International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-8, Issue-2, February 2021 Regular Edition E-ISSN: 2349-7084

An Experimental Study on the Accuracy and Efficiency of Some Similarity Measures for Collaborative Filtering Recommender Systems

Abba Almu[1*], Ziya'u Bello[2]

[1*,2] *Dept. of Mathematics, Computer Science Unit, Usmanu Danfodiyo University, P.M.B 2346, Sokoto – Nigeria.*

Email: almul2003@yahoo.com, ziyaulhaqbello@gmail.com

**Corresponding Author: almul2003@yahoo.com*

Available online at: http://www.ijcert.org

Received: 15/09/2020 Revised: 17/02/2021 Accepted:08/03/2021 Published: 17/03/2021

Abstract: Similarity measure is the fundamental component used in collaborative filtering recommendation technique to provide ratings prediction to users by employing either item-based or user-based recommender algorithms. The collaborative filtering has been widely implemented using various similarity measures but ignores to consider the time taken by the similarity measures to make accurate predictions in different application domains. This paper attempted to assist recommender systems developers to understand appropriate similarity measure depending on the application domain under consideration with less execution time and error rate. It also takes the effect of neighbrhood sizes (k) on the prediction accuracy and efficiency into consideration. The experimental evaluations were conducted on the four similarity measures with the same dataset using Python programming language implementation. The evaluation metrics considered during the experiments are Execution Time, Mean Absolute Error and Root Mean Square Error. The results of the evaluation demonstrated that, Manhattan Distance similarity measure had the best accuracy as well as efficiency of predictions in this study.

Keywords: Similarity Measures, Recommender Systems, Collaborative Filtering, Accuracy, Efficiency

1. Introduction

Recommender system is a kind of system that provides recommendations of goods and services to interested users. it filters the data and recommends the items. It is commonly used in movies, books, music, social media etc. application domains to find items of interest to the users [6]. The User previous ratings/likes and dislikes are processed to generate recent recommendations. Generally, the recommender systems are classified into Content based, Collaborative filtering, and Hybrid [6].

Content-based recommender system recommends items that are similar to the ones the user likes before. It identifies the users' preferences based on the features of the items such as products, movies, jokes, books etc. The Similarity is then calculated based on the items features the user likes previously in comparison with the items the user did not seen before [6]. Collaborative Filtering (CF) recommender system identifies the similar users and provides recommendations based on what user prefers. The system recommends items to the user by using preferences of the similar users in the past with that of the target user. The similarity is then computed using similar preferences in the ratings history of other users [6]. Hybrid recommender system is based on the combination of the content based and the collaborative filtering systems [6]. For example, Collaborative filtering system suffers from cold start problem, since it cannot recommend items with no ratings. Whereas the content based system limit its prediction with ratings since it relies on the content of the items that depends on the features and description that can be easily obtained. Among these systems, the CF classifications are the most popularly implemented techniques in today's recommender systems [11].

Several studies have been conducted in the area of recommender system similarity measures comparison ([1],[13],[5],[12],[14], [6],[4],[2]). Among these studies, the study [13] fails to consider manhattan distance similarity. It ignores the time taken by each similarity measure to make predictions. The effect of neighbourhood size which influences the quality of the prediction is also not considered.

This paper conducted experimentations to evaluate the performances of the four similarity measures on the same platform and determine the best similarity measure based on accuracy and efficiency. The study discovers the Similarity Measure with least execution time and the one that provides more accurate predictions. This is needed since the number of ecommerce sites increase with the number of users accessing the sites and the users prefer less time in getting good result. Hence, it is very important that recommendations are generated accurately in less time.

The remainder of the paper is presented as follows: Section 2 gives brief reviews on some previous studies of similarity measures comparison. Section 3 describes the material and methods used for the implementations and comparisons. Section 4 discusses and analyses the evaluation results obtained, then finally conclusions and future directions are highlighted in section 5.

2. Related Works

Jayvardhan, Thomas and Yadav [4] presented a CF Recommender for user based and item based approaches using various Similarity Measures. The work compares the performances of three different measures and the effect of predictive accuracy when building recommender systems. The work is evaluated using RMSE metric and result demonstrated that, Euclidean Distance tends to be a better choice when building user-based CF while for item-based CF the accuracy of prediction of the Tanimoto Coefficient demonstrated to be the best. However, the study ignores the effect of efficiency of the prediction on these measures while building recommender systems.

Madhuri [6] performed a comparative study on some similarity measures for Item-based recommendations. The study evaluated and analyses the execution time taken by each to generate Top-N recommendations. The similarity measures experimented includes Cosine, Adjusted Cosine, Correlation and Extended Jaccard Co-efficient using Jester dataset. The result reveals that, the Cosine and the extended Jaccard metrics have the less execution time during recommendations. Among these measures, the Extended Jaccard demonstrates the least execution time. However only four similarity measures were considered in the study, hence different similarity measure can be used to see which gives the most accurate answer and least execution time when compared with the other similarity measure. However, the study ignores accuracy of the prediction.

Ali, Saeed and Teh [2] conducted a study on similarity measures for data clustering algorithms. The fifteen publically available datasets classified as high and low dimensional were evaluated on a single framework in this study. The results of the study indicated that, Average Distance tends to be the best accurate measure for all clustering algorithms considered in the study. Moreover, it is also one of the fastest in terms of convergence. Likewise, Pearson Correlation is not suitable for Low dimensional dataset and fails the compatibility test on centroid based algorithms. But, it is mostly suggested for use in high dimensional dataset based on hierarchical techniques. However, this work is limited to only continuous data.

Taner, Efecan, and Zeki [14] conducted a comparative study of CF algorithms on a number of similarity measures for user-based and item-based movie recommendations, to ascertain the goodness of their performances. The study is experimented on Apache Mahout Framework with a view to observing how a neighbour affects the prediction quality. The result of the study shows that for user-based similarities, uncentered cosine provides the fastest recommendations while for item-based similarities, Tanimoto Coefficient gives the fastest recommendations. The algorithms were implemented on a movie recommender system only, but it can be used in any other dataset or recommender. However, the study focused only on the speed of the recommendation without taking recognition for the accuracy of the predictions.

Ajay and Minakshi [1] conducted a study on some similarity measures for recommender systems. The study considers traditional similarity measures consisting of Pearson correlation coefficient, Cosine similarity, Mean squared difference, Jaccard similarity, and Proximity–Impact–Popularity similarity. The result reveals that traditional similarity measures failed to detect effective similar neighbours for cold user. However, the study fails to implement these similarity measures using real dataset.

Shalini, Hong and Shri [12] compared some similarity measures for memory-based CF recommender systems. It investigates users' interest with a view to provide good recommendations and also reduce the data sparsity influence. The study proposed that items should be categorized in a hierarchy so as to enable correct predictions for all missing items ratings. However, the work ignores the implementation in order to understand the validity or invalidity of the hypothesis.

Lamis, Chadi, Jacques, and Demerjian [5] presented a work on CF recommender systems for evaluating the accuracy of some similarity measures. The study focused on most used similarities namely; Pearson Correlation, Cosine Vector, Mean Squared Difference, Spearman Rank Correlation, Frequency-weighted Pearson Correlation, Weighted- Pearson Correlation and Discounted Similarity. The study implemented the measures on the MovieLens with different sample using MAE and RMSE accuracy metrics. The result of the study shows that, weighted Pearson correlation demonstrated better accuracy of predictions in a sparse dataset followed by Frequency weighted Pearson correlation. While in dense dataset the Spearman rank correlation demonstrated good accuracy. However, the study only considers the accuracy of the predictions without minding the speed of predictions.

Suganeshwari and Syed [13] performed a comparison on the study of similarity measures for CF recommendations. The study investigated the influence of popular similarity metrics used for continuous data predictions on a unified framework. The MovieLens dataset with accuracy and mean average precision metrics were used during experimental evaluation. The study reveals that, Pearson correlation produces better results than the remaining similarity measures considered. It also shows that item-based CF methods produce good quality prediction than the user - based CF methods. However, the study fails to consider Manhattan Distance similarity measure and the time taken to make prediction.

3. Methodology

In this section, the measures of similarity, proposed experimental architecture, dataset description as well as the evaluation metrics are discussed.

3.1 Measures of Similarity

These are the description of similarity measures considered in this study.

3.1.1 Manhattan Distance

Manhattan distance is one of the similarity measures used for finding similar neighbours in recommender systems. Perlibakas [10] stated that, it is among the best distance measures. This measure is defined in equation below

$$d = \sum_{i=1}^{n} |x_{i-} y_i|$$

Where x_i and y_i denotes the two vectors. It appropriately works well with datasets containing isolated clusters. But it is sensitive to the outliers.

3.1.2 Euclidean Distance

This is the distance measure between the two objects connected by the path. This measure is typically used with the dense and continuous data [13]. It is given as in the equation below

$$Sim\ (p,q) = \sqrt{\sum_{j=1}^{n} (p_j - q_j)^2}$$

Where *p* and *q* are represented as Euclidean vectors, starting from the origin *n*-space. The two objects are similar when the distance between them is zero while dissimilar objects have higher distances. However, this measure is sensitive to the outliers.

3.1.3 Cosine Similarity

Cosine similarity is the measure of the angle between the two objects. The smaller the angle between the objects, the higher similarity scores. This measure tends to have accurate values in a sparse dataset [13]. The cosine similarity is described in the equation below

$$sim(p,q) = \cos(p,q) = \frac{\vec{p}\ .\ \vec{q}}{|p|\ X\ |q|}$$

Where "**.**" denotes dot product of the two objects, $\vec{p}$ and $\vec{q}$ represent the vector ratings for users *p* and *q*.

The drawback of the cosine similarity is considering the missing values as the negative ones. It fails to consider rating scales. To address this problem, Pearson correlation coefficient utilises cosine similarity with some sort of normalisation.

3.1.4 Pearson Correlation Coefficient

This is the popular similarity metric that utilises co-rated items with the deviation of the average ratings of the item [13]. It measures the two users or items in terms of linear relationship between them. Also, it is similarity scores ranges from -1 to +1. The Pearson correlation is shown as in equation below

$$\mathrm{sim(p,q)} = \frac{\sum_{u \in U_{pq}} (r_{up} - \overline{r_p})(r_{uq} - \overline{r_q})}{\sqrt{\sum_{u\varepsilon U} (r_{up} - r_p)^2}\ \sqrt{\sum_{u\varepsilon U} (r_{uq} - r_q)^2}}$$

Where, U_{pq} denotes common users ratings for both items *p* and *q*, $\overline{r_p}, \overline{r_q}$ denotes average ratings of user *u* for items *p* and *q*. This measure fails to provide accurate result when two users have common ratings.

3.2 Proposed Experimental Architecture Design

The proposed similarity measures comparison, based on accuracy and speed of execution is presented in fig. 1.

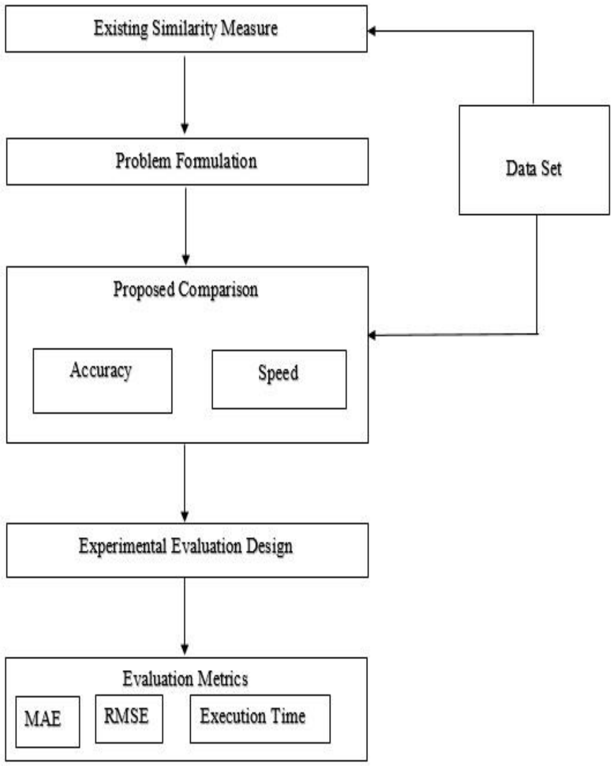

Fig. 1. Proposed Experimental Architecture

3.3 Previous Approaches Analysis

In this component, the various studies on the performance of different similarity measures with various features are examined. The two problems are obtained from the related works conducted in section two. The existing study considered failed to include Manhattan Distance similarity measure and the time taken to make prediction. It also ignores the effect of neighbourhood size which influences the quality of the prediction during recommendations. This study proposes an extension to the existing comparison by studying the tradeoffs between the accuracy and efficiency of the similarity measures under study as it affects the performance of the recommender system.

3.4 Problem Formulation

The research work conducted intensive review of related works on similarity measures comparative study. Several methods, strengths and weaknesses of the existing approaches were described. The research problems derived from previous works and its scope were extensively identified in section 1 and section 2 respectively.

3.5 Conducting Experimental Evaluation

The experiments in this work were conducted to verify and evaluate the accuracy and efficiency of the four similarity measures considered in this study. These measures were then compared in terms of MAE, RMSE and Execution time. The experiments were carried out using movies related datasets. The four similarity measures considered were then coded in python language, where the dataset is accessed as stored in an excel file and find out the nearest neighbours of the active user. It then computes the predictive ratings, i.e. the value that will replace a missing value in active user vector. Evaluation metrics will then be use to evaluate the accuracy and speed of each similarity measure during prediction.

3.6 Dataset Description

MovieLens datasets consisting of 100,000 data points related to ratings given by the users to some movies over the years are used in this study [7]. The ratings in the datasets comprises of integers ranging from 1and 5. All the selected users for the experiment have rated at least 20 movies. In order to validate the results, "ratings.csv" file of the dataset was taken into consideration, having contained the complete dataset information needed. The attributes of the MovieLens dataset are user_id, item_id, ratings and timestamp.

3.7 Performance Metrics Evaluation

The evaluation metrics used for the experimental evaluations of these similarity measures for recommender system in order to verify the performance of the proposed experimental study include:

3.7.1 Mean Absolute Error (MAE)

This metric evaluates the accuracy of the recommendation algorithm using predicted value as against the actual user's ratings [14]. The MAE value is simply calculated by summing up all the pairs and then divides them by the total number of predicted rating pairs. The MAE is given as in the equation below

$$MAE = \frac{\sum_{i=1}^{n} |(p_i - r_i)|}{n}$$

Where, p_i denotes the predicted score of user i, $\boldsymbol{r}_i$ denotes the actual rating score of user i and n denotes total number of predicted ratings pairs.

3.7.2 Root Mean Square Error (RMSE)

It is also accuracy metric with slight modification to the MAE. As it takes the power of 2 to the calculated predicted and actual ratings difference [14]. The RMSE is given as in the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_{i-}\ r_i)^2}{n}}$$

Where, p_i, $\boldsymbol{r}_i$ and n are as already defined in MAE.

3.7.3 Execution Time

Execution time shows how fast the algorithm executes or performs prediction. Each similarity measure has its way of finding similar neighbours and predicting item. By taking the average prediction time taken by a particular similarity measure to predict item ratings, the fastest similarity measure can be obtained.

4. Experimental Results and Discussion

In this section the results of the four similarity measures compared based on efficiency and accuracy were presented, discussed, and analysed.

4.1 Experimental Evaluation Environment

All the experimentations on the four similarity measures were ran on windows based computer consisting of these specifications (i) Intel Core i5-4200M CPU (ii) 2.50GHz processor speed, and (iii) 6GB RAM. In these experiments, the data is divided into training and test sets, with an 80-20 split for appropriate calculations and comparisons of the prediction efficiency and accuracy. The environment where the experiments were conducted is PyCharm that provides the necessary essential tools needed by Python users. It was used to write, modify as well as ran the experimental codes. For each similarity measure, the accuracy is measured using MAE and RMSE Error while the speed (efficiency) of the algorithm is measured using execution time as implemented using python programming language.

4.2 Experimental Evaluation Results

While running the experimentations, the execution time (efficiency) and the accuracy (MAE and RMSE) of prediction of the whole program of each of the similarity measures were taken into account using different neighbourhood sizes (K). The experimental results of the four similarity measures were presented and discussed as follows:

4.2.1 Prediction Efficiency Results Discussion

The Execution Times (ms) of the Similarity Measures are presented as follows:

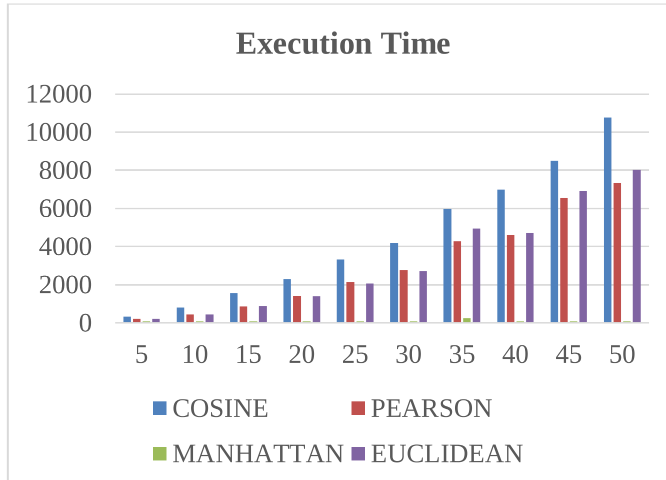

Fig. 2. Similarity Measures Based on Predictive Efficiency (Execution Time Metric)

Figure 2 shows the execution time when testing the similarity efficiency with different value of neighborhood sizes (K). It was observed that Manhattan Distance takes less execution time as compared to Pearson Correlation Coefficient, Euclidean Distance and Cosine similarities, because Manhattan Distance places less emphasis on outliers. As the number of neighbors increases the execution time for all the similarity measures keeps increasing, except for Manhattan Distance, which execution time keeps changing between 70ms and 253ms maintaining lowest execution time.

4.2.2 Prediction Accuracy Results Discussion

A. The MAE of the Similarity Measures

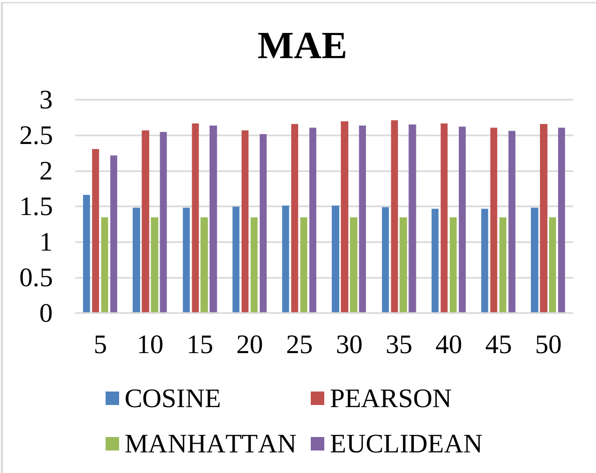

Fig. 3. Similarity Measures Based on Predictive Accuracy (MAE Metric)

It was observed form the above result, Manhattan Distance outperformed among all four similarity measures, showing small error merging, followed by Cosine Similarity, Euclidean Distance and Pearson Correlation Coefficient, because Manhattan Distance tries to reduce all errors equally since the gradient has constant magnitude.

Also, the error value of all the similarity measures continuously increases as the number of neighbors increases when performing MAE check, except for Manhattan which maintained a constant value.

B. The RMSE of the Similarity Measures

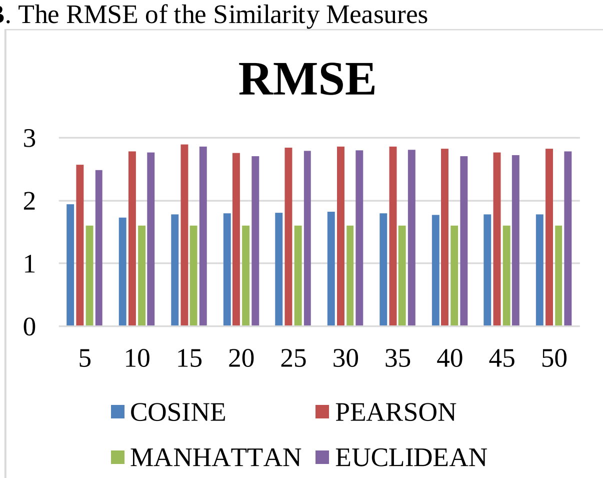

Fig. 4. Similarity Measures Based on Predictive Accuracy (RMSE Metric)

It was observed from the above result, Manhattan Distance outperformed among all four similarity measures, showing small error merging, followed by Cosine Similarity, Euclidean Distance, and Pearson Correlation Coefficient, because Manhattan Distance tries to reduce all errors equally since the gradient has constant magnitude.

Also, the error value of all the similarity measures continuously increases as the number of neighbors increases when performing RMSE check, except for Manhattan which maintained a constant value.

5. Conclusion and Future Work

In this study, four similarity measures used in neighbourhood-based collaborative filtering were compared (Manhattan Distance, Euclidean Distance, Pearson Correlation Coefficient and Cosine Similarity) against three metrics, Execution Time, MAE and RMSE. The study was done on the 100K MovieLens dataset, and the results of the comparison were analyzed.

The Manhattan Distance similarity measure had the best accuracy of predictions as well as fastest execution time. As the number of neighbors increases the execution time for all the similarity measures keeps increasing except for Manhattan Distance, which execution time keeps reducing by maintaining the lowest execution time. Also, the error value of all the similarity measures continuously increases as the number of neighbors increases when MAE and RMSE were computed, except for Manhattan which maintained a minimal error rate.

However, as a future work, the study will be performed on larger datasets with a view to obtaining good and accurate results. The similarity measures will also be subjected to larger number of performance metrics so as to find a better way of ranking these measures from best to worst.

References

[1] Ajay A. & Minakshi C. Similarity Measures used in Recommender Systems: A Study: *International Journal of Engineering Technology Science and Research (IJETSR)*, 2017, 4 (6): 619-626.

[2] Ali S. S., Saeed A., & Teh Y. W. A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data: *PLoS ONE, 2015, 10 (12):* 1-20.

[3] Elmutafa, S. A. A. Principle of Computer System: *LAMBERT Academic Publishing (LAP)*, 2015, pp. 1-2.

[4] Jayvardhan , S., Thomas, V., and Yadav, M. L. Prediction Accuracy Comparison Of Similarity Measures In Memory Based Collaborative Filtering Recommender Systems. *International Journal of Engineering Research & Technology (IJERT)*, 2013, 2 (6): 1105- 1108.

[5] Lamis, A. H., Chadi A. J., Jacques B. A., & Jacques D. Similarity Measures for Collaborative Filtering Recommender Systems *IEEE Middle East and North Africa Communications Conference (MENACOMM), 2018*, pp. 1-5.

[6] Madhuri A. B. Comparative study of similarity measures for item based top n recommendation. *Unpublished project for Bachelor of Technology Degree in Computer Science and Engineering:* National Institute of Technology, Rourkela (Deemed University), 2014.

[7] Harper, F. M., & Konstan, J. A. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS), 2015, 5(4): 1–19*.

[8] Minh-Phung T. D., Dung V. N, & Loc N. Model-based Approach for Collaborative Filtering*: The 6th International Conference on Information Technology for Education (IT@EDU2010) Ho Chi Minh city,2010, Vietnam*. 218-219.

[9] Odunayo, E. O., Ibrahim A.A., Adeleye, S. F., & Olumide, O. O. A Comparative Analysis of Euclidean Distance and Cosine Similarity Measure for Automated Essay-Type Grading: *Journal of Engineering and Applied Sciences, 2018, 13 (11):* 4198-4204.

[10] Perlibakas, V. Face Recognition Using Principal Component Analysis and Wavelet Packe Decomposition. *Informatica, Lith. Acad. Sci., 2004, 15 (2) pp. 243-250*.

[11] Ping, H. Q., and Ming X. Research on Several Recommendation Algorithms. *Procedia Engineering*, 2012, 29: 2427-2431.

[12} Shalini C. S., Hong X.., & Shri R. Measures of Similarity in Memory-Based Collaborative Filtering Recommender System – A Comparison*: MISNC '17 Bangkok, 2017, Thailand,* pp.1-8.

[13] Suganeshwari, G., & Syed I. S. P. A Comparison Study on Similarity Measures in Collaborative Filtering Algorithms for Movie Recommendation: *International Journal of Pure and Applied Mathematics, 2018, 119 (15):1495-1505.*

[14] Taner A. Efecan K. & Zeki B. Comparison of Collaborative Filtering Algorithms with various Similarity Measures for Movie Recommendation*: International Journal of Computer Science, Engineering and Applications (IJCSEA), 2016,* 6 (3):12-13.

Authors Profile

Abba Almu is a lecturer at the Computer Science Unit, Faculty of Science, Usmanu Danfodiyo Univesity Sokoto, Nigeria. He received his PhD from Usmanu Danfodiyo Univesity Sokoto in 2019. His current areas of specialty consist of recommender systems, case-based reasoning and machine learning.

Ziya'u Bello is currently a Masters student in Information Technology at the Computer Science Unit of Usmanu Danfodiyo University, Sokoto. His areas of research interests consist of the applications of information and communications technology in business domain.