# Music Genre Classification and Feature Comparison using ML

Zhengxin Qi
Fordham University
NY, NY, USA
zqi7@fordham.edu

Mohamed Rahouti
Fordham University
Bronx, NY, USA
mrahouti@fordham.edu

Mohammed A. Jasim
University of Mount Union
Alliance, OH, USA
jaesimad@mountunion.edu

Nazli Siasi
Newport University
Newport News, VA, USA
nazli.siasi@cnu.edu

## ABSTRACT

An essential feature of the music is the genre, which can be considered a high-level description of an individual piece of music. In this sense, genre as a music feature is similar to typical descriptive features from the ML perspective. Although a genre can be understood as a principal component of a piece of music, the process of breaking it down to meaningful representation is a grand challenge. Identifying the genre with lower-level features is a key part of music genre recognition (MGR), which is an important field of research in music information retrieval (MIR). Understanding how to describe music genres in a quantitative way can be useful in analyzing the music for use in music recommender systems and the general understanding of music. This research aims to compare and analyze the feasibility, performance, and understandability of features used to describe music by predicting the genre using machine learning (ML) techniques. Using the mel-frequency cepstral coefficients (MFCC), a popular audio feature extraction method, key features from GTZAN, and human-understandable features from Spotify, this paper demonstrates a trade-off between classification accuracy, understandability, and interpretability of the features.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Machine learning algorithms**; *Artificial intelligence*; • **Information systems** → Information retrieval.

## KEYWORDS

music, genre classification, music information retrieval, machine learning

**ACM Reference Format:**
Zhengxin Qi, Mohamed Rahouti, Mohammed A. Jasim, and Nazli Siasi. 2022. Music Genre Classification and Feature Comparison using ML. In *ICMLT '22: International Conference on Machine Learning Technologies, March 11–13, 2022, Rome, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3529399.3529407

*ICMLT 2022, March 11–13, 2022, Rome, Italy*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9574-8/22/03…$15.00
https://doi.org/10.1145/3529399.3529407

## 1 INTRODUCTION

Music genre can be defined as a conventional category, which identifies specific pieces of music as belonging to a shared set of traditions or conventions [7]. Music from the same genre can use the same instrumentations (e.g., drum machine, guitar, piano, synthesizer, etc.), convey similar mood (e.g., happy, sad, calm, depressed, etc.), have similar themes (e.g., love, death, nostalgia, etc.), and comprise similar tempo (e.g., slow, fast, etc.). Researchers are generally presented with samples of music pieces and qualitative descriptions when exploring and investigating music genres, but they are rarely presented with quantitative descriptions. However, some understanding from the quantitative perspective can be of immense help to researchers and musicians.

Notably, studies such as Bhat et al. [2] attempted to classify music mood using four features (Intensity, Timbre, Pitch, Rhythm) as eight categories (Happy, Exuberant, Energetic, Frantic, Sad, Depression, Calm, Contentment). The moods of songs are divided according to psychologist Robert Thayer's traditional model of mood, and the features are generated using a specialized algorithm and then normalized. Bhat's results show that happy, energetic, depressed, and calm moods are the most successfully identified, with accuracies above 0.9.

There are numerous other features that can be extracted from audio. This paper aims to examine two types of features that are relatively on the extremes by using the famous GTZAN dataset and a Spotify dataset. The GTZAN dataset was collected and used by Tzanetakis and Cook [19] and contains low-level features derived mathematically from frequency, intensity, etc. The Spotify dataset was initially generated from the Spotify Developer API (then called Echo Nest API) and contained human-understandable, high-level features. An in-depth introduction to the two datasets will be given in the next section.

The contributions of this study are summarized as follows:

- Explore the feasibility of Spotify features for the use of genre classification using machine learning (ML).
- Compare the performance of the classifiers that use Spotify features directly to those that use the GTZAN dataset with the same methodology.
- Examine and describe music genres in a quantitative way using various ML models.
- Gain insights into some music genres (for music recommendation systems) while further exploring the Spotify attributes and features dataset.

The remainder of this paper is organized as follows. In Section 2, we discuss the state-of-art studies related to our work. Next, Section 3 covers our methodology and implementations. Furthermore, this section discusses and explores the datasets used in this study and the models' parameters tuning. Section 4 presents the performance results and analysis, while Section 5 highlights future works to be built upon this study. Last, Section 6 concludes the paper.

## 2 RELATED WORK

Music genre classification has been considered a substantial component in various music information retrieval (MIR) systems [7]. It has further been growing as a research field while attracting attention to the emergence of digital music on the Internet in general [3, 8].

Over the past couple of decades, ML techniques have been widely leveraged in countless fields and applications such as music, medicine, biology, cybersecurity, etc., to enable solutions to mine the information hidden in the data [9, 10, 12, 13]. Research studies leveraging ML techniques for music modeling, generation, and analysis primarily aim at training techniques and datasets to gauge system performance based upon quantitative and qualitative measures [1, 11, 20]. Such studies rarely build on sophisticated outcomes of music datasets to inform the application of music genre classification [17]. Notably, music genre classification has been significantly studied in the past several years using various ML algorithms [14, 21], and the GTZAN dataset is at the center of it. In the paper where the dataset was originated, Tzanetakis et al. [19] achieved 61% accuracy when classifying ten genres using the mel-frequency cepstral coefficients (MFCC) features.

Other notable works such as Elbir and Aydin [4] proposed a music genre classification system and music recommendation engine, which focuses on extracting representative features that have been obtained by a novel deep neural network model. The standalone classifier based on convolutional neural network (CNN) achieved an accuracy of 81.8% on the GTZAN dataset. Kaur and Kumar [6] studied and analyzed the feature-based automatic music genre classification using Gaussian mixture model. Moreover, Senac et al. [14] proposed feature maps with CNN for music genre classification. In contrast, Sharma et al. [15] proposed a novel music genre classification model based on stacking of support vector machine (SVM) with relevance vector machine (RVM) and decision trees. Training on the GTZAN dataset, the model in [15] reached an accuracy of 87%.

While numerous novel models have been proposed, and countless experiments have been done on the GTZAN dataset, to the best of our knowledge, none of these studies have examined the feasibility of using the Spotify features as means for genre classification. If models built on the Spotify features can achieve a reliable accuracy, they could help us describe music in a way that is both quantitative and meaningful to humans.

## 3 METHODOLOGY

### 3.1 Dataset and Features

*3.1.1 GTZAN Dataset.* The GTZAN dataset originally contains ten genres (Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, Rock), each with 100 pieces of 30s audio files (1000 pieces in total). The tracks are all 22050Hz Mono 16-bit audio files in *.wav* format. There are 60 features further extracted based upon the characteristics of music related to texture, timbre and instrumentation. Among the 60 features, the key features deployed in this study are given in Table 1. Among the features, there are 20 calculated as the mean and variance of MFCC over a one-second window.

*3.1.2 Spotify Dataset.* The Spotify dataset includes 17996 entries of songs with 17 features each. The features are shown in Table 2.

### 3.2 Algorithms and Evaluation Metrics

*3.2.1 Algorithms.* The algorithms that will be used for analysis are decision Tree, random forest (RF), and k-nearest neighbors (KNN) since all three natively handle multi-class classification tasks efficiently. The overall framework illustration is depicted in Figure 1. The reasons for using a decision tree are, first, it is not susceptible to redundant and irrelevant features, meaning when it is hard to extract or reduce features, it will not be greatly affected. Second, it can possibly help understand the differences and similarities among genres by visualizing the splits. RF, since it is a better version of the decision tree, serves as a test of performance and feasibility rather than a tool for understanding. KNN serves a similar purpose but offers another perspective since it is a lazy learner, unlike decision tree and RF.

All models will be coded in python using Scikit-learn library [5], which includes built-in cross-validation, using test sets of 33% of original data.

*3.2.2 Evaluation Metrics.* Accuracy score will be used for the GTZAN dataset since it is extremely balanced. For the Spotify dataset, class imbalance may be present. Tischio and Weiss [18] suggest that the F1 score values rare cases and classes in the presence of class imbalance because both precision and recall (components of F1) are defined with respect to the rare class. Accuracy of classification, in a broader sense, is very important for music genre classification since a wrong prediction can have disastrous effects on recommender systems. The confusion matrix will be heavily relied upon for interpreting results qualitatively.

### 3.3 Exploratory Data Analysis

It is important to first explore the dataset to understand the distribution of the class, the features, and their correlations. Therefore, below are some key visualizations for both datasets.

Since the GTZAN dataset is extremely balanced, it is unnecessary to show its distribution. However, we can observe how the features correlate to each other with the heatmap of correlations depicted in Figure 2.

Although most features have no correlations with others, some features seem to have strong positive or negative correlations (particularly on the top left corner). Notably, the "spectral centroid mean" has high correlation with "spectral bandwidth mean," "spectral bandwidth variance," "rolloff mean," and "zero crossing rate mean". There is also a somewhat strong negative correlation between the MFCC2 mean and almost all other non-MFCC attributes. This might pose a problem for the KNN algorithm since it can be affected by highly correlated or redundant features. Therefore, they will be considered in the KNN model.

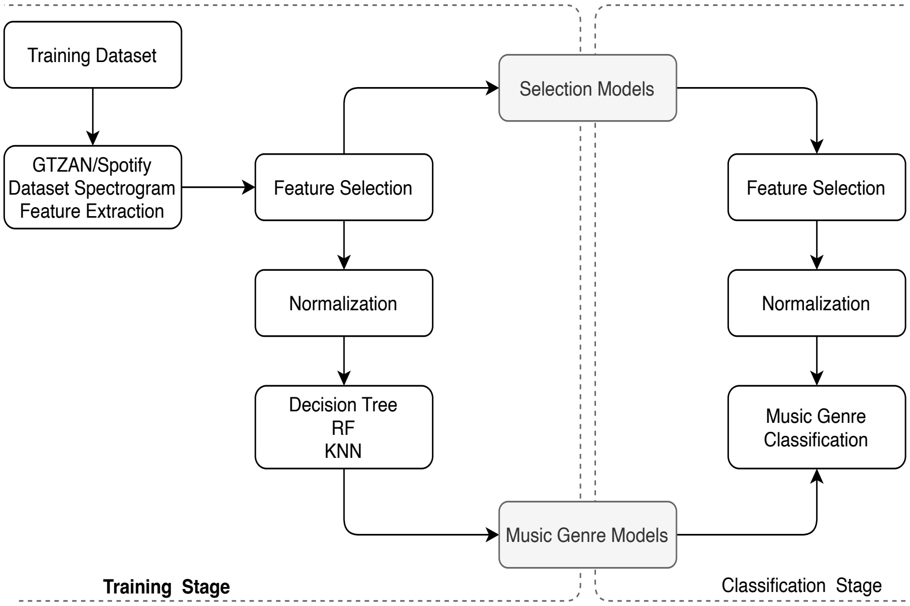


**Figure 1: The diagram for our ML-enabled music genre classification.**

**Table 1: GTZAN dataset features.**

| Feature | Description |
|---|---|
| Filename | Name of the file |
| Length (int) | In milliseconds |
| Centroid (float) | A measure of spectral brightness |
| Rolloff (float) | A measure of spectral shape |
| Flux (float) | A measure of spectral change |
| ZeroCrossing (float) | The number of time domain zerocrossings of the signal (useful to detect the amount of noise in a signal) |
| Low Energy (float) | The percentage of analysis windows that have energy less than the average energy of the analysis windows over the texture window |
| MFCC (float) | Mel-frequency cepstral coefficients: perceptually motivated features commonly used in speech recognition research |

Next, the distribution of classes in the Spotify dataset is examined and shown in Figure 3. There is a high-class imbalance, which could resemble reality since the actual distribution of genres is quite impossible to be balanced. Sampling techniques will be used in future work to address this issue. Furthermore, Figure 4 shows the heatmap of correlations between features of the Spotify dataset.

Although most of the features have little to no correlations with each other, there are three very notable correlations; loudness vs. energy, acousticness vs. energy, and acousticness vs. loudness represented by one orange and two blue squares. This will be addressed and discussed in the next subsection.

Last, the distribution of classes in the Spotify dataset is explored and depicted in Figure 3. It is found that some durations are in minutes (representing the tallest bar on the left end) while others are in milliseconds. Therefore, a conversion factor of 60000 will be applied to the minutes so that all song durations are in ms.

## 3.4 Data Preprocessing

For both datasets, the categorical data is dropped. For the GTZAN dataset, it is the filename. It serves no other purpose than recordkeeping. It may also giveaway information since the file names contain genre. For Spotify dataset, it is the artist name and track name. The reason for its removal is to eliminate the effect of the artist in determining the genre of the music. Since most artists specialize in one genre, predicting the genre given the artist's name is sometimes trivial.

GTZAN dataset is complete because of its small size. Therefore, it contains no duplicates or empty values. However, the Spotify dataset has over 4000 entries with the instrumentalness feature missing. They will be dropped as there is no meaningful way to fill in the missing values. Additionally, the length feature is dropped for the GTZAN dataset since the length of all the audio files is the same (30s).

**Table 2: Spotify dataset features.**

| Feature | Description |
|---|---|
| Artist Name | Name of the artist |
| Track Name | Name of the track |
| Popularity (int) | A 0-to-100 score that ranks how popular a track is relative to other artists on Spotify |
| Acousticness (float) | A confidence measure from 0.0 to 1.0 of whether the track is acoustic |
| Danceability (float) | Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity |
| Duration (int) | The duration of the track in milliseconds |
| Energy (float) | A measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity |
| Instrumentalness (float) | Predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content |
| Key (int) | The key the track is in. 0 = C, 1 = C♯/D♭, and so on. -1 = No key detected |
| Liveness (float) | Detects the presence of an audience in the recording |
| Loudness (float) | The overall loudness of a track in decibels (dB). Values typical range between -60 and 0 db |
| Mode (int) | Indicates the of a track. Major = 1, minor = 0 |
| Speechiness (float) | Detects the presence of spoken words in a track |
| Tempo (float) | The overall estimated tempo of a track in beats per minute (BPM) |
| Time Signature (int) | The overall time signature of a track (how many beats are in each bar or measure) |
| Valence (float) | A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track |
| Class (int) | 11 classes encoded with [0, 10] representing Acoustic/Folk, Alt music, Blues, Bollywood, Country, Hip-Hop, Indie Alt, Instrumental, Metal, Pop, Rock |

When using KNN, data must be standardized to be on the same scale. For this reason, sklearn's standard scaler and robust scaler will be used to test if the latter, which is robust to outliers, will improve the results. Furthermore, KNN is susceptible to redundant features. For this reason, features from both datasets with high (positive or negative) correlations with other features will be dropped for the analysis. For the GTZAN dataset, the features are spectral centroid mean, spectral bandwidth mean, rolloff mean, MFCC1 mean, MFCC2 mean which are in the top left corner of the heatmap. For the Spotify dataset, the features are loudness and acousticness.

## 3.5 Parameter Tuning

Grid search is performed here to help select the best parameters for each model using scikit-learn's GridSearchCV. After selecting the best parameters, most of them will be kept, but some key hyperparameters will still be tuned within a range, and graphs will be generated to show their effects.

The algorithms, their corresponding parameters, and the values that will be used are shown in Table 3.

# 4 EVALUATION

## 4.1 GTZAN Dataset

*4.1.1 Decision Tree.* First, we construct the confusion matrix, an $NxN$ matrix (where $N$ is the number of target classes), to examine our classification model's performance. The confusion matrix will contrast the actual target values with the values predicted by the ML model. The confusion matrix is shown in Figure 5.a. In this evaluation scenario, the best parameters are criterion: entropy, max_features: None, and splitter: best.

Based on Figure 5.a, with an accuracy of 0.65, it is shown that these features can be indeed used to distinguish between the genres. Among all the genres, the most easily distinguishable ones for the algorithm are Classical and Metal, with accuracies of 0.83 and 0.77, respectively. This is because they are sonically on the extremes. Classical music is characterized by quiet, but high, dynamics (having a large difference in loudness between the loud and quiet parts), and complexity in a harmonic organization (i.e., superposition of multiple ensembles of acoustic instruments in all registers). Metal (or heavy metal) music is often characterized by its consistent loudness, thick and massive sound, and distortion. This can be evident in the spectrograms generated using the GTZAN dataset. A spectrogram is a visual representation of a sound signal that shows the frequency on the y-axis and how they change over time on the x-axis. The brightness of color represents the intensity of that frequency at that point in time.

Moreover, Figure 6 depicts the two most representative spectrograms. On the one hand, we can see that the Classical piece on the left has many highly distinct horizontal lines (or blocks) of frequencies, corresponding to complex instrumentations and harmonics. It also has brighter and darker parts, corresponding to its varying dynamics. On the other hand, the Metal piece on

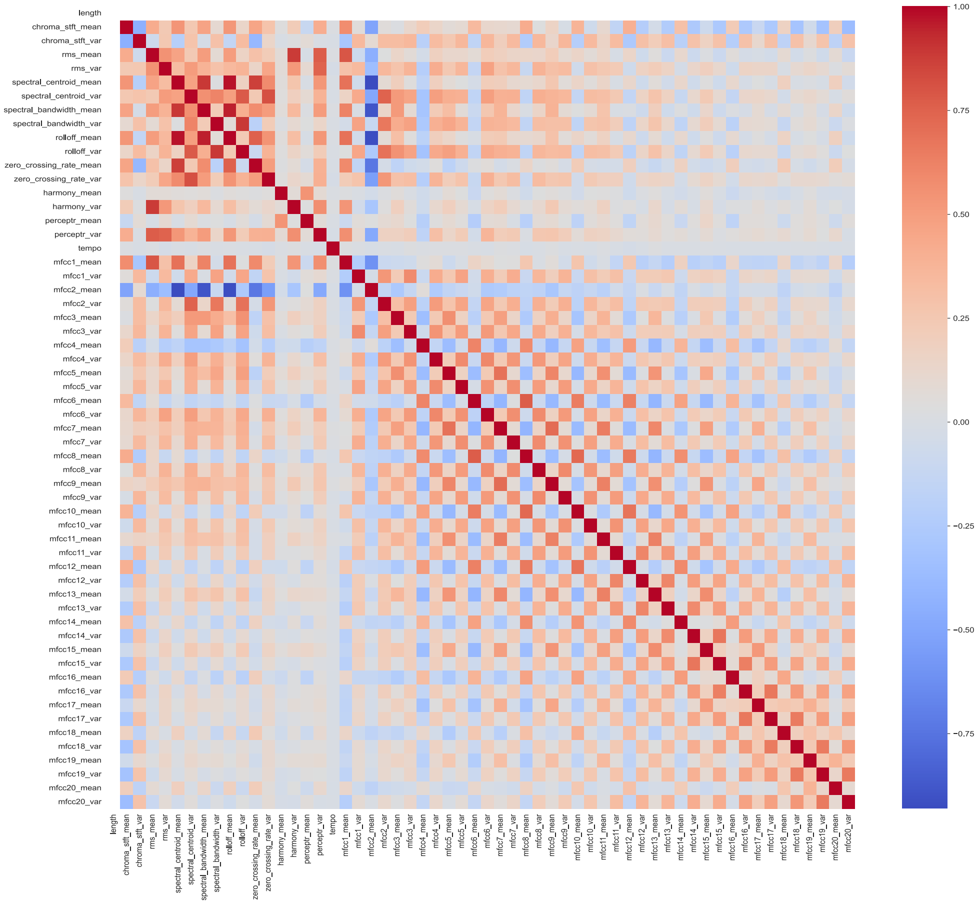
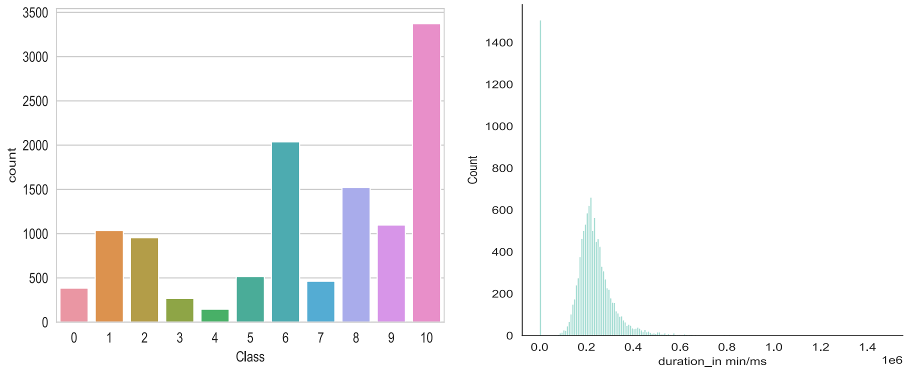


**Figure 2: Heatmap of correlations between features in GTZAN dataset.**



(a)

(b)

**Figure 3: Distribution of classes in GTZAN dataset (a) and Spotify dataset (b).**

the right has highly consistent and loud dynamics with the indistinguishable lines representing distorted frequencies, which often occupy a wider range.

*4.1.2 RF.* Next, we construct the confusion matrix for the RF model to evaluate its performance. The matrix is shown in Figure 5.b. The best parameters herein are criterion: gini, max_features: sqrt, and n_estimators: 230.

As shown in Figure 5.b, the accuracy is improved significantly compared to a single decision tree (from 0.65 to 0.86). Although these features produced promising results, the level of accuracy

**Table 3: Parameters to be tuned. The intervals used for the N estimators and N neighbors in RF are 10 and 5, respectively.**

| Algorithm | Decision Tree | | | RF | | KNN | | | |
|---|---|---|---|---|---|---|---|---|---|
| Parameter | Criterion | Splitter | Max Features | N estimators | Bootstrap | N neighbors | Weights | Algorithms | Metric |
| Value | GINI<br>Entropy<br>-<br>- | Best<br>Random<br>-<br>- | Sqrt<br>$Log_2$<br>Auto<br>None | 1 to 251<br>-<br>-<br>- | True<br>False<br>-<br>- | 1 to 201<br>-<br>-<br>- | Distance<br>Uniform<br>-<br>- | Ball tree<br>Kd tree<br>Brute<br>Auto | Euclidean<br>Minkowski (p=1,3)<br>-<br>- |

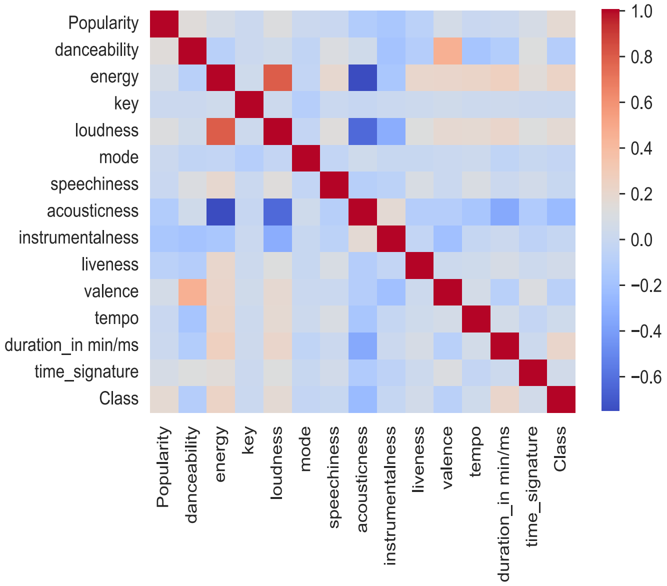


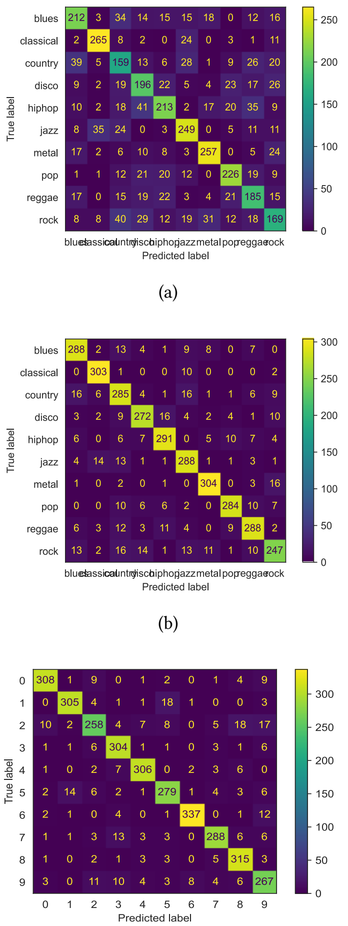
**Figure 4: Heatmap of correlations between features in Spotify datasets.**

might still not be enough to be called reliable. Therefore, a thorough examination of the accuracy variation is needed. Figure 7.a shows a graph of how accuracy varies with different parameters. Irrelevant parameters are not shown so the focus is on the number of estimators.

As expected, accuracy increases as the number of estimators increases. It increased sharply from 1 to 10 estimators and plateaued at over 100 estimators. It appears that although the number of estimators is high, it is still relatively quick to build (on the order of minutes), although the slowest among the three.

*4.1.3 KNN.* Next, we also construct the confusion matrix for the KNN model to evaluate its performance. The matrix is shown in Figure 5.c. The best parameters in this experiment scenario are algorithm: auto, metric: euclidean, n_neighbors: 1, and weights: uniform.

As shown in Figure 5.c, the KNN model performed slightly better than RF, with an accuracy of 0.9. Interestingly, misclassified genres (True label vs. Predicted label) are largely the same. This may suggest that there is a flaw with the data itself. This could be improved either in the preprocessing phase or data collection phase and will be examined in future works. One point to note is that dropping redundant features and switching between standard scaler and robust scaler made the accuracy fluctuate by less than 0.02. However, the improvement is not highly meaningful.

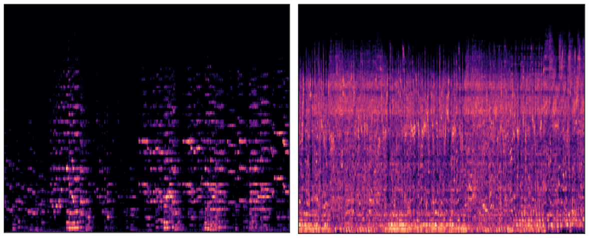


(a)

(b)

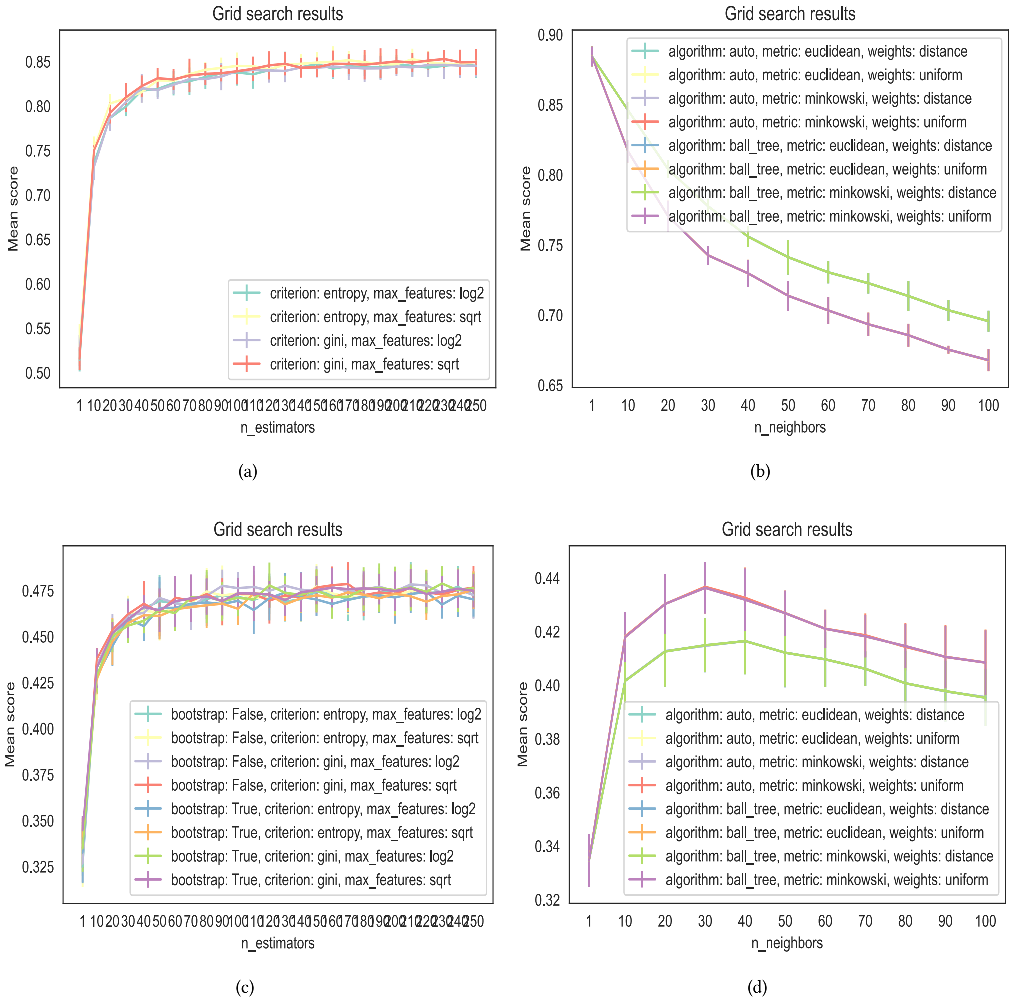
(c)

**Figure 5: GTZAN dataset confusion matrix for decision tree (a), RF (b), and KNN (c).**



(a) (b)

**Figure 6: Spectrogram of Classical (left) vs Metal (right).**



(a)

(b)

(c)

(d)

**Figure 7: Grid search results based on GTZAN for RF (a) and KNN (b). Grid search results based on Spotify for RF (c) and KNN (d).**

Furthermore, Figure 7.b shows the grid search results for the KNN model. Interestingly, 1 nearest neighbor produces the highest accuracy, and the accuracy continues to decrease as the number of neighbors increases. The high accuracy is not due to memorized training data because of the train-test-split and cross-validation (also, the error rate is not 0). Specifically, this is because $1NN$ produces low bias and high variance, meaning that the model is the closest to the training data. However, it is not robust and can easily model the noise in the data. Therefore, although $1NN$ produces the highest accuracy, a better number of neighbors is likely greater than 1 (so that the variance is lowered). Last, KNN did not fall to the curse of dimensionality as dealing with more than 50 features as the features are neither redundant nor irrelevant after pruning.

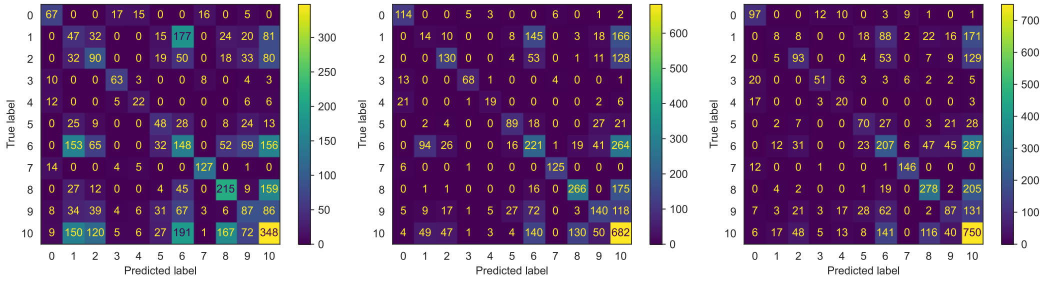
## 4.2 Spotify Dataset

*4.2.1 Decision Tree.* For the Spotify dataset, we first construct the confusion matrix for the decision tree model to evaluate its performance. The matrix is shown in Figure 8.a. The best parameters here are class_weight: None, criterion: gini, max_features: log2, and splitter: best.

As depicted in Figure 8.a, decision tree using Spotify dataset produced bad results with weighted average F1 score 0.33. It appears that the model has genuine mislearning about the genres. Upon investigation, it was found that the dataset contains some genre overlap (i.e., genres are not significantly distinct from each other). Most notably, Alt Music (1) and Indie Alt (6), which suggests overlap just by their names, as well as Metal (8), are all subgenres of Rock (10). This can explain the significance of misclassification in rows 1, 6, 8, 10, which falls into the remaining three genres.

*4.2.2 RF.* Next, a confusion matrix is also calculated for the RF model, which is shown in Figure 8.b. The best parameters here are bootstrap: True, criterion: gini, max_features: $log_2$, and n_estimators: 230.

As shown in Figure 8.b, RF again improved the results from decision tree (from 0.33 to 0.46 for f1-score) and further validated the genre overlap assumption. Among true labels 1,6, 8, and 10 (Alt,



(a) (b) (c)

**Figure 8: GTZAN dataset confusion matrix for decision tree (a), RF (b), and KNN (c).**

Indie/Alt, Metal, and Rock), a significant portion has been predicted to be in the other three genres. However, with improved accuracy, some other patterns and observations have emerged. First, the most trivial among the observations is that the Instrumental (7) genre has an F1 score of 0.93. This is due to instrumental pieces scoring highly in instrumentalness and acousticness feature, as well as having a low loudness. This may suggest that Instrumental might not be a good "genre" instead of just a description. Second, the algorithm confuses a significant portion (42%) of Blues (2) as Rock (10). By comparing the mean and standard deviation of both genres, it has been found that they differ in many features significantly, such as loudness, energy, acousticness. The remarkable misclassification could be due to Rock being the majority class in this dataset, outweighing any other genre. It is interesting to note that Blues scores the $3^{rd}$ highest in valence scale (0.61), suggesting that they are not particularly "blue" and actually convey positive energy.

Moreover, Figure 7.c gives the grid search result for the RF model. Unsurprisingly, the grid search result looks highly similar to the GTZAN dataset. This time, bootstrap is used and marginally improves the result. Because the accuracy keeps increasing, and in fear that the best result is not produced by 230 estimators, which is near our arbitrary upper limit of 250, another grid search is run in the range of 150 to 300 estimators. The best result is produced by 180 estimators with slightly different other parameters, and the curve even sees a decreasing trend (or just bouncing around).

*4.2.3 KNN.* Last, the confusion matrix for the KNN model using the Spotify dataset is calculated and shown in Figure 8.c. The best parameters here are algorithm: auto, metric: minkowski, n_neighbors: 30, weights: uniform, and p: 1.

Unlike the KNN for the GTZAN dataset, the best result is produced by 30 neighbors. Perhaps the Spotify dataset contains a lot of noise and outliers. Some features that may contain extreme values, like duration, have been examined. Apart from standardizing all the duration in milliseconds, an arbitrary boundary of 60000 ms (1 min) to 360000 ms (6 min) has been set that eliminates about 8% of extreme data (in accordance with a reasonable length for songs). Based upon this, the model surprisingly performed worse. This is due to four genres in which durations are all expressed in minutes instead of ms. Putting all durations on the same scale sacrifices performance on training data, but it will make the model perform better on new data samples.

**Table 4: Accuracy measures of algorithms (highest).**

| Algorithm | GTZAN | Spotify |
|---|---|---|
| Decision Tree | 0.65 | 0.33 |
| RF | 0.86 | 0.46 |
| KNN | 0.90 | 0.44 |

However, dropping redundant or irrelevant features (e.g., 1: loudness, which is a component of energy; 2: time signature, for which the majority of songs have the value of 4; 3: instrumentalness, for which the majority of songs score very low; 4: popularity, which might have more to do with how recent a song is rather than its genre) did not improve the performance. Using a robust scaler over a standard scaler still marginally improves performance (only by 0.02). The increase here is evidence that robust scaler enables KNN to handle extreme outliers better.

Last, Figure 7.d presents the grid search results for the KNN Spotify. The grid search results are in line with expectations, with a sharp increase at the start, a peak around a precise value, and lowered f1 scores as the number further increases (introducing high bias while reducing the variance). The $1NN$ produces a bad learning result likely suggests that the variance of the original data is high and that although songs in a genre have commonalities, they can be significantly different. The overall poor performance of KNN can also be partly attributed to the significant class imbalance, aside from the infeasibility of this dataset.

## 4.3 Overall Performance

Finally, we examined and compared the overall performance achieved by all three ML models across the GTZAN and Spotify datasets. The overall performance accuracy results are summarized in Table 4.

The algorithms' performances on the GTZAN dataset are almost two times those on the Spotify dataset. On the one hand, it is relatively reasonable to conclude that the low-level features from the GTZAN dataset are more suitable for ML implementations than the high-level features from the Spotify dataset. However, the MFCC values that make up most features in the GTZAN dataset merely describe the perception of the sound signal of that 1-second window. Because a sound signal must undergo transformations to arrive at an MFCC value, it no longer provides any easily understandable insights.

On the other hand, the Spotify dataset suggests that some features exist that can allow us to interpret music both qualitatively and quantitatively. The Spotify features are a step in the right direction, but still not good enough.

## 5 DISCUSSION

There are a few aspects in this work that can be improved or further extended in future studies. The aspects are summarized as follows.

- Class imbalance can be better addressed using sampling techniques such as Synthetic Minority Oversampling TEchnique (SMOTE).
- Further algorithms that deal with multi-class classification problems, particularly neural networks, can be used to test the feasibility of the audio features for ML use.
- Further features can be examined or extracted to better understand how the Spotify features are generated. Future features can include images to enable the classification of spectrograms using computer vision. It would be interesting to explore how the ML algorithms perform the classification of audio using computer vision.
- Adoption of information fusion techniques to rank the features' types based on their ML models performance and their rank combination understandability.
- Deploying unsupervised learning, particularly clustering, to gain more insights into music genres. A partitional clustering like K-means may introduce new ways to categorize music. A hierarchical clustering may suggest unseen relationships between songs in different genres.
- Although the availability of the GTZAN dataset has made it a benchmark dataset and thus a measuring stick for comparing MGR systems, some faults are likely to reside in the dataset. Sturm [16] suggests that it may have several faults in its integrity, including repetitions, mislabeling, and distortions. On top of that, 1000 samples from 10 genres are far from producing meaningful real-world results. Therefore, more data from the real-world music should be tested.

Last, an interpretation (if possible) of how the split point is achieved in the decision tree can be a potential research direction for this work. The hope here is to generate rules (like rule-based classifiers) from the decision tree, so that we could obtain some interesting rules like $(\text{loudness} \geq -9\text{db}) \wedge (\text{danceability} \geq 0.66) \wedge (\text{speechiness} \geq 0.1) \rightarrow \text{Hip Hop}$ (not verified, although probably true).

## 6 CONCLUSION

Genre is a substantial feature of music, and it is regarded as a high-level description of an individual piece of music. As a result, if a genre can be interpreted as a principal component of a piece of music, it can be further broken as a set of features for ML implementations. To improve the field of research in MIR, in this work we aimed to identify the genre with lower-level features as key components of MGR. Specifically, to analyze the music for use in music recommender systems, we focused on examining and describing music genres in a quantitative way using various ML models. We further compared and analyzed the feasibility, performance, and understandability of some features used to describe music by predicting the genre using decision tree, RF, and KNN techniques. By leveraging the MFCC features from GTZAN and human-understandable features from Spotify, this paper demonstrates a trade-off between classification accuracy and understandability of the music features.

## REFERENCES

[1] Hareesh Bahuleyan. 2018. Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149* (2018).

[2] Aathreya S. Bhat, V.S. Amith, Namrata S. Prasad, and D. Murali Mohan. 2014. An Efficient Classification Algorithm for Music Mood Detection in Western and Hindi Music Using Audio Feature Extraction. In *ICSIP*. 359–364. https://doi.org/10.1109/ICSIP.2014.63

[3] Snigdha Chillara, AS Kavitha, Shwetha A Neginhal, Shreya Haldia, and KS Vidyullatha. 2019. Music genre classification using machine learning algorithms: a comparison. *Int Res J Eng Technol* 6, 5 (2019), 851–858.

[4] Ahmet Elbir and Nizamettin Aydin. 2020. Music genre classification and music recommendation by using deep learning. *Electronics Letters* 56, 12 (2020), 627–629.

[5] Jiangang Hao and Tin Kam Ho. 2019. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics* 44, 3 (2019), 348–361.

[6] Chandanpreet Kaur and Ravi Kumar. 2017. Study and analysis of feature based automatic music genre classification using Gaussian mixture model. In *ICICI*. IEEE, 465–468.

[7] Tao Li and Mitsunori Ogihara. 2005. Music genre classification with taxonomy. In *ICASSP*, Vol. 5. IEEE, v–197.

[8] Tao Li, Mitsunori Ogihara, and Qi Li. 2003. A comparative study on content-based music genre classification. In *ACM SIGIR conference on Research and development in informaion retrieval*. 282–289.

[9] Kevin Martin, Izzat Alsmadi, Mohamed Rahouti, and Moussa Ayyash. 2020. Combining Blockchain and Machine Learning to Forecast Cryptocurrency Prices. In *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 52–58.

[10] Kevin Martin, Mohamed Rahouti, Moussa Ayyash, and Izzat Alsmadi. 2022. Anomaly detection in blockchain using network representation and machine learning. *Security and Privacy* 5, 2 (2022), e192.

[11] Sergio Oramas, Francesco Barbieri, Oriol Nieto Caballero, and Xavier Serra. 2018. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.* (2018).

[12] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. 2016. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing* 2016, 1 (2016), 1–16.

[13] Mohamed Rahouti, Moussa Ayyash, Senthil Kumar Jagatheesaperumal, and Diogo Oliveira. 2021. Incremental Learning Implementations and Vision for Cyber Risk Detection in IoT. *IEEE Internet of Things Magazine* 4, 3 (2021), 114–119.

[14] Christine Senac, Thomas Pellegrini, Florian Mouret, and Julien Pinquier. 2017. Music feature maps with convolutional neural networks for music genre classification. In *CBMI*. ACM, 1–5.

[15] Srishti Sharma, Prasenjeet Fulzele, and Indu Sreedevi. 2018. Novel hybrid model for music genre classification based on support vector machine. In *ISCAIE*. IEEE, 395–400.

[16] Bob L. Sturm. 2013. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *CoRR* abs/1306.1461 (2013). arXiv:1306.1461 http://arxiv.org/abs/1306.1461

[17] Bob L Sturm, Oded Ben-Tal, Úna Monaghan, Nick Collins, Dorien Herremans, Elaine Chew, Gaëtan Hadjeres, Emmanuel Deruty, and François Pachet. 2019. Machine learning research that matters for music creation: A case study. *Journal of New Music Research* 48, 1 (2019), 36–55.

[18] Ray Marie Tischio and Gary M Weiss. 2019. Identifying classification algorithms most suitable for imbalanced data. (2019).

[19] G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10, 5 (2002), 293–302. https://doi.org/10.1109/TSA.2002.800560

[20] S Vishnupriya and K Meenakshi. 2018. Automatic music genre classification using convolution neural network. In *ICCCI*. IEEE, 1–4.

[21] Wenli Wu, Fang Han, Guangxiao Song, and Zhijie Wang. 2018. Music genre classification using independent recurrent neural network. In *CAC*. IEEE, 192–195.