Language Recognition from Audio for Pan Indian Languages

Ritika Thakur |zSaksham Singh | Sarthak Gupta | Sidhartha Garg

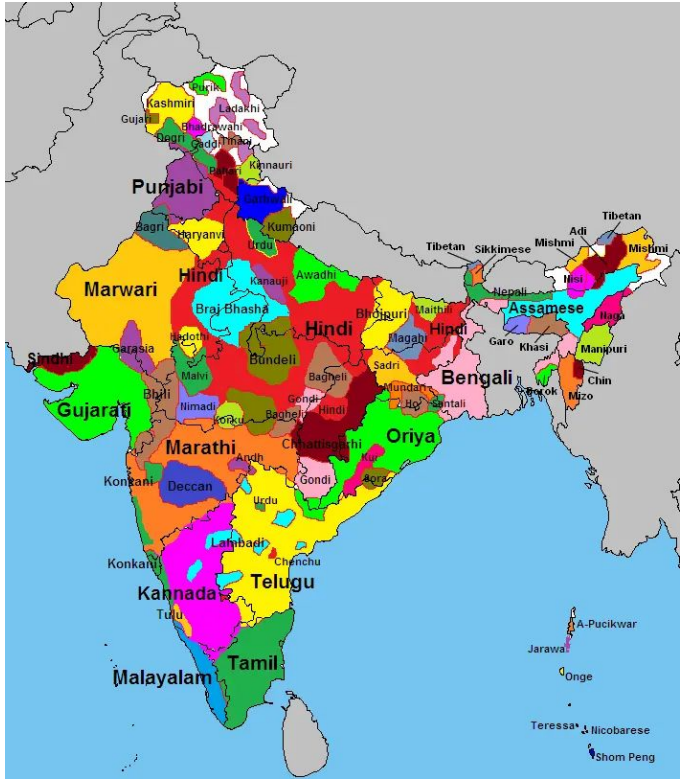
Group 30

ML Endsem Project

Github Repository

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation



Linguistic Diversity: India has 23 official languages and numerous dialects, creating a need for efficient language recognition systems.

Challenges in Multilingual Environments: Identifying spoken languages in India is complex due to overlapping phonetic patterns across languages.

Enhancing Communication Technologies: A reliable language recognition system can improve multilingual communication, making digital platforms more accessible for diverse users.

Literature review

Corpus ID: 5979747

Speech Recognition using MFCC

S. Suksri, T. Yingthawornsuk • Published 2012 • Computer Science

TLDR This paper describes an approach of speech recognition by using the Mel-Scale Frequency Cepstral Coefficients (MFCC) extracted from speech signal of spoken words, and shows the improvement in recognition rates significantly when training the SVM with more MFCC samples by randomly selected from database. Expand

[PDF] psrcentre.org | Save to Library | Create Alert | Cite

Literature review

DOI: 10.1109/NLPKE.2005.1598721 • Corpus ID: 17082759

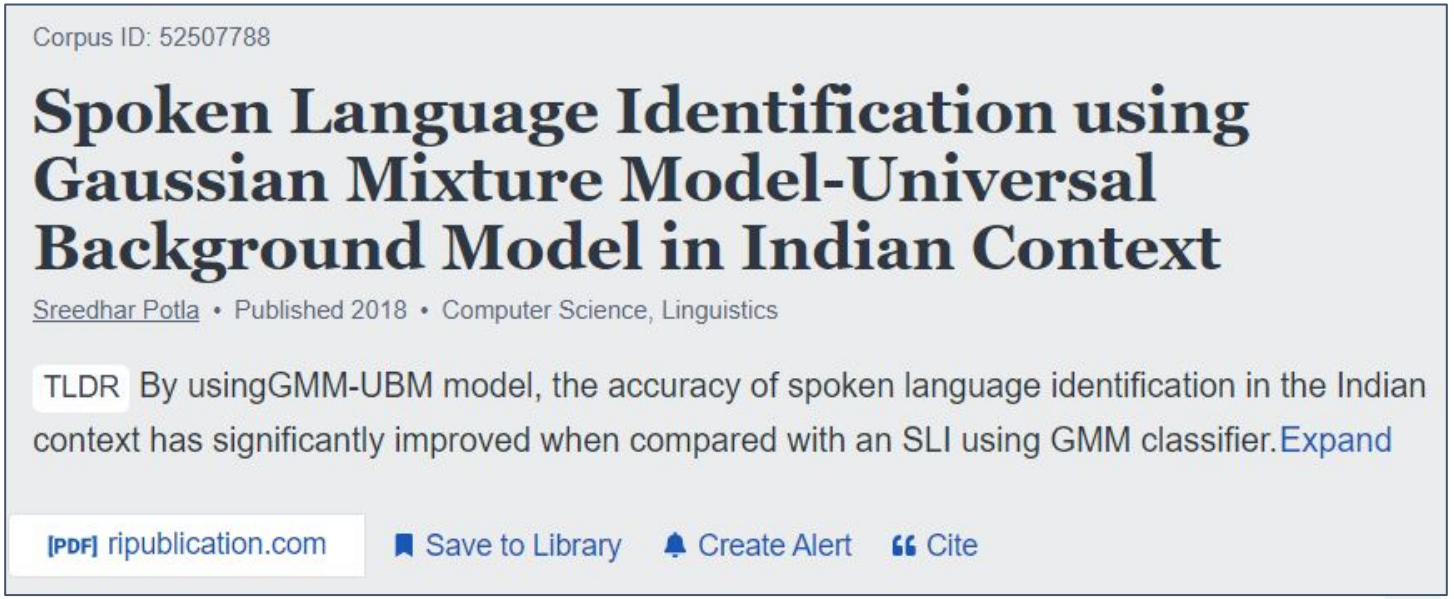
SVM-based audio scene classification

Hongchen Jiang, Junmei Bai, +1 author Bo Xu • Published in International Conference on... 30 October 2005 • Computer Science

TLDR 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. Expand

View on IEEE | doi.org | Save to Library | Create Alert | Cite

Literature review



Corpus ID: 52507788

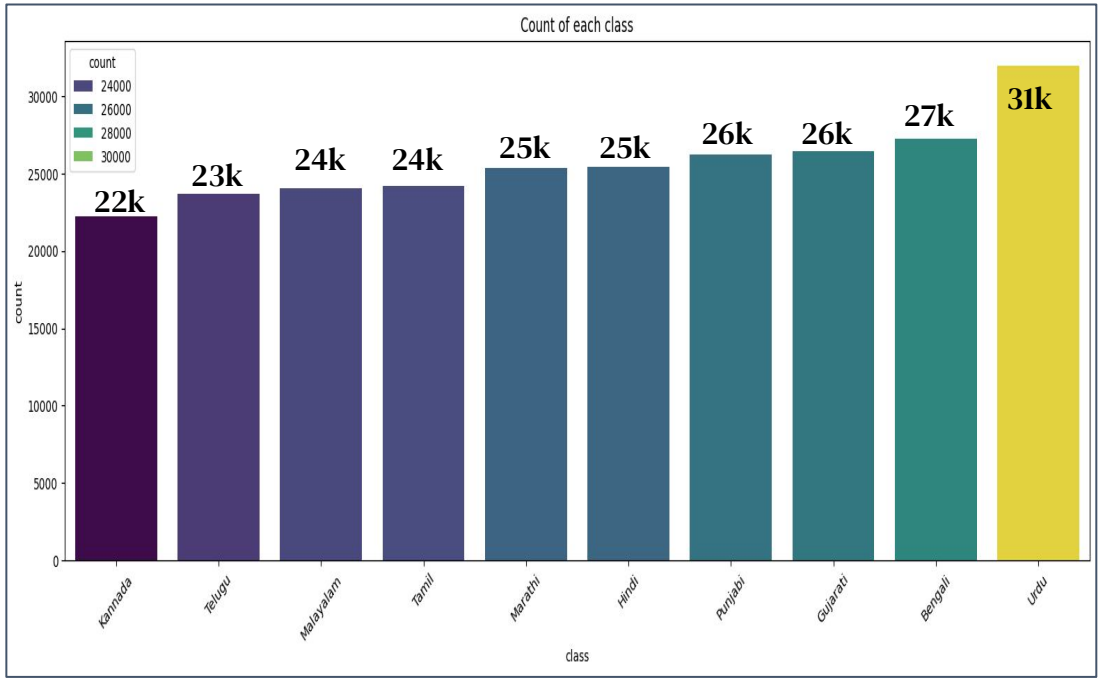
Spoken Language Identification using Gaussian Mixture Model-Universal Background Model in Indian Context

Sreedhar Potla • Published 2018 • Computer Science, Linguistics

TLDR By usingGMM-UBM model, the accuracy of spoken language identification in the Indian context has significantly improved when compared with an SLI using GMM classifier. Expand

[PDF] ripublication.com Save to Library Create Alert Cite

Dataset description



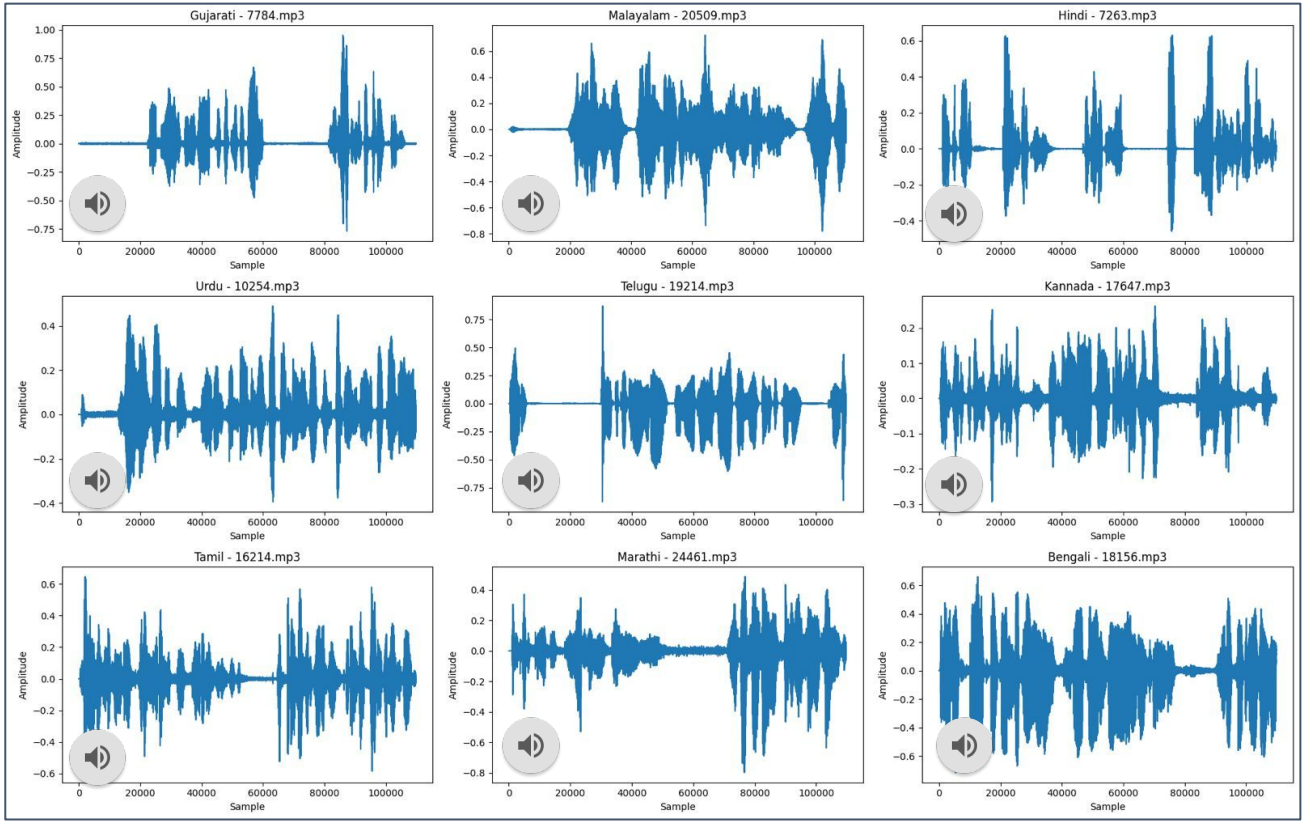
Sampling rate: 22050 bits/sec, 44.1 kHz

Audio Length: mostly about 5 secs (outliers ~ 0 secs)

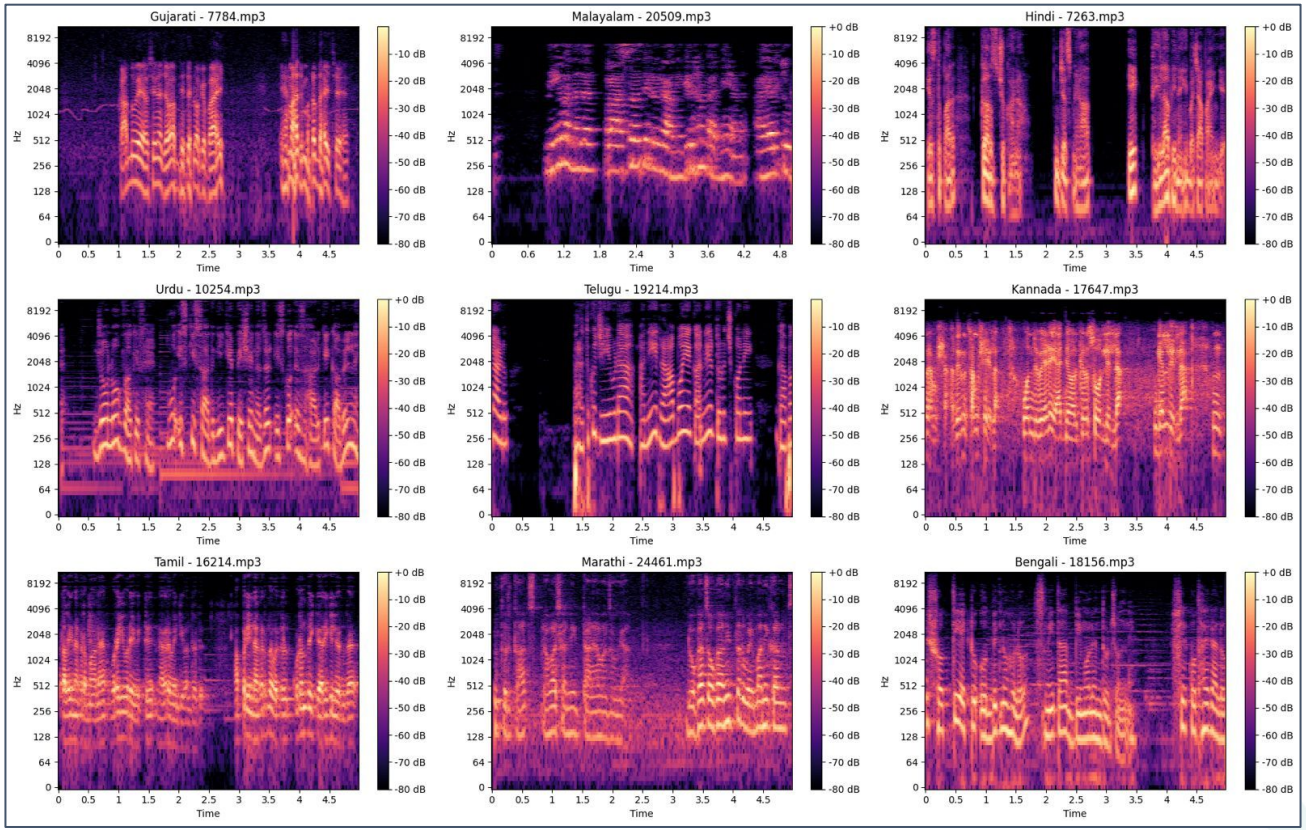
Class **Punjabi** CORRUPTED HENCE DISCARDED.

- Flute music
- Silent files
- And majorly - Gujarati audio

Dataset description



Dataset description

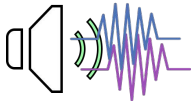


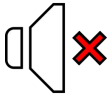
Methodology

- Pre-Processing
  - data cleaning
  - feature extraction
- Model selection
- Parameter hyper tuning
- Performance metrics

Methodology/Preprocessing/Data-Cleaning

Discarding of files if

- length is 0 seconds
- Signal-to-Noise Ratio above 0.02 (noisy) 

Removal of silent segments (segments with amp < 20dB) (Completely or partially silent) 

Padding of files if length < 5 seconds

Trimming of files if length > 5 seconds

Audio normalization for all files

Methodology/Preprocessing/Feature-extraction

3

FEATURE SETS

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

Methodology/Model-selection

Midsem

- Logistic Regression ✗
- Perceptron ✗
- Naive Bayes ✗
- Random Forest ✓
- Support Vector Machine ✓

Endsem

- Gaussian Mixture Models ✓
- Hidden Markov Model ✓
- Multilayer Perceptron ✓

Methodology/ SVM Parameters

Regularization parameter c: More c, lesser margin, chance of better accuracy but may overfit. Using **grid search** and **k-fold cross validation**, we found best **c = 10** (This grid search was conducted for different kernels of SVM: linear, rbf, poly and different values of regularization parameter **c: 0.1, 1, 10, 100**).

Gamma: How much more the boundary is influenced by training example, i.e. how complex the boundary is.

Kernels:

- **Linear:** for linearly separable data
- **Rbf:** projects data to a higher dimension for nonlinear classification
- **Poly:** uses a polynomial function for complex boundaries

Methodology/ GMM Parameters

Number of components (nnn): Refers to the number of Gaussian distributions (or clusters) used in the Gaussian Mixture Model. **nnn = 9** (since we are working with 9 languages).

Covariance type: Specifies the type of covariance matrix used in the GMM. It can be ‘full’, ‘tied’, ‘diagonal’ or ‘spherical’. **covariance type=tied** (assumes a shared covariance matrix across clusters to reduce complexity).

Regularization covariance (reg_covarreg_covarreg_covar): A small positive value added to the covariance matrix to ensure numerical stability during computation. **reg covar=$1e^{-3}$**.

Methodology/ GMM-UBM Parameters

Extends GMM for speaker and language recognition tasks by using a Universal Background Model (UBM) to capture general patterns.

Number of components (nnn): **nnn=512** (chosen to handle complex distributions).

Covariance type: **covariance type=diag**(improves computational efficiency).

Regularization covariance: **reg covar=$1e^{-6}$** (detailed modeling).

Relevance factor: **16** (balances adaptation from the UBM via MAP techniques).

Methodology/ HMM Parameters

Probabilistic models representing systems with hidden states that evolve over time, where each state generates observable outputs.

Number of Components (n_components): number of hidden states in the model.

Covariance Type: Specifies the structure of the covariance matrix of the Gaussian distributions:

- **full**: Full covariance matrix (most flexible).
- **diag**: Diagonal covariance matrix (less computationally expensive).
- **spherical**: Shared variance across features (simplest).

Methodology/ HMM Comparison

Probabilistic models representing systems with hidden states that evolve over time, where each state generates observable outputs.

(n_components): best accuracy with n_comp = 1. The more we increase hidden states, the more accuracy decreases.

Covariance Type: full gives the best accuracy as clearly the features are correlated. And we see a big jump from diagonal.

25 MFCC: **93.28%**
67 features: **94.97%**

n comp	full	diag	spherical
1	0.93	0.73	0.68
2	0.56	0.51	0.48
3	0.45	0.42	0.42
4	0.44	0.42	0.41

Methodology/ MLP Parameters

Architecture 1:

- **Structure**: Three hidden layers with **50, 30,** and **20 neurons**, respectively.
- **Activation Functions**: Tested multiple activation functions, including **ReLU, identity, tanh**, and **sigmoid**.
- **Optimizer**: Used the **Adam optimizer** for training.

Architecture 2:

- **Structure**: Three hidden layers with **134, 268,** and **134 neurons**, respectively.
- **Dropout probability: 0.3** applied after each hidden layer to prevent overfitting.
- **Activation Functions**: Applied **ReLU activation** to all hidden layers.
- **Optimizer**: Used the **Adam optimizer** for training.

Methodology/Performance-metrics

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. **Mean Accuracy is used as the final metric to measure the performance of each model**.

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$

$$\text{F}_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

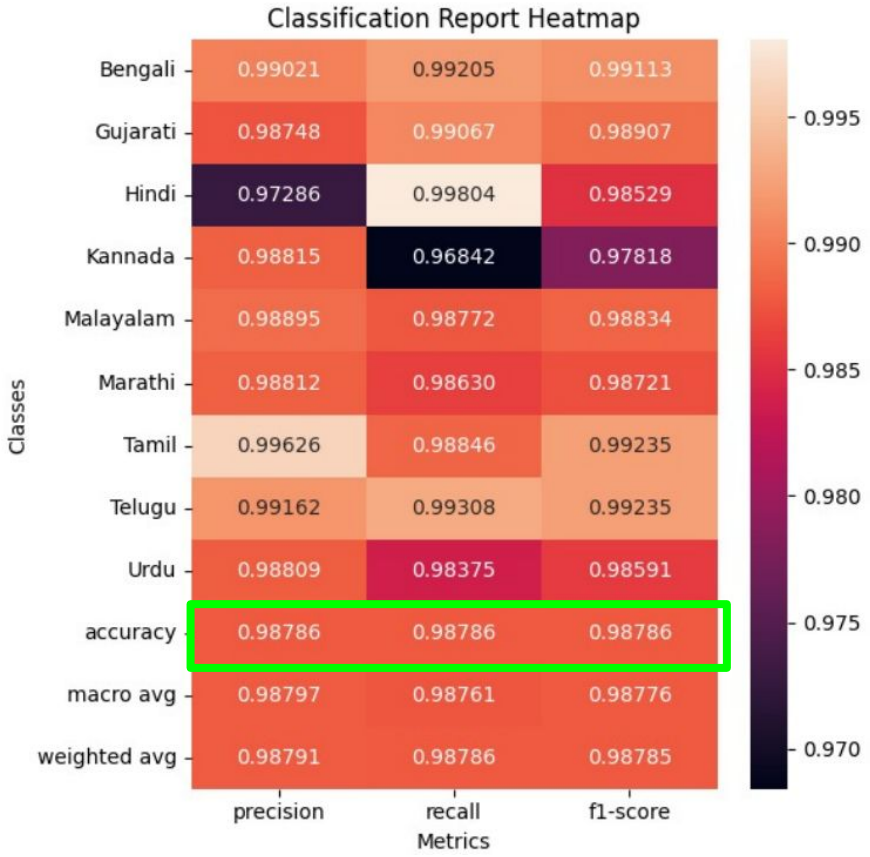
Results/Analysis/Conclusion (SVM)

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%

C = 1
gamma = 'scale'
degree = 3

- Feature Set 1 had more unique range of features as compared to Feature Set 2 which may have lead to extraction of irrelevant features, decreasing the performance.
- Feature Set 2 excelled as it had more MFCC and MFCC delta coefficients as compared to the rest, further backing the paper in our literature survey.

Results/Analysis/Conclusion (SVM)



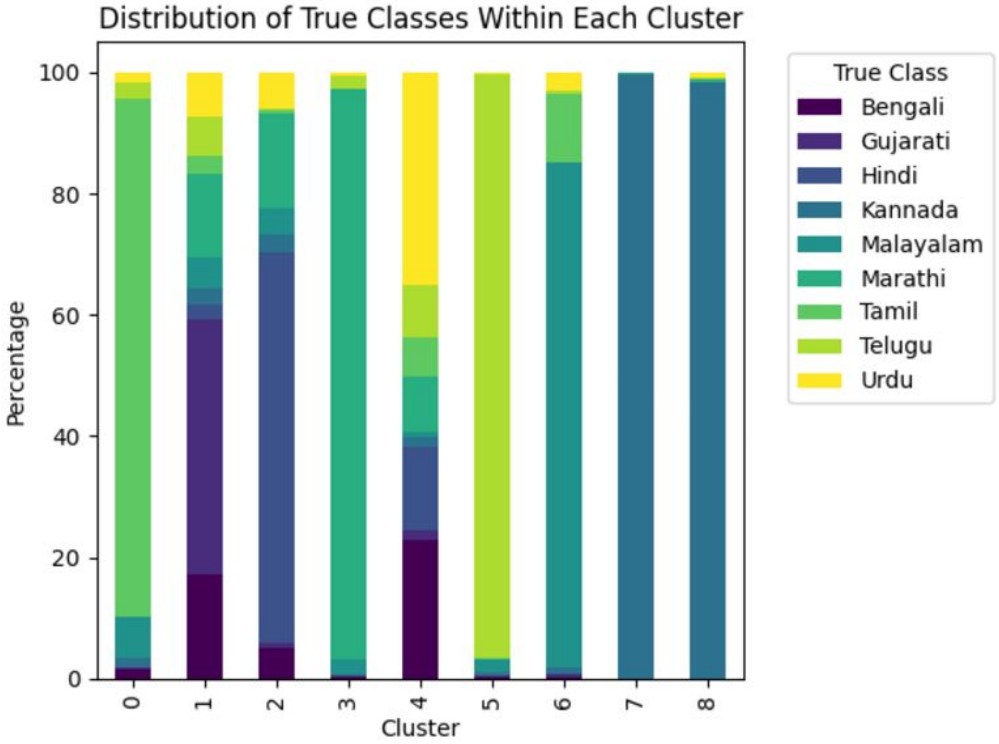
The final accuracy reported for SVM-RBF for Feature Set 2 for the complete dataset is **99.03%.**

On the other hand, our Random Forest classifier gives an accuracy of **98.3%** with 'n estimators=100'

Random forests are still good as it is both an ensemble learning methods, and can capture non linear data.

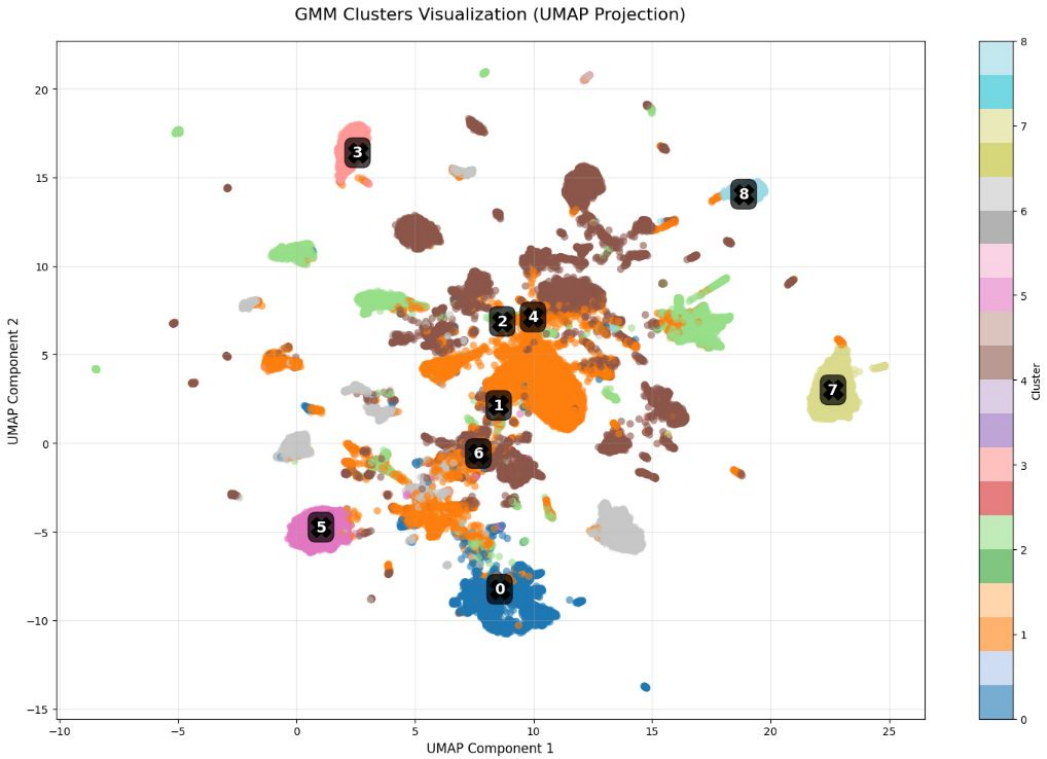
The results of SVM-RBF show that even after robustness of RF, SVM-RBF is able to outperform it for the problem.

Results/Analysis/Conclusion (GMM)



Clusters 1, 6, and 7 have high purity, dominated by a single language (Hindi, Malayalam, Telugu). Clusters 3 and 4 show more overlap, indicating lower purity. This suggests GMM works well for languages with distinct features but struggles with languages that have similar characteristics.

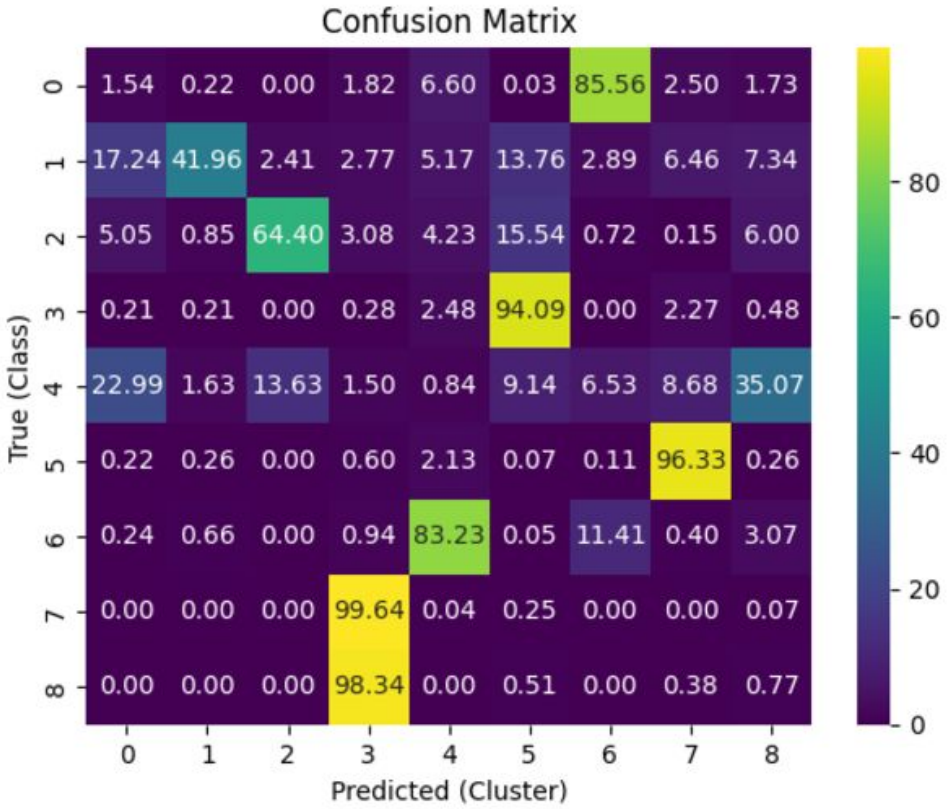
Results/Analysis/Conclusion (GMM)



Clusters 0, 1, and 7 are well-separated, matching the high purity seen in previous figure.

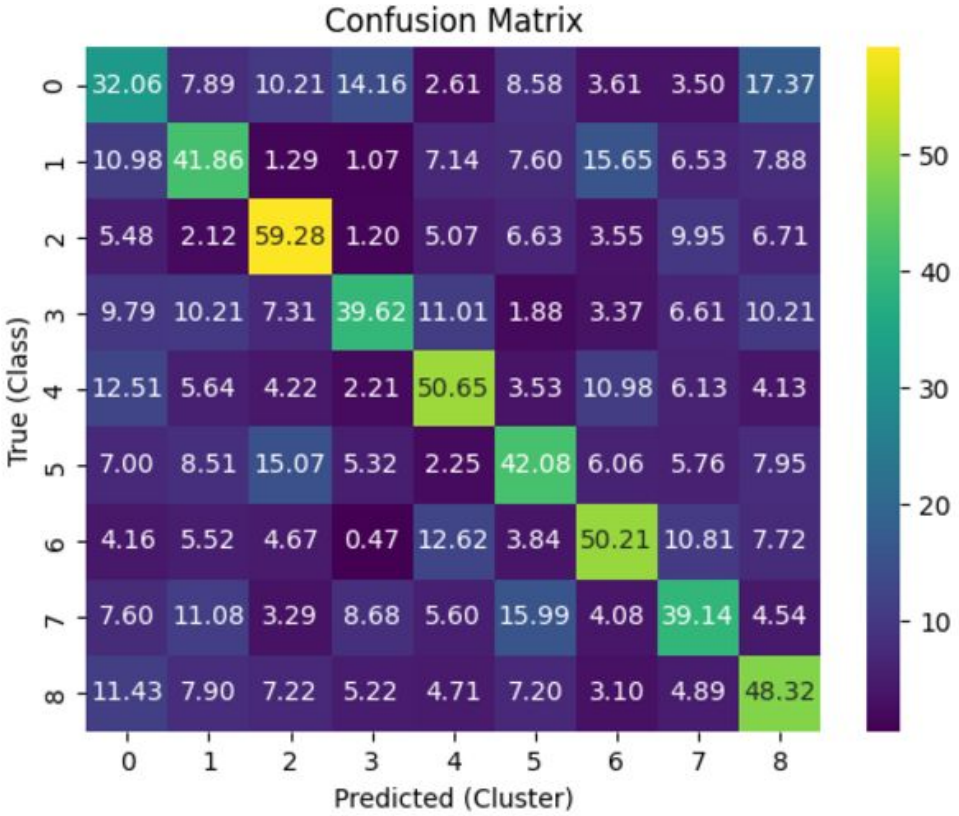
Clusters 2, 3, and 4 overlap, indicating poor separation, particularly for languages like Gujarati and Kannada, suggesting the need for better feature differentiation.

Results/Analysis/Conclusion (GMM)



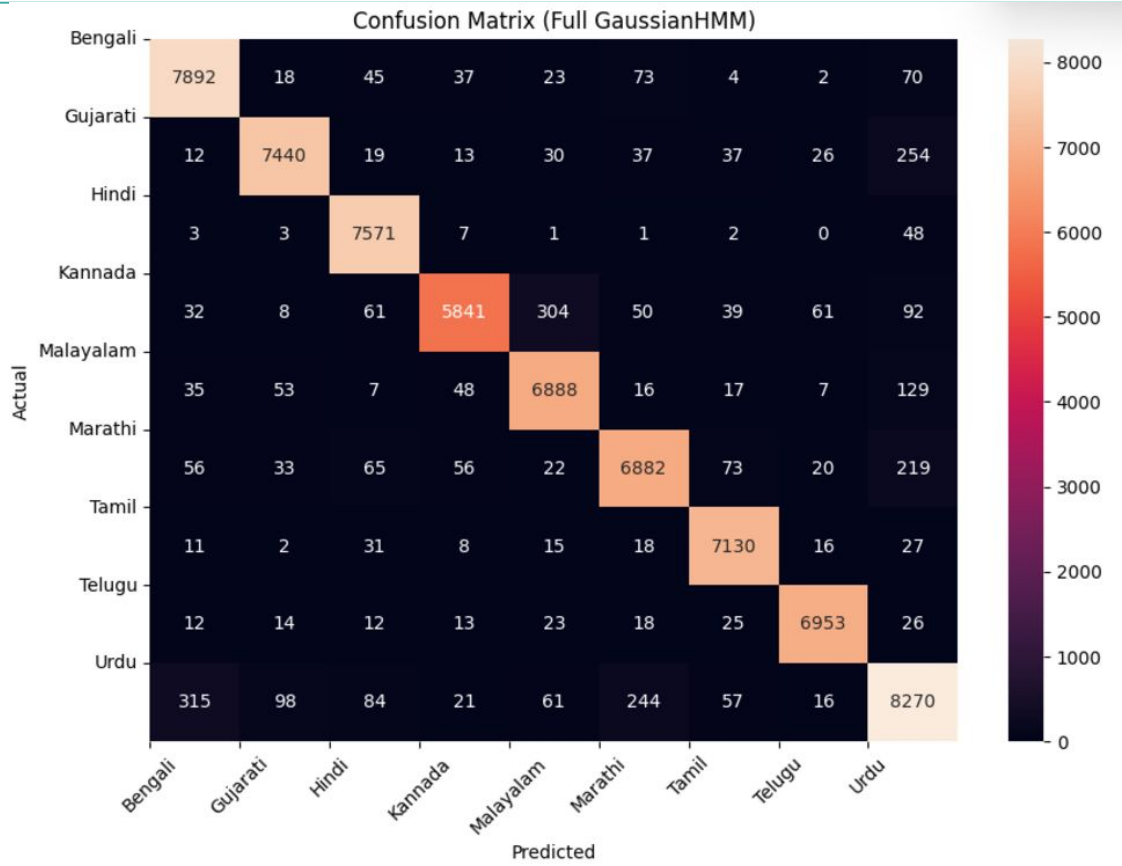
The accuracy of the GMM model is **59%**

Results/Analysis/Conclusion (GMM-UBM)



The accuracy of the GMM-UBM model is **44%**

Results/Analysis/Conclusion (HMM)

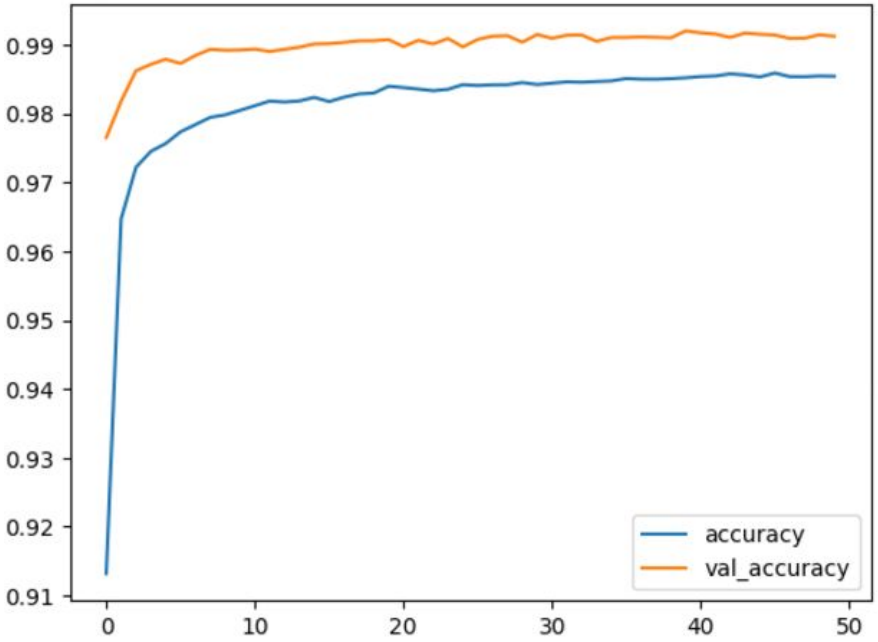


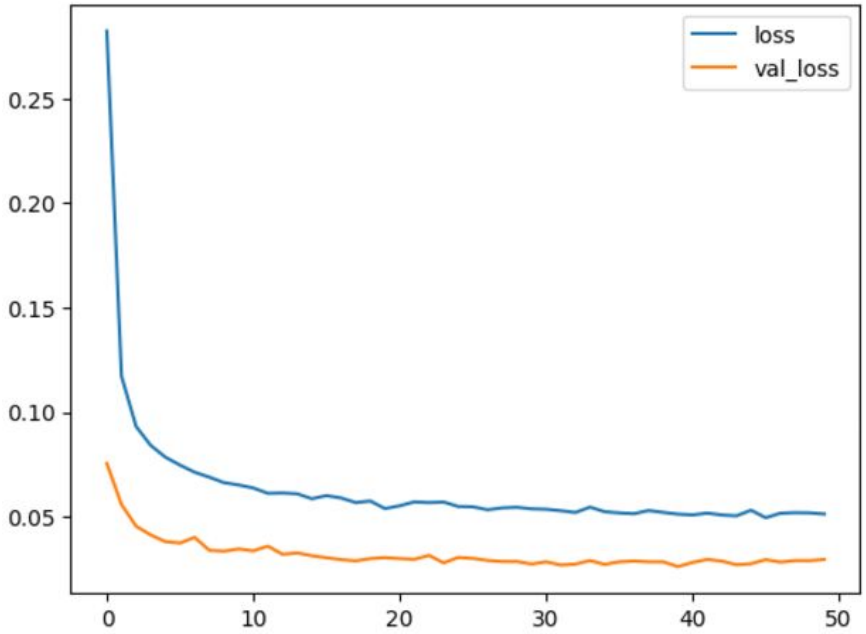
Results/Analysis/Conclusion (HMM)

n comp	full	diag	spherical
1	0.93	0.73	0.68
2	0.56	0.51	0.48
3	0.45	0.42	0.42
4	0.44	0.42	0.41

The above table is for only 25 MFCC features. When taking the best parameters from above but using all 67 features we get the accuracy of **94.97%**.

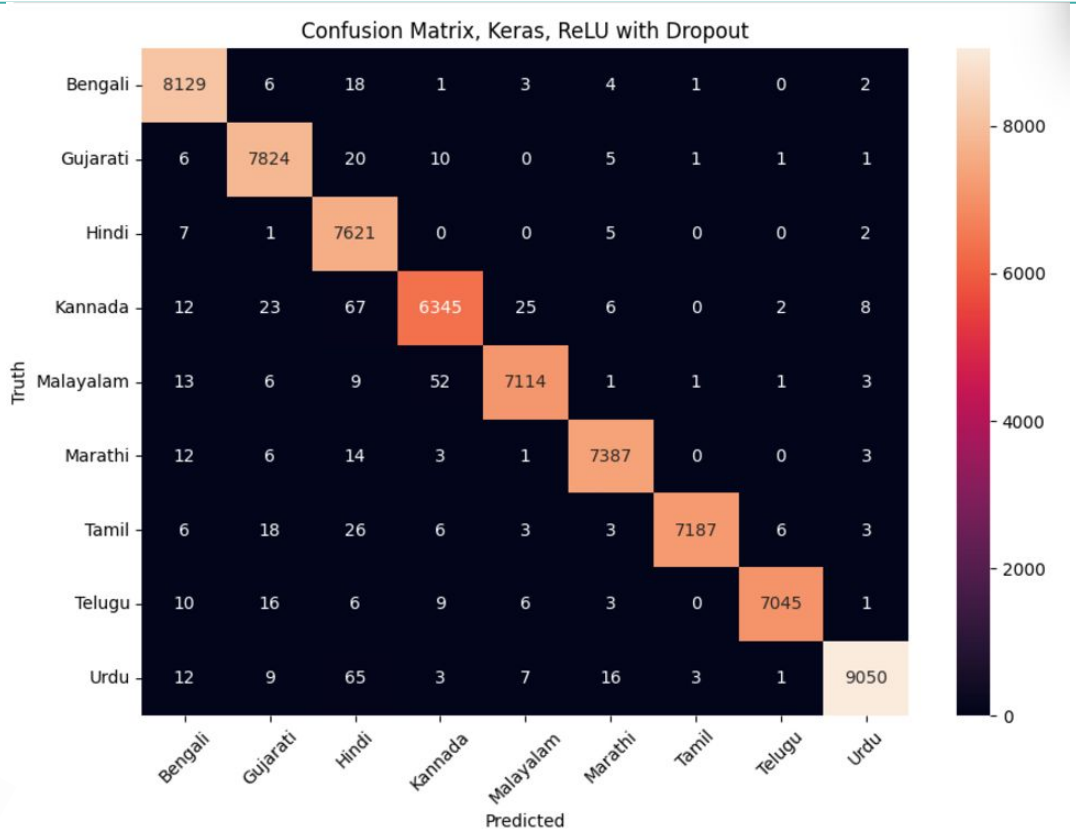
Results/Analysis/Conclusion (MLP)

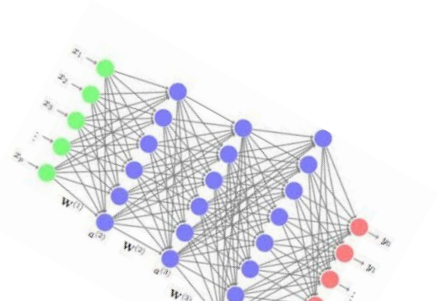




Final Accuracy Achieved: 99.12%

Results/Analysis/Conclusion (MLP)





Results/Analysis/Conclusion (MLP)

Architecture	ReLU	Sigmoid	Identity	Tanh
50,30,20	98.67%	98.30%	88.61%	98.40%
134,268,134	99.12%	–	–	–

The Multilayer Perceptron (MLP) with three hidden layers (134, 268, 134) and a dropout rate of 0.3 outperforms all other models, achieving an accuracy of **99.12%**. This suggests that MLP is highly effective for this classification task, likely due to its ability to learn complex non-linear relationships in the data.

Results/Analysis/Conclusion

- MLP was highly effective due to its ability to learn complex non-linear relationships in the data, enhanced by architecture structure and dropout.
- GMM and HMM struggle to distinguish between languages with similar acoustic features and thus require better feature differentiation and model tuning for improved performance.
- Inclusion of MFCC, spectral features, and derived features (e.g., Chroma STFT, ZCR) significantly boosted model performance.
- SVM is able to effectively handle high-dimensional data, acting as a strong benchmark.

Timeline (Midsem)

- Problem Statement and dataset selection	**27th August 2024**
- Ideation	**15th September 2024**

- Pre-Processing
  - data cleaning
  - feature extraction
	18th September 2024

- Model selection	**28th September 2024**
- Parameter hyper tuning	**12th October 2024**
- Performance metrics	**18th October 2024**

Timeline (Endsem)

- Explore Gaussian Mixture Models, Hidden Markov Models and SVM with Universal Background Models — **1st November 2024**
- Explore Multi-Layer Perceptron Models — **20th November 2024**
- Deploy end-to-end pipeline — **If Time Permits**

Individual team members' contributions

Ritika Thakur - Pre-processing, Feature Selection (Feature Set 2), Model Training (RF, Linear SVM, SVM-RBF), Report, GMM-UBM, PPT

Saksham Singh - re-processing, Feature Selection (Feature Set 1), Model Training (RF, Linear SVM, SVM-RBF, SVM-Poly), Grid Search, MLP, Report,
PPT

Sarthak Gupta - re-processing, Feature Selection (Feature Set 3), Model Training (RF, Linear SVM, SVM-RBF, SVM-Poly), HMM, PPT

Sidhartha Garg - re-processing, Feature Selection (Feature Set 1), Model Training (Linear SVM, SVM-RBF, SVM-Poly), GMM, Grid Search, Report, PPT

Thank You