MUSIC GENRE CLASSIFICATION OF AUDIO SIGNAL

Nivedit Jain(B18CSE039)
Muskan Bathla(B18CSE036)
Diya Sankhla(B18BB011)
Suniket Das(B18EE062)
Shashi Deegwal(B18EE048)

Abstract

This project was primarily aimed to predict the genre of an input music file. For this we used kNN model of classification in machine learning. We used the famous GTZAN dataset[4] and used Chroma frequency, Spectral roll-off, Spectral centroid, MFCC, Zero crossing rate in this project.

1. Introduction

Music for a long time has served as an art through which humans express their emotions. Music can lift anyone's mood up, it can act as a source of inspiration for someone or a kind of therapy for someone else. Professor Gideon Nave, says "When people are given the task of getting to know each other, music is one of the first things that they tend to talk about."[1]. Due to its irreplaceable role, the music itself is a fascinating topic to study.

With the rapid expansion of technology, there is also a rapid growth in the music industry and a lot of applications (like Spotify, Soundcloud, Youtube Music, etc) to stream music have also emerged. These platforms have a database of more than millions of songs; to predict what a user might like and to organize their data in a better way, they are always in need for automatic classification algorithms on different basis like genre, emotions, etc.

We have tried to classify a given music file as blues, classical, country, disco, hip-hop, jazz, reggae, rock, metal, or pop, similar to[2][3]. We have trained a machine learning model for prediction (kNN Model). All our implementation and testing are based on the famous, GTZAN dataset[4].

We have used **STFT(Short Time Fourier Transform), DTFT(Discrete Time Fourier Transform), Sampling, inverse DTFT** as tools from signals and systems.

The rest of this report is divided into 3 sections, section 2 is further divided into 2 sections.

2. Feature Extraction and Model training

The GTZAN dataset has 10 classes with 100 music files in each class. We have preprocessed files and clipped them to 30 second audio files. We have divided this section into 2 parts, which are followed in steps.

2.1 Feature Extraction

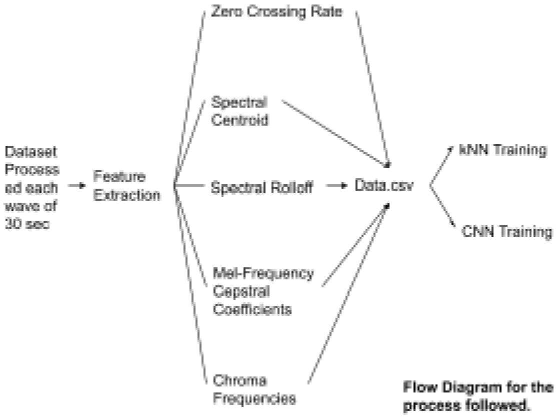

Fig 1. The process followed for the project.

2.1.1 Average Zero Crossing Rate

The zero-crossing rate is defined as the rate of crossing zero of a particular signal within a certain time period, whereas the Average Zero crossing rate is defined as the average of zero-crossing rates if the signal is divided into various parts of equal length. In our project, we have kept the size of a part (to calculate the zero-crossing rate) to be 2048 as done in [4].

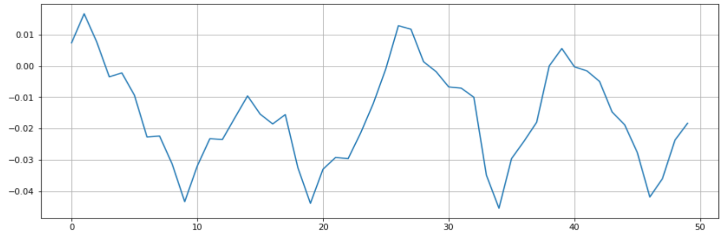

Fig 2.a. The above graph has 5 zero crossings with a zero-crossing rate of 0.102

$$\frac{1}{T-1}\sum_{t=1}^{T-1}\mathbb{I}\left\{s_t s_{t-1} < 0\right\}$$

Here, consider that if the signal is from 0 to N, then T=N, s_t and s_{t-1} are signal values at time t. □ is 1 if true else 0. [5]

Fig 2.b. Expression for calculating Zero Crossing Rate

2.1.2 Spectral Centroid

It represents the magnitude of the centroid of DTFT of an input signal. This feature can help us to differentiate between signals of consistent values and with rapidly varying values.

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)\,x(n)}{\sum_{n=0}^{N-1} x(n)}$$

Fig 3. Formula for Centroid

2.1.3 Spectral Rolloff

Spectral Rolloff basically provides information about the shape of the signal. It is defined as the frequency below which 85% of the magnitude spectrum is located.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n]$$

Fig 4.Here, Mt represents magnitude spectrum And Rt represents the value of Spectral roll-off.[2]

2.1.4 Mel-Frequency Cepstral Coefficients

The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features that describes the shape of a spectral envelope. It is used in many state-of-the-art human voice recognition systems.

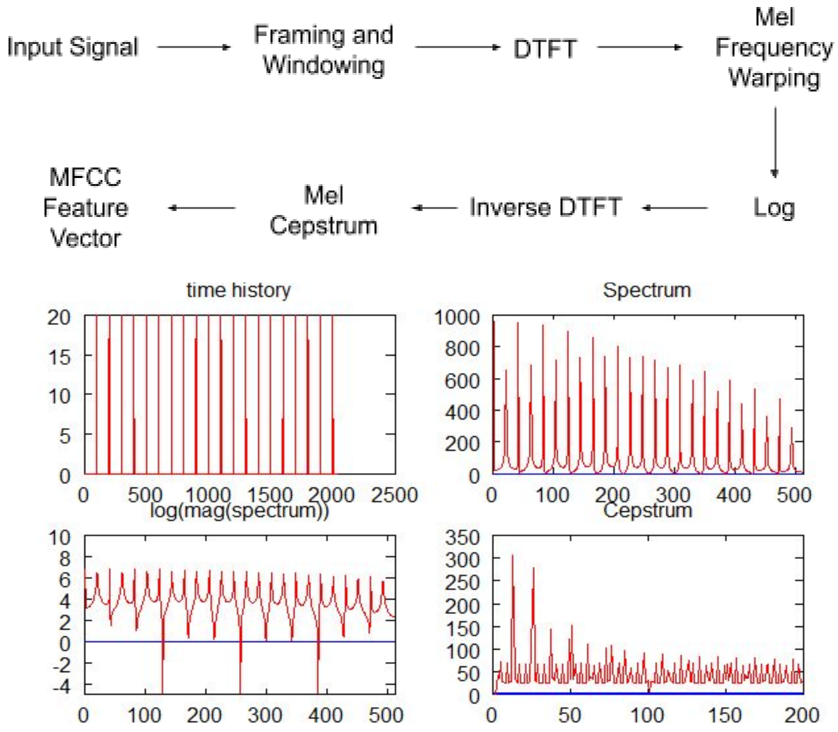

Fig.5. The process to Calculate MFCC [6]

$$\mathbf{Mel(f) = 2595*log(1+f/700)}$$

Fig.6. Formula to Calculate Mel Scale [6][7]

2.1.5 Chroma Frequencies

Chroma features are closely related to twelve different classes of the pitch. Chroma features help in identifying pitches that differ by an octave." They help us to gather information about the harmonic and melodic characteristics of music.

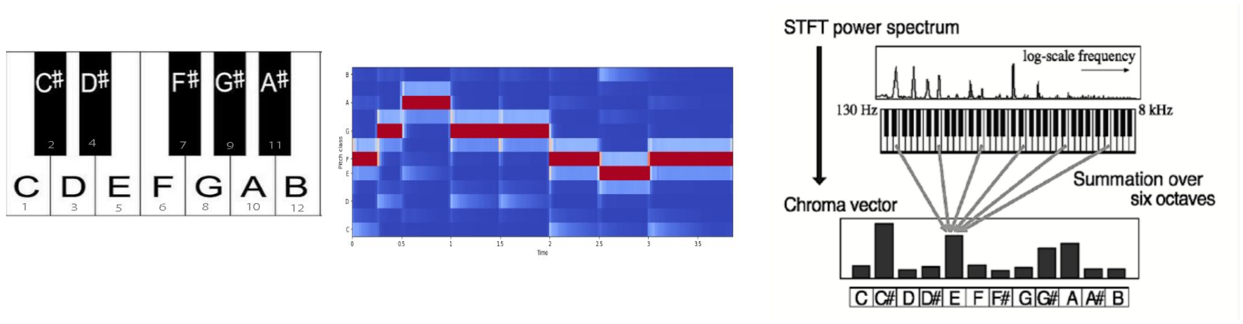

Fig 7. 12 chromas and chromatogram representation and algorithm[8]

2.2 Training kNN Model

k-Nearest-Neighbours(kNN) is a machine learning model used for classification. It is a supervised machine learning algorithm that accepts some inputs and their corresponding outputs for training so as to classify new data. kNN works by finding the 'k' closest labeled points (the nearest neighbours) to a given data point and assigning the majority label amongst them to the unlabeled data points into consideration. Following is the algorithm:

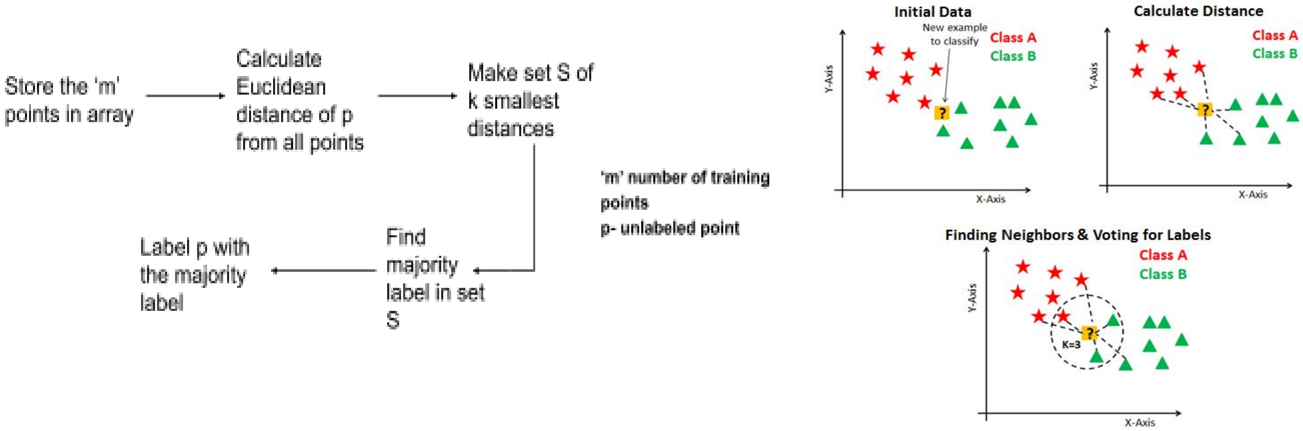
Store the 'm' points in array
Calculate Euclidean distance of p from all points
Make set S of k smallest distances
Find majority label in set S
Label p with the majority label
'm' number of training points
p- unlabeled point
Initial Data
New example to classify
Class A
Class B
Y-Axis
X-Axis
Calculate Distance
Class A
Class B
Y-Axis
X-Axis
Finding Neighbors & Voting for Labels
Class A
Class B
K=3
Y-Axis
X-Axis

Fig 8. kNN algorithm

2.2.1 Hypertuning model parameters

Hypertuning parameters is the process of finding optimal parameters for the model used; in order to achieve better accuracy. In this case, we find the optimal value for number of nearest neighbours of the unclassified point we want to take into consideration. Here, we have trained the model multiple times on a specified range of parameters and then figured out the optimal value which gives the best accuracy.

3. Results and Observations

Hypertuning model parameters

Parameter	Accuracy (%)	Parameter	Accuracy (%)
1	63.00	6	61.50
2	61.10	7	61.40
3	61.99	8	62.70
4	61.80	9	61.50
5	**63.70**	10	62.80

The maximum accuracy achieved in kNN model after 20-fold cross-validation and hypertuning model parameters is **63.70%** for 5 nearest neighbours.

4. Conclusions and Limitations

We have successfully used various tools of Signals and Systems for feature extraction which were the basis of machine learning model training. We were successful in training the kNN machine learning model and have achieved a decent accuracy. Limitations of our project include less optimization and high running time for feature extraction. Future works can be done on making faster algorithms and making better classification models to achieve better accuracy. One may try various other approaches like CNN on spectrogram, using Neural Networks etc.

Our Code base can be viewed on
https://colab.research.google.com/drive/19TT0Elf1YEO7qwqTnvQ58H0EFu1tRB2N

References

[1]. Nave, G., Minxha, J., Greenberg, D. M., Kosinski, M., Stillwell, D., & Rentfrow, J. (2018). Musical Preferences Predict Personality: Evidence From Active Listening and Facebook Likes. Psychological Science, 29(7), 1145–1158. https://doi.org/10.1177/0956797618761659

[2]. G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, July 2002.
DOI: 10.1109/TSA.2002.800560

[3] Parul Pandey, “A Guide to analyzing Audio/Music signals in Python”. Accessed 15 Nov. 2019 https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8

[4] Marsyas, GTZAN Genre Collection Dataset, http://marsyas.info/downloads/datasets.html. Accessed 14th November 2019.

[5] Archit Rathore, Margaux Dorido, Project Report, Accessed 19 November 2019,
https://www.cse.iitk.ac.in/users/cs365/2015/ submissions/archit/report.pdf

[6] Pratheeksha Nair, ” The dummy’s guide to MFCC”, Accessed 21 November 2019,
https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd

[7] Davis, S. Mermelstein, P. (1980) “*Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*”. In IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366

[8] Mellina, Alessio & Sentinelli, Alexandro & Marfia, Gustavo & Roccetti, Marco. (2012). AREEB: Automatic REfrain extraction for ThumBnail. 2012 IEEE Consumer Communications and Networking Conference, CCNC'2012. 10.1109/CCNC.2012.6181003.

References to Libraries Used in Code: Our Code base is on Python3 and all libraries below are of python3 (iPython Google Colab)

Google Colab:https://colab.research.google.com/
Pandas:https://pandas.pydata.org/
Keras:https://keras.io/

matplotlib: https://matplotlib.org/
librosa:https://librosa.github.io/librosa/
Scikit-Learn:https://scikit-learn.org/