Music Genre Classification with Machine Learning

Ian M Fraser

CSc 59929 Intro To Machine Learning

Spring 2019

Objective

The objective of this project is to train and test a variety of machine learning models to properly classify the musical genre of the audio given as input. The idea behind this project was started in my senior design class as a group project, where we are attempting to achieve similar results using deep learning techniques such as Convolutional Recurrent Neural Networks. In this project, I will instead investigate the accuracy of using different models that we have discussed in class such as Logistic Regression, Support Vector Machines, and Random Forests. The features used for these classifiers will be extracted manually with the help of the librosa library in python. The data being used for this project is referred to as the GTZAN dataset, a collection of 1000 audio recordings broken into 10 genres of 100 tracks each. Each mono audio recording in this dataset is 30 seconds in length, with a sample rate of 22050 Hz and bit rate of 16-bits.

Research

The idea behind using machine learning for music genre classification was found during my research on deep learning for my Artificial Intelligence course several semesters ago. In a blog post titled “Recommending music on Spotify with deep learning”, the author Sander Dieleman discusses his work during his internship with Spotify in the summer of 2014. He discusses how he has approached recommending audio for users on the platform by using Convolutional Neural Networks to analyze Mel-scaled spectrograms of the audio recordings to discover patterns within that could be used to find similar features. If successful, this approach could aid in providing users with recommendations not solely based on similarly labeled genres or preferences of other users (as shown in collaborative filtering) but instead provide the user a recommendation based on the features within a particular song that might match certain features in the song that you were listening to.

Although Dieleman’s article was primarily about recommendation, this approach to analyzing music for information sparked my interest in exploring classification of musical genres. In order to discover more information I had to pursue further research in order to understand the technical implementations behind this topic. This lead me to an essay by Perry Cook and George Tzanetakis titled “Music Genre Classification of Audio Signals” published through IEEE in July of 2002. In this essay, the authors go in depth discussing a variety of ways to extract features from audio recordings. These features are broken down into three categories; timbral texture features, rhythmic content features and pitch content features. The specifics of these methods will be discussed in a later section.

To classify the data used in this essay, the authors used a variety of methods such as statistical pattern recognition, Gaussian mixture models and K-nearest neighbor classifiers, which found a general accuracy ranging from 44-61%. Given that this research is 17 years old, I decided to look for more current examples to see if there were other options that might be worth considering for this project.

This lead me to an article by Hareesh Bahuleyan titled “Music Genre Classification using Machine Learning Techniques” from 2018. In Bahuleyan’s research he approached classification using two methods, the first using a deep learning model such as Convolutional Recurrent Neural Networks, the second using the same manually extracted features sited in Cook and Tzanetakis’ research. For the later example, Bahuleyan classified his datasets using Logistic Regression, Random Forests, Gradient Boosting, and Support Vector Machines. The accuracy of these models ranged from around 53- 59% depending on the model. Still not exceptional in performance, but generally a bit better overall. It should also be noted that Bahuleyan used a different dataset of audio than in the previous mentioned research.

Dataset

Now that I’ve discussed some of the research used for this project, I will explain some of the more technical aspects behind the implementation of this work. To start, I will go into some detail about the data being used for this project.

Since the research of Cook and Tzanetakis was one of the most cited bodies of work in most of the research I had found (including some essays not listed here), I decided to use their GTZAN dataset for my classifications. The GTZAN dataset consists of 1000 mono audio recordings, with a sample rate of 22050 Hz at 16-bits. The dataset is divided into 10 genres of 100 audio recordings, each at 30 seconds in length (usually selected from the middle of the track). The genres used in this dataset are Blues, Country, Classical, Disco, Jazz, Pop, Rock, Reggae, Metal, and Hip Hop. The audio recordings in this dataset were compiled using a variety of different methods including directly uploading from CD, to recording off of the radio. This means that the quality of each recording differs greatly, something that I will discuss later when we analyze the results.

Feature Extraction

With a dataset on hand, the question now is how do I find meaningful information from this data to use with my classification models? By themselves, the audio files do not provide any useful information. This is where feature extraction comes in. In Cook and Tzanetakis' article they describe feature extraction as "the process of computing a compact numerical representation that can be used to characterize a segment of audio" (Cook, et al. 2002). The main three categories of features used are timbral, rhythmic and pitch.

Timbral Features

The timbre of an audio recording can be thought of as the general quality or character of the sound. In order to find the timbral features of the audio recordings, a number of computations using short time Fourier transforms (STFT) are calculated for every short-time frame of sound (Cook, et al. 2002). For this project, I have used the same features found in my research.

1.) Spectral Centroid

The Spectral Centroid is defined as the center of gravity of the magnitude spectrum of STFT:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}$$

$M_t[n]$ is the magnitude of the Fourier transform at frame *t* and frequency bin *n*. This can be thought of as the spectral shape. The higher the centroid, the brighter the textures.

2.) Spectral Rolloff

The Spectral Rolloff is another way to measure the spectral shape. It can be described as the frequency below which 85% of the total energy in the spectrum lies.

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

3.) Zero Crossing Rate (ZCR)

Zero crossing rates provide a measure of the noisiness of the signal. It refers to the where the signal changes in sign from negative to positive.

$$Z_t = \frac{1}{2}\sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])|$$

where the *sign* function is 1 for positive arguments and 0 for negative arguments and *x[n]* is the time domain signal for frame *t*.

4.) Root-Mean-Square Energy (RMSE)

The energy of the signal is the total magnitude of the signal. It is a measurement of loudness. The energy can be defined as:

$$\sum_n |x(n)|^2$$

where the root-mean-square energy is defined as:

$$\sqrt{\frac{1}{N}\sum_n |x(n)|^2}$$

5.) Spectral Bandwidth

The *p*-th order moment about the special centroid.

$$(\sum_{k} S(k)(f(k) - f_c)^p)^{\frac{1}{2}}$$

where $S(k)$ is the spectral magnitude at frequency bin k, $f(k)$ is the frequency at bin k, and f_c is the spectral centroid. When p = 2, this is like a weighted standard deviation.

6.) Spectral Contrast

The Spectral Contrast considers the spectral peak, valley, and their differences in each frequency sub-band.

7.) Mel-Frequency Cepstral Coefficients

The Mel-Frequency Cepstral Coefficients (MFCCs) of a signal are a small set of features which concisely describe the overall shape of a spectral envelope. This is done by taking the log-amplitude of the magnitude spectrum, the FFT bins are grouped and smoothed according to the specified Mel-frequency scaling and then applied to a discrete cosine transform to decorrelate the feature vectors.

Rhythmic Features

The rhythm of an audio recording can be thought as the strong or regular repeated pattern within that recording. There are a variety of ways to find rhythmic features from time series data. For example, Cook and Tzanetakis recommend using a combination of techniques for beat analysis such as full wave rectification, low-pass filtering, downsampling, mean removal and enhanced autocorrelation. However, since I was limited on time, I decided to use a tempo based feature extraction that was available in the librosa library.

1.) Beat per minute (BPM)

The beats per minute gives an estimation of the overall tempo of the audio; how fast or slow the music is. To calculate the bpm using the librosa library, I first compute a spectral flux onset strength envelope using librosa.onset.onset_strength. Once the onset is computed, the librosa.beat.tempo is called to find the general tempo by measuring the strength of onsets.

The librosa library returns a single value estimation over the whole track, so no mean or standard deviation was required.

Pitch Features

The pitch of an audio recording measures how the frequency of the music is measured in relation to Western musical practices. For this feature, we used one feature, the constant-q chromagram.

1.) Contant-Q Chromagram.

The constant-q chromoagram uses a logarithmically spaced frequency axis that corresponds the total energy of the frequency axis to a 12 pitch scale used in Western music (C, C#, D, D#, E ,F, F#, G, G#, A, A#, B). The spacing of these pitch notes are similar to our Mel-Scaled spacing used in timbral features.

Feature Extraction Process

With the aid of the librosa library in python, I was able to extract the above features on my audio dataset. To save time and computational resources, I saved the extracted features into a CSV file to be used with my machine learning classification models.

Machine Learning

In order to predict the genres of my audio dataset, I will need to use various Machine Learning classifiers. In this section, I will briefly describe which models I decided to use, how they work, and what their advantages and disadvantages are.

Logistic Regression

Logistic Regression is a statistical method for predicting binary classes. It computes the probability of an event occurrence - so in our case, this means whether a point falls in a particular genre or not. Since Logistic Regression is a binary classifier, we had to use the One vs Rest (OvR) method of classification. In OvR, a separate model is trained for each class predicted and whether that observation is in that class or not. It assumes that each classification problem is independent.

The advantage of working with this model is that it is easy to implement, it has low variance, and provides a probability score for it's observations. The disadvantages are that it doesn't handle a large number of features very well, and requires transformation of non-linear features.

Support Vector Machines (SVM)

Support Vector Machine (SVM) is a supervised learning technique used for both classification and regression. It plots each data point as a point in an n-dimensional space (where n is the number of features) with the value of the feature being the value of a particular coordinate. Classification is performed by finding the hyper-plane that differentiates the two classes the best.

The advantages of this classifier is that it can perform very well with higher dimensions and is very effective when the number of features are more than the training examples. The disadvantages include large processing time for larger datasets, does not perform well with overlapped classes, and selecting appropriate kernels for performance can be tricky.

Random Forests

Random Forest is a supervised learning algorithm that works by building a “forest” of Decision Trees and merging them together to get a more accurate and stable prediction. In a decision tree, each leaf or node, represents a class label while the branches represent conjunctions of features leading to class labels.

The advantage of Random Forest is that it is less likely to overfit due to the random nature, and is extremely flexible and highly accurate. The disadvantage is it requires a lot of resources to compute which therefor can be considered time consuming.

Results

To show the results of my machine learning models, we will look at the accuracy, precision, recall, and F1 score of each of the models performance. In addition, I will show the confusion matrix that shows the number of correctly and incorrectly classified genres.

To understand these tables, we have to look first to some basic terminology that describes these results. First, we need to understand what these accuracy scores are actually representing, which is a series of calculation based on correctly and incorrectly classified predictions. These can be considered the true positives, true negatives, false positives and false negatives.

True positives (TP) are the number of cases in which we predicted correctly. True negatives (TN) is where we predicted incorrectly. False positive (FP) is where we predicted yes but it was not correct, and finally, false negative (FN) is where we predicted false when it was actually true.

The series of scores presented are calculated based on different measurements of these positives and negatives:

The **accuracy score** gives a ratio of correctly predicted observations total observations.

$$\mathbf{Accuracy = TP+TN/TP+FP+FN+TN}$$

The **precision score** gives a ratio of correctly predicted positive observations to the total predicted positive observations.

$$\mathbf{Precision = TP/TP+FP}$$

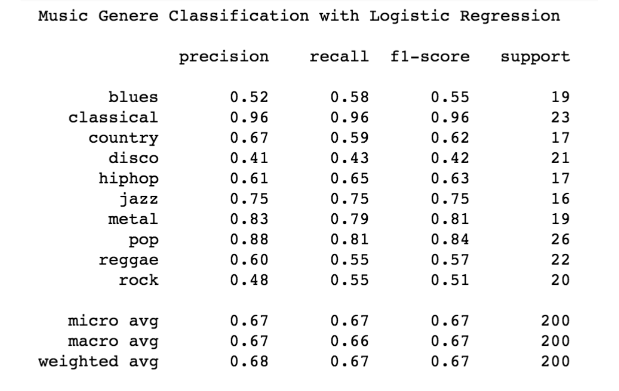
The **recall score** is the ratio of correctly predicted positive observations to all observations in the actually “yes” class.

Recall = TP/TP+FN

Finally, the **F1 score** is the weighted average of Precision and Recall.

F1 Score = 2*(Recall * Precission) / (Recall + Precision)

For logistic regression, we can see the different confusion metrics in the table below. We can see that classical music has one of the highest accuracies in this model, followed by pop and metal. Overall we have a 67% accuracy score.

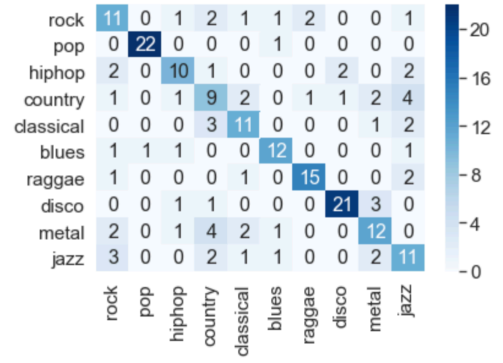


```
Music Genere Classification with Logistic Regression

              precision    recall  f1-score   support

       blues       0.52      0.58      0.55        19
   classical       0.96      0.96      0.96        23
     country       0.67      0.59      0.62        17
       disco       0.41      0.43      0.42        21
      hiphop       0.61      0.65      0.63        17
        jazz       0.75      0.75      0.75        16
       metal       0.83      0.79      0.81        19
         pop       0.88      0.81      0.84        26
      reggae       0.60      0.55      0.57        22
        rock       0.48      0.55      0.51        20

   micro avg       0.67      0.67      0.67       200
   macro avg       0.67      0.66      0.67       200
weighted avg       0.68      0.67      0.67       200
```



	rock	pop	hiphop	country	classical	blues	raggae	disco	metal	jazz
rock	11	0	1	2	1	1	2	0	0	1
pop	0	22	0	0	0	1	0	0	0	0
hiphop	2	0	10	1	0	0	0	2	0	2
country	1	0	1	9	2	0	1	1	2	4
classical	0	0	0	3	11	0	0	0	1	2
blues	1	1	1	0	0	12	0	0	0	1
raggae	1	0	0	0	1	0	15	0	0	2
disco	0	0	1	1	0	0	0	21	3	0
metal	2	0	1	4	2	1	0	0	12	0
jazz	3	0	0	2	1	1	0	0	2	11

Here we can see specifically how each audio file was correctly and incorrectly classified with Logistic Regression. The model correctly classified 22 tracks correctly, and only 1 as blues.

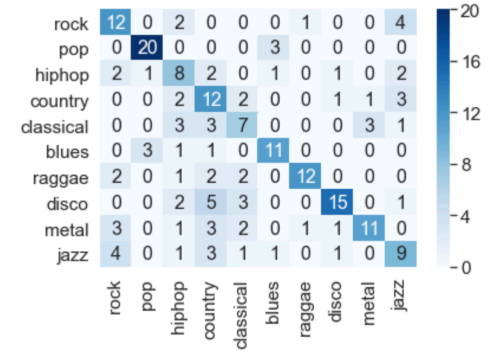
Next we will look at SVM. This model performed the worst out of the three with only 58% accuracy. This is still pretty good however, because it is better than a coin toss.

```
Music Genere Classification with SVM

                precision    recall  f1-score   support

       blues       0.52      0.63      0.57        19
   classical       0.83      0.87      0.85        23
     country       0.38      0.47      0.42        17
       disco       0.39      0.57      0.46        21
      hiphop       0.41      0.41      0.41        17
        jazz       0.69      0.69      0.69        16
       metal       0.86      0.63      0.73        19
         pop       0.79      0.58      0.67        26
      reggae       0.73      0.50      0.59        22
        rock       0.45      0.45      0.45        20

   micro avg       0.58      0.58      0.58       200
   macro avg       0.61      0.58      0.58       200
weighted avg       0.62      0.58      0.59       200
```



	rock	pop	hiphop	country	classical	blues	raggae	disco	metal	jazz
rock	12	0	2	0	0	0	1	0	0	4
pop	0	20	0	0	0	3	0	0	0	0
hiphop	2	1	8	2	0	1	0	1	0	2
country	0	0	2	12	2	0	0	1	1	3
classical	0	0	3	3	7	0	0	0	3	1
blues	0	3	1	1	0	11	0	0	0	0
raggae	2	0	1	2	2	0	12	0	0	0
disco	0	0	2	5	3	0	0	15	0	1
metal	3	0	1	3	2	0	1	1	11	0
jazz	4	0	1	3	1	1	0	1	0	9

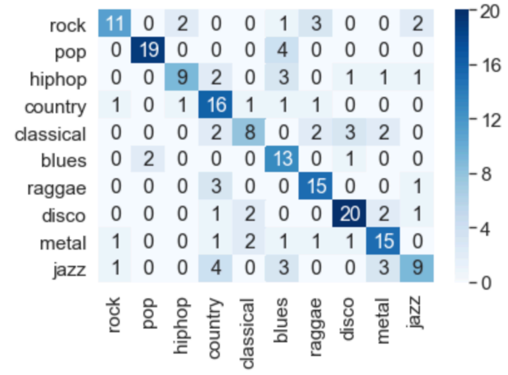
In the confusion matrix we see that out of 14 reggae songs, SVM correctly classified 12 of those tracks.

Finally, with Random Forests, we can see an accuracy of 68%. Just slightly better than Logistic Regression, but not by much.

```
                precision    recall  f1-score   support

       blues       0.79      0.58      0.67        19
   classical       0.90      0.83      0.86        23
     country       0.75      0.53      0.62        17
       disco       0.55      0.76      0.64        21
      hiphop       0.62      0.47      0.53        17
        jazz       0.50      0.81      0.62        16
       metal       0.68      0.79      0.73        19
         pop       0.77      0.77      0.77        26
      reggae       0.65      0.68      0.67        22
        rock       0.64      0.45      0.53        20

   micro avg       0.68      0.68      0.68       200
   macro avg       0.69      0.67      0.66       200
weighted avg       0.69      0.68      0.67       200
```



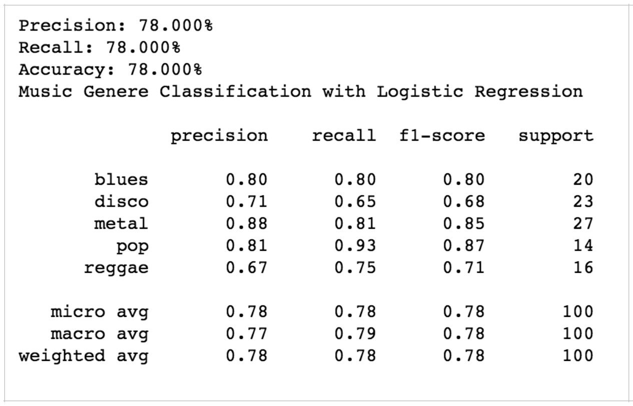
	rock	pop	hiphop	country	classical	blues	raggae	disco	metal	jazz
rock	11	0	2	0	0	1	3	0	0	2
pop	0	19	0	0	0	4	0	0	0	0
hiphop	0	0	9	2	0	3	0	1	1	1
country	1	0	1	16	1	1	1	0	0	0
classical	0	0	0	2	8	0	2	3	2	0
blues	0	2	0	0	0	13	0	1	0	0
raggae	0	0	0	3	0	0	15	0	0	1
disco	0	0	0	1	2	0	0	20	2	1
metal	1	0	0	1	2	1	1	1	15	0
jazz	1	0	0	4	0	3	0	0	3	9

Here, pop has shown to have the same accuracy reggae did in the previous model's prediction.

I was overall pretty unhappy with the accuracy's of these results so I started wondering if the number of classes being tested here too great. The more I thought about this, the more it started to make sense. This is because certain genres like Pop, Rock, and Country all share similarities in their features. It is likely that when applying feature extraction on these genres, that their results were too similar to differentiate from one another.

Because of this observation, I decided to do another run through these models using only 5 classes. In this case, I tested blues, metal, reggae, disco, and pop, 5 genres that have clear distinctions from one another in most cases.

In doing this, the accuracy scores shot up from my previous trials. Logistic Regression tested with a 78% accuracy, SVM with a 75% accuracy, and Random Forest with an 87% accuracy.

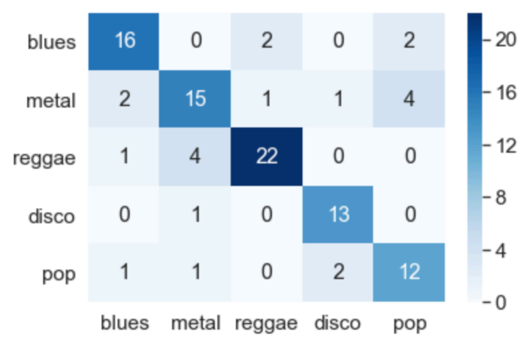


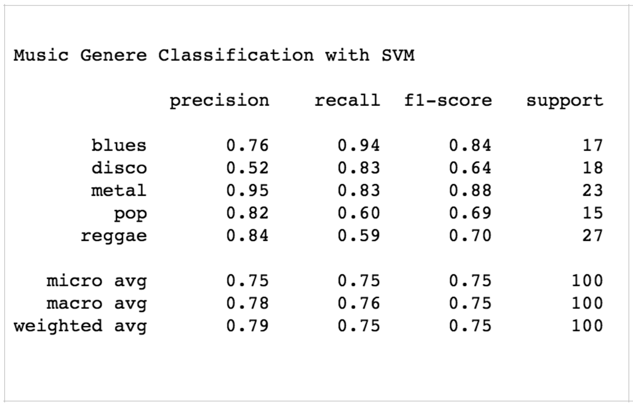
```
Precision: 78.000%
Recall: 78.000%
Accuracy: 78.000%
Music Genere Classification with Logistic Regression

              precision    recall  f1-score   support

       blues       0.80      0.80      0.80        20
       disco       0.71      0.65      0.68        23
       metal       0.88      0.81      0.85        27
         pop       0.81      0.93      0.87        14
      reggae       0.67      0.75      0.71        16

   micro avg       0.78      0.78      0.78       100
   macro avg       0.77      0.79      0.78       100
weighted avg       0.78      0.78      0.78       100
```



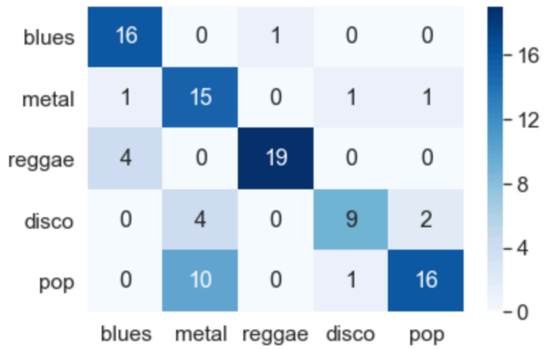


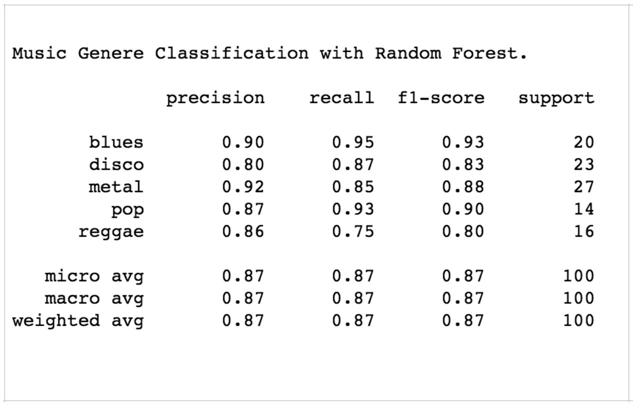
```
Music Genere Classification with SVM

              precision    recall  f1-score   support

       blues       0.76      0.94      0.84        17
       disco       0.52      0.83      0.64        18
       metal       0.95      0.83      0.88        23
         pop       0.82      0.60      0.69        15
      reggae       0.84      0.59      0.70        27

   micro avg       0.75      0.75      0.75       100
   macro avg       0.78      0.76      0.75       100
weighted avg       0.79      0.75      0.75       100
```



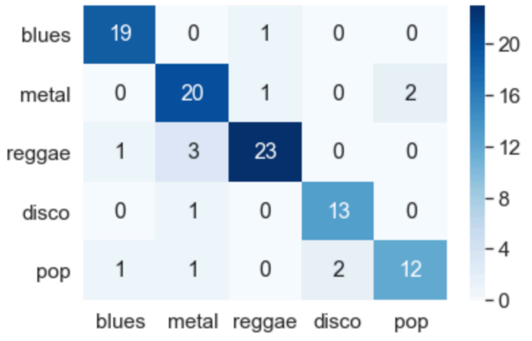


```
Music Genere Classification with Random Forest.

              precision    recall  f1-score   support

       blues       0.90      0.95      0.93        20
       disco       0.80      0.87      0.83        23
       metal       0.92      0.85      0.88        27
         pop       0.87      0.93      0.90        14
      reggae       0.86      0.75      0.80        16

   micro avg       0.87      0.87      0.87       100
   macro avg       0.87      0.87      0.87       100
weighted avg       0.87      0.87      0.87       100
```



Conclusion

If the primary objective of this project was to prove that a computer can accurately predict the genre of music better than a human, than I would have to say the conclusion is a no. There are many factors to consider when decided how to label a piece of music, its history, instrumentation, over all timbre, amongst many other smaller considerations. Certain genres draw influences from many others, which make giving them a concrete classification difficult and likely biased based on the listeners preference and knowledge.

Moving forward, to improve the results of these classifications, I would like to use a larger dataset with more examples of what qualifies for each genre. The GTZAN dataset is known to be flawed. It has inconsistencies in recording qualities, repeated tracks in certain genres, and misclassified data as well. All of these play into the accuracy of my predictions shown in this report. I would also prefer to do additional research on feature extraction, perhaps finding more examples for the categories of pitch and rhythm as I feel that I didn’t go far enough with what I used in this project. Finally, if given the time and proper resources, further explorations of deep learning techniques could be useful as they’ve shown to have higher accuracy in the field of genre classification.

Overall, this was an interesting topic to pursue. I hope to continue my research to learn more about Music Information Retrieval to learn more about how to analyze music for data. This project was useful to see how different machine learning models work with a somewhat larger dataset, and given me an opportunity to compare their accuracies and analyze how they could improve.

REFERENCES

Bahuleyan, Hareesh. *"Music Genre Classification using Machine Learning Techniques."* University of Waterloo, ON, Canada. April 2018, https://www.researchgate.net/publication/324218667

Cook, Perry, and George Tzanetakis. *"Music Genre Classification of Audio Signals."* IEEE Transactions On Speech and Audio Processing, VOL. 10, NO. 5, JULY 2002

Deileman, Sander. *"Recommending Music On Spotify with deep learning."* Web. Aug. 2014, http://benanne.github.io/2014/08/05/spotify-cnns.html

Tjoa, Steve. "*Music information retrieval."* Web. https://musicinformationretrieval.com/index.html Accessed 03 April 2019.

Tzanetakis, George, Perry Cook and George Essel. *"Automatic Musical Genre Classification of Audio Signals."* Web. 2001, http://ismir2001.ismir.net/pdf/tzanetakis.pdf