Music Recommendation System using Deep Learning

Dibyanshu Gautam(17BCE1328), Swapnil Rawat(17BCE1188), Priyali Poondir(17BCE1030)

Machine Learning Project (CSE4020)
Under
Dr. Suganya G.
SCSE, VIT University Chennai

Abstract- The "Music Recommendation System" is meant to work as a music player which is capable of adding similar songs to the playlist based on the genre of the initial song played by the user. This feature is helpful in recommending the user some new songs which he might like as they are similar to the song played by him. ML-based classification algorithms are used for predicting the genre of the song chosen and then this genre is used to search for similar songs from a collection (whose genre is already known). The names of this collection of songs contain the genre itself as part of the name, this fact is helpful for validation as we can check whether the genre of all the songs added to the playlist is same and is similar to the genre obtained by applying our classification model upon the song initially played.

Index Terms- CNN, GTZAN Dataset, MFCC, Music Recommendation,

I. INTRODUCTION

There is a history of variations in development of music information retrieval from a more content-based perspective. Recommendation can be described as a problem and the methods used for music recommendation have special focus on content-based recommendation by using the provided audio content features and content-based similarity measures. We used some of these features in our Content-based music recommender.

Music recommendation systems are becoming increasingly relevant due to increasee in the accessibility provided by a number of streaming services. One of the major streaming services, Spotify, includes a recommender system of its own. However, Recommendation systems still stand to be quite inaccurate when it comes to producing results and predicting songs.

Objective and subjective analysis of many previous recommender systems prove the findings in many papers that music recommendation based on content filtering get up to the mark precision. With the new modern era, People have a lot of songs to access and these millions of songs are obviously untouched by many of the users. With millions of songs to choose from, people sometimes feel overwhelmed. Thus, an efficient music recommender system is necessary in the interest of both music service providers and customers. Users will have no more pain to make decisions on what to listen while music companies can maintain their user group and attract new users by improving users' satisfaction. In the academic field, the domain of user centric music recommendation has always been ignored due to the lack of publicly available, open and transparent data.

II. BACKGROUND

Each Recommendation system that has been created in any aspect (music in our case) can be broadly classified in two filtering Models

a) Collaborative Filtering

The collaborative filtering approach to recommendation algorithms involves collecting a "large amount of information on users' behaviours, activities or preferences and predicting what users will like based on their similarity to other users". A key point to be made about this method is that the item itself, or its features, that is being recommended is not being analysed. Rather, it is making the assumption that previous information in a user's history about how they agree with other users (for instance User A liked Movie A and User B liked Movie A, so they will have similar interests), will be predictive in determining whether or not they will enjoy a certain item. Data collection under this approach includes both explicit data collection, like asking a user to rate an item, and implicit data collection, like keeping records on how often and for how long a user views an item. One popular machine learning technique used in this sort of recommender system is the k-nearest neighbour approach. One of the major issues with the collaborative filtering approach is the so called "cold start problem", in that the system need a large amount of data to make accurate recommendations.

b) Content-based Filtering

The content-based filtering approach differs from the collaborative filtering approach as it filters based on an analysis of both the item being recommended and the user. Content-based filtering closely examines the actual item to determine which features are most important in making recommendations and how those features interact with the user's preferences. Data collection can be much more complicated in content-based filtering as it is very difficult to select which features of an item will be important in creating some sort of predictive model (we will see that this is a major hurdle when it comes to music recommendation systems). Machine learning techniques such as naive Bayesian classifiers and cluster analysis are used to determine which features of an item can be used to classify it.

III. APPROACH TO SYSTEM

Choice of Dataset

Our project focuses on getting data from the audio dataset for implementing Content based Filtering. In this project, our choice of Dataset is GTZAN Dataset. GTZAN Dataset contains 1000 songs of 30 seconds which is equally divided in to 10 genres. Other Datasets that could have been used is Million Songs Dataset (MSD) and Free Music Archive (FMA). We continued with GTZAN Dataset for our convenience since it had 10 different genres to deal with.

Feature Generation

Each one of audio files from the dataset was taken and converted into equivalent spectograms. Spectogram can be defined as a visual representation of the complete spectrum of all the frequencies with respect to time. It can be seen as squared magnitude of short-term Fourier Transform (STFT) of the input audio signal. A Fourier transform is basically a representation that inputs a signal in time domain and outputs in frequencies. Mel Scale Spectogram uses Mel Scale in the y-axis to get Frequencies in different bins. This Mel scale Spectogram generation requires librosa built in library which directly converts an audio input into this. The function takes some parameters like – window length (window of time to perform transform) and hop length (number of samples between successive frames). We have used a window length of 2048 (appx 10ms). Hop length was taken as 512. Using Mel scale, we can distinguish between audio better as it scales the Hz scale using log function into dB (Decibels) and squashes the difference to relate more to human perceived pitch.
(Figure 1: All Major Genres are plotted).

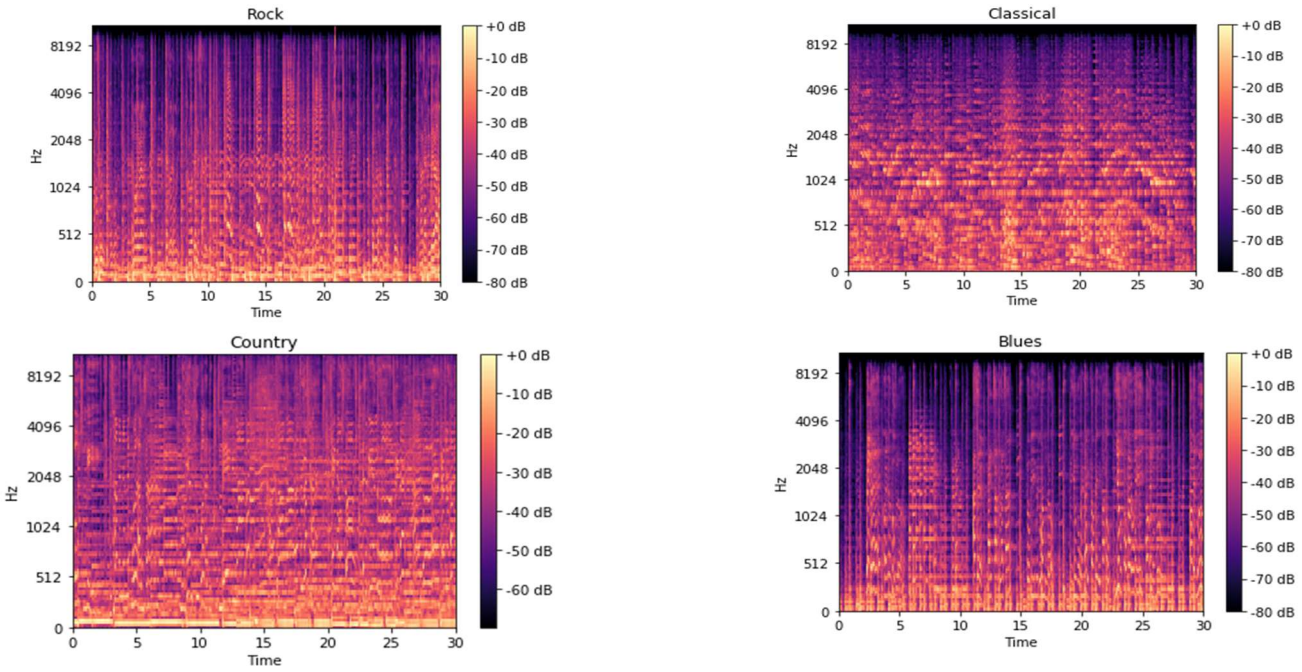


Figure 1: *Spectogram plot (From top left): Rock, Classical, Country, Blues*

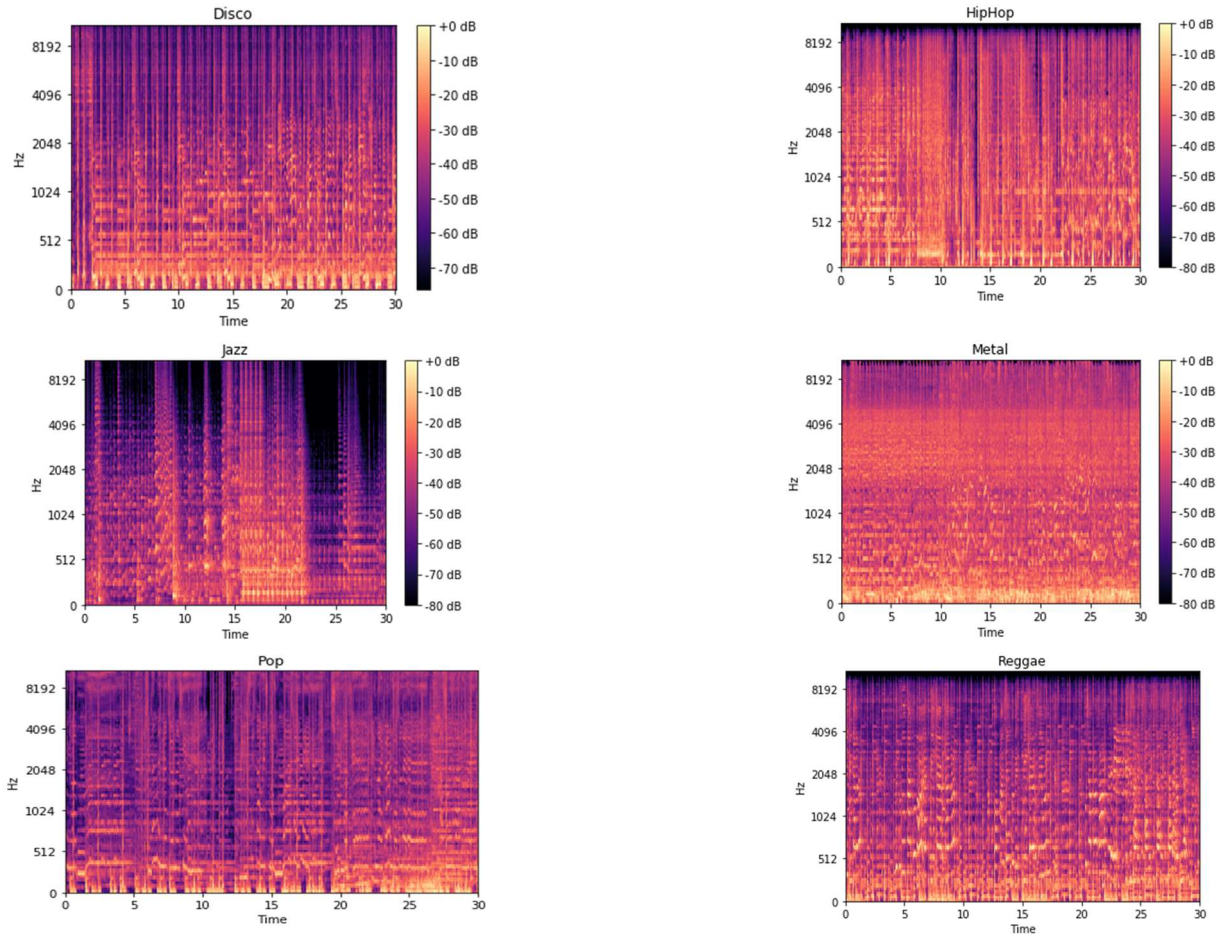


Figure2: Spectogram plot (From top left): Disco, HipHop, Jazz, Metal, Pop, Reggae

These Mel Spectograms form the basic feature for classification in our model. All these spectograms explain the Frequency to Time differences which would help us in determine the songs that are almost like the query song. We can use any model for constructing a prediction system, basically any classifier but we tried using Convolutional Neural Networks for enhanced accuracy and to use these images for better classification

IV. MODEL OF THE SYSTEM

Using the MFCC features, our feature dataset was created. It consists of RMSE, Spectral Centroid, Spectral Bandwidth, Roll off, Zero crossing rate and the other 20 MFCC Features. These completely makes our Feature dataset.

```
In [11]: data=data.drop('label',axis=1)
         data.head()
```

Out[11]:

	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	rolloff	zero_crossing_rate	mfcc1	mfcc2	mfcc3	mfcc4	...	mfcc11
0	0.349943	0.130225	1784.420446	2002.650192	3806.485316	0.083066	-113.596742	121.557302	-19.158825	42.351029	...	-8.324323
1	0.340983	0.095918	1529.835316	2038.617579	3548.820207	0.056044	-207.556796	124.006717	8.930562	35.874684	...	-5.560387
2	0.363603	0.175573	1552.481958	1747.165985	3040.514948	0.076301	-90.754394	140.459907	-29.109965	31.689014	...	-13.123111
3	0.404779	0.141191	1070.119953	1596.333948	2185.028454	0.033309	-199.431144	150.099218	5.647594	26.871927	...	-3.196314
4	0.308590	0.091563	1835.494603	1748.362448	3580.945013	0.101500	-160.266031	126.198800	-35.605448	22.153301	...	-13.083820

5 rows × 26 columns

Figure 3: Feature Dataset extracted from our GTZAN songs.

CNN Model

Image recognition related tasks have been using Convolutional Neural Networks (CNN's) since a long time. They work on the simple logic of Convolution over the complete data to understand the underneath layout. Since CNN's have a good precision on Image dataset, we thought to use this on our Mel Scale Spectograms to get better output results.

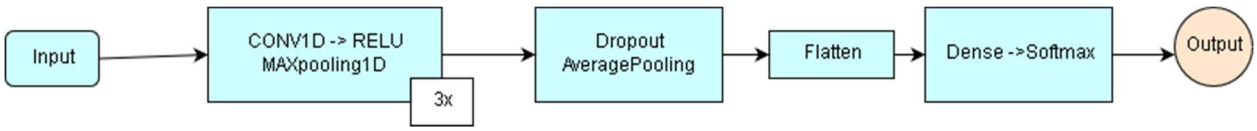
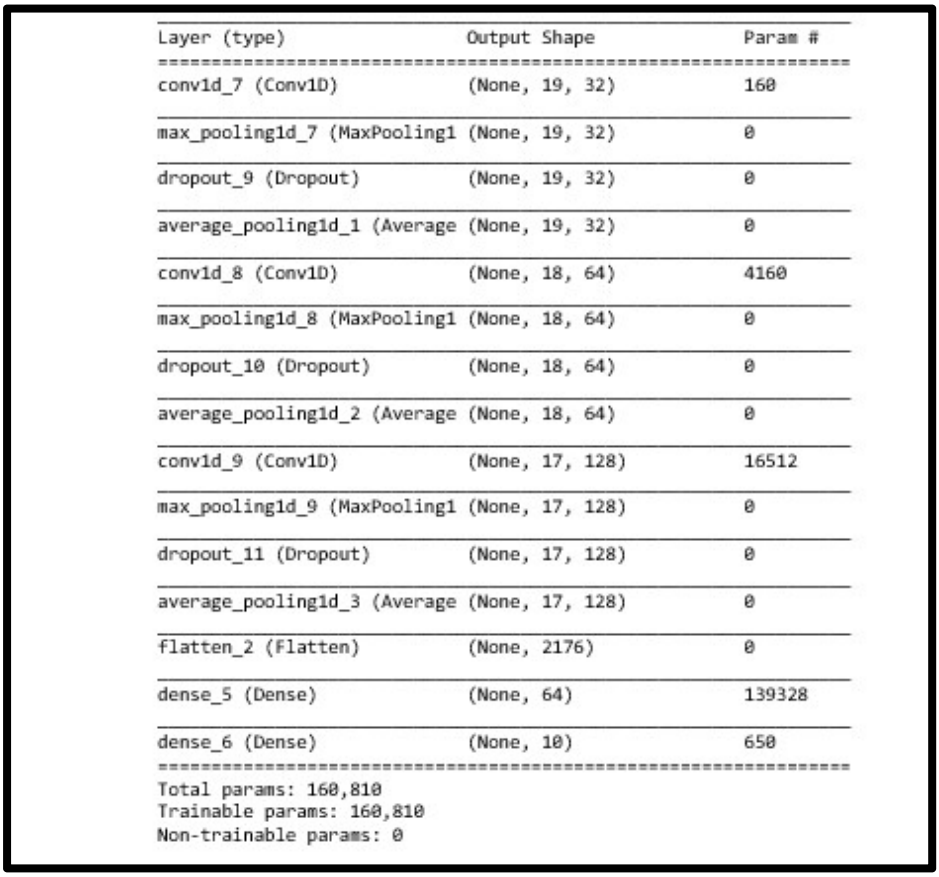


Figure 4: Model Flow Chart



Layer (type)	Output Shape	Param #
conv1d_7 (Conv1D)	(None, 19, 32)	160
max_pooling1d_7 (MaxPooling1	(None, 19, 32)	0
dropout_9 (Dropout)	(None, 19, 32)	0
average_pooling1d_1 (Average	(None, 19, 32)	0
conv1d_8 (Conv1D)	(None, 18, 64)	4160
max_pooling1d_8 (MaxPooling1	(None, 18, 64)	0
dropout_10 (Dropout)	(None, 18, 64)	0
average_pooling1d_2 (Average	(None, 18, 64)	0
conv1d_9 (Conv1D)	(None, 17, 128)	16512
max_pooling1d_9 (MaxPooling1	(None, 17, 128)	0
dropout_11 (Dropout)	(None, 17, 128)	0
average_pooling1d_3 (Average	(None, 17, 128)	0
flatten_2 (Flatten)	(None, 2176)	0
dense_5 (Dense)	(None, 64)	139328
dense_6 (Dense)	(None, 10)	650

Total params: 160,810
Trainable params: 160,810
Non-trainable params: 0

Figure5: Model information by Keras.

The model was trained with Adam optimizer and loss function categorical cross entropy. The model was then trained for around 72 epochs with batch sizes of 256 used. Total Parameters came out to be 160,810 with all of them being trainable.

V. RESULT

After the Hyper parameter Tuning, an Accuracy of 92 % on Training set was achieved and 54 % on Test Dataset which is much higher than a regular CRNN network model as well.

Following plots are the training and validation Accuracy and loss Graphs to discuss the flow.

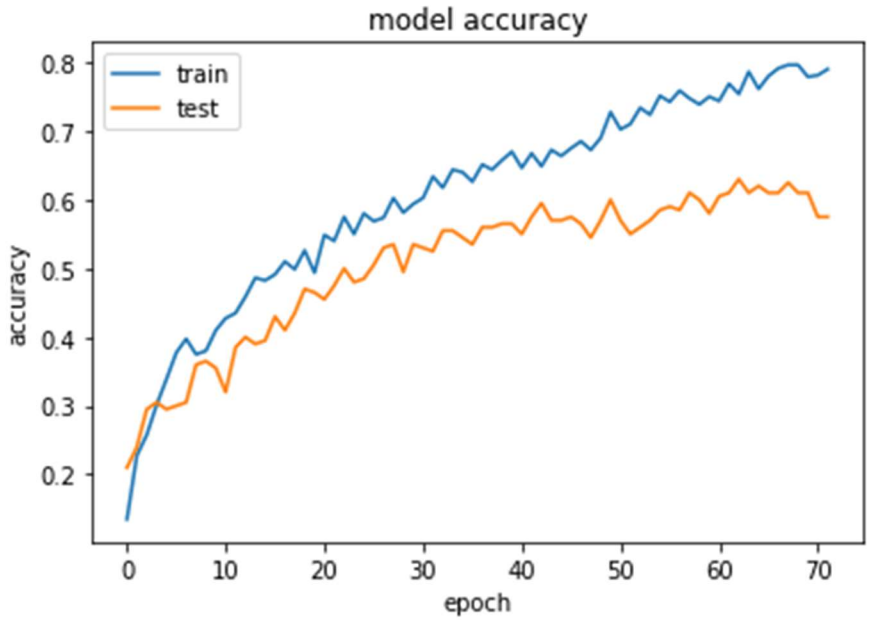


Figure 6: Accuracy over the Epochs

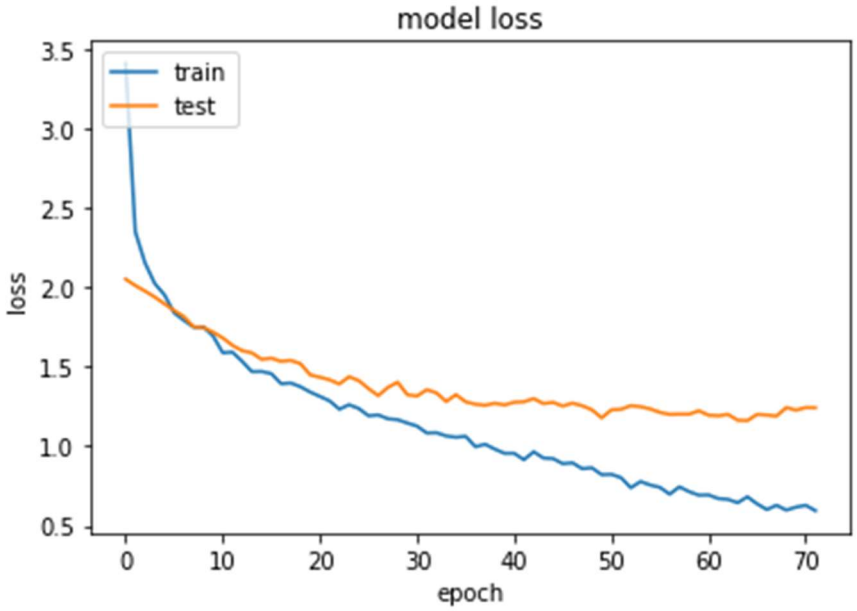


Figure 7: Model Loss over the Epochs

VI. CONCLUSION AND FUTURE WORK

Our proposed system is capable of extracting the audio features required for the analysis from any audio file (not constraining just to the files available in the trained dataset), and predict its genre as an intermediate step. The efficiency of the system while using CNN classification model is expected to be more than any other model, however since we are using a small dataset consisting of 1000 songs the obtained results are not as accurate as expected, rather the accuracy is similar to what obtained by using other common classification models such as random forest , KNN etc. This observation leads to the conclusion that CNN classification is best suited for larger datasets.

Neural networks and Deep learning techniques have been among the most potential fields of study for various kinds of applications in the recent years. This trend provides insights which suggest that they can prove instrumental in developing more efficient recommendation systems in future based on user activity and preferences rather than based on any other extracted parameters.

FINAL MUSIC PLAYER

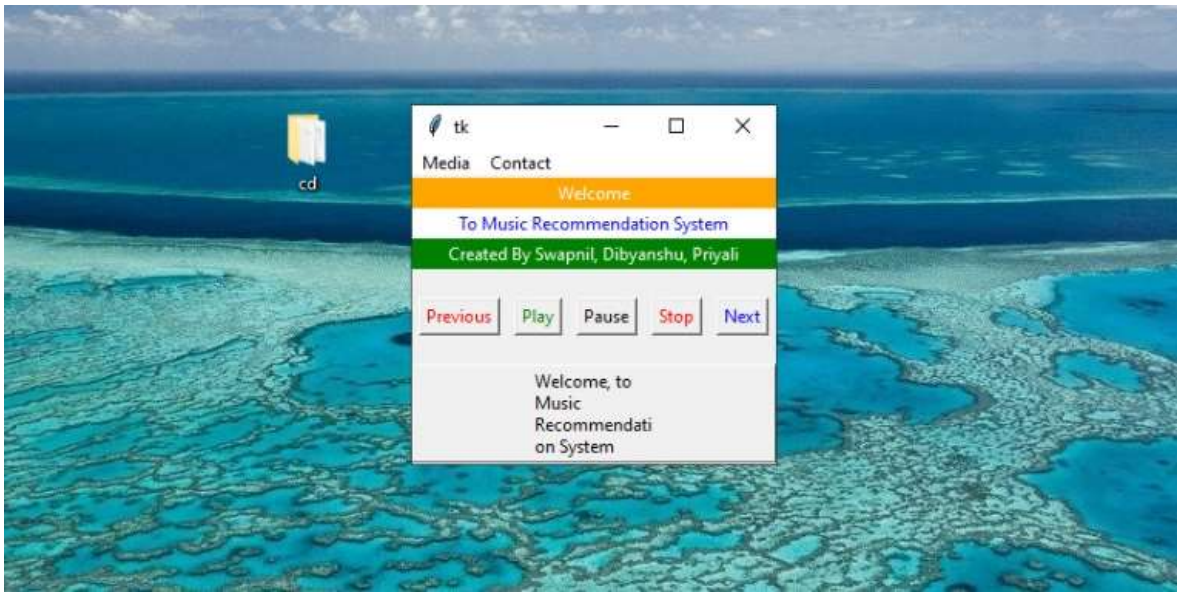


Figure 8: Music Player UI

ACKNOWLEDGMENT

We would like to thank Dr. Suganya. G for her able guidance on this project.

REFERENCES

[1] Deep content – based music recommendation => Aaron van der Oord, Sander Dieleman, Benjamin Schrauwen (2013)

[2] Evaluation of Musical Features for Emotion Classification => Yading Song, Simon Dixon, Marcus Pearce (2012)

[3] Music Recommendation Based on Acoustic Features and User Access Patterns => Bo Shao, Dingding Wang To, Tao Li, Mitsunori Ogihara

[4] A Music Recommender Based on Audio Features, January 2004 => Qing Li, Byeong Man Kim, Dong Hai Guan, Duk whan Oh

[5] A content – based music recommender system, May 2017 => Juuso Kaitila

[6] A Machine Learning based Music Retrieval and Recommendation System => Naziba Mostafa, Yan Wan, Unnayan Amitabh, Pascale Fung

[7] Deep Content – User Embedding Model for Music Recommendation => Jongpil Lee, Kyungyun Lee, Jiyoung Park, Juhan Nam (July 2018)

[8] Contextual music information retrieval and recommendation: State of the art and challenges, Marius Kaminskas, Francesco Ricci (April 2012)

[9] Music Recommendation using Collaborative Filtering and Deep Learning, Anand Neil Arnold, Vairamuthu S (May 2019)

[10] Content – based music recommendation using underlying music preference structure, Mohammad Soleymani, Anna Alijanaki, Frans Wiering, Remco C. Veltkamp

[11] A Survey of Music Recommendation system, Puja Deshmukh, Dr. Geetanjali Kale

[12] A New Benchmark Dataset for Building Context – Aware Music Recommender Systems, Ashmita Poddar, Eva Zangerle, Yi- Hsuan Yang

[13] Context – aware music recommendation based on latenttopic sequential patterns, Negar Hariri, Bamshad Mobasher, Robin Burke (Sept 2012)

[14] Location – aware music recommendation using auto – tagging and hybrid matching (October 2013)

[15] Music Recommendation: Audio Neighbourhoods to Discover Music in the Long Tail, Susan Craw, Stewart Massie, Ben Horsburgh (2015)

[16] A collaborative filtering method for music recommendation using playing coefficients for artists and users, Diego Sanchez Moreno, Ana B Gill Gonzalez, M. Dolores Munoz Vicente, Vivian F Lopez Batista, Maria N Moreno Garcia (2016)

[17] Learning Features from Music Audio with Deep Belief Networks, 2010, Philippe Harmel and Douglas Eck

[18] Deep Convolution Neural Network, Textual Features and Multiple Kernel Learning for Utterance – Level Multimodal Sentimental Analysis, Soujanya Poria, Erik Cambria, Alexander Gelbukh

[19] Deep multimodal learning for Audio – Visual Speech Recognition, 2015, Youssef Mrouch, Etienne Marcheret, Vaibhava Goel

[20] Multiscale approaches to music audio feature learning, 2013, Sander Dieleman and Benjamin Schrauwen

[21] Improving Content -Based and Hybrid Music Recommendation using Deep Learning, 2014, Xinxi Wang, Ye Wang

[22] Convolutional Deep belief networks for feature extraction of EEG signal, 2014, Yuanfang Ren, Yan Wu

[23] An Evaluation of Audio Feature Extraction toolboxes, David Moffrat, David Ronan, Joshua D. Reiss

[24] Librosa: Audio and Music Signal in Python, Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, Oriol Nietok

[25] Meyda: An audio feature extraction library for the Web Audio API, Hugh Rawlinson, Nevo Segal, Jakub Fiala