# PROACTIVE UNIVERSITY LIBRARY BOOK RECOMMENDER SYSTEM

Dissertation Submitted in Fulfillment of the Requirements for the Degree of

MAGISTER TECHNOLOGIAE:

In the

Department of Information and Communication Technology, Faculty of Applied and Computer Sciences, Vaal University of Technology

By

Tadesse Zewdu Mekonnen

217236170

**Supervisor:** Prof Tranos Zuva

**March 2021**

# DECLARATION

I hereby declare that this dissertation, which I submit for the qualification of **Magister Technologiae in Information Technology** to the Vaal University of Technology; Department of Information and Communications Technology, Faculty of Applied and Computer Sciences, apart from the recognized assistance of my supervisor and provided citations, is my own work and has not previously been submitted to any other institution for any degree.

Tadesse Zewdu Mekonnen ________ on this _______ day of ______________________
Candidate

Prof Tranos Zuva ______________ on this _________ day of ______________________
Supervisor

# ACKNOWLEDGEMENTS

I would like thank the almighty God the father for his presence in my work from the start to the end. Secondly, I would like to express my gratitude to my supervisor, Prof Tranos Zuva, for the useful comments, remarks and full guidance through the entire process of this Master's thesis. I would like to thank Prof Tranos Zuva for his unchanging positive attitude, for his fatherly approach and for his supportive ideas. Furthermore, I would like to thank my mother for her advice and encouragement. In addition, I owe my friends special thanks for encouraging and supporting me. I will forever be grateful for your support.

# ABSTRACT

Too many options on the internet are the reason for the information overload problem to obtain relevant information. A recommender system is a technique that filters information from large sets of data and recommends the most relevant ones based on people's preferences. Collaborative and content-based techniques are the core techniques used to implement a recommender system. A combined use of both collaborative and content-based techniques called hybrid techniques provide relatively good recommendations by avoiding common problems arising from each technique. In this research, a proactive University Library Book Recommender System has been proposed in which hybrid filtering is used for enhanced and more accurate recommendations. The prototype designed was able to recommend the highest ten books for each user. We evaluated the accuracy of the results using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). A measure value of 0.84904 MAE and 0.9579 RMSE found by our system shows that the combined use of both techniques gives an improved prediction accuracy for the University Library Book Recommender System.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

The following abbreviations are used in this study:

| | |
|---|---|
| MAE | Mean Absolute Error |
| RMSE | Root Mean Squared Error |
| EM | Expectation–maximization |
| CF | Collaborative Filtering |
| CBF | Content-Based Filtering |
| HF | Hybrid Filtering |
| KBF | Knowledge-Based Filtering |
| DF | Demographic Filtering |
| QoS | Quality of Service |
| BUS | Book Utilization System |
| XML | Extensible Markup Language |
| CBR | Case-based reasoning |
| SOM | Self-Organizing Map |
| GA | Genetic Algorithm |
| SPA | Pattern Analysis |
| PBL | Problem-Based Learning |
| TF | Term frequency |
| IDF | Inverse document frequency |

| | |
|---|---|
| DF | document frequency |
| LV | Length of Vector |
| NV | Normalized Vector |

## PUBLICATION

The following publication was extracted from this study:

T.Z. Mekonnen and T. Zuva (2020). A Proactive University Library Book Recommender System. 4th Computational Methods in Systems and Software 2020 in the Springer Series: Advances in Intelligent Systems and Computing, Vol.1294. Springer, Cham. ISBN: 978-3-030-63322-6, pp. 327-335, https://doi.org/10.1007/978-3-030-63322-6_26.

# CHAPTER 1: INTRODUCTION

## 1.1 BACKGROUND

The many options on the internet cause information overload. Information overload makes it difficult for many internet users to obtain relevant information on time. It is necessary to filter, prioritize and recommend relevant information to solve this problem. Even if search engines such as Google and Yahoo have solved some of the problems, prioritization and personalization are still needed to get more accurate and relevant information to individuals. Therefore, there is a need for more research in this area. This has increased the demand for recommender systems.

Recommender systems are machine-learning algorithms that have the ability to predict users' preferences based on their profiles. By taking a user's online experiences and input set, the recommender system generates a probable recommendation for the user. It provides users with a prediction closer to reality. Companies like Amazon.com use the recommender system for recommending products; Facebook for recommending people you may know and Netflix for recommending movies and TV shows to watch (Taheri et al., 2017).

Although there are few book recommender systems on the web such as  Radgeek, What should I read next, Bookish, Jelly books, My independent bookshop, Shelfari, Librarything, Amazon, Goodreads and Getglue, all of them recommend books for commercial activities. Similar systems of recommendation of books can be helpful if applied for university libraries.

Different libraries in Gauteng Province in South Africa have well-built automated library systems which are very helpful for library users to get information about books in the library. These libraries use search engines to access books using keywords and users depend on the search engine to retrieve items (or books) in the collection. Library users have to try different keywords repeatedly to get what they need in the library store until the relevant items are found. Even if it is very helpful for the users of a library to get different books, only using a search engine is not enough for users to find all books they want in a university library efficiently. Library users need a better assistant to access the books in which they may be interested. Therefore, there is a need for an optimum recommender system for university library books. The recommender system assists by increasing the visibility of available books (Tewari & Priyanka, 2014).

This study proposes a proactive University Library Book Recommender System, which provides library book recommendations based on a hybrid of collaborative and content-based filtering techniques. A proactive book recommendation system decides which available book is most likely to be relevant to the user after it retrieves large quantities of books and it gives recommendations without the user's request.

The existence of too many different books in one small library building results in increasing difficulty for the users in searching and finding what they want in a manner which best meets their requirements. The increase in the availability of digital information causes information overload. Users may not have the title and the key words of all books required for their studies due to an information gap. Even though library users may have a title of a book which they need for their studies, the other related books which are probably more important for their studies will remain undiscovered while the books are still available in the library (Isinkayea, Folajimib, & Ojokohc, 2015).

A proactive university library book recommender system is a system which recommends books without users request that are available in the university library.

The proposed scheme was useful in the management of existing book collection and document mining methods to address information overload problem. More precisely, it helped the users to pick a suitable collection of books in a library which contains the topics of users' interest.

## 1.2 PROBLEM STATEMENT

Information overloads, which are caused by the quick increments in accessibility of computerized data and numbers of web users, make it difficult to get items of interest on the internet.

The library management system (Sierra) in Vaal University of technology has a collection of 71,013 printed books and 64 electronic databases as of June 2017. However, only a small number of books are borrowed which means that many books are not borrowed at least once and are thus probably not known by readers. A recommender system is required to help readers in the university library to find what they want, especially for those users who do not know the books titles yet but may be interested in such books. This problem is referred to as low penetration rate. New books available in the library will stay unreachable because there is no way of informing users about and recommending the books for users of the library. As a result, users hardly get the required and related books. Forgotten old books that probably are very important for readers are also unreachable because the users are not able to remember all the old books.

The existence of too many different books in one small library building results in increasing difficulty for the users in searching and finding what they want in a manner that best meets their requirements. Users may not have the title and the key words of all books required for their studies due to an information gap.

## 1.3 RESEARCH QUESTIONS

- According to the literature, what has been done to recommend suitable and appropriate books to library users?
- How does one propose a proactive library recommender system for library users?
- How does one develop a proactive library recommender system prototype?
- How does one evaluate a proactive library recommender system?

## 1.4 AIM AND OBJECTIVES

The main aim of this dissertation is to recommend proactively library books to students and staff.

Objectives of the study are:

1. To investigate the literature to find out what has been done regarding recommender systems.
2. To propose a proactive university library book recommender system.
3. To develop a prototype for a university library recommender system
4. To evaluate the developed prototype using an algorithm

## 1.5 JUSTIFICATION FOR THE STUDY

In a proactive university library book recommender system different new and old related books are proactively recommended to the users according to the users' choice. The users' interests and the users' patterns of searching different books are categorized. Related books of the users' search and the most rated or liked books by other users and new books are then recommended to the users.

A proactive university library book recommender system is very helpful for the university library users by saving the time by searching for books and discovering books which users are not aware of. The system adds value to the library system by increasing the visibility and availability of books in the library.

## 1.6 SCOPE OF THE RESEARCH AND ITS LIMITATIONS

The system has a platform for login and search for different books. The users are proactively recommended to available and corresponding choice of books based on the system prediction. The system is applicable only in the university library and only registered users are able to access the system.

## 1.7 METHODOLOGY

### 1.7.1 DATA COLLECTION

Data collection was done by a literature search where the main sources of information were journal articles, conference papers and books dwelling on the use of recommender systems.

### 1.7.2 RESEARCH METHODS

The research study used a hybrid filtering approach, which encompassed both a collaborative filtering technique and a content-based filtering technique as shown in figure 1.0. The use of the combination of both techniques helped to reduce the drawbacks of both techniques.

The system computes the similarity between different library book users and then predicts probable ratings for unrated books by the users, which enables the system to give a good recommendation for users.

#### *1.7.2.1 Hybrid Filtering Method*

The hybrid filtering method is a combined method of collaborative and content-based filtering. This method was used to make a more accurate recommendation by avoiding common problems like cold start from the collaborative filtering method and lack of information about the book from other users' ratings and comments from the content-based filtering method.

#### *1.7.2.2 Collaborative Filtering Method*

The collaborative filtering method is a method to filter and predict the users' preferences by comparing them with other users' opinions or preferences (Ghadling, Belavad, Bhegade, Ghojage, & Supriya, 2015).

In this algorithm, different vectors are calculated by using similarity measures. These similarity measures are used to predict ratings. This can be done through finding the nearest neighbors and then recommending items. Pearson correlation is used to determine users' similarity.

$$\mathrm{sim}_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i}-\bar{r}_a)\times(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i}-\bar{r}_a)^2(r_{u,i}-\bar{r}_u)^2}} \tag{1.1}$$

Where $r_{a,i}$ is the rating given to item i by user a; $\bar{r}_a$ is the mean rating given by user a; and m is the total number of items.

Predictions are computed as the weighted average of deviations from the neighbor mean:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i}-\bar{r}_u)\times \mathrm{sim}_{a,u}}{\sum_{u=1}^{n} p_{a,u}} \tag{1.2}$$

Where $p_{a,i}$ is the prediction for the active user 'a' for item 'i'; $p_{a,u}$ is the similarity between users 'a' and 'u'; and 'n' is the number of users in the neighborhood. (Melville, Raymond , & Nagarajan, 2002).

*1.7.2.3 Content-based Method*

The content-based recommender method analyzes item descriptions to identify items that are of particular interest to the user (Pazzani & Billsus, Content-Based Recommendation Systems, 2007). The Cosine similarity formula is used.

$$u(c,s) = \cos(\overline{w}_c, \overline{w}_s) = \frac{(\overline{w}_c.\overline{w}_s)}{|\overline{w}_c|\times|\overline{w}_s|} = \frac{\sum_{i=1}^{k} w_{i,c}w_{i,s}}{\sqrt{\sum_{i=1}^{k} w^2{}_{i,c}\sqrt{\sum_{i=1}^{k} w^2{}_{i,s}}}} \tag{1.3}$$

Where 'c' is content base profile and 's' is content (Li & Han, 2013).

The system sorts items by the utility values and recommends other items to the user that have a degree of similarity to the user's profile.

## 1.8 RECOMMENDATION

The top ten (10) similar books (obtained from content-based method) with the highest predicted ratings (from collaborative method) were recommended to the users.

The top ten (10) similar books (obtained from the content-based method) with the highest predicted ratings (from collaborative method) are recommended to the users.

## 1.9 EVALUATION MEASURE

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used to evaluate the accuracy of the recommendation system.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are chosen over other methods because they are able to find the exact value of the difference between actual ratings given by users and estimated ratings.

MAE measures the average of the absolute deviance between the actual rating given by the users and the predicted rating. Mean Absolute Error (MAE) is calculated using equation 4.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} p_{i-q_i}}{n} \qquad (1.4)$$

where $p_1 \ldots . p_n$ are predicated ratings; $q_1 \ldots q_n$ are actual ratings and n is amount of ratings.

The lower the MAE, the more accurate the prediction is (Parvatikar & Bharti, 2015).

RMSE measures the average size of the error by calculating and finding the square root of the average of the squared differences between the prediction given by the prototype and the actual rating given by the user. RMSE gives a relatively bigger weight amount to large errors because the errors are squared before they are averaged. RMSE is calculated using equation 3.15

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(p_{i-q_i}\right)^2}{n}} \qquad (1.5)$$

where $p_1 \ldots . p_n$ are predicated ratings;$q_1 \ldots q_n$ are actual ratings and n is amount of ratings (Salam & Safir, 2016).

## 1.10 DISSERTATION OUTLINE

The rest of chapters are structured as follows:

Chapter 2 provides the literature review on the use of various algorithms in areas of recommendation systems. This chapter also deals with various studies on the comparison and assessment of recommender systems.

Chapter 3 draws up the methods used in this research by examining current methodologies in the field of information technology and discusses in depth the design research methodology used for this research.

Chapter 4 implements the method proposed, and analyses the results arising from the research

Chapter 5 presents the conclusion and future works.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 INTRODUCTION

Recommender systems are categorized under the class of personalized information filtering technologies that are targeted to help decision making in large information set (Lu, Wu, Mao, Wang, & Zhang, 2015). Recommendation systems are a new tool for assisting people in exploring information using the internet based on their preferences and help to identify products that users probably prefer to use (Rana & Jain, 2012). Therefore, different researchers have shown that the recommendation systems can assist users to find the items in which they are interested by helping users in their decision-making process. They recommend which items to buy, which movies to watch, which books to read, which music to listen to, which restaurants to choose, and so on (Porcel, Morales-del-Castillo, Cobo, Ruíz, & Herrera-Viedma, 2010). Recommender systems assist users while they are trying to obtain relevant information by evaluating and filtering a huge amount of information available on the internet (Porcel, Morales-del-Castillo, Cobo, Ruíz, & Herrera-Viedma, 2010).

Perugin, Gonçalves, & Fox (2004) conducted a research on recommender systems and proposed a system that assists customers to select a subset of items among a universal set of items based on their choice of preferences. A connection-oriented perspective was taken towards recommender systems research. The recommendation was taken as it connects people either directly because of user modeling or indirectly through relationships implicit in the data. Recommender systems are therefore characterized by how they model users to bring people together either explicitly or implicitly. However, the methodology used in this study is not clear about systematizing the process of the user's structure and developing systems. A location-based recommendation system using the Bayesian users' preference model in mobile devices was proposed by Park, Hong, & Cho (2007). It was mentioned that personalized services and user interface are necessary because of the small size of the mobile devices screen and inadequate resources on the proposed personalized recommendation system. The user's choice was modeled by Bayesian Networks. The parameter was learned from the database while an expert built the structure. The proposed system was displayed on the mini-map by collecting context information, location, weather, and time and user requests from mobile devices.

## 2.2 TECHNIQUES OF RECOMMENDER SYSTEM

There are many different techniques used in the recommendation system field. These techniques are classified into the following categories: collaborative Filtering (CF), Content-Based Filtering (CBF), Hybrid Filtering (HF), Knowledge-Based Filtering (KBF) and Demographic Filtering (DF) (Zuva, Ojo, Ngwira, & Zuva, 2012). Collaborative filtering and content-based Filtering techniques are the most commonly used techniques amongst others. The techniques used in recommender systems are shown in the figure below.

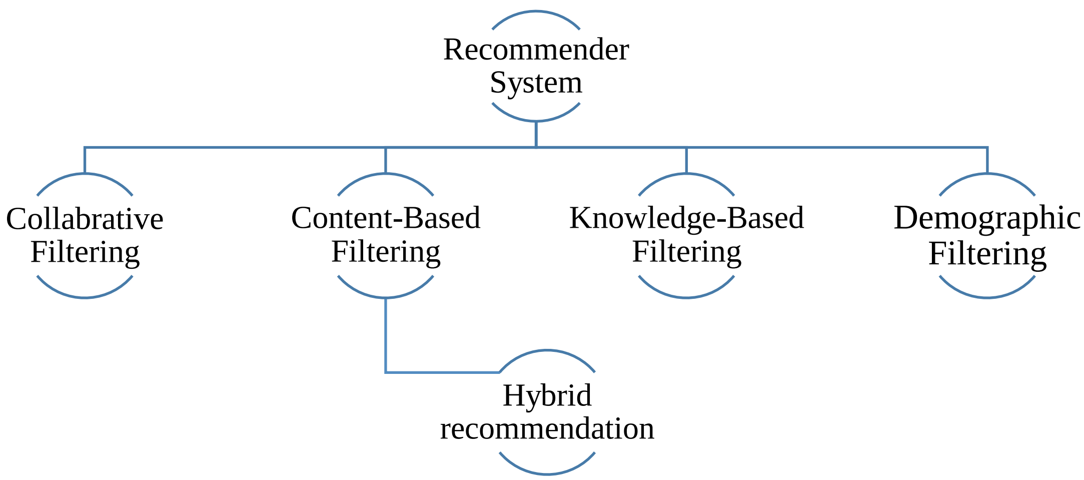


**Figure 2. 1** Techniques of recommender system (Zuva, Ojo, Ngwira, & Zuva, 2012)

### 2.2.1 COLLABORATIVE FILTERING

Collaborative filtering is the process of filtering or evaluating items using other peoples' opinions or choices (Schafer, Frankowski, Herlocker, & Sen, 2007). A collaborative filtering technique recommends items like books, CDs, and movies based on similar interest of users (Ekstrand, Riedl, & Konstan, 2011). Ungar and Foster (1998) used clustering methods for collaborative filtering. In their study, authors grouped people who have purchased similar items into clusters and it allowed for a more accurate recommendation for a new purchase. Grouping people into clusters with similar movies and grouping movies into clusters that are supposed to be liked by the same people can lead to predictions that have better results. Finding optimal clusters is not easy, as the groups of movies should be used to determine the people groups and vice versa. They presented a formal statistical model of collaborative filtering and they compared different algorithms including variations of K-means clustering and Gibbs sampling.

Collaborative filtering techniques provided good accuracy, correlation and SVD based methods, for instance they provided good accuracy except they were computationally expensive and could be used only in static off-line settings (George & Merugu, 2005). A novel collaborative filtering approach based on a proposed weighted co-clustering algorithm (Banerjee, Dhillon, Ghosh, Merugu, & Modha, 2004) was considered in the proposed scalable collaborative filtering framework based on co-clustering. This involved the simultaneous clustering of users and items. Incremental and parallel versions of the co-clustering algorithm were designed and used to build an efficient real-time collaborative filtering framework. The empirical evaluation demonstrated that their approach provided accuracy comparable to that of the correlation and matrix factorization-based approaches at a much lower computational cost. The approach used was compared to the correlation and matrix factorization-based approaches and provided the accuracy of a much lower computational course.

Ekstrand, Riedl, & Konstan (2011) wrote a book on collaborative filtering recommender systems. As recommender systems are very critical for the commerce eco-system, they introduced a powerful method for users to be able to filter relevant information among large information and product spaces. Design and evaluation of recommenders have to be done to address problems such as our information needs and item domains that are represented as a unique problem. Many options for choice of algorithms are available for system designers based on this analysis.

O'Mahony, & Smyth (2007) conducted a research on the development of a course recommender system for university college on-line enrolment application. A collaborative filtering style that suggests elective modules based on the past choices of like-minded students was used. Their study outlined the factors that influence student choices and designed solutions to address some of the key considerations that were identified. A simple content-based recommender for recommending similar modules based on similar keywords was also used. Recall and coverage were used to evaluate the system.

Chen & Chen (2007) developed personalized recommendatory system architecture to enable personalized services and management in a campus digital library. Using clustering algorithm and association rules, a two-phase data mining process was designed and recommendations were made. The process considered both the relationships in the users' cluster and the associations among the information accessed. The users' needs were closely met through recommendations given by this proposed system except that the system was not proactive.

Liao, Hsu, Cheng, & Chen (2010) made a research on a library recommender system based on a personal ontology model and collaborative filtering technique for English collections. In this study, a collaborative filtering method with personal ontology was adopted by using keywords of items in the library's collections to evaluate the preference of each user that minimizes data scarcity, improves accuracy, and solves the cold start of new coming items caused by the collaborative filtering method. This system has been operated at the National Chung Hsing University. This personal ontology approach lacked personal interest since keywords only are not enough to know somebody's choices.

Ze & Dengwen (2016) proposed an optimization collaborative filtering recommendation algorithm based on consistency of ratings. They designed a new similarity measure method called score based using a collaborative filtering technic. Users' common rating data was selected to calculate similarity and the nearest neighbors of users were brought and predicted item ratings were used to recommend and minimize the value of similarity calculation and actual value deviation. The prediction accuracy was increased with the improved algorithm proposed to improve the quality of the recommendation. Recommendations would be closer to users' expectation if the content-based algorithm was combined with existing collaborative algorithm.

Zheng, Ma, & Lyu (2009) proposed a web service recommender system that is a collaborative filtering-based web service recommender system. A web service recommender system was introduced to address recommender system critical problems. A user-contribution mechanism for web service QoS information collection and an effective and novel hybrid collaborative filtering algorithm were used for the web service recommender system included for web service QoS value prediction. Java language was used to implement the web service recommender system and it was deployed in the real-world environment. Twenty one thousand one hundred and ninety seven public web services were collected from the internet and a large-scale real-world experiment was conducted to study the prediction performance. Among 150 service users in different countries that have publicly available web services, 1.5 million test results were collected all over the world. The system achieved better prediction accuracy than others based on the comprehensive experimental analysis.

In their lecture book called 'Link prediction approach to collaborative filtering', Huang, Li, & Chen (2005) stated that collaborative filtering explores the correlations within user-item interactions to imply user preferences. A graph-based algorithm was designed to overcome the data sparsity problem that affects the quality of recommendations. The proposed graph-based algorithms introduced transitive user-item interactions and make use of link prediction approaches. These approaches gave a better performance than standard collaborative filtering algorithm experiments.

To address the problem of data sparsity and scalability, Kumar & Fan (2015) proposed a hybrid user-item based collaborative filtering algorithm and tried to achieve more personalized recommendations for users. They used a combination of case-based reasoning and average filling to solve the challenge of the sparsity of the dataset. A self-organizing map was optimized with the genetic algorithm that performs user clustering in large datasets to minimize the scope for item-based collaborative filtering. Encouraging results were seen after the proposed method was evaluated and compared with the traditional item-based collaborative filtering algorithm.

Herlocker, Konstan, Terveen, & Riedl (2004), studied on collaborative filtering recommender systems evaluation. The key decisions in evaluating collaborative filtering recommender systems were reviewed and made with regard to the user tasks, the type of analysis and datasets being used, the way the prediction quality was measured, prediction quality, prediction attributes other than quality, and user-based evaluation of the whole system. It presented empirical results compared to the evaluation used by previous researchers. All tested metrics fall into three equivalence classes from different accuracy matrices analysis on one content domain. Metrics that were uncorrelated were from different equivalency classes and metrics within each equivalency class that was highly correlated.

The two types of collaborative techniques are user-based and item-based collaborative techniques. They are studied in the next sections.

## 2.2.2 USER-BASED NEAREST NEIGHBOR ALGORITHM

The idea behind user-based collaborative filtering is that people with similar behavior have similar tastes. Users that are similar to a user are identified and the desired rating is estimated which is going to be a weighted average of the ratings of these similar users. User-based recommendation recommends new information to a user by considering other similar users' interests. User-based recommendations work by assuming if two users have similar preferences, then the two users will probably like to access the same information (Zeng, Li, Liu, Wen, & Hirokawa, 2016).

The algorithm can be divided into three processes. They computed users' similarity that is called user similarity; computing most-similar users as a group that is called User Neighborhood; recommend using the recommendation engine based on User Neighborhood and the user similarity. An improved user-based movie recommendation algorithm used the user-based recommendation algorithm, which rewrote user similarity by merging users' ages and genders into users' preference values on items. The (Root Mean Square Error) is used to evaluate and state the proposed algorithm is more accurate than the original one.

Lumauag, Sison, & Medina (2019) proposed an enhanced recommendation algorithm based on modified user-based collaborative filtering. Modified user-based collaborative filtering was used to solve the problem and improve the recommendation quality. Using the user-item rating matrix, the rating data of the active user and the rating data of the other users are compared. Similarity measures are used to calculate user similarity that is used to compute the prediction score. The predicted item to be recommended to the active user is determined by the prediction score based on the rating preference of other users. After using movie lens dataset and evaluating its accuracy and performance using root mean square error, precision, and recall, the enhanced algorithm was compared to the traditional algorithm and it outperformed and yielded an improved accuracy of recommendation.

The most common challenges faced when using a user-based nearest algorithm are the negative correlations that are in the Pearson coefficient that are not valued in the aspect of increasing the correlating prediction accuracy (Mohd, Hameed, al jadaan, & Sirandas, 2012 ). The equation to calculate is given in the equation (2.1) below.

$$userssim(x, y) = \frac{\sum_{i \in l_{xy}} (R_{xi} - \bar{R}_x)(R_{yi} - \bar{R}_y)}{\sqrt{\sum_{i \in l_{xy}} (R_{xi} - \bar{R}_x)^2} \sqrt{\sum_{i \in l_{xy}} (R_{yi} - \bar{R}_y)^2}} \quad (2.1)$$

where usersim(xy) represents the similarity between the users x and y which signifies the item set that was ranked by the user x and y simultaneously. $R_{xi}$ and $R_{yi}$ are the item score i rated by the user x and y and in that order $\bar{R}_X$ and $\bar{R}_Y$ represent the average scores of users X and Y respectively (Hu, Sun, & Liu, 2014).

The final phase is when $N_y$ denotes the target user y neighbor set (Hu, Sun, & Liu, 2014). Predicting y ratings for item j is shown in the equation 2.2 below.

$$pred(y, j) = \frac{\bar{A}_Y + \sum (R_{U_n, j} - \bar{R}_{U_n}) * usersim(y, u_n)}{\sum_{U_n \in N_y} sim(y, u_n)} \quad (2.2)$$

where $\bar{A}_Y$ signifies the average score for the user y for the rated items, $R_{un,j}$ is the score of item j rated by the neighbor Un $\bar{R}_{U_n}$ indicates the average score of the neighbor $u_n$ for the rated items. Sim $(y,u_n)$ denotes the similarity between y and the neighbor $u_n$ (Hu, Sun, & Liu, 2014).

### 2.2.3 ITEM-BASED NEAREST NEIGHBOR ALGORITHM

The study conducted by Wei, Ye, Zhang, & Huang (2012) proposed the item-based collaborative filtering recommendation algorithm combining item category with interestingness measure. To overcome the limitations of data scarcity and inaccurate similarity in personalized recommendation systems, a new collaborative filtering algorithm through use of items categories similarity and interestingness measure was proposed. In this algorithm, first the item category similarity matrix was constructed by calculating the item-item category distance, and then analyzing the correlation degree of different items by using the interestingness measure and, lastly, an improved collaborative filtering algorithm was conducted by combining the information of items categories with item-item interestingness and utilizing the improved conditional probability method as the standard item-item similarity measure. Experimental results showed this algorithm could effectively solve the dataset sparsity problem and achieve better prediction accuracy compared to other well-performing collaborative filtering algorithms.

Linden, Smith, & York (2003) proposed Amazon.com recommendations: item-to-item collaborative filtering. At Amazon.com, they used recommendation algorithms to personalize the online store for each customer. The store radically changed based on customer interests, showing programing titles to a software engineer and baby toys to a new mother. They compared three common approaches with their algorithm to solve the recommendation problem called traditional collaborative filtering, cluster models and search-based methods, which they called item-to-item collaborative filtering. Unlike traditional collaborative filtering, their algorithm's online computation scaled independently of the number of customers and the number of items in the product catalog. Their algorithm produced recommendations in real-time, scaled to massive datasets, and generated high-quality recommendations.

Sarwar, Karypis, Konstan, & Riedl (2001) studied item-based collaborative filtering recommendation algorithms. Important research questions in overcoming two fundamental challenges for collaborative filtering recommender systems were listed. The first challenge was to improve the scalability of collaborative filtering algorithms. These algorithms were able to search tens of thousands of potential neighbors in real-time, but modern systems demanded to search tens of millions of potential neighbors. It was mentioned that existing algorithms have performance problems with individual users for whom the site has large amounts of information. For instance, if a site was using browsing patterns as indications of content preference, it would have thousands of data points for its most frequent visitors. These "long user rows" slowed down the number of neighbors that could be searched per second, and further reduced scalability. The second challenge was to improve the quality of the recommendations for the users. It was deemed that users needed recommendations they can trust to help them find items they will like. Users would "vote with their feet" by refusing to use recommender systems that were not consistently accurate for them. In some ways these two challenges were in conflict, since the less time an algorithm spends searching for neighbors, the more scalable it would be, and the worse was its quality. For this reason, it became important to treat the two challenges simultaneously so that the solutions discovered were both useful and practical. In their suggestions, these issues of recommender systems were addressed by applying different approaches - item-based algorithms.

The bottleneck in conventional collaborative filtering algorithms was the search for neighbors among a large user population of potential neighbors. Item-based algorithms avoided this bottleneck by exploring the relationships between items first, rather than the relationships between users. Recommendations for users were computed by finding items that were similar to other items the user has liked. Because the relationships between items were relatively static, item-based algorithms could have been able to provide the same quality as the user-based algorithms with less online computation.

### 2.2.4 CONTENT-BASED FILTERING

Melville, Mooney, & Nagarajan (2002) proposed a content-based book recommending system by using learning for text categorization approach. The ability to recommend books and other information sources to the users based on their general interests rather than specific inquiries is an important service of digital libraries. It was stated that content-based recommenders were able to recommend effectively unrated items and provide efficient recommendations to users with unique, individual tastes. The authors used 10-fold cross, validation, classification accuracy, recall, precision, rating of top 3, rating of top 10, rank correlation to evaluate the proposed methodology (Mooney & Roy, 2000). Even though they solved the problem of a collaborative approach that fails to recommend unrated and unpopular items, the proposed system has a limitation of finding more information from other users' ratings and comments about the content of the book. Hence, the content-based approach does not consider items most rated by other users.

Kompan & Bieliková (2010) proposed a content-based news recommendation in the research project called SME FIIT. In their study, the authors presented a new approach for fast content-based news recommendations based on cosine-similarity search and effective representation of the news where the proposed method computed the list of similar articles for every article in the dataset. The evaluation was done based on the similarity computation of over 10 000 articles from the Slovak news portal per week. Therefore, they were able to estimate the similarity in 2-3 seconds. The pre-processing time was approximately 20 seconds while, the computation process of the whole dataset took approximately 22 seconds.

Deldjoo et al. (2016) designed a content-based video recommendation system using stylistic visual features. This study was primarily investigating the use of automatically extracted visual features of videos in the context of recommender systems that brought about some novel contributions in the domain of video recommendations. The content-based recommender system was proposed to encompass a new technique that automatically analyzes video content to extract a set of representative stylistic features (lighting, color, and motion) grounded on existing approaches of applied media theory. The evaluation of the proposed recommendations used performance evaluation metrics such as precision and recall. Thereafter, the recommendation was compared with existing content-based recommender systems that exploited explicit features such as movie genre. It showed that their technique leads to better recommendations and results. It would give a better result if they included audio features as well. A similar study was done by Oord, Sander, & Benjamin (2013) where authors proposed deep content-based music recommender system. The authors preferred to use a latent factor model for the recommendation, and predict the latent factors from music audio when they could not be obtained from usage data. The study compared a traditional approach using a bag-of-words representation of the audio signals with deep convolutional neural and evaluated the predictions quantitatively on the Million Song Dataset. The results showed that using predicted latent factors produces sensible recommendations, although there was a large semantic gap between the characteristics of a song that affect user preference and the corresponding audio signal. It is also clear that recent advances in deep learning translate very well to the music recommendation setting, with deep convolutional neural networks significantly outperforming the traditional approach. They tried to avoid the cold start problem from new and unpopular music.

Achakulvisut, Acuna, Ruangrong, & Kording (2016) researched a content-based recommendation system for scientific publications. They develop an algorithm and an accompanying Python library that implemented a recommendation system based on the content of articles. Design principles were to adapt to new content, provide near-real-time suggestions. They tested the library on 15K posters from the Society of Neuroscience Conference 2015. Human curated topics were used to cross-validate parameters in the algorithm and produce a similarity metric that maximally correlates with human judgments. They showed that their algorithm significantly outperformed suggestions based on keywords. The work presented promised to make the exploration of scholarly material faster and more accurate.

van Meteren & van Someren (2000) used content-based filtering for the recommendation. The recommender system PRES was described. It uses content-based filtering techniques to suggest small articles about home improvements. A domain such as this implicated that the user model has to be very dynamic and learned from positive feedback only. The relevance feedback method seemed to be a good candidate for learning such a user model, as it was both efficient and dynamic. The test results seemed to indicate that, on average, slightly more than one out of two of the suggestions that PRES made was relevant. The results were negatively influenced by the fact that the same concept could usually be described with several terms and many terms have more than one meaning. This made the user profile less accurate, especially because the documents in the collection were relatively short and normally only a few documents about the same topic are selected by the user. Better results could have been obtained by improving the vector space model. The effectiveness of PRES could be improved if content-based and collaborative filtering were combined. The menu structure also had a great influence on the effectiveness of PRES. Because it is not useful to recommend items that already appeared in the current menu, it became more difficult to make recommendations if the user had selected a menu that already contained most of the relevant items.

Manikrao & Prabhakar (2005) proposed a dynamic system that uses selection of web services with the recommendation system approaches. The realization of the semantic web was underway with the development of an arena of services providing similar properties, capabilities, interfaces, and effects. To pick one of such similar services that match the user's requirements was a difficult task and necessitated the use of an intelligent decision-making framework. Their methodology addressed precisely the above-mentioned component. They presented the design of a dynamic web-service selection framework that made use of a semantic matcher to support matching and composition of software services. The framework also used a recommendation system that helped a user to select the best service that matched their requirements. This recommendation system resulted in the evolution of the framework to adapt to the user's requirements.

Cao & Li (2007) proposed an intelligent fuzzy-based recommendation system for consumer electronic products. They mentioned that developing an intelligent recommendation system is a good way to overcome the problem of overloaded product information provided by enterprises. In his study, based on the consumer's current needs obtained from the system-user interactions, they proposed a fuzzy-based system for consumer electronics to retrieve optimal products. Experimental results showed that the system was feasible and effective.

Zhang & Koren (2007) proposed efficient Bayesian hierarchical user modeling for the recommendation system. A content-based personalized recommendation system learned user-specific profiles from user feedback so that it can deliver information tailored to each user's interest. A system serving millions of users could learn a better user profile for a new user, or a user with little feedback, by borrowing information from other users with a Bayesian hierarchical model. Learning the model parameters to optimize the joint data likelihood from millions of users was very computationally expensive. The commonly used EM algorithm converged very slowly due to the sparseness of the data in the IR applications. This study proposed a new fast learning technique to learn a large number of individual user profiles. The efficacy and efficiency of the proposed algorithm were justified by theory and were demonstrated on actual user data from Netflix and Movie-Lens.

Aygün & Yıldız (2016) proposed the development of a content-based book recommendation system using a genetic algorithm. The trend of making a presentation of the right content to the right user would be inevitable. For this purpose, recommendation systems were widely used for the areas of music, books, movies, tourist travel planning, e-commerce, education, and many more. The approach of recommender systems was based on the user's history of choices, likings, and reviews, each of which is interpreted to project the future choices of the user. In this study, a novel and strong recommender system for books was proposed. A content-based book recommendation application was developed which made recommendations according to the user's taste and choices.

### 2.2.5 HYBRID RECOMMENDATION

Huang, Chung, Ong, & Chen (2002) proposed a graph-based recommender system for digital libraries. In this study, authors developed a graph-based model that combines content-based and collaborative recommendation approaches and implemented the system in the context of an online Chinese bookstore. A Hopfield net algorithm was used to exploit high-degree book-book, user-user, and book-user associations.

Sample holdout testing and preliminary subject testing were conducted to evaluate the system. It was found that the system gained improvement with respect to both precision and recall metrics by combining content-based and collaborative approaches. Despite the use of both collaborative and content-based methods, no significant improvement was observed by exploiting high-degree associations.

Walter, Battiston, & Schweitzer (2008) presented a model of a trust-based recommendation system on a social network. The idea of the model was that agents use their social network to reach information and their trust relationships to filter it. They investigated how the dynamics of trust among agents affect the performance of the system by comparing it to a frequency-based recommendation system. Furthermore, they identified the impact of network density, preference heterogeneity among agents and knowledge sparseness to be crucial factors for the performance of the system. The system was self-organized in a state with performance near to the optimum; the performance on the global level as an emergent property of the system was achieved without explicit coordination from the local interactions of agents.

Debnath, Ganguly, & Mitra (2008) proposed a hybridization of collaborative filtering and content-based recommendation systems. Attributes used for content-based recommendations were assigned weights depending on their importance to users. The weight values were estimated from a set of linear regression equations obtained from a social network graph that captured human judgment about the similarity of items. In this study, content-based recommendation systems were studied. This definition refers to systems used in the web in order to recommend an item to a user based upon a description of the item and a profile of the user's interests. To start with, a definition of a recommendation system was generally given. Then, it was discussed why recommendation systems are necessary for web users and pinpointed the problem that was to be solved. Furthermore, there was a focus on techniques used in content-based recommendation systems in order to create a model of the user's interests and analyze an item collection, using the representation of the items. Additionally, the advantages and drawbacks of recommendation systems were emphasized both in the context of making recommendations and in contrast with other types of recommendation systems. Finally, there was a discussion about the LIBRA content-based recommendation system and emphasis on the CBMRS, PRES, and COBRA systems, which were implemented using the Java Platform.

Wu, Zhang, & Lu (2013) proposed a fuzzy tree similarity-based recommendation approach for telecom products. They justified the huge product assortments and difficult descriptions of telecom products and complex customers to select appropriate products. A fuzzy tree similarity-based hybrid recommendation approach was proposed to solve customers' challenges in selecting appropriate products. Fuzzy techniques were used to deal with the various uncertainties existing within the product and customer data. A fuzzy tree similarity measure was developed to evaluate the semantic similarity between tree-structured products or user profiles. The similarity measures for items and users both integrated the collaborative filtering (CF) and semantic similarities. The final recommendation hybridized item-based and user-based CF recommendation techniques. This technique was used to recommend products from telecommunication companies and the results reveal that the proposed approach was effective.

Li & Kim (2003) proposed an approach for combining content-based and collaborative filters. In their work, they applied a clustering technique to integrate the contents of items into the item-based collaborative filtering framework. The group rating information that was obtained from the clustering result provided a way to introduce content information into collaborative recommendations and solved the cold start problem. Extensive experiments have been conducted on Movie-Lens data to analyze the characteristics of their technique. The results showed that their approach contributes to the improvement of the prediction quality of the item-based collaborative filtering, especially for the cold start problem.

Kumar & Fan (2015) proposed a hybrid user-item based collaborative filtering algorithm. They tried to overcome challenges of data sparsity and scalability by proposing a hybrid method based on item-based CF trying to achieve a more personalized product recommendation for a user while addressing some of these challenges. Case-based reasoning (CBR) was combined with average filling used to handle the sparsity of the dataset, while self-organizing map (SOM) optimized with genetic algorithm (GA) performs user clustering in large datasets to reduce the scope for item-based CF. The proposed method showed encouraging results when evaluated and compared with the traditional item-based CF algorithm.

Choi, Yoo, Kim, & Suh (2012) proposed a hybrid online-product recommendation system that combines implicit rating-based collaborative filtering and sequential pattern analysis. The authors mentioned that any online shopping malls in which explicit rating information was not available still had difficulty in providing recommendation services using collaborative filtering techniques for their users. The Hopfield net spreading activation algorithm was used for the high-degree association graph search. However, applying temporal purchase patterns derived from sequential pattern analysis (SPA) for recommendation services often made users unhappy with the inaccurate and biased results obtained by not counting on individual preferences. The study derived implicit ratings so that CF can be applied to online transaction data even when no explicit rating information was available, and integrated CF and SPA for improving recommendation quality based on the results of several experiments that they conducted to compare the performance between them and others. Finally, the proposed model was compared with existing search algorithms and contended that implicit rating can successfully replace explicit rating in CF and that the hybrid approach of CF and SPA is better than the individual ones.

Milicevic, Vesin, Ivanovic, & Budimac (2011) proposed E-Learning personalization based on a hybrid recommendation strategy and learning style identification, goals, talents, and interests of their learners. In this research, they described a recommendation module of a programming tutoring system, which could automatically adapt to the interests and knowledge levels of learners. This system recognized different patterns of learning style and learners' habits through testing the learning styles of learners and mining their server logs.

The experiments were carried out on two real groups of learners, an experimental group and a control group. Learners from the control group learned in a normal way and did not receive any guidance through the course, while the students from the experimental group were required to use the system. The results showed the suitability of using this recommendation model in order to recommend online learning activities to learners based on their learning style, knowledge and preferences of learners.

Martinez, Arias, Vilas, Duque, & Nores (2009) conducted a study on efficient and effective personalized recommender system of TV programs called quevio.tv that intends to recommend what program on TV you can watch tonight. Personalization was achieved with the employment of algorithms and data collection schemes that could predict and recommend to television viewers content that matches their interests and/or needs. Their study introduced queveo.tv: a personalized TV program recommendation system. The proposed hybrid approach combined content-filtering techniques and collaborative filtering. Experimental results reveal that the proposed system can process some tasks such as downloading, storing and retrieving big files in databases offline.

Shinde & Kulkarni (2012) conducted a research study on a hybrid personalized recommender system using a centering-bunching based clustering algorithm. This research work proposed a novel centering-bunching based clustering algorithm that was used for hybrid personalized recommender systems. The proposed system worked in two steps; the first step was considering opinions from the users, which were collected in the form of a user-item rating matrix. In the second step the recommendations were generated online for the active user using similarity measures by choosing the clusters that had the higher rating. This helped to get further effectiveness and quality of recommendations for the active users. The experimental results using the Iris dataset showed that the proposed CBBC performed better than K-means and new K-modification methods algorithms.

Torres, McNee, Abel, Konstan, & Riedl (2004) undertook a research based on enhancing digital libraries using Tech-Lens technology. The hybrid algorithms were used to combine the strengths of collaborative filtering and content-based approaches to benefit from their potentials. Their results showed that the proposed method was efficient to be applied to develop recommender systems for other types of digital libraries.

Gao, Xing, Du, & Wang (2007) provided a personalized service system based on hybrid filtering for digital libraries. The authors suggested a new methodology that unified partition-based collaborative filtering and meta-information filtering where the user-item rating matrix can be partitioned into low-dimensional dense matrices using a matrix-clustering algorithm. Recommendations are generated based on these low-dimensional matrices. Additionally, the very low rating problem was solved using meta-information filtering while the unified method was applied to a digital resource management system. The experimental results indicate the high efficiency and good performance of the new method.

Ghazanfar & Prugel-Bennett (2010) proposed a scalable, accurate hybrid recommender system. In this research, authors proposed a unique cascading hybrid recommendation approach by combining different features such as rating and demographic information about items. The recommended method outperforms the standards of recommender system algorithms, and eliminates recorded problems from other recommender system approaches.

Gunawardana & Meek (2009) proposed a unified approach to building hybrid recommender systems. The study describes unified Boltzmann machines, which are probabilistic models that combine collaborative and content information in a coherent manner. They encode collaborative and content information as features, and then learn weights that reflect how well each feature predicts user actions. In doing so, information of different types is automatically weighted, without the need for careful engineering of features or for post-hoc hybridization of distinct recommender systems. The performed empirical results on movie and shopping domains shows that unified Boltzmann machines can be used to combine content and collaborative information to yield results that are competitive with collaborative techniques in recommending items that have been seen before, and also effective at recommending cold-start items.

Bostandjiev, O'Donovan, & Höllerer (2012) conducted a visual interactive hybrid recommender system. The proposed system generates item predictions from multiple social media platforms and semantic web resources such as Wikipedia and Facebook. The authors preferred hybrid techniques from the traditional recommender system. The evaluation from the comparison of different interactive and non-interactive hybrid strategies for computing recommendations across diverse social and semantic web indicate that explanation and interaction with a visual representation of the hybrid system increase user satisfaction and relevance of predicted content.

### 2.2.6 DEMOGRAPHIC-BASED FILTERING

Many modern recommender systems use hybridization, which combines two or more recommender techniques to gain better performance than when the systems are implemented individually. Demographic filtering is mostly employed in hybrid systems together with other types of recommender techniques in order to enhance prediction accuracy (Burke, 2002).

Demographic recommender systems categorize users or items based on their personal attributes and make recommendations based on demographic categorizations. Demographic filtering technique uses the demographic data of a user to determine which items may be appropriate for the recommendation. Demographic recommender systems utilize user attributes, classified as demographic data, in order to produce their recommendations, sometimes with the help of pre-generated demographic clusters.

The strength of the demographic filtering technique is that the new user problem does not apply to this type of recommender system since they do not need a list of ratings from a new user to make recommendations. However, according to earlier research, the major problem with demographic systems is that demographic data in combination with item ratings are difficult to acquire (Burke, 2002).

Park, Hong, & Cho (2007) proposed a location-based recommendation system using the Bayesian user's preference model in mobile devices. They proposed a map-based personalized recommendation system that reflected the user's preference modeled by Bayesian Networks. The structure of Bayesian Networks was built by an expert while the parameter was learned from the dataset. The proposed system collected context information, location, time, weather, and user request from the mobile device and inferred the most preferred item to provide an appropriate service by displaying onto the mini-map.

### 2.2.7 KNOWLEDGE-BASED

A knowledge- based recommender system is one that uses knowledge about users and products to pursue a knowledge-based approach to generate a recommendation and reason about what products meet the user's requirements (Bhargava, Sridhar, & Herrick, 1999).

Kuroiwa & Bhalla (2007) proposed a dynamic personalization for book recommendation system using web services and virtual library enhancements. They proposed to build a knowledge base by collecting book information using web services. They developed a book utilization system (BUS) that enabled users to edit additional book information by using a XML database. They used a book search using web services. They also created an infrastructure for sharing existing books among users by extracting featured keywords from KB for individuals' preference visualization. Their book search methodology made it possible to find and suggest the use of available books about "XML", when a user looked for books about "XQuery" by searching web resources using web services.

Chun & Hong (2001) proposed a framework that implements a knowledge-based recommender system for electronic commerce using the Java expert system library. This study dealt with the design and implementation of a product recommender system on an online shopping application. The system collected the users' information on a particular product by questioning the users and consulting its knowledge base to find the items that best meet the users' requirements.

Martin et al. (2012) suggested a recommender system that identifies a new set of media items responsive to an input set of media items and knowledge base metrics. Various metrics among media items were considered by analyzing how the media items are grouped to form the media sets in the knowledge base (Martin, Shur, & Torrens, 2012). Such association or "similarity" metrics are preferably stored in a matrix form that allows the system to identify a new set of media items that complements the input set of media items.

## 2.3 RELATED WORKS

Sase, Varun, Rathod, & Patil (2015) proposed a book recommendation engine that used data mining techniques to recommend books. The proposed recommender system would give its users the ability to view and search books as well as novels that would be used to draw out conclusions about the stream of a user and the genre of the books liked by that user. The system analyzed behavior by using the features of various recommendation techniques such as content-based, collaborative and demographic. Thus, in this study, a hybrid recommendation system was proposed which satisfied a user by providing the best and efficient book recommendations.

Tashkandi, Wiese, & Baum (2017) proposed an offline comparative evaluation of commonly used recommendation algorithms for book recommendation systems. The authors used collaborative technique to predict the books that the user will most probably like. The matrix of user-item ratings was used as an input. The output gave numerical rating prediction of the degree that the user likes or dislikes a specific book. Book Crossing data set was used and their study was implemented using three platforms called LensKit, Mahout, and MyMediaLite. The authors achieved 1.953229033 RMSE using Pearson similarity method and 1.953229033 RMSE as their final result.

Parvatikar & Joshi (2015) proposed an online book recommendation system use of collaborative filtering and association mining. This study solved the problem of data sparsity problem by combining the collaborative-based filtering and association rule mining to achieve better performance. The results obtained were demonstrated and the proposed recommendation algorithms performed better and solved the challenges such as data sparsity and scalability.

Jomsri (2014) proposed a book recommendation system for digital libraries based on user profiles by making use of association rule. The system used user profile loaning and applied association rule to create a model. Although the system was not proactive, the results showed that the new association rule algorithm was suitable to apply for recommender books in the library.

Bhure & Adhe (2015) proposed a system for book recommendation called the system using opinion mining technique. This system recommended books to the user by collecting and comments. For effective data analysis, opinion-mining techniques were deliberately used by this system. A collaborative method of commtrust and normalization algorithm was also used by the system in which the latter normalization contained the ranking of books, based on the weights assigned to them. The evaluation method used was a normalized opinion score (NS). NS = T/M where, T = Sum of total weights assigned to the book M = Sum of the maximum weights that can be assigned to each feature of the book. Despite the fact that the algorithm used for collaborative methods was a good one, the system still had a problem of cold start since it uses only a collaborative method.

Rana & Jain (2015) proposed a study on building a book recommender system using time-based content filtering. In this study, it was mentioned that recommendation systems are a new generation tool for helping the users in navigating information through the internet and retrieving information according to their preferences. A content-based approach with a new dimension called temporal dimension with the help of a counter each time the item is updated with the passage of time was used for their work. The users were then asked to rate the recommendations for recommendation evaluation. However, the system was limited to finding more information from other users' ratings and comments about the contents of the book; thus, the content-based approach does not consider most rated books by other users.

Ghadling, Belavad, Bhegade, Ghojage, & Supriya (2015) used only the surname of the main author proposed digital library with a hybrid book recommender engine that used collaborative, content-based, and proactive recommender system recommending techniques. A book recommender system for the digital library could be proposed by using a hybrid of those techniques since the hybrid book recommendation engine was very helpful to eliminate the weak side of each technique. In addition, the proposed system could have been used for college libraries, public libraries, and private online libraries. This technique was a very good approach for a book recommender system. Vaz, Martins de Matos, Martins, & Calado (2012) proposed improving a hybrid literary book recommendation system through author ranking. In their study, they presented a hybrid recommendation system to help readers decide which book to read next. They studied the book and author recommendations in a hybrid recommendation setting and tested their algorithm on the LitRec dataset. Their hybrid method combined two item-based collaborative filtering algorithms to predict books and authors that the user would like. Author predictions were expanded into a booklist that was subsequently aggregated with the former book predictions. Finally, the resulting booklist was used to give top similar books recommendations. By means of various experiments, they demonstrated that author recommendations could improve overall book recommendations.

Chen & Yang (2010) proposed an intelligent mobile location-aware book recommendation system that enhanced problem-based learning in libraries**.** It was mentioned that, by integrating the problem-based learning (PBL) model with book resources in libraries, one could identify the advantages of libraries in supporting e-learning, resulting in innovative and valuable research. Therefore, the study presented a novel intelligent mobile location-aware book recommendation system with map-based guidance to support cooperative PBL in a real-library environment. Using map navigation and book recommendation functionalities, learners could search for books associated with problem solving with increased ease and efficiency, thereby helping learners increase their PBL performance in a library environment. Experimental results revealed that learning performance during PBL supported by the proposed IMLBRS for book searches was superior to learning performance during PBL supported by the online public access catalog. Experimental results also showed that the proposed system facilitated better learning performance for learners with a field-dependent learning style than for learners with a field-independent learning style. Moreover, the proposed system facilitated learner contemplation, cooperative learning, and library user education as learners interacted with a real-library environment during cooperative PBL. Cui & Chen (2009) proposed an online book recommendation system based on web service. They mentioned that the book recommendation system has been developed rapidly due to the web technology and library modernization and existing recommendation systems could not supply enough information for readers to decide whether to recommend a book or not. In order to solve those problems, they designed a recommendation system for a novel book. Readers would be redirected to the recommendation pages when they could not find the required book through the library bibliographic retrieval system. The recommendation pages contained all the essential and expanding book information for readers to refer. Readers could recommend a book on these pages, and the recommendation data would be analyzed by the recommendation system to make a scientific purchasing decision. They proposed two formulas to compute the book value and the copy number respectively based on the recommendation data. The application of the recommendation system showed that both the recommended book utilization and readers' satisfaction were greatly increased.

Hahn (2011) proposed location-based recommendation services in library book stacks. This study suggested a model for location-based recommendation services that enable greater access to print and electronic resources. The design, methodology and approach used took the form of a synthesis of previous work in basic and applied collections-based way finding incorporating a library and information science literature on user context and system recommendations. The study identified problems that needed to be solved before the implementation of the production-level recommendation service and suggested possible implications the system may have on reference and instruction services. The study provided computing workflows necessary to implement a library recommendation service based on user location. IPhone software developer kit templates were leveraged for modeling data and interface prototypes. Use cases and user models were developed.

Mathew, Kuriakose, & Hegde (2016) proposed a book recommendation system through a content-based and collaborative filtering method. It was mentioned that the recommendation system is broadly used to recommend products to the end-users that were most appropriate. Online bookselling web sites were competing with each other by considering many attributes. Moreover, the previous systems lead to the extraction of irrelevant information and resulted in a lack of user satisfaction. This study presented the book recommendation system (BRS) based on combined features of content-based filtering, collaborative filtering and association rule mining to produce efficient and effective recommendations. For this, a hybrid algorithm was proposed which combined two or more algorithms which helped the recommendation system to recommend the book based on the buyer's interest. Tewari and Priyanka (2014) proposed a book recommendation system based on collaborative filtering and association rule mining for college students. This study presented an online book recommendation system for students reading textbooks. The main motive of this study was to develop the technique which recommended the most suitable books to the students according to their price range and publisher's name. This was based on the combined features of classification, user-based collaborative filtering and association rule mining.

Kurmashov, Latuta, & Nussipbekov (2015) proposed an online book recommendation system. This study proposed a quick and intuitive book recommendation system that helped readers to find an appropriate book to read next. The overall architecture was presented with its detailed description. A collaborative filtering method based on the Pearson correlation coefficient was used. Finally, the experimental results based on the online survey were provided with some discussions.

Rajpurkar & Bhatt (2015) proposed a book recommendation system. A book recommendation system based on combined features of content filtering, collaborative filtering, and association rule mining was presented. It mentioned that the recommendation system was one of the stronger tools to increase profit and retain buyers. The book recommendation system recommended books that were to a buyer's interest.

Parvatikar & Joshi (2015) proposed an online book recommendation system by using collaborative filtering and association mining. It was mentioned that various techniques have been introduced for recommending items, i.e. content, collaborative and association-mining techniques were used. This solved the problem of the data sparsity problem by combining the collaborative-based filtering and association rule mining to achieve better performance. The results obtained were demonstrated and the proposed recommendation algorithms performed better and solved the challenges such as data sparsity and scalability. If the proposed system was proactive, it could perform better than the obtained results.

Liao, Hsu, Cheng, & Chen (2010) proposed a library recommender system based on a personal ontology model and collaborative filtering technique for English collections. A collaborative filtering method with personal ontology was adopted by using keywords of items in the library's collections to evaluate the preference of each user that minimizes data scarcity, improves accuracy, and solves the cold start of new coming items caused by the collaborative filtering method. This system has been operated in the National Chung HOUSING University. However, this personal ontology approach lacked personal interest since keywords only are not enough to know one's choices.

Geyer-Schulz, Hahsler, Neumann, & Thede (2003) proposed behavior-based recommender systems as value-added services for scientific libraries. They developed a recommender system based on a stochastic purchase incidence model, presented the underlying stochastic model from repeat-buying theory, and analyzed whether the underlying assumptions on consumer behavior held for users of scientific libraries. A test prototype was already operational and evaluated. By June 2002, the recommender service was operational within the library system of the University at Karlsruhe. Heylighen & Bollen (2002) proposed Hebbian algorithms for a digital library recommendation system. They proposed a set of algorithms to extract metadata about the documents in a digital library from the way documents were used. Co-activation values were added, producing a matrix of associations. This matrix could be used to recommend the documents that are most strongly related to a given document, most relevant to the user's implicit interest profile, or most interesting to users overall. They calculated document similarity values, which in turn can be used to cluster similar documents. The data needed to feed such a recommendation system was readily extracted from the usage logs of document servers. Tejeda-Lorente, Porcel, Peis, Sanz, & Herrera-Viedma (2014) proposed a quality-based recommender system to disseminate information in a university digital library. To overcome the problem of the absence of taking the quality of items into account, they proposed a new recommender system based on quality. This system used the quality of the items to estimate their relevance. The system measured the item quality and took into account this measure as a new factor to be considered in the recommendation process. In such a way, they presented a recommender system based on items' quality to help users to access relevant research resources. They developed the recommender system using a fuzzy linguistic approach. It was tested in a university digital library and the result was satisfying.

Chen, Kuo, & Liao (2014) proposed an ontology-based library recommender system using map reduce recommender systems that have proven useful in numerous contemporary applications, helping users effectively to identify items of interest within massive and potentially overwhelming collections. They stated that the major weakness of the collaborative filtering mechanism was its performance in computing the pairwise similarity of users. Thus, the map reduce framework was examined as a potential means to address this performance problem. Their research study detailed the development and employment of the map reduce framework, examining whether it improves the performance of a personal ontology-based recommender system in a digital library. The results of this extensive performance study showed that the proposed algorithm could scale recommender systems for all-pairs similarity searching.

Geyer-Schulz, Neumann, & Thede (2013) proposed architecture for behavior-based library recommender systems. In this article, architecture for distributed recommender services based on a stochastic purchase incidence model is presented. Experiences with a recommender service that has been operational within the scientific library system of the University Karlsruhe since June 2002 were described. They mentioned that by utilizing lending and searching log files from online public access catalogs through data mining, customer-oriented service portals in the style of Amazon.com could easily be developed.

Porcel, Moreno, & Herrera-Viedma (2009) proposed a multi-disciplinary recommender system to advise research resources in university digital libraries. In this study, they analyzed the logical extensions of traditional libraries in the information society. It was stated that recommender systems are tools whose objective is to evaluate and filter the great amount of information available on the web to assist the users in their information access processes. They presented a model of a fuzzy linguistic recommender system to help the university digital library users to access their research resources. This system recommended researchers specialized and complementary resources in order to discover collaboration possibilities to form multi-disciplinary groups. In this way, this system increased social collaboration possibilities in a university framework and contributed to improving the services provided by a university digital library.

Smeaton & Callan (2005) proposed personalization and recommender systems in digital libraries. They reasoned that a simple search function, normally an integral part of any digital library, leads to user frustration as user needs become more complex and as the volume of managed information increases. They mentioned that proactive digital libraries, where the library evolves from being passive and untailored, were seen as having a great potential for addressing and overcoming these issues and included techniques such as personalization and recommender systems. In addition, authors outlined the working group’s vision for the evolution of digital libraries and the role that personalization and recommender systems will play, and they presented a series of research challenges and specific recommendations and research priorities for the field.

Jia & Shi (2013) presented a library management system which is based on content filtering and collaborative filtering. From the perspective of system application and design, the study designed the structure, function module, and user interface of the book recommendation system. Fu, Zhang, & Seinmin (2013) proposed a recommender system for the university library based on their experimental results with five million users' borrowing records. The necessity of a recommender system for university libraries; collaborative filtering technique was applicable and feasible; user-based CF technique was preferred over item-based; the performance of applying classical user-based collaborative filtering algorithm; the effectiveness of local recommendation and the great saving of computing resource were examined.

## 2.4 CONCLUSION

In this chapter, we discussed the background of the recommender system in general and the different methods and formulas used in the traditional recommender system from accredited journals. We discussed and reviewed techniques used in the recommender system domain including collaborative filtering, user-based nearest neighbor algorithm, item based nearest neighbor algorithm, content-based filtering, hybrid recommendation, demographic-based filtering and knowledge-based recommendation. Finally, this chapter discussed how this study is related to other studies previously done in context of recommender systems for libraries of public and private institutions using different methodologies including the one we used in the implementation phase of this study.

# CHAPTER 3: RESEARCH METHODOLOGY

## 3.1 INTRODUCTION

A proactive university library book recommender system is a system that proactively recommends books which are available in the university library. The proposed system is accommodating in overcoming most problems related with information overload within the current administration system of book collection. The proposed system more particularly helps the users to choose books in a library that contain the topic of their interest. The algorithm predicts the ratings of all users for unrated books and the books with highest rating are recommended to the users. This chapter used a hybrid recommender approach, which encompasses both a collaborative filtering technique and a content-based filtering technique. This chapter studies how the use of a combination of both techniques helps to reduce the drawbacks of both techniques to make the recommendations more accurate.

## 3.2 PROPOSED HYBRID APPROCH

The research study used the hybrid filtering approach that encompasses both a collaborative filtering technique and a content-based filtering technique as shown in figure 3.1. The use of a combination of both techniques helps to reduce the drawbacks of both techniques and makes the recommendations closer to reality. Similarity of books is computed using content-based a filtering technique and rating prediction for unrated books was done using the collaborative filtering technique. The books with highest predicted ratings together with other similar books are recommended to a user. The hybrid recommendation is demonstrated in figure 3.1.

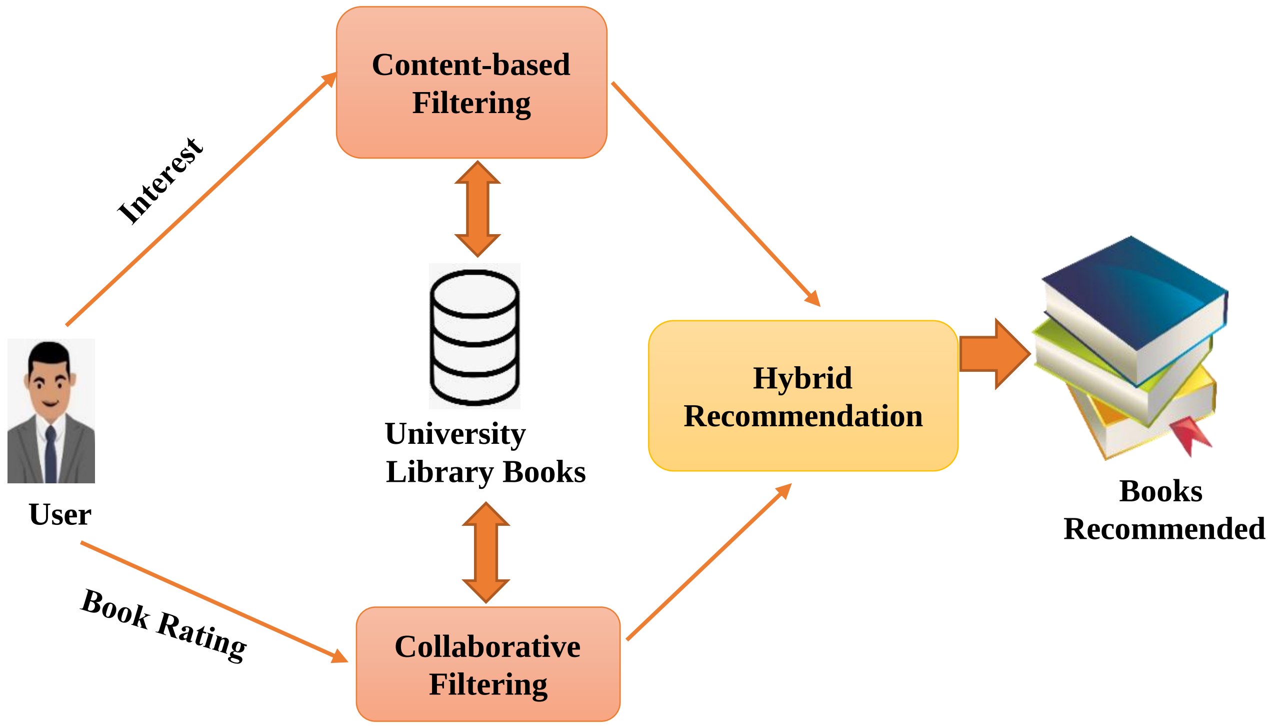


Figure 3.1: Techniques of hybrid recommender system

The system used similarity fusion to compute the similarity between different library book users and enables it to predict probable ratings for unrated books by the user.

We first used the content-based filtering method to compute the similarity between the books that are found in the library. After the system computed the similarity between each book, ranked, and stored them based on their similarity value using content-based filtering method, it used the collaborative filtering method to compute the similarity between different library book users and to predict probable ratings for unrated books by the users which enabled the system to give a good recommendation for users. Top 10 similar books that were ranked using the content-based method were used to recommend to the user that the system predicted to rate a similar book to the ranked books using the collaborative method.

Steps of the hybrid algorithm:

1. Compute similarity among books in the library using content-based technique
2. Predict the rating of unrated books using collaborative filtering technique

3. Recommend books with high predicted ratings from collaborative filtering algorithm together with books similar to them which are computed from content-based technique.

Both collaborative and content-based methods are described in the next sections.

## 3.3 CONTENT-BASED METHOD

The content-based recommender method analyses book details and picks out and recommends items in which the users are interested. In this case the content-based algorithm is used to compute similarity of books.

The steps used in content-based methods:

1. Measure term frequency and inverse document frequency scores

2. Measure weighted term vector

3. Use vector space model to determine similar books using cosine equation then ranking of similar books using the calculated values.

### 3.3.1 TERM FREQUENCY AND INVERSE DOCUMENT FREQUENCY (TF-IDF)

Term frequency and document frequency analysis (TF-IDF) assesses how relevant a word is to a document by counting the number of times it shows up in a document set.

Term frequency (tf) is a number of term repetitions in a book. Inverse document frequency (IDF) is the inverse of the document frequency (DF) in the whole document and shows how rare or common a term is in the whole document. The more it is close to zero (0), the more the term is common. If a term is rare and does not appear in many documents, the number approaches to one.

Using logarithm is helpful to diminish high recurrence words. It helps to dampen unessential terms like ‘to’, ‘the’, ‘very’, etc.… which are frequently used in a sentence. To demonstrate how this is working, a demonstration of the number of term repetitions in different books (Term frequency) and the number of term repetitions in the whole document (document frequency) were presented in the form of a table.

Number of term repetitions in different books (term frequency) and the number of term repetitions in the whole document (document frequency are demonstrated in table 3.1.

**Table 3. 1** Demonstration of number of term repetitions in different books (Term frequency) and number of term repetitions in the whole document (document frequency)

| Articles | Organic | Programming | Bonding | Better | Software | Economists | Audit | Finance |
|---|---|---|---|---|---|---|---|---|
| Into to ICT | 0 | 1500 | 0 | 70 | 30000 | 1 | 0 | 4 |
| Learning Java 2000 | 0 | 5000 | 0 | 60 | 4000 | 0 | 0 | 0 |
| General Chemistry | 2000 | 0 | 3000 | 80 | 1 | 1 | 0 | 1 |
| Fraud Examination | 0 | 0 | 1 | 90 | 2 | 2000 | 3000 | 2500 |
| Basic Economics | 0 | 0 | 0 | 80 | 1 | 5000 | 2500 | 2500 |
| Organic Chemistry 2012 | 5000 | 0 | 4000 | 70 | 0 | 0 | 0 | 0 |
| C++ Primer | 0 | 4500 | 0 | 60 | 4300 | 0 | 3000 | 3500 |
| The Periodic Table | 3000 | 0 | 1900 | 50 | 0 | 0 | 0 | 0 |
| DF | 200,000 | 400,000 | 100, 000 | 800,000 | 350,000 | 150,000 | 50, 000 | 90,000 |

In the next sections, we describe how term-frequency and inverse document frequency were used in phases in our algorithm.

### 3.3.1.1 TERM FREQUENCY

Term frequency is the frequency of a word in a document.

Phase 1: We calculated term frequency in the first phase and we used a logarithm to dampen high frequency words. Term frequency is measured using the equation 3.2.

$$tf = 1 + Log_{10}(TR) \tag{3.2}$$

where tf represents term frequency and TR term repetition (Kim, Sang-Woon, & Gil, 2019).

By applying the values given in table 3.1 on the above equation, term frequency measured values were listed in table 3.2.

**Table 3.2** Calculating term frequency

| ARTICLES | ORGANIC | PROGRAMMING | BONDING | BETTER | SOFTWARE | ECONOMISTS | AUDIT | FINANCE |
|---|---|---|---|---|---|---|---|---|
| INTO TO ICT | 0 | 4.176091 | 0 | 2.84509 | 5.47712 | 1 | 0 | 1.602059 |
| LEARNING JAVA 2000 | 0 | 4.698970 | 0 | 2.77815 | 4.602059 | 0 | 0 | 0 |
| GENERAL CHEMISTRY | 4.301029 | 0 | 4.477121 | 2.90308 | 1 | 1 | 0 | 1 |
| FRAUD EXAMINATION | 0 | 0 | 1 | 2.95424 | 1.301029 | 4.301029 | 4.477121 | 4.397940 |
| BASIC | 0 | 0 | 0 | 2.90308 | 1 | 4.698970 | 4.397940 | 4.397940 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Economics | | | | | | | | |
| Organic Chemistry 2012 | 4.698970 | 0 | 4.602059 | 2.84509 | 0 | 0 | 0 | 0 |
| C++ Primer | 0 | 4.653212 | 0 | 2.77815 | 4.633468 | 0 | 4.477121 | 4.544068 |
| The Periodic Table | 4.477121 | 0 | 4.278753 | 2.69897 | 0 | 0 | 0 | 0 |

The values make up the attribute vector for each book.

#### 3.3.1.2 INVERSE DOCUMENT FREQUENCY

IDF is the inverse of the frequency of the document for all records.

IDF was determined by measuring the inverse logarithm of document frequency between the entire documents (N). It is measures using the equation below.

$$IDF = log_{10} \frac{N}{DF} \tag{3.3}$$

where IDF represents Inverse document frequency, N represents the whole documents and DF represents document frequency (Nguyen & Eric, 2014).

IDF measured values are demonstrated in the table 3.3.

**Table 3.3** Demonstration of IDF calculated value

| Articles | Organic | Programming | Bonding | Better | Software | Economists | Audit | Finance |
|---|---|---|---|---|---|---|---|---|
| DF | 200,000 | 400,000 | 100,000 | 800,000 | 350,000 | 150,000 | 50,000 | 90,000 |
| IDF | 1.176091 | 0.875061 | 1.477121 | 0.574031 | 0.933053 | 1.301030 | 1.778151 | 1.522878 |

| N= | 2,140,000 |
|---|---|

We used the vector space model to represent each book as a vector and to calculate the cosine similarity values of the angles between the vectors.

### 3.3.2 VECTOR SPACE MODEL

Vector space model is utilized to decide which books are more similar to the other books. Vector space model is a numeric model that represents text documents as vectors and it is used to filter, retrieve, index and rank significance of data by computing the vicinity based on the angle between vectors. Books were saved as a vector of their attributes and the points between the vectors were calculated to decide the closeness between vectors. Vectors' lengths are measured as the square root of the summation of squared values of every attribute in the vector. Vector normalizing is done finally by dividing term vector by length of vector, and the cosine values of books are considered as the similarity measures between books. The cosine value of books is calculated as a sum product of a normalized term from both books. Cosine is used because the cosine value decreases with the increasing value of the angle that shows less similarity and vice versa.

Phase 2: in this phase, we calculated length of vector for a book that is the square root of summation of term frequencies square. Calculating the length of vector was done in order to represent each value as a vector that is later used to normalize the vector.

$$LV = \sqrt{\sum_i \left(tf_{t,d}\right)^2} \quad (3.4)$$

where LV represents length of vector, tf represents term frequency (Tata & Patel, 2007). Lengths of vector measured values are demonstrated in the table below.

**Table 3.4** Demonstration of length of vector values

| ARTICLES | ORGANIC | PROGRAMMING | BONDING | BETTER | SOFTWARE | ECONOMISTS | AUDIT | FINANCE | VL |
|---|---|---|---|---|---|---|---|---|---|
| INTO TO ICT | 0 | 4.17609 | 0 | 2.84509 | 5.47712 | 1 | 0 | 1.60205 | 7.68763 |
| LEARNING JAVA 2000 | 0 | 4.69897 | 0 | 2.77815 | 4.60205 | 0 | 0 | 0 | 7.13984 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| General Chemistry | 4.30102 | 0 | 4.47712 | 2.90308 | 1 | 1 | 0 | 1 | 7.06904 |
| Fraud Examination | 0 | 0 | 1 | 2.95424 | 1.30102 | 4.30102 | 4.47712 | 4.39794 | 8.32499 |
| Basic Economics | 0 | 0 | 0 | 2.90308 | 1 | 4.69897 0 | 4.39794 0 | 4.39794 | 8.37806 |
| Organic Chemistry 2012 | 4.69897 | 0 | 4.60205 | 2.84509 | 0 | 0 | 0 | 0 | 7.16616 |
| C++ Primer | 0 | 4.65321 | 0 | 2.77815 | 4.63346 | 0 | 4.47712 1 | 4.54406 | 9.56727 |
| The Periodic Table | 4.47712 | 0 | 4.27875 | 2.69897 | 0 | 0 | 0 | 0 | 6.75550 |

Phase 3: In this phase we normalized vectors to apply the cosine similarity algorithm.

$$NV = \frac{tf}{LV} \tag{3.5}$$

where NV represents normalized vector, tf represents term frequency and LV is length of vector (Singhal, Buckley, & Mitra, 2017).

Normalized vector of Into to ICT is computed as follows:

$$NV\ (\text{Into to ICT}) = \frac{tf(\text{Into to ICT})}{LV(\text{Into to ICT})} = \frac{4.17609}{7.68763} = 0.54322$$

Normalized vector measured values are demonstrated in the following table.

**Table 3.5** Normalized vector

| Articles | Organic (N) | Programming (N) | Bonding (N) | Better (N) | Software (N) | Economists (N) | Audit (N) | Finance (N) | LV | Summation of (NV) |
|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Into to ICT | 0 | 0.54322 | 0 | 0.37008 | 0.71245 | 0.13007 | 0 | 0.20839 | 7.68764 | 1 |
| Learning Java 2000 | 0 | 0.65813 | 0 | 0.38910 | 0.64456 | 0 | 0 | 0 | 7.13985 | 1 |
| General Chemistry | 0.60843 | 0 | 1.54219 | 0.41067 | 0.14146 | 0.14146 | 0 | 0.14146 | 7.06905 | 1 |
| Fraud Examination | 0 | 0 | 0.3385 | 0.35486 | 0.15627 | 0.51664 | 0.53779 | 0.52828 | 8.325 | 1 |
| Basic Economics | 0 | 0 | 0 | 0.34651 | 0.11935 | 0.56086 | 0.52493 | 0.52493 | 8.37807 | 1 |
| Organc Chemistry 2012 | 0.65572 | 0 | 1.61754 | 0.39701 | 0 | 0 | 0 | 0 | 7.16616 | 1 |
| C++ Primer | 0 | 0.4863 | 0 | 0.29038 | 0.48430 | 0 | 0.46796 | 0.47495 | 9.56727 | 1 |
| The Periodic Table | 0.66274 | 0 | 1.58533 | 0.39952 | 0 | 0 | 0 | 0 | 6.7555 | 1 |

After finding the normalized value of each vector, the similarities between contents were calculated using cosine similarity between the normalized vectors.

### 3.3.2.1 COSINE SIMILARITY OF TF.DF VECTORS

In this method, the cosine similarity algorithm was used to measure the cosine value of two vectors. The more the cosine value is closer to one, the more the vectors are similar to each other since the value of the cosine is closer to one when the angles between two vectors are smaller. In other words, the more the angles between the two vectors are smaller, the more the two vectors are similar.

Phase 4: In this phase, we measured book similarity values using the equation of cosine similarity. The cosine value of the two vector are computed as a sum product of normalized vectors. The equation in which the similarities are calculated is described in the equation below:

$$\mathrm{Similarity}_{a,b} = \sum_{i=1}^{n} (\mathrm{NV}_{ai} * \mathrm{NV}_{bi})$$

$$\mathrm{Similarity}_{a,b} = \mathrm{NV}_{a1} * \mathrm{NV}_{b1} + \mathrm{NV}_{a2} * \mathrm{NV}_{b2} + \mathrm{NV}_{a3} * \mathrm{NV}_{b3} + \cdots\ \mathrm{NV}_{an} * \mathrm{NV}_{bn} \quad (3.6)$$

where two book vectors were represented as 'a' and 'b', where Numbers from 1 – n are keywords and NV represents normalized vectors (Tata & Patel, 2007)

Ranked similarity values demonstration is listed in table 3.6.

**Table 3.6** Example of ranked similarity values demonstration

| BOOK VECTOR | COSINE VALUE |
|---|---|
| **INTO TO ICT** | |
| COS(INTO TO ICT, LEARNING JAVA) | 0.960737381 |
| COS(INTO TO ICT, C++ PRIMER) | 0.815696973 |
| COS(INTO TO ICT, FRAUD EXAMINATION) | 0.419968542 |
| COS (INTO TO ICT, BASIC ECONOMICS) | 0.395628071 |
| COS (INTO TO ICT, GENERAL CHEMISTRY) | 0.300652843 |

| | |
|---|---|
| COS (INTO TO ICT, THE PERIODIC TABLE) | 0.147858011 |
| COS (INTO TO ICT, ORGANIC CHEMISTRY 2012) | 0.146931538 |
| **LEARNING JAVA** | |
| COS (LEARNING JAVA, INTO TO ICT) | 0.960737381 |
| COS (LEARNING JAVA, C++ PRIMER) | 0.745246437 |
| COS (LEARNING JAVA, GENERAL CHEMISTRY) | 0.250976958 |
| COS (LEARNING JAVA, FRAUD EXAMINATION) | 0.238811278 |
| COS (LEARNING JAVA, BASIC ECONOMICS) | 0.21176336 |
| COS (LEARNING JAVA, THE PERIODIC TABLE) | 0.155456033 |
| COS (LEARNING JAVA, ORGANIC CHEMISTRY 2012) | 0.154481951 |

These ranked similarity values are later used for recommendations for similar books after we predict ratings for users using the collaborative filtering method.

## 3.4 COLLABORATIVE FILTERING METHOD

The collaborative filtering (CF) method was used to filter and predict the users' preferences by using other users' preferences (Ghadling, Belavad, Bhegade, Ghojage, & Supriya, 2015). The principle behind the collaborative technique is that, if users rated some other books similarly, a user's rating for a new book is definitely going to be similar. The description or Meta data of the book is not needed in collaborative filtering in order to make recommendations. In this technique, we measured the ratings of different users by using similarity measures used to predict ratings. This was done by identifying similar neighbors closest to each other and making recommendations.

A cosine similarity algorithm was applied for calculating the similarity between users. The cosine similarity algorithm was chosen over other algorithms because it is very efficient to evaluate the results obtained (Liu, Hu, Mian, Tian, & Zhu, 2014). The cosine similarity algorithm calculates the measure of cosine value of two vectors. The more the value is close to one (1) the more the vectors are similar to each other since the value of cosine approximates to one (1) when the angles between two vectors are smaller. Pearson correlation was used to determine the prediction rating of the user. Both the cosine similarity algorithm and Pearson correlation are described in detail in the next sections.

In this method we calculated how similar users are to one another based on their ratings for the books. Taking the first user and comparing him/her to the other users, we computed how similar the first user was to the rest of them using cosine similarity. We then calculated the prediction rating of the users for books using Pearson correlation.

The steps of the algorithm used are listed below:

1. Collect rating values from different users and represent them in a matrix form

2. Calculate cosine similarity between users.

3. Predict the value of unrated books

4. Recommend high predicted ratings to user

### 3.4.1 COLLECT RATING VALUES FROM DIFFERENT USERS AND REPRESENT THEM IN A MATRIX FORM

Step 1: We wrote user's book rating in matrix form

In this step, we represented and listed every user as a vector (or array) that contained the user preferences for books. The higher the similarity between vectors, the more similar they are to one another. After listing users, books and rating values on the list format, we wrote users' book rating values in a matrix form.

A sample of a small dataset was taken to demonstrate users' rating in a matrix form in the table below.

**Table 3.7** Users' book rating values in matrix form

| Into to ICT (ICT) | Learning Java 2002 (LJ) | General Chemistry (GC) | Fraud Examination (FE) | Basic Economics (BE) | Organic Chemistry 2012(OC) | C++ Primer (C++) | The periodic Table (PT) |
|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| U1 | ? | 5 | 1 | ? | 2 | 1 | ? | 1 |
| U2 | 2 | 1 | 5 | 2 | 2 | 4 | 1 | ? |
| U3 | 4 | ? | 1 | 1 | 1 | 1 | 5 | ? |
| U4 | 2 | 1 | ? | 1 | 2 | 4 | 1 | 5 |
| U5 | 2 | 1 | 2 | 4 | ? | 2 | 1 | 1 |
| U6 | 2 | 1 | 2 | ? | 5 | 2 | 1 | 1 |

The above sample data was used in the next sections to show how each method was applied.

Step 2: In this phase we created a user based similarity matrix to calculate similarity between users. We describe how we used the cosine similarity algorithm in the next section.

### 3.4.2 COSINE SIMILARITY

We used cosine similarity algorithm to calculate the measure of cosine value of two vectors. The more the cosine value is near to one, the more the vectors are similar to each other since the value of cosine slanted to one when the points between two vectors are smaller.

A zero (0) angle with cosine value 1 is a maximum similarity while a ninety (90) degree angle where the cosine value is 0 shows null similarity.

The equation to calculate cosine similarity is given below:

$$\mathbf{cos(V1, v2)} = \frac{\sum_{i=1}^{n} \mathbf{v1_i v2_i}}{\sqrt{\sum_{i=1}^{n} \mathbf{v1_i}^2} \sqrt{\sum_{i=1}^{n} \mathbf{v2_i}^2}} \tag{3.8}$$

where two vectors, V1 for user1 (U1) and V2 for User2 (U2) were formed with in the book space of (Int to

ICT, Learning Java (Lee, Park, Shim, & Lee, 2010) . Then we found the cosine of angles between vectors.

By taking sample data from Table 3.1, we calculated the similarity between each user as demonstrated below using equation 3.9 above.

$$\cos(\mathrm{V}_1, \mathrm{V}_2) = \frac{(\mathrm{R}_{\mathrm{B}_1\mathrm{U}_1} * \mathrm{R}_{\mathrm{B}_1\mathrm{U}_2} + \mathrm{R}_{\mathrm{B}_2\mathrm{U}_1} * \mathrm{R}_{\mathrm{B}_2\mathrm{U}_2})}{\sqrt{[(\mathrm{R}_{\mathrm{ICTU}_1}{}^2 + \mathrm{R}_{\mathrm{LJU}_1}{}^2) * (\mathrm{R}_{\mathrm{ICTU}_2}{}^2 + \mathrm{R}_{\mathrm{LJU}_2}{}^2)]}} \qquad (3.9)$$

Similarity (U1, U2)

$\mathrm{V}_1 = 5\mathrm{LJ} + 1\mathrm{GC} + 2\mathrm{BE} + 1\mathrm{OC} + 5\mathrm{C}^{++}$

$$\mathrm{V}_2 = 1\mathrm{LJ} + 5\mathrm{GC} + 2\mathrm{BE} + 4\mathrm{OC} + 1\mathrm{C}^{++}$$

$$\cos(\mathrm{V}_1, \mathrm{V}_2) = \frac{(5 * 1 + 1 * 5 + 2 * 2 + 1 * 4 + 5 * 1)}{\sqrt{[(5^2 + 1^2 + 2^2 + 1^2 + 5^2) * (1^2 + 5^2 + 2^2 + 4^2 + 1^2)]}} = 0.448316$$

Similarity (U1, U3)

$\mathrm{V}_1 = 1\mathrm{GC} + 2\mathrm{BE} + 1\mathrm{OC} + 5\ \mathrm{C}^{++}$

$\mathrm{V}_3 = 1\mathrm{GC} + 1\mathrm{BE} + 1\mathrm{OC} + 5\ \mathrm{C}^{++}$

$$\cos(\mathrm{V}_1, \mathrm{V}_3) = \frac{(1 * 1 + 2 * 1 + 1 * 1 + 5 * 5)}{\sqrt{[(1^2 + 2^2 + 1^2 + 5^2) * (1^2 + 1^2 + 1^2 + 5^2)]}} = 0.984324$$

Similarity (U5, U6)

$\mathrm{V}_5 = 2\mathrm{ICT} + 1\mathrm{LJ} + 2\mathrm{GC} + 2\mathrm{OC} + 1\mathrm{C}^{++} + 1\mathrm{PT}$

$\mathrm{V}_6 = 2\mathrm{ICT} + 1\mathrm{LJ} + 2\mathrm{GC} + 2\mathrm{OC} + 1\mathrm{C}^{++} + 1\mathrm{PT}$

$$\cos(\mathrm{V}_5, \mathrm{V}_6) = \frac{(2 * 2 + 1 * 1 + 2 * 2 + 2 * 2 + 1 * 1 + 1 * 1)}{\sqrt{[(2^2 + 1^2 + 2^2 + 2^2 + 1^2 + 1^2) * (2^2 + 1^2 + 2^2 + 2^2 + 1^2 + 1^2)]}} = 1$$

After we computed the cosine similarity of each vector, we put users' similarity values as a matrix form in order to make them ready for prediction as it shown in table 3.8.

**Table 3.8** User based similarities among users in matrix form

| | U1 | U2 | U3 | U4 | U5 | U6 |
|---|---|---|---|---|---|---|
| U1 | 1.0000 | 0.448316 | 0.984324 | 0.448316 | 0.621236 | 0.551888 |
| U2 | 0.448316 | 1.0000 | 0.527436 | 0.983819 | 0.818095 | 0.762362 |
| U3 | 0.984324 | 0.527436 | 1.0000 | 0.591312 | 0.587887 | 0.587887 |
| U4 | 0.448316 | 0.983819 | 0.591312 | 1.0000 | 0.638888 | 0.676802 |
| U5 | 0.621236 | 0.818095 | 0.587887 | 0.638888 | 1.0000 | 1.0000 |
| U6 | 0.551888 | 0.762362 | 0.587887 | 0.676802 | 1.0000 | 1.0000 |

Step 3: In this phase, we predicted ratings of unrated books by different users. The prediction method is described in the following section.

### 3.4.3 PREDICTION

Pearson correlation was used to determine the prediction rating of the active user ‘a’ for book item ‘i’. The weighted average of differences from the neighbor's mean was estimated as predictions:

Pearson correlation was used to determine the prediction rating of the active user ‘a’ for book item ‘i’. Predictions were computed as the weighted average of deviations from the neighbor’s mean. The equation to calculate prediction is given in the equation 3.9 below.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times sim_{a,u}}{\sum_{u=1}^{n} p_{a,u}} \tag{3.9}$$

where $p_{a,i}$ is the prediction for the active user ‘a’ for item ‘i’; $r_{u,i}$ is the rating given to item i by user u; $\bar{r}_a$ is the mean rating given by user a; $p_{a,u}$ is the similarity between users ‘a’ and ‘u’; and ‘n’ is the number of users in the neighborhood (Content-Boosted Collaborative Filtering for Improved Recommendations, 2002).

We demonstrated how the prediction was computed for User 1 using equation 3.9.

Prediction rating of User 1 for ICT book is computed as followed.

$$p_{U1,ICT} = \bar{r}_{u1} + \frac{\sum_{u=1}^{6}\left(r_{u,i} - \bar{r}_u\right) \times sim_{U1,u}}{\sum_{u=1}^{n} p_{U1,u}}$$

$$p_{U1,ICT} \\ = 2.5 \\ + \frac{[(2 - 2.428571) \times 0.448316] + [(4 - 2.166666) \times 0.984324] + [(2 - 2.285714) \times 0.448316] + [(2 - 2.285714) \times 0.621236] + [(2 - 2.000000) \times 0.551888]}{0.448316 + 0.984324 + 0.448316 + 0.621236 + 0.551888}$$

$$= 2.5 + \frac{-1.0714275 + 1.804594656216 + -0.128090157624 + -0.177495822504 + 0}{3.05408}$$

$p_{U1,ICT}$ = 2.894700901871604~ 3.0

Prediction rating of User 1 for Fraud Examination book is computed as followed.

$$p_{U1,FE} = \bar{r}_{u1} + \frac{\sum_{u=1}^{6}\left(r_{u,FE} - \bar{r}_u\right) \times sim_{U1,u}}{\sum_{u=1}^{n} p_{U1,u}}$$

$$p_{U1,FE} \\ = 2.5 \\ + \frac{[(2 - 2.428571) \times 0.448316] + [(1 - 2.166666) \times 0.984324] + [(1 - 2.285714) \times 0.448316] + [(4 - 2.285714) \times 0.621236]}{0.448316 + 0.984324 + 0.448316 + 0.621236}$$

$$= 2.5 + \frac{-0.192135236436 - 1.148377343784 - 0.576406157624 + 1.064976}{2.502192}$$

$p_{U1,FE}$ =-2.159521 ~ 2.2

We described how the final prediction was put in a matrix form. Two sample predictions that were calculated above are highlighted in the following table.

**Table 3.9** Two sample predicted ratings of a book by different users as a matrix form

| | Into to ICT (ICT) | Learning Java 2002 (LJ) | General Chemistry (GC) | Fraud Examination (FE) | Basic Economics (BE) | Organic Chemistry 2012(OC) | C++ Primer (C++) | The periodic Table (PT) |
|---|---|---|---|---|---|---|---|---|
| U1 | 3.0 | 5 | 1 | 2.2 | 2 | 1 | ? | 1 |
| U2 | 2 | 1 | 5 | 2 | 2 | | 1 | |
| U3 | 4 | ? | 1 | 1 | 1 | 1 | 5 | ? |
| U4 | 2 | 1 | ? | 1 | 2 | 4 | 1 | 5 |
| U5 | 2 | 1 | 2 | 4 | ? | 2 | 1 | 1 |
| U6 | 2 | 1 | 2 | ? | 5 | 2 | 1 | 1 |

Step 4: In this final step we recommend books with high predicted ratings to the users.

## 3.5 RECOMMENDATIONS

This section explains the recommendation algorithm step by step.

ALGORITHM

1. Select N-top rated books from the collaborative algorithm
2. Retrieve M-numbers of books similar to each of the N-top rated books
3. Recommend to the user the N+M books.

4. End

## 3.6 EVALUATION MEASURE

In this section, the performance or accuracy of the proposed university library recommender systems is evaluated through quantitative analysis. The evaluation models utilized as quantitative schemes are clarified below. The evaluation matrix of recommender system was utilized to measure the performance of the system prediction by getting the error during implementation.

Most common metrics to evaluate the quality of recommendation are mean absolute error (MAE), root mean squared error (RMSE), precision and recall.

**Precision:** This is the division of relevant items suggested to the items within the recommendation list. Precision shows the capacity to yield significant items. Precision answers how numerous recommended items are pertinent among the given recommendations by prototype.

**Recall**: This is the fraction of items recommended to be relevant by the user to the relevant items. Recall answers a question as to how many relevant items are recommended.

In our work, mean absolute error (MAE) and root mean squared error (RMSE) were used to evaluate the accuracy of the result by finding the difference between the actual ratings given by the user and the estimation ratings of the system using the algorithm.

We chose MAE and RMSE because they are the best methods to find the exact value of the difference between the actual rating and the estimated rating by the users. The combined evaluation of MAE and RMSE gives a better evaluation of the results.

We explained the evaluation process in chapter 4 by taking a test user for a sample book from a dataset. After training the data model and got the predicted rating for that test user for the above mentioned book, we calculated the difference between the actual rating and the predicted one.

### 3.6.1 MEAN ABSOLUTE ERROR (MAE)

MAE measures the average of the absolute difference between the predicted rating and the actual rating given by the users in the system.

Mean absolute error (MAE) is measured using statistical accuracy matrix. MAE is calculated using the formula below:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} p_{i-q_i}}{n} \tag{3.10}$$

where MAE is defined as average absolute difference between n pairs. $p_1 \ldots. p_n$ are predicated ratings; $q_1 \ldots q_n$ are actual ratings and n is amount of ratings. Lower the MAE, the more accurate the prediction is (Suhasini & Joshi, 2015).

### 3.6.2 ROOT MEAN SQUARED ERROR (RMSE)

RMSE measures the average size of the error by calculating and finding the square root of the average of squared differences between the prediction given by the prototype and the actual rating. Statistical accuracy metrics was used where mean absolute error (MAE) is determined.

RMSE gave a relatively bigger weight amount to large errors because the errors were squared before they were averaged.

RMAE was calculated using the formula below:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(p_{i-q_i}\right)^2}{n}} \tag{3.11}$$

where $P_1 \ldots. P_n$ represent predicated ratings; $q_1 \ldots q_n$ represents actual ratings and n is amount of ratings (Kamble & Deshmukh, 2017).

The evaluation process is applied and explained in chapter 4.

## 3.7 CHAPTER SUMMARY

This chapter explained in detail about the methods used to recommend books for library users. The proposed methods and their function were seen in detail. We explained how we used the combination of both collaborative and content-based methods. In the first phase, we used a content-based filtering technique by taking users' queries to calculate book similarity using TF-IDF weighting and the vector space model. We measured similarity of books and ranked them in similarity order using a content-based filtering method. We later used the collaborative filtering method to compute the similarity between different library book users and to predict probable ratings for unrated books by the users. Top 10 similar books that were ranked using the content-based method were used to make recommendations to the user. The system predicted to rate a similar book to the ranked books using the collaborative method. Mean absolute error (MAE) and root mean squared error (RMSE) were used to evaluate the accuracy of the result by finding the difference between the actual ratings given by the user and the estimation ratings of the system using the algorithm.

In the next chapter, experiments and results are evaluated.

# CHAPTER 4: EXPERIMENTAL RESULTS AND EVALUATION

## 4.1 INTRODUCTION

This chapter briefly explains about the experiments which were conducted on the implementation of the proactive university library book recommender system. The results obtained were discussed and the accuracy of the result was evaluated.

The objectives of this experiment are listed below:

- To design a prototype for a proactive university library book recommender system that would recommend books for library users.
- To develop a prototype for a proactive university library book recommender system.
- To evaluate the developed prototype to measure the accuracy.

The methods explained in chapter 3 were applied in this experiment.

## 4.2 EXPERIMENTAL SETUP

The software used and the dataset with their properties are discussed in the next section.

### 4.2.1 SOFTWARE AND HARDWARE SETUP

The experiment was conducted using a powerful Python Library called Surprise Library that was helpful to generate predictions.

The prototype was developed using Python programing language and Java, Windows 10 operating system, Intel (R) core (TM) i5 2410M CPU 2.30ghz, 4 Gig Ram and 500gb hard drive. An open-source web application called Jupyter Notebook and Eclipse Java 2019 were used for coding and running the prototype.

### 4.2.2 DATASET

We used a dataset called book-crossing dataset that has 278,858 users and 271,379 books. In the book crossing dataset, 1,149,780 ratings were used for training data and conducting the experiment. The dataset was taken from the University of Freiburg Department of Computer Science website (http://www2.informatik.uni-freiburg.de/~cziegler/BX/).

Explicit ratings were gathered on the range from zero to ten based on users' appreciation of the books. The dataset statistical information is listed in the table below.

**Table 4. 1** Book crossing dataset statistical information

| | |
|---|---|
| Number of Books | 271,379 |
| Number of Users | 278,858 |
| Number of Ratings | 1,149,780 |

In these experiments, we used a supervised learning technique. Supervised learning is a mechanism of dividing a dataset into training data and testing data. Seventy percent of the data were taken for training and then evaluated by 30% data points that were used as the test set.

The methodology described in chapter 3 was used and an analysis of the results was done. We conducted experiments using the prototype that was developed. The prototype and results are presented in the form of screenshots.

The books with their attributes and rating information of the dataset are illustrated in figure 4.1 and figure 4.2 respectively.

```
In [9]: books = pd.read_csv('BX-Books.csv', sep=';', error_bad_lines=False, encoding="latin-1")
        books.columns = ['ISBN', 'Title', 'Author', 'YrOfPublication', 'Publisher', 'ImageUrlS', 'ImageUrlM', ':
        print(books.info())
        print(books.head())
```

```
Data columns (total 8 columns):
ISBN               271360 non-null object
Title              271360 non-null object
Author             271359 non-null object
YrOfPublication    271360 non-null object
Publisher          271358 non-null object
ImageUrlS          271360 non-null object
ImageUrlM          271360 non-null object
ImageUrlL          271357 non-null object
dtypes: object(8)
memory usage: 16.6+ MB
None
         ISBN                                              Title  \
0  0195153448                                Classical Mythology
1  0002005018                                       Clara Callan
2  0060973129                               Decision in Normandy
3  0374157065  Flu: The Story of the Great Influenza Pandemic...
4  0393045218                             The Mummies of Urumchi

                 Author YrOfPublication                   Publisher  \
0    Mark P. O. Morford            2002     Oxford University Press
1  Richard Bruce Wright            2001       HarperFlamingo Canada
2          Carlo D'Este            1991             HarperPerennial
3      Gina Bari Kolata            1999        Farrar Straus Giroux
4       E. J. W. Barber            1999  W. W. Norton &amp; Company
```

**Figure 4.1** A screen shot of a sample of book information and its attributes

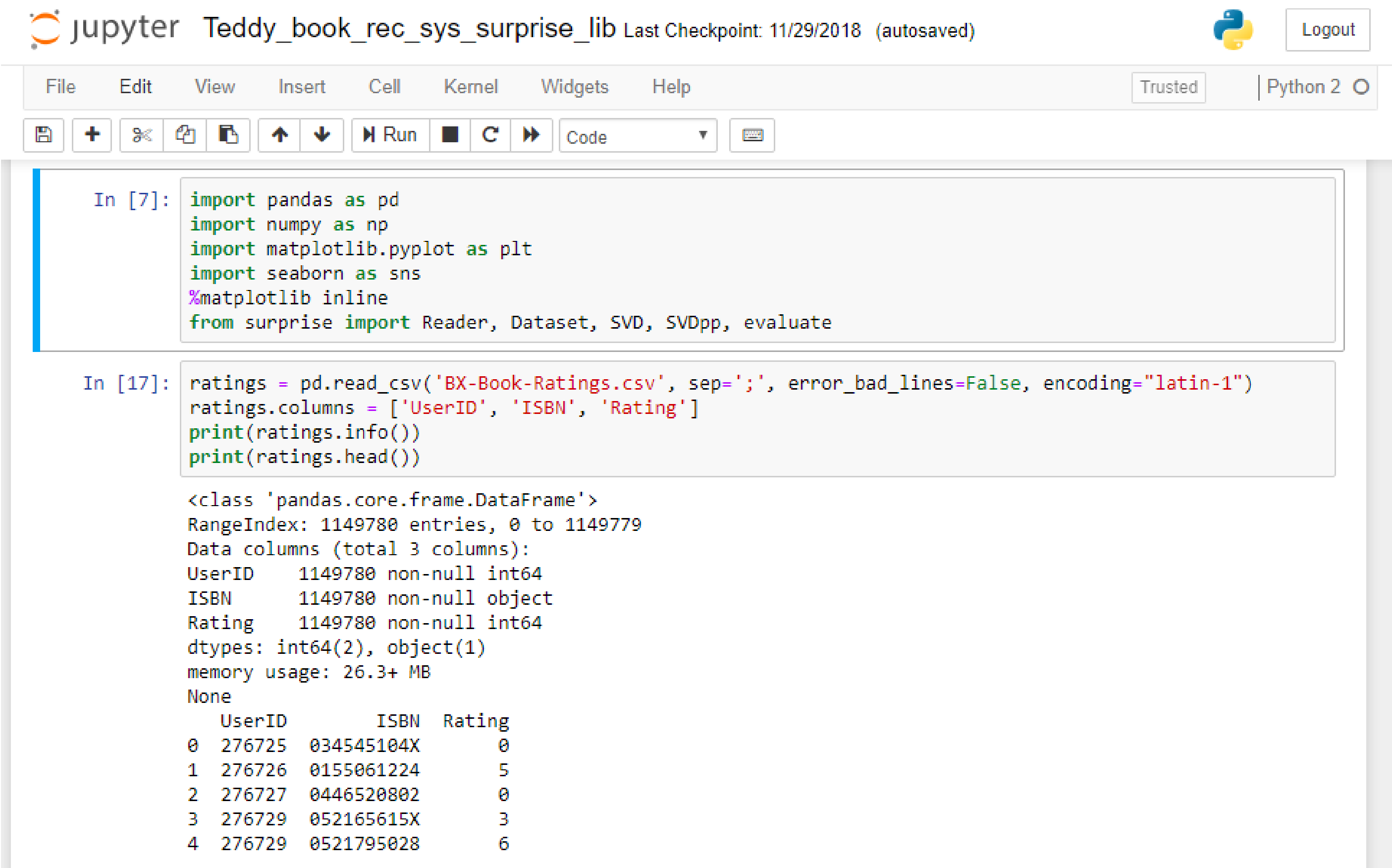


**Figure 4.2** A screen shot of import of dataset and rating information

The above figure shows a list of first five rating information. The rating was given from the range of zero to ten. A rating zero shows total dissatisfaction and a rating 10 shows maximum satisfaction. A number of records per rating was done to show how many times each rating is given. This demonstrated the frequency each rating ranged from zero to ten. Based on distribution of ratings frequency record, a rating value 8 was the biggest frequently rated value that counted for 100,000 times among all rating values. Overall distribution of ratings and frequency records are demonstrated as a screen shot in figure 4.3.

```
In [7]: #Check the overall distribution of ratings
        sns.set(style="whitegrid", font_scale=1.4)
        plt.subplots(figsize=(15,8))
        sns.countplot("Rating",data=ratings).set_title("Number of records per rating")

Out[7]: Text(0.5,1,'Number of records per rating')
```

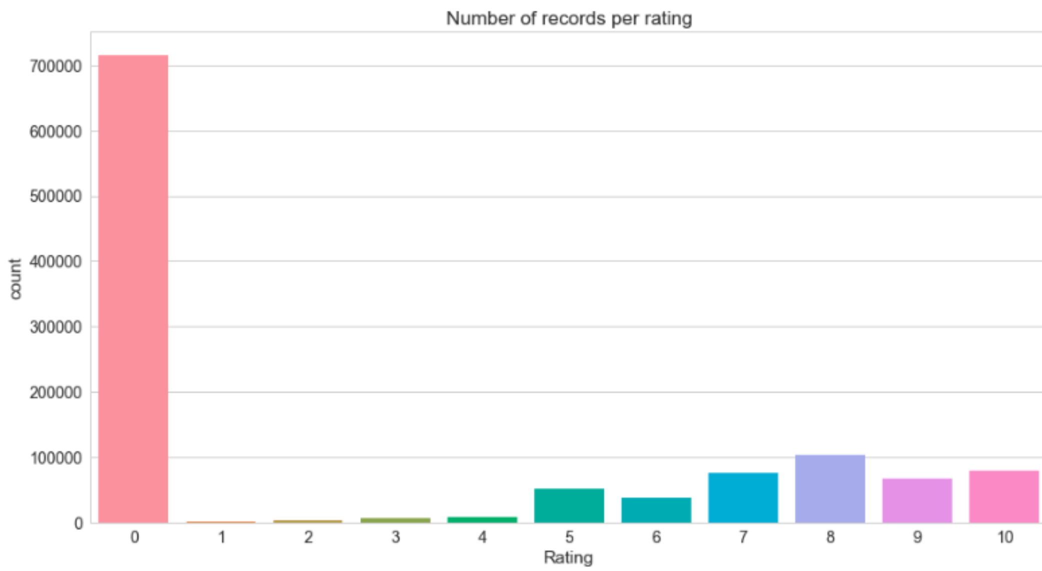


**Figure 4.3** A screen shot that demonstrates overall distribution of ratings frequency records

The prototype developed are presented in a form of screenshots. Figure 4.4 shows user registration interface of the developed prototype.

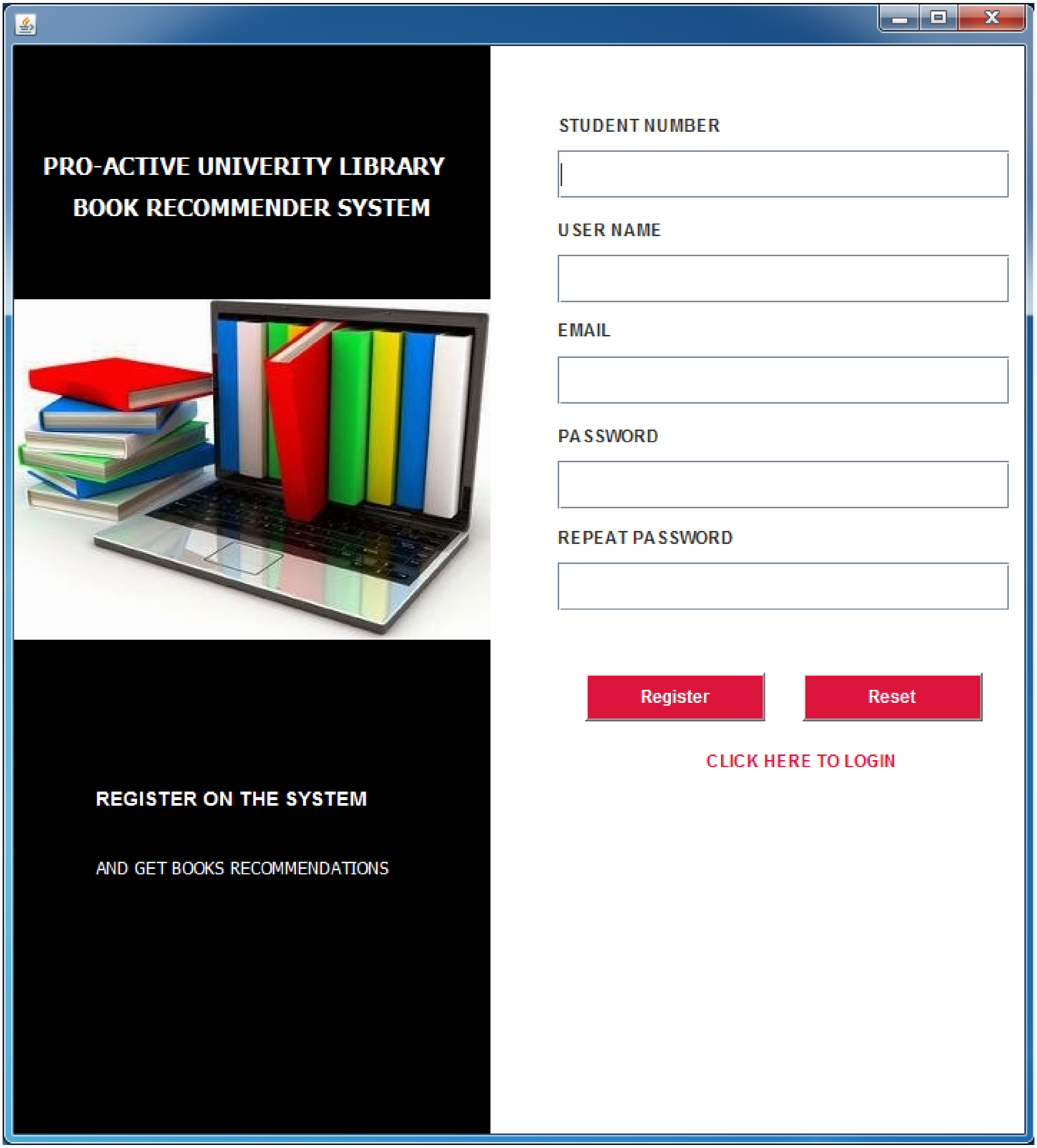


**Figure 4.4** A screenshot of user registration for new users

The above figure is an interface for registering new users. Users register using this register interface by entering their credentials. Once users are registered in the system, the system saves the users' information in the dataset as users. The proactive university library recommender system will not able to give personalized recommendations if users are not registered on the register interface.

There is a login interface for those who are already registered. The system allows the users to enter login information and log in to the system.

However, for already existing users they go straight to the login section of the system shown in figure 4.5.

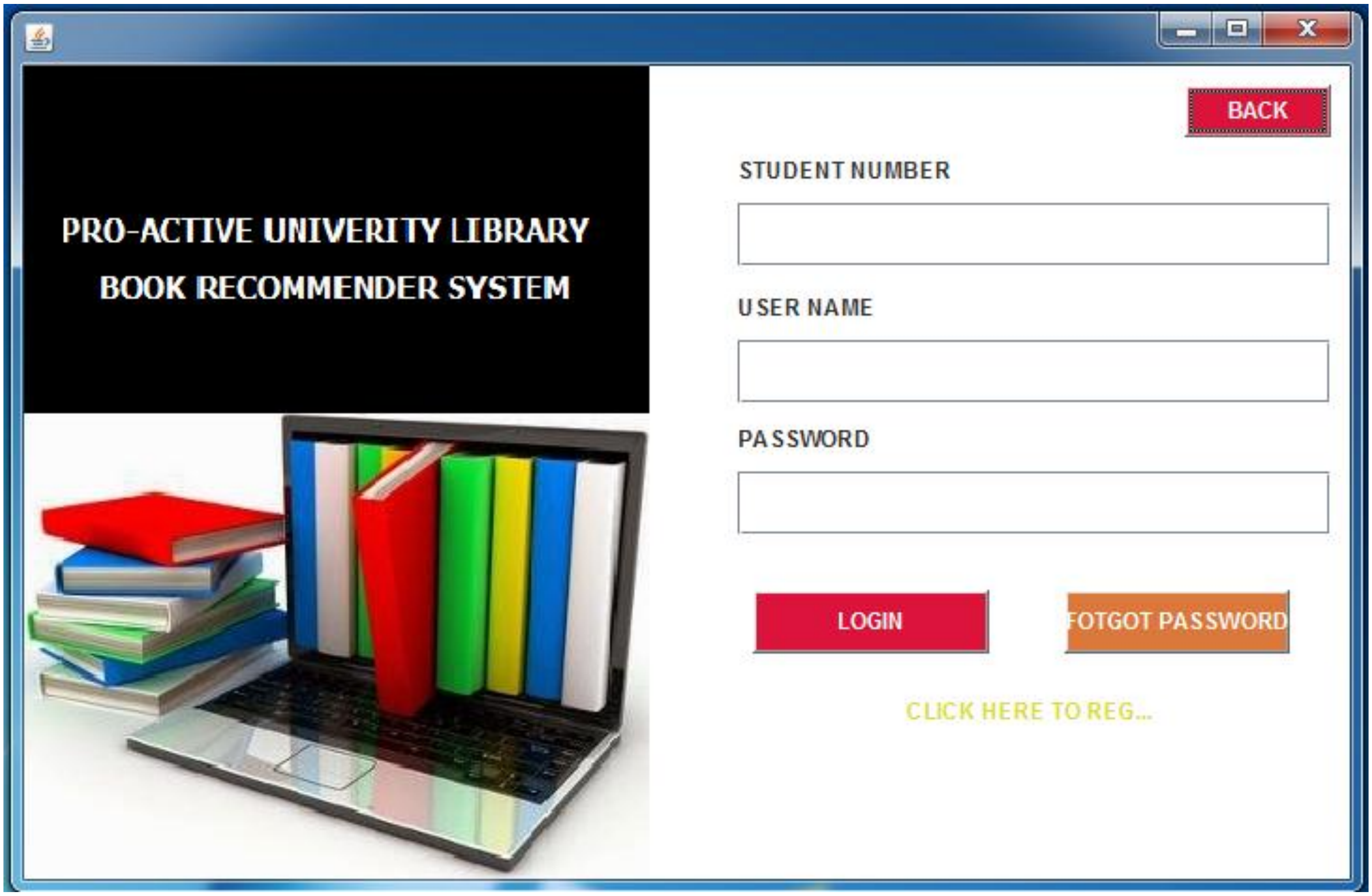


**Figure 4.5** A screenshot of the user login for existing users

The login section is used to identify and authorize existing users. After a first time user is logged in to the system, the user will be requested to insert his first book query. The system then display top 10 similar books to the book requested. Then the user will be requested to rate the top 10 similar books displayed as illustrated in figure 4.6 below:

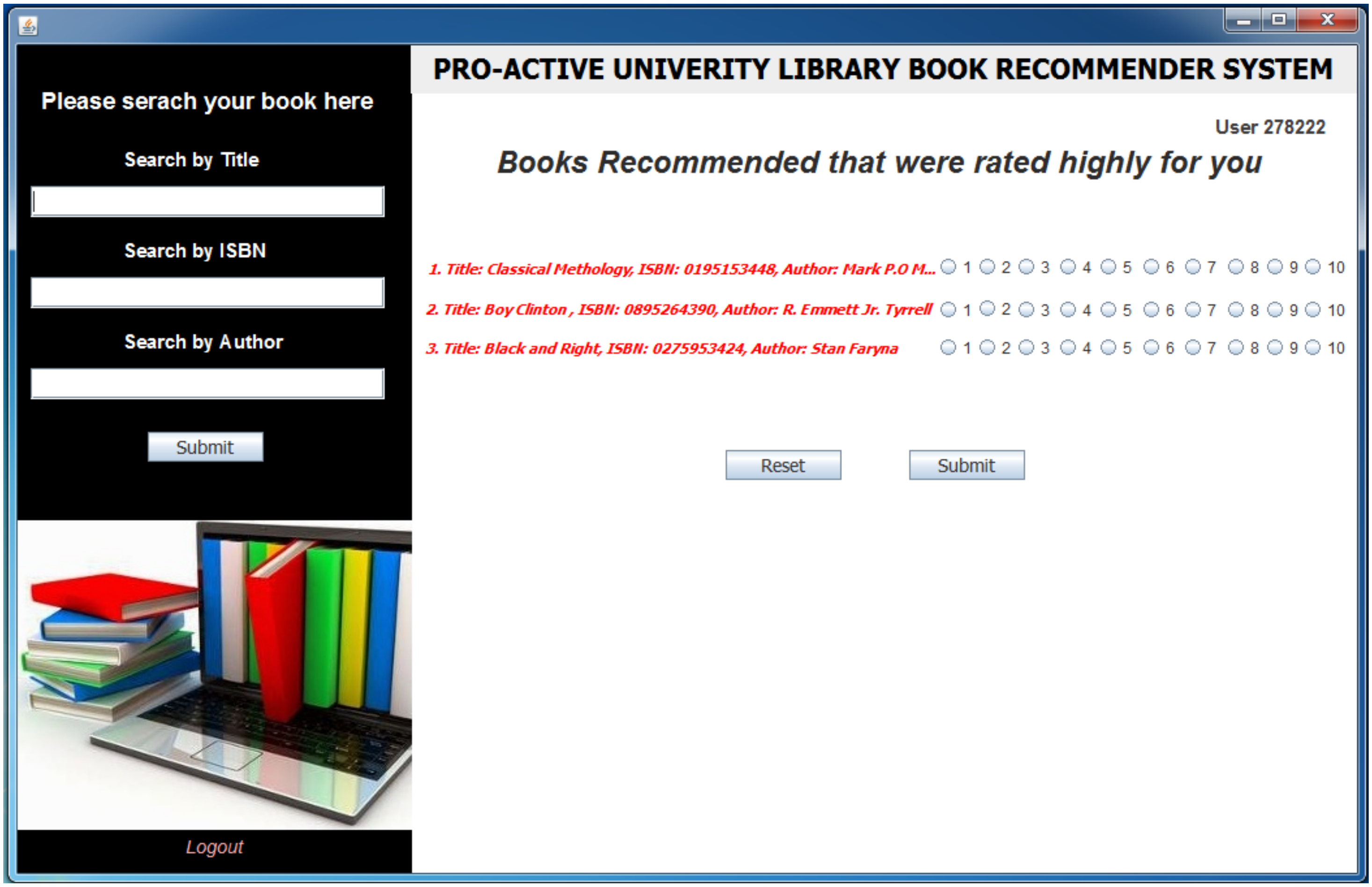


**Figure 4.6** Books proactively recommended that were rated highly for test user 278222

Once the user rates the books and submits, a new window with top 10 book recommendation will pop up. The lists of recommended books for test user 278222 are shown on the figure below.

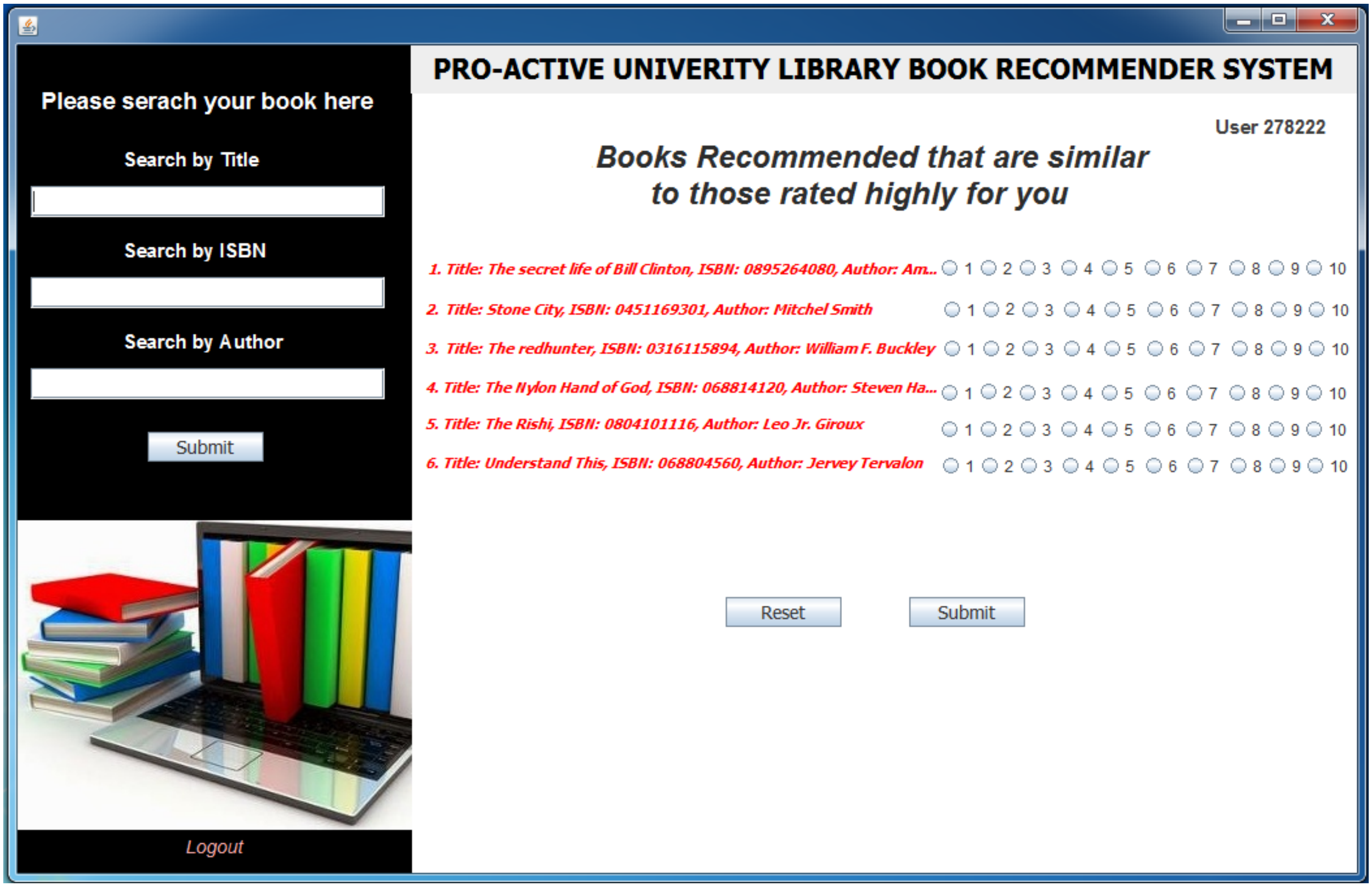


**Figure 4.7** List of books recommended that are similar to those rated highly for test user 278222

The recommender system in Figure 4.7 shows the book recommendation lists based on the user's rating experience. Once a user rates the books, the system will proactively recommend the top ten books every time the user logged in to the system and that implies the prototype was able to recommend the highest 10 predicted books for each user.

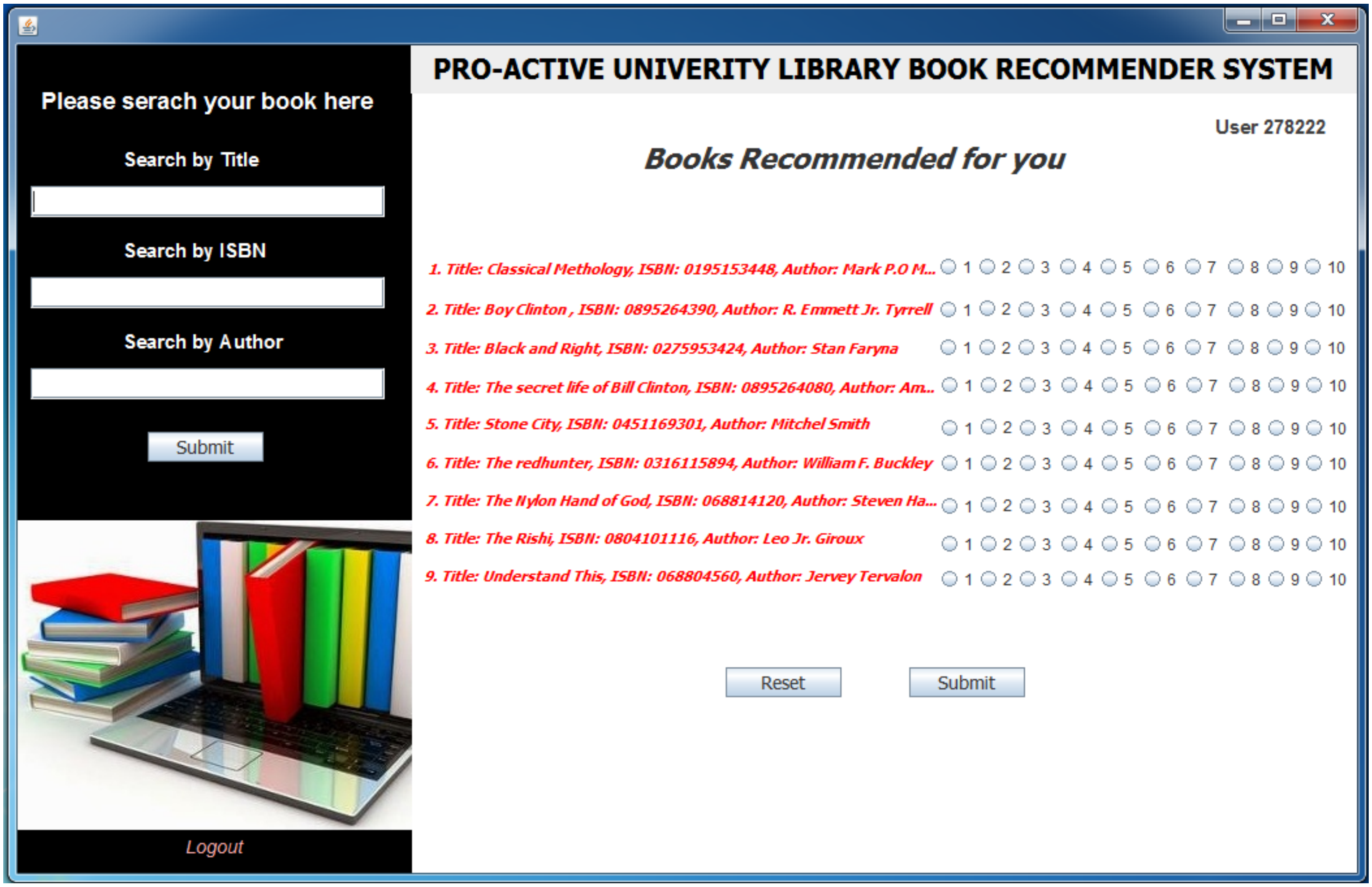


**Figure 4.8** The list of recommended books for test user 278222

To demonstrate the result obtained from experiment a screen shot of top 10 book recommendations for test user 278221 is taken from Python as a sample and shown in figure 4.8.

In [7]:
```python
#Recommending the highest 10 predicted books for user 278221
user_p = books.copy()
user_p = user_p.reset_index()

# getting full dataset
data = Dataset.load_from_df(ratings, reader)
trainset = data.build_full_trainset()
svd.fit(trainset)

user_p['Estimate_Score'] = user_p['ISBN'].apply(lambda x: svd.predict(278221, x).est)

user_p = user_p.drop('ISBN', axis = 1)

user_p = user_p.sort_values('Estimate_Score', ascending=False)

print(user_p.head(10))
```

```
         index                                              Title  \
0            0                                Classical Mythology
139164  139164              Boy Clinton: The Political Biography
139166  139166                                    Black and Right
139170  139170  The Secret Life of Bill Clinton: The Unreporte...
139171  139171                                Stone City: A Novel
139172  139172  The Redhunter : A Novel Based on the Life of S...
139173  139173                             The Nylon Hand of God
139176  139176                                          The Rishi
139181  139181                                    Understand This
139182  139182                                Palladio : A Novel

                       Author YrOfPublication                  Publisher  \
0          Mark P. O. Morford            2002  Oxford University Press
139164    R. Emmett Jr. Tyrrell            1996        Regnery Publishing
139166              Stan Faryna            1997        Praeger Publishers
139170  Ambrose Evans-Pritchard            1997        Regnery Publishing
139171            Mitchell Smith            1990     Simon &amp; Schuster
139172        William F. Buckley            1999              Little, Brown
139173            Steven Hartov            1996             Harpercollins
139176           Leo, Jr. Giroux            1985         Natl Book Network
139181          Jervey Tervalon            1994             Harpercollins
139182              JONATHAN DEE            2002                 Doubleday
```

**Figure 4.9** Top 10 book recommendations for user 278222

The above figure shows the top 10 recommendations given to a test user 278222.

## 4.3 EVALUATION OF RESULTS

Two different types of metrics for evaluating the effectiveness and accuracy of the methods were used. They are MAE and RMSE.

### 4.3.1 MEAN ABSOLUTE ERROR (MAE) AND ROOT MEAN SQUARED ERROR (RMSE)

Mean absolute error and root mean squared error were used to evaluate the accuracy of the results by finding the difference between the actual ratings given by the users and the estimation ratings of the system using the algorithm. MAE measures the average of the absolute deviance between the predicted ratings and the actual ratings given by the users in the system. RMSE measures the average size of the error by calculating and finding the square root of the average of squared differences between prediction given by the prototype and actual rating. RMSE gives a relatively bigger weight to large errors because the errors are squared before they are averaged.

The difference between the rating and the estimation was taken as an error. The more the difference is small the more the system is accurate.

To demonstrate the evaluation process, the rating of test user 22222 for book ID 111111 was taken. After training the data model, the rating for test user 222222 for book ID 111111 was predicted. We evaluated the accuracy of the result by comparing the predicted ratings directly with the actual ratings given by the users. MAE and RMSE were applied with the results shown in figure 4.10.

```
In [9]: # Train the model
        trainset = data.build_full_trainset()
        svd.fit(trainset)

        # Predict the rating for test user 22222 for book id 111111
        UserID = str(22222)
        ISBN = str(111111)
        Rating = 8
        print(svd.predict(UserID, ISBN, Rating))

        user: 22222      item: 111111      r_ui = 8.00   est = 7.10
```

**Figure 4.10** Train model and predict the rating for test user 22222 for book ID 111111

From figure 4.10, the result showed that the estimation rating was 7.10. The difference between the actual rating, which was 8.00, and the estimation value given by the algorithm, which was 7.10, was 0.90. This shows that the designed prototype was close to accurate compared to the previous result arrived at by other researchers.

Tashkandi, Wiese, & Baum (2017) proposed comparative evaluation for recommender systems for book recommendations achieved 1.953229033 RMSE using Pearson similarity method and 1.953229033 RMSE cosine similarity method for the item based collaborative method.

Our prototype showed a better performance over the above researches with 0.84904 MAE and 0.9579 RMSE as figure 4.10 Illustrates.

**4.4 CHAPTER SUMMARY**

In this chapter, experiments were done based on the methods we described in chapter 3. After we got the results and we were able to evaluate them and identified that our method was performing better comparing it with previous work done with the traditional collaborative filtering user a similarity matrix. The results showed that the experiment was successful to recommend high rated books using the hybrid recommendation technique. This specifies that the combination of the collaborative and content-based method can improve the prediction accuracy of the university library book recommender system.

# CHAPTER 5: CONCLUSION AND FUTURE WORK

## 5.1 CONCLUSION

This chapter provides the conclusion, contribution and future work of the research work. It also includes the limitations and possible future works to make them better in the future.

In chapter 1 the primary aim was to proactively recommend library books to students and staff. To achieve this aim, we accomplished different experiments based on the objectives which are mentioned in chapter 1.

The primary objective was fulfilled after we had studied what has been done in the literature in order to recommend suitable and appropriate books to library users. We identified the gaps, what must be done to fill the gaps and to improve previous work done. The importance of the reviews and studies was to understand the problems and issues that occurred in the techniques used. It also helped us to identify open issues, and advantages and disadvantages of the techniques used on previous recommender systems.

By designing and developing a proactive library recommender prototype for library users we fulfilled the second and third objectives mentioned in chapter 1. A hybrid of collaborative and content-based filtering methods was used in order to reduce the drawbacks of both techniques that made the recommendation more accurate. Scientific methodologies have been used in the studies. A standard dataset called book-crossing dataset, which has 278,858 users and 271,379 books, was used. For training data and conducting the experiment, 1,149,780 ratings were used. Cosine similarity algorithm was used to calculate the similarity between users while Pearson correlation was used to determine the prediction ratings of the target users. The prototype also used the vector space model to determine which books are more similar to each other and to the user profile. Each book is stored as a vector of its attributes, which are also vectors in an n-dimensional space, and the angles between the vectors are calculated to determine the similarity between vectors. The system computed the similarity between different library book users and enabled one to predict probable ratings for unrated books by the users that enabled the prototype to give a good recommendation for users.

Finally we evaluated the developed prototype to see if it could effectively recommend university library books for library users or not since the final objective which has been mentioned in chapter 1 was to evaluate the developed prototype. This was achieved by comparing the predicted ratings directly with the actual ratings given by the users. By getting a value of 0.84904 mean absolute error (MAE) and a value of 0.9579 root mean squared error (RMSE), we concluded that the developed prototype was evaluated with a better performance over most previous works done by other researchers.

Finally, this study shows that using the hybrid method/approach to recommend books to university libraries can increase the value of recommending books rather than using only one method.

## 5.2 SUMMARY OF CONTRIBUTIONS

Most of the existing recommender systems recommend books for commercial purposes. Our research showed that similar systems of recommendation of books could be applied to university libraries for helping university library users to discover related books of their own interest easily and timely.

The created prototype is valuable to unravel some of issues of data overload problems in the existing book collection systems. The prototype helps users to urge recommendation of suitable books in a library that contains the titles in which they are interested. It can increase the value of recommending books by increasing the visibility and availability of books in university libraries.

This research also will contribute a lot for other researchers in the recommender system field since not much research has been done before in this area in Africa. The improved results also suggested that it was helpful for researchers to start their research from somewhere not from scratch. This dissertation provided different contributions to knowledge by using the hybrid method, the results of which evaluated successfully. The research proved the positive effects of the hybrid recommender method in university library book recommender system. It also presented the advantages of the recommender system for university library management systems of book collections.

### 5.3 FUTURE WORK AND LIMITATION

Even if the algorithm applied, accuracy was evaluated and yielded better accuracy than most previous ones; there was only a small improvement in terms of recommendation accuracy. Therefore, there is still a necessity to research in this area to improve the accuracy of the recommendation. The algorithm in recommendation systems is domain restricted. Moreover, there were very limited datasets in the area of book collection management systems. Different algorithms can be used by future researchers to get results that are more accurate with the presence of enough training data from available datasets. The research on the book recommender system will remain an interesting research area until 100% accuracy is achieved.

# 6 REFERENCES

Achakulvisut, T., Acuna, D. E., Ruangrong, T., & Kording, K. (2016, July 6). Science Concierge: A fast content-based recommendation system for scientific publications. *PLoS ONE, 11*, 1-11.

Aygün, C., & Yıldız, O. (2016). Development of content based book recommendation system using genetic algorithm. *2016 24th Signal Processing and Communication Application Conference (SIU)* (pp. 1025-1028). Zonguldak: IEEE.

Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. (2019, January). Scientific Paper Recommendation: A Survey. *IEEE Access*, 1-1.

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2004). A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 509–514}). Seattle, Washington: Association for Computing Machinery.

Bhargava, H., Sridhar, S., & Herrick, C. (1999, March). Beyond spreadsheets: tools for building decision support systems. *Computer, 32*(3), 31-39.

Bhure, P., & Adhe, N. (2015). Book recommendation system using opinion mining. *International Journal of Research in Engineering and Technology, 04*(1), 2321-7308.

Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). TasteWeights: a visual interactive hybrid recommender system. *Proceedings of the Sixth ACM Conference on Recommender Systems* (pp. 35–42). Dublin: Association for Computing Machinery.

Braunhofer, M., Kaminskas, M., & Ricci, F. (2013, March). Location-aware music recommendation. *International Journal of Multimedia Information Retrieval, 2*, 31-44.

Burke, R. (2002, November 01). Hybrid recommender systems: Survey and experiments. *User Model User-Adap Interaction, 12*, 331–370.

Cao, Y., & Li, Y. (2007, July). An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Syst. Appl., 33*(1), 230-240.

Chai, T., & Draxler, R. (2014, June). Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*, 1247-1250.

Chen, C.-C., & Chen, A.-P. (2007). Using data mining technology to provide a recommendation service in the digital library. *The Electronic Library, 26*(6), 711-724.

Chen, C.-M., & Yang, Y.-C. (2010). *An Intelligent Mobile Location-Aware Book Recommendation System with Map-Based Guidance That Enhances Problem-Based Learning in Libraries* (Vol. 67). Springer, Berlin, Heidelberg.

Chen, L.-C., Kuo, P.-J., & Liao, I.-E. (2014, March). Ontology-Based Library Recommender System Using MapReduce. *Cluster Computing*, *18*, 113–121.

Choi, K., Yoo, D., Kim, G., & Suh, Y. (2012, July 01). A Hybrid Online-Product Recommendation System: Combining Implicit Rating-Based Collaborative Filtering and Sequential Pattern Analysis. *Electronic Commerce Research and Applications*, *11*, 309–317.

Chun, I.-G., & Hong, I.-S. (2001). The implementation of knowledge-based recommender system for electronic commerce using Java expert system library. *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570* (pp. 1766-1770). Pusan: IEEE.

Cui, B., & Chen, X. (2009). An Online Book Recommendation System Based on Web Service. *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 520-524). Tianjin: IEEE.

Debnath, S., Ganguly, N., & Mitra, P. (2008). Feature Weighting in Content Based Recommendation System Using Social Network Analysis. *Proceedings of the 17th International Conference on World Wide Web* (pp. 1041–1042). Beijing: Association for Computing Machinery.

Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., & Quadrana, M. (2016, January). Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, *5*, 99–113.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011, February). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human–Computer Interaction, 4,* 81-173.

Fu, S., Zhang, Y., & Seinmin, N. (2013, January). On the recommender system for university library. *Proceedings of the International Conference e-Learning 2013*, 215-222.

Gao, F., Xing, C., Du, X., & Wang, S. (2007, February). Personalized service system based on hybrid filtering for digital library. *Tsinghua Science and Technology, 12*, 1-8.

George, T., & Merugu, S. (2005). A Scalable Collaborative Filtering Framework Based on Co-Clustering. *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining* (pp. 625–628). Houston, TX: IEEE.

Geyer-Schulz, A., Hahsler, M., Neumann, A., & Thede, A. (2003, December). Behavior-based recommender systems as value-added services for scientific libraries. *Behavior-Based Recommender Systems as Value-Added Services for Scientific Libraries, 22*(4), 433-454.

Geyer-Schulz, A., Neumann, A., & Thede, A. (2013, December). An Architecture for Behavior-Based Library Recommender Systems. *Information Technology and Libraries, 22*, 165-174.

Ghadling, S., Belavad, K., Bhegade, S., Ghojage, P., & Supriya, K. (2015, November 28). Digital Library: Using Hybrid Book Recommendation Engine. *International Journal Of Engineering And Computer Science*.

Ghazanfar, M. A., & Prugel-Bennett, A. (2010). A scalable, accurate hybrid recommender system. *2010 3rd International Conference on Knowledge Discovery and Data Mining (WKDD 2010)* (pp. 94-98). Phuket: IEEE Computer Society.

Gunawardana, A., & Meek, C. (2009). A unified approach to building hybrid recommender systems. *Proceedings of the Third ACM Conference on Recommender Systems* (pp. 117–124). New York: Association for Computing Machinery.

Hahn, J. (2011, November 15 ). Location-based recommendation services in library book stacks. *Reference Services Review, 39*(4), 654-674.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004, January). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5–53.

Heylighen, F., & Bollen, J. (2002). Hebbian algorithms for a digital library recommendation system. *Proceedings. International Conference on Parallel Processing Workshop* (pp. 439-446). Vancouver, BC: IEEE.

Hu, L., Sun, A., & Liu, Y. (2014, July 03). Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 345–354.

Huang, Z., Chung, W., Ong, T.-H., & Chen, H. (2002). A Graph-Based Recommender System for Digital Library. *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 65–73). Portland: Association for Computing Machinery.

Huang, Z., Li, X., & Chen, H. (2005). Link prediction approach to collaborative filtering. *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)* (pp. 141-142). Denver, CO: IEEE.

Isinkayea, F. O., Folajimib, Y. O., & Ojokohc, B. A. (2015, November). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal, 16*(3), 261-273.

Jia, F., & Shi, Y. (2013). *Library Management System Based on Recommendation System* (Vol. 392). Springer, Berlin, Heidelberg.

Jomsri, P. (2014). Book recommendation system for digital library based on user profiles by using association rule. *Book recommendation system for digital library based on user profiles by using association rule* (pp. 130-134). Luton: IEEE.

Kamble, V. B., & Deshmukh, S. (2017). Comparision between Accuracy and MSE, RMSE by using. *Oriental Journal of Computer Science and Technology, 10*, 773-779.

Kim, Sang-Woon, & Gil, J.-M. (2019, August). Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences, 9*.

Kompan, M., & Bieliková, M. (2010). Content-Based News Recommendation. *E-Commerce and Web Technologies, 11th International Conference, EC-Web 2010* (pp. 61-72). Bilbao: Springer, Berlin, Heidelberg.

Kumar, N. P., & Fan, Z. (2015, December 31). Hybrid User-Item Based Collaborative Filtering. *Procedia Computer Science*, 1453-1461.

Kumar, N., & Fan, Z. (2015). Hybrid User-Item Based Collaborative Filtering. *Procedia Computer Science, 60*, 1453-1461.

Kurmashov, N., Latuta, K., & Nussipbekov, A. (2015). Online book recommendation system. *2015 Twelve International Conference on Electronics Computer and Computation (ICECCO)* (pp. 1-4). Almaty: ICCC.

Kuroiwa, T., & Bhalla, S. (2007). Dynamic Personalization for Book Recommendation System Using Web Services and Virtual Library Enhancements. *7th IEEE International Conference on Computer and Information Technology (CIT 2007)* (pp. 212-217). Aizu-Wakamatsu, Fukushima: IEEE Computer Society.

Lee, D., Park, J., Shim, J., & Lee, S.-g. (2010). An Efficient Similarity Join Algorithm with Cosine Similarity Predicate. *Database and Expert Systems Applications, 21th International Conference,. 6262*, pp. 422-436. Bilbao: Springer, Berlin, Heidelberg.

Li, B., & Han, L. (2013). Distance Weighted Cosine Similarity Measure for Text Classification. *International Conference on Intelligent Data Engineering and Automated Learning. 8206*, pp. 611-618. Springer, Berlin, Heidelberg.

Li, Q., & Kim, B. M. (2003). An Approach for Combining Content-based and Collaborative Filters. *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages. 11*, pp. 17-24. Sapporo: Association for Computational Linguistics.

Liao, I.-E., Hsu, W.-C., Cheng, M.-S., & Chen, L.-P. (2010, June 08). A library recommender system based on a personal ontology model and collaborative filtering technique for English collections. *The Electronic Library, 28*(3), 386-400.

Liao, I.-E., Hsu, W.-C., Cheng, M.-S., & Chen, L.-P. (2010, June). A library recommender system based on a personal ontology model and collaborative filtering technique for English collections. *The Electronic Library, 28*(3), 386-400.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76-80.

Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014, January). A new user similarity model to improve the accuracy of collaborative filtering. *Know.-Based Syst., 56*, 156-166.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Systems & e-Service Intelligence Lab, Centre for Quantum Computation & Intelligent Systems, 74*, 12-32.

Lumauag, R. G., Sison, A. M., & Medina, R. P. (2019). An Enhanced Recommendation Algorithm Based on Modified User-Based Collaborative Filtering. *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)* (pp. 198-202). Singapore: IEEE.

Manikrao, U. S., & Prabhakar, T. (2005). Dynamic selection of Web services with recommendation system. *International Conference on Next Generation Web Services Practices (NWeSP'05). 117-121*, pp. 117-121. Seoul: IEEE Computer Society.

Martin, F. J., Shur, J., & Torrens, M. (2012). *Patent No. 12685639.* US.

Martinez, A. B., Arias, J. J., Vilas, A. F., Duque, J. G., & Nores, M. L. (2009, March). What's on TV tonight? An efficient and effective personalized recommender system of TV programs. *Consumer Electronics, IEEE Transactions on, 55*, 286-294.

Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)* (pp. 47-52). Ernakulam: IEEE.

melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Eighteenth National Conference on Artificial Intelligence* (pp. 187-192). Edmonton, Alberta: American Association for Artificial Intelligence.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Eighteenth National Conference on Artificial Intelligence* (pp. 187-192). Edmonton, Alberta: American Association for Artificial Intelligence.

Melville, P., Raymond , J. M., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, (pp. 187-192). Edmonton, Canada.

Milicevic, A., Vesin, B., Ivanovic, M., & Budimac, Z. (2011, April). E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education, 56*, 885-899.

Mohd, A., Hameed, M. A., al jadaan, O., & Sirandas, R. (2012 , May 05). Collaborative Filtering Based Recommendation. *International Journal on Computer Science and Engineering (IJCSE), 4*.

Mooney, R. J., & Roy, L. (2000). Content-Based Book Recommending Using Learning for Text Categorization. *Proceedings of the Fifth ACM Conference on Digital Libraries* (pp. 195–204). San Antonio, Texas: Association for Computing Machinery.

Nguyen, & Eric. (2014). *Text Mining and Network Analysis of Digital Libraries in R.* Montreal, Quebec: Academic Press.

O'Mahony, P, M., & Smyth, B. (2007). A recommender system for on-line course enrolment: An initial study. *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 133–136). Minneapolis, MN: Association for Computing Machinery.

Oord, A. v., Sander, D., & Benjamin, S. (2013). Deep content-based music recommendation. *Proceedings of the 26th International Conference on Neural Information Processing Systems. 2*, pp. 2643–2651. Lake Tahoe: Curran Associates Inc.

Park, M.-H., Hong, J.-H., & Cho, S.-B. (2007). Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. *4611*, pp. 1130-1139. Verlag Berlin Heidelberg: Springer.

Park, M.-H., Hong, J.-H., & Cho, S.-B. (2007). Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing* (pp. 1130–1139). Hong Kong: Springer-Verlag.

Parvatikar, S., & Bharti, J. (2015). Online book recommendation system by using collaborative filtering and association mining. *Computational Intelligence and Computing Research (ICCIC) IEEE International Conference* (pp. 1-4). Madurai, India: IEEE.

Parvatikar, S., & Joshi, B. (2015). Online book recommendation system by using collaborative filtering and association mining. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC* (pp. 1-4). Madurai: IEEE.

Parvatikar, S., & Joshi, B. (2015). Online book recommendation system by using collaborative filtering and association mining. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-4). Madurai: IEEE.

Pazzani, M. J., & Billsus, D. (2007). *Content-based recommendation systems* (Vol. 4321). Springer, Berlin, Heidelberg.

Pazzani, M. J., & Billsus, D. (2007). *Content-Based Recommendation Systems* (Vol. 4321). German: Spring,Berlin, Heidelberg.

Perugin, S., Gonçalves, M. A., & Fox, E. A. (2004). Recommender Systems Research: A Connection-Centric Survey. *Journal of Intelligent Information Systems*, 107–143.

Porcel, C., Morales-del-Castillo, J., Cobo, M., Ruíz, A., & Herrera-Viedma, E. (2010). An improved recommender system to avoid the persistent information overload in a university digital library. *Control and cybernetics (CONTROL CYBERN), 39*, 901-923.

Porcel, C., Moreno, J., & Herrera-Viedma, E. (2009, December). A multi-disciplinar recommender system to advice research resources in University Digital Libraries. *Expert Systems with Applications, 36*, 12520-12528.

Rajpurkar, S., & Bhatt, D. (2015, April). Book recommendation system. *International Journal for Innovative Research in Science & Technology, 1*(11), 314-316.

Rana, C., & Jain, S. (2015, February). Building a book recommender system using time based content filtering. *WSEAS TRANSACTIONS on COMPUTERS*, *11*(2), 27-33.

Rana, C., & Jain, S. K. (2012, February). Building a Book Recommender system using time based content filtering. *WSEAS TRANSACTIONS on COMPUTERS* , *11*(2), 27-33.

Salam, P. Z., & Safir, N. (2016). *Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems.* School of Computer Science and Communication (CSC), STOCKHOLM, SWEDEN.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web. 1*, pp. 285-295. Hong Kong: Association for Computing Machinery.

Sase, A., Varun, K., Rathod, S., & Patil, P. (2015, February). A Proposed Book Recommender System. *IJARCCE, 4*(2), 481-483.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative Filtering Recommender Systems* (Vol. 4321). Verlag Berlin Heidelberg: Springer.

Shinde, S., & Kulkarni, U. (2012, January). Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications: An International Journal, 39*, 1381-1387.

Singhal, A., Buckley, C., & Mitra, M. (2017, June). Pivoted Document Length Normalization. *ACM SIGIR Forum, 51*, 176–184.

Smeaton, A., & Callan, J. (2005, August 01). Personalisation and recommender systems in digital libraries. *International Journal on Digital Libraries, 57*, 299-308.

Suhasini, P., & Joshi, B. (2015). Online book recommendation system by using collaborative filtering and association mining. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-4). Madurai: IEEE.

Suhasini, P., Joshi, B., & tadesse. (2015). Online book recommendation system by using collaborative filtering and association mining. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-4). Madurai: IEEE.

Taheri, S. M., Mahyar, H., Firouzi, M., Elahe, G., Grosu, R., & Movaghar, A. (2017). Extracting Implicit Social Relation for Social Recommendation Techniques in User Rating Prediction. *WWW '17 Companion: Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 1343–1351). Perth, Australia: ACM Digital Library.

Tashkandi, A., Wiese, L., & Baum, M. (2017). Comparative Evaluation for Recommender Systems for. *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband* (pp. 291-300). Bonn: Gesellschaft für Informatik e.V.

Tata, S., & Patel, J. M. (2007, June). Estimating the Selectivity of Tf-Idf Based Cosine Similarity Predicates. *Sigmod Record, 36*, 7–12.

Tejeda-Lorente, A., Porcel, C., Peis, E., Sanz, R., & Herrera-Viedma, E. (2014, March). A quality based recommender system to disseminate information in a university digital library. *Information Sciences, 261*, 52–69.

Tewari, A. S., & Priyanka, K. (2014). Book recommendation system based on collaborative filtering and association rule mining for college students. *2014 International Conference on Contemporary Computing and Informatics (IC3I)* (pp. 135-138). Mysore, India: IEEE.

Torres, R., McNee, S. M., Abel, M., Konstan, J. A., & Riedl, J. (2004). Enhancing digital libraries with TechLens+. *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 228–236). Tuscon, AZ: Association for Computing Machinery.

van Meteren, R., & van Someren, M. (2000, June). Using Content-Based Filtering for Recommendation. *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, *30*, pp. 47-56.

Vaz, P. C., Martins de Matos, D., Martins, B., & Calado, P. (2012). Improving an Hybrid Literary Book Recommendation System through Author Ranking. *Proceedings of the 12th ACM/IEEE-CS Joint*

*Conference on Digital Libraries* (pp. 387–388). Washington, DC: Association for Computing Machinery.

Walter, F., Battiston, S., & Schweitzer, F. (2008, February). A Model of a Trust-based Recommendation System on a Social Network. *Autonomous Agents and Multi-Agent Systems*, *16*, 57-74.

Wei, S., Ye, N., Zhang, S., & Huang, X. (2012). Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure. *International Conference on Computer Science and Service System* (pp. 2038-2041). Nanjing: IEEE Computer Society.

Wu, D., Zhang, G., & Lu, J. (2013). A fuzzy tree similarity based recommendation approach for telecom products. *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)* (pp. 813-818). Edmonton, AB: IEEE.

Ze, W., & Dengwen, Z. (2016). Optimization collaborative filtering recommendation algorithm based on ratings consistent. *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 1055-1058). Beijing: IEEE.

Zeng, J., Li, F., Liu, H., Wen, J., & Hirokawa, S. (2016). A Restaurant Recommender System Based on User Preference and Location in Mobile Environment. *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). 1*, pp. 50-60. Kumamoto: IEEE.

Zhang, Y., & Koren, J. (2007). Efficient Bayesian Hierarchical User Modeling for Recommendation System. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 47-54). Amsterdam: Association for Computing Machinery.

Zheng, Z., Ma, H., & Lyu, M. R. (2009). WSRec: A Collaborative Filtering Based Web Service Recommender System. *2009 IEEE International Conference on Web Services* (pp. 437-444). Los Angeles, CA: IEEE.

Zuva, T., Ojo, S. O., Ngwira, S., & Zuva, K. (2012). A survey of recommender systems techniques, challenges and evaluation metrics. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), 382-386.