Book Recommendation System

Eda Zhang, Radhika Kulkarni,Daye Kang, Eunhee Sung

Contents

1

Project purpose and problem

Project Description

- **Objective:** To recommend and discover new and interesting books based on users' tastes.
- **Purpose:** All people have different reading taste, so it is hard to get recommendation from others. The algorithm recommends books to users based on books they've read and their favorite genres.
- **Target Users:** Any people who would like to read a new book, especially for fantasy lover, children and their parents.
- **Possible Problem**: the review rating can be biased because people tend to rate when they really satisfied or when they disappointed.

2

Data sources and associated assumptions

Data Set | 1. Genre Data set

The datasets were collected in late 2017 from goodreads.com with updated several files in May 2019. Among 8 genres *Children's book and Fantasy* are selected for the app demo.

Contains detailed information about book id, authors, average rating of books, number of pages, and title of books.

dataset.csv

Genre : Children book + Fantasy
1,242 rows | 1-1 of 26 columns

	book_id	authors	average_rating	goodreads_book_id	country_code	description
1	1	[{'author_id': '3041852', 'role': ''}]	4.13	287141	US	Relates in vigorous prose the tale of Aeneas, the lege...
2	2	[{'author_id': '19158', 'role': ''}]	4.22	6066812	US	To Kara's astonishment, she discovers that a portal h...
3	3	[{'author_id': '5411', 'role': ''}]	4.43	89378	US	In Newbery Medalist Cynthia Rylant's classic bestselle...
4	4	[{'author_id': '169159', 'role': ''}]	3.57	1698376	US	WHAT DO YOU DO? A hen lays eggs... A cow gives mil...
5	5	[{'author_id': '1540277', 'role': ''}, {'author_id': '52621...	3.89	3631900	US	When Amadi disobeys his mother and runs off to the ...
6	6	[{'author_id': '2760047', 'role': ''}, {'author_id': '45890...	3.53	8030991	US	Loin tusaisen katseen pimeyteen. – Pidetaan toisiam...
7	7	[{'author_id': '7323', 'role': ''}, {'author_id': '3039530', '...	3.95	16250767	US	Illustrator Charles Reasonor's charming version of the...
8	8	[{'author_id': '7323', 'role': ''}, {'author_id': '3039530', '...	3.14	16250765	US	Illustrator Charles Reasonor's charming version of the...

Data Set | 2. Rating Data set

- This dataset is contributed by philip on Kaggle for collaborative filtering.
- Combining 3 datasets and created new rating data set.

ratings.csv

194,941 rows

	A	B	C
1	book_id	user_id	rating
2	1	314	5
3	1	439	3
4	1	588	5
5	1	1169	4
6	1	1185	4
7	1	2077	4
8	1	2487	4
9	1	2900	5
10	1	3662	4
11	1	3922	5

book_tags.csv

999,912 rows

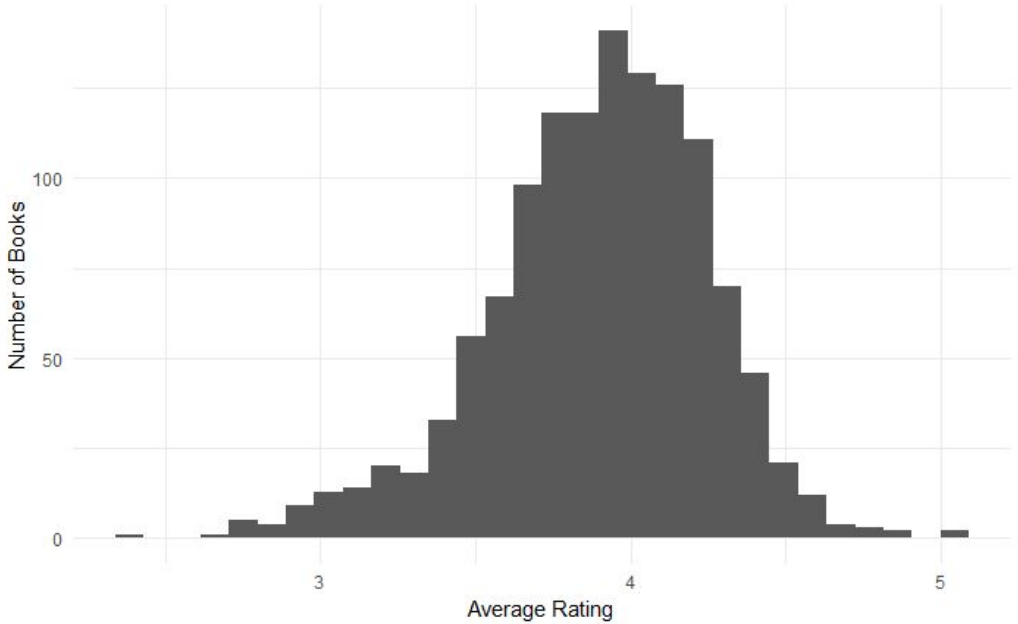
	A	B	C
1	goodreads_book_id	tag_id	count
2	1	30574	167697
3	1	11305	37174
4	1	11557	34173
5	1	8717	12986
6	1	33114	12716
7	1	11743	9954
8	1	14017	7169
9	1	5207	6221
10	1	22743	4974
11	1	32989	4364
12	1	27199	3857
13	1	11590	3456

tags.csv

34,252 rows

	A	B
1	tag_id	tag_name
2	0	-
3	1	--1-
4	2	--10-
5	3	--12-
6	4	--122-
7	5	--166-
8	6	--17-
9	7	--19-
10	8	--2-

Distribution Plot of Average Rating



To check if the dataset is biased with average rating, distribution plot is provided.

- The data set is approximately normally distributed with average of 3.90, so the average rating is not biased.

Exploratory Data Analysis | Logistic Regression

Conducted logistic regression to see if there is any relationship between dependent variable, genre and independent variables(average rating, number of pages, publication year, rating count, text reviews count, and title of length).

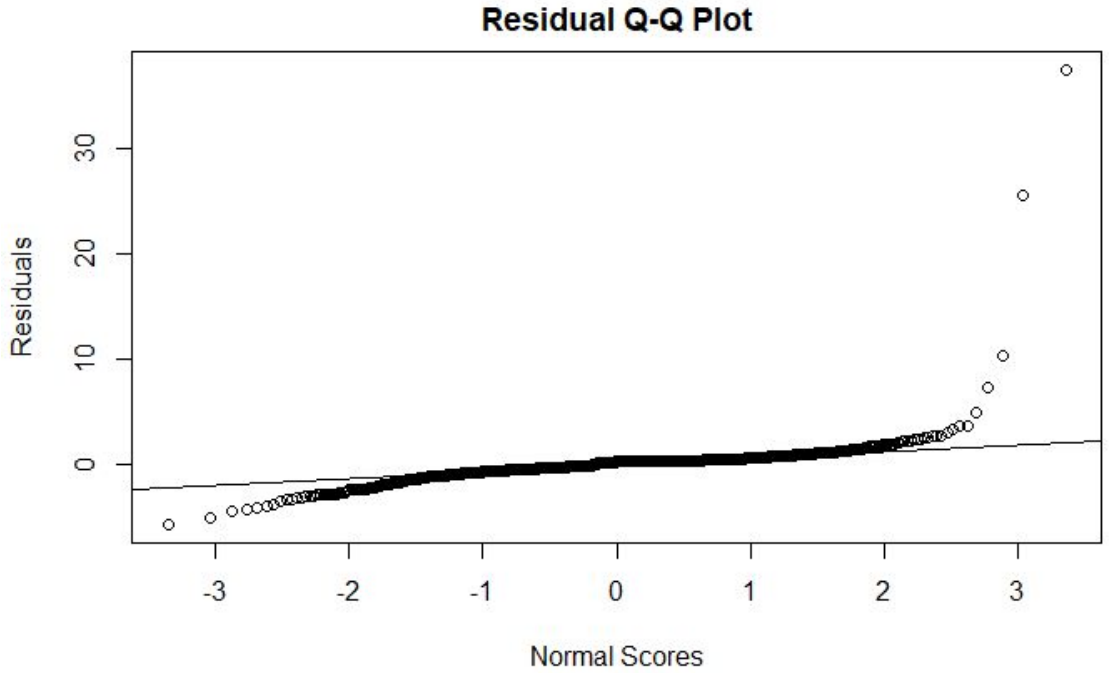
```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6416  -0.5981   0.3565   0.5727   3.8068

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                1.018e+00  2.020e+00   0.504    0.614
books_rd$average_rating    2.654e-01  2.262e-01   1.173    0.241
books_rd$num_pages        -1.294e-02  7.213e-04 -17.935  < 2e-16 ***
books_rd$publication_year -1.061e-04  9.086e-04  -0.117    0.907
books_rd$ratings_count    -7.372e-05  5.832e-05  -1.264    0.206
books_rd$text_reviews_count 1.240e-03  1.071e-03   1.157    0.247
books_rd$title_length      2.563e-01  4.708e-02   5.444  5.2e-08 ***
books_rd$dur                      NA         NA      NA       NA
```

.

Since the z-value of the number of pages and title length are very small, the two variables are significant.

Normal Probability Plot of Residuals



The plot clearly shows that there is a linear relationship among the variables and genres.

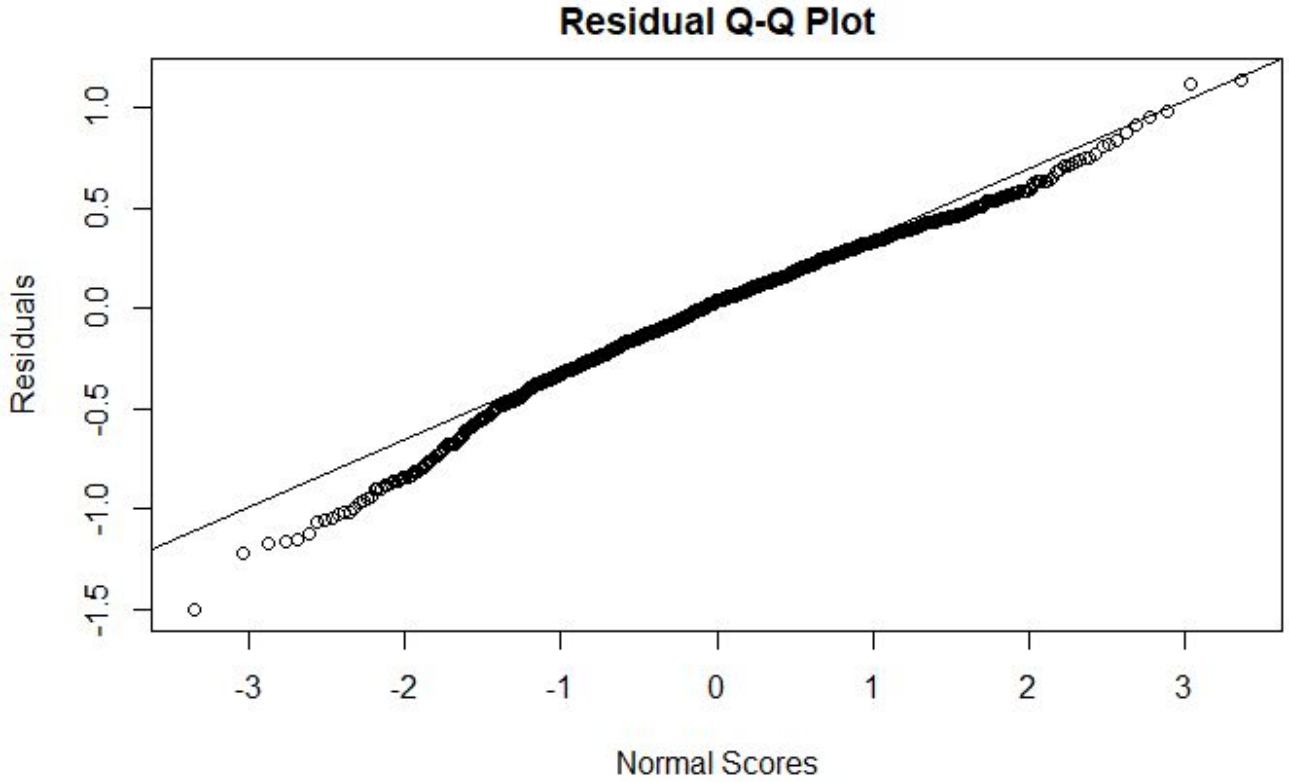
Exploratory Data Analysis | Linear Regression

Since the dataset is used for recommendation model, conducted linear regression to see if there is any relationship between average rating and other variables(genres, number of pages, publication year, rating count, text reviews count, and title of length).

```
Coefficients: (1 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        4.134e+00  2.663e-01  15.525  < 2e-16 ***
book_id           -5.136e-05  5.461e-05  -0.941    0.347
num_pages          2.945e-04  6.738e-05   4.370 1.35e-05 ***
publication_year  -1.339e-04  1.281e-04  -1.045    0.296
ratings_count      8.746e-06  6.775e-06   1.291    0.197
text_reviews_count -1.286e-04  1.211e-04  -1.062    0.289
genres            -2.040e-02  4.240e-02  -0.481    0.631
title_length       6.073e-03  5.829e-03   1.042    0.298
```

Since the p-value of the number of pages is very small, there are linear relationship between average rating and number of pages.

Normal Probability of Residuals



Since the p-value of the number of pages and title length are very small, the two variables are significant.

3

Data analysis and results

Data Analysis Method

1. For the recommendation: **Cluster Analysis & Collaborative Filtering**
2. Contents description: **Text Analysis**

Cluster Analysis on data

The data was clustered based mainly on the following parameters (using 'clValid' and 'kmeans' function in R):

1. Average Rating of books
2. Genre
3. Format (e-book, hardcover, paperback etc.)
4. Length of books

The K-means clustering method was chosen over hierarchical clustering since:

1. Size of the dataset is large (i.e. number of variables)
2. It allows more flexibility

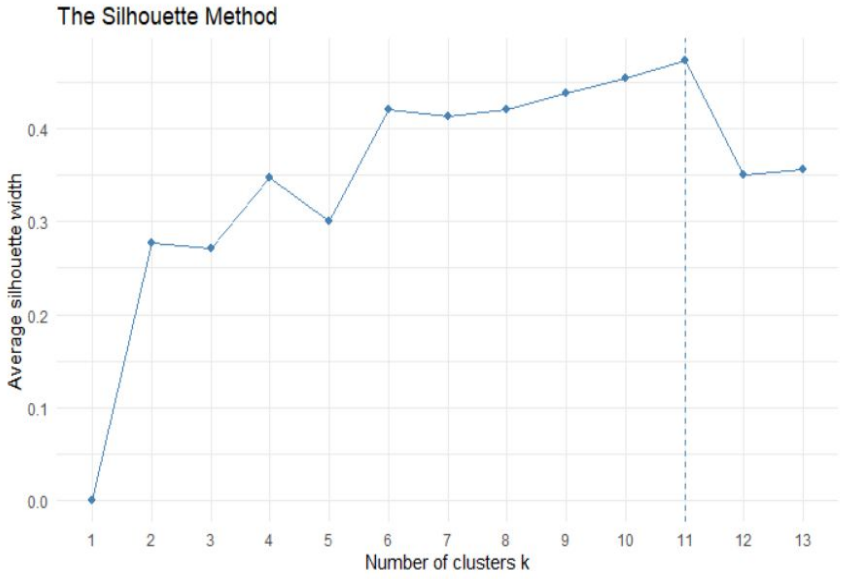
average_rating	num_pages	format.Hardcover	format.Mass Market Paperback	format.Paperback
4.55	662	1	0	0
4.29	40	1	0	0
3.91	360	0	1	0
4.13	288	0	0	1
4.35	40	1	0	0
4.09	471	0	1	0
4.04	404	1	0	0

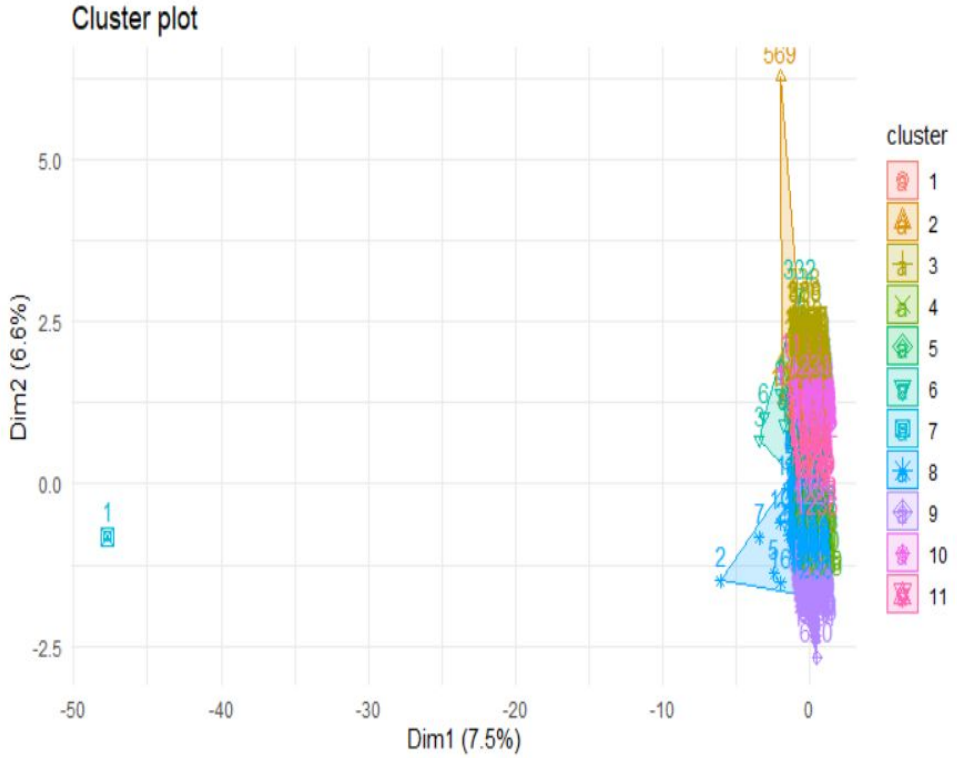
Preprocessing of data:

1. Binary encoding for categorical variables
2. Scaling (Standardization)
3. Removal of columns with NA variables (i.e. format.NA etc.)

Cluster Analysis on data

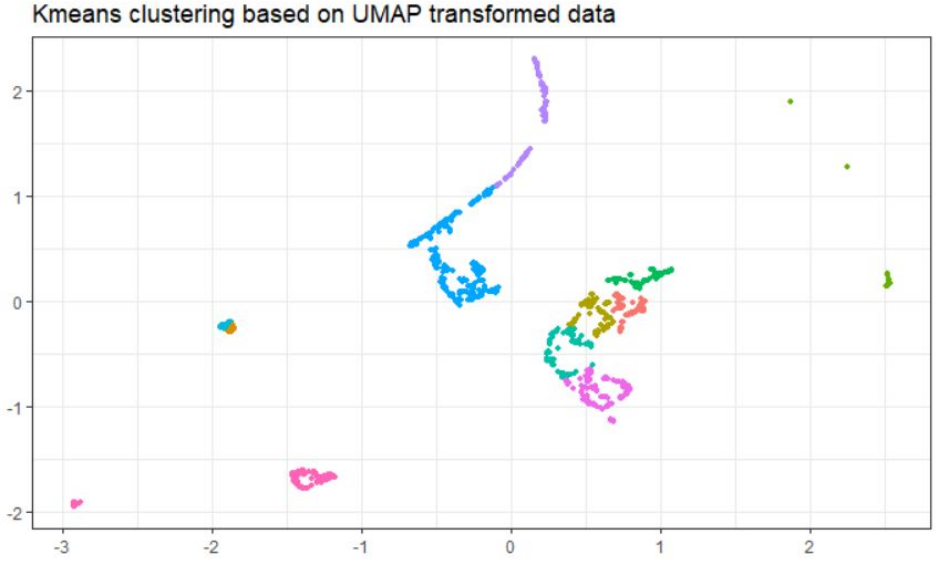
	Score <dbl>	Method <fctr>	Clusters <fctr>
Connectivity	0.0000	agnes	2
Dunn	0.2060	agnes	4
Silhouette	0.5525	kmeans	11

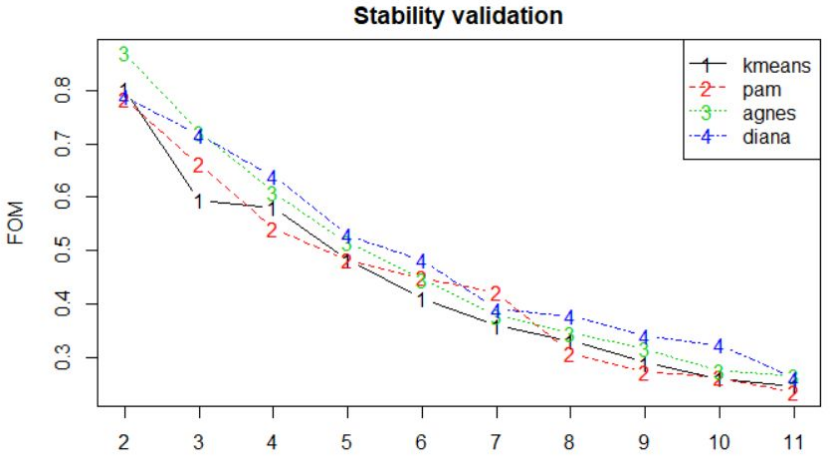


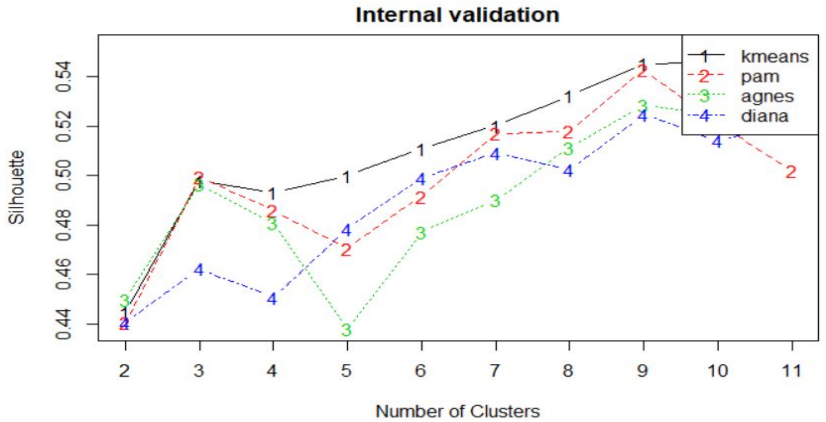


UMAP Analysis

- ‘umap’ function was used to reduce the data to two dimensions
- Cleaner, tighter clusters
- Some degree of overlap present
- Validation measures included silhouette and stability parameters







‘Kmeans’ Clustering Analysis

The ‘kmeans’ function inbuilt in R was also used for clustering the data points since it allows for more interpretability and is less complex.

```
fit.km<-kmeans(ratingslistdf_x, 11, nstart=25)
```

	title_without_series <chr>	genres <chr>	format <chr>	km_cluster <int>
1	The Name of the Wind (The Kingkiller Chronicle, #1)	fantasy	Hardcover	1
2	Don't Let the Pigeon Drive the Bus!	children	Hardcover	8
3	A Quick Bite (Argeneau #1)	fantasy	Mass Market Paperback	10
4	A Christmas Carol and Other Christmas Writings	fantasy	Paperback	6
5	Don't Let the Pigeon Stay Up Late!	children	Hardcover	8
6	Green Rider (Green Rider, #1)	fantasy	Mass Market Paperback	10

A deeper analysis of the components in every cluster reveals different characteristics of every cluster. For example, Cluster 2 had books which belonged to the ‘Children’s books’ genre and ‘Board Book’ format.

Text Analysis | TF-IDF Word Cloud

- Data: Book Description
- Feature:
  - Capture *some* info
  - Fast process
- Challenges:
  - Different languages
  - Small data set

Why Collaborative Filtering?

Collaborative filtering is a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users . The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.

Collaborative Filtering

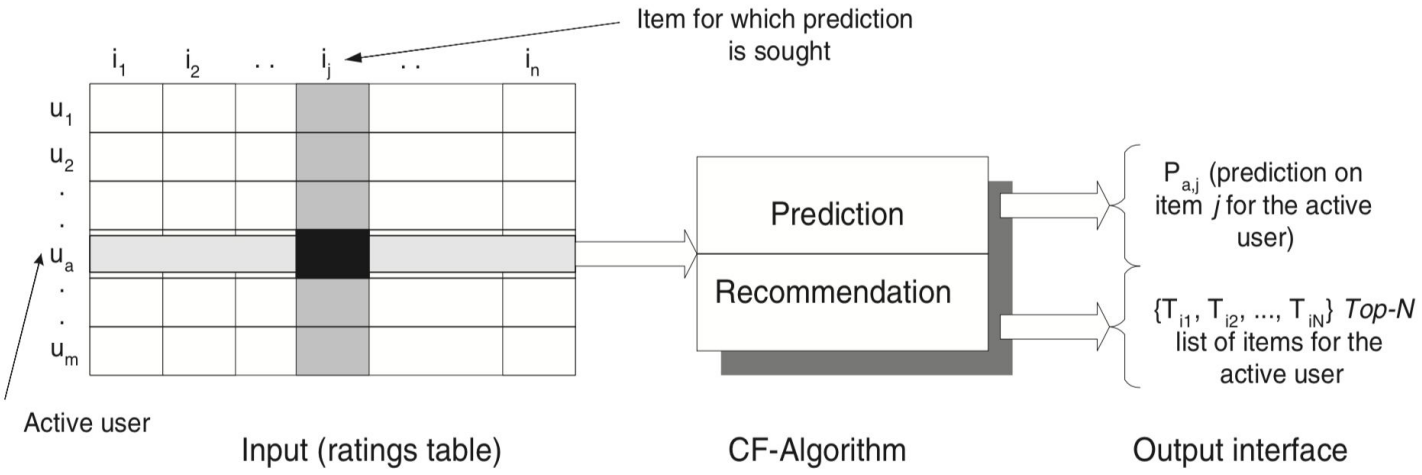


Figure 1: The Collaborative Filtering Process.

Item-Based Collaborative Filtering Recommendation Algorithms
http://files.grouplens.org/papers/www10_sarwar.pdf

Collaborative Filtering UBCF/IBCF

User-based collaborative filtering

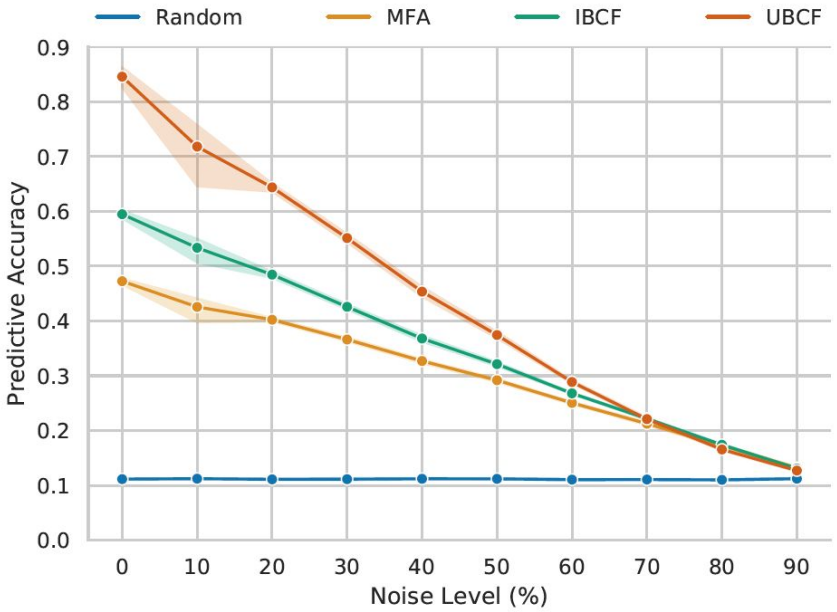
Memory-based Collaborative Filtering Algorithms. Memory-based algorithms utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as *neighbors*, that have a history of agreeing with the target user (i.e., they either rate different items similarly or they tend to buy similar set of items). Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or *top-N* recommendation for the active user. The techniques, also known as *nearest-neighbor* or user-based collaborative filtering, are more popular and widely used in practice.

THIS

Item-based collaborative filtering

Model-based Collaborative Filtering Algorithms. Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. The model building process is performed by different *machine learning* algorithms such as **Bayesian network, clustering,** and **rule-based** approaches. The Bayesian network model [6] formulates a probabilistic model for collaborative filtering problem. Clustering model treats collaborative filtering as a classification problem [2, 6, 29] and works by clustering similar users in same class and estimating the probability that a particular user is in a particular class C, and from there computes the conditional probability of ratings. The rule-based approach applies association rule discovery algorithms to find association between co-purchased items and then generates item recommenda-

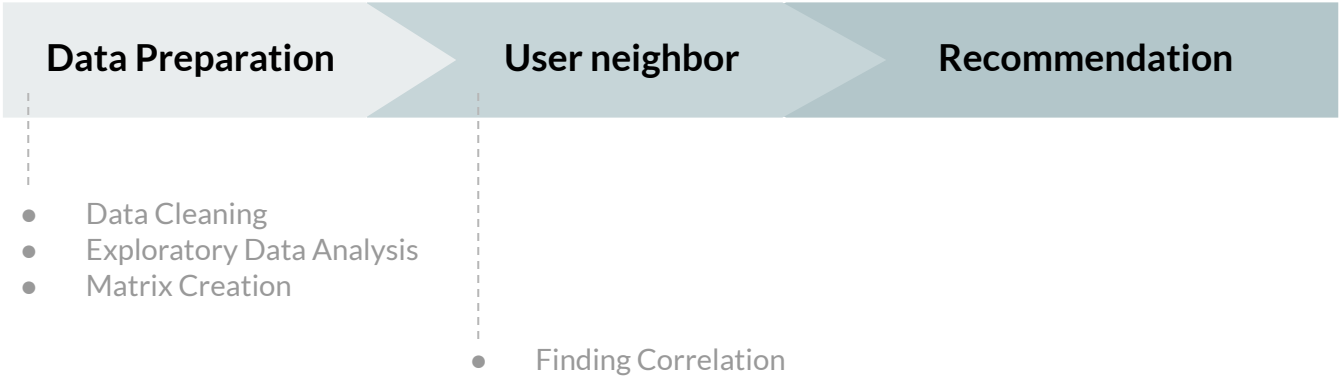
Collaborative Filtering UBCF/IBCF



Riesterer, N., Brand, D., & Ragni, M. (2020). Uncovering the Data-Related Limits of Human Reasoning Research: An Analysis based on Recommender Systems. *arXiv preprint arXiv:2003.05196*.

Item-Based Collaborative Filtering Recommendation Algorithms
http://files.grouplens.org/papers/www10_sarwar.pdf

UBCF | Process



https://www.kaggle.com/philippsp/book-recommender-collaborative-filtering-shiny/code

Data Cleaning

1. Remove the duplicate ratings

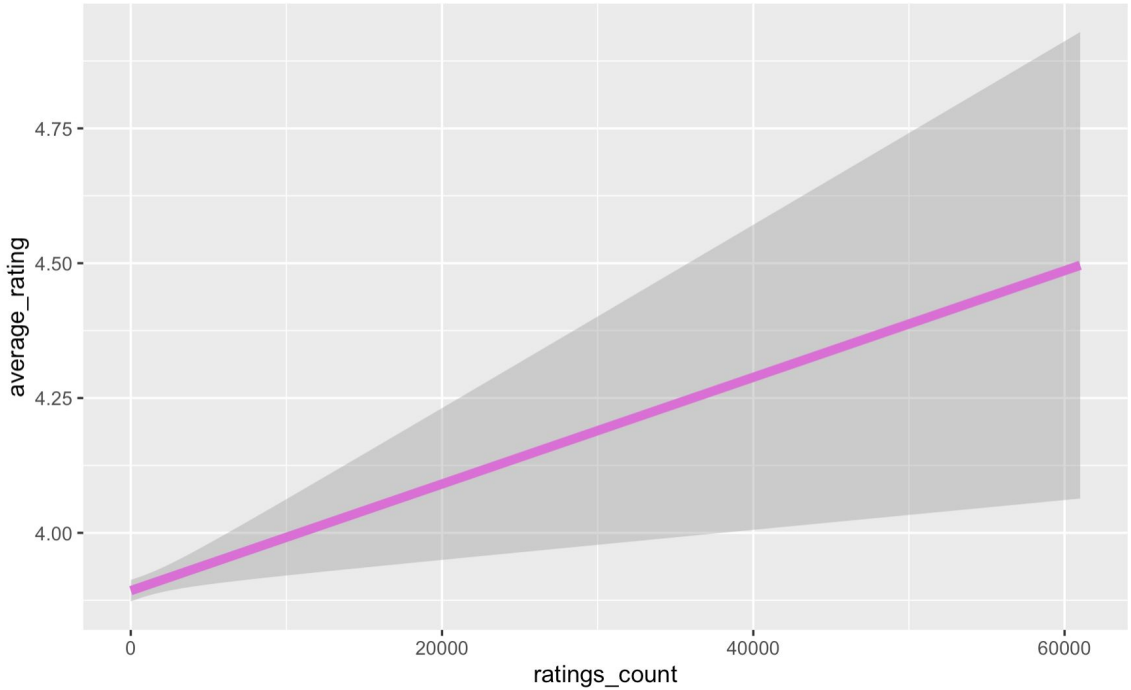
2. Remove users who rated fewer than 3 books.

3. To reduce calculation times, selected only a subset of users. (20%)

Exploratory Data Analysis

Number of ratings and the average rating

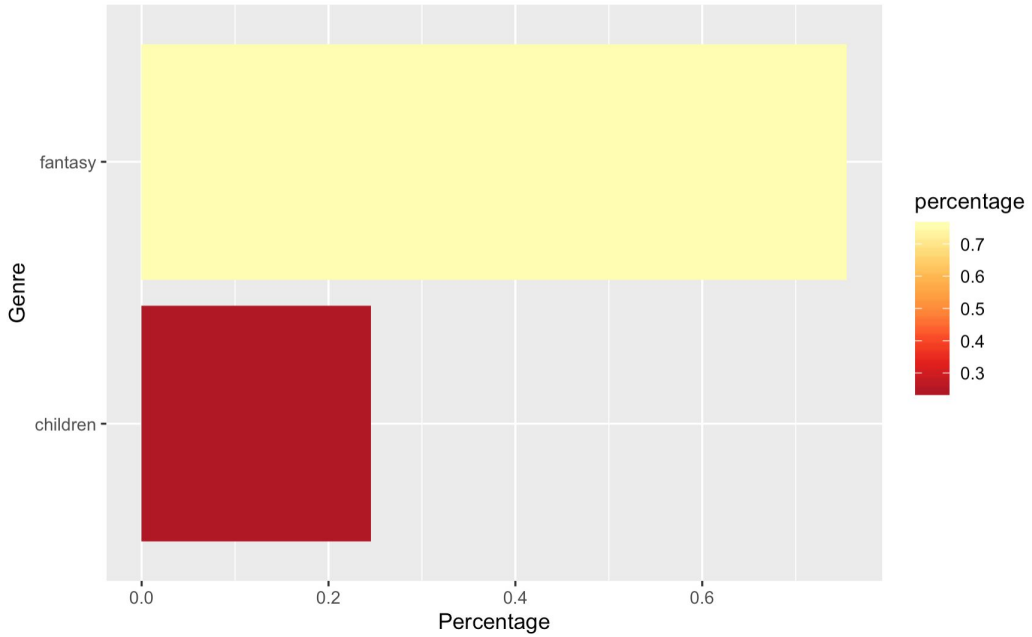
Frequent raters rate books differently from less frequent raters



Exploratory Data Analysis

Percentage of Genre

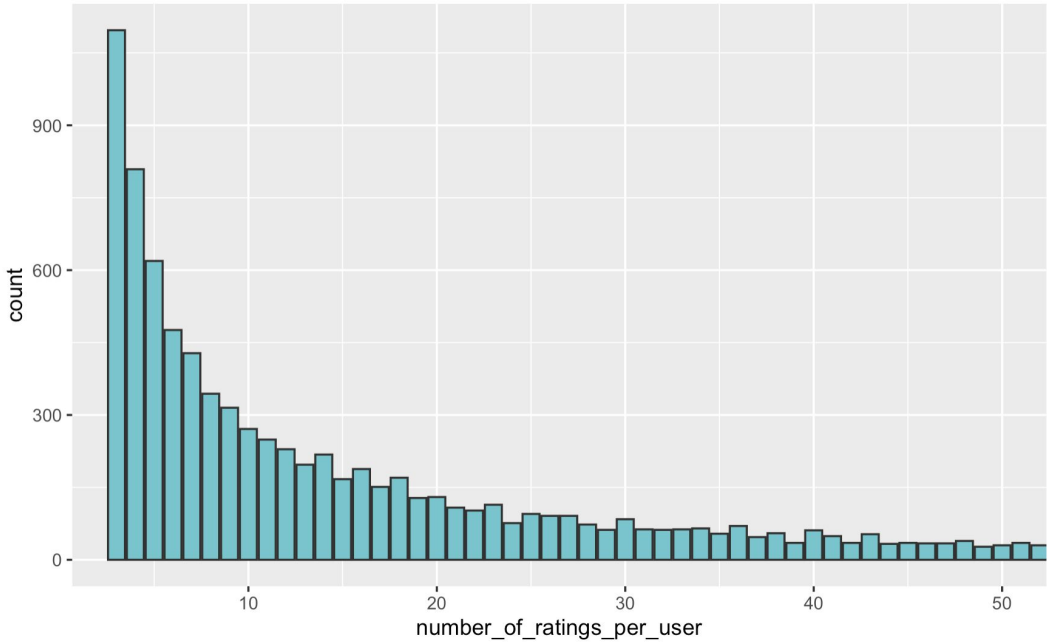
After the data cleaning fantasy genre show high percentage.



Exploratory Data Analysis

Number of ratings per users

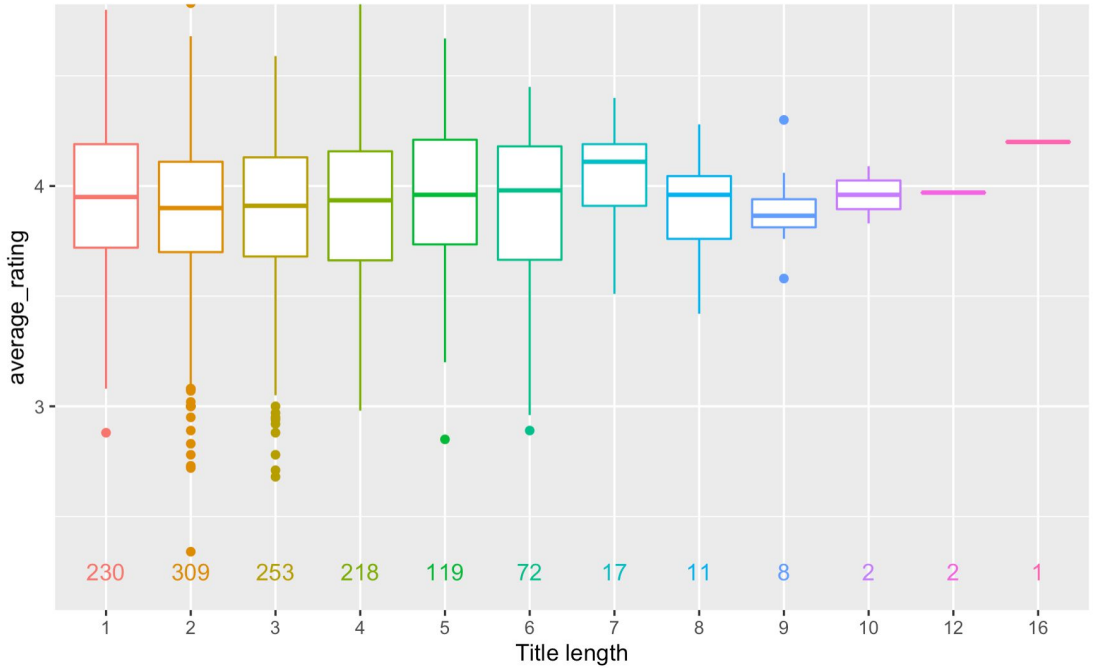
There are some users with many ratings.



Exploratory Data Analysis

How long should the title should be?

Titles with 5 or 7 words seem to have slightly higher ratings.

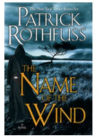


Exploratory Data Analysis

Top 10 rated books

image	title	ratings_count	average_rating
	Myrtle the Purple Turtle	13	5
	Witch Wars (The Underlayes, #1)	11	5
	Adventures Around the Globe: Packed Full of Maps, Activities and Over 250 Stickers	6	4.83

Top 10 popular books

	image	title	ratings_count	average_rating
1		The Name of the Wind (The Kingkiller Chronicle, #1)	416634	4.55
2		Don't Let the Pigeon Drive the Bus!	61003	4.29
3		A Quick Bite (Argeneau #1)	32140	3.91

Matrix Creation

```
        book_id
user_id  1  2  3  4  5
     18 NA NA NA NA NA
     19 NA NA NA NA NA
     22 NA NA NA NA NA
     31 NA NA NA NA NA
     51 NA NA NA NA NA
[1]  9003 10000
```

```{r}
dimension_names <- list(user_id = sort(unique(ratings$user_id)), book_id = sort(unique(ratings$book_id)))

ratingmat <- spread(select(ratings, book_id, user_id, rating), book_id, rating) %>% select(-user_id)

ratingmat <- as.matrix(ratingmat)

dimnames(ratingmat) <- dimension_names
ratingmat[1:5, 1:5]
dim(ratingmat)
```

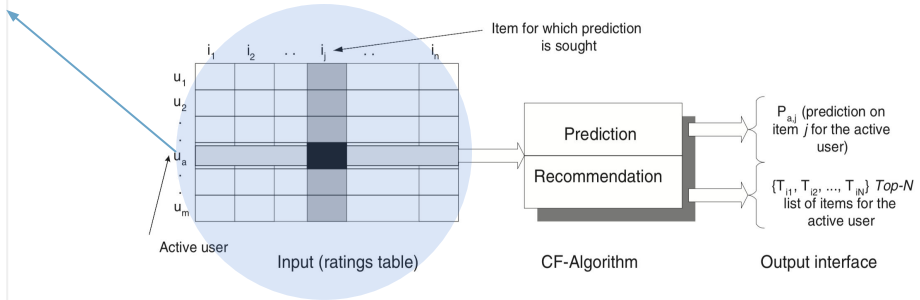


Figure 1: The Collaborative Filtering Process.

Find Similar Users

Current user id : 794

```
 [1] "173"   "794"   "937"   "1051"  "1660"  "1813"  "2095"  "2140"  "3290"  "4221"  "4691"  "4813"  "5410"  "5792"  "6387"  "6439"
[17] "6626"  "6690"  "6909"  "7325"  "7624"  "7635"  "7743"  "7762"  "7786"  "8159"  "8314"  "8882"  "8896"  "9578"  "10724" "10961"
[33] "11675" "12707" "12956" "13041" "13544" "14153" "14266" "14475"
```

Normalization

To reduce the influence of interindividual differences in mean ratings we normalized the user's ratings by subtracting the users mean from all individual ratings.

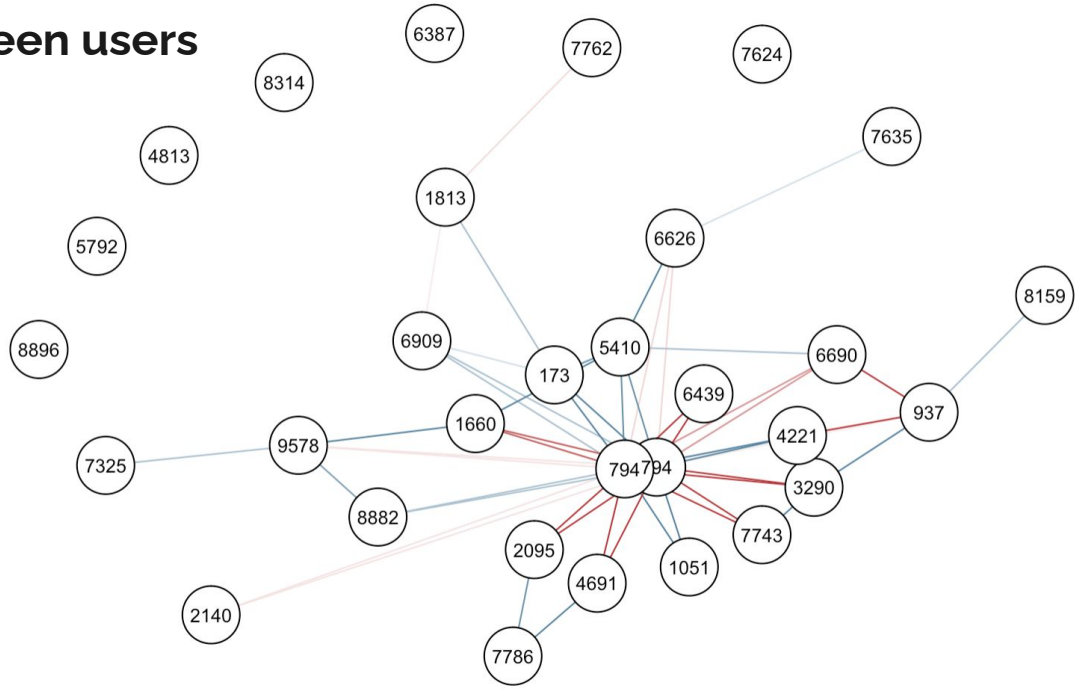
Sorting

We can calculate the similarity of all others users with the current user and sort them according to the highest similarity.

```
the standard deviation is zero        173        1051        4221       12956       14266       14475       16088       16546
17024       19731       20521
 1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000
1.00000000
      20814       21469       33450       33863       36535       42933       43137       49354       51814       16363
29882
 1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  1.00000000  0.94491118
0.91287093
      48299        5410       50273       37985       20392       26287        6909        8882       31018       50745
13544
 0.87038828  0.86602540  0.86602540  0.75925660  0.69994239  0.61237244  0.50000000  0.50000000  0.50000000  0.36514837
0.29880715
      44723       35724       21528       19930       24670       41861       38082
 0.13934660  0.13483997  0.13363062  0.13245324  0.00000000  0.00000000 -0.07559289
```

Visualizing similarities between users

Current user id : 794
With random 30 users




```
sim_mat <- cor(t(rmat), use = 'pairwise.complete.obs')
random_users <- selected_users[1:30]
qgraph(sim_mat[c(current_user, random_users), c(current_user, random_users)], layout = "spring", vsize = 5, theme =
"TeamFortress", labels = c(current_user, random_users))
```

Get predictions for other books

	item	mean_rating
1	1007	-1.35344827586207
2	106	-1.35344827586207
3	108	-0.353448275862069
4	112	1.64655172413793
5	1136	-0.353448275862069
6	1195	-0.353448275862069
7	124	-2.35344827586207
8	127	0.646551724137931
9	1281	-0.353448275862069
10	1385	0.646551724137931

Recommend the best 5 predictions

```
book5<-predictions[1:1200,] %>%
  arrange(-mean_rating) %>%
  top_n(5, wt = mean_rating)  %>%
  mutate(book_id = as.numeric(as.character(item))) %>%
  left_join(select(books2,  title,url, book_id), by = "book_id")

books6<-na.omit(book5, cols="title")
```

item <fctr>	mean_rating <dbl>	book_id <dbl>	title <chr>
112	1.646552	112	Πήτερ Πάν (Τα βιβλία που αγαπήσαμε, #3)
146	1.646552	146	Binny Bewitched
174	1.646552	174	Spirit of the Titanic
199	1.646552	199	Welcome to HorrorLand: A Survival Guide
258	1.646552	258	I Want to Grow
320	1.646552	320	Storms
329	1.646552	329	The Little Kitten
6	1.646552	6	Katso eteesi, Lotta!
797	1.646552	797	The Skull Collector
92	1.646552	92	The Ersatz Elevator (A Series of Unfortunate Events, #6)

Using recommenderlab

	user	rating	book_id	title
1	794	3.96719348659004	6	Katso eteesi, Lotta!
2	794	3.96719348659004	92	The Ersatz Elevator (A Series of Unfortunate Events, #
3	794	3.96719348659004	112	Πήτερ Πάν (Τα βιβλία που αγαπήσαμε, #3)
4	794	3.96719348659004	146	Binny Bewitched
5	794	3.96719348659004	174	Spirit of the Titanic

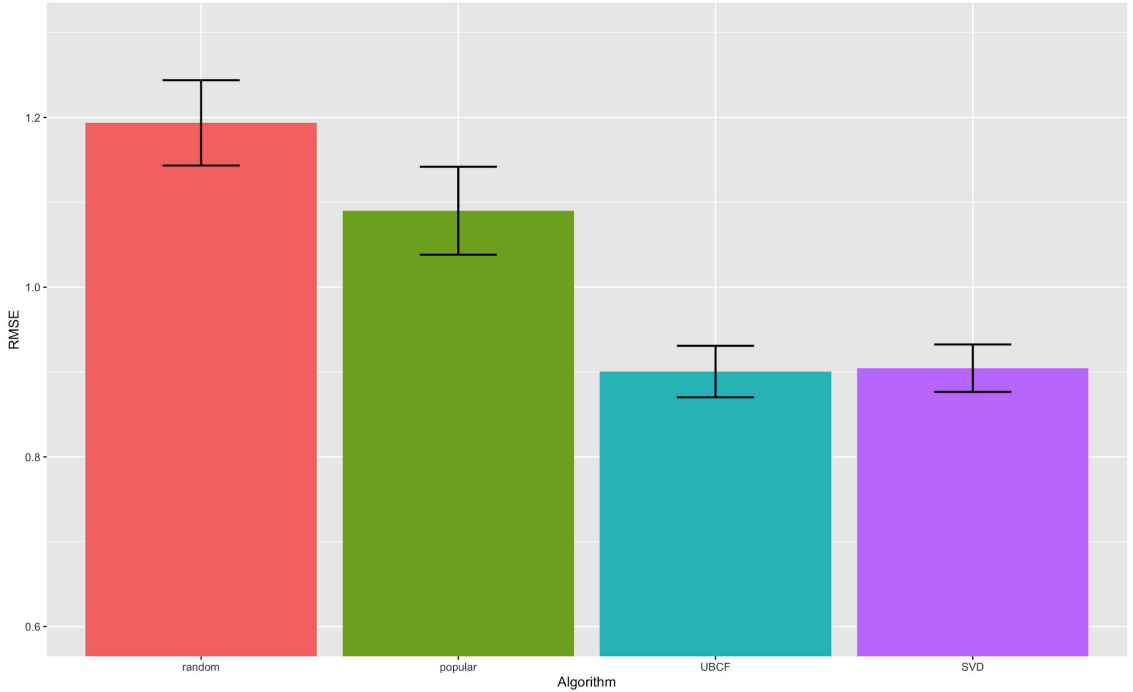
Performance of the algorithm

Comparing with other algorithms

Using Recommenderlab Library

-Can get similar recommendation results

-Can compare with other algorithms



Tradeoff of interpretability and model performance

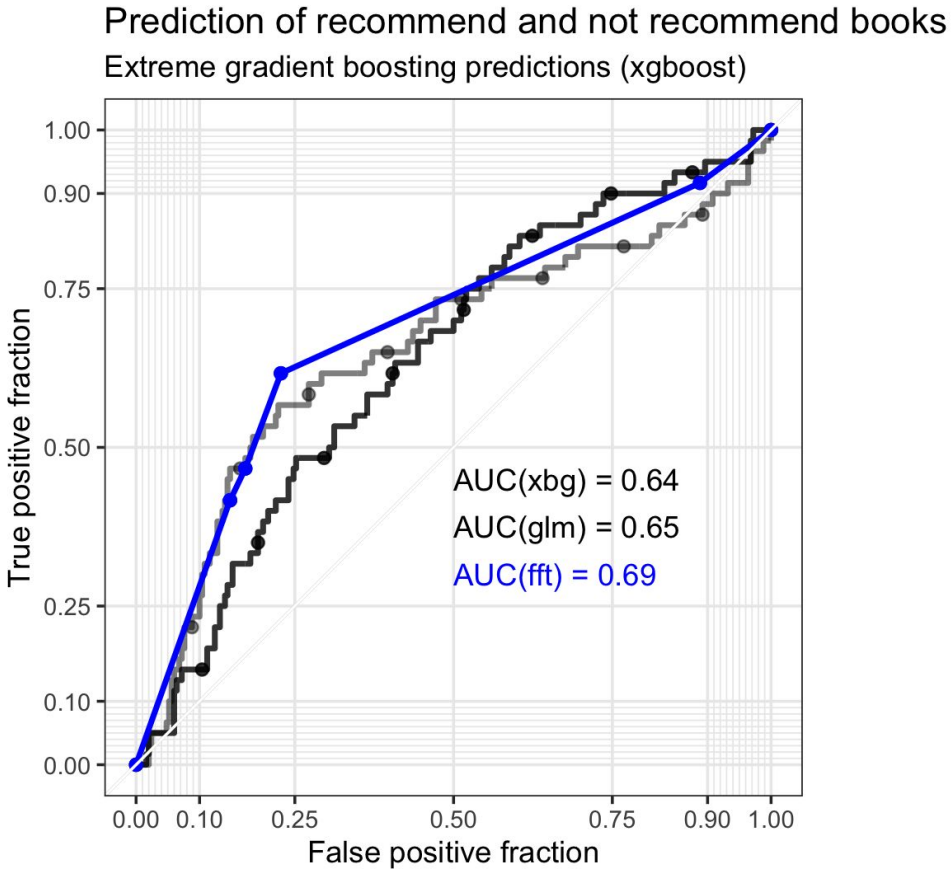
4

Model interpretation and explanation

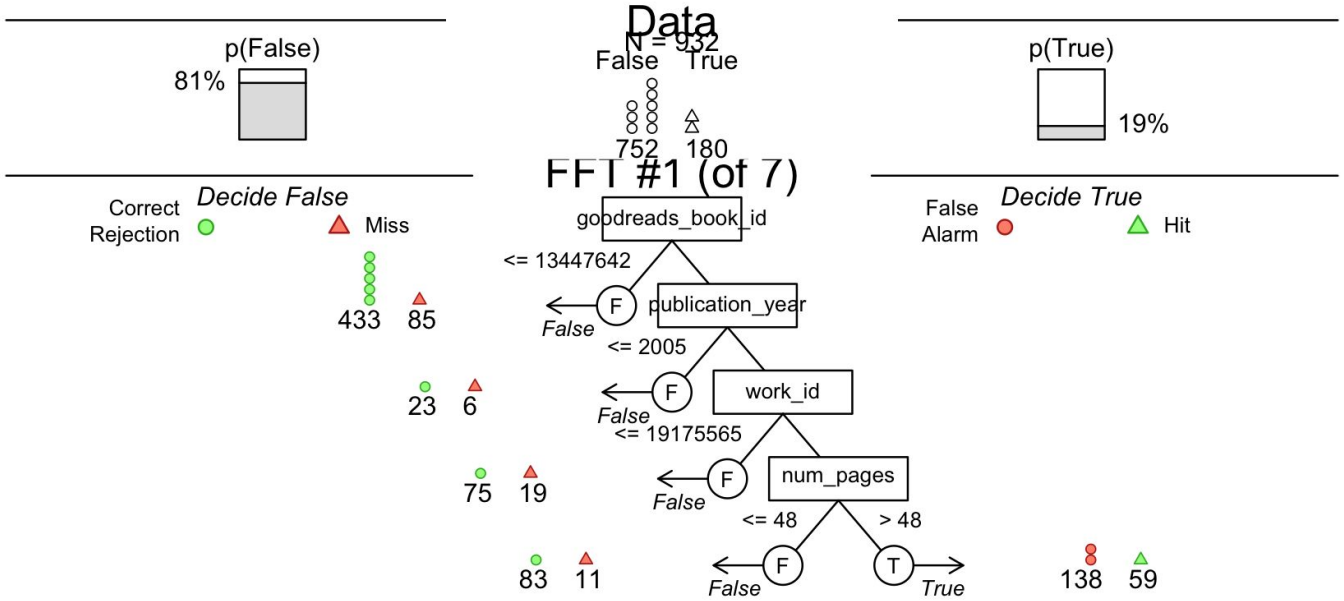
Supervised

Classifications

Classification Models Performance

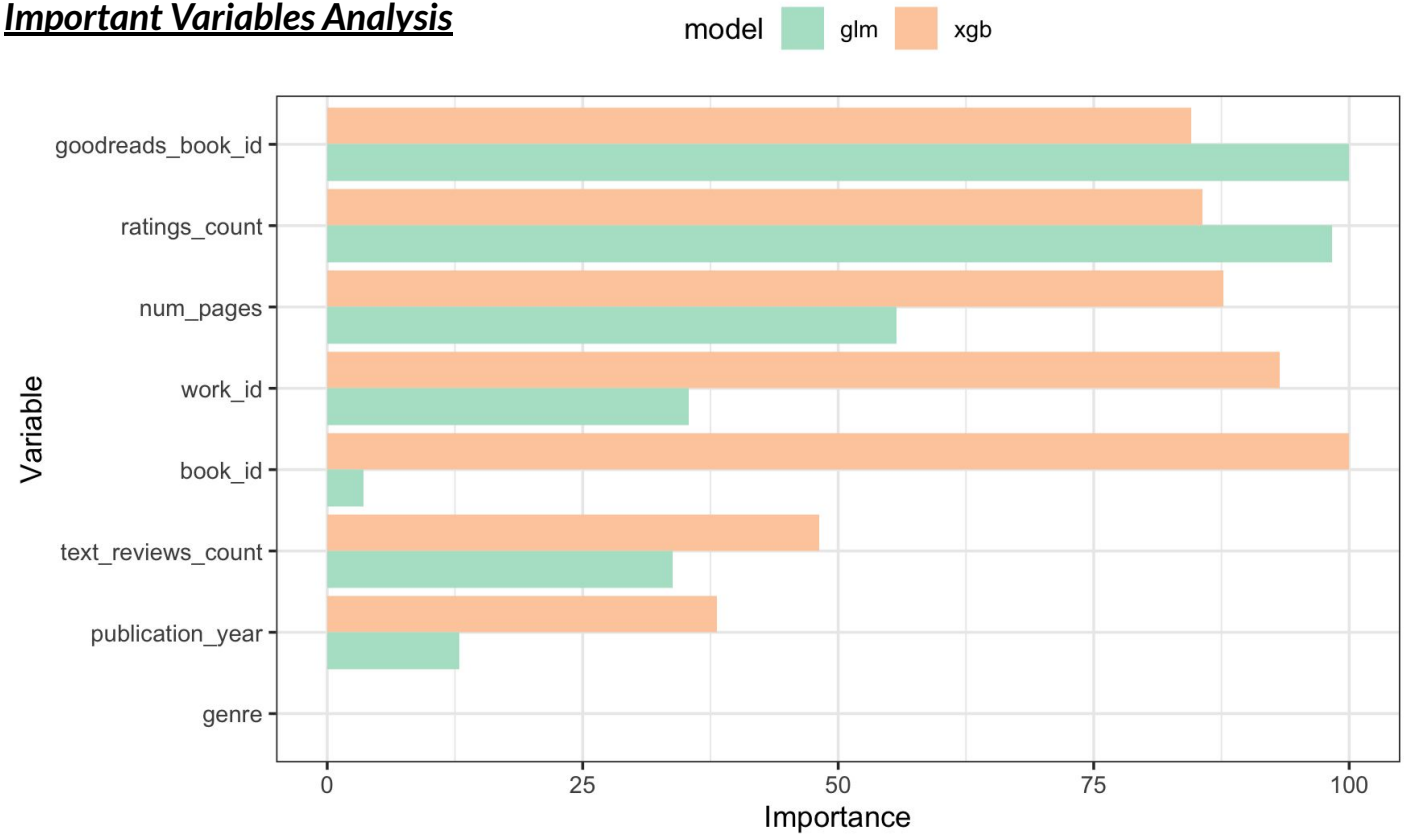


Classification>> Linear Model>> FFTree



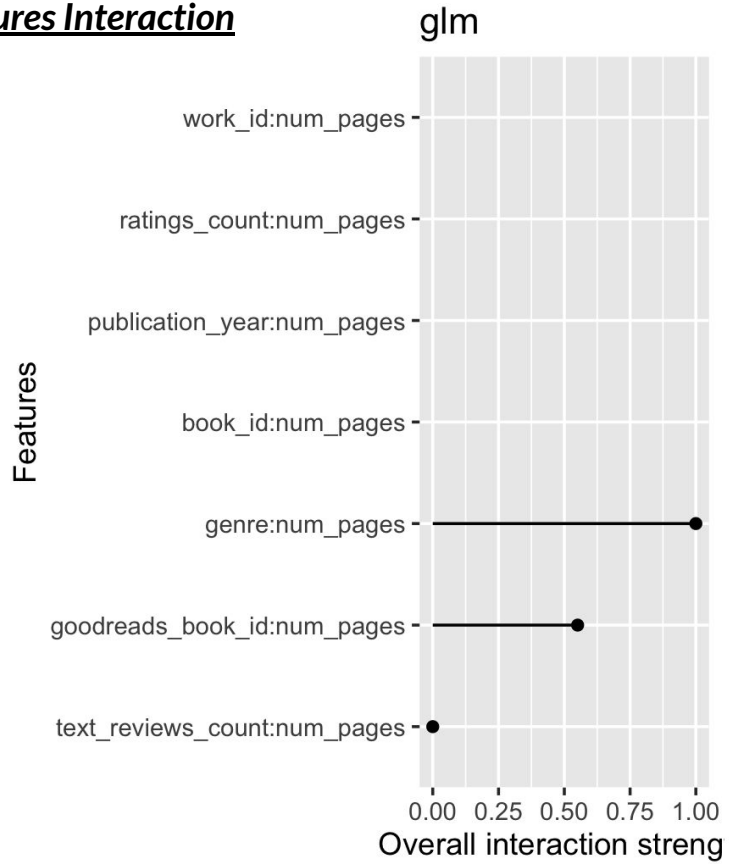
Classification>>Linear Vs. Non-Linear: glm vs. xgb

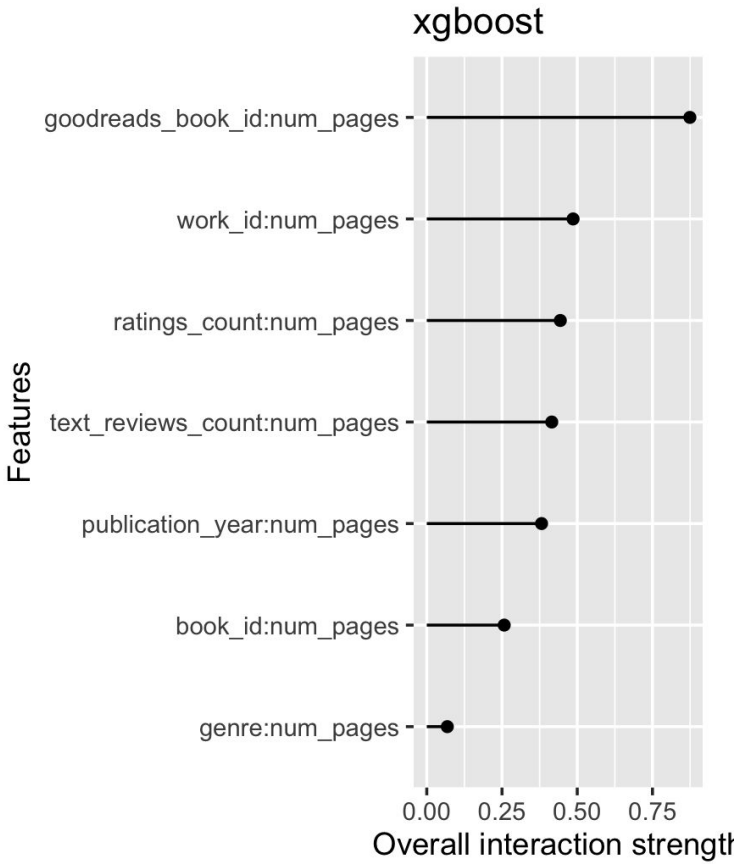
Important Variables Analysis



Classification>>Linear Vs. Non-Linear: glm vs. xgb

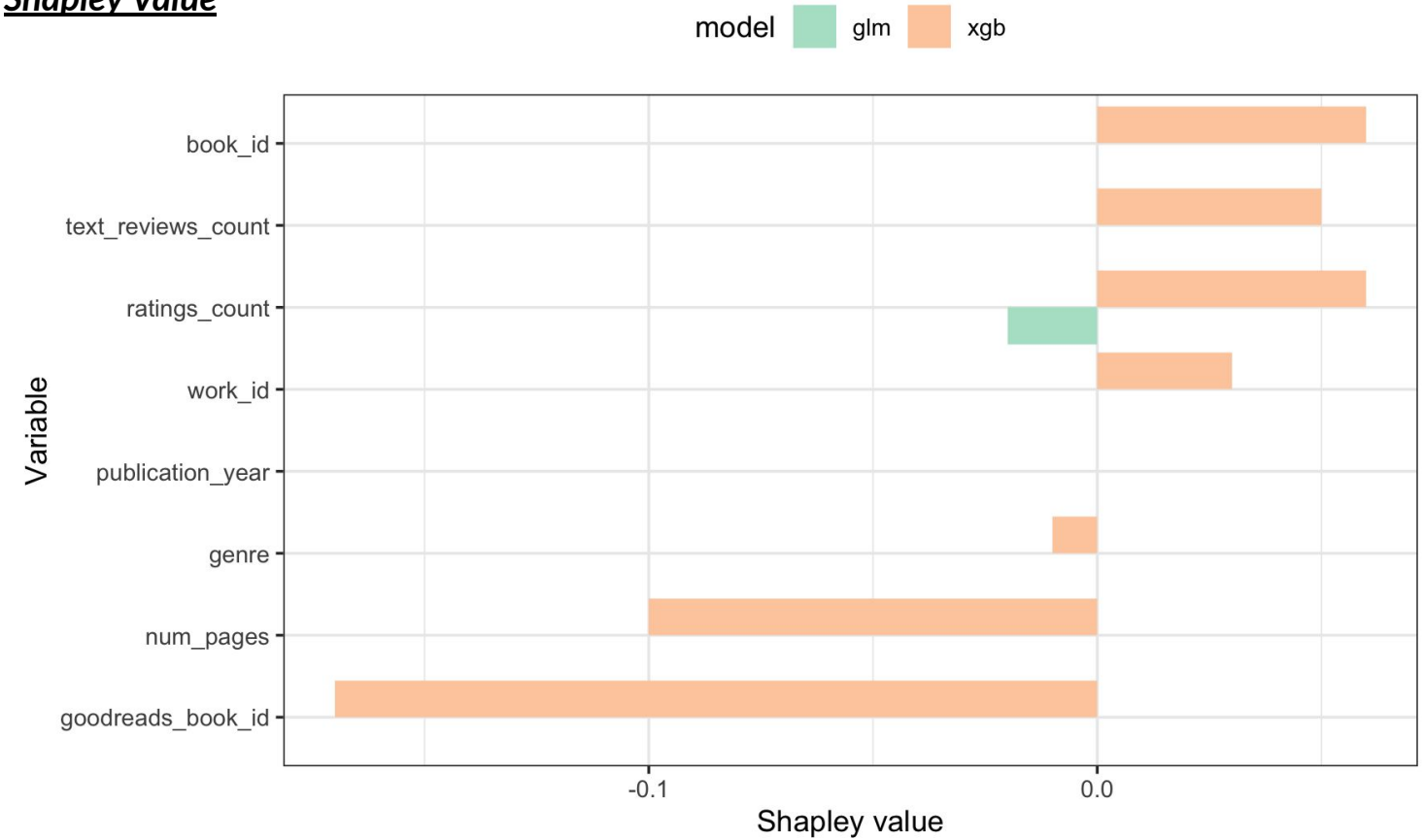
Features Interaction





Classification>>Linear Vs. Non-Linear: glm vs. xgb

Shapley Value



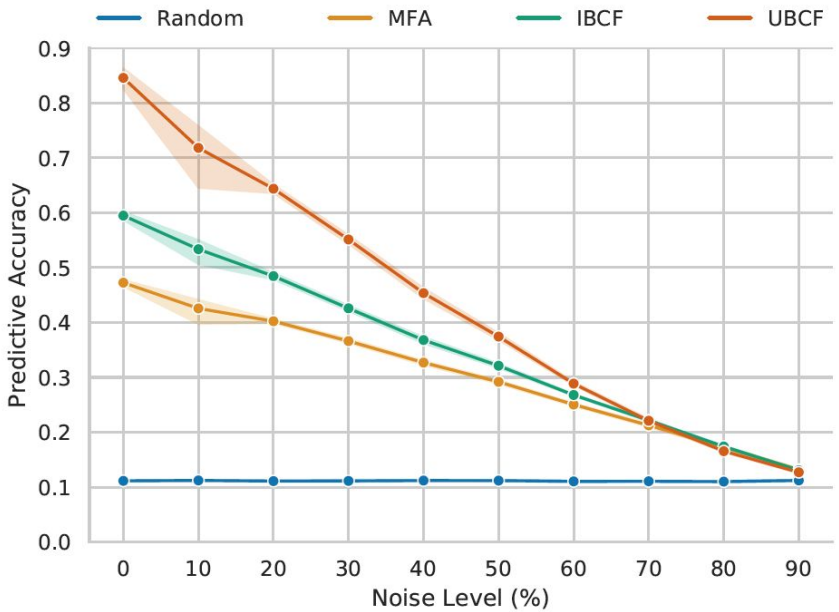
Unsupervised

Clustering

-Kmeans

-Collaborative filtering

Clustering >> Collaborative Filtering



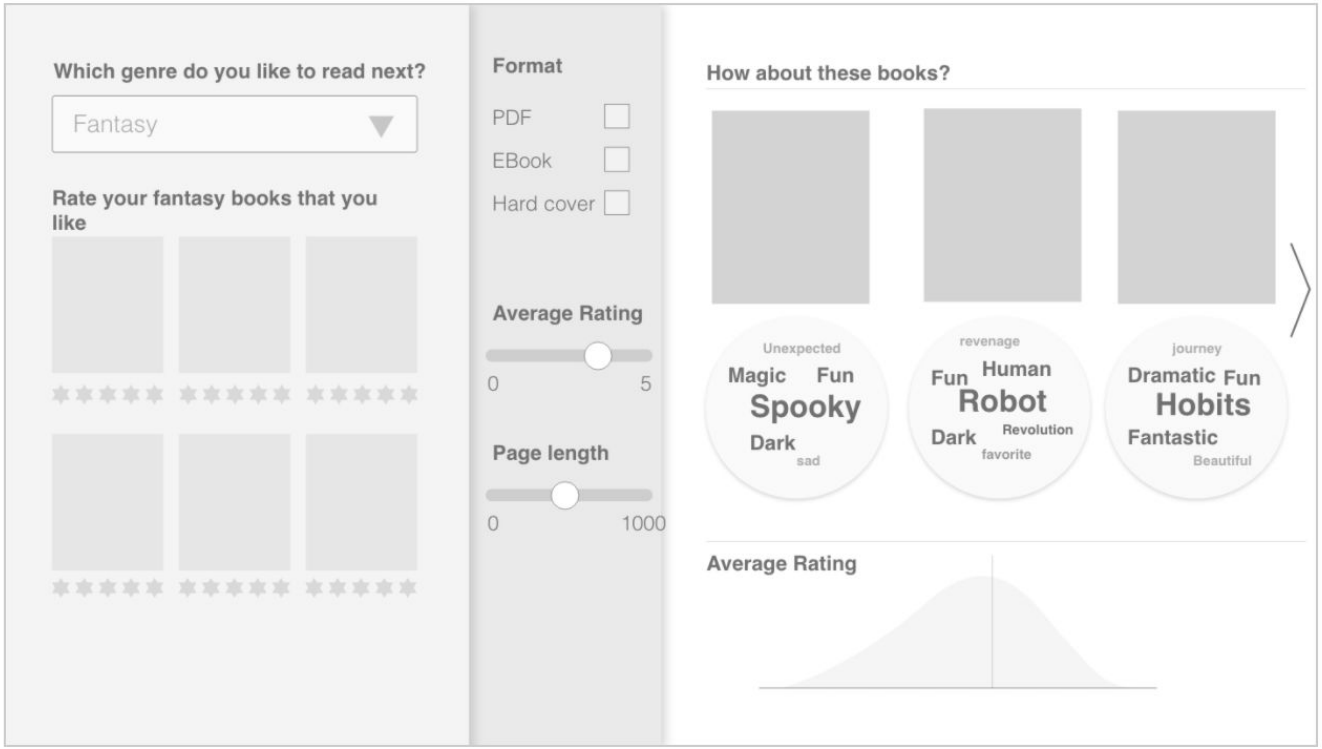
Riesterer, N., Brand, D., & Ragni, M. (2020). Uncovering the Data-Related Limits of Human Reasoning Research: An Analysis based on Recommender Systems. *arXiv preprint arXiv:2003.05196*.

5

Application and conclusions

Prototype

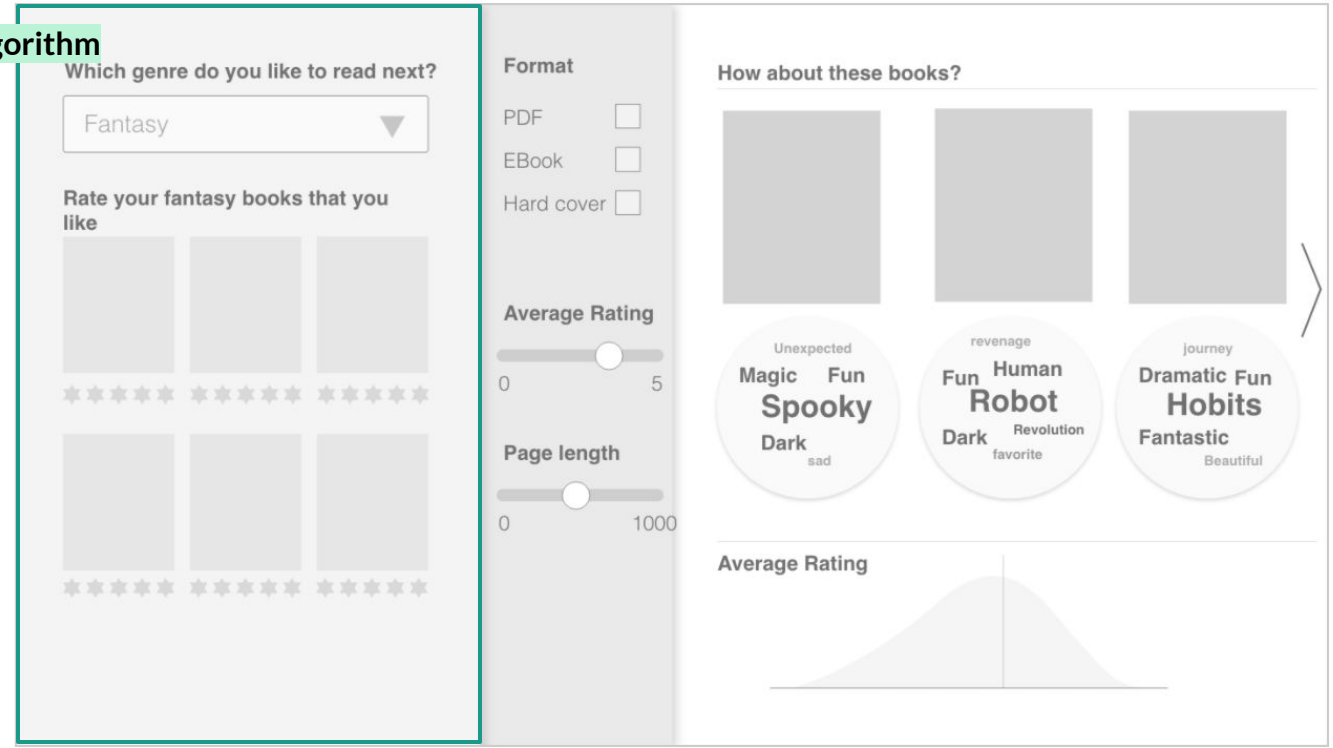
version1



Prototype

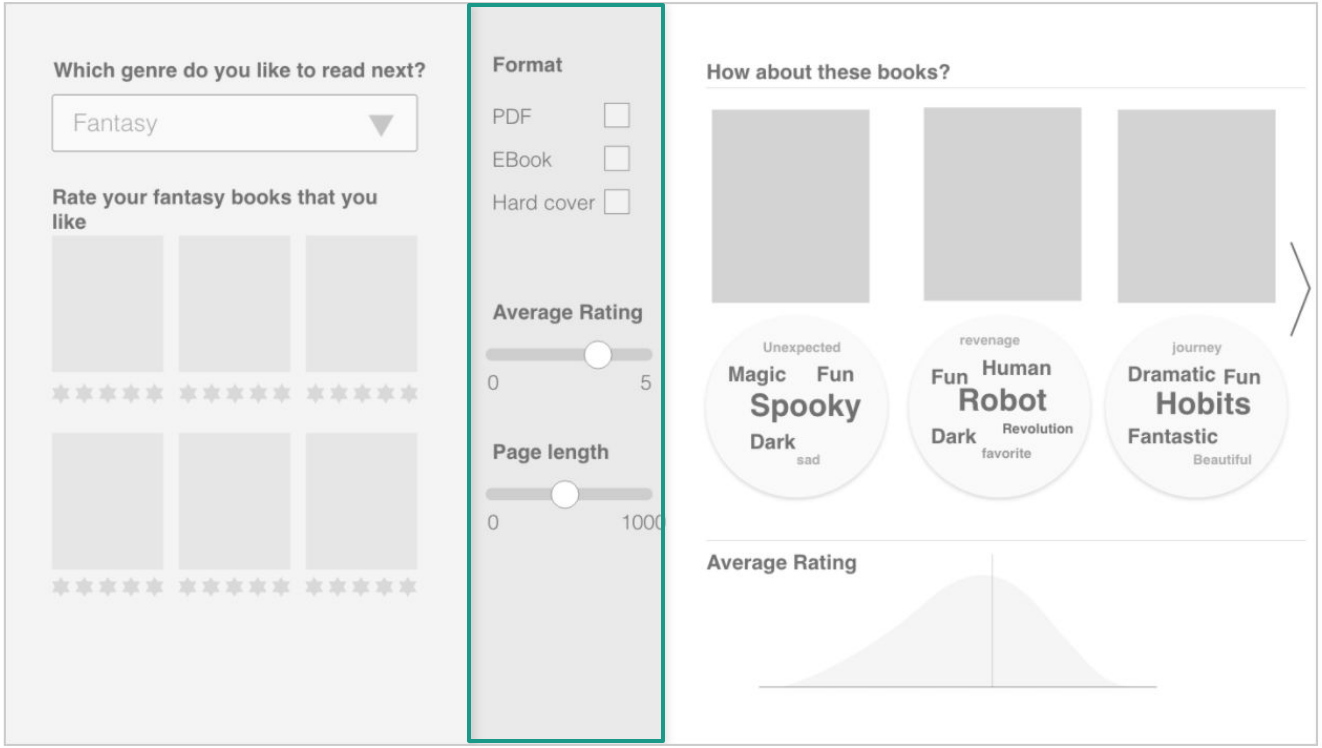
version1

Give Control of Algorithm



Prototype | version1

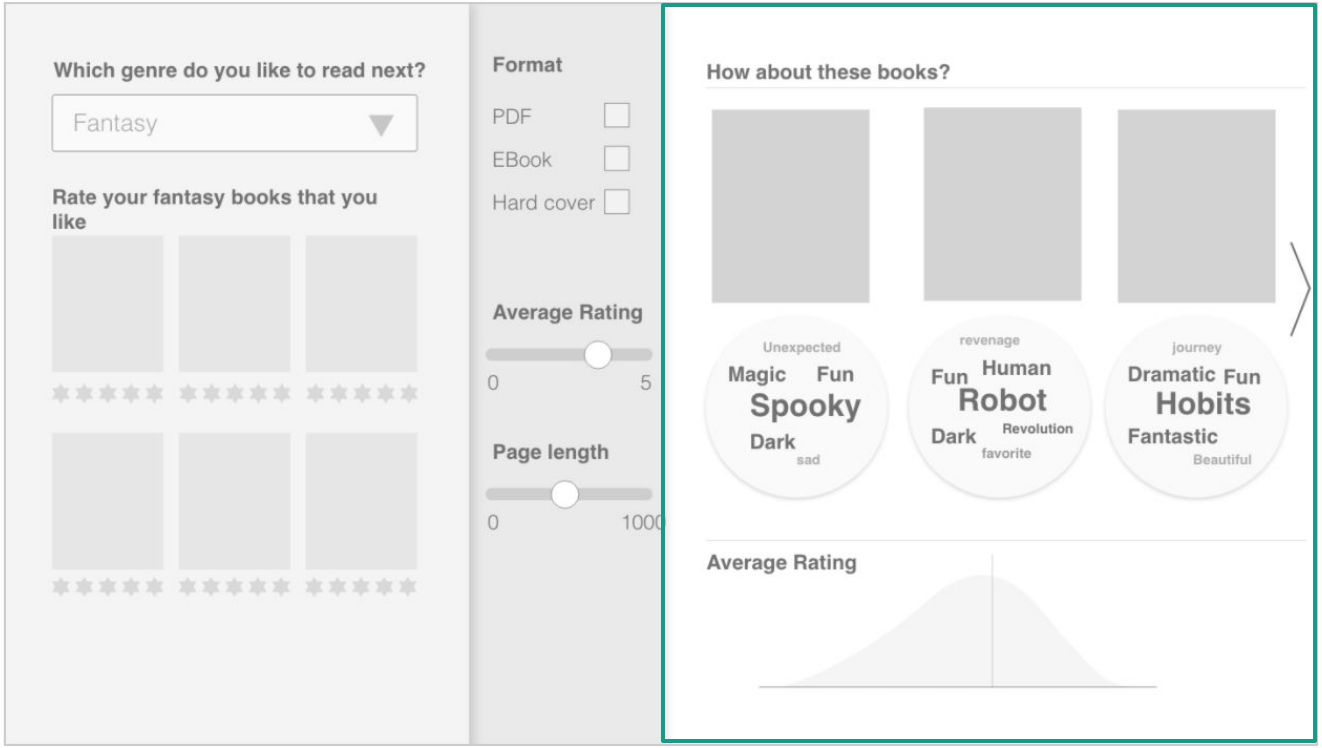
Filtering



Prototype

version1

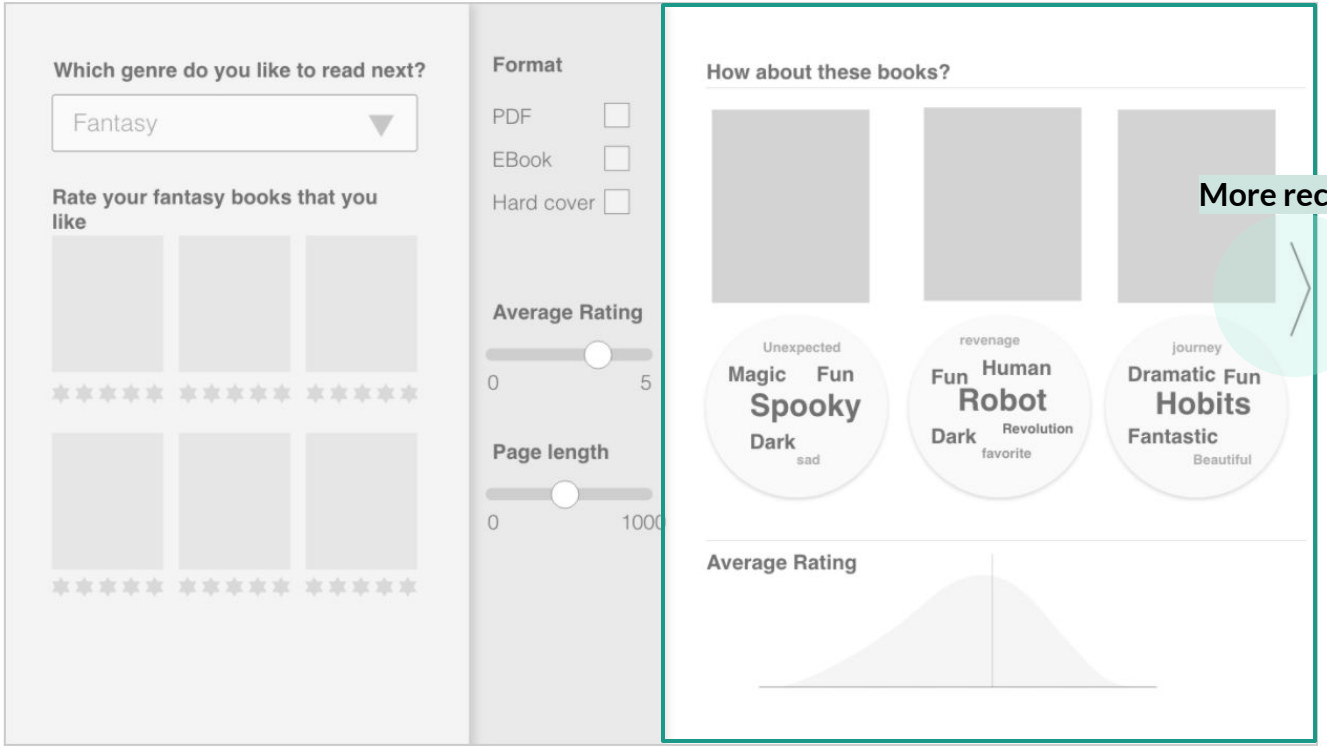
Recommendation



Prototype

version1

Recommendation

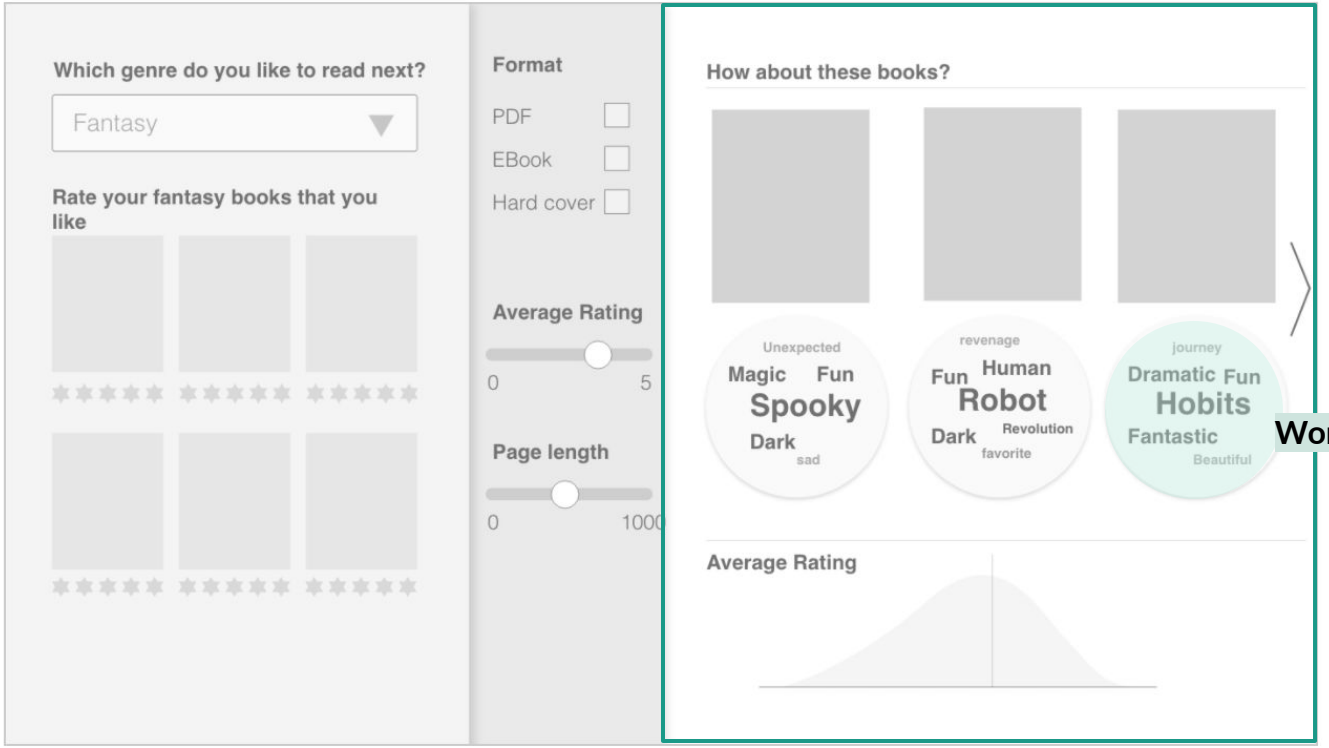


More recommendations

Prototype

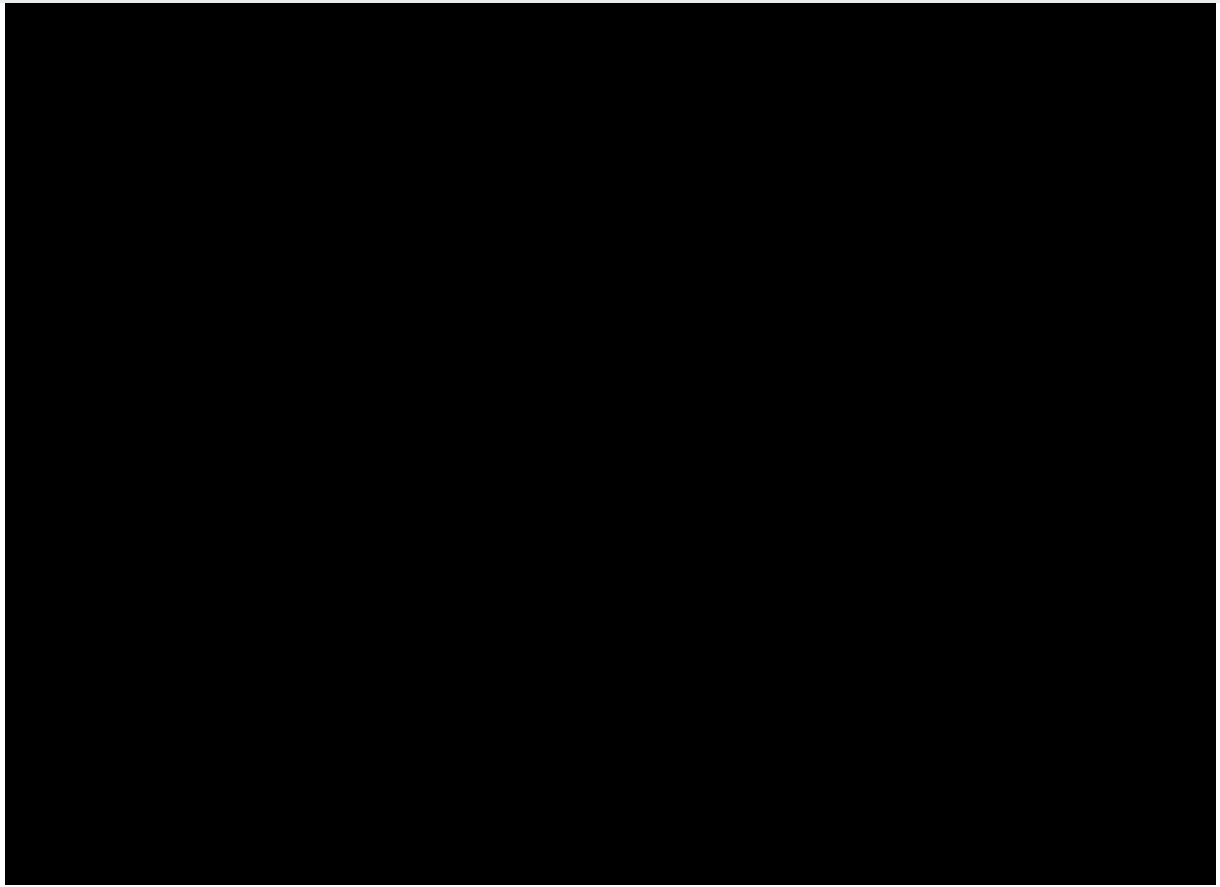
version1

Recommendation

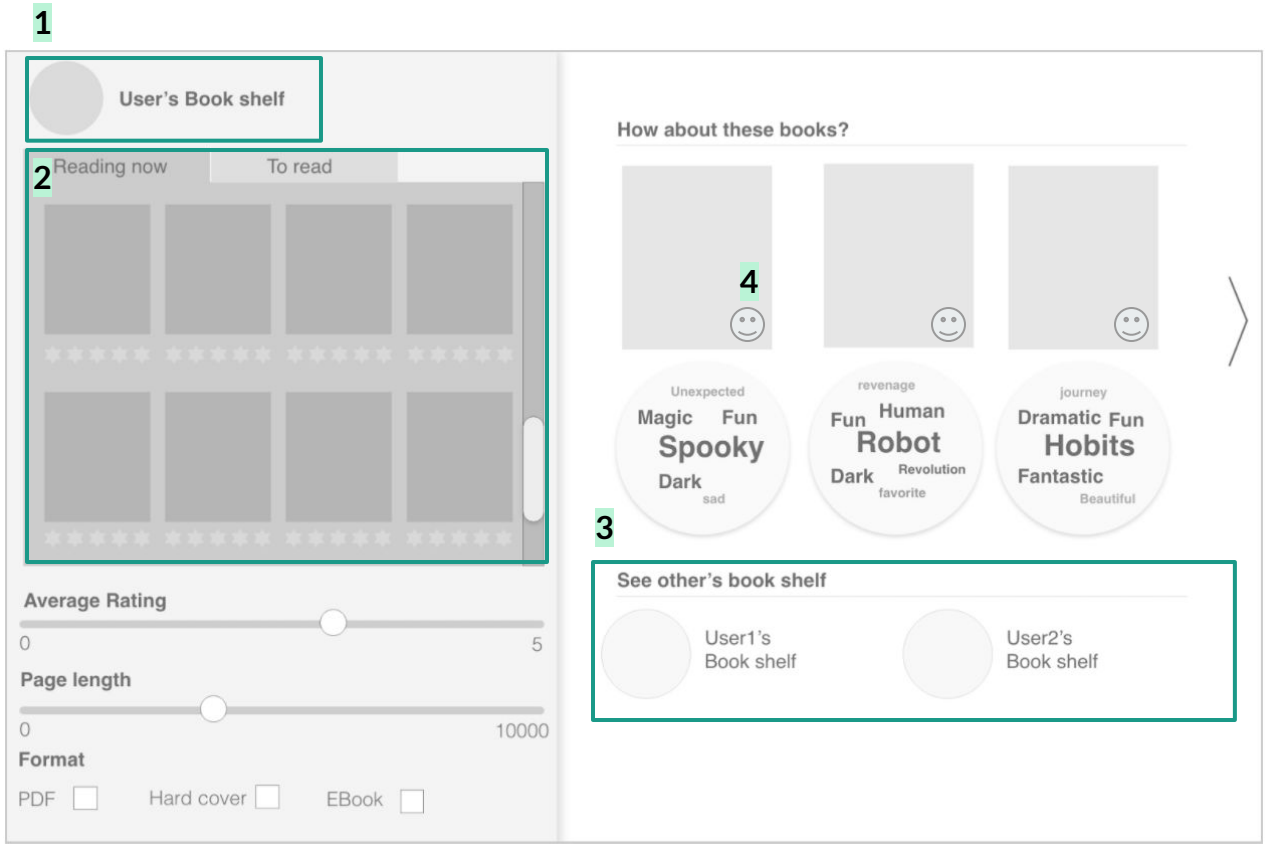


Words cloud

APP DEMO



Prototype



We made this prototype to improve trust and persuasion.

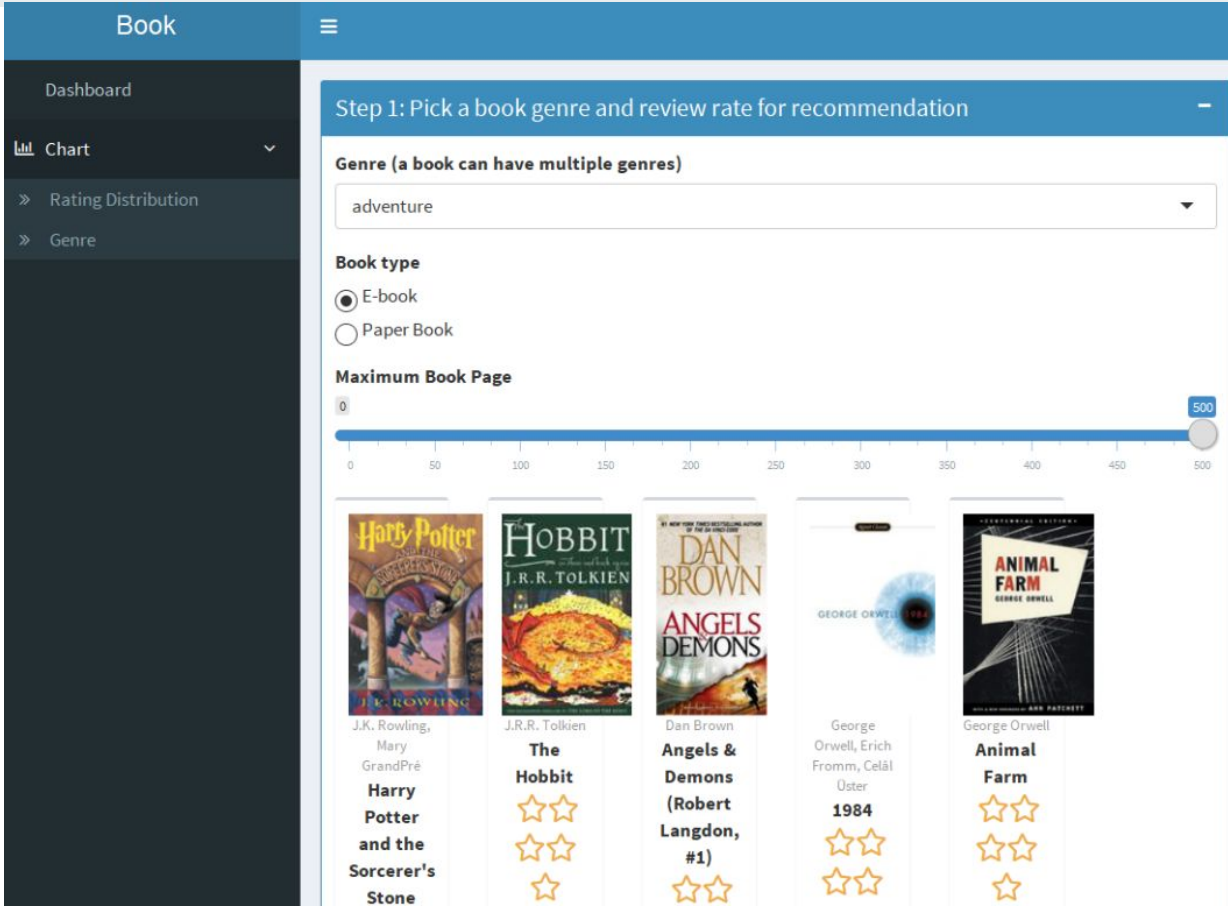
1. User's profile photo with the title 'OO's bookshelf' -> Increase the feeling of personalized service.
2. We separated users' bookshelf into 1. Reading now 2. To read. The algorithm can understand the user's current and future interests.
3. By showing other users' book self, we wanted to give users feel like it's a book reading community. We think we can further add curated book sections.
4. Liking/Disliking the recommendation -> algorithm can reflect the user's preference after the recommendation

Further improvements

Test on performances
- User testing

Give more control to users
- Add more customizable filters

Increase trust and persuasion



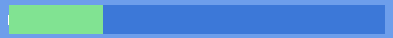
References

- Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", in *RecSys'18*. [bibtex]
- Mengting Wan, Rishabh Misra, Ndapa Nakashole, Julian McAuley, "Fine-Grained Spoiler Detection from Large-Scale Review Corpora", in *ACL'19*. [bibtex]
- Riesterer, N., Brand, D., & Ragni, M. (2020). Uncovering the Data-Related Limits of Human Reasoning Research: An Analysis based on Recommender Systems. arXiv preprint arXiv:2003.05196.

- Recommendation Algorithms:

  https://www.kaggle.com/philippsp/book-recommender-collaborative-filtering-shiny/code
- Choosing an optimal number of clusters:

  https://towardsdatascience.com/10-tips-for-choosing-the-optimal-number-of-clusters-277e93d72d92
- Working and benefits of UMAP: https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668



Questions?

Thank you