**Music genre classification using SVM**

**Vanja Šerfeze, Aleksa Stevanović, Stefan Milošević**

* Motivation

We wanted to make a classification solution that wasn’t the usual problem to solve, and music genre classification was the best choice for that. It has a lot of interesting features that can be extracted and theres a lot of different solutions to this same problem. We hope that we can further the reaserch in this growing field of Computer Science and make our small contribution.

* Research questions

The problem we are handling in our project is the classification of music by the genre of the song. We used the **GTZAN** dataset, which contains a collection of 10 genres of 100 audio files each. Each file is 30 seconds long. From this dataset we first created a 1000 audio files per each genre by splitting the 30 second long audio files into 3 second long ones. The result is more data that can be used to train the model. From this modified dataset we are extracting some audio file features that are commonly used in the classification of music. The features that we extracted are:

* **Zero Crossing Rate:** The zero crossing rate is the rate at which a signal changes its sign. In the context of audio, it represents the number of times the audio waveform crosses the zero-axis per unit of time. It provides information about the frequency content and changes in the audio signal.
* **Spectral Centroid:** The spectral centroid is a measure of the center of mass or the average frequency of an audio signal. It represents the distribution of spectral energy in the signal. A higher spectral centroid value indicates that the audio signal has a higher frequency content.
* **MFCC (Mel-frequency Cepstral Coefficients):** MFCCs are widely used features for audio signal processing. They represent the spectral shape of the audio signal by capturing the power spectrum of short overlapping frames of the signal and then applying a logarithmic transformation. MFCCs are commonly used for speech and music analysis tasks.
* **Spectral Contrast:** Spectral contrast measures the difference in magnitudes between peaks and valleys in the frequency spectrum. It provides information about the perceptual difference in the spectral content of an audio signal, capturing local variations and highlighting regions with significant energy differences.
* **Spectral Rolloff:** Spectral rolloff is a measure that represents the frequency below which a specified percentage of the total spectral energy lies. It indicates the cutoff point in the spectrum and provides information about the overall brightness or timbre of the audio signal.
* **Spectral Bandwidth:** Spectral bandwidth measures the spread of frequencies in the power spectrum. It represents the range of frequencies over which a signal is distributed. A higher spectral bandwidth indicates a wider distribution of frequencies in the audio signal.

The resulting dataset is a table of 10000 values that contains 7 columns. For the control dataset we used the extracted features from the same GTZAN dataset that we are using but a lot more features. That also has 10000 values but a lot more columns.

* Related work

The first thing that we saw based on this same problem, of music genre classification it was an article about how you can use a CNN to solve this same problem by transforming the original audio file into an image of a mel spectrogram. The idea stuck with us until finally it came time for us to make our solution, be it with a simpler model. Other solutions to this same problem are:

* The work of Dannenberg et al. based of naïve bayesian and neural network approaches.
* The work of Tzanetakis and Cook, they proposed three different feature sets to represent timbral texture, rhythmic and pitch content. Short-time Fourier Transform (STFT), Mel-frequency Cepstral Coefficients (MFCCs), Wavelet Transform (WT).

These works are only a sample of all the work done in this field, there are many more papers written on this subject but we can’t place them all here.

* Methodology

First of all we enlarged the dataset by splitting the original 30 second audio files into 10 3 second files. For the enlargment part of the process we used the pydub library and the AudioSegment class for the splitting and extraction of the many smaller audio files.

Then we extracted the features that we mentioned in the second part of this project paper using librosa and its built in functions for the extraction of features from the audio files.

After that we read the dataset file into a pandas dataframe, split it into training and test parts so that 80% of the data was used for training and 20% was used for the testing. After that we used a grid search to find the best hyperparameters of the model. For the training of the hyperparameters we used five-fold cross validation. From there on we calculated the top 3 top 2 and the normal accuracy scores. The result that we got were okay, but could be better probably with a larger model than a SVM. The measures that we got:

* Top 3 Accuracy: **0.6255**
* Top 2 Accuracy: **0.51**
* Accuracy: **0.322**
* Precision: **0.31011356376518356**
* F1 Score: **0.30438678644611405**

We made another classifier using the whole dataset without any selection of the features that we were going to use and we got these scores.

Reference model scores:

* Top 3 Accuracy**: 0.6491491491491491**
* Top 2 Accuracy: **0.495995995995996**
* Accuracy: **0.3203203203203203**
* Precision: **0.2866763410918696**
* F1 Score: **0.2776853581428256**
* Discussion

First we tried only using two features: the zero crossing rate and the central spectroid. But the model didn’t generalize well enough. We only got around 15% accuracy score which was close to a random guess. After we added more features the accuracy got much better. So the tried and tested method for machine learning stands, more data equals a better model. The best result that we got was with a C parameter of the SVM equal to 10. With this we avoided overfitting the model. We got a pretty balanced model because all of the metrics used were of a similar value.

Overall we believe that a more complex model would do a better job. Using Ensambling models that use bagging or boosting, a neural network or even a convolutional neural network.

* References

1. Song-Level Features and Support Vector Machines for Music Classification -

Mandel, Michael I.; Ellis, Daniel P. W.

2.Musical genre classification using support vector machines-

Changsheng Xu; N.C. Maddage; Xi Shao; Fang Cao; Qi Tian

3.Music Classification using MFCC and SVM-

R. Thiruvengatanadhan

4.Audio music mood classification using support vector machine -

Cyril Laurier; Perfecto Herrera