KuldeepSinh Chudasama, Sneha Gopinath , Nisarg Shah, and Freny Savalia

1. Introduction

1.1 Problem Statement

Recommender systems are information search and decision support tools used when there is a set of options or when the user lacks the domain-specific knowledge essential to take decision. It differentiates the data based on various algorithms and suggests the user with the most relevant product. The suggestion of the products is based on the customers' traits and needs.

This report presents an overview of the recommendation systems,

- The aim of the proposed system is to help individuals to provide one single platform that provides recommendations of most popular book among all the users for providing better user experience and helping the user to keep up with the trend.

- The proposed system provides various types of recommendations on the basis of book author, publication, ratings, location, age that in turn enhances the quality of information provided by system.

- This project also proposes a new book recommender system that combines user choices with not only similar users but other users as well to give diverse recommendation that change over time.

1.2 Motivation

Recommendation System are new generation internet tool which help the user in navigating through information on the internet and receive information related to their preferences. There are many famous recommendation systems that are present in various fields like online shopping, movie and music. The applications of recommendation system in various other fields are also researched on. This driving force was to develop a recommendation system in the field of online book shopping.

In the field of recommendation system of recent progress include intelligent method of filtering and choosing information. Example: TV listings in Netflix and YouTube video recommendations. The users have diverse and conflicting needs. Various differences exist between users due to personal, professional, private preferences, cultural, social and educational background. Due to all this the user would desire to have personalized recommendation systems that would process, filter, and display available information in a manner that suits each individual using them. The need for personalization is the motivation that led to the development of this project of recommendation system that adapt based on the inferred characteristics of the user interacting with them.

1.3 Review of other researches

The research paper that we have referred used kNN Collaborative filtering to generate the recommendations. The conventional content based recommendation systems recommends items with similar features to users while the collaborative filtering based systems predict the user preferences by analysing past relationships between users and interdependencies among items.
Thus the CF works under the belief that if a person A has same opinion as 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. Thus, serves as the fundamental principle to create user based and item based collaborative filtering techniques. The other research paper that we have referred which uses the RBM for collaborative filtering uses the different approach compared to conventional approaches. Most of the existing approaches to collaborative filtering cannot handle very large data sets. In this paper the authors have showed how a class of two-layer undirected graphical models, called Restricted Boltzmann Machines (RBM's), can be used to model tabular data, such as user's ratings of movies. They have applied the RBM models on Netflix's 1 million dataset and have achieved error rate that is well over 6% better than the score of Netflix's own system.

1.4 Open questions in the domain

The algorithm that is best for book recommender system. To explore other algorithms for book recommendation system.
Finding a good evaluation metric for the recommendation system other than RMSE. Most of the algorithms use RMSE to check the accuracy of the algorithm. But Netflix and YouTube have stopped using this type of approach as the lower RMSE doesn't

guarantee good recommendations. Now they use hit rate approach where there into consideration the users logging into their site to analyse.
Different methods to handle the sparsity present in the matrix. There can be another method like dense matrix, TFIDF, Word2Wek. Training this huge dataset using Hadoop/Spark.

1.5 Short summary of your proposed approach

Approach 1 (Proposed Approach):

We have referred the following articles regarding the recommender system. The research papers focus on analysing the k-nearest neighbor, collaborative filtering and Item based collaborative filtering techniques. They have looked into different techniques to compute item-item similarities based on correlation vs. cosine similarities between item vectors. The authors have used different techniques for obtaining recommendations from them (weighted sum vs. regression models). The authors have experimentally evaluated the results and compared the results of the various recommendation algorithms and have suggested that item-based algorithms provide better performance and quality than user-based recommendation algorithms.

- Item-Based Collaborative Filtering Recommendation Algorithms
- https://ieeexplore.ieee.org/document/7684166
- https://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf
- Understanding Basics of Recommendation Engines
- Quick guide to build Recommendation Engine in Python

Approach 2 (Our Approach):

We are using the K-Nearest Neighbor algorithm to cluster similar users based on similar interests of the users. For the collaborative filtering techniques we build the matrix which has users as rows and columns as the ISBN of books. The corresponding value represents the ratings given by the users to each item. This matrix is very sparse in nature.
To handle the sparsity of the matrix and to cope up with the computation power we have considered only those users and books for which explicit ratings are provided. The k Nearest Neighbors algorithm is used to find the k-similar users based on similarities of the ratings given to the books. In this way we will have the user based collaborative filtering.
For item based collaborative filtering, we are finding the k similar books based on the similarities with other items from the ratings matrix that we have created. We are utilizing the user’s like history to recommend similar books liked by other users.
We are also using the Restricted Boltzmann Machines for Collaborative Filtering.We have trained the RBM on the normalized ratings matrix to generate the results. At last we have built our hybrid recommender system algorithm that uses the RBM and

KNN based collaborative filtering approach and merge the results based on the weights given to each algorithm.

Difference/Novelty(Instructors Feedback):

We are creating a hybrid recommender system. First we are using the kNN model to find k similar users/items for user/item based collaborative filtering.
Similarly, we are also using the Restricted Boltzmann Machines (RBM) for Collaborative Filtering, The RBM based collaborative techniques uses the ratings matrix created by us for the recommendation. It basically tries to reconstruct the input given to its visible layer. The hidden layer learns the parameter learned during training. These parameters during back propagation reconstruct the input. Thus, when we feed it the similar kind of item in our case with null values, It will use the hidden weights to predict the missing values. Thus, predicting the values for the missing values. The, recommendation obtained are then sorted in descending order to obtain the most relevant recommendations.
In our Hybrid model we have combined the RBM and KNN to create a hybrid recommender. We have given them the weightage of 0.6 and 0.4, which gets multiplied to the similarity score to obtain the results.

2. Background

Recommendation Systems have been widely used in many Internet activities and their importance is increasing due to the "Information Overload" problem arising from Internet. According to a paper published by the National Sun Yat-sen University, “recommendation is not a new phenomenon arising from the digital era, but an existing social behaviour in real life”. In everyday life, we rely on recommendations from others. A huge amount of information is available electronically; moreover, the World Wide Web is still growing faster; as a result, the users suffer from the "Information Overload" problem, when searching on Internet. For example, when we search an e-commerce website, to find an object, sometimes we found thousands of results. Historically, the first recommendation system was the Tapestry which coming out in 1992 from Xerox PARC, then a variety of techniques and technologies of recommendation systems have been introduced.

While studying researches in this domain, we realised that the availability of huge size of data about items the catalogue and the disinclination of users to rate items make a dispersed profile matrix leading to less accurate recommendations. The sparse rating in collaborative filtering systems makes it difficult to make accurate predictions about items. Collaborative filtering uses nearest neighbours to recommend items and fewer ratings make it computationally hard to calculate neighbours.

Recommendation systems that implement a content-based (CB) approach recommend items to a user that is similar to the ones the user preferred in the past. Recommendation systems that implement collaborative filtering (CF) predict users' preferences by analysing relationships between users and interdependencies among items; from these, they extrapolate new associations. Finally, hybrid approaches combine content-based and collaborative approaches, which have complementary strengths and weaknesses, thus producing stronger results.

Dataset

We are using the Book-Crossing Dataset. It contains 278,858 users with demographic information providing 1,149,780 ratings about 271,379 books.
Link for the dataset:

https://grouplens.org/datasets/book-crossing/

The above dataset contains 3 tables:

BX-Users : Contains information regarding the user such as

- User ID
- Location
- Age

BX-Books: Contains information regarding the user such as

- isbn
- book_title
- book_author
- year_of_publication
- publisher
- image_url_s
- image_url_m
- image_url_l

`book-title`, `book_author`, `year-of-publication`, `publisher` values are obtained from Amazon Web Services.

URLs linking to cover images are also given, appearing in three different flavours (`Image-URL-S`, `Image-URL-M`, `Image-URL-L`), i.e., small, medium, large. These URLs point to the Amazon web site.

BX-Book-Ratings: Contain book rating information such has

- user_id
- isbn
- book_rating(It is expressed on a scale from 1-10)

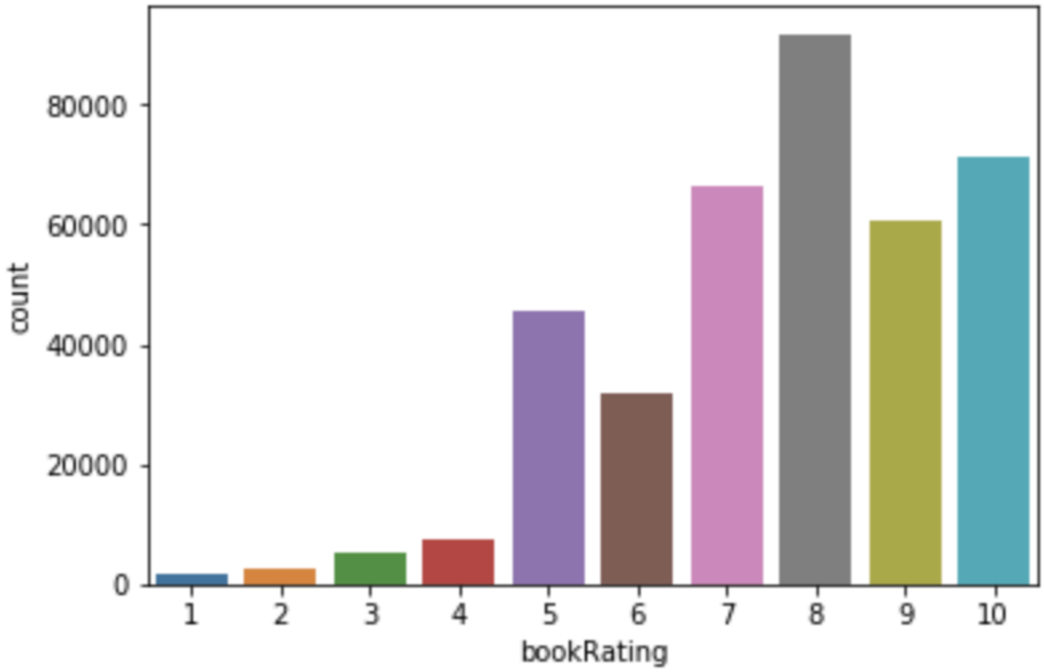

Figure 1: Distribution bookrating

3. Methods

User-Based Collaborative Filtering

In the user based collaborative filtering, we will create the matrix where users are represented by rows and books are represented as columns. The ratings are represented as there corresponding values. From the above created matrix we will find similar user based on the row wise cosine similarities of the user. The users who have most similar to the given users will be useful to generate the recommendations. Commonly used similarity measures are cosine, pearson, Euclidean etc. We have used cosine similarity. The cosine similarity is defined are shown below:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

In the code *findksimilarusers* function uses this method to return to return similarities and indices of k-nearest neighbors for active user. The function inputs userID and ratings matrix and returns similarities and indices of k similar users. The

predict_userbased predicts rating for specified user-item combination based on user-based approach.

recommend_userbased function uses above functions to recommend books for user-based approach.

The recommendations are computed as the weighted average of the deviations from the k nearest neighbors mean rating. These deviations are used to adjust the user based biases. These type of biases occur as users may tend to give high or low ratings to all items.

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u \in K} \left(r_{u,i} - \overline{r}_u\right) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

Where,

p (a,i) : the prediction for target or active user a for item i,

w (a,u) : the similarity between users a and u, and K is the neighborhood of most similar users.

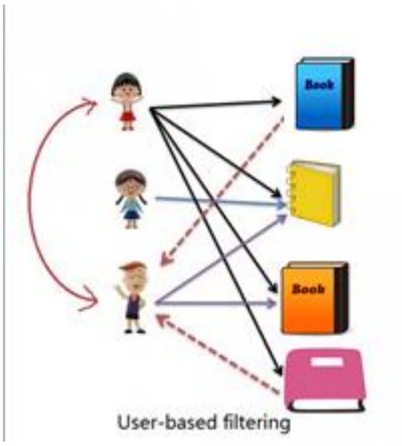

Figure 2: User Based Filtering

Item Based Collaborative Filtering

In this approach, the similarities between a pair of book items are computed using cosine similarity matrix between the columns of the ratings matrix that we created earlier. The rating of the target item i for a user. This can be predicted using the formula as follows:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

Where,

k : neighbourhood of most similar items rated by active user a

w (i,j) is the similarity between items *i* and *j*

The findksimilaritems function k nearest neighbor method employing cosine similarity to find k items similar to item i.

predict_itembased predicts rating for specified user-item combination based on item-based approach.

recommend_itembased function uses above functions to recommend books for item-based approach.

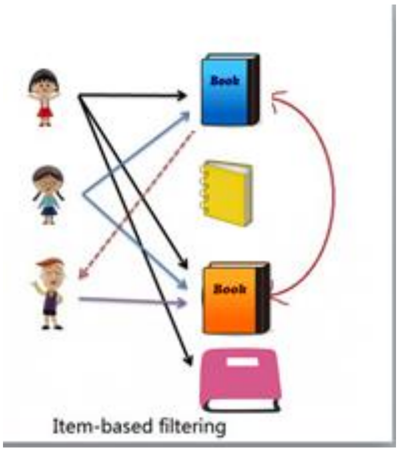

Figure 3: Item Based Filtering

Restricted Boltzmann Machines (RBM)

Considering we have M books, N users, and integer rating values from 1 to 10. The primary issue in applying RBMs to book ratings is dealing with the missing ratings. If all N users rated the same set of M books, we can treat each user as a single training case for an RBM which had M “softmax” visible units symmetrically connected to a set of binary hidden units. Each hidden unit can then learn to model a significant dependency between the ratings of different books. When most of the ratings are missing, we use a different RBM for each user (see the figure below). Every RBM has the same number of hidden units, but an RBM only has visible softmax units for the books rated by that user, so an RBM has few connections if that user rated few books. Each RBM only has a single training case, but all of the corresponding weights and biases are tied together, so if two users have rated the same book, their two RBM’s must use the same weights between the softmax visible unit for that book and the hidden units. The binary states of the hidden units, however, can be quite different for different users.

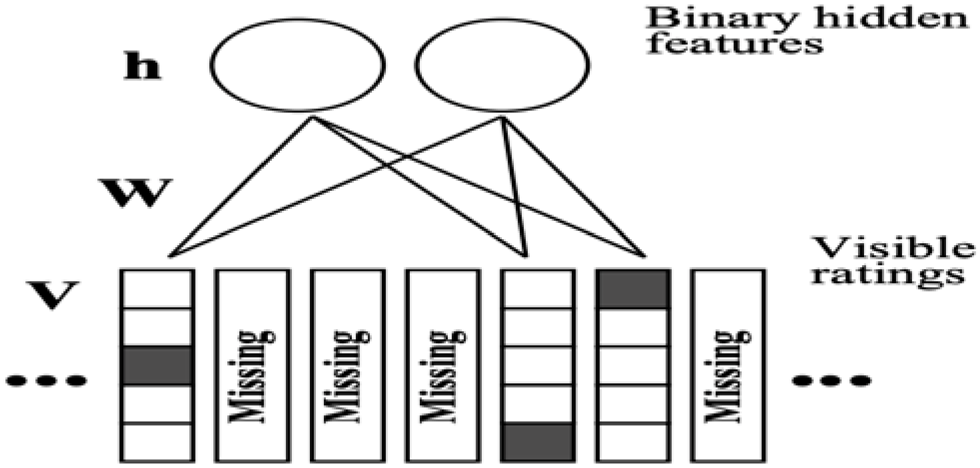

Figure 4: RBM model

<u>Hybrid</u>

To overcome the shortcomings of these individual methods, we decided to use a hybrid method to build a more efficient book recommendation system. We use the Collaborative Filtering and Restricted Boltzmann Machines in an environment where both these methods generate their individual recommendation scores. We have then combined both these scores by multiplying the recommendation scores obtained with appropriate weights. Then we will sort the recommendation score in the descending order to combine results from both the algorithms. Then we have concat the results to a single data frame and sorted them by the recommendation score in descending order.

4. Experiment

The framework for our project was built in python using pandas,numpy and sklearn for Collaborative filtering, We have used TensorFlow for implementing RBM and created hybrid recommender system based by combining the above mentioned two approaches.

Following are the recommendations for the users our algorithms generated. We have considered the user 2110 as our test subject and perform experiments to generate recommendation for the same. The user likes fantasy and fiction books. The kinds of recommendations are generated by each kind of recommendation techniques:

User Based Collaborative filtering:

The figure below shows the top 10 recommendations for user 2110 based on user-based CF approach.

```
knn.recommend_userbased(2110,ratings_matrix,'cosine',wt=1)
```

	ISBN	bookTitle	bookAuthor	yearOfPublication	publisher	RecommendationScore
35086	043965548X	Harry Potter and the Prisoner of Azkaban (Harry Potter)	J.K. Rowling	2004	Scholastic Paperbacks	6
30794	0525936831	The Forbidden Zone	Whitley Strieber	1993	Penguin USA	2
11296	9151830485	AprilhÃ¤xan	Majgull Axelsson	1997	RabÃ©n Prisma	2
30827	0064451178	How Do Apples Grow?	Betsy Maestro	1993	HarperTrophy	2
37726	3312008638	X = Liebe oder Ewig wÃ?Â¤hrt am lÃ?Â¤ngsten.	HansjÃ?Â¶rg Betschart	1999	Nagel & Kimche	2
30971	0843950323	Moon on the Water	Mort Castle	2002	Leisure Books	1
11481	0553283588	Juffie Kane	Beverly S. Martin	1990	Bantam Books	1
11829	0765304341	From a Whisper to a Scream (Key Books)	Charles de Lint	2003	Orb Books	1
11319	0865472807	Hard Laughter: A Novel	Anne Lamott	1987	North Point Press	1
54213	0345423089	Rookery Blues: A Novel	Jon Hassler	1998	Ballantine Books	1

Figure 5: Top 10 Book Recommendations for user 2110(User Based)

Item Based Collaborative filtering:

The figure below shows the top 10 recommendations for user 2110 based on item-based CF approach.

```
knn2.recommend(2110,ratings_matrix,'cosine',wt=1)
The Results using item based collaborative filtering are :
```

	ISBN	bookTitle	bookAuthor	yearOfPublication	publisher	RecommendationScore
20268	1569552924	Life-Giving Love : Embracing God's Beautiful Design for Marriage	Kimberly Hahn	2002	Charis Books	10
15158	0261103431	Tales from the Perilous Realm	J.R.R. Tolkien	1998	HarperCollins	10
15178	0836221362	It's A Magical World: A Calvin and Hobbes Collection	Bill Watterson	1996	Andrews McMeel Publishing	10
15177	0451627318	Child of the Dark: The Diary of Carolina Maria De Jesus	Carolina Maria De Jesus	1983	Signet Book	10
15176	0670894710	Bitten	Kelley Armstrong	2001	Viking Books	10
35845	0312865325	The Summoning God: Book II of the Anasazi Mysteries	Kathleen O'Neal Gear	2000	Forge	10
15170	0809468549	The Far Planets : Voyage Through the Universe	Time Life Books	1988	Time-Life Books	10
15169	0809468700	The Near Planets (Voyage Through the Universe)	Time Life	1989	Time-Life Books	10
15166	0773517634	War and Society in Europe 1870-1970 (War and European Society)	Brian Bond	1998	McGill-Queen's University Press	10
15165	0671638688	It's Always Something	Gilda Radner	1989	Simon & Schuster	10

Figure 6: Top 10 Book Recommendations for user 2110 (User Based)

RBM

The figure below shows the top 10 recommendations for user 2110 based on collaborative filtering using RBM approach.

```
rbm.recommend(4385,1)
```

	ISBN	bookTitle	bookAuthor	yearOfPublication	publisher	RecommendationScore
10345	0373194870	Hunter'S Vow (Silhouette Romance, No 1487)	Susan Meier	2000.0	Silhouette	0.137644
23462	0553563327	Thief of Hearts	TERESA MEDEIROS	1994.0	Bantam	0.102477
19630	0743250508	Fire Flight : A Novel (Nance, John J)	John Nance	2003.0	Simon & Schuster	0.100200
25396	0373226152	Conception Cover-Up (Top Secret Babies) (Harlequin Intrigue, No 615)	Karen Barrett	2001.0	Harlequin	0.091123
35086	8420482536	Amantes y Enemigos: Cuentos de Parejas	Rosa Montero	1998.0	Schoenhof Foreign Books Inc	0.088265
23356	0061010235	Veritas	William Lashner	1997.0	Regan Books	0.077272
11783	0553268074	Beet Queen	Louise Erdrich	1987.0	Bantam Doubleday Dell	0.076562
8160	0373031424	Lovespell (Harlequin Romance, No 3142)	Jennifer Taylor	1991.0	Harlequin	0.076209
23307	1551668955	Too Hot to Handle	Elizabeth Lowell	2002.0	Mira	0.074468
23350	1888451416	Southland	Nina Revoyr	2003.0	Akashic Books	0.073441

Figure 7: Top 10 Book Recommendations for user 2110 (RBM Approach)

Hybrid Recommendation System

The figure below shows the top 10 recommendations for user 2110 based on Hybrid approach.

:

	ISBN	bookTitle	bookAuthor	yearOfPublication	publisher	RecommendationScore
35086	043965548X	Harry Potter and the Prisoner of Azkaban (Harry Potter)	J.K. Rowling	2004.0	Scholastic Paperbacks	2.4
30794	0525936831	The Forbidden Zone	Whitley Strieber	1993.0	Penguin USA	0.8
37726	3312008638	X = Liebe oder Ewig wÃ?Â¤hrt am lÃ?Â¤ngsten.	HansjÃ?Â¶rg Betschart	1999.0	Nagel & Kimche	0.8
30827	0064451178	How Do Apples Grow?	Betsy Maestro	1993.0	HarperTrophy	0.8
11296	9151830485	AprilhÃ¤xan	Majgull Axelsson	1997.0	RabÃ©n Prisma	0.8
11319	0865472807	Hard Laughter: A Novel	Anne Lamott	1987.0	North Point Press	0.4
11829	0765304341	From a Whisper to a Scream (Key Books)	Charles de Lint	2003.0	Orb Books	0.4
11481	0553283588	Juffie Kane	Beverly S. Martin	1990.0	Bantam Books	0.4
30971	0843950323	Moon on the Water	Mort Castle	2002.0	Leisure Books	0.4
54213	0345423089	Rookery Blues: A Novel	Jon Hassler	1998.0	Ballantine Books	0.4

Figure 8: Top 10 Book Recommendations for user 2110 (Hybrid Recommendation)

As seen from the above experiment we can notice that hybrid recommender generates the more diverse set of results compared to other algorithms. The item based recommendations are better compared to user based recommendations. This is because of the sparsity of the data for the user-based CF.

5. Conclusion

We have learnt the fundamentals of Recommender Systems. The types of recommender systems and how to apply various recommendation techniques for content based and collaborative filtering based recommender systems. We build our recommender system by applying collaborative filtering using kNN algorithm for item based and user based collaborative filtering techniques. The novel part of our project was to use Restricted Boltzmann Machine for creating a collaborative filtering recommender system. We also built our hybrid recommender system which used both the RBM based and KNN based collaborative filtering techniques.

6. Response to the feedback

The following is our response to the feedback

What are you doing different from traditional approach?
In our Hybrid model we have combined the RBM and KNN to create a hybrid recommender. We have given them the weightage of 0.6 and 0.4, which gets multiplied to the similarity score to obtain the results.

Do you have plans of creating a repository?

We have created a github repository: https://github.com/KuldeepSinh24/Book-Recomendation-System

Have you cited the RBM paper ? You have not mentioned why you adopt this model? Any reasoning of yours or the authors?

We have cited the paper.
Restricted Boltzmann Machines for Collaborative Filtering

The authors have used the RBM based collaborative filtering to tackle the problem of generating recommendations on large dataset. Since, our dataset was large we have considered this approach to check how it performs on our dataset.

7. References

[1]Restricted Boltzmann Machines for Collaborative Filtering
[2]https://towardsdatascience.com/my-journey-to-building-book-recommendation-system-5ec959c41847
[3]https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-the-fundamentals-dfee054f978e
[4]https://cambridgespark.com/content/tutorials/implementing-your-own-recommender-systems-in-Python/index.html
[5]Item-Based Collaborative Filtering Recommendation Algorithms
[6]https://ieeexplore.ieee.org/document/7684166
[7]Understanding Basics of Recommendation Engines
[8] https://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf
[9]Tseng. C. (2002). Cluster-based Collaborative Filtering Recommendation Approach. Master's Thesis, Information Management department, National Sun Yat-sen University. Taiwan. July 22th.