For Project 1, I made a book recommendation system using Python. As an avid reader, I am always looking for new interesting books to read based on my interests. My fellow readers are too and because there are so many book options out there, it is easy to get overwhelmed with choices and overlook something that might become your next favorite.

Many companies like Goodreads, Audible, and Amazon take advantage of this desire by making recommendations for future purchases based on previous ones or in the case of Goodreads, by ratings and word of mouth. I would like to try to replicate a recommendation feature myself based on a selected title as an input. Assuming I just finished the book I inputted and enjoyed it, what should I read next?

I already mentioned that the methods employed by the recommendation systems vary from company to company. Part of this is due to the nature of the company. For example, Amazon is a retailer attempting to sell you your next book while Goodreads is more of a community of readers sharing their experiences and love for their favorite fandoms.

Each company also has different data available to them. Amazon has not only your purchase history but also product information and user ratings for most items in their marketplace. On the other hand, while Goodreads may lack a purchase history they instead have your voluntarily-inputted reading history which goes beyond purchases at a single retailer and can include books checked out from your local library. Combine that with a passionate user base that shares their opinions on the books they read, and they too have a wealth of data at their disposal.

Data

For my project, I found two different datasets on Kaggle that will enable me to try two different approaches: ratings-based and content-based. The first, “Book Recommendation Dataset” (https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset), contains information on a selection of over 242,000 books (e.g., title, author, year published), anonymized user data, and book rating values. This data will enable a rating-based recommendation approach where the system will look for other highly rated books to recommend.

The second dataset, "Goodreads' Best Books Ever" (https://www.kaggle.com/datasets/meetnaren/goodreads-best-books), includes more information about each book than the former dataset. Overall it includes entries on over 48,000 books including title, author, written plot summary, ISBN, rating, number of ratings and reviews, and the genres the book is classified as. The written plot summary data included in this data will enable a content-based recommendation approach where similarities between the contents of the inputted title can be compared with the database to make similar recommendations.

Methods

Rating-based

For the rating-based method, I am going to let the similarity in a book's user rating guide the recommendations. First I created a new TitleAuthor column in the books dataset combining the title and author information into one column for lookup. This is because

authors sometimes write very different books with the same name (e.g., *Joyland* by Stephen King and *Joyland* by Emily Schultz). Next, I joined the books and ratings datasets together into one dataframe based on the shared ISBN column.

Because my computer started repeatedly crashing later in this particular analysis, I came back to this point in the prep work and reduced the size of the dataset by removing ratings from users who had rated less than 10 books. This is NOT a step I would have normally taken, but it did seem to let to get the data to a point where my computer would work with it again.

Ratings with a value of "0" were assumed to be books the user had not rated. These values were replaced with nulls and removed from the active dataset. The dataset was then summarized into average ratings for each title-author combination and a count of the number of total ratings that went into that average.

Next, I created a pivot table showing each user's ratings for each book. Most of this table was null due to not every user rating every book, but the relationships were preserved in it despite its appearance.

To get the recommendations, I then inputted a book title-author combination from one present in the database and retrieved recommendations based on the correlated ratings. To avoid a highly rated book that only had a single rating, I filtered the results to only recommend books with 25 ratings or more. The resultant list of 10 books was then sorted to put more similarly rated books at the top.

Content-based

For the content-based method, I am going to let the similarity in a book's written summary guide the recommendations. First I trimmed down the active dataset by removing unnecessary columns and dropping rows with null values.

Next, I grouped books with the same name together to avoid duplicated results. In cases where multiple column values could have shown for the same book, I took the first one the grouping encountered to keep. Unlike the other database, combining title and author information would have caused more issues due to variations on each entry such as additional authors or illustrators. I decided for this method to only use the title as the input search parameter without the author.

Quotation marks were added around each book's description to facilitate each processing as a separate item through a TfidfVectorizer and fitting into a similarity matrix.

Finally, the book title was set as the index to enable searching, and a function was developed to compare similarity scores between the inputted book and those in the dataset to produce similar recommendations. The resultant recommendations are displayed in a dataframe showing both the recommended book title, author, and genres.

Challenges

The rating-based approach may have suffered from the fact that few books have a rating and fewer still have many ratings (Figure 2). This increases the influence that a single user rating has on the overall score of the book. Ideally, we would want to see at least a few ratings to be able to get a semi-balanced average. The content-based approach did not take user rating into account at this time but would run into the same issue (Figure 6) if it were to incorporate a rating component in the future. There some books have over 500,000 ratings and most others have few or no ratings at all.

A second challenge I faced was how to deal with repeated entries for the same book. In the rating-based system, I was able to combine the title and author information into a unique ID for the book. However, in the content-based system, I could not easily do this due to multiple factors including differences in the information level of detail (i.e., additional authors or illustrators included) or the fact that the same book, such as *The Hunger Games*, was represented in multiple different languages. The TfidfVectorizer I used with stopwords set to "English" most likely removed anything it did not recognize as something it could read thereby eliminating the foreign languages. There are still additional copies or variations of some books within the list however and these are visible even in the search results from my two sample searches.

Another challenge is response bias. Ratings oftentimes might not accurately reflect the general populace's true inclinations. Users who go through the effort of submitting a rating often feel more strongly about the item they are rating (both positively or negatively). For example, you are more likely to tell me that you hated something rather than go out of your way to tell me something was just ok. In both datasets used in this project, the data showed that ratings were more skewed to the higher side (Figures 1 and 5) meaning that users who liked the book were more likely to rate it. This could potentially artificially inflate the overall rating from the true overall and also provides less resolution between differences in ratings as they are all compacted into a smaller range.

Finally, the last challenge I'll mention is with my computing power. Although I would love to have a dataset for every single book title in the world, I ran into issues trying to run the rating-based approach and had to artificially trim down the small dataset I did have farther than normal to not crash my computer. If this exploration is repeated, I would attempt to avoid that trimming.

Analysis

For both approaches, I input the same two books as prompts for recommendations: *The Da Vinci Code* by Dan Brown and *A Christmas Carol* by Charles Dickens. Both books should be sufficiently popular enough to be in the dataset and have user ratings in addition to being familiar to the general reader in the sense of storyline. The recommendations returned are in Figures 3 and 4 for the rating-based approach and Figures 7 and 8 for the content-based approach.

Both approaches often listed the input search title as a result. This makes sense seeing as there were no safeguards put in place to prevent such a result, but also because the item would have an identical rating and plotline as itself.

The results shown for the rating-based approach appear to be all over the place in terms of topic, genre, and just about any other characteristic. They may be highly rated, but the list doesn't feel tailored to the inputted search. When I changed the search to *A Christmas Carol*, my recommendation list dropped down to a single item instead of the 10 that I was expecting and although I am not familiar with the book *Fried Green Tomatoes at the Whistle Stop Cafe*, I would hazard a guess that it is not anything like *A Christmas Carol*. and that it is a poor recommendation. Put simply, this recommender system seems to simply try to recommend the more popular books.

When we look at the recommendations produced by the content-based approach, not only are we able to see more information about the book (i.e., the genres it falls into), but we also get a sense of a theme and cohesiveness between the recommended items.

Where the items from the rating-based approach felt like random, albeit highly rated, items being returned, the list for *The Da Vinci Code* has Dan Brown's other works as well as themes of Leonardo Da Vinci and similar genres. The list returned for *A Christmas Carol* has strong holiday vibes and is a very different list from the one produced from *The Da Vinci Code*.

Conclusion

Overall, the output from the content-based recommendation approach is much more satisfying than the one based on simply ratings. In the content one, there is a visible connection within the recommendations that you feel like your search was seen. That feeling does not reappear in the rating-based. Just because a book is highly rated does not mean that it will necessarily be highly rated for me.

Both approaches have their limitations. While we saw here that a ratings-based approach has no concern with the subject matter of a book and thus produces rather random recommendations, the content-based approach only takes into account the keyword similarities it finds in the written description. It does not consider other factors like age group or genre preference or even book length which was available in this dataset. It also only works (at least for right now) for English titles. Before either approach, or even a hybrid approach between the two, is more widely used. These limitations should be addressed.

We all have different tastes and being able to cater to those individual tastes is an ability that can make companies money. This type of recommender system project is valuable as it applies to many different scenarios from books to movies to even places to eat and travel and more. This means that adapted versions of this project are useful for a broad range of companies even beyond the three book-related entities I mentioned at the beginning.

Ethical Concerns

There should not be many ethical concerns with this project considering all the data is composed of freely available information about the books themselves and the user ratings were provided voluntarily. The data itself does not contain any personally identifiable information and how we are using it should pose little harm to others.

How the data itself was collected from the source into these datasets I cannot speak to. According to the acknowledgments, the first set was collected from the Book-Crossing Community and any user data was anonymized before I downloaded it. The second set was scraped from the Goodreads website according to the dataset description. This second set has no personal information, merely summaries such as average user ratings.

One other possible concern could be that the recommendation system doesn't filter its recommendations to a particular audience demographic. A young reader inputting the same search might get the same recommendations as an adult reader which could include more mature material. Although this applies to both approaches, it applies more strongly to the ratings-based approach since the content-based one can take into account topics and provides the viewer visibility of the genres to help guide final selections.

Appendix

Figures

Ratings-Based Approach

Figure 1. Distribution of Average Book Ratings

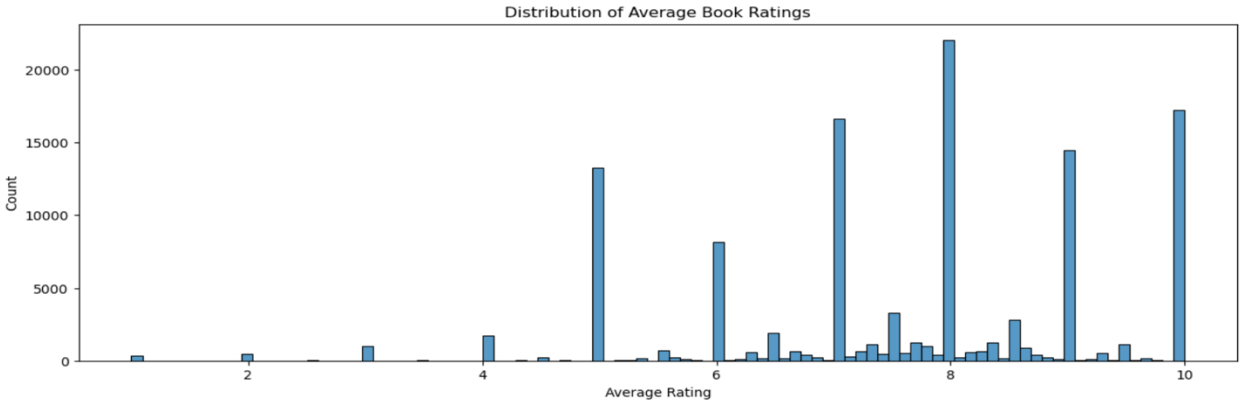

Figure 2. Distribution of Ratings Count Per Book

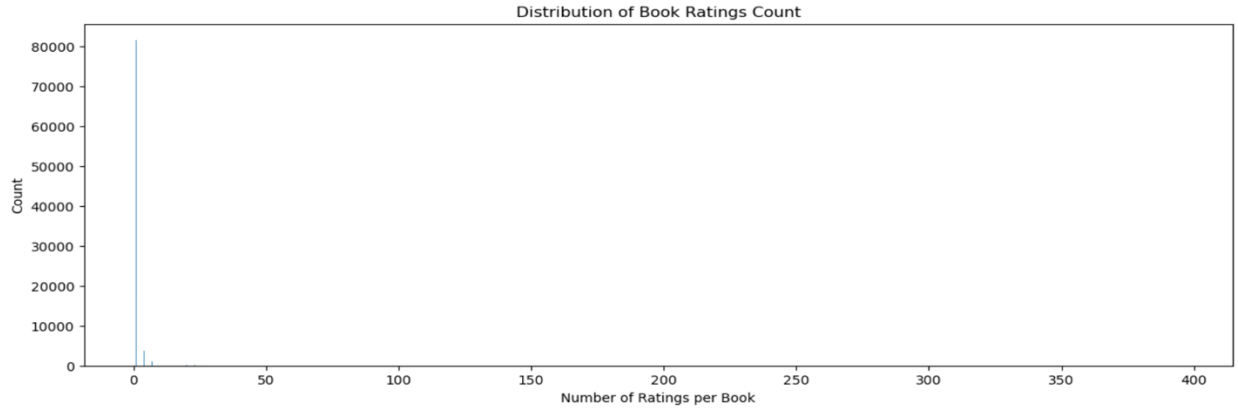

Figure 3. Recommendations for The Da Vinci Code *from Rating-based System*

TitleAuthor	Correlation	RatingsNum
Let Me Call You Sweetheart by Mary Higgins Clark	1.0	28
On the Road (Penguin 20th Century Classics) by Jack Kerouac	1.0	26
The Lion, the Witch, and the Wardrobe (The Chronicles of Narnia, Book 2) by C. S. Lewis	1.0	37
Foucault's Pendulum by Umberto Eco	1.0	30
The Da Vinci Code by Dan Brown	1.0	314
The Mulberry Tree by Jude Deveraux	1.0	41
Open House (Oprah's Book Club (Paperback)) by Elizabeth Berg	1.0	34
Shell Seekers by Rosamunde Pilcher	1.0	26
Speak by Laurie Halse Anderson	1.0	26
Milkrun by Sarah Mlynowski	1.0	30

Figure 4. Recommendations for A Christmas Carol *from Rating-based System*

TitleAuthor	Correlation	RatingsNum
Fried Green Tomatoes at the Whistle Stop Cafe by Fannie Flagg	-1.0	97

Content-based Approach

Figure 5. Distribution of Average Book Ratings

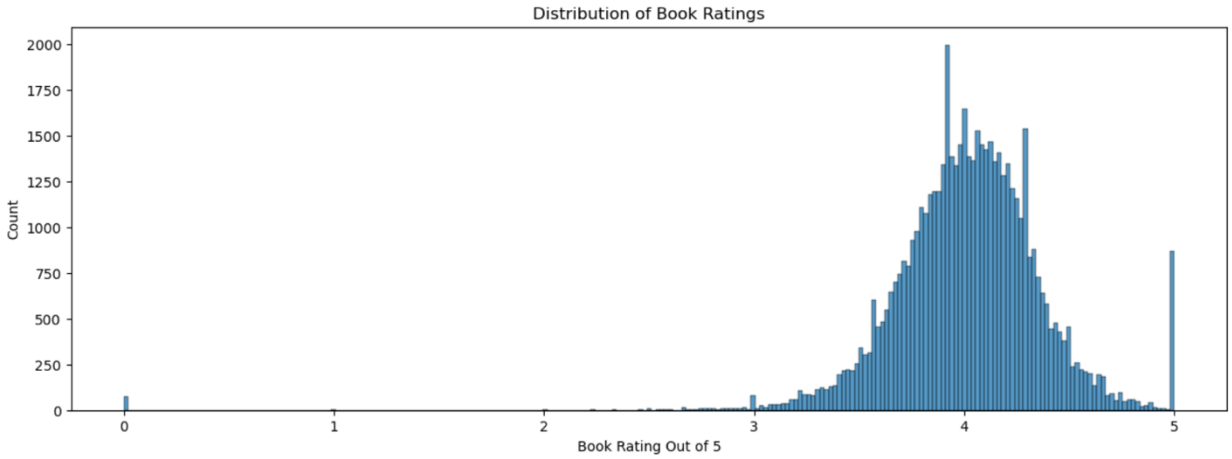

Figure 6. Distribution of Ratings Count Per Book

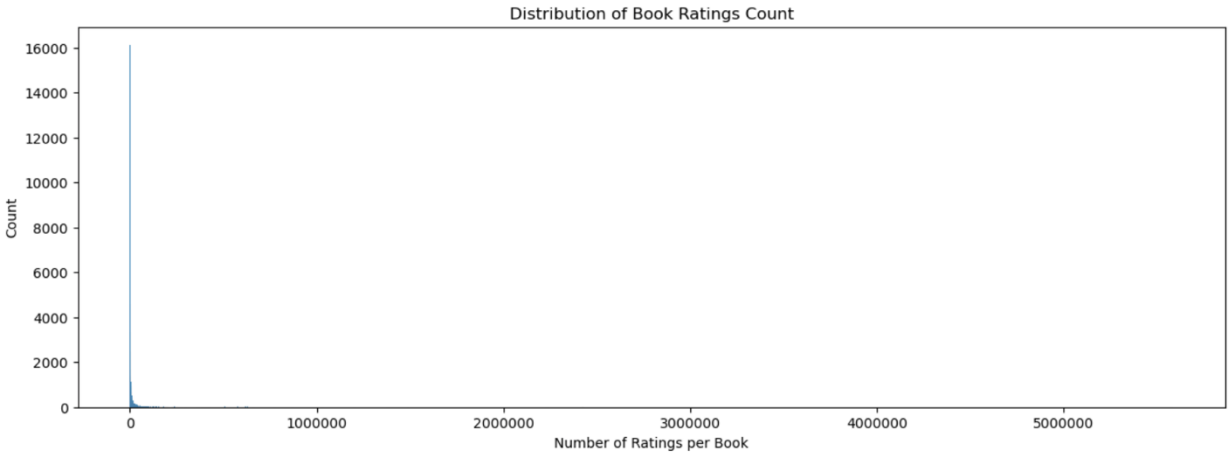

Figure 7. Recommendations for The Da Vinci Code *from Content-based System*

	book_title	book_authors	genres
30961	The Da Vinci Code	Dan Brown	Fiction|Mystery|Thriller
2783	Angels and Demons / The Da Vinci Code	Dan Brown	Fiction|Mystery|Thriller|Historical|Historical...
34365	The Lost Symbol	Dan Brown	Fiction|Mystery|Thriller
22419	Oprindelse	Dan Brown	Fiction|Thriller|Mystery|Thriller|Mystery Thri...
2774	Angels & Demons	Dan Brown	Fiction|Mystery|Thriller
2775	Angels & Demons - Malaikat dan Iblis	Dan Brown|Isma B. Koesalamwardi	Fiction|Mystery|Thriller
21792	O Código Da Vinci	Dan Brown|Celina Cavalcante Falck-Cook	Fiction|Mystery|Thriller
31282	The Devil's Chord	Alex Archer|Michele Hauf	Fantasy|Fiction|Fantasy|Urban Fantasy|Action|A...
37127	The Smile	Donna Jo Napoli	Historical|Historical Fiction|Young Adult|Hist...
10645	Evil in the Beginning	Gary Williams|Vicky Knerly	Mystery|Thriller|Adventure|Fiction
17844	Leonardo da Vinci	Walter Isaacson	Biography|Nonfiction|History|Art|Science
16681	King Dork	Frank Portman	Young Adult|Fiction|Humor|Young Adult|Teen|Mus...
3790	Be Great!: 365 Inspirational Quotes from the W...	Daniel Willey	Classics
17846	Leonardo, the Terrible Monster	Mo Willems	Childrens|Picture Books|Childrens|Childrens|St...
8831	Digital Fortress	Dan Brown	Fiction|Thriller|Mystery|Suspense
17845	Leonardo's Notebooks	Leonardo da Vinci|H. Anna Suh	Art|Nonfiction|History|Science|Classics|Biography
38082	The Uncanny	Sigmund Freud|Adam Phillips|David McLintock|Hu...	Nonfiction|Psychology|Philosophy|Philosophy|Th...
40473	Unbreakable	Kami Garcia	Young Adult|Fantasy|Paranormal|Fantasy|Paranor...
33190	The Hourglass Door	Lisa Mangum	Young Adult|Fantasy|Romance|Science Fiction|Ti...
29410	The Aylesford Skull	James P. Blaylock	Science Fiction|Steampunk|Fantasy|Science Fict...

Figure 8. Recommendations for A Christmas Carol *from Content-based System*

	book_title	book_authors	genres
416	A Christmas Carol	Charles Dickens|Joe L. Wheeler	Classics|Fiction|Holiday|Christmas|Fantasy|Lit...
417	A Christmas Carol and Other Christmas Writings	Charles Dickens|Michael Slater	Classics|Fiction|Holiday|Christmas|Short Stori...
418	A Christmas Carol, The Chimes and The Cricket ...	Charles Dickens|Katharine Kroeber Wiley	Classics|Fiction|Holiday|Christmas|Literature
95	12 Stocking Stuffers	Beverly Barton|Helen Bianchin|Janelle Denison|...	Romance|Contemporary|Holiday|Christmas|Antholo...
30393	The Christmas Box	Richard Paul Evans	Holiday|Christmas|Fiction|Holiday|Inspirational
11309	Finding Noel	Richard Paul Evans	Holiday|Christmas|Fiction|Romance|Holiday
400	A Charlie Brown Christmas	Charles M. Schulz	Holiday|Christmas|Childrens|Childrens|Picture ...
4101	Belstarr The Lost Toymaker	David Jacks|Daniel S. Morrow	Holiday|Christmas|Childrens|Picture Books
30391	The Christmas Basket	Debbie Macomber	Holiday|Christmas|Romance|Holiday|Fiction|Wome...
6321	Christine Kringle	Lynn Brittney	Holiday|Christmas
21604	North Pole Reform School	Jaimie Admans	Young Adult|Holiday|Christmas|Fantasy|Romance|...
32142	The First Christmas Carol	Marianne Jordan	Holiday|Christmas|Christian Fiction|Christian|...
34124	The Life and Times of Scrooge McDuck	Don Rosa	Sequential Art|Comics|Sequential Art|Graphic N...
30394	The Christmas Box Collection: The Christmas Bo...	Richard Paul Evans	Holiday|Christmas|Fiction|Romance
12878	Grace	Richard Paul Evans	Holiday|Christmas|Fiction|Romance|Holiday|Youn...
16799	Kissing Under the Mistletoe	Bella Andre	Romance|Romance|Contemporary Romance|Contempor...
21603	North Pole High: A Rebel Without a Claus	Candace Jane Kringle	Holiday|Christmas|Romance|Young Adult|Humor|Fi...
30392	The Christmas Books, Volume 1: A Christmas Car...	Charles Dickens|Michael Slater	Classics|Fiction|Holiday|Christmas|Short Stories
1674	A wartime Christmas	Carol Rivers	Historical|Historical Fiction|Holiday|Christma...
40496	Uncle Scrooge: Only a Poor Old Man	Carl Barks|Gary Groth|George Lucas	Sequential Art|Comics|Sequential Art|Graphic N...

Data Dictionary for Sources Used

"Book Recommendation Dataset"
(https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

books.csv

Variable	Description	Type	Count	Unique
ISBN	Book ID number	object	271360	271360
Book-Title	Title	object	271360	242135
Book-Author	Author	object	271359	102023
Year-Of-Publication	Publication Year	object	271360	202
Publisher	Publisher	object	271358	16807
Image-URL-S	Small book cover image URL	object	271360	271044
Image-URL-M	Medium book cover image URL	object	271360	271044
Image-URL-	Full size book cover image URL	object	271357	271041

ratings.csv

Variable	Description	Type	Count	Unique
User-ID	Distinguishes between different users	int64	1149780	105283
ISBN	Book ID number	object	1149780	340556
Book-Rating	User rating for that book	int64	1149780	11

"Goodreads' Best Books Ever"
(https://www.kaggle.com/datasets/meetnaren/goodreads-best-books),

book_data.csv

Variable	Description	Type	Count	Unique
book_authors	Author	object	54301	27159
book_desc	Written plot summary of book	object	52970	51781
book_edition	Print edition	object	5453	2134
book_format	Print format	object	52645	147
book_isbn	Book ID Number	object	41435	548
book_pages	Number of pages	object	51779	1403
book_rating	User rating for that book	float64	54301	259
book_rating_count	Number of user ratings	int64	54301	21860
book_review_count	Number of user written reviews	int64	54301	6895
book_title	Title	object	54301	48483
genres	Genres assigned to book	object	51059	30094
image_url	Book cover image URL	object	53618	53618

References

Book Recommendation Dataset. (2024, February 9). https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

Bouguezzi, S., & Bouguezzi, S. (2024, March 18). *How does the Amazon recommendation system work? | Baeldung on Computer Science*. Baeldung on Computer Science. https://www.baeldung.com/cs/amazon-recommendation-system

Domala, J. (2022, January 7). Movie similarity recommendations using Python - Better Programming. *Medium*. https://betterprogramming.pub/movie-similarity-recommendation-using-python-b98a2670a2ad

GfG. (2023, October 20). *Recommendation System in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/recommendation-system-in-python/

Goodreads' best books ever. (2018, December 28). https://www.kaggle.com/datasets/meetnaren/goodreads-best-books

Kharwal, A. (2022, July 19). *Book Recommendation System using Python*. Thecleverprogrammer. https://thecleverprogrammer.com/2022/07/19/book-recommendation-system-using-python/

Moench, K. (2021, May 27). *7 books that share the same title*. BOOK RIOT. https://bookriot.com/books-with-the-same-title/

Sisodia, R. (2022, January 4). Movie Recommendation System - Rahul Sisodia - Medium. *Medium*. https://medium.com/@rahulsisodia06/movie-recommendation-system-c8113226c0aa