Music Genre Classification

Laith Haddad

Spring 2024

1 Introduction

1.1 Motivation

Genre classification is an important task in audio signal processing and has wide applications in both academia and industry. By accurately categorizing music into different genres, this effort enables advanced music recommendation systems, personalized playlists, and efficient music collection management across digital platforms. In addition, it plays a central role in musicological studies, enabling researchers to analyze and understand cultural trends, genre developments, and artistic influences of different eras and geographical regions. In addition to recreational and scientific activities, music genre classification facilitates development in areas such as health and wellness, where music therapy and mood regulation can benefit from genre-specific customized playlists. Using datasets such as the GTZAN and FMA datasets, known for their coverage of various music genres, this project aims to develop robust machine-learning models that can accurately classify music, thus opening up new opportunities for creativity, research, and technological innovation, about music and more.

1.2 Background

The GTZAN dataset is a cornerstone in the field of music genre classification, serving as a benchmark for evaluating and developing genre classification systems. Comprising 1000 audio tracks, each 30 seconds long and categorized into 10 genres—blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock—this dataset provides a structured foundation for training machine learning models. Music genre classification involves extracting features such as timbral, rhythmic, harmonic, and dynamic elements from these tracks, and using algorithms like support vector machines, k-nearest neighbors, and convolutional neural networks to categorize the music accurately. Despite its limitations, including occasional repetition, mislabeling, and varying audio quality, the GTZAN dataset remains widely used, enabling advancements in automatic music recommendation systems, library

organization, and broadcasting. This synergy between a well-established dataset and sophisticated classification techniques underpins the continuous improvement and innovation in music information retrieval.

1.3 Problem definition

The input consists of raw audio files in WAV format, each 30 seconds long with a sample rate of 22,050 Hz and mono channels. These audio files undergo preprocessing, which involves segmenting the audio and extracting features such as MFCCs (Mel-Frequency Cepstral Coefficients), spectral centroid, spectral contrast, spectral rolloff, tempo (Beats Per Minute), and chroma features. The output provides a predicted genre label for each audio file (e.g., Blues, Classical, Country), along with confidence scores that indicate the probability distribution across potential genres (e.g., "Blues": 0.85, "Classical": 0.10, "Country": 0.05).

2 Related Work

2.1 Overview on Machine Learning for Music

The literature on machine learning applications in music spans several key texts, each offering distinct perspectives and methodologies. Dubnov and Greer [2023] explores the application of both deep and shallow machine learning techniques in the context of music and audio processing. This book provides a comprehensive overview of various methods and models, discussing their theoretical foundations and practical implementations. The authors delve into case studies and real-world applications, illustrating how these techniques can be used to solve complex problems in music analysis, synthesis, and classification. It is aimed at researchers and practitioners in the field, offering insights into the latest advancements and trends in machine learning for music and audio.

In contrast, Weihs et al. [2019] serves as a foundational guide to music data analysis, covering the essential concepts, methodologies, and tools used in the field. It addresses various aspects of music data, including audio signal processing, feature extraction, and machine learning techniques for music information retrieval. The authors discuss practical applications such as music recommendation systems, genre classification, and music similarity measures. This book is designed for students, researchers, and professionals interested in the analytical aspects of music data, providing both theoretical background and practical guidance.

Briot et al. [2019] focuses on the use of deep learning techniques for music generation, presenting a thorough exploration of the subject. It covers various deep learning models and architectures, such as recurrent neural networks (RNNs), generative adversarial networks (GANs), and variational autoencoders (VAEs), and their applications in creating music. The authors provide detailed explanations of how these models can be trained

and optimized to generate music that mimics human compositions. The book includes case studies and examples, making it a valuable resource for researchers and developers interested in the intersection of artificial intelligence and music creation.

Additionally, Schindler [2021] is dedicated to the topic of music genre classification using machine learning techniques. It covers a wide range of machine learning models, including traditional methods like support vector machines (SVM) and modern deep learning approaches such as convolutional neural networks (CNNs). The author discusses the challenges of feature extraction, dataset preparation, and model evaluation in the context of genre classification. Practical examples and case studies are provided to illustrate the application of these techniques in real-world scenarios. This book is suitable for researchers, students, and practitioners looking to understand and implement machine learning solutions for music genre classification.

2.2 Methodological Advances and Applications

In genre classification, researchers have explored diverse methodologies and applications in music analysis.

Tzanetakis and Cook [2002] focused on musical genre classification using audio signals, employing a dataset with diverse samples across genres. Their methods included Mel-frequency cepstral coefficients (MFCC) for feature extraction and support vector machines (SVM) for classification. Challenges included extracting genre-relevant features from audio signals. Their approach significantly improved classification accuracy, particularly through integrating MFCC features with SVM classifiers. Inputs were raw audio signals transformed into feature vectors, with outputs predicting genre categories.

In contrast, Pelchat and Gelowitz [2020] conducted music genre classification using neural networks on a large dataset of labeled music tracks, focusing on optimizing network architectures. Challenges included managing audio signal variability and preventing overfitting due to high feature dimensionality. Their study significantly improved classification accuracy compared to traditional methods, emphasizing an optimal deep neural network architecture tailored for this task. Inputs were preprocessed audio signals, with outputs being predicted music genre labels.

Oramas et al. [2018] integrates the MuMu dataset, combining data from Amazon Reviews and the Million Song Dataset (MSD) using MusicBrainz for accurate mapping. The dataset comprises 147,295 songs, 447,583 reviews, and cover art images, categorized with Amazon's genre taxonomy. Their methods include neural networks for audio processing, text aggregation from reviews, and Deep Residual Networks for image analysis. They faced challenges in data integration, feature extraction, and model optimization across modalities. By leveraging multimodal data, the study enhances genre classification, demonstrating superior performance over single-modal approaches. Results show significant accuracy gains by aggregating audio, text, and image features, with outputs predicting genre labels based on combined multimodal vectors.

Prabhakar and Lee [2023] focuses on music genre classification using efficient transfer and deep learning techniques on an extensive dataset of music tracks. Their methods include transfer learning to leverage pre-trained models and deep learning techniques to enhance genre classification accuracy. Challenges addressed efficient knowledge transfer and adaptation of pre-trained models to music data. Their approach successfully reduced training time and improved classification performance by employing transfer learning and fine-tuning deep learning models. They achieved excellent classification accuracy, highlighting the effectiveness of transfer learning in music genre classification. Inputs were pre-processed music data, and outputs were genre classifications.

Furthermore, studies like N. Farajzadeh [2023] developed PMG-Net, a specialized deep neural network for Persian music genre classification, using a dataset of categorized Persian music tracks. Their methods focused on adapting deep learning techniques to accommodate Persian music characteristics and ensuring accurate genre representation. Challenges included customizing models for Persian music and achieving precise genre classification. PMG-Net demonstrated superior classification performance, achieving high accuracy in classifying Persian music genres compared to traditional methods. Inputs were Persian music tracks, and outputs were predicted genre labels.

Kilickaya [2024] conducted genre classification and musical feature analysis using a dataset containing various musical features and genre-labeled tracks. The study involved analyzing these features and employing machine learning algorithms for genre classification. Challenges included extracting relevant features that effectively represented genre characteristics. The research focused on understanding and leveraging musical features to enhance genre classification accuracy. The best results were achieved through a combination of feature analysis and machine learning techniques, demonstrating high classification accuracy and validating the importance of feature analysis in music genre classification. Inputs were musical feature data, and outputs were genre classifications.

Similarly, Narkhede et al. [2024] conducted music genre classification using convolutional neural networks (CNNs) on a dataset of labeled music tracks. They utilized CNNs to exploit spatial hierarchies in spectrograms derived from audio signals. Challenges included optimizing CNN architecture for performance and complexity management to prevent overfitting. Their approach significantly improved genre classification accuracy, with the final CNN model, refined through extensive experimentation, achieving state-of-the-art performance. Inputs were spectrograms transformed from audio signals, and outputs were predictions of music genre labels.

Finally, Mogonediwa [2024] focused on training an artificial intelligence model for music genre classification using deep learning techniques. The study utilized a dataset of music tracks categorized into different genres. Challenges included managing large datasets, tuning hyperparameters, and mitigating overfitting. Their approach aimed to create a robust AI model capable of accurately classifying music genres, with the best results achieved through a well-tuned deep learning model, possibly employing CNNs or RNNs. The trained AI model demonstrated high accuracy in genre classification, surpassing traditional meth-

ods. Inputs were music track data, and outputs were predictions of genre classifications.

3 Data Description

The GTZAN and FMA datasets are widely used for music genre classification. This document provides an in-depth analysis of the dataset, including summary statistics, correlation, and various visualizations. We use two versions of the dataset: 3-second audio clips.

The GTZAN dataset is sourced from dat. It includes various audio features extracted from music tracks of different genres.

The FMA dataset is sourced from FMA, a large dataset for music analysis. It includes labeled tracks across multiple genres, allowing for genre classification tasks.

The dataset dimensions can be seen in Table 1.

Table 1: Dimensions of the datasets

Dataset	Number of Samples
3-second clips (GTZAN)	9990
3-second clips (FMA)	26000

3.1 Waveform

A waveform is a visual representation of an audio signal that displays how the amplitude of the sound changes over time. For the GTZAN dataset, which includes 30-second music clips across various genres, the waveform can show the temporal structure of these audio files. By plotting the amplitude on the vertical axis and time on the horizontal axis, one can observe the dynamics and intensity variations in the music. This visualization helps in understanding the rhythmic and structural patterns of the audio, which is useful for tasks like music genre classification (See Figures 1, 2, and 3).

3.2 Harmonic and Percussive

Separating the harmonic and percussive components of an audio signal provides a clearer analysis of its tonal and rhythmic elements. Using the librosa library in Python, you can load an audio file and apply the Harmonic-Percussive Source Separation (HPSS) technique. This process divides the audio into two parts: the harmonic component, which captures the tonal aspects like melodies and harmonies, and the percussive component, which highlights the rhythmic elements like beats and drums. Visualizing these components through waveforms helps in understanding the distinct structural features of the music (See Figures 4, and 5).

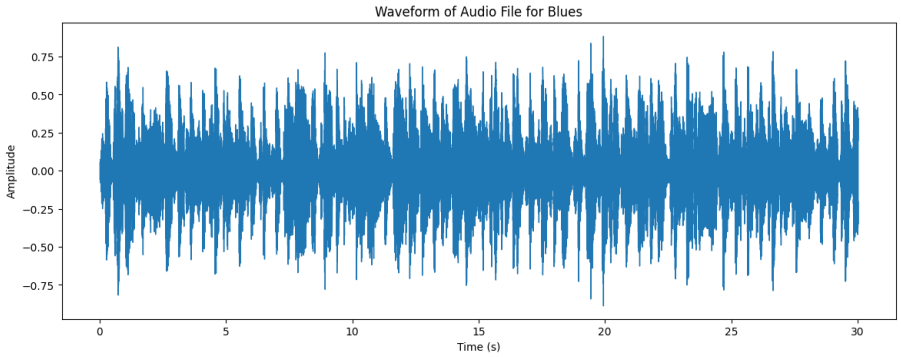

Figure 1: waveform of Audio File for Blues.

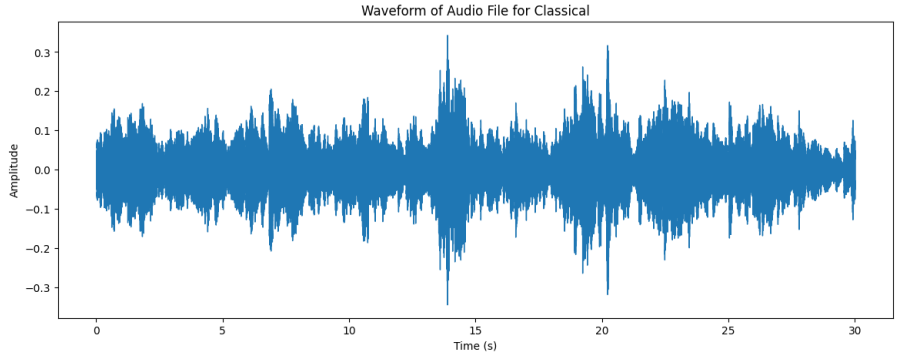

Figure 2: waveform of Audio File for classical.

3.3 Mel Spectrogram

A Mel spectrogram is a key tool in audio analysis, offering a representation of sound that aligns closely with human auditory perception. By converting an audio signal into the frequency domain using a Short-Time Fourier Transform (STFT), and then mapping these frequencies onto the Mel scale—a perceptual scale that mirrors the human ear's sensitivity to different frequencies—the Mel spectrogram captures essential auditory features. This process involves applying a series of triangular filters spaced according to the Mel scale and often converting the amplitude to a logarithmic scale for better perceptual accuracy. Widely used in applications such as speech recognition, music genre classification, and sound event detection, Mel spectrograms simplify and enhance the analysis of complex audio signals, making them indispensable in modern audio processing and machine learning tasks (See Figures 6, 7, and 8).

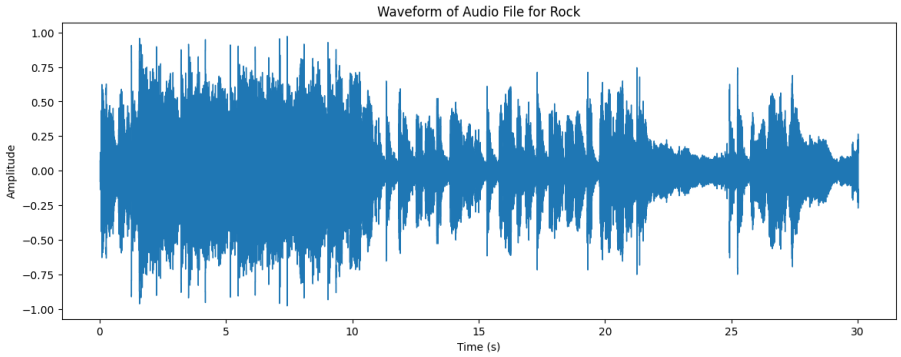

Figure 3: waveform of Audio File for rock.

3.4 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCCs) are essential features for audio analysis, capturing the short-term power spectrum of a sound. Derived from the Mel scale, which reflects human auditory perception, MFCCs are computed by taking the Fourier transform of an audio signal, mapping the power spectrum onto the Mel scale, taking the logarithm of the Mel frequencies, and applying the discrete cosine transform. This results in coefficients representing the audio's frequency content, useful for tasks like speech and music genre recognition. Using Python's librosa library, you can easily compute and visualize MFCCs to analyze audio characteristics (See Figure 9).

3.5 Distribution of RMS mean

To analyze the distribution of RMS mean values for audio segments of different lengths, we compute the RMS mean values for each segment type using a suitable audio processing library. By loading each audio file and dividing it into segments of specified lengths (e.g., 3 seconds and 30 seconds), we calculate the RMS mean for each segment and store these values separately. Visualizing the distributions with histograms reveals the frequency of RMS mean values for each segment duration, helping us understand the variability and central tendency of loudness in both short and long audio segments. This comparison provides insights into the overall loudness characteristics of the audio dataset across different time scales (See Figures 10, and 11).

3.6 Pair plot

A pair plot is a useful way to visualize the relationships between multiple features in a dataset. In this case, the features selected are RMS mean, spectral centroid mean, zero crossing rate mean, and tempo. The SNS pairplot function from the seaborn library creates a grid of scatter plots for each pair of features, while the hue parameter allows the points

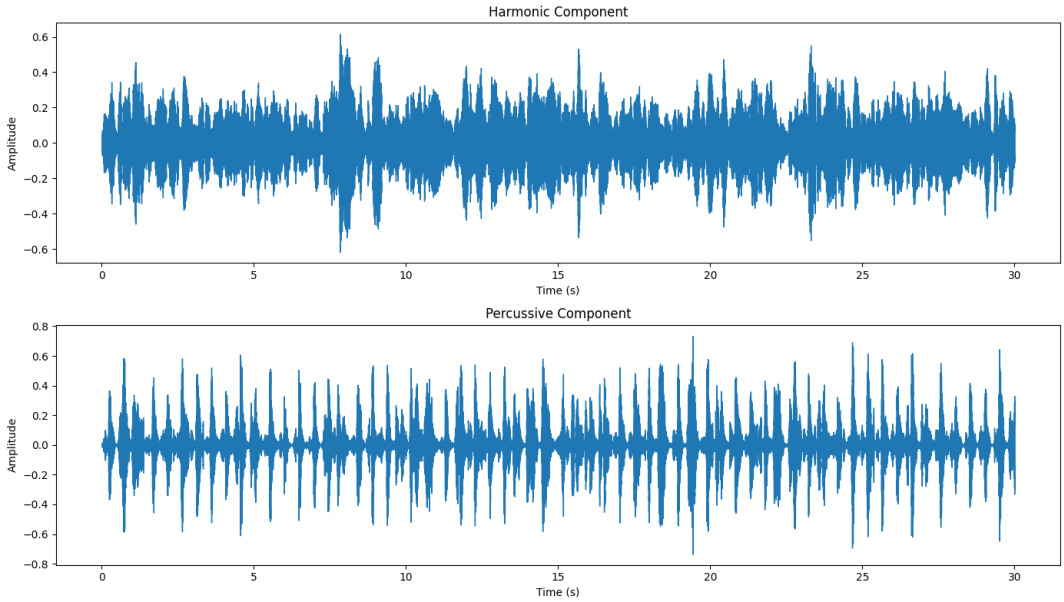

Figure 4: Harmonic and Percussive components for Blues.

to be color-coded based on the source column. This visualization helps in understanding how these features correlate with each other and with the source labels in the dataset (See Figure 12).

4 Proposed solution

4.1 Data Collection

For this project, we used two datasets, GTZAN and FMA (Free Music Archive), for developing a music classification model. GTZAN contains 1,000 songs labeled across 10 genres, namely, blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock. Each song is 30 seconds long. This dataset contains pre-labeled genre data and hence supports supervised learning approaches. Another filtered version of the FMA dataset was used to extend the data scope of the model. In that, again the same 10 genres were selected for consistency in both the models. Standardization was further improved by dividing each 30-second audio into samples of 3 seconds, thus enabling multiple inputs of one song in model training and testing. A genre-mapping scheme was adopted to harmonize the genre labels across the sets, which resulted in a well-balanced base for robust genre classification.

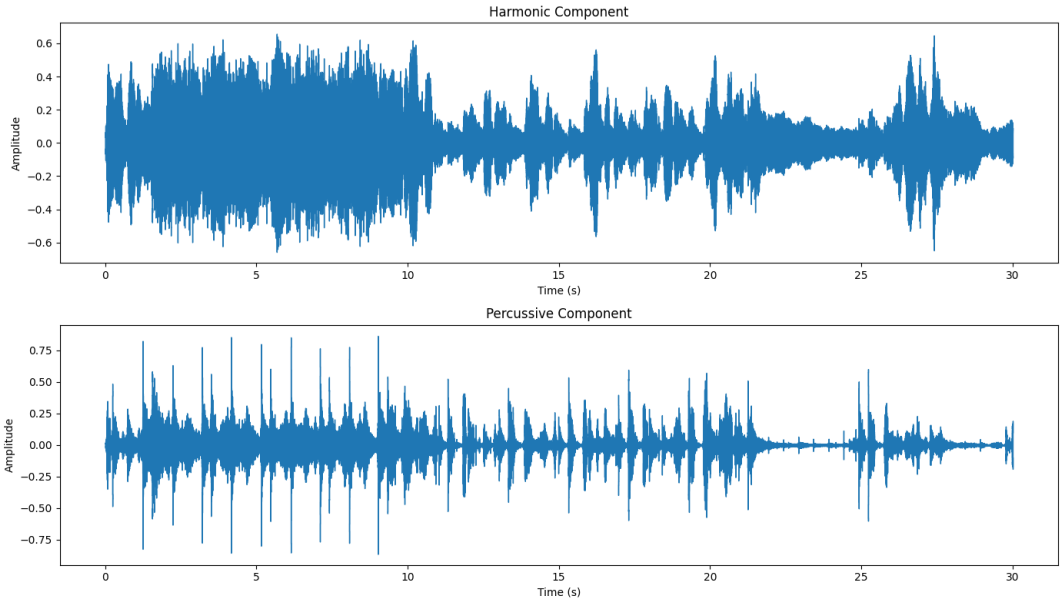

Figure 5: Harmonic and Percussive components for Rock.

4.2 Feature Extraction

Feature extraction was done using Mel Frequency Cepstral Coefficients, a feature extraction technique used in audio analysis, to convert the raw audio data into a more accessible and convenient format by machine learning models. For each 3-second segment extracted from the original 30-second audio clips, the 20 MFCCs necessary for the description of its essential features were computed, since the spectral properties are adequately represented. This was a crucial step, since MFCCs reduce the properties of sound into a compact form, highlighting perceptual features, which are in close correspondence to the way humans perceive music and speech. The focus on these coefficients may substantially enable the model to identify genre-dependent features, as MFCC encodes pitch, rhythm, and tone information that will help in distinguishing the music genres. This allowed the data to be represented in a lower-dimensional format while retaining most of the information, which further helped in improving computational efficiency and the model's generalizing ability across different genre types. The results are shown in Table 2.

4.3 Data Splitting

To improve the model performance in generalizing to new data, the balanced dataset was divided into separate training and testing sets. Overall, 80% would be the share of data for training, enabling the model to learn the pattern and feature across genres, while the

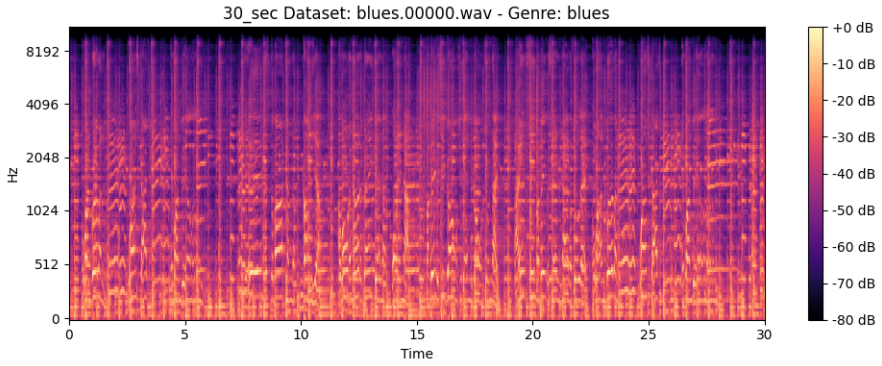

Figure 6: Mel Spectrograms for audio one blue label.

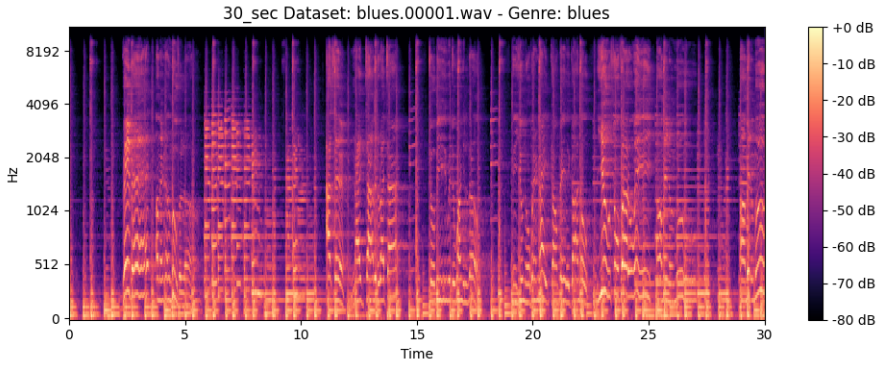

Figure 7: Mel Spectrograms for audio two blue label.

remaining 20% share was kept for testing to measure the performance of the model on testing data. This 80-20 split is a common practice in machine learning, as it provides sufficient data for training while retaining a substantial portion for unbiased performance assessment. By establishing a clear distinction between training and testing data, this approach helps ensure that the model's predictive abilities are robust and not merely memorized from the training set. The results are shown in Table 3.

4.4 Data Preprocessing and Standardization

After splitting the data, standardization was applied separately to the MFCC features in the training and testing sets. This ensures the training and testing distributions remain independent.

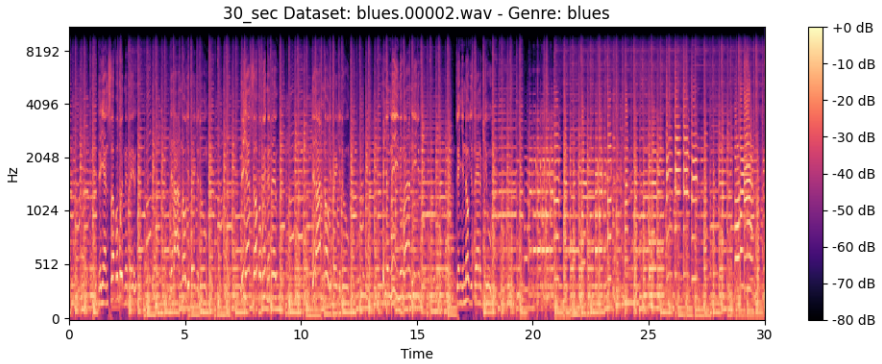

Figure 8: Mel Spectrograms for audio three blue label.

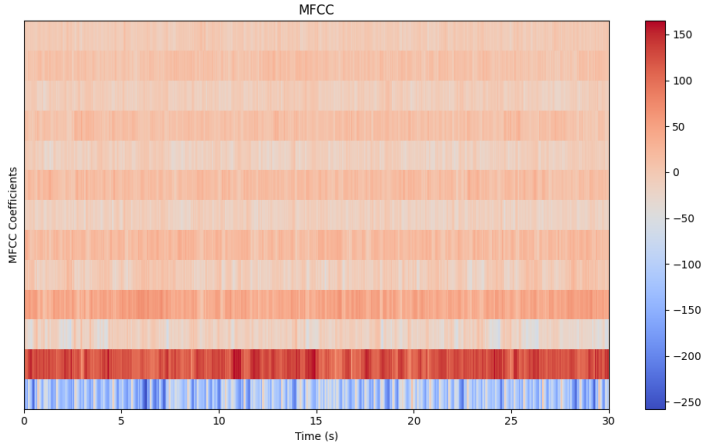

Figure 9: Mel-Frequency Cepstral Coefficientsn for Blues.

4.5 Class Weighting

Class weighting was then applied after preprocessing the data, in order to take care of genre imbalances. Instead of balancing the dataset by reducing the number of samples for overrepresented genres, class weighting adjusted the model's loss function to take these imbalances into consideration. It allowed the model to treat under-represented genres more equally, so that all genres were treated fairly during training. Class weighting helped to avoid bias while maintaining the full dataset for training, which can enhance the model's generalization capability.

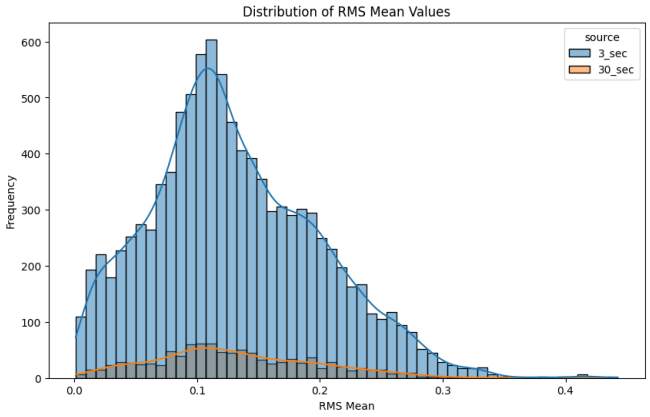

Figure 10: Distribution of RMS mean values.

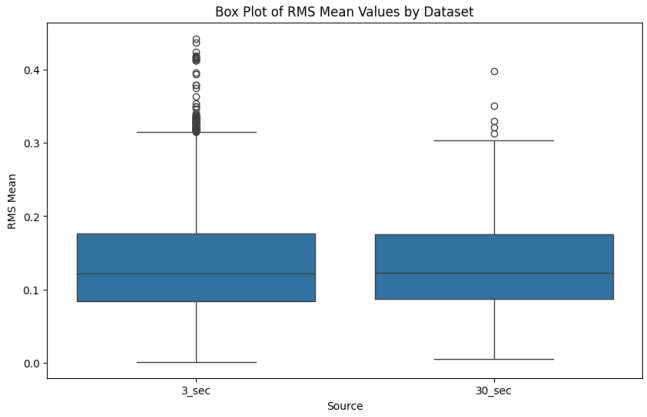

Figure 11: Box plot to compare feature distributions.

4.6 Model Selection and Training

A range of different machine learning algorithms from scikit-learn and deep learning models developed in TensorFlow were investigated to establish the best approach for genre classification. All models were compared on their performance according to the classification of music genres based on the set of MFCC features derived from audio. Traditional machine learning models tested initially included Support Vector Machines, Decision Trees, and Random Forests, along with a few deep learning architectures built in TensorFlow: neural networks. By doing some heavy comparisons among different models, the K-Nearest Neighbors algorithm from scikit-learn turned out to give the best performance for the classification task at hand by effectively modeling the genre-specific patterns within the dataset. It was then that the simplicity and reliance on structure in feature space allowed

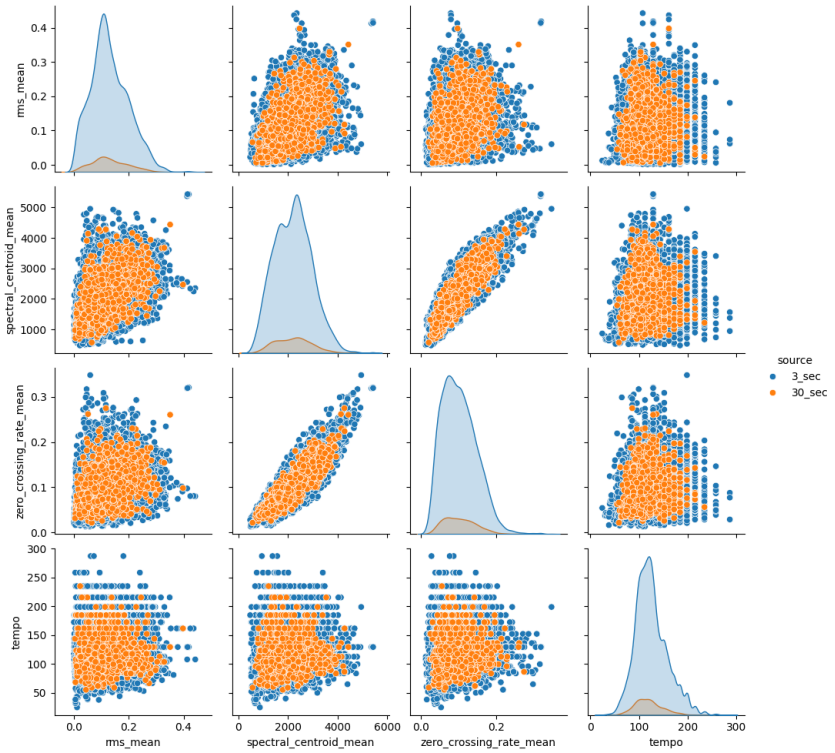

Figure 12: Pair plot of a few features.

KNN to do the best in identifying similarities of genres between samples. While models using TensorFlow were studied, too, KNN outperformed them on this particular task with a smaller computational budget, showing high accuracy in genre classification.

5 Experimental results

This section describes the experiments produced, and the results of those experiments to classify songs into their respective genres. The results are shown in Table 4.

5.1 KNN without Class Weighting

In the first experiment, we implemented the KNN algorithm without weighting the classes. Thus, it was trained on the raw, imbalanced dataset. The KNN algorithm treated all genres as equally important-with no allowances for class imbalance. This led to a test accuracy of 83% where the model performed commendably on most genres. However, the model showed a bias for the more represented classes because it had no mechanism to compensate for this class imbalance. Yet still, the KNN model generalized pretty well on

Table 2: MFCC Features Extracted from Audio Samples

MFCC_1	MFCC_2	...	MFCC_19	MFCC_20	Genre
-122.72	144.10	...	7.80	14.88	Blues
-87.50	69.30	...	-1.78	-2.67	Rock
-380.53	163.02	...	-10.53	-6.01	Classical
...	...	...	...	...	...

Table 3: Data Splitting

Dataset	Count
Train	30,000
Test	6,254

songs outside the original dataset, while poorly classifying those few represented genres. (See Figure 13).

5.2 KNN with Class Weighting

For the second experiment, we used class weighting in the KNN model to balance the dataset. We dynamically calculated the class weights during training and gave more importance to the minority genres. This helped the KNN model give more attention to underrepresented genres. We got a test accuracy of 86% for the KNN model. The model was able to classify all genres more effectively, especially the underrepresented ones, which boosted overall performance compared with the unweighted version. This experiment shows a way to internally weight classes so that KNN can adapt to class imbalances with strong generalization to songs outside the original dataset. (See Figure 14).

5.3 TensorFlow Model without Class Weighting

For the third experiment, we used an unweighted class network in TensorFlow that took input from this raw dataset. This setting allowed the model to give equal attention to all genres, and the model achieved an accuracy of 78% on the test set. The model demonstrated a good performance, but it was problematic when it came to classifying the minority genres, which got overpowered by the majority genre during training. Despite these challenges, the TensorFlow model was still in a position to make reasonably good predictions for songs outside the original dataset; however, the overall performance was lower than expected due to class imbalance. (See Figure 15).

Table 4: Model Performance Comparison

Model	**Accuracy**
KNN with Class Weighting	86%
KNN without Class Weighting	83%
TensorFlow Model with Class Weighting	74%
TensorFlow Model without Class Weighting	78%

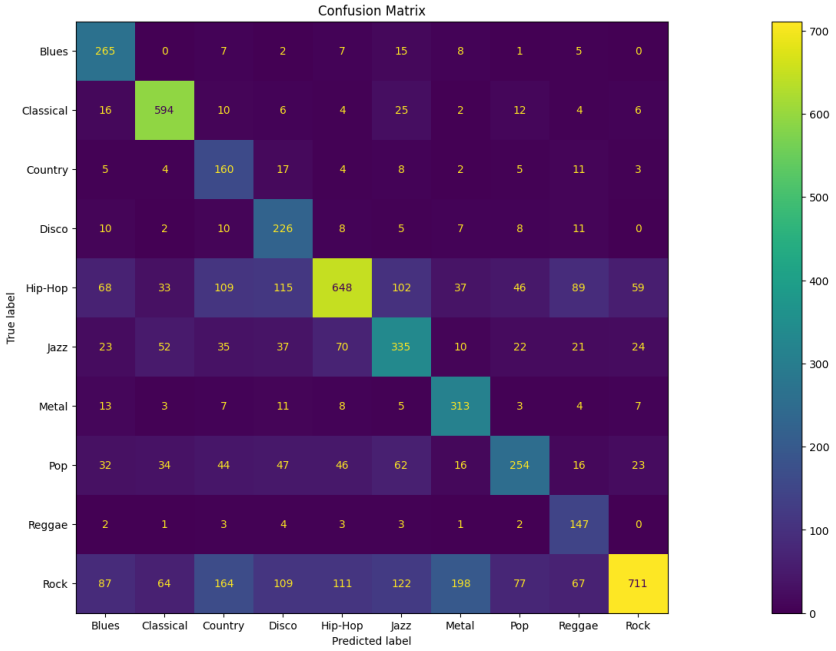

Figure 13: Confusion Matrix for KNN without Class Weighting.

5.4 TensorFlow Model with Class Weighting

Finally, for the fourth experiment, we trained the TensorFlow model with class weighting. This was done by applying the class weights to the loss function, which modified the model's training to give higher importance to the minority genres. Even after accounting for class imbalance, the test accuracy fell further to 74%. This suggests that even though class weighting improved the model's ability to classify underrepresented genres, it may have caused the model to struggle with the more prevalent genres. In this case, the TensorFlow model with class weighting still yielded good and reliable predictions for songs outside the original dataset but demonstrated a trade-off between improving generalization and high test accuracy. (See Figure 16).

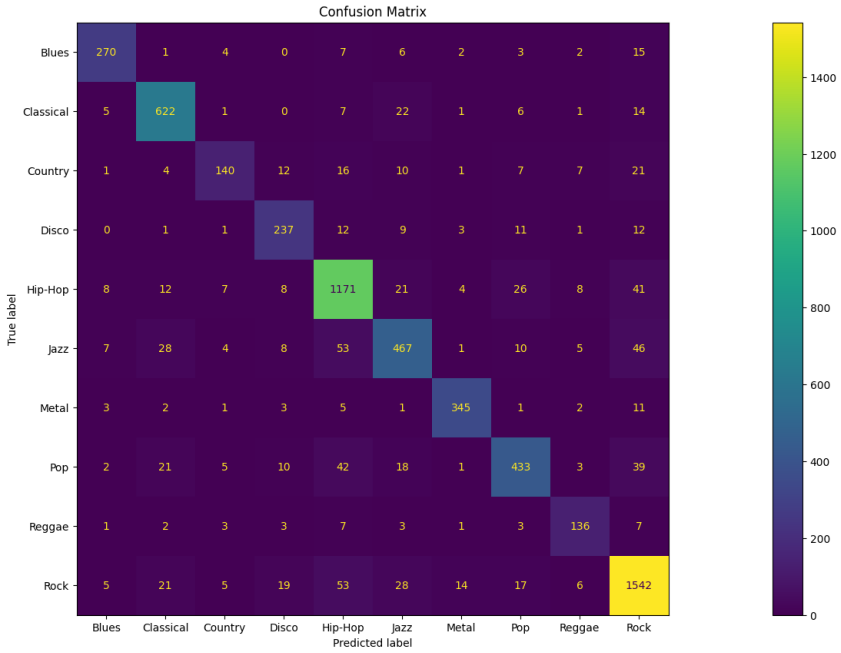

Figure 14: Confusion Matrix for KNN with Class Weighting.

5.5 Results with downloaded songs

To evaluate the performance of the trained model on songs outside the original dataset, a song was downloaded from YouTube and processed as follows: the audio was segmented into 3-second clips, with 20 Mel Frequency Cepstral Coefficients (MFCCs) extracted for each segment. These features were standardized to align with the preprocessing used during training, ensuring compatibility with the model. Each 3-second segment was then input into the model, which predicted a genre for each segment. A majority approach was applied to determine the overall genre of the song, selecting the most frequently predicted genre across all segments. For example, "Three Little Birds" was accurately classified as Hip-hop, matching its original genre. The results are shown in Table 5.

6 Conclusion

This work successfully proposes a music genre classification framework by leveraging robust data preprocessing, feature extraction, and machine learning techniques. Using the GTZAN and FMA datasets, we addressed challenges related to class imbalance and feature representation, ensuring a harmonized genre structure. The Mel Frequency Cepstral Coefficients extracted from the audio data served well in capturing essential spectral features

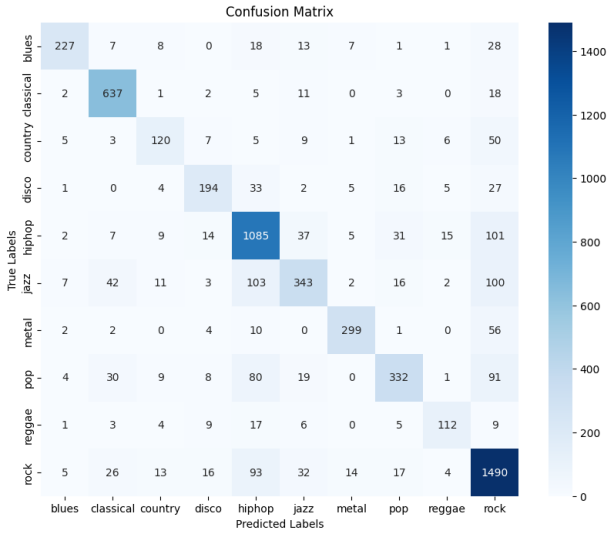

Figure 15: Confusion Matrix for TensorFlow Model without Class Weighting.

Table 5: Comparison of Predicted and Original Genres for KNN with Class Weighting

Song	**Predicted Genre**	**Original Genre**
The Thrill Is Gone	Rock	Blues
Spoonful	Rock	Rock, Blues
Chop Suey!	Metal	Metal, Pop, Rock
Waltz (No. 2)	Classical	Classical
Three Little Birds	Hip-hop	Hip-hop

that allowed the model to identify genre-specific patterns.

The main goal of developing a high-performance genre classification system was achieved. Within tested models, K-Neighbors Algorithm with class weighting has had the highest accuracy-86% and proved to be effective for class imbalance treatment tasks and showed good generalization. Besides, the experiments have shown trade-offs between various data balancing approaches and model architectures. Ultimately, the results emphasize the critical role of data preprocessing, feature engineering, and model selection in finding an optimal trade-off between classification performance and generalization capability.

References

S. Dubnov and R. Greer. *Deep and Shallow Machine Learning in Music and Audio*. Chapman, Hall, 2023.

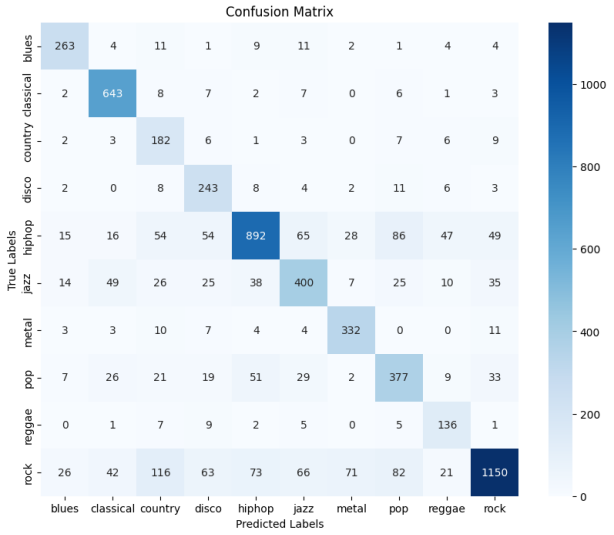

Figure 16: Confusion Matrix for TensorFlow Model with Class Weighting.

C. Weihs, D. Jannach, I. Vatolkin, and G. Rudolph. *Music Data Analysis Foundations and Applications*. Chapman, Hall, 2019.

J. P. Briot, F. D. Pachet, and G. Hadjeres. *Deep Learning Techniques for Music Generation*. SpringerLink, 2019.

S. Schindler. *Music Genre Classification Using Machine Learning*. Springer, 2021.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293–302, 2002.

N. Pelchat and C. M. Gelowitz. Neural network music genre classification. *Canadian Journal of Electrical and Computer Engineering*, 43:170–173, 2020.

S. Oramas, F. Barbieri, O. Nieto, and X. Serra. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval*, 1:4–21, 2018.

S. K. Prabhakar and S. Lee. Holistic approaches to music genre classification using efficient transfer and deep learning techniques. *Expert Systems with Applications*, 211:118636, 2023.

M. Hashemzadeh N. Farajzadeh, N. Sadeghzadeh. Pmg-net: Persian music genre classification using deep neural networks. *Entertainment Computing*, 44:100518, 2023.

O. G. Kilickaya. Genre classification and musical features analysis. *International Journal of Latest Engineering Research and Applications*, 9:18–33, 2024.

N. Narkhede, S. Mathur, A. Bhaskar, and M. Kalla. Music genre classification and recognition using convolutional neural network. *Multimedia Tools and Applications*, pages 1–16, 2024.

K. Mogonediwa. Music genre classification: Training an ai model. *University of Johannesburg*, 2024.

Gtzan dataset - music genre classification. *https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification*.

Fma dataset - music genre classification. *https://github.com/mdeff/fma*.