|  |  |
| --- | --- |
| GOODREADS BOOK RECOMMENDATION ENGINE  USING R LAUNGUAGE | CAPSTONE PROJECT  This project will able to help readers to choose best book according to other readers reviews & ratings.  Submitted by: Ruhi Choudhary  EDUBRIDGE – BATCH – 2022- 07265 |

# Problem Statement

When we plan to buy a book specially the scientific books, we normally ask about the goods ones, research the book domains, compare the books with similar or read the reviews so here the recommender system is the master of this problem. This will help readers to find relevant book according to their interest/taste.

The basic idea behind analyzing the Goodreads dataset is to get a fair idea about the relationships between the multiple attributes a book might have, such as: the aggregate rating of each book, the trend of the authors over the years and books with numerous languages. With over a hundred thousand ratings, there are books which just tend to become popular as each day seems to pass.

We've always considered the magical persona books seem to hold, and with this notebook, we step out on a journey to see what kind of books really drives people to read in this era of modern smart devices.

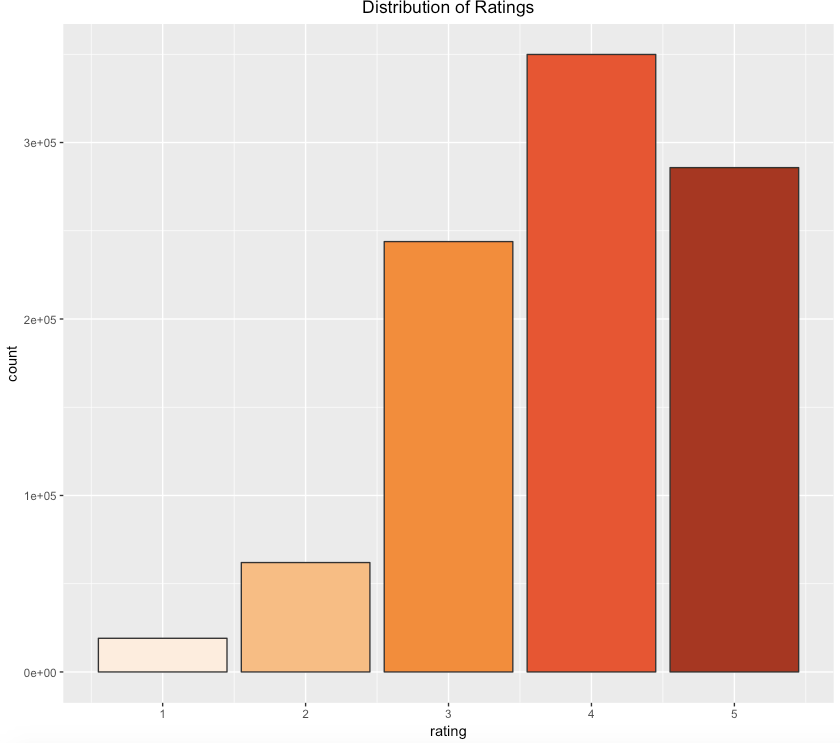
# Dataset description

* **ratings.csv** – contains all users’ ratings of the books (980k ratings, for 10k books, from 53424 users)
* **books.csv** – contains information on books such as author, year, etc.
* **book\_tags.csv** – contains all tag\_id’s users have assigned to that book and corresponding tags count.
* **ttags.csv** – contains the tag\_names corresponding to tag\_id’s .

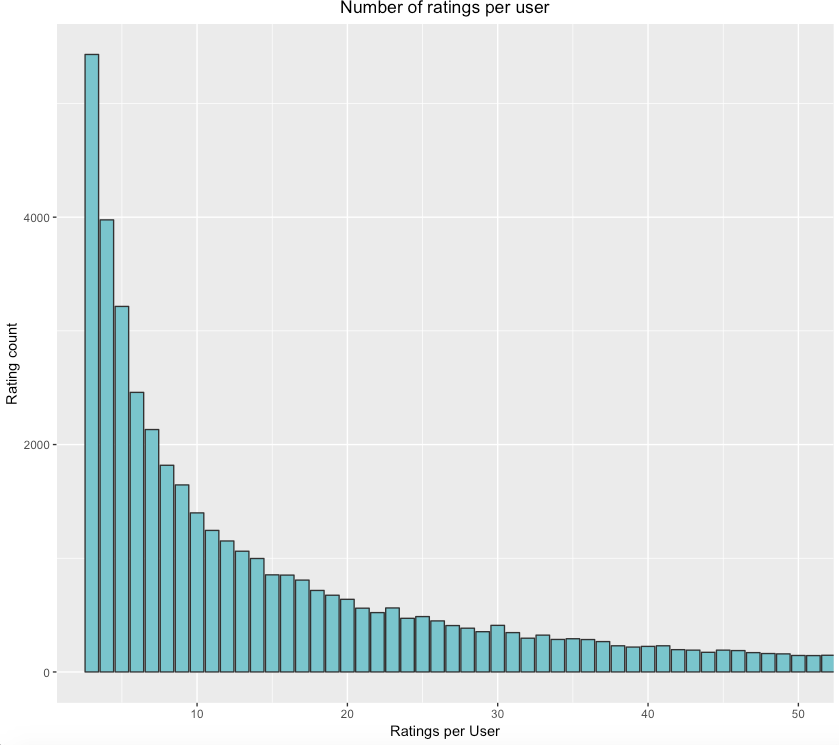
Last two files are linked by the book\_id.

# Data Exploration

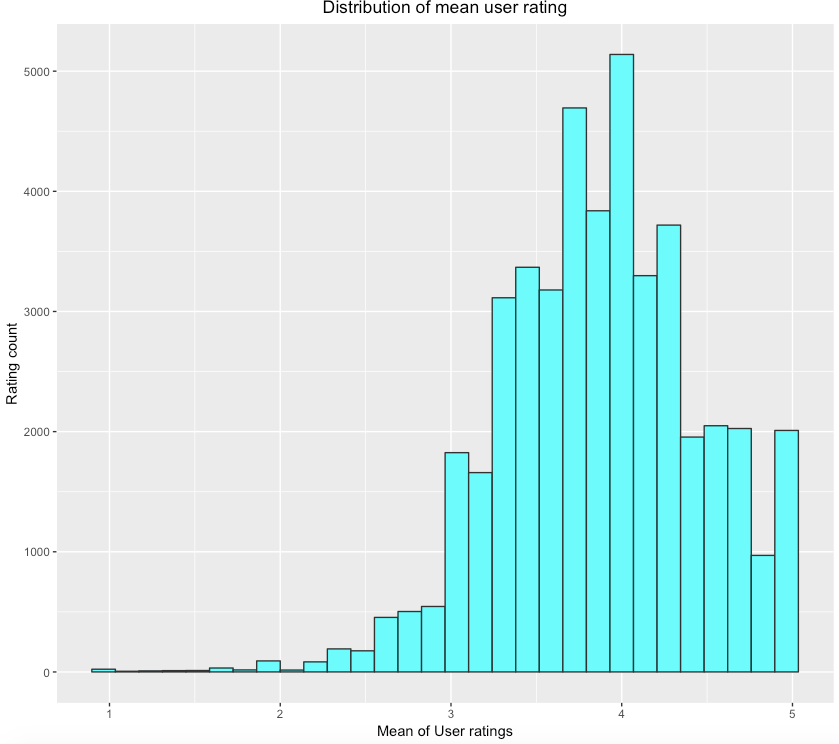
* 1. *Distribution of ratings:* Most of the ratings are in the 3-5 range, hence, people tend to give ratings from average to excellent.



* 1. *Number of ratings per user:*

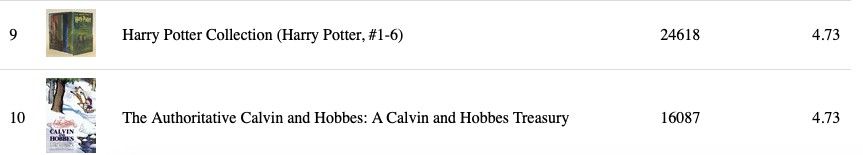


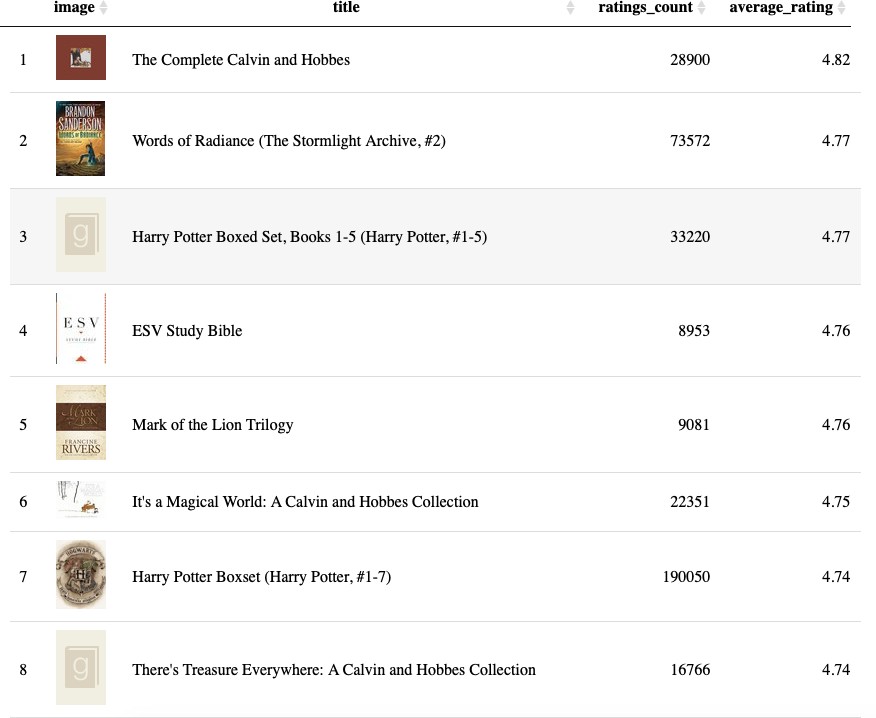
* 1. *Distribution of mean user rating:*



Some people give 5 rating to a mediocre book while others don’t unless its excellent. From the plot above, it can be seen that on the right side of the bump are the ones rated with a mean of 5 by the users, which means the users really liked that book.

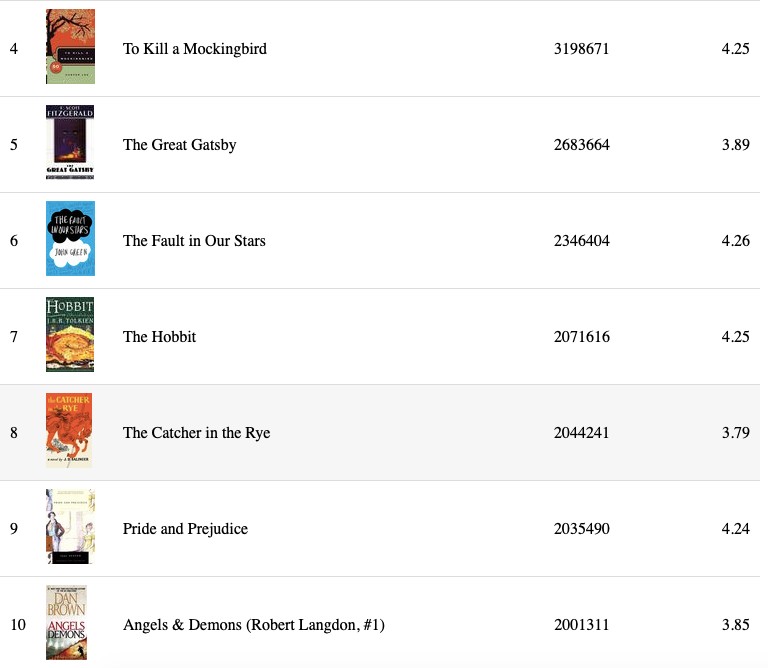
* 1. *10 highly rated books:*



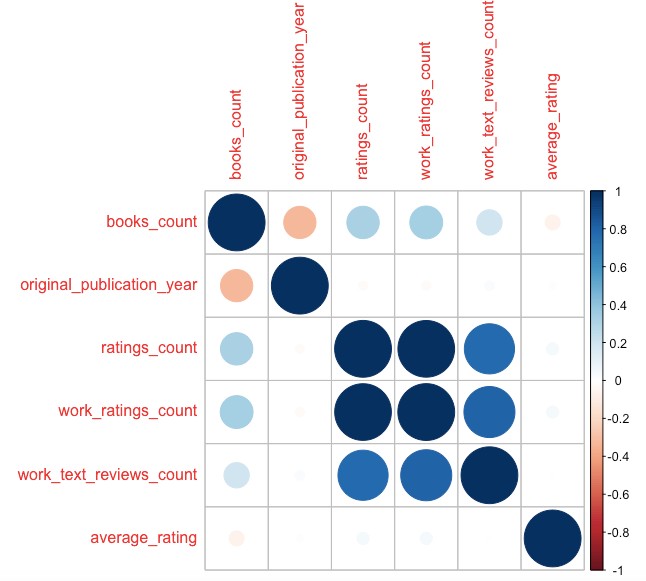


* 1. *10 most popular books:* Books that were rated more often.

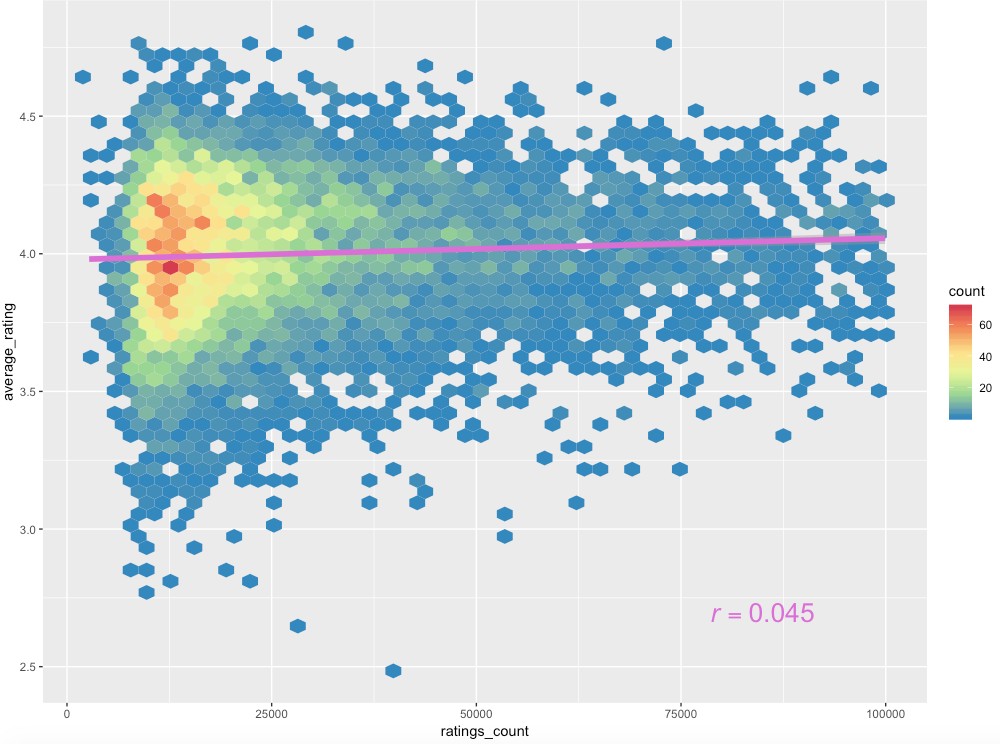


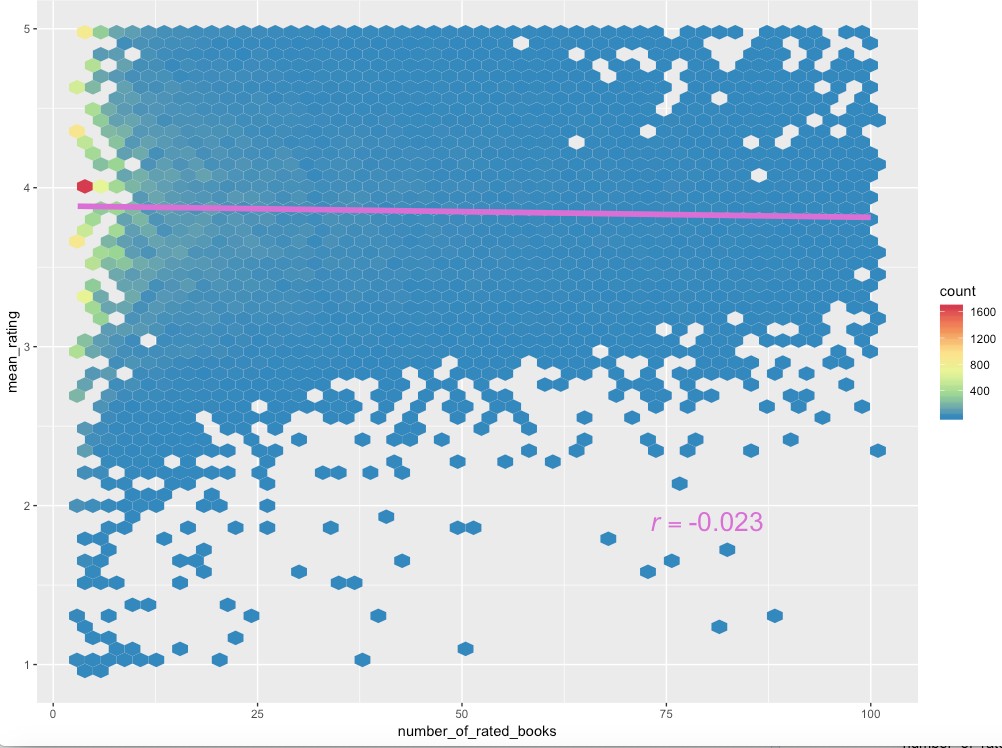


* 1. *Factors influencing book's rating:* Features other than this are affecting.



* 1. ***Relationship between Number of rating and Average rating:***Since value of ‘r’ is very low, there isn’t quite strong relation between the two.



* 1. *Frequent raters:* Frequent raters tend to be more critical, hence they don’t give high ratings.

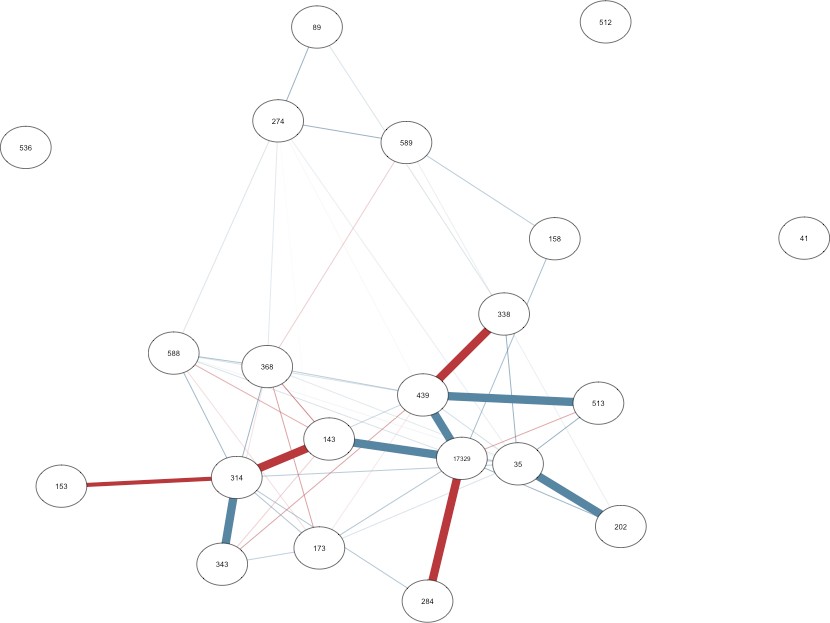
# Collaborative Filtering

In user-based collaborative filtering, following steps are required to build an algorithm for a recommender system:

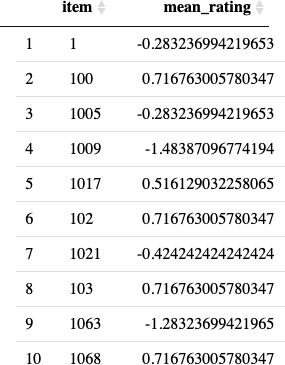
1. Identify other users who have similar interests to that of current user in terms of their ratings for the same books.
2. Take the average rating of books the current user has not yet read.
3. Recommend the books with the highest average rating to the current user.

Structuring the data in such a way that each row corresponds to a user and each column to a book.

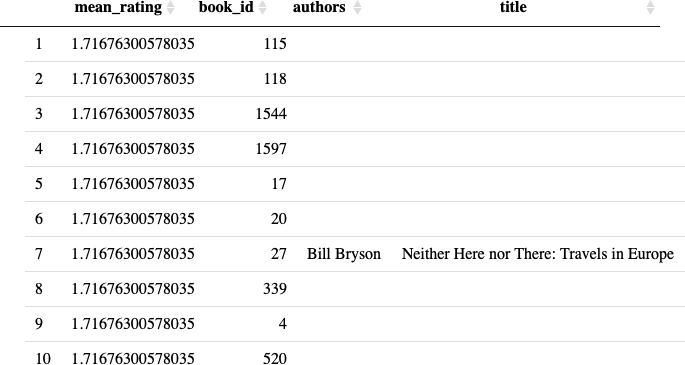
* Select a user (17329)
* Find similar users
* Normalize user ratings by subtracting the users mean from all individuals.
* Calculate similarity of 17329 with all other users.
* Visualizing the similarities: Blue edges being the most similar and red the least.



* Get predictions for other books: In order to get the recommendations for our user (17329) we would take the most similar users and average their ratings for books our user has not yet rated.



* Recommend the best predictions: From above sort the mean ratings and give the best predictions



* **Using Recommenderlab:** Recommender lab is a R-package that provides the infrastructure to evaluate and compare several collaborative-filtering algorithms. Many algorithms are already implemented in the package, and we can use the available ones to save some coding effort, or add custom algorithms and use the infrastructure
* Most of the values in the rating matrix are missing, because every user just rated a few of the 10000 books. Hence, representing this matrix in sparse format in order to save memory.