Speaker Verification

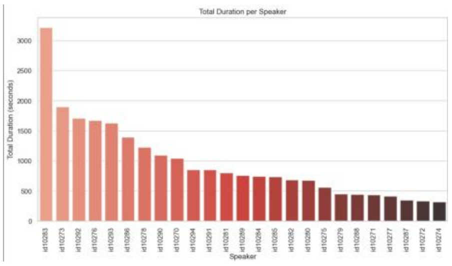
(Group 8)

1. Dataset Description:

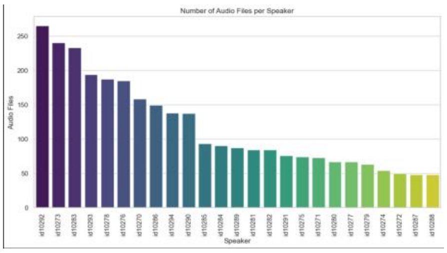
- **Number of Files**: 25 speakers, which include 2944 audio files
- **File Format**: .wav (Waveform Audio File Format)
- **File Duration**: All files have a different time duration.
- **Content**: The audio files contain the speech of different speakers at different speeds/pitches, etc.

2. Methodology:

- **Preprocessing**: Audio Normalization: Ensuring all audio files are sampled at a consistent rate (e.g., 16000 Hz) to maintain uniformity.
- **Segmentation**: Audio files are split into smaller, fixed-length segments (3 seconds and 8 sec). This allows for handling variable-length audio files while ensuring sufficient data for training.
- **Padding**: Shorter audio segments are padded with zeros to meet the required length. This avoids issues with variable-length input and ensures consistent feature extraction across all segments.
- **Noise Reduction**: Unwanted noise is minimized using techniques like band-pass filtering, which improves the quality of the extracted features.



- **Feature Extraction**: Key features like MFCCs (Mel-frequency cepstral coefficients) are extracted from each audio segment to represent the speaker's voice characteristics.



3. Exploratory Data Analysis (EDA):

Dataset Overview: The dataset contains audio files with a total duration of 24,330.31 seconds from 25 unique IDs (speakers).

Audio File Statistics:

Total Files: 2,944 Average

Duration: 8.26 seconds

Standard Deviation: 5.87 Seconds

Minimum Duration: 3.96 seconds

Median Duration: 6.42 seconds

Maximum Duration: 69.04 seconds

Duration Distribution: Most files have durations between 4.88 and 9.25 seconds, with a few outliers up to 69 seconds.

Challenges:

The project faced challenges such as time mismatches in audio duration, variations in pitch and tone, and background noise, which affected feature quality and model performance. Additionally, data imbalance introduced biases during training. Addressing these issues in the future could significantly enhance accuracy and robustness.

4. Models:

1. **Random Forest**: Ensemble learning method to enhance accuracy using multiple decision trees.
2. **K-Nearest Neighbors (KNN)**: Non-parametric model using proximity-based prediction.
3. **SVM(Based on similarity):** for Speaker Verification: The code performs speaker verification by extracting audio features and computing distance metrics between pairs of files. It trains machine learning models (SVM, Random Forest, KNN) on labeled pairs (same/different speakers) with hyperparameter optimization. The best model is evaluated using ROC-AUC and saved for future use.
4. **Gaussian Mixture Models (GMMs):** for each speaker using their audio data. It processes audio files, extracts MFCC features, trains a GMM for each unique userId, and stores all

models in a dictionary. The combined models are saved as a serialized .pkl file, with progress logged throughout the process.

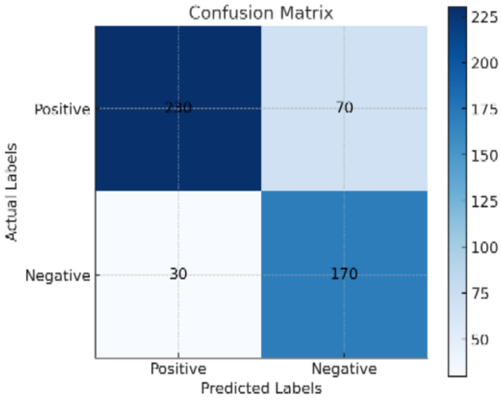
5. **Pairwise training:** Preprocessed audio files, extracted audio features, and trained random forest model with pairwise audio files labeled 1 or 0.

Analysis:

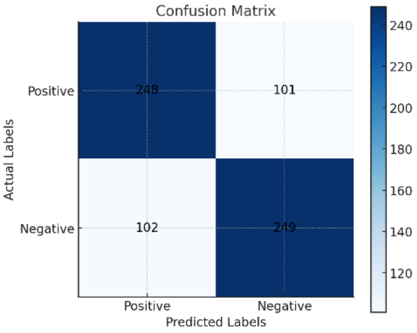
This project uses Gaussian Mixture Models (GMMs) and MFCC features to implement a speaker verification system. Audio files are processed to extract speech features, and a GMM is trained for each speaker. During verification, the system compares the features of two audio files and identifies whether they belong to the same speaker based on log-likelihood scores. The approach is practical for distinct voices but may face challenges distinguishing similar-sounding speakers, with potential feature extraction or model complexity improvements for better accuracy.

5. Results:

Seen Users:



Unseen Users:



6. Conclusion:

The traditional machine learning models trained on the dataset yielded low accuracy for speaker verification, highlighting the limitations of such approaches for this task. Factors like background noise, pitch variations, and dataset inconsistencies may have contributed to the suboptimal performance. This underscores the need for advanced techniques, such as deep learning models or feature engineering, to capture the complexities of speaker characteristics better and improve verification accuracy in future implementations.

7. Tools & Libraries:

Librosa, **Scikit-learn**, **Matplotlib**, **Pandas**, **Numpy**, **Scipy**, **Pickle.**

8. Contribution:

1. Literature survey: Nakul, Aniket.

2. EDA: Vardhana, Aniket.

3. Pre-Processing: Bikrant, Nakul.

4. Feature extraction: Aniket, Bikrant, Vardhana.

5. Model Training experiments: Aniket, Bikrant, Vardhana, Nakul.

6. Report and Presentation: Nakul.

9. References:

[1]. Sainburg, Tim. (2019). timsainb/noisereduce: v1.0 [Software]. Zenodo. https://doi.org/10.5281/zenodo.3243139

[2]. Sharma, Abhishek Manoj, "Speaker Recognition Using Machine Learning Techniques" (2019). Master's Projects. 685. DOI: https://doi.org/10.31979/etd.fhhr-49pm