Song Recommendation System

Namit Shah | Martand Javia | Suhanee Patel | Devarsh Patel

***Abstract*—Music is one of the most popular and important cultural aspects of the society. Over the time, numerous audio streaming platforms have been developed to satisfy our pursuit for music. Evidently, people have different musical taste and hence it is crucial to have a good recommendation engine as per one's preferences. Our idea is simple - to build a recommendation system which can recommend songs based on user's preferences by means of unsupervised learning mechanisms.**

***Index Terms*—spotify, unsupervised, clustering, recommendation**

I. Introduction

Spotify is one of the leading audio streaming platforms, hosting almost all song collections present in the world. We selected an independent dataset derived from Spotify Public API comprising of more than 160k songs [1], to design a machine learning model for the song recommendation system. The prediction is done based on different audio features applying unsupervised learning algorithms to classify the data into different clusters. The system will predict songs belonging to a particular set of clusters based on different features of the song. The dataset consists of various attributes and features of the songs, which helps in grouping them based on certain similar premises.

The rest of the letter is organized as follows. In Section II the existing body of works is explained. Section III focuses on the implementation process. The obtained results are presented in Section IV and Section V elucidates the conclusions of the work followed by References.

II. Literature Survey

Most of the platforms for streaming songs have their own recommendation systems. These recommendation systems analyze the user's preference by merely averaging the properties of music within the user's list. P. Cano presented "content-based system" where he extracted descriptions related to instrumentation, rhythm and harmony from music signal using similarity metrics [5]. G. Tzanetakis presented a way to extract feature from music, and divide musical genre automatically [4].

Another approach for recommendation includes collaborative filtering. Here, recommendations are made considering user preferences into account along with the attributes of the songs. Other models also include content based filtering which includes keyword searching, user actions and feedback. Though there are numerous methods for developing predictive models for songs, item based filtering is the most standard approach for creating an elementary recommendation system.

III. Implementation

A. Preprocessing Dataset

In order to derive valid inferences, it is important to make sure that the dataset does not have biased error terms. As a result, the selected dataset was initially pre-processed and cleaned to extract and remove all the indifferent values. Moreover, incomplete data values were removed from the dataset by employing different analysis functions.

After removing null and empty values from the dataset, correlation matrix of all the feature vectors was calculated. It turned out that, the pair-wise correlation of Energy, Acousticness, and Loudness was very high. It means that, both Energy and Loudness values can be predicted only using the Acousticness value. So, Energy and Loudness were removed from the feature space to lower the computational complexity of the model. Moreover, there were certain features like Duration, Album Name, etc. which were of no significant use in recommendation systems they did not contribute towards the audio properties of song were also removed from the dataset.

B. Observing trends in the audio features

Generally, every dataset observes some sort of trend or pattern to derive abstract intuitions for the analysis. After processing the dataset, we tried to visualise the trends and relationships between different features in the dataset. In this case, it helped finding patterns between different types of songs, and visualising the evolution of songs and music in last couple of decades. This process can also be used to study the change in people's song preference over the course of time.

C. Applying Unsupervised Learning Algorithm

To build a recommendation system, the dataset should be classified into several groups comprehending the relationship between the features. As the selected dataset does not contain any feature which can be considered as a target variable, unsupervised learning mechanisms can be used to model the problem. K-Means algorithm was implemented to cluster the dataset into different groups.

K-means basically divides the data into K different clusters taking into consideration various features and the nature of the dataset. Deciding the optimal value of K is very important considering the complexity of algorithm and size of the dataset. For this, elbow method was implemented which compares different values of K with their run-time performance to provide an optimal K-value. Using the k value obtained from the elbow method, K-Means algorithm was executed to perform clustering.

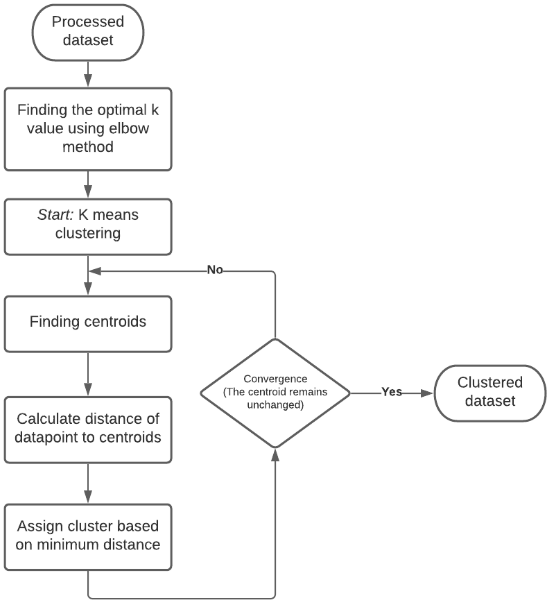

Fig. 1. K means algorithm

D. Optimizing K-Means algorithm

K-Means algorithm clusters the dataset in a brute force manner taking an average run time of order of $n \times k \times t$. Considering its high computational cost, K-means can be an inefficient approach to cluster the dataset. An effective method to optimize the complexity of K-means algorithm is proposed in article [7]. The idea is to reduce the number of computations for finding the nearest cluster centers when there is no significant shift in positions of the cluster centers. Using this approach, an optimized K-means algorithm was devised to cluster the dataset.

Algorithm 1 Optimized K means

Choosing k random data points as centroids
Initialize $clus$ vector of length = # datapoints
Initialize $dist$ vector of length = # datapoints
for each data point **do**
    Calculate euclidean distance to each centroid
    Assign cluster number based on min. distance
    Store cluster number in $clus[]$
    Store distance in $dist[]$
while convergence := False **do**
    Calculate new centroids based on the mean of the previous centroids
    Calculate the new distance from the centroid based on the cluster number
    if $dist_{new} < dist_{old}$ **then**
        Assign new cluster to the datapoint based on
        min. distance
        Update $clus[]$ and $dist[]$
return *clustered dataset*

E. Building Recommendation System

The primary function of recommendation system is to find songs similar to user's preference. Considering that analogy, the clustered dataset is used to differentiate the songs and find the records nearest to the input song(s), having similar feature values. The recommendation system created, can suggest number of similar songs based on user's input. But it is possible that the input list might contain more than 1 song records, and thus, the system should recommend songs similar to all the input records provided to it. To implement that, initially, songs are recommended individually for each input record, and then eliminated based on the similarity level with the input values.

The project uses 3 different recommendation systems, each one using a different clustering technique - Scikit learn K-means, Optimized custom K-means, and Fuzzy C-means technique (discussed in the next section).

F. Performance comparison

The custom optimized K-means algorithm used in this system has a smaller run-time complexity as compared to standard K-means algorithm. The results of the system is compared with other standard clustering algorithms to validate the performance of the model. Two clustering algorithms are used to compare the results with the proposed model - Standard K-means, Fuzzy C-means.

Fuzzy C-means [8] is a form of clustering in which each data point can belong to more than one cluster. Clustering or cluster analysis involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible.

To compare the similarity of the recommended songs with the input song(s), Cosine similarity measure is used. The cosine similarity index for all the approaches (Optimized K-means, K-means, Fuzzy C-means), having same input data is shown below.

$$similarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{1}$$

Moreover, distortion value for the clustered data of all the approaches have been calculated to compare the error performance. Here distortion is the mean squared distances of the data points from the clustered centers, where distance between data points is calculated using the Euclidean distance measure. If m is the number of datapoints in the dataset, distortion can be calculated as -

$$min_{c^{(1)},..,c^{(m)},\mu_1,..,\mu_K} \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_c^{(i)}|| \tag{2}$$

$c^{(i)}$ (Index of cluster $\{1, 2, 3, \ldots, K\}$ to which example $x^{(i)}$ is currently assigned), μ_k (Cluster centroid k) and $\mu_c^{(i)}$ (Cluster centroid of the cluster to which the example $x^{(i)}$ is assigned).

IV. Results

The processed data with selected feature vectors was clustered to build a recommendation system. The results obtained at various stages of model generation are shown in this section.

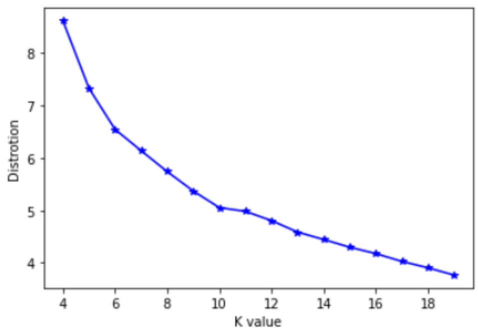

Fig. 2. Elbow Method: $distortion\ vs\ k$

Fig. 2 is the plot of distortion vs. k (number of clusters), where distortion here means the average of the squared distances (considering euclidean distance) from the data points to the mean of the respective clusters.

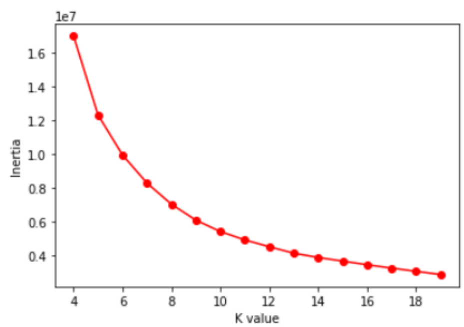

Fig. 3. Elbow Method: $inertia\ vs\ k$

Fig. 3 is the plot of inertia vs. k (number of clusters), where inertia is the sum of squared distances of samples to the mean of data points of the closest intra-cluster.

Table I and II shows the songs recommended by the recommendation system using different clustering techniques, for different user inputs. It can be seen that, all the techniques provides similar results with high cosine similarity value. Evidently, as all the techniques performs clustering individually, the cluster numbers for all of them are different.

TABLE I
Input Song : Perfect (Ed Sheeran)

	Cluster Number	S1	S2	Cosine Similarity
Custom K Means	8	Wasted	All through the night	0.99999825
Standard K Means	2	Wasted	All through the night	0.99999825
Fuzzy-C Means	4	Wasted	All through the night	0.99999825

TABLE II
Input Song : Lovely (Billie Eilish, Khalid)

	Cluster Number	S1	S2	Cosine Similarity
Custom K Means	7	The Nutcracker Suite, Op. 71a	The Christmas Song	0.9999992
Standard K Means	10	The Nutcracker Suite, Op. 71a	The Christmas Song	0.9999992
Fuzzy-C Means	3	The Nutcracker Suite, Op. 71a	The Christmas Song	0.9999992

Table III shows the songs recommended by the recommendation system using custom clustering technique, for multiple songs provided by the user as song inputs which are quite similar from each other.

TABLE III
Multiple Songs from same clusters

Input	Often, So beautiful
Output	Un Siglo Sin Ti, Brand New Year 2021

Table IV shows the songs recommended by the recommendation system using custom clustering technique, for multiple songs provided by the user as song inputs which are quite different from each other.

TABLE IV
Multiple Songs from different clusters

Input	7Rings, Perfect
Output	idfc - Tarro Remix, Lemonade, Wasted, All through the Night

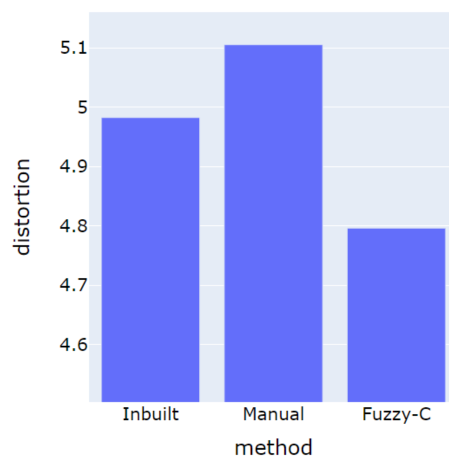

Fig. 4. Distortion for different clustering methods

Fig. 4 is the plot of distortion values for different clustering algorithms used in this project. From the graph, it can be seen that avg intracluster distortion using fuzzy C-means method is lowest of all. Moreover, the avg intracluster distortion using manual(optimized) K-means technique is slightly higher than the distortion obtained using standard K-means method.

V. Conclusion

Using classical unsupervised learning algorithms, the dataset can be clustered uniformly to differentiate various types of songs. Visualizing the dataset showed that the features are weakly correlated and hence, a large number of clusters are required to differentiate the data. These clusters are then used to recommend the songs based on the input feature space for single/multiple songs and to provide similar songs as results. Applying three different clustering approaches i.e. Custom K-means, Manual K-means, and Fuzzy C-means it was observed that the results which were obtained were quite similar and the recommendations provided by each algorithm had high cosine similarity. The distortion for each algorithm varies slightly where Fuzzy-C outperforms K-Means. The results obtained can be improved in terms of accuracy, for a vertically larger dataset, and by using more efficient algorithms.

References

[1] Y. E. Ay, "Spotify Dataset 1921-2020, 160k+ Tracks," Kaggle. [Online]. Available: https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks.

[2] Need, T., 2021. K Means Clustering — K Means Clustering Algorithm in Python. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering

[3] Chen, H. and Chen, A., 2001. A music recommendation system based on music data grouping and user interests. Proceedings of the tenth international conference on Information and knowledge management - CIKM'01.

[4] Tzanetakis, G. and Cook, P., 2002. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), pp.293-302.

[5] Cano, P., Koppenberger, M. and Wack, N., 2005. Content-based music audio recommendation. Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05.

[6] "Web API," Spotify for Developers. [Online]. Available: https://developer.spotify.com/documentation/web-api/.

[7] S. Na, L. Xumin, and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010.

[8] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," Computers amp; Geosciences, vol. 10, no. 2-3, pp. 191–203, 1984.