ADVANCED DATA ANALYTICS

GROUP PROJECT REPORT

E-COMMERCE PRODUCT RATING REVIEW MINING

Presented by: Group 3

GROUP MEMBERS

Tamanna Kapoor	**19BCE10011**
Yash Santani	**19BCE10069**
Aadya Iyer	**19BAI10172**
Khushii Guptaa	**19BCE10339**
Neelangshu Nath	**19BCE10224**
Paragi Vijayvargiya	**19BCE10267**
Indira Kar	**19BCE10351**
Chitrankshi Rathour	**19BCE10419**
Salman Mahboob	**19BCE10088**
Jaishree Sharma	**19BCE10307**
DVS Dinesh Kumar	**19BCE10233**

CONTENTS

1. INTRODUCTION

Recommendation systems are used in hundreds of different services - everywhere from online shopping to music to movies.
For instance, the online retailer Amazon had a heavy hand in developing collaborative filtering algorithms that recommend items to users. Music services like Pandora identify up to 450 uniquely identifying characteristics of songs to find music similar to that of their users' preferences. Other music streaming services, such as Spotify, heavily rely upon the music selections of similar users to make weekly song recommendations and personalized radio stations. Netflix, a popular television and movie streaming service, uses these systems to recommend movies that viewers may enjoy.
We can see how recommendation systems have a surprisingly large impact on the materials consumers engage with over the course of their daily lives.
For this project, we analyzed a dataset on books, observed various features and did exploratory analysis.

2. LITERATURE REVIEW

- Okon et.al. (2018) **[1]** proposed a model that generates recommendations to buyers, through an enhanced CF algorithm, a quick sort algorithm and Object Oriented Analysis and Design Methodology (OOADM). Scalability was ensured through the implementation of Firebase SQL. This system performed well on the evaluation metrics.
- Kurmashov et.al. (2015) **[2]** used Pearson correlation coefficient based CF to provide internet based recommendations to book readers and evaluated the system through an online survey
- Mathew et.al. (2016) **[3]** proposed a system that saves details of books purchased by the user. From these Book contents and ratings, a hybrid algorithm using collaborative filtering, content-based filtering and association rule generates book recommendations. Rather than Apriori, they recommended the use of Equivalence class Clustering and bottom up Lattice Transversal (ECLAT) as this algorithm is faster due to the fact that it examines the entire dataset only once.
- Ayub et.al. (2018) **[4]** proposed a similarity function similar to Jaccard Similarity to locate alike items and users for the enquiring item and user in nearest neighbour based collaborative filtering. They proposed that absolute value of ratings should be taken as against the ratio of co-rated items taken in Jaccard Similarity. They also compared performance of their method with other similarity measures.

3. BACKGROUND

There are three major approaches for recommendation systems:

(i) content-based
(ii) collaborative
(iii) hybrid

Broadly, recommendation systems that implement a content-based (CB) approach recommend items to a user that are similar to the ones the user preferred in the past.
On the other hand, recommendation systems that implement collaborative filtering (CF) predict users’ preferences by analyzing relationships between users and interdependencies among items; from these, they extrapolate new associations.
Finally, hybrid approaches meld content-based and collaborative approaches, which have complementary strengths and weaknesses, thus producing stronger results.

4. APPROACH / METHODOLOGY

To develop the recommendation system, we have performed the following steps:

1. **Data Searching**
   For this project, we have taken the dataset from Kaggle. Kaggle is an online platform that allows users to find and push datasets, explore and build models in a web-based data science environment and a lot more.

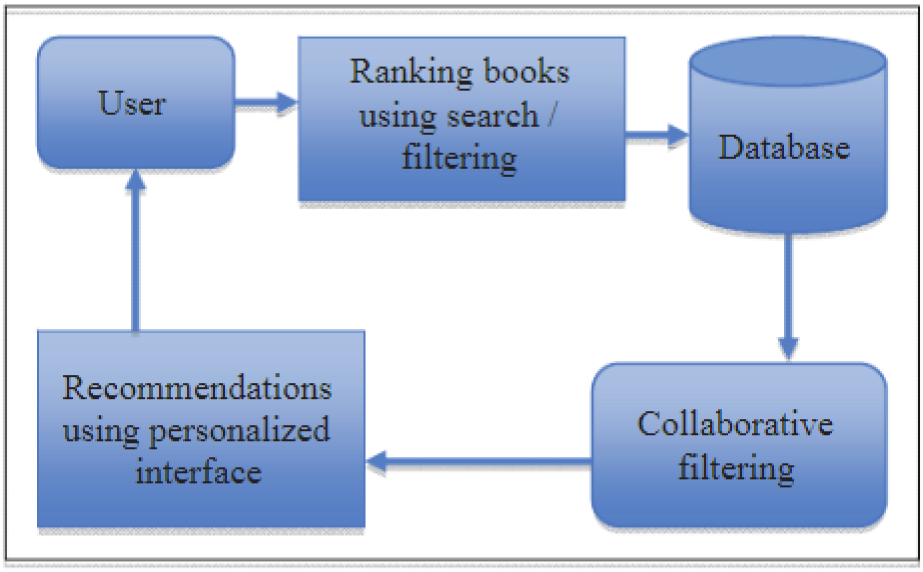
2. **Data Preparation**
   The Data preparation step includes all the activities used to create the data set used during the modeling phase. This includes cleansing data, combining data from multiple sources, and transforming data into more useful variables. In addition, feature engineering and text analysis can be used to derive new structured variables to enrich all predictors and improve model accuracy. The Data preparation phase is the longest.

3. **Exploratory Analysis**
   Used various plots such as bar graphs, scatter plots, pie charts to gain insights from the dataset. Also drew correlation among various variables to get better insights, and plotted graphs for the same.

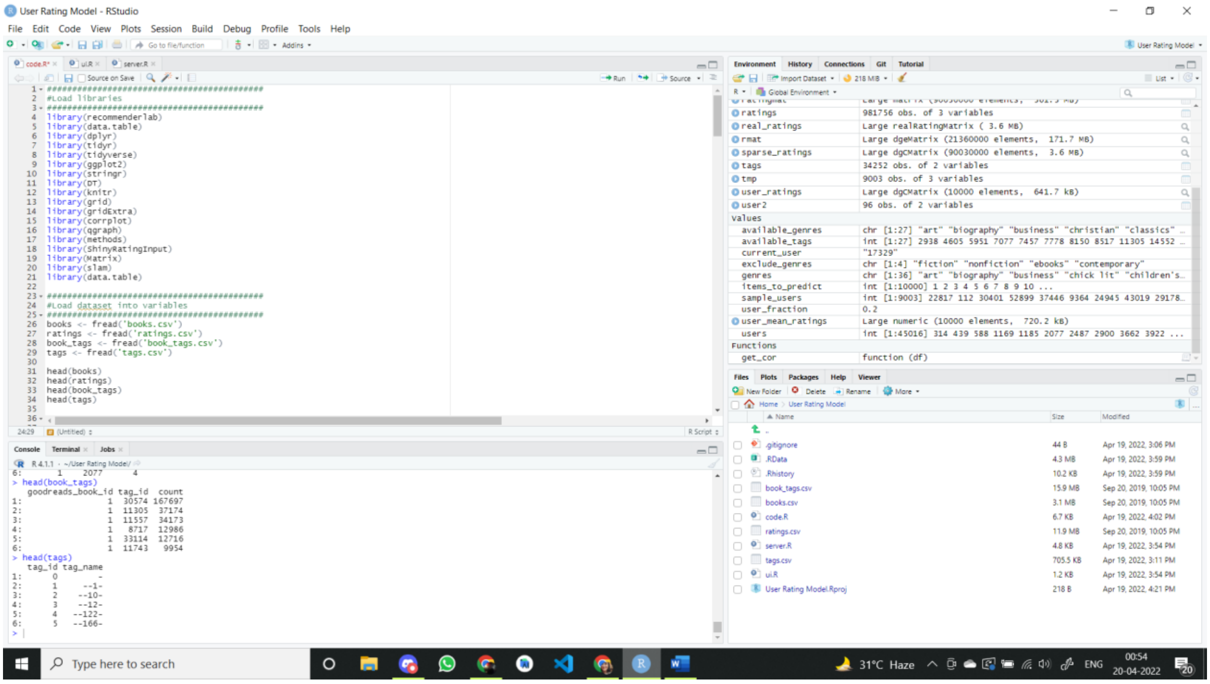
4. **Collaborative Filtering**
   User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user.
   Many websites use collaborative filtering for building their recommendation system.

5. **Shiny**

Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards.


User
Ranking books using search / filtering
Database
Collaborative filtering
Recommendations using personalized interface

5. IMPLEMENTATION

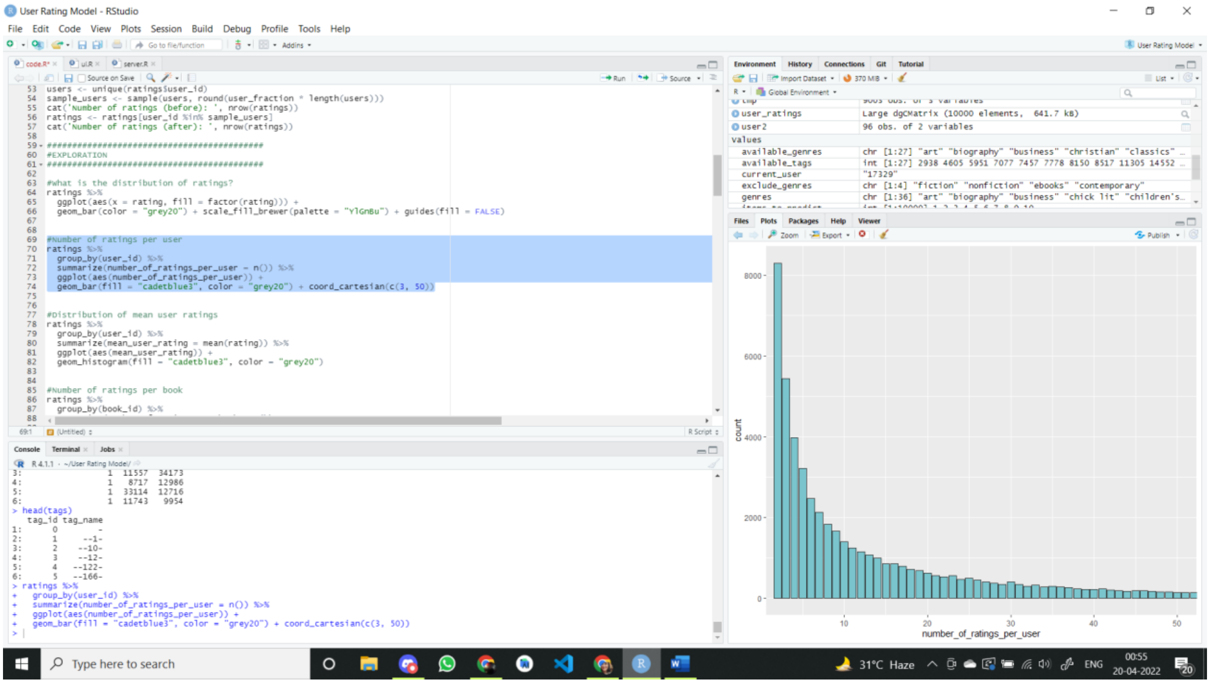


```
##########################################
#Load libraries
##########################################
library(recommenderlab)
library(data.table)
library(dplyr)
library(tidyr)
library(tidyverse)
library(ggplot2)
library(stringr)
library(DT)
library(knitr)
library(grid)
library(gridExtra)
library(corrplot)
library(qgraph)
library(methods)
library(ShinyRatingInput)
library(Matrix)
library(slam)
library(data.table)

##########################################
#Load dataset into variables
##########################################
books <- fread('books.csv')
ratings <- fread('ratings.csv')
book_tags <- fread('book_tags.csv')
tags <- fread('tags.csv')

head(books)
head(ratings)
head(book_tags)
head(tags)

```

```
6:       1     2077       4
> head(book_tags)
   goodreads_book_id tag_id  count
1:                 1  30574 167697
2:                 1  11305  37174
3:                 1  11557  34173
4:                 1   8717  12986
5:                 1  33114  12716
6:                 1  11743   9954
> head(tags)
   tag_id tag_name
1:      0        -
2:      1     --1-
3:      2    --10-
4:      3    --12-
5:      4   --122-
6:      5   --166-
> 
```

ratings	981756 obs. of 3 variables
real_ratings	Large realRatingMatrix (3.6 MB)
rmat	Large dgeMatrix (21360000 elements, 171.7 MB)
sparse_ratings	Large dgCMatrix (90030000 elements, 3.6 MB)
tags	34252 obs. of 2 variables
tmp	9003 obs. of 3 variables
user_ratings	Large dgCMatrix (10000 elements, 641.7 kB)
user2	96 obs. of 2 variables
Values	
available_genres	chr [1:27] "art" "biography" "business" "christian" "classics" …
available_tags	int [1:27] 2938 4605 5951 7077 7457 7778 8150 8517 11305 14552 …
current_user	"17329"
exclude_genres	chr [1:4] "fiction" "nonfiction" "ebooks" "contemporary"
genres	chr [1:36] "art" "biography" "business" "chick lit" "children's…
items_to_predict	int [1:10000] 1 2 3 4 5 6 7 8 9 10 ...
sample_users	int [1:9003] 22817 112 30401 52899 37446 9364 24945 43019 29178…
user_fraction	0.2
user_mean_ratings	Large numeric (10000 elements, 720.2 kB)
users	int [1:45016] 314 439 588 1169 1185 2077 2487 2900 3662 3922 ...
Functions	
get_cor	function (df)

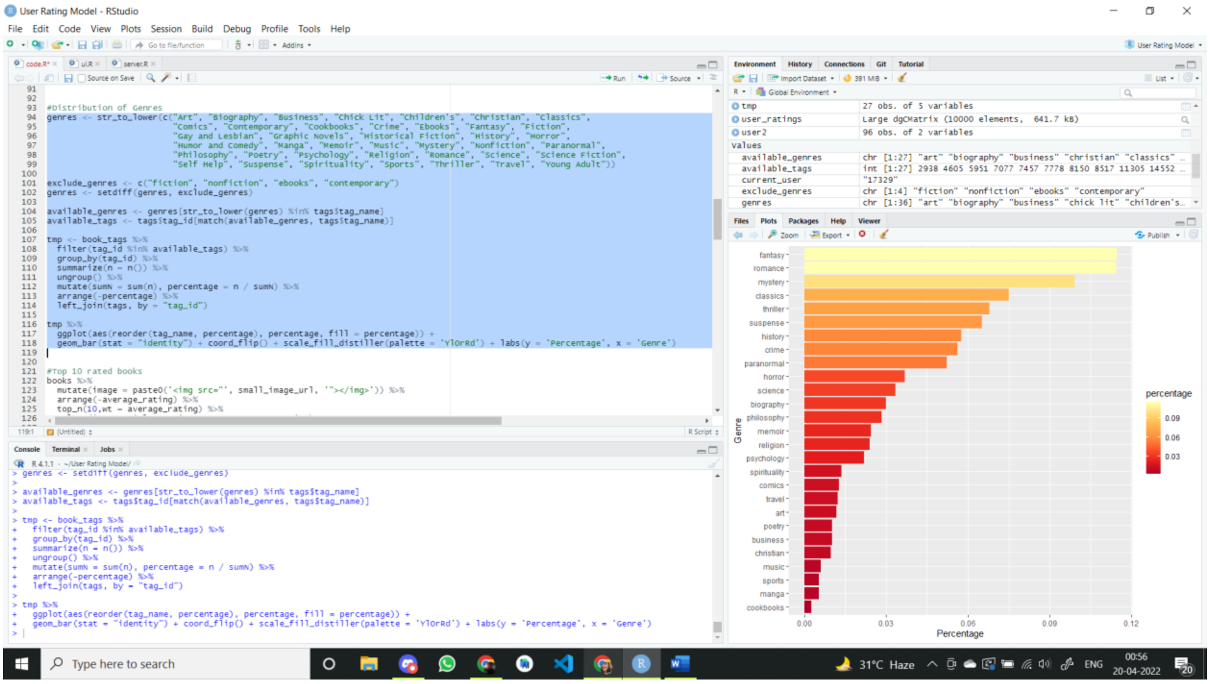
Name	Size	Modified
..		
.gitignore	44 B	Apr 19, 2022, 3:06 PM
.RData	4.3 MB	Apr 19, 2022, 3:59 PM
.Rhistory	10.2 KB	Apr 19, 2022, 3:59 PM
book_tags.csv	15.9 MB	Sep 20, 2019, 10:05 PM
books.csv	3.1 MB	Sep 20, 2019, 10:05 PM
code.R	6.7 KB	Apr 19, 2022, 4:02 PM
ratings.csv	11.9 MB	Sep 20, 2019, 10:05 PM
server.R	4.8 KB	Apr 19, 2022, 3:54 PM
tags.csv	705.5 KB	Apr 19, 2022, 3:11 PM
ui.R	1.2 KB	Apr 19, 2022, 3:54 PM
User Rating Model.Rproj	218 B	Apr 19, 2022, 4:21 PM



```
users <- unique(ratings$user_id)
sample_users <- sample(users, round(user_fraction * length(users)))
cat('Number of ratings (before): ', nrow(ratings))
ratings <- ratings[user_id %in% sample_users]
cat('Number of ratings (after): ', nrow(ratings))

##########################################
#EXPLORATION
##########################################

#what is the distribution of ratings?
ratings %>%
  ggplot(aes(x = rating, fill = factor(rating))) +
  geom_bar(color = "grey20") + scale_fill_brewer(palette = "YlGnBu") + guides(fill = FALSE)


#Number of ratings per user
ratings %>%
  group_by(user_id) %>%
  summarize(number_of_ratings_per_user = n()) %>%
  ggplot(aes(number_of_ratings_per_user)) +
  geom_bar(fill = "cadetblue3", color = "grey20") + coord_cartesian(c(3, 50))


#Distribution of mean user ratings
ratings %>%
  group_by(user_id) %>%
  summarize(mean_user_rating = mean(rating)) %>%
  ggplot(aes(mean_user_rating)) +
  geom_histogram(fill = "cadetblue3", color = "grey20")


#Number of ratings per book
ratings %>%
  group_by(book_id) %>%
```

```
3:                 1  11557  34173
4:                 1   8717  12986
5:                 1  33114  12716
6:                 1  11743   9954
> head(tags)
   tag_id tag_name
1:      0        -
2:      1     --1-
3:      2    --10-
4:      3    --12-
5:      4   --122-
6:      5   --166-
> ratings %>%
+   group_by(user_id) %>%
+   summarize(number_of_ratings_per_user = n()) %>%
+   ggplot(aes(number_of_ratings_per_user)) +
+   geom_bar(fill = "cadetblue3", color = "grey20") + coord_cartesian(c(3, 50))
> |
```

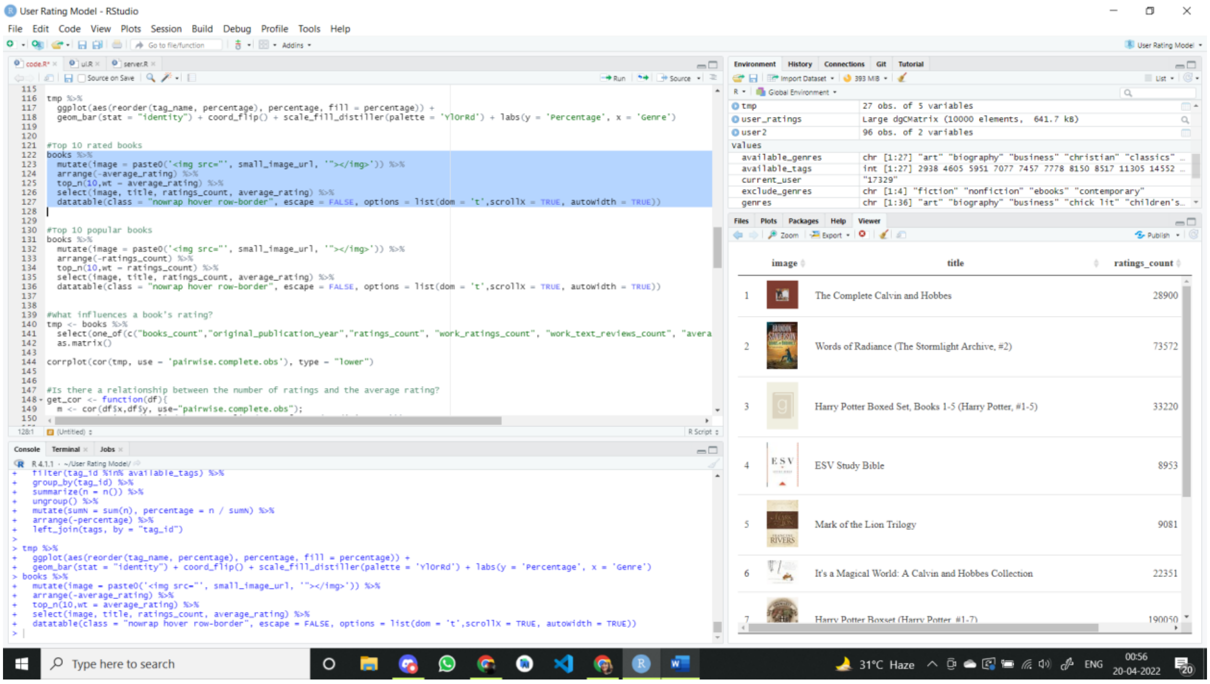
user_ratings	Large dgCMatrix (10000 elements, 641.7 kB)
user2	96 obs. of 2 variables
Values	
available_genres	chr [1:27] "art" "biography" "business" "christian" "classics" …
available_tags	int [1:27] 2938 4605 5951 7077 7457 7778 8150 8517 11305 14552 …
current_user	"17329"
exclude_genres	chr [1:4] "fiction" "nonfiction" "ebooks" "contemporary"
genres	chr [1:36] "art" "biography" "business" "chick lit" "children's…



```r
#Distribution of Genres
genres <- str_to_lower(c("Art", "Biography", "Business", "Chick Lit", "Children's", "Christian", "Classics",
                         "Comics", "Contemporary", "Cookbooks", "Crime", "Ebooks", "Fantasy", "Fiction",
                         "Gay and Lesbian", "Graphic Novels", "Historical Fiction", "History", "Horror",
                         "Humor and Comedy", "Manga", "Memoir", "Music", "Mystery", "Nonfiction", "Paranormal",
                         "Philosophy", "Poetry", "Psychology", "Religion", "Romance", "Science", "Science Fiction",
                         "Self Help", "Suspense", "Spirituality", "Sports", "Thriller", "Travel", "Young Adult"))

exclude_genres <- c("fiction", "nonfiction", "ebooks", "contemporary")
genres <- setdiff(genres, exclude_genres)

available_genres <- genres[str_to_lower(genres) %in% tags$tag_name]
available_tags <- tags$tag_id[match(available_genres, tags$tag_name)]

tmp <- book_tags %>%
  filter(tag_id %in% available_tags) %>%
  group_by(tag_id) %>%
  summarize(n = n()) %>%
  ungroup() %>%
  mutate(sumN = sum(n), percentage = n / sumN) %>%
  arrange(-percentage) %>%
  left_join(tags, by = "tag_id")

tmp %>%
  ggplot(aes(reorder(tag_name, percentage), percentage, fill = percentage)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette = 'YlOrRd') + labs(y = 'Percentage', x = 'Genre')


#Top 10 rated books
books %>%
  mutate(image = paste0('<img src="', small_image_url, '"></img>')) %>%
  arrange(-average_rating) %>%
  top_n(10,wt = average_rating) %>%
```

```
> genres <- setdiff(genres, exclude_genres)
>
> available_genres <- genres[str_to_lower(genres) %in% tags$tag_name]
> available_tags <- tags$tag_id[match(available_genres, tags$tag_name)]
>
> tmp <- book_tags %>%
+   filter(tag_id %in% available_tags) %>%
+   group_by(tag_id) %>%
+   summarize(n = n()) %>%
+   ungroup() %>%
+   mutate(sumN = sum(n), percentage = n / sumN) %>%
+   arrange(-percentage) %>%
+   left_join(tags, by = "tag_id")
>
> tmp %>%
+   ggplot(aes(reorder(tag_name, percentage), percentage, fill = percentage)) +
+   geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette = 'YlOrRd') + labs(y = 'Percentage', x = 'Genre')
> |
```

```
tmp                 27 obs. of 5 variables
user_ratings        Large dgCMatrix (10000 elements,  641.7 kB)
user2               96 obs. of 2 variables
values
  available_genres  chr [1:27] "art" "biography" "business" "christian" "classics" …
  available_tags    int [1:27] 2938 4605 5951 7077 7457 7778 8150 8517 11305 14552 …
  current_user      "17329"
  exclude_genres    chr [1:4] "fiction" "nonfiction" "ebooks" "contemporary"
  genres            chr [1:36] "art" "biography" "business" "chick lit" "children's…
```

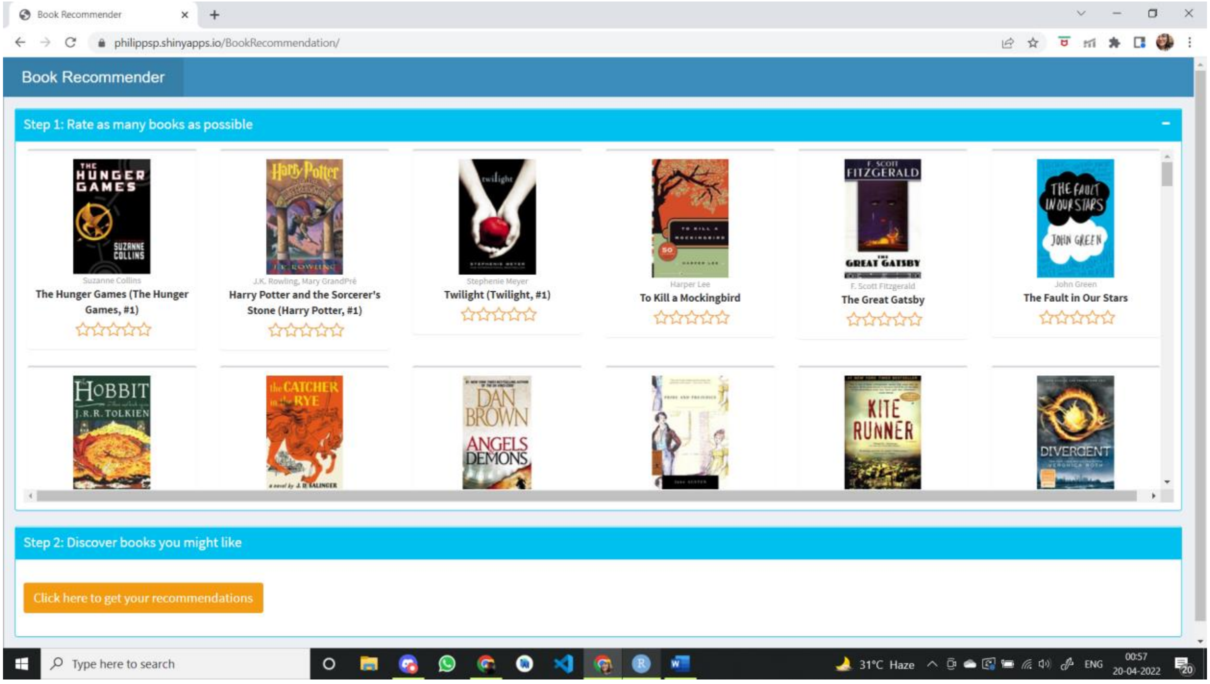


```r
tmp %>%
  ggplot(aes(reorder(tag_name, percentage), percentage, fill = percentage)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette = 'YlOrRd') + labs(y = 'Percentage', x = 'Genre')


#Top 10 rated books
books %>%
  mutate(image = paste0('<img src="', small_image_url, '"></img>')) %>%
  arrange(-average_rating) %>%
  top_n(10,wt = average_rating) %>%
  select(image, title, ratings_count, average_rating) %>%
  datatable(class = "nowrap hover row-border", escape = FALSE, options = list(dom = 't',scrollX = TRUE, autoWidth = TRUE))


#Top 10 popular books
books %>%
  mutate(image = paste0('<img src="', small_image_url, '"></img>')) %>%
  arrange(-ratings_count) %>%
  top_n(10,wt = ratings_count) %>%
  select(image, title, ratings_count, average_rating) %>%
  datatable(class = "nowrap hover row-border", escape = FALSE, options = list(dom = 't',scrollX = TRUE, autoWidth = TRUE))


#What influences a book's rating?
tmp <- books %>%
  select(one_of(c("books_count","original_publication_year","ratings_count", "work_ratings_count", "work_text_reviews_count", "avera
  as.matrix()

corrplot(cor(tmp, use = 'pairwise.complete.obs'), type = "lower")


#Is there a relationship between the number of ratings and the average rating?
get_cor <- function(df){
  m <- cor(df$x,df$y, use="pairwise.complete.obs");
```

```
+   filter(tag_id %in% available_tags) %>%
+   group_by(tag_id) %>%
+   summarize(n = n()) %>%
+   ungroup() %>%
+   mutate(sumN = sum(n), percentage = n / sumN) %>%
+   arrange(-percentage) %>%
+   left_join(tags, by = "tag_id")
>
> tmp %>%
+   ggplot(aes(reorder(tag_name, percentage), percentage, fill = percentage)) +
+   geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette = 'YlOrRd') + labs(y = 'Percentage', x = 'Genre')
> books %>%
+   mutate(image = paste0('<img src="', small_image_url, '"></img>')) %>%
+   arrange(-average_rating) %>%
+   top_n(10,wt = average_rating) %>%
+   select(image, title, ratings_count, average_rating) %>%
+   datatable(class = "nowrap hover row-border", escape = FALSE, options = list(dom = 't',scrollX = TRUE, autoWidth = TRUE))
> |
```

```
tmp                 27 obs. of 5 variables
user_ratings        Large dgCMatrix (10000 elements,  641.7 kB)
user2               96 obs. of 2 variables
values
  available_genres  chr [1:27] "art" "biography" "business" "christian" "classics" …
  available_tags    int [1:27] 2938 4605 5951 7077 7457 7778 8150 8517 11305 14552 …
  current_user      "17329"
  exclude_genres    chr [1:4] "fiction" "nonfiction" "ebooks" "contemporary"
  genres            chr [1:36] "art" "biography" "business" "chick lit" "children's…
```

	image	title	ratings_count
1		The Complete Calvin and Hobbes	28900
2		Words of Radiance (The Stormlight Archive, #2)	73572
3		Harry Potter Boxed Set, Books 1-5 (Harry Potter, #1-5)	33220
4		ESV Study Bible	8953
5		Mark of the Lion Trilogy	9081
6		It's a Magical World: A Calvin and Hobbes Collection	22351
7		Harry Potter Boxset (Harry Potter, #1-7)	190050



Book Recommender

Step 1: Rate as many books as possible

Suzanne Collins
The Hunger Games (The Hunger Games, #1)

J.K. Rowling, Mary GrandPré
Harry Potter and the Sorcerer's Stone (Harry Potter, #1)

Stephenie Meyer
Twilight (Twilight, #1)

Harper Lee
To Kill a Mockingbird

F. Scott Fitzgerald
The Great Gatsby

John Green
The Fault in Our Stars

Step 2: Discover books you might like

Click here to get your recommendations

6. LIMITATIONS OF THE PROJECT

The promising results of online survey users indicated several problems related to system.

One of them is that sometimes the same book appears in different categories which makes recommendations less useful.

This is because some of the genres are very closely related to each other and therefore the items of that genre can appear in both of genres.

The model can work more efficiently on datasets that have high variance as it results in more accurate predictions.

7. FUTURE WORK

The recommendation system proposed here takes the number of users who have rated the books into account, without factoring in the absolute rating.

Due to this, a recommendation might arise from a book that a user has given low rating to, in which case a book might be recommended from a genre that the user dislikes.

This recommendation system relies on the ratings given by users. So, trust is a major issue, like whether the feedback and rating given by the user is genuine or not.

This recommendation system does not solve the trust issue.

Therefore, future research should focus on resolving both these issues.

8. CONCLUSION

In this paper, we present a recommendation system that is based on collaborative filtering method. The main goal of the system was to create a such a system, which can give qualitative recommendations to their users without any need to be registered on a certain platform. Experimental results show that the proposed systems method provides relevant recommendations.

This system can be applied for other domains to suggest items like movies, music and other products.

9. REFERENCE

1) Okon, E.U., Eke, B.O. and Asagba, P.O. (2018). An improved online book recommender system using collaborative filtering algorithm. International Journal of Computer Applications(0975- 8887) Volume 179-No.46, June 2018.
2) Kurmashov, N., Konstantin, L., Nussipbekov, A. (2015). Online book recommendation System. Proceedings of Twelve International Conference on Electronics Computer and Computation (ICECC)
3) Mathew, P., Kuriakose, B. And Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. Proceedings of International Conference on Data Mining and Advanced Computing (SAPIENCE)
4) Ayub, M., Ghazanfar, M.A., Maqsood, M. and Saleem, A. (2018). A Jaccard base similarity measure to improve performance of CF based recommendation system. Proceedings of International Conference on Information Networking (ICOIN)