Tennessee Technological University
College Of Engineering

Final Project

Student names: *Joshua Adams, Weston Beebe, Parth Patel, Jonathan Sanderson, Samuel Sylvester*

Course: *Signal Analysis (ECE 6040)* – Professor: *Dr. Corey Cooke*
Due date: *May 4th, 2023*

Abstract

In this work, we used various Machine Learning methods in order to perform music identification on highly noisy recordings of various songs. To do this we used spectrograms and Power Spectral Density (PSD) to extract features from the recordings and sent the extracted features through several supervised machine learning models including, K-Nearest Neighbors, Naïve Bayes, Decisions Trees, and Random Forests. Using these, we found that the best performing models were Random Forests and Decision Trees with a testing accuracy of 99.8% and 96.2% respectively. With this, we believe we have successfully implemented a machine learning approach to music identification.

1 Introduction

This work explores the effectiveness of 10 different methods of music identification using machine learning. To reduce the dimesnionality of the sample songs, two different feature extractors were used. The first feature extractor tested is the spectrogram. Spectrograms are a representation of the frequency spectrum of a signal with respect to time. Spectrograms can be used to extract the characteristic features of a song. The pitch pattern of a song is identifiable as a repeating sequence of components in the spectrogram. The second feature extractor tested is the power spectral density (PSD) of the signal. The PSD and spectrogram of a signal are related in that they both analyze the frequency content of a signal. However, the PSD doesn't provide information on how the frequency of a signal changes with time. The PSD provides a distribution of energy across the different frequency components. This work chooses the 8 frequency bins with the highest PSD and inputs those bins into different supervised learning classifiers.

In combination with the 2 feature extractors, 5 different machine learning classifiers are used to classify each song. The supervised machine learning models used include: K-Nearest Neighbors, Gaussian Naïve Bayes, Decision Trees, Random Forests, and a boosted version of decision trees using Ada Boost. These models were selected using nested cross-validation. To gauge the performance of each model, the accuracy and F1 scores are calculated. The two methods with the highest accuracy were found to be the Random Forests and the Decision Trees models.

2 Background

Machine learning has shown to be practical in audio classification. Audio classification has used machine learning in some form for more than two decades. In 2002, genre annotation was done by extracting timbral texture, rhythmic content, and pitch content and training a statistical pattern recognition classifier [1]. In 2003, Shazam achieved song identification by computing the spectrogram of a song, using the spectrogram peaks to extract a fingerprint, and comparing the fingerprint to a database of songs [2]. Deep learning specifically has been used in genre designation [3] and cover song identification [4]. Machine learning has been used broadly in audio classification outside of music too, from biomedical sound identification to speech recognition [5].

There are many ways that song identification could be achieved with machine learning. The audio signal could be analyzed to extract different features, and different models could be trained to classify such features. The features and classifier should be chosen to yield the highest accuracy.

The features extracted from the audio signal must be highly immune to noise and distortion since a real-world signal will have these. The spectrogram is one potential candidate for identifying a song as the prominent frequencies in a song can be seen on top of the noise. The spectrogram is a time-frequency analysis technique that utilizes the short-time Fourier transform to show frequency components of the signal across time. Power spectral density is another candidate. The power spectral density is simply a way to break the signal power into frequency components, and it can be computed by taking the Fourier transform of the signal's auto-correlation function [6] [7]. Both the spectrogram and power spectral density provide methods of analyzing the frequency components of an audio signal.

The classifiers should be accurate to the test data. A few classification methods include support vector machines (SVMs), nearest neighbors, naive Bayes, decision trees, and ensemble methods. These all work a little differently, but they are all supervised machine learning methods capable of classification. Ensemble methods are unique in that they combine and capitalize on the strength of various individual classification methods. Ensemble learning can work in different ways. Bagging, or averaging, is an ensemble method that parallelizes classifiers and takes the decisions of the classifiers as a vote. One example of a bagging ensemble is the random forest. Boosting is a method that serializes classifiers, and the classifiers are trained more heavily on the harder to classify examples. One implementation of a boosting ensemble is AdaBoost [8].

Song identification with machine learning can be implemented with various combinations of features and classifiers. Only a few features and classifiers are introduced here, but these can be used to implement a simple song identification algorithm.

3 Methodology

The general flow of the solution is shown in Figure 1. After loading the data, feature extractors (FEs) are used to obtain features from the song snippets to be inputs for the models or classifiers. Then, model selection is performed to determine the best-performing models. These models are further tuned and then tested on the test dataset.

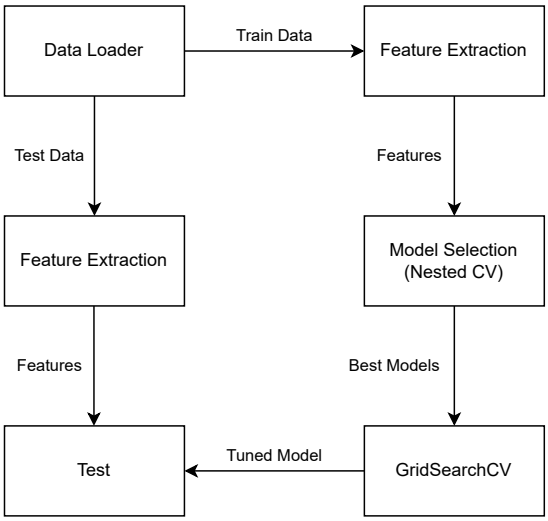


Figure 1: Project flow

3.1 Feature Extraction

Each sample in the training dataset contained 80,000 time steps. This is undesirable as it is computationally expensive and inefficient. We explored two main methods for extracting features and reducing dimensional space of the input.

3.1.1 Method 1: Spectrogram

To compute the spectrogram of the input data, we used the SciPy implementation of the Short Time Fourier Transform (STFT). Through an iterative process, we settled on using an FFT of size 2048. The result was 1025 frequency bins and 80 time bins. To reduce the dimensionality and remove noise at each time bin, we calculated the frequency with the maximum amplitude. The result was a vector of size 80, each element having the value of the frequency with the highest amplitude in the spectrogram.

3.1.2 Method 2: Power Spectral Density

To compute the power spectral density of the input data, we used the SciPy implementation of Welch's method. Welch's method was chosen over the periodogram since it reduced the noise in the power spectral density. With the chosen segment length of 4096, the result was power spectra across 2049 frequency bins. The 8 bins with the highest power spectral density are picked. The result was a vector of size 8, with elements having the frequencies with the highest power spectral density.

3.2 Model Selection

Table 1: Classifier candidates

Classifier
K-Nearest Neighbors (KNN)
Gaussian Naïve Bayes (GNB)
Decision Tree (DT)
Random Forest (RF)
Ada Boost + DT

For model selection, nested cross-validation was utilized to determine the top-performing models. Nested cross-validation gives a general idea of how the models will perform and generalize to the dataset. As the name suggests, nested cross-validation splits the training data into a number of folds with the outer cross-validation and further splits the data with the inner cross-validation. A visualization can be seen in Figure 2. The validation scores obtained from nested cross-validation are then averaged for each model and ranked in decreasing order. The potential models used during this step are listed in Table 1. After determining the top models, these models underwent hyperparameter-tuning using GridSearch Cross-validation with the entire train dataset. GridSearchCV finds the best hyperparameters by exhaustively searching through a search space of given parameters to find the best-performing model on a particular cross-validation fold.

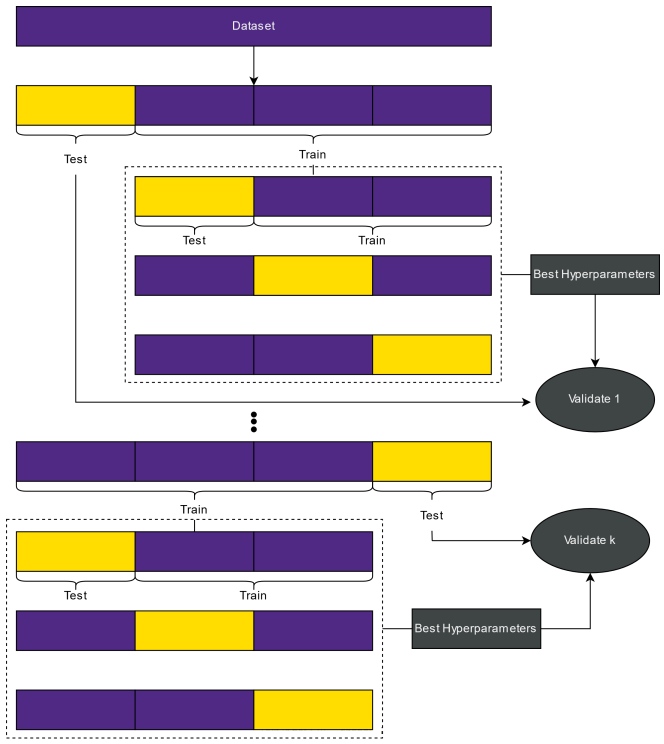


Figure 2: Nested Cross-validation

3.3 Testing

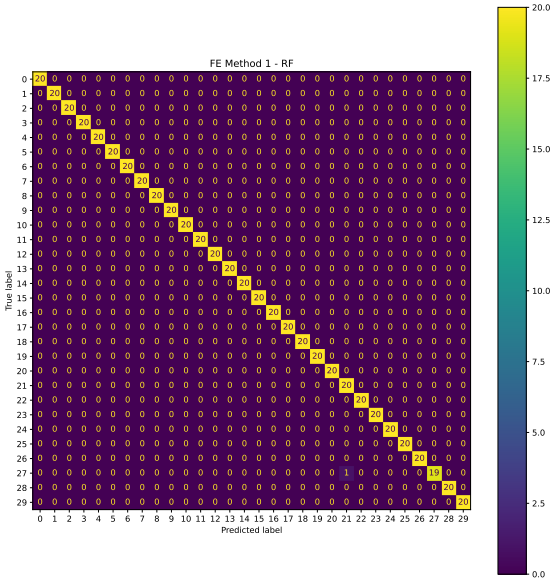
After obtaining and tuning the best models, the models were tested on the test dataset. To gauge how well each model did, the accuracy and F1 scores were calculated while also plotting a confusion matrix of the predictions. For reference, the equations to calculate accuracy and F1 are:

$$Accuracy = \frac{TP + TN}{P + N} \tag{1}$$

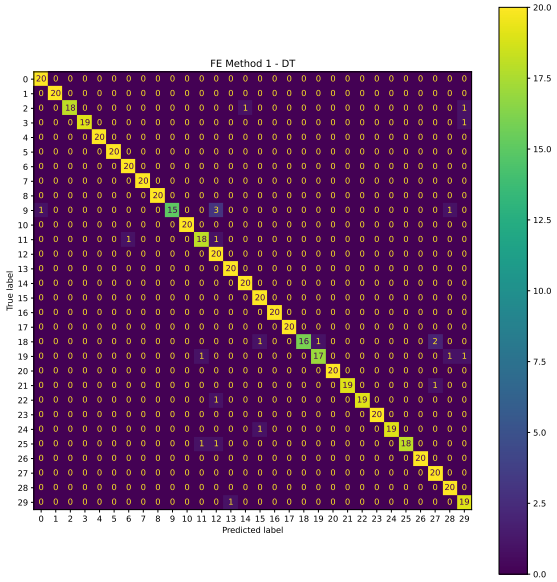
$$F1 = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{2}$$

4 Results

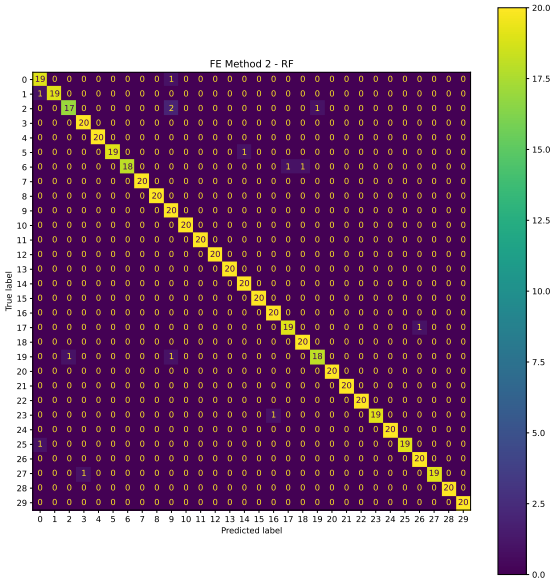
In Table 2 the F1 score of the model candidates for methods 1 and 2 can be seen. These were averaged for each model and the best two models with the highest score we selected to run on the testing dataset. The results of inferencing on the testing dataset can be seen in Table 3. In addition, the confusion matrices of the two models evaluated on both feature extraction methods is provided in Figure 3.



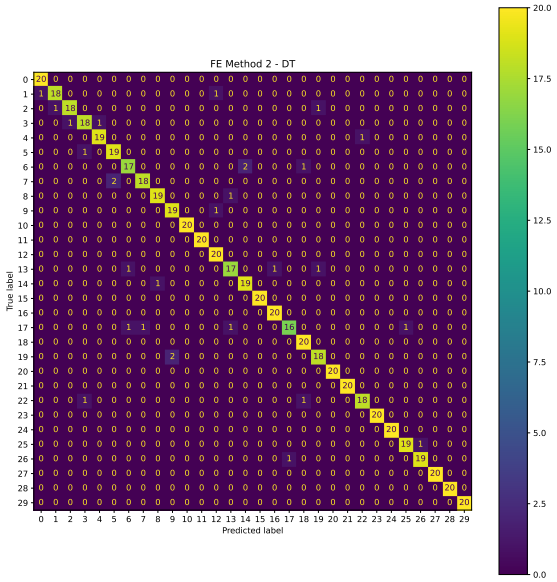
(a) FE method 1 with Random Forest



(b) FE method 1 with Decision Tree



(c) FE method 2 with Random Forest



(d) FE method 2 with Decision Tree

Figure 3: Confusion matrixes of both feature extraction methods with the two best models

Table 2: Results of nested cross-validation.

Model	Method 1 F1 Score	Method 2 F1 Score
KNN	0.978	0.912
GNB	*0.098	0.830
DT	0.976	0.939
RF	1.000	0.978
Ada	0.971	0.940

*Encountered multiply overflow error

Table 3: Accuracy and F1 of tuned models on testing dataset

	Metric	RF	DT
Method 1	Accuracy	0.998	0.962
	F1	0.998	0.961
Method 2	Accuracy.	0.977	0.952
	F1	0.977	0.951

5 Discussion

As mentioned in the Methodology section, we tested five different models. Most of these scored high in the nested CV step. When the best models were evaluated on the test dataset, Random Forest achieved a score greater than 97% across both FE methods and Decision Tree achieved a score greater than 95% on both FE methods. The best accuracies being 99.8% and 96.2% for Random Forest and Decision trees, respectively. The confusion matrices in Figure 3 show Random Forest and Decision Tree are performing well on all categories in the testing dataset. The goal of achieving an accuracy greater than 90% has been achieved.

6 Conclusion

This paper successfully shows applications of machine learning models to perform song identification. By using Random Forests and Decision Trees, we were able to successfully identify the recorded songs 99.8% and 96.2% of the time. This was done through extracting spectrograms and Power Spectral Density from the unknown song sample to compare to a bank of song samples for comparison. Random Forests was especially effective due to employing ensemble learning. These methods have shown to be effective, and could potentially be expanded onto a larger song bank to become a more robust song identifying software.

7 References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[2] A. Wang *et al.*, "An industrial strength audio search algorithm.," in *Ismir*, vol. 2003, pp. 7–13, Washington, DC, 2003.

[3] W. Hongdan, S. SalmiJamali, C. Zhengping, S. Qiaojuan, and R. Le, "An intelligent music genre analysis using feature extraction and classification using deep learning techniques," *Computers and Electrical Engineering*, vol. 100, p. 107978, 2022.

[4] Z. Yu, X. Xu, X. Chen, and D. Yang, "Learning a representation for cover song identification using convolutional neural network," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 541–545, 2020.

[5] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra, "Data augmentation and deep learning methods in sound classification: A systematic review," *Electronics*, vol. 11, no. 22, p. 3795, 2022.

[6] C. M. Spooner and R. B. Nicholls, "Chapter 18 - spectrum sensing based on spectral correlation," in *Cognitive Radio Technology (Second Edition)* (B. A. Fette, ed.), pp. 593–634, Oxford: Academic Press, second edition ed., 2009.

[7] J. Dempster, "Chapter six - signal analysis and measurement," in *The Laboratory Computer* (J. Dempster, ed.), Biological Techniques Series, pp. 136–171, London: Academic Press, 2001.

[8] C. Yi-Tung, *An Introduction to Approaches and Modern Applications with Ensemble Learning*. Computer Science, Technology and Applications, Nova, 2020.

Appendix