**Week 7 Report**

**Date:** 25.02.2025

**Student Name:** Abdullahi Jeylani

### **1. Completed Work This Week**

#### **1.1 Data Collection & Organization**

* Unfortunately no new voice recordings were collected this week due to the focus on data segmentation.
* Attempted to extract segments of the voice notes automatically using Python, but the approach was not efficient as it failed to produce accurate segmentation for different speech patterns.
* Decided to proceed with manual segmentation to ensure accuracy and consistency across all audio samples.
* Updated the metadata CSV file to include essential details such as participant ID, sentence spoken, timestamped audio segments, and corresponding file paths for better organization and retrieval.
* Postponed phonetic feature extraction until all audio files are properly segmented to maintain a structured workflow and avoid unnecessary reprocessing.

### **2. Findings & Insights**

* No new pronunciation patterns were identified this week due to the primary focus on segmentation.
* The initial attempt at automatic segmentation revealed inconsistencies, such as incorrect segment boundaries and overlap issues, leading to the decision to manually segment the data for better quality.
* The manual segmentation process confirmed that a uniform approach is needed to maintain consistency across all samples.

### **3. Research Review**

* Explored the Audacity application for audio segmentation, which provided a user-friendly and efficient tool for manual segmentation. Audacity allows for precise selection of speech segments and easy export into separate files, making it suitable for the project's needs.
* No significant research papers were reviewed this week specifically on speech segmentation techniques. However, future research may focus on refining segmentation methods to optimize the process.

### **4. Future Work**

* Implement a structured plan to complete the manual segmentation by processing at least 10 voice samples per day, with an estimated completion time of three weeks.
* Ensure that each segmented audio file is correctly labeled and stored systematically to facilitate efficient feature extraction.
* Once segmentation is complete, proceed with extracting key phonetic features such as MFCC (Mel-Frequency Cepstral Coefficients), formants, pitch, and energy to analyze pronunciation variations.
* Explore alternative automated segmentation techniques or machine learning approaches for potential improvements in future iterations.
* Review additional linguistic studies related to Somali accent characteristics to enhance the analysis phase of the project.

### **5. Segmentation code (not efficient)**

import os from pydub import AudioSegment from pydub.silence import split\_on\_silence

# Input and output directories

INPUT\_FOLDER = "raw\_audio"

OUTPUT\_FOLDER = "segmented\_audio"

# Ensure output folder exists

os.makedirs(OUTPUT\_FOLDER, exist\_ok=True)

def segment\_audio(file\_path, output\_folder, min\_silence\_len=300, silence\_thresh=-40): """ Splits an audio file into segments based on silence.

Parameters:  
- file\_path: Path to the input audio file.  
- output\_folder: Folder to save segmented words.  
- min\_silence\_len: Minimum length of silence (in ms) to be considered a split point.  
- silence\_thresh: Silence threshold in dB (lower values = more sensitive to noise).  
"""  
audio = AudioSegment.from\_file(file\_path)  
  
# Split based on silence  
segments = split\_on\_silence(audio, min\_silence\_len=min\_silence\_len, silence\_thresh=silence\_thresh)  
  
# Save each segment as a separate file  
base\_name = os.path.splitext(os.path.basename(file\_path))[0]  
for i, segment in enumerate(segments):  
 segment.export(f"{output\_folder}/{base\_name}\_part{i+1}.wav", format="wav")  
 print(f"Saved: {output\_folder}/{base\_name}\_part{i+1}.wav")

# Process all files in the input folder

for file\_name in os.listdir(INPUT\_FOLDER): if file\_name.endswith((".wav", ".mp3")): print(f"Processing {file\_name}...") segment\_audio(os.path.join(INPUT\_FOLDER, file\_name), OUTPUT\_FOLDER)

print("Segmentation complete!")