Audio analysis and feature extraction

David Štych
Aleksandra Jamróz

November 21, 2022

Introduction

Numerous industries have found extensive use for audio processing. Nowadays, we can meet applied audio processing every day, from using apps like Shazam to find a song to voice recognition used in translators, home assistants, etc.

In this laboratory, we focus on feature extraction, which means the extraction of various properties-features from the original sound file.

Feature extraction

On top of the baseline implementation, we added another three features. The five features we used are:

1. *Average value of the entropy of the energy of the audio signal's frames*
    - Included in the baseline code
2. *Maximum value of the entropy of the energy of the audio signal's frames*
    - Included in the baseline code
3. Average value of the spectral entropy of the audio signal's frames
    - Implemented in the provided code
    - Spectral entropy describes the pureness/noisiness of the signal. It might be helpful to our model.
4. Average value of the spectral flux of the audio signal's frames
    - Implemented in the provided code
    - Spectral flux is a measure of change in the signal power spectrum. Minor and major chords should have different rates of change. Therefore, it is useful for classification.
5. Average value of the zero crossing rates of the audio signal's frames
    - Implemented in the provided code
    - The zero crossing rate is the rate at which the signal goes through zero (changing from positive to negative and vice versa). It is one of the commonly used features to use when analyzing music samples. It led to noticeable improvement during our testing.

We have also tried experimenting with other features. We have been attempting to add *maximum value of the zero crossing rate of the audio signal's frames* and *average value of the spectral contrast of the audio signal's frames*. However, in both cases, a detrimental effect was observed, so we decided not to include them in our model.

Results

We kept the default split ratio in the dataset (70% training, 30% testing). Firstly, we kept the default parameters (frame length, number of subframes, and hop length) and experimented with the features. We tracked the progress and recorded the results in the table below.

Features	AUC
Baseline	0.486
Baseline Mean of spectral entropies	0.554
Baseline Mean of spectral entropies Mean of spectral flux	0.577
Baseline **Mean of spectral entropies** **Mean of spectral flux** **Mean of zero crossing rate**	**0.6**
Baseline Mean of spectral entropies Mean of spectral flux Mean of zero crossing rate Max of zero crossing rate	0.588
Baseline Mean of spectral entropies Mean of spectral flux Mean of zero crossing rate Mean of spectral contrast	0.583

After we had decided which features to use, we implemented an automatic algorithm to tune the parameters of the feature engineering. Sets of three parameters are defined in a list, and a function goes through the list and tries to train and test the model. AUC is calculated and stored in another list. After completion, the set of parameters with the highest AUC is selected. The effect of the parameter change can be seen in the table below.

The best parameters found were:

- Frame length: 1024
- Number of subframes: 8
- Hop length: 256

Frame length	Number of subframes	Hop length	AUC
512	10	128	0.6004
512	10	256	0.5944
512	8	128	0.6050
512	8	256	0.5541
1024	10	128	0.5399
1024	10	256	0.5794
1024	8	128	0.6409
1024	**8**	**256**	**0.6424**

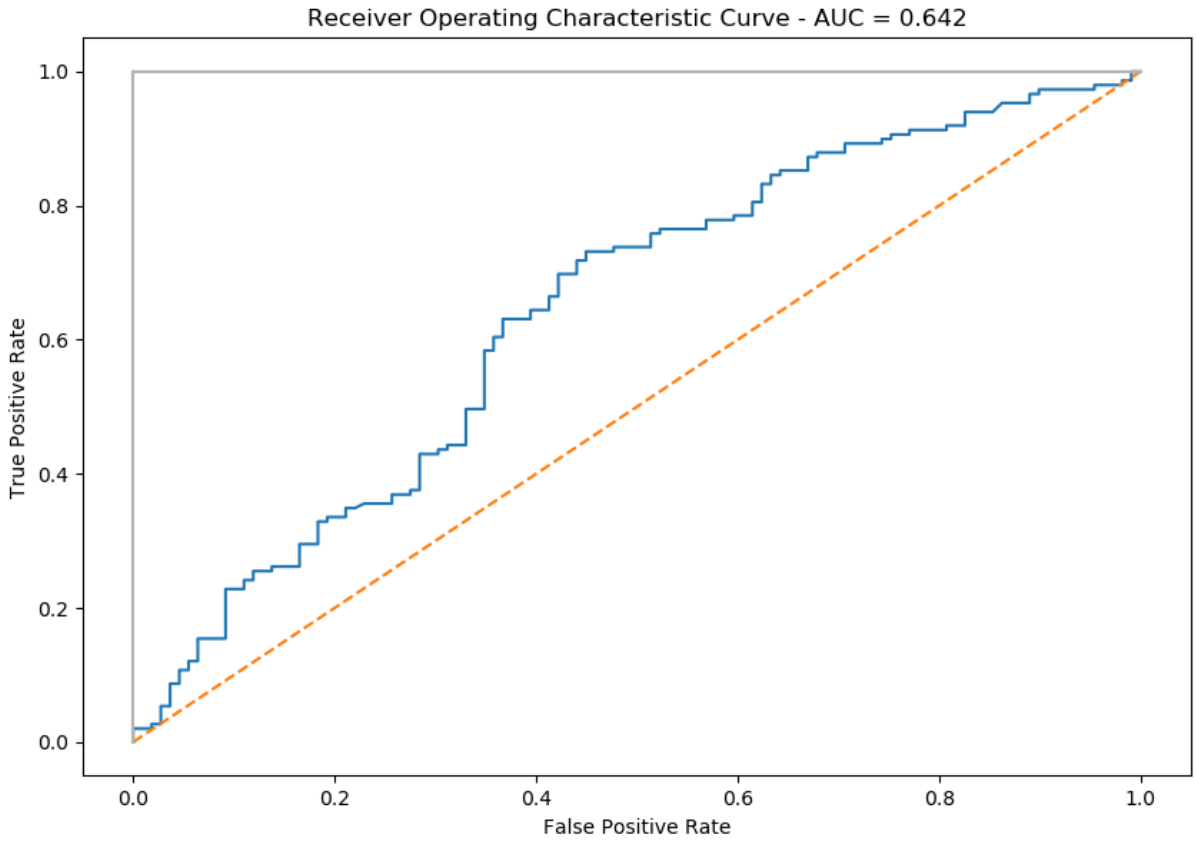

Figure 1: ROC curve plot - best-achieved result