ISyE 6740 – Fall 2020
Final Report

Team #66
Team Member Names:

- Contarino_Julia, 903554399
- Stephen_David, 903455230

Project Title: Good Books Recommendation

Problem Statement:

Recommendation systems are nothing new. You can find them on any platform where their main purpose is to sell goods or keep your attention. One could argue that due to our society being flooded with content, across all media types, it's more difficult to choose what content to consume. There are plenty of big-name companies that devote time and effort towards their tv, movie, and music platforms, but books are a different story.

There seem to be two main options for book recommendations; publications that review books, and word of mouth recommendations. Amazon lets users leave star ratings (1-5) with some text to explain the rating. Then there are forums like the Books Subreddit, Reddit.com/r/books, that publishes lists of community decided recommendations broken down by genre. While both options can produce good results, they lack the personalization of the algorithms devoted to tv, movies, and music.

One site that bridges the gap is GoodReads. The idea behind it is that you tell the site what genres you typically enjoy and what books you've previously read and provide a star rating (1-5) much like Amazon. They will then recommend books to you based on your personal taste, with the recommendations getting better the more feedback you provide. Our goal is to test different methods of recommending books and see if we can produce better results than GoodReads.

Data Source:

The main data source we have chosen to use is from the GoodReads site (https://www.kaggle.com/zygmunt/goodbooks-10k?select=to_read.csv)*. This dataset consists of many different csv files containing different information as follows:

- book_tags.csv: book_id, tag_id, and a count
- books.csv: book_id, best_book_id (the most popular edition of a work), work_id, books_count (the count of editions), authors, original_publication_year, original_title
- ratings.csv: book_id, user_id, rating (on a scale of 1 to 5)

- tags.csv: tag_id, tag_name (list of different subjects of books)
- to_read.csv: user_id, book_id (contains books user have selected “to read” on)

*A known issue with this dataset is that it has duplicate ratings. The data source with these duplicates removed is located at https://github.com/zygmuntz/goodbooks-10k.

This dataset contains 10,000 books, 34,525 tag names with 999,912 tags related to each book which comes out to about 100 tags per book. The ratings dataset contains 5,976,479 ratings from 53,424 unique users for the 10,000 books in our set. The to_read.csv file contains 912,705 books that the users have indicated they are interested in reading.

When we further explored our dataset, we noticed some peculiarities in the data. For instance, of the 10,000 books 186 of them are not in English. For our purposes, as we are recommending books without information about the users, we will assume they all speak English and only English and therefore would not want to be recommended books in other languages, so we have removed those records. There are also 1,084 books with missing values for language but looking through the values it seems that most of these are in English, so we kept the books with these missing values. This assumption means that some books that are not in English will have made it through the cracks, but we estimate that number to be about 1% of the total books in our dataset.

We also added a book_genres.csv file that contains a list of the most common book genres. We later used this file to group user-inputted tags into these genre categories to more easily compare the books in our dataset.

Methodology:

Our main method for recommending books to users is through the practice of collaborative filtering. Collaborating filtering is a way to recommend items to users based on their similarity to other users and based on the similarity of past books the user has indicated they enjoy. In essence these are our two metrics when considering books for recommendations: similarity of users and similarity of books. Each team member tackled one of these metrics.

First, let’s look at the case of finding similar users. The first step in this process is to create a matrix A of all the users (identified by their userID) as the rows (m as the number of users) and each book as the columns (n as the number of books):

$$A \in R_{m\times n}$$

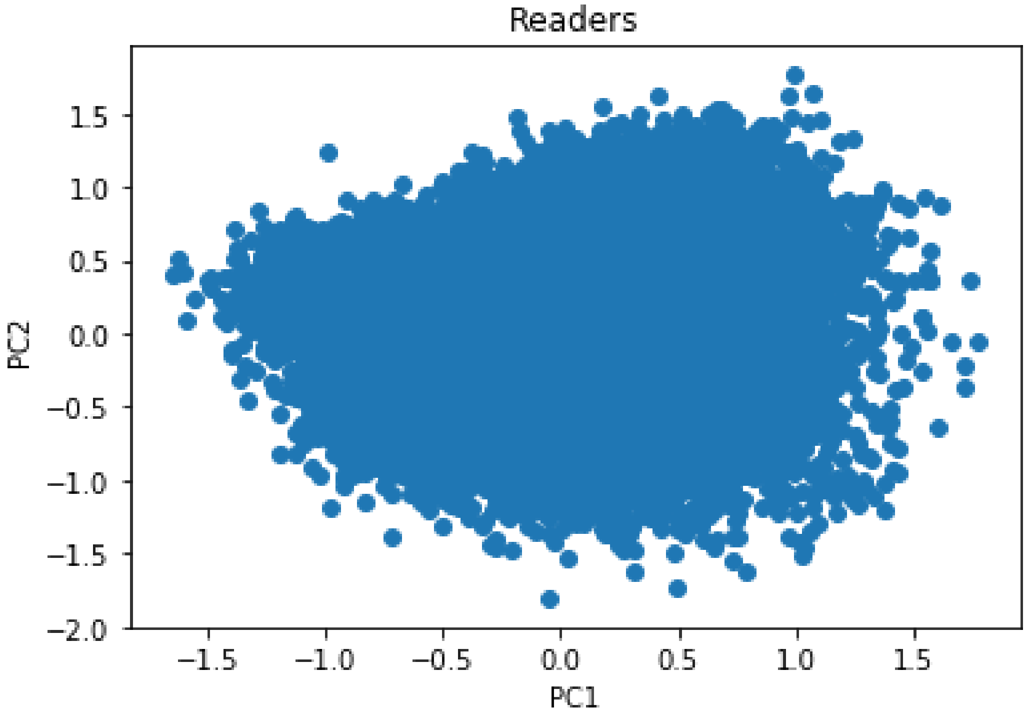
This is a sparse matrix as most users will not have rated every book. Once we have this matrix, we can use it to create a matrix U of all the users to their features for each book:

$$U \in R_{m\times d}$$

and a matrix V of all the books and their features:

$$V \in R_{n\times d}$$

. Using Principal Component Analysis (PCA), we performed eigendecomposition on matrix A to reduce the dimensions of each user's feature vector from 6815 to 2. This allowed us to plot each user as a data point in 2-dimensions to visually compare users. From here we can compare readers with similar ratings by simply using a pairwise distance matrix for all data points above. This proved more difficult than originally thought as we ran into memory issues producing a distance matrix of size 53424 x 53424. We eventually built a new PCA model using half the data; specifically, half the data with the highest number of book ratings to avoid including useless data points.



With this new pairwise distance matrix, we can then easily compare similar users by finding those with the lowest distance metric. We tested both Euclidean and cosine distance metrics as we read that cosine dissimilarity might produce better results but wanted something to compare results against. This helped us get a measure of which users have the smallest angle of difference in the space of the data. With the five most similar users we could then recommend their ten collectively highest rated books.

Secondly, to create a list of books based on a book a user has liked previously, we searched for all the users that have reviewed that book. We then found the books that are correlated closely with our sample book to find the books that are most similar.

To better ensure that the books our system recommended were most similar to the inputted book, we wanted to make sure the books had similar genres. This ended up being a bigger problem than initially accounted for. The dataset we found didn't label books by genre, instead there was a file in it that contained about 34,000 tags that GoodReads users manually entered into the site. Some of these tags made no sense in relation to the books (for instance some values were simply numbers) and so we reduced the number of tags to contain only useful tags. We then wanted to group these remaining tags names into categories that we could compare. We found a dataset of the most commonly used book genres and decided to use these as our genre labels. We then sorted the user inputted tags into these genres based on if the text the users inputted contained the key words of the genres. We then grouped each book by its genre and summed the number of times a user labeled it into each genre to get a ranking of its most labeled genres.

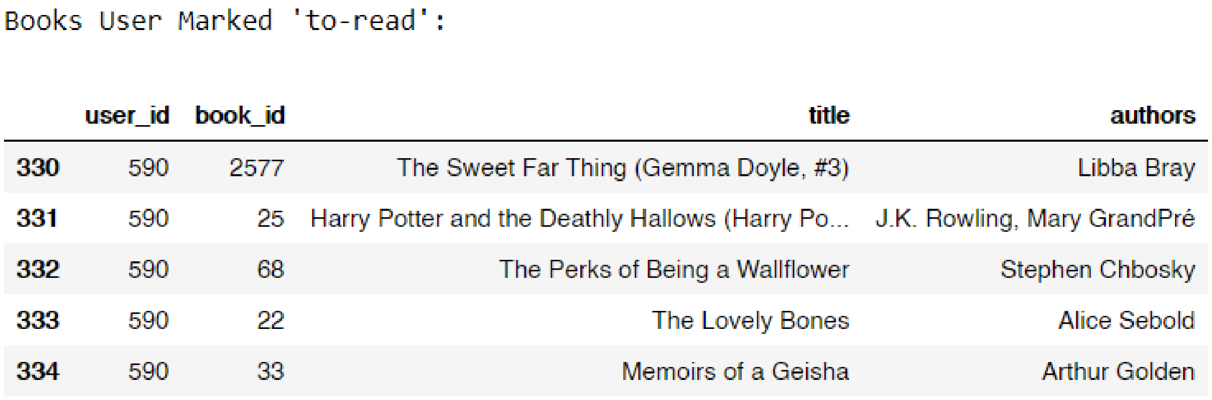
Once we had our set of most correlated books to our input book and the genres of that book, we then combined these results with a threshold for similar genres score of 70% or higher to create a list of the ten most similar books. However, we noticed that these similar books included the next few books in a series or other books from the author of the input book. We rationed that if a user knew about a certain book and liked it enough to find similar books to it, they would already know about reading the following books in that series or would explore that author on their own and recommending these options to them would be redundant. We therefore put a limit on repeating the author in the top ten recommended similar books.

Evaluation:

Our initial thought for testing our results was to use the dataset, "to_read", which includes all users and the book_id of books that they have marked as wanting to read in the future. The idea was that the success of our recommendations could be measured by the percentage of books on a user's "to_read" list that we could correctly identify. This plan didn't end up working so well when we went to implement it. The scores were very low and most times there wasn't a matching book. The problem with measuring the success of a recommender system is that the results will be opinion data. This makes tuning these models difficult. While the initial results of choosing between Euclidean and Cosine distance metrics seem like they would produce different results, it's difficult to tell whether one book (or set of books) is the correct recommendation without a user's input. However, the examples we ran through including those below show that while the exact books a user has in the "to_read" dataset don't match up to those we recommended all that often, the similarity of those books we recommend and those that are in the "to-read" dataset is still high.

Finding recommendations is a difficult problem to solve as it deals with a user's likes and dislikes which is dependent on a number of features and even if every factor is accounted for, there is still a chance the person behind the numbers won't like the book recommended.

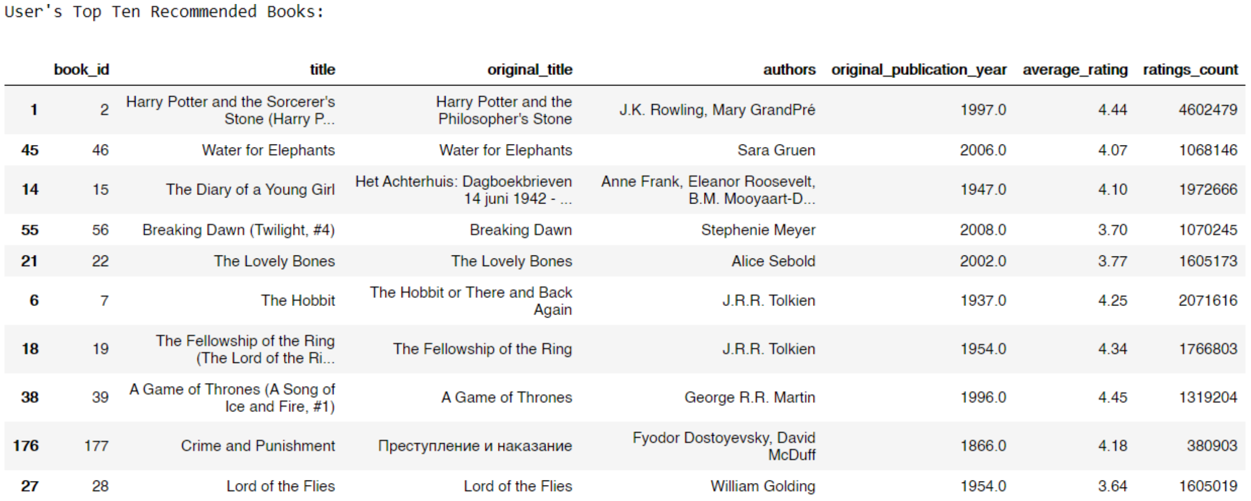
The example below shows one of our users and their list of books they have indicated they wanted to read:



```
Books User Marked 'to-read':
```

	user_id	book_id	title	authors
330	590	2577	The Sweet Far Thing (Gemma Doyle, #3)	Libba Bray
331	590	25	Harry Potter and the Deathly Hallows (Harry Po...	J.K. Rowling, Mary GrandPré
332	590	68	The Perks of Being a Wallflower	Stephen Chbosky
333	590	22	The Lovely Bones	Alice Sebold
334	590	33	Memoirs of a Geisha	Arthur Golden

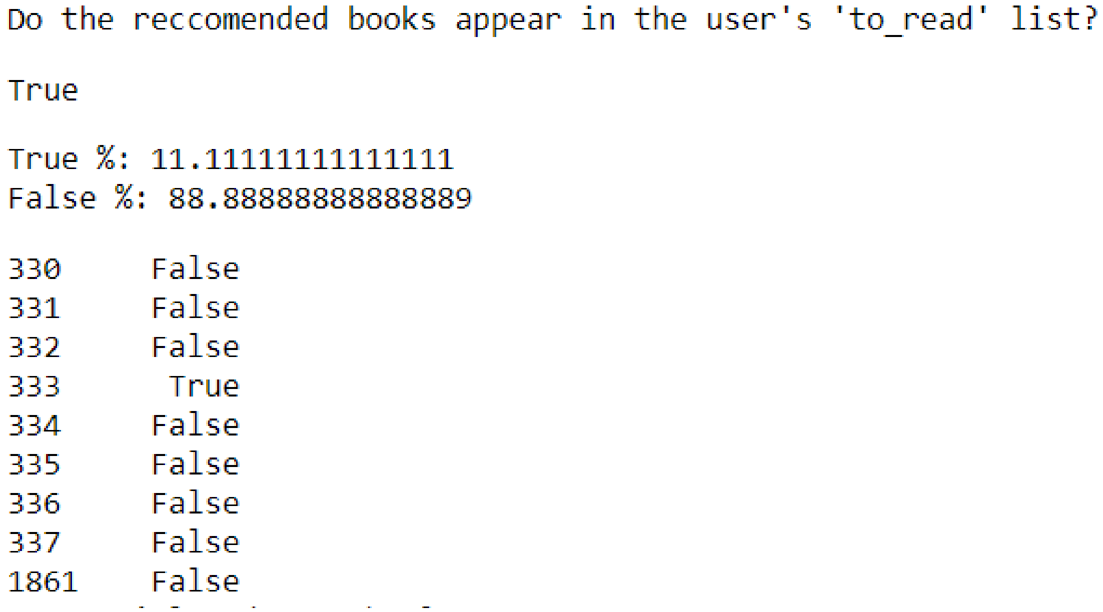
Below are the same user's top ten recommended books from our system based on similar users:



```
User's Top Ten Recommended Books:
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
1	2	Harry Potter and the Sorcerer's Stone (Harry P...	Harry Potter and the Philosopher's Stone	J.K. Rowling, Mary GrandPré	1997.0	4.44	4602479
45	46	Water for Elephants	Water for Elephants	Sara Gruen	2006.0	4.07	1068146
14	15	The Diary of a Young Girl	Het Achterhuis: Dagboekbrieven 14 juni 1942 - ...	Anne Frank, Eleanor Roosevelt, B.M. Mooyaart-D...	1947.0	4.10	1972666
55	56	Breaking Dawn (Twilight, #4)	Breaking Dawn	Stephenie Meyer	2008.0	3.70	1070245
21	22	The Lovely Bones	The Lovely Bones	Alice Sebold	2002.0	3.77	1605173
6	7	The Hobbit	The Hobbit or There and Back Again	J.R.R. Tolkien	1937.0	4.25	2071616
18	19	The Fellowship of the Ring (The Lord of the Ri...	The Fellowship of the Ring	J.R.R. Tolkien	1954.0	4.34	1766803
38	39	A Game of Thrones (A Song of Ice and Fire, #1)	A Game of Thrones	George R.R. Martin	1996.0	4.45	1319204
176	177	Crime and Punishment	Преступление и наказание	Fyodor Dostoyevsky, David McDuff	1866.0	4.18	380903
27	28	Lord of the Flies	Lord of the Flies	William Golding	1954.0	3.64	1605019

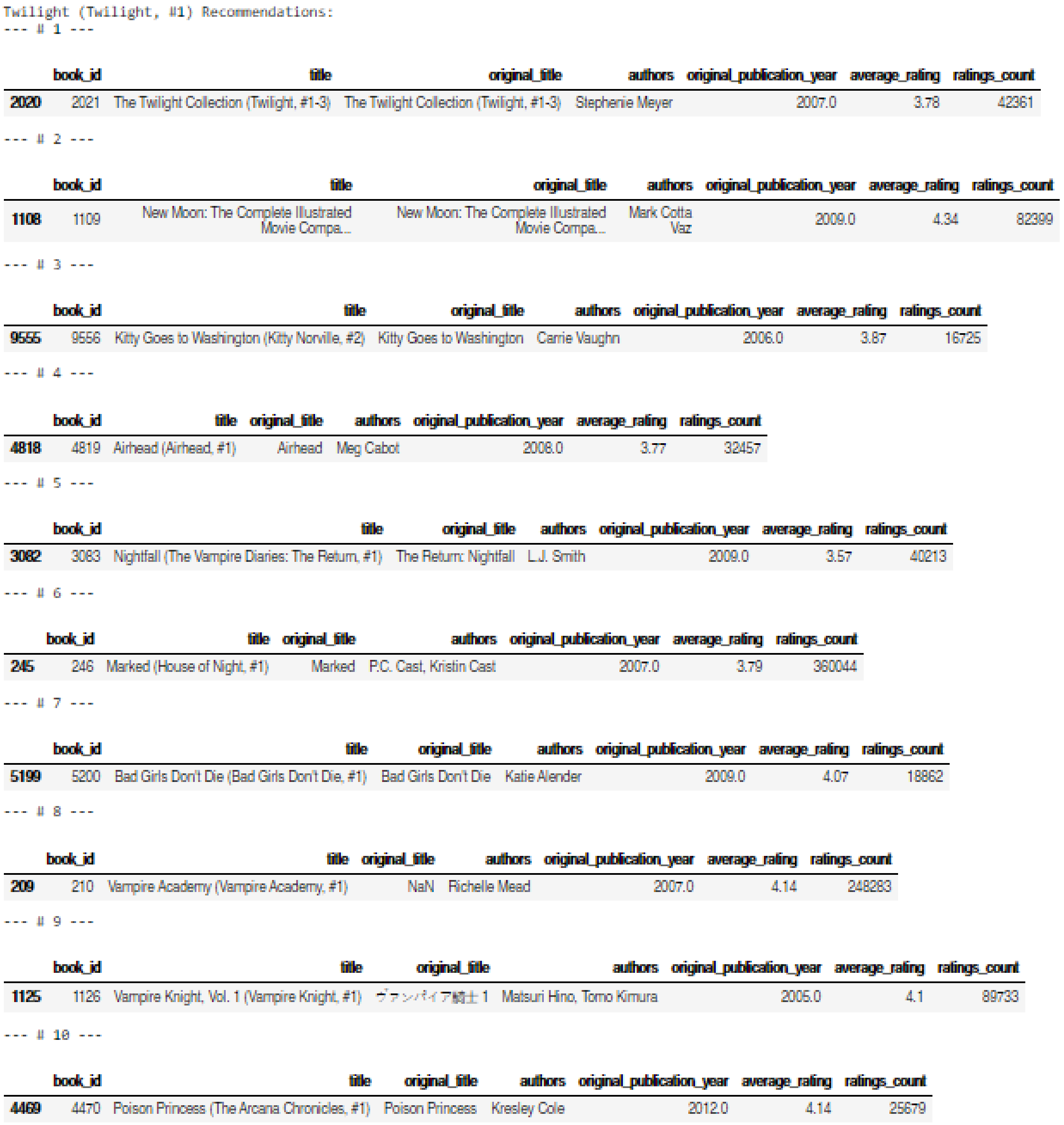
Below is the resulting accuracy score for this user. The system recommended one of the books the user has indicated they are interested in:



```
Do the reccomended books appear in the user's 'to_read' list?

True

True %: 11.11111111111111
False %: 88.88888888888889

330     False
331     False
332     False
333      True
334     False
335     False
336     False
337     False
1861    False
```

This example user and their resulting recommendations show that even though only one of the books we recommended are in their "to_read" list, the books we recommend that aren't in their list are still similar to those that in their list. For instance, the user indicated they are interested in reading the last book in the Harry Potter series, but as they haven't ranked any of the books in that series, the system recommended them the first book in that series.

The example book below shows that the book similarities are working as we would expect as well. When we plug in the book Twilight, a book that notoriously features fantasy vampires, we see that the system recommends other fantasy themed novels (recommendations 3, 6,7, and 10). The system, however, takes it a step further. Even though we don't have a genre specifying vampire novels, the system recommends books with vampire in their title (recommendations 5, 8, and 9):



```
Twilight (Twilight, #1) Recommendations:
--- # 1 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
2020	2021	The Twilight Collection (Twilight, #1-3)	The Twilight Collection (Twilight, #1-3)	Stephenie Meyer	2007.0	3.78	42361

```
--- # 2 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
1108	1109	New Moon: The Complete Illustrated Movie Compa...	New Moon: The Complete Illustrated Movie Compa...	Mark Cotta Vaz	2009.0	4.34	82399

```
--- # 3 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
9555	9556	Kitty Goes to Washington (Kitty Norville, #2)	Kitty Goes to Washington	Carrie Vaughn	2006.0	3.87	16725

```
--- # 4 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
4818	4819	Airhead (Airhead, #1)	Airhead	Meg Cabot	2008.0	3.77	32457

```
--- # 5 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
3082	3083	Nightfall (The Vampire Diaries: The Return, #1)	The Return: Nightfall	L.J. Smith	2009.0	3.57	40213

```
--- # 6 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
245	246	Marked (House of Night, #1)	Marked	P.C. Cast, Kristin Cast	2007.0	3.79	360044

```
--- # 7 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
5199	5200	Bad Girls Don't Die (Bad Girls Don't Die, #1)	Bad Girls Don't Die	Katie Alender	2009.0	4.07	18862

```
--- # 8 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
209	210	Vampire Academy (Vampire Academy, #1)	NaN	Richelle Mead	2007.0	4.14	248283

```
--- # 9 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
1125	1126	Vampire Knight, Vol. 1 (Vampire Knight, #1)	ヴァンパイア騎士 1	Matsuri Hino, Tomo Kimura	2005.0	4.1	89733

```
--- # 10 ---
```

	book_id	title	original_title	authors	original_publication_year	average_rating	ratings_count
4469	4470	Poison Princess (The Arcana Chronicles, #1)	Poison Princess	Kresley Cole	2012.0	4.14	25679

If we were to replicate this analysis, not only would a bigger dataset with more clearly defined genres be useful but including more recent books would help the analysis as well. The most recent books in our current dataset were released in 2017. Times and opinions are constantly altering, and a great recommendation system would capture the intricacies of these changes. Who knows, maybe there'd even be a whole new genre of quarantine books from 2020.