PROCESS ON GTZAN: MUSIC SIGNAL FEATURE EXTRACTION AND MATCHING WITH RANDOM FOREST AND K-NEAREST NEIGHBORS

Author:
Louis Widi Anandaputra (22/492218/PA/21090)

DEPARTMENT OF COMPUTER SCIENCE AND ELECTRONICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
GADJAH MADA UNIVERSITY

2024

I. Introduction

The use of digital signals had undergone a long development to reach its current state in the twenty-first century. The conversion of analog signals into digital signals through the process of quantization and sampling had enabled signals to be used for various purposes. A common digital music streaming service Spotify, had listed more than 100 million songs are available in its system [1], in which those songs were stored in a digital format acquired from sampling and quantizing the analog signals of the music itself. This would indicate the further use of digital signals are interesting to be explored.

Various services such as Apple Siri, Google Home, and Amazon Alexa used a method of speech recognition in its application. It tried to acquire information and features from signals to determine what words are being spoken by the users and incorporate further natural language processing (NLP) steps to provide feedback and service. Other services such as Shazam are able to identify songs by asking the user to put the music they are listening into the microphone. One commonality of these services is that a classification system is used to provide service, either it's classifying spoken words or samples of music. Therefore, it is important to understand the process behind classifying the signals.

This work would try to understand the process of 1-dimensional signal classification through classifying music of 10 different genres. The process would involve steps such as acquiring the array of the sample and quantized signals, followed by extracting information (features) from the signals and representing it in the form of a tabular data, which then it would be matched by using a classifier. The classifier being used would take the form of a slow learner machine learning model. This document would present the work on solving a music genre recognition (MGR) classification problem.

II. Method

The method used to tackle this MGR problem would be first by gathering the dataset, where a dataset consists of 10 different music genres with each genre having 100 audio file samples, present in the GTZAN dataset [2]. After acquiring the dataset, the audio files are read and converted into an array of amplitudes in respect to time, which would then be followed by extracting both time and frequency domain features of the signal. Those features are then represented in a tabular representation with each column representing different features alongside 1 target column consisting genres of different music samples. Lastly, the process would involve a model learning from the tabular data in constructing a predictive classification model for predicting future music genres.

A. Dataset

The GTZAN dataset consists of 10 different genres with each genre having 100 audio file samples, creating a total of 1000 audio samples dataset [2]. Each sample would take the form of 30 seconds audio files and the 10 genres able to be classified through using this dataset are: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock. The dataset also contains the extracted spectrogram and was used by another researcher to classify the genres through convolutional neural network (CNN)-based deep learning architectures [3]. Therefore, the audio files of this dataset would have its features to be extracted through acquiring the values of the digital signals.

The GTZAN dataset is also used by more than 100 researchers throughout the first decade of the twenty-first century [4]. Some would also argue due to the small number of samples, this dataset is not suitable for real-life implementations, some others would also argue regarding the consistency of the dataset itself [5]. Therefore, this dataset would be used as a benchmark dataset in this work.

B. Extracted Features

The extracted features are divided into two domains, the time domain features and the frequency domain features. The time domain features values measured in response to time, consisting of Zero Crossing Rate (ZCR), Root Mean Square Energy (RMS), and Tempo. While on the other hand the frequency domain features are values measured in response to the frequency. This feature is acquired through a fast fourier transform (FFT) operation and would consist of Chroma of Short Time Fourier Transform (C-STFT), Spectral Centroid, Spectral Rolloff, Harmonicity, Percussiveness, and the first 25 Mel-Frequency Cepstral Coefficients (MFCC).

In each of the 30 seconds audio file, the sampling rate being used would have the value of 22050 Hz, meaning that there are 22050 samples per second and a total of 661500 samples in one audio file being represented. Then, in each of our processing, the data is split into frames with each having 512 samples, meaning the data are split into

1292 frames. This is done due to the fact that the duration of the audio samples are long and therefore the average and standard deviation of each feature except for the tempo are being represented rather than directly measuring the feature values of all of the 661500 samples. Therefore, there would be a total of 67 features and 1 target genre label being used.

Time Domain Features

a. Zero Crossing Rate

As the amplitude of a signal consists of negative and positive values, the ZCR describes the rate at which the signal crosses from positive value to negative value and from negative to positive value. The ZCR of each frame is explained by (1) as N would be the number of samples and sign would have the value of 1 if the value of the sample at position n is positive and 0 otherwise. The ZCR for each frame is then represented through the average and standard deviation to represent the whole 30 second audio file ZCR.

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

b. Root Mean Square Energy

As several other sources may utilize the regular average of energy or amplitude, the RMS is beneficial as it would avoid the cancellation of positive and negative values through squaring. By also taking the root, the unit would be the same as the original. As described by (2), the RMS value is the square root of the sum of squared energy *x* divided by *N* number of samples.

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

c. Tempo

Regarded as the speed of the music being played, tempo is measured in beats per minute (bpm). This feature can represent how fast the different music from different genres are. The tempo is calculated by first measuring the strength of the onset where usually onset is an indicator at which an increase to the signal’s amplitude happens [6]. As correlation of the onset is able to be acquired, the tempo is then able to be calculated. This feature would then be able to explain when distinctive beats are showing, leading to uncovering how fast the music is.

Frequency Domain Features

a. Chroma

The chroma feature can be represented in a chromagram acquired after performing a Short Time Fourier Transform. It is a feature that shows the *pitch class profile* where the main idea is that there are two different pitches that can be regarded as pitches belonging to the same chroma[7]. The pitches are color coded to show different intensity. This feature would be able to represent the music on what are the dominant pitch classes. Figure 1 would show the comparison between the pitch profile between a sample of the pop genre and the classical genre music where one key difference is there is less change of pitch class intensity in classical music than in the pop music over the course of the duration. The average and standard deviation value of all the 13 pitches would then be taken as a feature representation on the data to be used for the feature matching process.

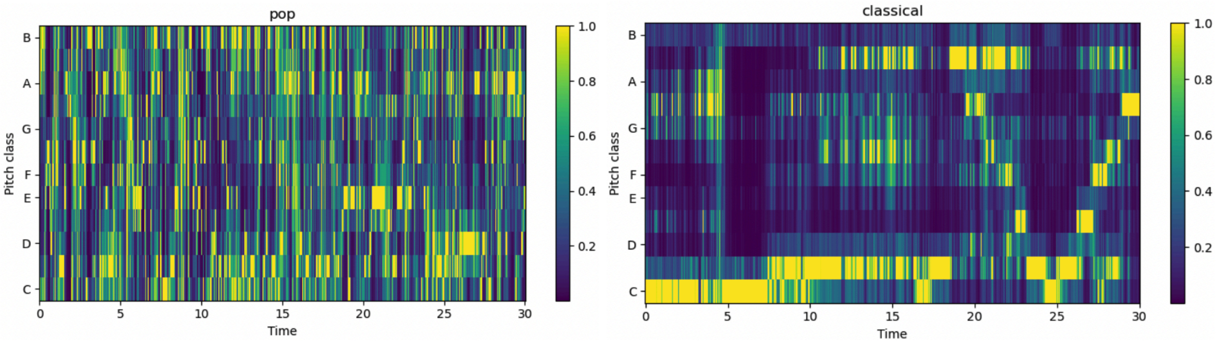


Figure 1. Comparison of a sample pop and classical music chroma feature.

b. Spectral Centroid

As the signal is divided into frames, the spectral centroid as an energy distribution measure is regarded as the point at which the most energy is centered around. It is defined as the average frequency weighted by amplitudes, divided by the sum of the amplitudes [8] as described by (3) where F[k] would be the amplitude in respect to bin k in the DFT spectrum.

$$Spectral\ \ Centroid\ = \frac{\sum_{k=1}^{N} kF[k]}{\sum_{k=1}^{N} F[k]} \tag{3}$$

c. Spectral Bandwidth

The spectral bandwidth is the width of the band at which the intensity is more than a certain threshold. Such a threshold may come from half of the maximum intensity or come from a conventionalized value such as 3 dB. This method is done to remove noise as it gives the threshold to how much decibels is needed for the data to be regarded as signals. As shown in Figure 2, the spectral bandwidth value measures the different wavelengths (which corresponds to the frequencies) range from λ_1 to λ_2.

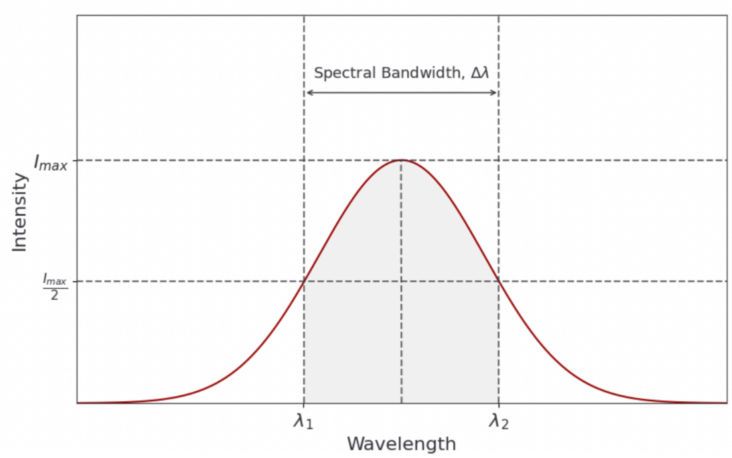


Figure 2. Calculation of the bandwidth value (image source: [9]).

d. Spectral Rolloff

The spectral rolloff feature is regarded as the feature that describes the frequency below which a specified percentage threshold of the total spectral energy resides. Usually, the threshold for the amount of total spectral energy percentage is 85%.

e. Harmonicity

The harmonicity feature would describe the existence of dominant frequency. Meaning, for all frequencies present in the signal, there is a fundamental frequency. For example, the signal would consist of 400 Hz, 800 Hz, 1200 Hz until an indefinite sample, the fundamental frequency would be regarded as 400 Hz.

f. Percussiveness

In contrast to the harmonic feature, the percussiveness of the music correlates highly to the timbre, describing the characteristic of the music. This percussiveness is a sharp attack-decay characteristic of the music and can be acquired alongside the process of acquiring the harmonicity.

g. 25 Mel-Frequency Cepstral Coefficients

The MFCCs are coefficients which describe the overall shape of the spectral envelope. It is a feature computed logarithmically and would consist of several different coefficients. The number of coefficients on existing studies are between 12 and 13 but it is agreed upon that the performance of the classification process would be better with more coefficients [10]. A research yielding higher accuracy used a number of 25 MFCCs, therefore this work would also implement the first 25 MFCCs as part of the feature.

C. Feature Matching

In the feature matching process, slow learner machine learning models are incorporated into training based on the extracted data and are compared with each other. The slow learners used are Random Forest (RF) and K-Nearest Neighbors (K-NN). By splitting the dataset into 75% training data and 25% validation data, a 10 fold cross-validation method would also be used in the model selection process. In evaluating the performance result, the F1-Score Macro metric would be used as this dataset consists of 10 different classes with a balanced proportion.

a. Classification model

K-Nearest Neighbors

The K-NN is a non linear classification algorithm which determines the class of the object based on the neighboring objects[11]. This would be ideal in classifying the music genres. The K is the number of neighbors that would be selected in order to determine the class of the music. The other parameter of this algorithm would be the distance measurement method between the object and the neighbors. The several distances available are the Manhattan (4), Euclidean (5), and Minkowski (6) Distance.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{4}$$

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^c \right)^{\frac{1}{c}} \tag{6}$$

Random Forest

Random forest is an ensemble algorithm based on the decision tree algorithm [12]. This algorithm works by constructing a *forest* of *decision trees.* Each instance would go to each tree where in the end, a voting/majority decision based on the outputs of each tree is made to determine the class of the object.

b. Evaluation of Classification

As all 10 classes on the dataset are proportional, meaning there would be no class imbalance, an F1-Macro Average evaluation metric is used. The F1 metric is twice the value of precision multiplied by recall divided by the addition of the precision and recall as described by (7) [13]. The F1 value is not only describing how accurate the model is but how wrong it is through precision and recall. The value of precision and recall itself would then be described by (8) and

(9) where TP is the number of true positive predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

The F1-Macro itself regards each class as independent and are then averaged. The given formula by (10) provides that i is the different classes and n is the number of classes. This would lead to a more clear understanding of how each model in the model selection method would perform as each class is treated differently.

$$\text{Macro F1 Score} = \frac{\sum_{i=1}^{n} \text{F1 Score}_i}{n} \tag{10}$$

c. Model Selection

As there are two different models being tested, both RF and K-NN would undergo a repeated K-fold cross validation method. The K-fold cross validation is done to have a clear understanding of the model performance across different data partitions. Then, the repetition of the cross validation is used to find the best hyper parameter of the models. Figure 3 illustrates how the data is partitioned and then used for the fitting or training process on the model, although the main difference would be the use a Kof 10 splits of cross validation to evaluate the model.

The parameter tuning process in this work would utilize a linear process. Meaning, across the given hyper parameter choices, the repetition of K-fold cross validation would iterate how each choice of parameters affects the model. Therefore, a measurement on the duration of model selection, model fitting based on the selected parameters, and lastly duration of the inference using the model is done to provide information about efficiency in regard to the classification ability by the model.

Another to understand the model’s performance better, a confusion matrix with the columns showing the predicted labels and the rows showing the true labels of the data is used. This matrix is used to have an understanding on which genre the model is best at predicting and which would be the worst genre to predict.

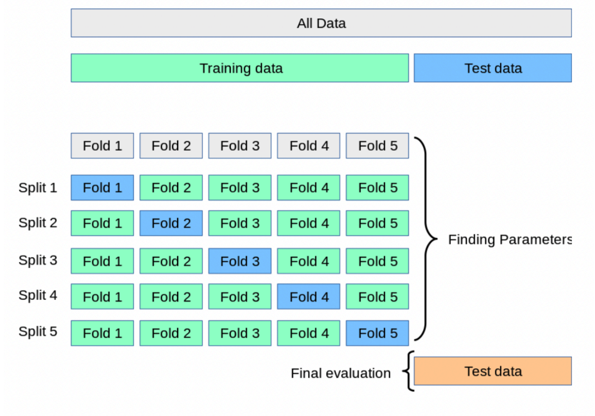


Figure 3. Data partition on model selection process.

III. Results

The results of this project would provide information on the model selection process, training and testing process, and lastly the overall evaluation process. While also providing information regarding the duration or efficiency of each process, the model would work on a table consisting of 67 features and 1 target label. Overall, the RF model performance is more accurate but the K-NN model performance is way faster than the RF model.

a. Model Selection Process

Table 1. Model Selection Evaluation

Model	Selected Parameters	Duration	Average Macro F1-Score (10-Fold CV)
K-NN	p (distance): 1 number of neighbors: 9	**CPU times: 12.2 s**	0.61552
RF	min_samples_leaf : 1 n_estimators: 800 max_depth: 10	CPU times: 2h 25min 19s	**0.69475**

Throughout the model selection process, the K-NN was subjected to a parameter selection of distance measure: Minkowski, Euclidean, and Manhattan, while the RF model needed to select three different parameters: number of estimators, maximum tree depth, and minimum leaf samples. As Table 1 informed, the RF model yields a longer duration for the model selection process compared to the K-NN model, although it may yield a better average Macro F1-Score from the 10-Cross validation process. The cpu runtime for the RF model is more than 700 times longer than the K-NN runtime while only having 11% improvement.

b. Training and Testing process

By utilizing the selected parameters, the training process would be done by using the 75% of the initial split of the data and tested against the rest of the data, having a 25% proportion to the original data. As there is a corrupted audio file, the training process would use only 749 data while the testing process would still keep using the 250 data.

The result on Table 2 shows that the RF model managed to put up an amount of 70% Macro F1-Score, while the K-NN managed to yield 67% of Macro F1-Score. It is also shown that the duration it takes for the RF model to be built or trained is longer than the inference time, having more than two times the duration of the inference process. Although the RF may seem to have lower duration both in training and inference process, the K-NN model actually has a higher inference duration per music than the training duration. Having a difference of more than 0.01 ms.

Table 2. Training and Testing Process Evaluation

Model	Training Duration	Inference Duration	Macro F1-Score on Test Data
K-NN	**CPU times: 3.12 ms, 0.00416 ms for 1 music**	**CPU times: 21.7 ms, 0.0868 ms for 1 music**	0.6725828546881522
RF	CPU times: 2.13 s, 2.84 ms for 1 music	CPU times: 47.1 ms, 0.1884 ms for 1 music	**0.700388235612284**

c. Overall Evaluation

Both RF and K-NN models struggled to predict several music genres. Rock, Jazz, Hiphop, Reggae, and Country are the notable genres the model struggled to predict.

Corresponding to the performance evaluation using the Macro F1-Score, there are more false classification on the K-NN model than the RF model. It should be noted that as described by the confusion matrix on K-NN and RF models which are available at the appendix, several misclassifications of a certain genre are centralized to only one or two genres.

IV. Discussions and Conclusion

The performance of the RF model may be more accurate than the K-NN model, but the K-NN model yields a faster process. Issue would come to the bigger complexity an RF model has as it would try to construct a large number of decision trees, while on the other hand a K-NN model would only construct one *nearest neighbor* model. As expected by the bigger number of *rules* the RF could store compared to the K-NN model which only classifies based on the distance with the neighbors, the RF managed to acquire a higher Macro F1-Score. Although with the testing done in this project, the complexity trade-off for the RF model to acquire higher classification measure may not be worth the time in a real-life scenario where an industry may need fast decisions or outputs.

In classifying the music, the misclassification of the different genres may come due to the limited and low number of music samples. Having only 100 instances for each class may come as a handicap in this project. The thin border between each genre may come as a contributing factor in this work as some music may be categorized to multiple genres. This would indicate future works to try to gather a more comprehensive dataset in which each genre is defined clearly and more samples are present.

Therefore in concluding this work, the use of feature extraction on one dimensional signals was able to yield information, useful for a classification process. Certain algorithms for the classification process such as K-NN may be less powerful in terms of its classification evaluation metric result, but it may perform better in a real-life situation as it is able to be trained at a faster rate while having small classification performance trade-off. The use of a comprehensive and well-structured dataset is also needed if the goal of the project is to be implemented in a business scenario.

References

[1] Spotify, “About Spotify.” 2024 [Online]. Available: https://newsroom.spotify.com/company-info/. Accessed at: 15 March 2024

[2] A. Olteanu, “GTZAN Dataset - Music Genre Classification”, Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genreclassification.

[3] D. S. Lau and R. Ajoodha, ‘Music Genre Classification: A Comparative Study Between Deep Learning and Traditional Machine Learning Approaches’, in Proceedings of Sixth International Congress on Information and Communication Technology, 2022, pp. 239–247.

[4] B. L. Sturm, ‘The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use’, CoRR, vol. abs/1306.1461, 2013.

[5] B. L. Sturm, ‘The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval’, Journal of New Music Research, vol. 43, no. 2, pp. 147–172, 2014.

[6] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler, "A tutorial on onset detection in music signals," in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, Sept. 2005, doi: 10.1109/TSA.2005.851998.

[7] M. Müller, S. Balke, “Lab Course Short-Time Fourier Transform and Chroma Features,” Lab Course Friedrich-Alexander-Universit¨at Erlangen-N¨urn, 2018.

[8] U. Nam, “Special Area Exam Part II,” Stanford University, 2001.

[9] Cph Nano, “What Is Spectral Bandwidth within UV-Vis Spectroscopy?” (Online) Available: https://knowledge.cphnano.com/en/pages/what-is-spectral-bandwidth-within-uv-vis-spectroscopy. Accessed at: 19 March 2024

[10] M. R. Hasan, M. M. Hasan, and M. Z. Hossain, ‘How many Mel-frequency cepstral coefficients to be utilized in speech recognition? A study with the Bengali language’, The Journal of Engineering, vol. 2021, no. 12, pp. 817–827, 2021.

[11] A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, ‘k-Nearest Neighbor Classification’, in Data Mining in Agriculture, New York, NY: Springer New York, 2009, pp. 83–106.

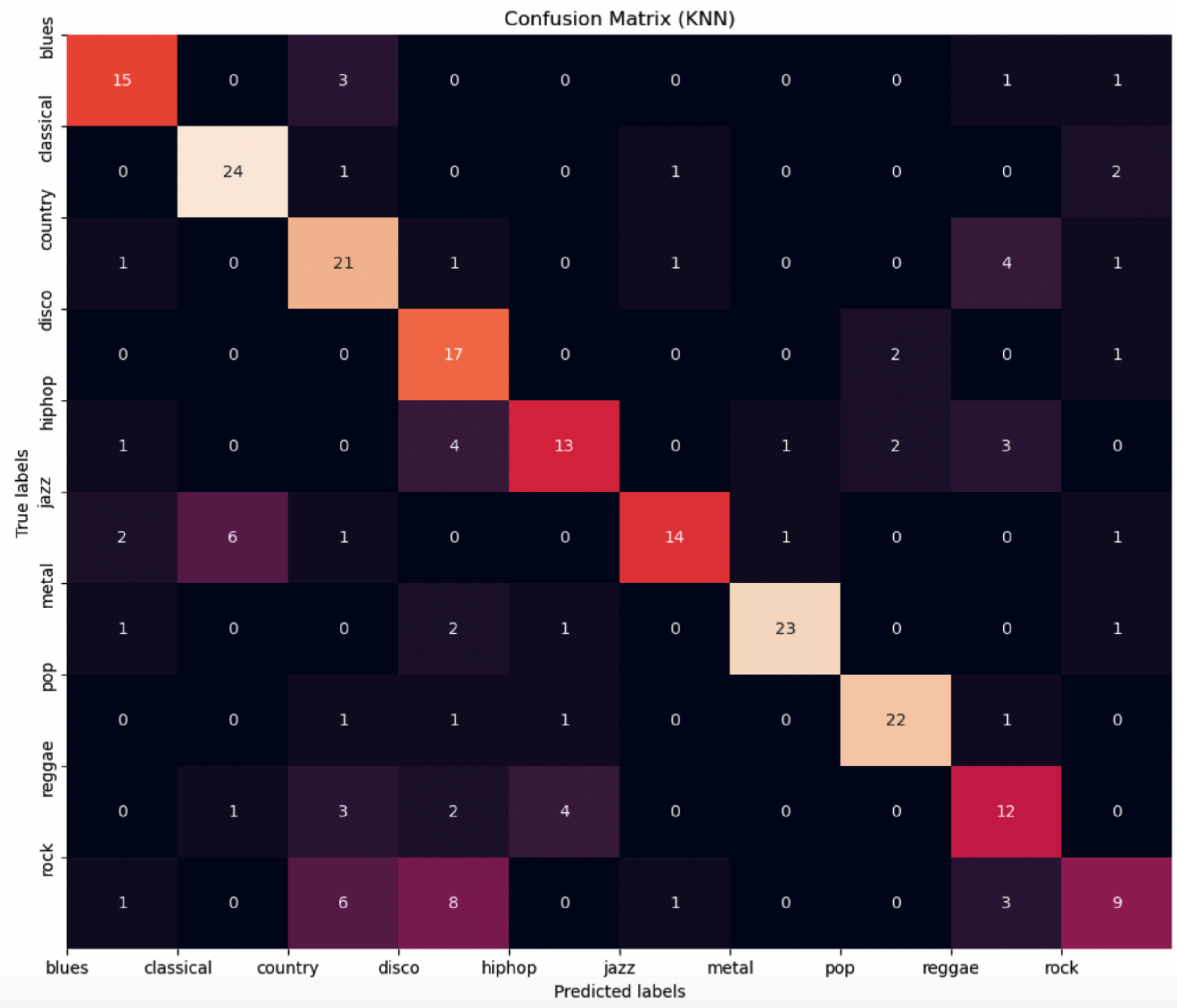
[12] L. Breiman, ‘Random Forests’, Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[13] C. Goutte and E. Gaussier, ‘A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation’, in Advances in Information Retrieval, 2005, pp. 345–359.

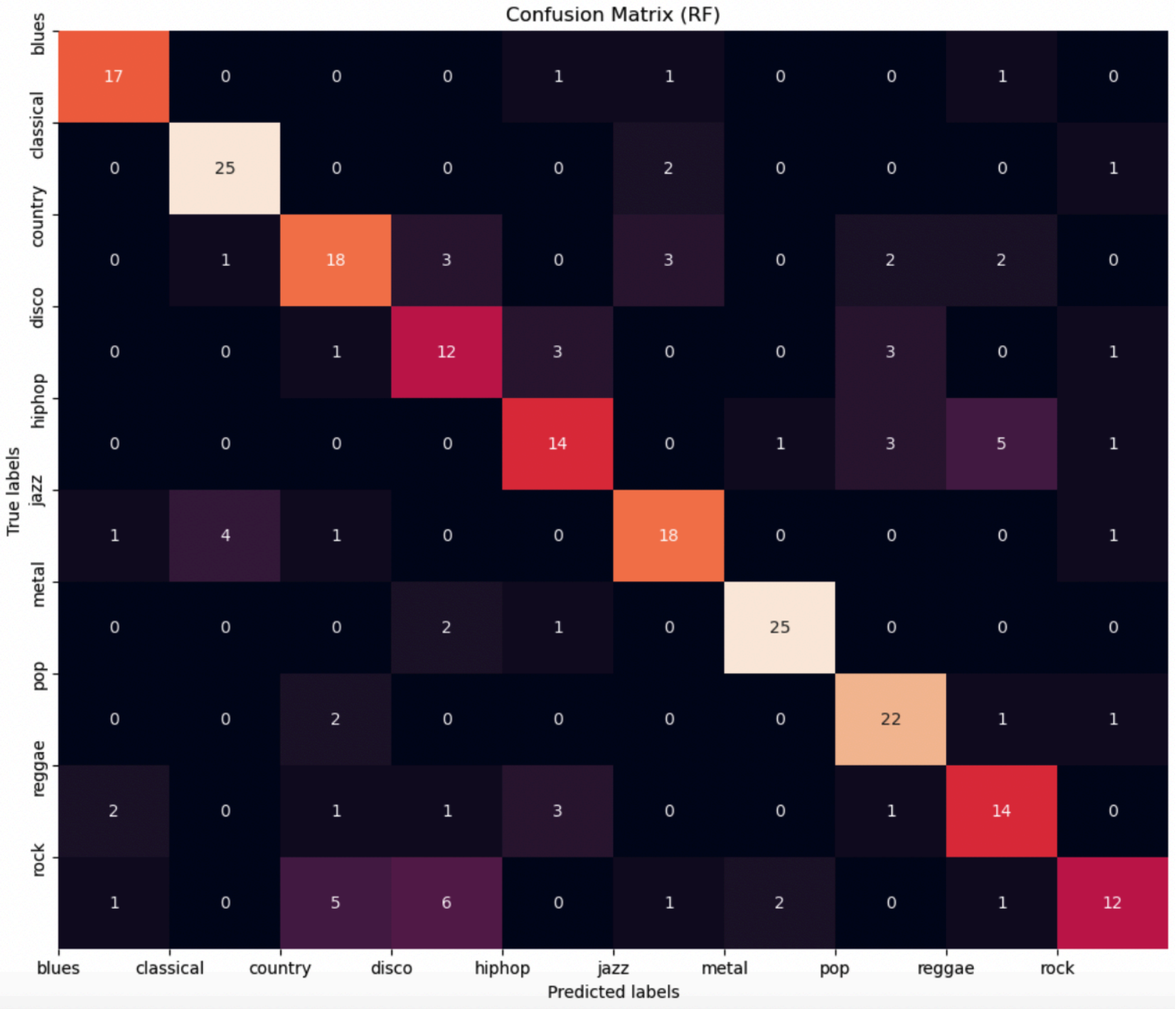
Appendix:

This project is able to be accessed in the github repository of : https://github.com/louiswids/Music-Genre-Classification where a graphical analysis, feature extraction, and the feature matching process is present on three different python notebooks.

Confusion matrices for K-NN and Random Forest model:



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	15	0	3	0	0	0	0	0	1	1
classical	0	24	1	0	0	1	0	0	0	2
country	1	0	21	1	0	1	0	0	4	1
disco	0	0	0	17	0	0	0	2	0	1
hiphop	1	0	0	4	13	0	1	2	3	0
jazz	2	6	1	0	0	14	1	0	0	1
metal	1	0	0	2	1	0	23	0	0	1
pop	0	0	1	1	1	0	0	22	1	0
reggae	0	1	3	2	4	0	0	0	12	0
rock	1	0	6	8	0	1	0	0	3	9



Confusion Matrix (RF)

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	17	0	0	0	1	1	0	0	1	0
classical	0	25	0	0	0	2	0	0	0	1
country	0	1	18	3	0	3	0	2	2	0
disco	0	0	1	12	3	0	0	3	0	1
hiphop	0	0	0	0	14	0	1	3	5	1
jazz	1	4	1	0	0	18	0	0	0	1
metal	0	0	0	2	1	0	25	0	0	0
pop	0	0	2	0	0	0	0	22	1	1
reggae	2	0	1	1	3	0	0	1	14	0
rock	1	0	5	6	0	1	2	0	1	12