

Northeastern University

DS3000 Final Project: Book Recommendation System

Submitted by

Madeline Jin

Shru Kumar

Grace Pietak

Melina Yang

Date Submitted: 11/29/23

Course Instructor: Eric Gerber

Table of Contents

Abstract-----	3
Introduction-----	3
Method-----	4
Data Description-----	4
Visualizations-----	5
Results-----	6
Discussion-----	6
Works Cited-----	7

DS3000 Final Project: Book Recommendation System

DS3000: Foundations of Data Science

by : Madeline Jin, Shru Kumar, Grace Pietak, Melina Yang

Abstract







This project aims to create a book recommendation system for books solely within a user's pre-made shelf or list. We collected data from the Goodreads website using our own premade reading list by extracting information about each individual book (ie. title, author, genres, rating, etc). Given this information, our program outputs a recommended book or list of books from their list. In order to do this, we calculated pairwise similarity scores and then used k-means clustering. Additionally, we have functions with an option to search purely by genre and keyword if desired. The versatility of our code functionality allows the user to modify their desired book characteristics and number of books to recommend.

Introduction

For many readers, deciding what book to read next can be a difficult task, especially with the overwhelming number of options available both in print and digitally. Goodreads is one of the largest cataloging websites for books that allows users to explore new books, search for books, review books they have read or are interested in reading, etc. One of the most popular features of the website is the ability to create virtual bookshelves or lists to keep track of books to read with ease. Overtime, many of these lists can grow and choosing a book can become very overwhelming. Goodreads already has a recommendation feature, however this system often recommends brand new books for readers to buy and read, ignoring the ones already in user's pre-existing lists. By suggesting books within a user's pre-existing list, we hope to counteract excessive consumerism and wasteful behaviors. By examining different factors of a book a user has input and enjoyed, we are able to output a list of most similar books within their bookshelf. To start, we created a reading list on the Goodreads website and scraped relevant data about each of the individual books to add to our model using Beautiful Soup. Our framework is based on pairwise similarity scores from the book the user input and k-means clustering; however, there are also two more options to search a given list by genre or keyword as well. A sample image of a Goodreads reading list can be found below in Figure 1 [1].

Figure 1 - Goodreads Bookshelf

Want to Read (70) x Search and add books Compare Books Settings Stats Print

#	cover	title	author	rating	rating	my rating	read	added	
1		I Know Why the Caged Bird Sings (Maya Angelou's Autobiography, #1)	Angelou, Maya	4.29	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view
32		The Catcher in the Rye	Salinger, J.D.	3.80	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view
2		Where the Crawdads Sing	Owens, Delia *	4.40	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view
3		All the Light We Cannot See	Doerr, Anthony *	4.32	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view
4		Anna Karenina	Tolstoy, Leo	4.09	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view
5		The Seven Husbands of Evelyn Hugo	Reid, Taylor Jenkins *	4.43	★★★★★	★★★★★ add to shelves	not set	Nov 29, 2022	view

Method

With the information we collected, we decided to create a few different ways for our recommendation system to work. A user is able to search by either genre or keyword and be presented with the books that match sorted by the number of reviews and ratings. If a user doesn't know what they would like to read they can also input a book they enjoyed and recommend similar ones on their reading list. This system is based on cosine similarity and K-means clustering. The cosine similarity calculates similarity scores based on the books genres and descriptions. We found this to be the best way to compare these since some books belonged to multiple genres. From there we extracted and scaled all the numeric features we collected and performed K-means clustering. With this method the rating and popularity of the books are also taken into account.

Data Description

Our project contains data from each individual book on a premade Goodreads reading list. Given the url to the users book list, our code retrieves information about each individual

book storing them all into a dataframe. We used the BeautifulSoup library to web scrape each book's titles, authors, genres, descriptions, book URLs, ratings, number of ratings, and number of reviews. The web scraping functions can be found below along with the first few rows of the created dataframe.

Visualizations

Figure 2 - Overall Book Ratings vs. Number of Ratings

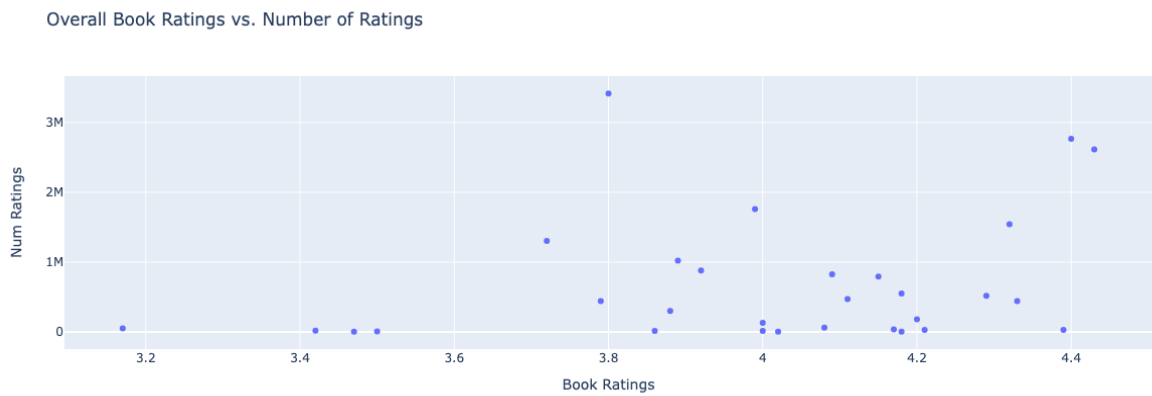
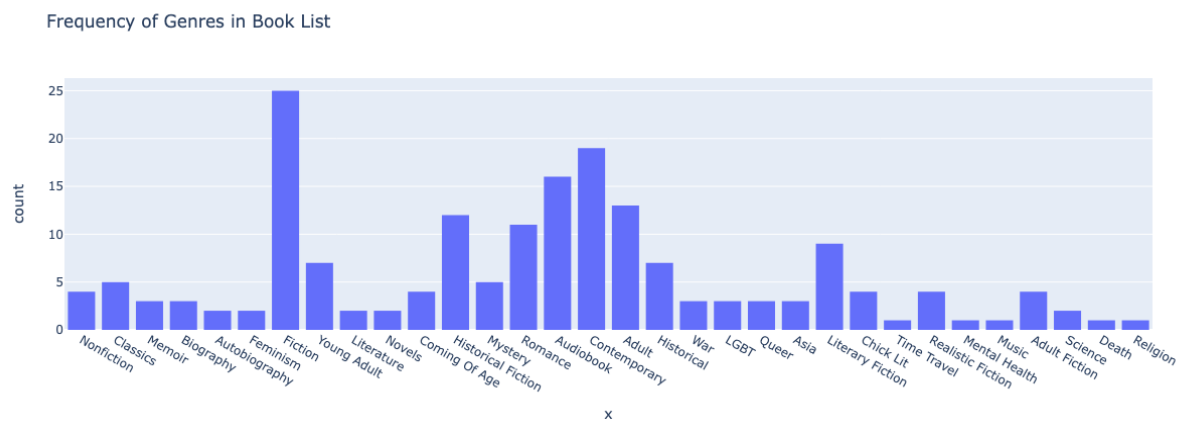
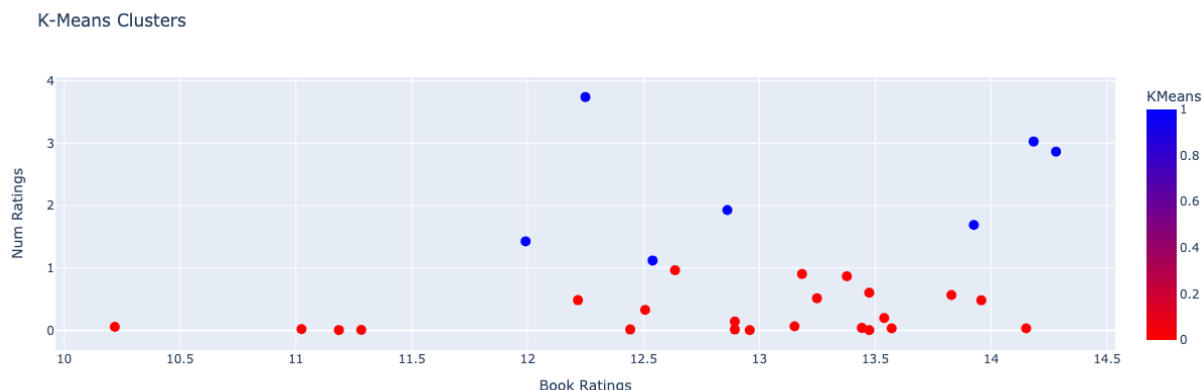


Figure 3 - Frequency of Genres in Book List



Results



Discussion

The resulting code from our project allows us to recommend books to users given a genre, keyword or book. In our project, we explored a couple of different methods of recommending books to the user based on the information that the user wanted to give to us. Some of our functions that output a list of book recommendations are not original concepts—many popular book catalog websites such as Goodreads provide resources so that users are able to filter and provide new book recommendations to users based on data about their likes and dislikes. However, we don't have knowledge of the method behind the recommendation of these books, or even if the process is done using math and data science methods.

Our system primarily operates on a content-based recommendation system, using the details of the books to provide a suggestion. It is highly likely that the book recommendation system used by Goodreads and other book catalog websites is largely done by a collaborative recommendation system, in which users with similar preferences also liked a given book. Since we do not have this database of user information, our model is lacking on this front. Another aspect our model may fail upon is through sentiment analysis of blurbs. At the moment, our model operates on mere similarity of words or keywords within the book description. More language processing methods may allow for more accurate recommendations.

Some additional factors that we could have added to make our recommendation system more accurate and nuanced are as follows:

- Taking into account a user's reading history including genres or authors previously enjoyed.
- Using data on a user's non-reading interests such as music listening history, movies they've enjoyed, places they have traveled to/hope to travel to.
- Having the user input factors that influenced their enjoyment of the book they read.

- Allowing a user to input multiple books they enjoyed, or possibly an entire bookshelf of their favorite reads

Works Cited

[1] Goodreads. "Madeline Jin's *To Read*." goodreads.com, [Online] Available: https://www.goodreads.com/review/list/158906936-madeleine-jin?order=a&ref=nav_my_books&shelf=to-read&sort=position [Accessed: November 29, 2023]