MP2 Part 2 – Audio classification

Objective:

One of the ways of detecting anomalies in an industrial device is by listening to acoustic sound that they produce and then being able to detect if it sounds like normal or abnormal operation.

The task in the mini project will be to develop an application to classify acoustic sounds from industrial valves as 'Normal' or 'Abnormal'.

This project shall be done with the classical ML approach using the Scikit-Learn library.

Details:

- The data is in the form of folders for different valve Id's (product types) and sub-folders for normal / abnormal data. Compressed version of the individual valve folder is uploaded. It is recommended to unzip each of those and place the folder for each valve id in an overarching folder called valve as this will be convenient for the next actions. I have also provided as an alternative a link from where you can download the same zipped folders.
- Clearly you will have to find a programmatic way to load each audio file from each folder, extract features from it, and add it to a DataFrame to build the overall common dataset. You will also populate a label column in the DataFrame for each audio file.
- Develop the above part carefully. As a tip, you could use the python 'os' library to loop through the folders to read in the audio file one by one, extract features and store it in a list before converting the final list into a Dataframe. The process of running the for loop can take a while.
- Once that is done, you can split the data to train and test and do all the usual model building steps.
- Feature extraction will be done based on developing a traditional ML solution using the 'sklearn' library and one of the classification algorithms available in that library.
- Minimum features to be extracted: ***mfccs, chroma_stft, spectral_centroid, spectral_bandwidth, spectral_rolloff, zero_crossing_rate, rmse***. You can choose to extract more features if you so desire.
- Because each of these features, when extracted, give a range of values based on their sampling rate, you can take the mean of each of them and maybe the standard deviation of some too**. E.g.** *"mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)", "mfccs_mean = np.mean(mfccs, axis=1)"*
- You can extract the features using the 'librosa' library, examples of which have been shown in lectures.
- It may help to scale the values of the features in the dataset using the standard scaler - to get a more efficient gradient descent process.
- Develop the complete solution in a Jupyter Lab notebook.

Prediction on new data:

1. New data in the form of audio files for valves are provided to predict upon as normal or abnormal.
2. It is expected that the model developed will be able to predict the classes of the new data correctly.

Grading Criteria for each task:

1. The Solution code runs flawlessly and all requirements in the instructions are addressed – 45%
2. The features from the audio data of the various valves, provided in different folders, have been extracted and combined into a single DataFrame – 20%
3. The classification model has been developed and performs with 90% accuracy or greater - 15%
4. The predictions on new data are 100% correct – 10%
5. Questions are answered with a clear understanding of the concepts and in depth (check the requirements mentioned in the questions) – 7.5%
6. Code is well structured and well commented – 2.5%

<u>Answer the following questions in the last cell of the Notebook</u>

1. While you built this application in the classical ML way using tabular data, what is the other way you could have built this application where Deep Learning can be used? Explain the basic concept of that method and explain how it works.
2. Explain how one would have to preprocess the data to adopt the above deep learning method and how that can be implemented?
3. In completing this application what were the challenges and what were three takeaways that you would consider significant in understanding audio analytics?

This Mini Project will be due by October 28th midnight.