2025

Voice Authentication

PHASE II

INTRODUCTION TO ML FALL 1403

PREPARED BY:

ASRA MEHROLHASSANI SID: 810800024
ZANKO KARIMI SID: 810801075
ZANIYAR GHAZALI SID: 810801072
HOSSEIN NOROOZI SID: 810899073

THE CURRENT DOCUMENT IS A REPORT OF THE SECOND PHASE OF MACHINE LEARNING FINAL PROJECT

Table of Contents

Voice Authentication: Phase II

The current document is a report of the second phase of the Voice authentication project, prepared for the class Machine Learning- ML 8111-097-01.

Knowledge of '**Audio Signal Processing**" is required for understanding this document.

1. An Introduction to Voice Authentication

Voice Authentication definition, its importance & applications

Since human exists, he has always wanted to make life easier for themselves. Due to the innate desire to survive or whatever, it resulted in advancement of science and technology and even change of the way human thinks nowadays or the way they live now.

As we all feel and see, everything has changed (not necessarily positively), life is not as was before, and human is tending to ease everything, special thanks to technologies of artificial intelligence nowadays briefly known as AI.

AI is somehow capturing human positions due to its convenience, speed, and also accuracy, and in some fields, we can maybe also consider security too. One of the fields that AI is capturing human's position is in diagnosis, even the diagnosis of illnesses, AI is little by little tending to be better than humans in performance.

An important AI technic is speech recognition, in other words, voice recognition, or specifically voice authentication. Voice authentication refers to the process of verifying an individual's identity using their unique vocal characteristics. It is a biometric method that captures and analyzes the voice to match it with a pre-stored voiceprint.

We all feel the importance of diagnosis of someone by their voice everywhere in our lives. For instance, someone calls you, and says: "You know me?", that's awkward to say no, ha? That was a small encounter to voice authentication. Think of a credit card industry. It's vital to ensure that a particular purchase is being

authorized by the actual card-holding customer and not someone else. Merchants only incur this fee if a Voice Authorization is initiated, and for most merchants it is a rare occurrence. It's hard to do all of these without these technics. They're quick and more accurate than humans in speech recognition.

Humans were in need of speech recognition systems due to a combination of practical, technological, and accessibility-driven needs as mentioned above. Let's go through some other key reasons:

- Efficiency and convenience, for example it has Hands-free interaction and we also save time by speaking than typing or other interactions.
- Accessibility for people with disabilities
- Enhancing User Experience and Human-Computer interaction
- Economic and Practical advantages
- Safety and many other important reasons

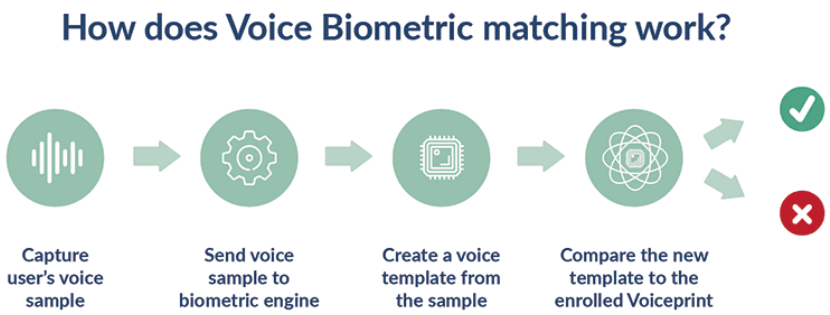


Figure 1. Voice Biometric Matching

2.Voice Authentication & Gender Classification Challenges

Challenges, solutions, and ongoing researches

Despite many advantages voice authentication has as mentioned above, it also comes with important challenges. These challenges should be considered in businesses and researches. Some key challenges are as below:

Model Accuracy

One of the key challenges in voice authentication is achieving a high accuracy. Model accuracy is sensitive to background noise. In noisy environments, such as crowded areas or places with poor acoustics, the accuracy of voice analysis may be compromised. This leads to potential false rejections or acceptances.

There are effective approaches to achieve a high accuracy and overcome the challenges in noisy environments. Improving the dataset can enhance speech recognition model accuracy. A larger, more diverse, and high-quality dataset helps the model better understand different accents, dialects, background noise, and speaking styles, leading to more accurate predictions. Knowing the user's environment before developing the model can be beneficial in understanding what kind of background noise the system will be required to ignore and we can also use linear noise reduction filters.

Vulnerabilities to voice replication techniques

One of the challenges with voice biometrics is establishing that the original sample voice is authentic and that it belongs to the intended identity – there is no trusted source or identity document to remotely verify against. Even if a genuine sample is obtained, the enrollment samples may not be diverse enough or may not adequately represent the variability of an individual's voice, which could lead to difficulties in accurate identification. Ultimately, biometric voice technology does not assure that an individual is who they claim to be – only that a voice matches the original template. By this logic, voice biometrics cannot secure the highest risk point in the user journey: onboarding. As such, it provides limited defense against the most pervasive and damaging identity fraud types: deepfakes and synthetic identity fraud. This fundamental flaw is critical to financial services organization's security; the inability to bind a digital identity to a real-world person means that financial institutions must rely on other methods to establish high levels of assurance.

Figure 2. Identity fraud

Ethical and privacy concerns

Voice authentication collects and stores individuals' biometric data, raising ethical and privacy concerns. Organizations must adhere to strict privacy regulations and secure storage practices to prevent unauthorized access or misuse of sensitive voice data. Transparency and clear user consent mechanisms are essential to address these concerns effectively.

Accessibility issues

Voice biometrics can be adversely affected by certain medical conditions or disabilities that impact speech patterns, making it difficult or impossible for some individuals to use the technology effectively. Individuals may not always speak clearly or consistently, which can impact the system's performance. The voice changes over time, which is problematic.

Voice diversity in gender classification

Variability in vocal frequency, pitch, and other vocal characteristics, especially among non-binary and transgender individuals, makes classification more challenging.

The topics mentioned above were disadvantages and challenges voice authentication may face. As mentioned, for some of them, there are approaches to consider. A more general approach is to use a multi-layer authentication system or use of face authentication along with voice authentication. Although, achieving a 100 percent assurance is never possible.

3.Dataset

About The Provided Dataset

The provided dataset contains above **800** students' recorded voices concerning ML class's first homework. File's names are of the "**HW1_Qx_SID_gender.mp3**" format and it is worth noting that the average audio length is **3 minutes**.

Missing Values

As expected, some inconsistencies were perceived in the dataset, including 42/822 (5%) misnamed (mislabeled) files and few mis-recorded audios.

4. Data Preprocessing and Feature Extraction

Skipping Silence and Resampling

There exists some silent intervals in the audio signal that we need to skip. In order to sample the data, a fixed sampling frequency is required, in this project the sampling rate is equivalent to 16000 Hz(where 22050 Hz Is equivalent to 1 second).

Noise Cancelling

In order to cancel the noise in the audio signals, two methods can be used. One is the Spectral Subtraction which is based on the Fourier Transform and subtracts the noise from the signal. The other method is Bandpass Filtering which is better when we know what interval of frequency is recognized as noise. As described by the above paragraph, It is evident that, for this project **Spectral Subtraction is of priority.**

Our code:

```
def spectral_subtraction(noisy_audio, sr = 22050, noise_start = 0, noise_end = 1):
    f, t, Zxx = stft(noisy_audio, fs=sr, nperseg=1024)

    noise_idx = (t >= noise_start) & (t <= noise_end)
    noise_spectrum = np.mean(np.abs(Zxx[:, noise_idx]), axis=1, keepdims=True)

    magnitude = np.abs(Zxx) - noise_spectrum
    magnitude = np.maximum(magnitude, 0)

    Zxx_denoised = magnitude * np.exp(1j * np.angle(Zxx))

    _, denoised_audio = istft(Zxx_denoised, fs=sr, nperseg=1024)

    return denoised_audio
```

Signal Normalization

To normalize the data we've normalized the loudness as it is variant across different audio signals. The below code demonstrates our solution:

```
def loudness_normalization(audio, sr = 22050):
```

```
    meter = pyln.Meter(sr)
    loudness = meter.integrated_loudness(audio)
    target_loudness = -23.0
    gain = 10**((target_loudness - loudness) / 20.0)
    audio_normalized = audio * gain
    return audio_normalized
```

Audio Split

In order to generate more data, we split the audio signal into 3 seconds-long intervals. Maximum …… number of samples is taken from each signal.

Our code:

```
def split_audio(audio, segment_length = 3, sr = 16000):
    segment_samples = segment_length * sr
    num_segments = len(audio) // segment_samples
    segments = [audio[i * segment_samples : (i + 1) * segment_samples] for i in
range(num_segments)]
    if len(segments) > 5:
        segments = random.sample(segments, 5)
    return segments
```

Audio Signal's Feature Extraction

In order to analyze the audio signals and generate valid data for the training phase of the project, the frequency related features including: **Log Mel Spectrogram (128 features), MFCC (13 features), Spectral Centroid, Spectral Bandwidth, Spectral Contrast (7 features)**, **FFT** along with the time domain features: **Energy, Zero-Crossing Rate** and **LPC (13 features), PLP(12 features) and Chroma(12 features)** features were extracted .

Relevant code:

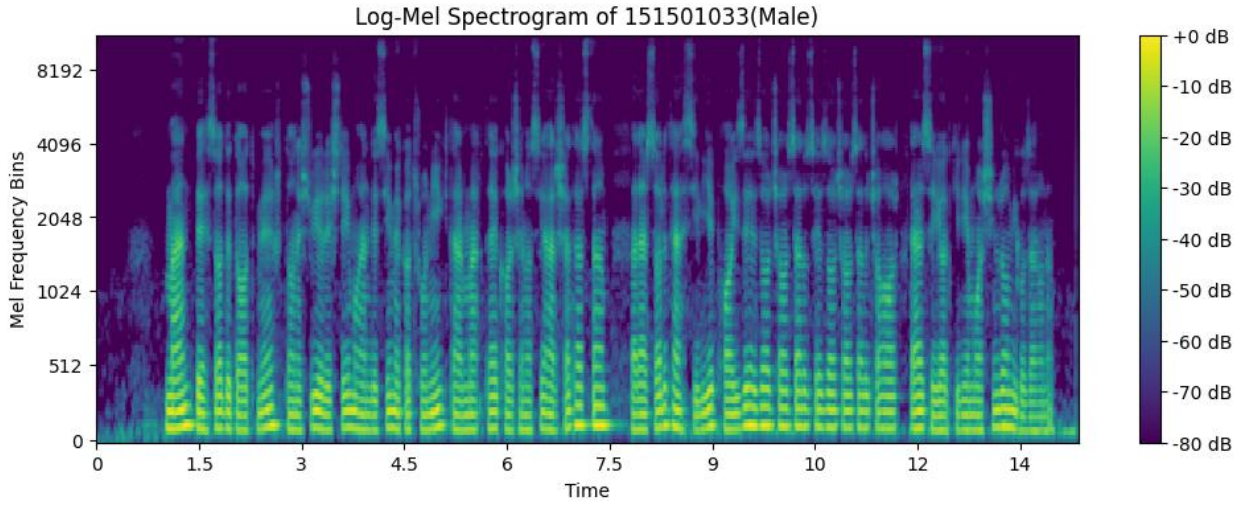
```
mfcc = np.mean(librosa.feature.mfcc(y=audio, sr=sr, n_mfcc=13), axis=1)
            log_mel = np.mean(librosa.feature.melspectrogram(y=audio, sr=sr,
n_mels=128), axis=1)  # Log Mel-Spectrogram (128 values)
            spec_centroid = np.mean(librosa.feature.spectral_centroid(y=audio,
sr=sr), axis=1)  # Spectral Centroid (1 value)
            spec_bandwidth = np.mean(librosa.feature.spectral_bandwidth(y=audio,
sr=sr), axis=1)  # Spectral Bandwidth (1 value)
            spec_contrast = np.mean(librosa.feature.spectral_contrast(y=audio,
sr=sr), axis=1)  # Spectral Contrast (7 values)
            zcr = np.mean(librosa.feature.zero_crossing_rate(y=audio), axis=1)  #
Zero-Crossing Rate (1 value)
```

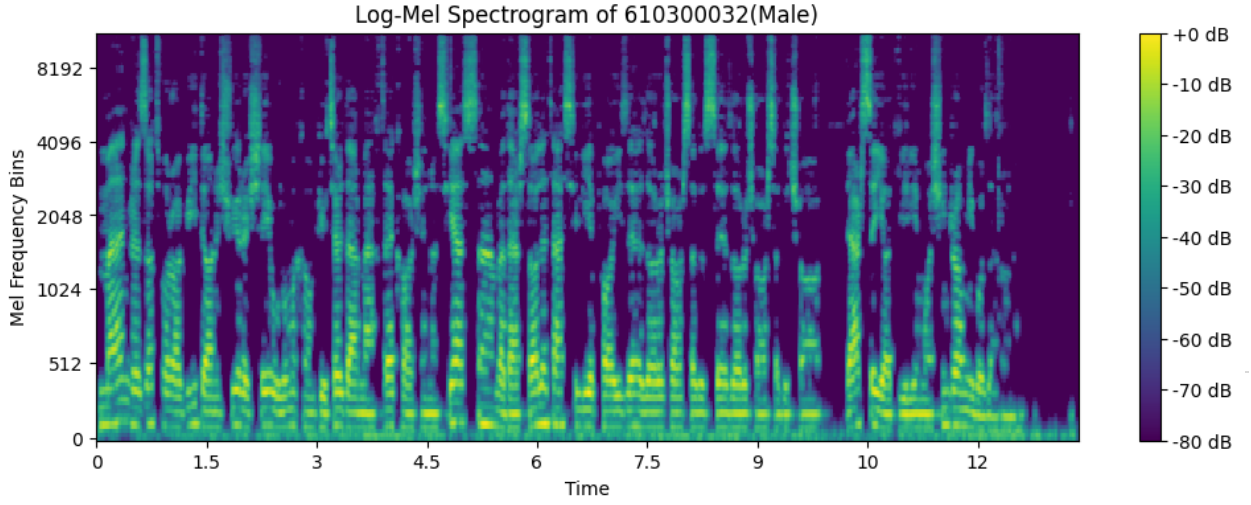
```
            energy = np.mean(np.square(audio))  # Energy (1 value)
            lpc_features = librosa.lpc(segment, order=12)
            plp_features = np.mean(logfbank(segment, samplerate=sr, nfilt=12),
axis=1)
            chroma = np.mean(librosa.feature.chroma_stft(y=segment, sr=sr,
n_chroma=12), axis=1)
            fft_spectrum = np.fft.fft(segment)  # Compute FFT
            fft_magnitude = np.abs(fft_spectrum[:len(fft_spectrum) // 2])  # Take
only positive frequencies
            fft_features = np.mean(fft_magnitude, axis=0)
            feature_vector = np.hstack([mfcc, log_mel, spec_centroid,
spec_bandwidth, spec_contrast, zcr, energy, lpc_features, plp_features, chroma,
fft_features])
            with_labels = np.hstack([feature_vector, ID_label, gender_label])
            features.append(feature_vector)
```

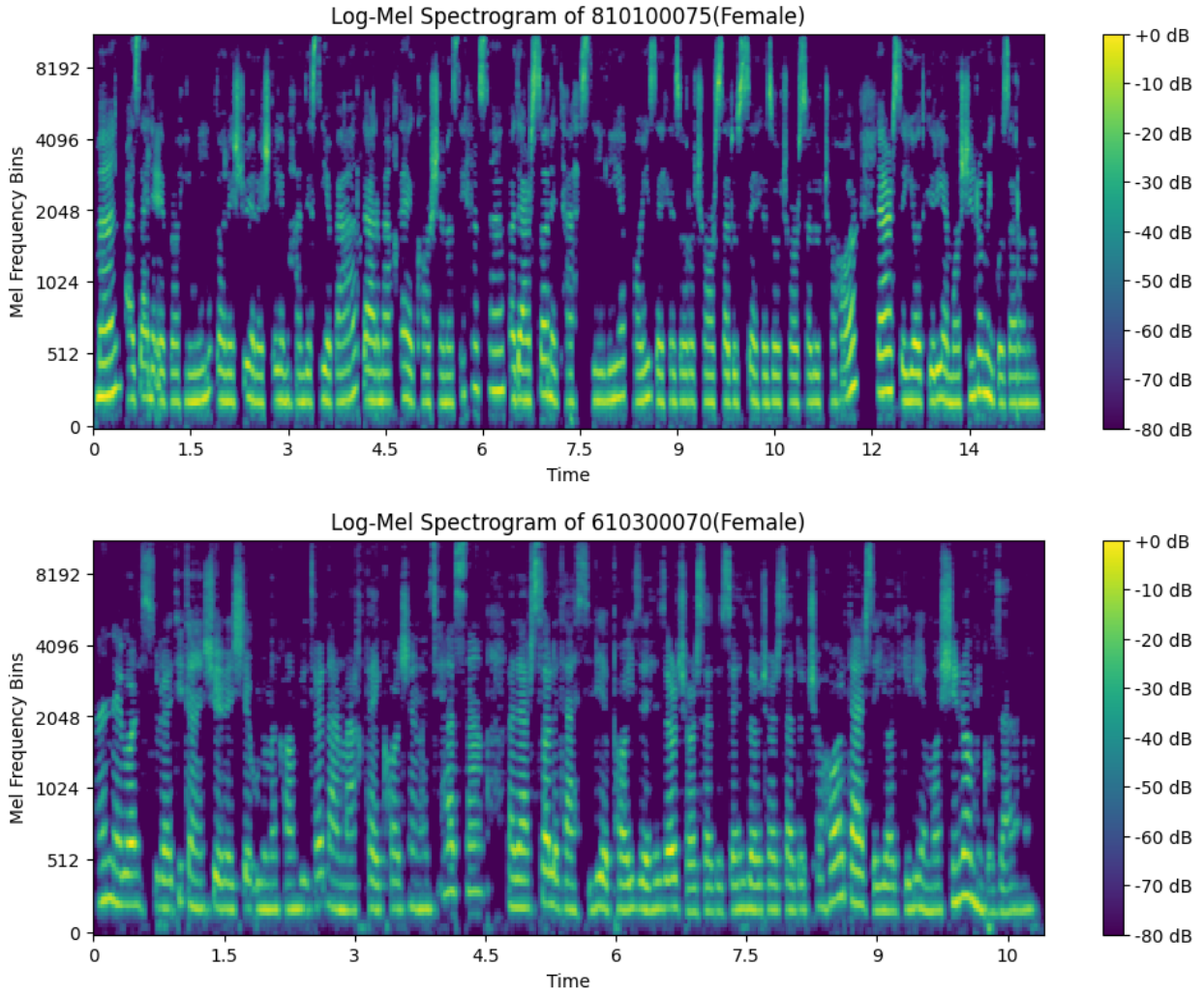
Data Visualization

Log Mel Spectrogram

In order to visualize the data, we have used 4 data samples (2 from men and two from women) and visualized the signals with Log Mel-Spectrogram (as it is closer to human's perception of the sound).





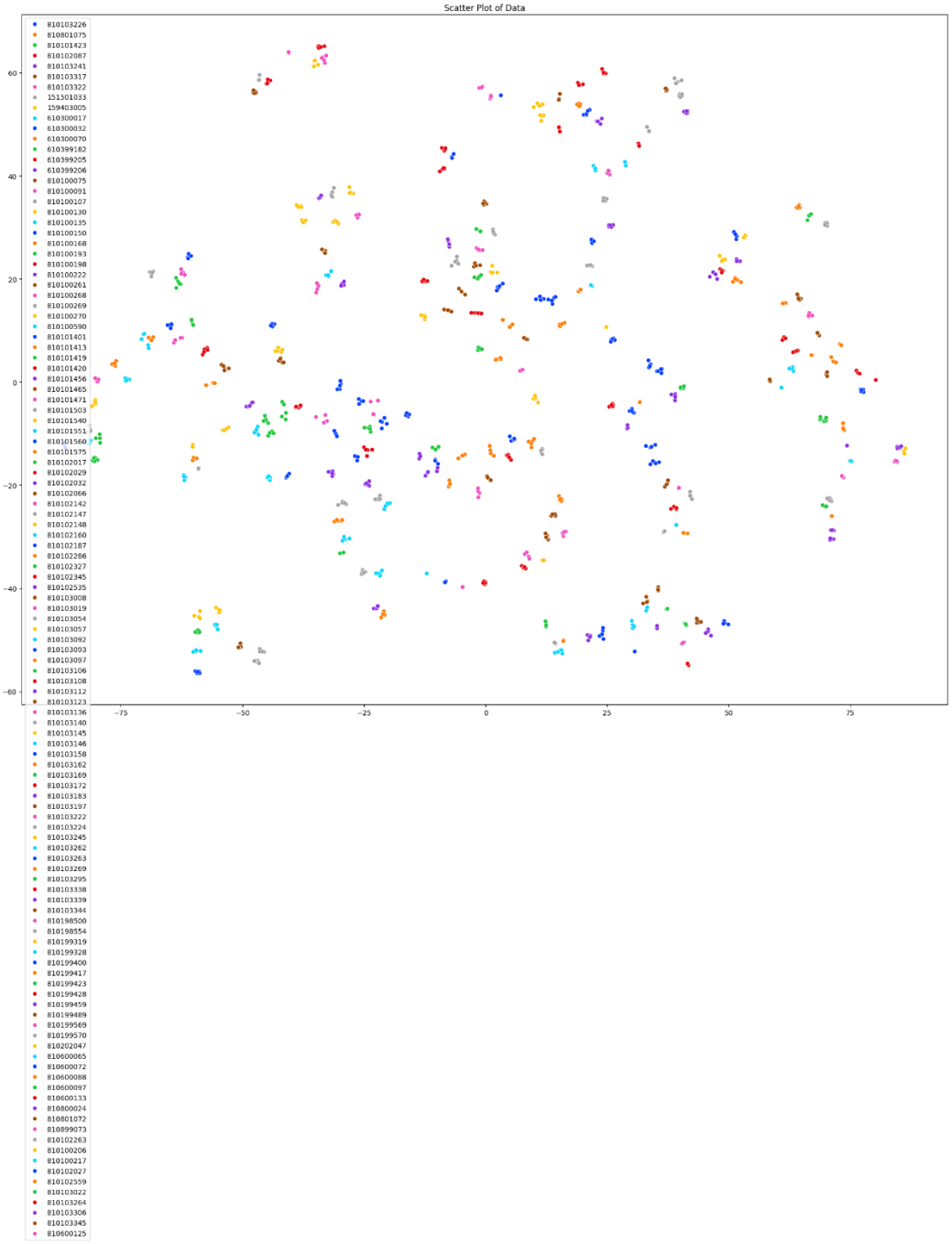


Notice how the **X-axis** represents how sound evolves as time passes and the **Y-axis** represents Mel frequency bands denoting pitch and tonal characteristics.

Color intensity represents the amplitude (loudness) of frequencies over time—brighter colors usually indicate higher intensity.

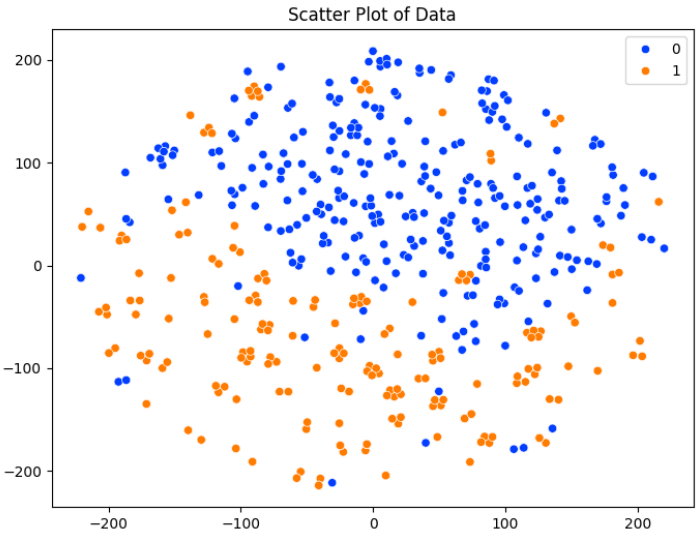
In order to decrease the dimensions of the dataset (selecting the dominant features only), we have applied PCA and t-SNE and visualized the results.

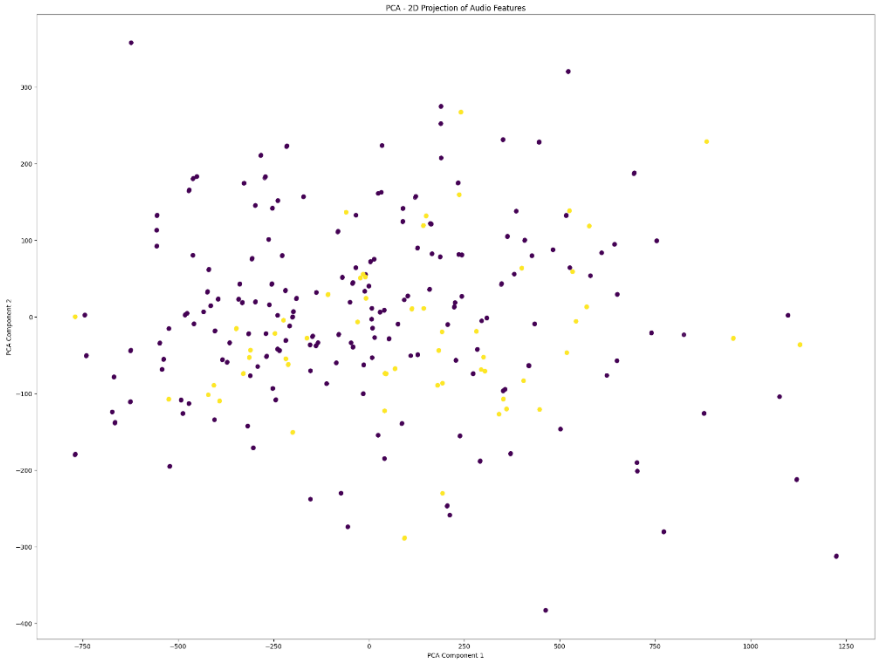
PCA and t-SNE



The above figure shows t-SNE applied (data with the studentID labels) and it shows how similar samples(from each person) are closer.

The above figure shows t-SNE applied (data with gender labels) and it shows how similar samples (each gender) are closer.





As it is evident, two Components are not enough for differentiating between the genders **with two features**.

In order to find out how many features are required for PCA, we analyzed the eigenvalues as below

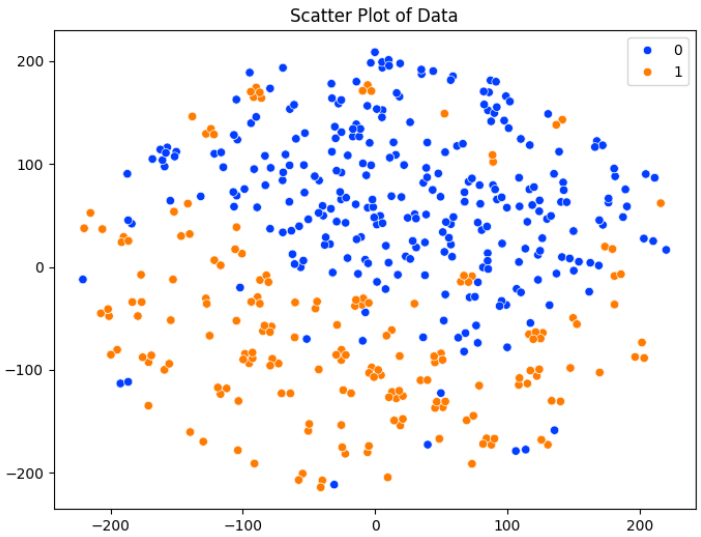
```
features = np.array(features)
cov = np.cov(features.T)
eigenvalues, eigenvectors = np.linalg.eig(cov)
eigenvalues = np.real(eigenvalues)
```

Afterwards we omitted the features with eigenvalues less than 1 and chose 15 of them. This helped us understand the problem better.

4. Gender Classification

In order to prepare the dataset for training, all the provided audios are used:)) However, feature extraction from such a dataset is quite laborious. Note that the number of female participant are the same as the number of men.

We have prepared 794 data to train and test the required model for gender classification.



Results

For the training process the Naïve Bayes algorithms were used which achieved:

F1 Score = 84% , Recall = 79%, Precision = 89%

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score

tsne_2d = TSNE(n_components=3, perplexity=10, random_state=42)
gender_features = tsne_2d.fit_transform(gender_features)


X_train, X_test, y_train, y_test = train_test_split(gender_features,
balanced_gender_labels, test_size=0.25, random_state=0)

nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

y_pred = nb_classifier.predict(X_test)
```

5. Closed-set Authentication (based on students ID)

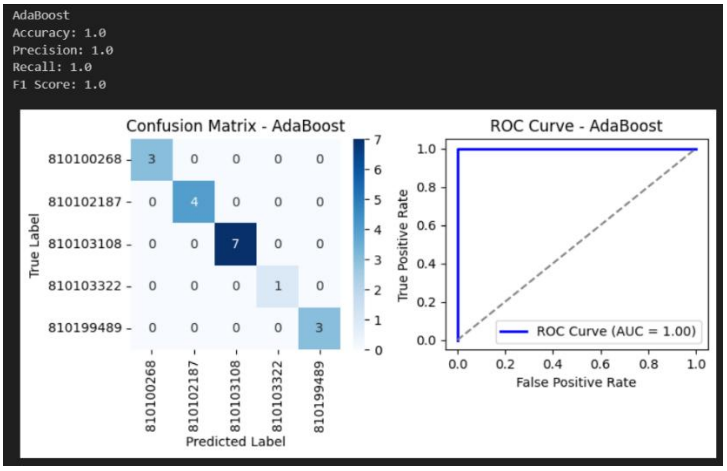
First we found a list of student's that have more samples than 14 (14 was obtained by observing how many samples exists for each student ID, 14 seemed to be the appropriate number). We selected 6 of these studentIDs randomly to train and test a model. Notice how we ensured that each studentID has the same number of data samples with other IDS.

For the training process the below algorithms were used:

SVM
MLP
KNN
Logistic Regression
AdaBoost

For comparing the above methods ROC curve is graphed and also the confusion matrix is studied for each method.

AdaBoost

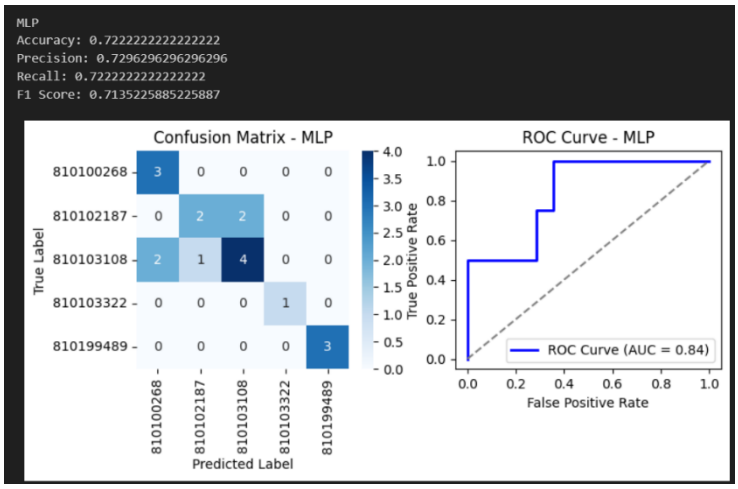


Notice how the x-axis denotes False Positive Rate and the y-axis denotes True Positive Rate. The more the area under the curve, the better. Since the area under the curve is equal to 1 (square shape) the model is acceptable. (f1 score = 1)

By studding the confusion matrix, accuracy, precision and recall are calculated.

MLP

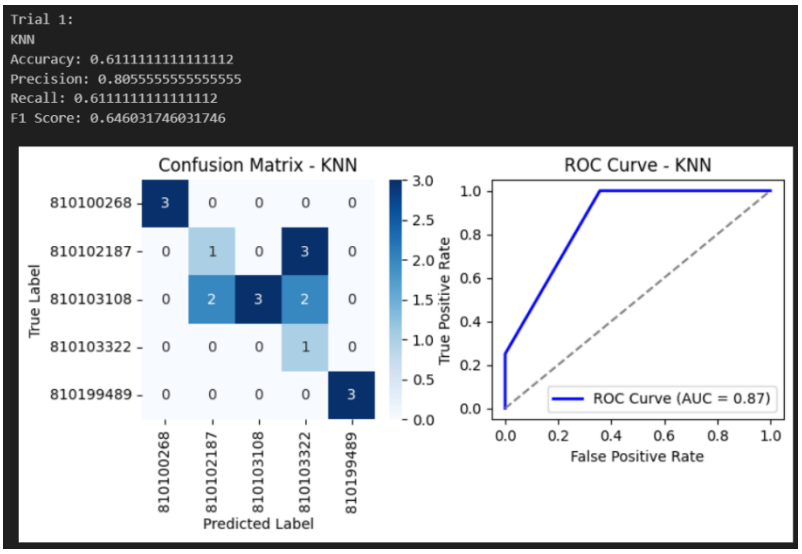
Notice how the x-axis denotes False Positive Rate and the y-axis denotes True Positive Rate. The more the area under the curve, the better. Since the area under the curve is not equal to 1 (stair shape, AUC = 0.84) the model is not as accurate as AdaBoost. By studding the confusion matrix, accuracy, precision and recall are calculated.



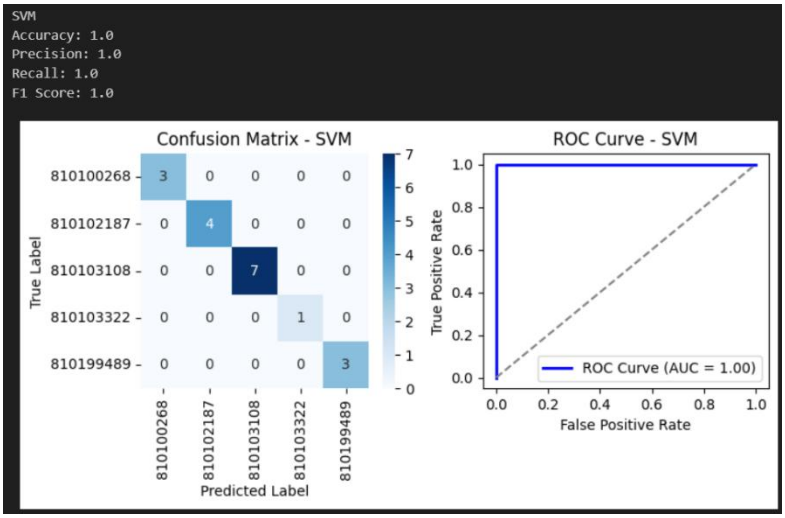
KNN

Notice how the x-axis denotes False Positive Rate and the y-axis denotes True Positive Rate. The more the area under the curve, the better. Since the area under the curve is not equal to 1 (AUC = 0.87) the model is not as accurate as KNN.

By studding the confusion matrix, accuracy, precision and recall are calculated. The result is presented in the figure.



SVM



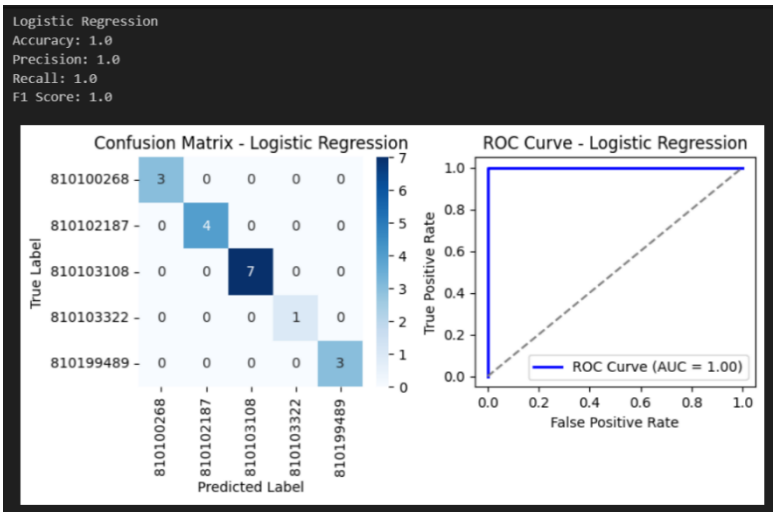
Notice how the x-axis denotes False Positive Rate and the y-axis denotes True Positive Rate. The more the area under the curve, the better. Since the area under the curve is equal to 1 (square shape) the model is acceptable. (f1 score = 1)

By studding the confusion matrix, accuracy, precision and recall are calculated. The result is presented in the figure.

Logistic Regression

Notice how the x-axis denotes False Positive Rate and the y-axis denotes True Positive Rate. The more the area under the curve, the better. Since the area under the curve is equal to 1 (square shape) the model is acceptable. (f1 score = 1)

By studding the confusion matrix, accuracy, precision and recall are calculated. The result is presented in the figure.



By analyzing the above figures, below represents the performance of each model:

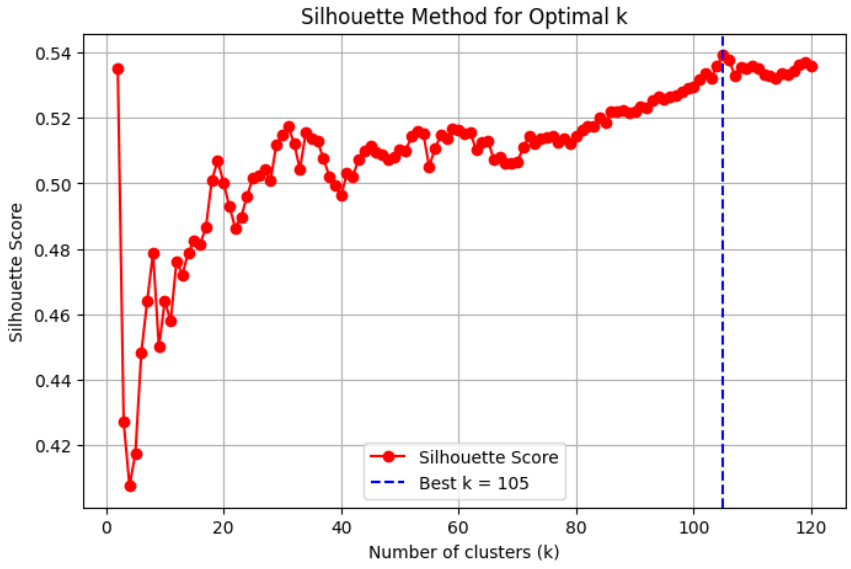
1. Logistic regression, SVM, AdaBoost

2. MLP

3. KNN

6. Clustering

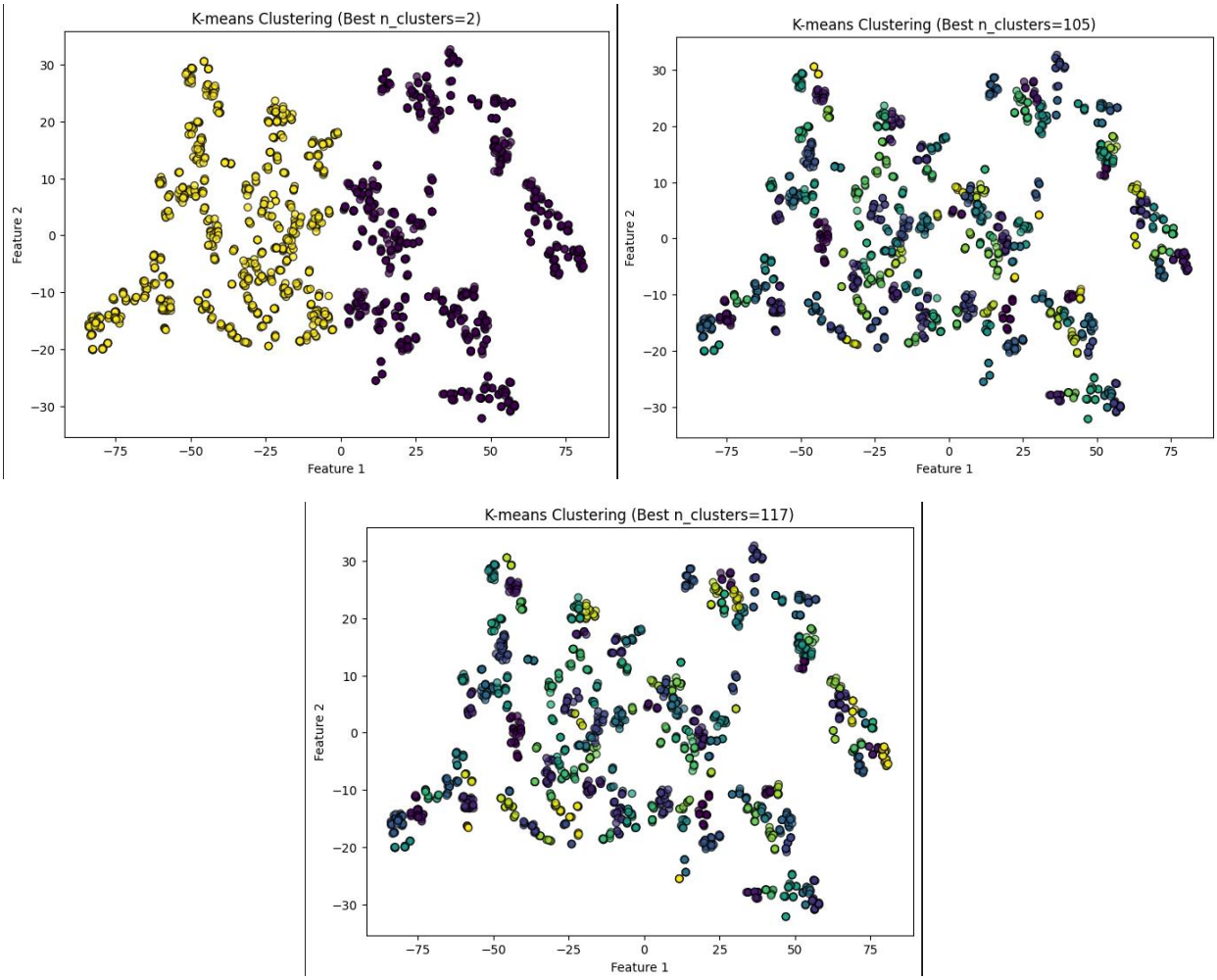
In this section, we set aside the labels and perform clustering on the data. First, we need to determine the number of clusters. To do this, we use the silhouette score method, calculating and comparing the silhouette score for different numbers of clusters. The corresponding chart is shown below:



t-SNE has been applied to the samples, and now, our feature vector length is 2.

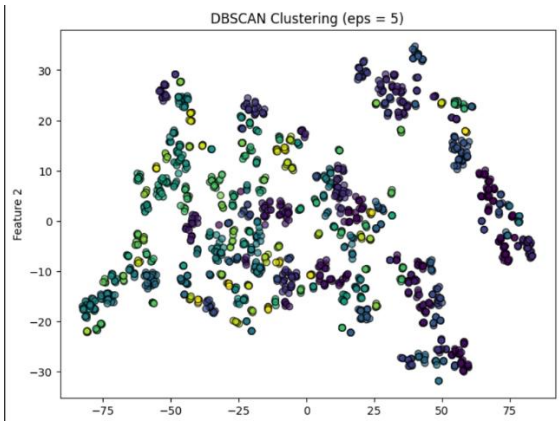
From this chart, it is interpreted that the optimal number of clusters is 105. We perform clustering using this number of clusters, as well as with 2 clusters and 117 clusters, using three different methods. These two values represent the number of classes we had earlier in gender detection and voice authorization.

K-Means

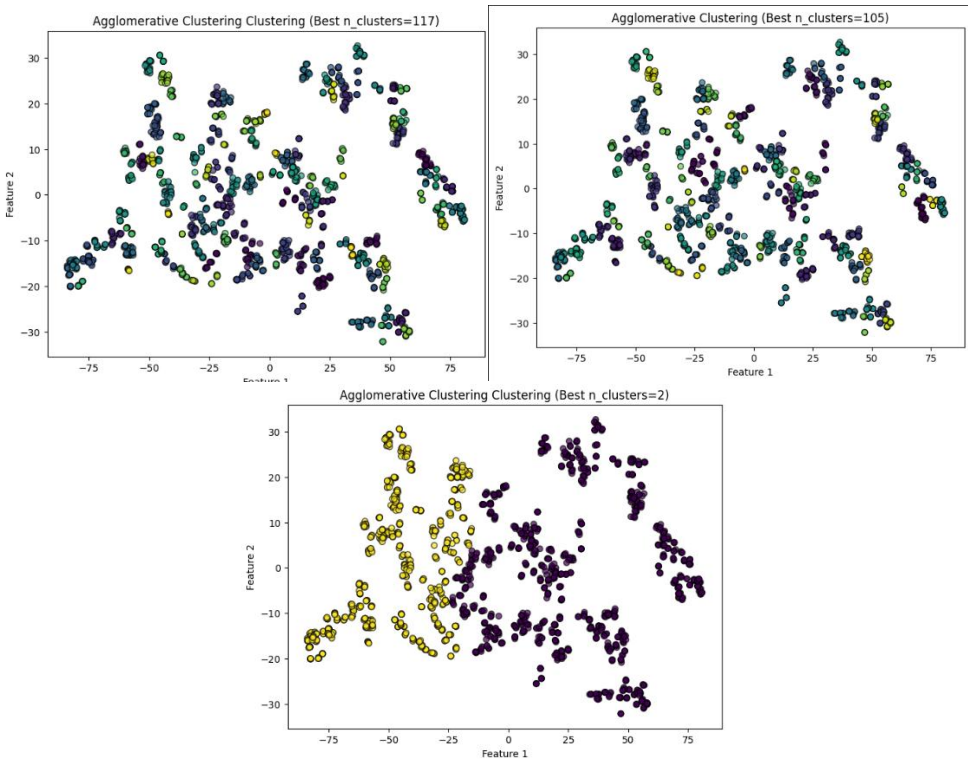


DBSCAN

The number of clusters in the DBSCAN method is not predefined; instead, it is determined based on the data, epsilon, and min-samples parameters.



Agglomerative Clustering



Results Analysis

Clustering with two classes does not reveal a meaningful relationship between class members. However, we previously observed that combining a large number of extracted features led to excellent separability in voice authentication. Thus, with 105 or 110 clusters, the clustering appears to be effective, suggesting that the samples in each cluster likely correspond to the voice of a specific person. In fact, the audio features of the data within a cluster are highly similar to each other.