Language Recognition from Audio for Pan Indian Languages
Midsem Report : CSE343 Machine Learning, Monsoon 2024, IIIT-Delhi

Ritika Thakur 2022408 | Saksham Singh 2022434 | Sarthak Gupta 2022451 | Sidhartha Garg 2022499

Abstract

India's rich linguistic diversity, with 23 official languages and countless dialects, presents unique challenges for language recognition systems. This project aims to create an advanced system for recognizing Pan Indian languages by leveraging classical machine learning techniques. Starting with an extensive dataset of audio recordings, we focus on extracting rich features like MFCCs and Spectral to capture unique acoustic features. By applying SVM and exploring different kernels, our approach demonstrates significant potential in accurately classifying a wide range of Indian languages.

1. Introduction

In a country as diverse as India, effective language identification from audio is essential for improving communication technologies and making digital platforms more accessible in multilingual environments.

We want to develop a language recognition system for Pan Indian languages using classical machine learning techniques on a comprehensive dataset of audio recordings. By analyzing acoustic features like MFCCs and spectral properties, we aim to build models capable of accurately classifying languages from audio samples.

2. Literature Survey

- **Speech Recognition using MFCC** by S. Suksri, T. Yingthawornsuk.: This paper explores speech recognition using MFCC for feature extraction and Principal Component Analysis (PCA) for dimensionality reduction. SVM outperforms Maximum Likelihood (ML) classifiers, especially with larger MFCC samples, in recognizing spoken words.

- **Speaker Gender Recognition via MFCCs and SVMs** by Ernest Fokoue, Zichen Ma.: An algorithm involving MFCCs and SVMs is provided to perform speaker gender recognition and the RBF kernel is compared with polynomial kernel and considered as a better kernel function in this gender recognition task.

- **SVM-based audio scene classification** by Hongchen Jiang, Junmei Bai: This paper presents an approach that uses support vector machine (SVM) for audio scene classification, which classifies audio clips into one of five classes: pure speech, non-pure speech, music, environment sound, and silence.

- **Spoken Language Identification using Gaussian Mixture Model-Universal Background Model in Indian Context** by Sreedhar Potla, Vishnu Vardhan B.: The study identifies one of twenty-three Indian languages using Mel-Frequency Cepstral Coefficient (MFCC) features, showing improved accuracy with a GMM-UBM model compared to a standard GMM.

In "Speech Recognition using MFCC", SVM outperforms ML classifiers when combined with MFCC and PCA, which can help improve accuracy, through dimensionality reduction. "Speaker Gender Recognition via MFCCs and SVMs" demonstrates SVM's effectiveness, especially with the RBF kernel, suggesting experimenting with different kernels for better performance. "SVM-based Audio Scene Classification" reinforces SVM's suitability for audio tasks, while "Spoken Language Identification using GMM-UBM" highlights GMM-UBM's superior accuracy, offering a promising model for the language classification system.

3. Dataset Details and Pre-processing

We have used a dataset, created by Chaitanya Bharadwaj H B on Kaggle, containing approximately 257K five-second audio samples evenly distributed across 10 Indian languages: Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Punjabi, Tamil, Telugu, and Urdu. The audio samples were sourced from regional videos on YouTube.

3.1. Dataset Details

In our dataset the audio samples are mostly 5 seconds long, with outliers even reaching as low as 0 seconds. All the audio samples were sampled at a rate of 22050 bits/sec. For every class, the number of samples were: 'Kannada' : 22k, 'Telugu' : 23k, 'Malayalam' : 24k, 'Tamil' : 24k, 'Marathi' : 25k, 'Hindi' : 25k, 'Punjabi' : 26k, 'Gujarati' : 26k, 'Bengali' 27k, 'Urdu' : 31k. Upon manual inspection, we also noticed that 'Punjabi' class had mostly flute music, silent files or files containing audio of other languages like 'Gujarati'. Since, this class was corrupted in the dataset we chose to omit it for further pre-processing and model training.

We can analyse the details of amplitudes of audio files by visualising them using audio signals.

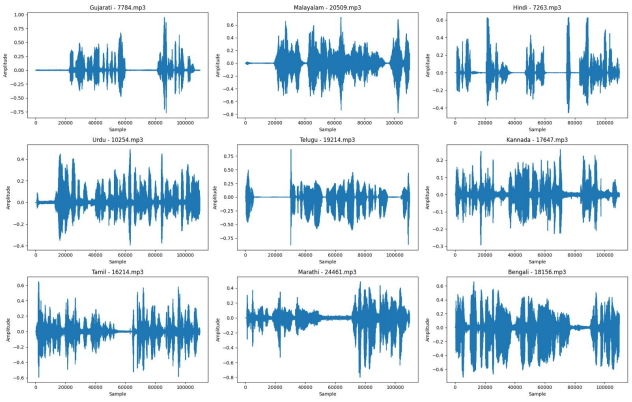


Figure 1. Audio signals of one instance from each class

We also used spectograms to visualise the frequency content of a signal over time, unlike simple audio signals which tell about the amplitude for a sample. They provide insights about how the sound energy is distributed about different frequencies and how it changes over time. From our spectograms we can tell that the data does not look too unclean.

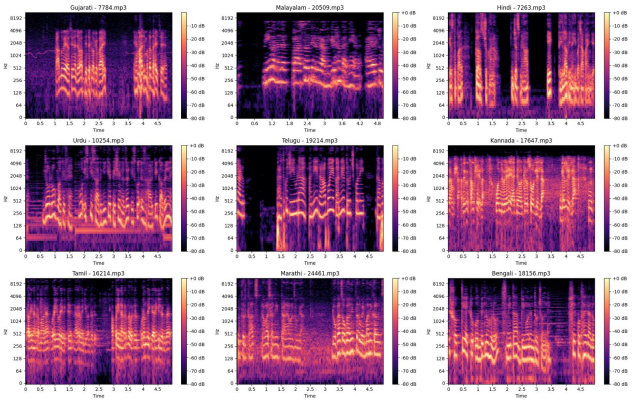


Figure 2. Spectograms of one instance from each class

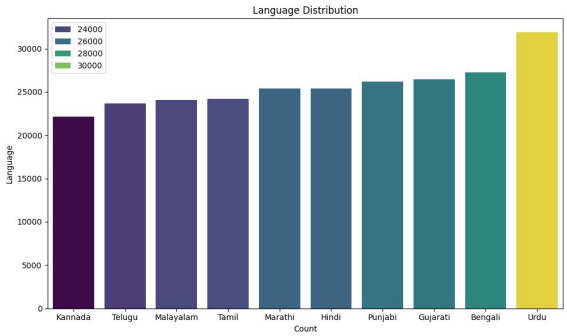


Figure 3. Classes are mostly balanced throughout the dataset

3.2. Data Pre-Processing

Data Cleaning

Our dataset consisted of audio files which were silent, noisy, and those which had silent segments. To handle these, we detected silent segments(those less than 20 dB) and removed them using the Librosa library. If an audio sample had a length of 0 seconds then we removed it. For samples with duration less than 5 seconds we padded them with zeroes, and those greater than 5 seconds were trimmed. All samples were normalised and noisy samples were removed by calculating the SNR(signal to noise ratio), with a noise threshold of 0.02, for each using the Librosa library.

Feature Extraction

We trained our models on three different feature sets, for an undersampled dataset of 45k samples (5k samples from each class) and chose the best one for the entire dataset.

Features	Feature Set 1	Feature Set 2	Feature Set 3
MFCC	15	25	13
MFCC Delta	15	25	NIL
MFCC Delta2	15	NIL	NIL
Chroma STFT	12	12	12
Spectral	5	5	1
ZCR	1	NIL	1
RMS	1	NIL	NIL
Mel Spectogram	1	NIL	NIL
Tempogram	1	NIL	NIL
Tempo	1	NIL	NIL
Pause Ratio	1	NIL	NIL
Tonnetz	NIL	NIL	1

- **MFCC (Mel-Frequency Cepstral Coefficients):** These model how humans perceive sound by representing the power spectrum. It's crucial for both speech and music analysis since it mimics our hearing system.
- **MFCC Delta:** Shows how MFCCs change over time - essentially tracking the velocity of spectral changes.

Think of it as capturing the "movement" in sound.

- **MFCC Delta2:** Measures the acceleration of these changes - how quickly the Delta values themselves are changing. Helps capture more subtle dynamic features.
- **Chroma STFT:** Maps all music frequencies into 12 basic pitch classes (like piano keys in one octave). Useful for analyzing harmony and musical structure.
- **Spectral Features:** These describe the overall distribution of frequencies, helping characterize the sound's "texture" or timbre.

4. Methodologies and Model details

After cleaning and preprocessing the data, we undersampled it to 45,000 files, ensuring an equal distribution of 5,000 files per class. Standard scaling was applied to the feature set, followed by a grid search to evaluate the performance of various SVM kernels, including linear, polynomial and Radial Basis Function (RBF).
For training the models, we split the dataset into training and testing set (80 : 20).
The RBF SVM yielded the highest accuracy on the undersampled data. Subsequently, we applied this model to the full dataset for training, and also trained a Random Forest as a baseline comparison.

SVM

Support Vector Machines (SVM) are supervised learning models used for classification and regression tasks. SVM is ideal for multilanguage classification because it handles high-dimensional data (e.g., MFCCs, chroma) effectively. Its use of kernels, like RBF, enables it to classify nonlinear data. Additionally, SVM is robust to outliers by focusing on support vectors, improving generalization.
The key parameters in SVM are:

- **Kernel**: Defines the type of decision boundary. Common kernels are Linear (for linearly separable data), Polynomial (uses a polynomial function for complex boundaries), RBF (projects data to a higher dimension for nonlinear classification), Sigmoid (similar to neural network activation), Precomputed (uses a custom kernel matrix).
- **C (Regularization)**: Controls the trade-off between margin size and classification accuracy. Smaller C allows more misclassifications for a larger margin. Larger C prioritizes accuracy but may overfit.
- **Gamma**: Determines the influence of a training example. High gamma focuses on close neighbors, creating complex boundaries. Low gamma has a broader influence, making simpler boundaries.

Performance Metrics

For each model that we train, we are generating a classification report using the scikit learn library which gives us the precision, recall, f1-score, accuracy, weighted avg and macro avg. Accuracy is used as the final metric to measure the performance of each model.

5. Result and Analysis

To determine the best feature set we implemented SVM with three different kernels (Linear, RBF, Polynomial) on the three feature sets independently on default hyper-parameters of scikit learn (C = 1, gamma = 'scale', degree = 3).

Feature set	1	2	3
SVM-Linear	89.7%	92.6%	86.2%
SVM-RBF	96%	98%	91.8%
SVM-Poly	94.1%	96.7%	95%
RF	95.1%	97.2%	91%

As can be inferred from the table, feature set 2 is more relevant to the audio dataset and gives greater accuracy.
For hyper-parameter tuning and comparing between the three SVM kernels, the following graph represents the trend of accuracy and score w.r.t different values for parameter 'C' for all three kernels on the undersampled data.

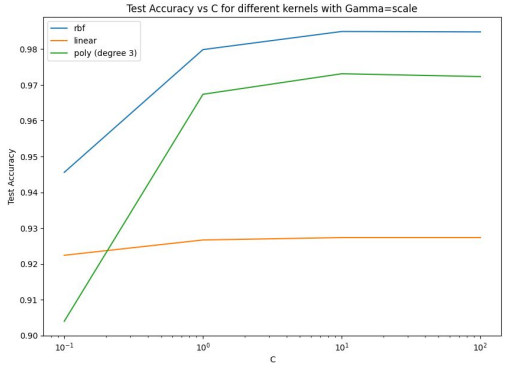


Figure 4. Comparison of different SVM kernels for undersampled Feature Set 2 with degree = 3

We can infer that the best model is SVM with Radial Basis Function kernel with parameter 'c' = 10 obtained by implementing grid search as shown above.
Doing 5-fold cross-validation for SVM-RBF for different values of 'C' shows that same trend is followed for all folds, with best value of 'C' at 10 on undersampled data. We were constrained by the training runtime of SVM models on large dataset, thus carried out this strategy of testing on smaller data first.

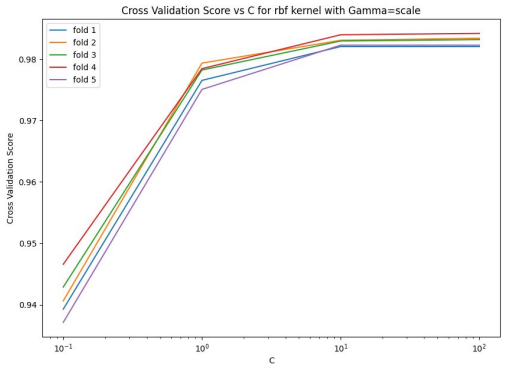


Figure 5. 5-fold CV for SVM-RBF with different values of 'C'

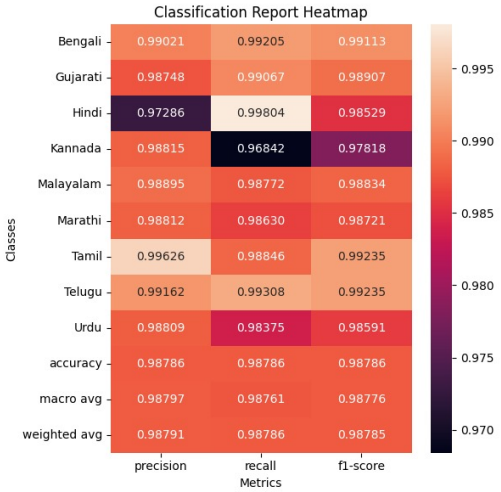


Figure 6. Classification Report for SVM-RBF on complete data

The final accuracy reported for SVM-RBF for Feature Set 2 for the complete dataset is **98.78%**. On the other hand, our Random Forest classifier gives an accuracy of **98.3%** with 'n_estimators=100'

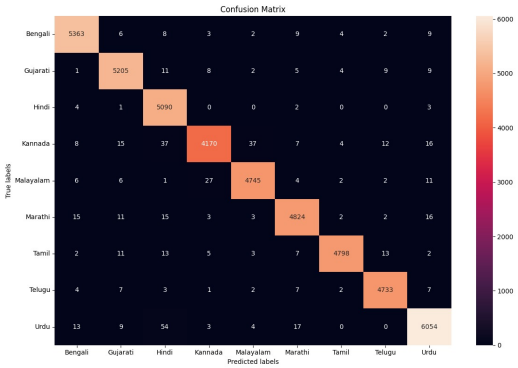
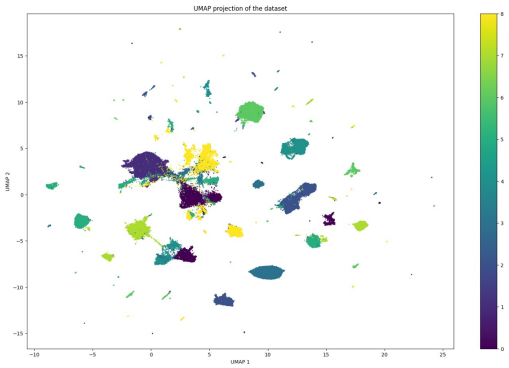


Figure 7. Confusion matrix for SVM-RBF

The attached UMAP projection shows distinct clusters where each colour represents a different language. This shows that our features were selected correctly and are relevant. This is also why SVM gives us a good result when used for training on this feature set due to the non linear nature of the dataset.



6. Conclusion

Learnings

- As suggested by the literature survey and our trained models, the best feature set appears to be the one consisting of high number of MFCCs. This hints that MFCC features are highly relevant to an speech dataset.
- SVM models are good at the task of audio classification since the dataset is non-linear and in clusters. Comparing SVM kernels, we find that RBF-SVM gives us the best performance across the models due to its ability to handle non-linear patterns in high dimensional feature spaces.

Work Left

Moving forward, we will explore models like GMM, HMM and SVM-UBM. We will finish our analysis of these models by finally training our data using Multi-Layer Perceptron Models. If time permits, we would try an end-to-end pipeline and deploy it using GitHub Pages.

Contributions

- Ritika Thakur - Pre-processing, Feature Selection (Feature Set 2), Model Training (RF, Linear SVM, SVM-RBF), Report, PPT
- Saksham Singh - Pre-processing, Feature Selection (Feature Set 1), Model Training (RF, Linear SVM, SVM-RBF, SVM-Poly), Grid Search, Report, PPT
- Sarthak Gupta - Pre-processing, Feature Selection (Feature Set 3), Model Training (Linear SVM), PPT
- Sidhartha Garg - Pre-processing, Feature Selection (Feature Set 1), Model Training (Linear SVM, SVM-RBF, SVM-Poly), Grid Search, Report, PPT

7. References

- S. Suksri and M. Yingthawornsuk, "Speech Recognition using MFCC," *Semantic Scholar*. Available: https://www.semanticscholar.org/paper/Speech-Recognition-using-MFCC-Suksri-Yingthawornsuk.
- P. Fokoue and W. Ma, "Speaker Gender Recognition via MFCCs and SVMs," *Semantic Scholar*. Available: https://www.semanticscholar.org/paper/Speaker-Gender-Recognition-via-MFCCs-and-SVMs-Fokoue-Ma.
- D. Jiang and H. Bai, "SVM-based audio scene classification," *Semantic Scholar*. Available: https://www.semanticscholar.org/paper/SVM-based-audio-scene-classification-Jiang-Bai.
- A. R. Mishra and N. Sharma, "Spoken Language Identification using Gaussian Mixture Model-Universal Background Model in Indian Context," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 4002–4008, 2018. Available: https://www.ripublication.com/ijaer18/ijaerv13n5$_7$9.*pdf*.
- C. Bharadwaj, "Audio Dataset with 10 Indian Languages," *Kaggle*. Available: https://www.kaggle.com/datasets/hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages.
- *Scikit learn Documentation*, "SVM," Available: https://scikit-learn.org/1.5/modules/svm.html.
- *Librosa Documentation*, "Librosa Library Python," Available: https://librosa.org/doc/latest/index.html.
- *Scikit-learn*, "Scikit-learn for model training, metric calculations, and feature scaling," Available: https://scikit-learn.org/stable/.

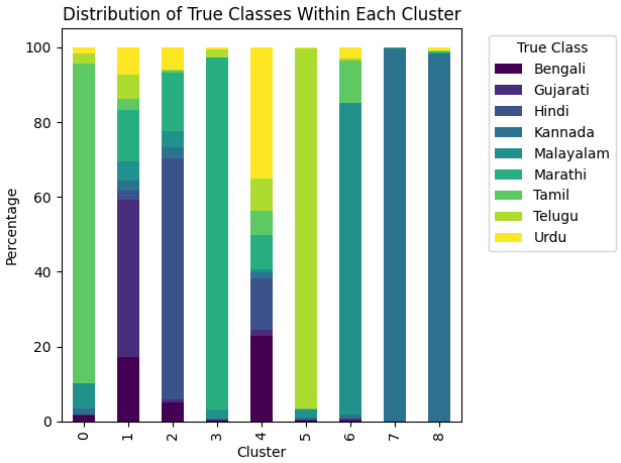


Figure 8. Pain.jpeg