Music Recommendation System

Akshay A S[1], Madhura M[2], B Mounika[3], H S Rohit[4], Vikranth B M[5]

[1-4]*U.G Scholar, Department of Computer Science and Engineering, BMS College of Engineering*
[5]*Assistant Professor, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, India*

***Abstract*—Music recommendation system is a recommendation engine which recommends songs to the user based on his interest. Music can raise someone's mood, get them excited, or make them calm. There are many types of music to which one can listen. World of so music is so big that a person can't explore all the songs he'd like and listen to them. So, we build a model which assists a person in identifying songs he might like. These notes, which are the pitch and duration of a sound, identify music. There are unique features of songs which are determined by these notes. We can group the songs which are similar based on these features. In our project we use "million song datasets" and extract features of each song and create a dataset of features of songs. This dataset helps us to create a model. This model is built on clustering concept which forms the clusters of songs which have similar features. Whenever user listens to a song S, the song which has similar features to S is recommended to him. The history of the songs that user has listened to is also maintained according to the genre and a playlist of songs of his interest is created.**

In this way our system helps user listen to songs which he will appreciate

***Keywords*— Feature Extraction, k-means clustering, music recommendation, unsupervised learning, content-based filtering.**

I. INTRODUCTION

These days recommendation systems are used in many fields such as item recommendation by amazon, show recommendation by Netflix, etc. Such recommendation systems are based mainly on collaborative filtering [1] , which involves recognizing similar users and combining their recommendation based on each user's preferences. The other method used in combination with collaborative filtering is content based filterirng [2] . Content based filtering involves recommendations based on the item's characteristics and user's preferences. Generally, the characteristics of the item would be its rating, popularity, usefulness, etc. These characteristics are explicit. They are defined by other users or rating system. For a song, these features may include its artist, year of release, rating, popularity, album, etc. which are features are associated with a song. In our proposed recommendation system, we aim to cluster songs based on implicit properties of a song that are perceived by the listener such as timbre, acoustics, etc. "A million songs dataset" trains the model. clustering of the songs are done using unsupervised learning. The algorithm used for clustering is k-means clustering [3] . The songs are aimed to be classified into to their respective genres based on these features. As mentioned before, the recommendation systems can be broadly classified into content-based filtering and collaborative filtering. The main advantage of content-based filtering over collaborative filtering is that collaborative filtering requires large amounts of data related to user's preferences in order to map similar users. On the other hand, content-based filtering can operate with little amount of information. Content based filtering requires two types of information- a model of user's preferences and a history of user's interaction with the recommender system.

A. Song features

Five features are selected among many available features to distinguish songs from one another. These features are not abstract features like danceability, popularity, etc. which are used to classify music. Features which relate to the sound wave in the songs are used so as to distinguish songs accurately. These five features are MFCC(Mel-frequency cepstral contrast) [4] , Mel-Spectogram, chroma stft [5] , spectral contrast [6] and tonnetz [7] . MFCC is the representation of sound made by humans in the power spectrum. Mel-Spectogram too is the representation of power spectrum, but the acoustic power spectrum. Chroma stft represents the 12 different pitch classes in music. Spectral contrast deals with strength of the peaks and valleys in the spectral wave. Tonnetz is space which represents the tonal space in the music. These 5 features could help us to classify the songs accurately, and help us to accurately provide the next song to the user.

B. Content-Based filtering

Content based filtering is an approach used to build recommender systems based on user's preferences and item's characteristics. Usually there are a set of features that identify an item, further recommendations are based on a cumulative value of the features present in the user's history. A simple way of calculating this cumulative value is taking average. There also exist other sophisticated methods such Bayesian classifier, decision trees, etc. To create a user's profile, the two important types of information required are a model of user's preference and history of user's interaction with the recommender system. A key issue with content- based filtering is learning user's preferences from one content source and transferring it to another content source. The system creates a profile for each user by assigning weights to different attributes or features of an item depending on user's history. This weight denotes the importance of each feature to the given user.

C. K-means Clustering

K-means algorithm is a widely used algorithm in clustering. The algorithm works iteratively by assigning a cluster to each data point. It forms K clusters of data points. Initially, it uses k random centroids for the dataset. The centroids need not be from data points. The distance between each data point and each centroid is calculated again. The data point would then belong to a cluster .Initially, the value of k is determined. Elbow method is used in determining the value of 'k'.

II. PREVIOUSLY PUBLISHED ARTICLES

A. Collaborative filtering

Collaborative filtering is the most popular method used to recommend the next song to the user. The method used here is to keep track of all users in the system. Users who listened to similar songs are taken together. For example, if person A has the same taste of music as person B, then the system would recommend songs to person A based on what person B listened to before, or vice versa [7].

The problem with this technique was that a lot a data was required of the user's preference, and many systems don't have the capacity to do this. The other problem with this technique is that it doesn't suit the model we are trying to build. The model we want is to accurately predict the song according the previous song the user has listened [4][5][6].

B. Features Used

The Features used in these works are usually abstract features which are not strong features to distinguish the songs. The features used in many of these works are popularity, danceability, a cousticity, etc. Due to this abstractness, we believe that these features do not recommend songs accurately. The features such as MFCC, Mel-Spectogram, Chroma STFT, Spectral contrast and tonnetz are useful features which actually help in distinguishing songs accurately.

III. IMPLEMENTATION

A. Disadvantages of Existing Methods

The existing method for recommendation usually uses collaborative filtering. This method requires a large dataset regarding user's preferences. Although many recommendation systems used a hybrid of collaborative and content-based filtering, the features considered for content- based filtering are explicit features such as popularity, rating, etc.. This way the model tends to be biased towards popular songs by popular artists.

B. Future Scope: Our Approach

The proposed system implements additional features where the next song is recommended based on a particular user's interest. The grouping of songs is done using k means clustering algorithm. Songs with similar features are grouped into one cluster. Recommendation of the next song is done from the same cluster of the current playing song. The history of the songs is also maintained separately. Playlists are also created for different types of songs. The main objectives of the project can be outlined as below:

1. To extract features from the songs and create a dataset which maps the songs to its features. The features are extracted using a python library called Librosa. The various features are – Mel-Spectrogram, MFCC, chroma-stft, spectral contrast, tonnetz.
2. Applying k-means clustering to the given dataset based on the features extracted in the previous step. The number of clusters 'k' is determined using elbow method. Based on that 'k' number of clusters are generated. The similarity between data points is calculated using Euclidean distance. This clusters the dataset into 'k' clusters
3. Based on the clusters generated in the previous step, a recommendation engine is built which performs two tasks. The first functionality of recommendation engine is recommending the next song to the user, based on the user's history and the clusters generated.
4. The second functionality of the recommendation engine creating playlists. Playlists are created using the songs from the cluster that the user has previously listened to. The number of songs in playlist generated for each cluster is proportional to the number of songs the user has listened to from that particular cluster.

C. Software Used

"Librosa" [8] python package is used to extract features from the songs. The features extracted are Mel-Spectrogram, MFCC, chorma-stft, spectral contrast and tonnetz.

"Pandas" [9] python package is used for data analysis. It performs the task of reading the dataset. This dataset has the details of all features extracted for each song.

"NumPy" [10] is a python package used for mathematical calculations. It will help us in calculations such as Euclidean distance between points.

"Matplotlib [11]" is a python package used for plotting graphs and analyzing various patterns in the dataset. "Sklearn" [12] has different packages such as cluster, regression, etc. We are using the cluster package in order to perform k- means clustering on the data points.

D. Project Breakdown

The overall project can be modularized into the following tasks that will help in obtaining the desired objective:

1) The first task will be to extract the features of all songs from the "million songs dataset" available on Kaggle. The features are extracted using the python package called "Librosa".
2) The second task is to perform k-means clustering on the above dataset. The value of 'k' is determined using the elbow method. Firstly, 'k' number of random centroids are selected. Each of these centroids determine a cluster. Every data point is grouped into a cluster depending on its Euclidean distance from each centroid. Once all points are classified, 'k' centroids are generated by taking the mean of all data points belonging to a cluster. This process is repeated over a few iterations. The process is complete when no point changes it's cluster
3) The third task is to build a recommendation engine based on the above clustering and the user's listening history. There are two parts to the recommendation engine built. The first part is to recommend the next song. This is achieved by selecting a song from that is very close to the song that user is currently listening to.
4) The fourth task is to build the second part of recommendation engine which creates playlists for the user. A playlist is created for each cluster that the user has listened to. The number of songs in the playlist in dynamic. It is calculated based on the number of songs from the given cluster that the user has listened to.

IV. CONCLUSION

This paper presents the project to implement a music recommendation system which recommends the next song based on preferences of the user and on songs history the user has listened by using a machine learning model. The model is built using k means clustering algorithm.

ACKNOWLEDGEMENT

The authors extend immense gratitude towards their professor B.M. Vikranth for supporting them throughout the study and providing them the necessary guidance whenever they needed. We are grateful to BMS College of Engineering for having provided us with the facilities needed

REFERENCES

[1]. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998, pp. 43– 52.
[2]. R. V. Meteren and M. V. Someren, "Using content-based filtering for recommendation," in Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, 2000, pp. 47–56.
[3]. J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100–108, 1979.
[4]. B. Logan et al., "Mel Frequency Cepstral Coefficients for Music Model- ing." in ISMIR, vol. 270, 2000, pp. 1–11.
[5]. G. Peeters, "Chroma-based estimation of musical key from audio-signal analysis." in ISMIR, 2006, pp. 115–120.
[6]. D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in Multimedia and Expo,2002. ICME'02. Proceedings. 2002 IEEE International Conference on, vol. 1, 2002, pp. 113–116.
[7]. D. Tymoczko, "The generalized tonnetz," Journal of Music Theory, pp. 1–52, 2012.
[8]. B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in Proceedings of the 14th python in science conference, 2015, pp. 18–25.
[9]. W. McKinney, "pandas: a foundational Python library for data analysis and statistics," Python for High Performance and Scientific Computing, pp. 1–9, 2011.
[10]. T. E. Oliphant, A guide to NumPy. Trelgol Publishing USA, 2006, vol. 1.
[11]. J. D. Hunter, "Matplotlib: A 2D graphics environment," Computing in science & engineering, vol. 9, no. 3, pp. 90–95, 2007.
[12]. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit- learn: Machine learning in Python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011.