TROUVAILLE
BBM406 FINAL PROJECT REPORT

Ecem Varma Pelin Azizoğlu Tunahan Pınar [1]

Abstract

Music genre is a way to specify music into categories by humans. Machine learning takes it data from knowledge or facts that people have identified. But classifying music into genres is a hard task even for people because there are no certain ground rules. And a music usually does not have a single genre which makes the problem even harder. Deep learning is a trending field with usages in daily life, this paper provides a comparative study on classifying music genres using deep neural network approach. Feature selection is used, and spectrograms are generated. This music genre classifier uses one of the most used datasets for this problem, GTZAN. The result of our classifier in test data is nearly 91%. Index Terms—music genre classification, feature extraction, deep-learning, machine learning, spectrograms, Deep Neural Network

1. Introduction

Music has been present in human lives since the very early ages of humankind in every society and tradition. But of course, in every tradition and society the form of music shows uniqueness. Africa is the place where music was first born and since then it has a special place in our lives. [1] First form of music was created by simply clapping hands. After that, with the human evolution, sticks and stones replaced hands.[2]

[3] Stone wall with ancient musicians

In 1920s popularity of mainstream started to increase which led to radio stations and record companies starting to market genres to people.[4] Music genres are labels created by people to identify different types of music. Music genres don't have any certain rules because music composition is a natural process that occurs from interaction between humans and humans, nature, and historical events. Music genre classification has been a popular topic with many applications in real-world. Usually music genre classifiers use pattern recognition algorithms to classify short interval of audio records. In Spotify which is one of the most used music streaming applications globally, there exists more than 1,300 music genres.[5] Instrumentation, rhythm, and pitch content of the music are means to differentiate genres despite of the huge numbers of genres. Genre classification can be performed by programs instead of human experts. This will be led to automatic process which will gain time and energy. It also helps to provide a framework and assess features for categorizing music. In this paper, the problem of classifying music by genres is explained. To give more details, fifty-two features are used as: Chroma STFT, Root Mean Square Error, Spectral Centroid.

We found the idea of a music classier exciting because it has many real-world applications that we use in daily life. Lots of similar work has been done and is being done because music is everywhere and reachable to anyone.

The topics that this paper contains is as follows. Some related work to our topic music genre classifier is given in Section 2, our approach to this classification problem is given in Section 3, the experimental results that we obtained are presented in Section 4, and our conclusions are given in Section 5.

2. Related Work

Automatic audio analysis systems depend on the extraction of feature vectors. A large number of different feature sets exist to represent audio signals. Generally, these features are based on some form of frequency-time representation. Although a full overview of sound features cannot be covered in this study, many quality sound feature extraction references are given.

G.Tzanetakis and C Perry, who both created this data set and made one of the oldest studies in the field of music

genre classification. G. Tzanetakis and C Perry explore the classification of audio signals into musical genre hierarchies to label and categorize musical genres based on similar characteristics. To achieve this goal, they proposed 3 different feature sets for representing rhythmic, timbral texture, and pitch. They used the k-Nearest Neighbor and Gaussian Mixture model to learn and model this classification. With this dataset they created and the model they used, they achieved an accuracy rate of around 61 percent.

In addition to this basic work, if we look at the present day, we can see how much work has been done. One of them, close to the present, done by R. Ajoodha, R. Klein, and B. Rosman. They uses the GTZAN dataset for this classification, just like the previous example. In addition, they uses four different feature sets, namely magnitude spectrum, pitch, tempo and chord features. The best and worst features of these features are determined by using knowledge acquisition rankings according to their contribution to the learning process. This study uses 6 of the standard machine learning classification algorithms, namely Multilayer Perceptron, Logistic Regression, Support Vector Machines, Random Forests, k-Nearest Neighbors and Naive Bayes. These classifications were compared among each other, and among them, the logistic regression model achieved the best accuracy of 81 percent.

The research conducted by H. Bahuleyan, which is one of the examples close to today, is also a very good source on this subject. The biggest feature that distinguishes it from the other examples we mentioned is that it uses Convolutional Neural Networks (CNN) instead of standard machine learning algorithms. H. Bahuleyan used MEL spectrograms as input for CNN. Using the Audio Set dataset, the CNN which uses VGG-16 CNN + Extreme Gradient Boosting, reached performance accuracy up to 64 percent. Since this study by H. Bahuleyan uses the ”Audio Set” data set, which is different from the others, we cannot make a true comparison.

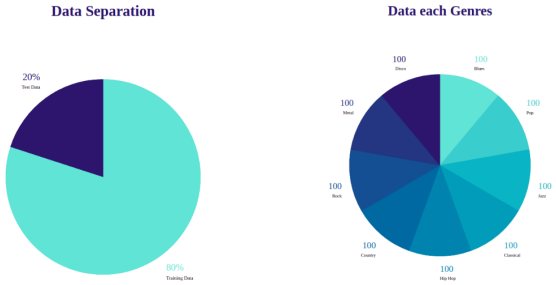
In addition, both H. Bahuleyan’s study and R. Ajoodha, R. Klein, and B. Rosman’s study, characterizes Mel-frequency cepstral coefficients (MFCC) as one of the features that contribute the most to music genre classification.

3. The Approach

In this experiment, many different types of classification algorithms were used to find the algorithm that best fits our data and produces the best results. Many different combinations of experiments have been performed with data from related works, our lectures, and research. Due to the number of features and classification algorithms, many comparison tables and graphs were obtained. As a result of long efforts, we achieved the best result with DNN. The DNN model was created with 5 layers (1 input layer, 3 hidden layers, 1 output layer). It achieved results with an accuracy of 91 percent.

3.1. Dataset

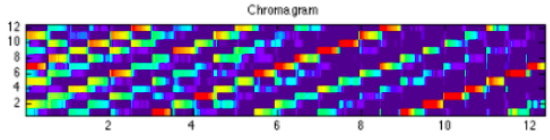
The GTZAN dataset consists of 1000 audio files that include 100 audio files for each 10 music genres. Genres’ names are Blues, Pop, Jazz, Classical, Hip Hop, Country, Rock, Metal, and Disco. In this data set, each audio file is 30 seconds long. The data set exists on Kaggle and it is one of the most popular datasets for music genre recognition(MGR). It has been used in many published articles. In our experiment, the data set split into 80 percent - 20 percent, training data, and test data respectively. The training data was further split into 5-folds for cross-validation purposes to obtain more consistent data. Also, each music file in the dataset was divided into 10 parts. The new dataset consists of 3-second long audio and we had 10.000 audio files with 10 different genres(each of them has 1000 audio data). Actually, it worked surprisingly well.



3.2. Features

The dataset includes the raw audio files and these audio files contain many different features. These feature names are ”chroma short-time Fourier transform”, ”root means square error”, ”spectral centroid”, ”spectral bandwidth”, ”spectral roll-off”, ”zero-crossing rate”, and ”MFCC”. For our experiment, mean and variance values are calculated for each feature. Also, MFCC include 20 different MFCC variant. So, in the end, the output of the feature extraction process gives us 52 different types of features.

Chroma STFT: This feature is an interesting and powerful representation for music audios. It corresponding to the total energy of the signal for each of the 12 bins representing the 12 distinct semitones (or chroma) of the musical octave.



Root Mean Square Error(RMSE): It equals the frequency in a discrete-time signal after taking the root mean square of it.

Spectral Rolloff: The spectral roll-off, basically defined as the frequency where the frequency below the frequency Rt which 85 percent of the total signal energy is contained under. Actually, it is a measure of the shape of the signal.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n].$$

Zero-Crossing Rate: The zero-crossing rate is the rate of sign-changes in which discrete-time signal changes sign from positive to negative or back along with a signal. It heavily using in music information retrieval, and also speech recognition.

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])|$$

Spectral Bandwidth: The spectral bandwidth is defined as the bandwidth of an audio wave at one-half the peak maximum.

Spectral Centroid: The spectral centroid is defined as the center of gravity for our music file which the most energy is centered around for a given frame. It indicates where the center of mass for our music data is located and it is calculated as the weighted mean of the frequencies present in the sound.

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}$$

Mt[n] where is the magnitude of the Fourier transform at frame and frequency bin n.

Mel-Frequency Cepstral Coefficients (MFCC): Mel-frequency Cepstrum (MFC) is based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. So, it is a representation of the short-term power spectrum of a sound. MFCC of a signal is a small set of features that collectively make up an MFC. These small sets give us concisely describe the overall shape of a spectral envelope.

3.3. Classification Models

Logistic Regression (LR): Although Logistic Regression is generally used for binary classification, it can also be used for multiple classifications with the one versus rest method. It trains independent classifiers for each music genre and treats the classifier with the highest probability among them as the final selected genre label. In our result, LR achieved an accuracy of 69 percent with the GTZAN dataset.

K-Nearest Neighbours (KNN): The k-nearest neighbors (KNN) algorithm is a simple algorithm and it is easy to implement. It can classify songs based on other K songs of most similar. K determines the number of neighboring songs are used to determine the label of selected data. K should be chosen as an odd number to avoid ties between classes so we always do that. In this method, we tried many different k numbers and also we tried to weighted K-NN. By doing this, the 30-second dataset and the 3-second dataset tried separately and observed differences between them. We tried with 30-sec data and the best result was 68 percent accuracy. We got our best result with the 3-sec dataset and the test accuracy is 90 percent. Both of them achieved with the K=3.

Support Vector Machines (SVM): SVM is a supervised machine learning algorithm. It transforms the given input into a high-dimensional space using a kernel. Multi-layer SVM classifiers try to find an optimal boundary between the genre labels. In this method, we tried different types of kernels with SVM. These kernels are ”linear”, ”rbf”, ”poly”, ”sigmoid”. We got our best results with rbf kernel. Results are 83.63 percent and 68.0 percent accuracies with 3sec dataset and 30-sec dataset respectively.

Random Forest (RF): RFs is an ensemble learning method for classification, regression, and other tasks that generates a large number of decision trees at training time and then takes the mode of classes or the class output, which is the average estimate of the individual trees. It trains each decision tree using a subset of the training data and makes a prediction using only a random subset of features. Random forest is based on trees, so we tried a different number of tree parameters. Best 30 sec dataset result is 69.5 percent accuracy, number of estimators = 600. Best 3 sec dataset result is 84.13 percent, number of estimators = 600.

Deep Neural Network (DNN): A deep neural network (DNN) is a kind of machine learning method that consists of multiple artificial neural networks (ANN) between the input and output layers. There are many different types of neural networks, but the components are generally the same. These neural networks consist of neurons, synapses, weights, biases, and functions. These components work similarly to the human brain and can be trained like other machine learning algorithms. In our experiment, we try many different layers and functions. For DNN, there are many different kinds of parameters like relu, sigmoid, tanh, etc. Also, DNNs are tunable with different sizes of layers and different numbers of layers. It was a real challenge to choose which one is best. We got our best result with DNN and it was 91 percent test accuracy.

DNN Approach Detailed: Our deep neural network

architecture was built using Keras models and consists of the input layer followed by 5 different layers. This model used adam as an optimizer, used sparse categorical cross-entropy as loss function, and used accuracy as a metric.

Input layer: Relu used as an activation function. Size = 512.
First hidden layer: Relu used as an activation function. Size = 256.
Second hidden layer: Relu used as an activation function. Size = 128.
Third hidden layer: Relu used as an activation function. Size = 64.
Output layer: Softmax used as an activation function. Size = 10 due to a number of genres.

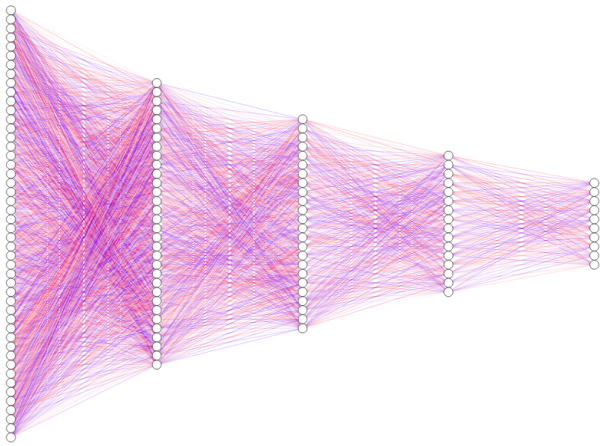


Figure 1. DNN with 512x256x128x64x32x10

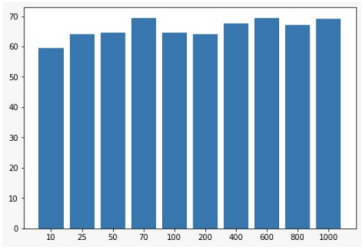
As a result of the traditional machine learning approaches above, the 3-sec dataset gives enormous improvement to our test results. Relu worked great as an activation function in each layer. Also, we can say that more number hidden layers give us better accuracy.

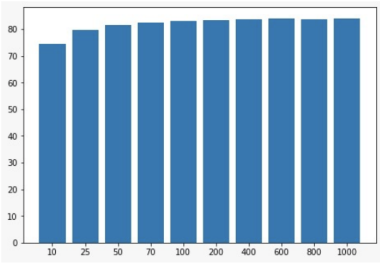
4. Experimental Result

Dataset Description: The dataset includes 10 genres with 100 audio files, each of them having a length of 30 seconds. Firstly, we converted label categories into numerical values that "0, 1, 2, 3, 4, 5, 6, 7, 8, 9". After this process, the dataset is splitted into train (80%) and test (20%) sets. So we come up with a decision to have a bigger dataset by rearranging our dataset by converting 30-second songs into 10, 3-second songs. And it was the right step for increasing accuracy.

4.1. Random Forest

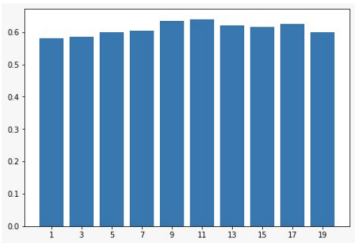
One of the machine learning methods is Random Forest which builds multiple decision trees and combines them to get a more stable and proper prediction. Random forest algorithm has a lot of advantages; it reduces overfitting, gives higher accuracy than one decision tree, and runs efficiently on large datasets. In our experiments, we tried a different number of trees(10 to 1000) in the forest and plotted. In the first graph is audio files that have a length of 30 seconds, the second one is 3 seconds. Our best result in 3 seconds length audio files with 84% accuracy.

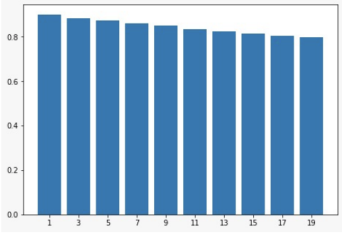




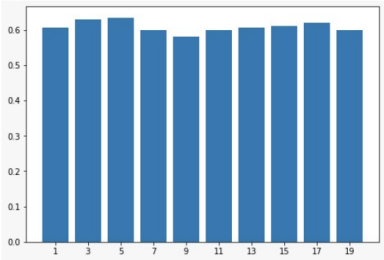
4.2. K-nearest Neighbors Algorithm

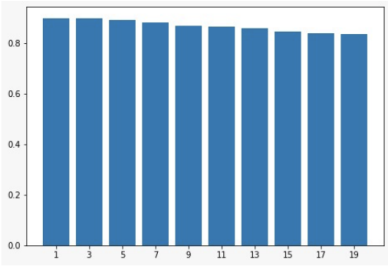
K-Nearest Neighbor classifier is a machine learning method that we applied and it resulted in 88% accuracy. In the first graphs are audio files that have a length of 30 seconds, the second ones are 3 seconds. As we see, 3 seconds length audio files have better accuracy for both of them. In weighted k-nearest neighbor, closer neighbors have a greater influence than neighbors who are further away. And generally weighted k-nn has better accuracy.





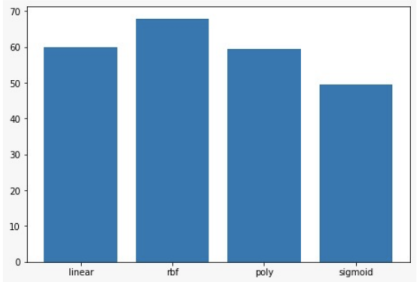
Weighted K-nearest Neighbors Algorithm:

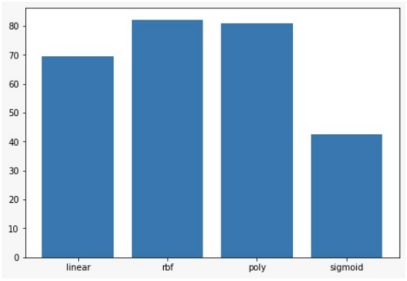




4.3. Support Vector Machine

Support Vector Machine uses different mathematical functions defined as the kernel. The kernel function is to take data as input and change it to the required form. We used different types of kernel functions: linear, radial basis function, polynomial, and sigmoid. In the left graph is audio files that have a length of 30 seconds, the right one is 3 seconds. As we see, 3 seconds length audio files have better accuracy for all function types, on the other hand, radial basis function got the best accuracy level for both of them.



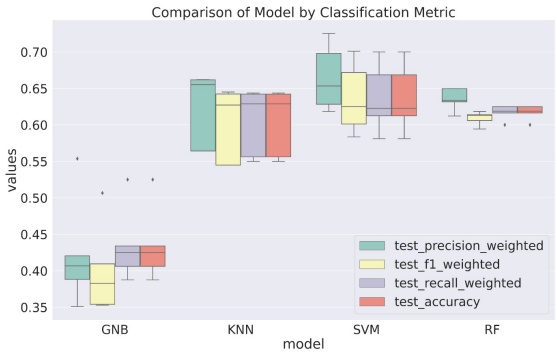


4.4. Naive Bayes

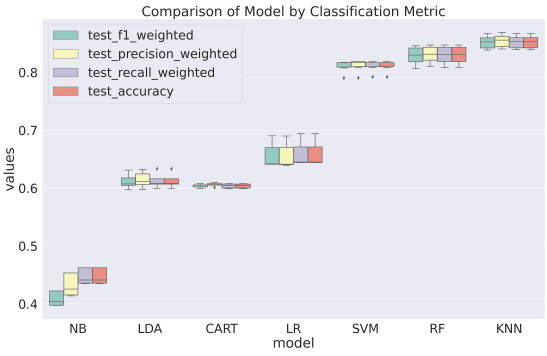
We also applied Gaussian Naive Bayes Classifier. Naive Bayes assumes that the features are independent during this if this assumption is not correct, the probabilities are incorrect too. It can be the reason for our low outcome because this approach was not successful and resulted in 44% accuracy.

Overview of algorithms and their accuracies that we used before decided on neural network.

30-seconds audio files:

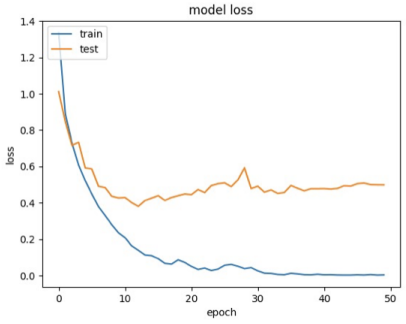


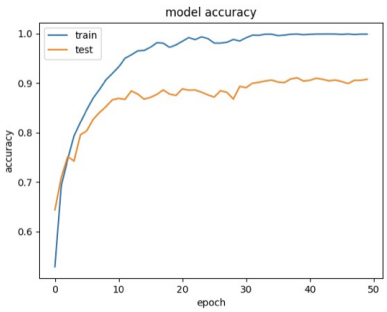
3-seconds audio files:



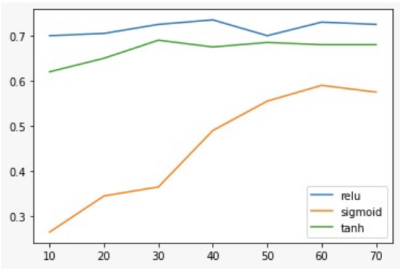
4.5. Neural Network

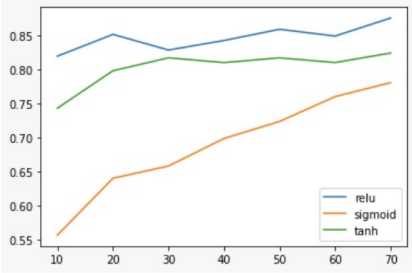
In a neural network, there are different activation functions used to the weighted sum of the input is transformed into an output, different loss functions used to calculate the gradients, and different optimizers used to solve optimization problems by minimizing the function and all of them affect accuracy. On the other hand, if a model cannot learn enough with a training dataset, increasing accuracy is impossible. To make an efficient learning process, we used 10000 audios of 3 seconds dataset in our neural network. We had 52 different features for classification. We decided to use pca, it affects models to become more efficient as the reduced feature set boosts learning rates and diminishes computation costs by removing redundant features. We used pca and we had 11 features that are affected mostly the label. Then, we started training with a neural network model. There added one more layer in our network, so we have 5 layers in our network. The first one is an input layer. Then there are 3 hidden layers, and the last layer is an output layer. The output layer has 10 neurons for 10 genres. We used Relu as an activation function, we also tried sigmoid and tanh activation functions but we got the best accuracy with Relu. It can be about not vanishing gradient. We also tried different optimization algorithms such as Adam, Adagrad, sgd, and RMSProp and in the end, we used the Adam algorithm which is largely used in deep learning and works well with large datasets. Our model's goal is to reduce sparse categorical cross-entropy loss function. We used the fit function with 30 epochs and 128 batch sizes. And with these properties, accuracy is equal to 91%.
In the following example, in the left graph are audio files that have a length of 30 seconds, the right one are 3 seconds. The loss function and accuracy of an architecture trained on our dataset. The loss function and accuracy of the model are across training epochs.



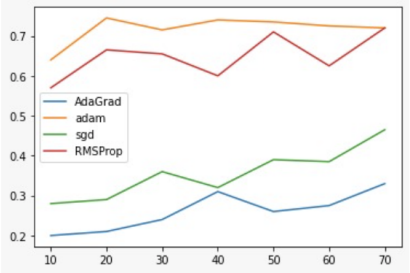


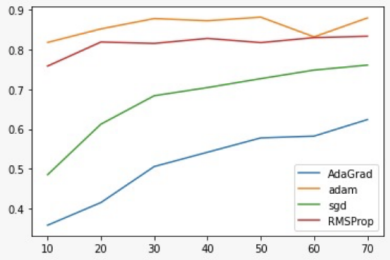
In the following example, in the left graph are audio files that have a length of 30 seconds, the right one are 3 seconds. The activation function of an architecture trained on our dataset. The activation function of the model are across training epochs.





In the following example, in the left graph are audio files that have a length of 30 seconds, the right one are 3 seconds. The optimization function of an architecture trained on our dataset. The optimization function of the model are across training epochs.





5. Conclusion

Music genre classification will continue to have a major role in the music industry since it is a huge industry still growing. People's interest in music and different genres is increasing day by day. And this subject turns to valuable gradually. And that affects us to choose this subject.

We began our project with around 60% percent and took it to a 90% accuracy level. We tried different methods, and we choose one of them to develop. Increasing accuracy is turned to our aim. Looking into producing more training data using existing training data by further cutting audio samples into smaller samples resulting in more samples. For that, we changed 30 seconds audios into 10, 3-seconds audio is a good start. After that, using pca and increasing the number of hidden layers also improved accuracy. We used Relu as an activation function, we also tried sigmoid and tanh activation functions but we got the best accuracy with Relu. We also tried different optimization algorithms such as Adam, Adagrad, sgd, and RMSProp and in the end, we decided to use the Adam algorithm.

Even though music genre classification is not specific it can be performed automatically with better results than luck, and performance than humans. Using the features specified we have achieved 90% accuracy in a data-set which have ten musical genres of thirty seconds. Our success with usage of features specified for musical genre classification shows that there is a potential for automatic classification of music using similarity and extracting features to categorize audio tracks. A probable future direction research topic is enlarging the genres and semantic descriptions such as emotion.

References

[1] https://en.wikipedia.org/wiki/history_of_music, a.

[2] https://www.sciencedaily.com/releases/2017/06/170620093153.htm: :text=our%20early%20ancestors%20may%20have,hurt%20your%20hands%20as%20much.text=so%2c%20we%20know%20that%20music,from%20when%20humans%20first%20evolved., b.

[3] featured image credit: Stone wall with ancient musicians, by repina valeriya © via shutterstock.., c.

[4]https://www.telegraph.co.uk/travel/discover-america/evolution-of-music/: :text=its%20mainstream%20popularity%20began%20in,the%20genre%20to%20the%20masses.text=i%20n%20the%201940s%2c%20afro,many%20subsidiary%20genres%20were%20born. d.

[5]https://askwonder.com/research/music-genres-there-ovchwa91d: :text=according%20to%20the%20popular%20music,music%20genres%20in%20the%20world., e.

H. bahuleyan. "music genre classification using machine learning techniques." arxiv preprint arxiv:1804.01149 (2018)., f.

R. Ajoodha, R. K. and Rosman, B. Single-labelled music genre classification using content-based features. *In 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pp. 66–71, 2015.

Tzanetakis, G. and .Perry, C. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing 10, no.5*, pp. 293–302, 2002.