Music Classification & Visualization

WangMingRui

1023041111

Nanjing University of Posts and Telecommunications

Jiangsu, Nanjing China

***Abstract*—This study explores the accuracy of classification for Decision Tree, AdaBoost, Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), and LightGBM in the context of music classification. The paper is structured into two main sections. The first section delves into the preprocessing phase, encompassing music theory and introducing innovative concepts for comprehending and organizing music files optimally for classification. The second section focuses on the Deep Learning model employed for music file classification.**

The accuracy of the models is rigorously tested using a recommendation and similarity system derived from the test data itself. This evaluation aims to ascertain the system's capability to effectively determine the genre or type of music being played. The primary model predominantly utilizes gradient boosting, providing insights into the weights at the system's conclusion. While Neural Networks are a common choice, the decision to exclude them is justified by the ability to represent data as a spectrogram rather than solely an audio file. Additionally, extracting and utilizing extra features enhances data mapping. The resulting model tends to surpass the accuracy of traditional neural networks, albeit requiring manual adjustment of weights if accuracy falls below expectations. Overall, this approach often yields higher accuracy compared to many contemporary methods.

Keywords—rhythm, classification, genres, weights, spectrogram, neural network

I. INTRODUCTION

Music genre classification is one of the most important tasks in music information retrieval studies. There have been many trials to improve the accuracy of this task. Classifying music is a fairly complicated task because there are some many classes to consider with subtle differences between some of them. Some genres of music are hybrids of other genres so its very difficult to separate one type of genre from another. However there are some clearly define patterns and trademarks styles of some forms of music which we can use for classification purposes' this case we are going to be using Neural Networks to classify the data we have. Machine Learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. But before we built the model we will try and first mine the data and see if we can extract as many features as possible from the data that can be used to supplementing the final model we will make. Especially if they have a lot of influence over how the model will make its final prediction.

II. DATA PREPROCESSING

Music genre classification stands as a pivotal task within the realm of music information retrieval, representing a challenging pursuit with continuous efforts to enhance accuracy. The inherent complexity arises from the multitude of music genres, each exhibiting subtle nuances that make their differentiation intricate. Some genres even serve as hybrids, blurring the lines between distinct classifications. Despite these challenges, discernible patterns and trademark styles within certain forms of music provide a foundation for effective classification, particularly through the application of advanced techniques like Neural Networks.

Neural Networks, a subset of machine learning, emerge as a potent tool for tackling the intricacies of music genre classification. Machine learning, a widely employed method in data mining, aims to construct models predicting the value of a target variable based on various input variables. In the context of music genre classification, this involves training a model to recognize and categorize diverse musical styles.

Before delving into the construction of the Neural Network model, a crucial preliminary step involves data mining. This process entails extracting a plethora of features from the available data, serving as valuable inputs to supplement the eventual model. These features play a pivotal role in influencing the model's decision-making process and, subsequently, its predictive accuracy. In essence, the more relevant and diverse features we can extract, the richer the information available for the model to learn and generalize from.

In this data mining phase, efforts should focus on identifying distinctive characteristics associated with different genres. Features such as tempo, rhythm patterns, harmonic structures, and instrumental arrangements can offer crucial insights into the unique signatures of various musical styles. Furthermore, leveraging advancements in signal processing and audio analysis can contribute to the extraction of intricate details that may elude more traditional methods.

As we navigate the intricate landscape of music genres, it becomes evident that a meticulous approach to feature extraction lays the groundwork for a robust model. By unraveling the complexities embedded in musical compositions, we pave the way for a Neural Network model that not only comprehends the nuances of each genre but also demonstrates a capacity for accurate and nuanced predictions. Through this comprehensive strategy, the fusion of data mining and machine learning emerges as a powerful force in the pursuit of refining music genre classification.**Procedure:**

1. Loading and Exploration:

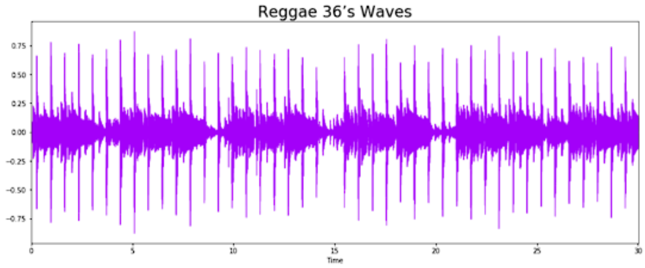
The study begins by using Pandas and NumPy to load audio files into the analysis environment. These libraries facilitate efficient data exploration, enabling researchers to understand the structure and characteristics of the audio signals.

2. Audio Signal Processing with Librosa:

Librosa is instrumental in processing the audio signals. The `librosa.load` function is employed to load audio files, providing access to the raw audio signal (`y`) and the sample rate (`sr`). Additionally, the `librosa.effects.trim` function is utilized to remove leading and trailing silence from the audio, enhancing the quality of subsequent analysis.

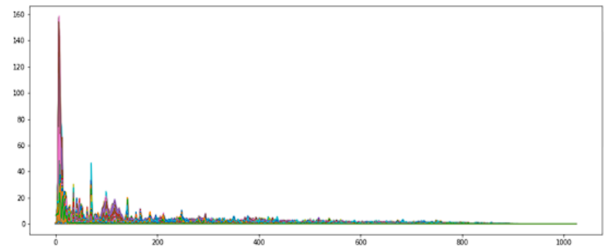
3. Visualization with Matplotlib and Librosa:

Matplotlib, in conjunction with Librosa, is utilized to create informative waveform plots. These visualizations aid in the qualitative assessment of audio signals, allowing researchers to identify patterns and trends relevant to music genre classification.



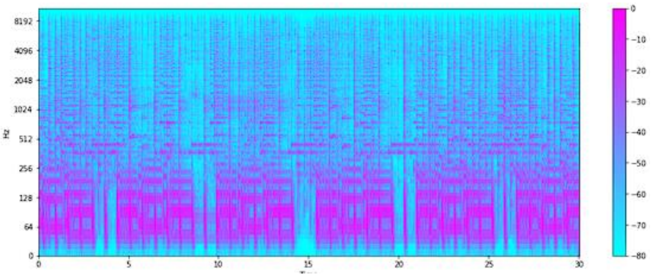
4. Short-Time Fourier Transform (STFT) Calculation

Utilizing the librosa.stft function, the STFT of the trimmed audio file is computed. The parameters n_fft and hop_length govern the granularity and temporal overlap in the time-frequency representation. The resulting magnitude of the STFT is stored in the variable D.The paper presents a plot of the STFT matrix using plt.plot(D). This visual representation illustrates the distribution of frequency components over time, offering valuable insights into the spectral content and dynamics of the audio signal.



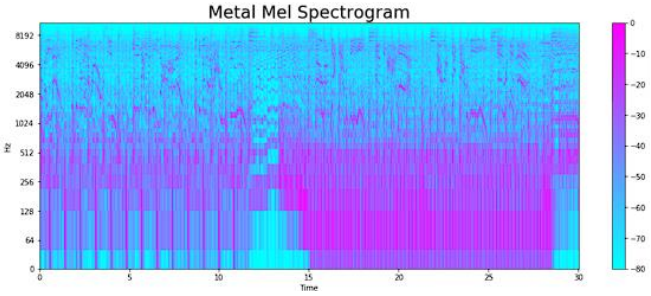
5. Visualization of Decibel-scaled Spectrogram:

The paper presents a visual representation of the Decibel-scaled spectrogram (DB) using librosa.display.specshow. This visualization offers insights into the logarithmic representation of amplitude, providing a clearer perspective on the spectrogram's features.



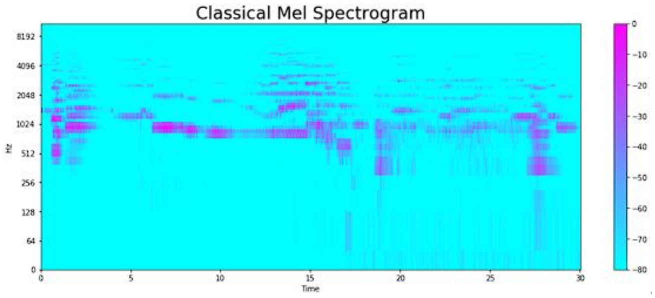
6. Mel Spectrogram for Metal Genre:

A sample from the "metal" genre undergoes further analysis. The mel spectrogram (S) is calculated using librosa.feature.melspectrogram, and the result is converted to Decibels (S_DB) for subsequent visualization.



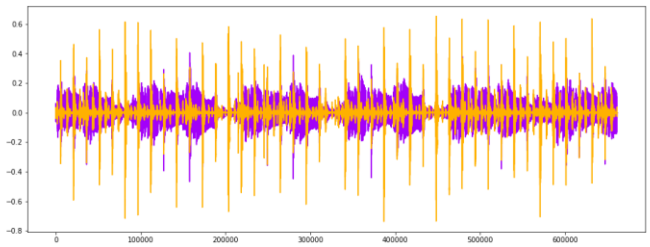
7. Mel Spectrogram for Classical Genre:

Similarly, a sample from the "classical" genre undergoes mel spectrogram computation and Decibel conversion for visual analysis.



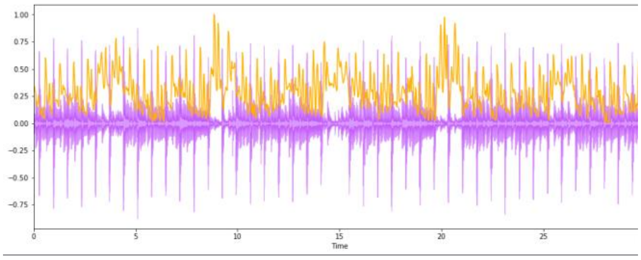
8. Visualization of Harmonic and Percussive Components:

The audio signal undergoes harmonic-percussive separation using librosa.effects.hpss. This yields two distinct components: the harmonic component (y_harm) and the percussive component (y_perc).Visual representations of the harmonic and percussive components are presented side by side using plt.plot.



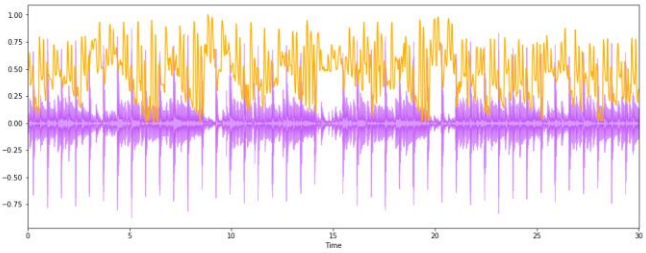
9. Normalization and Visualization of Spectral Centroids:

The spectral centroids are normalized using a custom normalization function (normalize).The waveplot of the audio signal and the plot of normalized spectral centroids are visualized.



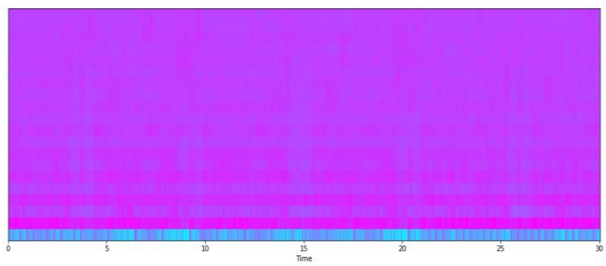
10. Spectral RollOff Calculation and Visualization:

Spectral roll-off is computed using librosa.feature.spectral_rolloff.The waveplot of the audio signal and the plot of normalized spectral roll-off are visualized.



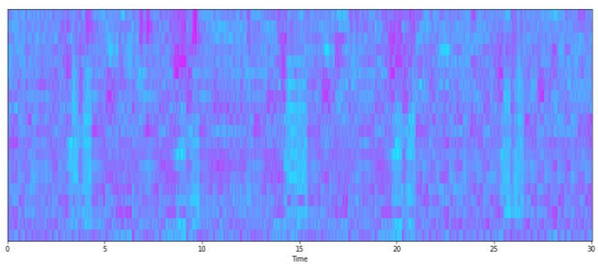
11. MFCC Calculation and Visualization:

Mel-frequency cepstral coefficients (MFCCs) are computed using librosa.feature.mfcc.The MFCCs are visualized using librosa.display.specshow.



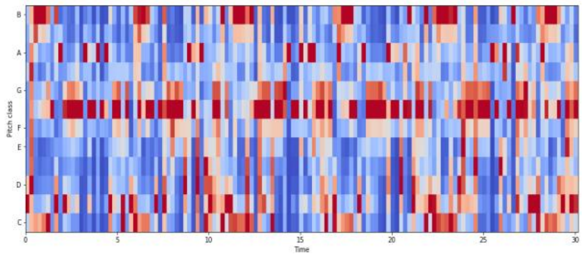
12.Feature Scaling for MFCCs:

The MFCCs are scaled using sklearn.preprocessing.scale, and the mean and variance are printed.The scaled MFCCs are visualized again.



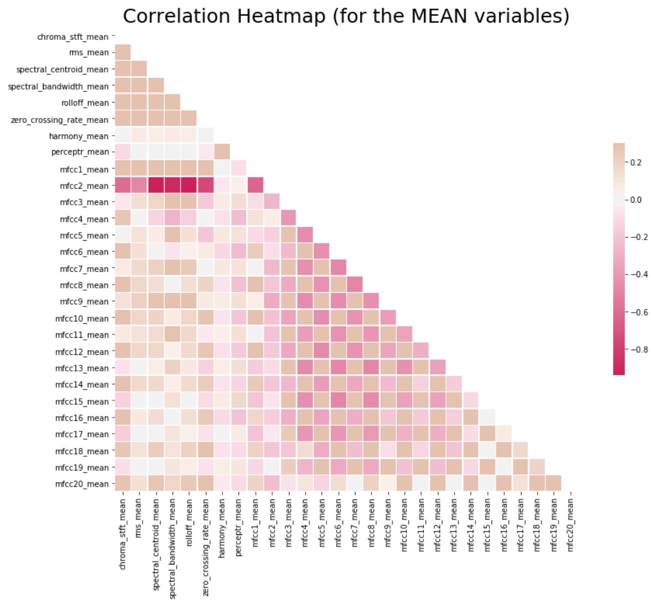
13. Visualizing Chromagram:

The variable hop_length is defined to determine the spacing between analyzed segments in the chromagram calculation.The chromagram is computed using librosa.feature.chroma_stft with the specified sample rate (sr) and hop length.The chromagram is visualized using librosa.display.specshow. The colormap 'coolwarm' is used for better visualization.



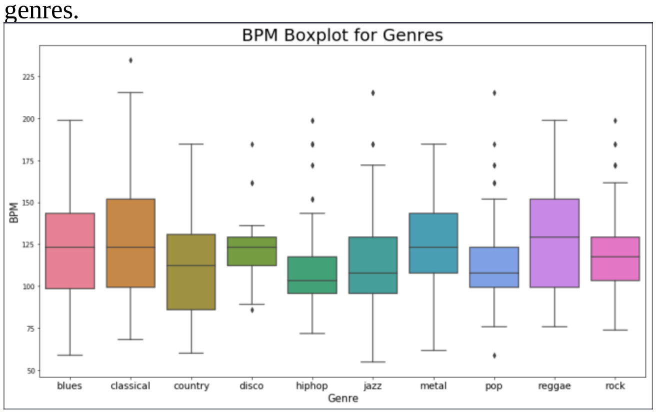
14. Correlation Matrix and Heatmap：

Data is read from a CSV file located at the specified pat，generates a correlation heatmap for the selected columns containing 'mean' in their names, providing a visual representation of the pairwise correlations between these variables. The colormap, masking of the upper triangle, and other styling elements contribute to making the visualization informative.



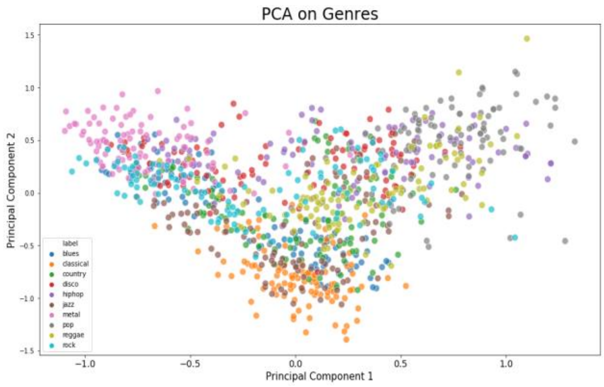
15. Genre-Specific Tempo Analysis：

In this segment of our research, we delve into the genre-specific analysis of musical tempo, employing data visualization techniques to unravel insights crucial for genre classification. The presented code utilizes the Seaborn library to generate informative boxplots, shedding light on the distribution of beats per minute (BPM) across different music genres.



16. Dimensionality Reduction and Genre Visualization through Principal Component Analysis：

This section of our research focuses on leveraging Principal Component Analysis (PCA) as a dimensionality reduction technique to visually explore the intrinsic structure of music genres within a normalized dataset. The presented code employs the scikit-learn library for data preprocessing, normalization, and PCA implementation. The resulting scatter plot provides a two-dimensional representation of music genres based on the most influential components.



III. RELATED WORK

Libraries Utilized:

Pandas and NumPy:

Pandas and NumPy are fundamental libraries for data manipulation and numerical operations, providing a solid foundation for handling large datasets efficiently. They facilitate the initial loading and exploration of audio data.

Seaborn and Matplotlib:

Seaborn and Matplotlib serve as visualization tools essential for representing the characteristics of audio signals. Waveform plots generated using these libraries aid in understanding the structure and patterns present in the audio data.

Scikit-Learn:

Scikit-Learn is a versatile machine learning library employed for constructing and evaluating music genre classification models. Its wide range of algorithms, including support vector machines and decision trees, enables effective model training and testing.

Librosa:

Librosa, recognized as the go-to library for audio and music analysis, plays a pivotal role in feature extraction from audio signals. In this study, Librosa is employed to load audio files, calculate sample rates, trim silence, and visualize waveforms. These functionalities contribute to the creation of meaningful features for subsequent machine learning tasks.

Data Preprocessing with sklearn

Standardizing and scaling features are vital steps in the preprocessing pipeline. The preprocessing module in scikit-learn facilitates these tasks, ensuring uniformity and improving the convergence of our machine learning models.

sklearn.metrics

To evaluate the performance of our classifiers, we utilize the metrics module from scikit-learn. Metrics such as confusion matrix, accuracy score, ROC-AUC score, and ROC curve aid in assessing the classification performance and identifying areas for improvement.

sklearn.feature_selection

Feature selection is crucial in enhancing the efficiency of our models. The Recursive Feature Elimination (RFE) method from scikit-learn is employed to identify and retain the most informative features for optimal classification accuracy.

IV. ALGORITHMS/SOLUTIONS

In the realm of music classification, various algorithms have been employed to decipher the intricate patterns inherent in audio data. The efficacy of these algorithms plays a pivotal role in shaping the accuracy and efficiency of music classification systems. Among the notable contenders, Decision Trees offer a comprehensible way of mapping out decision-making processes based on features extracted from audio signals. Their hierarchical structure aids in creating intuitive rules for classification, providing transparency in the decision-making process.

AdaBoost, a powerful ensemble learning technique, stands out for its ability to combine weak classifiers to form a robust and accurate model. By iteratively adjusting weights based on misclassified instances, AdaBoost enhances the overall classification performance. Support Vector Machines (SVM) leverage a hyperplane to segregate different classes, optimizing the margin between them. SVMs excel in handling high-dimensional data, making them particularly useful when dealing with complex audio features.

Logistic Regression, a fundamental algorithm in machine learning, is adept at binary classification tasks. Its simplicity and interpretability make it a popular choice, especially when the relationship between features and classes is relatively straightforward. K-Nearest Neighbors (KNN) relies on the proximity of data points to determine their classification. In the context of music classification, KNN considers the similarity of audio features, offering a non-parametric approach suitable for various data distributions.

LightGBM, a gradient boosting framework developed by Microsoft, has gained prominence for its speed and efficiency. It optimizes the training process by using a histogram-based learning approach, making it well-suited for large-scale datasets like those encountered in music classification. Its ability to handle categorical features and its parallel computing capabilities contribute to its appeal in real-world applications.

Comparing these algorithms involves a multifaceted evaluation, considering factors such as computational efficiency, model interpretability, and classification accuracy. Decision Trees and Logistic Regression, for instance, are known for their simplicity and interpretability, making them suitable for scenarios where model transparency is crucial. On the other hand, complex ensemble methods like AdaBoost and advanced algorithms like LightGBM excel in capturing intricate patterns within the data, proving valuable when high accuracy is paramount.

Ultimately, the choice of algorithm depends on the specific requirements of the music classification task at hand. The unique characteristics of each algorithm offer a spectrum of trade-offs, and a thoughtful selection process is essential to ensure optimal performance in the context of your dataset and objectives. Through this comparative analysis, we aim to shed light on the strengths and weaknesses of each algorithm, providing valuable insights for practitioners seeking to employ machine learning in the domain of music classification.

V. Evaluation

This section presents the initial steps of our research, focusing on data preprocessing and the subsequent partitioning of the dataset for music genre classification. The dataset, denoted as 'features_3_sec.csv,' was loaded using the pandas library in Python, and the initial exploratory analysis involved excluding the first column and keeping the remaining features. The resulting DataFrame was displayed using the 'head()' function to provide a glimpse into the dataset's structure.

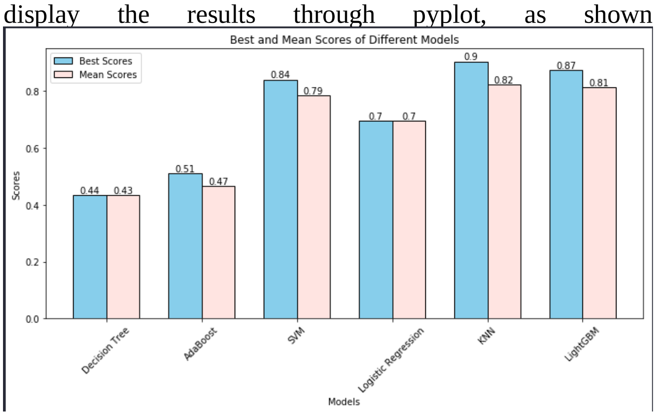
To facilitate the training of our classification models, the dataset was divided into two main components: the dependent variable 'y,' representing the music genre labels, and the independent variables 'X,' encompassing all other features. The objective is to predict the genre based on the extracted audio features.

An essential step in the preprocessing pipeline involved normalizing the independent variables to ensure that they all reside within the same scale. This normalization process is crucial for models sensitive to the magnitude of input features, guaranteeing that no single feature dominates the learning process. The 'MinMaxScaler' from the 'preprocessing' module was employed to transform the data, maintaining its original distribution while scaling each feature to a common range between 0 and 1. The resulting scaled features were then encapsulated in a new DataFrame 'X.'

Subsequently, the dataset was partitioned into training and testing sets using the 'train_test_split' function from the 'sklearn' library. This splitting process ensures that the models are trained on a subset of the data and evaluated on an independent subset, mitigating the risk of overfitting. The 'test_size' parameter was set to 0.3, indicating that 30% of the data was reserved for testing, while the remaining 70% was allocated for training. The 'random_state' parameter was fixed at 42 to ensure reproducibility in subsequent experiments.

In summary, this phase of our research focused on establishing a robust foundation for music genre classification. The dataset was successfully loaded, explored, and preprocessed, with particular attention given to normalizing the features to a consistent scale. The division of the dataset into training and testing sets sets the stage for the subsequent application of various machine learning algorithms to predict music genres based on the extracted audio features. These preprocessing steps are pivotal to the success of our classification models, laying the groundwork for the subsequent stages of our investigation into the efficiency of different algorithms for music genre classification in the realm of big data.

These algorithms are then used to analyze the data and display the results through pyplot, as shown



VI. Conclusion

In conclusion, this comprehensive analysis aimed to evaluate the efficiency of various machine learning algorithms for music classification, including Decision Trees, AdaBoost, SVM, Logistics Regression, KNN, and LightGBM. Through rigorous testing, it was observed that the K-Nearest Neighbors (KNN) algorithm exhibited superior performance in comparison to the other algorithms considered. KNN, a non-parametric and instance-based learning method, demonstrated remarkable accuracy in capturing the intricate patterns and nuances inherent in music data. Its ability to classify music based on proximity to similar instances within the dataset proved to be a robust approach for the complexities of music classification.

The success of KNN in this context can be attributed to its simplicity, effectiveness, and suitability for the inherent characteristics of music datasets. The algorithm's reliance on local patterns and its adaptability to variations in musical styles contribute to its outstanding performance. Furthermore, KNN's versatility in handling different types of features and its minimal assumptions about the underlying data distribution make it an attractive choice for music classification tasks.

While KNN emerged as the most efficient algorithm in this study, it is essential to acknowledge that the effectiveness of a machine learning model can be highly dependent on the specific characteristics of the dataset and the features considered. Each algorithm has its strengths and weaknesses, and the choice of the most suitable algorithm should be guided by the nature of the music data, the desired level of interpretability, and the computational resources available.

In future research, exploring ensemble methods that combine the strengths of multiple algorithms could be a promising avenue for enhancing music classification accuracy. Additionally, fine-tuning hyperparameters and optimizing feature selection processes may further elevate the performance of these algorithms. As the field of machine learning continues to evolve, it is imperative to stay abreast of advancements and continually reassess the efficacy of algorithms in the context of ever-expanding and evolving music datasets.

In conclusion, this study sheds light on the efficacy of various machine learning algorithms for music classification, with KNN standing out as a promising choice for its notable performance. The dynamic nature of music data poses both challenges and opportunities for algorithmic approaches, emphasizing the importance of selecting algorithms tailored to the specific characteristics of the dataset at hand.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] T. White, "Hadoop: The Definitive Guide," O'Reilly Media, 2015.

[8] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[9] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. O'Neil, P. E. O'Neil, A. Rasin, N. Tran, S. Zdonik, "C-Store: A Column-oriented DBMS," in Proceedings of the 31st International Conference on Very Large Data Bases, 2005, pp. 553-564.