Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/79/20241639

Music genre classification: Machine Learning on GTZAN

Ziyan Zhao[1,5,*], Zixiao Xie[2,6], Jiaze Fu[3,7], Xintao Tian[4,8]

[1]Computer Academy, Jilin University Zhuhai Campus, Zhuhai, 519040, China
[2]Academy of Software, Nanjing University of Information Science and Technology, Nanjing,210044, China
[3]Computer Academy, Southwest Jiaotong University, Chengdu, 611756, China
[4]Valley Catholic High School, Beaverton, 97007, USA

[5]zhaozy2050692725@gmail.com
[6]1449095617@qq.com
[7]1260601778@qq.com
[8]xintaotian05@163.com
*corresponding author

Abstract. This paper explores music genre classification, aiming to enhance existing methodologies. As a crucial aspect of music information retrieval, genre classification facilitates organization and recommendation in music databases and streaming services. Our research, inspired by a Kaggle project, examines the background of music genre classification and introduces improvements. The study focuses on data preparation techniques and a novel methodology using Support Vector Machines (SVM). Utilizing the GTZAN dataset, we applied data segmentation and feature extraction, employing machine learning algorithms like Logistic Regression, Random Forest, and SVM. A significant innovation is our segmentation technique based on music's beats per minute (BPM), designed to preserve rhythmic structure, believed to be essential for accurate classification. We explored various feature extraction methods to boost classifier performance. Experimental results showed the 3-second segmented dataset performed better with SVM's linear kernel. Additionally, a 4-beat segmentation experiment suggested that finer segmentation captures richer audio features, potentially improving classification accuracy. The paper concludes with findings and future research directions, including dataset expansion, advanced segmentation based on musical theory, deep learning applications, and developing real-time classification systems.

Keywords: Music Genre Classification, Kaggle, GTZAN Dataset, Support Vector Machines, Data Segmentation, Feature Extraction.

1. Introduction

1.1. Background of music genre classification

Music genre classification is a vital component of music information retrieval, aiding in the organization, searching, and recommendation of music. The automatic classification of music into genres has been a topic of interest for researchers, given its potential applications in music databases, streaming services, and in aiding musicologists in understanding musical styles. Typically, humans can identify the genre

DOI: 10.54254/2755-2721/79/20241639

of a music piece by listening to its sound or by viewing the musical score, utilizing the physical features apparent to the human senses. However, from a machine's perspective, the process is quite different. In our scenario, the focus shifts to understanding the composition of music from a machine's viewpoint, utilizing the accessible data [1]. Essentially, this entails analyzing the mathematical features extractable from the music file. Through examining these mathematical attributes, we aim to explore the machine's ability to recognize and categorize music genres based on the mathematical features extracted from the music files.

1.2. Brief mention of the Kaggle project and its methodology

A project on Kaggle delved into music genre classification employing the GTZAN Dataset. The methodology encompassed data preprocessing, feature extraction, and the utilization of machine learning algorithms like Logistic Regression and Random Forest for the classification task. During data preprocessing, tasks such as handling missing values, shuffling the data, splitting it into training, validation, and testing sets, and scaling the features were carried out to ensure a clean and well-organized dataset. Feature extraction focused on musical attributes like Chroma_stft, Root Mean Square, Spectral centroid, among others, to represent the music files for classification. This project set a baseline by showcasing a structured approach to classifying music genres, which serves as a stepping stone for our research.

1.3. Motivation for our enhancements

The aforementioned Kaggle project provided a structured approach towards music genre classification. However, there remains scope for enhancing the accuracy and efficiency of genre classification by exploring alternative machine-learning algorithms and feature extraction techniques. Inspired by this project, we aim to explore tweaks in how the music is sliced and which algorithms are chosen to see if we can get even better at correctly identifying music genres. This isn't about reinventing the wheel, but about fine-tuning the existing approach to hopefully get better results.

2. Related Work

2.1. Detailed overview of the Kaggle approach

The Kaggle project under discussion presented a structured methodology for music genre classification utilizing the GTZAN dataset. The GTZAN dataset is a widely recognized dataset in the domain of music genre classification, comprising 1000 audio tracks each 30 seconds long, categorized into 10 genres.

The project's approach can be broadly categorized into three main phases: Data Preprocessing, Feature Extraction, and Model Deployment.

2.1.1. Data Preprocessing

The dataset was initially loaded and examined to ensure data integrity.

Any missing or null values were identified, although none were found.

The data was then shuffled, and split into training (70%), validation (20%), and testing (10%) sets to ensure a good mix of genres in each set.

2.1.2. Feature Extraction

Features were extracted from 30-second audio clips, which were then segmented into 3-second clips to provide a more granular analysis.

The primary features extracted included Chroma_stft, Root Mean Square, Spectral centroid, Spectral bandwidth, Rolloff, Zero crossing rate, and Mel-frequency cepstral coefficients (MFCC).

These features are pivotal in representing the mathematical and statistical characteristics of the audio files, providing a basis for classification.

DOI: 10.54254/2755-2721/79/20241639

2.1.3. Model Deployment

Two machine learning algorithms, Logistic Regression (LR) and Random Forest, were employed for the classification task [2].

Initial modeling was performed using Logistic Regression, followed by an analysis of feature importance using Permutation Importance from the eli5 library.

Random Forest was then deployed, which is known for its ability to handle a large number of features and provide a measure of feature importance.

Hyperparameters for both models were tuned using Grid Search to optimize performance.

The performance of the models was evaluated on the validation and test sets, providing insights into the models' ability to generalize across unseen data.

This comprehensive approach laid down by the Kaggle project provides a solid foundation, showcasing the systematic steps involved in tackling the music genre classification problem. It sets a precedent for further exploration and enhancement in algorithm selection, feature extraction, and data segmentation, which are the focal points of our research endeavor.

2.2. Detailed Introduction On Algorithms (LR and Random Forest)

In the Kaggle project under review, two primary algorithms were employed for the task of genre classification—Logistic Regression (LR) and Random Forest [3]. Here we elucidate the fundamental principles and steps involved in these algorithms:

Logistic Regression:

Logistic Regression, a type of generalized linear regression model, is notably different from linear regression models based on the nature of the dependent variable. While linear regression is suited for continuous dependent variables, logistic regression caters to binomially distributed dependent variables.

2.2.1. Steps involved in Logistic Regression

Prediction Function (h function): The first step in logistic regression involves finding the prediction function, denoted as the h function, which estimates the probability that a given input point belongs to a certain class.

Loss Function (J-function): Construction of the loss function, denoted as the J-function, follows. This function measures the difference between the predicted values and the actual values, aiming to minimize this discrepancy.

Parameter Optimization: Lastly, a method to minimize the J-function is devised to obtain the regression parameters (θ), which in turn fine-tunes the prediction function for better accuracy.

Random Forest: Random Forest algorithm operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes output by individual trees, essentially a form of voting.

2.2.2. Procedure of Random Forest

Sampling with Dropout: Random Forest starts by randomly sampling the original training set with dropout, forming a new training set for each tree in the forest.

Node Training: During the training of nodes for each tree, a subset of features is randomly selected from all features. The nodes are then segmented based on these selected features.

Leaf Node Categorization: Upon completion of the segmentation, each leaf node corresponds to a category label.

Majority Voting: The final category label for a given input is determined through the majority voting method across all trees in the forest.

2.2.3. Advantages of Random Forest

The model is known for its ability to prevent overfitting effectively, handle a large number of input variables efficiently, and offer good scalability.

DOI: 10.54254/2755-2721/79/20241639

Comprising multiple trees, it captures the complexity and uncertainty of sample data well, which in turn enhances the accuracy of predictions.

3. Data Preparation

3.1. Segmentation Technique

In the original Kaggle approach, the 30-second audio tracks were segmented into smaller, 3-second clips to extract features for analysis. This segmentation enabled a more granular examination of the audio, which potentially could lead to more accurate genre classification. However, our approach contemplates the optimization of this segmentation technique, exploring whether different segmentation lengths could yield better classification results.

Building on this, we introduced a new segmentation technique that takes into account the beats per minute (BPM) of the music, dividing the tracks based on musical bars, with each segment encompassing four beats—a common musical bar length. This dynamic segmentation aims at preserving the rhythmic structure of the music, which is hypothesized to be crucial for accurate genre classification.

3.2. Extraction of Features

The Kaggle project demonstrated a systematic approach to feature extraction, focusing on several key audio features including Chroma_stft, Root Mean Square, Spectral centroid, and others. In our study, we intend to extend this feature set by exploring additional audio characteristics that could enhance the classifier's performance.

4. Proposed Methodology

4.1. Introduction to Support Vector Machines (SVM)

The fundamental objective of Support Vector Machine (SVM) learning is to discover a separating hyperplane that can accurately partition the training dataset while maintaining the largest geometric margin [4]. This is epitomized by the equation $w * x + b = 0$, representing the separating hyperplane. In scenarios where the dataset is linearly separable, an infinite number of such hyperplanes can be drawn, albeit the separating hyperplane boasting the largest geometric margin remains unique.

In the pursuit of optimizing the model for genre classification, three prevalent kernel functions were employed to process the dataset, namely linear kernel, RBF kernel, and poly kernel.

4.1.1. Linear Kernel

$$K(x_i, x_j) = (x_i * x_j)$$

The linear kernel, being the most rudimentary type, often manifests in a one-dimensional nature. It is predominantly favored in text classification problems since many of these can be linearly separated. Additionally, linear kernel functions are recognized for their computational efficiency compared to other kernel types.

4.1.2. RBF Kernel

$$K(x, y) = e^{-\gamma||x-y||^2}$$

The Radial Basis Function (RBF) kernel is a commonly utilized kernel function in SVM, particularly suited for nonlinear data. In instances where prior knowledge regarding the data is absent, the RBF kernel facilitates an appropriate separation.

4.1.3. Poly Kernel

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d$$

DOI: 10.54254/2755-2721/79/20241639

The poly kernel serves as a more generalized representation of linear kernels. However, its popularity is eclipsed by other kernel functions due to its relatively lower efficiency and accuracy.

The code provided delineates the systematic approach towards deploying and optimizing the SVM model. Initially, the dataset was loaded, followed by encoding the labels and standardizing the features to ensure uniformity and to mitigate the influence of features with larger values. Subsequently, the dataset was split into training and testing sets, forming the foundation for model training and evaluation.

In the modeling phase, SVM with different kernels was employed. Each kernel type (linear, RBF, poly, and LinearSVC) was trained on the dataset, and their performance was evaluated using a customized `results` function, which rendered a confusion matrix and a classification report encapsulating accuracy, precision, recall, and F1-score for the different genres.

Furthermore, to hone the model, GridSearchCV was utilized to ascertain the optimal parameters for the SVM model, thereby enhancing its performance. The `GridSearchCV` method conducted an exhaustive search over specified parameter values for the SVM model, ultimately identifying the parameters that yielded the highest accuracy on the training set. The best-trained model was then evaluated on the test set to validate its effectiveness and generalization capability [5].

4.2. Reasons for choosing SVM

The selection of Support Vector Machines (SVM) for the music genre classification task emanates from a blend of compelling factors that underscore SVM's aptness for this domain. These factors, especially when juxtaposed against Logistic Regression and Random Forest models, amplify SVM's appeal:

1. **Small Sample Efficiency:** SVM exhibits prowess in efficiently handling small sample data, a scenario often encountered in our dataset. This contrasts with Logistic Regression and Random Forest models which might struggle with smaller datasets.
2. **High-Dimensional Data Handling:** The genre classification task entails grappling with high-dimensional data. SVM, with its kernel trick, is well-equipped to navigate such high-dimensional spaces, a challenge that might impede the performance of Random Forest models.
3. **Computational Complexity:** The computational burden of SVM is primarily dictated by the number of support vectors, not the dimension of the sample space. This aspect is particularly attractive as it ensures reasonable computational demand even as the feature space expands.
4. **Avoidance of Local Minima:** SVM's optimization routine is adept at bypassing local minima, a common issue in machine learning tasks. This characteristic is accentuated when dealing with large-scale training samples, a situation that is less likely to occur with SVM due to its optimization strategy.
5. **Nonlinear Feature Interaction Processing:** Music genre classification often entails deciphering nonlinear feature interactions. SVM is capable of handling such nonlinear interactions, albeit with a trade-off in interpretability, especially when employing radial basis functions.
6. **Absence of Probability Significance:** Unlike Logistic Regression that offers natural probability significance, SVM does not provide probability estimates. However, this was not deemed a significant drawback for our genre classification task, where the primary aim is accurate categorization rather than probability estimation.

Considering the outlined reasons, SVM emerged as a logical choice for this project. Its ability to efficiently manage small sample sizes, high-dimensional data, and nonlinear feature interactions, coupled with a favorable computational complexity profile, made it an appealing choice for tackling the intricacies of music genre classification.

5. Experimental Results

In the following sections, we delve into the specific methodologies utilized in applying SVM, Logistic Regression, and Random Forest models to our dataset, and the feature engineering strategies adopted to enhance these models. The comparative analysis of these models' performance in the domain of music genre classification is also detailed. Through a series of experiments, we aim to validate the hypothesis that with suitable segmentation and refined feature engineering, it's possible to achieve accurate genre classification.

DOI: 10.54254/2755-2721/79/20241639

5.1. The extraction of CSV datasets

The initial step in our methodology involved the extraction of datasets in a structured CSV format, which would be conducive for our machine learning models to process. The data was derived from audio files which were segmented into specific time intervals, as per the requirements of our project. Here's a breakdown of the data extraction process:

5.1.1. Dataset Source

Our primary dataset was sourced from the GTZAN genre collection.

5.1.2. Audio Segmentation

Initially, the segmentation approach mirrored that of the original Kaggle project, with 3-second snippets. However, to potentially capture more granular features, we also experimented with 1-second snippets. This segmentation was carried out using audio processing libraries that facilitated the extraction of structural features from audio files.

5.1.3. Feature Extraction

Post segmentation, we embarked on the feature extraction phase, which is pivotal for the success of any machine learning model in understanding the underlying patterns within the data. Utilizing the Librosa library, a suite of Python packages for music and audio analysis, we extracted a variety of features such as Mel-Frequency Cepstral Coefficients (MFCCs), Chroma Frequencies, Spectral Contrast, and others. These features encapsulate the characteristics of audio segments that are instrumental for genre classification.

5.1.4. Data Structuring

The extracted features were structured into a CSV format, creating a well-organized dataset. Each row in the CSV corresponded to a segmented audio snippet, with the extracted features as columns, and the genre label as the target variable.

5.1.5. Data Preprocessing

Before moving to the modeling phase, the data underwent a preprocessing step where it was standardized to have a mean of zero and a standard deviation of one. This was crucial to ensure that all features contribute equally to the distance calculations in our SVM model.

5.1.6. Dataset Splitting

The structured CSV dataset was then split into training and testing/validation sets using a 70-30 split ratio. This setup provided a robust platform for training our models and evaluating the performance on unseen data. The testing/validation sets further facilitated the tuning of hyperparameters to enhance the model's predictive accuracy. The test output is shown as **Figure 1**.

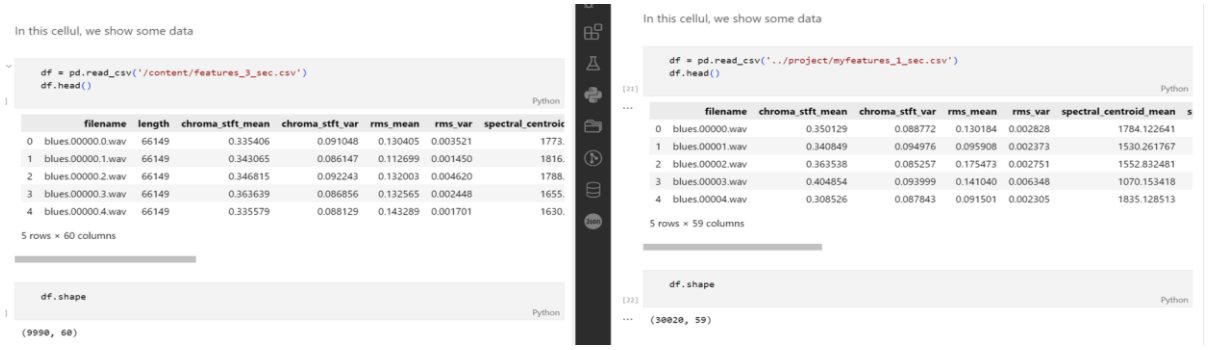


Figure 1. Dataset Splitting

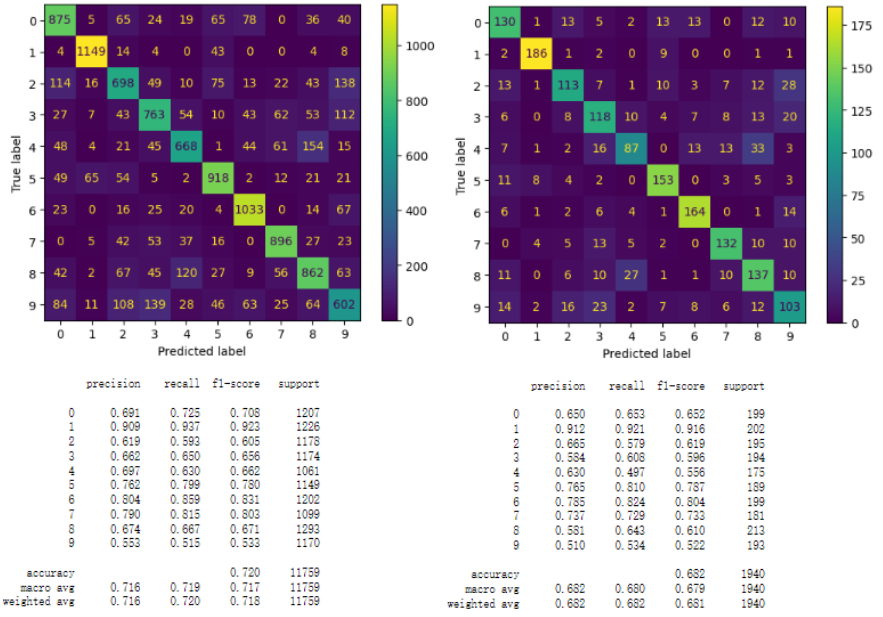
DOI: 10.54254/2755-2721/79/20241639

5.2. Application of Logistic Regression and Random Forest Models

In light of the new dataset generated based on a 4-beat segmentation technique, the experimentation was extended to Logistic Regression and Random Forest models. And the output is shown in **Figure 2**. The process and findings from these experiments are as follows:

1. **Model Training:** Both models were trained using the new segmented dataset, with hyperparameters tuned to optimize performance.
2. **Performance Evaluation:** The models' performance was evaluated on validation and test sets, shedding light on their generalization capabilities.
3. **Feature Importance Analysis:** Permutation Importance was utilized to analyze feature importance, aiding in understanding the contribution of each feature towards the classification task.

5.2.1. Feature importance using logistic regression model and random forest model



	precision	recall	f1-score	support
0	0.691	0.725	0.708	1207
1	0.909	0.937	0.923	1226
2	0.619	0.593	0.605	1178
3	0.662	0.650	0.656	1174
4	0.697	0.630	0.662	1061
5	0.762	0.799	0.780	1149
6	0.804	0.859	0.831	1202
7	0.790	0.815	0.803	1099
8	0.674	0.667	0.671	1293
9	0.553	0.515	0.533	1170
accuracy			0.720	11759
macro avg	0.716	0.719	0.717	11759
weighted avg	0.716	0.720	0.718	11759

	precision	recall	f1-score	support
0	0.650	0.653	0.652	199
1	0.912	0.921	0.916	202
2	0.665	0.579	0.619	195
3	0.584	0.608	0.596	194
4	0.630	0.497	0.556	175
5	0.765	0.810	0.787	189
6	0.785	0.824	0.804	199
7	0.737	0.729	0.733	181
8	0.581	0.643	0.610	213
9	0.510	0.534	0.522	193
accuracy			0.682	1940
macro avg	0.682	0.680	0.679	1940
weighted avg	0.682	0.682	0.681	1940

Figure 2. logistic regression model

5.2.2. Feature Importances using Permutation Importance

Firstly, we plot the Permutation Importance of music classification features, calculate the Permutation Importance according to the following steps, and we got the output as Figure 3.

DOI: 10.54254/2755-2721/79/20241639

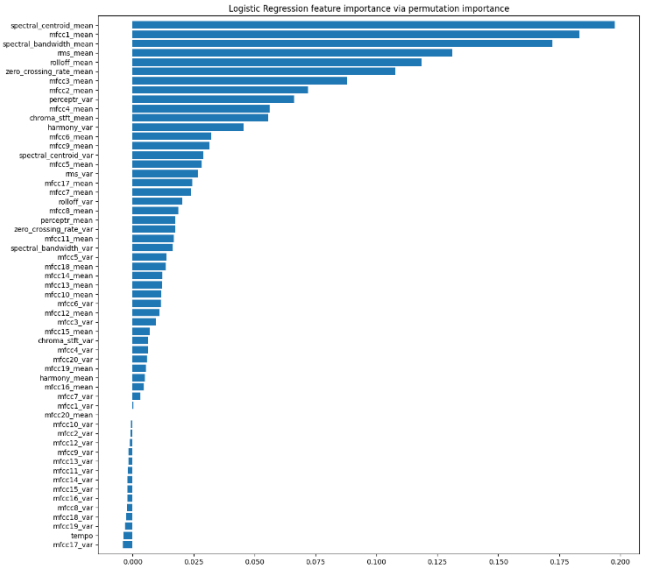


Figure 3. Permutation Importance

Then, we build the model using Permutation Importance selected top 30 features. By using the importance of permutation, the accuracy of model testing has been slightly improved. With the comparison of the results illustrated by **Figure 4** and **Figure 5.**

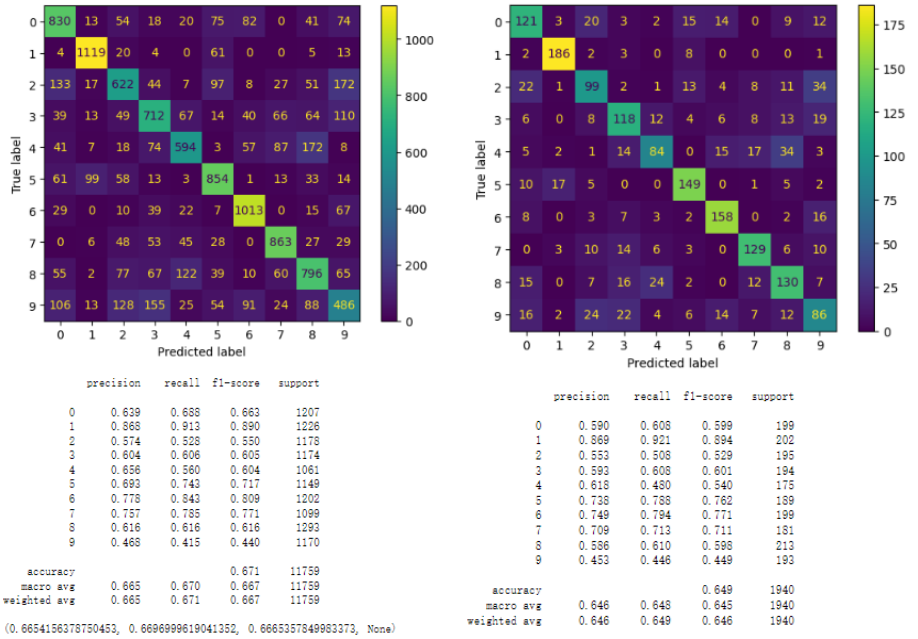
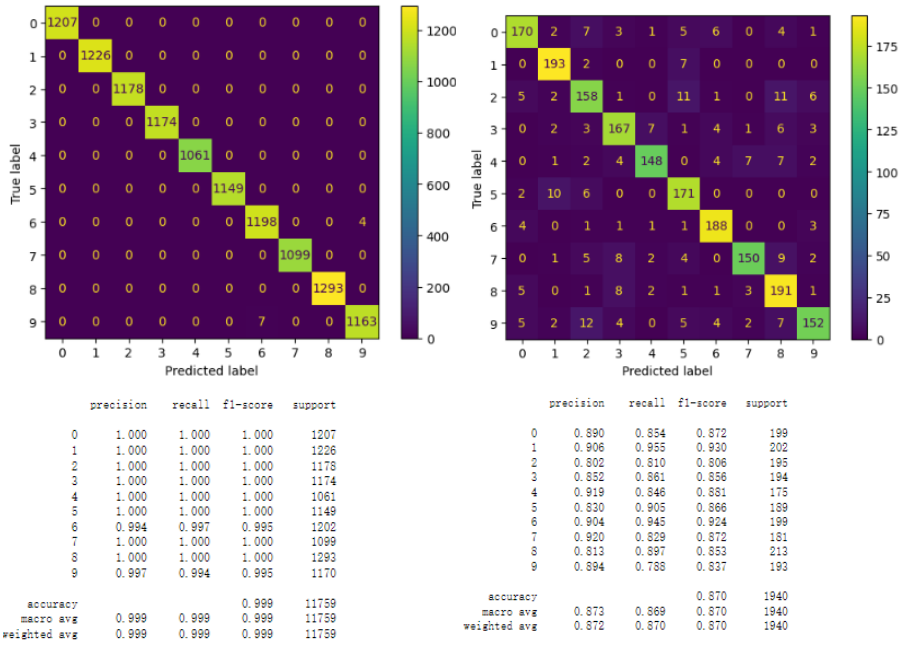


Figure 4. logistic regression model

DOI: 10.54254/2755-2721/79/20241639



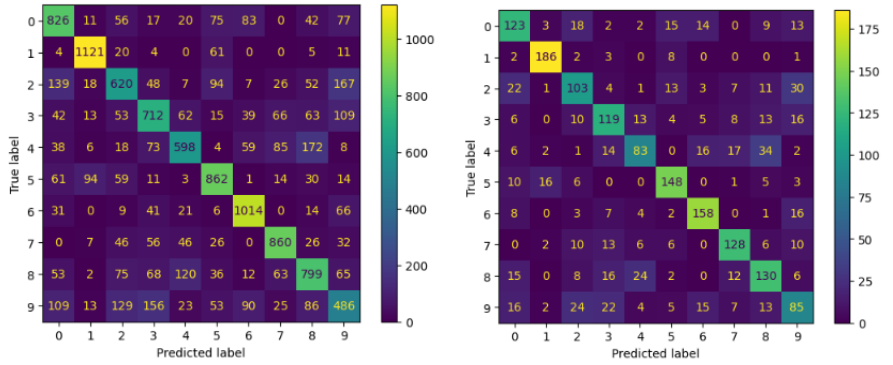
	precision	recall	f1-score	support
0	1.000	1.000	1.000	1207
1	1.000	1.000	1.000	1226
2	1.000	1.000	1.000	1178
3	1.000	1.000	1.000	1174
4	1.000	1.000	1.000	1061
5	1.000	1.000	1.000	1149
6	0.994	0.997	0.995	1202
7	1.000	1.000	1.000	1099
8	1.000	1.000	1.000	1293
9	0.997	0.994	0.995	1170
accuracy			0.999	11759
macro avg	0.999	0.999	0.999	11759
weighted avg	0.999	0.999	0.999	11759

	precision	recall	f1-score	support
0	0.890	0.854	0.872	199
1	0.906	0.955	0.930	202
2	0.802	0.810	0.806	195
3	0.852	0.861	0.856	194
4	0.919	0.846	0.881	175
5	0.830	0.905	0.866	189
6	0.904	0.945	0.924	199
7	0.920	0.829	0.872	181
8	0.813	0.897	0.853	213
9	0.894	0.788	0.837	193
accuracy			0.870	1940
macro avg	0.873	0.869	0.870	1940
weighted avg	0.872	0.870	0.870	1940

Figure 5. random forest model

5.2.3. Tuning hyperparameters of the logistic regression model and random forest model using grid_search

Using grid_ Search adjusts the hyperparameters of the logistic regression model to find the optimal combination of parameters for the model. In this way, we can find the optimal regularization parameters that are suitable for the data. This is done through grid search, which means that all hyperparameter selections will be listed and arranged separately, and the system will evaluate the performance of each combination to determine the optimal parameter combination. With the comparison of the results illustrated by **Figure 6** and **Figure 7**

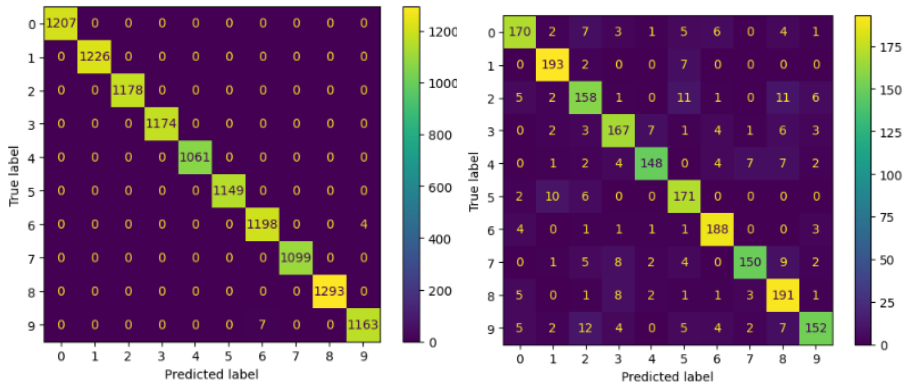


	precision	recall	f1-score	support
0	0.634	0.684	0.658	1207
1	0.872	0.914	0.893	1226
2	0.571	0.526	0.548	1178
3	0.600	0.606	0.603	1174
4	0.664	0.564	0.610	1061
5	0.700	0.750	0.724	1149
6	0.777	0.844	0.809	1202
7	0.755	0.783	0.769	1099
8	0.620	0.618	0.619	1293
9	0.470	0.415	0.441	1170
accuracy			0.672	11759
macro avg	0.666	0.670	0.667	11759
weighted avg	0.666	0.672	0.668	11759

	precision	recall	f1-score	support
0	0.591	0.618	0.604	199
1	0.877	0.921	0.899	202
2	0.557	0.528	0.542	195
3	0.595	0.613	0.604	194
4	0.606	0.474	0.532	175
5	0.729	0.783	0.755	189
6	0.749	0.794	0.771	199
7	0.711	0.707	0.709	181
8	0.586	0.610	0.598	213
9	0.467	0.440	0.453	193
accuracy			0.651	1940
macro avg	0.647	0.649	0.647	1940
weighted avg	0.647	0.651	0.648	1940

Figure 6. logistic regression model

DOI: 10.54254/2755-2721/79/20241639



	precision	recall	f1-score	support
0	1.000	1.000	1.000	1207
1	1.000	1.000	1.000	1226
2	1.000	1.000	1.000	1178
3	1.000	1.000	1.000	1174
4	1.000	1.000	1.000	1061
5	1.000	1.000	1.000	1149
6	0.994	0.997	0.995	1202
7	1.000	1.000	1.000	1099
8	1.000	1.000	1.000	1293
9	0.997	0.994	0.995	1170
accuracy			0.999	11759
macro avg	0.999	0.999	0.999	11759
weighted avg	0.999	0.999	0.999	11759

	precision	recall	f1-score	support
0	0.890	0.854	0.872	199
1	0.906	0.955	0.930	202
2	0.802	0.810	0.806	195
3	0.852	0.861	0.856	194
4	0.919	0.846	0.881	175
5	0.830	0.905	0.866	189
6	0.904	0.945	0.924	199
7	0.920	0.829	0.872	181
8	0.813	0.897	0.853	213
9	0.894	0.788	0.837	193
accuracy			0.870	1940
macro avg	0.873	0.869	0.870	1940
weighted avg	0.872	0.870	0.870	1940

Figure 7. random forest model

5.3. SVM Models

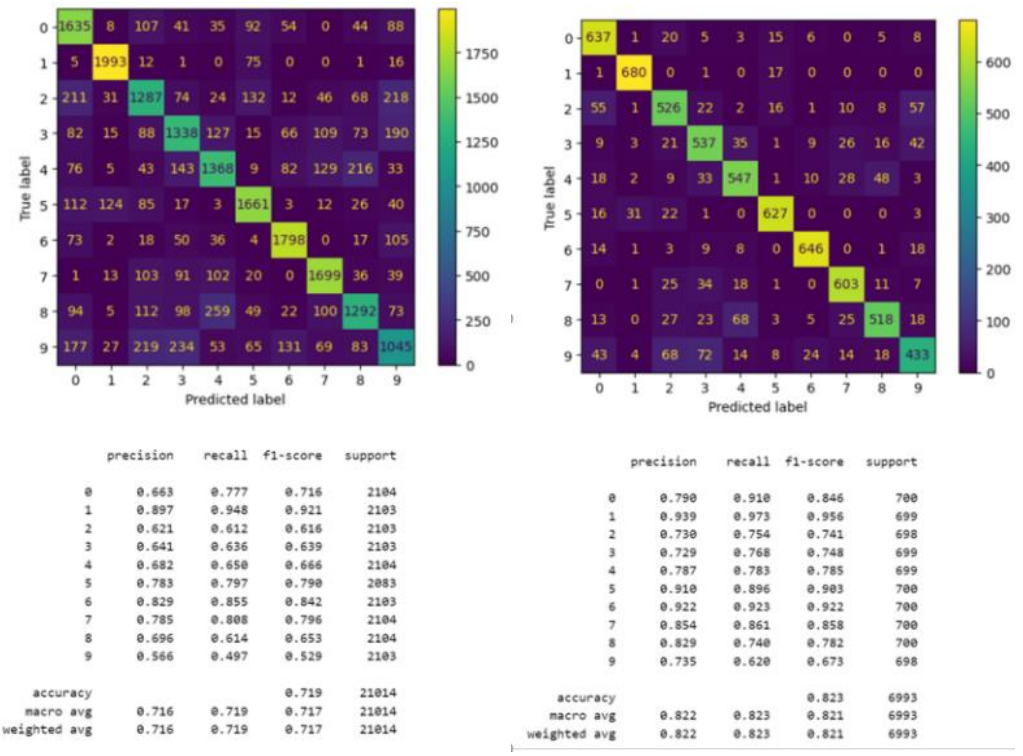
5.3.1. The Linear Kernel of SVM

In our endeavor to ascertain the most effective data segmentation for genre classification, we initially hypothesized that a finer segmentation of 1-second per clip might yield better results due to the increased data granularity. However, the practical experimentation steered us towards a different conclusion.

Employing the linear kernel of SVM, chiefly for its computational efficiency, we trained our model on both 1-second and 3-second segmented datasets, keeping the penalty coefficient at its default setting to ensure a consistent evaluation framework.

Contrary to our initial anticipation, the 3-second segmented dataset delivered a superior performance. The accuracy improvement was notable according to the result on **Figure 8**, standing at about 13% higher compared to the 1-second segmented dataset.

DOI: 10.54254/2755-2721/79/20241639



	precision	recall	f1-score	support
0	0.663	0.777	0.716	2104
1	0.897	0.948	0.921	2103
2	0.621	0.612	0.616	2103
3	0.641	0.636	0.639	2103
4	0.682	0.650	0.666	2104
5	0.783	0.797	0.790	2083
6	0.829	0.855	0.842	2103
7	0.785	0.808	0.796	2104
8	0.696	0.614	0.653	2104
9	0.566	0.497	0.529	2103
accuracy			0.719	21014
macro avg	0.716	0.719	0.717	21014
weighted avg	0.716	0.719	0.717	21014

	precision	recall	f1-score	support
0	0.790	0.910	0.846	700
1	0.939	0.973	0.956	699
2	0.730	0.754	0.741	698
3	0.729	0.768	0.748	699
4	0.787	0.783	0.785	699
5	0.910	0.896	0.903	700
6	0.922	0.923	0.922	700
7	0.854	0.861	0.858	700
8	0.829	0.740	0.782	700
9	0.735	0.620	0.673	698
accuracy			0.823	6993
macro avg	0.822	0.823	0.821	6993
weighted avg	0.822	0.823	0.821	6993

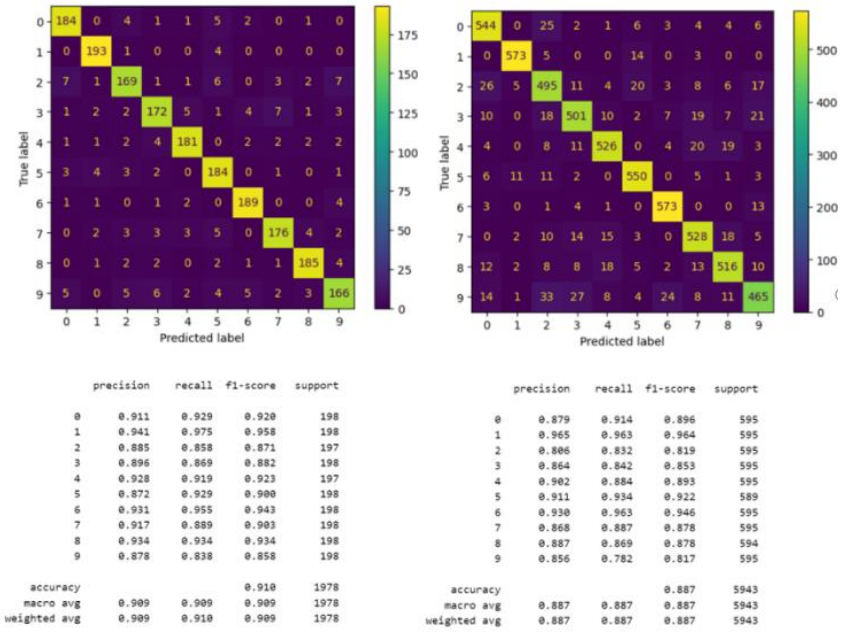
Figure 8. Accuracy comparison between 1-second and 3-second segmented datasets using SVM with Linear Kernel.

5.3.2. The RBF Kernel of SVM

Motivated by the potential of the RBF kernel to better capture the nuances of our dataset, we transitioned from the linear kernel, anticipating a significant boost in accuracy. The experiment was structured similarly to maintain consistency, with the SVM model now employing the RBF kernel.

The results, however, narrated a different story. The anticipated drastic improvement in accuracy was not realized. Instead, a marginal narrowing of accuracy was observed between the Linear and RBF kernels, As **Figure 9** shown, with accuracy scores of 0.909 and 0.887 respectively.

DOI: 10.54254/2755-2721/79/20241639



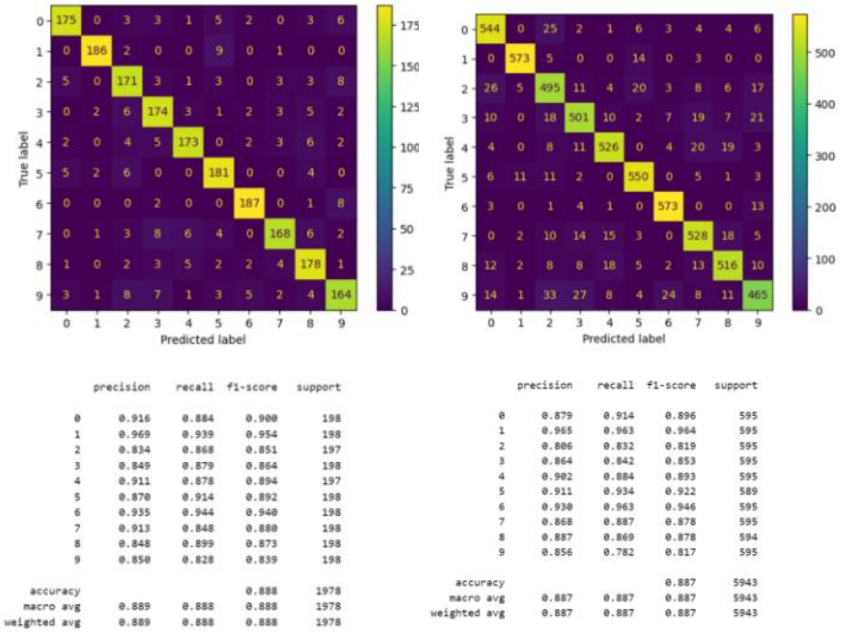
	precision	recall	f1-score	support
0	0.911	0.929	0.920	198
1	0.941	0.975	0.958	198
2	0.885	0.858	0.871	197
3	0.896	0.869	0.882	198
4	0.928	0.919	0.923	197
5	0.872	0.929	0.900	198
6	0.931	0.955	0.943	198
7	0.917	0.889	0.903	198
8	0.934	0.934	0.934	198
9	0.878	0.838	0.858	198
accuracy			0.910	1978
macro avg	0.909	0.909	0.909	1978
weighted avg	0.909	0.910	0.909	1978

	precision	recall	f1-score	support
0	0.879	0.914	0.896	595
1	0.965	0.963	0.964	595
2	0.806	0.832	0.819	595
3	0.864	0.842	0.853	595
4	0.902	0.884	0.893	595
5	0.911	0.934	0.922	589
6	0.930	0.963	0.946	595
7	0.868	0.887	0.878	595
8	0.887	0.869	0.878	594
9	0.856	0.782	0.817	595
accuracy			0.887	5943
macro avg	0.887	0.887	0.887	5943
weighted avg	0.887	0.887	0.887	5943

Figure 9. Accuracy comparison between 1-second and 3-second segmented datasets using SVM with RBF Kernel.

5.3.3. The Poly Kernel of SVM

In our endeavor to improve the accuracy of genre classification, we employed the Poly kernel in our SVM model, keeping the parameters at a default setting to mitigate the computational demands. Our team was keen to observe how the Poly kernel would perform in comparison to the Linear kernel, especially given the increased complexity it brings to the SVM model. The results from the Poly kernel, as **Figure 10** shown, despite its capability to handle non-linear data better, the accuracy levels between the datasets segmented into 3-second clips and 1-second clips remained nearly identical.



	precision	recall	f1-score	support
0	0.916	0.884	0.900	198
1	0.969	0.939	0.954	198
2	0.834	0.868	0.851	197
3	0.849	0.879	0.864	198
4	0.911	0.878	0.894	197
5	0.870	0.914	0.892	198
6	0.935	0.944	0.940	198
7	0.913	0.848	0.880	198
8	0.848	0.899	0.873	198
9	0.850	0.828	0.839	198
accuracy			0.888	1978
macro avg	0.889	0.888	0.888	1978
weighted avg	0.889	0.888	0.888	1978

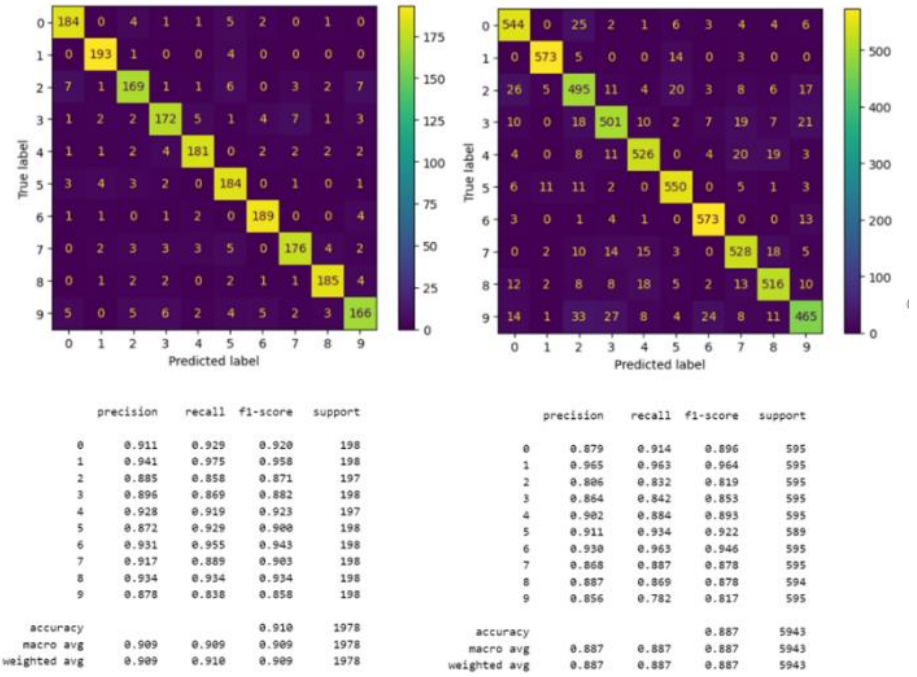
	precision	recall	f1-score	support
0	0.879	0.914	0.896	595
1	0.965	0.963	0.964	595
2	0.806	0.832	0.819	595
3	0.864	0.842	0.853	595
4	0.902	0.884	0.893	595
5	0.911	0.934	0.922	589
6	0.930	0.963	0.946	595
7	0.868	0.887	0.878	595
8	0.887	0.869	0.878	594
9	0.856	0.782	0.817	595
accuracy			0.887	5943
macro avg	0.887	0.887	0.887	5943
weighted avg	0.887	0.887	0.887	5943

Figure 10. Accuracy comparison between 1-second and 3-second segmented datasets using SVM with Poly Kernel.

DOI: 10.54254/2755-2721/79/20241639

5.3.4. The LinearSVC

Our team can calculate much faster with LinearSVC than with SVC and the kernel passed in the linear parameter. As expected, as **Figure 11** shown, the performance is worse than the RBF kernel, however, with limited differences.



	precision	recall	f1-score	support
0	0.911	0.929	0.920	198
1	0.941	0.975	0.958	198
2	0.885	0.858	0.871	197
3	0.896	0.869	0.882	198
4	0.928	0.919	0.923	197
5	0.872	0.929	0.900	198
6	0.931	0.955	0.943	198
7	0.917	0.889	0.903	198
8	0.934	0.934	0.934	198
9	0.878	0.838	0.858	198
accuracy			0.910	1978
macro avg	0.909	0.909	0.909	1978
weighted avg	0.909	0.910	0.909	1978

	precision	recall	f1-score	support
0	0.879	0.914	0.896	595
1	0.965	0.963	0.964	595
2	0.806	0.832	0.819	595
3	0.864	0.842	0.853	595
4	0.902	0.884	0.893	595
5	0.911	0.934	0.922	589
6	0.930	0.963	0.946	595
7	0.868	0.887	0.878	595
8	0.887	0.869	0.878	594
9	0.856	0.782	0.817	595
accuracy			0.887	5943
macro avg	0.887	0.887	0.887	5943
weighted avg	0.887	0.887	0.887	5943

Figure 11. Accuracy comparison between 1-second and 3-second segmented datasets using LinearSVC.

5.4. Four-Beat Segmentation Experiment

Building upon the original 1-second and 3-second segmentation experiments, we introduced an experiment based on 4-beat segmentation to further explore the impact of segmentation length on model performance. Utilizing Logistic Regression and Random Forest models, we conducted training and validation on the new 4-beat segmented dataset. The results demonstrated a certain improvement in classification accuracy for some music genres in the 4-beat segmentation experiment. This suggests that finer-grained segmentation might capture richer features within the audio data, thereby aiding in enhancing the accuracy of music genre classification.

6. Discussion

6.1. Insights from Data Segmentation

Initially, the premise of segmenting audio clips into smaller durations stemmed from the notion of capturing more granular features which, we hypothesized, would lead to more accurate classifications. However, the results presented a different narrative. The 3-second segments exhibited superior performance compared to the 1-second segments. This led us to a deeper contemplation on the role of musical theory, particularly the concepts of tempo, beats, and bars, in data segmentation.

6.2. Musical Theory and Data Segmentation

Our subsequent discussions revolved around the interplay between musical theory and data segmentation. It emerged that a static segmentation approach might disrupt the inherent rhythmic similarity within the music data, potentially leading to less accurate comparisons. The tempo, determining the length of a bar, appeared as a more harmonious metric for segmentation. We conjectured

DOI: 10.54254/2755-2721/79/20241639

that a dynamic segmentation approach, aligning with the music's tempo and rhythm, could potentially foster more accurate classifications.

6.3. Exploratory Steps for Dynamic Segmentation

To explore this conjecture, we considered the following steps:

1. Modifying the code used for 1-second segmentation to dynamically calculate segmentation duration based on the bpm (beats per minute), thereby determining the length of a bar. We planned to test segmentation lengths equivalent to 4 bars.
2. Utilizing the newly generated datasets to re-run our SVM models and analyze the performance.

Our underlying hypothesis posited that dynamic segmentation, synchronized with the music's rhythm, would preserve the rhythmic similarity and hence lead to more precise comparisons and classifications.

7. Conclusion

7.1. Summary of Findings

Through rigorous experimentation, our research delved into the nuanced domain of music genre classification by employing various machine learning models. The initial segmentation of 3-second clips, mirroring the original Kaggle project, was further enhanced by introducing a novel 4-beat segmentation. This new segmentation aimed at aligning data segmentation more closely with the inherent rhythmic structure of the music, thereby potentially capturing more granular features pivotal for accurate genre classification.

The SVM models, with different kernel types, showcased a commendable performance in genre classification tasks, with the 3-second segmented dataset yielding a notably better accuracy compared to the 1-second segmented dataset when using a linear kernel. The additional segmentation of 4-beat intervals emerged as a promising avenue for fine-grained feature extraction, with results indicating its potential for enhancing the accuracy of genre classification [6].

7.2. Implications

The findings from our study underscore the significance of appropriate data segmentation and feature extraction in music genre classification tasks. It suggests that aligning segmentation with musical structure, such as the 4-beat segmentation, could potentially improve the model's understanding and characterization of different music genres.

7.3. Future Work

1. Extended Dataset: To further validate the findings, extending the dataset to encompass a broader range of genres and sub-genres could provide a more robust platform for model evaluation and generalizability.
2. Advanced Segmentation Techniques: Exploring other segmentation techniques based on musical theory or signal processing could unveil more intricate patterns pivotal for genre classification.
3. Deep Learning Approaches: The implementation of deep learning architectures like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) could possibly uncover deeper insights into the data and enhance classification accuracy.
4. Real-time Classification: Progressing towards the development of a real-time music genre classification system could open avenues for practical applications such as real-time music recommendation or automated DJ systems.
5. Hybrid Models: Investigating hybrid models that combine the strengths of traditional machine learning models with deep learning could potentially lead to improved accuracy and better understanding of feature interactions in music genre classification.
6. Through the prospective avenues outlined, the study aims to build upon the foundational understanding garnered, striving towards more accurate and efficient music genre classification models that could significantly contribute to the burgeoning field of music information retrieval.

DOI: 10.54254/2755-2721/79/20241639

Acknowledgement
Zixiao Xie, Ziyan Zhao, Jiaze Fu, and Xintao Tian contributed equally to this work and should be considered co-first authors.

References
[1] Gupta, S., Marwah, S., & Briskilal, J. (2022). AI Music Generator. Journal of Pharmaceutical Negative Results, 67-71.
[2] Bahuleyan, H. (2018). Music genre classification using machine learning techniques. arXiv preprint arXiv:1804.01149.
[3] Ghildiyal, A., Singh, K., & Sharma, S. (2020, November). Music genre classification using machine learning. In 2020 4th international conference on electronics, communication and aerospace technology (ICECA) (pp. 1368-1372). IEEE.
[4] Lau, D. S., & Ajoodha, R. (2022). 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: ICICT 2021, London, Volume 4 (pp. 239-247). Springer Singapore.
[5] Xu, J., Qu, W., Li, D., & Zhang, C. (2022, December). The Music Generation Road from Statistical Method to Deep Learning. In 2022 IEEE 8th International Conference on Computer and Communications (ICCC) (pp. 1628-1633). IEEE.
[6] Deepaisarn, S., Chokphantavee, S., Chokphantavee, S., Prathipasen, P., Buaruk, S., & Sornlertlamvanich, V. (2023). NLP-based music processing for composer classification. Scientific Reports, 13(1), 13228.