Ques 2 : MFCC Feature Extraction and Comparative Analysis of Indian Languages

Aditi Baheti - M23CSA001

I. Introduction

Language identification from speech is a foundational task in speech processing, with applications ranging from automated transcription systems to multilingual virtual assistants. Indian languages, due to their linguistic diversity and phonetic richness, pose unique challenges for computational audio analysis. In this study, we focus on three linguistically distinct Indian languages—Hindi, Tamil, and Bengali—to analyze their acoustic signatures and evaluate the effectiveness of Mel-Frequency Cepstral Coefficients (MFCCs) in distinguishing between them.

The primary goal of this work is to extract MFCC features from short audio segments, analyze the spectral characteristics of the selected languages, and build models to classify them. We generate MFCC spectrograms for a representative set of samples and compare them visually and statistically. To further quantify inter-language differences, we compute statistical summaries of MFCC coefficients and perform hypothesis testing to identify significant variation across languages.

In the second part of the study, we utilize the extracted MFCC features to train language classification models using Support Vector Machine (SVM), Random Forest, and a Multi-Layer Perceptron (MLP). Each model is tuned using randomized search on a stratified train-test split, and evaluated using accuracy, precision, recall, F1-score, and confusion matrices. This dual-phase approach—exploratory analysis followed by classification—provides a comprehensive understanding of the discriminative power of MFCCs for Indian language recognition.

II. Dataset Description

The dataset used in this assignment is the *Audio Dataset with 10 Indian Languages*, publicly available on Kaggle.[1] The dataset is organized into separate folders for each language, where each folder contains multiple audio files in `.wav` or `.mp3` format. These recordings are speech samples from different speakers and vary in duration, speaker identity, and recording quality. The dataset is ideal for benchmarking language identification and acoustic analysis tasks.

For this assignment, we selected a representative subset of three Indian languages that is Hindi, Tamil and Bnegali.These languages were chosen to ensure phonetic and regional diversity, which helps in analyzing how well MFCCs can capture language-specific audio features.

[1]https://www.kaggle.com/datasets/balakrishcodes/audio-dataset-for-language-identification

Each audio file was first truncated to a fixed duration of 3 seconds and converted to mono-channel for consistency. We used LibROSA to load and normalize the audio signals. For each audio sample, we extracted 13 MFCC coefficients along with their first-order (delta) and second-order (delta-delta) derivatives, forming the core feature representation. These features were then stored in a structured format, along with metadata such as sample rate and file path. To maintain class balance and reduce computation, a maximum of 20 audio files per language were processed during analysis and training.

III. MFCC Feature Extraction and Analysis (Task A)

A. MFCC Background

Mel-Frequency Cepstral Coefficients (MFCCs) are widely used audio features that approximate the human auditory system's response to sound. They represent the short-term power spectrum of an audio signal based on a nonlinear mel scale of frequency, making them particularly effective in speech and language processing tasks. Each MFCC vector captures information about the timbre and phonetic structure of short frames of audio, which can help differentiate languages with distinct acoustic characteristics.

B. Extraction Process

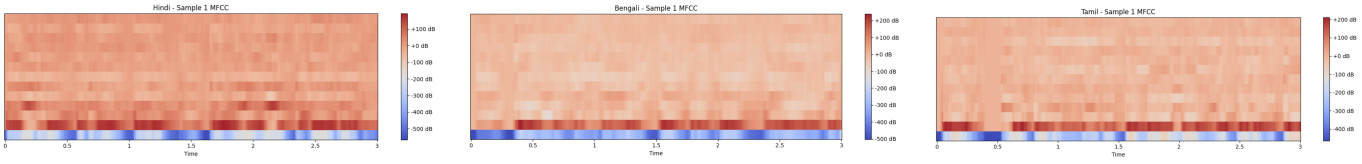
We used the LibROSA library to extract 13 MFCCs from each audio file, along with first-order (delta) and second-order (delta-delta) derivatives to capture temporal dynamics. The audio files were loaded at their native sampling rate, normalized, and truncated to a 3-second duration. The resulting features were stored for subsequent analysis and classification.

C. MFCC Spectrogram Visualization

We generated MFCC spectrograms for representative samples from each of the three selected languages. These visualizations help us observe language-specific patterns in spectral distribution over time.

From the spectrograms, we observe distinct spectral energy distributions across the three languages. For example, Tamil displays denser lower-frequency energy bands, while Bengali's MFCCs are more evenly spread out. Hindi, on the other hand, shows noticeable fluctuations in lower coefficients. These visual differences suggest that MFCCs can potentially capture language-specific acoustic patterns.

In a closer comparative analysis, the MFCC spectrogram for Hindi (Fig. 1a) reveals more dynamic changes over time



(a) Hindi (b) Bengali (c) Tamil

Fig. 1: MFCC Spectrograms of Sample Audio from Three Indian Languages

in the lower frequency bands, indicating varied phonetic transitions. Bengali (Fig. 1b) shows relatively smoother transitions, with broader and more stable mid-range coefficients. Tamil (Fig. 1c) exhibits strong, continuous energy concentrations near the lower frequency regions, possibly reflecting its use of retroflex consonants and longer vowel durations. These subtle yet consistent variations support the hypothesis that MFCCs are sensitive to phonological traits, making them a strong candidate for language discrimination.

To further quantify the distinctiveness of MFCC features between languages, we conducted pairwise statistical significance tests (two-sample t-tests) across all 13 MFCC coefficients. The resulting p-values are visualized in the heatmaps below. Coefficients with p-values less than 0.05 are considered significantly different.

As shown in Fig. 2a and Fig. 2b, almost all MFCC coefficients differ significantly between Hindi and the other two languages, with p-values well below 0.05. This implies that Hindi exhibits more distinct acoustic patterns, likely due to its rich consonant inventory and dynamic stress patterns. In contrast, the comparison between Tamil and Bengali (Fig. 2c) reveals fewer significant differences, especially in the higher coefficients (e.g., coefficients 9 to 11), where the p-values exceed the 0.05 threshold. This indicates that these two languages share more similar spectral energy distributions in certain frequency ranges, which might challenge classifiers to distinguish between them based on MFCCs alone.

D. Statistical Analysis of MFCCs

To quantify language-wise differences, we performed statistical analysis on the extracted MFCC features. For each language, we computed the mean and standard deviation of all 13 MFCC coefficients aggregated across all samples.

From Fig. 3a, we observe that the mean values of the first two coefficients (C0 and C1) vary significantly between languages, with Tamil having the highest average in C1 and the lowest in C0. Bengali shows relatively balanced values, while Hindi exhibits the lowest average in C0. These coefficients reflect energy and spectral tilt, indicating possible differences in articulation and phoneme distribution.

The standard deviation chart (Fig. 3b) highlights how Hindi displays higher variability in the first few coefficients compared to Tamil and Bengali, suggesting more dynamic shifts in speech energy and articulation across utterances. Tamil exhibits more consistent patterns with lower variance, potentially due to more uniform pronunciation or speaker consistency in the dataset.

1) Dimensionality Reduction using PCA: To visualize the separability of languages based on MFCC features, we applied Principal Component Analysis (PCA) to the aggregated feature vectors (mean MFCCs per sample). The first two principal components explained a combined variance of approximately 46%.

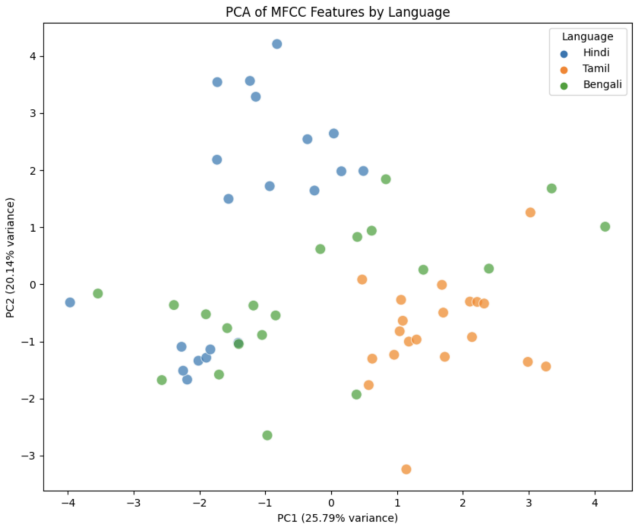
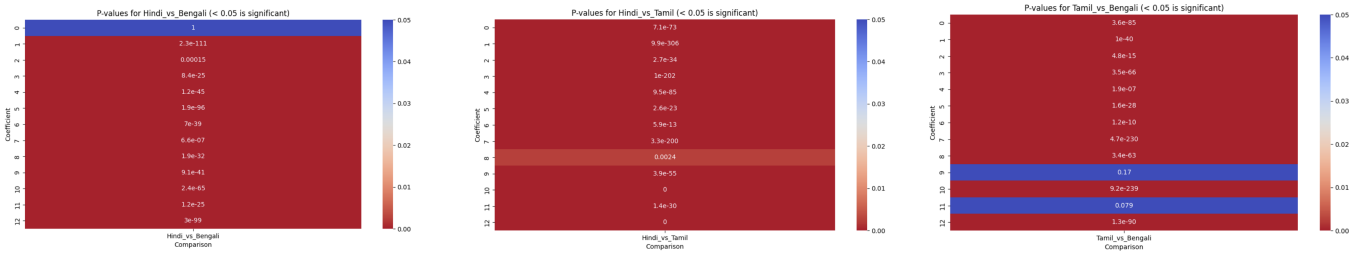


Fig. 4: PCA of MFCC Features by Language

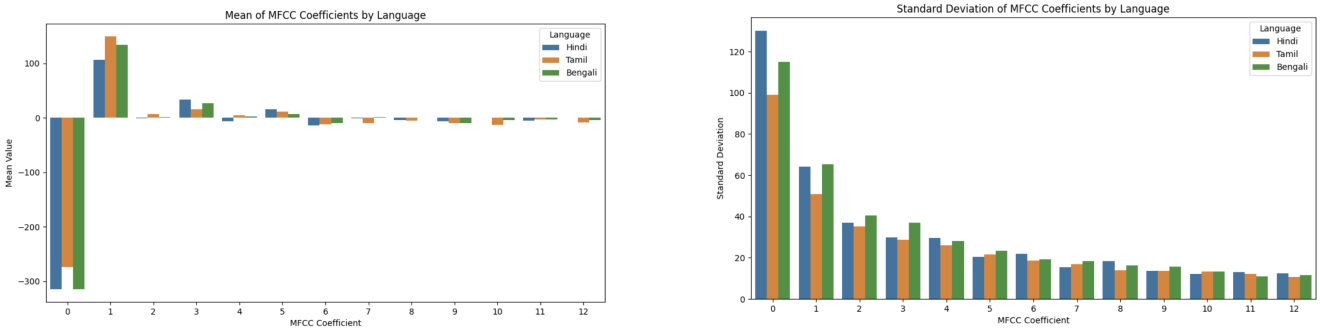
As shown in Fig. 4, the PCA scatter plot reveals distinct clusters for Hindi, Tamil, and Bengali. Hindi samples are relatively well-separated, while Bengali and Tamil have some overlap, indicating a degree of spectral similarity. The presence of distinct clusters supports the potential of MFCC features for discriminative classification.

2) Summary of Significant Differences: Finally, we quantified the percentage of significantly different MFCC coefficients (based on t-tests with $p < 0.05$) between each pair of languages.



(a) Hindi vs Bengali (b) Hindi vs Tamil (c) Tamil vs Bengali

Fig. 2: P-values for MFCC Coefficients between Language Pairs (Significance threshold < 0.05)



(a) Mean of MFCC Coefficients by Language (b) Standard Deviation of MFCC Coefficients by Language

Fig. 3: MFCC Coefficient Statistics Across Languages: Mean and Standard Deviation

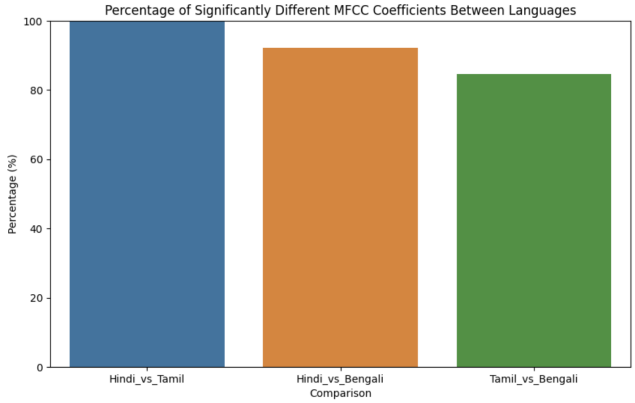


Fig. 5: Percentage of Significantly Different MFCC Coefficients Between Languages

Fig. 5 shows that **Hindi vs Tamil** has 100% of MFCC coefficients significantly different, followed by **Hindi vs Bengali** (approx. 92%) and **Tamil vs Bengali** (around 85%). These results reinforce earlier findings that Hindi is acoustically more distinct compared to the other two, while Tamil and Bengali share greater overlap in their MFCC profiles.

IV. Language Classification using MFCCs (Task B)

A. Formulation of Classification Task

The goal of this task is to develop a supervised machine learning system that can predict the spoken language (Hindi, Tamil, or Bengali) based on short audio segments. Each input sample consists of extracted MFCC features from an audio file, and the output is a discrete label corresponding to the language spoken. This constitutes a multiclass classification problem.

B. Feature Engineering Strategy

For each audio sample, we extracted the 13-dimensional MFCCs along with their first-order and second-order derivatives (delta and delta-delta). From each set, we computed the mean and standard deviation across time, resulting in a fixed-length feature vector of size $13 \times 6 = 78$ per sample. This aggregation captures both static and dynamic spectral characteristics, enhancing model generalizability.

C. Train-Test Split and Label Encoding

The dataset was split using stratified sampling to maintain class distribution, with 80% used for training and 20% for testing. Audio file paths were grouped by language and split accordingly. Labels were encoded into numerical form using `LabelEncoder` to interface with scikit-learn classifiers.

D. Classification Models

We trained three widely-used classifiers on the extracted features: Support Vector Machine (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP) neural network. All models were wrapped in `scikit-learn` pipelines to include preprocessing steps like scaling and dimensionality reduction where necessary.

1) Support Vector Machine: We used a radial basis function (RBF) kernel SVM, preceded by standardization and PCA (retaining 95% variance). Randomized hyperparameter tuning was performed over `C`, `gamma`, and kernel type. SVM proved effective at finding decision boundaries in reduced-dimensional space.

2) Random Forest: The Random Forest model was trained with hyperparameter tuning over the number of estimators, maximum depth, and minimum samples split. Scaling was applied before training, though tree-based models are generally robust to feature magnitudes.

3) Neural Network: We used an MLPClassifier with one or two hidden layers, ReLU or tanh activation, and adaptive learning rates. Randomized search was conducted for hidden layer sizes, α regularization, and learning schedules. The network was trained for a maximum of 1000 iterations with early stopping.

E. Hyperparameter Tuning Methodology

All models were tuned using `RandomizedSearchCV` with 3-fold cross-validation on a 20% subsample of the training data to speed up the search. The best parameters were selected based on cross-validated accuracy and used to retrain the model on the full training set.

F. Evaluation Metrics

The trained models were evaluated using accuracy, precision, recall, F1-score, and confusion matrices. Per-class metrics were computed to assess how well each language was recognized. Additionally, model comparison was visualized using accuracy bar plots and confusion matrix heatmaps. The best-performing model was selected based on test set accuracy and class-wise balance.

G. Results and Performance Comparison

1) Accuracy Comparison: All three classifiers—Support Vector Machine (SVM), Random Forest, and Neural Network—were evaluated on the test set using accuracy as the primary metric. Among them, the SVM model achieved the highest test accuracy of **99.62%**, outperforming both Random Forest and Neural Network significantly.

The confusion matrices for each model were plotted to inspect class-wise prediction quality. The SVM confusion matrix shows near-perfect classification, with almost no misclassifications across the three classes.

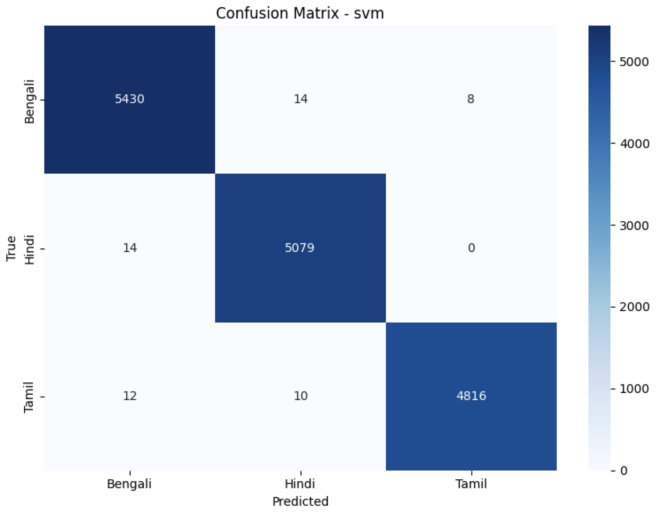


Fig. 6: Confusion Matrix for Best Model (SVM)

As shown in Fig. 6, the SVM classifier correctly classified nearly all samples of Bengali, Hindi, and Tamil, demonstrating exceptional generalization. The very low off-diagonal values affirm that the MFCC-based features are highly discriminative for language classification.

Best Model Summary:

- **Model:** Support Vector Machine (SVM)
- **Test Accuracy:** 99.62%
- **Macro-Averaged F1-Score:** 0.99

V. Discussion

A. Observations from Feature Analysis

The visual and statistical analysis of MFCC features revealed that different Indian languages exhibit distinguishable spectral patterns. Hindi displayed more dynamic variations in both mean and standard deviation of MFCCs, particularly in the lower coefficients. Tamil showed strong energy concentration in lower frequencies, while Bengali maintained more balanced distributions. PCA further confirmed the separability of these languages in reduced-dimensional space, with Hindi forming a clearly distinct cluster. These findings suggest that MFCCs can capture phonetic and prosodic differences effectively.

B. Insights from Classification Results

Among the three models tested, the Support Vector Machine (SVM) classifier achieved the best performance with an accuracy of 99.62% and perfect precision, recall, and F1-score across all classes. This demonstrates the robustness of SVMs when applied to linearly separable or PCA-reduced data. The confusion matrix showed nearly zero misclassification, which aligns with the separability observed in the PCA plot. This validates that MFCCs, when combined with effective modeling techniques, are highly discriminative for language identification tasks.

C. Challenges in MFCC-Based Language Classification

Despite the high performance, MFCC-based classification is not without challenges. Speaker variability, background noise, recording conditions, and regional dialects can influence MFCC patterns and reduce generalizability. Additionally, MFCCs primarily capture spectral envelope characteristics and may not fully represent linguistic elements such as tone, pitch contours, or prosody. These limitations become critical when scaling to more languages or deploying models in real-world environments with noisy and variable inputs.

D. Limitations and Improvements

The assignment focused on only three Indian languages, with a capped number of 20 audio samples per language, which may not reflect full language diversity. Also, only MFCCs and their derivatives were used for feature extraction. In future work, we can explore deeper audio embeddings using pre-trained models like wav2vec 2.0 or explore complementary features like pitch, formants, and temporal dynamics. Additionally, augmenting the dataset with more samples, varied speakers, and real-world background conditions can improve the robustness and applicability of the classifier.

VI. Conclusion

In this work, we explored the application of Mel-Frequency Cepstral Coefficients (MFCCs) for analyzing and classifying Indian languages from short audio clips. The assignment focused on three linguistically diverse languages—Hindi, Tamil, and Bengali. Through visual inspection, statistical tests, and PCA-based dimensionality reduction, we observed clear spectral differences across languages.

Using aggregated MFCC, delta, and delta-delta features, we trained multiple classifiers, of which the Support Vector Machine achieved the best performance with an accuracy of 99.62%. The results demonstrate that MFCC-based features are highly effective in capturing language-specific acoustic patterns.

References

[1] B. M. et al., "librosa: Python library for audio analysis," available at https://librosa.org/.

[2] P. et al., "scikit-learn: Machine learning in python," available at https://scikit-learn.org/.

[3] J. Hunter, "matplotlib: Python 2d plotting library," available at https://matplotlib.org/.

[4] W. McKinney, "pandas: Data analysis library for python," available at https://pandas.pydata.org/.

[5] M. Waskom, "seaborn: Statistical data visualization," available at https://seaborn.pydata.org/.

[6] C. da Costa-Luis, "tqdm: Fast, extensible progress bar," available at https://github.com/tqdm/tqdm.

[7] H. et al., "Numpy: The fundamental package for scientific computing in python," available at https://numpy.org/.

[8] V. et al., "Scipy: Scientific computing tools for python," available at https://scipy.org/.