Music Genre Classification

Yazid MOULINE, Guillaume REQUENA

Abstract

Musical genres are categorical labels created by humans to characterize pieces of music. A musical genre is a conventional category that identifies pieces of music as belonging to a shared tradition or set of conventions. These characteristics, that classify music by genre, typically are related to the instrumentation, rhythmic structure, and harmonic content of the music. Genre hierarchies are commonly used to structure the large collections of music available on the Web. Nowadays, it is widely spread, used by music streaming platforms such as Spotify, Deezer or Youtube Music, to automatically classify their "discotheque" and to offer a wide range of services, using classification and clustering algorithms.
In this paper, the automatic classification of song extracts into musical genres is explored by using machine learning classification algorithms (logistic regression, a neural network, support vector machines).

Keywords: Musical genre classification, Feature extraction, Machine Learning
Link of the GitHub: https://github.com/yazidmouline/MusicGenreClassification

1. Introduction

Music streaming platforms such as Spotify or Deezer have totally modified the way we listen to music today. Instead of always listening to albums we tend to listen to playlists of songs grouped by genre. Also, these applications allow us to discover many songs easily. Indeed, these platforms have developed techniques (Flow for Deezer and Mix for Spotify) that suggest songs or tracks that are similar to one song or an artist etc. Of course, these softwares suggest similar songs, meaning that they go further than classifying music only by genre.

We implemented a Logistic Regression algorithm and a basic feed-forward neural network. The input of our programs is a dataset (a csv file) that we constituted by extracting musical features with the LibROSA library from a collection of wav files. The constitution of our dataset and feature extraction is described in the third part of this report. And the Machine Learning algorithms implementation is described in the fourth part of this report.

2. Related work

Machine Learning has a tremendous number of applications in the musical world. Generating music, analyzing similarities between compositions, analyzing musical characteristics and many other. Classifying music by genre is one of these applications that has now been studied for decades.

In 2002, G. Tzanetakis and P. Cook [1] used a mixture of a Gaussian model and kNN classifier with only three features very well chosen. They used the same dataset that us, the GTZAN Genre Collection[2].

A very interesting web article on the Medium section "towards data science" [3] inspired us for the feature extraction using the LibROSA library. They used also a CNN algorithm to analyse directly the spectrograms and not features reduced to real numbers, a potential extension to our work.

A project from a group of Stanford students on Music Genre Classification. They used kNN algorithm, Support Vector Machines, a basic feeed-forward neural network and a convolutional neural network[4].

3. Dataset and feature extraction

3.1 Dataset structure

In order to create our dataset we used the well-known GTZAN genre collection database. It is a zip file composed of 1000 audio tracks of 30 seconds long, divided into 10 different genres (100 tracks per genre). We have blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock.

Human beings (or musicians at least) easily differentiate a genre from another by ear, no one would confuse classical music with metal for example. We tried to look for computable features that would differentiate one music genre from another. We used LibROSA, a Python package for music and audio analysis. By computing some characteristics we realised that they can have some tendencies according to the genre, we thus constituted our dataset with these features.

The general idea is to constitute our dataset with these features for every song we have in our database. We constituted a csv file (Music_data_set.csv) that contains as features mean_mfccs, mean_chroma_stft, tempo, pulse, flatness, contrast, zero_crossing.

We encourage you to have a look at the explanations of these features and the reason why we choose them in the next subsection and in the iPython notebook (Dataset.ipynb) where we plotted the different spectrograms for songs from two different genres. The features we extracted represent timbral texture, rhythmic and pitch content. A profound musical analysis and feature extractions might give us more interesting features. Also, we decided to compute the mean of many spectrograms, which is a projection and thus a loss of a lot of information about the extract. An image analysis of these spectrogram might give us better results.

3.2 Features extraction

3.2.1 MFCCs

We compute the Mel-frequency cepstral coefficients (MFCCs). These coefficients are widely known to be used for speech recognition or anything including a human voice and also in music information retrieval (genre recognition for instance). We thought that it would be interesting to compute the mean over a song to make it a feature. The plots below (on the same scale) show how this feature can be very different for two genres.

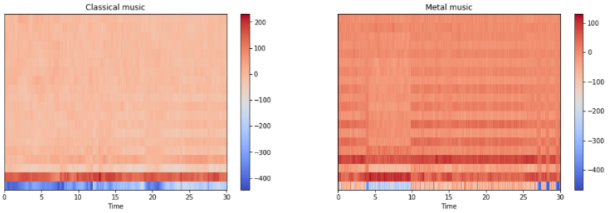


Figure 1 - MFCC comparison (classical/metal)

3.2.2 Chroma stft

Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto the 12 distinct semitones (or chroma) of the musical octave. Since, in music, notes exactly one octave apart are perceived as particularly similar, knowing the distribution of chroma even without the absolute frequency can give useful musical information about the audio. We also decided to compute the mean over a song, as a characteristic of the "purity" of the instruments. Classical music for example will have less "noise" than metal music (drums and saturated guitars).

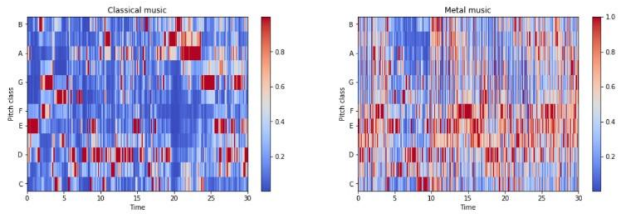


Figure 2 - Chroma stft comparison (classical/metal)

3.2.3 Tempo

The tempo can also be a characteristic of the music genre. However, two totally different songs can have the same tempo, but a genre tends to have a certain range of tempos for its songs. We decided to make it a feature of our dataset.

3.2.4 Pulse

This is also a rhythmic feature that can characterize the presence or not of drums in a song. It is then interesting to compute and put it as a feature in our dataset to differentiate genres that would have more or less present drums.

3.2.5 Spectral flatness

This feature allows also to analyze the purity of the sound. A coefficient equal to zero is a pure note and a coefficient equal to one is white noise, which explains these two spectra. Classical music has a lot more "pure" notes than metal music which has a lot of distorted instruments and drums. We decided to compute the mean as well.

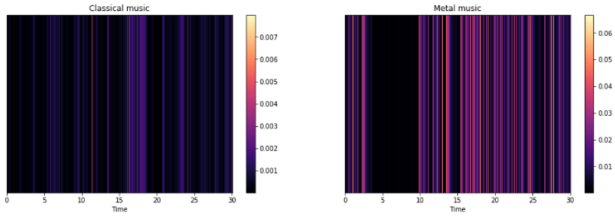


Figure 3 - Spectral flatness comparison (classical/metal)

3.2.6 Spectral contrast

This feature allows us to study the noisiness of a song, a low contrast might indicate a lot of noise. A contrast equal to zero is that of a white noise, and a very high contrast gets closer to a pure sound.

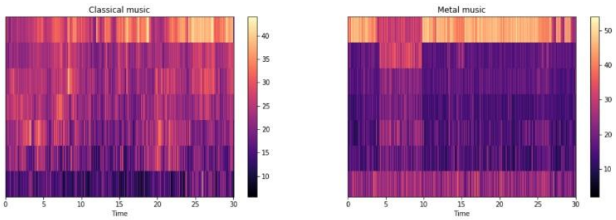


Figure 4 - Spectral contrast comparison (classical/metal)

3.2.7 Zero-crossings

Finally, the sum of the zero-crossings over the extract. This feature might not be very relevant, it might indicate the presence of many different instruments.

3.3 Rescaling

Since the range of values of raw data varies widely, we decided to standardize our features at first only for few methods that required it such as Random Forest or SVM classifier methods. We did by using a method from sklearn which is called StandardScaler. It consists in removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$$\text{New feature} = \frac{(x-\mu)}{\sigma}$$

4. Classification methods and machine learning algorithms

We decided to implement machine learning models to classify music by genre. A logistic regression model with a linear solver. We applied it with different number of genres and different genres and the results were pretty convincing. Then a random forest classifier trying as well with different number of genres. A basic feed-forward network, trying with a different number of layers, and same as for before, different number of genres and different genres, meaning different outputs.

4.1 Logistic Regression

We decided to use the simplest classification algorithm, a logistic regression, with a linear solver. We start by importing our dataset, shuffling it, rescaling the features with sklearn StandardScaler and splitting into a training and testing set. We chose to classify first genres two by two. We ran the program with all the possible inputs according to our dataset (10 genres).

This table show all the accuracies obtained with all the different tuples possible. The test accuracies are generally high, ranging between 0.65 and 1.0 with an average of 0.88. We can notice however that some genres tend to differ more easily to others, for example classical music always obtained very high accuracies. On the other side, other genres have difficulties differencing from, the others, for instance rock music. This encouraged us consider the case "one versus all" that we implemented for the neural network.

accura	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	1	0,95	0,7	0,825	0,95	0,85	0,925	1	0,85	0,725
classical	0,95	1	0,95	0,925	0,975	0,9	1	0,975	0,95	0,95
country	0,7	0,95	1	0,8	0,925	0,775	0,95	0,875	0,65	0,75
disco	0,825	0,925	0,8	1	0,925	0,925	0,9	0,85	0,8	0,7
hiphop	0,95	0,975	0,925	0,925	1	0,95	0,875	0,875	0,8	0,85
jazz	0,85	0,9	0,775	0,925	0,95	1	1	0,9	0,9	0,825
metal	0,925	1	0,95	0,9	0,875	1	1	0,95	1	0,825
pop	1	0,975	0,875	0,85	0,875	0,9	0,95	1	0,875	0,825
reggae	0,85	0,95	0,65	0,8	0,8	0,9	1	0,875	1	0,8
rock	0,725	0,95	0,75	0,7	0,85	0,825	0,825	0,825	0,8	1

Figure 5 - Accuracies of the comparison of two genres using Logistic Regression

Then we applied the logistic regression for more than two genres. For example, classifying four genres (classical, metal, blues, country) we get a test accuracy of 72.5%.

Finally, trying to classify ten genres gets difficult for the logistic regression, with our dataset. However, we still obtain a test accuracy of 48%, meaning that the program gets the genre right about one time out of two for the ten genres we have.

4.2 MLP Neural Network

After having seen the results from the logistic regression, we realized that according the fact that we have 10 classes, it could be a good try to perform our data in a neural network using the Multi Layer Perceptron Classifier. It trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. We decided to use two input layers of ten neurons each plus the bias for our Neural Network. After having tried with three or more layers, we

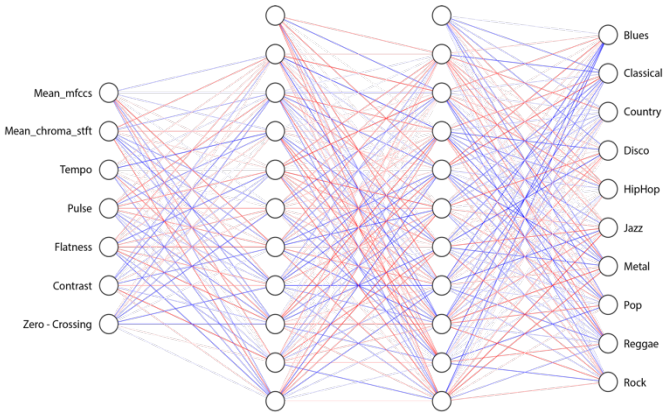
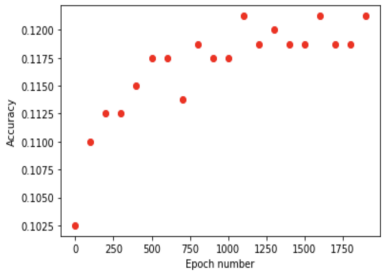


Figure 6 - Neural network representation for ten genres

realized that the changing in accuracy was not significant, but it was way time longer.

We first trained our neural networks with all the 10 genres, but we obtained a really low result of accuracy.

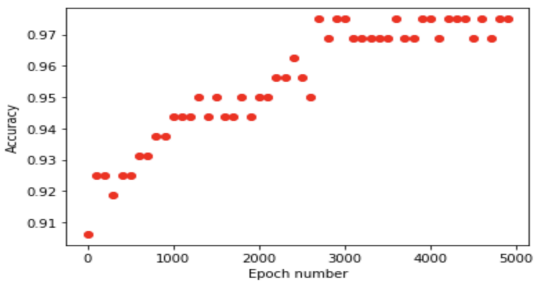
We came to the conclusion that it could be because of the number of classes and because we don't have that many amounts of data, a thousand in total and a hundred for each genre. It is not enough to train well our Neural Network. The second idea was to say that probably our features are not all relevant to this issue. So, we decided to see what the results could be if we kept only two classes that are quite different according the popular opinion: classical and rock.




```
=== Result on test data ===
Accuracy: 12.5 %
```

Figure 7 - Accuracy over Epoch number for ten genres classification neural network

The results here showed how our neural network is able to clearly compare classical and rock. It immediately refuted the fact that the features of our dataset are not relevant.




```
=== Result on test data ===
Accuracy: 95.0 %
```

Figure 8 - Accuracy over Epoch number for two genres classification neural network

4.3 Random Forest

In a way we were not really convinced about the efficiency of using a neural network for the classification of the 10 genres. We were not really sure about what the origin of a really low accuracy. So, we decided to implement the random forest classifier method with sklearn. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

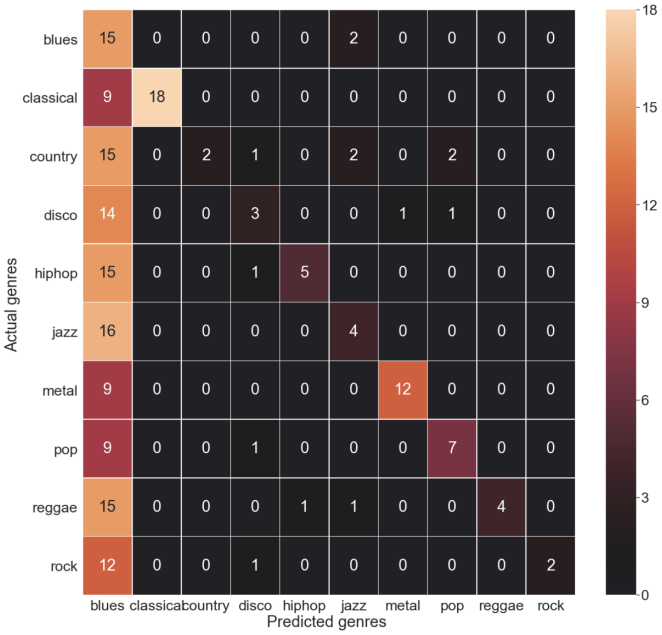


Figure 9 - Confusion matrix for ten genres using the Random Forest method

In order to understand where the issue of a really low accuracy come from, we chose to print the confusion matrix of the results from the results of the Random Forest method. You can see the confusion matrix in *Figure 8* We realized that some of the classes were most of the time not classified as they were supposed to be. Moreover, they are often misclassified as a blues song. So, we came to the conclusion that perhaps we didn't have enough features to distinguish the genre that looks the most similar such as reggae and blues. But four of the classes were well classified most of the time as you can see it in figure 8. These genres are blues, classical metal and pop.

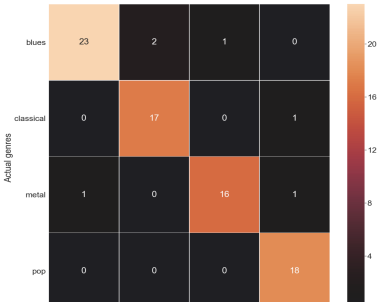


Figure 10 - Confusion matrix for 4 genres using Random Forest

What we did next, was to try the same Random

Forest Classifier with only these four genres in order to see how the confusion could looks like: blues, classical, metal and pop.

As we expected, on Figure 10, we obtained an accuracy of 92% and almost a perfect classified confusion matrix. Our data set and our features were at least really effective if we kept only the four genres beyond.

4.4 Support Vector Machine / One VS Rest

After the discovery we made with the confusion matrix and the fact that some features are bewildered with blues sometimes, we thought about another interesting problem. How is the accuracy going when we try to classify one class versus all the others. Thus, we decided to try the OneVSRestClassifier from sklearn on our Dataset. It appeared that this classifier uses at first a linear support vector machine method.

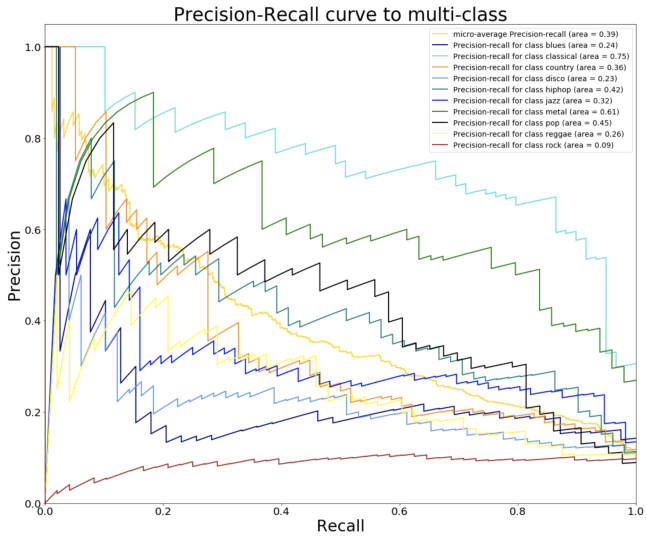


Figure 11 - Precision-Recall curve using One VS Rest Classifier with SVM

So we decided to print the precision recall curve for each classes. Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

As you can see beyond, the chart revealed that it is not relevant to try to predict the rock. While if you try to know if the genre of your song is classical or not, the output answer will be rather relevant. It is actually something that we already observed in the logistic regression accuracy table of one genre vs one genre in Figure 5.

While rock used to have the worst accuracy when comparing itself one by one with other genres using logistic regression, classical was the one with the highest accuracy.

accura	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
rock	0,725	0,95	0,75	0,7	0,85	0,825	0,825	0,825	0,8	1
classical	0,95	1	0,95	0,925	0,975	0,9	1	0,975	0,95	0,95

Figure 12 - The rock and classical accuracies of the comparison of it with each genre using Logistic Regression

The point here is that regarding to the dataset we have rock is not enough distinguishable to use it in the model.

Conclusion and future work

All our models gave similar results which were sometimes surprisingly accurate, always being able to differentiate two genres between them, or classifying with a very high accuracy songs of 4 different genres. One of the major conclusion we got to is that classical or metal music is easily distinguishable with our actual features. Whereas rock and some others are not, probably because we need more specific features because of the complexity and diversity of the rock music.

What could be interesting for us to study and improve our models is using as inputs directly the spectrograms or chromatograms of the features instead of computing the mean which is a loss of information about the music. We could also try to find few other features that could be more specific to rock music in general, such as determination of the instruments used in the song. Rock always comes along with a guitar, a bass and a drum as a basis so it could be relevant to identify the instruments that are used and add it as a feature.

Moreover, enlarging our database with other songs could provide us with better results for the neural network for instance. It would be interesting also to only test these models with songs not provided by the GTZAN database.

References

[1] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5):293–302, July 2002.

[2] GTZAN Genre Collection
http://marsyas.info/downloads/datasets.html

[3] Music Genre Classification with Python
https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8

[4] Stanford Machine Learning Project: Music Genre Classification
http://cs229.stanford.edu/proj2018/report/21.pdf