Music Genre Classification

Clark Brown, Sam Cochran, Daniel Swingle

December 7, 2020

1 Introduction

Our work seeks to curate audio features to train a music genre classifier. Such a classifier would be able to take in a set of audio features for a song and accurately determine the genre of that song–a task that is accomplished by most humans with minimal background in music. There are a number of difficulties in such a problem not limited to the definition of "genre" and selecting appropriate audio to train the model.

1.1 Motivation

It is a somewhat simple task for a trained musician or musicologist to listen to a work of music and label its genre. What do we need to help a computer complete the same task? Questions we want to answer:

1. What features of music make it a part of its genre?

1.2 Related Work

There have been many studies in the area of genre classification in machine learning. Traditionally models have used learning algorithms for SVM and KNN and have relied heavily on commone spectral features including the MFCCs (1). The state of the art has improved over time with most classical machine learning classifiers managing 60-70% accuracy. This is similar to human capabilities with short song intervals according to some human trials (2). In more recent years, neural networks have been able to make more accurate predictions near 80-90% accuracy in some cases.

2 Data

Our data comes from the Free Music Archive ([https://github.com/mdeff/fma]) reated by Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. International Society for Music Information Retrieval Conference (ISMIR), 2017.

We use the audio files and genre tags, but build our own features. We also use the small data set composed of 8000 30-second songs (8 GB in `.mp3` fromat). We convert each file to a `.wav` for simplicity. Each song is designated by a `track_id` and labeled with one of eight genres: Hip-Hop, Pop, Folk, Experimental, Rock, International, Electronic, and Instrumental. There songs are distributed evenly across genres with 1000 songs per genre.

2.1 Potential Issues

One potential issue with our data is that the dataset is composed entirely of free music (creative commons), and therefore our model may have difficulty analyzing other kinds of music, which may be quite different.

Specifically, we have reason to believe that the genre definitions, quality, and style of a free music database may differ from commercial music, so we will have to find a way to evaluate how well a model trained on a free music database can generalize to samples of commercial music

2.2 Missing Data

The dataset is fairly robust, but of the 8000 tracks, there are 6 that are not actually 30 seconds long. We ignore these tracks from our analysis.

2.3 Ethical Concerns and Implications

The music used in our work comes from the Creative Commons and is liscensed for this kind of use. We see no privacy concerns with the collection of this data. As music genre does not make a serious impact on the commercialization of music or the daily lives of non-musicians, we do not anticipate and negative repercussions from our work. The lines around genre are vague enough to ensure that professors of music theory and music history need not worry that they shall be out of a job.

3 Feature Engineering

Since our original data was made up only of track IDs corresponding to wav files, and their genre labels, our feature extraction makes up all of our useful data. We created a dataframe that has the following features as its columns. In the next section, we discuss the meaning of each added feature column.

3.1 Feature Descriptions and Reasoning

Track ID: each wav file corresponds to a number, and we have a function that generates the file path to access each track if needed. Genre Code: We have encoded our eight genres by a 1:1 mapping to integers 0-7.

Zero Crossing Rate: Indicates the average rate at which the sign of the signal changes. Higher zero crossing rates match with higher percussiveness in the song. We added this feature because genres often have a certain feel relative to beat and percussive sound.

Frequency Range: The max and min frequency the audio ignoring the top 20% and bottom 20%. Clipping the top and bottom was important because almost all of our audio files go from 10 Hz to 10000 Hz. But seeing the range in where most of the sound of a song is seems to be connected to genre. Some genres have greater ranges while others are in a small range.

Key and Tonality: We used the Krumhansl-Schmuckler algorithm to estimate the most likely key that the audio sample is in, and whether the key is major or minor. We chose this because even though most genres have songs in different keys, knowing the key will aid in normalizing pitch information for other features.

Mel Frequency Cepstral Coefficients (MFCCs): Represents the short term power spectrum of the sound. Aligns closely with the human auditory system's reception of sound. These 30 coefficients describe the sound of a song in a human way. MFCCs are being used more and more

in Music Information Retrieval specifically with genre tasks because they encapsulate the human experience of sound. We feel this will improve accuracy.

Spectral Rolloff: The frequency below which a certain percent of the total spectral energy (pitches) are contained. When audio signals are noisy, the highest and lowest pitches present do not convey much information. What is more useful is knowing the frequency range that 99% of the signal is contained in, which is what the spectral rolloff represents.

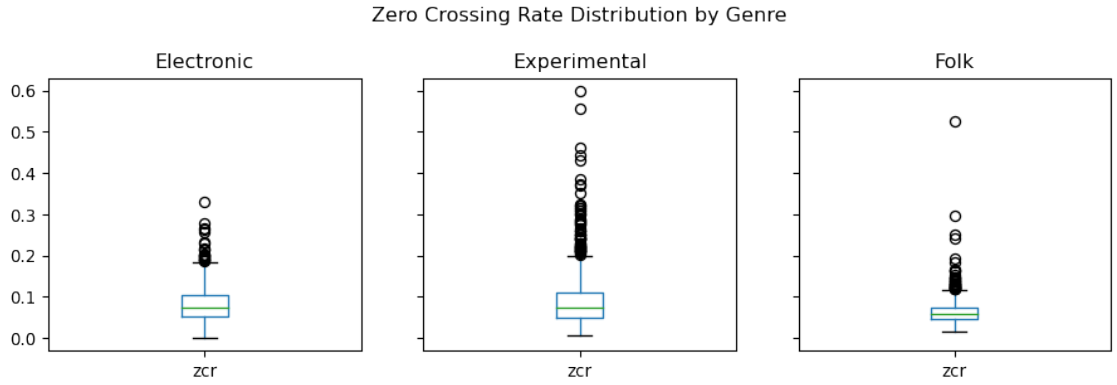
The Three Highest Tempo Autocorrelation Peaks: Indicative of what we would guess the average BPM will be for this audio file (3 columns). This is a way of summing up the entire tempogram array in just a few numbers so that comparing tempo features for each track is tractable.

Average Tonnetz over all Time: The mean and variance of the x and y dimensions of the tonal centers for the major and minor thirds, as well as the fifths (this ends up being 6 means and 6 variances for a total of 12 columns). Here we take the means and variances to reduce the information down from a 6xt matrix (where t is the number of time values, about 1200) to just 12 numbers that sum up that matrix for each track.

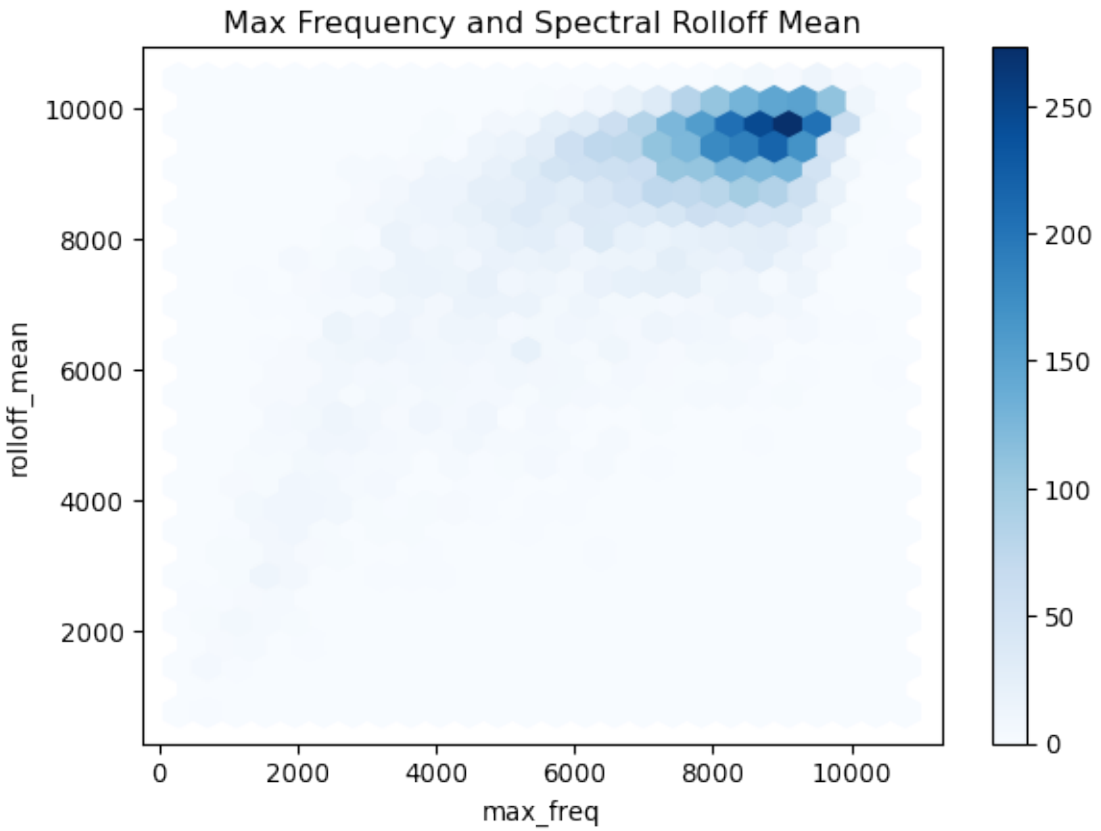
4 Visualization and Analysis

4.1 Visualization

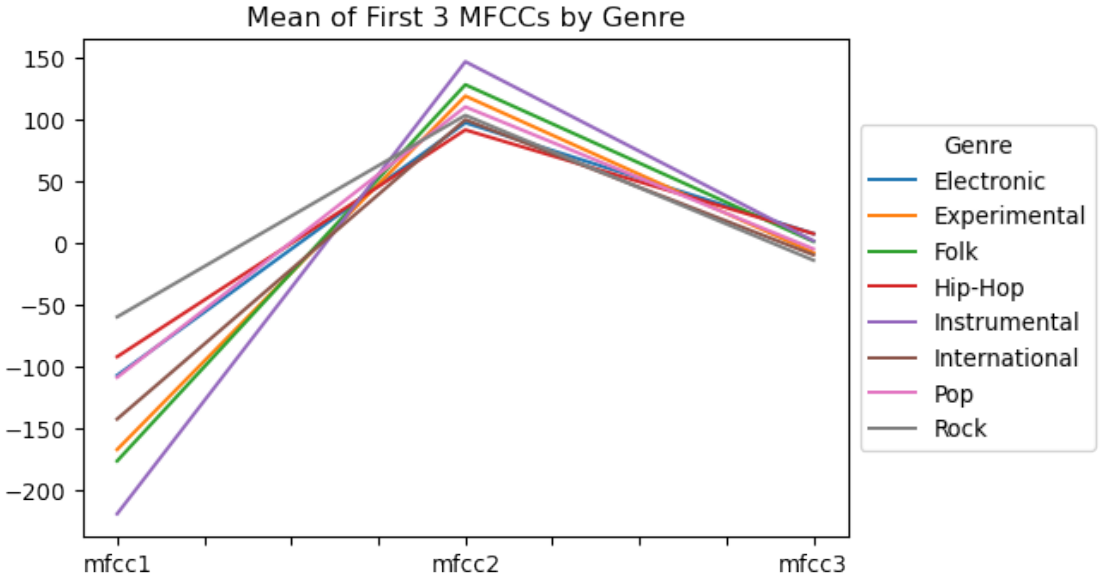
For now we present visualizations of most of these features. We will eventually be more selective.

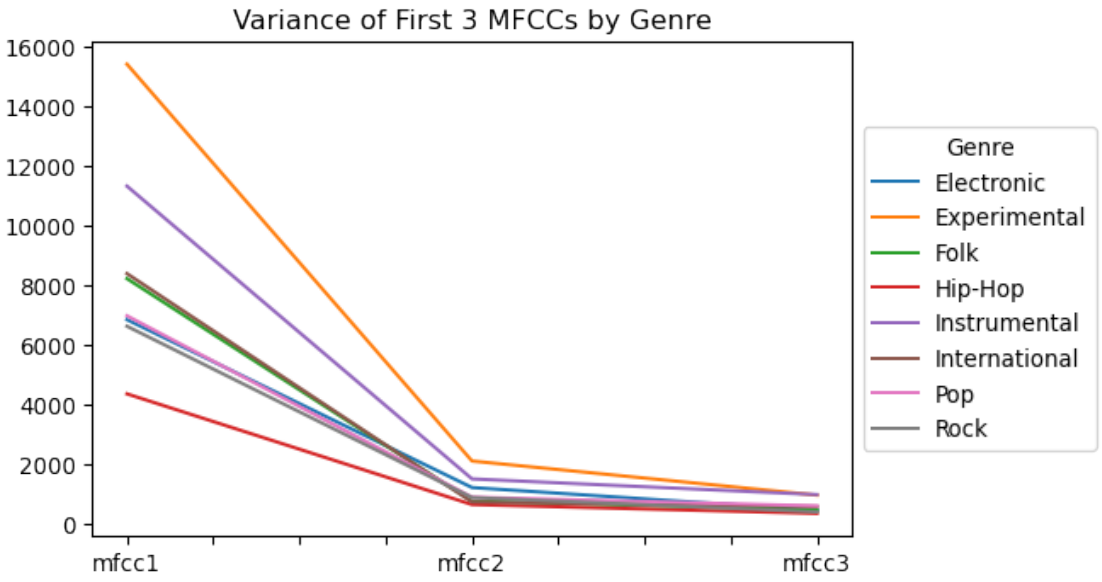


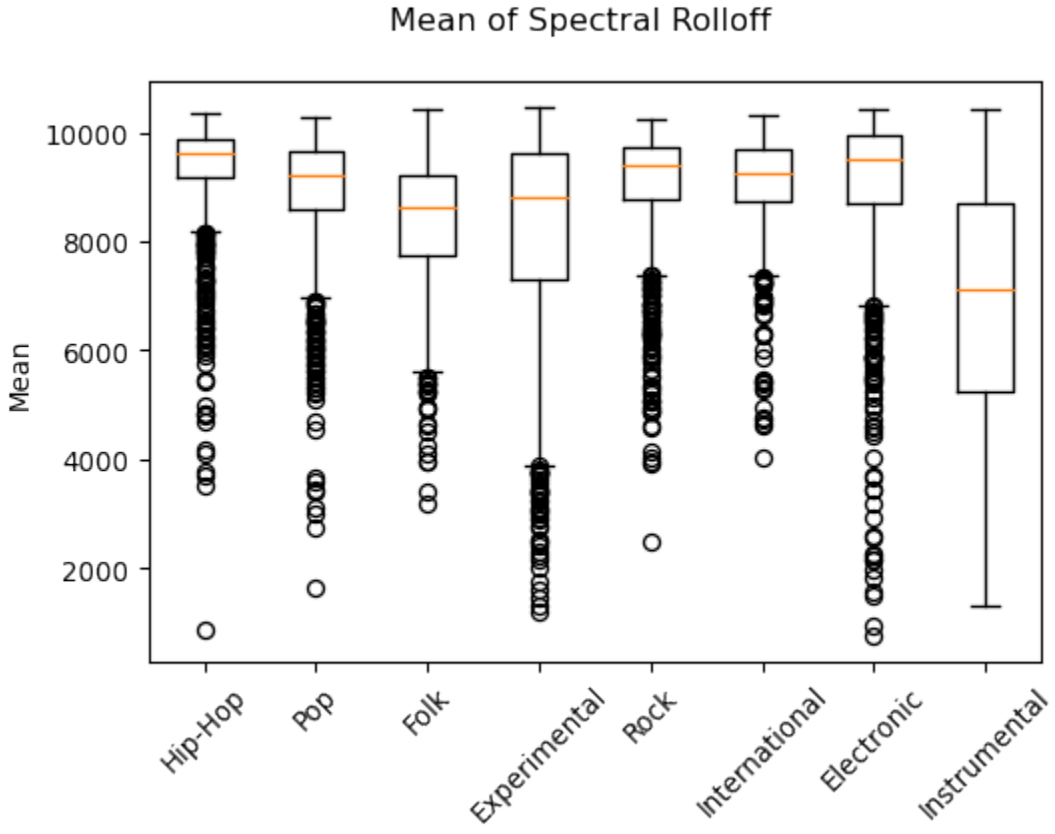
These boxplots fore each genre show the Zero Crossing Rate distribution by genre. ZCR is usually thought of as a good measure to include when doing a genre analysis because it conveys something of the percusiveness of the song. We see that the distributions differ enought to justify including it, but some genres are more drastic than others.



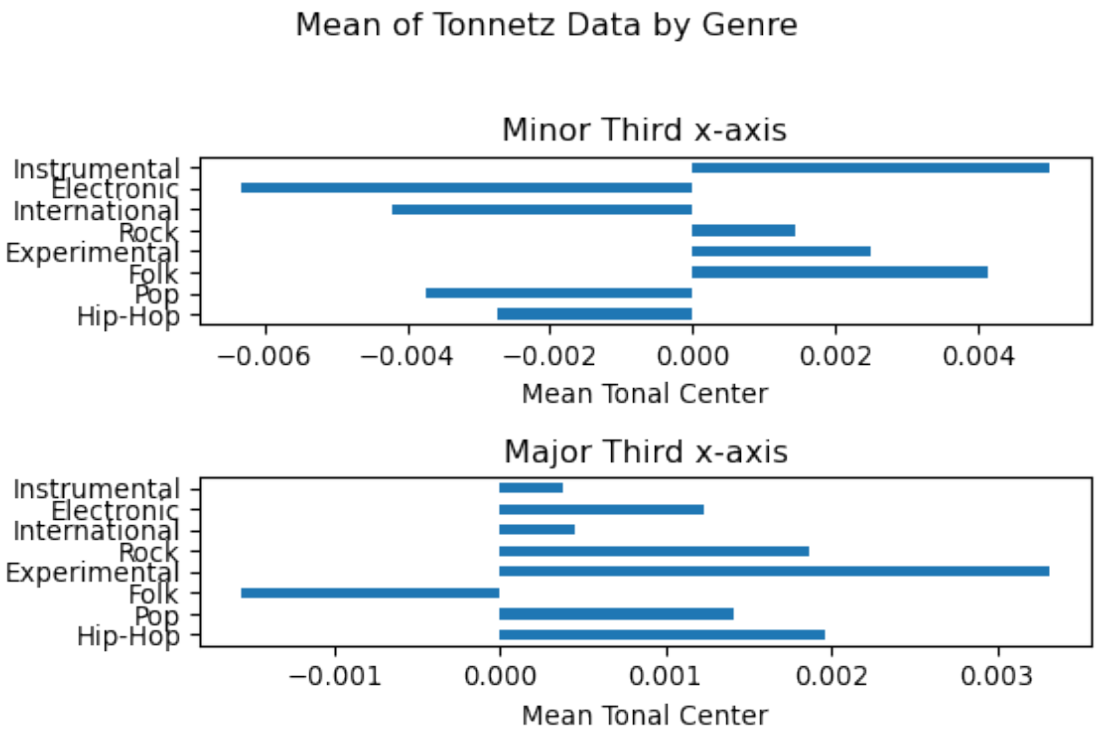
The hexbin plot compares the max frequency and the spectrall rolloff mean. Because the spectrall rolloff mean is the mean frequency greater than 99% of a time frame's frequencies, it make sense that it may be redundant information or colinear with max_frequency.




Variance of First 3 MFCCs by Genre
16000
14000
12000
10000
8000
6000
4000
2000
0
mfcc1
mfcc2
mfcc3
Genre
Electronic
Experimental
Folk
Hip-Hop
Instrumental
International
Pop
Rock


Mean of Spectral Rolloff
Mean
10000
8000
6000
4000
2000
Hip-Hop
Pop
Folk
Experimental
Rock
International
Electronic
Instrumental

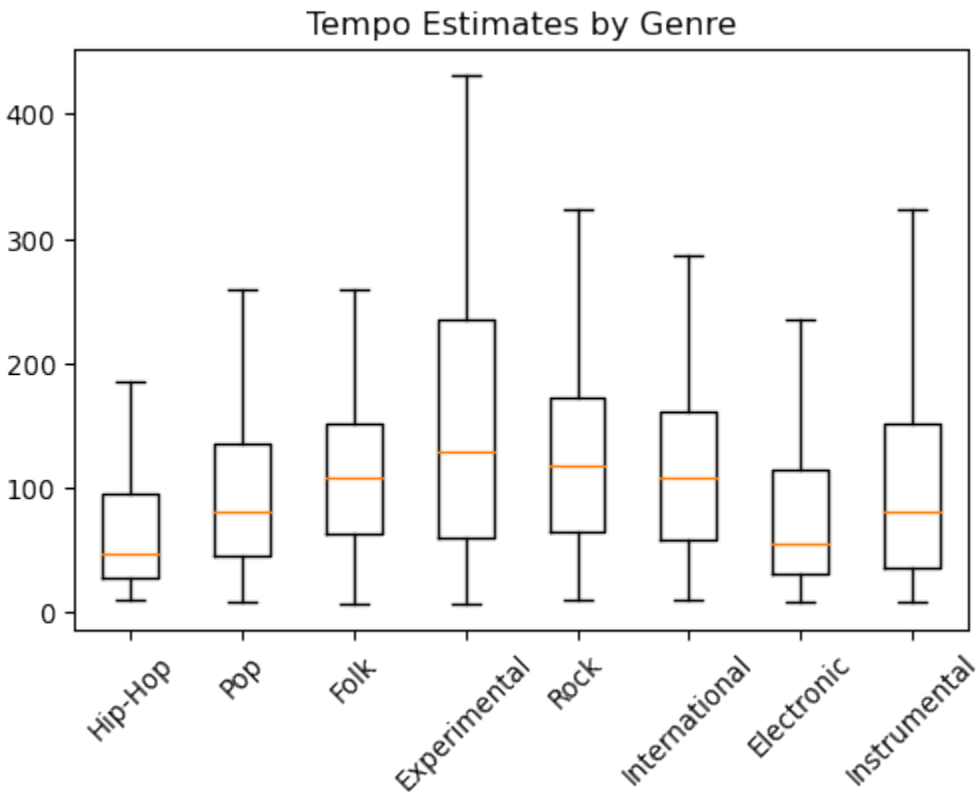
A couple things to note from the plot above are the distributions of the mean spectral rolloff of experimental and instrumental music, which tend to be skewed lower than for other genres.



For each tonnetz, we calculated the mean and variance of the x and y directions for that tonal center for each song. Above are the plots of the averages of two of those means across each genre. We show plots of the major and minor third x-axis means, and much of the other data behaves similarly.

We note that for the minor third, the average tonal center is negative for half of the genres, and positive for others, and the magnitude of each mean changes drastically from genre to genre. In contrast, the major third x-axis mean is positive for all but one genre (folk). Though we plot only two tonnetz features here, many of the other tonnetz features behave similarly, with different tones having postive and negative means that vary by genre. Which tones are positive and negative changes for each tone, indicating that the mean tonal center data could be useful in making decisions between genres.

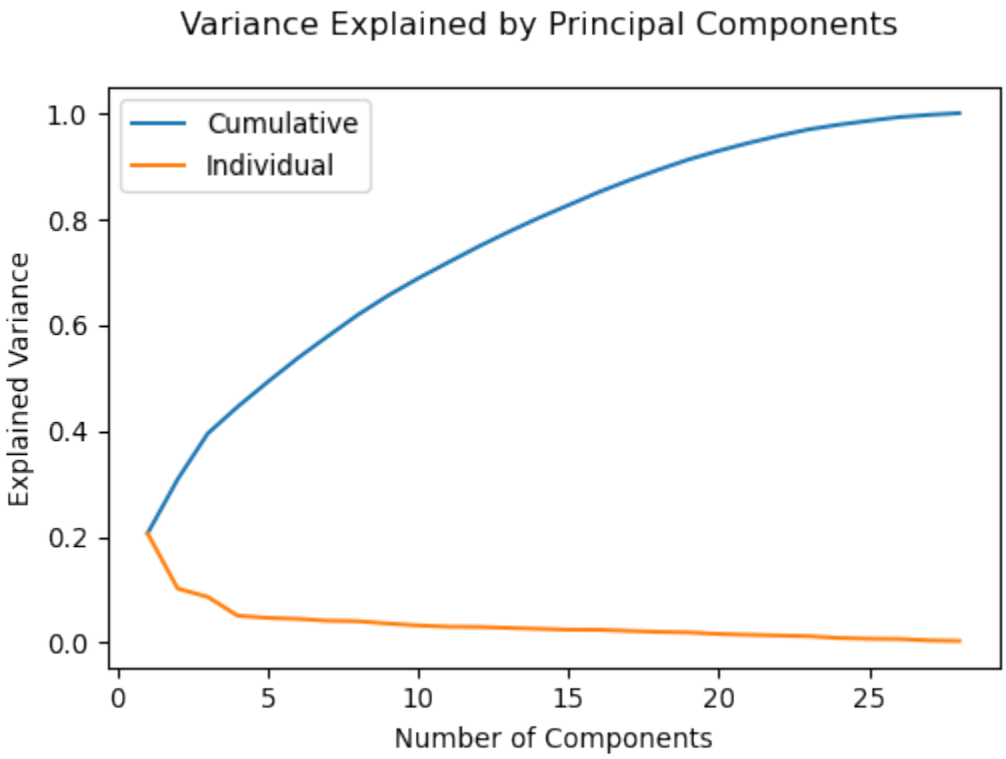
For example, for the major third x-axis mean tonal centers, folk music is strongly negative, while the rest of the genres are positive. Thus, this particular feature could be useful for classifying folk vs. not folk, and a similar idea could be used to interpret the other tonal features.



The tempo estimates are all somewhat similar in shape, in that all are skewed toward the lower end of the tempo ranges and all have outliers in the higher tempo ranges. We do see, however, that electronic and hip-hop songs appear to have a stronger clustering of tempo estimates at the lower/slower end of the spectrum, which could indicate that the tempo data may be useful for classification. But we do note that the similarity of the tempo distributions from genre to genre indicate that this will certainly not be the strongest feature to classify on. This makes sense, since each genre will have a lot of variablity in the tempo of each of its songs.

We have only plotted the first tempo estimate here, but in our data we have the top three tempo estimates for each song, corresponding to the highest three autocorrelation peaks found using the librosa tempo methods. This could result in colinearity in our data without adding much new information, since the distribution for each tempo feature is similar.

We are ignoring the outliers to focus more on the distribution of the tempo estimates; some of the outliers had values as high as 1200. That may indicate that the algorithm failed to pick out a tempo for these songs, or that some of the experimental music doesn't have a tempo.



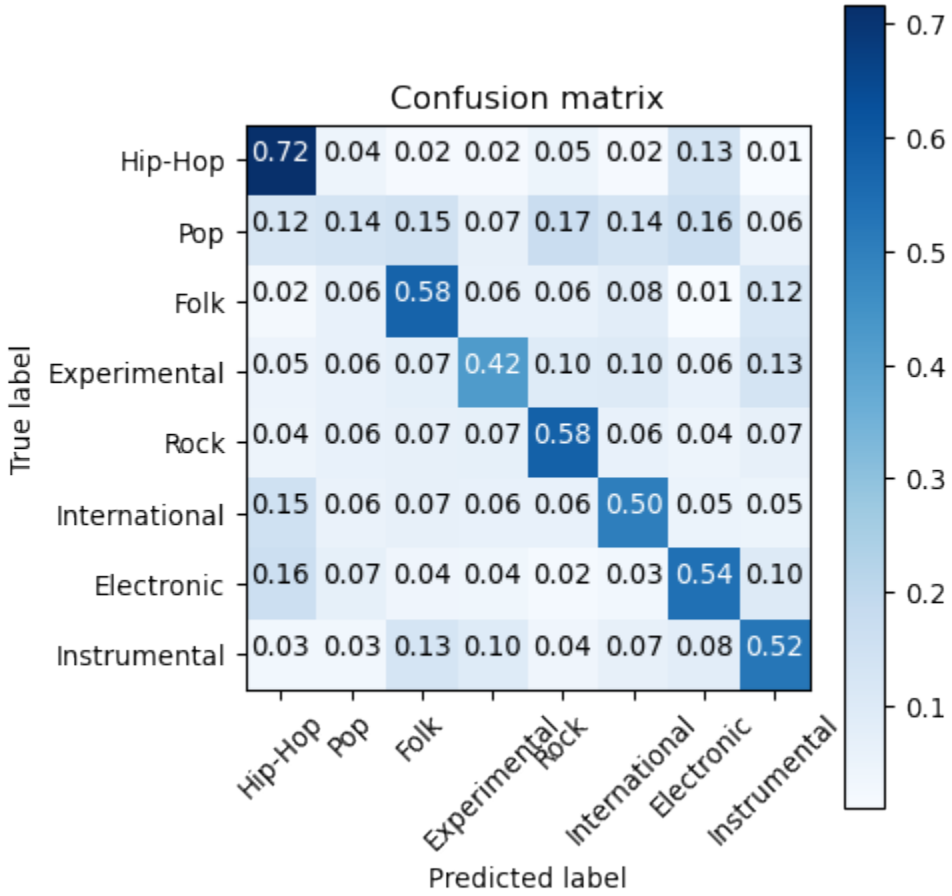
Using principal component analysis, we see that most of the variation in our features (90%) are explained by about 30 components. There is a strong dropoff in the amount of variance explained by each individual component after about the fourth component, seen in the scree plot (orange).

4.2 Models

```
Logistic Regression Accuracy: 0.42910758965804835


XGBoost Accuracy: 0.45954962468723937


RF Accuracy: 0.5
```



4.2.1 Table of Accuracy

Model	Accuracy
Logistic Regression	44%
XGBoost	49%
Random Forest	53%
Multilayer Perceptron	43%
K-nearest Neighbors	40%

Among the models we trained on the features, XGBoost and random forests (with around 1000 trees) had the highest accuracy.

4.2.2 Answering Questions

With the additional information from machine learning are you able to answer any of your reserach questions?

We found that the the most predictive features for the genre of an audio sample are the minimum and maximum frequencies, the first two MFCCs, and the first tempo estimate.

5 Conclusion

Music classification is hard. Deep learning may be useful.

6 Bibliography

(1) G. Tzanetakis and P. Cook. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5):293–302, July 2002.

(2) D. Perrot and R. Gjerdigen, "Scanning the dial: An exploration of factors in identification of musical style," in Proc. Soc. Music Perception Cognition, 1999, p. 88

(3) Mingwen Dong. Convolutional neural network achieves human-level accuracy in music genre classification. CoRR, abs/1802.09697, 2018