IST 691

Music Genre Classification

Group 6

Mike DeMaria
Noah Goldie
Wei Liao
Nien Tzu Shih

Overview

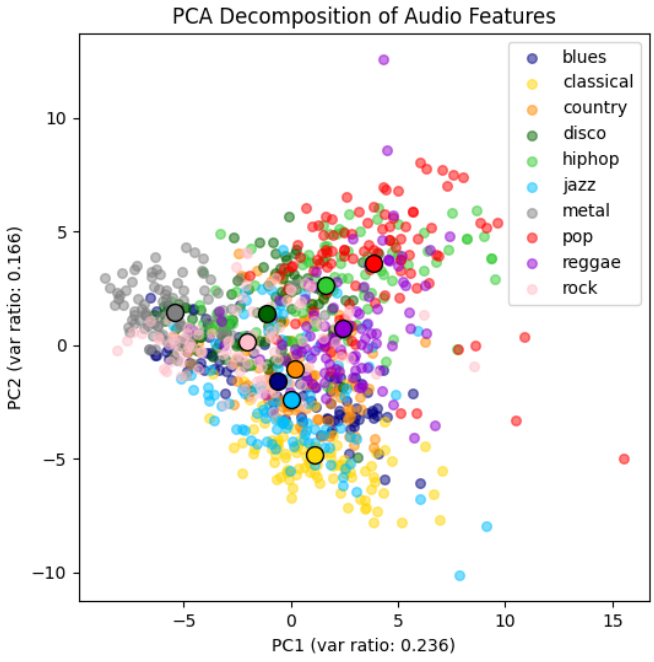
Music can come in a near infinite combination of sounds and styles. Throughout history and across cultures, people have composed music that reflect certain sounds or conventions. Music is grouped into genres, wherein each genre has its own distinct format. Much of music classification is subjective. Our goal is to programmatically identify an unclassified piece of music into a particular genre through the use of deep learning, taking advantage of pre-classified audio clips from https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/data. We wanted to then use this ANN in combination with the RateYourMusic.com Top 5,000 Most Popular Albums dataset (https://www.kaggle.com/datasets/tobennao/rym-top-5000) to also make recommendations on similar sounding albums, based on the genre output from the neural network. This recommendation component is intended to be a simple lookup table.

Prediction Goals

Our goal is to accurately predict which genre a piece of music falls into based on a 30 second audio waveform. Our data source includes 1000 songs that have been pre-classified across 10 genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock.

Data Exploration

The dataset consists of audio files (wav), mel spectrograms (png) and text representations (csv). Each genre is equally balanced, with 100 songs across the 10 genres. We started with a PCA decomposition on the csv.



The pronounced circles in the plot represent the centers of each genre. The first two principal components of the audio summary features account for a combined 40% of the total variance.

Even with less than half of the variance represented, the genres do appear to form rough clusters. Metal and classical appear to be the most distinct from the other genres in this space, while country, jazz, and blues are distributed similarly. This decomposition shows broad patterns from broad features.

We wanted to generate our own test samples, but could not figure out how the data in the csv was generated. However, we were able to determine how to generate mel spectrograms (Librosa). We were unable to determine the settings used to generate the spectrograms included with the downloaded data set. However, we noticed that the ones we generated ourselves were of higher quality. The dataset's images were 336x218 pixels, while ours were 497x370 pixels, exclusive of white space. This is a fidelity improvement of nearly 150%. As such, we opted to re-generate all of the spectrograms from the supplied wav files.

We also discovered a way to split wav files into multiple shorter files (Stack Overflow). We opted to divide the 30 second clips into three 10 second clips, then generate mel spectrograms on these. This, in theory, triples the amount of pre-classified audio samples. In practice, we did not get much utility from this activity.

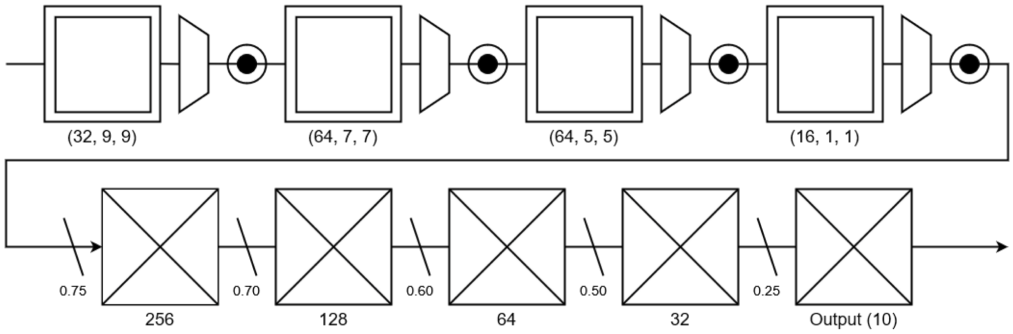
Text Based Models

We started with 5 non-neural network based ML models. The model accuracy for random forest was 79%, decision tree 65%, logistic regression 27%, KNN 28% and SVM 22%. The test and training data was split 80/20. Decision tree and random forest were both poor at identifying rock, while all the models were good at identifying classical. We observed similar results between the 30 second clips CSV and 3 second clips CSV. Despite the 3 second version having an order of magnitude more data to train against, we saw no appreciable difference in accuracy. In fact, some of the 3 second models performed a percent or two worse than the 30 second models. However, we also did not spend a significant amount of time fine tuning these linear regressions.

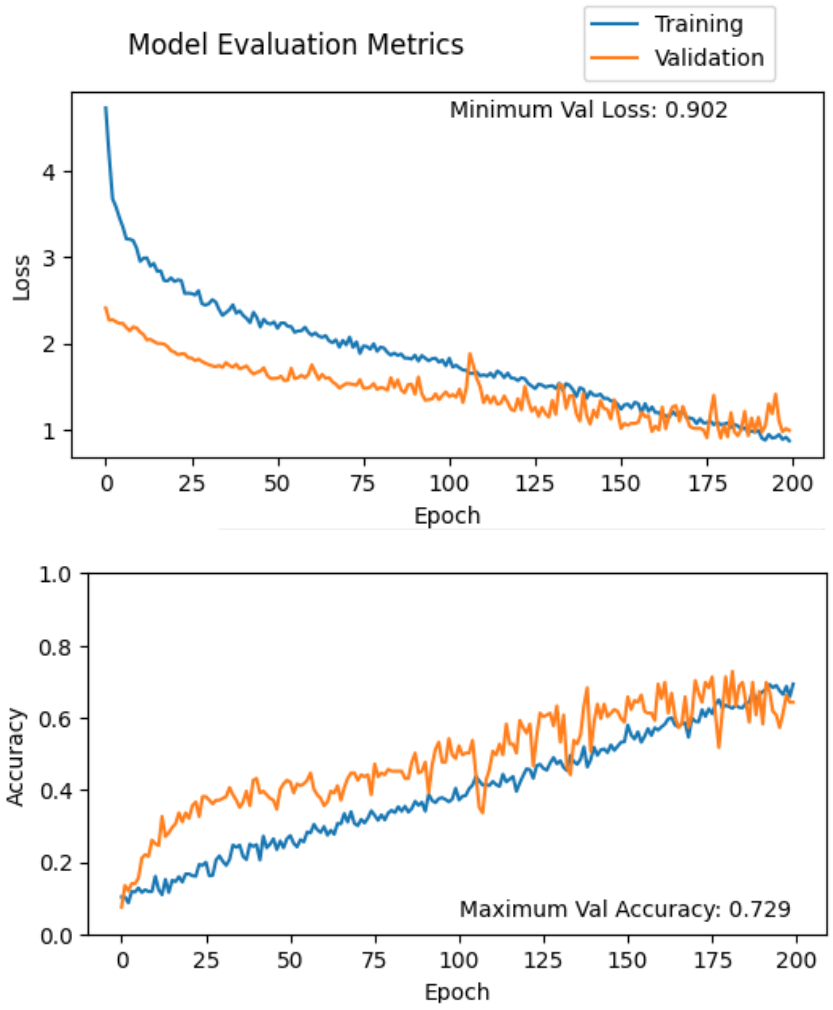
We produced a feed forward neural network against the 30 second csv file. We created a model that contained an input layer with 128 neurons, two hidden layers with 64 and 32 neurons, and an output layer with 10 classes. The hidden layers used ReLU as the activation function, and the output layer used softmax. We used an 80/20 training data split with 10 epocs and a batch size of 32. The model showed an accuracy of 67%.

Mel Spectrogram Models

To process the mel spectrograms, we used a convolutional neural network. The model used multiple convolutional layers with max pooling and a ReLU activation function. Several dense layers were used, each with an ELU activation function. Most notably, we made heavy use of dropouts between the dense layers. Our working theory is that music is heavily repetitive, and by using such a high amount of dropout, we were able to avoid overfitting.



The model was trained with a batch size of 32 and 200 epocs. We saw an accuracy of about 72%, and a minimum validation loss of 0.902.



Results

We wanted to check if our model could detect genres based off the same underlying song. As such, we tested 6 versions of Für Elise, by Ludwig van Beethoven. We found a classical

version, slightly retooled for a full orchestra, as opposed to a piano solo. We also found disco, jazz, reggae, metal and dub step covers. Note that dub step was not part of our 10 genre classifications. This was intentional, as we wanted to see what would happen when we threw a new and unclassified genre at the model.

	Classical	Disco	Jazz	Reggae	Metal	Blues	Country	Hip Hop	Pop	Rock
Classical	61%	12%	12%	0%	1%	6%	4%	0%	1%	3%
Disco	0%	34%	1%	1%	1%	2%	9%	3%	27%	19%
Jazz	0%	5%	32%	0%	0%	5%	3%	44%	2%	5%
Reggae	0%	0%	0%	85%	0%	0%	0%	13%	0%	0%
Metal	0%	1%	0%	0%	93%	0%	0%	0%	0%	4%
Dub Step	0%	1%	0%	12%	2%	0%	0%	25%	59%	0%

Our model correctly predicted 4 of the 5 genres, with jazz being the only notable miss. However, the model's second highest probability for the jazz cover was correct. We noticed that the classical and disco covers had small probabilities across nearly all of the genres. Conversely, reggae and metal had very few. Looking down the spreadsheet's columns, we can see that the classical and reggae categories were pretty clear cut: either a song is classical/reggae or it isn't. From this data, we surmise that many songs will have elements of classical music or disco music, but songs that are actually classical or reggae are quite distinct from each other. While blues and country had small probabilities in this table, none of our 6 Für Elise covers contained strong elements of these genres.

We tested our several other songs. Money by Pink Floyd came up as 33% disco, 29% rock. Our opinion is that this is a blues song, but that is debatable (Epstein). Back in Black by AC/DC came up 38% metal, 31% rock, and Copacabana by Barry Manilow came up 48% disco, 25% pop, which is within our personal expectations. Riders in the sky, generally considered a country song (American Cowboy), came up as 39% rock, 32% disco. This song has elements of rock and country in it. Mundian To Bach Ke, by Panjabi MC, came up as 94% hip hop. This song is slightly unique in that its lyrics are in Punjabi as opposed to English. Finally, Old Town Road came up 45% pop, 25% hip hop.

Problems

We took our best model and re-trained it with the 10 second data. This produced a model with high validation accuracy (over 60%), but did not perform any better with our self selected test songs. It had a tendency to rate everything in one genre, classifying every song as hip-hop or country. We suspect that, as popular music can be inherently repetitive, we were not actually supplying additional datapoints to the model, but repetition of the same data, and thus were overfitting. As the 30 second models were performing very well, and took significantly less time

to train, we abandoned trying to further tune the 10 second models. Perhaps 10 second clips would have performed better with additional tuning.

Another issue found was that the quality of the clips mattered in producing a good outcome. We initially found a low quality clip of Copacabana by Barry Manilow, and the model claimed 74% probability it was a metal song. Copacabana is more along the lines of a disco, tropical or pop song (Discongs), but would not be considered metal. A higher quality version yielded a 48% probability of disco and 25% pop, which is more in line with our expectations.

We ran into memory issues in generating the spectrograms. It appears that Python was not releasing memory after each generation, and thus the program kept filling up all available RAM. We could only generate the spectrograms one genre at a time, otherwise it would slow down tremendously as it filled up all available memory.

We generated spectrograms with two different values for the y axis and color values: one based on frequency amplitude and one based on pitch. Ultimately, both spectrograms performed equally well. Our final model was based on the pitch based spectrograms, but the power/amplitude ones would have worked just as well. Given more time, we could have experimented with additional spectrogram options.

Finally, we were not fully pleased with the recommendation engine. We used the top two genres outputted by the CNN, and used those to recommend similar albums. We did this by selecting 3 albums: the highest average rating, the most ratings left by users, and the most reviews. We intentionally de-duplicated albums that fell into more than one category, meaning the user would see anywhere from 1 to 9 recommendations. However, we noticed that St. Anger by Metallica was listed in the recommendations. This album had a rating of 1.82 / 5.0, and is generally considered very divisive (Louder), so there were a lot of reviews. This component of our solution did not utilize neural networks, so it was not given a high priority. Were we to one day productionalize this project, we would want to beef up this piece.

Summary

We were able to create a neural network that could determine music genre with a reasonable degree of accuracy. This was possible on a fairly small amount of training data. With our train/test splits, each genre only had about 40 minutes of music to train against. That is fairly miniscule, considering tens of thousands of songs are uploaded to spotify every day (Ingham). With additional data, we could train with higher accuracy across even more genres.

Citations

"Librosa.Display.Specshow." Librosa Display Documentation, http://man.hubwiz.com/docset/LibROSA.docset/Contents/Resources/Documents/generated/librosa.display.specshow.html. Accessed 12 Dec. 2023.

"How to splice an audio file (wav format) into 1 sec splices in Python?". Stack Overflow, https://stackoverflow.com/questions/36799902/how-to-splice-an-audio-file-wav-format-into-1-sec-splices-in-python?rq=1. Accessed 12 Dec. 2023.

Epstein, Dan. "Pink Floyd's 'Dark Side of the Moon': 10 Things You Didn't Know". Rolling Stone, 1 March 2018, https://www.rollingstone.com/feature/pink-floyds-dark-side-of-the-moon-10-things-you-didnt-know-201743/

American Cowboy Magazine. "The Top 100 Western Songs". https://web.archive.org/web/20101019002745/http://americancowboy.com/culture/top-100-western-songs. Accessed 12 Dec. 2023 from The Internet Archive as of 19 Nov. 2010.

"Barry Manilow - Copacabana", https://www.discogs.com/release/5981243-Barry-Manilow-Copacabana. Accessed 12 Dec. 2023.

Louder. "St. Anger: A banger or a clanger?". Louder, 20 Sept. 2022. https://www.loudersound.com/features/st-anger-a-banger-or-a-clanger

Ingham, Tim. "Nearly 40,000 Tracks are Now Being Added to Spotify Every Single Day". Music Business Worldwide, 29 April 2019. https://www.musicbusinessworldwide.com/nearly-40000-tracks-are-now-being-added-to-spotify-every-single-day/