Linguistix: Speaker Recognition System - Mid-Progress Report

Shashank Parchure Atharva Honparkhe Vyankatesh Deshpande
Abhinash Roy Namya Dhingra Damarasingu Akshaya Sree

CSL2050: Pattern Recognition and Machine Learning

Abstract

This mid-progress report outlines our work on Linguistix: Speaker Recognition System, which classifies speakers using various machine learning models. We implemented KNN, SVM, Bayesian Learning, Decision Tree, and K-Means clustering, enhanced with LDA, PCA, and Correlation based Feature Selection. Initial results indicate that KNN, SVM, and Bayesian Learning perform well with LDA and PCA, while Decision Tree results were suboptimal. Future work involves hyperparameter tuning and advanced feature extraction to improve model performance.

1 Introduction

The goal of the project is to identify who is speaking from the audio data. We are given a Kaggle dataset with different audio files of 50 different speakers. For each speaker a number of audio files are given each of length around 1 minute. The goal is to try out different traditional ML methods on the dataset and experiment with them and compare the results systematically using different performance measures and qualitative analysis.

2 Approaches Tried

We implemented:

- **KNN, SVM, Naïve Bayes, Decision Tree, K-Means Clustering** - Evaluated for speaker classification.
- **Pre-Processing Techniques:**
  - Feature Extraction: Extracted features using PCA,LDA,t-SNE,UMAP
  - Feature Selection: Used Top n features using Corelation based Feature Selection

3 Experiments and Results

3.1 Dataset

We used the Kaggle dataset containing speech samples and extracted MFCC features.
Dataset Link

3.2 Results

Preliminary results indicate:

- KNN, SVM, and Naïve Bayes yielded high accuracy with LDA and PCA.
- Decision Tree performed suboptimally due to overfitting.
- Feature selection improved efficiency with minimal accuracy loss.

4 Performance Summary of Various Models

Model	Train Acc (%)	Validation Acc (%)	Test Acc (%)
KNN (With LDA)	99.89	100.00	99.80
KNN (With PCA)	89.58	76.98	79.08
KNN (Entire Dataset)	98.27	95.22	94.63
SVM (With FS)	98.95	99.24	100.00
SVM (With PCA)	98.12	99.21	99.40
SVM (With LDA)	91.43	97.78	95.56
Bayesian Learning (Entire Dataset)	95.96	75.00	74.70
Bayesian Learning (With FS)	96.87	79.76	82.27
Bayesian Learning (With LDA)	99.89	100.00	99.80
Bayesian Learning (With PCA)	52.49	50.25	48.11
Decision Tree (Entire Dataset)	100.00	63.35	61.43
Decision Tree (With PCA)	100.00	14.34	14.12
Decision Tree (With LDA)	100.00	14.34	14.12
Decision Tree (With PCA + LDA)	100.00	63.35	61.43
Decision Tree (With LDA + t-SNE)	99.60	1.86	1.59
Decision Tree (With LDA + UMAP)	99.94	84.88	88.59
Decision Tree (With t-SNE)	15.77	1.86	2.65
Decision Tree (With UMAP)	16.51	11.41	12.20
Decision Tree (Raw Features)	16.79	8.22	10.61
AdaBoost	20.78	18.92	19.28
SAMME	7.37	7.17	7.16
Bagging	100.00	81.08	82.70
K-Means (With LDA)	88.61	92.61	87.67
K-Means (With LDA + PCA)	91.03	94.03	87.67
K-Means (Raw Features)	8.68	9.09	37.27

Table 1: Performance comparison of various models

5 Future Work

We aim to:

- Optimize models through hyperparameter tuning.
- Enhance feature extraction with deep learning techniques and implement further ML Techniques such as GMM, Perceptron and ANN.
- Experiment with ensemble learning methods.
- Create a Web Demo of the best approach.

6 Contribution of Each Member

- Shashank Parchure: Implemented KNN with PCA, Bayesian Learning with Correlation based Feature Selection and SVM with PCA.
- Atharva Honparkhe: Implemented Decision Tree with PCA, LDA and Ensemble Methods
- Vyankatesh Deshpande: Extracted MFCC Features of the Dataset. Implemented KNN with LDA, Bayesian Learning with LDA and SVM with Correlation based Feature Selection.
- Abhinash Roy: Implemented KMeans Clusturing with LDA
- Namya Dhingra: Implemented Decision Tree with UMAP and t-SNE, KMeans Clustering with PCA
- Damarasingu Akshaya Sree: Implemented KNN, Bayesian Learning with PCA and SVM with LDA