Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 1 (2022) 57 - 63 ISSN Media Electronic: 2580-0760

E-commerce Recommender System Using PCA and K-Means Clustering

Dendy Andra A.N[1], Z. K. A. Baizal[2]
[1,2]School of Computing, Telkom University
[1]dendyandra@students.telkomuniversity.ac.id, [2]baizal@telkomuniversity.ac.id*

Abstract

Recently, recommender system has an important role in e-commerce to market products for users. One of recommender system approach that used in e-commerce is Collaborative Filtering. This system works by providing product recommendations based on products liked by other users who have similar preferences. However, sparse conditions in user data will cause sparsity problems, namely the system is difficult to provide recommendations because of the lack of important information needed. Therefore, we propose an e-commerce product recommendation system based on Collaborative Filtering using Principal Component Analysis (PCA) and K-Means Clustering. K-Means is used to overcome sparsity problems and to form user clusters to reduce the amount of data that needs to be processed. While PCA is used to reduce data dimensions and improve clustering performance of K-Means. The test results using the sports product dataset on the Olist e-commerce show that the proposed system has a lower RMSE value compared to other methods. For the number of neighbors of 10, 20, 30, and 40, our system obtains values of 0.771806, 0.75747, 0.75304, 0.75304, and 0.75270.

Keywords: recommender system, collaborative filtering, principal component analysis, k-means, e-commerce

1. Introduction

Recommender system is an important part of a business strategy in e-commerce [1]. Recommender system is used to provide product recommendations to users with the aim of helping users get the desired product [2]. Collaborative filtering (CF) is a recommendation system method used in e-commerce. Basically, this CF-based system generates recommendations by utilizing information from other users who have similar preferences.

However, user data stored in e-commerce is sometimes in a sparse state. The incompleteness of user-related data, such as a lot of empty or unavailable data creates a sparsity problem, that is the lack of important information for recommender system to produce recommendations for users [3]. As a result, the accuracy of recommendations will decrease and the system cannot generate recommendations that are relevant to the user. This inaccuracy of product recommendations to users can lead to decreased user confidence in the system [4].

Several studies have been conducted to overcome sparsity problem in the recommendation system, one of solution is to apply the clustering technique. Research [5] uses the K-Means model to recommend products in stores based on customer categorization. In the e-commerce domain, researchers [1] designed a product recommendation system for users and a stock keeping unit (SKU) for sellers by combining the recommendation results from PCA and K-Means with the recommendation results using K-Means. In education, K-Means has been used to recommend specialization courses [6].

Clustering has also been implemented on a CF-based system. In [7] a recommendation system is proposed using Principal Component Analysis (PCA) and K-Means to recommend films. In the field of music, researchers [8] designed a CF-based song recommendation system using a combination of PCA with K-Means and Hierarchical Clustering methods.

K-Means is one of the clustering algorithms that is widely used in recommender systems and data mining [9]. In the CF-based recommendation system, the K-Means Clustering approach is able to overcome sparsity problem effectively [3][10]. However, K-Means has a weakness, the performance of data clustering decreases if the dimensionality of the processed data is very large [8]. In addition, the optimal number of clusters from K-Means is unknown, so it needs to be determined beforehand so that the clustering results are good.

Accepted: 11-01-2022 | Received in revised: 26-01-2022 | Published: 01-02-2022

To overcome the problem of data dimensionality, K-Means can be combined with data dimension reduction methods, one of which is the PCA method. The PCA method is able to improve the performance of the clustering algorithm well [11]. Meanwhile, the exact measurement to determine the optimal number of clusters and also to know the data in each cluster has been well partitioned is by using Silhouette coefficient [5][12].

Therefore, in this study, we propose a recommender system for e-commerce products based on Collaborative filtering using Principal Component Analysis (PCA) and K-Means Clustering methods. The K-Means method was chosen because it can be used to overcome sparsity problems in e-commerce. While the PCA method is used to reduce the dimensions of the data and improve the clustering performance of K-Means. Thus, this combination is suitable for handling product data and e-commerce users that have a large number and dimensions. In this study, the optimum number of clusters was determined using the Silhouette coefficient. This measurement is used because it is able to identify the accuracy of the partition for each cluster that is formed. The dataset used in this study is a user dataset and product ratings with the sports category in the Olist e-commerce.

This paper is arranged in the following order: Chapter II discusses the research methods used. Chapter III discusses the research results. And Chapter IV discusses the conclusions of the research and future work.

2. Research Methods

The workflow of the proposed recommendation system has 4 main stages, that is reducing data dimensions, forming user clusters, handling sparsity, and forming recommendations. Our system input data is user data which contains transaction history and geolocation. As well as user rating data that contains user ratings for certain products.

In the data dimensionality reduction stage, the dataset that has been cleaned and ready to be processed, then the dimensions are reduced using PCA. The user cluster forming stage contains the processes of forming a number of user clusters using K-Means. The number of clusters formed is obtained from calculating the Silhouette coefficient. The stages of handling sparsity on the data contain the process of handling sparsity by filling in empty values using the average value of each cluster. The stage of forming product recommendations is the process of forming products that will be recommended to users using the Collaborative Filtering algorithm. Figure 1 shows the workflow of our system.

2.1 Dataset

In this study, there are two datasets used, i.e the user dataset and the rating dataset. These two datasets were created by the Olist from 2016 to 2018. These datasets can be freely downloaded for research. User dataset consists of 7,422 data with 7 features. This dataset contains the user's geolocation and transaction history. The user dataset will be used by PCA and K-Means to form a user cluster. Table 1 shows an example of user data. The product rating dataset consists of 15,944 data and 3 features. This dataset contains user ratings for certain products. Product rating dataset will be used on CF to form recommendations to users. Table 2 shows examples of product rating data.

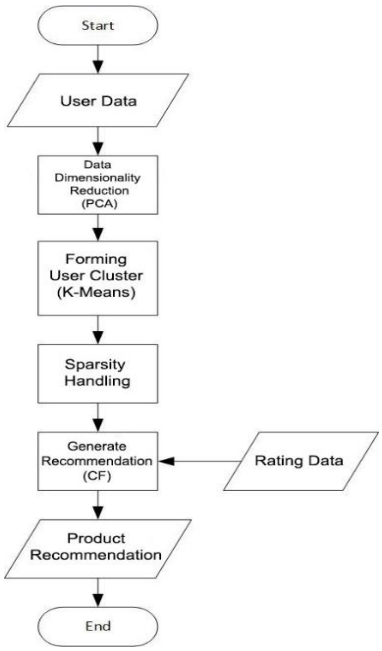

Figure 1. System workflow

Table 1. User data example

Feature	Value
customer_unique_id	00053a61a98854899e70ed204dd4bafe
payment_type	credit_card
total_payment_installments	3.0
total_payment_value	419.18
geolocation_lat	-25.4313
geolocation_lng	-49.2792
total_payment_sequential	1.0

Table 2. Product rating data example

Customer_unique_id	Product id	Rating
bee8c71dfec2314e63b7ef0e2ce70bdd	5581bb179770e44255d6bb9b9e1bcca9	4
4841d5835d0ab4894f2b0a0ac60aa49a	bc911e68db068530ee4d709f33920330	4
4841d5835d0ab4894f2b0a0ac60aa49a	a8fe47ad6f852f93cc92c7b408687de3	5
4841d5835d0ab4894f2b0a0ac60aa49a	ac8e7cf3a658f63dc68dd7f3b8b5f97b	5
…	…	…

DOI: https://doi.org/10.29207/resti.v6i1.3782

2.2 Data Dimensionality Reduction (PCA)

Data that has been preprocessed then will be reduced its dimensions using the PCA. PCA performs dimensionality reduction of a set of correlated data [1]. PCA works by utilizing orthogonal transformation techniques to transform a set of data with correlated variables into a collection of unrelated linear variables. The data that is processed by PCA is numerical data of continuous type so that the vector calculation results are correct. Here are the steps of the PCA algorithm:

Step 1. Standardize the data to have an equivalent value scale. Standardization is done by dividing a feature data value by the average value of all feature data so that the data will be centered at 0.

Step 2. Calculate the covariance matrix to determine the correlation of each feature in the data.

Step 3. Calculate the eigenvector and eigenvalues from the covariance matrix, then sort from the largest value.

Step 4. Choose the principal component by considering the eigenvectors that have the largest or most significant eigenvalues.

PCA will generate a new dataset as a result of the principal component that has the most significant information from the original dataset. Thus, the system needs to rearrange the original data using the selected principal component data. So that the dimensionality of the data that will be used in the next process has been reduced.

2.3 Sparsity Handling

Collaborative Filtering Algorithm will experience problems when the input data is in sparse condition. Thus, it is necessary to handle sparsity in the dataset so that the recommendation results can be better. We need to assume that all users in the same cluster have the same preferences.

Thus, the empty value in the user-item matrix can be filled with the average value of users who are in the same cluster because they have similar preferences. This method is able to improve the performance of the recommendation system based on Collaborative Filtering [3]. Filling in the blank values in this user-item matrix will make it easier to calculate similarity between users.

2.4 Forming User Clusters (K-Means)

All e-commerce users are grouped based on the similarity of data between users using the K-Means Clustering method. K-Means works by calculating the distance or similarity between data objects and grouping the data based on the similarity or proximity to other data [13]. K-Means will form data clusters that have the same resemblance. Here are the steps of the K-Means algorithm:

Step 1. Determine the number of clusters to be formed.

Step 2. Specifies the centroid or center of the cluster. For the initial initiation of the centroid, we can choose data randomly.

Step 3. Calculate the distance of each data to the centroid. In this study, the method of calculating the distance at K-Means uses Euclidean distance with equation 1 as follows [5]:

$$d(x, y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (1)$$

where x and y is data to be measured its distance, n is number of features on data, x_k and y_k is k-th feature on data x and y.

Step 4. Group data based on proximity to the centroid. The smaller d is obtained, the closer the data is to the centroid of the cluster.

Step 5. For the same number of clusters, find new centroid of the cluster by calculating the average of each data in the cluster.

Step 6. Repeat steps 2 to 5, the loop stops when cluster or all data positions do not change anymore.

To find the optimum number of clusters, K-Means will be iterated with different number of clusters. For each iteration, the quality of the cluster will be measured using the Silhouette coefficient in equation 2 as follows [5]:

$$g_i = \frac{m_i - n_i}{\max(m_i, n_i)} \quad (2)$$

where m_i is the minimum average distance between clusters, n_i is the average distance from a point in a cluster to all data in the same cluster.

Silhouette values will be in the range -1 to 1. If the silhouette results are close to -1, it indicates that the data is in the wrong cluster, if it is close to 0 then the data clusters intersect, and if it is close to 1, the data is well clustered. The optimum number of clusters is the number of clusters that have a silhouette value close to 1.

2.5 Generate Recommendation (CF)

Collaborative Filtering (CF) algorithm is used to make product recommendations for users. CF is one of the filtering algorithms in the recommendation system that has been widely used in recent years [14]. CF provides recommendations to users by looking for ranking patterns based on ranking data from other users who have the same preferences [11][15]. The working steps of the Collaborative Filtering algorithm are as follows:

Step 1. Determine the number of neighbors. Neighbors are users who are close/similar to the active

DOI: https://doi.org/10.29207/resti.v6i1.3782

user. The number of neighbors defines the number of users to be considered in calculating the rating prediction.

Step 2. Calculate the similarity of active users to their neighbors using Cosine similarity in equation 3 below [16]:

$$sim(i,j) = \cos(\theta) = \frac{i.j}{\|i\|.\|j\|} \quad (3)$$

where i and j are the dot products between the two users. The range of similarity values is between -1 to 1, where the closer to 1, the more similar. Vice versa, if it is close to 0 then it is not similar, and if it is close to -1 then it is opposite.

Step 2. Predict rating values for products that have never been rated by active users. Predicted value is obtained by equation 4 as follows [17]:

$$\widehat{r_{ua}} = \overline{r_a} + \frac{\sum_{j \in N_u(a)} sim(i,j)(r_{ja} - \overline{r_j})}{\sum_{j \in N_u(a)} |sim(i,j)|} \quad (4)$$

where $\widehat{r_{ua}}$ is the predicted value of user u's rating of product a, $\overline{r_a}$ is the average value of product a 's rating, r_{ja} is the rating of product a by user j, and $\overline{r_j}$ is the average rating of user j.

Step 3. Sort rating predictions from highest to lowest. Then take a number of top products as recommendations.

The output of this CF algorithm are products that are recommended to active users. Where the number of recommendations needs to be defined to indicate how many recommendations should be displayed.

2.6 Evaluation Metrics

In this study, the performance of the recommendation system was measured using the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). RMSE and MAE are evaluation metrics that are commonly used to measure the performance of recommender systems [18]. Both methods evaluate the accuracy of the system's predictions by comparing the predicted rating with the actual rating.

Comparison is carried out to find the error value, that is the deviation between the predicted value and the actual value. The smaller the RMSE and MAE values, the smaller the error value of the prediction. That is, the predicted value is getting better because it is closer to the true value.

MAE is used to calculate the average error value of all data. The MAE value is obtained from the following equation 5 [19]:

$$MAE = \frac{1}{N} \sum_{(u,i)\epsilon N} |r_{ui} - \hat{r}_{ui}| \quad (5)$$

where N is the number of data, r_{ui} is the actual rating of the user, $\hat{r}_{ui}$ is the rating predicted by the recommender system. RMSE is used to calculate the average error by giving a large penalty if the error value is also large. In addition, RMSE also considers the direction of the error (positive error or negative error) where this is not done in MAE. The RMSE value is obtained from equation 6 as follows [19]:

$$RMSE = \sqrt{\frac{1}{|N|} \sum_{(u,i) \in N} (r_{ui} - \hat{r}_{ui})^2} \quad (6)$$

where N is number of data, $\hat{r}_{ui}$ is the predicted rating, and r_{ui} is actual rating from user.

3. Results and Discussions

All datasets used will be preprocessed first so that the data is ready for use in the next stage. This preprocessing includes outlier removal, data scaling, and encoding. The data is then divided into two, namely training data consisting of 4376 data, and test data consisting of 1096 data. Training data is used to train PCA and K-Means in order to generate predictions. While the test data is used to test the performance of the system on a dataset that has never been used before.

At the training stage, we used 5-fold cross-validation so that the results of the PCA and K-Means training could be more accurate. After the training is complete, the system is tested using test data. The system performance is then measured using RMSE and MAE. The system performance will also be compared with the performance of the comparison method.

3.1. Number of Principal Component

The number of principal components in PCA needs to be determined before reducing the dimensions of the data. The selection of the amount is done by considering the large or the most significant explained variance (eigenvalues) compared to the others. In addition, the determination of the number can also be done by testing the number of PCA components on the clustering performance of K-Means. PCA is trained with numerical input data features such as total_payment_installments, total_payment_value, geolocation_lat, and geolocation_lng.

The results show PCA obtained 4 principal components with an average value of explained variance as shown in Table 3. From the table, it can be seen that the initial two components have large and significant explained variance. The two components have a total explained variance of about 0.63, which means that they contain 63% of the important information from the four features. Meanwhile in components 2 and 3, the explained variance is much smaller than the two initial components.

DOI: https://doi.org/10.29207/resti.v6i1.3782

Table 3. Explained variance for each principal component

Principal Component	Average Explained Variance
1	0.32207686399999996
2	0.31023692199999997
3	0.188339114
4	0.17934709999999998

Then we tested the effect of the number of PCA components on the clustering performance of K-Means to ensure the optimum number of components. The test is done by testing the K-Means clustering on the same number of clusters, but the number of PCA components is different. Assuming the number of clusters is 2, then the K-Means clustering performance is obtained for each number of PCA components as shown in Table 4.

Table 4. Clustering performance for each num of component

Num of component PCA	Num of cluster	Silhouette coefficient
1	2	0.849219
2	2	0.791772
3	2	0.757907
4	2	0.724850

From the table it can be observed that the greater the number of components used, the lower the silhouette value. In addition, the number of components that have the largest silhouette value is 1. However, by considering the explained variance of the components, the number of components 1 cannot be used because there will be a lot of missing information.

Therefore, it can be concluded that the optimum number of components of PCA is 2 which consists of principal components 1 and 2. The choice of the number of components is based on the large and significant amount of explained variance, as well as the results of clustering tests. Although there is still missing information about 37% because it does not include components 3 and 4, the amount of information contained in principal components 1 and 2 is still above 50%.

3.2. Number of K-Means Clusters

The optimum number of K-Means clusters is sought by measuring the clustering performance on different number of clusters. In this study, the number of clusters tested had a value range of 2 to 10. K-Means was tested with data that had reduced dimensions using PCA. The results of the K-Means clustering performance test for each number of clusters can be seen in Figure 2.

Figure 2 show us that the number of clusters = 4 has the largest silhouette coefficient value, which is 0.83501. In addition, there is no significant increase in the silhouette value after the number of clusters = 4. This means that 4 clusters are indeed the highest and optimum value of all the other clusters.

3.3 System Test Results

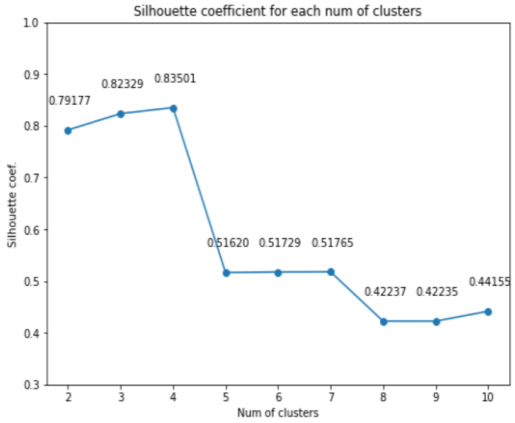

Figure 3. Silhouette coefficient for each num of clusters

Performance of the proposed system is tested with the optimum parameters that have been obtained from previous tests. The system performance is then compared with the comparison system with the same parameters. In this study, the comparison system used is a CF-based recommendation system that uses K-Means (KMCF) and also uses Hierachical Clustering (HCCF). The whole system is tested with the same number of neighbors in the CF algorithm, that is 10, 20, 30, and 40.

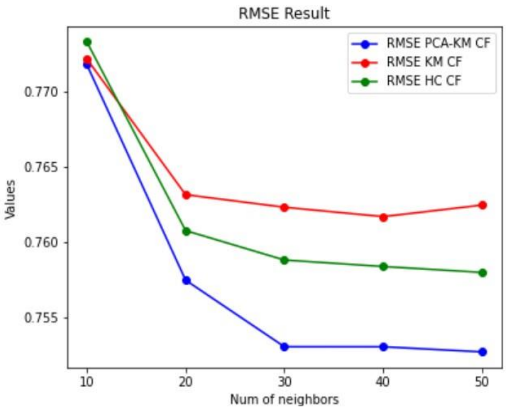

Figure 2. Comparison of RMSE values between methods for different numbers of neighbors

Figure 3 shows the comparison of the RMSE values for each method at different numbers of neighbors. The test results show that the proposed system (PCA-KM CF) has the lowest RMSE value for each number of neighbors compared to the value of the KMCF or HCCF system. PCA-KM CF obtained RMSE values for the five neighbors of 0.771806, 0.75747, 0.75304, 0.75304, and 0.75270, respectively. This explains that by considering the large error, the average error value of the proposed system is still lower than the KMCF and HCCF methods. This means that the proposed system has better accuracy in predicting user rating values.

DOI: https://doi.org/10.29207/resti.v6i1.3782

From the test results, it can be seen when the number of neighbors is too small system produces a large error. When the number of neighbors is 10, the RMSE value of the whole method is very high. However, as the number of neighbors increases, the RMSE value tends to decrease. This shows that a large number of neighbors can reduce errors in the prediction results. At the number of neighbors 30 or more, the RMSE value of PCA-KM CF tends to be stable. This shows that there is no significant change in the error in the number of neighbors of 30 or more.

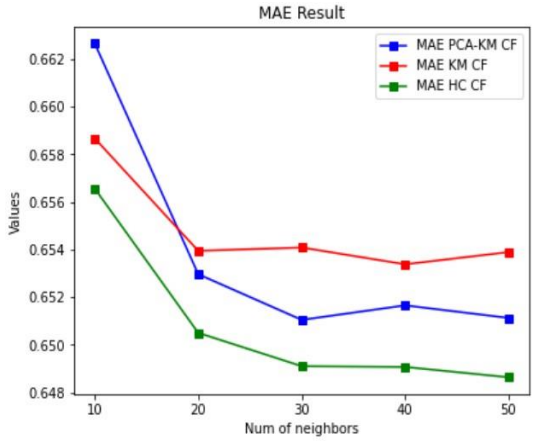

Figure 4. Comparison of MAE values between methods with different numbers of neighbors

Figure 4 shows the comparison of MAE values for each method. From the figure, it can be seen that the proposed system has an MAE value lower than KMCF and higher than HCCF. This means that PCA-KM CF has a smaller average error than the system using K-Means. However, the average error is still higher than the system using Hierarchical Clustering. PCA-KM CF obtained MAE values for the five neighbors of 0.66265, 0.65296, 0.65104, 0.65165, and 0.65113, respectively.

As shown from the RMSE results, the MAE value also tends to be high when the number of neighbors is small. The value starts to decrease as the number of neighbors gets bigger. The test results on the number of neighbors 10 show the MAE value of the PCA-KM CF system is very high compared to other methods. In addition, for the number of neighbors more than 30, the MAE value of the proposed system also does not change significantly.

4. Conclusion

In this paper, we have implemented a recommender system based on Collaborative Filtering using PCA and K-Means Clustering in the e-commerce field. System performance was evaluated using RMSE and MAE. The test results show that our proposed system is able to handle the sparsity problem well. Based on the results of the RMSE, the proposed system has the lowest value compared to other methods. For the number of neighbors of 10, 20, 30, and 40, our system obtains RMSE values of 0.771806, 0.75747, 0.75304, 0.75304, and 0.75270, respectively. Meanwhile, from the MAE results, our system obtains 0.66265, 0.65296, 0.65104, 0.65165, and 0.65113, respectively. The MAE value of our system is lower than the system with K-Means, but still high compared to the system using Hierarchical Clustering.

Based on the evaluation results from RMSE and MAE, the two methods showed almost similar results. However, RMSE is better at representing the performance of the recommender system because it considers a large error from the system's prediction results. Therefore, the proposed system allows it to be applied to e-commerce even though it does not achieve the most optimum value in MAE. So that in the future, this research can be developed again in order to achieve the optimum value. For example, by testing performance on other e-commerce data, or trying a combination of dimensions and other clustering methods that have not been used. Such as PCA with Hierarchical Clustering, Singular Value Decomposition (SVD) with K-Means, or a combination of PCA with K-Means and Hierarchical Clustering.

Reference

[1] S. Bandyopadhyay, S. S. Thakur, and J. K. Mandal, “Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: benefit for the society,” Innov. Syst. Softw. Eng., vol. 17, no. 1, pp. 45–52, 2021.

[2] Z. K. A. Baizal, D. H. Widyantoro, and N. U. Maulidevi, “Computational model for generating interactions in conversational recommender system based on product functional requirements,” Data Knowl. Eng., vol. 128, no. October 2018, p. 101813, 2020.

[3] J. Choi, S. Yun, and J. Kim, “Improvement of Data Sparsity and Scalability Problems in Collaborative Filtering Based Recommendation Systems,” 2020.

[4] M. I. Ardimansyah, A. F. Huda, and Z. K. A. Baizal, “Preprocessing matrix factorization for solving data sparsity on memory-based collaborative filtering,” Proceeding - 2017 3rd Int. Conf. Sci. Inf. Technol. Theory Appl. IT Educ. Ind. Soc. Big Data Era, ICSITech 2017, vol. 2018-Janua, pp. 521–525, 2017.

[5] B. Mulyawan, M. Viny Christanti, and R. Wenas, “Recommendation Product Based on Customer Categorization with K-Means Clustering Method,” IOP Conf. Ser. Mater. Sci. Eng., vol. 508, no. 1, pp. 0–6, 2019.

[6] R. Ananda, M. Zidny Naf’an, A. B. Arifa, and A. Burhanuddin, “Sistem Rekomendasi Pemilihan Peminatan Menggunakan Density Canopy K-Means,” RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 4, no. 1, pp. 172–179, 2020.

[7] and A. M. and others Yadav, Vikash, Rati Shukla, Aprna Tripathi, “A New Approach for Movie Recommender System using K-means Clustering and PCA,” J. Sci. Ind. Res., vol. 80, no. 2, pp. 159–165, 2021.

[8] C. Langensiepen, A. Cripps, and R. Cant, “Using PCA and K-Means to predict likeable songs from playlist information,” Proc. - 2018 UKSim-AMSS 20th Int. Conf. Model. Simulation, UKSim 2018, pp. 26–31, 2018.

[9] H. Zarzour, Z. Al-sharif, M. Al-ayyoub, and Y. Jararweh, “A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques,” 2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018, vol. 2018-Janua, pp. 102–106, 2018.

DOI: https://doi.org/10.29207/resti.v6i1.3782

[10] [M. Singh, "Scalability and sparsity issues in recommender datasets: a survey," Knowl. Inf. Syst., vol. 62, no. 1, pp. 1–43, 2020.
[11] W. Hong-Xia, An Improved Collaborative Filtering Recommendation Algorithm, vol. 1. Springer International Publishing, 2019.
[12] T. Li, P. Ye, and S. Zheng, "State grid office system user clustering analysis based on K-means algorithm," 2018 IEEE 3rd Int. Conf. Big Data Anal. ICBDA 2018, pp. 438–442, 2018.
[13] M. K. Gupta and P. Chandra, "An Empirical Evaluation of K-Means Clustering Algorithm Using Different Distance/Similarity Metrics," Lecture Notes in Electrical Engineering, vol. 605. pp. 884–892, 2020.
[14] P. K. Singh, P. K. Dutta Pramanik, A. K. Dey, and P. Choudhury, "Recommender systems: An overview, research trends, and future directions," Int. J. Bus. Syst. Res., vol. 15, no. 1, pp. 14–52, 2021.
[15] H. Zarzour, Y. Jararweh, and Z. A. Al-Sharif, "An Effective Model-Based Trust Collaborative Filtering for Explainable Recommendations," 2020 11th Int. Conf. Inf. Commun. Syst. ICICS 2020, pp. 238–242, 2020.
[16] M. Rahul, P. Pal, V. Yadav, D. K. Dellwar, and S. Singh, "Impact of similarity measures in K-means clustering method used in movie recommender systems," IOP Conf. Ser. Mater. Sci. Eng., vol. 1022, no. 1, 2021.
[17] P. K. Singh, P. K. D. Pramanik, and P. Choudhury, "Collaborative Filtering in Recommender Systems: Technicalities, Challenges, Applications, and Research Trends," New Age Anal., pp. 183–215, 2020.
[18] P. Kumar and R. S. Thakur, "Recommendation system techniques and related issues: a survey," Int. J. Inf. Technol., vol. 10, no. 4, pp. 495–501, 2018.
[19] M. Chen and P. Liu, "Performance evaluation of recommender systems," Int. J. Performability Eng., vol. 13, no. 8, pp. 1246–1256, 2017.

DOI: https://doi.org/10.29207/resti.v6i1.3782