Q1. Speech Enhancement and Speaker Verification in Multi-Speaker Scenarios

Name - Prabhat Ranjan
Roll No. - M23CSA017
Git- https://github.com/M23CSA017/speech_asgn_2

This assignment focuses on speaker verification in multi-speaker environments, which is a common challenge in real-world applications. When multiple people speak at the same time, it's harder to recognize who is speaking. To address this, we use:

A pre-trained wav2vec2 model for speaker verification.
Fine-tuning techniques like LoRA and ArcFace to improve performance.
A pre-trained SepFormer model to separate mixed speech.
Evaluation of speaker verification both before and after separation.

Dataset Description

VoxCeleb1: Used for evaluation. It includes 1,251 speakers and 150,000 utterances from YouTube interviews.

VoxCeleb2: Used for training. It contains 6,112 speakers and over 1 million utterances. Fine-tuning was done on the first 100 speakers. Multi-speaker training used speakers 1–50, and testing used speakers 51–100.

Q1.2 Pre-trained Model Evaluation

The wav2vec2 XLSR model was evaluated on VoxCeleb1-cleaned data:

Metric	Value
Equal Error Rate (EER)	47.93%
TAR @ 1% False Acceptance	2.46%
Speaker Identification Accuracy	34.56%

The results show weak performance, suggesting the need for fine-tuning.

Fine-Tuning with LoRA and ArcFace:

Methodology

LoRA (Low-Rank Adaptation): Injects low-rank trainable parameters into the transformer layers to reduce computational overhead.

ArcFace Loss: Enforces larger angular margin between different speaker embeddings to improve discriminability.

Hyperparameters

Key hyperparameters used for fine-tuning:

1. Batch Size: 8
2. Learning Rate: 3e-5
3. LoRA Parameters: Rank (r) = 8, Alpha = 8, Dropout = 0.05

The model was fine-tuned on VoxCeleb2 using LoRA and ArcFace.

Metric	Pre-trained	Fine-tuned
Equal Error Rate (EER)	47.93%	41.45%
TAR @ 1% FAR	2.46%	7.99%
Identification Accuracy	34.56%	63.70%

All metrics improved after fine-tuning.

Q1.3 Speech Separation Using SepFormer

The SepFormer model was used to separate overlapping speech in the test set. Metrics were computed using the 'mir_eval and pesq' libraries to evaluate signal quality and perceptual clarity. The evaluation covered Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), Signal-to-Artifacts Ratio (SAR), and Perceptual Evaluation of Speech Quality (PESQ). Statistical summaries are presented below:

Metric	Mean	Standard Deviation
SDR	3.73 dB	7.32 dB
SIR	19.38 dB	11.34 dB
SAR	5.21 dB	4.45 dB
PESQ	1.22	0.25

What this means:

SIR shows that SepFormer did a good job separating speakers.
SDR and SAR show that there's still some distortion and artifacts.
PESQ is low, meaning the audio doesn't sound very clear to humans.

Post-Separation Verification

After separating the mixed audio using SepFormer, we tested speaker identification using both the fine-tuned and pre-trained models. The separated audio files were processed one by one to check which speaker they belong to.

Model	**Rank-1 Accuracy**
Fine-tuned	**53.50%**
Pre-trained	**44.50%**

The fine-tuned model outperformed the pre-trained model by 9%, showing that fine-tuning helped the model learn stronger and more specific speaker features. Even though the speech was distorted after separation, the fine-tuned model handled it better. This indicates that fine-tuning made the model more effective, even in harder, noisier conditions.

Analysis:

The results clearly show that fine-tuning the wav2vec2 XLSR model with LoRA and ArcFace significantly improved speaker verification performance in single-speaker conditions. The Equal Error Rate dropped from 47.93% to 41.45%, and identification accuracy rose from 34.56% to 63.70%, confirming the benefits of adapting the model to the speaker domain.

In multi-speaker scenarios, after separation with SepFormer, the fine-tuned model achieved 53.50% Rank-1 accuracy, compared to 44.50% for the pre-trained model. This 9% improvement shows that fine-tuning helped the model generalize better, even in the presence of separation artifacts.

However, the overall accuracy remained lower than in clean conditions. The average SDR of 3.73 dB and PESQ of 1.22 highlight that while SepFormer suppressed interference well (SIR = 19.38 dB), it introduced distortions that affected speech quality. These artifacts likely impacted speaker embeddings and made verification more difficult.

Key factors impacting performance:

1. Domain Mismatch: Fine-tuning was done on clean speech but tested on separated, distorted audio.
2. Lack of Robust Augmentation: No noise or overlap was introduced during training, limiting real-world adaptability.
3. Signal Artifacts: Even after separation, residual noise and distortion affected embedding quality.

To improve performance, future work should include data augmentation with overlaps, and explore joint training for both separation and verification to handle noisy multi-speaker scenarios more effectively.

Q 2. MFCC Feature Extraction and Comparative Analysis of Indian Languages

Task A: MFCC Feature Extraction and Comparative Analysis

Dataset Overview

The dataset used for this analysis was sourced from Kaggle under the title Audio Dataset with 10 Indian Languages. It contains 10,000 audio samples spanning 10 different languages. For this study, audio samples from three languages were selected:Hindi, Punjabi and Telugu.
Each language was represented by 5 samples for MFCC visualization and statistical analysis.

MFCC Extraction and Visualization

MFCC Extraction: MFCCs were extracted using the Librosa library, with 13 coefficients extracted per frame.
Representative Samples: MFCC spectrograms were generated for a representative set of 5 samples from each language.

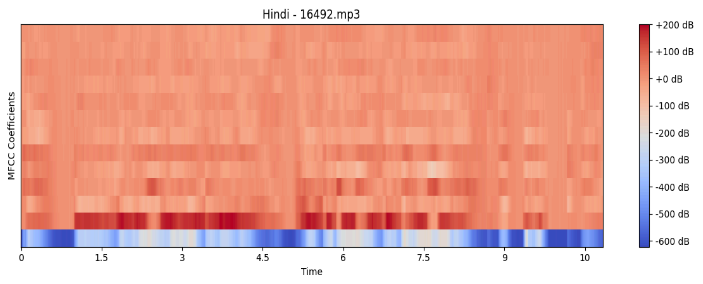
MFCC Spectrograms for Selected Languages:

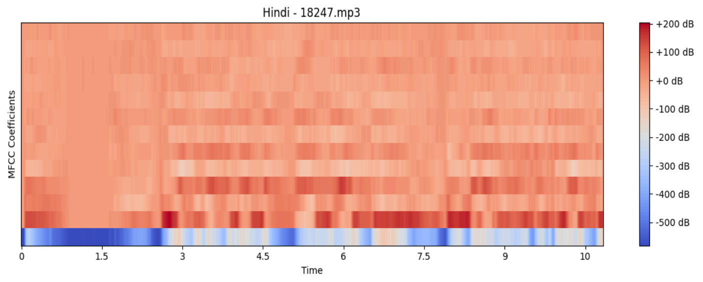
Hindi: Shows moderately spaced formants with denser patterns at lower frequencies.
Punjabi: Exhibits concentrated energy bands around 0–3 kHz, reflecting distinct phonetic patterns.

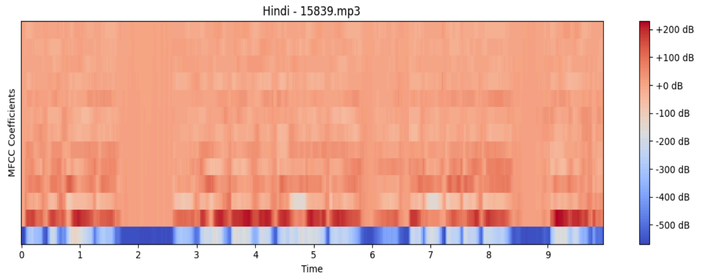
Telugu: Energy is distributed more evenly across time, with relatively clear formant transitions.

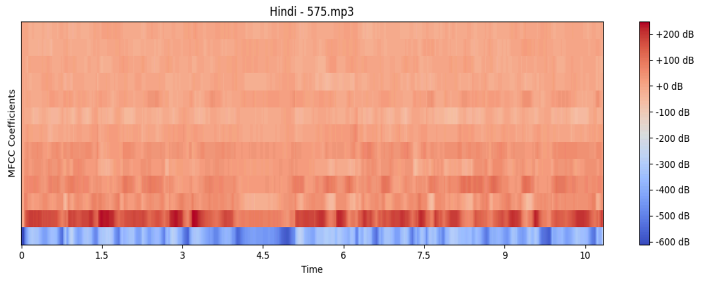
Hindi

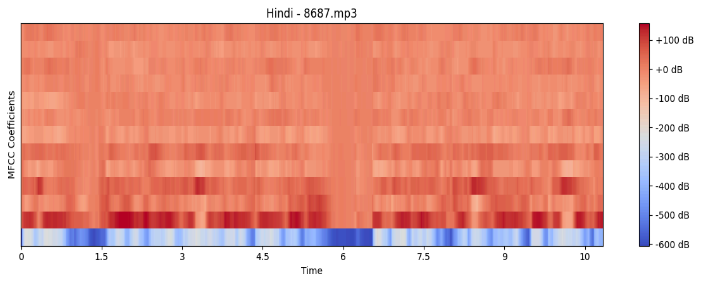
Punjabi

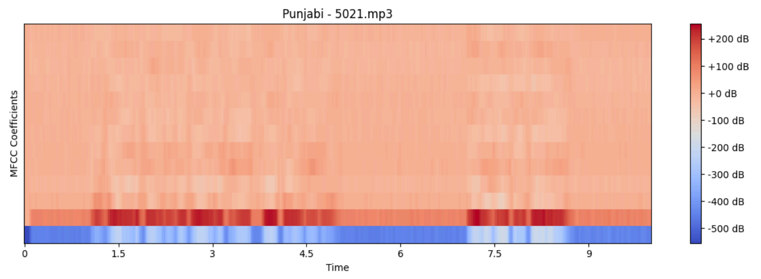


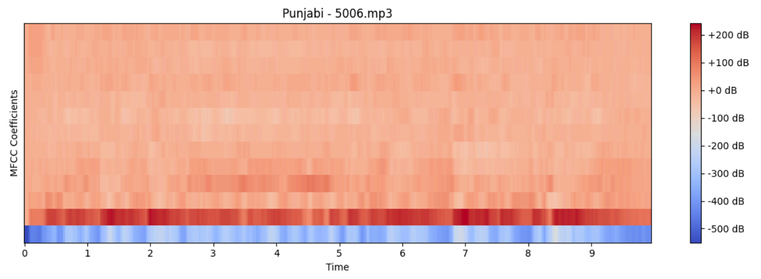


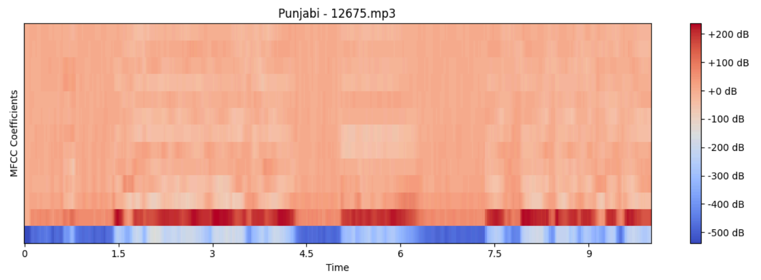


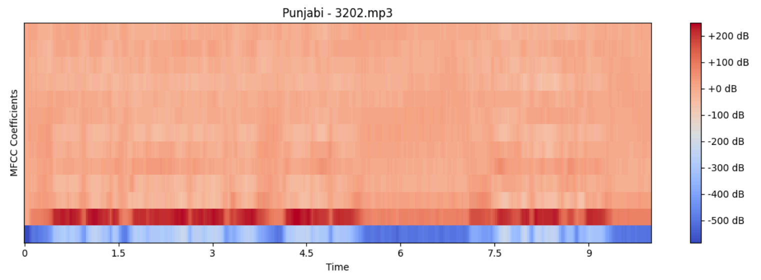


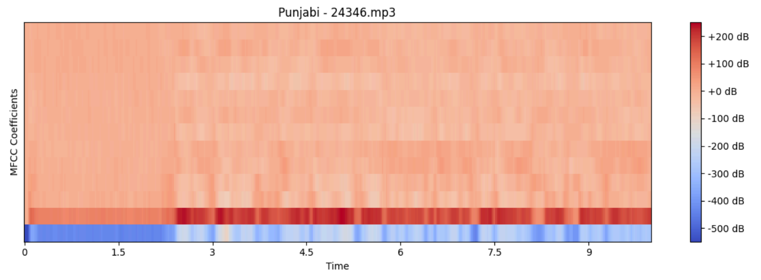




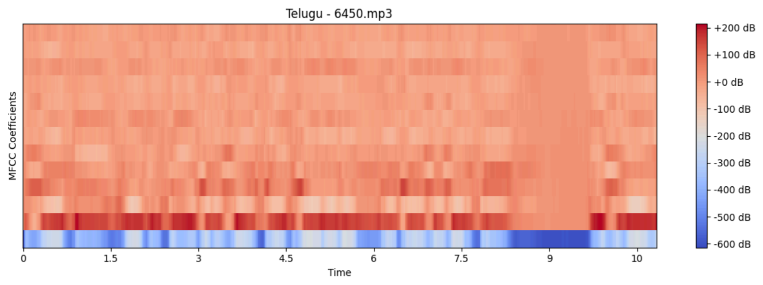


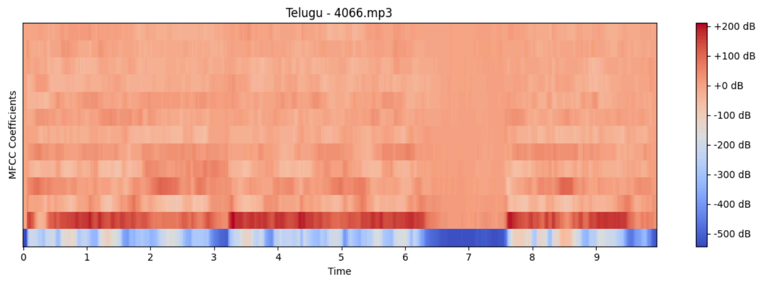


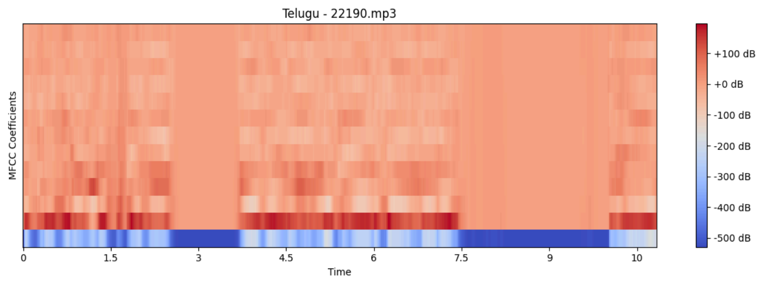


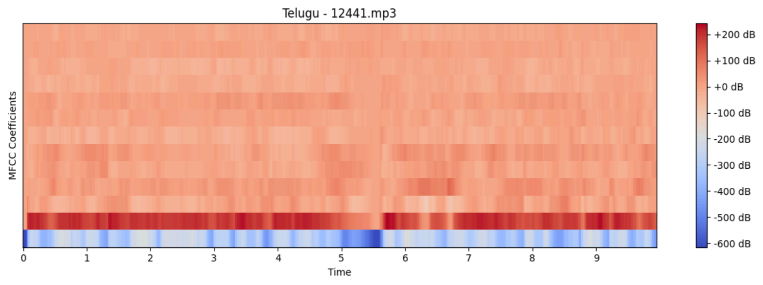


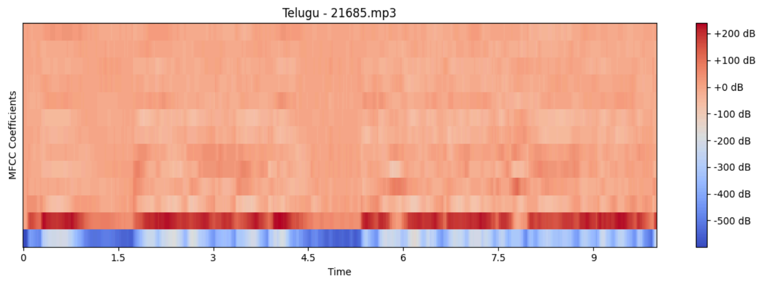
Telugu











Comparative Analysis of MFCC Spectrograms

Description:

Visual Patterns:All three languages show strong energy concentration in the lower frequency bands, reflecting the dominance of voiced phonemes**.**

Transition regions in the spectrograms align with diphthongs and plosive consonants across languages.

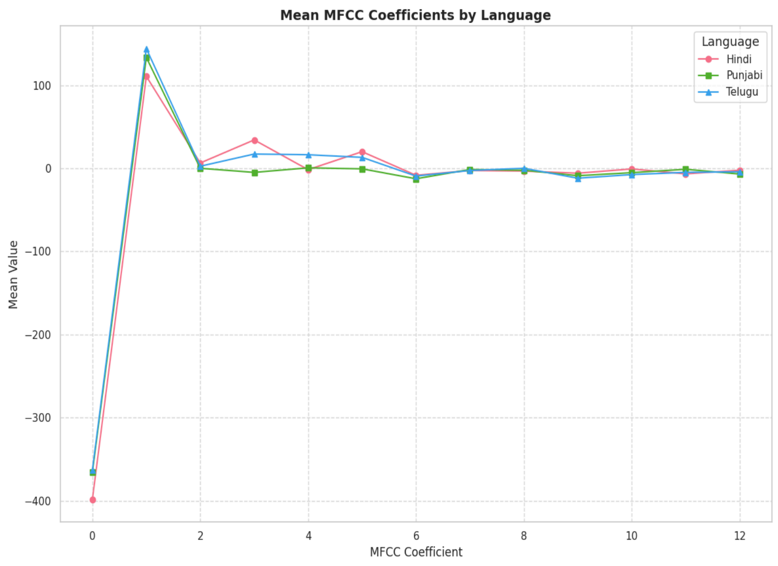
Key Differences:

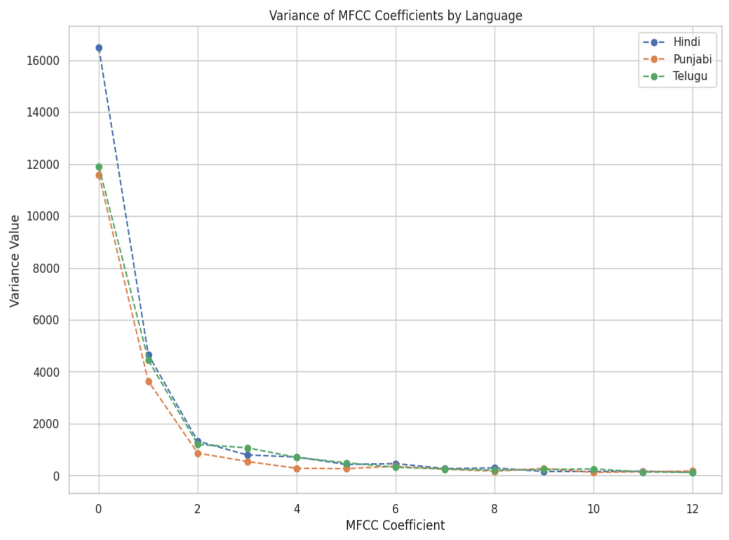
Hindi: Denser spectral structure with more variations.

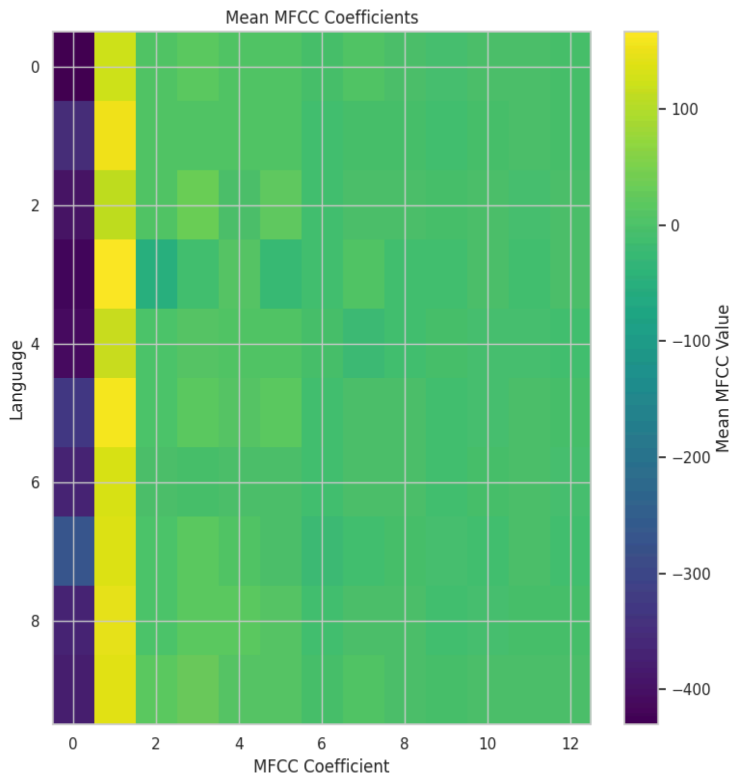
Punjabi: Higher variability across frames, reflecting tonal variations.

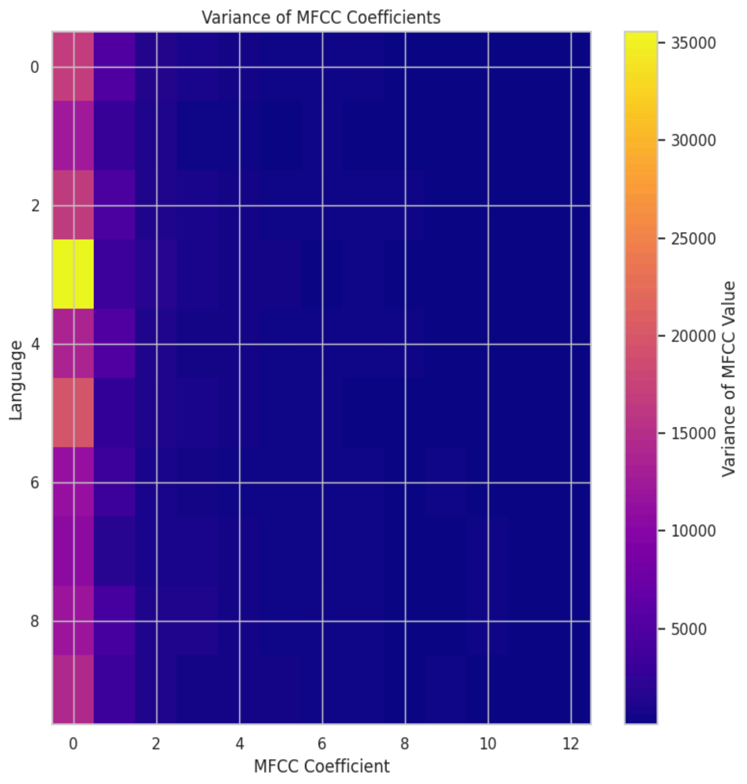
Telugu: Clearer formant

transitions suggesting distinct phonetic boundaries.









Analysis of the above plots:

Variance Analysis: Hindi exhibits the highest variance in the initial coefficients, while Punjabi and Telugu show relatively lower variance.

Mean Analysis: The first MFCC coefficient dominates the mean values across all languages, capturing the signal's energy.

Heatmap Analysis: Mean and variance heatmaps reveal that the majority of spectral energy and variability is captured by the first few MFCCs, with less impact from higher coefficients.

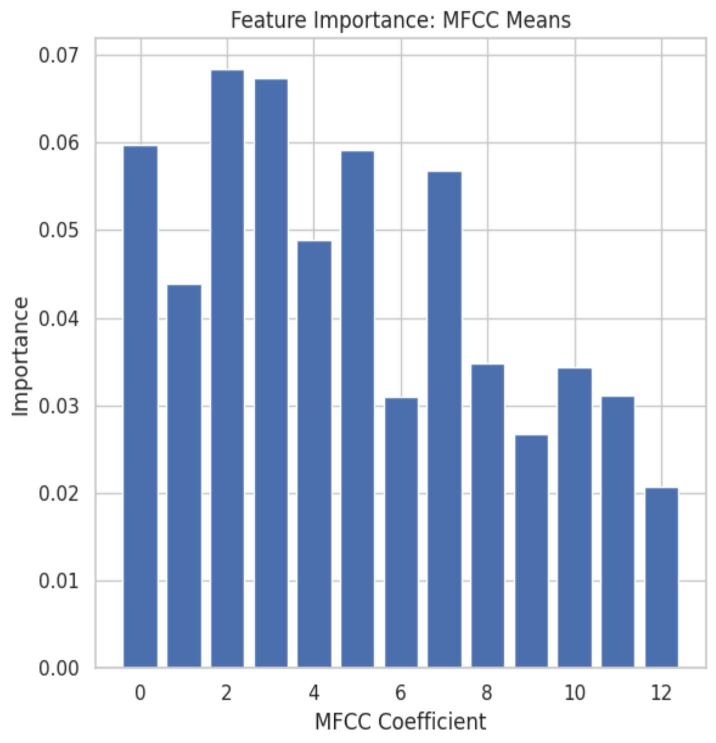
Task B: Classifier Implementation and Evaluation

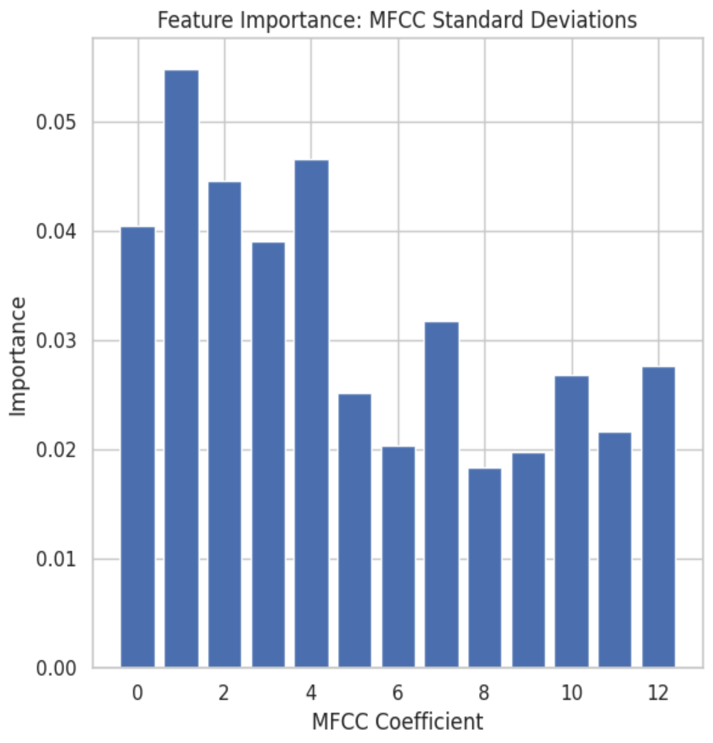
Feature Extraction for Classification

Feature Matrix:
A feature matrix was created using the mean and variance of MFCC coefficients. The matrix had dimensions (10000, 26), where:

13 features captured the mean MFCC coefficients.
13 features captured the variance of MFCC coefficients.

Figures



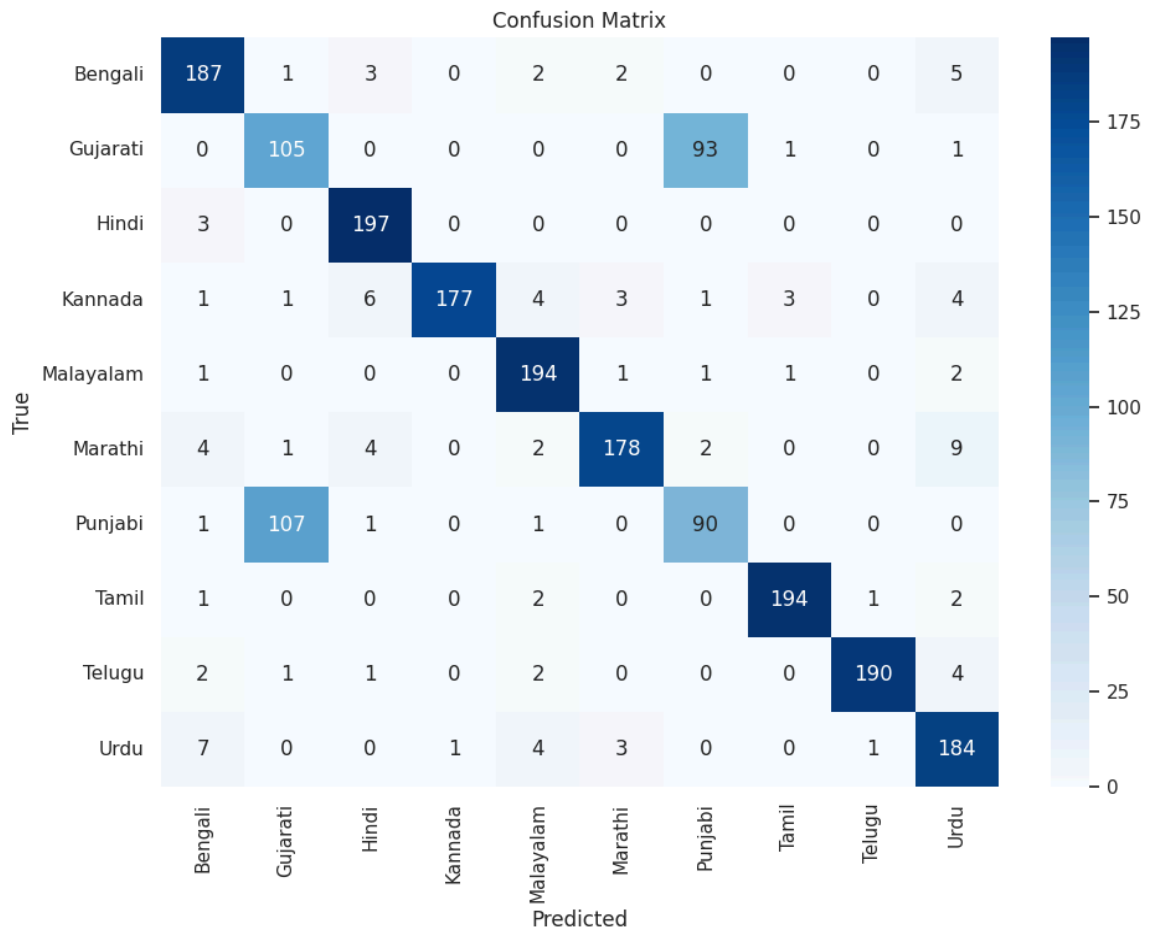


Model Selection and Training

Model:A Random Forest Classifier was chosen because it handles multi-class classification tasks effectively.

Accuracy:The model achieved an impressive 84.80% accuracy on the test data.

Confusion Matrix



	Bengali	Gujarati	Hindi	Kannada	Malayalam	Marathi	Punjabi	Tamil	Telugu	Urdu
Bengali	187	1	3	0	2	2	0	0	0	5
Gujarati	0	105	0	0	0	0	93	1	0	1
Hindi	3	0	197	0	0	0	0	0	0	0
Kannada	1	1	6	177	4	3	1	3	0	4
Malayalam	1	0	0	0	194	1	1	1	0	2
Marathi	4	1	4	0	2	178	2	0	0	9
Punjabi	1	107	1	0	1	0	90	0	0	0
Tamil	1	0	0	0	2	0	0	194	1	2
Telugu	2	1	1	0	2	0	0	0	190	4
Urdu	7	0	0	1	4	3	0	0	1	184

The confusion matrix shows that the model performed well with Hindi, Tamil, and Telugu, while Punjabi and Gujarati had more classification errors.

Classification Results and Performance Analysis

Performance Highlights:

High-Performing Languages:
Hindi (F1: 0.96) and Tamil (F1: 0.97) showed excellent performance.
Telugu (F1: 0.97) also performed very well.
Low-Performing Languages:
Punjabi (F1: 0.47) and Gujarati (F1: 0.50) performed poorly, likely due to accent overlaps.

Challenges and Limitations

Speaker Variability

Challenge: Differences in pitch, tone, and speaking style affect MFCC patterns since they capture speaker-specific vocal traits.
Impact: This variability makes it difficult to maintain consistency, especially when working with audio from diverse speakers.
Solution: Techniques like Cepstral Mean and Variance Normalization (CMVN) help reduce these differences, but they can't fully eliminate the problem, especially in multilingual datasets.

Background Noise

Challenge: Noisy environments introduce unwanted frequency components that distort MFCC features, leading to misclassifications.
Impact: Studies show that MFCC-based systems can lose up to 80% accuracy in low Signal-to-Noise Ratio (SNR) conditions.

Solution: Noise reduction methods like spectral subtraction and wavelet denoising can help, but using Mel-Spectrograms with deep learning models (like CNNs or RNNs) tends to improve robustness in noisy settings.

Regional Accents and Dialects

Challenge: Dialectal variations (e.g., Telugu dialects or variations in Hindi) alter phonetic structures, making it hard for MFCCs to capture subtle differences.
Impact: Accent variations often lead to misclassifications, especially in diverse multilingual datasets.
Solution: Fine-tuning models with accent-specific data or using hybrid models that combine MFCCs with prosodic features can improve performance. Multi-task learning can also help models differentiate between accents and speaker traits.

Key Limitations

Language Mismatch:
MFCC models often struggle when tested on languages they haven't seen before, leading to poor generalization in cross-lingual tasks.
Low-Resource Languages:
Many Indian languages and dialects lack sufficient training data, making it difficult to build accurate models for them.
Non-Stationary Noise:
Real-world environments introduce unpredictable noise, making it hard for MFCC-based models to perform reliably in such scenarios.

References:

1. SpeechBrain – A toolkit for speech processing, used here for the pre-trained SepFormer model for speaker separation. https://speechbrain.github.io.
2. PEFT (Parameter-Efficient Fine-Tuning) – A library for efficiently fine-tuning large models using methods like LoRA.(https://github.com/huggingface/peft).
3. Hugging Face Transformers – A library that provides pre-trained models (e.g., Wav2Vec2) and tokenizers for natural language and audio processing tasks.