Nume: Nastase Alexandru

Grupa: 233

**Brief introduction**

For this machine learning project I have decided to go with Support-Vector Machines (SVM)

Algorithm (after trying KNN model but it’s results were bad compared to SVM), and tried to focus on the LIBROSA library along with mel-frequency cepstrum (MFCC) spectrogram feature from librosa.

I have tried a numerous parameters from different functions from librosa, like librosa.load(), librosa.feature.mfcc() and svm.SVC method.

**A look into the code and the step-by-step results**

After I finished reading and stocking the data from the .txt files, I iterated through the train files, and for each filename, I searched for it in the actual files from train folder, since they are not in the same order in the train.txt file.

After that, I used librosa.load() to decode my audio. After many tries of “playing” with the parameters and values of them, I found that the very default parameters work the best (as far as I discovered), but the most important parameter was res\_type to be set to ‘kaiser\_best’.

Now, since I had my audio as a numpy ndarray of floating points and the sample rate of it, I searched the whole internet in the hope of finding the best librosa feature that can match with my dataset. After many tries of different functions that generated and rendered different spectrograms like librosa’s melspectrogram, rms, spectral\_contrast and others, I have reached the conclusion that librosa’s mfcc spectrogram shows the best results, which is basically the highest accuracy.

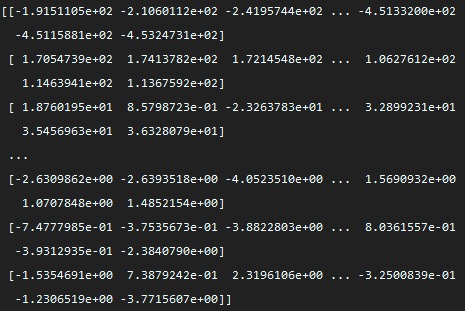
**For example, we take the first audio file and examine it’s transformation along the code:**

The first audio file: 102333.wav



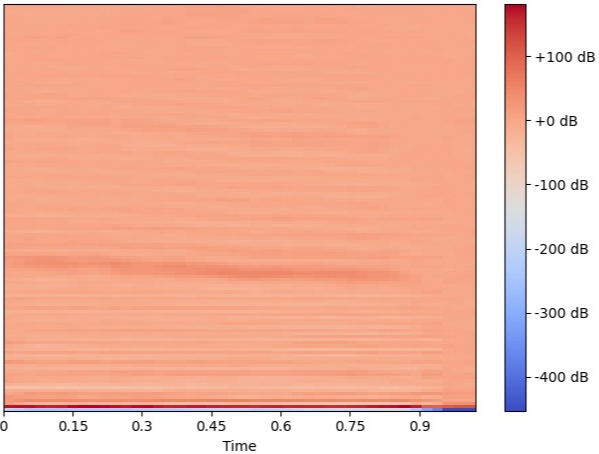
It’s numpy ndarray after using librosa.load():

The same file after I used MFCC on it:



I have observed that if the number of mfccs processed is higher, the accuracy will be higher too. The default value of the parameter n\_mfcc from librosa.feature.load is set to 20. If I increase it to a value greater than 100, there will be differences even greater than 0.05 – 0.07, which is a huge 5-7% impact on the accuracy.

I have rendered the spectrogram so I have a visualization of it:



After adding all the mfcss to a list and converthis the list to a numpy ndarray, I called the function “normalize” from lab5 and used different scalers. After some online research and tries of scalers, the best results were given by the standard scaler and minmax scaler.

Now for the SVC function I have opted for the next tuning parameters:

* **Regularization parameter C = 1** (basically the more I increased it, the more strict the model was, and was avoiding misclassifying each training sample but it was resulting in a worse accuracy)
* **Kernel parameter = linear**

I have tried modifying the other parameters but nothing helped.

**Conclusion**

After many tries of getting the best results on the data that I have been given, with SVM model, these are the results:

Accuracy score = 0.693

F1 score = 0.7095553453169348

I have tried using KNN model too but the best result was 0.565 or something like that, so I dropped the idea of using it.