Speaker Diarization Using Deep Learning and pyannote.audio

Abhimanyu Kumar
22B1257
Department of Electrical Engineering, IIT Bombay
Supervisor: Prof. Preeti Rao

November 26, 2025

Abstract

Speaker diarization refers to partitioning an audio stream into homogeneous segments according to speaker identity. This report presents a hybrid pipeline integrating Voice Activity Detection, LSTM-based speaker change detection, and pyannote.audio embeddings evaluated on a real-world dataset.Used two clusttering algorithem:- **1.Mean Shift clusttering** and **2. K-mean clusttering** and compared it's results along with speaker diarization.

1 Introduction

Speaker diarization addresses the challenge of determining "Who spoke when?" in audio recordings. Applications span meeting transcription, broadcast media, and conversational analysis.

2 Dataset

Experiments were conducted on the **2.audio6min.wav** dataset.

- Audio prepared using **audacity**
- **Number of speakers: 3**
- **audio duration is 6 minutes.**
- Annotation format: RTTM
- Sampling rate: 16kHz WAV

3 Preprocessing & Voice Activity Detection

- Stereo audio is converted to mono using pydub, resampled to 16kHz, and segmented via webrtcvad.

```
sound = AudioSegment.from_wav(stereo_audio_path)
sound = sound.set_channels(1)
sound = sound.set_frame_rate(16000)
sound.export(mono_audio_path, format="wav")
```

- Frame generated and pulse code modulation to convert analog voice to digital**(pcm_data = wf.readframes(wf.getnframes())**.
- VAD output is a NumPy array of time intervals (seconds) where speech is detected in the audio.

4 Speaker Change Detection

- Feature extraction (uses 11 MFCC coefficients)is performed using MFCCs along with their first- and second-order derivatives (Δ and $\Delta\Delta$), followed by normalization.resulting in 35-dimensional features per frame
- A bidirectional LSTM-based sequence model is trained for frame-level speaker change detection and saved as "my_trained_model.h5".
- During inference, this pretrained bidirectional LSTM model is loaded and applied to unseen audio to estimate, for each frame, the probability of a speaker change and to localize speaker change points.

```
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(TimeDistributed(Dense(32)))
model.add(TimeDistributed(Dense(1, activation='sigmoid')))
```

5 Embedding Extraction (pyannote.audio)

For each speech segment, a 512-dimensional speaker embedding is extracted using pyannote's pretrained models:

```
from pyannote.audio import Inference
embedding = inference.crop(file_dict, Segment(start, end))
```

These embeddings, $\mathbf{E}_i \in \mathbb{R}^{512}$, are used for clustering.

6 Clustering and Diarization Output

- Clustering Algorithems Used **1.MeanShift clusttering** and **2. Kmean clusttering**.
- Output is formatted as RTTM file for metric evaluation

7 Plots

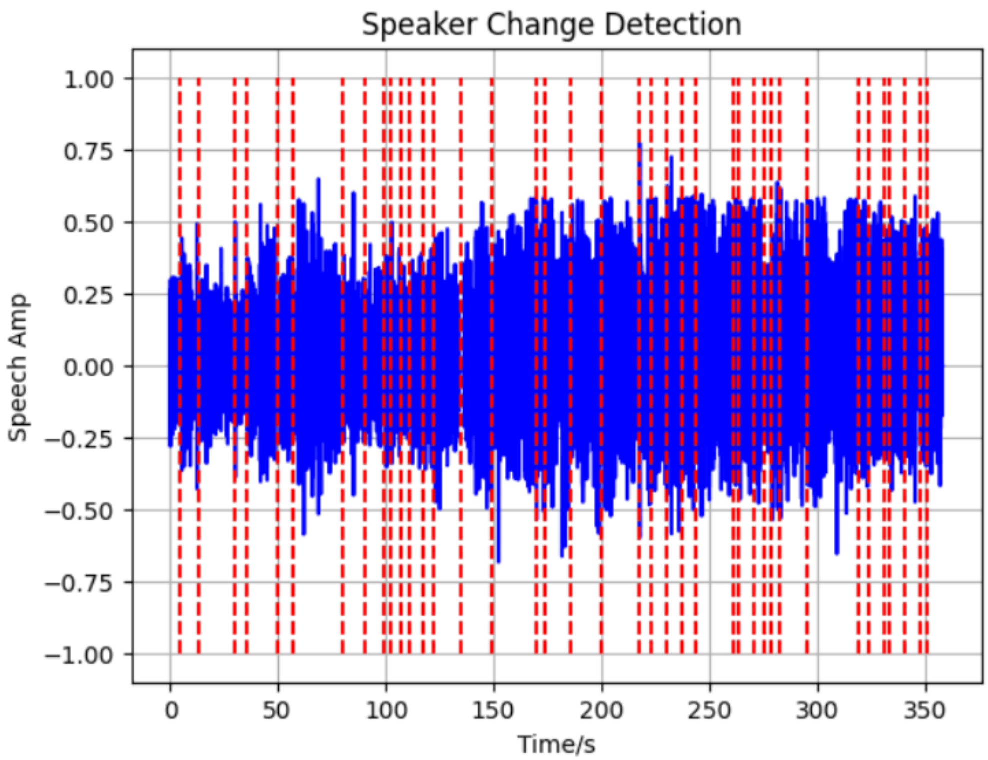

Figure 1: Speaker change detection

7.1 1. Using mean shift clustter

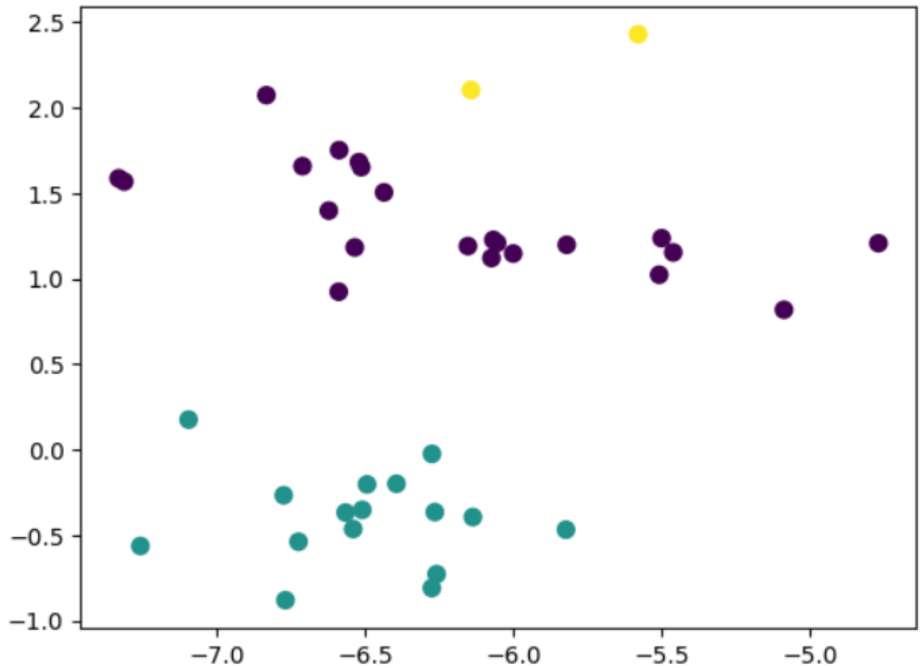

Figure 2: Speaker Embeddings Clustered via MeanShift

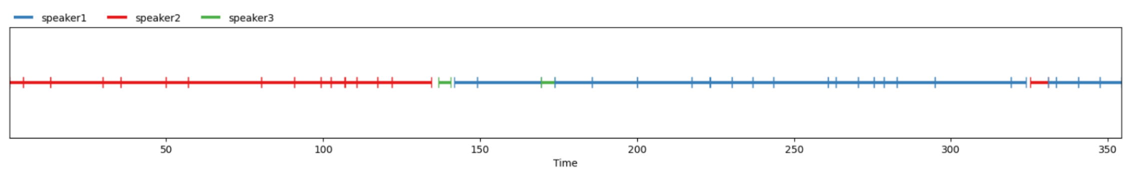

Figure 3: Preedicted speaker time interval

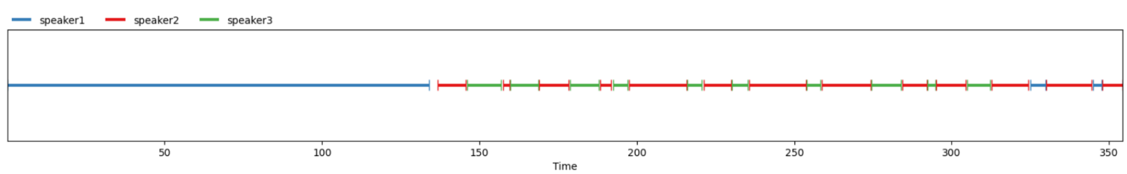

Figure 4: Reference speaker time interval

7.2 Using k mean clustter

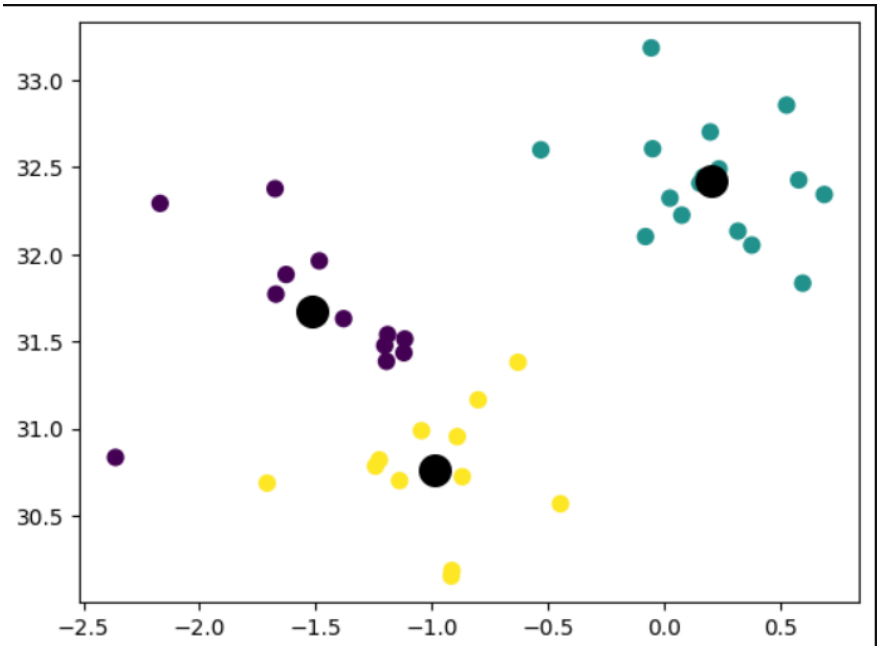

Figure 5: Speaker Embeddings Clustered via MeanShift

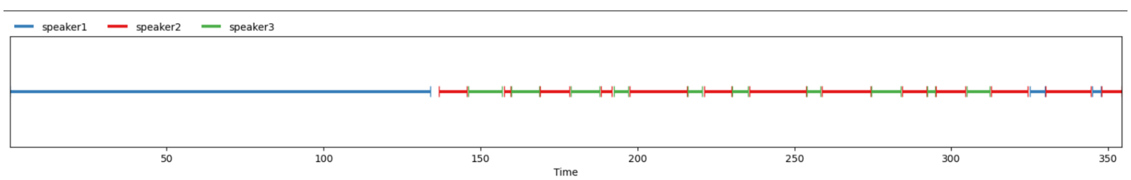

Figure 6: Preedicted speaker time interval

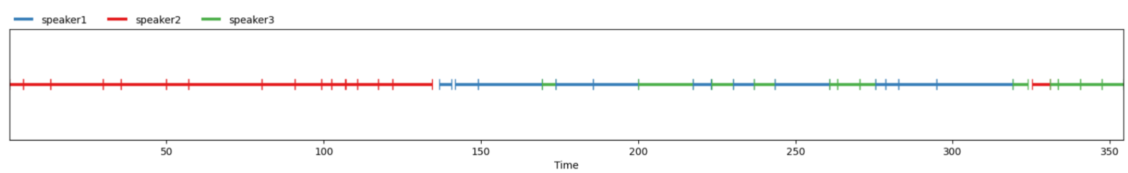

Figure 7: Reference speaker time interval

8 Results

Experimental Results for mean shift clustter

- In mean shift clustter, No required number of speaker.So using this , can be predict number of speaker also. But in K-mean clustter need number of speaker prior.
- Clusters found (Number of speaker): 3

Evaluation Scores for mean shift clustter

- Confusion:65.37 sec
- false alarm:7.62 sec
- missed detection:1.40 sec
- DER: [25.64%]

Experimental Results for k mean clustter

Evaluation Scores

- Confusion:51.77 sec
- false alarm:7.62 sec
- missed detection:1.4 sec
- DER: [20.95%]

8.0.1 "RTTM" for Mean shift clustter in page 6.

8.0.2 "RTTM" for K-Mean clustter in page 7.

Table 1: Reference Timeline for mean shift

Serial No.	Speaker	Start (s)	End (s)
1	speaker1	0.00	134.08
2	speaker2	136.69	145.62
3	speaker3	146.08	156.87
4	speaker2	157.56	159.50
5	speaker3	159.83	168.67
6	speaker2	168.96	178.39
7	speaker3	178.89	187.81
8	speaker2	188.35	191.73
9	speaker3	192.50	197.08
10	speaker2	197.42	215.85
11	speaker3	216.08	220.66
12	speaker2	221.35	229.93
13	speaker3	230.12	235.31
14	speaker2	235.58	253.77
15	speaker3	254.00	258.27
16	speaker2	258.69	274.23
17	speaker3	274.46	283.85
18	speaker2	284.35	292.16
19	speaker3	292.43	294.85
20	speaker2	295.12	304.58
21	speaker3	304.93	312.23
22	speaker2	312.73	324.31
23	speaker1	325.00	329.81
24	speaker2	329.96	344.43
25	speaker1	345.00	347.77
26	speaker2	347.81	354.50
27	speaker1	354.68	358.04

Table 2: Predicted Timeline for mean shift

Serial No.	Speaker	Start (s)	End (s)
1	speaker2	0.07	4.37
2	speaker2	4.38	13.11
3	speaker2	13.12	29.88
4	speaker2	29.89	35.48
5	speaker2	35.49	49.85
6	speaker2	49.86	56.92
7	speaker2	56.93	80.18
8	speaker2	80.19	90.71
9	speaker2	90.72	99.35
10	speaker2	99.36	102.39
11	speaker2	102.40	106.90
12	speaker2	106.91	110.74
13	speaker2	110.75	117.27
14	speaker2	117.28	121.94
15	speaker2	121.95	134.44
16	speaker3	136.64	140.67
17	speaker1	141.70	149.17
18	speaker1	149.18	169.46
19	speaker1	169.47	173.75
20	speaker1	173.76	185.56
21	speaker1	185.57	200.05
22	speaker1	200.06	217.33
23	speaker1	217.34	223.22
24	speaker1	223.23	230.29
25	speaker1	230.30	236.92
26	speaker1	236.93	243.41
27	speaker1	243.42	260.82
28	speaker3	260.83	263.32
29	speaker1	263.33	270.42
30	speaker1	270.43	275.45
31	speaker1	275.46	278.61
32	speaker1	278.62	282.71
33	speaker1	282.72	294.90
34	speaker1	294.91	319.06
35	speaker1	319.07	323.95
36	speaker2	325.38	330.93
37	speaker3	330.94	333.46
38	speaker1	333.47	340.57
39	speaker1	340.58	347.38
40	speaker1	347.39	354.29

Table 3: Reference Timeline for K-mean

Serial No.	Speaker	Start (s)	End (s)
1	speaker1	0.00	134.08
2	speaker2	136.69	145.62
3	speaker3	146.08	156.87
4	speaker2	157.56	159.50
5	speaker3	159.83	168.67
6	speaker2	168.96	178.39
7	speaker3	178.89	187.81
8	speaker2	188.35	191.73
9	speaker3	192.50	197.08
10	speaker2	197.42	215.85
11	speaker3	216.08	220.66
12	speaker2	221.35	229.93
13	speaker3	230.12	235.31
14	speaker2	235.58	253.77
15	speaker3	254.00	258.27
16	speaker2	258.69	274.23
17	speaker3	274.46	283.85
18	speaker2	284.35	292.16
19	speaker3	292.43	294.85
20	speaker2	295.12	304.58
21	speaker3	304.93	312.23
22	speaker2	312.73	324.31
23	speaker1	325.00	329.81
24	speaker2	329.96	344.43
25	speaker1	345.00	347.77
26	speaker2	347.81	354.50
27	speaker1	354.68	358.04

Table 4: Predicted Timeline for K-mean

Serial No.	Speaker	Start (s)	End (s)
1	speaker2	0.07	4.37
2	speaker2	4.38	13.11
3	speaker2	13.12	29.88
4	speaker2	29.89	35.48
5	speaker2	35.49	49.85
6	speaker2	49.86	56.92
7	speaker2	56.93	80.18
8	speaker2	80.19	90.71
9	speaker2	90.72	99.35
10	speaker2	99.36	102.39
11	speaker2	102.40	106.90
12	speaker2	106.91	110.74
13	speaker2	110.75	117.27
14	speaker2	117.28	121.94
15	speaker2	121.95	134.44
16	speaker1	136.64	140.67
17	speaker1	141.70	149.17
18	speaker1	149.18	169.46
19	speaker3	169.47	173.75
20	speaker1	173.76	185.56
21	speaker1	185.57	200.05
22	speaker3	200.06	217.33
23	speaker1	217.34	223.22
24	speaker3	223.23	230.29
25	speaker1	230.30	236.92
26	speaker3	236.93	243.41
27	speaker1	243.42	260.82
28	speaker3	260.83	263.32
29	speaker3	263.33	270.42
30	speaker3	270.43	275.45
31	speaker1	275.46	278.61
32	speaker1	278.62	282.71
33	speaker1	282.72	294.90
34	speaker1	294.91	319.06
35	speaker3	319.07	323.95
36	speaker2	325.38	330.93
37	speaker3	330.94	333.46
38	speaker3	333.47	340.57
39	speaker3	340.58	347.38
40	speaker3	347.39	354.29

9 Challenges

- Accurate handling of overlapping speech
- Robustness in low-resource/noisy environments
- Scalability to long recording and large datasets

10 Conclusion

This report presented a hybrid speaker diarization pipeline that integrates Voice Activity Detection (VAD), LSTM-based speaker change detection using MFCCs with delta features, and pyannote.audio neural embeddings for robust speaker representation. The system was evaluated on a 6-minute audio recording containing 3 speakers, comparing two clustering approaches: Mean Shift and K-means.

The experimental results demonstrate that K-means clustering achieves superior performance with a Diarization Error Rate (DER) of **20.95%**, compared to Mean Shift clustering which yielded a DER of **25.64%**. Both methods exhibited identical false alarm (7.62s) and missed detection (1.40s) rates, indicating that the performance difference stems primarily from speaker confusion errors, where K-means (51.77s) outperformed Mean Shift (65.37s) by reducing speaker label assignment errors.

11 References

- pyannote.audio documentation
- T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization Recent advances with deep learning," Computer Speech & Language, vol. 72, p. 101317, 2022.