Lab 9 & 10 - Voice Recognition

Classical Voice Recognition

In the classical approach we will use Mel-frequency cepstral coefficients (MFCCs) as features and Gaussian mixture models (GMMs) to classify them.

First of all, download the Voice Dataset:
https://drive.google.com/file/d/1BPIH87RhrypLmsIdrSX5zoPOMezr17yq/view?usp=sharing
It consists of a single word recorded by 9 different people in the .wav format and it's divided into train and test subset. Our goal is to extract MFCC from each train recording and fit GMM to them. After that we will validate our simple system on the test subset.

1. Import all necessary libraries: numpy, os, librosa (for MFCC extraction) and from sklearn.mixture import GaussianMixture.
2. Training process - for each recording from the train subset:
    a. Load recording - librosa.load;
    b. Extract MFCC features - librosa.features.mfcc with `n_mfcc=39` and `sr` set accordingly to the input data (later try manipulate the n_mfcc value and observe the results);
    c. Fit GMM to the extracted features - GaussianMixture with `n_components=32` and `random_state=0` (later try to manipulate the n_components value and observe the results). Note that - according to the sklearn documentation - features' vector should be transposed to form a row vector instead of column vector (you can use transpose from numpy);
    d. Add GMM to some collection (e.g. append to the list).
3. After successful training, classify the test recordings based on GMMs - for each test recording:
    a. Load and extract MFCC;
    b. For each GMM in the collection, compute the score for MFCC extracted from the test recording. GMM with the maximum score represents the detected speaker.
4. Now try to manipulate parameters of MFCC extraction and GMM fitting - write down (e.g. in the comments) the achieved results.