# CUSTOMIZED BOOK RECOMMENDATION SYSTEM

[1]Ashish Kamble, [2]Nitesh Gawade, [3]Subhash Bhoir, [4]Abhay Patil
[1]Student BE, [2] Student BE, [3] Student BE, [4] Mentor
[1]Department of Information Technology,
[1]MCT's Rajiv Gandhi Institute of Technology, Mumbai, India

***Abstract:*** Recommender systems are found in many ecommerce applications today. Recommender systems customarily provide the utilizer with a list of recommendations that they might prefer, or supply prognostications on how much the utilizer might prefer each item. Culling what book to read next has always been a question for many. Even for students, deciding which textbook or reference book to read on a topic unknown to them is an immensely colossal question. There are two prevalent approaches for providing recommendations, they are collaborative filtering and content predicated filtering. In this report we endeavor to present a model for a personalized recommendation system for books that utilizes hybrid recommendation approach which is cumulation of content predicated and collaborative filtering. The proposed recommendation system endeavors to learn the user's predilections and recommends the books to the utilizer predicated on their predilections. The system withal recommends the books to the utilizer predicated on the user's demographic parameters like age and location. The system withal endeavors to understand the user's favorite author and recommends accordingly.

***Index Terms*** **- Recommendation System, Collaborative filtering, Content-based filtering, Demographic parameters.**

## I. INTRODUCTION

Recommender systems have propagated from the last decennium. Since the number of products has grown in number, the desideratum for recommender systems has additionally incremented. Recommendation system endeavors to prognosticate the interest of a utilizer and recommend products that match their interest as accurately as possible. Withal, e-commerce business will be profited by the incrementation of sales which will conspicuously occur when the utilizer is presented with more items that he/she would likely find to match the interest. It simplifies the user's job thus making it benign for both the utilizer as well as the dealer. They withal avail the users to manage their reading list by kenning their predilection. Recommendation system endeavors to prognosticate the interest of a utilizer and recommend products that match their interest as accurately as possible. Withal, e-commerce business will be profited by the incrementation of sales which will conspicuously occur when the utilizer is presented with more items that he/she would likely find to match the interest. There are fundamentally two approaches for Recommendation System:

Collaborative filtering: Collaborative filtering (CF) is the process of filtering or evaluating items through the opinions of other people. CF technology brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data [1].

Content-based filtering: Content-based filtering recommender systems recommend items based on the textual information of an item, under the assumption that users will like similar items to the ones they liked before [2].

A recommender platform with the combination of all these techniques is also possible. It is called a Hybrid Recommender System. Different techniques have been developed over time to give accurate recommendations. Apart from the regular filtering techniques, other approaches are being adopted. Ontology-based recommendations, Demographic-based recommendations have gained importance in recent times. Natural Language Processing is also being incorporated nowadays to analyze user feedback. Context aware recommendations too are gaining popularity [3].

## II. EXITING SYSTEM

There are book recommendation systems already exists which uses collaborative or content-based filtering but there are very few book recommendation systems which use hybrid recommendation system. Firstly, we should know about issues and challenges faced by recommendation system:

1. Cold-start problem:

It's arduous to give recommendations to incipient users as his profile is virtually empty and he hasn't rated any items yet so his taste is unknown to the system. This is called the cold start quandary. In some recommender systems this quandary is solved with survey when engendering a profile. Items can additionally have a cold-start when they are incipient in the system and haven't been rated afore. Both of these quandaries can be withal solved with hybrid approaches.

2. Scalability:

The voices of individuals with a brief history might not be that relevant because the voices of these who have rich history in their profiles. the difficulty of trust arises towards evaluations of a particular customer. the matter might be solved by distribution of priorities to the users.

3. Sparsity:

In online shops that have a plethora of users and items there are virtually always users that have rated just a few items. Utilizing collaborative and other approaches recommender systems generally engender neighborhoods of users utilizing their profiles. If a utilizer has evaluated just few items then its pretty arduous to determine his taste and he/she could be cognate to the erroneous neighborhood. Sparsity is the quandary of lack of information.

4. Privacy:
Privacy has been the most consequential quandary. In order to receive the most precise and redress recommendation, the system must acquire the most quantity of information possible about the utilizer, including demographic data, and data about the location of a particular utilizer. Naturally, the question of reliability, security and confidentiality of the given information arises. Many online shops offer efficacious aegis of privacy of the users by utilizing specialized algorithms and programs.
5. Changing User Preference:
The voices of individuals with a brief history might not be that relevant because the voices of these who have rich history in their profiles. the difficulty of trust arises towards evaluations of a particular customer. the matter might be solved by distribution of priorities to the users.

## III. LITERATURE REVIEW

The literature survey on the subsisting recommendation system for books also because the literature survey on the advice system for other applications are as follows:
1. Recommendation System and its Approaches:
Mukta Kohar, Chhavi Rana in their paper have described what's meant by recommendation system, the varied approaches like content-based, collaborative filtering and hybrid approaches that are involved for creating recommendations[4] Lipi Shah, HetalGaudani, Prem Balani in their work have described different techniques involved in content-based , collaborative filtering and hybrid approach and therefore the issues related in each of the category[5].

2. Problems related to Recommendation System:
Soanpet. Sree Lakshmi, Dr. T. Adi Lakshmi in their work have described intimately the issues like overspecialization, data sparsity, cold start, scalability, ranking of the advice, etc. that are associated with the advice system [6].

3. Existing Recommendation system for books:
S. Vinodhini, V. Rajalakshmi, B. Govindarajalu proposed a customized recommendation system which recommends books to the user supported the keyword extraction [7] using hadoop and therefore the ratings given by user. They also implemented Region Aggregation will recommending the book.
Pijitra Jomsri developed a library book recommendation system supported user profile loaning and apply association rule to make model [8].
Salil Kanetkar, Akshay Nayak, Sridhar Swamy, Gresha Bhatia proposed web-based personalized hybrid book recommender system [9] which exploits varied aspects of giving recommendations aside from the regular collaborative and content-based filtering approaches. Temporal aspects for the recommendations are incorporated. Also, for users of various age, gender and country, personalized recommendations are often made on these demographic parameters. Scraping information from the online and using the knowledge obtained from this process are often equally useful in making recommendations.
Punit Gupta and Shankar proposed a tagging [10] based evolving recommendation system for digital library system for user and library administration.
Guangqian Zhangl, Wei Sunl they performed a knowledge analysis so as to explore the importance of various attributes [11] within the user preference to a book and provided the suggestion for book recommendation design.
CaiNicolas Ziegler et al (2005) proposed a recommendation system that considers an idea called topic diversification [12]. consistent with this idea, the list of top n recommendation is going to be balanced because the users extended interest also will be taken under consideration. Thus, the user won't be bored upon the similar quite recommendations often made. The concept of User-based Collaborative filtering and Item-based Collaborative filtering are combined and therefore the recommendations are made.

4. Other Recommendation system:
1. Recommendation system for recommending courses to students:
Boban Vesin et al (2012) developed a recommendation system termed as PROTUS (PRogramming TUtoring System) [13] that recommended courses to the scholars. The courses are usually recommended to the scholars supported their age and domain of study but during this system semantic web technology concepts are used. Navigation patterns are obtained from the past history of the scholar and from that pattern, future recommendations are made.

2. Personalized online news recommendation system:
Saranya.K.G and G.Sudha Sadhasivam, proposed personalized news recommendation approach supported dynamic updating policy and collaborative filtering[14]. This paper also solves the matter of scalability asscociated with personalized news recommendation system with the assistance of hadoop framework.

3. Recommendation system for music:
Brian McFee et al (2012) developed a recommendation system for music by learning the content similarity [15]. It used content-based similarity method initially then collaborative similarity method is imposed on the results. It avoided the cold start problem and therefore the overhead of query-to-answer technique.

4. Recommendation for fashion Company Lyst [16]:
Maciej Kula in his work presents a hybrid matrix factorization model and proves that this model outperforms both collaborative and content-based models in cold-start or sparse interaction data scenarios. He compares all the hybrid algorithm with lightfm algorithm and shows the lightfm algorithm outperforms the remainder of the algorithm.

## IV. IMPLEMENTATION

### 1.Data and Sources of Data

This dataset contains ratings for ten thousand popular books. on the source, for instance that these ratings were found on the web. Generally, there are 100 reviews for every book, although some have less - fewer - ratings. Ratings go from 1 to 5. Both book IDs and user IDs are contiguous. For books, they're 1-10000, for users, 1-53424. All users have made a minimum of two ratings. Median number of ratings per user is 8. There also are books marked to read by the users, book metadata (author, year, etc.) and tags.

ratings.csv contains ratings sorted by time. it's 69MB and ratings go from 1 to 5 . Both book IDs and user IDs are contiguous. For books, they're 1-10000, for users, 1-53424. Other file to_read.csv provides IDs of the books marked “to read” by each user, as user_id,book_id pairs, sorted by time. There are on the brink of 1,000,000 pairs. Another file books.csv has metadata for every book (goodreads IDs, authors, title, average rating, etc.File book_tags.csv contains tags/shelves/genres assigned to books by users. Tags during this file are represented by their IDs. they're sorted by goodreads_book_id ascending and count descending. tags.csv translates tag IDs to names.

All files are available at https://github.com/zygmuntz/goodbooks-10k.

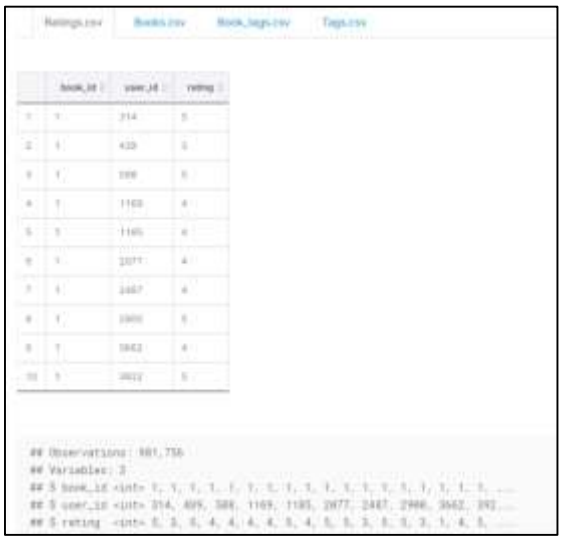


Fig 1 Rating.csv

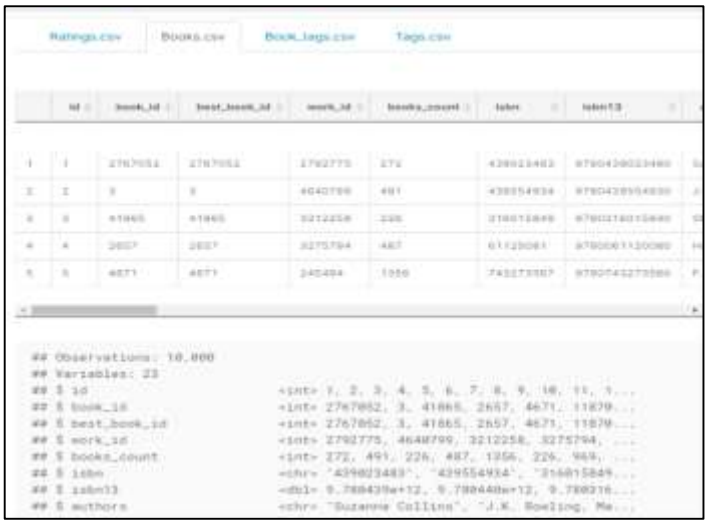
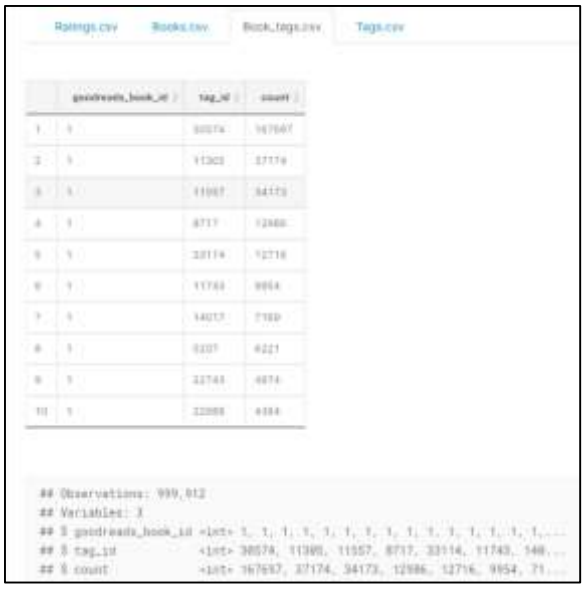
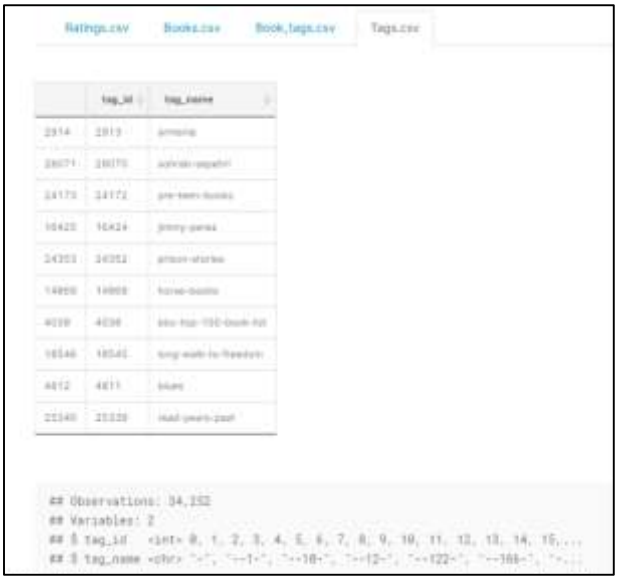


Fig 3 Book_tags.csv





**2. Cleaning the Dataset**

As with nearly any real-life dataset, we'd like to try to to some cleaning first. When exploring the information, we noticed that for a few combinations of user and book there are multiple ratings, while in theory there should only be one (unless users can rate a book several times). Furthermore, for the collaborative filtering partially II it's better to possess more ratings per user. So i made a decision to get rid of users who have rated fewer than 3 books. The data contains nearly 1mio rows, so for this step I found data to be significantly faster than dplyr. If you've got not yet tried it out, i like to recommend you to try to to so. It helped me tons e.g., within the Instacart competition. However, within the remainder of this kernel I'll attempt to use dplyr, as we personally find it easier to read. So, we first removed the duplicate ratings.

As we filtered our ratings all users have a minimum of 3 ratings. However, we will also see that are some users with many ratings. this is often interesting, because we will later examine whether frequent raters rate books differently from less frequent raters. we'll come to the present later. People have different tendencies to rate books. Some already give 5 stars to a mediocre book, while others don't give 5 stars unless it's the right book for them. Such tendencies are often seen within the figure below. On the proper side there's a bump from users with a mean rating of 5, indicating that they really liked all books (or they only rated books they really like…). we will also see that there are nearly no notoriuous downvoters rating all books with a 1. Such tendencies are getting to be important for collaborative filtering later, and are typically addressed by subtracting the user's mean rating from their ratings.Extracting the genres of the books isn't trivial since users assign self-chosen tags to books, which can or might not be an equivalent as genres defined by goodreads. As a practical way I chose only the tags the match those provided by goodbooks. this might be improved by grouping similar tags together (like 'self-help', 'self help' etc. to 'Self Help'). But i feel my approach is ok for a primary glance.We see that the majority books are "Drama", "Biography", or "Science Fiction" books, while there aren't very many "Sports" within the database.

You might have seen within the books.csv that there's language information on the books. this is often interesting because GoodReads is an English-speaking site. However, the dataset contains some books in several languages. the rationale is that typically there are multiple editions of a book (both within the same language and in several languages). For this dataset, it seems that the foremost popular edition was included, which for a few books is their original language.

Next, we will see, whether we will find any associations of features with a book's rating. For a fast look, let's first plot the matrix between the books average_rating and a few variables. In summary, we see only small correlations between the features and therefore the average rating (last row), indicating that there are not any strong relationships between the rating a book receives and meta-variables (like rating counts etc.). this suggests that the rating depends more strongly on other features (e.g. the standard of the books itself).

Theoretically, it'd be that the recognition of a book (in terms of the amount of ratings it receives) is related to the typical rating it receives, such once a book is becoming popular it gets better ratings. However, our data shows that this is often true only to a really small extent. The correlation between these variables is merely 0.045. The dataset contains information about what percentage editions of a book are available in book_count. These can either vary editions within the same language or also translations of the book into different languages. So one might assume, that the higher the book is that the more editions should be available. In fact, data show precisely the opposite pattern: The more editions a book has the lower is that the average rating. The causal direction of this association is in fact unclear here.

**3. Creating Recommender System**

Recommenderlab may be a R-package that gives the infrastructure to gauge and compare several collaborative-filtering algorithms. Many algorithms are already implemented within the package, and that we can use the available ones to save lots of some coding effort, or add custom algorithms and use the infrastructure (e.g. cross validation). There is a crucial aspect concerning the representation of our rating matrix. As we could already see above, most of the values within the rating matrix are missing, because every user just rated a couple of of the 10000 books. this enables us to represent this matrix is sparse format so as to save lots of memory. Creating predictions is then also straight-forward. you only call predict() and pass the model, the ratings for the user you would like to predict ratings for, and a parameter to inform the function that you simply want to urge predicted ratings back.

The good thing about Recommenderlab is that it offers the likelihood to simply evaluate and compare algorithms. so as to try to so, one first has got to create an evaluation scheme. Here, as an illustration I chose to try to to 10-fold crossvalidation. The given parameter determines what percentage ratings are given to make the predictions and successively on what percentage predictions per user remain for the evaluation of the prediction. during this case -1 means the predictions are calculated from about 1 rating, and performance is evaluated for 1 for every user. In a second step, we will list all the algorithms we would like to match. As we've a tuneable parameter nn, which is that the number of most similar users which are wont to calculate the predictions. I vary this parameter from 5 to 50 and plot the RMSE. Furthermore, as a baseline one also can add an algorithm ("RANDOM") that randomly predicts a rating for every user.

Here, the system learns the probability of two or more products being bought together. for instance, when a user buys a smartphone from an ecommerce store, it's more probable that an equivalent user will buy a group of headphones on a return visit, instead of another smartphone. As such, the algorithms are based around recommending products that are complementary to other products – they're product-defined, as against user-defined, as in CBF and CF. The Naïve Bayes algorithm is most ordinarily utilized in complementary filtering. Hybrid approaches essentially work by combining CBF and CF methods. this will be achieved during a number of the way – for instance, by making content-based and collaborative-based predictions separately then combining them, by adding collaborative-based capabilities to a content-based approach (and vice versa), or by purposefully unifying the 2 approaches into one model.

The actual building and deployment of the app are very easy also. Actually you only got to create two files ui.R and server.R. ui.R contains the static UI interface, which is essentially a dashboard with two big boxes (one for the user ratings and one for the recommendations).server.R firstly populates the user rating box with books which may be rated, secondly calculates the recommendations when the user hits the button, and thirdly populates the box for the recommendations. Finally, deploying the app may be a matter of seconds. Just create an account on shinyapps.io and tell Rstudio to publish your app.

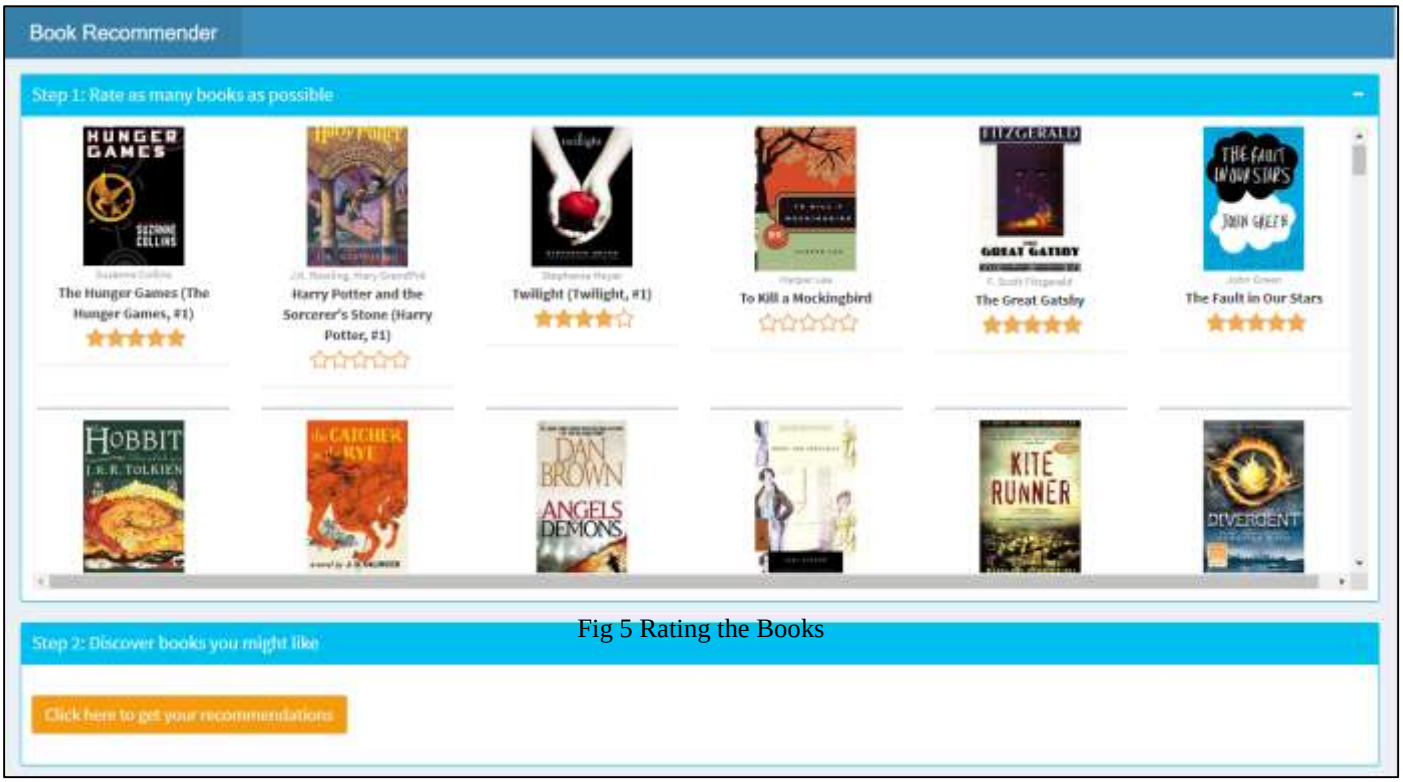



Fig 5 Rating the Books

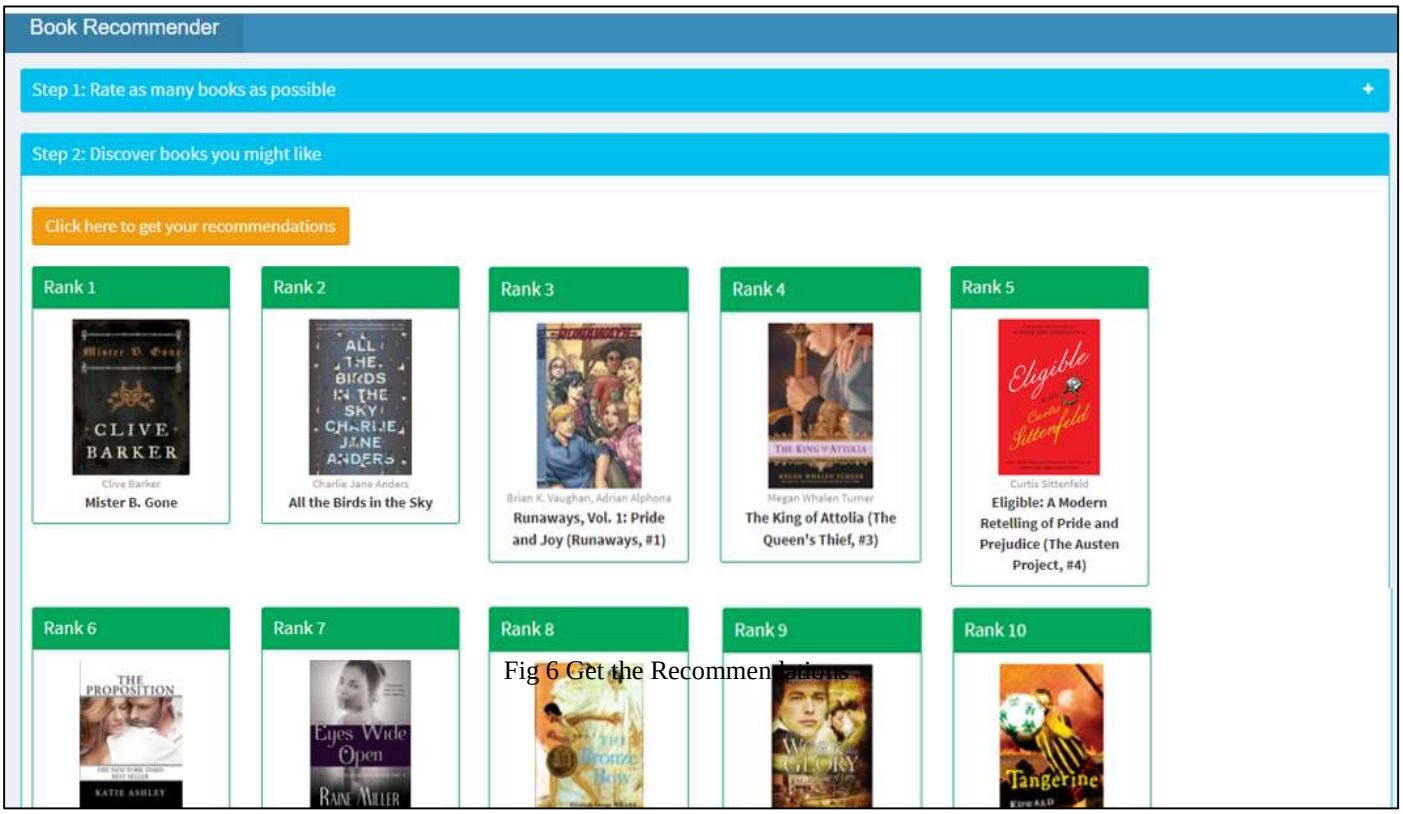



Fig 6 Get the Recommendations

## V. CONCLUSION

This system provides personalized recommendation of books to the user. this system considers big data of books. The system makes use of both content-based and collaborative filtering algorithm so as reduce the cold start problem and provides the user with recommendation list. The system tries to predict the ranking by considering the item's similarity also as user's similarity in order that a user can get recommendations of latest books

## REFERENCES


**[1]** P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS 4321, pp. 291 – 324, 2007.

**[2]** M. A. Ghazanfar and A. Prugel-Bennett, "A Scalable, Accurate Hybrid Recommender System," 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket, 2010, pp. 94-98.

**[3]** Salil Kanetkar, Akshay Nayak, Sridhar Swamy, Gresha Bhatia," Web-based Personalized Hybrid Book Recommendation System ", IEEE, pp. 978-1-4799-63935/14,2014.

**[4]** Survey Paper on Recommendation System, Mukta koharet al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2), 2012,3460-3462

**[5]** Survey on Recommendation System, Lipi Shah, Hetal Gaudani, Prem Balani, International Journal of Computer Applications (0975 – 8887) Volume 137 – No.7, March 2016

**[6]** Recommendation Systems:Issues and challenges, Soanpet .Sree Lakshmi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5771-5772

**[7]** S. Vinodhini, V. Rajalakshmi, B. Govindarajalu," Building Personalised Recommendation System with Big Data and Hadoop Mapreduce ", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181. ***3*** Page 21153

**[8]** Pijitra Jomsri," Book Recommendation System for Digital Library Based on User Profiles by Using Association Rule", IEEE,pp. 978-1-4799-4233-6/14,2014.

**[9]** Salil Kanetkar, Akshay Nayak, Sridhar Swamy, Gresha Bhatia," Web-based Personalized Hybrid Book Recommendation System ", IEEE, pp. 978-1-4799-6393- 5/14,2014.

**[10]** Punit Gupta, Ravi Shankar, "Tagging based evolving Recommendation System for digital library system",2015 4th International Symposium on emerging Trends and Technologies in libraries and information services.

**[11]** Guangqian Zhangl, Wei Sunl, "User Preferences to Attributes of Books for Personalized Recommendation", IEEE pp 978-1-4673-2008-5/12,2012.

**[12]** CaiNicolas Ziegle.R, Sean M. McNee., Joseph A. Konstan and Georg Lausen (2005), „Improving Recommendation Lists Through Topic Diversification", International World Wide Web Conference Committee (IW3C2), ACM, pp. 5959-30- 469.

**[13]** Boban Vesin., Mirjana Ivanovic., Aleksandra Klasnja-Milic and Zoran Budimac (2012), „Ontology-based semantic recommendation in programming tutoring system", Journal on expert systems with applications, Vol. 39, pp 1229-12246.

**[14]** Saranya.K.G, G.Sudha Sadhasivam," A Personalized Online News Recommendation System ", International Journal of Computer Applications (0975 – 8887) Volume 57– No.18, November 2012.

**[15]** Brian McFee, Luke Barrington and Gert Lanckriet, "Learning Content Similarity for Music Recommendation" IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 8, 2012.

**[16]** Metadata Embeddings for User and Item Cold-start Recommendations, Maciej Kula, arXiv:1507.08439v1[cs.IR] 30 Jul 2015.

**[17]** https://github.com/zygmuntz/goodbooks-10k.

**[18]** Message Clustering based Matrix Factorization Model for Retweeting Behaviour Prediction, Bo Jiang, Jiguang Liang, Ying Sha, Lihong Wang, ACM 978-1-4503-3794-6/15/10