International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 6, Issue 5 (SEPTEMBER-OCTOBER 2018), PP. 12-18

MUSIC RECOMMENDER SYSTEM

Hardik Jain[1], Jeevanjot Singh[2], Prashant Singh Rana[3],

***Abstract*— Music recommendation comes under category of Music Information Retrieval (MIR) which has been quite a topic of interest these days. Music is categorized by various features including rhythmic structures, member form and instrumentation. To determine the interest of the user is a big challenge for the MIR community. Through the means of this research paper we aim to present a music recommendation system, which provides personalized recommendations to each user. It is based on users past likes and listening history. The features are first extracted from the database of audio files which are in .au format. The proposed model is used to train these audio files and different clusters are formed accordingly allotting the songs in the database to predicted category. Now the user liked songs features are extracted and the built model predicts the recommendations for the user by matching the category allotted to this song, to that of other songs. An accuracy of close to 75% was achieved during the course of the project.**

***Index Terms*— MIR, MUSIC RECOMMENDER SYSTEM.**

I. INTRODUCTION

With the gradual technological advancements and rise of digital content distribution it is no surprise of the vast amount of music collections one has access to. Music libraries exceed 80 million songs which by far out limit the listening capacity of an individual. One can easily get overwhelmed by this gigantic number, making it an absolute necessity for the introduction of an efficient recommender system in the interest of both the user and service provider. This gives the service provider an upper edge by being able to maintain a high user count for providing the premium benefit of saving time, energy and pain of going through many different songs before finding the one matching their taste.

By the means of the paper we wish to propose a data driven/content-based approach to cluster music according to a person's previous likings/ ratings. The method does not involve any knowledge of the in- dividual genres or their total number in the dataset. Hence it is not guided by the subjective knowledge of the genres alone, but accounts audio features thereby giving a higher range of interests the user might be inclined to. Audio features include vocals, instru- mentation, bass, dance ability, loudness, acousticness, genre etc. which come into play when recommending tracks to the user. We aim to propose a model which clusters music on the described features and provides a better performance than existing machine learning models. This includes use of three different models and their weighted combination to provide the best results.

A subset of the GTZAN Audio Dataset has been taken and features have been extracted from all the audio files. Based on those features, machine learning models have been trained. It is validated that the combina- tional model performs better than the individual models. Various techniques like feature selection and cross validation have been applied to achieve maximum accuracy for user specific recommendation.

II. LITERATURE REVIEW

Most of the work in this field has occurred under the genre detection and classification topic. The most sig- nificant contribution in the field of genre classification has been given by the creators of the GTZan dataset

- Tzanetakis and Cook [1]. Till date, it is considered as the standard for audio genre classification. They used Gaussian Mixture Model (GMM) and achieved a highest accuracy of 61.0%.

Michael, Yang, and Kenny [2] investigated vari- ous machine learning algorithms including KNN, K Means, Multiclass SVM and Neural Networks for clas- sification of genres. However, they relied completely on Mel Frequency Cestral Coefficients to characterize genres.

Tao Li et al. [3] used SVM and LDA for content based music genre classification on the GTZan dataset and custom dataset constructed by the author. They achieved the best accuracy of 78.5%.

Bergstra et al. [4] used decision stumps as classifiers on MIREX 2005 dataset achieving an accuracy of 82.34%.

Pampalk et al. [5] achieved an accuracy of 82.3% on MIREX 2004 dataset using Neural Net and GMM as classifiers.

Carlos, Alessandro, and Celso [6] proposed an ensem- ble approach using a combination of various classical machine learning models on a Latin music dataset. They also included feature selection and conducted various experiments related to feature selection using genetic feature paradigm.

Tao Feng [7] used restricted Boltzmann Machine to build Deep Belief Neural Networks to perform a multiclass classification task of labeling music genres and compared it to that of vanilla neural networks. Arjun, Kamelia, Ali, and Raymond [8] used deep neural networks for the said classification and inferred that neural networks are comparable to classical mod- els when the data is represented in a rich feature space.

Miguel [9] used deep learning approach in music genre classification. He used mel spectrograms as input to the convolution neural networks. However, the results were not at par with the ones computed from the conventional methods.

III. METHODOLOGY

The workflow of the present work is shown in Fig. 1. The extracted feature vector is fed into the machine learning models. The combined models and the in- put song from the user together are used to output recommendations. The performance is computed ac- cordingly.

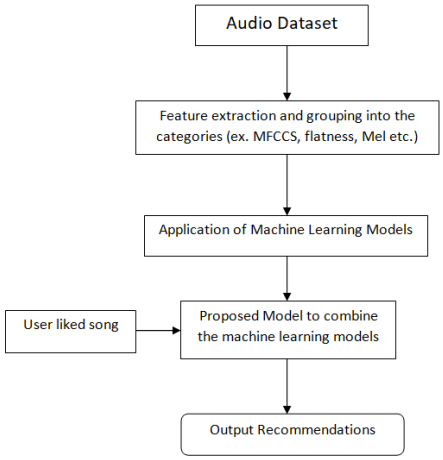


Fig. 1: Flowchart of Methodology

A. Dataset

The dataset used for the project is the GTZAN audio dataset. The dataset includes 10 music genres namely Blues, Classical, Country, Disco, Hip-Hop, Jazz, Metal, Pop, Reggae and Rock with 100 songs in each genre. The audio clips are 22050 Hz Mono 16 bit audio files in .au format.

B. Feature Extraction and Description

This is the first step to be accomplished before diving into the details of working of the system. This sec- tion describes the different features we included as a part of our research. The detailed number of features extracted are listed in the Table 2.

1) Mel-Frequency Cepstral Coefficients(MFCCS)

Mel-frequency cepstral coefficients (MFCCS) are coef- ficients that represent short term power spectrum of a sound based on a linear cosine transform of a log power spectrum. This feature group is a large part of the final feature vector (40). MFCCS is derived as follows

- The first step involves dividing the audio into several short frames. The aim of the step is to keep the audio signal constant.
- A periodogram estimate of the power spectrum is then calculated for each frame which represent the frequencies present in the short frames.
- Power spectra is then pushed into the mel filter bank and summing the collected energy in each filter.

$$M(f) = 1125\ln(1 + f/700) \quad (1)$$

- The logarithm of filter bank energies is evaluated.
- The Discrete Cosine Transform is calculated.
- Keep first 40 DCD features.

2) Spectral Flatness

Spectral flatness is used to characterize an audio spectrum. It provides a way to quantify how noise- like a sound is, opposed to being tone-like. A higher value (close to 1) indicates similar amount of power in all spectral bands thereby getting sound similar to white noise, giving graph a smooth and flat look. A low value (close to 0) indicates spectral power is concentrated in small number of bands giving a sound like mixture of sine waves and a spiky appearance to the graph.

3) Zero Crossing Rate

Zero crossing rate is defined as the rate at which signal swaps from positive to negative or vice versa. It is a key feature in classifying a percussive sound.

4) Spectral Bandwidth

It is the wavelength interval, wherein the radiated spectral quantity is not less than a specified fraction of the magnitude of the component having the maxi- mum value.

5) Chroma

Chroma relates to 12 different pitch classes when re- ferred to in a music context. These classes also known as pitch class profiles can play a very powerful role in analyzing music whose pitch can be meaningfully characterized. The chroma features capture melodic and harmonic characteristics of music and hence are an ideal set to be incorporated in our system.

TABLE 1: Feature Table

AUDIO NO.	F1	F2	F3	F4	F5	–	F192	F193	F194	F195	F196	LABEL
1	0.30847	0.703094	-0.472691	0.364664	-0.427321	–	0.364165	0.195952	-0.519954	0.109396	-0.645405	blues
2	-0.629159	0.780355	0.824518	-0.024904	0.33173	–	0.82919	1.374903	0.152211	0.971883	-0.841406	blues
.												
.						–						
101	-1.235011	1.153998	-1.159193	-0.641014	-0.968688	–	-1.744357	-0.609401	-1.381024	-0.752405	-0.44188	classical
102	-1.792668	1.087021	-0.743751	-0.704825	-0.884321	–	-1.270641	1.11642	-1.757755	-1.256539	-1.382437	classical
.												
.						–						
501	-1.074077	1.224611	-0.101969	-1.67668	0.15349	–	0.579038	0.58003	1.856817	0.712789	0.461453	jazz
502	-2.008884	1.514235	0.387308	-0.819594	0.569391	–	0.473699	-0.896019	-0.683298	-1.71043	1.450543	jazz
.												
.						–						
701	0.499061	-1.401829	1.761899	-0.554589	1.735851	–	2.166136	-0.278418	0.222247	1.48294	0.876384	pop
702	0.851785	-1.733222	2.11079	-1.236748	1.376335	–	1.18568	-0.162726	0.549592	0.013887	0.798537	pop
.												
.						–						
999	-0.803907	0.769354	-0.03708	1.219519	0.948107	–	1.655937	-0.124175	1.071157	1.046128	1.155128	rock
1000	-0.905267	0.776486	-0.628146	0.999458	-0.024546	–	-1.892227	1.449155	2.52681	0.423387	-1.873858	rock

TABLE 2: Components of Feature Vector

S. No.	Feature Group	Number of Features
1	MFCCS	40
2	Flatness	1
3	Zero Crossing rate	1
4	Specral Bandwidth	1
5	Chroma	12
6	Mel	128
7	Contrast	7
8	Tonnetz	6
	Total	**196**

6) Mel

Mel spectrogram is a time frequency representation of a sound. It is sampled into a number of points around equally spaced times t_i and frequency f_i on a Mel frequency scale.

$$Mel = 2595 * log(1 + f/700) \quad (2)$$

128 features were extracted from each audio file making it an integral part of the final feature vector.

7) Contrast

Contrast refers to small differences in speech sounds that are perceived by a listener helping him differentiate between different words (eg. pat and bat); the minimal difference of voicing in these words cause the listener to perceive them differently. Seven contrast features were extracted from each audio to make up the files feature vector.

8) Tonnetz

It is a representation of the tonal centroid features. Six features were extracted from each audio to make up the file feature vector.

C. Dataset Preprocessing

Before the data is fed to the model, pre processing is done to give more relevant and accurate results. The techniques of standardization and normalization were implemented aiming at better results. Later the accuracy was compared with the initial unprocessed data, and the results were found to be superior.

***D.** Dataset Description*

A sample of the extracted feature vector consisting of a total of 196 features is shown in Table 1.

IV. PROPOSED ENSEMBLE MODEL

The description of the project and steps involved include the following stages of operations and working.

Step I: The first step includes the selection of the dataset as mentioned earlier which contain

30 second clips. Different features as discussed in section 3.2 are now extracted from the loaded songs and a matrix of features is formed. The matrix contains different number of features from each parent categories which are defined above. This matrix is standardized using sklearns standard scaler for achieving improved results. The feature extraction has been discussed in the previous section.

Step II: Elbow method finds an optimum number of clusters for grouping the songs. K-means clustering is now applied to the matrix of features with the previously found number of clusters. The matrix of features was fed into two machine learning models namely Artificial neural Network and XG Boost. Hyperparameter tuning of these models was done individually for achieving best possible results on the test set. The Table 4 shows the models with their tuned parameters. Detailed Working in this step is shown in the Figure 2.

TABLE 3: Incorporated Packages

Package Installed	Uses
pandas	Reading and creating matrices
numpy	Array/Matrix manipulation
sklearn	Data Preprocessing, Hyperparameter tuning
keras	Creating and optimising neural networks
xgboost	Creating boosted tree model

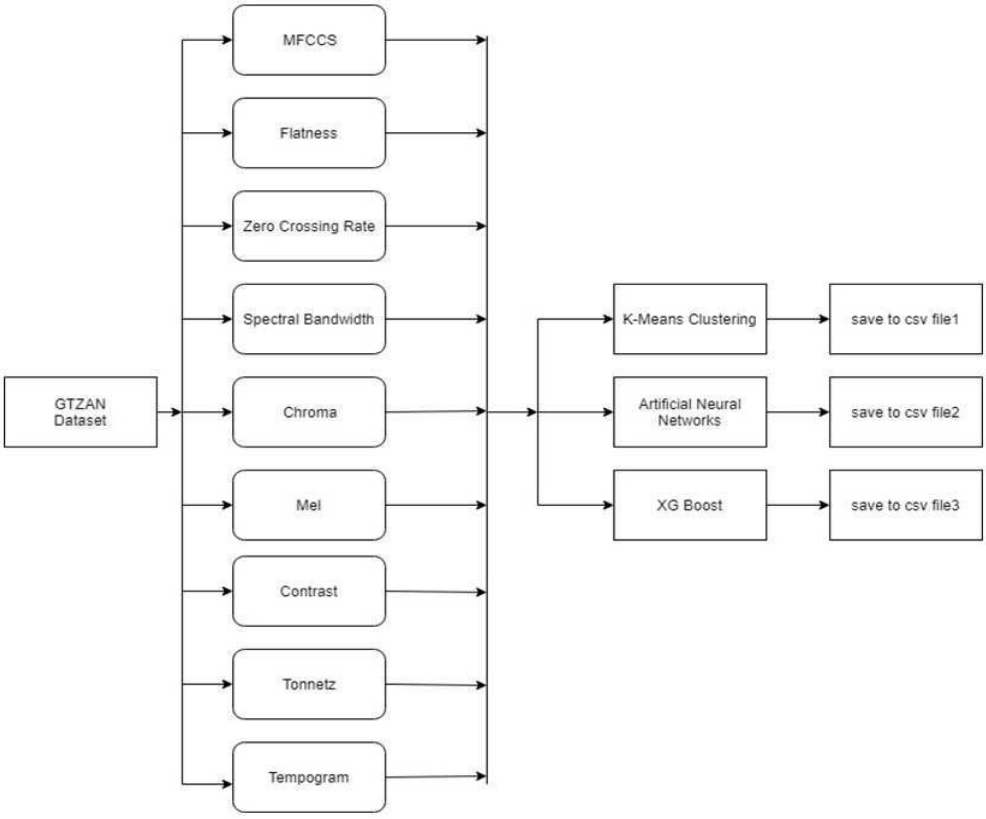


Fig. 2: Workflow of the Proposed Model (Step II)

Step III: Finally the three models are considered together to provide the user with a recommendation list with respect to the user liked song. This is done such that list outputs songs in an order where the first recommendations are the ones which are common to both the outputs of artificial neural network and Xgboost models. This aids in the total accuracy of the model as our purpose is not to just categorise the songs by genres alone and take into account all the features of the songs(like vocals, beats, bass, instruments etc.). Now the remaining songs from the two sets formed by ANN and Xgboost are matched with the relevant cluster formed through k-means clustering. Any matching songs are added to the recommendation list. Working of this step is depicted in 3.

Step IV: K-fold cross validation is applied to all the models individually to further check the consis- tency of the proposed model. The recommendation list formed in the above step is now output to the user.

V. Model Evaluation

Various parameters such as precision, recall and accuracy are calculated to evaluate the performance of the proposed ensemble model. The confusion matrix is formed for different models individually and is used to calculate the evaluation parameters. The confusion matrix for ANN is depicted in Figure 4(a)and for xgboost is depicted in 4(b). Repeated K-fold cross validation has been performed to test the robustness of the model

TABLE 4: Machine Learning Models

Model	Required Package	Tuning Parameters
Artificial Neural Network	Keras	Input dim=200,units=111, kernel initializer=uniform,activation=relu, Dropout(rate=0.1),output units=10,output activation=softmax Optimiser=adam,loss=sparse categorical crossentropy, Metrics=accuracy, batch size=16,epochs=35
xgboost	xgboost	Max depth=3, learning rate=0.1, n estimators=255,subsample=0.79

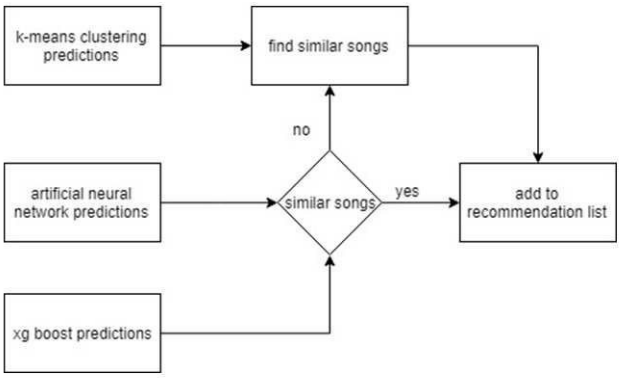


Fig. 3: Workflow of the Proposed Model (Step III)

A. Model Evaluation Parameters

Model evaluation parameters are calculated using the confusion matrix. The confusion matrix for the proposed model is given in Fig.4.

B. Precision

Precision is the fraction of relevant instances among the retrieved instances. Precision is computed as: Precision = TP/TP+FP

$$Precision = TP/TP + FP \quad (3)$$

1) Recall

Recall is the fraction of relevant instances that have been retrieved over the total number of relevant instances. Recall is computed as:

$$Recall = TP/TP + FN \quad (4)$$

2) F1-Score

F-1 Score is the harmonic average of precision and recall. F-1 Score is computed as:

$$F-1Score = 2 * Precision * Recall/Precision + Recall \quad (5)$$

3) Accuracy

Accuracy is the measure of correctness of the classifier. Accuracy is computed as:

$$Accuracy = (TP + TN)/TotalData \quad (6)$$

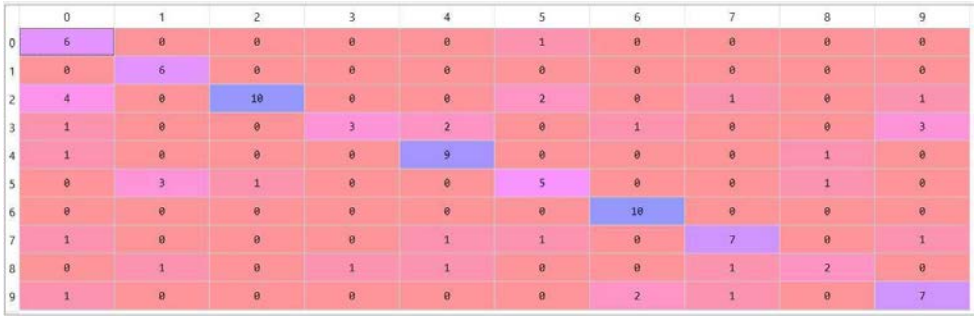
VI. Result Analysis, Comparison and Discussion

A problem of underfitting/overfitting may be encountered while model training. To make sure that such a problem does not arise, cross validation using an independent dataset must be performed. Overfitting occurs when model learns to much from the provided data and underfitting occurs when the learning is too less. By performing cross validation if the n executions of the model records highly fluctuating accuracies, we can conclude the model to be overfitting/underfitting. In the project, cross validation is used and it is seen that the model does not suffer from any inconsisten- cies. The machine learning models are trained with the tuning parameters mentioned in the above table. The dataset is divided into two parts with 80% dataset considered as training set, and the remaining 20% is the test set. The proposed model is the combi- nation of the three models mentioned. The model when tested on any unseen songs recommended very similar songs with an accuracy of 74.6%.

VII. Conclusion and Future Work

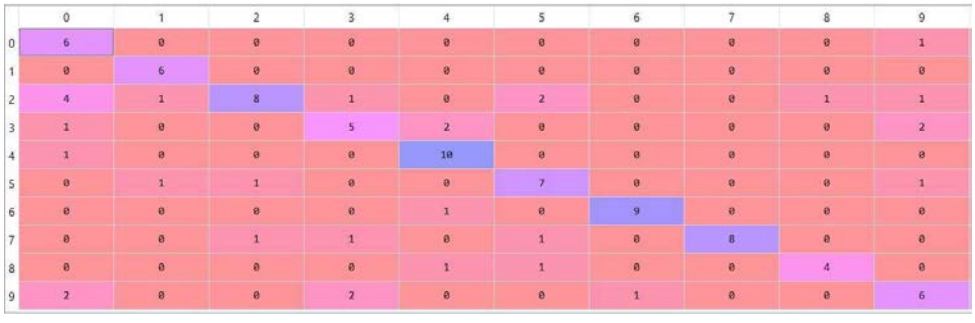
In today's world, Music Clustering finds numerous applications in content based searching and recommendation systems. A combinational model for Music Recommendation is proposed which is created by voting among Artificail Neural Network, Xg Boost and K-means Clustering which achieves an average accuracy of 74.6%.

In future we intend to study different learning architectures like Random Forset, Convolution Neural Networks and stacked autoencoders for incorporation in our system on a larger dataset and higher computational power providing more accurate results. Our proposition of songs would be more accurate if weighted average of many machine learning model's prediction is taken into account.



	0	1	2	3	4	5	6	7	8	9
0	6	0	0	0	0	1	0	0	0	0
1	0	6	0	0	0	0	0	0	0	0
2	4	0	10	0	0	2	0	1	0	1
3	1	0	0	3	2	0	1	0	0	3
4	1	0	0	0	9	0	0	0	1	0
5	0	3	1	0	0	5	0	0	1	0
6	0	0	0	0	0	0	10	0	0	0
7	1	0	0	0	1	1	0	7	0	1
8	0	1	0	1	1	0	0	1	2	0
9	1	0	0	0	0	0	2	1	0	7

Confusion Matrix of neural network model



	0	1	2	3	4	5	6	7	8	9
0	6	0	0	0	0	0	0	0	0	1
1	0	6	0	0	0	0	0	0	0	0
2	4	1	8	1	0	2	0	0	1	1
3	1	0	0	5	2	0	0	0	0	2
4	1	0	0	0	10	0	0	0	0	0
5	0	1	1	0	0	7	0	0	0	1
6	0	0	0	0	1	0	9	0	0	0
7	0	0	1	1	0	1	0	8	0	0
8	0	0	0	0	1	1	0	0	4	0
9	2	0	0	2	0	0	1	0	0	6

Confusion Matrix of xgboost model

Fig. 4: Here, in both (a) and (b) the x axis represents predicted value of the model, and the y axis represents actual model value. From the 1000 songs taken as dataset, we train the models on 800 random songs and the remaining 200 songs are used to test model prediction(on basis of genre alone).Confusion matrix represents random 100 songs from the test dataset. We see, Neural gives accuracy of 65%, and xgboost gives accuracy of 69%. When both models are combined an improved accuracy of 74.6% is achieved over a k cross validated set.

TABLE 5: Comparison with Existing Works on GTZan

Sno.	Author	Classifier	Number of Genres	Best Accuracy
1	Tzanetakis et al. [1]	Gaussian Mixture Model	10	61%
2	Michael et al. [2]	Neural Networks	4	96%
3	Miguel [9]	Convolution Neural Networks	10	58.73%
4	Tao Feng [7]	Deep Belief Neural Networks	4	63.75%
5	**Present Work**	**Proposed Combined Model**	**10**	**74.6%**

REFERENCES

[1] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals. In Proc. of 2nd Annual International Symposium on Music Information Retrieval, Indiana University Bloomington, Indiana, USA, 2001.

[2] Michael Haggblade, Yang Hong, and Kenny Kao. Music genre classification. Department of Computer Science, Stanford University, 2011.

[3] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 282–289. ACM, 2003.

[4] Emmanouil Benetos and Constantine Kotropoulos. A tensor- based approach for automatic music genre classification. In Signal Processing Conference, 2008 16th European, pages 1–4. IEEE, 2008.

[5] Elias Pampalk, Arthur Flexer, Gerhard Widmer, et al. Improve- ments of audio-based music similarity and genre classificaton. In ISMIR, volume 5, pages 634–637. London, UK, 2005.

[6] Carlos N Silla Jr, Alessandro L Koerich, and Celso AA Kaest- ner. A machine learning approach to automatic music genre classification. Journal of the Brazilian Computer Society, 14(3):7– 18, 2008.

[7] Tao Feng. Deep learning for music genre classification. private document, 2014.

[8] Arjun Raj Rajanna, Kamelia Aryafar, Ali Shokoufandeh, and Raymond Ptucha. Deep neural networks: A case study formusic genre classification. In Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on, pages 655– 660. IEEE, 2015.

[9] Miguel Flores Ruiz de Eguino. Deep music genre.