

Music - Genre classification system using Machine learning

Varshit Manepalli
School of Engineering and Science
Stevens Institute of Technology
Hoboken, NJ, USA
vmanepal@stevens.edu

Eshan Kaushik
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

Kishan Gangarama
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

Abstract—By creating a trustworthy and effective system for automatic classification utilizing machine learning algorithms and audio attributes taken from a dataset of 8,000 recordings, this project seeks to address the issue of manual classification of music genres. Moreover, clustering strategies will be used to find commonalities among overlapping genres. The outcomes of this project could be applied to music-related applications including playlist makers, recommendation systems, and music search engines.

I. INTRODUCTION

It is a challenging task that might have a big impact on many music-related applications to automatically classify musical genres and identify similarities between them. The present manual classification techniques take a lot of time and are subjective, which causes genre classification to be inconsistent. In order to solve this issue, the project's goal is to create an accurate and effective system for automatically classifying genres using machine learning algorithms and audio attributes taken from a data set of 8,000 tracks.

The aim of the project is to investigate how well different machine learning algorithms, including decision trees, support vector machines, and deep learning models, can categorize different musical genres using audio parameters including spectral properties, rhythmic patterns, and tone. In addition, clustering strategies like k-means and hierarchical clustering will be used to find connections between various genres, particularly those that overlap or have comparable traits.

The ultimate objective of this project is to develop a reliable and precise system for automatically classifying music genres that can be incorporated into music-related applications, such as playlist generators, recommendation systems, and music search engines, to enhance user experience and deliver more impartial genre classification results.

II. RELATED WORK

Automatic music genre classification methods now in use can be generally divided into two primary categories as Feature-based methods and Deep learning methods.

Using machine learning algorithms, feature -based methods entail collecting audio properties from music songs, such as spectral features, rhythm patterns, and tone, and categorizing genres based on these features. Decision trees, support vector

machines, and k-nearest neighbors are a few of the frequently employed machine learning methods. These approaches frequently rely on manually created features and conventional machine learning techniques, which are easily understood and computationally effective. Unfortunately, they might not be able to capture subtle and complicated genre features, and they might perform worse when faced with huge and varied music data sets.

Without the requirement for manually created features, deep learning techniques use deep neural networks to learn hierarchical representations of audio features from the original raw audio data. In deep learning-based techniques for genre categorization, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are frequently employed. In capturing complex genre patterns and obtaining high accuracy in genre classification tasks, deep learning techniques have demonstrated promising results. Yet, because deep neural networks are black boxes, they may demand a lot of data for training, be expensive computationally, and be difficult to understand.

Tzanetakis and Cook's "A Survey and Experiments on Music Genre Classification using Audio Features and Machine Learning Algorithms" (2002), which explores various feature-based methods for music genre classification, and Oord et al "Deep 's content-based music recommendation" (2013), which suggests a deep learning-based method for music recommendation that incorporates genre classification, are two notable works in this field. Given the benefits and drawbacks of each approach, the development of the proposed system for automatic music genre classification in this project can be guided and inspired by these current solutions.

III. OUR SOLUTION

A. Description of Dataset

The Data set that will be used in this project is the free music archive dataset. It contains a sizable, carefully curated library of audio files and related metadata. About 100,000 audio tracks from diverse genres and styles—including jazz, rock, electronic, and more—make up the FMA data set. To ensure a wide variety of musical content for study, the tracks are drawn from a variety of artists and labels. The data set

contains audio files in both compressed and lossless forms, making it appropriate for various analysis.

Along with the audio files, FMA also offers comprehensive metadata, including track details, artist information, genre tags, album art, and more. This metadata enables in-depth analysis and insights into music structure, genre classification, and other music-related tasks. It also enables academics to investigate and evaluate numerous musical qualities, such as tempo, key, mode, and instrumentation. Overall, the FMA dataset is extensive and diverse, offering a wealth of information for music analysis. This information enables researchers and analysts to investigate different facets of music, create music recommendation systems, and advance the field of music analysis through machine learning and data-driven research.

B. Machine Learning Algorithms

In this study, we want to investigate several machine learning methods for categorizing music genres using audio attributes taken from the dataset. Convolutional neural networks, support vector machines, K-nearest neighbors, and clustering techniques are a few of the various algorithms we may take into account.

CNNs are suited for assessing audio aspects in music because of their well-known capacity to learn hierarchical representations from input data with spatial correlations. They have demonstrated promise in audio analysis tasks and have been effectively applied to image recognition challenges. For genre categorization, we might use a CNN architecture with several convolutional and pooling layers, followed by fully connected layers. In order to improve the performance of the model, we can experiment with various kernel sizes, pooling techniques, and activation functions.

Due to their capacity for managing high-dimensional data and ability to identify the best hyperplanes to divide classes, SVMs are a popular choice for classification problems. We can experiment with different parameters like regularization strength and kernel coefficients when using SVMs with various kernel functions, such as linear, polynomial, or radial basis function (RBF). In order to feed different feature representations into the SVMs for genre classification, we may also investigate spectrogram-based features like Mel-frequency cepstral coefficients (MFCCs).

KNN is a straightforward and understandable technique for classification tasks that works by locating the k nearest neighbors in the feature space and assigning the query sample the majority class label. To improve the performance of the model, we can experiment with various k values and distance metrics, such as Euclidean distance or cosine similarity. These parameters can then be changed while the model is being trained. To find similarities between various genres and increase the precision of genre categorization, we may use unsupervised clustering approaches in addition to supervised machine learning algorithms. For instance, we may combine related files based on their audio attributes using methods like K-means or hierarchical clustering, and then use the resulting clusters as further information for classification.

C. Implementation Details

We will employ a standardized procedure for evaluating the effectiveness of our machine learning models during implementation. With a standard ratio of 70-15-15, we will partition our dataset of 8,000 tracks into training, validation, and test sets. The validation set will be utilized for hyperparameter tuning and model selection, the training set for training the models, and the test set for assessing the final performance of the chosen model. We will employ methods like cross-validation, grid search, or randomized search to fine-tune the hyper-parameters of our models. In cross-validation, the training set is divided into several folds, with one fold serving as a validation set while the remaining folds are used to train the model.

We will also use regularization strategies like L1 or L2 regularization, early halting, and dropout to prevent overfitting. These methods can aid in enhancing our models' generalization abilities and stop them from memorizing training material. We also use feature engineering methods like feature scaling, feature selection, or feature augmentation to further enhance the performance of our models. Normalizing the input characteristics with feature scaling can help keep them from being too dominant during training. The dimensionality of the input features can be decreased and unnecessary or duplicate characteristics can be eliminated with the use of feature selection. In order to diversify the training data and strengthen the model's capacity to generalize to new data, feature augmentation may involve the addition of synthetic data.

Based on a number of criteria, including accuracy, precision, recall, F1-score, and confusion matrix, we will assess the performance of our models. Using the test set to get a final estimate of performance, we will choose the model with the best performance based on how it performed on the validation set.

IV. COMPARISON

This section includes the following: 1) comparing the performance of different machine learning algorithms that you used, and 2) comparing the performance of your algorithms with existing solutions if any. Please provide insights to reason about why this algorithm is better/worse than another one.

V. FUTURE DIRECTIONS

This section lays out some potential directions for further improving the performance. You can imagine what you may do if you were given extra 3-6 months.

VI. CONCLUSION

This section summarizes this project, i.e., by the extensive experiments and analysis, do you think the problem is solved well? which algorithm(s) might be better suitable for this problem? Which technique(s) may help further improve the performance?

Last but not the least, don't forget to include references to any work you mentioned in the report.

REFERENCES