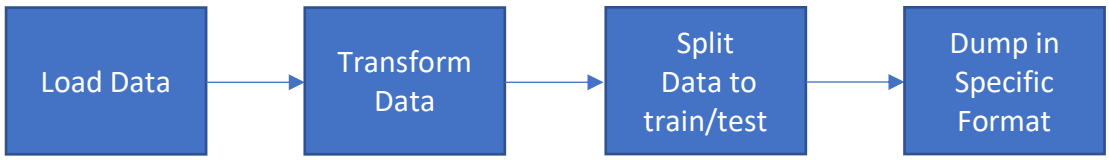
Task for Future Rider

Clone from github repository *https://github.com/Jamiroquai88/Rider-Candidate.git*
In this task, we want to transform input data using pre-trained transformations. Imagine audio recording with length of **1000** seconds segmented into **1000** chunks (every chunks is **1** second). 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). At first, we want to load this data, then we want to transform them. For our special purpose, we want to split this transformed data into 2 groups – **test** and **train** – and then we want to save them in specific text format shown below.

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*. Data are transformed as follows: *lda_matrix.dot(data)*

Data Clustering

We want to split our input data into **test** and **train**, each of them in separate files. We want to use 30 percent for **test** and 70 percent for **train**.

Dump Results

Result are saved in text format in following format:
<segment_start> <segment_end> <transformed_vector>