

# Review-based analysis of clustering approaches in a recommendation system

Sabeena Yasmin Hera, Mohammad Amjad

Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia University, New Delhi, India

## Article Info

### Article history:

Received Mar 8, 2022

Revised Apr 13, 2023

Accepted May 24, 2023

### Keywords:

Hierarchical clustering

K-mean clustering

Natural language processing

Recommendation systems

Spectral clustering

## ABSTRACT

Because of the explosion in data, it is now incredibly difficult for a single person to filter through all of the information and extract what they need. As a result, information filtering algorithms are necessary to uncover meaningful information from the massive amount of data already available online. Users can benefit from recommendation systems (RSs) since they simplify the process of identifying relevant information. User ratings are incredibly significant for creating recommendations. Previously, academics relied on historical user ratings to predict future ratings, but today, consumers are paying more attention to user reviews because they contain so much relevant information about the user's decision. The proposed approach uses written testimonials to overcome the issue of doubt in the ratings' pasts. Using two data sets, we performed experimental evaluations of the proposed framework. For prediction, clustering algorithms are used with natural language processing in this strategy. It also evaluates the findings of various methods, such as the K-mean, spectral, and hierarchical clustering algorithms, and offers conclusions on which strategy is optimal for the supplied use cases. In addition, we demonstrate that the proposed technique outperforms alternatives that do not involve clustering.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Sabeena Yasmin Hera

Department of Computer Engineering, Faculty of Engineering and Technology

Jamia Millia Islamia University

Jamia Nagar, New Delhi 110025, India

Email: [sabeenayasminhera@gmail.com](mailto:sabeenayasminhera@gmail.com)

## 1. INTRODUCTION

The expansion of the world wide web during the 1990s resulted in an equal expansion of the amount of data that is accessible online, exceeding the capacity of individual clients to process this data. Early research in recommendation system (RS) grew out of information filtering and retrieval research [1]. RS assist users by analyzing and evaluating information from other users in order to identify content, items, and services (such as websites, digital gadgets, movies, books, music, and television series) [2]. Recommender frameworks successfully anticipate what the user could be interested in and add them to the data streaming to the user. Many advancements have been made in the field of RS, allowing a large number of customers to successfully access data for nearly any service, including education, travel, health, cuisine, gaming, and electronics. The ability of a recommender system to create a user's preferences and interests by evaluating this user's behavior and/or the behavior of other users to provide custom recommendations is its most essential feature [1], [3].

Initially, two information filtering strategies for recommendation were introduced: content-based filtering [4] and collaborative-based filtering [5]. Content-based recommender makes recommendations based on the user's previous choices. The previous interest of users on their profile represents the contents.

The item can be represented by the document's attributes and features. Recommendations are created in collaborative recommender based on the prior preferences of other like interested users. Later, a hybrid recommendation model was created by combining these two recommendation techniques [6]. User ratings and written reviews are two of the most common data sources used by recommendation algorithms. Many components and criteria influence a client's decision, including location, quality, quantity, area, purchase history, and client personality, to mention a few. Customers can use the opinions offered for a service, such as reviews and posts, to assist them purchase an item or service. In this regard, various statistics demonstrate the utility of this data from the perspective of the user. According to a BrightLocal local market survey conducted in 2018, 91% of clients between the ages of 18 and 34 trust online reviews as much as personal recommendations. Approximately 57% of clients will only use a company that has four or more ratings. According to a 2019 survey, 82% of consumers read online reviews, with 52% indicating they 'always' read reviews.

This study takes advantage of the fact that customer reviews can accurately predict rating scores for a recommender system that is based on reviews. A system that comes highly recommended will frequently have issues with scalability and sparsity. The problem is addressed by the framework that has been proposed, which does so by employing clustering strategies as a prerecommendation step. In this study, we cluster the users in the dataset using K-means clustering [7], spectral clustering [8], and hierarchical clustering [9] methods, according to the reviews and descriptions representing the users preferences that were provided by the users themselves. The report is broken up into five distinct components. In the first section, we provide an overview of the history, setting, and significance of the study. The work that is relevant to this topic is discussed in section 2. In section 3, we will discuss the methods and models that will be utilized in the work that is being suggested. In addition to this, the workflow of the steps that are contained inside it is described. In the section 4, we take a high-level look at the datasets as well as the assessment metric. In addition to this, it explains the outcomes of the tests that were conducted using the aforementioned datasets and the approach that was proposed. The conclusion and recommendations for further research are offered in section 5.

## 2. LITERATURE REVIEW

In RS, rating prediction can be stated as either review-based score (RBR) prediction or missing score (MS) prediction in a [user, item] matrix. The primary distinction between both is that RBR is based on textual feedback provided by consumers, whereas MS prediction in [user, item] matrix is based on consumer rating history. When the [user, item] matrix is sparse, predicting the rating score becomes challenging. As a result, few studies examine employing text-based information to predict scores. Research by Saumya *et al.* [10] employed a neural network to predict the helpfulness score in order to find the top online product review. The review sentences were incorporated into low-dimensional vectors by a pre-trained model. Three tri-gram, four-gram, and five-gram filters were used to learn the best features of the review text. The lower mean squared error validates the suggested method's forecast accuracy. Existing machine learning-based models that rely on hand-crafted features outperform the proposed method. To solve the problem of data sparsity, Cheng *et al.* [11] used textual review information with ratings. They proposed an aspect-aware topic model for learning user preferences, different aspects of objects, and estimating a user's priority for an item. An aspect rating is then weighted by an aspect importance, which is determined by the targeted user's preferences and the features of the targeted item. According to the results, the suggested method outperforms strong baseline methods, especially for users with a limited amount of ratings. They experimented with and validated the model using datasets from the Amazon and Yelp 2017 challenges. Ratings are sentiments was proposed in [12] as an extension of the hidden factors as topics (HFT) model. That is a model that combines the latent factor model with the latent dirichlet allocation based on the aspect and sentiments unification model. By mixing users sentiments in review texts with their rating ratings, the model could learn more precise latent components of users and things than baseline models. Extensive experiments on large, real-world datasets reveal that the ratings are sentiments (RAS) model outperforms both the latent factor and the HFT models, to some extent, alleviates the cold-start problem.

RS uses clustering algorithms to recognize clusters of consumers with comparable preferences. Many collaborative filtering (CF) based recommendation algorithms have integrated clustering methods to mitigate the problems of scalability and sparsity. Research by Pandya *et al.* [13] presented a new methodology that combines the clustering algorithm with the Eclat algorithm for the generation of prediction rules. First of all, they clustered the rating matrix by user similarity. The clustered data is then converted to boolean data and efficient rules are generated using the Eclat algorithm on boolean data. Finally, a recommendation was generated depending on the rules. The investigation shows that the strategy not only reduces sparsity but also improves system accuracy. Research by West *et al.* [14] proposed a RS based on a hierarchical clustering technique. They proposed a method named EigenfactorRecommends, a citation-based

technique for optimising academic navigation. To put the theory into practice, they have used the AMiner dataset. Research by Wei *et al.* [15] evaluated a RS film using a hybrid strategy that customizes both the sentence-level tags posted on the films and the personal scores given by the user.

Traditional methods measure [[user, item]] matrix similarity, whereas this method uses singular value decomposition (SVD) techniques and matrix factorization. Research by Mishra *et al.* [16] created a new method that takes into account both sequential and content information in online navigation patterns. Soft clusters were also taken into account during clustering, which aids in capturing users diverse interests. The suggested system used similarity upper approximation and SVD for the production of user recommendations. The MSNBC benchmark dataset, a simulated dataset, and the cyber threat intelligence (CTI) dataset were used to test our method. To assert the viability of the proposed model, it was compared to a first-order Markov model and a random prediction model. Research by Ghazarian and Nematbakhsh [17] suggested a collaborative approach that produces group recommendations based on the item and the user's similarity. Similarity of items is recognized using support vector machines (SVM) and similarity of users is recognized using similarity measures. In addition, it fills the vacant entries of the user-item matrix by predicting the most suitable values. While the scheme demonstrates a greater accuracy value and decreased error rate, the issue of scalability continues to be unresolved. Research by Xue *et al.* [18] proposed a cluster-based CF system based on K-means to smooth out the unrated data by cluster for each user. The clusters created from the training data serve as the foundation for data smoothing and neighborhood selection in the discussed technique. As a result, recommendations were more accurate and efficient. According to empirical investigation datasets from MovieLens and EachMovie, the proposed approach consistently outperforms current state-of-the-art CF algorithms. From the available literature, it can be seen that although textual reviews and clustering algorithms are used for score prediction in RS, they have not been used together. Therefore, in this paper perform partitioning using clustering techniques on the textual information and study the effect of these on the rating prediction of the existing reviews.

### 3. METHOD

The strategy that has been suggested for predicting rating scores will be illustrated in this section. The suggested method includes the following processes: acquiring datasets, doing data pre-processing, extracting features, clustering the dataset, and applying recommendation algorithm. These procedures are illustrated in Figure 1.

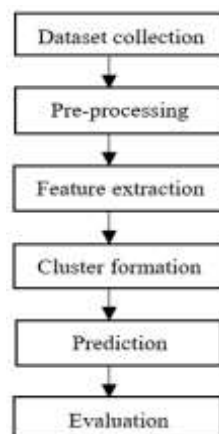


Figure 1. Flowchart of the proposed work

#### 3.1. Pre-processing

Once the data is collected, it needs some pre-processing before analyzing and evaluating the data. Incomplete, inaccurate, and contaminated data analysis can lead to inappropriate and below quality results. Since, the data is in natural language form, a set of natural language processing tasks needs to be performed before further processing. It includes tokenization, stopwords removal, and stemming.

Tokenization is the first step in pre-processing, it involves dividing longer text strings into smaller parts or tokens. It is possible to tokenize larger pieces of text into phrases and tokenize phrases into words. Tokenization is also known as text segmentation or lexical analysis. It detaches numbers, words, symbols, and other characters from string. Next step is stopwords removal. Stopwords are words with high prevalence

all over the sentences; these words don't contain much valuable information. Such words are generally used to connect components of a sentence instead of displaying topics, items or purpose. We can remove words like "the" or "and" by comparing text to a stopwords list. Then stemming is performed which reduces words to its root, generally a suffix, by dropping unnecessary characters. Here the morphological forms of words are removed. Several stemming designs are available, including porter and snowball. In our work porter stemmer [19] is used.

### 3.2. Feature extraction

The texts are transformed into a structured form by converting them into vectors for further processing. This vector space modelling deals with feature extraction from texts. A statement is transformed into a number vector based on the bag of words model with a fixed length; each term of the statement is Tf-idf score. Tf-idf weight is a statistical measure used in a set or corpus to assess how essential a term is to a text and is calculated in (1)-(3).

$$Tf - idf \text{ score} = TF \times IDF \quad (1)$$

$$TF(t) = \text{No. of times } t \text{ appears in a document} / \text{Total no. of terms in the document} \quad (2)$$

$$IDF(t) = \log(\text{Total number of documents} / \text{No. of documents with term } t \text{ in it}) \quad (3)$$

Where TF is term frequency, IDF is inverse document frequency, and t is term in the document.

### 3.3. Clustering algorithms

The clustering approach is used to combine users who are similar together into one cluster and users who are different together into another cluster. The approach of clustering is utilized in the study that has been proposed. This clustering tool clusters user datasets using K-means, spectral, and hierarchical methods.

#### 3.3.1. K-means clustering

K-means clustering is supervised learning algorithm. The algorithm here takes textual information of users as an input and divides them into K number of clusters. The clusters here are formed so that the intracluster sum of squares is minimized.

$$\text{minimize } J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c^j\|^2 \quad (4)$$

Where  $x_i^{(j)}$  = the user's review;  $c^j$  = chosen centroid for the cluster generation;  $\|x_i^{(j)} - c^j\|^2$  = distance measure used for clustering; k = number of required clusters.

#### 3.3.2. Spectral clustering

In recent times, spectral clustering has emerged as a widespread clustering method for grouping texts into clusters. Spectral clustering is very helpful when the composition of the individual clusters is highly non convex. The Laplacian matrix is the most important step in spectral clustering technique. It is also called as graph Laplacian [20]. It uses the spectral decomposition of the Laplacian matrix built on the input dataset [21]. The graph for Laplacian matrix is an undirected weighted graph. The users text graph can be built using  $\epsilon$ - neighborhood graph, K-nearest neighbor graph or fully connected graph. The number of clusters existing in the dataset can be deduced by projecting the points into a non-linear embedding and analyzing the eigen values of Laplacian matrix [22].

$$L = D - Wt \quad (5)$$

Where L is un-normalized Laplacian matrix, D is diagonal matrix, and Wt is weight matrix with  $w_{ij} = w_{ji} \geq 0$ , since graph is undirected.

#### 3.3.3. Hierarchical clustering

In hierarchical clustering technique, clusters are formed by dividing or integrating data in top down or bottom-up manner respectively. It is categorized into agglomerative clustering and divisive clustering based on the approach taken for forming clusters [9]. The agglomerative follows the bottom-up strategy, which builds clusters by considering every user belonging to an individual cluster and then merging these nuclear clusters into larger clusters until all users are lastly placed in a single cluster or otherwise until a

certain termination criterion is fulfilled. The divisive clustering follows the top-down strategy, which breaks down clusters comprising all users into smaller clusters until each user forms a cluster on its own or until certain termination criteria are met. In agglomerative clustering two users are chosen to be merged using linkage function. In this work wards minimum variance linkage is used for cluster formation. In this linkage function the decision to merge two clusters is based on the minimum merging cost. The merging cost of two clusters say A and B can be defined by the sum of squares of the data points to its center [23].

$$cost(A, B) = \sum_{i \in A \cup B} \|\vec{x}_i - \vec{m}_{A \cup B}\|^2 - \sum_{i \in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i \in B} \|\vec{x}_i - \vec{m}_B\|^2 \quad (6)$$

Where  $\vec{m}_j$ =center of cluster j and  $\vec{x}_i$ =each user i.

### 3.3.4. Recommendation and prediction

After the clustering step in order to predict ratings for target users the system computes the similarity of the target user to all its neighborhood users belonging to the same cluster based on their available data. The similarity between users is calculated using cosine similarity [24]. The user with which the target user is found to have maximum similarity is used for prediction of ratings. The detailed for proposed framework is shown in Algorithm 1.

#### Algorithm 1. Algorithm for the recommendation

*Input:* dataset D containing reviews and ratings

*Output:* Predicted rating for a given user

begin

D= dataset

K= number of clusters

N=number of reviews in D

y= rating

n= range of test set for each cluster k

m= range of training set for each user k

Step1: Randomly select K reviews from N as the centroid of initial clusters.

Step2: Iteratively divide D into K clusters using aforementioned clustering techniques

Step3: For each cluster

a) Split data into test-set and training set, test set=0.30 and training set= 0.70

b) Calculate cosine similarity between (test[i], training[j]) say, W[i][j]

For i ← 1 to n

For j ← 1 to m

If max\_value[i] == W[i][j] // search the matrix W[i][j] for maximum value corresponding to each test[i].

Assign test\_y[i] == train\_y[j] // Assign the y value of training[j] to test [i] to get the predicted value of y

End For

End For

End For

End

## 4. RESULTS AND DISCUSSION

Two datasets have been used for the proposed work. One of them is consumer reviews of amazon for beauty and health. It contains 12,071 reviews given by users out of which those having a neutral rating were dropped. The other dataset is amazon fine food reviews dataset and after the pre-processing, top 5,000 reviews were taken into consideration for experimentation. Although the datasets use a 5-star rating system, we've converted it into the "high" ({4, 5}) and "low" ({1, 2,}) binary groups. In addition, we restructured rating estimation as a classification problem where we estimate the probability that a user would "like" an item or not.

### 4.1. Evaluation metric

Mean absolute error (MAE) is used to assess the performance measurement of the proposed framework because of its accuracy and simplicity that suits the experiment's objective. It is one of the most widely used metric for performance evaluation of recommender systems. MAE estimates in a collection of projections the median magnitude of the errors. The relative scores between the predictions and the resulting observation over the test set are averaged [25]. The MAE is calculated in (7).

$$Mean\ absolute\ error = \frac{\sum_{j=1}^n \|p_j - a_j\|}{n} \quad (7)$$

Where n=number of ratings;  $p_j$ =predicted rating of  $j^{th}$  user and  $a_j$ =actual rating of  $j^{th}$  user.

## 4.2. Result and analysis

The study was set out to enhance the accuracy of the rating prediction for RS and to identify the effect of various clustering scheme on it. Hence, this section shows the analysis and results discussion. To conduct the experiment, we used python for implementation and the parameters of the host system were Win 10, intel® core™ i5-8265U, X64 and 8 GB RAM.

Table 1 demonstrates the experimental result of the clustering-based framework for recommender system using textual information. We have evaluated different number of clusters and presented the results for numbers that were close to the optimal result. It shows the effect on different clustering scheme i.e., K-means clustering based recommender system (KCRS), spectral clustering-based recommender system (SCRS) and Agglomerative clustering-based recommender system (ACRS) on the accuracy of the prediction in terms of MAE. We have also determined the threshold value which gave us the minimal MAE.

After extensive testing it is found to be near  $\log_2[n/2] \pm 1$ . It also appears that the performance of the RS depends on the type of dataset and the number of clusters. The clustering-based framework outperforms the non-clustering one only when the above two conditions are met. Figure 2 shows the comparison between the optimal values of KCRS, SCRS, and ACRS. For the Amazon consumer reviews dataset, SCRS outperforms the other two approach. It is because the dataset was concave in nature. Whereas for Amazon fine food reviews ACRS performed better than the KCRS and SCRS as shown in Figure 2(a). The proposed approach is then compared to the non-clustering approach for RS as shown in Figure 2(b). It shows that the proposed clustering approach performs better than the non-clustering one.

Table 1. MAE of KCRS, SCRS, and ACRS for different number of clusters

Number of clusters	MAE					
	Amazon consumer reviews			Amazon fine food reviews		
	KCRS	SCRS	ACRS	KCRS	SCRS	ACRS
C=11	0.0871	0.0233	0.0857	0.1858	0.1718	0.1051
C=12	0.0911	0.0202	0.092	0.1556	0.1592	0.0908
C=13	0.0737	0.0211	0.0717	0.1166	0.1911	0.0842
C=14	0.0747	0.0209	0.0616	0.1968	0.1849	0.0917

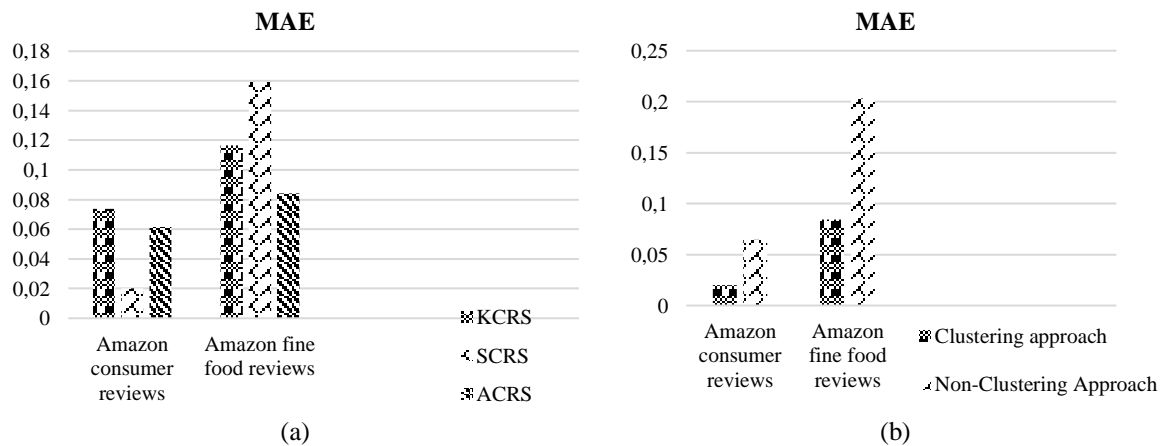


Figure 2. Comparison of optimal results KCRS, SCRS, and ACRS for the datasets in (a) frame 1 comparison of the best performing clustering-based approach and (b) frame 2 non-clustering based approach each of the algorithm

## 5. CONCLUSION





User evaluations are extremely valuable when it comes to the process of making recommendations since they not only provide an objective rating of a service but also more accurately define the objectives of a user. In this research, we describe the recommendation framework by utilizing a variety of clustering methods. By using clustering, we are able to narrow the search space and more quickly locate the user or item that is most analogous to the one now being used. In addition to this, it solves the scalability issue. The results of the experiments led us to the conclusion that the system for predicting rating scores that is based on clustering works better than the one that is not based on clustering when the number of clusters that are

produced is optimal. When the number of clusters is optimally considered, a best-fit clustering-based prediction system will perform better than other clustering methods-based prediction systems for any given number of clusters, and it will perform better than a non-clustering-based RS when the number of clusters is optimally considered. The first suggestion for upcoming advances would be to evaluate evaluations on the grounds of each phrase, given that each phrase may have a unique polarity and perspective based on objectivity. This would be the first step in a series of proposed future developments. Further refinements could involve analyzing reviews with the punctuation in them and figuring out what it all means. The scope of the textual analysis can be expanded even further to incorporate the determination of the emojis that are written alongside the textual material. Other methods of clustering can also be used to group items that are similar to one another. The framework also lends itself well to research in a variety of other subject areas.





## REFERENCES

- [1] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1–2, pp. 101–123, 2012, doi: 10.1007/s11257-011-9112-x.
- [2] E. Frias-Martinez, G. Magoulas, S. Chen, and R. Macredie, "Automated user modeling for personalized digital libraries," *International Journal of Information Management*, vol. 26, no. 3, pp. 234–248, 2006, doi: 10.1016/j.ijinfomgt.2006.02.006.
- [3] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.
- [4] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, Berlin, Heidelberg: Springer, 2007, pp. 291–324. doi: 10.1007/978-3-540-72079-9\_9.
- [5] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, Berlin, Heidelberg: Springer, 2007, pp. 325–341. doi: 10.1007/978-3-540-72079-9\_10.
- [6] Z. Ji, H. Pi, W. Wei, B. Xiong, M. Wozniak, and R. Damasevicius, "Recommendation based on review texts and social communities: a hybrid model," *IEEE Access*, vol. 7, pp. 40416–40427, 2019, doi: 10.1109/ACCESS.2019.2897586.
- [7] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005, doi: 10.1109/TNN.2005.845141.
- [8] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007, doi: 10.1007/s11222-007-9033-z.
- [9] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012, doi: 10.1002/widm.53.
- [10] S. Saumya, J. P. Singh, and Y. K. Dwivedi, "Predicting the helpfulness score of online reviews using convolutional neural network," *Soft Computing*, vol. 24, no. 15, pp. 10989–11005, 2020, doi: 10.1007/s00500-019-03851-5.
- [11] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web-WWW '18*, New York, USA: ACM Press, 2018, pp. 639–648. doi: 10.1145/3178876.3186145.
- [12] D. Yu, Y. Mu, and Y. Jin, "Rating prediction using review texts with underlying sentiments," *Information Processing Letters*, vol. 117, pp. 10–18, 2017, doi: 10.1016/j.ipl.2016.08.002.
- [13] S. Pandya, J. Shah, N. Joshi, H. Ghayvat, S. C. Mukhopadhyay, and M. H. Yap, "A novel hybrid based recommendation system based on clustering and association mining," in *2016 10th International Conference on Sensing Technology (ICST)*, IEEE, 2016, pp. 1–6. doi: 10.1109/ICSensT.2016.7796287.
- [14] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, "A recommendation system based on hierarchical clustering of an article-level citation network," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 113–123, 2016, doi: 10.1109/tbdata.2016.2541167.
- [15] S. Wei, X. Zheng, D. Chen, and C. Chen, "A hybrid approach for movie recommendation via tags and ratings," *Electronic Commerce Research and Applications*, vol. 18, pp. 83–94, 2016, doi: 10.1016/j.elerap.2016.01.003.
- [16] R. Mishra, P. Kumar, and B. Bhasker, "A web recommendation system considering sequential information," *Decision Support Systems*, vol. 75, pp. 1–10, 2015, doi: 10.1016/j.dss.2015.04.004.
- [17] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3801–3812, 2015, doi: 10.1016/j.eswa.2014.11.042.
- [18] G.-R. Xue *et al.*, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, USA: ACM, 2005, pp. 114–121. doi: 10.1145/1076034.1076056.
- [19] M. Bounabi, K. El Moutaouakil, and K. Satori, "A comparison of text classification methods using different stemming techniques," *International Journal of Computer Applications in Technology*, vol. 60, no. 4, pp. 298–306, 2019, doi: 10.1504/IJCAT.2019.101171.
- [20] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE Comput. Soc., 2002, pp. 107–114. doi: 10.1109/ICDM.2001.989507.
- [21] Z. Zhang and S. R. Kulkarni, "Detection of shilling attacks in recommender systems via spectral clustering," in *FUSION 2014 - 17th International Conference on Information Fusion*, 2014, pp. 1–8.
- [22] H. D. Menendez and D. Camacho, "GANY: a genetic spectral-based clustering algorithm for large data analysis," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2015, pp. 640–647. doi: 10.1109/CEC.2015.7256951.
- [23] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, 2005, doi: 10.1007/s10618-005-0361-3.
- [24] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004, doi: 10.1145/963770.963772.
- [25] H. N. Kim, A. T. Ji, I. Ha, and G. S. Jo, "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation," *Electronic Commerce Research and Applications*, vol. 9, no. 1, pp. 73–83, 2010, doi: 10.1016/j.elerap.2009.08.004.

**BIOGRAPHIES OF AUTHORS**

**Sabeena Yasmin Hera**     graduated with honors from Abdul Kalam Technical University in Lucknow, India in 2016. In 2019, she graduated with honors from the Department of Computer Engineering at Jamia Millia Islamia in New Delhi, India. She worked as a data analyst in 2016-2017, and she developed algorithms and software in Python and R. She has a strong understanding of machine learning algorithms and concepts such as data preprocessing, regression, classification, and so on, as well as appropriate model selection techniques. She worked as a research fellow from 2019 to 2022, conducting research and analyzing extensive literature, data, and results for heart disease prognosis using machine learning. She has been working as an associate business analyst since 2022, focusing on conversational AI, Chatbot development, and platform integration. She can be contacted at email: [sabeenayasminhera@gmail.com](mailto:sabeenayasminhera@gmail.com).



**Mohammad Amjad**     received his B.Tech. in computer engineering with honors from Aligarh Muslim University in Aligarh, India in 1997. He received his M.Tech. (information technology) degree with honors from IP University in New Delhi, India in 2008, and his Ph.D. (computer engineering) degree from Jamia Millia Islamia in 2013. In 1997, he began working as a senior network engineer for Crescent Computer Pvt. Ltd. in New Delhi. He was in charge of the networks design for Northern Indian Railway, Telecommunications of India Limited (TCIL), and Videsh Sanchar Nigam Limited (VSNL). Following that, he worked as a lecturer in the Department of Computer Science and Information Technology at MJP Rohilkhand University in Bareilly, Uttar Pradesh. He joined the Department of Computer Engineering University Polytechnic, F/o engineering and technology Jamia Millia Islamia in 2002, and then in 2006, he joined the Department of Computer Engineering F/o engineering and technology, Jamia Millia Islamia New Delhi. He has approximately 18 years of teaching experience. Dr. Amjad is a professor in the Department of Computer Engineering at Jamia Millia Islamia's Faculty of Engineering and Technology in New Delhi. Dr. Amjad received his Ph.D. for his work in the field of Mobile Ad hoc Networks, where he investigated clustering in MANETs and energy consumption in wireless sensor networks, as well as designing a "Quality of Service Framework for Mobile Ad hoc Networks." Dr. Amjad's research has been published in prestigious refereed international journals and international conferences. He published 69 research papers in national/international conferences and journals such as IEEE explore/Springer/International Journals in the United States of America/International Conferences in China, Malaysia, and the United States of America. He is actively involved in R&D activities in the areas of MANET, WSN, IoT, and Network Security systems. Dr. Amjad is an ISTE Life Member and an expert lecturer on IoT, wireless sensor networks, mobile communication, and network security. He can be contacted at email: [mamjad@jmi.ac.in](mailto:mamjad@jmi.ac.in).