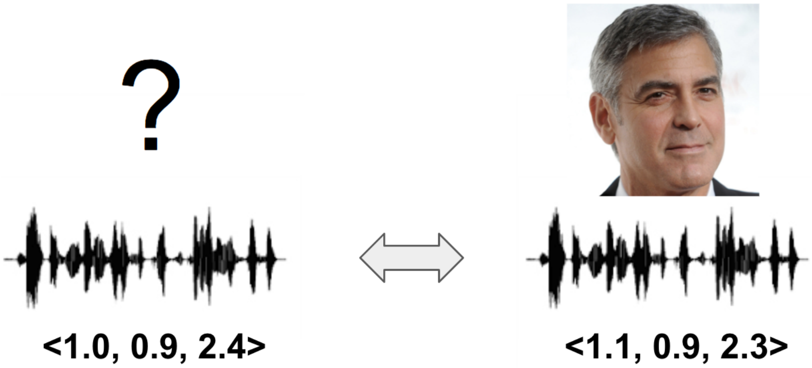
speaker recognition and verification

Eddouks Salima
Eddouks Oumayma

May 2023



1 Introduction

Speaker recognition, also known as voice recognition or voice biometrics, is a fascinating area of research that focuses on identifying and verifying individuals based on their unique vocal characteristics. It involves analyzing speech signals to extract relevant features from speech signals and utilizing GMM models for classification The project will begin by collecting a diverse dataset of speech samples from a wide range of speakers. The speech data will undergo preprocessing to remove Silence . Feature extraction techniques, such as Mel Frequency Cepstral Coefficients (MFCCs), will be applied to capture the distinctive characteristics of each speaker's voice. Gaussian Mixture Models (GMMs) will be employed as the classification models for speaker recognition. GMMs are statistical models that represent the probability distribution of features extracted from speech signals. They have been widely used in speaker recognition due to their ability to model the variability in speaker-specific features effectively.

2 Objective1:

The main objective of this project is to analyze and evaluate the performance of speaker recognition systems using different segment lengths of features (3 seconds, 10 seconds, 15 seconds, and 30 seconds) for both female and male speakers. Specifically, the project aims to generate error rate bar_charts that depict the accuracy of speaker identification and verification across these segment lengths. By comparing the performance of the system for different durations and gender categories, the objective is to gain insights into the optimal segment length for accurate speaker recognition and to understand any potential gender-based variations in the recognition performance

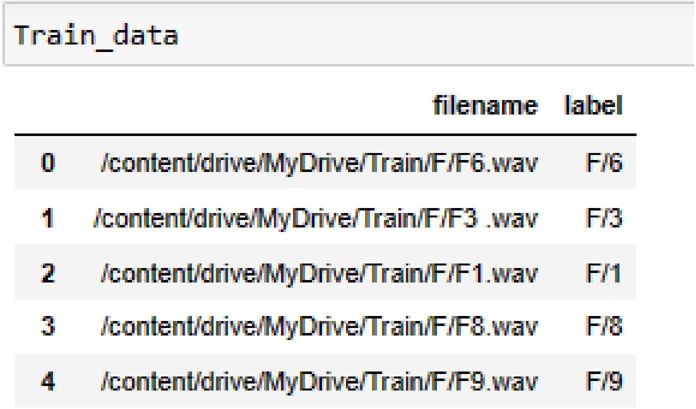
3 Objective2:

The second objective of this project is to perform speaker verification using the developed speaker recognition system based on Gaussian Mixture Models (GMMs). This involves computing the scores of all GMMs corresponding to each female and male speaker. Additionally, the project aims to generate Detection Error Tradeoff (DET) curves for different segment lengths (3 seconds, 10 seconds, 15 seconds, and 30 seconds). These curves will represent the relationship between the False Acceptance Rate (TFA) - the ratio of impostors accepted to the total number of impostors - and the False Rejection Rate (TFR) - the ratio of clients rejected to the total number of clients. By analyzing the DET curves, the objective is to assess the system's performance across different segment lengths and evaluate its tradeoff between false acceptance and false rejection rates. In addition this project aims to determine the optimal threshold for speaker verification based on the Detection Error Tradeoff (DET) curve. The objective is to find the threshold point on the DET curve that minimizes the Equal Error Rate (EER) or achieves the desired balance between the False Acceptance Rate (FAR) and the False Rejection Rate (FRR). This optimal threshold will serve as a decision boundary to classify speaker verification outcomes accurately. By finding the optimal threshold, the objective is to enhance the system's performance and strike an optimal balance between accepting legitimate speakers and rejecting impostors.

4 Corpus :

The corpus for this project contain 10 females and 7 males :

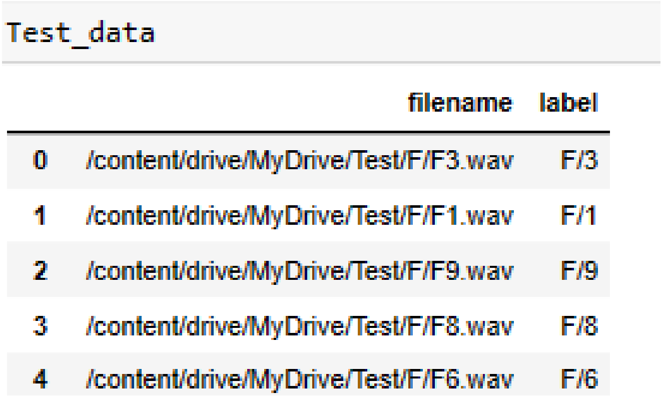
For the training data, 1 minute of audio data is available from each female and male speaker. This data will be used to train the speaker recognition system, allowing it to learn and model the unique vocal characteristics specific to each individual. The training process will involve feature extraction, such as Mel Frequency Cepstral Coefficients (MFCCs), to capture the relevant information from the audio data and train Gaussian Mixture Models (GMMs) for each speaker category.



```
Train_data
```

	filename	label
0	/content/drive/MyDrive/Train/F/F6.wav	F/6
1	/content/drive/MyDrive/Train/F/F3 .wav	F/3
2	/content/drive/MyDrive/Train/F/F1.wav	F/1
3	/content/drive/MyDrive/Train/F/F8.wav	F/8
4	/content/drive/MyDrive/Train/F/F9.wav	F/9

The testing data consists of 1 minute of audio data from each female and male speaker, separate from the training data. This data will be used to evaluate the performance of the trained speaker recognition system. By applying the trained GMM models to the testing data

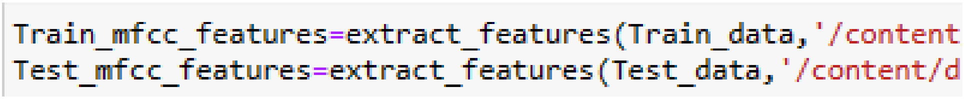


```
Test_data
```

	filename	label
0	/content/drive/MyDrive/Test/F/F3.wav	F/3
1	/content/drive/MyDrive/Test/F/F1.wav	F/1
2	/content/drive/MyDrive/Test/F/F9.wav	F/9
3	/content/drive/MyDrive/Test/F/F8.wav	F/8
4	/content/drive/MyDrive/Test/F/F6.wav	F/6

5 Extraction features :

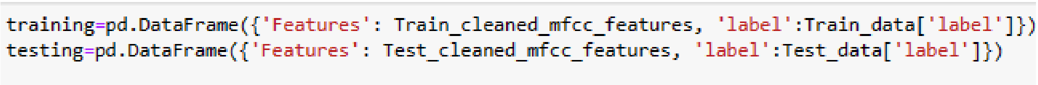
A function extract_features that takes in a DataFrame (data) containing information about audio files, and a directory path (dir_path) specifying the location of the audio files. The function extracts MFCC (Mel Frequency Cepstral Coefficients) features from the audio files. we applied this function to our Train_data and Test_data , The extracted MFCC features from the training data are stored in Train_mfcc_features , and the extracted MFCC features from the test data are stored in Test_mfcc_features



```
Train_mfcc_features=extract_features(Train_data,'/content
Test_mfcc_features=extract_features(Test_data,'/content/d
```

6 Eliminate Silence :

A function eliminate_low_mean_frames that takes in audio_data as input. It eliminates frames with low energy based on a bi-Gaussian model.The function first calculates the energy of each audio frame by computing the logarithm of the sum of squared values of the audio data across the frames. It then fits a bi-Gaussian model to the energy distribution, estimating the mean (energy_mean) and standard deviation (energy_std). A threshold is determined as the difference between the energy mean and the energy standard deviation. the function is applied to each feature in the Train_mfcc_features list and Test_mfcc_features list using a loop. The cleaned features are stored in the Train_cleaned_mfcc_features list and Test_cleaned_mfcc_features list. And testing and training DataFrame are created



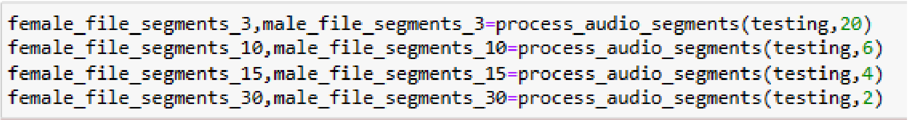
```
training=pd.DataFrame({'Features': Train_cleaned_mfcc_features, 'label':Train_data['label']})
testing=pd.DataFrame({'Features': Test_cleaned_mfcc_features, 'label':Test_data['label']})
```

7 GMM Models :

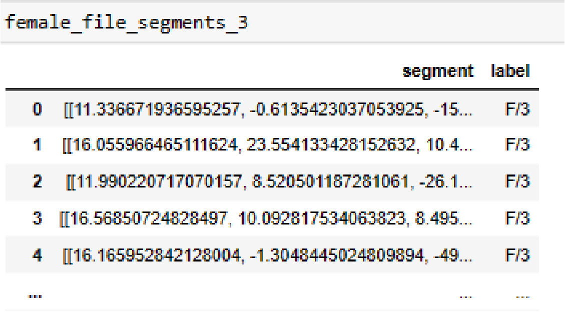
The function GMM_MODEL, which takes in audio data and the number of Gaussian components (N_Gaussien) as input. It fits a Gaussian Mixture Model (GMM) to the audio data , the DataFrames, F_models and H_models, are created to store the GMM models for female and male with different numbers of Gaussian components (128, 256, 512, and 1024) The outputs are F_models_128 , H_models_128, F_models_256 , H_models_256, F_models_512 , H_models_512, F_models_1024 , H_models_1024

8 Devide testing data into segments :

In this report, we present a method for segmenting the audio features , We adopted a segmentation strategy that involves dividing the features into segments of different durations. Specifically, we segmented the features into 20 segments, each spanning 3 seconds, allowing for a fine-grained analysis of short-duration patterns in the audio. Additionally, we created 6 segments of 10 seconds, 4 segments of 15 seconds, and 2 segments of 30 seconds to capture longer-term characteristics and variations in the audio data. This multi-segment approach enables us to examine the impact of segment duration on the speaker recognition system's performance,



```
female_file_segments_3,male_file_segments_3=process_audio_segments(testing,20)
female_file_segments_10,male_file_segments_10=process_audio_segments(testing,6)
female_file_segments_15,male_file_segments_15=process_audio_segments(testing,4)
female_file_segments_30,male_file_segments_30=process_audio_segments(testing,2)
```



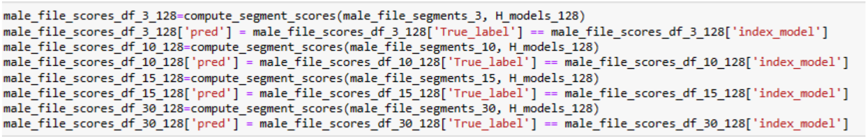
```
female_file_segments_3
```

	segment	label
0	[[11.336671936595257, -0.6135423037053925, -15...	F/3
1	[[16.055966465111624, 23.554133428152632, 10.4...	F/3
2	[[11.990220717070157, 8.520501187281061, -26.1...	F/3
3	[[16.56850724828497, 10.092817534063823, 8.495...	F/3
4	[[16.165952842128004, -1.3048445024809894, -49...	F/3
...	...	...

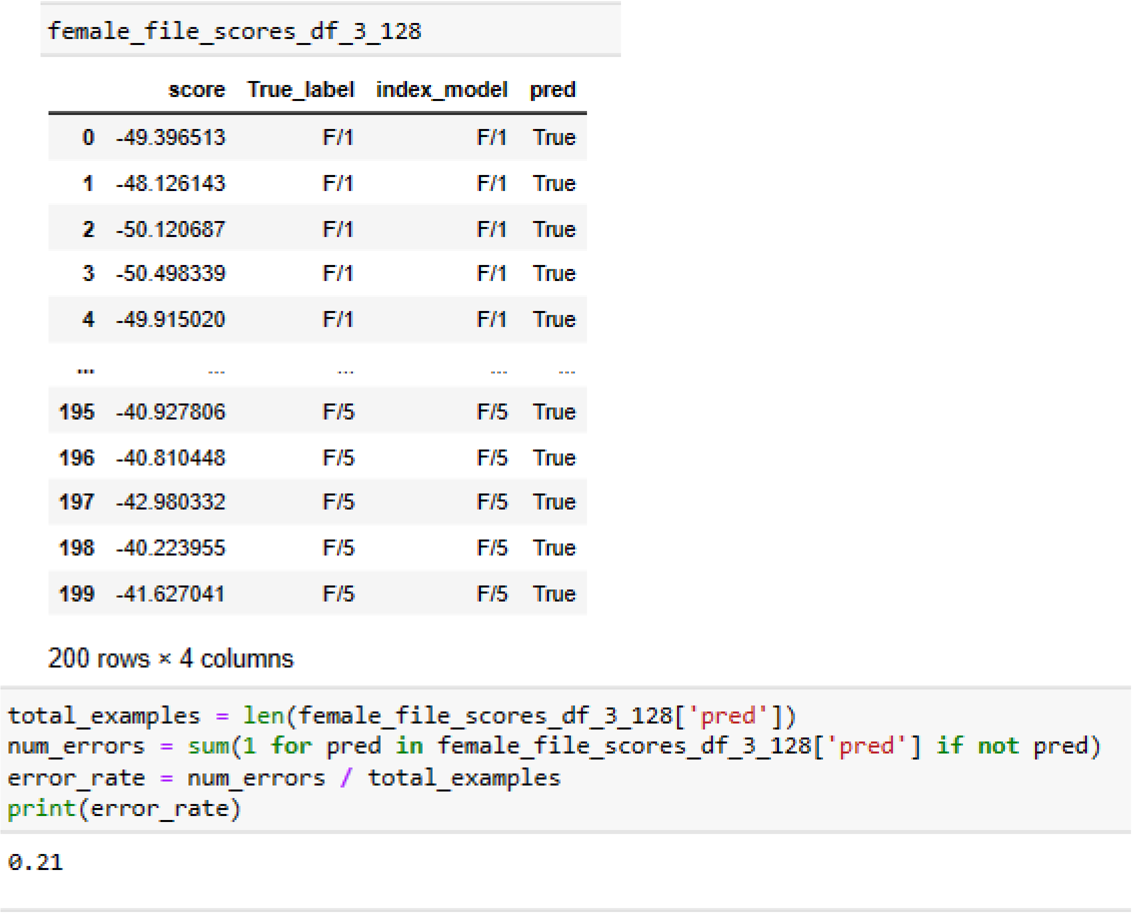
9 Identification:

9.1 computing scores of segments:

In our report, we performed speaker recognition using Gaussian Mixture Models (GMMs) with different numbers of components (128, 256, 512, and 1024). We segmented the audio data into various durations (3 seconds, 10 seconds, 15 seconds, and 30 seconds) and computed the segment scores for each model. For each segment, we calculated the scores by comparing the segment with the corresponding GMM model. The maximum score and the corresponding model index were recorded for each segment. We then evaluated the accuracy of the recognition by comparing the true labels with the predicted labels based on the maximum score. We repeated this process for different GMM configurations. The results were saved in separate dataframes for each duration and number of components. This analysis allowed us to examine the performance of the GMMs with varying complexity and segment durations, providing insights into the optimal configuration for our speaker recognition system.



```
male_file_scores_df_3_128=compute_segment_scores(male_file_segments_3, H_models_128)
male_file_scores_df_3_128['pred'] = male_file_scores_df_3_128['True_label'] == male_file_scores_df_3_128['index_model']
male_file_scores_df_10_128=compute_segment_scores(male_file_segments_10, H_models_128)
male_file_scores_df_10_128['pred'] = male_file_scores_df_10_128['True_label'] == male_file_scores_df_10_128['index_model']
male_file_scores_df_15_128=compute_segment_scores(male_file_segments_15, H_models_128)
male_file_scores_df_15_128['pred'] = male_file_scores_df_15_128['True_label'] == male_file_scores_df_15_128['index_model']
male_file_scores_df_30_128=compute_segment_scores(male_file_segments_30, H_models_128)
male_file_scores_df_30_128['pred'] = male_file_scores_df_30_128['True_label'] == male_file_scores_df_30_128['index_model']
```



```
female_file_scores_df_3_128
```

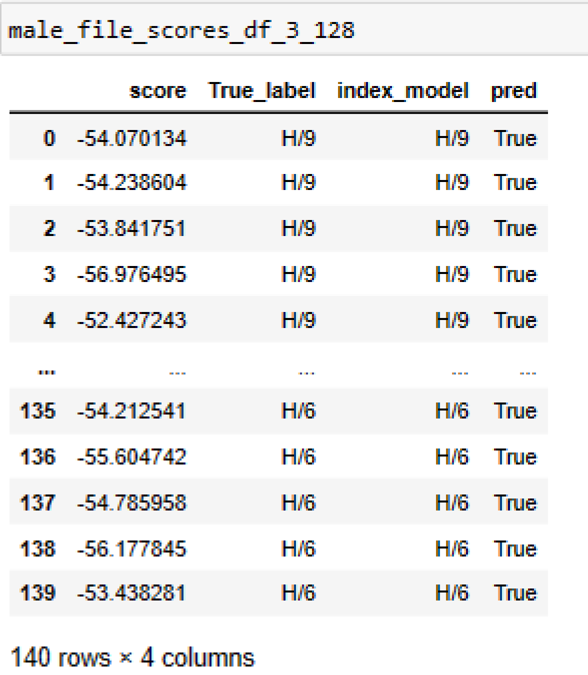
	score	True_label	index_model	pred
0	-49.396513	F/1	F/1	True
1	-48.126143	F/1	F/1	True
2	-50.120687	F/1	F/1	True
3	-50.498339	F/1	F/1	True
4	-49.915020	F/1	F/1	True
...	...	...	...	...
195	-40.927806	F/5	F/5	True
196	-40.810448	F/5	F/5	True
197	-42.980332	F/5	F/5	True
198	-40.223955	F/5	F/5	True
199	-41.627041	F/5	F/5	True

200 rows × 4 columns

```python
total_examples = len(female_file_scores_df_3_128['pred'])
num_errors = sum(1 for pred in female_file_scores_df_3_128['pred'] if not pred)
error_rate = num_errors / total_examples
print(error_rate)
```

```
0.21
```

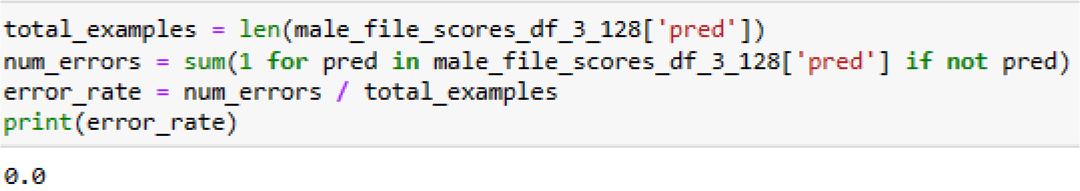
for example here , The speaker recognition system achieved an error rate of 0.21 for all the segments of 3s of females with GMM 128



```
male_file_scores_df_3_128
```

	score	True_label	index_model	pred
0	-54.070134	H/9	H/9	True
1	-54.238604	H/9	H/9	True
2	-53.841751	H/9	H/9	True
3	-56.976495	H/9	H/9	True
4	-52.427243	H/9	H/9	True
...	...	...	...	...
135	-54.212541	H/6	H/6	True
136	-55.604742	H/6	H/6	True
137	-54.785958	H/6	H/6	True
138	-56.177845	H/6	H/6	True
139	-53.438281	H/6	H/6	True

140 rows × 4 columns



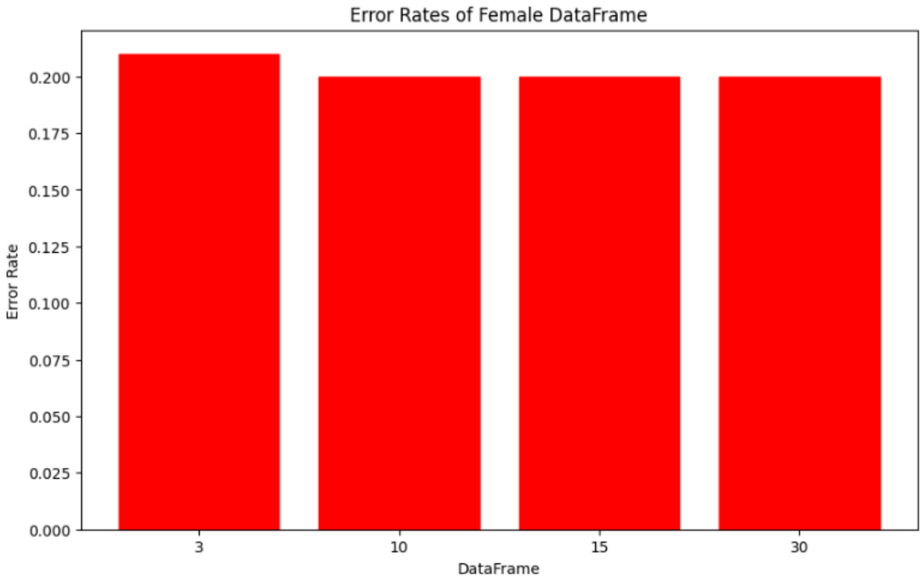
```
total_examples = len(male_file_scores_df_3_128['pred'])
num_errors = sum(1 for pred in male_file_scores_df_3_128['pred'] if not pred)
error_rate = num_errors / total_examples
print(error_rate)
```

```
0.0
```

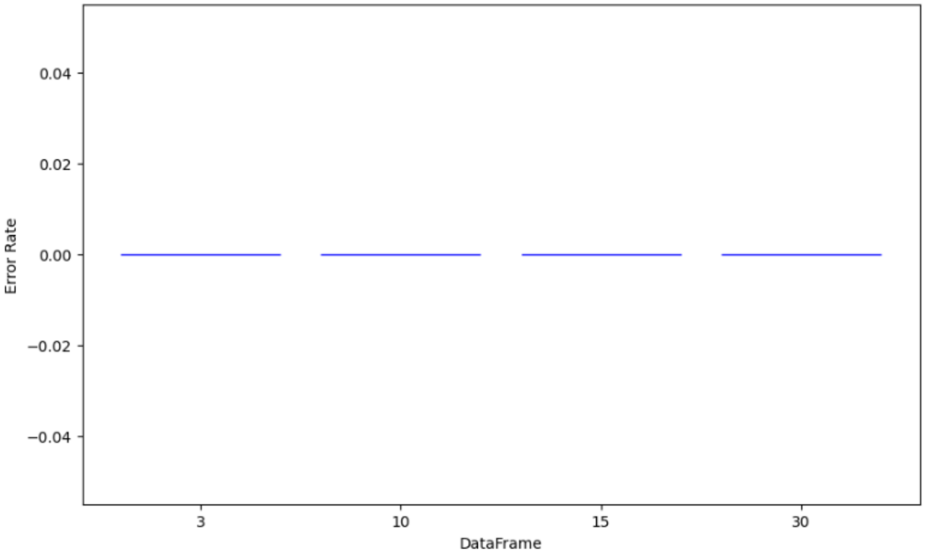
The speaker recognition system achieved an error rate of 0.00 for all the segments of 3s of males with GMM 128

9.2 plot the error rates:

The error rates of female and male DataFrames obtained using different configurations of Gaussian Mixture Models (GMMs) are plotted in the "Error Rates of Female and Male DataFrames" chart. The female_dfs list consists of four DataFrames, namely female_file_scores_df_3_128, female_file_scores_df_10_128, female_file_scores_df_15_128, and female_file_scores_df_30_128, each representing different segment durations. Similarly, the male_dfs list includes the corresponding DataFrames for male speakers. The error rates are calculated and displayed as bar plots, with different colors representing female and male error rates. The evaluation is performed for GMMs with 128, 256, 512, and 1024 components.



These results indicate that the performance of the model varied across different female DataFrames, with higher error rates observed for smaller segment sizes (3 and 10) compared to larger segment sizes (15 and 30).

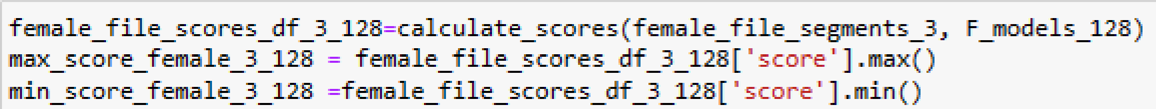


In contrast to the female DataFrames, the error rates for the male DataFrames were found to be consistently zero for all segment sizes,The absence of errors suggests that the model was highly successful in classifying the male segments correctly, demonstrating its robustness and effectiveness in distinguishing male speech patterns or characteristics.

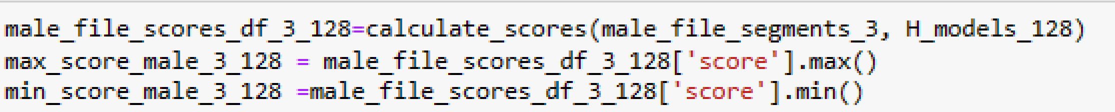
10 Verification:

10.1 Calculate scores of segments

The scores for the segments of female and male speakers using different GMM configurations (128, 256, 512, and 1024 components) have been calculated and stored in separate DataFrames. For example, female_file_scores_df_3_128 contains the scores for female segments of 3 seconds using GMMs with 128 components. The maximum and minimum scores for this DataFrame are obtained as max_score_female_3_128 and min_score_female_3_128, respectively. This process is repeated for other segment durations and GMM configurations, resulting in separate DataFrames for each combination. Similarly, the same calculations are performed for male segments, generating male_file_scores_df DataFrames for each configuration.



```
female_file_scores_df_3_128=calculate_scores(female_file_segments_3, F_models_128)
max_score_female_3_128 = female_file_scores_df_3_128['score'].max()
min_score_female_3_128 =female_file_scores_df_3_128['score'].min()
```

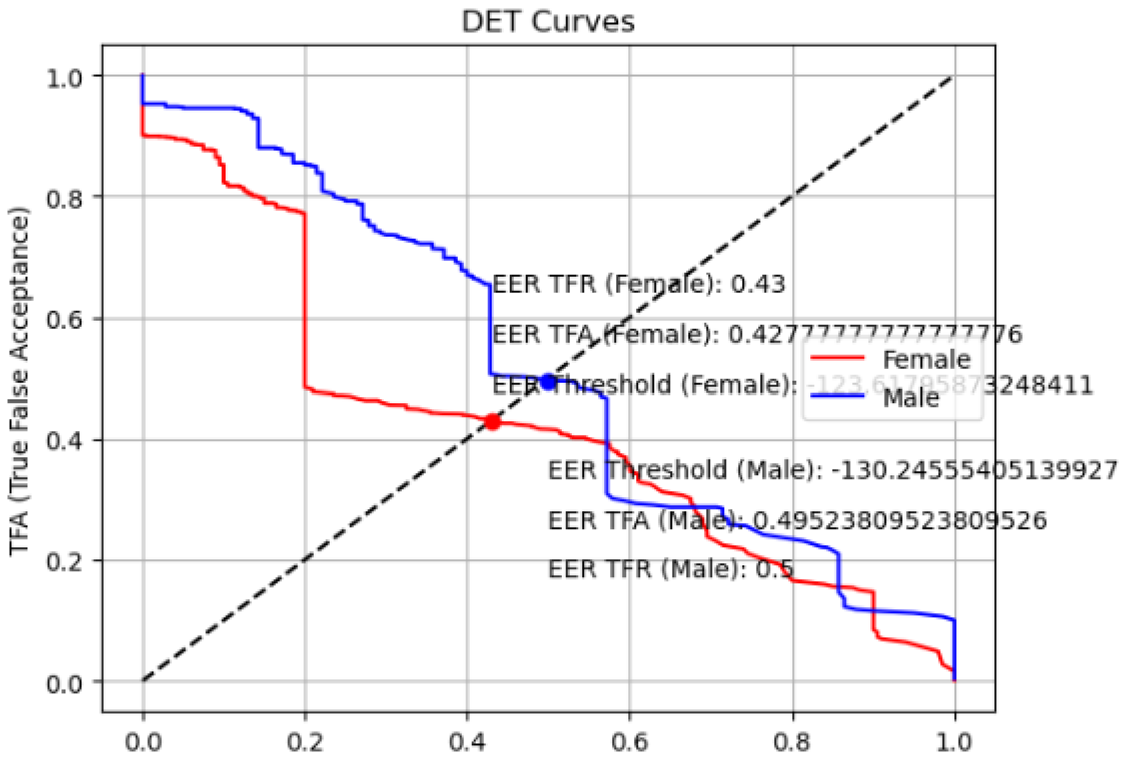


```
male_file_scores_df_3_128=calculate_scores(male_file_segments_3, H_models_128)
max_score_male_3_128 = male_file_scores_df_3_128['score'].max()
min_score_male_3_128 =male_file_scores_df_3_128['score'].min()
```

10.2 calculate TFA and TFR:

TFA = calculated by dividing the number of impostor samples incorrectly accepted by the total number of impostor samples

TFR = calculated by dividing the number of clients samples incorrectly rejected by the total number of clients.



The DET (Detection Error Tradeoff) curve was plotted for the female DataFrame in red, and the male DataFrame in blue with a segment size of 3 seconds. The curve represents the relationship between the True False Rejection (TFR) and True False Acceptance (TFA) rates . The Equal Error Rate (EER) is a crucial point on the DET curve that represents the threshold at which the TFR and TFA rates are equal

For the female DataFrame:
EER threshold = -130
TFA = 0.49
TFR = 0.5

For the male DataFrame:
EER threshold = -123
TFA = 0.42
TFR = 0.43

11 Conclusion:

In conclusion, this project aimed to analyze the performance of a speaker identification and verification system using female and male datasets. it was observed that using Gaussian Mixture Models (GMMs) with different numbers of components had a significant impact on the performance. While higher numbers of components, such as GMMs with 256, 512, and 1024 components, yielded promising results, it was also noticed that the computational requirements for executing these models were demanding, surpassing the capabilities of the hardware.

Considering the limitations of the hardware, the focus was shifted towards GMMs with 128 components. Despite having a lower number of components compared to the higher-performing models, the GMM 128 still exhibited commendable performance.