Bachelor in Data Science and Engineering - 2021/2022
University Carlos III Madrid
Audio processing, Video processing and Computer vision

SERGIO AIZCORBE (**100406602**) | BERNARDO BOUZAS (**100406634**) | LS6 REPORT

Introduction

The provided dataset contains a collection of barking and meowing audio samples in uncompressed WAV format with various background noises. The aim of this practice is to study the significance of different audio filters and the quality of the prediction results based on the extracted features. Additionally, extra functions have been implemented and tested and two cross-validation tools have been designed in order to get both the best combination of feature extraction algorithms and their respective hyperparameters.

Feature extraction: Introduction

All of the filters included in the *audio_features.py* have been used to test its relevance in the classification stage. Additionally, **two extra filter functions** have been implemented based on the *LibROSA* library.

- **Root-Mean-Square**: it squares the amplitude (signal value), averaged over a period of time. Then, it square-roots the result, which gives a value proportional to the effective power of the signal. We use it as a way of obtaining the mean of values over time.
- **Polynomial Features**: It obtains the coefficients of fitting an nth-order polynomial to the columns of a spectrogram.

In addition to the metrics that were provided by the code template, two more metrics have been stored in the result matrix (M):

Maximum value	**Average value**
Minimum value	**Sum of values**

Thus, the results fed into the SVC will have dimension (*num_data*, *n_feat * 4*), in this case.

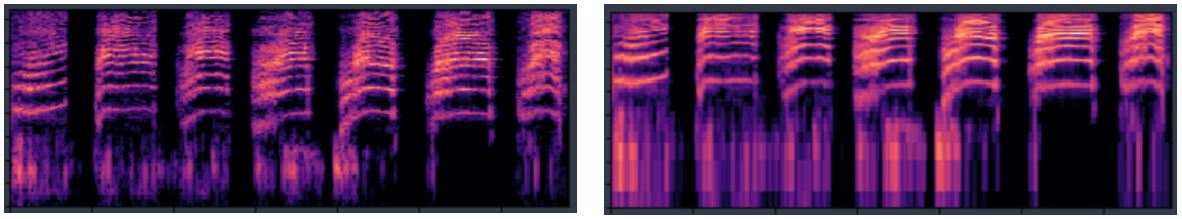
Following this, an in-depth study of the predefined functions and their possible impact on a binary classification problem has been carried out. Special emphasis has been given to the research of two one in particular: the **mel-frequency cepstral coefficients** and their representation in **mel-scaled spectrograms,** as we found they are widely used**.**

Studies have shown that humans do not perceive frequencies on a linear scale. The mel scale relates the perceived frequency of a tone to the frequency that is actually measured, scaling it to more closely resemble what is perceived by the human ear.

The mel spectrogram remaps the values in hertz to the mel scale. The y-axis is converted to a logarithmic scale and the color dimension becomes decibels, which could be interpreted as a logarithmic scale of the amplitude.

Mel Frequency Cepstral Coefficients are thus coeficients for extracting features from the components of a speech audio signal based on human auditory perception. Here is a comparison between a normal spectrogram and a mel-scaled spectrogram, obtained by us:

SPECTROGRAM **MEL-SPECTROGRAM**



Although these two feature extractors are widely used in audio automatic recognition, we have investigated their use cases and believe that with a large amount of data and strong classifiers, such as convolutional neural networks, the mel spectrogram can often work better than MFCCs. This assumption will be elucidated later on.

The rest of the plots are contained in the *visualization.ipynb* file.

Feature extraction: Feature functions Cross-Validation

Given that the significance of each filter might differ depending on the number of samples to train with and the respective classification task, **cross validation** has been used to test all the possible combinations of filters. In order to use the cross validation built-in by the *sklearn* framework, a **custom object derived from a transformer abstract class** has been designed to search for the best possible combinations of functions. The grid-search tries to maximize the **AUC score** for the classification.

Due to the large number of combinations and thus, the computational time, cross validation has been made with 5 validations folds. The following functions are the ones which maximize the AUC score of the dataset:

Preprocessing	The basic preprocessing function, mandatory.
Energy	The area under the squared magnitude of the signal.
Energy entropy	A measure of the amount of information a signal carries.
Zero-crossing rate	The rate at which a signal changes from positive to zero, and vice versa.
Polynomial features	Coefficients of fitting an nth-order polynomial to the columns of a spectrogram

Concerning our initial approach emphasizing mel-frequency cepstral coefficients and spectrograms, we are not surprised that they haven't shown up as optimal features, as yielded by our CV tool. After all, they are very efficient in speech, dictated numbers and speaker recognition, which is a quite different framework from our specific use case with animal sounds.

Moreover, the Zero-Crossing-Rate is usually a key feature to classify percussive sounds, which could fit the context of the sounds we have been given.

Regarding the energy-related features, we believe that their presence makes sense, since the audio signal contains different amounts of entropy and energy levels over time, and the distribution of these can be a determining factor for their classification. We have found that entropy-based algorithms in information theory literature are considered as well-performing when measuring the irregularities in the tested signals.

Feature extraction: Hyperparameter tuning Cross-Validation

Once the best combination of functions is known, another transformer has been custom-designed to implement cross-validation with the parameters of these functions. These are the tested hyperparameters for the used filters:

	USED IN	PARAMETER RANGE
Threshold (*thr*)	*preprocess_data()*	[10, 20, 30]
Frame Length (*flen*)	*get_energy(), get_energy_entropy(), get_zero_crossing_rate()*	[512, 1024, 2048]
HOP Length (*hop*)	*get_energy(), get_energy_entropy(), get_zero_crossing_rate()*	[512, 1024, 2048]
No. of Subframes (*nsub*)	*get_energy(), get_energy_entropy()*	[8, 10, 12]
Polynomial Order (porder)	*get_poly_features()*	[1, 2, 3]

A total of 5 validations folds have been used and these are the obtained optimal results for this functions:

- ➔ ***preprocess_data***
  - *thr = 20*
- ➔ ***get_energy***
  - *flen = 2048 hop = 512 nsub = 12*
- ➔ ***get_energy_entropy***
  - *flen = 2048 hop = 512 nsub = 12*
- ➔ ***get_zero_crossing_rate***
  - *flen = 2048 hop = 512*
- ➔ ***get_poly_features***
  - *porder = 2*

Evaluation and discussion of results

We have combined an approach based on theoretical assumptions and evidence justification with a fairly effective cross-validation methodology and, in our opinion, the results are very satisfactory. In particular, we believe that the course of action followed will to some extent avoid overfitting. It has been noticed that in our evaluation the results vary slightly depending on the seed, obtaining an accuracy in the classification between **0.93** and **0.97**.