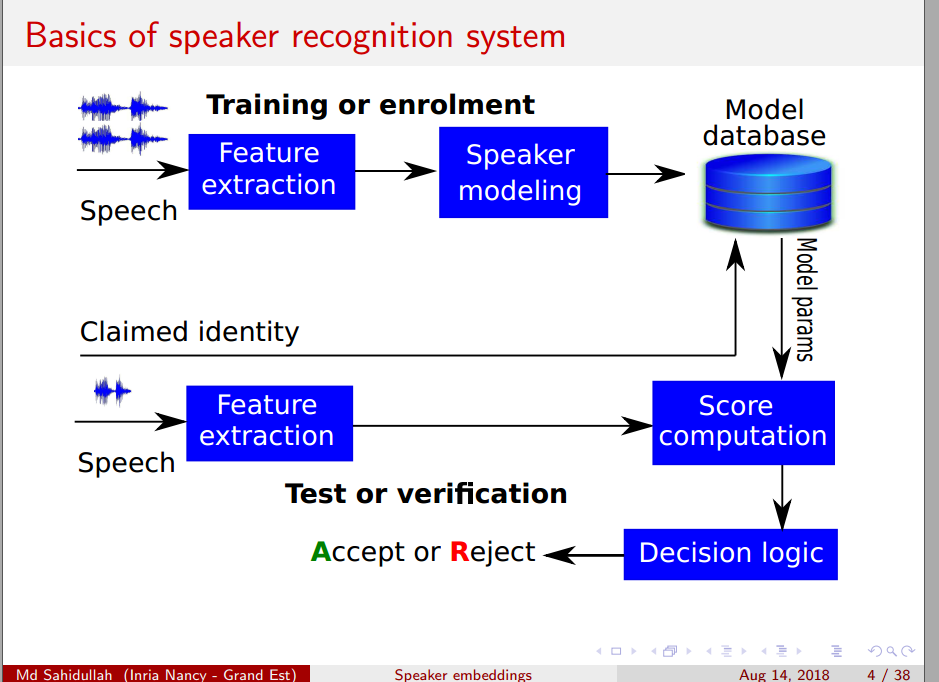
CELEBRITY VOICE MATCHING PROJECT WORK

1. Environment:

* Windows 10
* Spyder platform
* Python language

1. Theory:



The purpose is to find the celebrity voice that best matches to my own voice. Compare similarity of voices by using an automatic speaker recognition system, so the speaker recognition system is needed in this project, after the speech is recorded, there is feature extraction to get the features of the speech, process the features help to make the model, then the model can be trained with those processed features, with this model, it is able to score the speech, then we can compare the similarity and select the highest one to match the result.

1. Implement:

1) Record my speech : I record it as .wav file

2) Use pretrained models to set up: I load the ivectors.npy, plda.npz, tmatrix.npy, ubm.npz, those are the pretrained objects that use the celebrities data.

3) Extract MFCC -> Sufficient statistics -> I Vector -> I-Vector processing

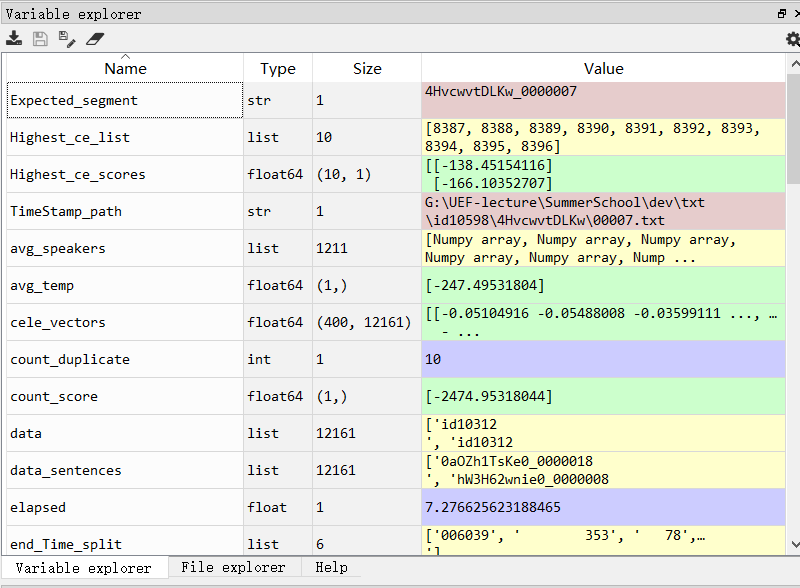
In this step, I use the function mfcc get the extracted MFCC, and store the features into npy file, then I use the pretrained ubm model to get sufficient statistics, and store it into another file, next I use the pretrained t-matrix to get the I-vector of my speech, it is in a variant, and use the function in backend.py to do i-vector preprocessing.

4) Score my vector against celebrities’: In this step, I use the pretrained PLDA model to do the score against celecrities’

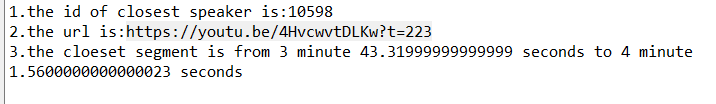
5) Average scores, in order to get the highest score related celebrity and segment, I take the average score for every celebrities, so each celebrity has an average score. It will be used to find the highest one, which is the most similar celebrity compared with my voice.

6) Get url and according to timestamp, get start time and end time: In this step, the url can be obtained by combine the youtube and the segment id, I download timestamps for the dev set to match the i-vectors to the segments of Youtube videos from which the i-vectors were extracted., what I do is use the start timestamp and end timestamp divided by 25, so I get the start time and end time for the most similar segment,

1. Result:



The output is:



1. Conclusion:

The time cost for my recording to get the result is 7.42 secondes, the longest part is processing the ubm model.

We can jump to the start part of the video by adding the start time, for example, <https://youtu.be/4HvcwvtDLKw?t=223> is able to just jump to the start part of the segment, because there is a ‘?t=223’ it means start to play at the time which t=223