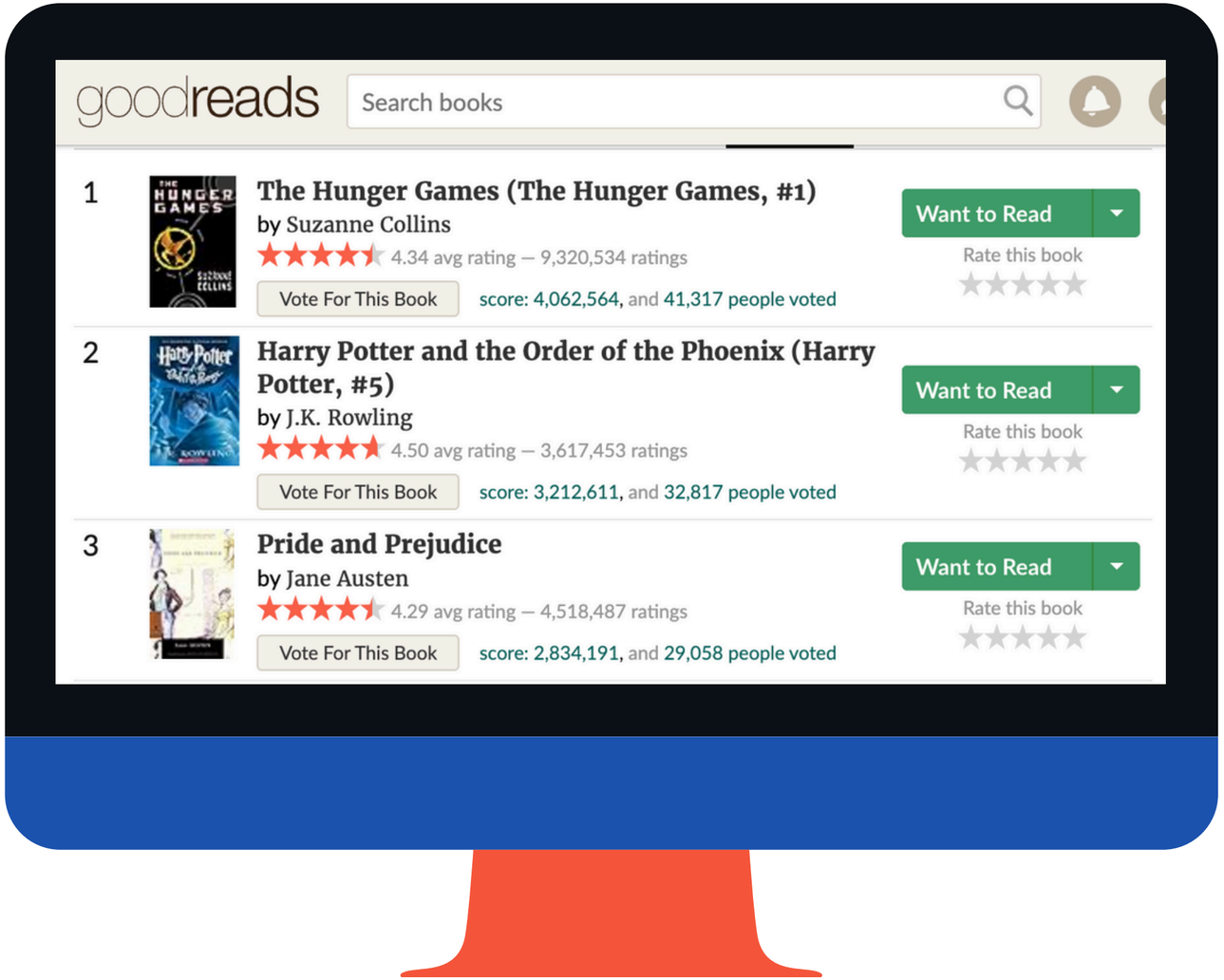
- Lots of quality user reviews
- Large assortment of books
- Good data coverage/quality

Weaknesses:

- Time intensive to pull data via web scraping (Selenium)



GoodReads.com - Best Books Ever List



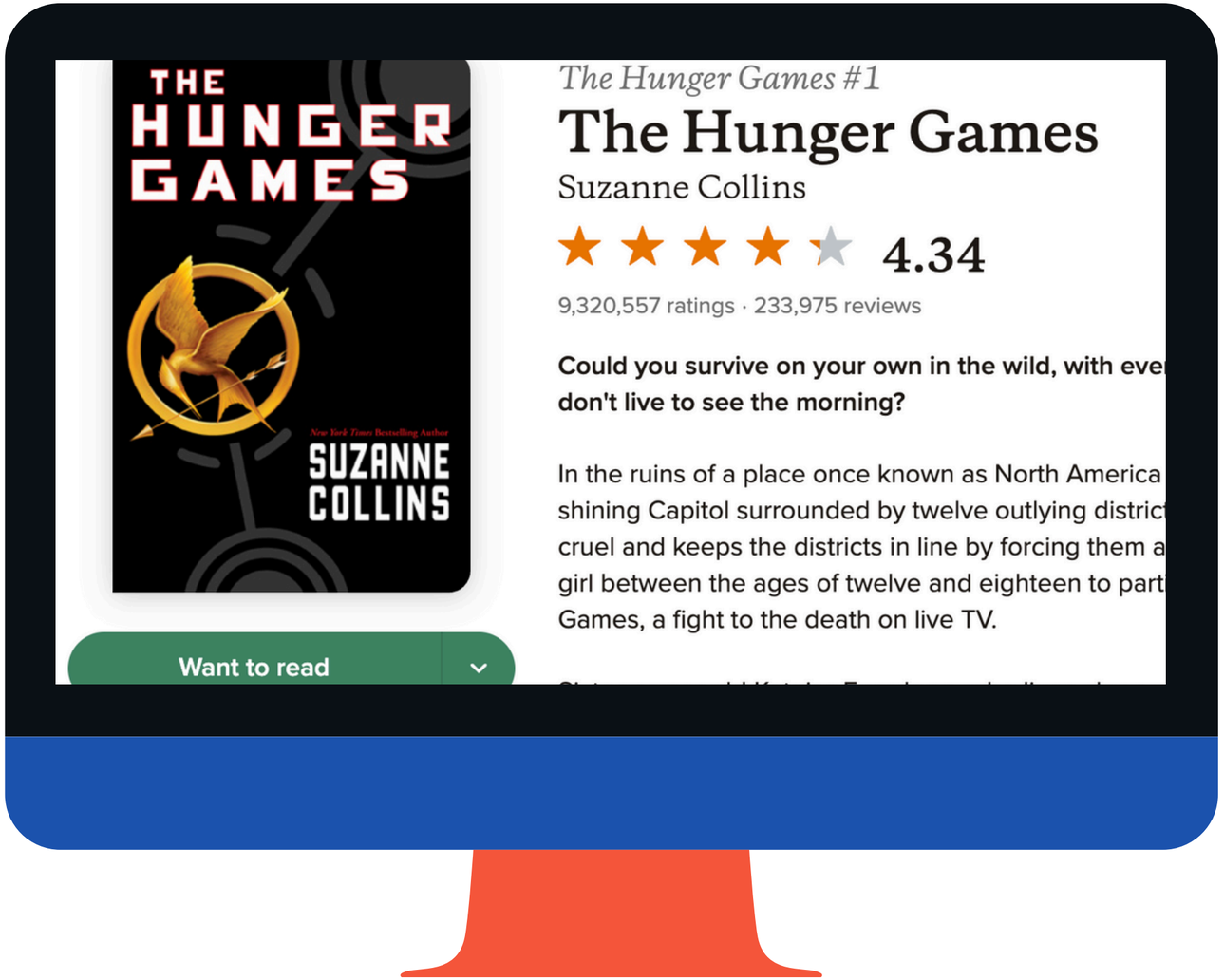
Contains

- Popularity
- Average Rating
- Book Title
- Book Author
- Book Details Page (URL)

Missing:

- Genres***
- Other details (ISBN, Page #, etc)

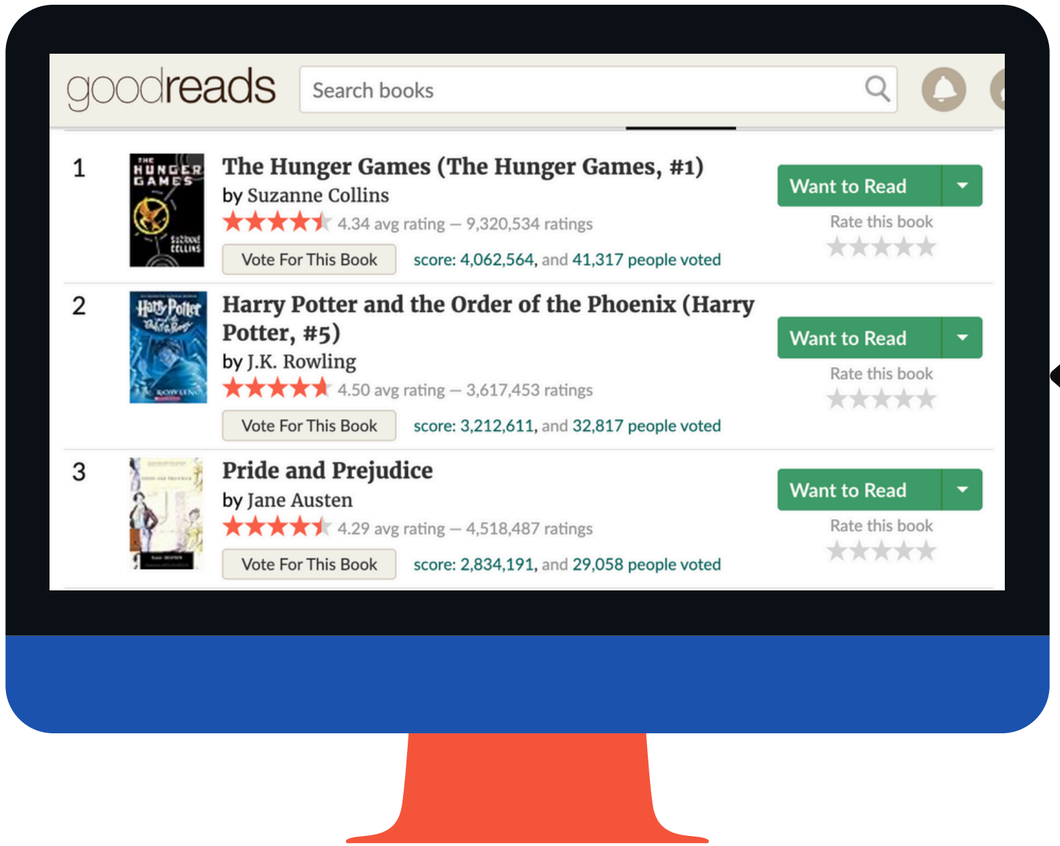
Book Details Page



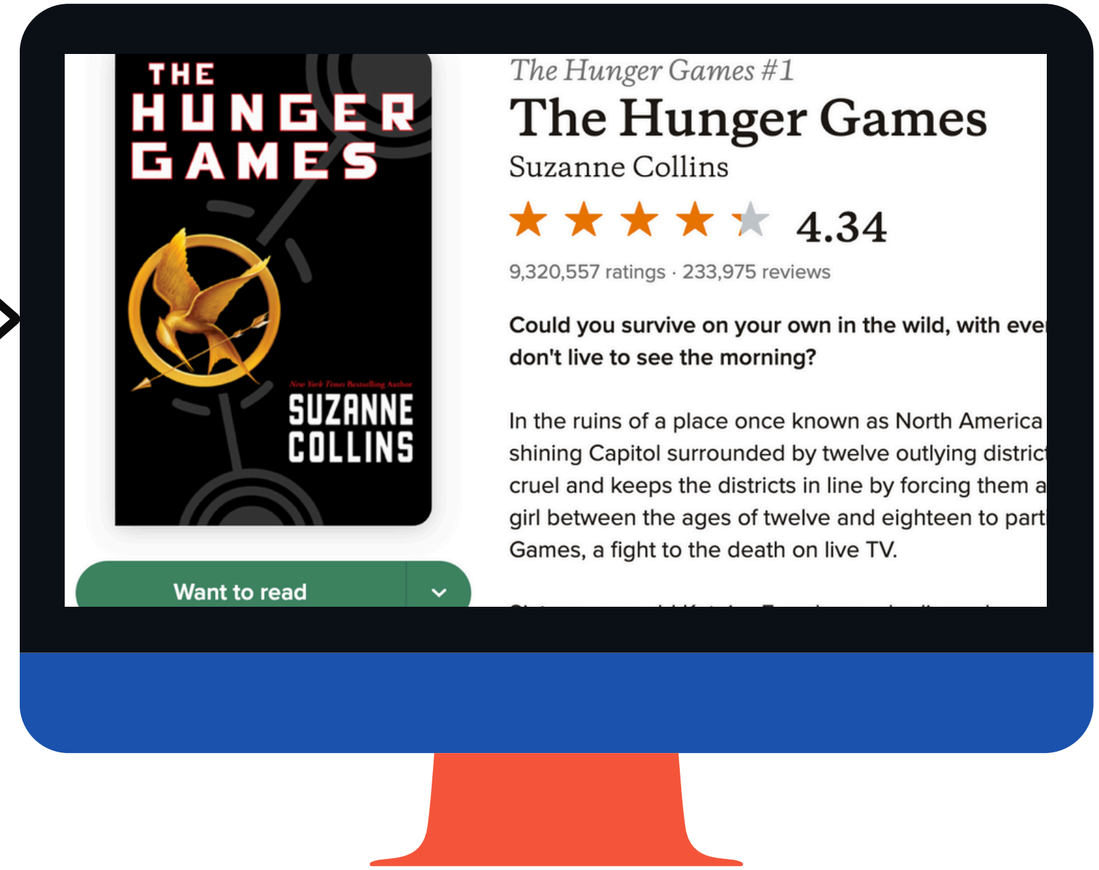
Contains

- Genres
- ISBN
- Page #’s
- Reviews

🙏❤️ Web scraping



500x



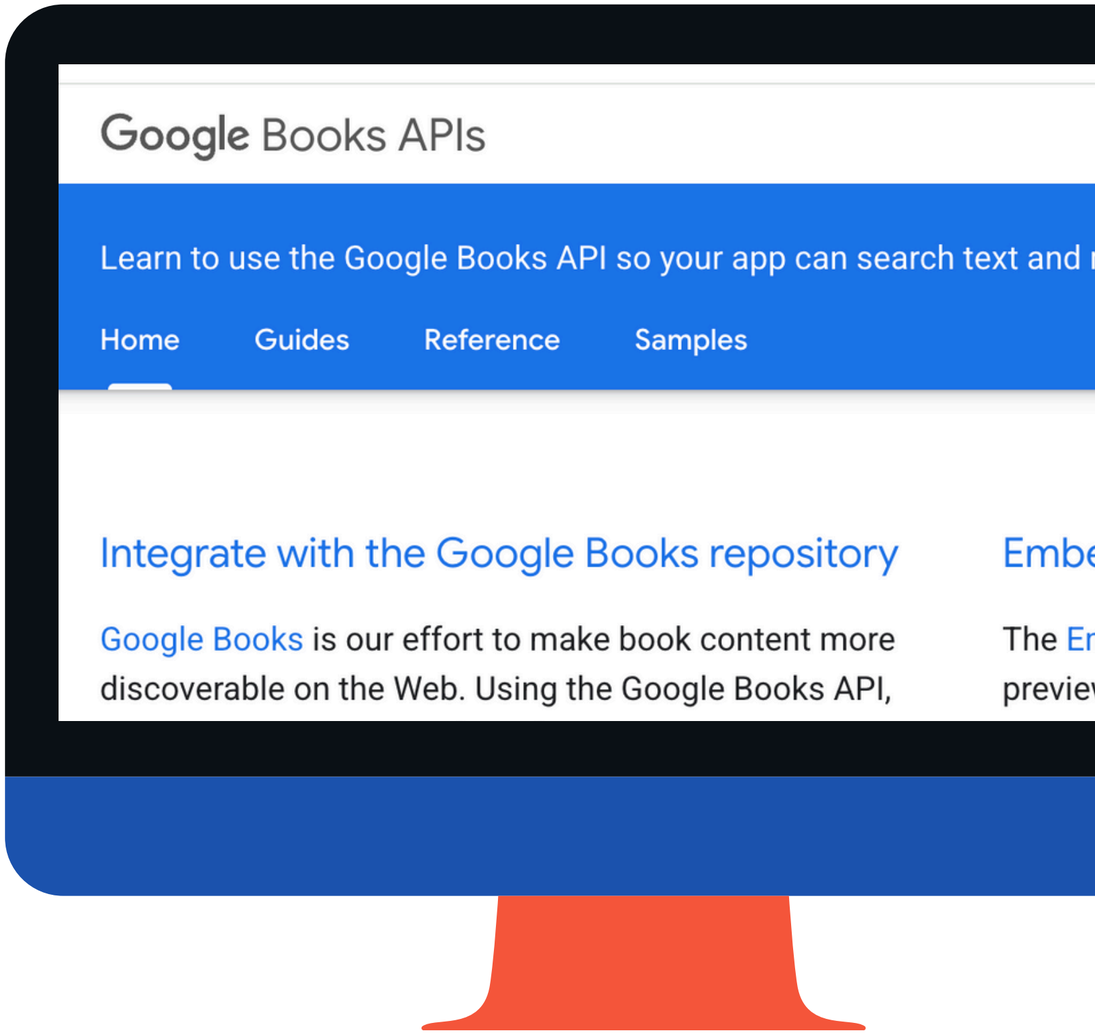
Google Books API

Strengths:

- Very easy and quick to use
- Contains a broad range of books (new, old classic)

Weaknesses:

- Lots of missing data
- Very few ratings and data on popularity/quality



Data Processing 🧹

Cleaned GoodReads Data

- Removed duplicates, extracted series, cleaned genres.

Processed Google API Data

- Normalized ratings, converted types, etc.

Aligned Columns

- Matched structures for seamless merging.

Merged Datasets

- Combined 1,000+ books into a unified dataset

Cleaned GoodReads Data → Processed Google API Data → Aligned Columns → Merged Datasets

Analysis - Ratings & Popularity

🥇 Popularity* vs. ⭐ Avg Rating

- Weak negative correlation (-0.10)
- **Takeaway:** popular ≠ always highly rated (e.g Twilight)

🥇 Popularity* vs. 📈 Num Ratings

- Moderate negative correlation (-0.40)
- **Takeaway:** more ratings = higher popularity*

⭐ Avg Rating vs. 📈 Num Ratings

- Weak positive correlation (0.06)
- **Takeaway:** more ratings ≠ better ratings

** Popularity (low number = high popularity)*

Analysis - Balanced Quality Score

Goal:

- **Balance** popularity and reviews
- Create **one normalized score**

Inputs:

- **Avg_Rating** ⭐ **:** Weight 0.5
- **Popularity** 🥇 **:** Weight 0.25
- **Num_Ratings** 📈**:** Weight 0.25

Note: All scores were normalized and scaled first

Challenge - Missing Quality Data

⚠️ **Challenge:**

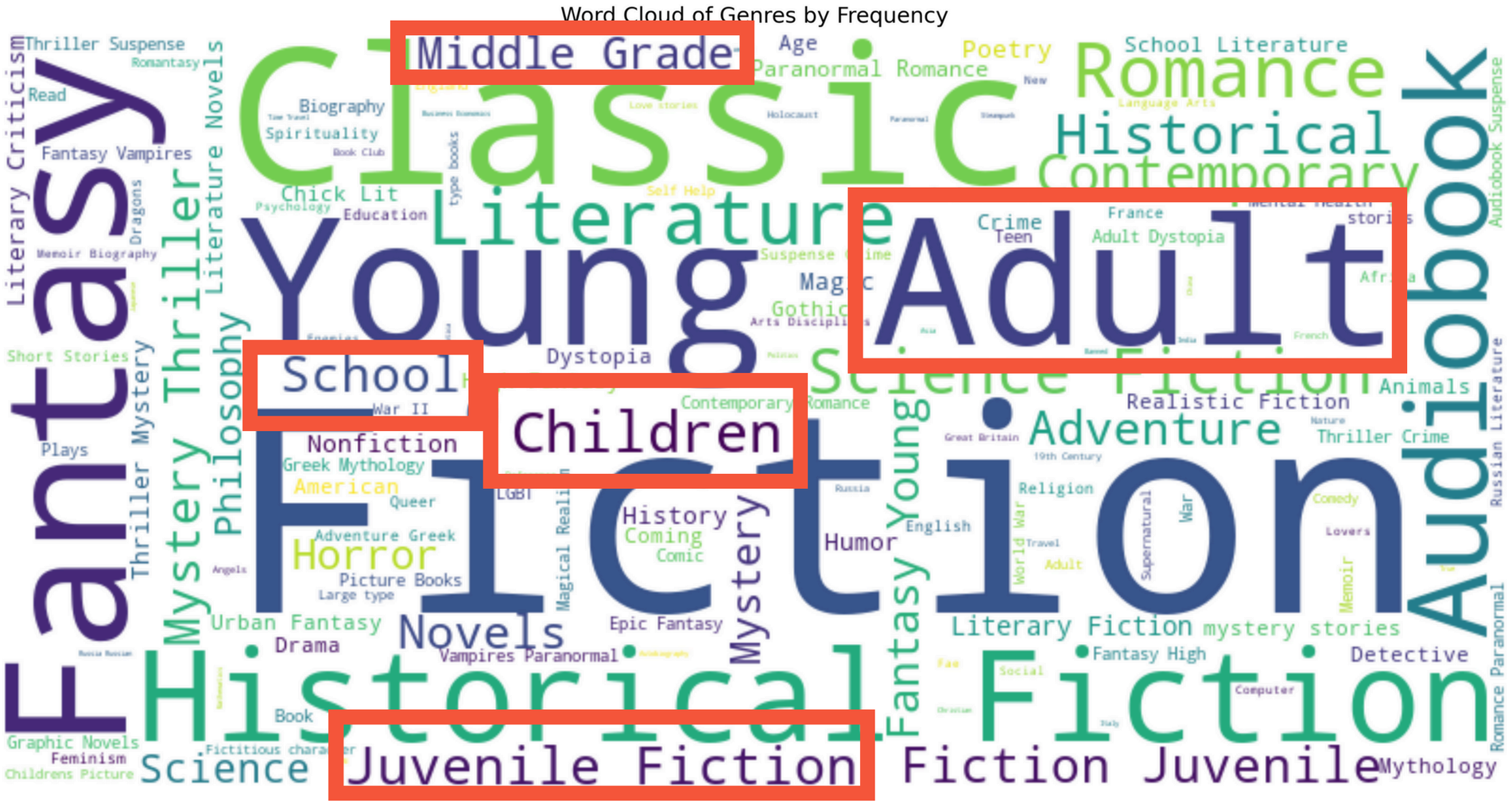
- Many books from the Google API lacked ratings, reviews, or popularity data.

🛠️ **Solution:**

- Filled missing scores using the **average score of other books by the same author.**
- KNN Imputation (Backup)

For remaining gaps, used **K-Nearest Neighbors (K=3)** to estimate scores based on similar books.

Analysis - Book Genres



Keywords by Age Group

Create a new **numerical column** representing age group

👶 **Ages 0–8:**

- "Childrens", "Juvenile Fiction"

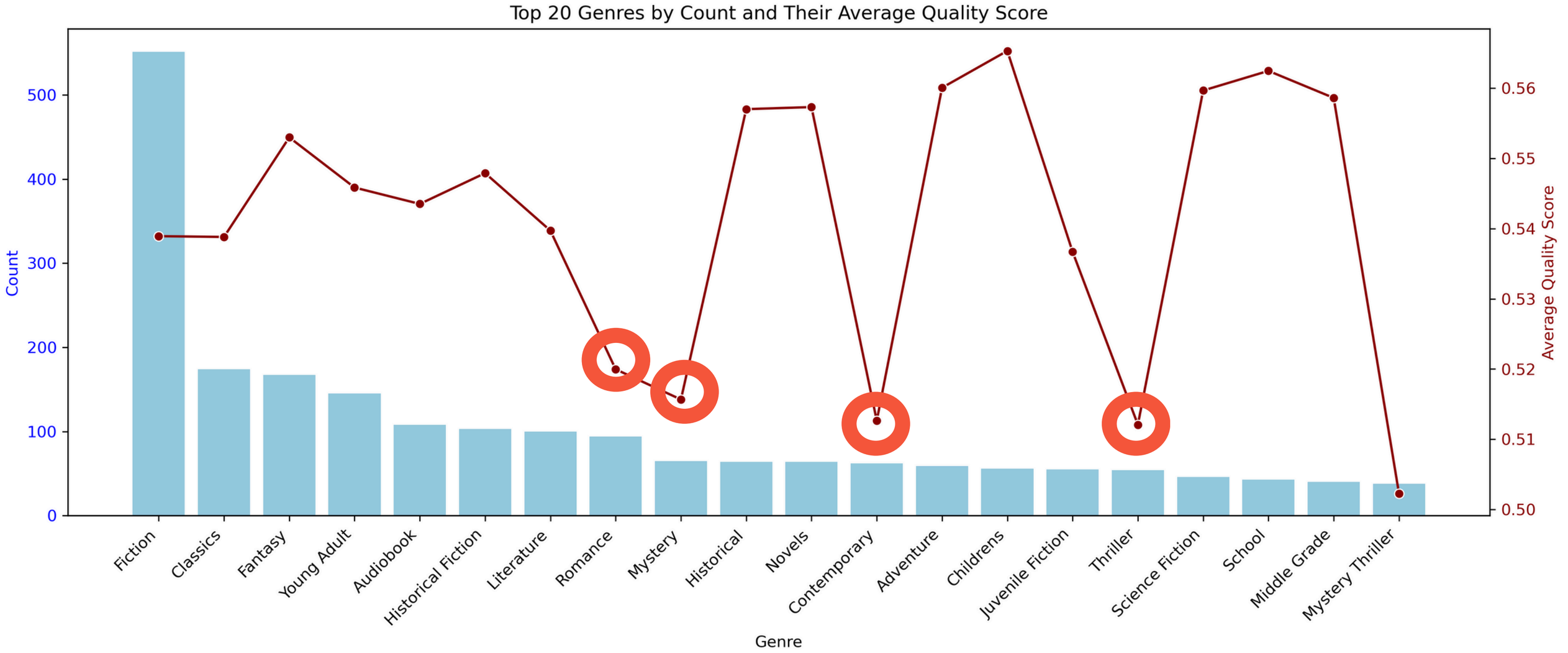
👦 **Ages 8–18:**

- "School", "Middle Grade", "Young Adult"

🔞 **Ages 18–100 (default):**

- "Romance", "Thriller", "Mystery", "Horror" (and anything else not matched above)

Analysis - Book Genres & Quality



Most Controversial Genres

Genres with the most mixed ratings and quality scores — often loved or loathed:

- ❤️ Romance
- 🕵️ Mystery
- 🟦 Contemporary
- 👦 Juvenile Fiction
- 🔪 Thriller
- 🔍 Mystery Thriller
- 👻 Paranormal

"Controversial" = high variance in reader ratings and quality perception

Book Clustering Overview

We used **unsupervised machine learning** to group similar books based on content and reader data.

🔍 **Step 1: Feature Preparation**

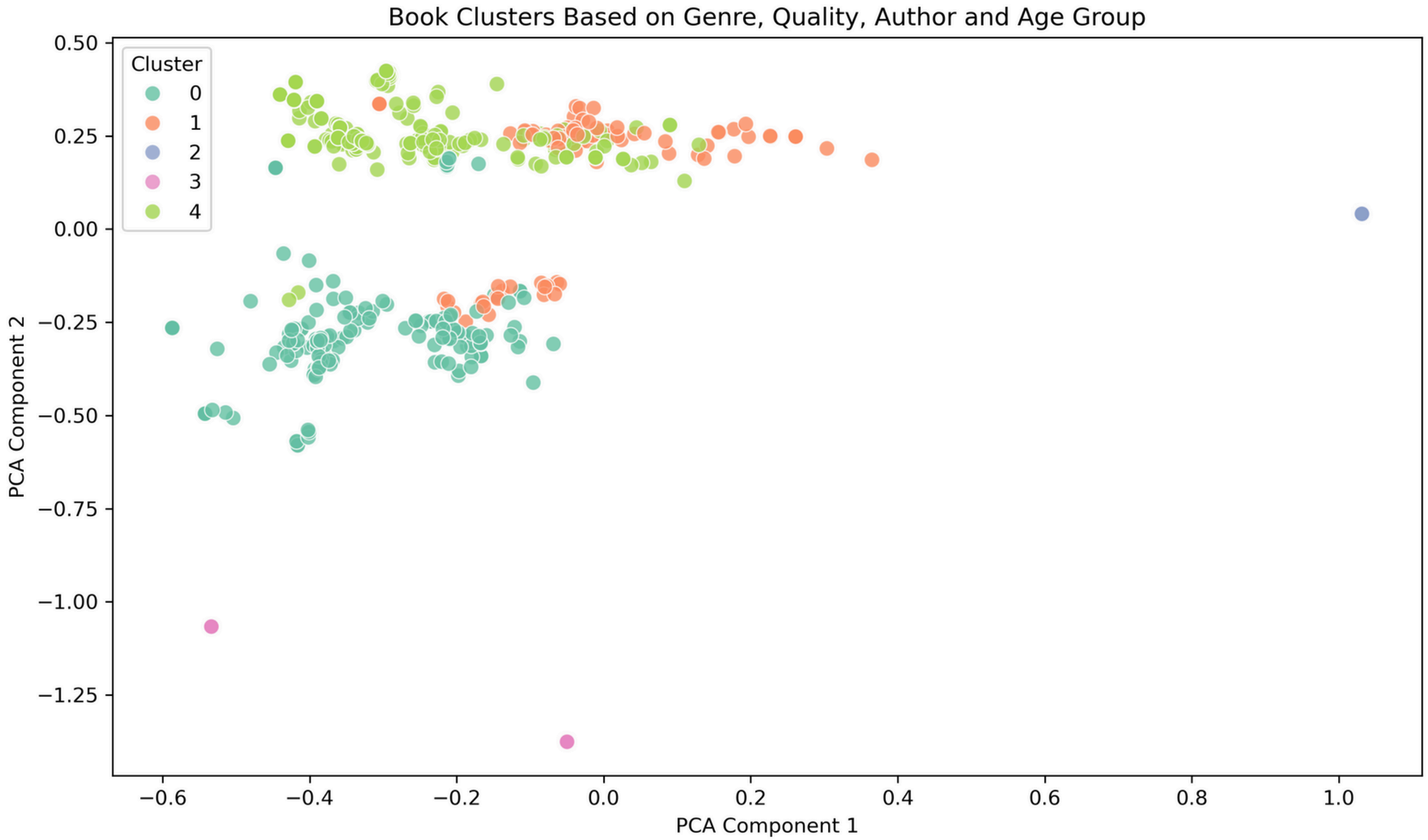
- Converted **genres** and **authors** into text features using **TF-IDF***
- Normalized numeric features: **quality score** and **age group**

🤖 **Step 2: Clustering**

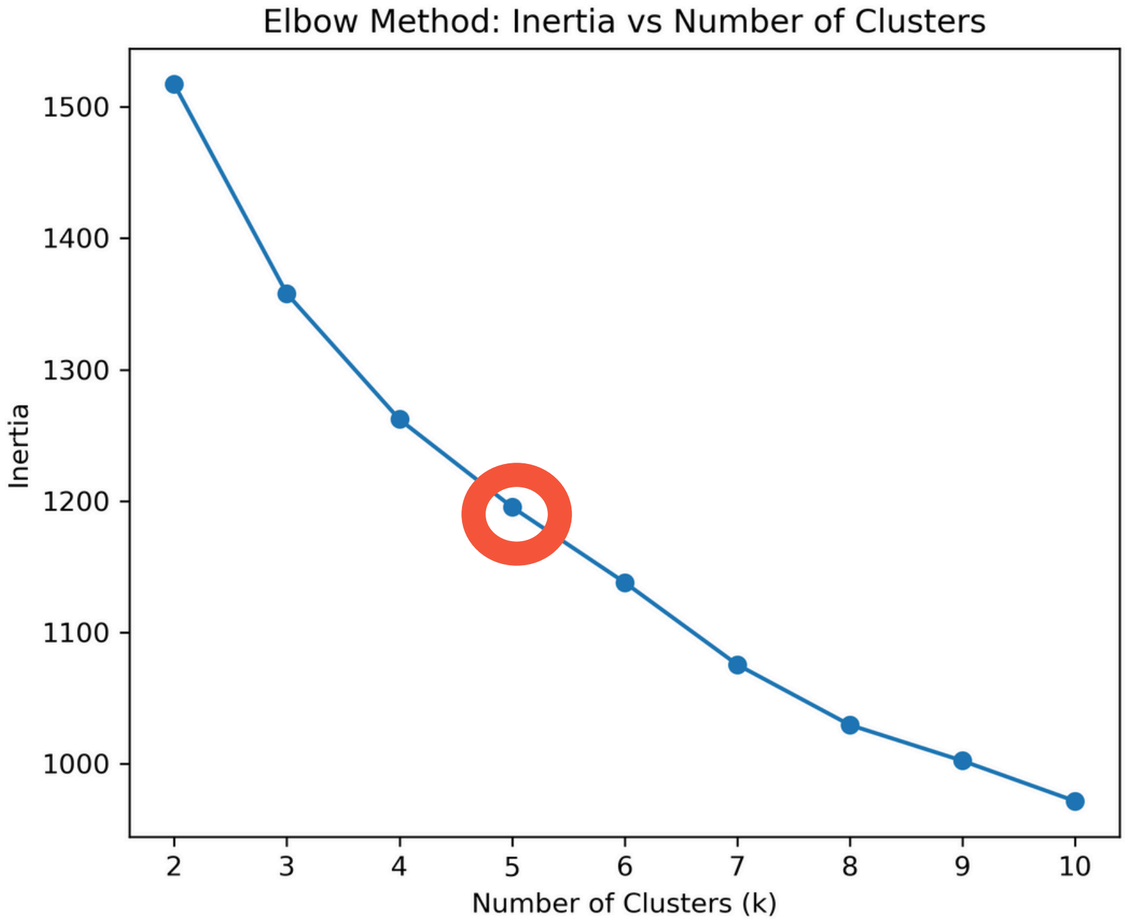
- Combined all features and applied **KMeans** clustering--
- Grouped books into 5 distinct clusters based on content and metadata

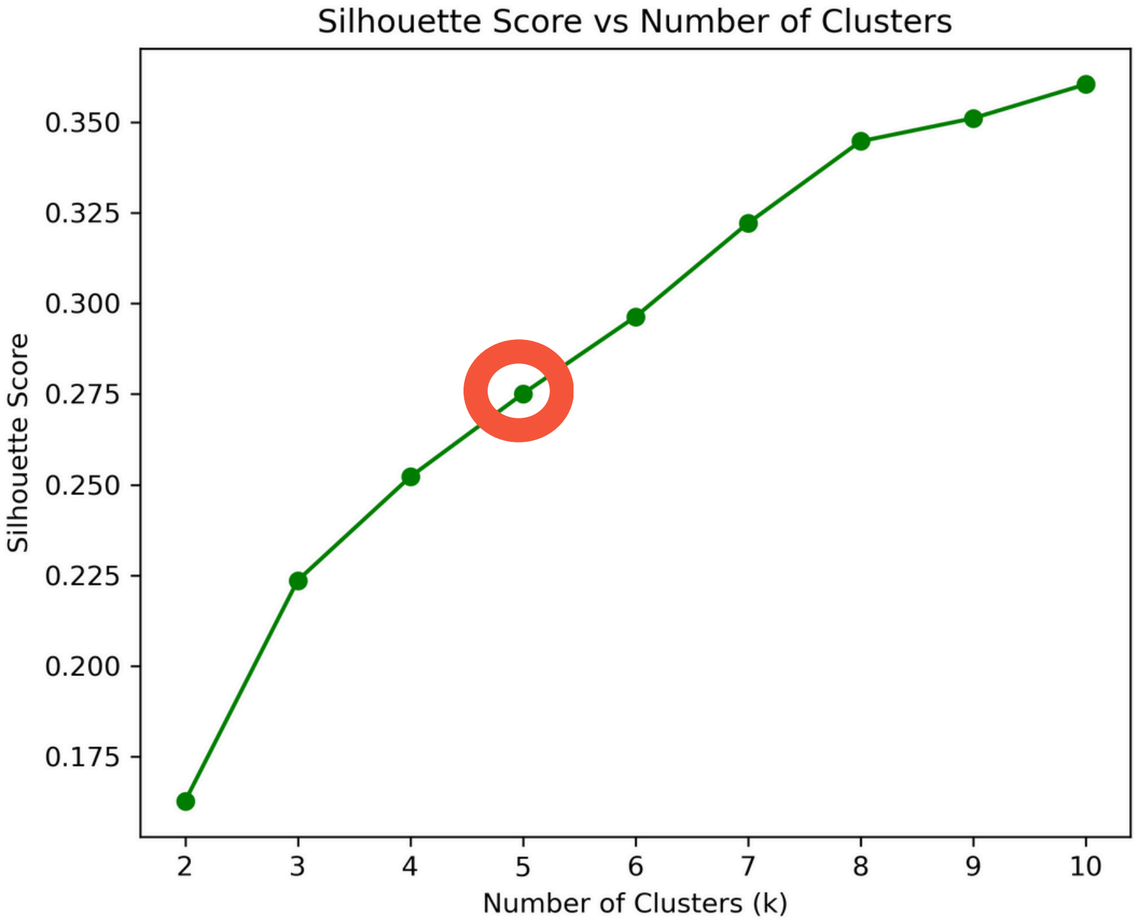
**TF-IDF: Term Frequency–Inverse Document Frequency*

Book Clusters Visualized



Optimizing Clusters





📌 **Takeaway:**

Both the elbow method and silhouette score indicate that **5 clusters** offer the best balance of cohesion and separation for grouping similar books.

Final Recommendations

Primary Data Source

→ Prioritize scraping from **GoodReads.com**

Enhance Recommendation Criteria

→ Incorporate **Genre, Author, Age Group,** and **Popularity**

Deeper Analysis Needed

→ Investigate **low-popularity** books to understand gaps and improve filtering or labeling.

→ More grouping and cleaning of **genre** category to remove duplicates

App Demo