Explanation of the Bonus task:

Bonus Task (Audio Section):

In this task, I employed end-to-end deep learning algorithm for the classification of audio signals. I used the GTZAN dataset. The dataset is provided with labels, so employing a supervised learning algorithm is most suitable choice. In the other tasks, the feature extraction algorithms e.g., Spectral bandwidth, MEL-Frequency Cepstral Coefficient (MFCCS) are tested. For classification, we can employ STFT and easily generate spectrograms, which can be used as extracted features for training a machine/ deep learning algorithm. But I was curious to know the classification results of deep learning algorithm by using raw time series signal. Therefore, I employed EEG-Net, which is convolutional neural network-based algorithm for classifying EEG data. EEG-Net is one of the most successful algorithms for classifying motor-imagery tasks. Since, EEG signal and audio signals are time-series signal, so it makes sense to check the performance of EEG-Net on the audio signals as well.

In the following section, I will explain how to run the source code. I will also briefly explain the process of feature extraction from the raw audio signals and lastly briefly explain the architecture of EEG-Net and its performance on the test dataset.

Execution of Source code:

1. All the libraries that are required to run the source code are available in requirement.txt file.
2. Original GTZAN dataset ("genres_original" folder) is required: please provide the path on the 'line 85' of the 'main_bonus_audio_task.py'
3. All the parameters of EEG-net and the parameters to extract data snippet from the source audio file are also available in 'main_bonus_audio_task.py'
4. 'models.py' file defines the architecture of EEG-Net

Process of feature extraction:

In this task, I aimed for end-to-end learning, so only raw signals are used for feature extraction. I used the snippet of only 3-second to classify the given signal. I agree that using a snippet of bit more duration make more sense because it can contain more discriminative information. But it considerably increases the size of input because the sampling rate is quite high and as a result high computational power is required. As the main aim of this task is not to increase the classification accuracy, I also do not consider the resampling. Long story short, I consider 3-seconds to construct a feature vector.

But the code is quite flexible, we can extract the feature vector of different time duration. We also introduce the overlapping window. Off course, we can down sample the signal. Here, we used the original sampling frequency (22.05 kHz).

Architecture of EEG-Net: EEG-Net is a compact architecture, which contains three convolutional layers, which are a simple convolution layer, a depth convolutional layer and

separable convolutional layers. The hyperparameters are mentioned in 'main_bonus_audio_task.py'.

Training data and test data: In this task because of lack of computational power I only used the feature vector of first four classes. The dataset is divided into training and test dataset, with 70% and 30% ratios.

Classification accuracy: After training the algorithm for 100 epochs, I achieved 72% classification accuracy, however it can be improved by deepening the architecture or enhancing the window size or introducing the overlap size between different segments as well.

Should you require further information in running or understanding this script, please do contact me.