Music Genre Classification Using Machine Learning

Eymen Topçuoğlu, Ahmet Faruk Yılmaz, Muhammed Raşit Ayaz

ABSTRACT

Categorizing music files according to their genre is a challenging and important task that is used by many music streaming services. Even though there are no sharp differences between music genres, the genre of the music can be determined according to certain dynamics.

In this project, we aim to create different machine learning models that will predict the genre of a given music file. First, we did literature research and analyzed the different related works about music genre classification. There are many features to use for genre classification and each of them has a different effect on the different types of music. For the feature selection, we have put together what we have learned from the literature research that we did and combinations we have tried manually on the dataset.

We used GTZAN [1] as our dataset. GTZAN consists of 1000 audio tracks each 30 seconds long. Music files are organized into folders with their respective genres. There are 10 genres in total, each is represented by 100 tracks. All of the tracks are 22050 Hz Mono 16-bit audio files in .wav format. We separated each song into 10 segments to increase the size of our dataset to get better results. In the end, we had 10000 tracks, each of them 3 seconds long.

As a technology, we used Python as the programming language. We used Librosa for feature extraction, Matplotlib for visual content (graphs, plots), Pandas for data manipulation, and finally Tensorflow and Scikit for machine learning models.

OVERVIEW

The main aim of the project is to build different machine learning models that will predict the genre of the given music file and compare these models.
We used the pair programming technique during the development phase. That's why all the tasks were completed together as a team.
We divided the whole project into 6 subtasks:

1. **Literature Research - 19.11.2021**
   Initial literature research about the topic. Each team member picked different research and analyzed it. At the end of the research phase, we shared the knowledge that we gained from our research. During the whole process of the project, we continued to analyze research papers.

2. **Feature Selection - 10.12.2021**
   This subtask took more time compared to other subtasks because the key feature to get better results in the experiment is feature selection. We combined the knowledge that we gained from the research and the combinations that we tried on the model.

3. **Dataset Preprocessing - 21.12.2021**
   In the dataset preprocessing part, we extracted the features that we decided on the feature selection subtask. A library called Librosa is used for feature extraction. We saw that this library contains all of the features that are required for our project. This library is widely used in audio analysis with machine learning.

4. **Classification - 03.01.2022**
   We used 5 different classification algorithms in our machine learning models. These are Deep Neural Network, XGBoost Classifier, CNN (Convolutional Neural Network), K-Nearest-Neighbor-Classifier, and Support Vector Classifier. We got relatively poor results in K-Nearest-Neighbor-Classifier and Support Vector Classifier algorithms compared to other algorithms.

5. **Hyperparameter Tuning - 12.01.2022**
   In this part, we tried to tune the parameters to get higher accuracy. We tried different learning rates and different optimizer algorithms. We decided to use the Adam algorithm for optimization. We used 70% of the dataset for training and 30% for testing.

6. **Testing - 15.01.2022**
   After we trained our model with different classifier algorithms, we randomly downloaded the songs from the internet and used them as input. We experienced that for some music genres, our model is pretty accurate but for some genres, it gives very unexpected results.

PROJECT ACCOMPLISHMENTS

1. LITERATURE RESEARCH

Jean-Julien Aucouturier and François Pachet won the ISMIR 2004 genre classification contest at the 5th International Conference on Music Information Retrieval with their Representing Musical Genre: A State of the Art [3] research. In their article, they discussed and proposed to classify various approaches in representing musical genres. They argue that the taxonomies for titles would be more appropriate than taxonomies for albums because albums often contain titles of many different genres. They also compared three different genre taxonomies with more than hundreds of genres and concluded that there is neither consensus nor any shared structure. While doing classification, they used supervised learning methods but some part of their training data was not manually labeled, thus they had to be classified using recognition techniques. As a result, they have described three approaches for extracting musical genres.

One of them was manual classification. A manual classification of titles can be useful for bootstrapping and evaluating automatic systems, but it is not realistic for large databases and does not easily scale up. In the CUIDADO project, the authors have initially built a rather detailed taxonomy of genre for music titles, described in Pachet and Cazaly [4].

Another approach was using signal processing techniques and machine learning schemes. They reviewed 8 other contributions and compared the algorithms these researchers use. All these works proceed in two steps, which are frame-based feature extraction and machine learning classification.

The last approach was data mining techniques such as co-occurrence analysis. They have determined that 70% of the clusters constructed from such data mining distances translate interesting similarities. Some of the music genres were well distinguished using this approach.

G. Tzanetakis et al. [1] suggested an automatic classification method of audio signals. Three feature sets for representing timbral texture, rhythmic content, and pitch content are suggested.

Timbral texture feature set consists of standard features that are based on general audio classification, rather than for musical content. It is formed by 19 features: means and variances of spectral centroid, roll-off, flux, zero crossings over the texture window, low energy, and means and variances of the first five MFCC (Mel-frequency cepstral coefficients) over the texture window.

The rhythmic content feature set is based on the beat histogram of the corresponding audio signal and it consists of 6 features: relative amplitude of the first and second histogram peak, the ratio of the amplitude of the second peak divided by the amplitude of the first peak, period of the first and second peak in bpm, and the overall sum of the histogram.

The Pitch content feature set is computed from UPH (unfolded pitch histogram) and FPH (folded pitch histogram). It consists of 5 features: amplitude of the maximum peak of the FPH, period of the maximum peak of the UPH, period of the maximum peak of the FPH, pitch interval between the two most prominent peaks of the FPH, and the overall sum of the histogram.

For classification, they used Gaussian mixture model (GMM) classifiers with diagonal covariance matrices. To initialize the classifier, they used the K-means algorithm with multiple random starting points.

The training dataset they used consists of 20 musical genres and 3 speech genres, each having 100 representative excerpts. Each excerpt was 30 seconds long, so the training data was formed by 19 hours of an audio signal.

As a result, they claimed that the proposed automatic classification model's performance was comparable with human genre classification. The classifier using the proposed 30-dimensional feature vector was capable of correctly classifying 61% of the testing dataset, which consists of 10 musical genres.

Panagakis et al. [5] approached the experiment of classifying musical genres from a multilinear perspective. Features extracted such as multiscale spectro-temporal modulation and auditory cortical processing. They say that spectro-temporal modulation was used in various audio classification tasks but not in music genre classification back in these times. Each recording is represented by a third-order feature tensor generated by the auditory model. To handle large data sensors, they examined three multilinear subspace techniques. These are the Non-Negative Tensor Factorization (NTF), the High-Order Singular Value Decomposition (HOSVD), and the Multilinear Principal Component Analysis (MPCA). Used support vector machine for classification.

They used GTZAN and ISMIR 2004. For the GTZAN dataset, they managed to get 78.20% accuracy for Non-Negative Tensor Factorization, 77.90% for High-Order Singular Value Decomposition, and 75.01% for Multilinear Principal Component Analysis. On the other hand for the ISMIR 2004 dataset, the accuracy results are 80.47% for NTF, 80.95% for HOSVD, and 78.53% for MPCA. They also mentioned that for the ISMIR 2004 dataset, it is not possible to compare the results directly with other research experiments because of the quite different experimental settings [6,7,8]. It is also mentioned that each song belongs to only one genre class.

Bahuleyan [9] compares the performance of two classes of models wherein the Convolutional Neural Network (CNN) model is trained end-to-end, to predict the genre label of an audio signal, solely using its spectrogram. The second approach comes from time and frequency domain features. Logistic Regression (LR), Random Forest (RF), Gradient Boosting (XGB), and Support Vector Machines (SVM) were used for classification.

For the dataset, they extracted 10 second sound clips from a total of 2.1 million Youtube videos. When filtering applied to these videos, the dataset contains a total of 40540 samples of 7 categories.

When we check the results, they obtained 0.64 accuracy using Convolutional Neural Network with Fine Tuning and 0.59 accuracy using Extreme Gradient Boosting.

In a study done in Witwatersrand University [10], the categories of features utilized for automatic genre classification were presented and the researchers implemented an information gain ranking algorithm to determine the features most contributing to the classification. Their aim was to investigate the performance of machine learning models. They have identified four categories of features that are generally assumed to contribute to the classification. These categories are *magnitude-based, tempo-based, pitch-based,* and *chordal progression* features.

After utilizing the information gain ranking algorithm, they decided on the following features on their model: Chroma, root mean square, spectral centroid, spectral bandwidth, spectral roll-off, zero-crossing rate, MFCC, harmony, and tempo. They applied multiple learning algorithms using the Scikit library such as k-Nearest Neighbors, Multilayer Perceptron, Random Forests, Support Vector Machines, and Logistic Regression. After training several classifiers, k-Nearest Neighbors provided the best accuracy at 92.69%, furthermore, they observed that the kNN had a relatively low training time of 78 milliseconds.

2. FEATURE SELECTION

a. Overview

We've spent most of our time researching about the features that can be extracted from an audio signal and we saw that there are lots of sophisticated approaches. We decided to evaluate the effects of different features on different models. We started with learning the characteristics about the features. Essentially, features are extracted by applying different functions/aggregations over the waveform of an audio signal. Example waveforms of 2 genres can be examined in Figure 1 and Figure 2:

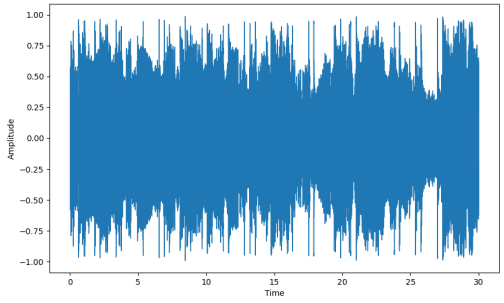


Figure 1: Waveform of a metal song - Apokalips by Danzig

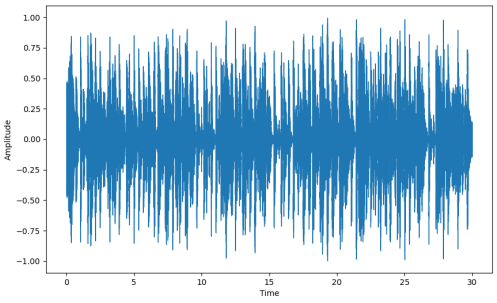


Figure 2: Waveform of a reggae song - Prophecy by Freddie McGregor

Another important tool for feature extraction is the fourier transform. It is the fundamental building block for various features. The fourier transform, applied to the waveform of an audio signal, converts the time domain signal to the frequency domain, and shows the distribution of contributions of different frequencies. Visualization can be seen in Figure 3 and Figure 4:

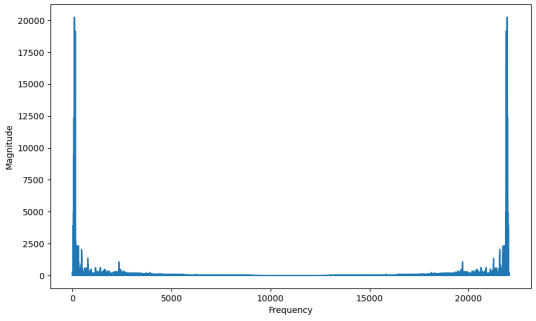


Figure 3: Fourier transform of a metal song - Apokalips by Danzig

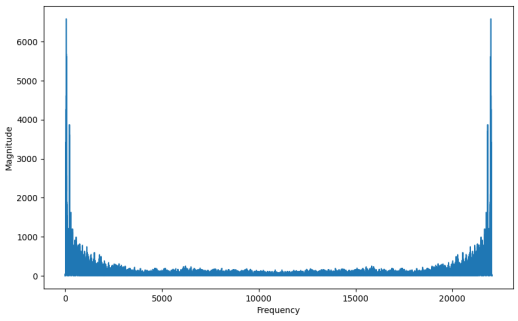


Figure 4: Fourier transform of a reggae song - Prophecy by Freddie McGregor

Short-time fourier transform (STFT) provides 3-dimensional data: time, frequency and magnitude. Each point in this space represents the magnitude of the frequency at the given time instant. Inspecting the magnitude of different frequencies as it varies over time, provides a deep insight for the classification of different songs. Spectrograms generated by applying STFT to the waveforms can be seen on Figure 5 and Figure 6:

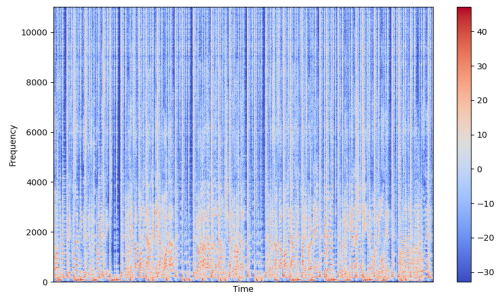
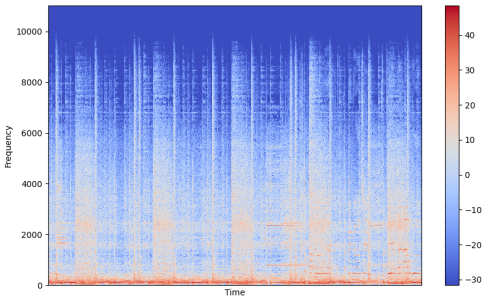


Figure 5: STFT of a metal song - Apokalips by Danzig Figure



6: STFT of a reggae song - Prophecy by Freddie McGregor

In the following section, we'll briefly go over the description of each of the features that we evaluated.

b. Time Domain Features

For the following features, we calculated them for each frame of the signal and calculated the mean and the variance of the resulting vectors, and put it in our feature set.

Root mean square (RMS): This feature approximates how we perceive sound. It can be defined as the square root of the mean square. For each frame of the time domain signal, RMS is calculated.

Zero crossing rate (ZCR): It is the rate at which a signal changes from positive to zero to negative or from negative to zero to positive. We computed the zero-crossing rate of the audio time series.

Harmonic and percussive components: These 2 features are extracted using harmonic/percussive source separation (HPSS).

Tempo: It is the beats per minute.

c. Frequency Domain Features

For the following features, we again calculated them for each frame of the signal and calculated the mean and the variance of the resulting vectors and put it in our feature set.

Spectral centroid: This feature is a good predictor of the "brightness" of a sound [1]. It can be defined as the center of gravity of the magnitude spectrum of the STFT (Figure 5 and 6).

Spectral bandwidth: Bandwidth is the difference between the upper and lower frequencies in a continuous band of frequencies.

Spectral rolloff: It is the frequency below which 85% of the magnitude distribution is concentrated [1].

Chromagram: It is the chromagram of the STFT spectrogram. It correlates to the musical aspect of harmony.

Mel Frequency Cepstral Coefficients (MFCC): MFCCs are very popular for audio processing applications. To obtain them, a discrete fourier transform should be applied to the waveform, then the log-amplitude spectrum should be evaluated. After, mel-scaling should be applied. And finally, a discrete cosine transform should be performed. A visualization of the coefficients can be seen on Figure 7 and Figure 8. Each row of the plot represents a different coefficient. There are up to 40 coefficients, and we've worked with the first 20 of them.

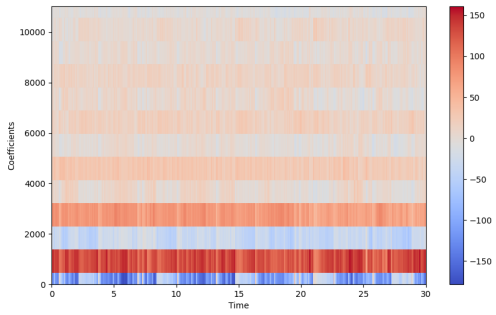


Figure 7: MFCCs of a metal song - Apokalips by Danzig

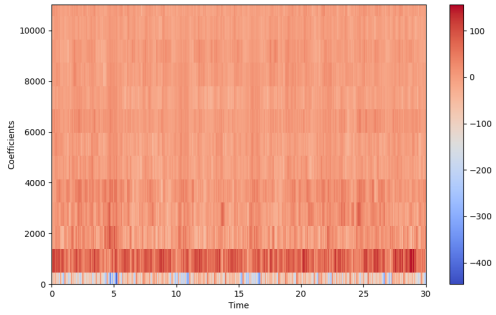


Figure 8: MFCCs of a reggae song - Prophecy by Freddie McGregor

3. DATASET PREPROCESSING

Our dataset has 1000 songs but with more songs, we decided that we can get better accuracy, so because of this we divided each song into 10 segments and our dataset grew 10 times. Of course, it is not as efficient as 10000 songs with 30 seconds long but it is an alternative way.

In the dataset, there is a file that contains different features. When we used the default features we got relatively poor results compared to our expectations. After that, we decided to extract the features manually and analyze them to get higher accuracy. We used a library called Librosa that is mainly used for audio analyzing and different machine learning approaches about sound. First, we extracted the features using Librosa and combined them in a feature vector. In the classifier algorithms, we used a different part of the vector.

4. MODEL SELECTION

Before building the models, we wanted to enlarge our dataset, so we divided each song in our dataset into 10 pieces, each piece 3 seconds long. We also introduced a mapping to describe each genre in a numerical form. We worked on five different classifiers on this project.

First, we built a Deep Neural Network model and created multiple hidden layers before the output layer. Our accuracy with this model was 81%. We achieved 80% accuracy with extreme gradient boosting, which was among the other classifiers we use. Another classifier was Convolutional Neural Networks and we used 13 MFCC as features. These classifiers we mentioned provided us decent accuracy values but we could not observe high accuracy values with the other two classifiers: k-Nearest Neighbors and Support Vector Classifier. Mostly, we focused on the Deep Neural Network classifier and came up with some features that give us the best accuracy.

```
Features: ['mfcc1_mean' 'mfcc1_var' 'mfcc2_mean' 'mfcc2_var' 'mfcc3_mean'
 'mfcc3_var' 'mfcc4_mean' 'mfcc4_var' 'mfcc5_mean' 'mfcc5_var'
 'mfcc6_mean' 'mfcc6_var' 'mfcc7_mean' 'mfcc7_var' 'mfcc8_mean'
 'mfcc8_var' 'mfcc9_mean' 'mfcc9_var' 'mfcc10_mean' 'mfcc10_var'
 'mfcc11_mean' 'mfcc11_var' 'mfcc12_mean' 'mfcc12_var' 'mfcc13_mean'
 'mfcc13_var' 'mfcc14_mean' 'mfcc14_var' 'mfcc15_mean' 'mfcc15_var'
 'mfcc16_mean' 'mfcc16_var' 'mfcc17_mean' 'mfcc17_var' 'mfcc18_mean'
 'mfcc18_var' 'mfcc19_mean' 'mfcc19_var' 'mfcc20_mean' 'mfcc20_var']

Epoch 199/200
219/219 [==============================] - 2s 11ms/step - loss: 0.0428 - accuracy:
0.9887 - val_loss: 1.4125 - val_accuracy: 0.8105
```

```
Features: ['chroma_stft_mean' 'chroma_stft_var' 'rms_var' 'spectral_bandwidth_mean'
 'harmony_mean' 'harmony_var' 'perceptr_mean' 'perceptr_var' 'tempo'
 'mfcc1_mean' 'mfcc1_var' 'mfcc2_mean' 'mfcc2_var' 'mfcc3_mean'
 'mfcc3_var' 'mfcc4_mean' 'mfcc4_var' 'mfcc5_mean' 'mfcc5_var'
 'mfcc6_mean' 'mfcc6_var' 'mfcc7_mean' 'mfcc7_var' 'mfcc8_mean'
 'mfcc8_var' 'mfcc9_mean' 'mfcc9_var' 'mfcc10_mean' 'mfcc10_var'
 'mfcc11_mean' 'mfcc11_var' 'mfcc12_mean' 'mfcc12_var' 'mfcc13_mean'
 'mfcc13_var' 'mfcc18_mean']

Epoch 200/200
219/219 [==============================] - 3s 12ms/step - loss: 0.0546 - accuracy:
0.9861 - val_loss: 1.2592 - val_accuracy: 0.8048
```

```
Features: ['mfcc1_mean' 'mfcc1_var' 'mfcc2_mean' 'mfcc2_var' 'mfcc3_mean'
 'mfcc3_var' 'mfcc4_mean' 'mfcc4_var' 'mfcc5_mean' 'mfcc5_var'
 'mfcc6_mean' 'mfcc6_var' 'mfcc7_mean' 'mfcc7_var' 'mfcc8_mean'
 'mfcc8_var' 'mfcc9_mean' 'mfcc9_var' 'mfcc10_mean' 'mfcc10_var'
 'mfcc11_mean' 'mfcc11_var' 'mfcc12_mean' 'mfcc12_var' 'mfcc13_mean'
 'mfcc13_var']

Epoch 233/599
219/219 [==============================] - 2s 11ms/step - loss: 0.0241 - accuracy:
0.9921 - val_loss: 1.6791 - val_accuracy: 0.8048
```

5. HYPERPARAMETER TUNING

In all of our models, we used 70% of our dataset for training and 30% of it for testing.

In the Deep Neural Network model, we used seven hidden layers in total. They had 4096, 2048, 1024, 512, 256, 128 and 64 nodes respectively. We set the learning rate to 0.001 and the batch size to 32.

In the eXtreme Gradient Boosting model, we set the learning rate to 0.05 and number of gradient boosted trees to 1000.

In the k-Nearest Neighbors model, we set the number of neighbors to 19.

6. TESTING

For the testing part, as we mentioned before we downloaded different types of songs from the internet and tested them in our trained model. There are 10 different genres in the dataset and because there are many features, some genres are easy to guess and some genres are not based on the complexity of the song. Here are the results for different songs from different genres:

Dua Lipa - Break My Heart (Pop):

```
File: test/dua-lipa-break-my-heart.wav
Label: pop

blues: 9.550509408661419e-28
classical: 6.282863498126984e-15
country: 3.785220292588931e-13
disco: 2.2963915213836117e-09
hiphop: 8.259573860414093e-07
jazz: 1.7682966202414718e-10
metal: 1.1315576669841242e-14
pop: 0.9999991655349731
reggae: 7.02292607748789e-12
rock: 4.678670645352767e-13
```

When we check the results, our model predicts the genre correctly with an almost 100% rate.

Beethoven - Für Elise (Classical):

```
File: test/beethoven-für-elise-piano.wav
Label: classical

blues: 8.74967470438498e-13
classical: 1.0
country: 2.42643175652835e-16
disco: 1.293229941834798e-21
hiphop: 5.431598774268201e-18
jazz: 2.2149417855388265e-09
metal: 2.6972625214319428e-33
pop: 2.2554446781196973e-23
reggae: 0.0
rock: 1.2172435368690282e-13
```

For Beethoven, it gives the result of 100% classical and because the others are almost 0 the computation ignores these small numbers.

Dr Dre - The Next Episode (Hiphop)

```
File: test/dr-dre-the-next-episode.wav
Label: rock

blues: 0.07183708995580673
classical: 3.186172398272902e-05
country: 0.05722988024353981
disco: 0.19166478514671326
hiphop: 0.18449872732162476
jazz: 0.0002110584027832374
metal: 0.014557148329913616
pop: 0.040530718863010406
reggae: 0.09866613894701004
rock: 0.3407725393772125
```

In the results, it is %34 rock, 19% disco, 18% hip hop and it goes like this for the other genres. Our model predicted this song wrong.

Queen - Bohemian Rhapsody (Rock):

```
File: test/queen-bohemian-rhapsody.wav
Label: reggae

blues: 5.229822636465542e-05
classical: 2.7146652428200 47e-05
country: 0.0006789013859815896
disco: 0.007842855527997017
hiphop: 0.4065662920475006
jazz: 0.0004430591652635485
metal: 6.690232112305239e-05
pop: 0.02386755496263504
reggae: 0.5602481961250305
rock: 0.00020689750090241432
```

Our model labeled this song as reggae 56% and after that hiphop comes with 40%. Again it was predicted wrong.

During our tests, we saw that when the song contains different vibes from different genres, it is hard to get correct results. Since the machine learning model uses the mathematical part of the songs, it detects many different features in one song and gives the outputs according to that.

SUMMARY

In this project, we had the chance to put our theoretical knowledge into practice. We've worked with different machine learning libraries and used different models. We've learned how an audio signal can be processed in different ways to extract various features that define its characteristics. The limited computational power and the small sized dataset resulted in the misclassification of new songs. However, we achieved an accuracy of 0.81 in the GTZAN dataset.

Possible future work for us can be to use different spectrograms together to increase the depth level in CNN to achieve better results.

REFERENCES

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, July 2002, doi: 10.1109/TSA.2002.800560.

[2] Cano P, Gómez E, Gouyon F, Herrera P, Koppenberger M, Ong B, Serra X, Streich S, Wack N. ISMIR 2004 audio description contest. Barcelona: Universitat Pompeu Fabra, Music technology Group; 2006. 20 p. Report No.: MTG-TR-2006-02

[3] Aucouturier, Jean-Julien, and Francois Pachet. "Representing musical genre: A state of the art." Journal of new music research 32.1 (2003): 83-93.

[4] Pachet, François, and Daniel Cazaly. "A taxonomy of musical genres." RIAO. 2000.

[5] Panagakis, I., Benetos, E. and Kotropoulos, C. (2008). Music genre classification: a multilinear approach. Paper presented at the International Symposium Music Information Retrieval, 14 - 18 September 2008, Philadelphia, USA.

[6] Holzapfel, A. and Stylianou Y. “Musical genre classification using nonnegative matrix factorization-based features”, IEEE Transactions on Audio, Speech, and Language Processing, Vol.16, No. 2, pp. 424-434, 2008.

[7] Lidy, T. and Rauber, A. “Evaluation of feature extractors and psycho-acoustic transformations for music genre classification”, Proceedings of the Sixth International Symposium on Music Information Retrieval, London, UK, 2005.

[8] Pampalk, E., Flexer, A. and Widmer, G. “Improvements of audio-based music similarity and genre classification”, Proceedings of the Sixth International Symposium on Music Information Retrieval, London, UK, 2005.

[9] Bahuleyan, H. (2018). Music genre classification using machine learning techniques. arXiv preprint arXiv:1804.01149.

[10] Ndou, Ndiatenda, Ritesh Ajoodha, and Ashwini Jadhav. "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches." 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE, 2021.