Classifying Speakers in Audio Recordings

Rahul Jain & Earn Wonghirundacha

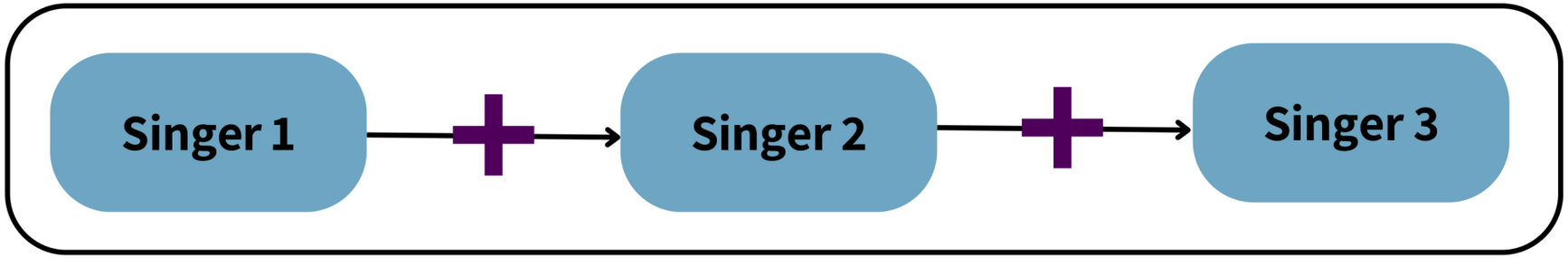
01

Problem Statement

Given an audio recording,
can we classify how many
people are singing?

Data

- **MedleyVox Dataset**
  - **21 duet songs, 42 singers**
- **Generated new dataset**



different solo voices concatenated together to form **one** audio clip

02

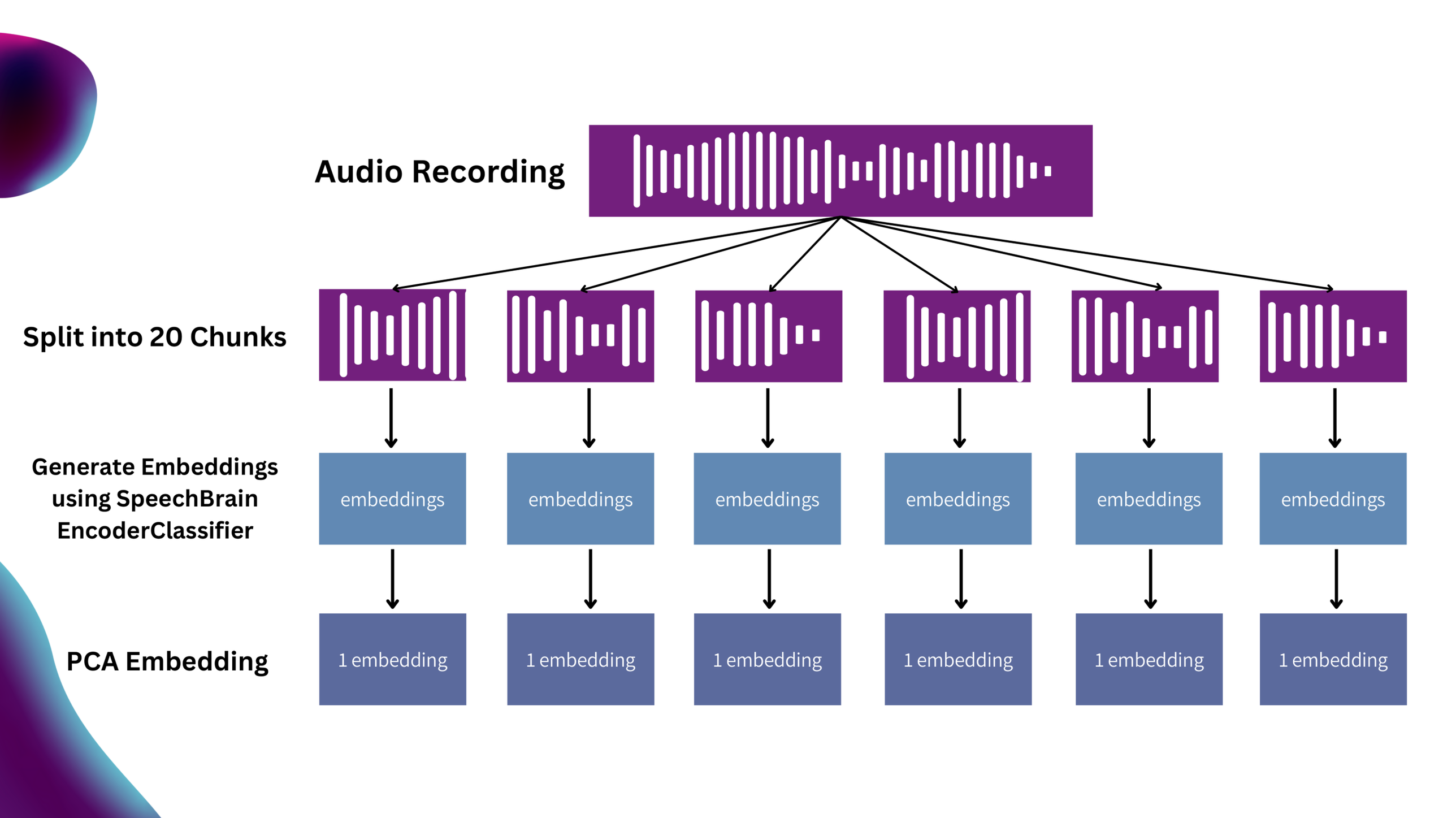
System Design

Approach 1: Clustering

Approach 2: Pre-Trained Speaker Diarization

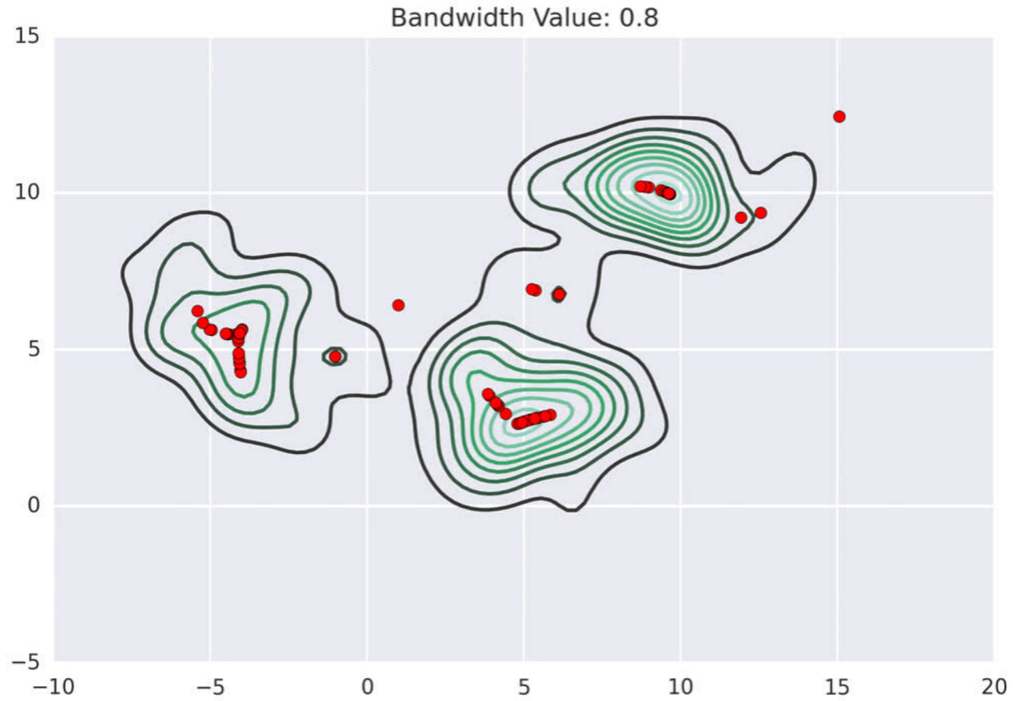
Approach 3: RNN

Benchmark Model Approach 1: Generate Embeddings and Cluster


Audio Recording
Split into 20 Chunks
Generate Embeddings using SpeechBrain EncoderClassifier
PCA Embedding
embeddings
embeddings
embeddings
embeddings
embeddings
embeddings
1 embedding
1 embedding
1 embedding
1 embedding
1 embedding
1 embedding

Mean Shift Clustering

1. Clustering without specify how many clusters there are in advance
2. Used Principle Component Analysis to simplify the data
3. Estimate the bandwidth with a quantile of 0.5



Benchmark Model Approach 2: Pre-trained Pyannote Speaker Diarization Model

Speaker Diarization Classification

- Open-source toolkit used for speaker diarization
- Partitioning an audio stream into segments according to the identity of the speaker
- Give the model our entire .wav file
- Take the output of the the diarization model
- Classify how many speakers there are

Pipeline

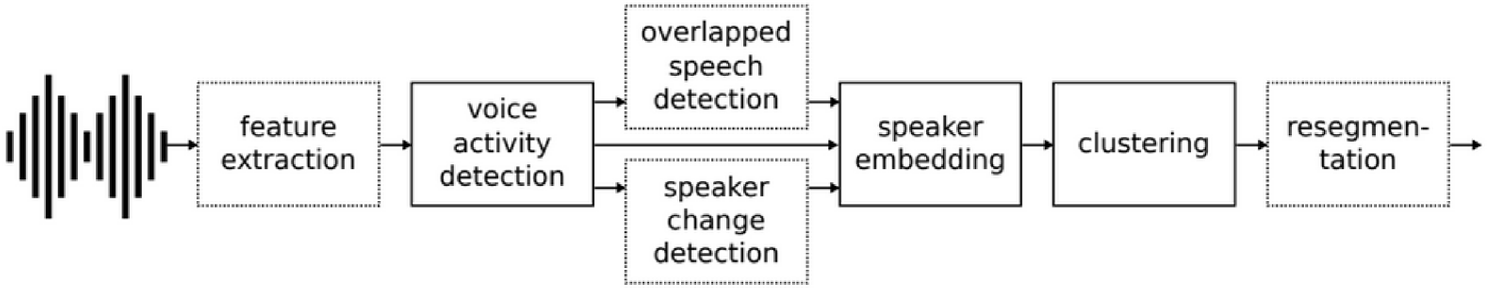


Fig. 1. `pyannote.audio` provides a collection of modules that can be jointly optimized to build a speaker diarization pipeline.

Example Output

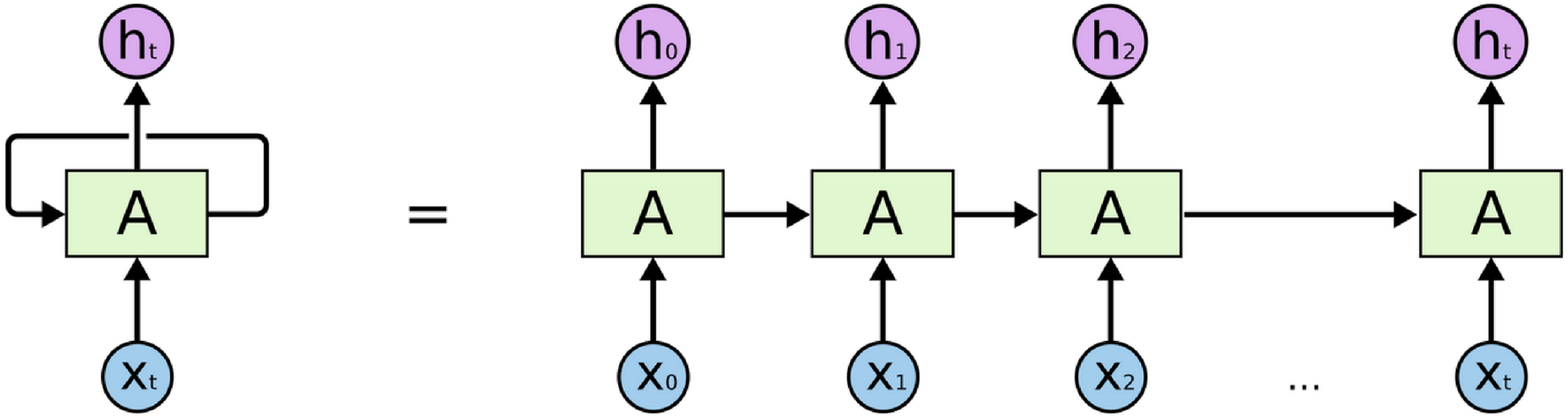
```
start=0.0s stop=0.0s speaker_SPEAKER_01
start=0.0s stop=0.7s speaker_SPEAKER_04
start=0.7s stop=5.0s speaker_SPEAKER_01
start=5.2s stop=7.8s speaker_SPEAKER_00
start=8.5s stop=10.2s speaker_SPEAKER_01
start=10.5s stop=12.0s speaker_SPEAKER_01
start=12.0s stop=13.3s speaker_SPEAKER_00
start=13.3s stop=14.0s speaker_SPEAKER_02
start=14.0s stop=14.5s speaker_SPEAKER_03
start=14.5s stop=15.0s speaker_SPEAKER_00
Number of Singers: 5
Number of Predicted Singers: 5
```

```
start=0.9s stop=3.0s speaker_SPEAKER_00
start=3.1s stop=6.6s speaker_SPEAKER_01
Number of Singers: 2
Number of Predicted Singers: 2
```

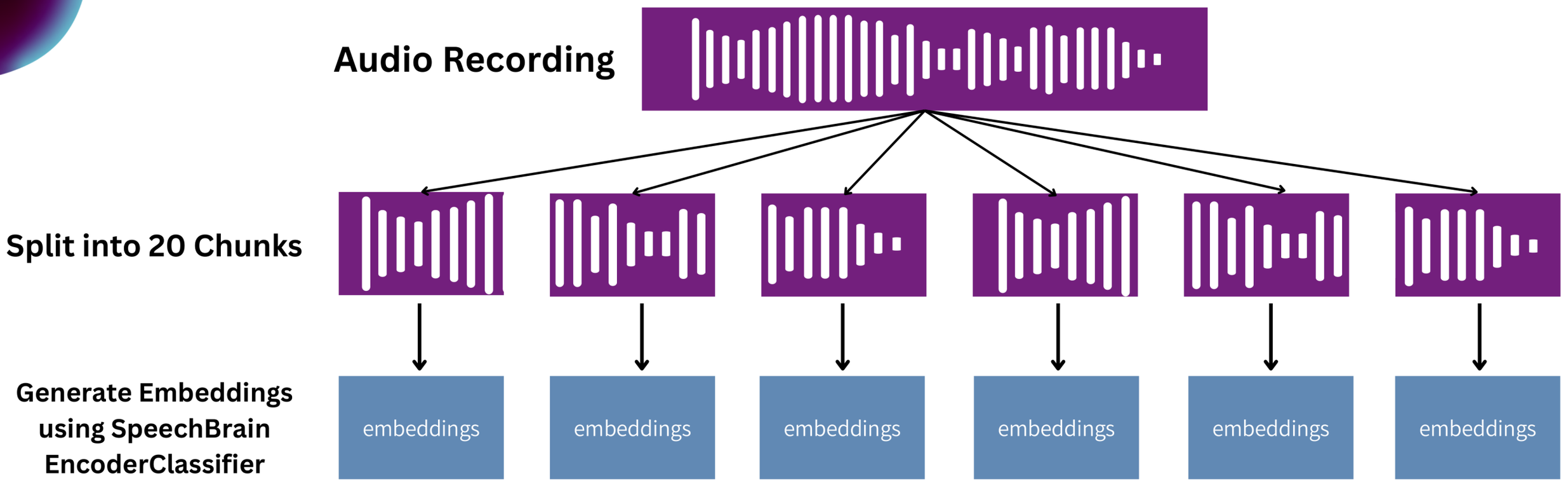
Main Approach
Approach 3: Detecting Singer Change with RNN

Main Idea

- RNNs process data sequentially (very important for audio)
- Compare embeddings at time t to embeddings in the past
- Idea is that the RNN can learn to predict where the singer switches
- Count the number of switches at the end



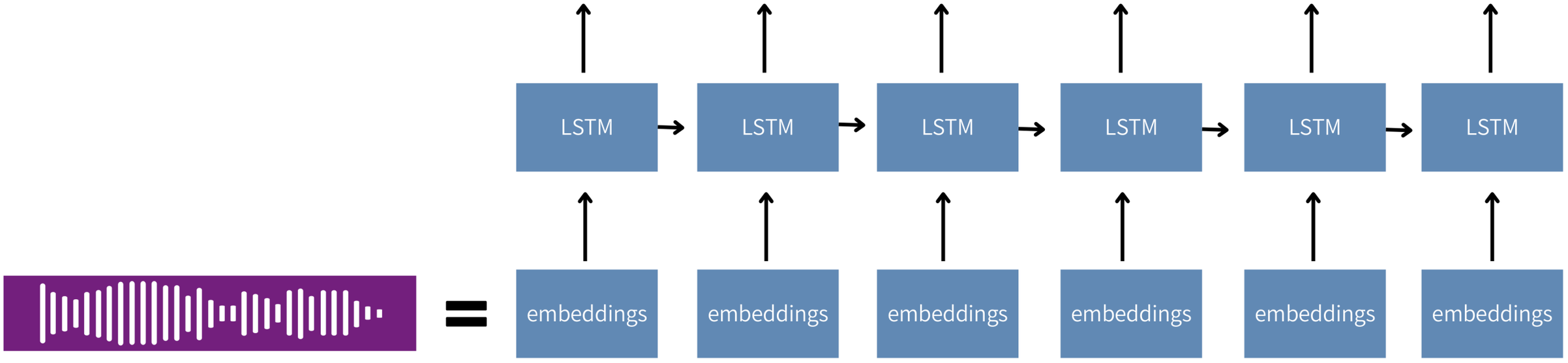
Pre-processing



Created 200 “songs”: 80% training, 20% testing

Method

y_hat = sum([1, 0, 0, 1, 0, 0])



y_true = sum([1, 0, 0, 1, 0, 0])

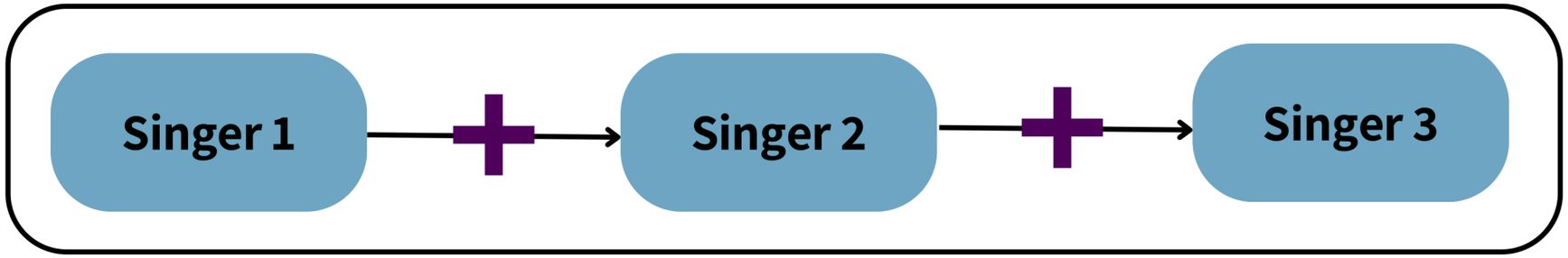
03

Experimental Setup & Results

First Iteration

2 Singers with no randomization

Data: No randomization



≈ 5 seconds ≈ 5 seconds ≈ 5 seconds

No Randomization

Approach 1: Mean Shift Clustering

No PCA

Accuracy: 20.13%

```
Cluster Assignments: [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
Estimated Number of Singers: 2
Cluster Assignments: [0 0 0 0 0 0 0 0 0 0 0 0 3 4 2 1 0 0 0 0]
Estimated Number of Singers: 5
```

5 Components

Accuracy: 29.16%

2 Components

Accuracy: 38.88%

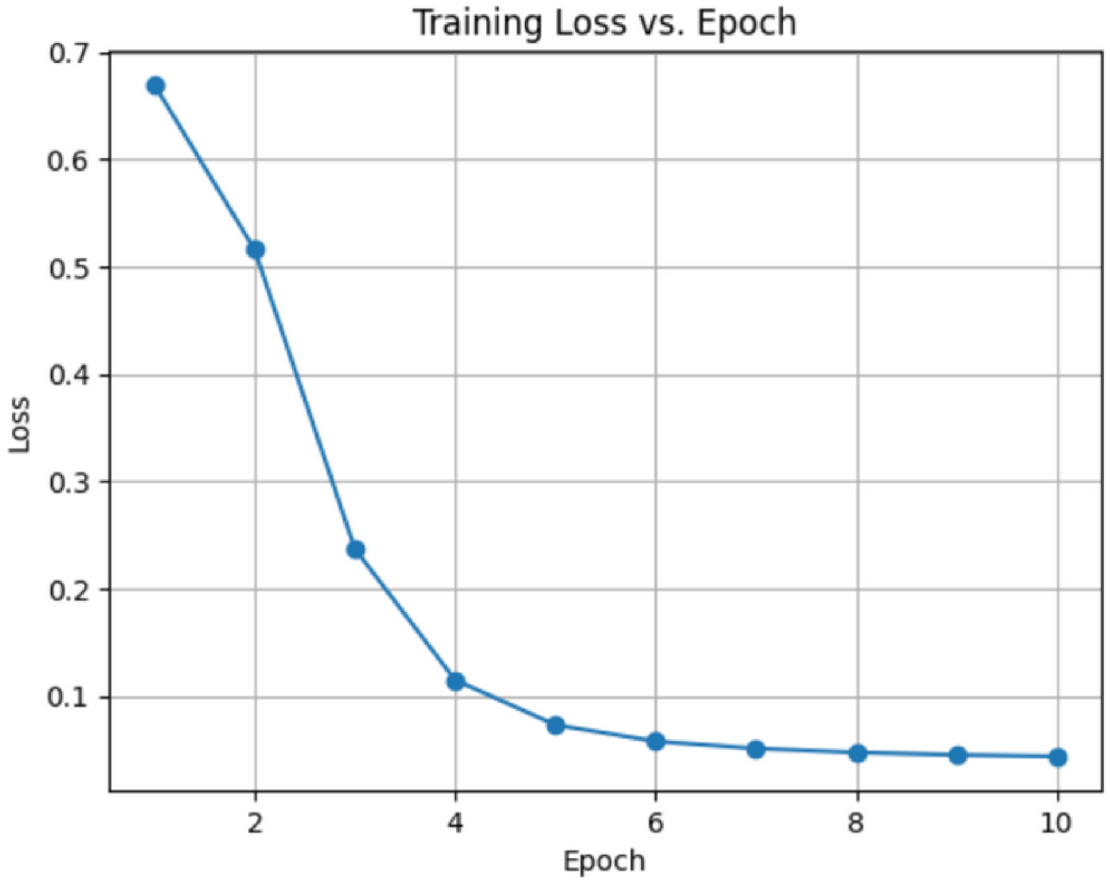
Accuracy: 75.69%

```
Cluster Assignments: [2 0 1 0 0 0 0 1 1 2 0 0 0 1 0 1 1 1 1 1]
Estimated Number of Singers: 3
Cluster Assignments: [0 0 0 0 0 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0]
Estimated Number of Singers: 2
Cluster Assignments: [0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0]
Estimated Number of Singers: 2
```

No Randomization

Approach 3: RNN

- Currently testing with 2 singers
  - Only one switch
- Still need to run more tests on 3-8 singers



Problems

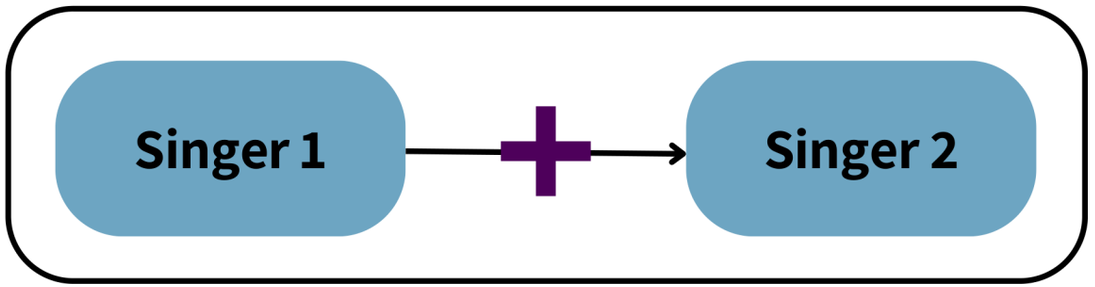
- Curve is way too smooth
- Is it only predicting zeros?
- Is it only predicting 1s every 5 seconds?
- 99.7% accuracy which seems very fishy

Second Iteration

2 Singers with randomization

Improvement #1

Randomizing split



≈ 2 seconds

≈ 3 seconds

Benchmarks

1) Clustering

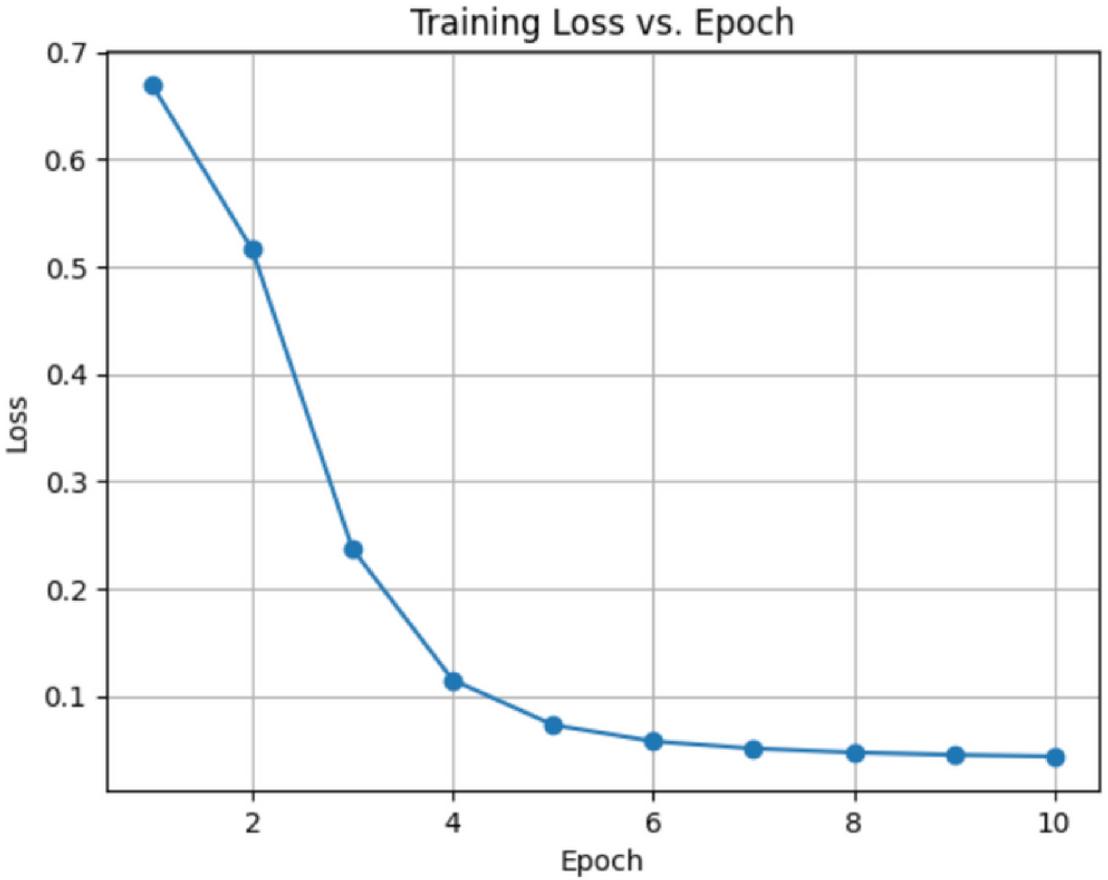
1 Component, Quantile = 0.5 Accuracy: 72.5%

2) Speaker Diarization

Accuracy: 45.16%

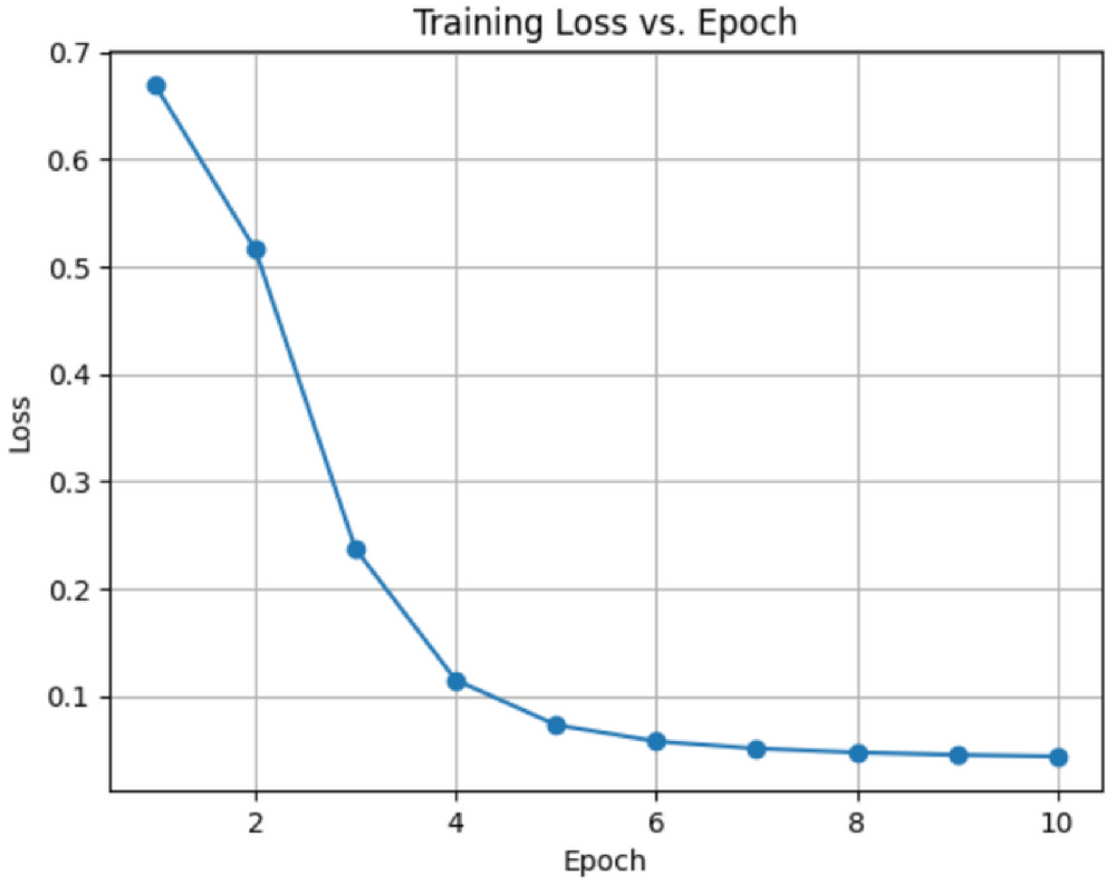
Testing with 2 singers and one randomized switch

95% accurate..?



Testing with 2 singers and one randomized switch

95% accurate..?




```
000. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
001. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
002. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
003. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
004. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
005. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
006. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
007. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
008. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
009. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
010. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
011. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
012. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
013. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
014. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
015. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
016. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
017. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
018. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
019. [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  predicted: 1
```

Third Iteration

2- 8 Singers with randomization and smaller chunks

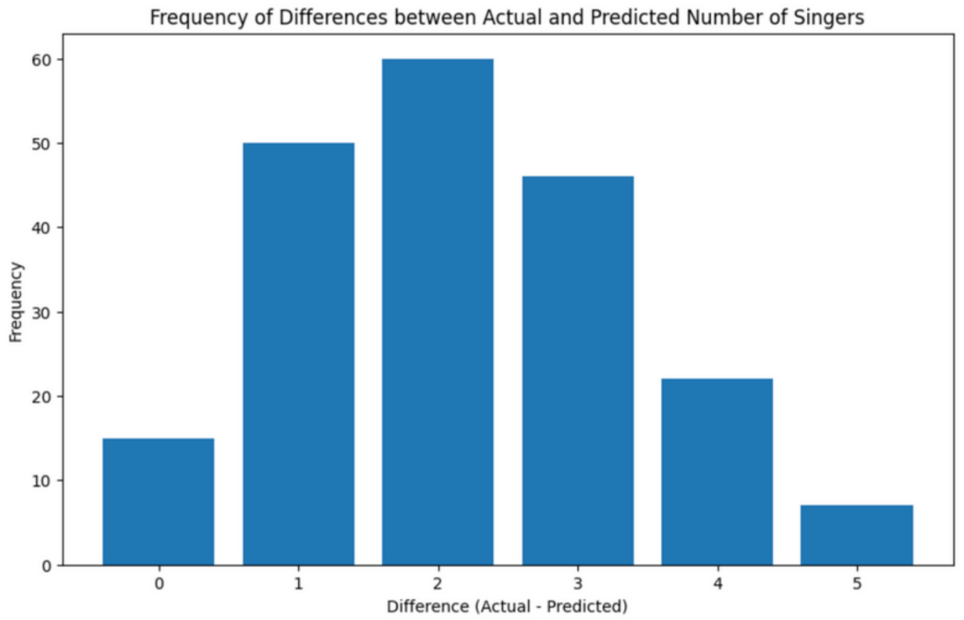
Benchmarks

1) Clustering

3 Component, Quantile = 0.25 Accuracy: 18%

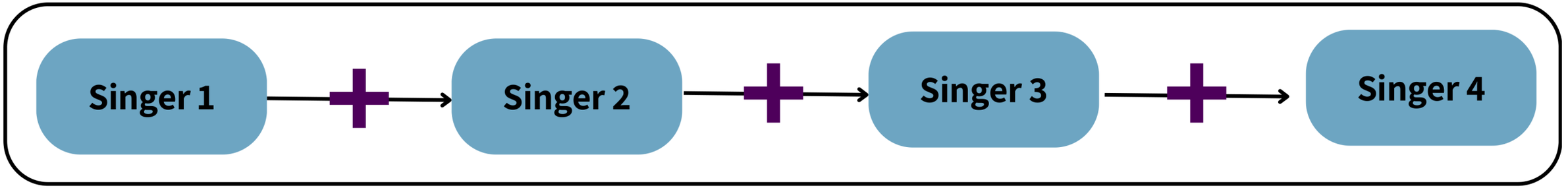
2) Speaker Diarization

Accuracy: 7.5%



Improvement #2

Randomizing # singers (2-7)



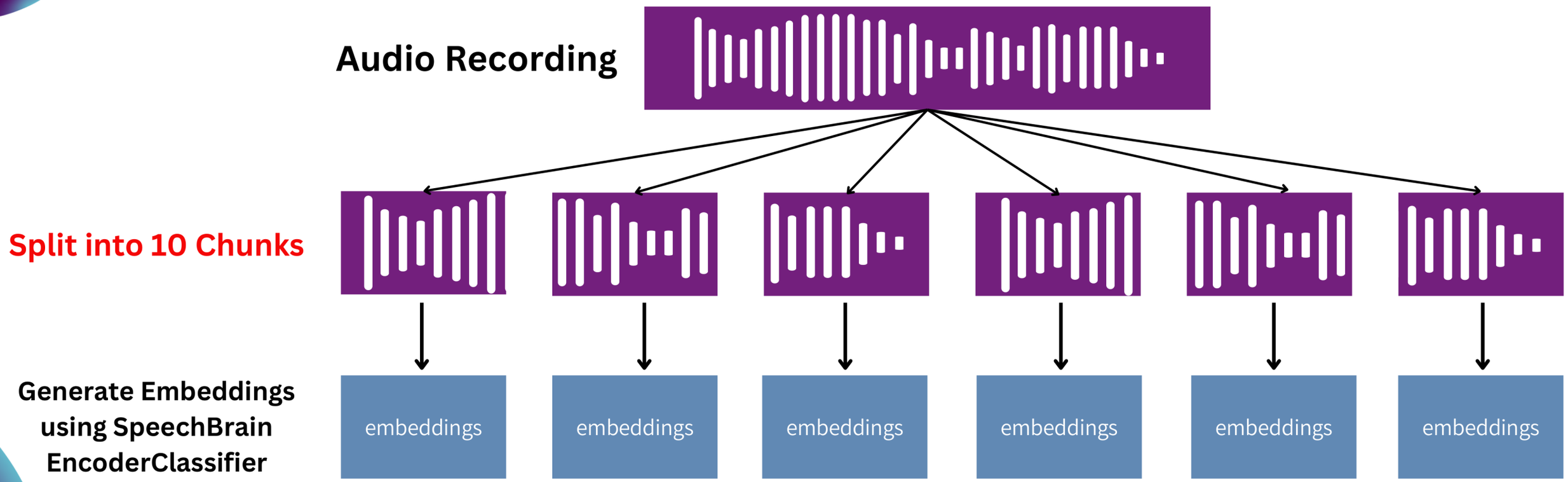
≈ 2 seconds

≈ 3 seconds

≈ 4 seconds

≈ 6 seconds

Improvement #3



New Dataset with:

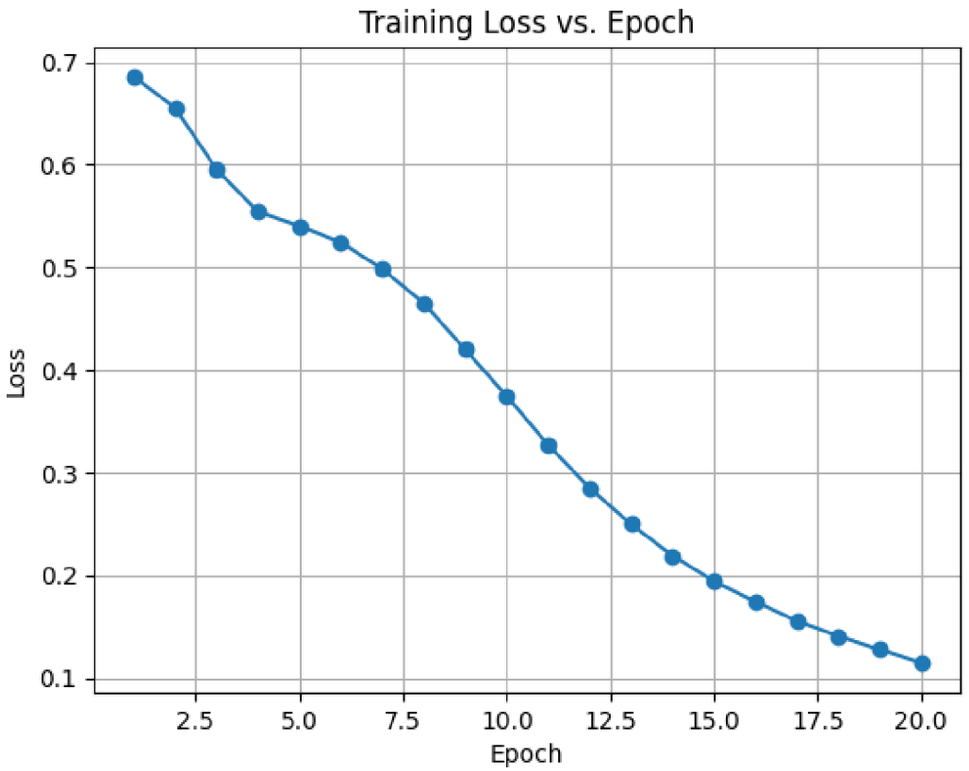
- Randomized split length (between 2-5 seconds)
- Randomized number of singers (2-8)
- 10 chunks, instead of 20

```
loaded training_songs:
  000. combined_output_4_singers_20231209072028703087.wav has 04 singers with switches at [0, 3526, 7928, 11422] and 10 chunks
  001. combined_output_4_singers_20231209072029283782.wav has 04 singers with switches at [0, 4154, 8734, 12609] and 10 chunks
  002. combined_output_6_singers_20231209072030025047.wav has 06 singers with switches at [0, 3875, 6167, 11091, 15048, 18932] and 10 chunks
  003. combined_output_5_singers_20231209072030932043.wav has 05 singers with switches at [0, 4226, 6914, 11191, 15351] and 10 chunks
  004. combined_output_6_singers_20231209072031699817.wav has 06 singers with switches at [0, 4016, 8787, 13686, 17269, 22194] and 10 chunks
  005. combined_output_3_singers_20231209072032663755.wav has 03 singers with switches at [0, 3493, 7339] and 10 chunks
  006. combined_output_5_singers_20231209072033166257.wav has 05 singers with switches at [0, 3112, 6608, 11395, 15480] and 10 chunks
  007. combined_output_2_singers_20231209072034031946.wav has 02 singers with switches at [0, 3757] and 10 chunks
  008. combined_output_4_singers_20231209072034509299.wav has 04 singers with switches at [0, 3622, 8195, 10883] and 10 chunks
  009. combined_output_3_singers_20231209072035524977.wav has 03 singers with switches at [0, 4705, 6799] and 10 chunks
  010. combined_output_4_singers_20231209072036205872.wav has 04 singers with switches at [0, 2597, 4691, 8964] and 10 chunks
  011. combined_output_2_singers_20231209072036924873.wav has 02 singers with switches at [0, 2688] and 10 chunks
  012. combined_output_7_singers_20231209072037545651.wav has 07 singers with switches at [0, 3635, 8352, 11677, 16472, 19305, 21767] and 10 chunks
  013. combined_output_3_singers_20231209072039247148.wav has 03 singers with switches at [0, 3224, 4932] and 10 chunks
  014. combined_output_6_singers_20231209072039888129.wav has 06 singers with switches at [0, 4541, 8437, 12846, 15165, 17040] and 10 chunks
  015. combined_output_3_singers_20231209072041171958.wav has 03 singers with switches at [0, 3640, 6734] and 10 chunks
  016. combined_output_6_singers_20231209072041649619.wav has 06 singers with switches at [0, 2698, 5016, 7097, 8805, 12045] and 10 chunks
  017. combined_output_2_singers_20231209072042492627.wav has 02 singers with switches at [0, 2200] and 10 chunks
  018. combined_output_7_singers_20231209072042860487.wav has 07 singers with switches at [0, 4031, 8074, 11205, 15281, 19543, 22667] and 10 chunks
  019. combined_output_6_singers_20231209072043924305.wav has 06 singers with switches at [0, 4797, 7782, 9907, 12879, 15841] and 10 chunks
```

Training Data:

```
song 000. embeddings shape: (10, 512)   switches: [1 0 1 0 0 1 0 0 1 0]
song 001. embeddings shape: (10, 512)   switches: [1 0 1 0 0 1 0 1 0 0]
song 002. embeddings shape: (10, 512)   switches: [1 1 1 0 1 0 1 0 1 0]
song 003. embeddings shape: (10, 512)   switches: [1 0 1 1 0 0 1 0 1 0]
song 004. embeddings shape: (10, 512)   switches: [1 1 0 1 0 1 1 0 1 0]
song 005. embeddings shape: (10, 512)   switches: [1 0 0 1 0 0 1 0 0 0]
song 006. embeddings shape: (10, 512)   switches: [1 1 0 1 0 0 1 0 1 0]
song 007. embeddings shape: (10, 512)   switches: [1 0 0 0 0 0 1 0 0 0]
song 008. embeddings shape: (10, 512)   switches: [1 0 1 0 0 1 1 0 0 0]
song 009. embeddings shape: (10, 512)   switches: [1 0 0 0 0 1 0 1 0 0]
song 010. embeddings shape: (10, 512)   switches: [1 0 1 0 1 0 0 1 0 0]
song 011. embeddings shape: (10, 512)   switches: [1 0 0 1 0 0 0 0 0 0]
song 012. embeddings shape: (10, 512)   switches: [1 1 0 1 1 0 1 1 1 0]
song 013. embeddings shape: (10, 512)   switches: [1 0 0 1 0 1 0 0 0 0]
song 014. embeddings shape: (10, 512)   switches: [1 0 1 1 0 1 1 1 0 0]
song 015. embeddings shape: (10, 512)   switches: [1 0 0 1 0 0 0 1 0 0]
song 016. embeddings shape: (10, 512)   switches: [1 1 0 1 1 1 0 1 0 0]
song 017. embeddings shape: (10, 512)   switches: [1 0 0 1 0 0 0 0 0 0]
song 018. embeddings shape: (10, 512)   switches: [1 1 1 0 1 1 0 1 1 0]
song 019. embeddings shape: (10, 512)   switches: [1 0 1 0 1 1 1 0 1 0]
```

Training Loss:



Accuracy:

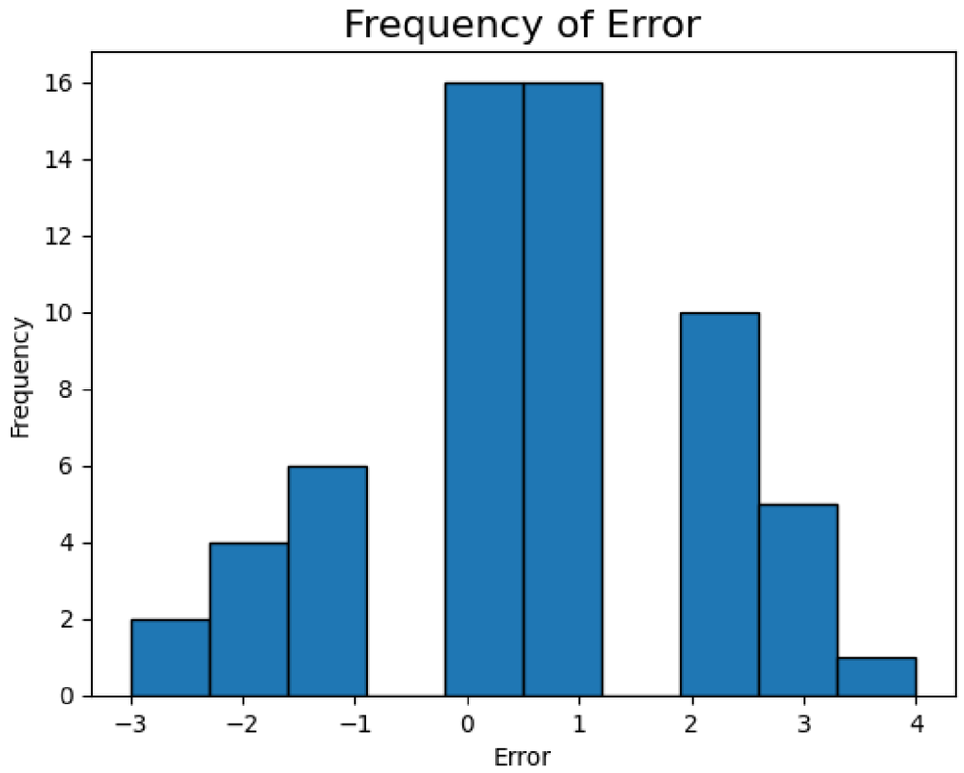
```
001. [1 1 0 1 1 1 0 1 0 0]     predicted: 6    actual: 3      diff: -3
002. [1 1 0 1 0 0 1 1 0 0]     predicted: 5    actual: 6      diff: 1
003. [1 0 1 0 1 0 0 0 0 0]     predicted: 3    actual: 2      diff: -1
004. [1 0 0 1 0 0 0 1 0 0]     predicted: 3    actual: 2      diff: -1
005. [1 0 1 0 0 1 0 0 0 0]     predicted: 3    actual: 6      diff: 3
006. [1 1 0 0 1 0 1 0 1 0]     predicted: 5    actual: 6      diff: 1
007. [1 0 0 1 0 0 1 0 0 0]     predicted: 3    actual: 5      diff: 2
008. [1 0 0 0 1 0 0 0 0 0]     predicted: 2    actual: 2      diff: 0
009. [1 1 0 1 1 0 1 1 0 0]     predicted: 6    actual: 5      diff: -1
010. [1 0 0 0 1 0 1 0 0 0]     predicted: 3    actual: 5      diff: 2
011. [1 1 0 1 0 0 0 1 0 0]     predicted: 4    actual: 6      diff: 2
012. [1 1 1 0 1 1 1 0 1 0]     predicted: 7    actual: 7      diff: 0
013. [1 0 1 0 1 0 1 0 1 0]     predicted: 5    actual: 7      diff: 2
014. [1 0 0 0 1 1 0 0 0 0]     predicted: 3    actual: 3      diff: 0
015. [1 0 1 0 0 1 1 0 1 0]     predicted: 5    actual: 6      diff: 1
016. [1 0 1 1 0 1 0 1 0 0]     predicted: 5    actual: 5      diff: 0
017. [1 0 0 0 1 0 0 1 0 0]     predicted: 3    actual: 3      diff: 0
018. [1 0 1 1 0 1 1 0 1 0]     predicted: 6    actual: 5      diff: -1
019. [1 0 1 0 1 1 0 0 0 0]     predicted: 4    actual: 2      diff: -2
020. [1 0 0 0 1 0 0 0 0 0]     predicted: 2    actual: 2      diff: 0
021. [1 0 1 0 0 0 0 0 0 0]     predicted: 2    actual: 2      diff: 0
022. [1 1 1 0 1 0 1 0 1 0]     predicted: 6    actual: 4      diff: -2
023. [1 1 0 1 0 0 1 0 1 0]     predicted: 5    actual: 7      diff: 2
024. [1 0 1 0 0 1 1 0 1 0]     predicted: 5    actual: 7      diff: 2
025. [1 0 0 0 1 0 0 1 0 0]     predicted: 3    actual: 2      diff: -1
026. [1 0 0 1 0 0 1 1 0 0]     predicted: 4    actual: 6      diff: 2
027. [1 1 0 0 1 0 0 0 0 0]     predicted: 3    actual: 3      diff: 0
028. [1 0 0 0 0 0 0 0 0 0]     predicted: 1    actual: 3      diff: 2
029. [1 1 1 1 1 1 0 1 1 0]     predicted: 8    actual: 5      diff: -3
030. [1 1 0 1 1 1 1 0 1 0]     predicted: 7    actual: 7      diff: 0
031. [1 0 0 0 1 0 0 1 0 0]     predicted: 3    actual: 4      diff: 1
032. [1 0 0 1 0 0 0 0 0 0]     predicted: 2    actual: 3      diff: 1
033. [1 0 0 0 1 0 0 0 0 0]     predicted: 2    actual: 3      diff: 1
034. [1 1 1 1 1 0 1 0 1 0]     predicted: 7    actual: 5      diff: -2
035. [1 0 1 0 0 1 0 0 1 0]     predicted: 4    actual: 7      diff: 3
036. [1 1 0 1 0 1 1 1 0 0]     predicted: 6    actual: 4      diff: -2
037. [1 1 1 0 1 1 1 0 1 0]     predicted: 7    actual: 4      diff: -3
038. [1 0 1 0 0 1 1 0 1 0]     predicted: 5    actual: 7      diff: 2
039. [1 0 1 0 0 1 0 1 0 0]     predicted: 4    actual: 4      diff: 0
```

Chunk switches accuracy: 64.50%

Accuracy +/- 0: 46.67%

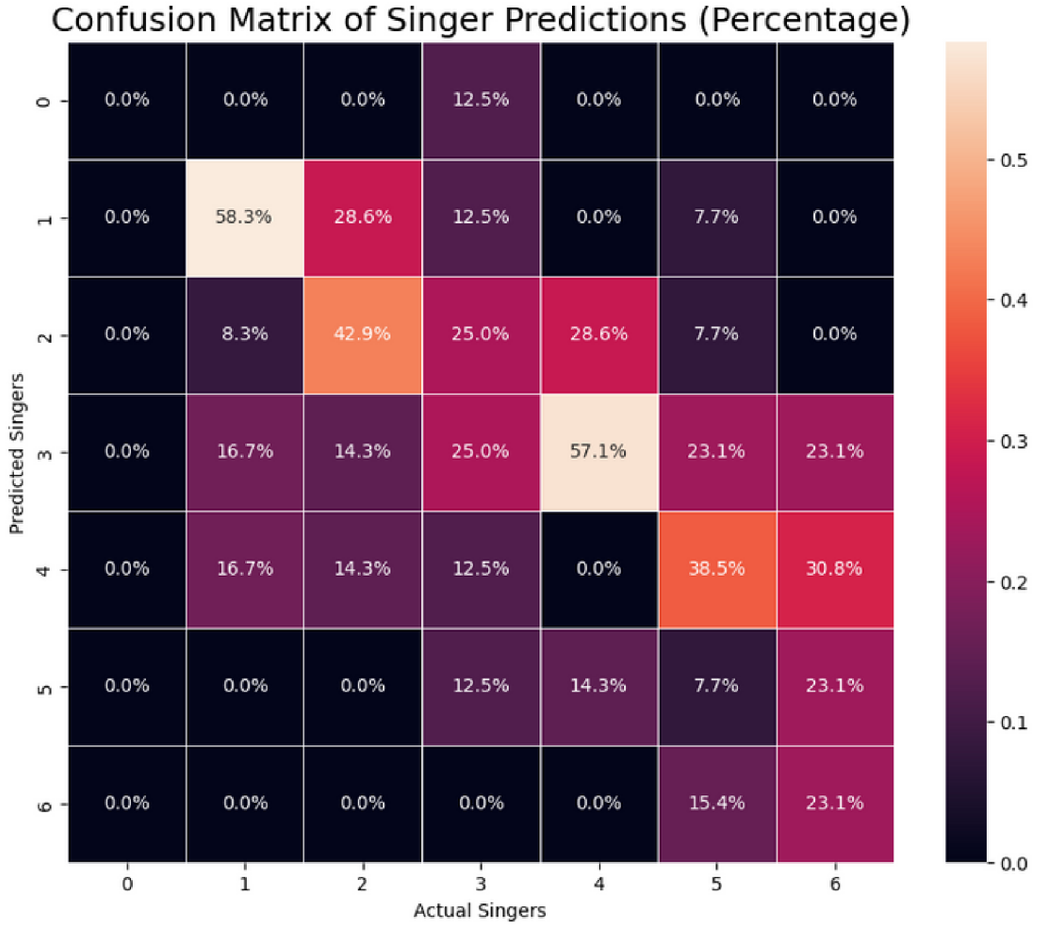
Accuracy +/- 1: 73.33%

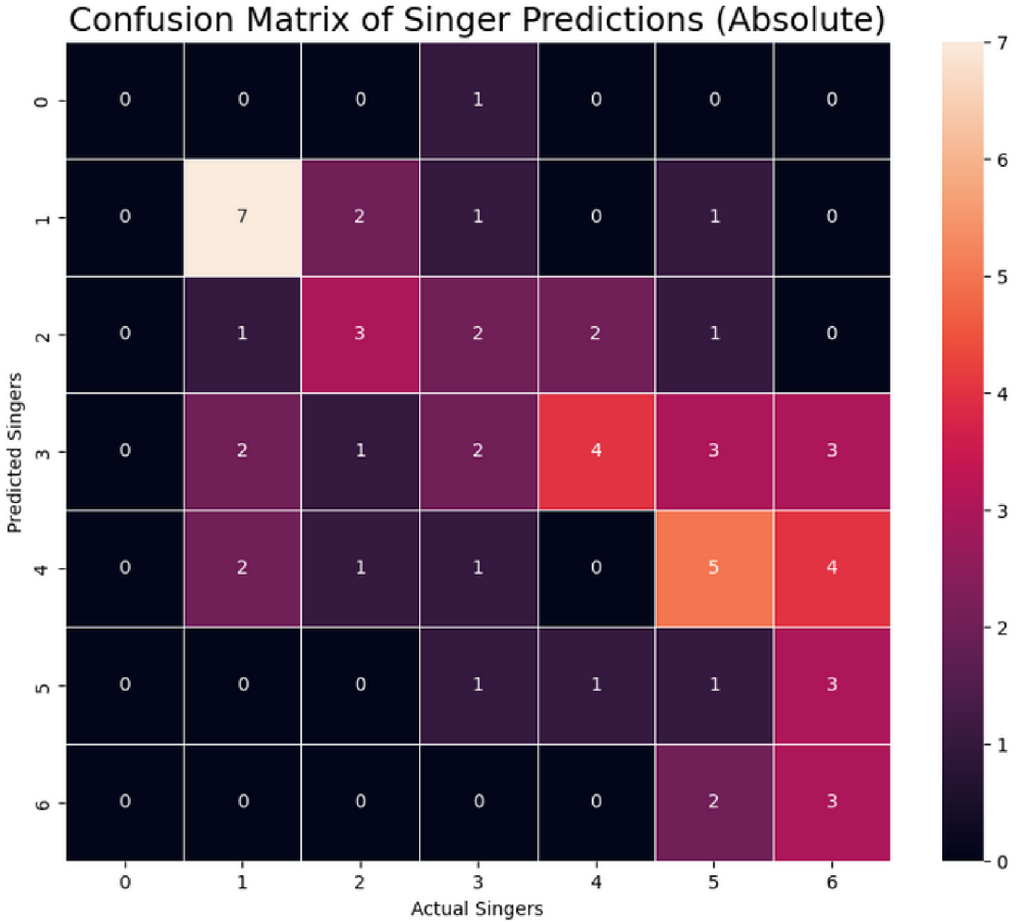
Accuracy +/-2: 90.00% **n = 60**



Confusion Matrix:

Chunk switches accuracy: 64.50%
Accuracy +/- 0: 46.67%
Accuracy +/- 1: 73.33%
Accuracy +/-2: 90.00% **n = 60**





04

Summary of Results

Accuracy

	1 - Clustering	2 - Pyannote	3- RNN
2 Singers - No Randomization	**75.6%**	-	99.7%
2 Singers - With Randomization	**72.5%**	45.16%	95.0%
2 - 8 Singers - With Randomization and Smaller Chunks	18%	7.5%	**46.7%**

Final Reflections

Unexpected Challenges & Takeaways

Impact of Preprocessing

- Past assignments required minimal changes
- There are often many approaches to preprocessing
- Minor changes can cause massive performance differences

There's not always a good benchmark

- Many models are trained on conversation
- Conversation != singing!

Audio is hard.

- Ran into many audio-specific issues:
  - Preprocessing into ...chunks/feature extraction?
  - Fine-tuning requires processing in .yaml file for dataset structure, .rttm file for audio chunks, etc.

C.-B. Jeon, H. Moon, K. Choi, B. S. Chon, and K. Lee, “Medleyvox: An evaluation dataset for multiple singing voices separation,” ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023. doi:10.1109/icassp49357.2023.10095425

H. Bredin et al., “Pyannote.audio: Neural building blocks for speaker diarization,” ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020. doi:10.1109/icassp40776.2020.9052974