Analysis of Music Recommendation System using Machine Learning Algorithms

Suja Panickar
MIT-WPU, Pune, India
Pune, India.
suja,panickar@mitwpu.edu.in

Himanshu Chaudhari
MIT-WPU, Pune, India
Pune, India.
himanshuchaudhari2346@gmail.com

Prashant Bansude
MIT-WPU, Pune, India
Pune, India
1032200665@mitwpu.edu.in

Abhishek Anand
MIT-WPU, Pune, India
1032192154@mitwpu.edu.in

Abstract—
For modern people who have always had cellphones in their pockets, music listening is a particularly personal and situational activity. As a result, contextual data such as the user's current song's acousticness, energy, danceability, popularity, instrumentality, loudness, and year aspects are employed to dramatically improve music recommendations.
Recommendation systems are algorithms that offer precise recommendations based on the preferences or needs of the user. They're an often-overlooked part of our daily lives, impacting how we watch movies, listen to music, and shop online. Given the variety of applications, there are many options and approaches to consider while designing such systems.

In this paper, We provide a personalised music suggestion system based on the songs that the user is currently listening to. We describe a collaborative filtering and content filtering recommendation algorithm that combines the output of the network while the user listens to the current song, and predicts the next 15 comparable songs for the user based on the songs he chooses.

Keywords—

Clustering Algorithm, Machine Learning, Decision Tree & Random Forest, Linear Regression, SVM, Kmeans, Kmodes, Recommendation System, PCA..

I. Introduction

Music listening is a particularly intimate and situational activity for modern people who have always had telephones in their pockets. As a consequence, contextual data such as the acousticness, energy, danceability, popularity, instrumentality, loudness, and year characteristics of the user's current song are used to significantly enhance music suggestions.
A music recommender system is one that learns from a user's prior listening experiences and selects tracks they may like hearing in the future. We experimented with a variety of different algorithms to see what we might come up with. a strong recommender system. We began with a simple and intuitive popularity-based model. Additionally, collaborative filtering algorithms are developed, which forecast (filter) a user's preferences and likes based on the preferences and tastes of several other users. Algorithms that provide exact suggestions based on the user's preferences or requirements are referred to as recommendation systems. They are a frequently neglected aspect of our everyday lives, having an effect on how we watch movies, listen to music, and purchase online. Given the breadth of applications, there are several alternatives and techniques to take into account while creating such systems.
In this paper, we recommended that the dataset be cleaned up using pre-processing techniques, and then that PCA techniques be used to reduce the dimensions. Following the preparation and standardization of the data, we collected some sample data and applied the K-means Algorithm to the data in question. This is how we determine the optimal Cluster value (K). following the discovery of the k value and application of the k-means algorithm.

II. Literature Survey

Our recommendation system's methodologies and procedures are extensively employed in a variety of fields, including music, movies, news, and e-commerce. Such algorithms are also used by companies like Facebook, Twitter, and LinkedIn to propose friends/followers/connections [5]. As a result, there is a substantial quantity of literature on the issue. Netflix famously challenged competitors to create an algorithm that would recommend movies to consumers based on their watching behaviour. The winning proposal included a variety of techniques, but one of the most successful algorithms was based on Latent Factor Models, which we will look at more in this project. Amazon, for example, recommends products to consumers based on both item-item and user-user correlations [6]. We'll use cosine similarity to try to replicate this method. Traditional methodologies such as collaborative filtering (CF) are employed in the music recommendation area to find the relationships between users and music items based on the rating a user gives for each music item, as noted by Ja-Hwung Su. [2]. (A musical piece, such as a song, is referred to as a music item in this article.) CF-like recommenders come in a variety of shapes and sizes.

in [1] This study provides a hybrid music recommendation system that rates musical works while effectively retaining collaborative and content-based data, such as user rating scores and audio signal acoustic properties.

We employed a three-way aspect model, a probabilistic generative model, to build our system. By incorporating a collection of latent variables that conceptually correlate to genres, this model may theoretically explain the generating mechanism for both types of observed data.

Alexandros Nanopoulos [3], Alexandros In music information retrieval, social tagging is becoming increasingly common (MIR). This paper allows users to categorise musical things such as songs, albums, and artists. MIR benefits from social tags since they provide a multifaceted source of information on genre, style, mood, user opinion, and instrumentation. They suggest using three-order tensors to represent social tagging data, which capture cubic (three-way) relationships between users, tags, and music items. The Higher Order Singular Value Decomposition (HOSVD) is used to find latent structure in this model, which aids in providing accurate and customised suggestions, i.e., recommendations tailored to the interests of specific users.

Deger Ayata [4], offers an emotion-based music recommendation system that learns a user's emotion from data acquired via wearable physiological sensors, as stated in the article. A wearable computer device containing galvanic skin response (GSR) and photo plethysmography (PPG) physiological sensors, in particular, is used to classify a user's emotion. As extra data, this emotion data is sent to any collaborative or content-based recommendation engine. As a result, these data may be used to improve the performance of existing recommendation engines.

II. Proposed System Architecture

The suggested system for identifying Current Songs Details and subsequently giving music suggestions is depicted in Figure 1.

Individuals often keep their favourite music files on their mobile devices in order to listen to them anytime they want, or they utilise the favourite playlist option to add their favourite songs. However, this music does not generate new song recommendations. The tempo (pace) of music is one of the most critical characteristics of a song. If the tempo of a song is properly identified, content-based music browsing may make use of this feature to search for and propose songs that fall into the same tempo category. To optimise music classification performance based on user preferences, the song classifier predicts the next 15 songs for users based on their energy, acousticness, danceability, instrumentality, valence, popularity, tempo, liveliness, loudness, speechiness, and year. The categorization results from all of these characteristics, as well as each user's listening history, are then aggregated and applied to music files.

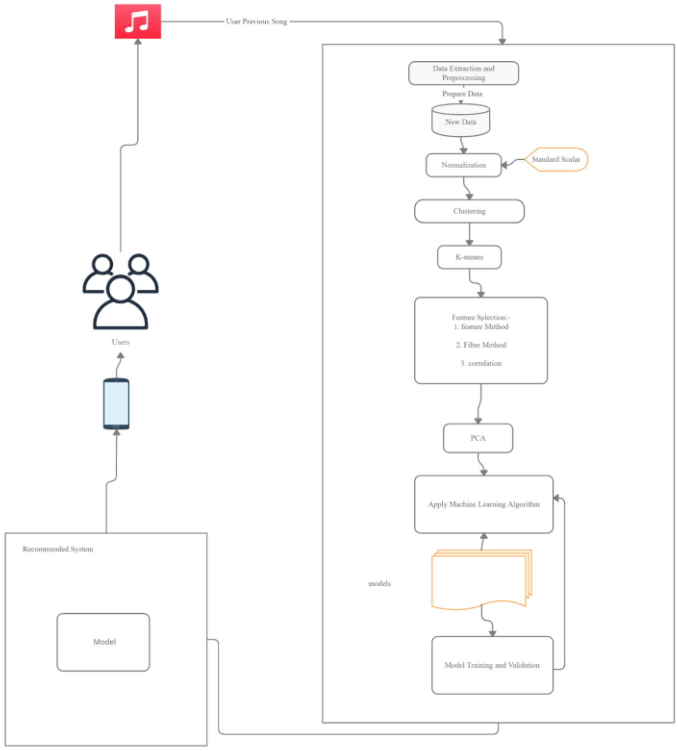

Figure 1. The above figure depicts all the processes.

III. Data Description

The dataset contains all of the music's data, from classic to contemporary. Additionally, data is available for many languages such as English, Hindi, Marathi, and Punjabi. There are 174389 rows and 19 columns in this dataset. The data set contains energy (Ranges from 0 to 1), acousticness (Ranges from 0 to 1), danceability (Ranges from 0 to 1), instrumentality (Ranges from 0 to 1), valence (Ranges from 0 to 100), tempo, liveliness (Ranges from 0 to 1), loudness (Float typically ranging from -60 to 0), speechiness, year, artists, release date, name, explicit [19].

IV. Preprocessing of DataFrame

On a dataset of 174384 music records, we performed the pre-processing tools. We identified all missing values and outliers in the dataset, as well as all the data's metadata. next we remove any superfluous columns from the collection, such as song IDs and artist IDs. then create some graphs, such as a distribution plot, a bar graph, or a line graph, to get insight into the data. Thus, during pre-processing and visualisation, we place a premium on energy, acousticity, danceability, instrumentality, valence, popularity, pace, liveliness, loudness, speechiness, and year in order to achieve the best result possible [19].

A. *Visualization on data*

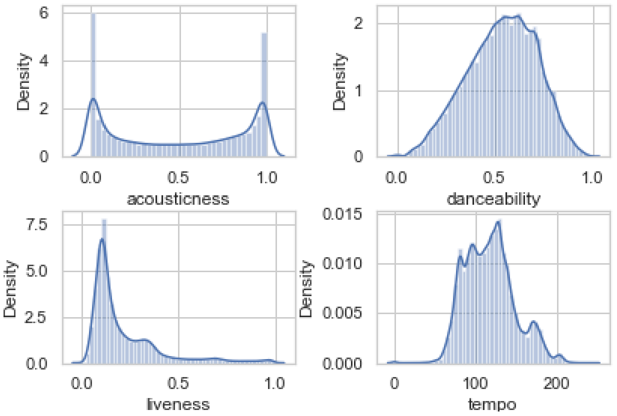

Figure 2. The following graph is show amount acousticness, danceability, liveness, tempo present in song

The above image, Figure 2 show that amount of variance in songs according to density. So if density is high then liveness, acousticness around 2 and liveness is 7. So according to these attributes we can cluster the data.

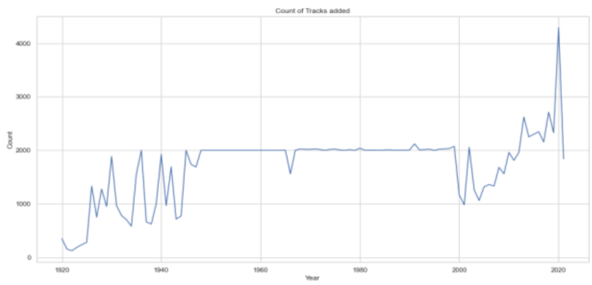

Figure 3. This graph illustrates the number of songs release per year

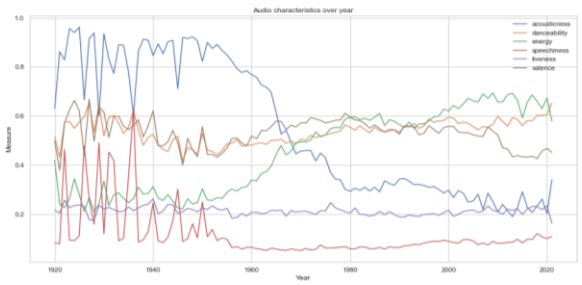

Figure 4. This specific statistic displays the audio characteristics over year.

From above figure 3, graph showing counts of songs release per year 1955 to 1990 seem like constant song release. From year 2018 increase count of songs per year. Another figure, figure 4, illustrates audio characteristics over year.

B. *Pre-processing*

After pre-processing, this dataset has no null values. Then we apply data normalisation methods. Normalization and standardisation are the two most often used approaches for scaling numerical data prior to modelling. Normalization adjusts each input variable independently to the range 0-1, which is the most precise range for floating-point data [20]. Standardization scales each input variable independently by subtracting the mean and dividing by the standard deviation, therefore shifting the distribution to have a mean of zero and a standard deviation of one.

Standardization is predicated on the assumption that your data conform to a Gaussian distribution (bell curve) with a well-behaved mean and standard deviation. If this expectation is not realised, you can still normalise your data, but you may not obtain trustworthy findings.
The scaler is defined, fitted to the entire dataset, and then used to transform the dataset into a transformed version with each column normalised individually. As can be seen, the mean value in each column is set to 0.0 if it exists, and the values are centred on 0.0 for both positive and negative values.

A value is standardized as follows:

y = (x – mean) / standard_deviation (1)

Where the mean is calculated as:

mean = sum(x) / count(x) (2)

And the standard_deviation is calculated as:

standard_deviation = sqrt(sum((x – mean)^2) / count(x)) (3)

V. Clustering

Clustering, or cluster analysis, is a task that requires unsupervised learning. It is frequently used in data analysis to uncover intriguing patterns in data, such as consumer groupings based on their behaviour. There are several clustering methods to select from, and no single technique is optimal in all circumstances. Rather than that, it is prudent to investigate a variety of clustering methods and their various configurations. Numerous algorithms search for dense regions of observations by comparing or separating examples in the feature space. As such, scaling data prior to employing clustering methods is frequently a recommended strategy. Affinity Propagation,Agglomerative Clustering, BIRCH, DBSCAN, K-Means, Mean Shift, Mini-Batch, OPTICS,Gaussian Spectral Clustering Mixture
Each method takes a unique approach to the problem of identifying natural groupings within data [21].

We attempt to locate the best clusters in the dataset using the kmeans and kmodes algorithms [9]. We used these two strategies to cluster our data. We experimented with cluster sizes ranging from two to sixteen and used kmeans and kmodes, as well as the elbow and silhouette methods, to determine the optimal k value [12], [16], [18].

A. *Elbow Method and silhouette Method*

We'll use a clustering algorithm to find K clusters. The three clusters created with the silhouette approach are displayed below. We reduced the dimensions based on these findings, then gave domains and colors to identify them based on the cluster values.

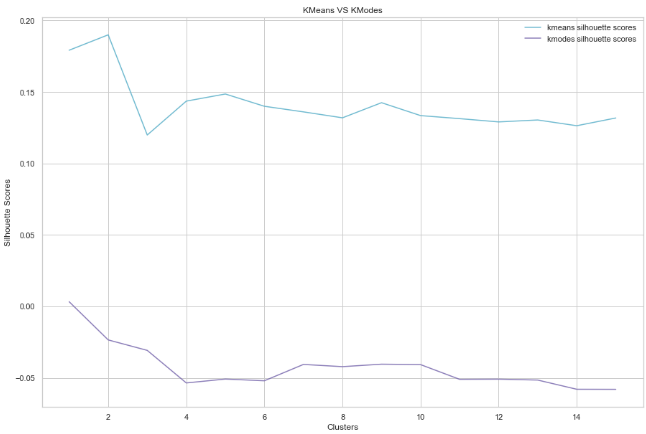

Figure 5. kmeans vs kModes

Set K centroids to different values. As we add more clusters, the total variation between each group diminishes. As a result, these data do not indicate the optimal number of K clusters to seek. We must reduce the within-cluster sum of squares to get the right K number of centroids for our prediction analysis [18]. To accomplish so, we first used the Elbow approach to figure out how many clusters were optimum [13], [15].

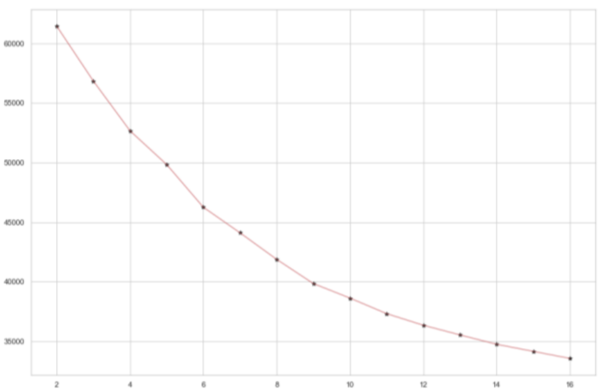

Figure 6. shows the effect of altering the number of clusters using the Elbow approach, as shown in the graph and illustration below. When calculating the total number of squares inside groups, two should appear at the elbow's end.

When you look at K equals 4, you'll see that it looks like an elbow. When the number of clusters is added to the within-group mean square, the effect is significantly worse [13], [14]. The ideal number of cluster centers in this case is two. However, the narrative is susceptible to interpretation, and the fact that we aren't sure because the other point is at 3, undermines our argument even more. We employed the second option, known as the 'Silhouette technique,' to verify that the right number of k clusters were produced [14]. The ideal number of clusters for the Silhouette approach is shown in Figure 7.

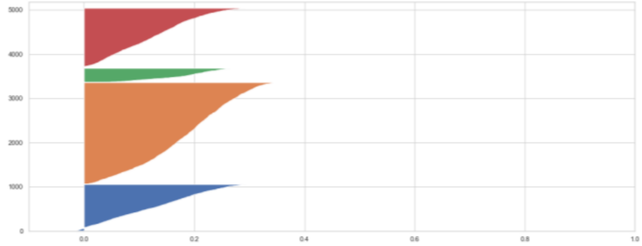

Figure 7. silhouette Score

The silhouette approach may be used to investigate and confirm consistency among data clusters. It is calculated by a comparison of an entity's proximity to its own (cohesion) cluster vs its distance from all other (separation) clusters. A high score suggests that the item is well-connected to its cluster [12], [13], whereas a low score shows that it is poorly matched to neighboring clusters. If the majority of the items have a high value, clustering is appropriate. There may be an oversupply or shortage of clusters in the clustering arrangement if there are a substantial number of points with a negative or insignificant value. In the picture below, notice that cluster number 4 has a negative value. When we look at the other side, we can see that each point is brilliant, uniform, and unbroken in groups. Our prediction process is based on three clusters, one of which is inferred to be the most optimal, as shown in Figure 7.

V. Machine Learning Algorithm

When k = 4, after building clusters with kmeans, we add the target value to the instance that was generated in the first place [9]. After that, we run the data through a variety of machine learning algorithms and create a categorization report. As a result, we experiment with algorithms such as linear learning, logistic learning, SVM, Decision Tree, and Random Forest. After obtaining all of the accuracy of each model, we compare them and choose the method with the highest accuracy.

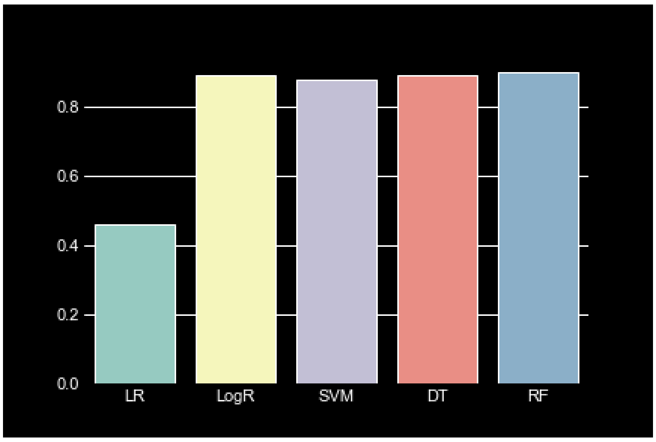

Figure 8. comparison of different algorithm

We develop a recommendation function that takes the parameters spotify link, suggestion length, data1, data5, and other parameters. We obtain all audio properties from spotify link, which we then feed to the cluster algorithm in order to obtain the suggestion length instance. and it will provide a list of audios.

V. Results

We received the greatest results possible for our music recommendation algorithm. We attempted to cluster the data based on auditory qualities such as tempo, density, energy, liveness, and so on, and then we searched for missing values in the data. Visualize the data to get insight into the information included within these qualities. We attempted a variety of approaches and graphs, including bar graphs, lineplot graphs, piechart graphs, and box plots, and for each of these, we used the Seaborn Python library to obtain inside information, and we were successful in our endeavors. Following the acquisition and comprehension of the data, we eliminate some columns that are superfluous and do not make any sense. after obtaining all of the final data that is comprised of high-quality information Following standard normalization of data in order to reduce greater values between 0 and 1, we apply different clustering algorithms to that dataset in order to determine which cluster contained which type of songs and how many audio characteristics values were present in each particular song in that particular cluster. as a result, we experimented with k-means and k-mediodes In order to determine which k value is optimal, we apply the elbow technique and silhouette score to obtain k values ranging from 2 to 16 permutations of the k value. When k=4, we get a better categorization of clusters.
The target value is subsequently set to the original data. After that, we used PCA methods to improve the accuracy of the machine learning algorithm we had developed.

Table 1. comparison of different algorithm

ML Algorithm		**Linear reg**	**Logistic reg**	**SVM**	**Decision tree**	**Random forest**
Acc		0.46	0.89	0.88	0.89	0.90
F1 - score	Class 0	0.49	0.97	0.86	0.97	0.98
	Class 1	0.65	0.91	0.88	0.88	0.90
	Class 2	0.43	0.86	0.55	0.84	0.86
	Class 3	0.32	0.92	0.91	0.92	0.93
Precision	Class 0	0.93	0.99	1.00	0.97	0.99
	Class 1	0.55	0.92	0.91	0.89	0.91
	Class 2	0.32	0.87	0.88	0.84	0.88
	Class 3	0.99	0.91	0.86	0.91	0.91
recall	Class 0	0.34	0.96	0.75	0.96	0.97
	Class 1	0.80	0.90	0.86	0.87	0.89
	Class 2	0.62	0.85	0.82	0.83	0.84
	Class 3	0.19	0.93	0.95	0.93	0.95

In order to compute accuracy and f1 score and to illustrate the classification report, we experimented with many machine learning algorithms such as linear regression, logistic regression, random forest, decision trees, SVM and other similar ones. Finally, random forest outperforms all of the other algorithms when it comes to accuracy and precision. After that, we use the random function to process the data.

In the end, we build a function called recommendation functions, which takes three parameters: spotify link, the number of recommendations you require, and the data you entered earlier. As a result, we first obtain through the connection all of the audio characteristics of that particular participant from the dataset. Then we'll figure out where these songs go in relation to one other. in addition to returning the specified number of recommendations from that cluster

VI. Conclusion

Its usefulness for discovering similar music has been put to the test, and it has met the project's aims. It has been demonstrated that a large information system is not required to present the user with associated music that suits his tastes. Utilizing the available clustering technique, which provides comprehensive music collections for each cluster. However, it was discovered that developing a recommendation system with commercial features necessitates the creation of a large relational database to hold the previously catalogued music. Making connections between artists, energy, musical genres, and epochs in a strong relational database of music might considerably enhance the capabilities of a recommender system to assist people find relevant music.
Recommender System for Music There are several methods to this problem, and we learn about some of them in depth, particularly the four models that we attempted to employ in the study following clustering. We had a lot of trouble dealing with this massive dataset, figuring out how to properly examine it, and working out certain technical specifics. However, with a lot of effort, we were able to overcome all of these obstacles.
The cooperation is the most enjoyable aspect of this endeavors. This endeavors has taught us a great deal. In terms of research, we still have a lot of work to do to improve our studies. We can take some efforts and conduct a lot more testing in the future because Music Recommender System is such a broad, open, and challenging topic. We also learned that developing a recommender system is not an easy undertaking. The fact that it is a huge dataset makes it challenging in many ways.

References

[1] Yoshii, K.; Goto, M.; Komatani, K.; Ogata, T.; Okuno, H.G. (2008). An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model. , 16(2), 435–447. doi:10.1109/tasl.2007.911503

[2] J. -H. Su, H. -H. Yeh, P. S. Yu and V. S. Tseng, "Music Recommendation Using Content and Context Information Mining," in IEEE Intelligent Systems, vol. 25, no. 1, pp. 16-26, Jan.-Feb. 2010, doi: 10.1109/MIS.2010.23.

[3] A. Nanopoulos, D. Rafailidis, P. Symeonidis and Y. Manolopoulos, "MusicBox: Personalized Music Recommendation Based on Cubic Analysis of Social Tags," in IEEE Transactions on Audio, Speech,

and Language Processing, vol. 18, no. 2, pp. 407-412, Feb. 2010, doi: 10.1109/TASL.2009.2033973.

[4] D. Ayata, Y. Yaslan and M. E. Kamasak, "Emotion Based Music Recommendation System Using Wearable Physiological Sensors," in IEEE Transactions on Consumer Electronics, vol. 64, no. 2, pp. 196-203, May 2018, doi: 10.1109/TCE.2018.2844736.

[5] Ricci, Francesco; Rokach, Lior; Shapira, Bracha; Kantor, Paul B. (2011). *Recommender Systems Handbook || . , 10.1007/978-0-387-85820-3(), –.* doi:10.1007/978-0-387-85820-3

[6] Schafer, J. Ben; Konstan, Joseph; Riedi, John (1999). [ACM Press the 1st ACM conference - Denver, Colorado, United States (1999.11.03-1999.11.05)] Proceedings of the 1st ACM conference on Electronic commerce - EC '99 - Recommender systems in e-commerce. , (0), 158–166. doi:10.1145/336992.337035

[7] Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014.

[8] A. M. Martinez and A. C. Kak, "PCA versus LDA," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, Feb. 2001.

[9] Aristidis Likas, Nikos Vlassis, Jakob J. Verbeek,"The global k-means clustering algorithm",Pattern Recognition,Volume 36, Issue 2,2003,Pages 451-461,ISSN 0031-3203.

[10] Wagstaff, Kiri et al. "Constrained K-Means Clustering with Background Knowledge." Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc.

[11] Anil K. Jain,"Data clustering: 50 years beyond K-means",Pattern Recognition Letters, Volume 31, Issue 8,2010.

[12] S. Aranganayagi and K. Thangavel, "Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure," International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), 2007.

[13] Thinsungnoen, Tippaya & Kaoungku, Nuntawut & Durongdumronchai, Pongsakorn & Kerdprasop, Kittisak & Kerdprasop, Nittaya. (2015). "The Clustering Validity with Silhouette and Sum of Squared Errors".

[14] Kodinariya, T. and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." (2013).

[15] Purnima Bholowalia and Arvind Kumar. Article: "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN". International Journal of Computer Applications 105(9):17-24, November 2014.

[16] Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. J 2019.

[17] Purnima Bholowalia and Arvind Kumar. Article: "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN". International Journal of Computer Applications 105(9):17-24, November 2014.

[18] Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. J 2019.

[19] McFee, Brian; Bertin-Mahieux, Thierry; Ellis, Daniel P.W.; Lanckriet, Gert R.G. (2012). *[ACM Press the 21st international conference companion - Lyon, France (2012.04.16-2012.04.20)] Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion - The million song dataset challenge. , (), 909–.* doi:10.1145/2187980.2188222

[20] Luo C., Zhan J., Xue X., Wang L., Ren R., Yang Q. (2018) Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks. In: Kůrková V., Manolopoulos Y., Hammer B., Iliadis L., Maglogiannis I. (eds) Artificial Neural Networks and Machine Learning – ICANN 2018. ICANN 2018. Lecture Notes in Computer Science, vol 11139. Springer, Cham. https://doi.org/10.1007/978-3-030-01418-6_38

[21] K. Bindra and A. Mishra, "A detailed study of clustering algorithms," 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2017, pp. 371-376, doi: 10.1109/ICRITO.2017.8342454.