Explainable and Cross-Cultural Audio Genre Classification

Anonymous

Computer Science

The University of Texas at Austin

Introduction

The modern history of recorded music begins in the late 19th century. Thomas Edison's 1877 phonograph inaugurated the era of mechanical sound reproduction, and Emile Berliner introduced the "Gramophone" in 1887, which used a disc rather than a cylinder - a format that would dominate the next century. While music had long been part of daily life through classical traditions, the emergence of reproducible technologies enabled the phonograph to become a mass medium for popular music in the 1910–20s. Recordings of large-scale orchestral works and other classical instrumental performances also made music more accessible in everyday contexts [20].

Electronic music is broadly defined as "music made or modified using electronic processing or manipulation." Around the turn of the twentieth century, Thaddeus Cahill invented the first musical instrument capable of generating sound electrically. The subsequent development of computer music technology has had an immense impact on nearly every aspect of musical creation and distribution [10]. Although composers were experimenting with computer sound synthesis and algorithm-based composition by the mid-twentieth century, digital technologies did not exert broad influence on the music industry until the introduction of digital synthesizers in the late 1970s and early 1980s [20].

Similarly, the evolution of physical media (such as vinyl records, cassette tapes, and CDs) and the later introduction of electronic and digital media have made approaches to producing, distributing, and interacting with music increasingly sophisticated. In contemporary contexts, streaming services provide convenient access to music and offer features such as music recognition and genre classification. However, classification mechanisms within these services often appear opaque and lack clear explanation. For instance, Spotify, one of the most widely used global streaming platforms, notes that "at the moment there isn't a way to check the genres that Spotify classifies songs into via the Spotify app itself," as discussed in a user forum [9].

Furthermore, genre classification is a rather subjective and variable task, and therefore there is no single answer that fits all cases. As musical cultures increasingly converge, genre labels require attention to diverse cultural contexts, especially given recent trends toward stylistic hybridity. This issue is highlighted in prior research noting that "with the recent advent of various kinds of music, the boundaries between global music genres such as rock, pop, jazz, blues, and hip-hop are blurring, and music classification by genre continues to require new challenges due to the essentially abstract and subjective nature of music. Most of these genres are classified in relation to the structure of musical instrument arrangements,

rhythms, and harmonics, but it is not uncommon, for example, to encounter misclassified ID3 genre tags in MP3 files" [14].

Current approaches to music classification largely consist of two categories: text-based classification using metadata (which relies on manual labeling and does not incorporate the audio signal) and content-based classification using audio data, which involves extracting musical characteristics such as pitch, timbre, and melody that cannot be effectively captured through text alone. These extracted features can then be used as input to a classifier [29]. Furthermore, a survey examining music information use and seeking behaviors discusses the "importance of extramusical information" and the need for "new types of metadata" that include relational categories such as "genre." Support for such "relational metadata" was reinforced by the finding that more than half of the respondents expressed positive opinions toward "genre" as a metadata category (62.7%). Although this study approaches genre as metadata, it nonetheless underscores the need for reliable and carefully constructed information, suggesting that further development of classification systems, particularly ensuring their reliability in real-world contexts, is necessary [16]. The practical need for efficient genre classification continues to grow, given the increasing volume of user-generated music uploaded to streaming platforms.

Therefore, in this paper, I address the explainability and transparency of classification algorithms, the crucial determinants for genre categorization, and the cross-cultural applicability of classification models across different musical contexts.

Research background, and literature review

Digital audio data refers to digital representation of sound waves, captured by converting analog sound signals into series of numerical values. This process involves sampling the continuous sound wave at discrete intervals and storing it in formats that can be digitally processed. Key characteristics of audio data include the sampling rate, bit depth, number of channels, dynamic range, and frequency [30].

To process audio signals for input into classifiers, preprocessing is required, specifically, the extraction of audio features that transform raw sound into analyzable data. Feature extraction involves identifying and isolating meaningful information embedded within a raw audio signal. These techniques convert raw audio into structured data, which helps

machines to process and interpret audio more effectively. These reveal patterns and features crucial for tasks like speech recognition and music analysis [30]. By applying feature extraction techniques, the data becomes more manageable, as ineffective or redundant attributes are removed without discarding any information that is important or relevant to the task. In addition, feature extraction reduces computational load, enabling systems to operate more efficiently and improving the speed of learning and generalization in machine learning processes [2].

For classification purposes, we extract time and frequency domain features from the audio files. These features are Spectrograms, Mel-spectrograms, MFCC, Spectral centroids, Chromagrams, Energy, Spectral Roll-off, Spectral Flux, Spectral Entropy, Zero-crossing rate, and Pitch [13].

- A spectrogram is a time–frequency representation of an audio signal that provides a 2D visualization of how its frequency content changes over time [6]. It combines time, frequency, and amplitude into a single graph into a single graphical display. A spectrograms are generated using the Short-Time Fourier Transform (STFT), which splits the audio into small time segments and computes the frequency spectrum for each [30]. Darker regions in spectrogram indicate lower energy at a given frequency and time while brighter tones indicate the opposite. These patterns support various tasks such as speech recognition, musical analysis and sound pattern identification [6].
- For example, a spectrogram can reveal the distinct frequencies of instruments like trumpets or violins, or help identify vowel sounds, as each vowel has a unique frequency pattern (Zillilz).
    1. A **mel spectrogram** is a variation that maps frequencies to the mel scale, which mimics human auditory perception. Unlike a linear frequency scale, the mel scale emphasizes lower frequencies, where human hearing is more sensitive. This makes mel spectrograms ideal for speech and music analysis. To create mel spectrogram, the STFT is applied, and the resulting spectra are passed through a mel filter bank (Zillilz).
- **Mel-Frequency Cepstral Coefficients (MFCC)**: Mel-Frequency Cepstral Coefficients (MFCCs) build upon mel spectrogram extraction by compressing the spectral information into a compact set of features. MFCCs encode the overall spectral envelope of the audio signal and provide a vectorized representation that approximates how humans perceive sound. Because they effectively capture timbral and textural qualities including tone color and instrument characteristics,

MFCCs have become widely used in audio analysis [5].

The concept of MFCCs was originally introduced by Davis and Mermelstein in the early 1990s and has since emerged as a pivotal feature in various audio-processing applications, most notably in speech recognition. MFCCs are also widely used as timbral descriptor in many genre classification systems and have been employed by many researchers to model music and other audio sounds [3]. The computation of MFCCs involves several key steps as follows:

1. Sampling and windowing: the audio signal is divided into short, overlapping frames. A window function such as Hanning or Hamming window is applied to each frame to enhance harmonics, smooth edges, and diminish edge effect while taking a DFT on the signal [3].
2. STFT (Short-Time Fourier Transform) is then applied to each frame to get its frequency spectrum. It is a key element in music classification, as it gives us insights about tonal differences, which plays a crucial role in genre determination [3].
3. Mel Filter bank - Mel band-pass filter is a bank of filters, which is constructed based on pitch perception applying a Mel filter bank to the obtained power spectra calculating the logarithm values of all filter banks. It targets extracting non-linear representation of the speech signal [2].
4. Discrete Cosine Transform expresses a finite sequence of data points regarding a summation of cosine functions oscillating at different frequencies. The DCT is applied on the Mel filter bank to select most accelerative coefficients or to separate the relationship in the log spectral magnitudes from the filter-bank [2].

Algorithms selection

- **Convolutional Neural Networks (CNNs),** as defined by Aparna Goel, “is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers” [7]. They have recently been applied to a wide range of audio-processing tasks, including pitch estimation, speech analysis and music genre classification, particularly due to their robustness in challenging audio environments [30]. Although CNNs were originally designed for visual data processing tasks such as image classification, object detection, and semantic

segmentation [7], their ability to detect local patterns translates effectively to audio representations such as spectrograms. Spectral patterns can be treated similarly to image textures, enabling CNNs to learn hierarchical structure in time–frequency audio data [30]. Because raw time-domain audio is difficult to process due to high sampling rates, time–frequency transformations such as spectrograms provide a more practical representation for neural network models (Ahktar). Traditional classification methods rely on hand-crafted features, whereas CNN-based systems can automatically learn relevant representations from time–frequency inputs. Several studies demonstrate the effectiveness of CNNs for music genre classification. For example, one study converts raw audio into mel spectrograms using Librosa, feeds them into a CNN, and applies majority voting across ten classifiers, achieving an average accuracy of 84% on the GTZAN dataset [8]. Another study compared an optimized CNN model with a human baseline and found that while humans were only able to correctly classify 43.3% of samples, CNN achieved 98% accuracy on the training set and 68.7% on the test set, completing inference on 10 samples in 36–37 seconds [26].

- **XGB -** XGBoost (XGB), proposed by Dr. Chen in 2016, is a large-scale machine-learning framework for tree boosting and an optimized extension of Gradient Boosting Decision Trees (GBDT). GBDT is an ensemble learning algorithm that achieves high classification accuracy by iteratively combining many weak models, typically decision trees, sequentially, into a stronger predictive model. In this sequential process, each new tree is trained to correct the errors made by the previous one, a procedure known as boosting. XGBoost improves upon this framework by leveraging multi-threaded parallel computation on the CPU to accelerate training, and by incorporating additional regularization terms to reduce model complexity and mitigate overfitting [22, 27]. As an ensemble model composed of multiple decision trees XGBoost is well-suited for music genre classification tasks. When configured with a multi-class softmax objective function, it predicts probability distributions for each genre class, with the number of output classes set to match the dataset. Final predictions for each song can then be obtained using the predict_proba method, which outputs the probability of belonging to each genre category [17]. Unlike RF, boosting algorithms employ forward stagewise additive modeling during sequential training iterations. [3]. In addition, XGBoost also supports parallel computation and missing value processing, which makes the training process of the model more efficient and stable [24].

- **Random Forest -** Random Forest is an ensemble learning method that constructs a large number of decision trees to improve predictive accuracy and reduce overfitting. Each tree is trained on a random subset of the data, and final predictions are obtained by bootstrap aggregating (bagging) the outputs of all trees, a process that enhances model stability and robustness to noise and bias. Random Forest also selects a random subset of features per each split (creation of individual tree), increasing tree diversity and reducing the risk of overfitting, thereby outperforming a single decision tree in generalization ability [3, 25]. Random Forest introduces diversity by training each decision tree on a randomly selected subset of the dataset and a random subset of features, which increases model robustness and reduces the risk of overfitting. This randomness enables Random Forests to excel at handling high-dimensional data, computing feature-importance metrics, and modeling nonlinear relationships, making them widely used across classification and regression tasks [19]. Random forest model has good performance in processing high-dimensional data and nonlinear feature fitting [24].
- In contrast, XGBoost includes explicit regularization mechanisms—such as L1 and L2 penalties—to control model complexity, which Random Forest does not typically incorporate [19]. Also, **Random Forest** can be slow in training, especially with a very large number of trees and on large datasets because it builds each tree independently and the full process can be computationally expensive. However, prediction is fast, as it involves averaging the outputs from all the individual trees.
- Random Forest can be slow to train on very large datasets because each tree is built independently, though inference remains fast since prediction only requires averaging tree outputs. XGBoost, however, is optimized for speed and scalability, supports multi-core and distributed computation and performs more computation with fewer resources. It generally achieves higher accuracy on test data and tends to perform better on tasks requiring high precision or involving class imbalance. Although XGBoost is more challenging to tune, it is often better suited for large and complex datasets [19].

Data Preparation & Feature Extraction

GTZAN Dataset Overview:

1. Genres original
    A. A collection of 10 music genres, each containing 100 audio files

   B. Each file is 30 seconds long
   C. Widely known as the GTZAN dataset, often referred to as the *MNIST of sounds*
2. images original
   A. Mel-spectrogram images generated from each audio file
   B. Used for neural-network–based classification
   C. Audio files were converted to mel spectrograms to allow CNN models to process them as images
3. 2 CSV files
   A. Contain extracted audio features for machine learning
   B. File 1: Mean and variance of multiple audio features per 30-second song
   C. File 2: Same structure, but audio was split into 3-second segments
      i. Produces 10× more data, improving model training
   D. More data increases classifier robustness and performance

Korean Dataset Overview:

1. Audio original
   A. 120,000 raw audio clips (WAV)
   B. Phrase-level recordings sourced from Korean popular music
   C. Includes vocals, Western instruments, synthesizers, and Korean traditional instruments
   D. Designed for loop generation, MIR tasks, and AI music modeling
2. Annotations original
   A. 120,000 MIDI transcription files
   B. Score-style symbolic representations aligned with each audio clip
   C. Supports note-level analysis, transcription research, and symbolic–audio alignment
3. Metadata original
   A. 120,000 JSON metadata files
   B. Contains labeled attributes such as:
      i. instrument type
      ii. phrase characteristics
      iii. structural information
   C. Used for supervised learning and dataset organization
4. Instrument distribution

A. 15,000 vocal melody clips
B. 70,000 Western-instrument clips
    i. drums, piano, guitar, bass, strings, brass, wind, organ
C. 15,000 synthesizer clips
    i. lead, pad
D. 10,000 Korean traditional instrument clips
E. Korean string + Korean wind instruments

Total: 120,000 samples

Preprocessing:

The preprocessing stage required standardizing both datasets (GTZAN and the Korean Popular Music Dataset) so that they are directly comparable for cross-cultural classification and analysis. The steps below summarize the transformations applied before model training.

A. **Genre Re-mapping:** Since the two datasets used different genre labels, the labels were re-mapped into meta-genre categories that serve as supersets of the original genre classes.

Table1. Genre re-mapping table

Meta-Genre	**GTZAN Labels**	**Korean Labels**
Blues-like	blues	Folk & Blues
Rock/Metal	rock, metal	Rock & Metal
HipHop/Rap	hiphop	Rap & Hiphop
Pop/Dance/RnB	disco, pop	Dance, RnB & Soul

B. **Instrument Filtering (Korean Dataset Only):** The Korean dataset originally contained many irrelevant categories (for example, Korean traditional instruments, whereas the focus of this project is mainstream international and Korean pop music). To match GTZAN's modern music focus, only core instruments commonly used in contemporary popular music were retained:
    - Drums / Percussion
    - Bass
    - Electric Guitar
    - Synth / Keys
    - Strings
C. **Audio normalization – same technical profile for all files for feature extraction [21]**

- Audio setting:

  Table 2. Table showing parameter and value pairs for standard audio seting

Parameter	**Value**
Sample rate	22,050 Hz
n_fft	2048
hop_length	512
Target audio length	5 seconds
Mel bins	128
MFCC count	20
Final feature shape (for CNN)	128 × 128
Power	log-power mel

- Length normalization
  - Korean dataset: padded → *exactly 5 seconds* (raw clips ≈3 sec)
  - GTZAN: trimmed → *exactly 5 seconds* (raw clips 30 sec)
- Output formats
  - Classical ML models: CSV feature vectors
  - CNN models: .npy mel-spectrogram tensors
  - Same label schema for both datasets
- Feature Extraction (Per 5s Clip):
  - MFCCs (20): mean + std → 40 features
  - Chroma (12): mean + std → 24 features
  - Spectral Centroid: mean + std
  - Spectral Bandwidth: mean + std
  - Spectral Rolloff: mean + std
  - Zero-Crossing Rate: mean + std
  - RMS energy: mean + std
  - Tempo (BPM): 1 scalar

D. Dataset Splits:
- For all datasets:
  - 70% training
  - 15% validation
  - 15% testing
- *Due to imbalance in the final data counts, additional stratification was required. The data sizes before stratification:*

Table 3. Table showing imbalance in the data counts

Meta-Genre	GTZAN Count	Korean Count
Pop / Dance / RnB	200	1932
Rock / Metal	200	1020
HipHop / Rap	100	325
Blues-like	100	183

Models & Experiment Setup

Model architecture design and parameter selection were guided by recent academic literature in Music Information Retrieval (MIR), with the aim of constructing compact, interpretable, and comparable baseline models for subsequent analysis. This design choice prioritizes transparency and explainability, which is essential for examining potential cross-cultural differences in musical and genre structures. The resulting architectures reflect current best practices in music-genre classification research and are aligned with experimental setups commonly used in the field [12, 18, 28].

Model architecture:

1. CNN model construction: Three convolution–batchnorm–ReLU–pooling blocks (32, 64, and 128 channels), followed by global average pooling and two fully connected layers (256 → n_classes) with dropout regularization.
    A. Input: 128×128 mel-spectrogram
    B. Output: Genre classification
2. Random Forest: 500 bootstrapped decision trees with random feature subsets → majority vote.
3. XGBoost: 500 boosted trees trained sequentially with regularization and subsampling → softmax classifier.

Experiments:

The experiments consisted of three rounds, where all models were trained, tested, and evaluated on GTZAN-only, Korean-only, and combined datasets. This setup enables the examination of feature robustness, the observation of model performance trends across datasets, and the assessment of potential cross-cultural differences arising from variations in musical structure and dataset composition.

Results

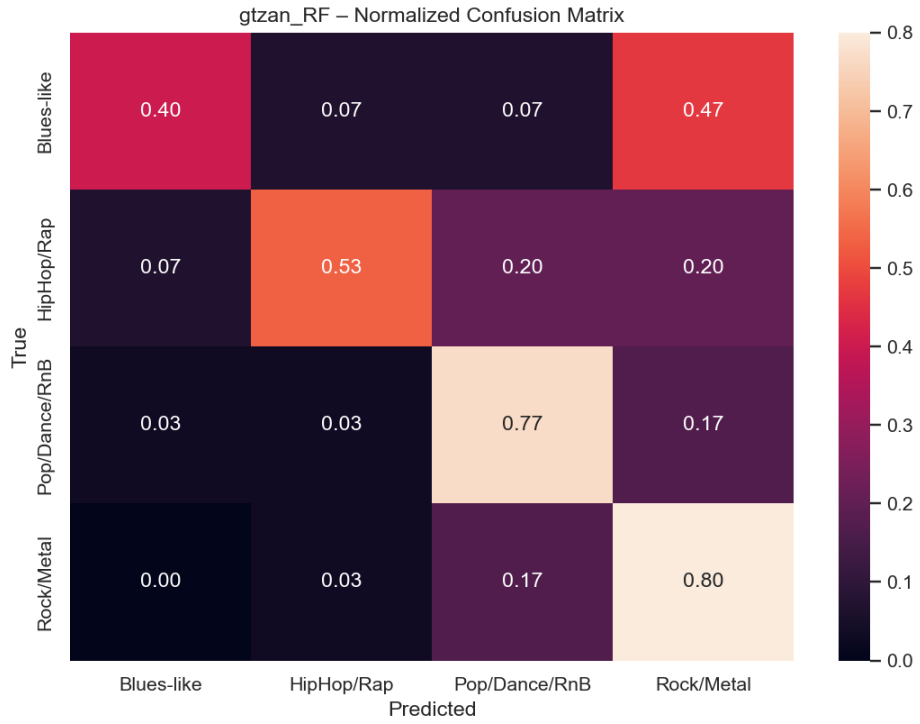
Explainability methods

They provide interpretable insights into the decision-making processes of the models and help validate whether the models were able to capture meaningful musical features.

- SHAP (SHapley Additive exPlanations) by Lundberg and Lee (2017) is a method to explain individual predictions. It is based on the game-theoretically optimal Shapley values, providing a game-theoretic framework for explaining the output of any machine learning model. SHAP connects optimal credit allocation with local explanations using classic Shapley values and their extensions [1].
- Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique for producing “visual explanations” for decisions made by CNN-based models. It uses the gradients of a target concept flowing into the final convolutional layer to create a coarse localization map that highlights important regions in the image for the prediction [23].

Ex.1 — Random Forest (GTZAN → GTZAN)

Fig 1. Random Forest (GTZAN→GTZAN) – Confusion Matrix



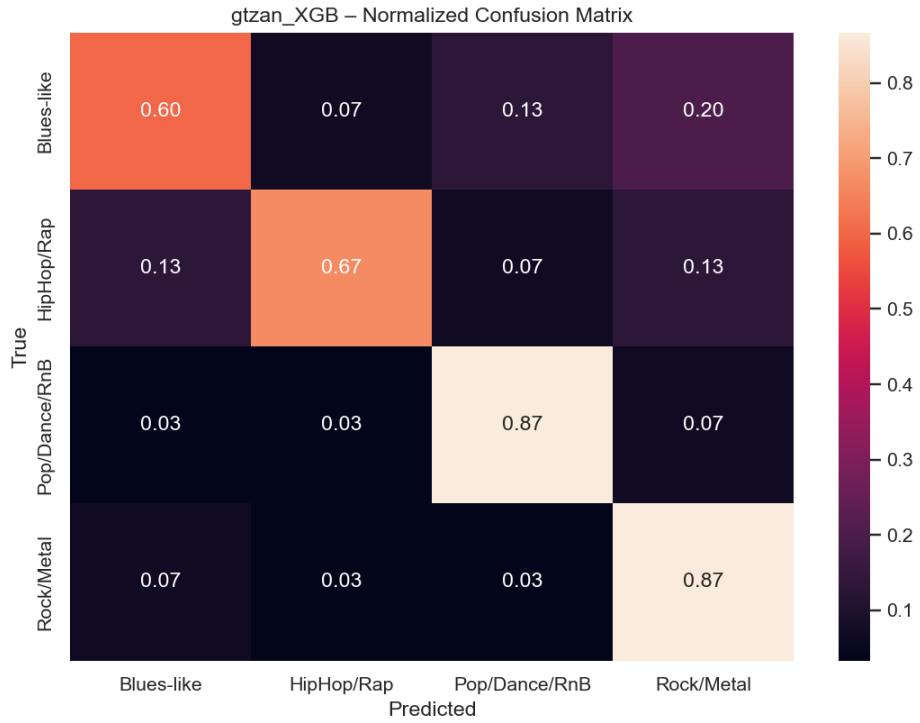
Listing 1. Random Forest (GTZAN→GTZAN) – Classification Report


```
{
    "Blues-like": {
        "precision": 0.75,
        "recall": 0.4,
        "f1-score": 0.5217391304347826,
        "support": 15.0
    },
    "HipHop/Rap": {
        "precision": 0.7272727272727273,
        "recall": 0.5333333333333333,
        "f1-score": 0.6153846153846154,
        "support": 15.0
    },
    "Pop/Dance/RnB": {
        "precision": 0.71875,
        "recall": 0.7666666666666667,
        "f1-score": 0.7419354838709677,
        "support": 30.0
    },
    "Rock/Metal": {
        "precision": 0.6153846153846154,
        "recall": 0.8,
        "f1-score": 0.6956521739130435,
        "support": 30.0
    },
    "accuracy": 0.6777777777777778,
    "macro avg": {
        "precision": 0.7028518356643357,
```

```
        "recall": 0.625,
        "f1-score": 0.6436778509008523,
        "support": 90.0
    },
    "weighted avg": {
        "precision": 0.6909236596736597,
        "recall": 0.6777777777777778,
        "f1-score": 0.6687165102312367,
        "support": 90.0
    }
}
```


Ex.2 — XGBoost (GTZAN → GTZAN)

Fig 2. XGBoost (GTZAN→GTZAN) – Confusion Matrix



Listing 2. XGBoost (GTZAN→GTZAN) – Classification Report


```
{
    "Blues-like": {
```

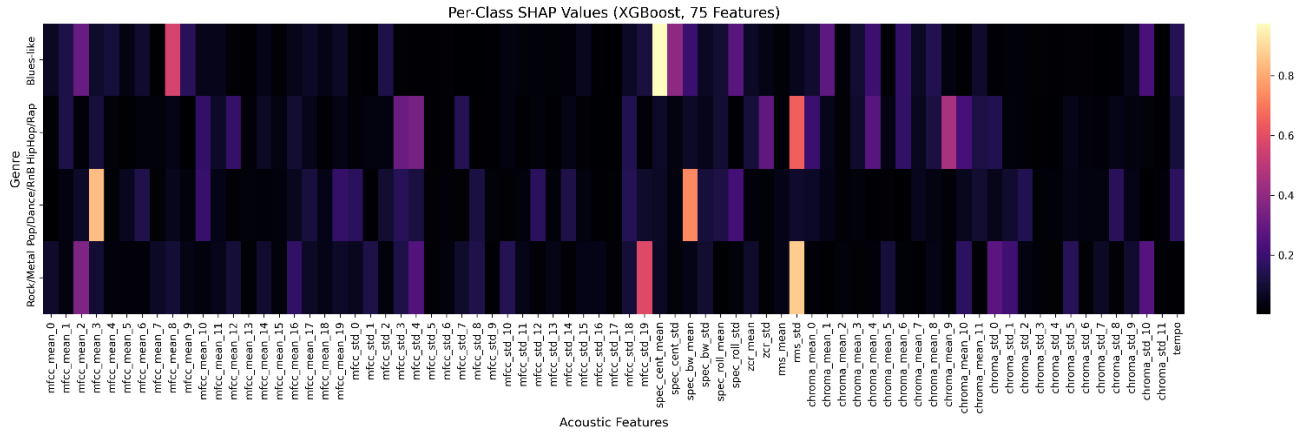
```
    "precision": 0.6428571428571429,
    "recall": 0.6,
    "f1-score": 0.6206896551724138,
    "support": 15.0
},
"HipHop/Rap": {
    "precision": 0.7692307692307693,
    "recall": 0.6666666666666666,
    "f1-score": 0.7142857142857143,
    "support": 15.0
},
"Pop/Dance/RnB": {
    "precision": 0.8666666666666667,
    "recall": 0.8666666666666667,
    "f1-score": 0.8666666666666667,
    "support": 30.0
},
"Rock/Metal": {
    "precision": 0.7878787878787878,
    "recall": 0.8666666666666667,
    "f1-score": 0.8253968253968254,
    "support": 30.0
},
"accuracy": 0.7888888888888889,
"macro avg": {
    "precision": 0.7666583416583417,
    "recall": 0.75,
    "f1-score": 0.7567597153804051,
    "support": 90.0
```

```
    },
    "weighted avg": {
        "precision": 0.7868631368631369,
        "recall": 0.7888888888888889,
        "f1-score": 0.7865170589308519,
        "support": 90.0
    }
}
```


Listing 3. Ranked list of top 10 features

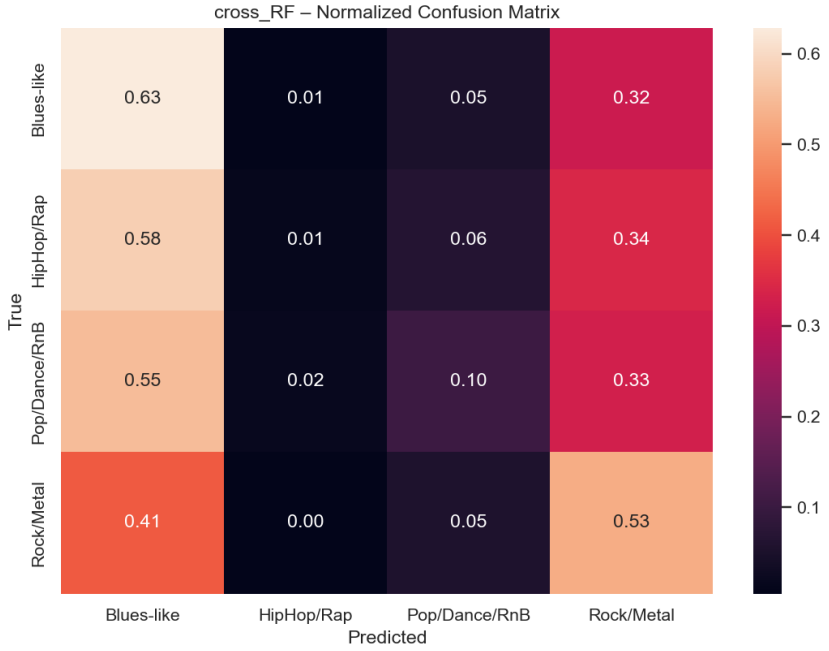
	feature	mean_abs_shap
1	rms_std	0.400632
2	mfcc_mean_3	0.277269
3	spec_bw_mean	0.249394
4	spec_cent_mean	0.245131
5	mfcc_mean_8	0.214751
6	mfcc_std_4	0.180814
7	mfcc_std_19	0.176501
8	mfcc_std_3	0.172206
9	mfcc_mean_2	0.160437
10	spec_roll_std	0.144613

Fig3. Per-class SHAP values (75 features)



Ex.3 — Random Forest (GTZAN → Korean)

Fig 4. Random Forest (GTZAN→Korean) – Confusion Matrix



Listing 4. Random Forest (GTZAN→Korean) – Classification Report

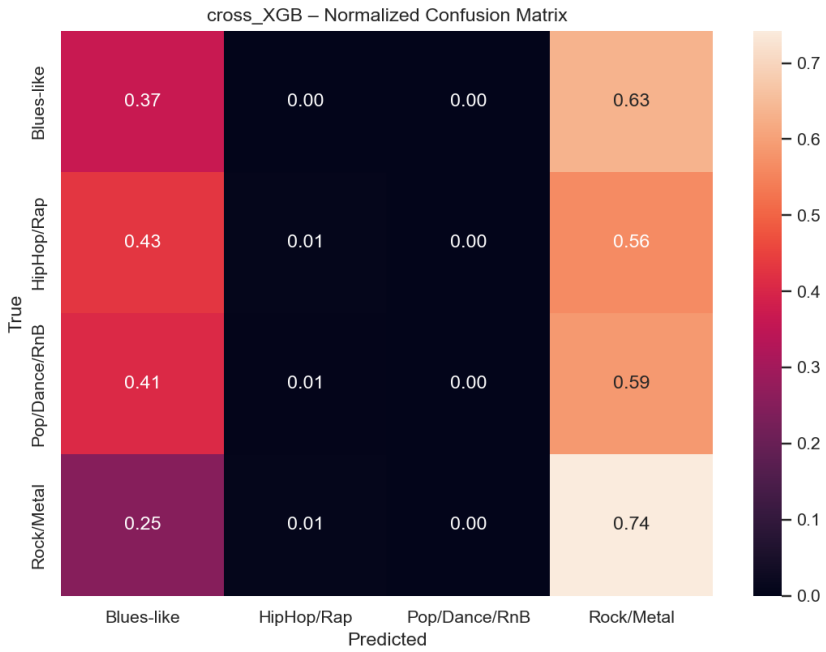

```
{
    "Blues-like": {
        "precision": 0.06413831567205801,
        "recall": 0.6284153005464481,
        "f1-score": 0.11639676113360324,
```

```
    "support": 183.0
},
"HipHop/Rap": {
    "precision": 0.09302325581395349,
    "recall": 0.012307692307692308,
    "f1-score": 0.021739130434782608,
    "support": 325.0
},
"Pop/Dance/RnB": {
    "precision": 0.7017543859649122,
    "recall": 0.10351966873706005,
    "f1-score": 0.18042399639152007,
    "support": 1932.0
},
"Rock/Metal": {
    "precision": 0.40179238237490666,
    "recall": 0.5274509803921569,
    "f1-score": 0.45612547689699023,
    "support": 1020.0
},
"accuracy": 0.2476878612716763,
"macro avg": {
    "precision": 0.3151770849564576,
    "recall": 0.3179234104958394,
    "f1-score": 0.19367134121422402,
    "support": 3460.0
},
"weighted avg": {
    "precision": 0.522424154223739,
```

```
        "recall": 0.2476878612716763,
        "f1-score": 0.24340837345147417,
        "support": 3460.0
    }
}
```


Ex.4 — XGBoost (GTZAN → Korean)

Fig 5. XGBoost (GTZAN→Korean) – Confusion Matrix



Listing 5. XGBoost (GTZAN→Korean) – Classification Report

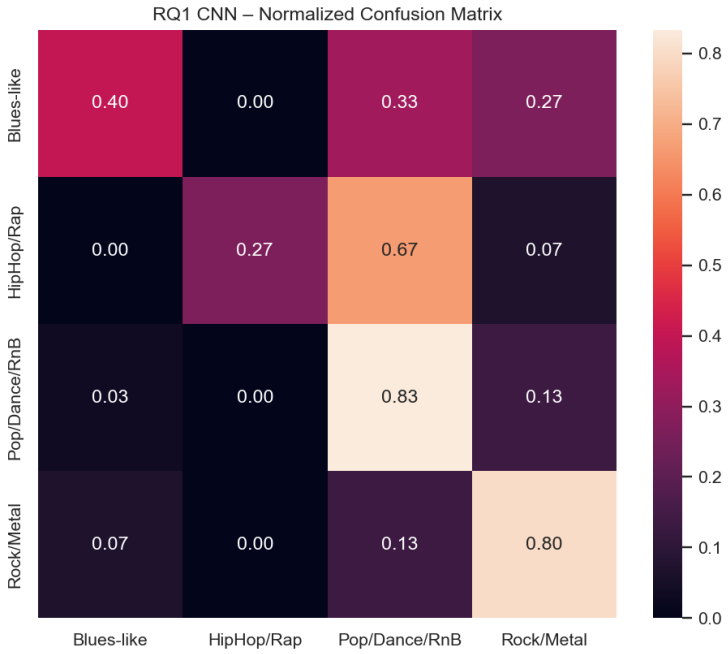

```
{
    "Blues-like": {
        "precision": 0.0536,
        "recall": 0.366120218579235,
        "f1-score": 0.09351011863224006,
        "support": 183.0
    },
    "HipHop/Rap": {
```

```
    "precision": 0.09523809523809523,
    "recall": 0.006153846153846154,
    "f1-score": 0.011560693641618497,
    "support": 325.0
},
"Pop/Dance/RnB": {
    "precision": 0.0,
    "recall": 0.0,
    "f1-score": 0.0,
    "support": 1932.0
},
"Rock/Metal": {
    "precision": 0.34582000913659205,
    "recall": 0.7421568627450981,
    "f1-score": 0.4717980679339358,
    "support": 1020.0
},
"accuracy": 0.23872832369942196,
"macro avg": {
    "precision": 0.12366452609367182,
    "recall": 0.2786077318695448,
    "f1-score": 0.1442172200519486,
    "support": 3460.0
},
"weighted avg": {
    "precision": 0.11372762724615747,
    "recall": 0.23872832369942196,
    "f1-score": 0.14511664925891343,
    "support": 3460.0
```

```
    }
}
```


Ex.5 — CNN (GTZAN → GTZAN)

Fig 6. CNN (GTZAN→GTZAN) – Confusion Matrix



Listing 6. CNN (GTZAN→GTZAN) – Classification Report


```
{
    "Blues-like": {
        "precision": 0.6666666666666666,
        "recall": 0.4,
        "f1-score": 0.5,
        "support": 15.0
    },
    "HipHop/Rap": {
        "precision": 1.0,
        "recall": 0.26666666666666666,
```

```
        "f1-score": 0.42105263157894735,
        "support": 15.0
    },
    "Pop/Dance/RnB": {
        "precision": 0.5681818181818182,
        "recall": 0.8333333333333334,
        "f1-score": 0.6756756756756757,
        "support": 30.0
    },
    "Rock/Metal": {
        "precision": 0.7272727272727273,
        "recall": 0.8,
        "f1-score": 0.7619047619047619,
        "support": 30.0
    },
    "accuracy": 0.6555555555555556,
    "macro avg": {
        "precision": 0.740530303030303,
        "recall": 0.575,
        "f1-score": 0.5896582672898463,
        "support": 90.0
    },
    "weighted avg": {
        "precision": 0.7095959595959597,
        "recall": 0.6555555555555556,
        "f1-score": 0.6327022511233037,
        "support": 90.0
    }
}
```

Fig 7. Grad-CAM visualization (Blues-like) 1

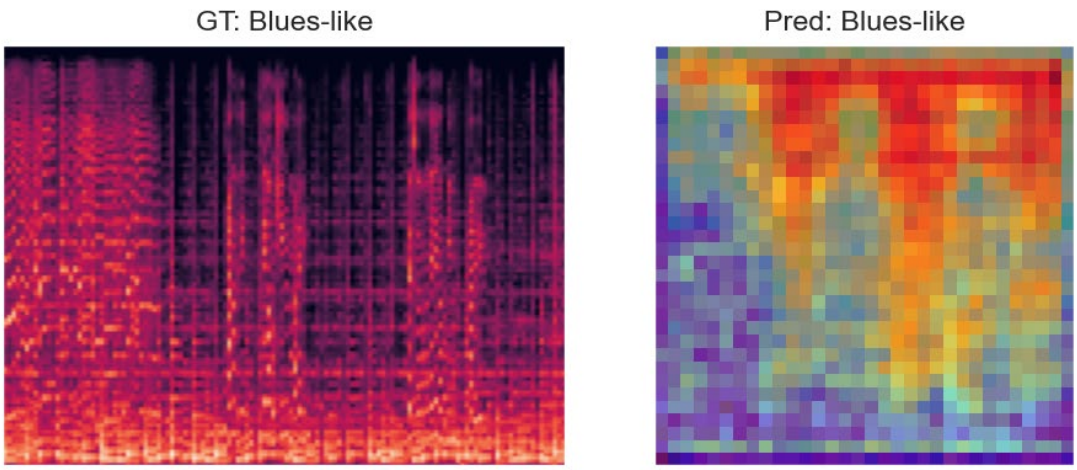
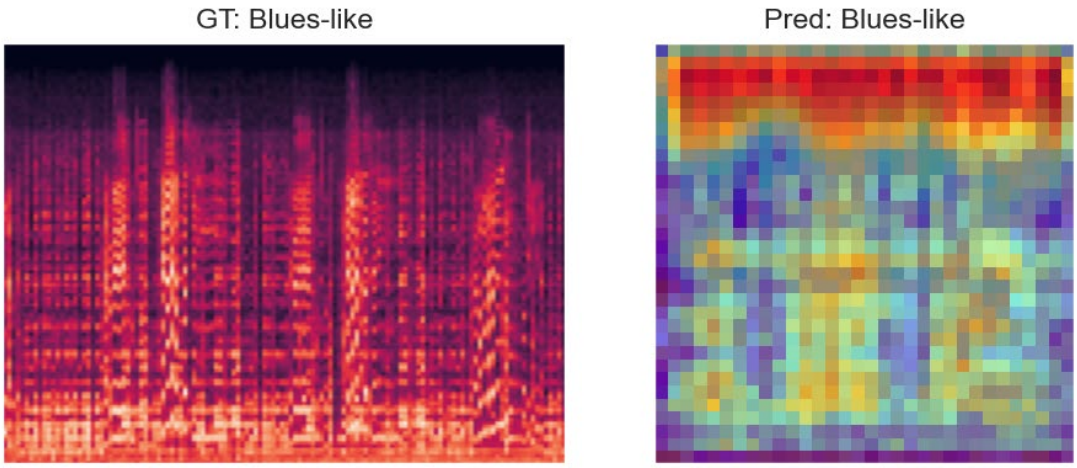
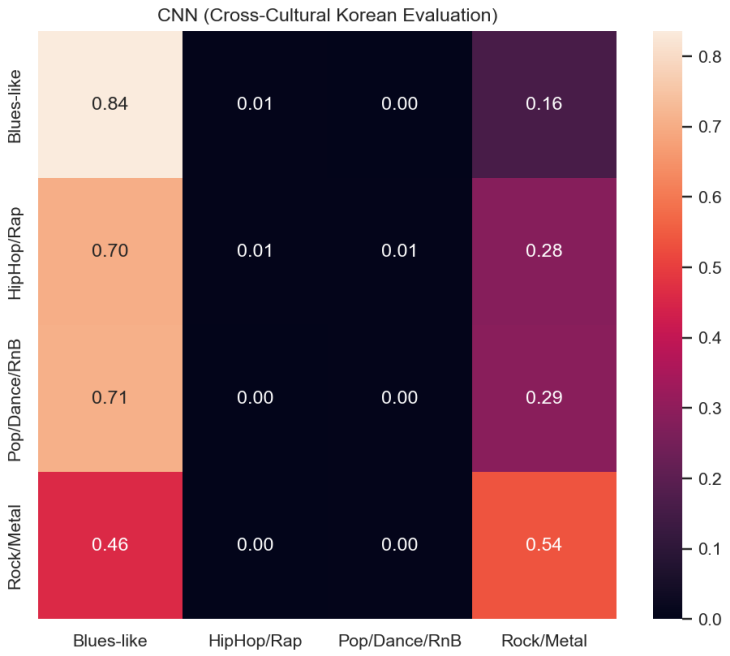


Fig 8. Grad-CAM visualization (Blues-like) 2



Ex.6 — CNN (GTZAN → Korean)

Fig 9. CNN (GTZAN→Korean) – Confusion Matrix



Listing 7. CNN (GTZAN→Korean) – Classification Report


```
{
    "Blues-like": {
        "precision": 0.0690744920993228,
        "recall": 0.8360655737704918,
        "f1-score": 0.12760633861551293,
        "support": 183.0
    },
    "HipHop/Rap": {
        "precision": 0.4,
        "recall": 0.006153846153846154,
        "f1-score": 0.012121212121212121,
        "support": 325.0
    },
    "Pop/Dance/RnB": {
        "precision": 0.5384615384615384,
        "recall": 0.0036231884057971015,
        "f1-score": 0.0071979434447300775,
```

```
        "support": 1932.0
    },
    "Rock/Metal": {
        "precision": 0.44661776691116545,
        "recall": 0.5372549019607843,
        "f1-score": 0.48776145972407653,
        "support": 1020.0
    },
    "accuracy": 0.20520231213872833,
    "macro avg": {
        "precision": 0.3635384493680067,
        "recall": 0.3457743775727299,
        "f1-score": 0.1586717384763829,
        "support": 3460.0
    },
    "weighted avg": {
        "precision": 0.4735544643385136,
        "recall": 0.20520231213872833,
        "f1-score": 0.15569782357219925,
        "support": 3460.0
    }
}
```


Conclusion

The 6 experiments, along with Grad-CAM and SHAP analyses, revealed several consistent patterns.

GTZAN RF: Blues-like shows the best precision but the lowest recall, while

Pop/Dance/RnB achieves the most consistent performance overall (relatively high precision and recall). Rock/Metal exhibits the lowest precision but the highest recall, meaning the classifier frequently confused its characteristics with other genres. Meanwhile, the classifier tended to be conservative when labeling Blues-like tracks.

GTZAN XGBoost: Compared to the Random Forest model, the XGBoost classifier shows higher precision, recall, and F1 across all genres, indicating that MFCC-based patterns were captured more effectively, aligning with the literature supporting XGBoost's stronger performance. Similar to the Random Forest case, Pop/Dance/RnB again achieves the most stable and strongest scores. In contrast, Blues-like - unlike its RF results - shows both low precision and recall, while the model demonstrates better performance on Rock/Metal, suggesting more confident and stable classification for this genre.

Cross-cultural evaluation (GTZAN → Korean): When models trained on GTZAN are evaluated on the Korean dataset, both Random Forest and XGBoost suffer substantial accuracy drops. This reflects strong differences in musical structure across cultures and a pronounced domain shift between the two musical contexts.
Rock/Metal again exhibits the best accuracy, suggesting its acoustic characteristics transfer more reliably across datasets. In contrast, Pop/Dance/RnB shows the sharpest degradation: despite strong performance in the GTZAN domain, and high precision under Random Forest, it yields extremely low recall, and even reaches 0 precision/recall with XGBoost. This implies that cross-cultural divergence is strongest for the most mainstream/popular genre.

CNN: The CNN model shows an interesting pattern. It generally exhibits higher recall than precision, indicating that it is the most conservative classifier among the three, but

relatively confident in its positive predictions. Rock/Metal once again shows high precision and recall in both GTZAN and cross-cultural evaluations, suggesting it is the most distinguishable and consistent genre across musical contexts. Pop/Dance/RnB displays the lowest recall when evaluated on the Korean dataset, replicating the instability observed in classical models.
Overall performance ranking was: **XGB > CNN > RF** in-domain, and **RF > XGB > CNN** in cross-cultural settings.

Only the Blues-like genre produced a stable Grad-CAM visualization. This suggests that only this genre yielded sufficiently consistent activation patterns for the CNN, while for other genres the activation maps were likely too noisy, aligning with existing MIR findings that CNNs often compress temporo-spectral structure too aggressively, impairing interpretability.

SHAP analysis shows the most influential musical features for the GTZAN dataset. RMS_std appears as the strongest contributor, reflecting dynamic variability (changes in loudness over time). Its prominence, along with the strong cross-domain performance of Rock/Metal, suggests a meaningful link between genre characteristics and how models rely on dynamics for discrimination.

Overall, these patterns demonstrate that models trained on Western-dominant datasets do not reliably transfer to Korean music, and the severity of this mismatch varies significantly across genres.

Limitations include the structural differences between datasets (Korean data using loop-based segments by instrument vs. GTZAN's 30-second audio tracks). For future work, I could explore zero-shot cross-cultural evaluation more systematically and curate more

rigorously aligned datasets that better represent distinctive cultural musical features.

Bibliography

1. “18 Shap.” *18 SHAP – Interpretable Machine Learning*, christophm.github.io/interpretable-ml-book/shap.html. Accessed 3 Dec. 2025.
2. Abdul, Zrar Kh., and Abdulbasit K. Al-Talabani. “Mel frequency cepstral coefficient and its applications: A Review.” *IEEE Access*, vol. 10, 2022, pp. 122136–122158, https://doi.org/10.1109/access.2022.3223444.
3. Akhtar, Ryyan, and Alokit Mishra. *Music Genre Classification Using Machine Learning Techniques*, 1 Sept. 2025, https://arxiv.org/html/2509.01762v1#S2.
4. Andrada, and Victor Basu. “GTZAN Dataset - Music Genre Classification.” 2019, Accessed 2025.
5. Aristorenas, Aris J. *Machine Learning Framework for Audio-Based Content Evaluation Using MFCC, Chroma, Spectral Contrast, and Temporal Feature Engineering*, 31 Oct. 2024, https://arxiv.org/html/2411.00195v1.
6. Bhakta, Susmit Sekhar. “Understand Audio Data.” *GeeksforGeeks*, GeeksforGeeks, 23 July 2025, www.geeksforgeeks.org/nlp/understand-audio-data/.
7. Buhl, Nikolaj. “Convolutional Neural Networks (CNN) Overview.” *Convolutional Neural Networks Cheat Sheet | Encord* , July 2023, encord.com/blog/convolutional-neural-networks-explained/.
8. Cheng, Yu-Huei, et al. “Convolutional neural networks approach for music genre classification.” *2020 International Symposium on Computer, Consumer and Control (IS3C)*, Nov. 2020, pp. 399–403, https://doi.org/10.1109/is3c50286.2020.00109.
9. Damon, Adam. “How Do I Find What Genre a Song Is.” *Home - The Spotify Community*, 3 Dec. 2023, community.spotify.com/t5/Content-Questions/how-do-i-find-what-genre-a-song-is/td-p/5719043.
10. Dewan, Brian. “Thaddeus Cahill’s ‘Music Plant’: Brian Dewan.” *CABINET /*, 3 winter 2002, www.cabinetmagazine.org/issues/9/dewan.php.
11. Ding, Yang, et al. “Efficient Music Genre Recognition Using ECAS-CNN: A Novel Channel-Aware Neural Network Architecture.” *MDPI*, Multidisciplinary Digital Publishing Institute, 31 Oct. 2024, www.mdpi.com/1424-8220/24/21/7021#:~:text=A%20convolutional%20neural%20network%20(CNN,genre%201%20to%20genre%20N.
12. Dutt, Namrata. “Music Genre Classification Using CNN: Part 1- Feature Extraction | by Namrata Dutt | Medium.” *Medium*, 2022, medium.com/@namratadutt2/music-genre-classification-using-cnn-part-1-feature-extraction-b417547b8981.

13. Kim, Jonghwa. International Journal of Internet, Broadcasting and Communication, Cheju, 2024, pp. 27–32, *Multiclass Music Classification Approach Based on Genre and Emotion*.
14. Korean Culture Center. "'Korean Popular Music Loop Sound Generation Dataset.'" 2023, Accessed 2025.
15. Kumar, Ravin. "Audio Features." *Project Name Not Set*, 2024, ravinkumar.com/GenAiGuidebook/audio/audio_feature_extraction.html.
16. Lee, Jin Ha. *Survey of Music Information Needs, Uses, and Seeking Behaviours: Preliminary Findings*, 2004.
17. Leonhardt, Juan Francisco. "Music Genre Classification: A Machine Learning Exercise | by Juan Francisco Leonhardt | Medium." *Music Genre Classification: A Machine Learning Exercise*, 2024, medium.com/@juanfraleonhardt/music-genre-classification-a-machine-learning-exercise-9c83108fd2bb.
18. Li, Teng. "Optimizing the configuration of deep learning models for music genre classification." *Heliyon*, vol. 10, no. 2, Jan. 2024, https://doi.org/10.1016/j.heliyon.2024.e24892.
19. M, Content. "Top 6 Machine Learning Classification Algorithms." *GeeksforGeeks*, GeeksforGeeks, 6 Aug. 2025, www.geeksforgeeks.org/machine-learning/top-6-machine-learning-algorithms-for-classification/.
20. Marcus, Leonard. "Music | Art Form, Styles, Rhythm, & History | Britannica." *Encyclopedia Britannica*, 29 July 2024, www.britannica.com/art/music.
21. "MFCC." *Extract Mel-Frequency Cepstral Coefficients from Audio - Simulink*, 2022, kr.mathworks.com/help/audio/ref/mfccblock.html.
22. Saxena, Pawan. "XGBoost." *GeeksforGeeks*, GeeksforGeeks, 24 Oct. 2025, www.geeksforgeeks.org/machine-learning/xgboost/).
23. Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 618–626, https://doi.org/10.1109/iccv.2017.74.
24. Song, Jiulin. "Comparison and analysis of accuracy of traditional random forest machine learning model and XGBoost model on Music Emotion Classification Dataset." *2023 4th International Conference on Machine Learning and Computer Application*, 27 Oct. 2023, pp. 712–716, https://doi.org/10.1145/3650215.3650340.
25. Stihec, Jan. "Random Forests in ML for Advanced Decision-Making." *Shelf*, 19 June 2024, shelf.io/blog/random-forests-in-machine-learning/.
26. Suo, Wency. "CNN Music Genre Classification." *Regeneron Isef*, 2025,

isef.net/project/robo046---cnn-music-genre-classification.

27. Tan, Lifeng, et al. “Music style classification with compared methods in XGB and BPNN.” *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, June 2019, pp. 403–407, https://doi.org/10.1109/icis46139.2019.8940287.
28. Wang, Yaohua, and Haoqi Chen. “An improved random forest model for music genre classification algorithm based on Sparrow Search algorithm.” *Applied and Computational Engineering*, vol. 77, no. 1, 16 July 2024, pp. 84–90, https://doi.org/10.54254/2755-2721/77/20240658.
29. Zhang, Jingwen. “Music feature extraction and classification algorithm based on Deep Learning.” *Scientific Programming*, vol. 2021, 25 May 2021, pp. 1–9, https://doi.org/10.1155/2021/1651560.
30. Zilliz. “Getting Started with Audio Data: Processing Techniques and Key Challenges | by Zilliz | Medium.” *Medium*, 28 Feb. 2025, medium.com/@zilliz_learn/getting-started-with-audio-data-processing-techniques-and-key-challenges-420dc5233163.

Appendix: Use of AI

I used ChatGPT to support idea brainstorming (e.g. topic and concept exploration), receive guidance on tool and data selection (all final choices were made personally), refine phrasing, and resolve minor coding bugs. All text was drafted by me, based on the literature and my own understanding. AI assistance was limited to minimal linguistic refinement for clarity, conciseness, and academic tone. I confirm that no AI system was used to generate original content from scratch and that all core writing, interpretation, and analysis were my own.