

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325792825>

An Improved Online Book Recommender System using Collaborative Filtering Algorithm

Article in International Journal of Computer Applications · June 2018

DOI: 10.5120/ijca2018917193

CITATIONS

15

READS

8,134

3 authors, including:



[Emmanuel Okon](#)

University of Lagos

8 PUBLICATIONS 36 CITATIONS

SEE PROFILE

An Improved Online Book Recommender System using Collaborative Filtering Algorithm

E. Uko Okon

Department of Computer Science
University of Port Harcourt
Nigeria

B. O. Eke

Department of Computer Science
University of Port Harcourt
Nigeria

P. O. Asagba

Department of Computer Science
University of Port Harcourt
Nigeria

ABSTRACT

In e-commerce today, contents available for users to explore are overwhelming because an average ecommerce website is about seventy per cent (70%) more than a physical store in total number of users and items. Hence, the need to filter, prioritize and efficiently deliver relevant information using recommender systems. We will design and develop a recommendation model that uses object-oriented analysis and design methodology (OOADM), improved collaborative filtering algorithm and an efficient quick sort algorithm to solve these problems. This will be achieved by implementing the stated model with python model-view-controller (MVC) framework known as Django Framework. This improved system is implemented using a real-time, cloud-hosted NOSQL database called FireBase which guarantees scalability. From the results, the speed and scalability of book recommendation was improved with a performance record obtained within the range of ninety (90) to ninety-five (95) per cent using the root mean square error (RMSE) of several recommendations obtained from the system.

General Terms

Books, improved, system, recommendation, algorithm, online bookshop, online.

Keywords

Recommender system, collaborative filtering, recommender algorithms, collaborative filtering algorithm, machine learning algorithm, NoSQL, Firebase, scalability.

1. INTRODUCTION

1.1 Background

A recommender system is a type of information filtering system that predict the rating or preference that a user would give to an item. Recommender systems are sometime referred to as recommendation system. Recommendation system describes web applications that predicted response to options. According to [10], Recommender Systems are targeted to individuals who do not have enough personal experience to evaluate the potentially overwhelming alternatives that a web site for instance may offer.

Some companies such as Amazon, Facebook and google are already are already utilizing the power of recommender systems in building their new and existing systems. Recommender systems often provide personalized recommendation of items (to users) using a ranked list of predicted items.

According to [5] individuals rely on recommendations provided by others in making routine daily decisions. As examples, it is common to rely on peer recommendations when selecting a book to read; employers use recommendation letters in their recruiting decisions; and when

selecting a movie to watch, individuals do read and rely on the movie reviews that a film critic has written.

In attempt to replicate these behaviors, the first Recommender System used algorithms to create a prototype of recommendations produced by a community of users to deliver recommendations to an active user. These recommendations were designed for every item that similar users had liked. This approach is popularly known as collaborative-filtering and its aim is to ensure that if the active user agreed in the past with some users, then the other recommendations coming from these similar users should be relevant as well and of interest to the active user.

In this paper, we introduce a novel architecture for the implementation of recommender system algorithm which can be used to improve the recommendation of books to users.

1.2 Recent Application of Machine Learning and Recommender Systems

Machine learning constitutes one of the most recently researched and applied field in artificial intelligence and its capabilities has been applied in diverse areas. The following list will better explain the recently applied area of machine learning and recommender systems today.

1. Wildlife Preservation: Machine learning has been applied in marine to help identify whales in the oceans based on audio recording so that ships can avoid hitting them.

2. Anomaly Detection: Machine Learning is applied to flag any malpractice in very high volume high frequency data transactions / communications. Machine learning systems can today detect a possible insider trading in a stock market; also machine learning can flag a malicious customer transaction as a fraudulent transaction in high volume business doing market place websites.

3. Classification: Machine learning helps to group online questions asked by visitors in to different categories that can be easily accessible by new users. The same classification (or topic modeling) algorithms are behind how news articles from thousands of sources gets neatly segregated under topics in Google News or any major news aggregating portals.

4. Financial Trading: Today, people are attempting to predict stock market capitals but machine learning algorithms are presenting better attempts. Stock market prediction systems uses probabilities but a trade with a low probability at a high volume can turn generate much gain for an organization. And this machine out-performs humans in big data consumption and speed of execution.

5. Search Engines: Search engines are perhaps the most widely known application of recommender systems and machine learning. Each time search query is executed on

search engines like Google, the application observes how the user responds to the results. If the user clicked the results at the top and remains on the web page, it assumes that the user was satisfied with the information searched for. Inversely, if the user clicks to the next page of the results or inputs a new search string without clicking on any of the displayed results, it can be concluded that the required results was not presented to the user and the program automatically learns from the current mistake to increment on knowledge and deliver a better result subsequently.

6. Natural Language Processing (NLP): Natural Language processing is applied in many applications across different

disciplines. Machine learning algorithms with natural language replicate a customer service agent and direct customers to the needed information faster

7. Smart Cars: Smart cars are developed today by automobile companies to utilize Internet of Things (IoT) by learning about the owner and the environment.

8. Computer Vision/ Personal Security: Machine learning could help security staff in public places such as airport/stadia detect defaulters faster. Machine learning can help eliminate false alarms and spot things human might not be able to see in security screenings at airports, stadiums, concerts, and other venues.

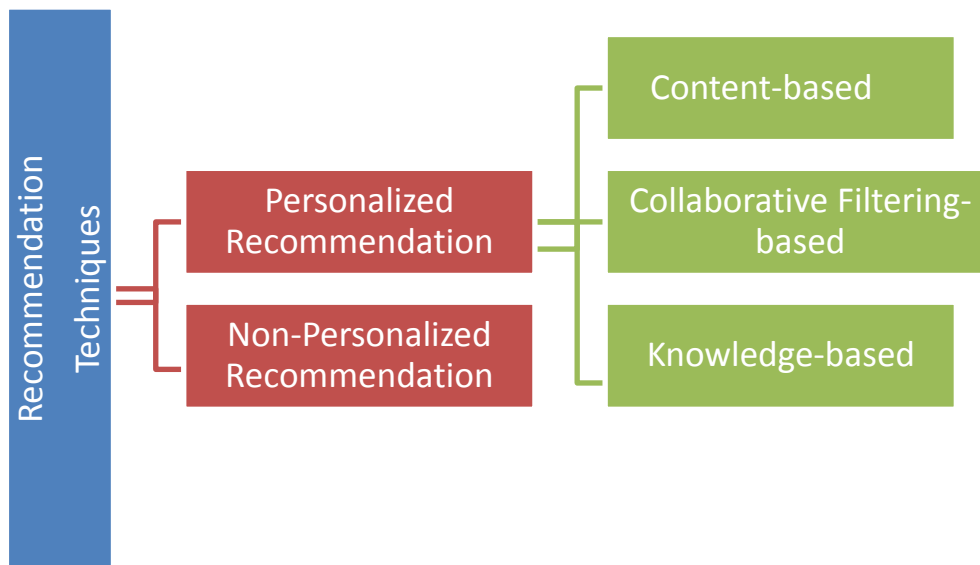


Fig. 1 Classification of recommendation techniques [1]

1.3 Definition of Terms

Recommender: a recommender is a word used to portray someone who puts forward (something or someone) as being suitable for a particular purpose or role [12].

Machine Learning: This is a field in artificial intelligence (AI) focuses on calculations (algorithms) that enable computer systems to learn. This means that in machine learning, a dataset is fed into algorithms and the algorithms interprets information about the data properties and that data enables it to make forecasts about other information that it may find later on.

Algorithm: An algorithm is a self-contained succession of activities to be performed with a specific end goal to tackle a particular problem. Algorithm can perform computations, information handling and mechanized thinking errands.

Collaborative Filtering: Collaborative filtering (CF) is a technique applied mainly in recommendation systems to make automatic predictions about the interests of a client by gathering inclinations or taste data from numerous clients (collaboration). The underlying assumption of the collaborative filtering approach is that on the off chance that a man A has an indistinguishable assessment from a man B on an issue, A will probably have B's conclusion on an unexpected issue in comparison to that of a haphazardly picked individual. For example, a collaborative filtering recommendation system for television tastes could influence forecasts about which television show a client should like

given a halfway rundown of that client's tastes (likes or dislikes). In the more broad sense, collaborative filtering is the procedure of filtering for data or patterns by utilizing methods including coordinated effort among different agents, data sources and viewpoints, etc.

Online: To be considered online, one of the next statements may apply to a system: it is under the direct control of another device; it is under the direct control of the framework with which it is associated; or it is accessible for immediate use on demand by the system without human intervention.

System: This is a set of interacting or interdependent component parts forming a complex or intricate whole. Every system is delineated by its spatial and temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed in its functioning. All parts of a system works together to achieve a common goal.

Book: A book is a set of written, printed, illustrated, or blank sheets, made of paper, parchment, or other materials, fastened together to hinge at one side, with text and/or images printed in ink.

Bookshop: A bookshop is a location where books are stocked for the purpose of being rented or purchased.

Online Bookshop: An online bookshop is an internet website that runs an e-commerce service where online users can access information about books, buy or rent books whenever their timing is ideal.

2. REVIEW AND RELATED WORK

2.1 Review

Recommender Systems (RSs) are tools and techniques which uses software to provide suggestions of useful items to a user [6]. The suggestions identify with different decision-making processes, such as what things to purchase, what music to tune in to, what online news to peruse, or even which book to lease/purchase, etc. The recommender systems are concerned with identifying the learning about the comparable client or the occasion and infer the favorable predictions or suggestions using the context. It is the condition of “individualized” and “intriguing and helpful” that distinguishes the recommender system from search engines [1].

The term “Item” is generally used to represent what the system suggests to users. A Recommender System typically canters around a specific kind of thing (for example, CDs, or news) and as needs be its outline, its graphical UI, and the core recommendation technique applied to obtain the recommendations are altogether modified to give valuable and

successful recommendations to that specific sort of item [2].

Recommender systems assist e-commerce sites increase their deals, recommender systems intermittently creates a rundown of recommendations in three ways which is enlisted as; through collaborative, content based, and knowledge based filtering [1]. A common practice in recommender systems research is the hybridization of recommendation techniques to accomplish top execution or rebuild a current model. These systems can be utilized independently or consolidated together in various ways. Figure 1 explains the different classes of techniques commonly used in building recommendation (or recommender) systems.

2.2 Related Work

[8] proposed a general framework for content-boosted collaborative filtering. This work improved on recommendation by boosting collaborative filtering algorithms with contents. Hence with their results, it was clear that a naïve hybrid method tends to perform better than a pure content-based or collaborative filtering algorithm.

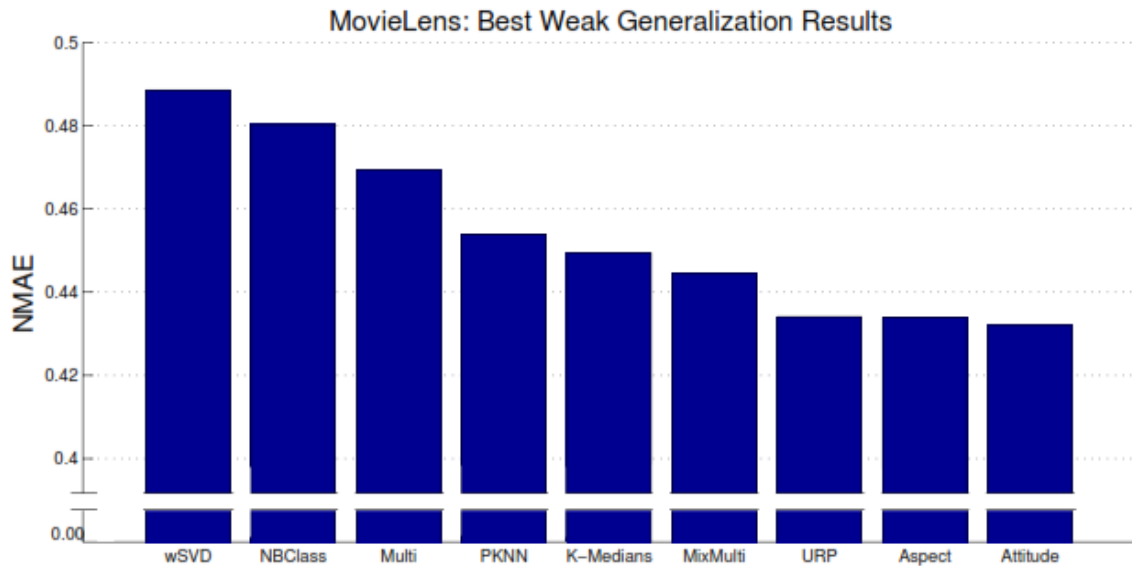


Fig. 2 Best weak Generalization results from nine prediction models [7].

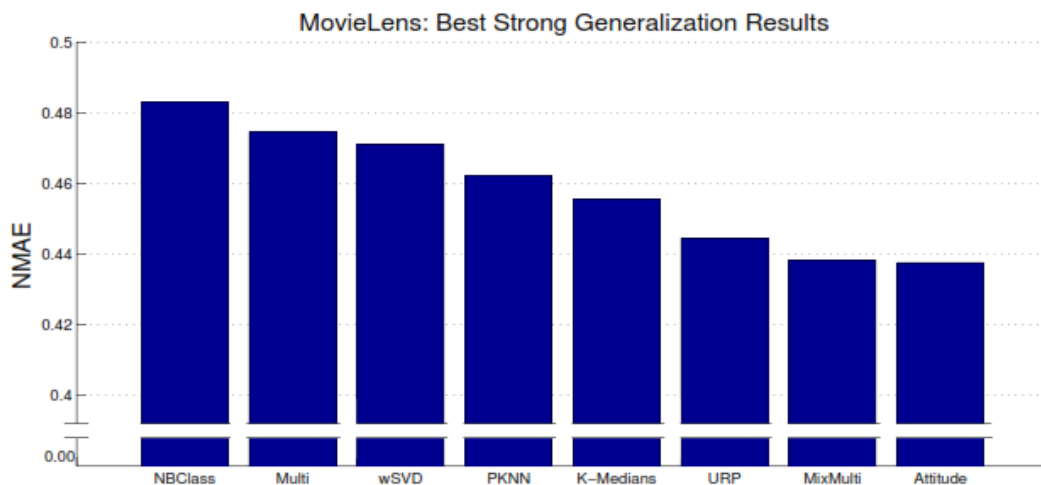


Fig. 3 Best strong Generalization results from nine prediction models [7].

[13] carried out a survey of collaborative filtering techniques by first identifying the collaborative filtering tasks and the challenges facing them such as data sparsity, scalability, gray sheep, shilling attacks, privacy protections etc. and possible solutions to these challenges.

The different collaborative filtering techniques such as memory-based, model-based and hybrid collaborative filtering algorithms were represented and analyzed to determine their predictive performance and their ability to address the previously stated challenges. [5] proposed Amazon.com recommendations: an item-to-item collaborative filtering. This approaches solved recommendation problems with the use of an algorithm different from the traditional collaborative filtering algorithm, cluster models algorithm and search-based algorithm methods. The item-to-item collaborative filtering algorithm applied to Amazon's online shop computations was able to scale independently of the number of customer's and items in the product catalog.

[14] proposed a user-based collaborative filtering algorithm on Hadoop which involved the implementation of user-based collaborative filtering algorithms on a cloud-based platform (case study of Hadoop). The work aimed at solving the scalability problems of collaborative filtering and was able to establish the fact that a simple method that partitions users into groups according to two basic principles namely the tidy arrangement of mapper number to overcome the initiation of mapper and the partitioning of tasks such that all processors finish task at the same time can actually result in a linear speedup of performance.

[3] proposed RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized web service recommendations. The above study considered the difference between web service recommendations and product recommendations used in e-commerce websites. RegionKNN was designed and presented for large scale web services recommendation. RegionKNN used an approach that employs the characteristics of QoS by building efficient region model that quickly generates web service recommendations using modified memory-based collaborative filtering algorithm. From the result of the study, experiments show the high scalability of the RegionKNN.

[11] analyzed the different item-based recommendation algorithm like the item similarity computations: cosine-based similarity, correlation-based similarity, adjusted cosine similarity and prediction computations: weighted sum, regression. Finally, results were experimentally obtained and compared with the k-NN approach. The experiment was able to show that item-based algorithms provide a more efficient performance and better quality recommendations than its user-based counterparts.

[4] presented a comprehensive survey of neighborhood-based methods for item recommendation problem. The major benefits of neighbourhood-based recommender systems and their key characteristics were clearly described. Also, the document addressed the important decisions that must be taken in the course of implementing a recommender system using a neighborhood-based method. The document also gives practical guide on how to carry out such process. Finally, the challenges of sparsity and limited coverage, often observed in large commercial recommender systems were discussed, and a few solutions were provided to overcome these problems.

[7] conducted a study on the different machine learning approaches to collaborative filtering. In the study, the different K-NN classification and regression methods were applied to derive the class of neighborhoods from a recommenders system. A new method for prediction was introduced; this method was able to learn from a set of Naïve Bayes classifiers. The study also illustrated the application of K-medians clustering for prediction rating. Furthermore, dimensionality reduction techniques such as the weighted singular value decomposition, the principal component analysis and the probabilistic principal component analysis were applied in rating predictions. A couple of density estimation methods in probabilistic models such as multinomial model, aspect model and user rating profile model were described and a new family of models known as Attitude model family was introduced. A total of nine (9) rating prediction models were implemented and compared as shown in Figure 2 and 3.

3. MATERIALS AND METHODS

3.1 Architecture of the Existing System

This describes the proposed system, explaining how modules and components integrate and communicate to bring about the working application of the proposed system. The systems design is developed to satisfy the requirement of modern collaborative filtering recommendation system architecture including computational structures and model training algorithms. The system design will also capture the major functional building blocks needed to understand the process of building an online book recommender software system. The architectural design of the proposed system is illustrated in figure 4.

The architecture of the existing system as shown in Fig. 4 comprises the following components;

1. **The User:** The user represents the individual or client that utilizes the interface of the web application at any time and location globally. Typically, in the scenario of the massive open online course (MOOC) system [9], the user is the individual that accesses and explores the online courses. Users

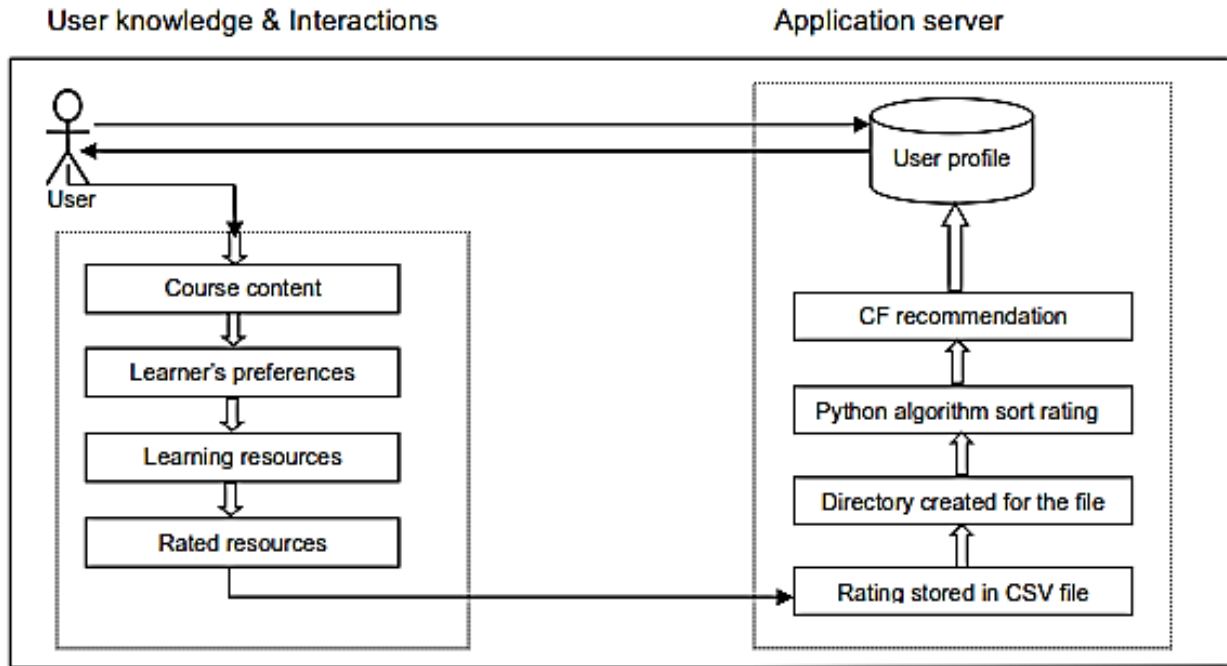


Fig. 4. Architecture of the Existing System [9].

can rates previously taken courses. Here, the term user may refer to a learner in the existing system.

2. Course Content: In the existing system, the course content is the target and many course contents (resources) are available for user access. It forms the collection of the different contents of each online course.

3. Learner's preference: The learner's preference refers to the users preferred choice of online resources. Different users/learners have different choice of online courses and a collection of these choices contribute to the effective recommendation of courses for other users.

4. Learning Resources: Learning resources in this context refers to the different forms of learning materials available for user of the chosen course content. These learning resources can be available in different forms, for instance, a video tutorial, audio tutorial, slide presentation, images, etc.

5. Rated Resources: A rated resource refers to the learning resources that has been rated over 5 by the learner (user) of the online course.

6. Rating stored in a (.csv) file: A collection of different rated resources and user list forms a comma separated value file and is used to perform recommendation of items at the backend.

7. Directory Created for the file: The directory is a path or location that provides an absolute or relative link to the rating file.

8. Python algorithm for sorting rating: Python algorithm in this context involves the step by step implementation of the sorting of ratings with the use of python programming language.

9. A Collaborative Filtering (CF) Recommendation: This involves the process of recommending learning resources to users using previous ratings from other learners. The type of CF algorithm used by the existing system is the user-based Collaborative Filtering (CF) algorithm.

10. User Profile: The user profile describes a database containing user personal details, implicit and explicit ratings and list of recommendations provided based on user's similarity with other users.

Thus, the existing system uses Scrum - a leading agile software development methodology and applies a user-based collaborative filtering (CF) algorithm to compute and perform recommendations.

3.2 Architecture of the Proposed System

This describes the proposed system, explaining how modules and components integrate and communicate to bring about the working application of the proposed system. The systems design is developed to satisfy the requirement of modern collaborative filtering recommendation system architecture including computational structures and model training algorithms. The system design will also capture the major functional building blocks needed to understand the process of building an online book recommender software system. The architectural design of the proposed system is illustrated in figure 5.

The proposed system architecture is described in Fig. 5:

The User: The user represents the individual or client that utilizes the interface of the web application at any time and location

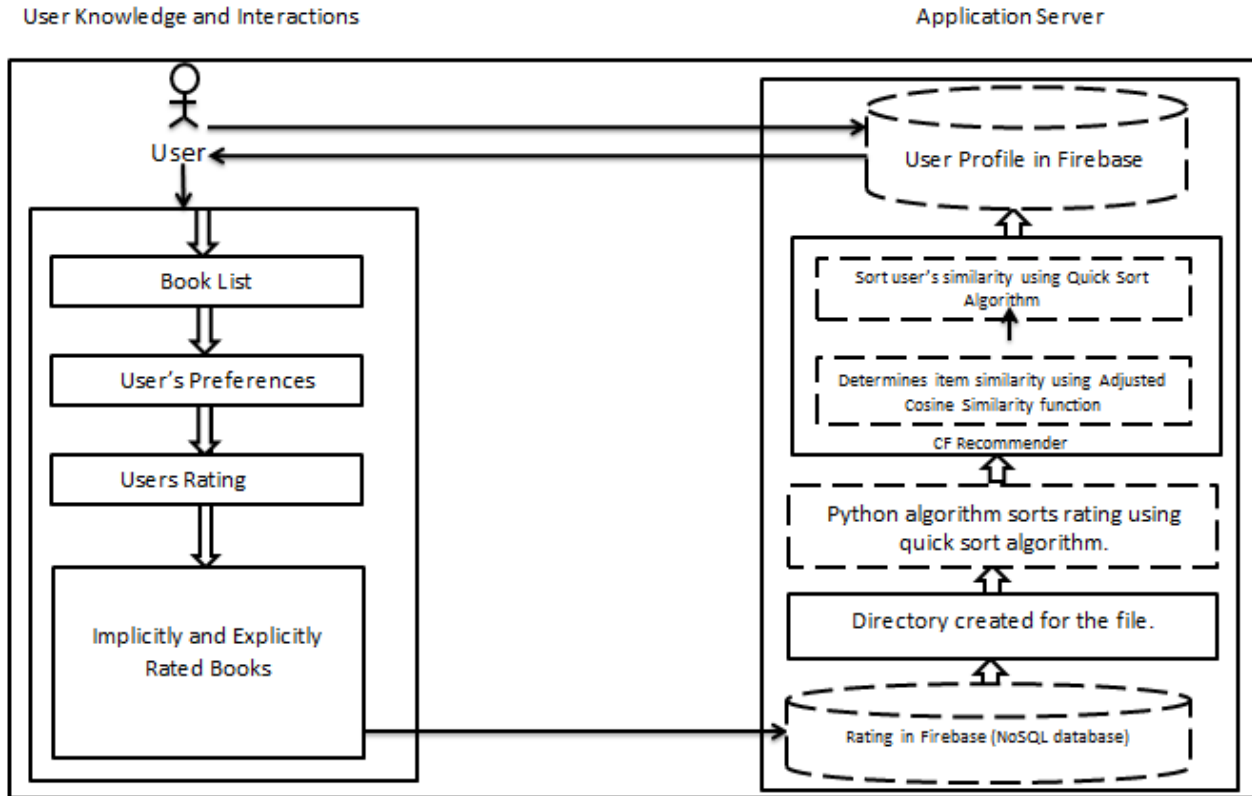


Fig. 5 Architecture of the Proposed System.

globally. User activities includes querying the web application for specific choices, viewing of reading materials (books), adding books to cart, completing the purchases or rental processes and manually rating books, etc.

Book List: A book list in the proposed system refers to a collection of recommended or non-recommended books presented to the user through the web interface.

User's Preferences: The user's preference is used describe the user's preferred choice of suggested books. Different users have different choice of books. A collection of these user choices contribute to the effective recommendation of books for other users.

Users rating: The user rating describes the act of the user providing ratings for the books in the online book shop.

Implicitly and Explicitly Rated Books: This represents a collection of book ratings which are obtained by explicitly providing users with an interface/option to rate books online. Rated books data is also obtained by implicitly observing and taking note of how a user interacts with the online bookshop.

Rating stored in Firebase: A collection of different rated books and user list is saved into a real-time NoSQL database called Firebase and used to perform recommendation of items.

Directory Created for the file: For each entry in the Firebase database, links are created to ensure access to data from the EMMA Recommender web application.

Python Algorithm sorts rating using quick sort algorithm: Python algorithm implements quick sort algorithm which ensures fast sorting of rating list.

Collaborative Filtering (CF) Recommender: This involves the generation of books recommendation list for users using

the inputs in the database. The engine of the CF Recommender performs the computation of similarity among items with other rated items in the firebase database for a given user using an adjusted cosine similarity model given as:

Directory Created for the file: For each entry in the Firebase database, links are created to ensure access to data from the EMMA Recommender web application.

Python Algorithm sorts rating using quick sort algorithm: Python algorithm implements quick sort algorithm which ensures fast sorting of rating list.

Collaborative Filtering (CF) Recommender: This involves the generation of books recommendation list for users using the inputs in the database. The engine of the CF Recommender performs the computation of similarity among items with other rated items in the firebase database for a given user using an adjusted cosine similarity model given as:

$$s(i, j) = \frac{\sum_{u \in U} (R_{u,i} - R_u)(R_{u,j} - R_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - R_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - R_u)^2}} \quad (3.1)$$

Where $s(i, j)$ represents the similarity score between a given item i and another item j in the rating system. $u \in U$ represents every user u in a set of users U . $R_{u,i}$ is the rating user u gives to item (book) i . Also, R_u represents the average rating user u gave for all items user u rated. $R_{u,j}$ is the rating user u gives to next item (book) j . When the similarity index is computed and results are obtained, quick sort algorithm is applied to sort the rated items in ascending order. And finally the results are forwarded into the user profile base.

User Profile: The user profile describes a database containing

user's personal details, their implicit/explicit ratings for books and list of recommendations provided using user's similarity with other users.

4. RESULTS AND DISCUSSION

Table 4.1 Experiment data results for the ONLINE BOOK RECOMMENDER system.

Users ratings		Recommendation
User id	Number of books rated	% Accuracy using RMSE
278859	20	91
345	122	95
456	12	87
567	19	89
67455	35	90
63532	16	89
65	14	75

234564	12	84
34789	55	88
5635	65	75
17	123	98
118	2123	98.
904	13	75
206	44	78
215	99	60
2442	111	92
32111	291	82

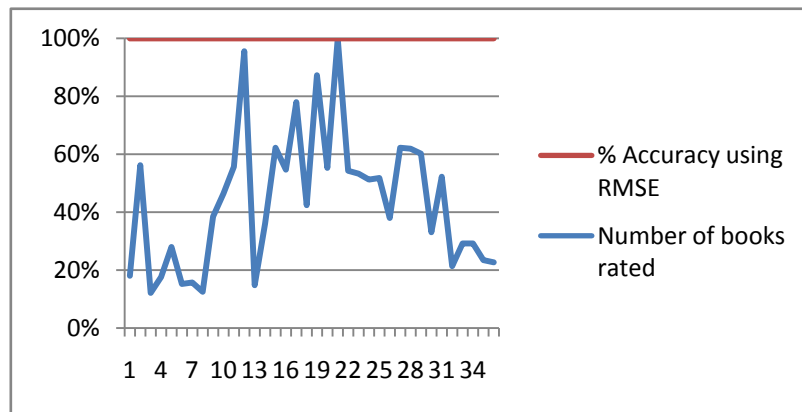


Fig. 6 a graph of the number of books rated against the percentage accuracy of recommendation

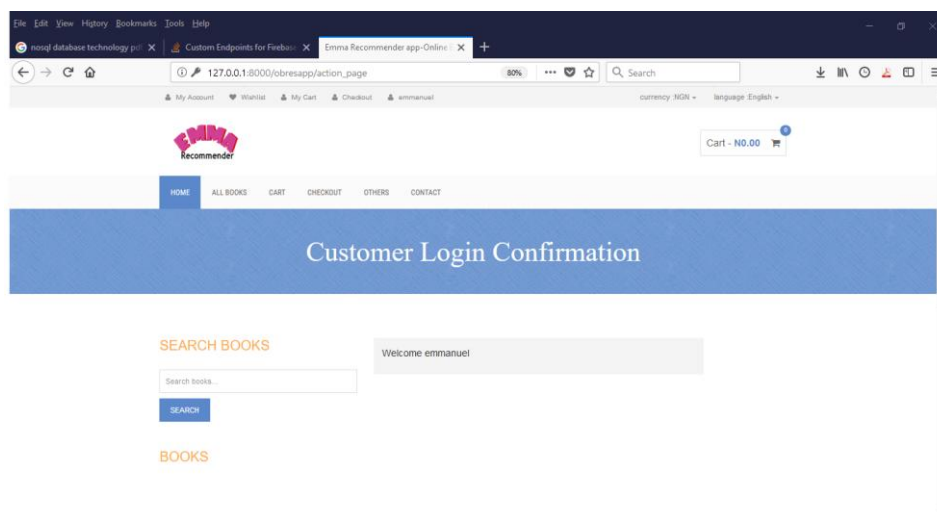


Fig. 7 A User Login into the recommender system.

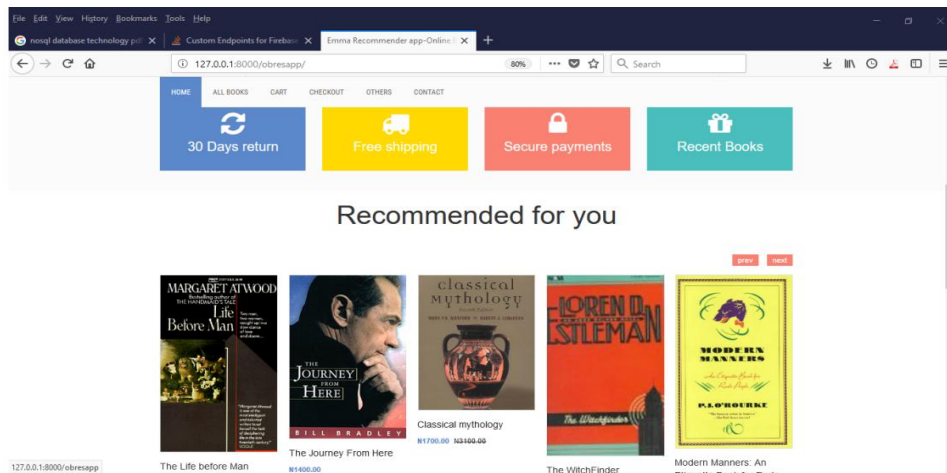


Fig. 8 Personalized recommendation result for a signed-in user.

Table 4.1 tabulates the results of the experiment carried out by observing the number of books rated by each user and the root mean square error (RMSE) recorded in each experiment. The result in the table presents the user ratings using the different user ids and the number of learning materials (books) that were rated in all. From the result it can be observed that the items rated does not directly determine the percentage accuracy of the recommendation. However, the complete absence of a book rating will still yield a zero percent recommendation as values will not be available for similarity computations

The experiment on the EMMA RECOMMENDER App is further visualized using a line chart. From the picture seen in Fig. 6, the graph presents a bell-shape is interpreted as follows: the increase in the number of books rated is not corresponding with the increase in accuracy. Hence, the accuracy is not determined by high number of ratings but yet is high consistently. Figure 7 and 8 shows the result and the user interface available for users of the app.

5. CONCLUSION

This paper was able to present a comprehensive review on researches previously targeted on improving recommender systems. It also introduced a real-time database, an efficient quick sort algorithm and adjusted cosine similarity algorithm to improve on recommender systems. From the results and visualizations, we can deduce that the accuracy of rating followed a normal distribution which suggests consistency and efficiency.

Hence, future works should target on securing recommender system data against attacks and also improving on the algorithms used.

5. REFERENCES

- [1] Akshita, J., and Smita, A. (2013). Recommender system: review. International Journal of computer application, 71(24), 38-42.
- [2] Burke, R. (2007). Hybrid web recommender systems. The Adaptive Web, Springer Berlin, Heidelberg, 377-408.
- [3] Cheng, Z., and Hurley, N. (2010). Effective diverse and obfuscated attacks on model-based recommender systems. RecSys '09: Proceedings of the Third ACM Conference on Recommender Systems, ACM, New York, NY, USA, 141-148.
- [4] Desrosier, C., and Karypis, G. (2012). A comprehensive

survey of neighbourhood-based recommendation methods. Department of Computer Science and Engineering. University of Minnesota, Minneapolis, USA. 5-33.

- [5] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76-80.
- [6] Mahmood, T., and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. Hypertext ACM publications, 73-82.
- [7] Marlin, B. (2004). Collaborative filtering: a machine learning perspective. Thesis research. Department of computer science, University of Toronto. 1-118.
- [8] Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In Proceedings of the eighteenth national conference on artificial intelligence (AAAI-02), Edmonton, Alberta, 187-192.
- [9] Onah, D. and Sinclair, J. (2015). Collaborative filtering recommendation system: a framework in massive online courses. Proceedings of the 9th International Technology, Education and Development Conference, Madrid, Spain. 1249-1257. isbn:9788460657637.
- [10] Resnick, P., and Sami, R. (2007). The influence limiter: provably manipulation-resistant recommender systems. RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems, ACM, New York, USA. 25-32.
- [11] Sarwar, B., Karypis, G., Konstan, J., and Reid, J. (2001). Item-based collaborative filtering algorithms. GroupLens research group, Army HPC research center, University of Minnesota, Minneapolis, 1-11.
- [12] Simpson, J., & Weiner, E. (2016). Recommender. Oxford English Dictionary. Oxford: Oxford University Press.
- [13] Su, X., and Khoshgoftaar T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence. Hindawi publishing corporation. 1-19.
- [14] Zhao, Z. and Shang, M. (2010). User-based Collaborative Filtering Recommendation Algorithms on Hadoop. Third International Conference on Knowledge Discovery and Data Mining. IEEE journal. 478-481. doi: 10.1109/WKDD.2010.54.