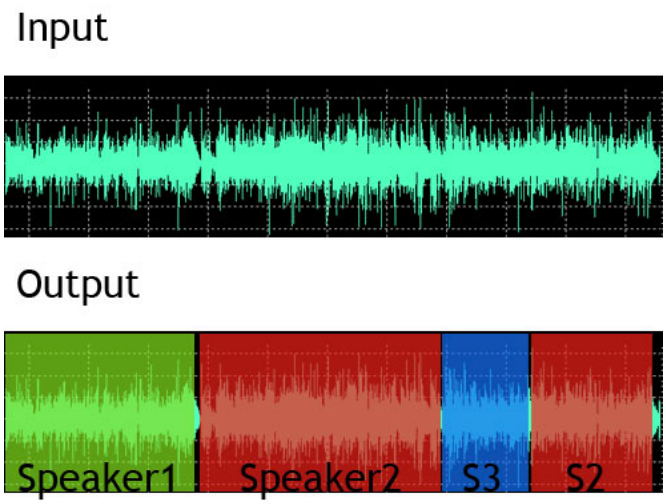
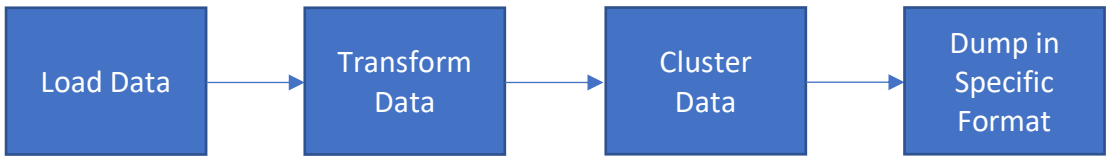
Task for Future Rider

Clone from github repository https://github.com/Jamiroquai88/Rider-Candidate.git
In this task, we want to simulate data clustering similar to task of speaker diarization. Imagine audio recording with length of **1000** seconds segmented into **1000** chunks. For each frame, we want to assign this 1 second long chunk to one of **2** speakers and to **silence**. Of course, we are not going to use waveform as input, but we use vector embedding generated from MFCC features (which hopefully contain some information about speech signal).



Processing pipeline can look for example like this:



Input data

Input data are stored in subdirectory *data*. Input embeddings are stored in **cPickle** format in file *data/embeddings.pkl*.

Data transformations

We do not want to use raw data, but we want to somehow transform them. We want to use two transformations

1. Subtract mean from all data
2. Use pre-trained **LDA** (Linear Discriminant Analysis) to reduce dimensionality of embeddings from **600** to **150**. LDA model is saved as **numpy** array in file *data/lda.npy*. (optional)

Data Clustering

We want to cluster data (or assign in smart way) to **2** speakers and **silence**. It is recommended to use **K-Means**, **Agglomerative Hierarchical Clustering** or any other clustering method.

Dump Results

Result are saved in text format in following format:
<segment_start> <segment_end> <speaker_label/silence>