**Extracting and Analyzing the Data in Sound Files to Train a Music Genre Classification Model**

Tom Constertina

Case Studies in Machine Learning  
University of Texas at Austin

**ABSTRACT**

Music classification has become increasingly important over recent years, with the widespread use of streaming services that provide music suggestions for its users. We have developed a simple music classification model that has been trained on the songs from the free-to-use GTZAN Dataset to classify sound files into a musical genre based on their features [1]. For each song, we decomposed the song into different musical features using the Python library of LibROSA [2]. Musical features such as MFCCs, spectral bandwidth, chroma features, and tempo were extracted and analyzed. After that, we used all of these features to train a model and evaluate its performance. We trained on pre-built models from the Python library of *sklearn* and our own model with a custom neural network architecture.

**INTRODUCTION**

Music is an incredibly diverse and rich form of art, with many different genres and styles from numerous cultures throughout thousands of years. Even with the culmination of the experience and development of music throughout the years, music is hard to quantify. What makes a good singer good or bad if they both sing on-key? How does emotion change the way music is perceived? How does vibrato change the perception of a song [3]? With the introduction of recording equipment and sound analysis, some of these questions are becoming easier to define. What was once a very abstract art form can now be analyzed, stored as data, and decomposed to determine what makes up a sound. Generative AI models have taken great strides in interpreting music and then being able to generate unique music.

Though AI is no replacement to human-created music, it can play a key role in understanding sound and shaping the next era of music [5]. With technology playing a role in assisting music classification and generation, this paper aims to learn what features make up a sound, how to create a model on the data extracted from a song, and the best-performing model in classifying music into a genre. While in this paper we are only focused on genres, this approach can also be used for more abstract classifications, like finding other songs that are similar to a user’s preference in melody, rhythm, instrument types, and more.

**BACKGROUND**

There has already been a great deal of research and work gone into developing AI models for music tasks. Oord et al. explored how transfer learning can be utilized to overcome the issue of lack of availability of free music [6]. In this paper, it discusses being able to transfer the knowledge from a model for a different, but related task, and utilize it for a new task with a different dataset. This helps to overcome the issue of scarcity of copyright-free music as well as the large file sizes and lengthy training times that go into model development.

Jia discussed creating a convolutional neural network to train a model to classify the overall ‘emotion’ of a song [7]. The input features for this model were the MFCC features of a song, and we will also use MFCC features in our model. MFCC stands for Mel-Frequency Cepstral Coefficient and has become instrumental in many aspects of speech processing [8]. MFCC will be helpful in our project as it can represent aspects of a song such as its rhythmic, harmonic, and timbral properties.

Lazaro et al. uses Audio Spectrum Centroid (ASC), Audio Spectrum Flatness (ASF), and Audio Spectrum Spread (ASS) to classify a song’s tempo [9]. ASC is used to determine the most prominent frequency at a certain point in time. ASF is used to measure the ‘flatness’ of a song at a point in time. A high flatness would mean there are a lot of frequencies present in similar strengths, such as white noise. A low flatness would indicate just a few frequencies dominating the sound. ASS is used to quantify how variant the frequencies are in a song, in relation to the ASC. If a song has several prominent frequencies which are vastly different from each other, the spectral spread will be very large [10]. In our project, we used ASC and ASF as features in our project, and the combination of them help to identify the tempo of a song.

Das et al. used a similar approach to our music classification task to develop a model to classify urban noises such as a siren or a jackhammer [11]. They created a model that combined both a convolutional neural network (CNN) and a long short-term memory (LSTM) model and compared their performance to popular pre-built models. While we have come up with a different set of features, the architecture of theirs is going to be heavily used as a basis for our own model’s architecture. One of the features they used was Chroma STFT, which we also used. This represents the intensity of different pitch classes, or notes, of a sound at a particular time.

**METHODOLOGY**

1. *Dataset*

The music dataset I used for model training was the GTZAN Dataset [1]. This dataset provides 30 second clips of different songs for 10 unique genres, all which are free-to-use and publicly available. The 10 different genres are Blues, Classical, Country, Disco, Hip-Hop, Jazz, Metal, Pop, Reggae, and Rock. There is a total of 1000 songs, as there are 100 song clips for each genre. They also provide a list of features, similarly extracted from LibROSA, but their features were not used. The path to a song is defined as ‘/Data/genres\_original/{genre}/{genre}.000{00-99}.wav’. One minor caveat to be mentioned is that the ‘jazz.00054.wav’ file was corrupted, so it was not used in training the model.

1. *Audio Features*
   1. *Chroma STFT*

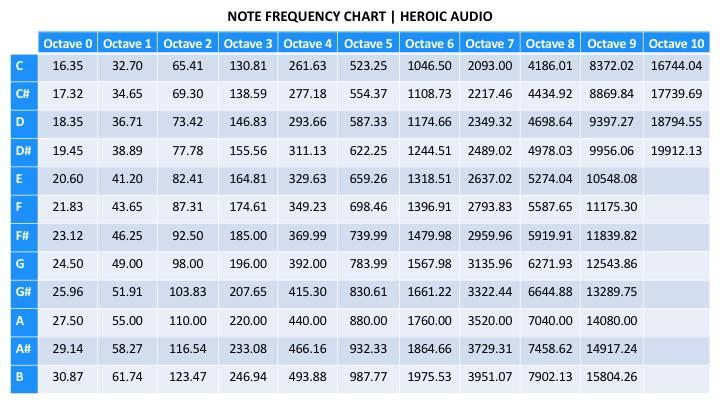
The first feature extracted from the songs was the Chroma STFT. STFT stands for short-term Fourier transform, and maps a signal into a function of time and frequency [12]. We can use this on a song and then use this to generate a chroma feature matrix that will represent the frequency as a musical pitch, or note (C, D, E, F…), and portray music notes that dominate the sound with a brighter light to represent what percentage of the sound is made up by frequencies of that musical note. We then pass both the mean and variance of this Chroma STFT into our model as a feature.

A close-up of a graph

Description automatically generated

*Figure 1. A STFT representation of a 3-second song clip (upper) and a Chroma STFT representation of the same sound clip.*

Note that a higher frequency does not necessarily correlate to a higher pitch class, because of the presence of octaves. There are multiple frequency ranges for one particular note, each capturing the note at a different octave. The Chroma STFT does not capture octaves, but only the type of note [13].



*Figure 2. A Note Frequency Chart, representing the frequency of notes at different octaves [14].*

**REFERENCES**

\* Indicates a non-scholarly source

\*[1] GTZAN Dataset. (2023). *Kaggle.* Retrieved from <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>.

[2] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. “librosa: Audio and music signal analysis in python.” In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

[3] Seashore, C. E. (1937). The Psychology of Music. Music Educators Journal, 23(4), 30-33. <https://doi.org/10.2307/3385146>.

[4] Barenboim, G., Debbio, L.D., Hirn, J. *et al.* Exploring how a generative AI *interprets* music. *Neural Comput & Applic* **36**, 17007–17022 (2024). <https://doi.org/10.1007/s00521-024-09956-9>.

\*[5] Semanick, Alex. (2024, April 3). *How AI is transforming the creative economy and music industry*. OHIO News. <https://www.ohio.edu/news/2024/04/how-ai-transforming-creative-economy-music-industry>.

[6] Oord, A.V., Dieleman, S., & Schrauwen, B. (2014). Transfer Learning by Supervised Pre-training for Audio-based Music Classification. *International Society for Music Information Retrieval Conference*.

[7] Jia, Xiaosong, A Music Emotion Classification Model Based on the Improved Convolutional Neural Network, *Computational Intelligence and Neuroscience*, 2022, 6749622, 11 pages, 2022. <https://doi.org/10.1155/2022/6749622>.

[8] Tiwari, V. (2010). MFCC and its applications in speaker recognition. *International journal on emerging technologies*, *1*(1), 19-22.

[9] A. Lazaro, R. Sarno, R. J. Andre and M. N. Mahardika, "Music tempo classification using audio spectrum centroid, audio spectrum flatness, and audio spectrum spread based on MPEG-7 audio features," *2017 3rd International Conference on Science in Information Technology (ICSITech)*, Bandung, Indonesia, 2017, pp. 41-46, doi: 10.1109/ICSITech.2017.8257083.

[10] Giannakopoulos, T., & Pikrakis, A. (2014). Chapter 4 - Audio Features. In T. Giannakopoulos & A. Pikrakis (Eds.), *Introduction to Audio Analysis* (pp. 59–103). doi:10.1016/B978-0-08-099388-1.00004-2.

[11] J. K. Das, A. Ghosh, A. K. Pal, S. Dutta and A. Chakrabarty, "Urban Sound Classification Using Convolutional Neural Network and Long Short Term Memory Based on Multiple Features," *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, Fez, Morocco, 2020, pp. 1-9, doi: 10.1109/ICDS50568.2020.9268723.

[12] Yang, Y., Peng, Z., Zhang, W., & Meng, G. (2019). Parameterised time-frequency analysis methods and their engineering applications: A review of recent advances. *Mechanical Systems and Signal Processing*, *119*, 182–221. doi:10.1016/j.ymssp.2018.07.039.

\*[13] Ultimate Guide to Musical Frequencies. *iDrumTune*.Obtained from <https://www.idrumtune.com/ultimate-guide-to-musical-frequencies/>.

\*[14] Understanding Note Frequency Charts (And Why You Should Be Using One). *Produce Like a Pro*. Obtained from <https://producelikeapro.com/blog/note-frequency-chart/>.