Dillon Dugan

**Midterm Project Paper: Music Genre Classification**

Classification of music genres can provide insight of user preferences and tastes in audio to give recommended new music or help aspiring artists to classify their music style. For this project, a dataset of audio files retrieved from a paper on genre classification, “Musical genre classification of audio signals” by G. Tzanetakis and P. Cook. This paper was published in IEEE Transactions on Audio and Speech Processing in 2002 and their dataset was digitally uploaded to http://opihi.cs.uvic.ca/sound/genres.tar.gz. There are 10 genre classification with 100 audio tracks for each genre resulting in 1000 audio files to use for the classifier. According to the paper, each audio track is exactly 30 seconds long and are in 22050 Hz monophonic 16-bit wave format. The audio tracks were collected from various means of recording included their personal CD collection, radio samplings, and microphone recordings over 2000 to 2001 to stimulate multiple ways that audio are perceived.

Downloading the dataset from the source directly yields in tar files and thus needs to extracted using the tarfile package in python. Once the files are ready to be accessed, the use of the librosa python package allowed the files to be processed for feature extraction. Looping over the genres so the target variable is the genre, the features are written to a csv file. The following features were extracted: tempo, harmonics, chroma Short-Time Fourier Transform (STFT), root-square-mean error, spectral centroid, spectral bandwidth, spectral contrast, spectral flatness, spectral rolloff, zero crossing rate, and a set of Mel-Frequency cepstral coefficients (MFCCs). Each feature can help describe the spectrum of the audio file. Tempo describes the pace of the file in beats per minute, harmonics demonstrate a sequence of frequencies that are harmonizing in the file, chroma STFT measures the intensity of the twelve distinct pitch classes, the roots-square-mean is calculated per frame of the audio file, spectral centroid is the center of mass of the spectrum, spectral bandwidth characterizes the variance of the spectrum from the spectral centroid, spectral contrast measures the difference between the peaks and valleys of the spectrum, spectral flatness demonstrates the frequency of peaks and valleys in the audio file, spectral rolloff measures the percentage of frequencies under a given threshold, zero crossing rate is the rate at which the spectrum pass through zero (positive signal to negative and vice versa), and the Mel-Frequency cepstral coefficients (MFCCs) are a set of coefficients that gives the shape of the spectrum, which can describe the timbre of the audio track. For all of these except the tempo, the features were averaged to avoid 30 observations for each feature causing our dataset to have too many independent features.

Initially, the completed data frame with each feature for all 1000 audio files was scaled using StandardScaler package to have a standard deviation of 1. Having the data scaled, it can be split using train\_test\_split with 70/30 split so 70% of the data is used to train the classifier model while the remaining 30% tests the model for accuracy. The training set was used on a RandomForestClassifier and KNeighborClassifier using default settings. The random forest model yielded an accuracy score of 0.6333 while the KNN model was at 0.5833, therefore the random forest has slightly more predictive power and should be hyper-tuned to increase accuracy of the model. Using GridSearchCV with the following parameters: {'n\_estimators':[100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000], 'max\_depth':[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None], 'max\_features':['auto', 'sqrt'], 'bootstrap':[True, False]}, the best parameters were {'bootstrap': False, 'max\_depth': 20, 'max\_features': 'auto', 'n\_estimators': 200}. Even with hyper-tuned parameters, the random forest model was still only 0.6633 accurate in predicting music genre on the testing set although it was 100 percent accurate on the training set, implying the model is good at predicting given data but has difficulty predicting unseen data. Possibly with a change in features or classifier model, there could be a higher accuracy in the testing data.

Overall, the model is better at predicting genre than I thought it would given the subjectivity of genre classification. Possibly with another classifier or some less features/files there could be an increase in accuracy as the accuracy of the training versus testing sets implies the model is currently overfitted. I would also say that some additional genres or more modern audio files might be useful for working on music classification as some genres have evolved over the past two decades. I would also be interested to work with full audio files as 30 second clips could lose critical parts of the song that makes it the genre it is. I decided to use the model with my own custom track created with one of my friends in a record studio around 2019. According to him, the track would be classified under jazz, however, this model decided it fits the features of rock genre. There is a 66 percent accuracy of that classification, however, I thought it was interesting that based on the main artist, the audio track was different genre than the model predicted it to be.