Stacking-based Model Fusion for Music Classification

RunQing,Wang
Nanjing University of Posts and Telecommunications
School of Computer Science and Technology
Student number: 1023041109

***Abstract*—In the context of artificial intelligence, tasks such as music classification and emotion recognition have become pivotal technologies in fields like music retrieval, recommendation, and therapy. However, due to the diversity and complexity of music, the task of music classification poses significant challenges, requiring consideration of various music genres, some of which may exhibit subtle differences, while others may be blends of multiple genres. This study contributes significantly to the field of music classification in several aspects. Firstly, we propose and validate an innovative music classification method named Stacking. By stacking the outputs of multiple machine learning models, this method markedly improves the accuracy of the model, excelling in metrics such as accuracy, precision, recall, and F1 score. Secondly, through in-depth data exploration and visualization analysis of audio data, we uncover the inherent features and distribution patterns of music data, providing a foundation for subsequent modeling. Lastly, we comprehensively consider multiple machine learning and deep learning algorithms, such as MLP, GBDT, ResNet, XGBoost, Random Forest, and LightGBM, conducting comparative experiments to thoroughly evaluate their performance in music classification tasks. Through this series of work, we offer effective avenues for improvement in music classification tasks and make unique contributions to methodology and experimental comparisons.**

***Index Terms*—machine learning; stacking; music classification; deep learning**

I. Introduction

In the rapidly evolving realms of data science and machine learning, the remarkable success achieved by integrating big data and deep learning has become increasingly apparent. This success is propelled by the persistent and urgent demand for handling vast datasets and conducting precise data mining. This need holds significance not only in academic research but also plays a pivotal role in industrial applications. Machine learning services have become ingrained in our daily lives, enabling a deeper understanding of phenomena through extensive data utilization and the accomplishment of various complex functionalities, thereby contributing to societal progress.

Within the field of music, tasks such as music classification and sentiment analysis have emerged as crucial technologies for applications like music retrieval, recommendation systems, and music therapy. The diversity and complexity of music pose significant challenges in music classification tasks, requiring consideration of multiple music genres, some with subtle differences and others representing blends of several genres. To address this challenge, researchers have employed various machine learning models and methods, continually enhancing the accuracy and universality of music classification through experiments.

In this study, we employ an innovative approach – stacking fusion based on multiple machine learning models – aimed at further improving the accuracy of music classification. We compare multiple algorithms, including stacking fusion, MLP, GBDT, ResNet, XGBoost, Random Forest, LightGBM, etc., to comprehensively evaluate their performance in music classification tasks. Through in-depth analysis of experimental results, we find that the stacking fusion algorithm excels in accuracy, precision, recall, and F1 score metrics. Specifically, the stacking algorithm achieves the highest accuracy at 92.60%, with corresponding precision, recall, and F1 scores also reaching 92.60%, highlighting its outstanding performance in music classification.

In addition to focusing on the classification performance of algorithms, we also pay special attention to the comprehensive comparison of training time and model parameters. This not only aids in gaining a holistic understanding of the training efficiency of different algorithms but also provides a more comprehensive reference for selecting appropriate algorithms, balancing accuracy and efficiency in the implementation of music classification tasks. Through these comprehensive comparisons, we aim to provide valuable guidance and insights for future research in music classification, with the anticipation that this series of work will contribute to a deeper understanding and breakthroughs in the development of the music classification field.

This study has made various contributions to the field of music classification, primarily manifested in the following aspects:

- Proposal and Validation of the Stacking Method: Innovatively, this paper proposes a stacking method based on multiple machine learning models for music classification tasks. By stacking the output results of different algorithms, this method effectively enhances the model's accuracy. Experimental validation of the stacking method demonstrates its outstanding performance across various metrics such as accuracy, precision, recall, and F1 score. Achieving the highest accuracy, it provides an effective path for improving music classification tasks.
- Comparative Experiments with Multiple Algorithms: This paper comprehensively considers multiple machine learning and deep learning algorithms, including MLP, GBDT,

ResNet, XGBoost, Random Forest, LightGBM, among others. Comparative experiments are conducted to assess their performance in music classification tasks. This series of comparisons aids in a deeper understanding of the strengths and weaknesses of each algorithm, providing crucial insights for selecting the most suitable algorithm for music classification.

- Data Exploration and Visualization Analysis: Through in-depth data exploration and visualization analysis of audio data, we reveal the intrinsic features and distribution patterns of music data. This contributes to a better understanding of the diversity and complexity of music data, laying the foundation for subsequent modeling.

The remaining structure of the paper is as follows: Chapter 2 reviews current research in music classification. Chapter 3 defines the research questions, while Chapter 4 details our proposed music classification method. Chapter 5 explores data and conducts model comparison experiments, and Chapter 6 concludes the study.

II. Related Works

This section will provide a brief overview of the existing work and achievements in recent years related to music classification. Keunwoo Choi [1] proposed a convolutional recurrent neural network (CRNN) for music tagging. CRNNs take advantage of convolutional neural networks (CNNs) for local feature extraction and recurrent neural networks for temporal summarization of the extracted features. Yang Yu [2] proposed a new model incorporating with attention mechanism based on a Bidirectional Recurrent Neural Network. Furthermore, two attention-based models (serial attention and parallelized attention) are implemented in this paper. Compared with serial attention, parallelized attention is more flexible and gets better results in our experiments. Especially, the CNN-based parallelized attention models taking STFT spectrograms as input outperform the previous work. Mohsin Ashraf [3] proposed a hybrid architecture of CNN and variants of RNN such as long short-term memory (LSTM), Bi-LSTM, gated recurrent unit (GRU), and Bi-GRU. Yun-Ning Huang [4] proposed a novel method for leveraging pre-trained speech models for low-resource music classification based on Neural Model Reprogramming (NMR). Hang Zhao [5]proposed S3T, a self-supervised pre-training method with Swin Transformer for music classification, aiming to learn meaningful music representations from massive easily accessible unlabeled music data. Junfei Zhang [6] proposed a novel approach using visual spectrograms as input and proposed a hybrid model that combines the strength of the Residual neural Network (ResNet) and the Gated Recurrent Unit (GRU). This model is designed to provide a more comprehensive analysis of music data, offering the potential to improve the music recommender systems through achieving a more comprehensive analysis of music data and hence potentially more accurate genre classification.

III. Problems

The objective of the music classification task is to automatically recognize audio data as belonging to specific music genre categories through the extraction of features from audio signals and model training. To rigorously define this task, we introduce the following symbols:

- Training data set: $D_{\text{train}} = (X_i, y_i)i = 1^{N\text{train}}$, where X_i represents the i-th audio The feature representation of the file, y_i is the corresponding music genre label.
- Test data set: $D_{\text{test}} = (X_j, y_j)j = 1^{N\text{test}}$, where X_j represents the jth audio The feature representation of the file, y_j is the corresponding music genre label.
- Model parameters: θ represents the parameters of the music classification model.
- Model prediction function: $f_\theta(X)$ represents the prediction function of the music classification model with parameter θ for the input audio feature X.

The music classification task can be formalized as an optimization problem aiming to minimize the loss function $L(y,\ f_\theta(X))$ to learn the optimal model parameters θ, where L denotes the loss function, and y represents the true music genre labels. The training process can be expressed by the following formula:

$$\min_{\theta} \sum_{i=1}^{N_{\text{train}}} L\left(y_i, f_\theta\left(X_i\right)\right) \tag{1}$$

The trained model $\hat{f}_\theta$ can be utilized for music genre classification of audio files in the test set. In order to thoroughly assess the model performance, we employ metrics including classification accuracy, precision, recall, and F1-score as indicators of model effectiveness.

The specific definitions of performance evaluation metrics are as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

where:

- True Positives (TP): The number of samples correctly predicted as positive by the model.
- False Positives (FP): The number of samples actually negative but incorrectly predicted as positive by the model.
- False Negatives (FN): The number of samples actually positive but incorrectly predicted as negative by the model.
- True Negatives (TN): The number of samples correctly predicted as negative by the model.

Furthermore, model training time and the number of model parameters are considered as measures of temporal and spatial complexity for the model. The results are shown in the section.

IV. Stacking-based model fusion for music classification

We presents a exploration of ensemble stacking as a powerful fusion technique for improving music classification accuracy. As Fig.1 shows, a collection of diverse machine learning models, including K-Nearest Neighbors (KNN), XGBoost, Gradient Boosting Decision Trees (GBDT), LightGBM, and Random Forest, are amalgamated through the stacking paradigm. The final ensemble model is constructed using a meta-classifier, in this case, the XGBoost classifier, to synthesize the collective wisdom of the individual models.

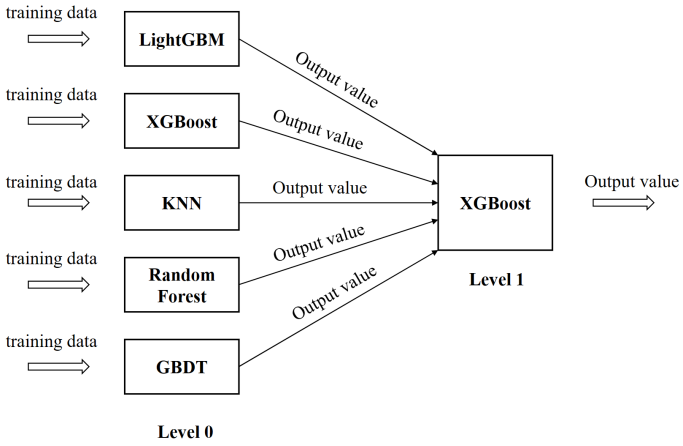

Fig. 1. Stacking-based model fusion

The ensemble stacking framework is implemented using the scikit-learn library in Python. The base models selected for stacking include KNN, XGBoost, GBDT, LightGBM, and RandomForest. These models are chosen for their diverse learning strategies and ability to capture different aspects of music features.

The stacking classifier is instantiated with a list of base estimators and a final estimator, which serves as the meta-classifier responsible for making the ultimate prediction. The selected meta-classifier, XGBoost, is known for its robust performance and adaptability to various datasets.

V. Evaluation

This section primarily outlines the setup of the music recognition experiments, explores the data, and presents performance comparative experiments based on multi-class algorithms.

A. Setup

The experimental setup utilized a server equipped with an Intel Core i7-10th Gen processor and an RTX 2060 GPU. Software tools included JupyterLab as the code editor, Sklearn for machine learning modeling, Matplotlib for data visualization, and Tensorflow 2.1 coupled with Keras for the construction and training of deep learning models.

The dataset chosen for this study consists of audio files and utilizes the publicly available GTZAN dataset. This dataset comprises 10 distinct genres of songs, namely blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock. Each genre includes 100 songs numbered from 0 to 99, resulting in a total of one hundred songs per category. The GTZAN dataset is widely used in the field of Music Genre Recognition (MGR) research and evaluation. These files were collected between 2000 and 2001 from various sources, including personal CDs, radio broadcasts, and microphone recordings, to represent diverse recording conditions.

B. Exploratory data analysis

1) Waveform Visualization: The data preprocessing involves loading audio files and outputting basic information about them, including audio data, audio amplitude, sampling rate, and estimated audio length. Subsequently, the audio files undergo processing to remove leading and trailing silence from the audio signal. Information about the processed audio file, including the data and amplitude after removing silence, is then output.

To visualize the waveform of the processed audio file, a graphical representation is created. A figure object is instantiated with a size of 12×4 inches, utilizing the librosa.display() function. The waveform graph is generated based on the audio file's data and sampling rate, and the resulting graph is saved. The format of the audio waveform visualization is depicted in the figure below:

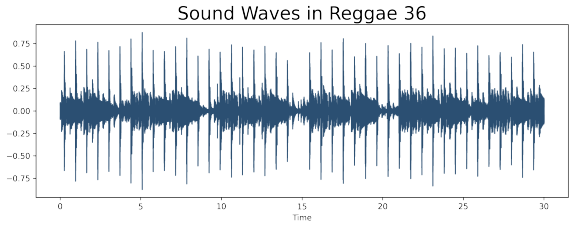

Fig. 2. Waveform Visualization

The image displays the original waveform graph of a music segment. The original waveform graph visualizes the amplitude variations of the audio signal over the time axis. It illustrates how the amplitude of the audio signal changes over time, providing an intuitive observation of the sound intensity, waveform shape, and characteristics of the audio signal. From the graph, we can observe that the amplitude range of the music waveform is between -1 and 1, and the duration is 30 seconds.

2) Mel Spectrogram: The Mel Spectrogram is a graphical representation of audio features across frequency and time, presenting the audio signal in the frequency domain by mapping its spectrum to the Mel frequency domain using Mel filters. The Mel Spectrogram is an effective feature representation method in audio signal processing. For this purpose, we utilize the Mel Spectrogram.

Firstly, the metal audio file is loaded, and a silence removal process is applied. Subsequently, the librosa.feature() function is employed to compute the Mel Spectrogram of the audio file, and it is converted into a spectrogram in decibels (dB) for a better display of the audio signal's features across frequency and time. A color bar is added to the plotted spectrogram to represent the dB color mapping. Finally, the generated Mel Spectrogram of the metal audio file is saved, as shown in the figure.3. Following the aforementioned steps, the Mel Spectrogram of the classical audio file is depicted in the figure.4

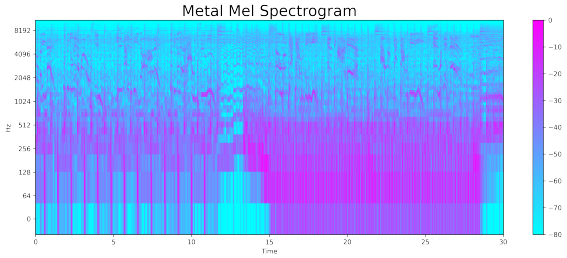

Fig. 3. Mel Spectrogram

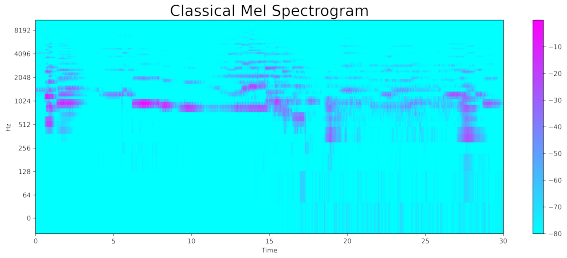

Fig. 4. Classical Mel Spectrogram

3) Harmonic: What humans actually perceive is the fundamental pitch played by musical instruments and a synthesized sound composed of a series of frequencies. Harmony is a feature that humans cannot discern, and data analysis can visually display the harmonics (audio components whose frequency is integer multiples of the fundamental frequency) that make up the harmony for better music classification. Decompose audio files into harmonic and non-harmonic components and plot their waveforms. First, use the librosa.effects() function to separate the harmonic and non-harmonic components of the audio file. Then, draw the waveform diagram of the harmonic component in purple and the waveform diagram of the non-harmonic component in orange. Finally, save the drawn waveform graph. The waveform diagram of the drawn audio file is shown in the figure below, which separates the harmonic and non-harmonic components for a better understanding of the harmonic and non-harmonic parts of the audio file.

4) Chroma: The chroma feature categorizes meaningful pitches into twelve classes, typically closely aligned with the equal temperament scale. Chroma features may be associated

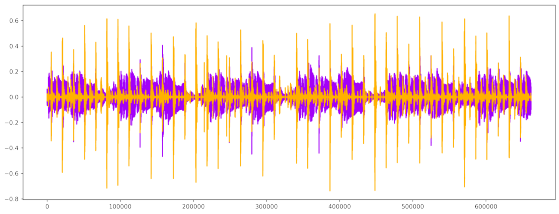

Fig. 5. Harmonic

with tonality and pitch, with a key focus on capturing harmonic and melodic characteristics in music, demonstrating robustness to timbral and instrumental variations. In audio signal analysis, chroma features are employed to differentiate sounds of different instruments, identify music styles, and perform audio classification tasks. We attempt to extract chroma features from the data, utilizing the librosa.feature() function to compute the chromagram of the audio file, representing the distribution of the audio signal in terms of pitch. The librosa.display() function is used to visualize the chromagram, where the horizontal axis represents time, and the vertical axis represents pitch. Finally, the generated chromagram is saved. The chromagram aids in analyzing the tonal characteristics of the audio signal. The plotted chromagram is depicted in the figure below:

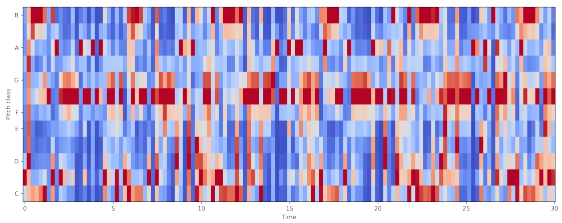

Fig. 6. Chroma

5) PCA: Principal Component Analysis (PCA) is a commonly used technique for data dimensionality reduction and feature extraction. Its objective is to project high-dimensional data into a lower-dimensional space through linear transformations, preserving the most significant information in the data. We preprocess the loaded data, extract features and labels, normalize the features, and then employ PCA to reduce the dimensionality to two principal components. The reduced data is transformed back into a DataFrame, and the variance ratios explained by the principal components are computed and output. A scatter plot of the PCA-transformed data is created using the Seaborn library, where the x-axis and y-axis represent the two principal components. Principal Component 1 is plotted on the x-axis, and Principal Component 2 is plotted on the y-axis. Different colors are used to represent data points with different labels. Finally, the generated scatter plot is saved. The plotted scatter plot of the data is depicted in the figure below:

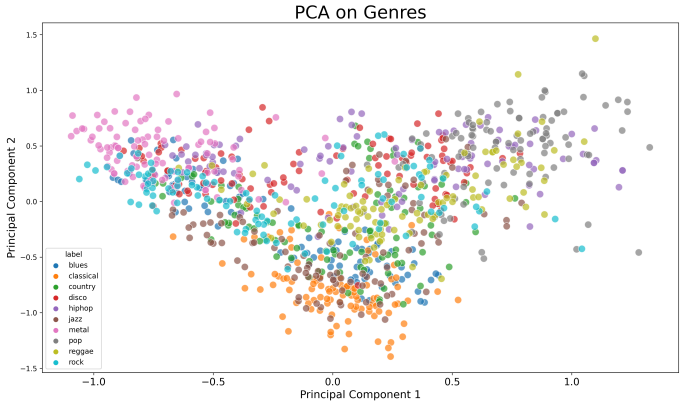

Fig. 7. PCA

C. Results of music classification

The purpose of this comparative experiment is to assess the performance of different machine learning algorithms in music classification. We employed a variety of classical algorithms, including MLP, LightGBM, XGBoost, KNN, RandomForest, ResNet, GBDT, CART, GaussianNB, and BernoulliNB, along with a model fusion method implemented through Stacking.

1) Performance Evaluation:: As shown in Table I, the experiment conducted a comprehensive performance evaluation of music classification algorithms, comparing their performance in terms of accuracy, precision, recall, and F1 score. The Stacking algorithm demonstrated outstanding performance, achieving the highest accuracy and overall performance score of 92.60%. MLP closely followed, with an accuracy of 91.17%, indicating relatively good classification performance. LightGBM and XGBoost algorithms exhibited excellent performance in accuracy and other evaluation metrics, reaching around 90%.

However, the deep learning model ResNet, while achieving a relatively high accuracy (85.22%), experienced a significant increase in training time and model parameter count, requiring more computational resources. In contrast, the Naive Bayes algorithms (GaussianNB and BernoulliNB) showed relatively poor overall performance, especially in accuracy and F1 score.

TABLE I
COMPARATIVE RESULTS OF MUSIC CLASSIFICATION BASED ON ML&DL

Algorithm	Accuracy	Precision	Recall	F1-score
Stacking	**92.60%**	**92.62%**	**92.60%**	**92.60%**
MLP	91.17%	91.20%	91.17%	91.16%
LightGBM	90.29%	90.37%	90.29%	90.29%
XGBoost	89.57%	89.71%	89.57%	89.56%
KNN	87.02%	87.35%	87.02%	86.95%
RandomForestr	86.47%	86.68%	86.47%	86.38%
ResNet	85.22%	85.29%	85.22%	85.23%
GBDT	82.98%	83.16%	82.98%	82.92%
CART	63.39%	63.18%	63.39%	63.25%
GaussianNB	52.14%	53.23%	52.14%	50.29%
BernoulliNB	48.20%	46.57%	48.20%	45.00%

2) Time and Space Complexity:: As depicted in Table II, the assessment of time and space complexity revealed distinct features of each algorithm in terms of training time and model parameter count. Stacking and MLP algorithms exhibited relatively long training times, at 521.56 seconds and 438.05 seconds, respectively. In comparison, KNN, GaussianNB, and BernoulliNB algorithms showed relatively efficient training times but with relatively poorer performance.

The deep learning model ResNet had an advantage in terms of model parameter count but required more computational resources. Conversely, LightGBM and XGBoost algorithms had relatively fewer model parameters and achieved a balance between training time and performance.

TABLE II
TRAINING TIME AND MODEL PARAMETERS OF MODELS

Algorithm	Training Time	Model Parameters
Stacking	521.56	136
MLP	438.05	16166
GBDT	88.23	20
ResNet	62.23	56621
XGBoost	8.71	27
RandomForestr	3.05	18
LightGBM	2.87	20
CART	0.41	12
KNN	0.01	8
GaussianNB	0.01	2
BernoulliNB	0.01	4

In conclusion, the choice of the most suitable algorithm depends on the specific requirements of the music recognition task, considering factors such as accuracy, training time, and computational resources. Stacking, MLP, and LightGBM are highlighted as top performers in this comparative analysis.

VI. CONCLUSION

This study contributes significantly to music classification in several key areas. The introduction of the innovative Stacking method successfully boosted model accuracy to an impressive 92.6%. This method's validation underscores its effectiveness in leveraging diverse algorithms for improved music classification, a valuable asset for practical applications. In data exploration, our thorough analysis of audio data unveiled essential features and distribution patterns. This not only enhanced our understanding of music diversity and complexity but also laid a solid foundation for constructing more effective classification models. Through comprehensive experiments comparing multiple algorithms, including MLP, GBDT, ResNet, XGBoost, Random Forest, and LightGBM, we provided insightful evaluations. This analysis shed light on the strengths and weaknesses of each algorithm, offering practical guidance for researchers and practitioners in making informed choices tailored to their specific tasks.

In summary, our study, employing innovative methods, in-depth data exploration, and thorough algorithmic comparisons,

provides valuable insights for advancing research and applications in music classification, contributing positively to the development of related fields.

REFERENCES

[1] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.

[2] J. Pons and X. Serra, "Randomly weighted cnns for (music) audio classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 336–340.

[3] C. Xu, N. Maddage, and X. Shao, "Automatic music classification and summarization," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 441–450, 2005.

[4] Y.-N. Hung, C.-H. H. Yang, P.-Y. Chen, and A. Lerch, "Low-resource music genre classification with cross-modal neural model reprogramming," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[5] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, "S3t: Self-supervised pre-training with swin transformer for music classification," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 606–610.

[6] J. Zhang, "Music genre classification with resnet and bi-gru using visual spectrograms," 2023.