Book Recommedation System

Group 6: Vogli Aleksandro[1820382], Chun-Yi Chen[1820370], Piin Shiuan Ho[1859808], Agarwal Mayank[1725206], Omogha Oghenekeno Utomudo[1725331]

Department of Business Informatics
University of Mannheim, Germany

1 Introduction

Online book reading and selling websites compete against each other on many factors. One of those vital factors is their book recommendation system. A good recommendation system can help them to retain customers in order to increase profits.

However, when the number of book choices is overwhelming, how to provide users with personalized content and services will become a potential problem. The main objective of our Web Mining Project is to address the problem of information overload and improve better user experiences for book recommendations.

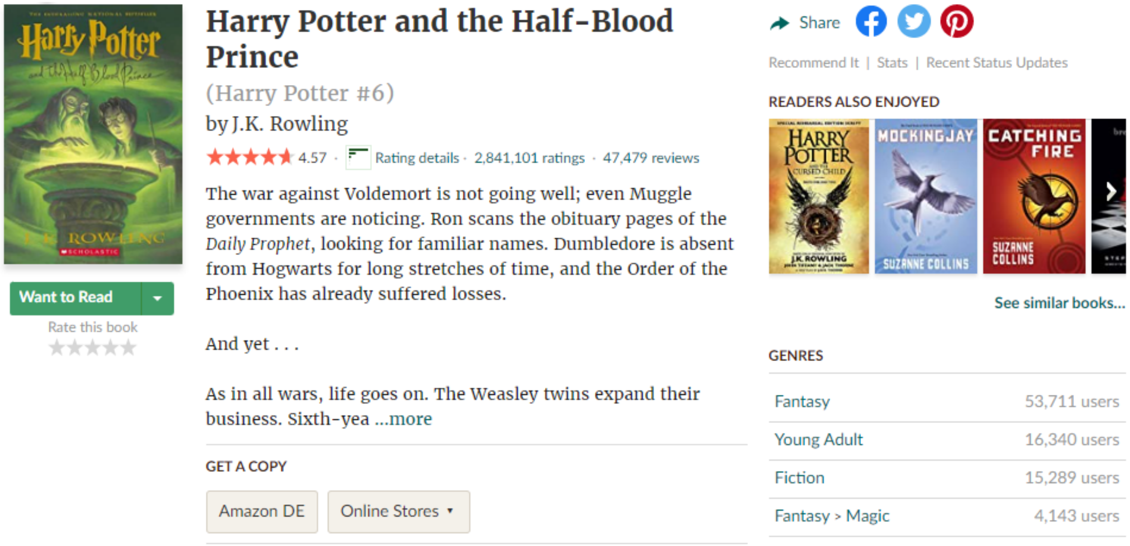
2 The Dataset

Code of the project : https://github.com/AustinChen123/WebMining

Dataset	Format	# of attributes	List of attributes
2M Goodreads Book Datasets	.csv	18	ISBN Id Language
Books(Book Recommendation Dataset)	.csv	8	ISBN Book-Title Book-Author Year-of-Publication Publisher
Users(Book Recommendation Dataset)	.csv	3	User Id Location User Age
Ratings(Book Recommendation Dataset)	.csv	3	User Id ISBN Rating

Table 1: Dataset attributes

Aside from these data we found on Kaggle, we also need more text variable such as genre to implement the content-based method. But because the Goodreads API does not provide book genre, eventually we developed a web crawler to get the genre data using the ID and ISBN in our dataset.



Harry Potter and the Half-Blood Prince

(Harry Potter #6)

by J.K. Rowling

★★★★★ 4.57 · Rating details · 2,841,101 ratings · 47,479 reviews

The war against Voldemort is not going well; even Muggle governments are noticing. Ron scans the obituary pages of the *Daily Prophet*, looking for familiar names. Dumbledore is absent from Hogwarts for long stretches of time, and the Order of the Phoenix has already suffered losses.

And yet . . .

As in all wars, life goes on. The Weasley twins expand their business. Sixth-yea ...more

GET A COPY

Amazon DE | Online Stores ▾

Want to Read ▾

Rate this book ☆☆☆☆☆

Share

Recommend It | Stats | Recent Status Updates

READERS ALSO ENJOYED

See similar books...

GENRES

Fantasy	53,711 users
Young Adult	16,340 users
Fiction	15,289 users
Fantasy > Magic	4,143 users

Fig. 1: Example image of our web crawler

3 Methodology

Recommender system is a strategy that allows user to make decision when they have various information around. It can handle these information and provide personalized recommendations to the users. For this recommendation tasks, there are several methods based on different information. For our project, we aims to create a good book recommender with our data, so we are going to perform different types of recommender system method, e.g. Collaborative Filtering, Content Based RS, and compare the performance to see which one is better, and how to improve. The approaches we used for the project are:

1. Collaborative Filtering
   - Memory based
     - Calculating similarity for each user based on the books they rated.
     - KNN
   - Model Based
     - Matrix Factorization using SVD and Stochastic Gradient Descent as a loss function

2. Content Based
   - Based on “Genre” and users’ preferences, similarity using TF-IDF and Cosine Similarity will be calculated
   - Based on “Genre” and "Book title", create BERT embedding vectors to calculate cosine similarity
3. Hybrid Recommendation System
   - A combination of the user, item and content based approaches, aims to have more information to recommend.
4. Popularity based
   - Weighted average of scores, which further consider the counts of the vote(popularity).
   - Make the same recommendation to every user, based on the popularity of an item.

The idea of Collaborative filtering is to recommend items to a user based on the ratings or reactions of others users that are similar to a user. This poses the question of how to find users or items that are similar to each other, how to determine the rating a user would rate an item given the ratings of similar users and how to evaluate the accuracy of the ratings the recommender system predicts for a user based on the ratings of similar users.

To get the similarity of the users, the first approach we use is a rating similarity based approach. We find similar users on the basis of their ratings and use KNNWithMeans for computing user similarities. The KNNWithMeans algorithm is derived from the nearest neighbours approach and a basic collaborative filtering approach that considers the mean ratings of each user. We apply pearson correlation coefficient to measure the user based similarity as described below, where a, b : users $r_{a,p}$: rating of user a for item p and P: set of items, rated by both a and

$$\operatorname{pred}(a, p)=\frac{\sum_{b \in N} \operatorname{sim}(a, b) * r_{b, p}}{\sum_{b \in N} \operatorname{sim}(a, b)} \tag{1}$$

We further use a prediction function below for predicting a users rating given the rating of similar users.

$$\operatorname{sim}(\boldsymbol{a}, \boldsymbol{b})=\frac{\sum_{p \in P}\left(\boldsymbol{r}_{a, p}-\overline{\boldsymbol{r}}_{a}\right)\left(\boldsymbol{r}_{b, p}-\overline{\boldsymbol{r}}_{b}\right)}{\sqrt{\sum_{p \in P}\left(\boldsymbol{r}_{a, p}-\bar{r}_{a}\right)^{2}} \sqrt{\sum_{p \in P}\left(\boldsymbol{r}_{b, p}-\bar{r}_{b}\right)^{2}}} \tag{2}$$

For our model based collaborative filtering approach, we have used matrix factorization to identify the relationship between item features and user preferences. This technique when applied with collaborative filtering generates latent features from the user and the item. Importantly, since every book reader does not read every book in the dataset, the resulting effect will be a sparse matrix with missing values. In this project, we have used Singular Value Decomposition (SVD) for matrix factorization to get the most important singular values of our features and eliminate less important values which could result in noise. The Content based approach are useful in scenarios when there is an insufficient amount of rating data available and therefore it is very applicable in our case. An algorithmic approach to content based recommendation system relies on defining a set of rules, which can be used to rank books. Our first approach which we use as a baseline to content based recommendation would be using TF-IDF.

Another approach, we have looked upon is based on the Genre and Book title. The Book title often is used as a way to understand user preferences and behaviour. For Genre, We have extracted top 20 Genres for this analysis which we would use it in BERT embedding and cosine Similarity to get better understanding of user preferences.

We also used a Hybrid recommendation approach, which is using best of both worlds, Collaborative Filtering and Content based. For this approach, we had to first find the similar books using cosine similarity (Content-based) approach based on user preference. Then, we use these similarity ratings to predict the book ratings using the collaborative filtering approach which ultimately gives us the highest rated books by user.

We have further used a Popularity based approach which works on the principle of popularity, otherwise known as trends. As the name implies, this method recommends to a user the most popular items that are currently trending or most popular among other users. To identify the popularity class of a book, instead of using the arithmetic mean. We consider the rating scores and the total ratings count(a proxy of the popularity) that the books had garnered. We further define a threshold of the minimum number of votes that determine if a book gets on the trending chart. Then, this could give us a greater popularity list of books with the general populace.

One of the most important goal in using matrix factorization is to minimize the loss present in the sparse matrix. SGD is one algorithm used to minimize the loss, which we have adopted to minimize the squared error on the known user ratings.

4 Experimental setting

For the final data we used, we have 67047 ratings, 8359 unique users and 7462 books with 95% non-na attributes. For those na features, there are 90% na values in "description" and 100% na in "PagesNumber". Overall, most of the data is available and all of the columns we need are 100% non-na.

For the rating distribution, in Figure 2, we can see the distribution is seriously skewed, most of the ratings are scoring 0, and this is not a good pattern to a recommendation system. We found that this is a data quality problem since the original rating data with 340,000 books also have the same issue so we are going to try binning the ratings and log transform to fix the problem.

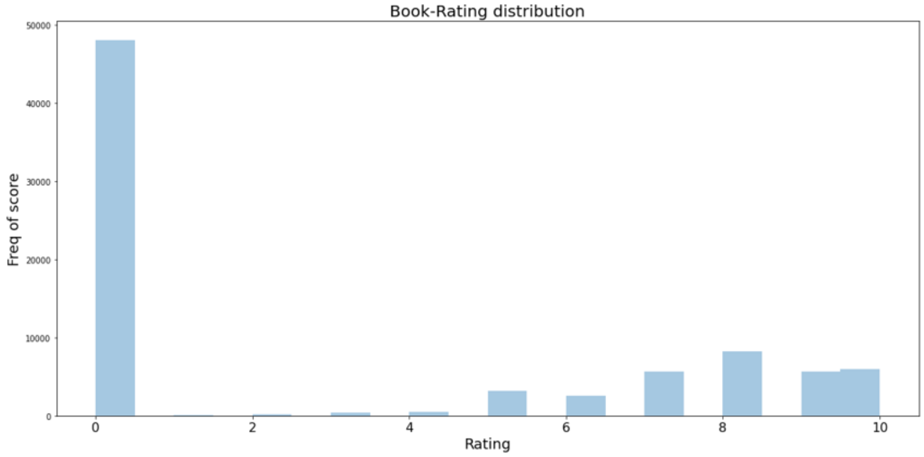


Fig. 2: Rating distribution of our dataset.

For book binning, we generate the new rating score $\hat{x}$, $\hat{x}$ is defined as:

$$rating(\hat{x}) = \begin{cases} 0 & \text{, if original rating x between 0 and 2} \\ 1 & \text{, if original rating x between 2 and 4} \\ 2 & \text{, if original rating x between 4 and 6} \\ 3 & \text{, if original rating x between 6 and 8} \\ 4 & \text{, if original rating x between 8 and 10} \end{cases} \tag{3}$$

And for the log transform, we apply log1p, which add 1 to the original value before taking log, to prevent taking log to 0 value. Its inverse, expm1, will be used when calculating the RMSE. The formulas are shown as follow:

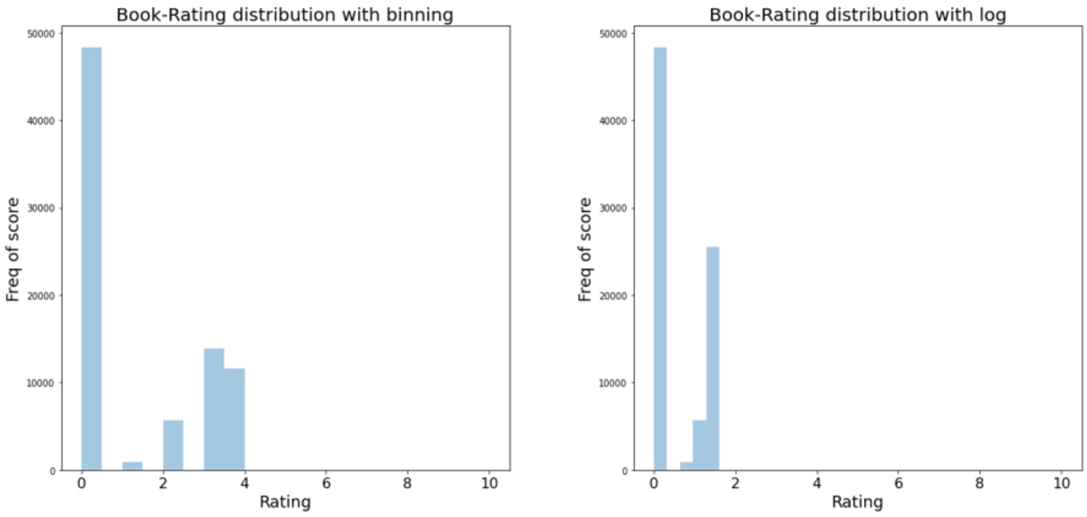
$$log1p(x) = \log(x + 1) \tag{4}$$

$$expm1(x) = exp(x) - 1 \tag{5}$$

We implemented matrix factorization using the Surprise library, and trained 75% of the known ratings of the dataset. To evaluate the performance of our model on the test set, we used the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as metrics.

The rating distributions after these adjustment are shown in Figure 3. We can see the distribution of both method become more concentrated but still not a normal distribution. We will compare the performance of different rating adjustment in the later part.

To further improve our model, we have used 5 folds cross-validation on the split training set. We have also tuned the latent factors to extract on "n_factors" of 50, 100 and 150 respectively.



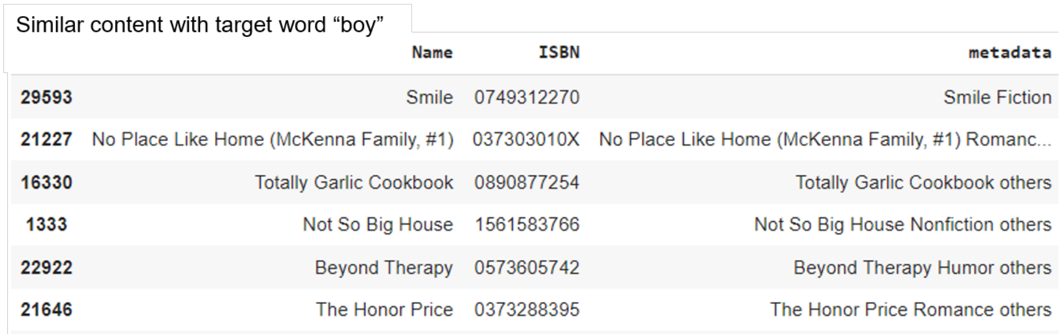
(a) 1a

(b) 1b

Fig. 3: Rating distribution after adjustment.

5 Evaluation and discussion of the results

For BERT embedding content-based RS, Figure 4 is the result when we type "boy" as our target word of content. After really look into the recommendation result, we think the BERT embedding result is worse than basic TF-IDF method. And the reason might because our metadata only contain book title and genre, which might be not enough for a deeplearning framework. Furthermore, it is not easy to know the content of the book just by genre and title, so simple TF-IDF would work better when the task is just finding similar book name.



Similar content with target word “boy”

	Name	ISBN	metadata
29593	Smile	0749312270	Smile Fiction
21227	No Place Like Home (McKenna Family, #1)	037303010X	No Place Like Home (McKenna Family, #1) Romanc...
16330	Totally Garlic Cookbook	0890877254	Totally Garlic Cookbook others
1333	Not So Big House	1561583766	Not So Big House Nonfiction others
22922	Beyond Therapy	0573605742	Beyond Therapy Humor others
21646	The Honor Price	0373288395	The Honor Price Romance others

Fig. 4: BERT embedding content-based RS.

Initial results of our baseline model in 3 and 2 show a score of 3.66 and 3.08 for RMSE and MAE respectively, without experimentally binning the rating distribution and handling the skewed distribution.

	Baseline	with Binning	with LogTransform
SVD(50)	3.1147	1.6216	0.5672
SVD(100)	3.1025	1.6145	0.5648
SVD(150)	3.0765	1.5997	0.5620
KNN(20,Pearson,user-based)	3.6079	2.5172	0.7200
KNN(20,Cosine,item-based)	3.6326	3.5475	0.9553

Table 2: MAE of different algorithms and preprocessing steps using cross-validation.

When look at the performance of different preprocessing steps, we found out that both steps work well, but the RMSE in KNN increase significantly than baseline. It might because the original ratings are merged together, so some of the patterns exist between the neighbors are disappear after we bin the ratings

	Baseline	with Binning	with LogTransform
SVD(50)	3.6587	3.3542	0.9301
SVD(100)	3.6568	3.3765	0.9230
SVD(150)	3.6505	3.3883	0.9214
KNN(20,Pearson,user-based)	4.1693	7.9847	1.5470
KNN(20,Cosine,item-based)	4.2569	9.7760	1.9105

Table 3: RMSE of different algorithms and preprocessing steps using cross-validation.

Compared all results we have above, we found SVD with n_factor=150 is the best model and parameter in our project.

6 Conclusions

6.1 Model result

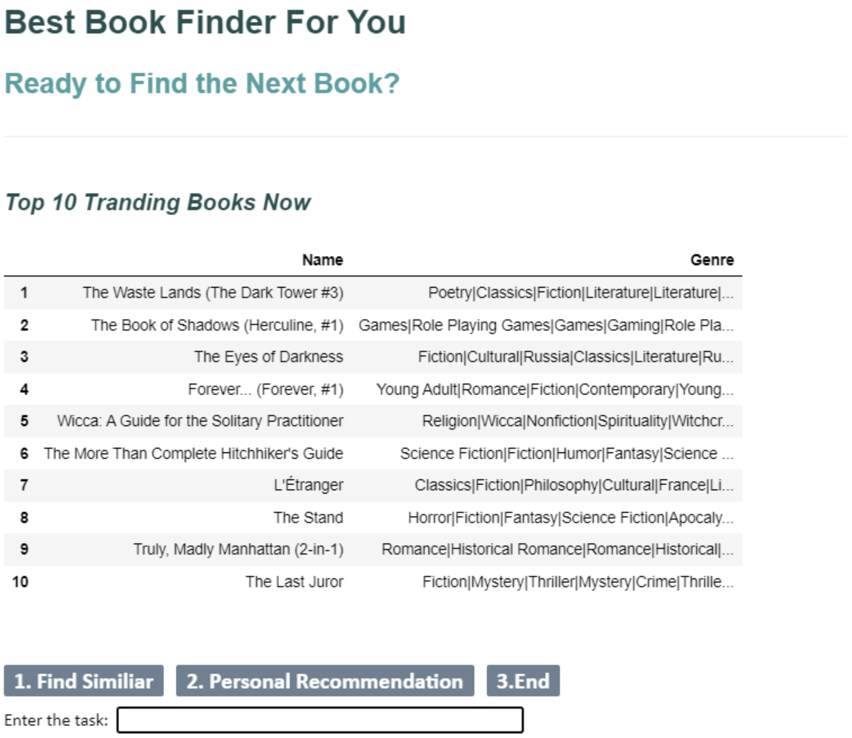
The result of our book recommendation system can be measured and evaluated using several metrics that can be compared. One of the metrics is the RMSE between baseline and other preprocessing methods. With a massive reduction from 3.65 to 0.93, we can say a good result is achieved. The main problems that prevent a better result of RMSE arise from the data quality, f. ex, the massive amount of 0 existed in the rating records.

Another important role is played by the density of the dataset and the density of individual attributes. All of our data and attribute we used is non-na. Thus, it supports our system to expand the information on user and book to consequently perform better recommendation.

Overall, our models are working fine and the RMSE reduced to a acceptable level. But the modelling process can be improve by using better quality data, such as more text variable (e.g. book introduction) for content-based model and better rating distribution.

6.2 Use scenario

After the implementation, we decided to build a sample book recommendation system to show how those approaches could be realized in a real scenario. First, when people reach the system, they will see the top 10 Trending books Now. The rating list is generated by the Popularity Based model considering the rating counts and average rating.(Figure 5) Second, for different customer groups, we designed different approaches.



Best Book Finder For You

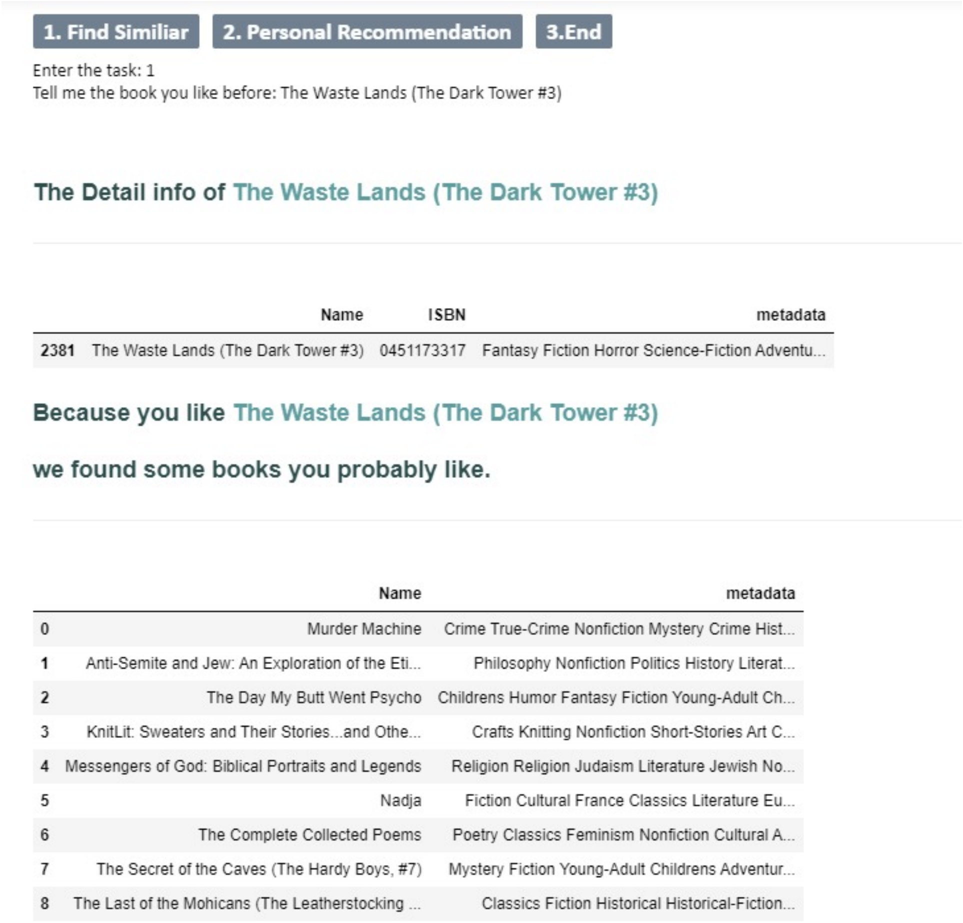
Ready to Find the Next Book?

Top 10 Tranding Books Now

	Name	Genre
1	The Waste Lands (The Dark Tower #3)	Poetry|Classics|Fiction|Literature|Literature|...
2	The Book of Shadows (Herculine, #1)	Games|Role Playing Games|Games|Gaming|Role Pla...
3	The Eyes of Darkness	Fiction|Cultural|Russia|Classics|Literature|Ru...
4	Forever... (Forever, #1)	Young Adult|Romance|Fiction|Contemporary|Young...
5	Wicca: A Guide for the Solitary Practitioner	Religion|Wicca|Nonfiction|Spirituality|Witchcr...
6	The More Than Complete Hitchhiker's Guide	Science Fiction|Fiction|Humor|Fantasy|Science ...
7	L'Étranger	Classics|Fiction|Philosophy|Cultural|France|Li...
8	The Stand	Horror|Fiction|Fantasy|Science Fiction|Apocaly...
9	Truly, Madly Manhattan (2-in-1)	Romance|Historical Romance|Romance|Historical|...
10	The Last Juror	Fiction|Mystery|Thriller|Mystery|Crime|Thrille...

Fig. 5: Make the same recommendation to every user, based on the popularity of an item.

For new incoming users in our system, we will use the Content-Based model to recommend similar products that liked before.(Figure 6)



```
Enter the task: 1
Tell me the book you like before: The Waste Lands (The Dark Tower #3)
```

The Detail info of The Waste Lands (The Dark Tower #3)

	Name	ISBN	metadata
2381	The Waste Lands (The Dark Tower #3)	0451173317	Fantasy Fiction Horror Science-Fiction Adventu...

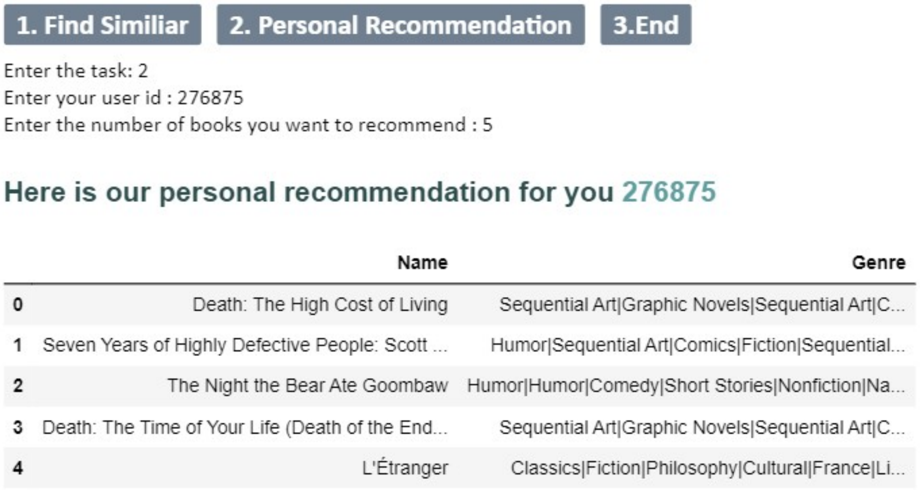
Because you like The Waste Lands (The Dark Tower #3)

we found some books you probably like.

	Name	metadata
0	Murder Machine	Crime True-Crime Nonfiction Mystery Crime Hist...
1	Anti-Semite and Jew: An Exploration of the Eti...	Philosophy Nonfiction Politics History Literat...
2	The Day My Butt Went Psycho	Childrens Humor Fantasy Fiction Young-Adult Ch...
3	KnitLit: Sweaters and Their Stories...and Othe...	Crafts Knitting Nonfiction Short-Stories Art C...
4	Messengers of God: Biblical Portraits and Legends	Religion Religion Judaism Literature Jewish No...
5	Nadja	Fiction Cultural France Classics Literature Eu...
6	The Complete Collected Poems	Poetry Classics Feminism Nonfiction Cultural A...
7	The Secret of the Caves (The Hardy Boys, #7)	Mystery Fiction Young-Adult Childrens Adventur...
8	The Last of the Mohicans (The Leatherstocking ...	Classics Fiction Historical Historical-Fiction...

Fig. 6: List of book recommendation based on the book you like before.

For users that had a rating history before, we use the Hybrid Recommendation model(combined with the Collaborative Filtering approach and Content-Based approach) to give them a better personal experience on item recommendation.(Figure 7)



```
Enter the task: 2
Enter your user id : 276875
Enter the number of books you want to recommend : 5
```

Here is our personal recommendation for you 276875

	Name	Genre
0	Death: The High Cost of Living	Sequential Art|Graphic Novels|Sequential Art|C...
1	Seven Years of Highly Defective People: Scott ...	Humor|Sequential Art|Comics|Fiction|Sequential...
2	The Night the Bear Ate Goombaw	Humor|Humor|Comedy|Short Stories|Nonfiction|Na...
3	Death: The Time of Your Life (Death of the End...	Sequential Art|Graphic Novels|Sequential Art|C...
4	L'Étranger	Classics|Fiction|Philosophy|Cultural|France|Li...

Fig. 7: List of book recommendation based on the user previous books rating.

References

1. kellypeng: ScentMate, https://github.com/kellypeng/scentmate_rec
2. ecbenezra: recommender-system,https://github.com/ecbenezra/recommender-system