

Project Report
Option 1
Audio Classification System
(Team: Apurva Sharma & Harleen Bhamrah)

This project is to implement an audio classification system that involves audio signal processing and machine learning model.

The graphic user interface allows users to browse given test audio files from the given database, select the query audio, and to classify if it belongs to music or speech class. User can also listen the audio by playing the selected audio from the database.

Technical Details:

The project is implemented in Python using different libraries for audio classification and analysis. For the GUI, “*PySimpleGUI*” library is used to build the interface of the Audio Classification System in this assignment.

Next to play the audio in the system, “*simpleaudio*” library is used. *WaveObject.from_wave_file* of *simpleaudio* is used to read the selected audio file and *play()* method helps to play the read audio file.

For extracting the features and building a machine learning model from the training data, “*pyAudioAnalysis*” library is used for machine learning process. So, let's see briefly what happens using this library, what steps are taken for building a machine learning process.

pyAudioAnalysis implements the following functionalities:

- **Feature extraction:** several audio features both from the time and frequency domain are implemented in the library.
- **Classification:** supervised knowledge (i.e., annotated recordings) is used to train classifiers. A cross-validation procedure is also implemented to estimate the optimal classifier parameter (e.g., the cost parameter in Support Vector Machines or the number of nearest neighbors used in the kNN classifier). The output of this functionality is a classifier model which can be stored in a file. In addition, wrappers that classify an unknown audio file (or a set of audio files) are also provided in that context. In this system, we have used SVM model.
- **Segmentation:** the following supervised or unsupervised segmentation tasks are implemented in the library: fix-sized segmentation and classification, silence removal, speaker diarization and audio thumbnailing. When required, trained models are used to classify audio segments to predefined classes, or to estimate one or more learned variables (regression).

Feature Extraction:

The list of features can be extracted in a short-term basis: the audio signal is first divided into short-term windows (frames) and for each frame all 34 features are calculated. This results in a sequence of short-term feature vectors of 34 elements each. Widely accepted short-term window sizes are 20 to 100 ms. In *pyAudioAnalysis*, the short-term process can be conducted either using overlapping (frame step is shorter than the frame length) or non-overlapping (frame step is equal to the frame length) framing.

In mid-term basis, according to which the audio signal is first divided into mid-term windows (segments), which can be either overlapping or non-overlapping. For each segment, the short-term processing stage is carried out and the feature sequence from each mid-term segment, is used for computing feature statistics (e.g., the average value of the ZCR). Therefore, each mid-term segment is represented by a set of statistics. Typical values of the mid-term segment size can be 1 to 10 seconds. In cases of long recordings (e.g., music tracks) a long-term averaging of the mid-term features can be applied so that the whole signal is represented by an average vector of mid-term statistics.

Some of the important features listed below:

- Zero Crossing Rate: The rate of sign-changes of the signal during the duration of a particular frame.
- Energy: The sum of squares of the signal values, normalized by the respective frame length.
- Entropy of Energy: The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
- Spectral Centroid: The center of gravity of the spectrum.
- Spectral Spread: The second central moment of the spectrum.
- Spectral Entropy: Entropy of the normalized spectral energies for a set of sub-frames.

Feature values:

```
[array([[2.45040676e-01, 3.15111865e-02, 3.17557721e+00, ...,  
        5.21406781e-02, 1.99080187e-03, 3.10526811e-02],  
       [1.24101673e-01, 4.11929880e-02, 3.21445943e+00, ...,  
        1.07481839e-02, 1.05824959e-02, 8.19589886e-03],  
       [6.17021277e-02, 3.44112485e-02, 3.21701485e+00, ...,  
        2.12415452e-02, 1.95352752e-02, 9.63912630e-03],  
       ...,  
       [1.29053587e-01, 3.10897357e-02, 3.21956105e+00, ...,  
        6.26658625e-03, 9.39610061e-03, 4.58068415e-03],  
       [2.06304756e-01, 3.99737587e-02, 3.19051838e+00, ...,  
        5.89724712e-03, 5.27619131e-03, 9.01711025e-03],  
       [9.25375469e-02, 7.20151757e-02, 3.03012135e+00, ...,  
        6.38563498e-02, 1.53508001e-03, 2.05885666e-02]]), array([[0.15525657, 0.01947824,  
3.09194586, ..., 0.04942403, 0.02469654,  
        0.01699908],  
       [0.13762516, 0.13756768, 2.82146353, ..., 0.03408678, 0.00670018,  
        0.01872909],  
       [0.13482478, 0.01551619, 2.99203477, ..., 0.03851239, 0.01086711,  
        0.01751281],  
       ...,  
       [0.0810388, 0.04766432, 3.07821006, ..., 0.03185346, 0.02522733,  
        0.01330875],  
       [0.07352941, 0.05030176, 2.89828662, ..., 0.04468325, 0.00661992,  
        0.01605208],  
       [0.09324155, 0.14803006, 2.74645757, ..., 0.03569315, 0.00925073,  
        0.01650709]])]
```

Audio Classification:

The library provides functionalities for the training of supervised models that classify either segments or whole audio recordings. Support vector machines and the k-Nearest Neighbor classifier have been adopted towards this end. In addition, a cross-validation procedure is provided to extract the classifier with optimized parameters. In particular, the precision and recall rates, along with the F1 measure are extracted per audio class. Parameter selection is performed based on the best average F1 measure. High-level wrapper functions are provided so that the feature extraction process is also embedded in the classification procedure. In this way, the users can directly classify unknown audio files or even groups of audio files stored in particular path.

Execution Steps:

The project consists of one important python file “*AudioClassification_main.py*” with all the implementations. The given 40 files of (music and speech) are divided into training data and test data for machine learning process. From each 20 files of music and speech, 12 each of music and speech are selected as a training data and remaining 8 of each music and speech are kept for testing and validation.

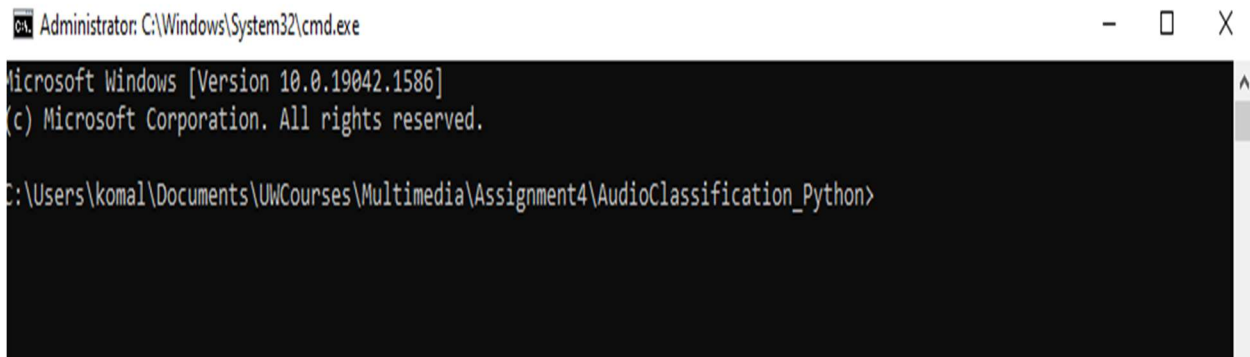
Extract the “**AudioClassification_Python.zip**” folder. The folder consists of train_data, test_data and main python file “*AudioClassification_main.py*” and requirements.txt file.

To get the UI interface, run *AudioClassification_main.py* as a python application, i.e.,

1- Open the command prompt from this extracted folder “**AudioClassification_Python**”.

For example, please find attached screenshot for your reference, the path may differ for each user local system where the folder is extracted.

Dependencies: Make sure you have python3 installed. Install required packages with ***pip install -r requirements.txt*** on the cmd.



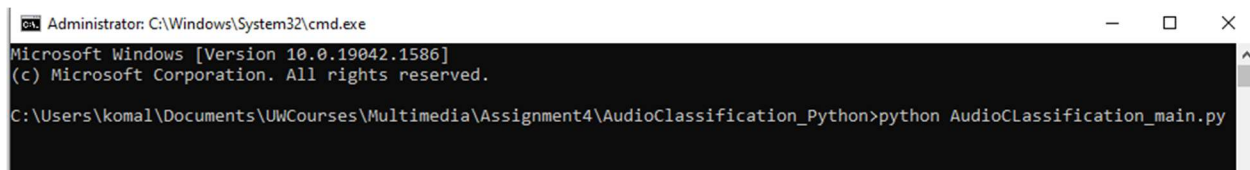
```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\komal\Documents\UWCourses\Multimedia\Assignment4\AudioClassification_Python>
```

2- At the extracted folder path – run the main python file using following command:

python AudioClassification_main.py

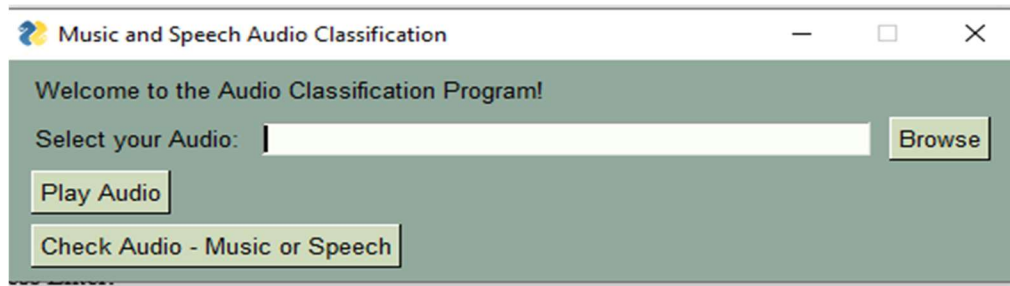
and press Enter.



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\komal\Documents\UWCourses\Multimedia\Assignment4\AudioClassification_Python>python AudioClassification_main.py
```

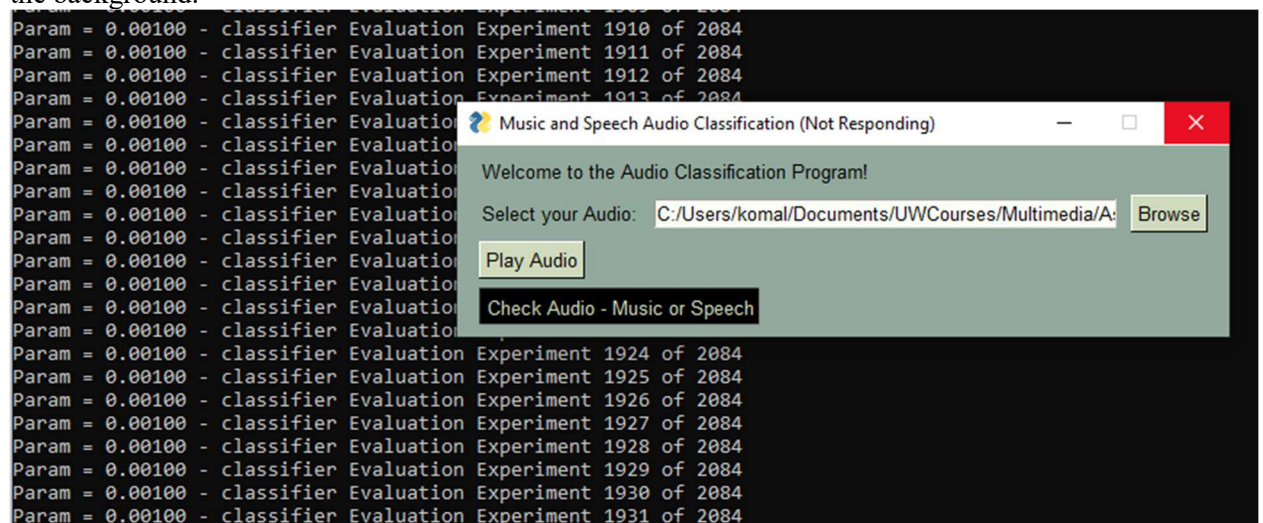
3- GUI for the Audio classification will get open, after step 2 is executed.



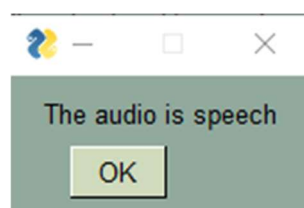
4- On the GUI, user can select the desired audio, he/she wants to classify to whether its music or speech and play the audio files as well. So, in order to check for the classification class, please follow below steps:

- 1- Select any audio from the given audio files under “**test_data**” folder from the extracted folder as your query audio for classification using “**Browse**” button.
- 2- User can play the audio if he wants to listen the audio using “**Play Audio**” button.
- 3- To check the audio if it belongs to music or speech class, user can click on “**Check Audio – Music or Speech**” button. The machine learning process takes around 1 minutes and 20-30 secs to train the model using train_data and build a classification model with the highest accuracy. Once the model predicts the given query audio based on machine learning model, result is displayed on the different pop-up window, classifying the class of the given selected audio.

Different feature extraction, model training, evaluation of the classification model takes place at the background.



The result is displayed after the machine learning model predicts the given audio. In this case, its speech.



Evaluation results: These results can also be seen on command prompt after getting above classification result.

SVM Model:

	music			speech			OVERALL			
C	PRE	REC	f1	PRE	REC	f1	ACC	f1		
0.001	100.0	82.4	90.3	84.8	100.0	91.8	91.1	91.0		
0.010	100.0	89.8	94.6	90.4	100.0	94.9	94.8	94.8		
0.500	100.0	91.2	95.4	91.9	100.0	95.8	95.6	95.6		
1.000	100.0	92.2	95.9	92.9	100.0	96.3	96.1	96.1	best f1	best Acc
5.000	100.0	92.0	95.9	92.6	100.0	96.2	96.0	96.0		
10.000	100.0	91.1	95.3	92.1	100.0	95.9	95.6	95.6		
20.000	100.0	91.7	95.7	92.1	100.0	95.9	95.8	95.8		

Confusion Matrix:

```

      mus  spe
mus  45.60  3.87
spe   0.00  50.53
Best macro f1 96.1
Best macro f1 std 15.9
Selected params: 1.00000

```

KNN Model:

	music			speech			OVERALL			
C	PRE	REC	f1	PRE	REC	f1	ACC	f1		
1.000	100.0	91.0	95.3	91.7	100.0	95.7	95.5	95.5	best f1	best Acc
3.000	100.0	83.9	91.3	86.1	100.0	92.5	92.0	91.9		
5.000	100.0	84.7	91.7	87.5	100.0	93.3	92.6	92.5		
7.000	100.0	84.4	91.6	86.3	100.0	92.6	92.1	92.1		
9.000	100.0	82.7	90.6	85.3	100.0	92.1	91.4	91.3		
11.000	99.3	82.8	90.3	85.7	99.4	92.1	91.3	91.2		
13.000	98.3	84.0	90.6	86.3	98.5	92.0	91.3	91.3		
15.000	99.3	83.5	90.7	86.0	99.5	92.2	91.5	91.5		

Confusion Matrix:

```

      mus  spe
mus  45.67  4.51
spe   0.00  49.82
Best macro f1 95.5
Best macro f1 std 16.6 Selected params: 1.00000

```

REFERENCES

[1] Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, NCSR Demokritos, Patriarchou Grigoriou and Neapoleos St, Aghia Paraskevi, Athens, 15310, Greece.

[2] <https://towardsdatascience.com/extracting-features-from-audio-samples-for-machine-learning-7b6a9271984>