Music Genre Recognition with Handcrafted Audio Features

André Luís de Oliveira[1], **Luiz Fernando Carvalho**[1],
Daniel Prado Campos[1], **Rafael Gomes Mantovani**[1]

[1]Federal University of Technology – Paraná (UTFPR), Apucarana, Paraná, Brazil

andre.2002@alunos.utfpr.edu.br,
{luizfcarvalho,danielcampos,rafaelmantovani}@utfpr.edu.br

Abstract. *Music has been increasingly prominent in people's lives in recent years. Technology integration with music is progressively increasing, directly contributing to the enhancement and understanding of this art form. In this sense, Music Genre Recognition (MGR) applies Machine Learning (ML) to identify similar patterns in music and classify them. This study assessed traditional ML algorithms in the GTZAN dataset, classifying audio songs into ten genres. The results showed that the XGBoost classifier statistically outperformed all the other algorithms evaluated, with an accuracy value of* 0.722*. Future experiments can improve it with a more robust feature engineering process, exploring and mixing deep with traditional features.*

1. Introduction

Music has been playing an increasingly prominent role in people's lives in recent years. With the universalization of the Internet, music has become more accessible and democratic, thanks to software and websites that facilitate access, such as Spotify and YouTube. Additionally, with each passing day, the number of songs created and made available to the public increases, making an analytical study of this type of art increasingly necessary. This is where technology comes into play.

Technology integration with music is progressively increasing, directly contributing to the enhancement and understanding of this art form. Artificial intelligence (AI) [Russell and Norvig 2010] plays a significant role in this integration. In this sense, Data Mining (DM) [Tan et al. 2005] techniques and Machine Learning (ML) [Mitchell 1997] algorithms can be employed to extract various types of information from musical pieces, facilitating studies based on the obtained data. The field dedicated to extracting and analyzing information from musical works is known as Music Information Retrieval (MIR). Different tasks encompassed by MIR include: cover song detection [McFee et al. 2015], music recommendation [Song et al. 2012], tempo estimation [Alonso et al. 2004], query by humming [Ghias et al. 1995], instrument recognition [Livshin 2007], and genre recognition [Tzanetakis and Cook 2002].

Due to the vast amount of music created continuously, manual classification often becomes tiring and inefficient. Hence, Music Genre Recognition (MGR), a sub-field of MIR, aims to identify similar patterns in music of the same genre and classify them accordingly [Ghildiyal et al. 2020]. In a few words, a sound can be represented by an audio signal, from which various features can be extracted in both the time and frequency domains. These features can then be used to train a ML algorithm, obtaining a model to predict the genre of new songs.

Recent studies explore the MGR through traditional ML algorithms and Deep Learning (DL) architectures [Aggarwal 2018]. However, even though Convolutional Neural Networks (CNNs) often yield highly satisfactory results, they come with significant computational costs and require extensive training time. Conversely, traditional ML algorithms are less computationally intensive but do not achieve the same level of accuracy as deep neural networks. Thus, this study aims to develop efficient ways to apply traditional ML algorithms to achieve satisfactory results in MGR without incurring high computational costs. The desired pipeline includes the following sub-tasks: audio preprocessing, feature extraction, application of ML algorithms, and evaluation of the obtained results.

This paper is organized as follows: Section 2 presents some of the necessary concepts about MGR and describes related work. The experimental methodology is presented in Section 3. The results are discussed in Section 4, while the conclusions and final considerations of the study are presented in Section 5.

2. Related Works

Machine Learning (ML) can be described as a system's ability to find inductive biases while performing a task, improving performance in future executions of that task and similar ones [Mitchell 1997]. This concept is widely applied to classification problems, where labels are assigned to elements based on their characteristics. To classify a song into a genre, it is essential to identify relevant features present in those songs. As an audio signal, a song can be mathematically represented as a function of one or more independent variables [Oppenheim and Willsky 1997]. Using signal analysis and processing techniques, various features can be extracted from these audio signals, both in the time and frequency domain, and, with these characteristics, researchers aim to use ML algorithms to effectively classify songs into specific genres.

2.1. Studies on Music Genre Recognition

One of the earliest studies aimed at performing Music Genre Recognition (MGR) is [Tzanetakis and Cook 2002]. In this study, the authors extracted features based on timbre, rhythm, and pitch, and attempted to classify music into ten distinct genres, as well as classify some of these genres into subgenres. They utilized models from standard statistical pattern recognition (SPR), such as the Simple Gaussian Classifier (GS), Gaussian Mixture Model Classifier (GMM), and K-Nearest Neighbors (KNN). The best result was achieved with the GMM, with an accuracy of 0.61.

Recently, [Bahuleyan 2018] utilized a Convolutional Neural Network (CNN) fed with images of the song's spectograms, achieving an accuracy of 0.64 in a dataset with ten classes. Additionally, this study also evaluated traditional algorithms like: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM) and Extreme Gradient Boosting (XGB). For these algorithms, the features used included: Central Moments, Zero Crossing Rate (ZCR), Root Mean Square Energy (RMSE), Tempo, Mel-Frequency Cepstral Coefficients (MFCC), Chroma Features, Spectral Centroid, Spectral Bandwidth, Spectral Contrast and Spectral Rolloff. The best result was obtained with XGB with an accuracy of 0.59. Furthermore, an ensemble between the CNN and XGB was also evaluated, achieving an accuracy of 0.65.

In [Shah et al. 2022], both traditional ML algorithms and a CNN were used in experiments, resulting in relatively similar accuracy for all of them. The XGB achieved

an accuracy of 0.712, the CNN 0.741, the RF 0.662, and the SVM 0.691. In this study, the CNN was fed with spectrogram images of the songs, while the other algorithms received hand-crafted features obtained with the Librosa library.

In [Ghildiyal et al. 2020], the authors adopted a slightly different approach when training neural networks. Instead of using complete spectrograms of the music, each spectrogram was divided into 64 slices, thereby increasing the data quantity by 64 times. A larger portion of these slices was used for training, a smaller part for validation, and the rest for testing. This approach resulted in an accuracy of 0.91 with the CNN. The study also provides an analysis of the effectiveness of each tested algorithm in classifying each specific genre.

Finally, in [Ndou et al. 2021] the authors adopted an approach of splitting the 30-second audio clips from the GTZAN dataset into 3-second slices, thereby increasing the data volume. Various sets of features were extracted for both the 30-second and 3-second audio clips. This study achieved remarkably high results for traditional methods, particularly with kNN, which obtained an accuracy of 0.92 using a specific set of features extracted from the 3-second audio slices.

2.2. Literature Summary

From this, it is evident that neural network-based algorithms, especially CNNs, tend to achieve better results. However, the considerably longer training time is a significant drawback of this approach. Studies that have achieved good results with traditional methods, such as [Ndou et al. 2021], likely suffer from bias. The audio clips were divided into 3-second slices, but slices from the same song could fall into different sets when splitting the training and testing sets. This likely inflated the accuracy because the models were trained on some slices of a song and tested on others from the same song. Similarly, the division of spectrograms into 64 slices in [Ghildiyal et al. 2020] may have experienced the same bias.

3. Experimental Methodology

This section presents the experimental methodology used in the benchmarking experiments of the ML algorithms. An overview of the flow of experiments, including sub-steps, is shown in Figure 1. The following subsections will explain each sub-task in detail.

3.1. Dataset

This study performed experiments with the GTZAN [Tzanetakis et al. 2001], a pre-labeled dataset with songs from different genres. This dataset is widely used in similar studies, allowing future comparisons. The GTZAN is a balanced dataset containing 1000 songs across 10 distinct genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, and Rock. Each class/genre has 30-second clips from 100 different songs with a sample rate of 44.1 KHz. The dataset is available on Kaggle[1].

[1] https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

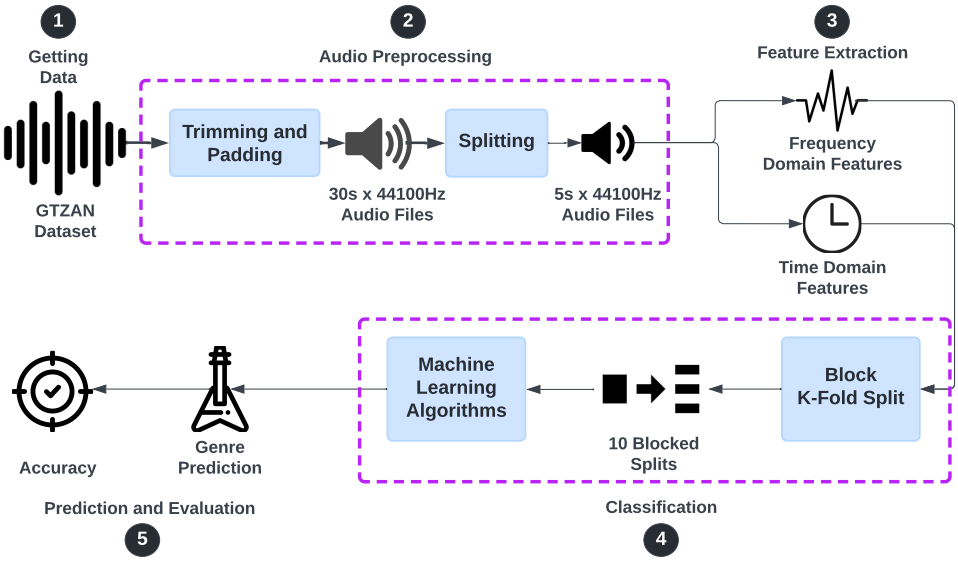

Figure 1. Methodology Pipeline adopted in the experiments for Music Genre Recognition

3.2. Audio Preprocessing

After the data acquisition, several preprocessing operations were performed to ensure the experiments could progress more effectively. An audio signal can be digitally represented by a sequence of samples in the discrete domain, given a specific sampling rate [Oppenheim and Willsky 1997]. As stated before, each dataset instance was a 30-second audio clip with a sampling rate of 44.1 kHz, resulting in 1,323,000 audio samples. However, some of the audio files had slightly more or fewer samples. To resolve this, trimming and padding were performed. Instances with extra samples were trimmed, and those with fewer samples were padded with zeros, ensuring all audio files had the same number of samples.

Next, similarly to [Ndou et al. 2021], the 30-second dataset instances were split into smaller slices to increase the data available for analysis. Prior empirical tests were conducted with slices with size = $\{3, 5, 10\}$ seconds. Results demonstrated that slices with 5 seconds presented the best accuracies and a fast running time. Thus, each 30-second audio clip was divided into six 5-second clips, with these new clips retaining the label of the original audio instance. Thus, the 1000 original instances became 6000 after the splitting.

3.3. Feature Extraction

DL algorithms handle raw data and extract abstract features innerly. However, traditional ML algorithms do not. They require a feature extraction sub-task to generate handcrafted descriptors to proceed with the model induction. We extracted two categories of features: time domain (TDF) and frequency domain features (FDF). The TDF extracted were:

- **Zero Crossing Rate (ZCR):** it measures the rate at which the waveform crosses the zero axis [Torres-García et al. 2021]. It was obtained using a frame length of 2048 and a hop length of 1024;

- **Root Mean Square Energy (RMSE):** it evaluates the signal amplitude and energy. It is calculated as the square root of the mean of the sum of squares of the signal [Sehgal et al. 2022]. It was obtained using a frame length of 2048 and a hop length of 1024.

Complementary, we also extracted the following FDF:

- **Chroma features:** A chroma feature is a key audio melody feature in MIR, based on the Twelve-Tone Equal Temperament. It captures musical notes perceived as similar to an octave, providing useful musical information even without the original octave's frequency. Represented as a 12-dimensional vector corresponding to the 12 pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B), it reflects the local energy distribution of the audio signal at these semitones [Shi et al. 2019];

- **Mel-frequency Cepstral Coefficients (MFCC):** they represent the spectral envelope of a sound signal. They are calculated by performing a Fourier transform on the sound signal, mapping the spectrum onto the Mel scale, taking the logarithm of the powers at each Mel frequency, applying the discrete cosine transform (DCT), and retaining a subset of the resulting coefficients [Bäckström et al. 2022]. To obtain them, `n_mfcc = 20` was used;

- **Spectral Centroid:** The spectral centroid is the average frequency of a spectrum, weighted by amplitude. It is computed by considering the frequencies and their normalized amplitudes as a probability distribution [Peeters 2004]. To obtain it, `n_fft = 2048` and a hop length of 1024 were used;

- **Spectral Bandwidth:** Spectral bandwidth is the difference between the highest and lowest frequencies in a continuous frequency band. It can be considered as the range of frequencies around the spectral centroid, measuring the maximum deviation of the signal on both sides of this point within a given time frame. To obtain it, `n_fft = 2048` and a hop length of 1024 were used;

- **Spectral Rolloff:** Spectral rolloff is a measure that indicates the frequency below which a certain percentage of the total spectral energy of a signal is contained. It can differentiate sounds with more energy in the lower frequencies and those with more energy in the higher frequencies [Peeters 2004]. In this work, two Rolloffs were used, one at 99% and one at 1%, extracted with the roll percent parameters of 0.99 and 0.01, respectively, as well as `n_fft = 2048` and a hop length of 1024 in both cases.

All the features were extracted by using the `librosa`[2] library. These features were previously explored in [Ndou et al. 2021]. Each of these features generates a floating matrix as output, always with the same number of rows, but different columns. This high volume of data generates preprocessed datasets with high dimensionality. One strategy explored in [Ndou et al. 2021] was to divide the matrices into segments, and summarize them using the mean and standard deviation of the corresponding values. While reducing

[2]`https://librosa.org/doc/latest/index.html`

the dimensionality of the sets, data that could positively impact the classification task is lost. In the initial experiments of this work, we replicated the methodology described above. Additionally, the calculation of the mean and standard deviation for the rolloffs and MFCCs was done independently, with separate means and standard deviations for each set of coefficients obtained. Table 1 presents all the resulting features, where '*n means*' denotes the number of means obtained from the feature matrices, and '*n sds*' is the number of standard deviations.

Table 1. Features extracted from audio samples

Feature	n means	n sds
Chroma Features	12	12
Root Mean Square Energy (RMSE)	1	1
Spectral Centroid	1	1
Spectral Bandwidth	1	1
Spectral Rolloff	2	2
Zero-crossing Rate (ZCR)	1	1
Mel-frequency Cepstral Coefficients (MFCC)	20	20
Total	**38**	**38**

3.4. Machine Learning Algorithms

A total of seven ML algorithms were evaluated in experiments. They follow different learning biases so they can learn different decision surfaces from the same data. All algorithms are listed and briefly explained below:

- **K-Nearest Neighbors (KNN):** it considers each dataset instance as a point in an N-dimensional space, where N represents the number of features. The nearest neighbors of an instance are identified using a chosen distance metric. The classification of a new unlabeled instance is determined by the labels of its k closest neighbors [Mitchell 1997];

- **Decision Trees (DT):** the DT classifies instances by guiding them from the root to a leaf node, where the final classification is made. Each node tests a specific attribute, with branches representing possible values for that attribute. Starting at the root, the instance follows the branch corresponding to its attribute value, repeating this process down the tree until a leaf node is reached, which provides the classification [Breiman et al. 1984];

- **Logistic Regression (LR):** is a classification technique that improves linear classifiers by replacing the hard threshold function with a logistic function. This creates a continuous and differentiable boundary, addressing non-differentiability and overconfidence near decision boundaries. The logistic function outputs a probability between 0 and 1, which can be interpreted as the probability of belonging to a particular class [Russell and Norvig 2010];

- **Multiplayer Perceptron (MLP):** is a type of Artificial Neural Network (ANN) that includes one or more hidden layers between the input and output layers,

which help capture complex patterns in the data. The network is highly connected through synaptic weights, which determine the strength of the connections between neurons. These weights are adjusted during the learning process using the backpropagation algorithm, which improves the network's accuracy by minimizing the error between the predicted and actual output [Haykin 2009];

- **Support Vector Machines (SVM):** a SVM creates a decision boundary that maximizes the distance to the nearest data points, promoting effective generalization. They start with a linear hyperplane for separation but can use the kernel trick to handle data that is not separable in the original space by mapping it to a higher-dimensional space. SVMs are nonparametric, relying on a small set of critical data points (support vectors) to define the boundary rather than a fixed set of parameters. This approach blends flexibility to model complex functions with resistance to overfitting [Cortes and Vapnik 1995];

- **Random Forest (RF):** The RF algorithm improves prediction accuracy using a collection of DTs trained on different bootstrap data samples. Additional randomness is introduced by considering only a random subset of features at each node, which speeds up training and enhances diversity. This method reduces variance without affecting bias and eliminates the need for pruning. Final predictions are made by aggregating the outputs of all trees, using majority voting for classification[Breiman 2001];

- **Extreme Gradient Boosting (XGB):** it is a high-performance, scalable gradient boosting implementation known for its efficiency in handling large datasets. It uses advanced techniques like parallel and distributed computing, sparse data handling, and regularization to optimize DTs. XGB minimizes a regularized objective function, utilizing innovations like weighted quantile sketch and efficient algorithms for tree splits [Chen and Guestrin 2016].

All the traditional ML algorithms used in experiments were set with their corresponding default hyperparameter values provided by the scikit-learn library[3].

3.5. Evaluation and Reproduction of Experiments

Once features were extracted, data was divided into training and testing folds. This step performed a 10-fold cross-validation split in blocks (Blocked K-Fold). Thus, 5-second splits from the same song are arranged to the same fold. For example, if split "0" from song "A" is selected for a particular fold, all other splits from that song must also belong to that fold. This was done to prevent bias in the results, as splits from the same song have similar characteristics. All algorithms executed in experiments underwent this Blocked K-Fold process. Also, before the classification was performed, the data of all features were normalized within [0,1].

The simple predictive accuracy metric was used for evaluation since GTZAN is a balanced dataset. The experiments were repeated ten times with different seeds, reporting

[3]https://scikit-learn.org/stable/

the average accuracy values and their corresponding standard deviations. The datasets, codes, and experimental results are all publicly available in GitHub [4]

4. Results

Overall experiment results performing ten times the Blocked 10-Fold process are summarized in Figure 2. Algorithms are ordered increasingly according to their mean BAC values on the x-axis. As can be seen, the DT algorithm had the worst result, while the XGB algorithm outperformed all the others. Nevertheless, all the induced models were better then the two weak baselines: *Random*, which randomly predicts a label; and *Majority*, which always predicts a unique label to all the testing examples. They performed 0.104 and 0.10, respectively.

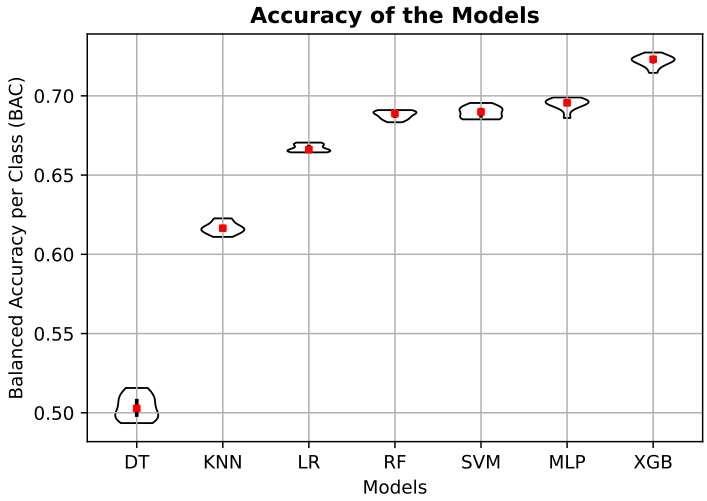

Figure 2. Overall results with BAC values obtained on the experiments.

The XGB algorithm obtained the best accuracies, averaging 0.722 considering all tested seeds. Next, a group of algorithms performed similarly (RF, SVM, and MLP), with BAC values nearly but above 0.7 and low standard deviation. The LR algorithm performed slightly lower but yielded consistent results around 0.66. Interestingly, traditional ML algorithms with handcrafted features can provide accurate results with a with a reduced computational cost. Lastly, the DT algorithm had the worst performance in average accuracy, indicating that nodes likely failed to separate the data efficiently.

The Wilcoxon statistical test was also conducted to verify the significance of the performance differences between all the tested algorithms. It was applied a 95% confidence level with $alpha = 0.05$. Significant differences were found between all algorithms except for SVM and RF. The average accuracies for SVM and RF were 0.690 and 0.688, respectively. Still, all the algorithms obtained lower performance compared to literature works with the same dataset [Ndou et al. 2021]. Using 30-second audio segments might provide more information in the spectrograms, potentially improving accuracy values. Another option is to experiment with different hyperparameters in each classifier to enhance overall performance.

[4] `https://github.com/andryll/MGR-IC`

4.1. Best Model's Predictions

Due to the strong performance of the XGB algorithm, a more detailed analysis of its results was necessary to understand where its errors are concentrated. The confusion matrix shown in Figure 3 will be used to assist in this analysis.

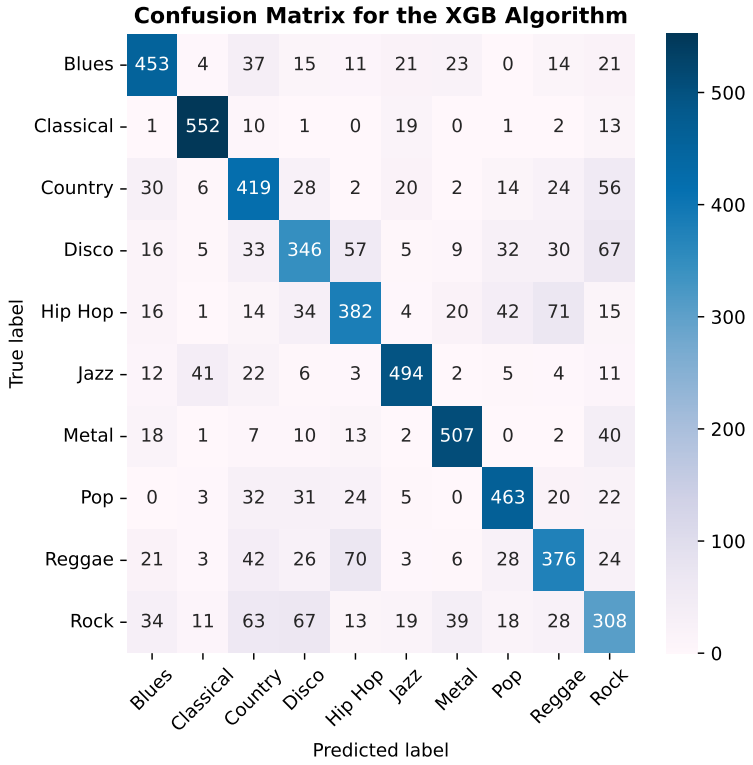

Figure 3. XGB Confusion Matrix.

From the figure, it is evident that most correct classifications are concentrated in the "Classical", "Metal" and "Jazz" labels. On the other hand, the "Rock" and "Disco" classes performed significantly worse. There is noticeable confusion between these two classes, as they represent the largest errors in each other's precision. Additionally, a considerable number of "Rock" clips were mislabeled as "Country" or "Metal." For "Jazz", the most confusions, apart from "Rock", were with "Hip Hop" and "Pop".

Based on these results, several hypotheses can explain this distribution of accuracies and errors. Genres with a more prominent instrumental focus, such as "Classical", "Jazz" and "Metal" were classified more accurately. In contrast, genres where clear vocals share prominence with instruments, like "Rock", "Reggae" and "Disco" had poorer results. Another factor contributing to the difficulty in classifying "Rock" is its many diverse subgenres, which can increase the likelihood of confusion with other genres. In future work, analyzing and utilizing features that recognize vocal patterns might enable more accurate classification for both instrumentally prominent and vocally dominant genres.

4.2. Features Importance Analysis

Due to the hypotheses raised earlier, a deeper analysis of the impact and importance of each feature in the classification was necessary. The graph in Figure 4, generated using

the XGB algorithm, will be utilized to assist with this analysis. In the figure, features are increasingly sorted in the x-axis, from left to right, to the most important ones.

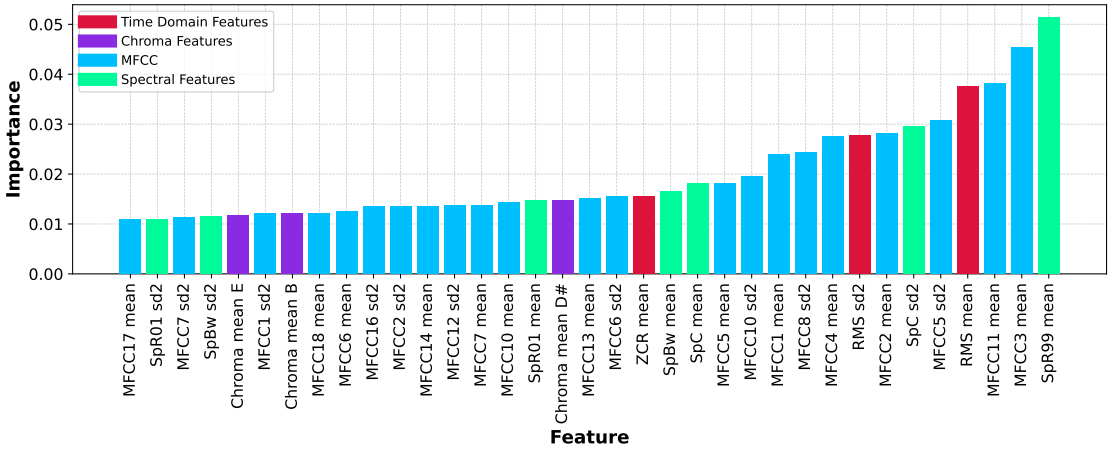

Figure 4. Features with importance above 1% color-coded by type.

Based on the figure, it is evident that among the TDFs, the RMSE demonstrated significant importance. Regarding FDFs, the Spectral Rolloff with a roll percent of 99% was highly influential in classification, with other spectral features also contributing notably. The MFCCs are well distributed throughout the graph, with some having a substantial impact while others show a low impact. Most of the lower-impact MFCCs correspond to standard deviations. No Chroma feature showed significant importance, with the highest being below 2%.

When analyzing the importance of mean and standard deviation, it is clear that the means have a much more significant impact than the standard deviations, suggesting that the use of standard deviations could be considerably reduced in future work. Regarding the TDFs, RMSE had a considerable importance, as well as other features like ZCR. It is possible that other unused TDFs could also positively contribute to the classification. For MFCCs, reducing the number of coefficients might yield more precise results. Additionally, the low impact of Chroma features indicates that they could be replaced with more impactful features. Finally, since the 99% spectral Rolloff was highly important for classification, other Rolloff percentages might also be crucial for improving classification results.

5. Conclusions

This study performed a MGR task evaluating seven different ML algorithms in the GTZAN dataset. Experiments showed that the XGB algorithm obtained the best results, with an average accuracy value of 0.722, and it proved to be an excellent choice for this task when combined with appropriate signal processing and feature extraction techniques. Dividing the audio tracks into smaller segments to increase the amount of data was effective, and using blocked slicing for cross-validation ensured unbiased evaluation. However, there are still improvements that can be made.

Understanding how to select the features to be used proved to be a challenging task. Poor feature selection can significantly degrade the results, and this selection needs

to be done optimally. In future work, based on the results obtained here, it will be possible to identify a more suitable set of features for recognizing different patterns in genres, particularly regarding the vocal characteristics of each genre. Hyperparameter tuning can also be performed to configure the algorithms for this specific task better, as well as DL algorithms could provide more powerful feature extractor.

Acknowledgments

The authors would like to thank the Federal University of Technology – Parana (UTFPR), campus of Apucarana, for the financial support.

References

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, Cham.

Alonso, M. A., Richard, G., and David, B. (2004). Tempo and beat estimation of musical signals. In *International Society for Music Information Retrieval Conference*.

Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.0114*.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Bäckström, T., Räsänen, O., Zewoudie, A., Zarazaga, P. P., Koivusalo, L., Das, S., Mellado, E. G., Mansali, M. B., Ramos, D., Kadiri, S., and Alku, P. (2022). *Introduction to Speech Processing*. 2 edition.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *arXiv:1603.02754v3*.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. (1995). Query by humming: musical information retrieval in an audio database. In *MULTIMEDIA '95*.

Ghildiyal, A., Singh, K., and Sharma, S. (2020). Music genre classification using machine learning. *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1368–1372.

Haykin, S. S. (2009). *Neural networks and learning machines*. Pearson Education, third edition.

Livshin, A. (2007). *Automatic Musical Instrument Recognition and Related Topics*. PhD thesis, Université Pierre et Marie Curie.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill.

Ndou, N., Ajoodha, R., and Jadhav, A. (2021). Music genre classification: A review of deep-learning and traditional machine-learning approaches. *2021 IEEE International IOT, Eletronics and Mecatronics Conference*.

Oppenheim, A. V. and Willsky, A. S. (1997). *Signals and Systems*. Prentice Hall.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition.

Sehgal, R., Gupta, N., Tomar, A., Sharma, M. D., and Kumaran, V. (2022). *Smart Electrical and Mechanical Systems: An Application of Artificial Intelligence and Machine Learning*. Academic Press.

Shah, M., Pujara, N., Mangaroliya, K., Gohil, L., Vyas, T., and Degadwala, S. (2022). Music genre classification using deep learning. *6th International Conference on Computing Methodologies and Communication (ICCMC)*.

Shi, L., Li, C., and Tian, L. (2019). Music genre classification based on chroma features and deep learning. In *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 81–86.

Song, Y., Dixon, S., and Pearce, M. T. (2012). A survey of music recommendation systems and future perspectives. In *The 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Addison Wesley, us ed edition.

Torres-García, A. A., Garcia, C. A. R., and Luis Villasenor-Pineda, O. M.-M. (2021). *Biosignal Processing and Classification Using Computational Learning and Intelligence: Principles, Algorithms, and Applications*. Academic Press.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5).

Tzanetakis, G., Essl, G., and Cook, P. (2001). Automatic musical genre classification of audio signals.