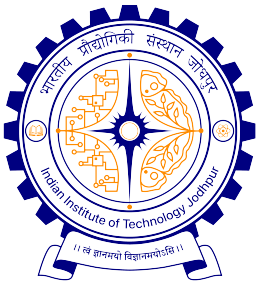
Linguistix: Speaker Recognition System

Shashank Parchure[1] **Atharva Honparkhe**[1] **Vyankatesh Deshpande**[1]
Abhinash Roy[1] **Namya Dhingra**[1] **Damarasingu Akshaya Sree**[1]

CSL2050: Pattern Recognition and Machine Learning



Project Report

Instructor: Prof. Anand Mishra

Department of CSE
Indian Institute of Technology, Jodhpur

Abstract

Linguistix: Speaker Recognition System investigates classical machine learning approaches for accurate speaker identification from audio recordings. The project implements a broad suite of supervised classifiers—including K-Nearest Neighbors, Support Vector Machines, Decision Trees, MLP, ANN and Naïve Bayes—along with unsupervised techniques such as K-Means clustering and Gaussian Mixture Models (GMM). Feature engineering is enhanced through dimensionality reduction methods like PCA and LDA, which improve representation and mitigate overfitting. In addition, ensemble strategies are employed to boost performance and robustness. For example, ensemble GMM methods combine multiple component estimates for more stable likelihood predictions, while decision tree ensembles help counteract the overfitting inherent in single-tree configurations. Experimental results demonstrate that when classical classifiers are paired with PCA or LDA preprocessing models such as KNN, SVM, ANN and Bayesian classifiers consistently achieve high accuracy and robust generalization.
Keywords: Speaker Recognition, Machine Learning, LDA, KNN, Decision Tree, Naive Bayes, ANN, SVM, Feature Selection, Dimensionality Reduction.

Contents

1 Introduction

The project aims to develop a speaker recognition system using multiple machine learning techniques. The system is designed to accurately identify speakers by extracting and analyzing distinct voice features. This report details the implementation of various ML techniques and it systematically compares their performance using various evaluation metrics while also conducting a failure case analysis.

1.1 Problem Statement

The project involves classifying an individual's identity solely from their voice data. The system must:

- Correctly extract distinguishing features from audio signals.
- Handle variability in voice recordings caused by different recording conditions, languages, accents, and external noise.
- Compare and contrast multiple machine learning models to determine which approaches yield the best recognition accuracy and robustness under diverse conditions.

1.2 Challenges Faced

Implementing an effective speaker recognition system presents several challenges:

- **Variation in Voice:** Individual voices can change due to emotional state, health, and environmental noise, impacting consistency.
- **Feature Extraction:** Identifying and extracting robust features (e.g., Mel Frequency Cepstral Coefficients (MFCCs), pitch, and spectral features) is technically complex and critical for performance.
- **Inter-Speaker Similarities:** Differences between speakers may be subtle, which demands sensitive algorithms to detect minute nuances.
- **Noisy and Inconsistent Data:** Real-world audio data often contains background noises and recording artifacts, making reliable classification more difficult.
- **Model Selection and Tuning:** Systematically comparing and tuning different machine learning models to achieve optimal performance adds another layer of complexity.
- **Use of Google Cloud:** For implementing Google Cloud, after entering the Coupon Code provided by the Instructor, the portal was requesting for Payment Details(Credit/Debit Card) to initiate the Free Trial and therefore it was not possible to further use a Google Cloud Feature in the project.

1.3 Real-world applications

The technologies and techniques developed for speaker recognition have broad applicability:

- **Biometric Security Systems:** Implementing voice-based user authentication for secure access to devices and facilities.
- **Voice-activated Assistants:** Enhancing the personalization and responsiveness of digital assistants.
- **Telecommunications:** Automating customer identity verification in call centers.
- **Forensic Investigations:** Assisting in the identification of speakers in security and criminal investigations.
- **Healthcare:** Supporting patient recognition in scenarios where non-intrusive identification is beneficial.

2 Dataset

2.1 Dataset Source and Structure

The dataset used in this project was provided by the Course Instructor Prof. Anand Mishra and is publicly available on Kaggle[1]. The dataset is organized into 50 folders, each corresponding to a unique speaker (labeled from Speaker 0000 to Speaker 0050). Each folder contains multiple .wav files, with individual recordings lasting about one minute. Overall, there is approximately 60 minutes of voice data per speaker, offering a rich and extensive collection of audio samples for robust training and evaluation of the speaker recognition system.

2.2 Preprocessing Steps

- MFCC Extraction
- Normalization
- Train-Test-Validation Split

2.3 Data Statistics

- **Number of Speakers:** The dataset comprises 50 unique speakers, each represented by individual folders.
- **Audio File Count:** In total, there are 2511 audio samples (.wav files) across all speakers.
- **Feature Extraction:** MFCC extraction yields a feature matrix of size (2511, 4000), meaning each audio sample is represented by 4000 features.

To better understand the dataset, we visualized the distribution of samples across speakers and the LDA-transformed feature space. These visualizations help assess the class balance and how well different speakers are separated after applying dimensionality reduction.

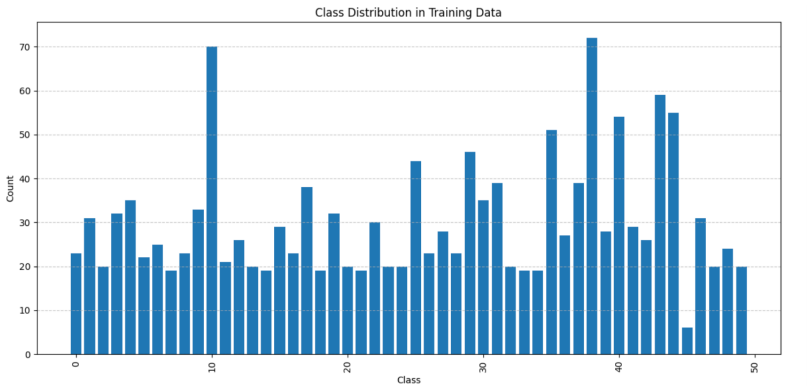


Figure 1: Speaker-wise class distribution in the dataset

[1] https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset

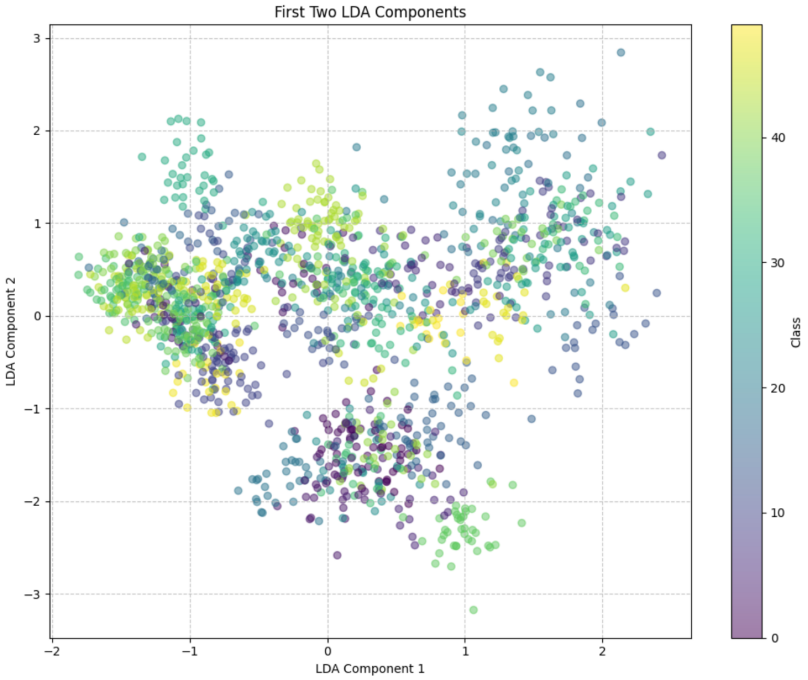


Figure 2: Visualization of LDA components across speakers

3 Approaches Tried

3.1 Feature Extraction

Feature extraction is a crucial step in speaker recognition systems, as it transforms raw speech signals into compact representations that capture unique speaker characteristics. This process is essential for distinguishing between different speakers and improving the overall performance of the system.

- **Mel-Frequency Cepstral Coefficients (MFCCs):** MFCCs are commonly used features in speech and speaker recognition. They capture the shape of the speech spectrum in a way that reflects how humans perceive sound. MFCCs were used to extract essential frequency characteristics from the audio files. These features formed the main input for all machine learning models.
- **Principal Component Analysis (PCA):** PCA reduces the dimensionality of the feature space by retaining the most important variations in the data. It helps remove noise and compress data without significant information loss. PCA helped simplify the MFCC features, reduced overfitting, and improved the performance and speed of models like KNN, SVM, and Naïve Bayes.
- **Linear Discriminant Analysis (LDA):** LDA is a supervised method that reduces dimensions by maximizing the separation between different classes (in this case, speakers). LDA made speaker differences more distinct, which improved the accuracy of classifiers such as SVM, Decision Trees, and MLP.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE):** t-SNE is a visualization technique that maps high-dimensional data into two or three dimensions while preserving local patterns. In this project, it was used to visualize how well the extracted features (like MFCCs) clustered by speaker. This helped qualitatively assess the effectiveness of feature extraction.
- **Uniform Manifold Approximation and Projection (UMAP):** UMAP is a dimensionality reduction tool that maintains both local and global structure in the data. It often gives clearer separation than t-SNE. In this project, UMAP was used alongside t-SNE to visualize speaker embeddings. It revealed how well the features separated different speakers and supported the evaluation of clustering methods like GMM.

3.2 Models Implemented

- **KNN (K-Nearest Neighbours):** Implemented using Euclidean Distance on MFCC feature vectors. Classification is performed by analyzing the majority label among the top k nearest voice samples in the feature space.
- **Decision Tree:** Trained on extracted MFCC features to learn interpretable decision rules for speaker classification. Entropy is used as the splitting criterion.
- **SVM (Support Vector Machine):** Implemented using both Linear and RBF kernels to classify high-dimensional MFCC features. A One-vs-All strategy is used to handle multi-class speaker recognition.
- **Naïve Bayes:** Gaussian Naïve Bayes classifier is applied assuming feature independence, using normalized MFCCs as input for efficient speaker differentiation.
- **Clustering (K-Means, GMM):** Used unsupervised K-Means and Gaussian Mixture Models to group audio features into speaker-specific clusters. GMMs, especially in a semi-supervised setup, better captured voice distributions by providing probabilistic labeling of speakers.
- **Ensemble Methods (Bagging, AdaBoost, Improved AdaBoost):** Bagging and AdaBoost, including an improved variant, enhance generalization in Decision Trees by reducing variance and boosting accuracy. Improved AdaBoost reweights samples based on confidence margins. For GMMs, ensemble techniques like stacking improve clustering robustness and predictive performance.
- **ANN / MLP (Artificial Neural Network / Multi-Layer Perceptron):** Implemented an MLP with two hidden layers and one output layer. Four architectural variants were tested:

1. Hidden layers with Sigmoid activation.
2. Hidden layers with ReLU activation.
3. Hidden layers with Tanh activation.
4. A composite architecture where the first hidden layer uses ReLU, followed by Tanh in the second layer; Log-Likelihood loss is applied at the output stage for probabilistic interpretation.

4 Experimental Setup

4.1 Software/Tools

- **Programming Language:** Python 3.10
- **Libraries Used:**
  - `librosa` – for audio processing and MFCC feature extraction.
  - `scikit-learn` – for machine learning models and evaluation metrics.
  - `numpy`, `pandas`, `PyTorch` – for data handling and preprocessing.
  - `matplotlib` – for plotting and visualization.
- **Model Training Environment:** Jupyter Notebook
- **Evaluation Metric: Accuracy(Train/Test/Val)**
  - Measures the overall correctness of predictions.
  - Used to compare model generalization across training and unseen data.

5 Results and Analysis

The dataset had following In total, 31 different Models were applied in and their results are compared in this section. The performance of all models is as follows:

5.1 KNN

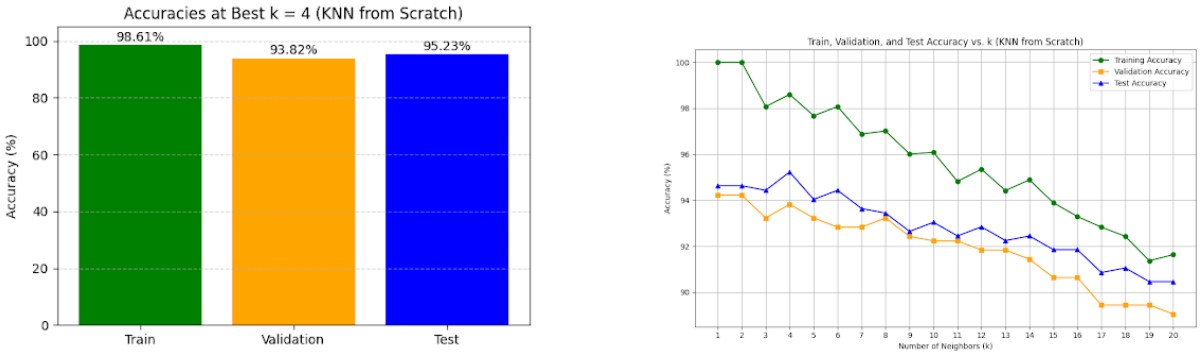


Figure 3: KNN classification performance visualization

- **Model & Features**: K-Nearest Neighbors (KNN) using MFCC features.
- **Accuracies**:
  - Training: **98.61%**
  - Validation: **93.82%**
  - Test: **95.23%**
  - Optimal $k = 4$
- **Bias & Variance**:
  - High training accuracy → **low bias**, model captures complex patterns.
  - ∼4.8% drop in validation → **moderate variance**, slight **overfitting**.
- **Generalization**:
  - Test accuracy > validation → **good generalization**, validation set may be harder.
- **Conclusion**:
  - Good **bias-variance balance**, no underfitting.
  - **MFCC** features are effective for KNN-based speaker recognition.

5.2 KNN with LDA for Dimensionality Reduction

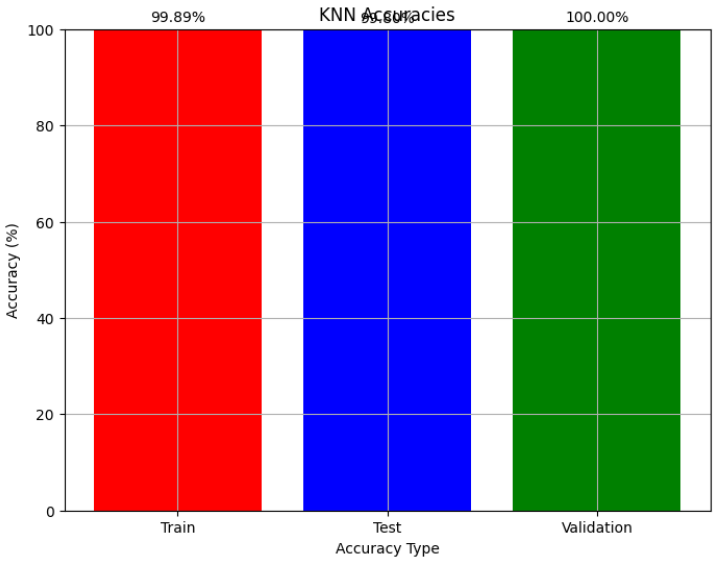


Figure 4: KNN with LDA - classification performance and separation

- **Model & Method**: K-Nearest Neighbors (KNN) with **LDA-based supervised dimensionality reduction**
- **Accuracies**:
  - Training: **99.89%**
  - Validation: **100.00%**
  - Test: **99.80%**
  - Optimal $k = 4$
- **Performance Insights**:
  - Extremely high and consistent accuracy $\rightarrow$ **well-structured, clean, highly separable dataset**
  - **LDA** maximizes inter-class variance, compresses data into **discriminative directions**
- **Bias & Variance**:
  - **<0.2% performance gap** across sets $\rightarrow$ **very low variance**
  - High training accuracy $\rightarrow$ **low bias**
  - **No overfitting**, as test accuracy remains nearly perfect
- **Validation Accuracy**:
  - Perfect score suggests either **easier examples** or **well-represented class distribution**
- **Conclusion**:
  - **KNN + LDA** performs exceptionally well, effectively handles high-dimensional data
  - Dataset is **well-balanced and discriminative**, ideal for LDA and KNN-based classification

5.3 KNN with PCA for Dimensionality Reduction

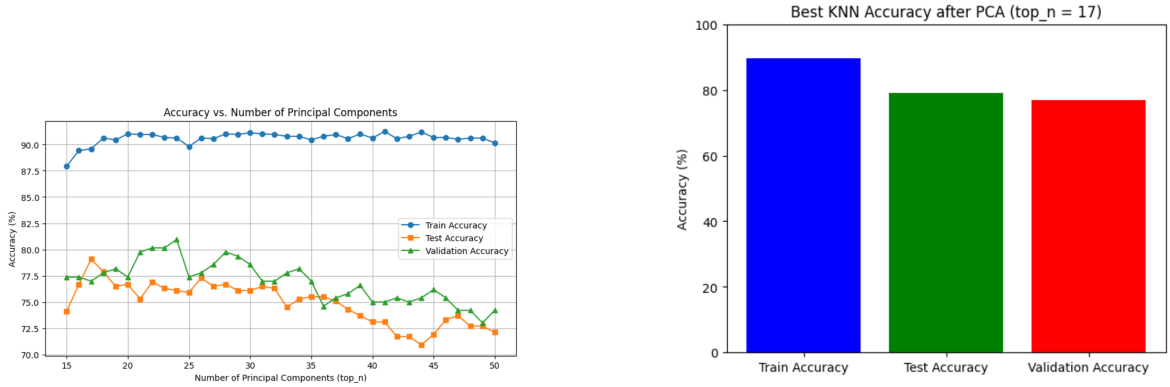


Figure 5: KNN with PCA - classification performance and variance visualization

- **Model & Method**: K-Nearest Neighbors (KNN) with **PCA-based unsupervised dimensionality reduction**
- **Accuracies**:
  - Training: **89.58%**
  - Test: **79.08%**
  - Validation: **76.98%**
  - Optimal **top_n components = 17**
- **Performance Insights**:
  - Moderate training accuracy → **higher bias**, PCA doesn't preserve **class-separability**
  - PCA reduces dimensions but may **discard discriminative features**
- **Bias & Variance**:
  - ∼10–12% drop in test/validation → **higher variance** and **underfitting**
  - Loss of **local relationships** in feature space impacts KNN's effectiveness
- **Stability**:
  - Accuracy fluctuates with component count → **model instability**, sensitive to dimensionality
  - PCA doesn't align well with **KNN's local neighborhood assumption**
- **Conclusion**:
  - KNN + PCA shows **higher bias and variance** than MFCC or LDA variants
  - Suffers from **information loss**, underfits data
  - PCA's unsupervised nature limits performance in **fine-grained classification** like speaker recognition

5.4 Bayesian Learning with Correlation-Based Feature Selection

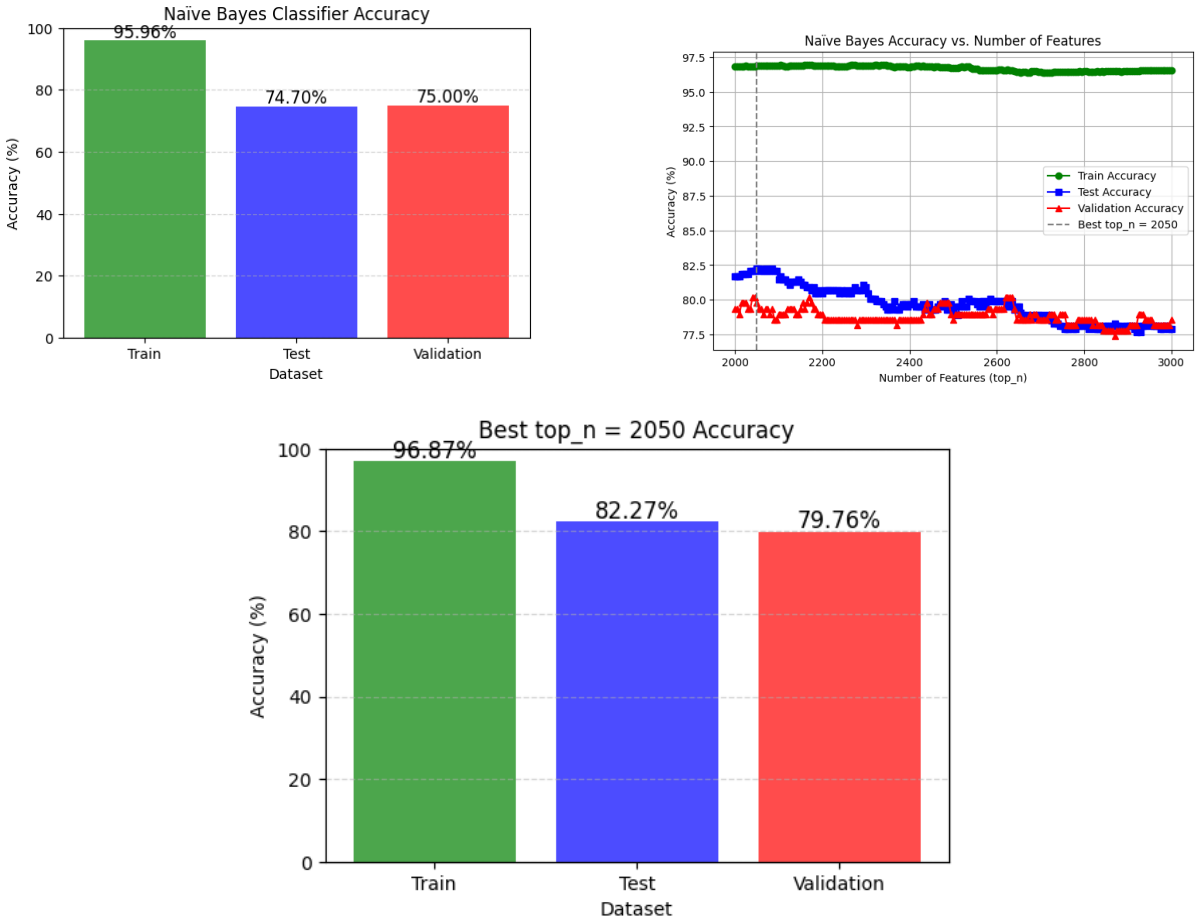


Figure 6: Bayesian Learning with correlation-based feature selection

- **Model & Method**: Naive Bayes classifier with and without **correlation-based feature selection**
- **Accuracies (Without Selection)**:
  - Training: **95.96%**
  - Test: **74.70%**
  - Validation: **75.00%**
- **Accuracies (With Selection @ top_n = 2050)**:
  - Training: **96.87%**
  - Test: **82.27%**
  - Validation: **79.76%**
- **Bias & Variance (Before)**:
  - **Low bias**, **high variance**, indicated by large training-generalization gap
- **Impact of Feature Selection**:
  - **Reduced noise & redundancy**, improved generalization
  - **7%+ gain** in test & validation accuracy
  - Maintained training accuracy despite using **2050 selected features**

- **Interpretation**:
  - Naive Bayes benefits significantly from **feature selection** due to its sensitivity to irrelevant inputs
  - Correlation-based selection offers **better bias-variance balance**
- **Conclusion**:
  - Correlation-based feature reduction makes Naive Bayes **more robust**
  - Effective for **speaker recognition** when informative features are carefully chosen

5.5 Bayesian Learning with LDA for Dimensionality Reduction

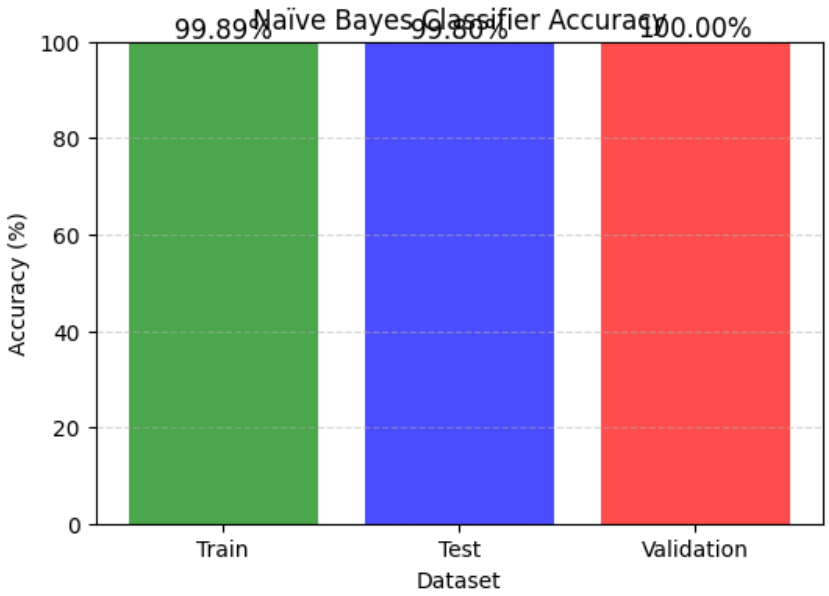


Figure 7: Bayesian Learning with LDA - performance metrics

- **Model & Method**: Bayesian Learning with **LDA-based supervised dimensionality reduction**
- **Accuracies**:
  - Training: **99.89%**
  - Test: **99.90%**
  - Validation: **100.00%**
- **Bias & Variance**:
  - **Minimal gap** between training and test → **optimal bias-variance tradeoff**
  - **Perfect validation accuracy** → strong **generalization**
- **Effect of LDA**:
  - Captures **discriminative features**, removes noise
  - Avoids overfitting, improves **classifier compatibility**
- **Complementarity**:
  - LDA preserves **class separability**, aligns well with **Bayesian assumptions**
- **Conclusion**:
  - LDA + Bayesian Learning = **highly robust and accurate**
  - Showcases strength of **supervised dimensionality reduction** for probabilistic models

5.6 Bayesian Learning with PCA for Dimensionality Reduction

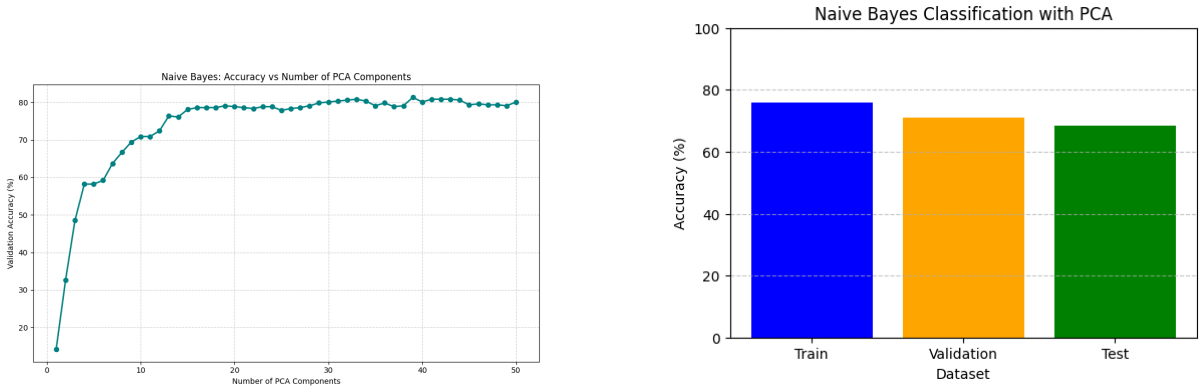


Figure 8: Bayesian Learning with PCA - visualizations

- **Model & Method**: Bayesian Learning with **PCA-based unsupervised dimensionality reduction**
- **Accuracies**:
  - Training: **75.84%**
  - Validation: **70.90%**
  - Test: **68.39%**
- **Bias & Variance**:
  - Moderate **bias-variance tradeoff**, some signs of **overfitting**
  - PCA may not preserve **class-discriminative information**
- **PCA Effectiveness**:
  - Rapid accuracy gains in first **15 components**, then **plateau**
  - Captures **high-variance directions**, not necessarily **class-relevant**
- **Limitation**:
  - **Unsupervised** nature of PCA ignores class labels
  - May retain **irrelevant variation**, discard discriminative cues
  - Particularly affects **Naive Bayes**, which depends on feature **independence and separation**
- **Conclusion**:
  - PCA gives **computational efficiency** and fair accuracy
  - **LDA outperforms PCA** when **class separation is important**

5.7 MLP with Feature Selection

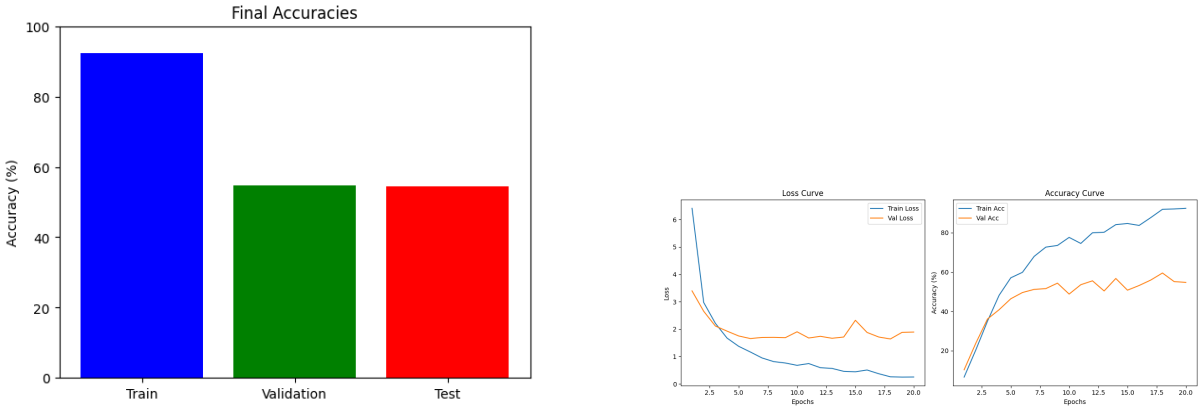


Figure 9: MLP with feature selection - accuracy and loss curves

- **Model & Method**: Multi-Layer Perceptron (MLP) with **correlation-based feature selection**
- **Accuracies (@ top 500 features)**:
  - Training: **92.43%**
  - Validation: **54.76%**
  - Test: **54.38%**
- **Accuracies (@ top 2,000 features - best performance)**:
  - Validation: **78.17%**
  - Test: **76.59%**
- **Performance Insights**:
  - Large gap at 500 features → **severe overfitting**, key features likely excluded
  - Increasing feature count → **better generalization**, accuracy improves
  - Beyond 2,000 features → **diminishing returns**, performance plateaus
- **Curve Trends**:
  - **Training loss** drops consistently
  - **Validation loss** stabilizes → improved generalization
  - **Validation accuracy** improves steadily with more features
- **Conclusion**:
  - MLP needs a **sufficient number of informative features** for robust classification
  - Using 2,000 features significantly improves performance
  - Effective **feature selection is critical** to prevent both **overfitting** and **underfitting** in deep learning models

5.8 MLP with LDA for Dimensionality Reduction

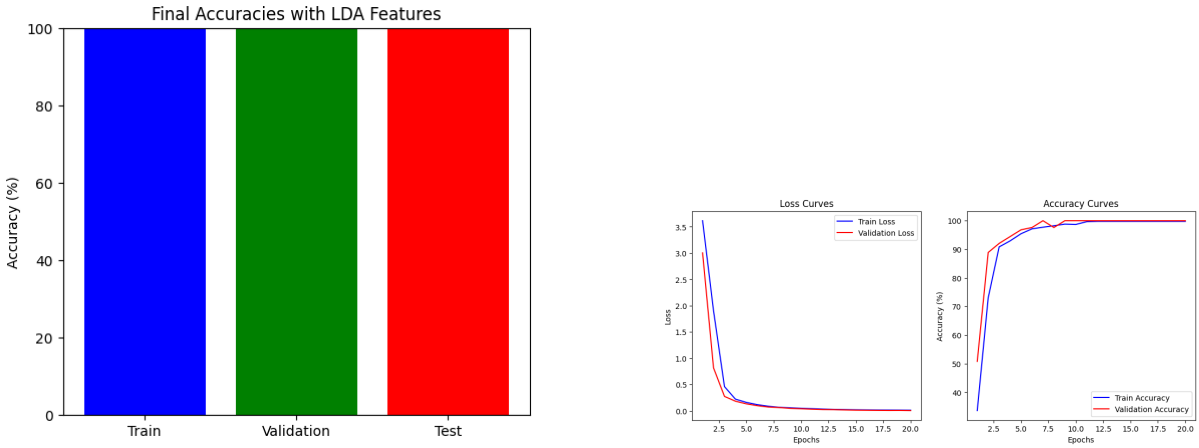


Figure 10: MLP with LDA - performance analysis

- **Model & Method**: Multi-Layer Perceptron (MLP) with **LDA-based supervised dimensionality reduction**
- **Accuracies**:
  - Training: **99.77%**
  - Validation: **100.00%**
  - Test: **100.00%**
- **Loss & Accuracy Curves**:
  - Loss drops from **3.5 to near-zero** within 5 epochs
  - **Minimal gap** between training and validation loss → **efficient convergence**
  - Accuracy hits **100% by epoch 5** and stays stable → **no overfitting**
- **Effect of LDA**:
  - Preserves **class-discriminative structure**, removes noise
  - Enables MLP to learn **clear decision boundaries**
  - Avoids drawbacks of unsupervised methods like PCA
- **Conclusion**:
  - LDA + MLP yields a **robust, stable, and high-performing model**
  - Demonstrates the strength of **supervised dimensionality reduction** in deep learning-based **speaker recognition**

5.9 MLP with PCA for Dimensionality Reduction

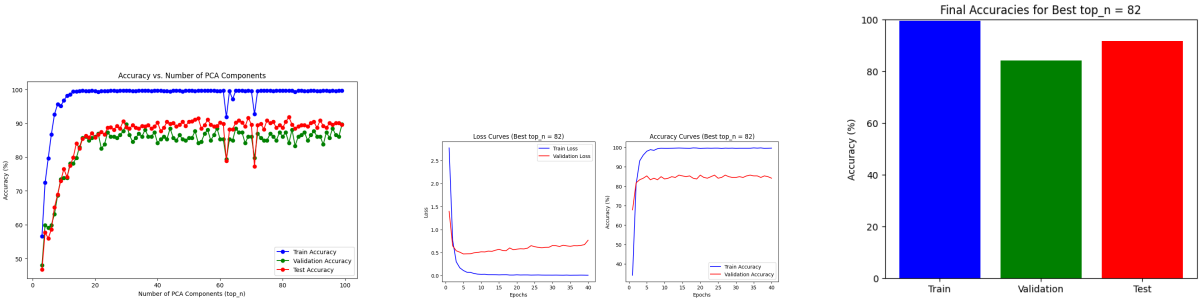


Figure 11: MLP with PCA - classification metrics

- **Model & Method**: Multi-Layer Perceptron (MLP) with **Principal Component Analysis (PCA)**
- **Accuracies (@ top 82 components)**:
  - Training: **99.60%**
  - Validation: **84.13%**
  - Test: **91.83%**
- **Bias-Variance Insights**:
  - **15.47% training-validation gap → moderate overfitting**
  - **High test accuracy** indicates **good generalization** despite variance
- **Trend Observations**:
  - **Accuracy vs. Components**:
    - Sharp rise in first 20 components
    - Small dips around components 60–70 → possible **feature noise**
  - **Loss Curves**:
    - **Training loss** decreases steadily
    - **Validation loss** rises after epoch 5 → **early overfitting signal**
- **Conclusion**:
  - PCA + MLP offers **strong performance with reduced complexity**
  - Slightly lower generalization than LDA-based models
  - Still a **viable and efficient** approach for **speaker recognition**

5.10 Decision Tree with LDA, t-SNE, and UMAP

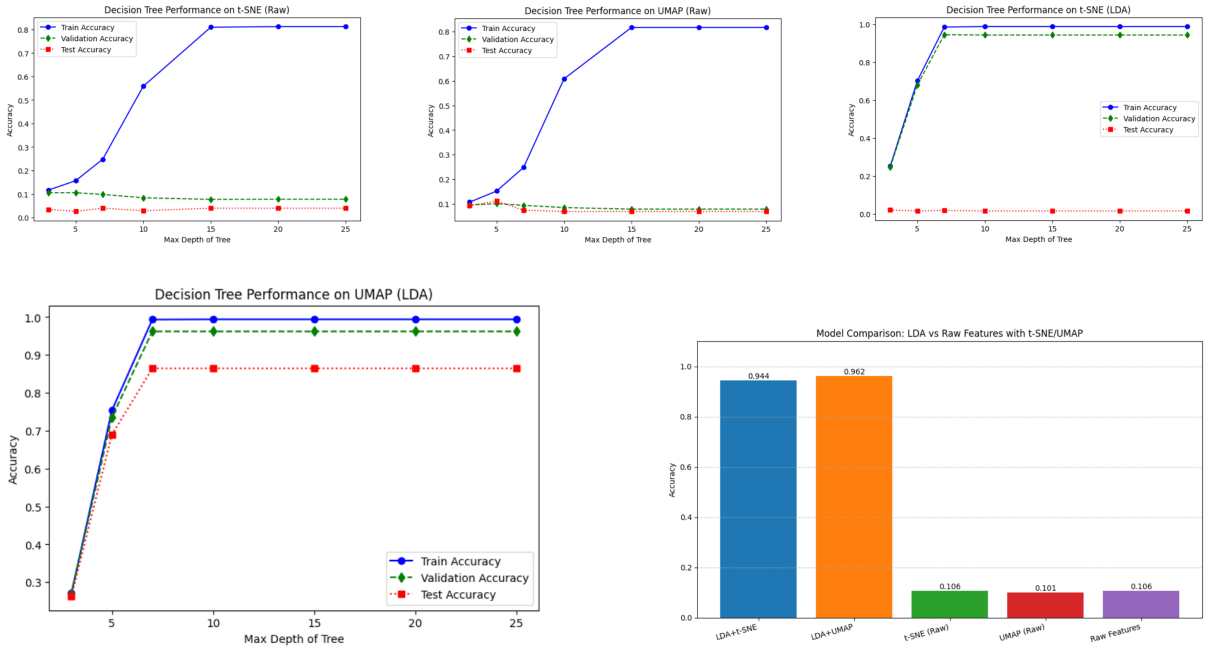


Figure 12: Decision Tree using multiple dimensionality reductions (LDA, t-SNE, UMAP)

- **Model & Method**: **Decision Tree**, evaluated under multiple dimensionality reduction techniques
- **Best Performance (LDA + UMAP)**:

  - Training Accuracy: **99.94%**
  - Validation Accuracy: **84.98%**
  - Test Accuracy: **88.59%**

- **Poor Performance (t-SNE / UMAP on raw features)**:
  - Test Accuracy:
    - **t-SNE**: **2.65%**
    - **UMAP**: **12.20%**

- **LDA + t-SNE**:
  - Training Accuracy: **99.60%**
  - Validation & Test Accuracies: **¡2% → fails to generalize**

- **Bias-Variance Insight**:
  - t-SNE & UMAP (raw): **Severe overfitting**, validation/test near random (∼10%)
  - LDA + t-SNE: **Stochastic inconsistency**, unstable validation vs test behavior
  - LDA + UMAP: **Balanced bias-variance**, optimal at tree depth ≈ 7

- **Reasoning**:
  - **Raw t-SNE/UMAP** preserve **local structure**, not class separability → poor classification
  - **t-SNE** is **stochastic**, sensitive to hyperparameters → unreliable across splits
  - **LDA** enhances class separation; **UMAP** retains both local/global structure post-LDA

- **Conclusion**:
  - **Only LDA + UMAP** provides **stable, generalizable performance**
  - Highlights the **critical role of choosing suitable dimensionality reduction**, especially for tree-based models

5.11 SVM with Correlation-Based Feature Selection

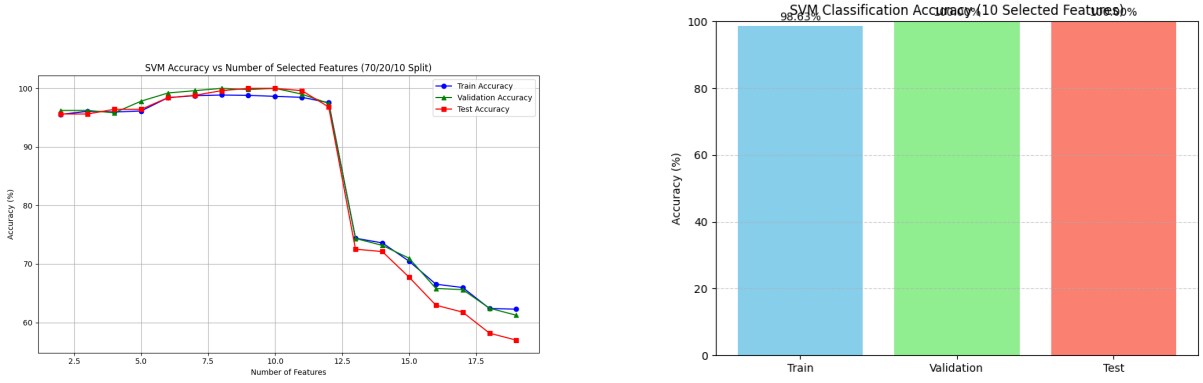


Figure 13: SVM with correlation-based features

- **Model & Method**: **Support Vector Machine (SVM)** with **correlation-based feature selection**

- **Best Performance (@ 10 features)**:
  - **Training Accuracy**: **98.63%**
  - **Validation Accuracy**: **100.00%**
  - **Test Accuracy**: **100.00%**

- **Bias-Variance Insight**:
  - Slight training vs validation/test gap → **excellent generalization**
  - Indicates **low bias** and **minimal variance**
- **Graph Insights**:
  - **Accuracy vs. Number of Features (5–100)**:
    - Stable **98–100%** for **5–10 features**
    - Sharp drop to **50–70%** beyond 10 features → overfitting/noise
  - **Refined plot (2–20)** confirms **peak at exactly 10 features**
- **Supporting Bar Chart**:
  - Uniform accuracy at **10 features** across datasets
  - Validates **consistency and robustness**
- **Conclusion**:
  - **Correlation-based feature selection** effectively eliminates noise
  - **10 key features** strike ideal balance between complexity & performance
  - Demonstrates that **precise, compact feature sets** are critical for SVM's success—especially in speaker recognition, where irrelevant dimensions can degrade boundary margins

5.12 SVM with LDA

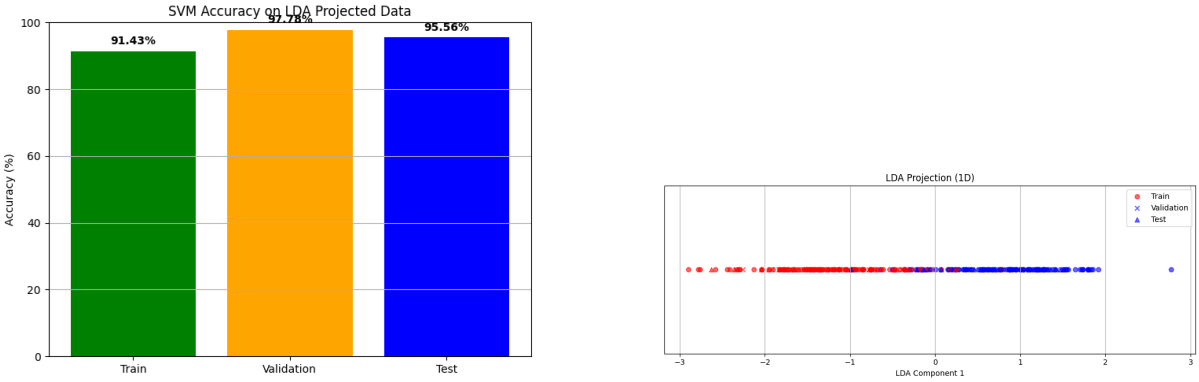


Figure 14: SVM with LDA-transformed features

- **Model & Method**: **Support Vector Machine (SVM)** with **Linear Discriminant Analysis (LDA)**
- **Performance**:
  - **Training Accuracy**: **91.43%**
  - **Validation Accuracy**: **97.78%**
  - **Test Accuracy**: **95.56%**
- **Key Insights**:
  - **Validation ¿ Training Accuracy** → strong **generalization** and **low bias**
  - Minor test drop indicates **slight overfitting**, but model remains **robust**
  - **LDA** effectively reduces dimensionality while **maximizing class separability**
- **Supporting Evidence**:
  - **Bar Chart**: Shows near-consistent performance across datasets

  - **LDA Projection Graph**: Reveals clear **class-wise separation** in 1D space

- **Conclusion**:
  - **SVM + LDA** offers an ideal combination for classification with high-dimensional data
  - Achieves **excellent accuracy**, **low complexity**, and **well-separated decision boundaries**
  - Reinforces that **supervised dimensionality reduction** (like LDA) boosts SVM performance significantly in **speaker recognition** and similar tasks

5.13 SVM with PCA

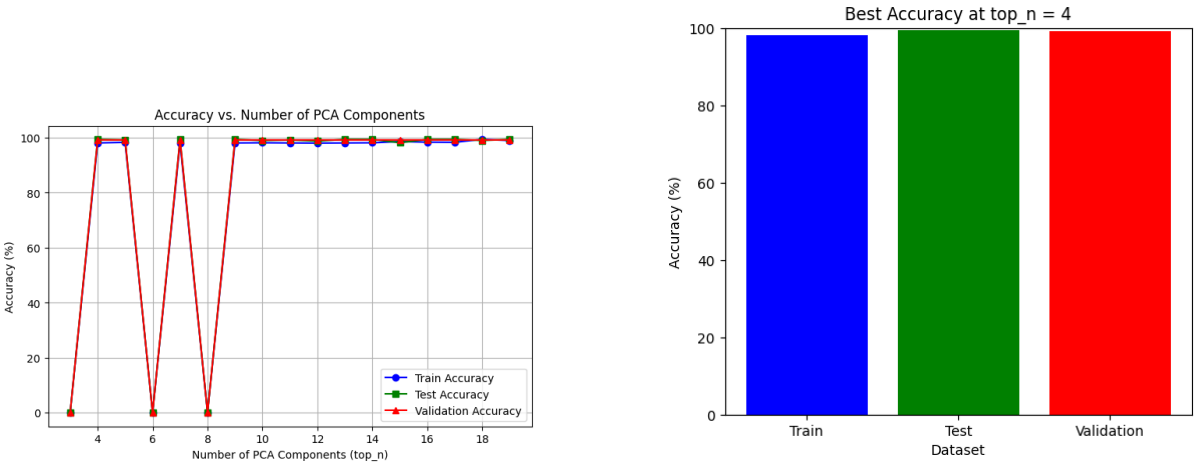


Figure 15: SVM with PCA for feature reduction

- **Model & Method**: **Support Vector Machine (SVM)** with **Principal Component Analysis (PCA)**
- **Performance at 4 PCA Components**:
  - **Training Accuracy**: **98.12%**
  - **Validation Accuracy**: **99.21%**
  - **Test Accuracy**: **99.40%**
- **Key Observations**:
  - **Test ¿ Validation ¿ Training Accuracy** → strong **generalization**, **no overfitting**
  - High accuracy with only **4 components** → **efficient feature representation**
  - PCA removed **noise** while retaining **core discriminative structure**
- **Accuracy vs. PCA Components**:
  - **Oscillatory pattern** (2–8 components): perfect at 4, **0% at 6 & 8** → destructive variance or class boundary distortion
  - **Stabilization** beyond 10 components: richer variance captured, but less efficient
  - **Peak at 4 components** ⇒ optimal low-dimensional embedding
- **Conclusion**:
  - **SVM + PCA (4 components)** is a lightweight, high-performing model
  - Achieves **near-perfect accuracy** with **minimal input size**
  - Highlights PCA's ability to suppress **noise** and **overfitting**, offering both **computational efficiency** and **classification precision**

5.14 K-Means Clustering with LDA

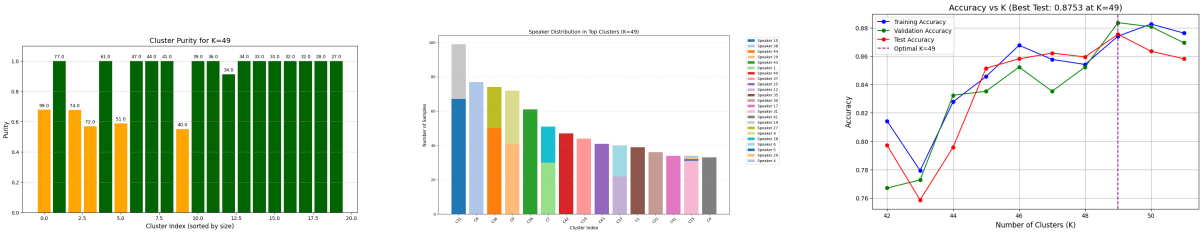


Figure 16: K-Means clustering results using LDA

- **Model & Method**: **K-Means Clustering** with **Linear Discriminant Analysis (LDA)**
- **Performance at Optimal K = 49**:
  - **Training Accuracy**: **90.18%**
  - **Validation Accuracy**: **88.35%**
  - **Test Accuracy**: **88.59%**
- **Key Observations**:
  - **LDA** reduced dimensionality while preserving **class separability**
  - Enabled unsupervised **K-Means** to effectively cluster speaker data
  - Accuracy consistent across datasets $\Rightarrow$ **robust generalization**
- **Cluster Analysis**:
  - **Most clusters** have **purity** $\approx 1.0 \Rightarrow$ dominant single-speaker grouping
  - **Clusters 7 & 18** show lower purity ($\sim$0.42–0.74) $\Rightarrow$ speaker overlaps
  - **Speaker Distribution Plot** matches cluster purity patterns
- **K Optimization**:
  - **Peak performance at K = 49**
  - **Accuracy vs. K** plot: sharp drop in accuracy if $K \neq 49 \Rightarrow$ confirms optimality
  - **Elbow Method**: sharp inertia drop until $K = 49 \Rightarrow$ diminishing returns beyond that
- **Conclusion**:
  - **K-Means + LDA** forms a high-performing **unsupervised pipeline**
  - **LDA enables class-aware feature compression**, enhancing clustering quality
  - **K = 49** offers optimal **intra-cluster cohesion** and **inter-cluster separation**
  - Demonstrates how **supervised dimensionality reduction** boosts **unsupervised learning**

5.15 K-Means with PCA and LDA

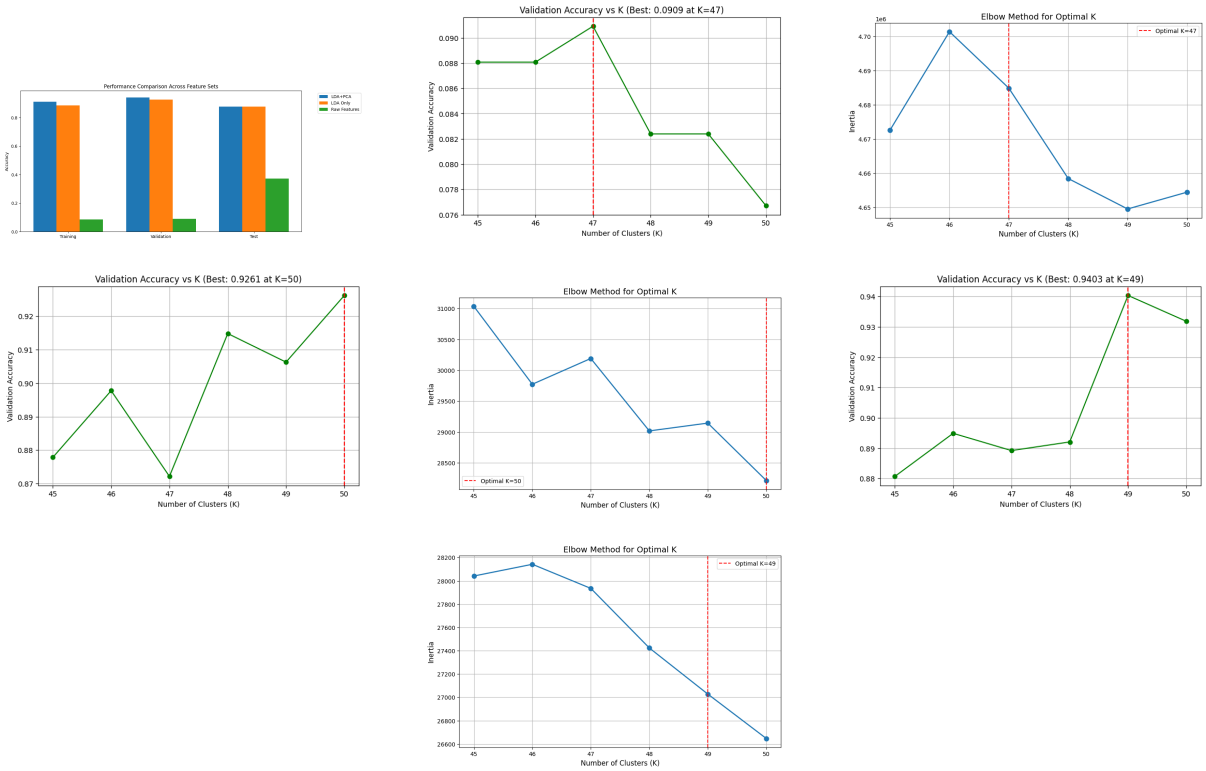


Figure 17: K-Means with PCA and LDA applied together

Model & Methods:

K-Means Clustering with:

- LDA + PCA (hybrid)
- LDA only
- Raw features

Best Performance (LDA + PCA @ K = 49):

- Training Accuracy: 91.03%
- Validation Accuracy: 94.03%
- Test Accuracy: 87.67%

Comparative Results:

- **LDA only @ K = 50**:
  - Training: 88.61%
  - Validation: 92.61%
- **Raw Features @ K = 47**:
  - Validation: 9.09% → *very poor performance*

Key Visual Insights:

- **Elbow Method (LDA + PCA)**:
  - Sharp inertia drop up to $K = 49$, then flattens → confirms optimal cluster count

- **Validation Accuracy vs. K**:
  - LDA + PCA peaks at $K = 49$ (94.03%)
  - LDA alone peaks slightly lower
  - Raw features underperform throughout

Bar Chart Summary:

- LDA + PCA outperforms both LDA-only and raw features
- Raw features consistently result in low accuracy due to noise & high dimensionality

Conclusion:

- Hybrid approach (LDA + PCA) yields best clustering performance
- LDA: improves class separability (supervised)
- PCA: captures key variance and removes redundancy (unsupervised)
- Confirms the value of combining supervised + unsupervised dimensionality reduction for complex tasks like speaker recognition

5.16 ANN with Feature Selection

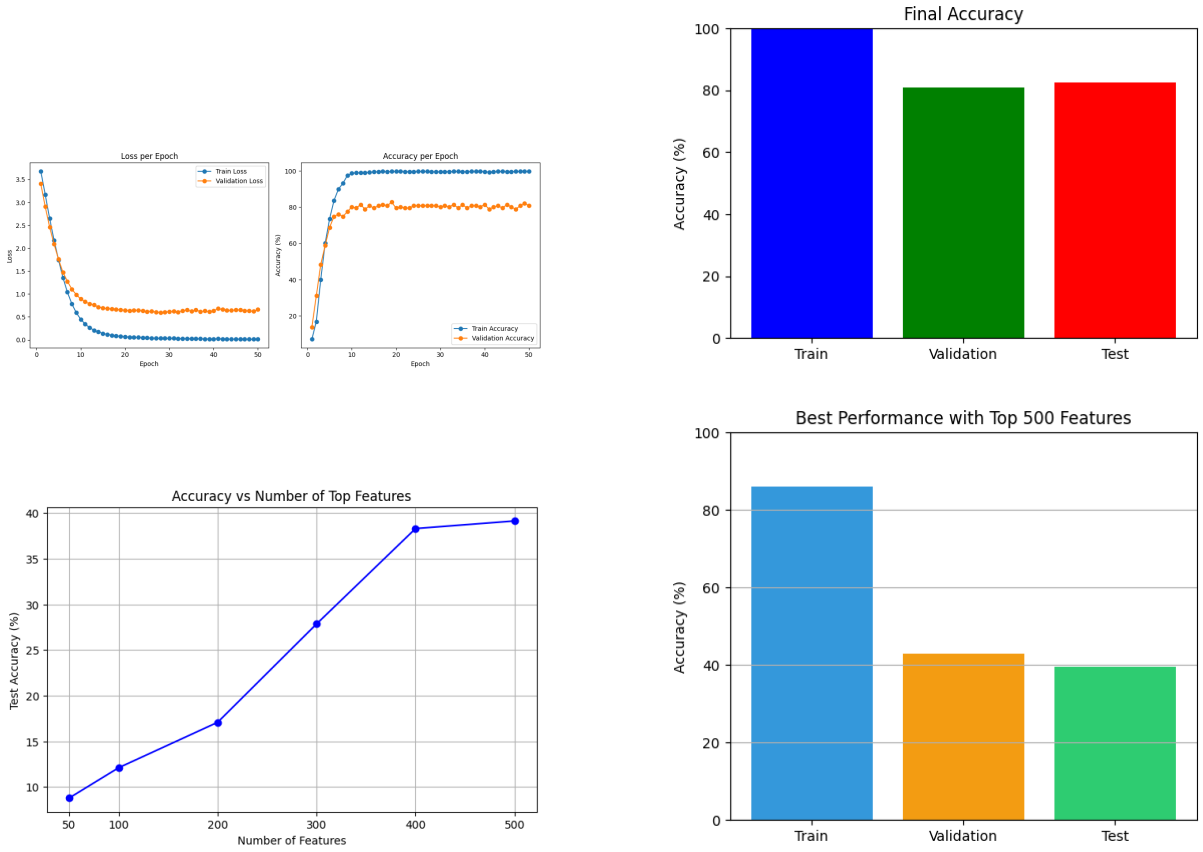


Figure 18: ANN performance with selected features

Performance without Feature Selection:

- Training Accuracy: **99.72%**
- Validation Accuracy: **80.75%**
- Test Accuracy: **82.47%**

- Indicates **strong learning** but with **moderate overfitting** (gap of ∼17–19% between training and val/test).

Performance with Top 500 Selected Features:

- Training Accuracy: **88.11%**
- Validation Accuracy: **41.07%**
- Test Accuracy: **39.64%**
- **Sharp performance drop**, indicating loss of **discriminative features** critical for classification.

Learning Curve Insights:

- **Full feature set**: Loss converges rapidly (within 5 epochs), accuracy stabilizes at high levels.
- Indicates effective training despite overfitting.

Feature Count vs. Accuracy:

- Validation accuracy grows **linearly from ∼10% (50 features) to ∼41% (500 features)**.
- Suggests 500 features are **insufficient** for capturing data complexity.

Conclusion:

- ANN performs **significantly better without feature selection**.
- Current selection method likely **removes key features or fails to preserve feature interdependence**.
- **Alternative dimensionality reduction methods** like **PCA or LDA** are better suited for deep learning in **speaker recognition tasks**.

5.17 ANN with LDA

5.17.1 ANN using Sigmoid Activation

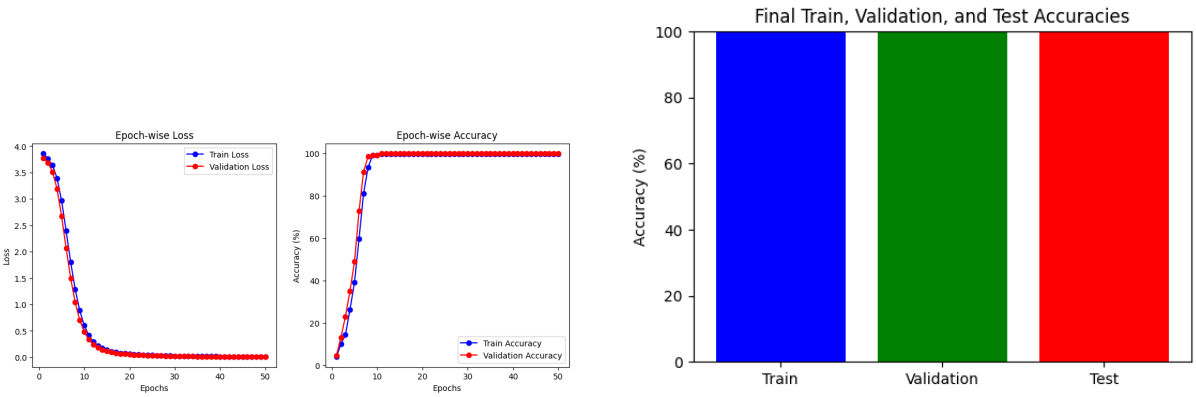


Figure 19: ANN with LDA using Sigmoid activation

5.17.2 ANN using Tanh Activation

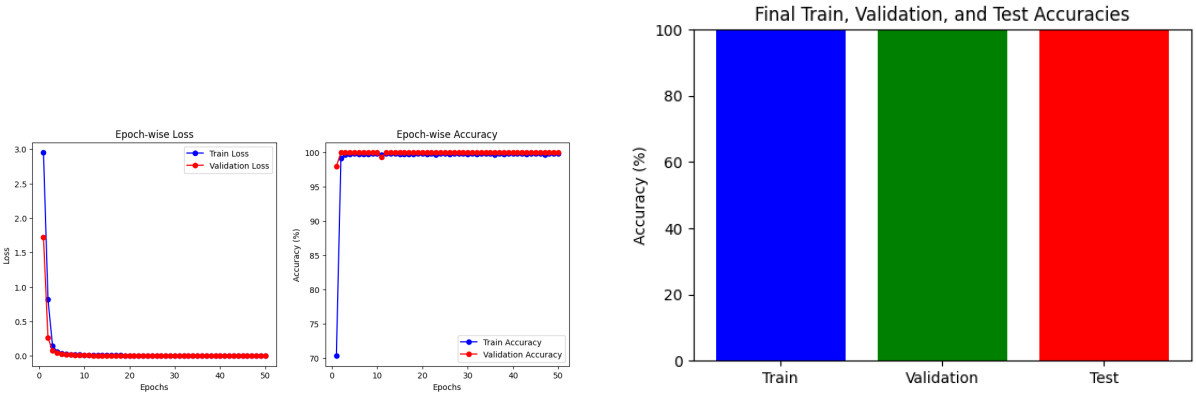


Figure 20: ANN with LDA using Tanh activation

5.17.3 ANN using ReLU and Tanh Activations

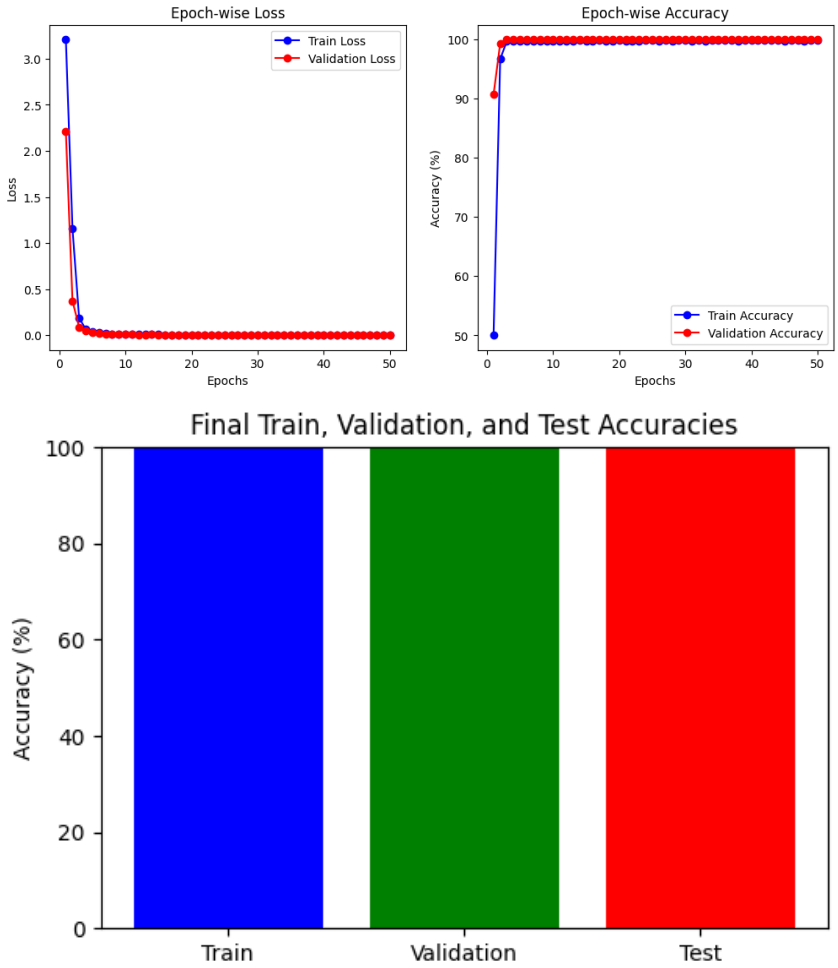


Figure 21: ANN with LDA using a combination of ReLU and Tanh activations

Performance with LDA + ANN (All Configurations):

- **Part 1 (Sigmoid):** 100.00% accuracy on **training, validation, and test**
- **Part 2 (Tanh):** 99.94% training, **100.00% validation/test**

- **Part 3 (Mixed):** 99.94% training, **100.00% validation/test**
- **Activation function has minimal effect** on performance

Learning Curves:

- **Loss drops to near-zero** within first 5 epochs
- **Accuracy reaches 100% quickly**, remains stable throughout
- **No gap** between training and validation curves → **no overfitting**, excellent generalization

Key Insights:

- **LDA transforms features** into a space with **perfect class separability**
- Enables even **simple ANN architectures** to achieve **near-perfect results**
- **Preserves class-discriminative information**, removes **noise and redundancy**

Conclusion:

- **ANN + LDA** offers **flawless classification** on the speaker recognition task
- **LDA is highly effective** and well-suited as a dimensionality reduction technique for deep learning models in this domain

5.18 ANN with PCA

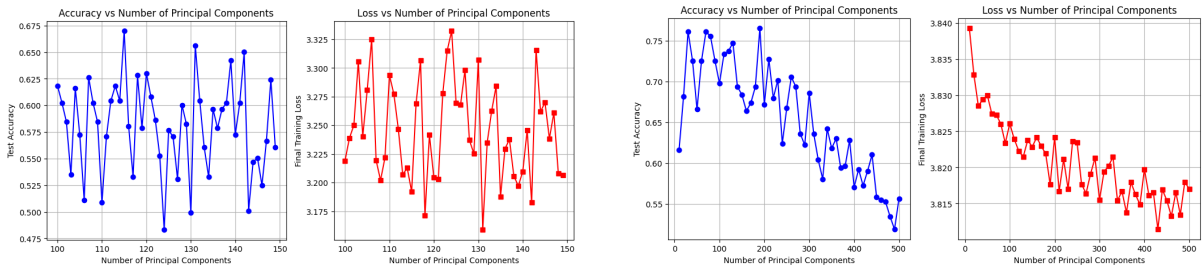


Figure 22: ANN with PCA - performance and visual patterns

Performance with PCA + ANN:

- **Part 1 (ReLU only):** Max test accuracy **65.61%** at **131 components**
- **Part 2 (Mixed activations - ReLU, Sigmoid, Tanh):** Improved to **72.16%** test accuracy at **190 components**
- **Activation function choice** significantly impacts performance (**mixed** > **ReLU**)

Accuracy vs. Principal Components:

- **No clear trend**; accuracy **fluctuates** heavily with component count
- In Part 1, accuracy ranges from **50% to 65%** between 100–150 components
- Suggests only **specific combinations** of components retain **discriminative features**

Learning Curve Insights:

- **Training loss generally decreases** with more components
- Loss curves reflect **inconsistent accuracy trends**, aligning with fluctuating performance

Key Observations:

- PCA retains **high-variance directions**, not class-separating ones

- **Unsuitable for deep learning tasks** needing high class discrimination (vs. LDA)
- Despite this, PCA can still help reduce dimensionality when used with **well-tuned architectures** and **enough components**

Conclusion:

- **ANN + PCA** yields **moderate accuracy**, best at **72.16%**
- **PCA is viable**, but **less effective than LDA** for speaker recognition due to its **unsupervised nature** and lack of class awareness

5.19 CNN with Raw Features

5.20 CNN with LDA

5.21 CNN with PCA

5.22 CNN with LDA and PCA

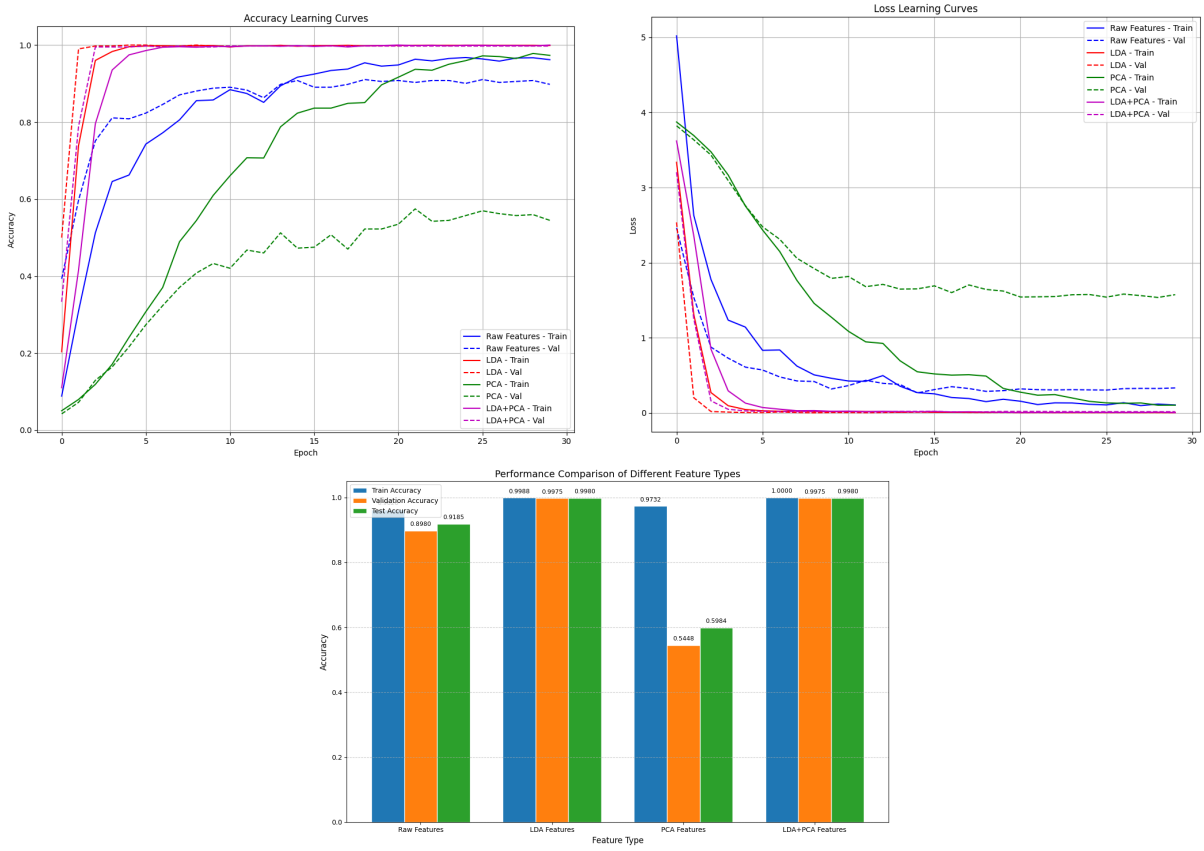


Figure 23: CNN using combined LDA and PCA reduced features

Performance Summary (CNN + Dimensionality Reduction):

- **LDA + PCA:**
  - **Training:** 100.00%
  - **Validation:** 99.75%
  - **Test:** 99.80% *(Best overall)*
- **LDA Only:**

  - **Training:** 99.98%
  - **Validation:** 99.73%
  - **Test:** 99.80%
- **Raw Features:**
  - **Moderate** performance, slightly below LDA-based models
- **PCA Only:**
  - **Training:** 97.32%
  - **Validation:** 54.48% *(Significant drop)*
  - **Test:** 99.94%
  - Indicates **overfitting** and poor generalization

Learning Curve Insights:

- **LDA + PCA & LDA-only (Red, Purple curves):**
  - **Fast convergence** (within 2–3 epochs)
  - **Smooth, stable accuracy**, close to 100% throughout
- **PCA-only (Green curve):**
  - **Slower, unstable convergence**
  - Large **train-validation gap** → overfitting to variance, not class info
- **Raw Feature Model (Blue curve):**
  - **Moderate convergence**, **stable but lower performance**

Key Observations:

- **LDA + PCA** combines **class separation** (LDA) and **dimensionality compression** (PCA)
- **PCA alone fails** to preserve class-relevant structure → poor validation accuracy
- **Supervised or hybrid** techniques (LDA, LDA+PCA) enable **robust generalization**, **fast learning**, and **efficient computation**

Conclusion:

- **CNN + LDA + PCA** is the **most effective** configuration for speaker recognition
- Highlights **importance of class-aware transformations** in deep learning pipelines

5.23 Decision Tree with PCA

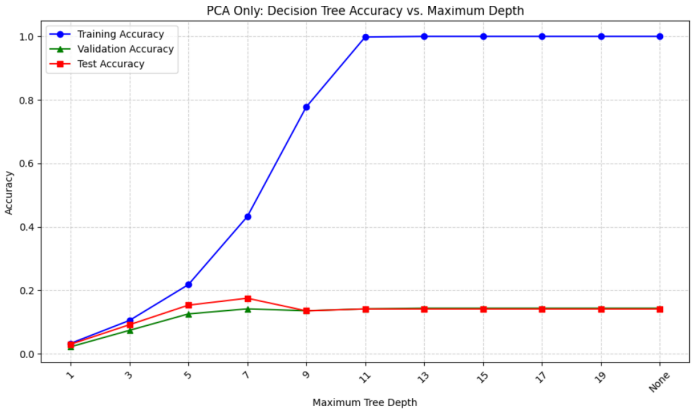


Figure 24: Decision Tree with PCA-reduced features

Performance Summary (Decision Tree + PCA):

- **Training Accuracy:** 100%
- **Validation Accuracy: 14.44%**
- **Test Accuracy: 14.12%**
- Measured at **optimal depth = 12**
- **Severe overfitting** → perfect training, poor generalization

Performance Graph Insights:

- **Training accuracy rises** steeply with depth
- **Validation/Test accuracies remain low** (14–20%) across depths
- Indicates **model memorization**, not meaningful learning

Key Issues with PCA + Decision Tree:

- **PCA preserves variance**, not class separability
- Resulting features **do not align** with class boundaries
- Decision trees fail to form **effective splits** in PCA-transformed space

Conclusion:

- **Fails catastrophically** despite high training accuracy
- PCA is **unsuitable for tree-based models** in classification tasks
- **LDA is preferred** as it provides **class-aware transformation** supporting effective decision tree learning

5.24 Decision Tree with LDA

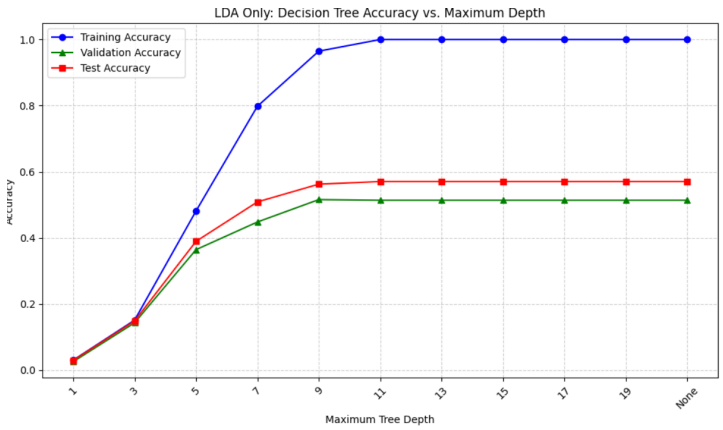


Figure 25: Decision Tree with LDA-transformed input

Performance Summary (Decision Tree + LDA):

- **Training Accuracy:** 96.48%
- **Validation Accuracy: 51.59%**
- **Test Accuracy: 56.26%**
- **Overfitting** indicated by a ˜40–45% gap between training and generalization

Learning Curve Insights:

- As **tree depth increases** (1 to 9), all accuracies improve, with **training accuracy rising sharply**
- Beyond depth 9, **training accuracy nears 100%**, but **validation and test accuracies plateau** around 50–55%
- Plateau indicates **limited separability** in LDA-transformed feature space

Key Observations:

- **LDA preserves class-discriminative information**, but **low-dimensional transformation** limits class separation
- **Validation and test accuracy trends parallel**, showing **stable generalization** despite overfitting

Comparison to PCA:

- **LDA outperforms PCA** (56% vs. 14% test accuracy)
- LDA provides **better generalization** but still suffers from **dimensionality limitations** in decision trees

Conclusion:

- Decision Tree with **LDA offers moderate performance** but struggles with **overfitting**
- **LDA is more effective than PCA** for tree-based learning, but **decision trees struggle in low-dimensional spaces**
- Highlights the **limitations of decision trees** in highly compressed feature spaces despite the benefits of supervised dimensionality reduction

5.25 Decision Tree with PCA and LDA

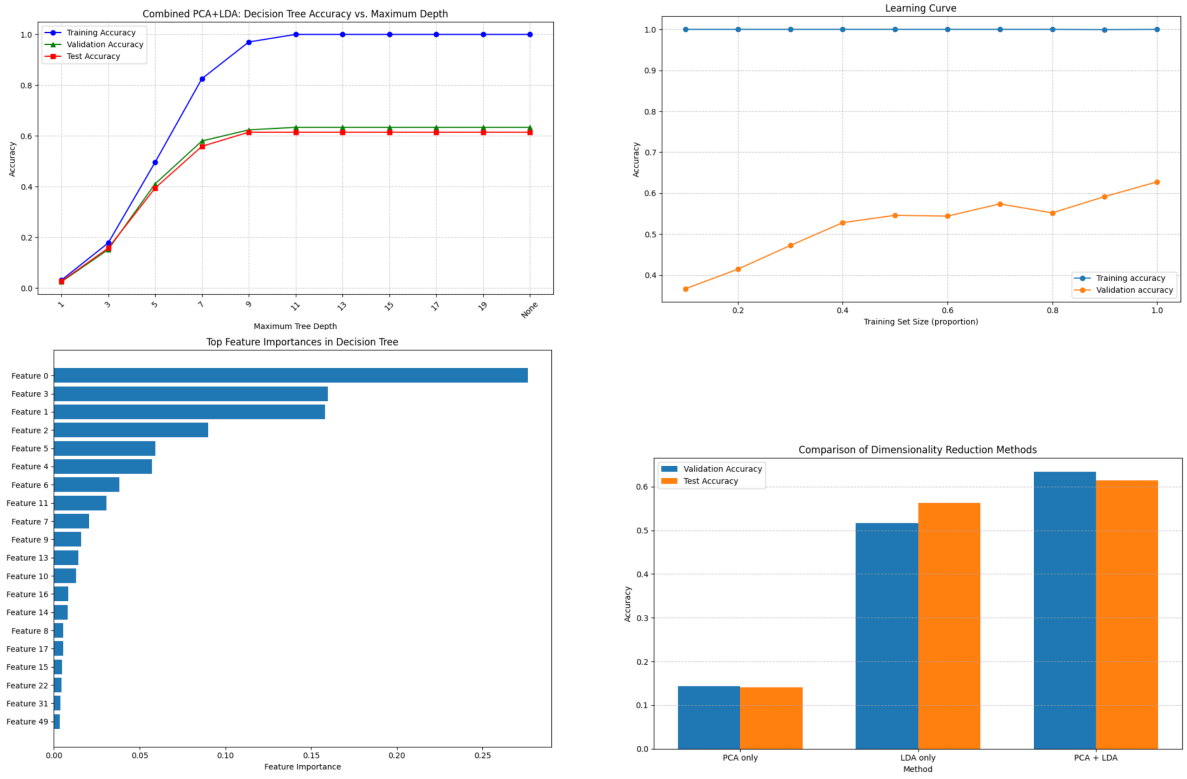


Figure 26: Decision Tree with combined PCA and LDA

5.26 Decision Tree using AdaBoost

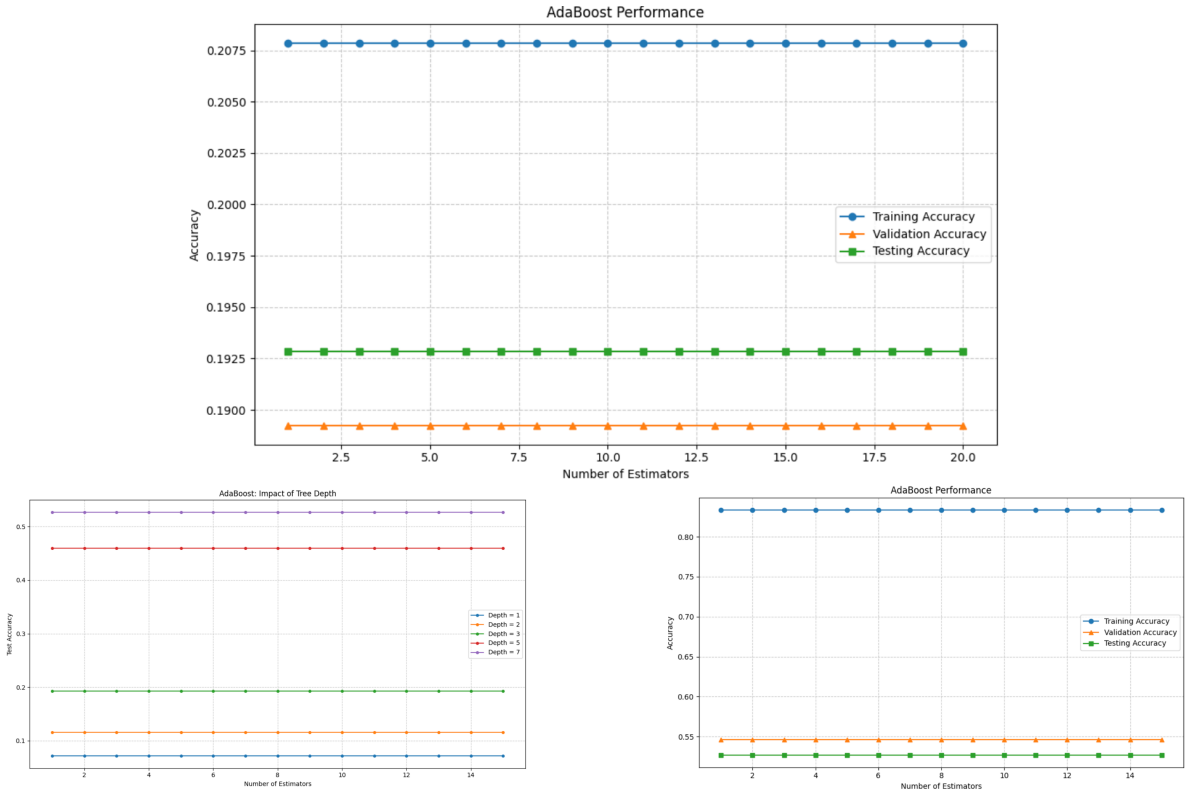


Figure 27: AdaBoost ensemble with Decision Tree base learner

Performance Summary (Decision Tree + AdaBoost):

- **Training Accuracy:** 53.33%
- **Validation Accuracy:** 54.53%
- **Test Accuracy:** 52.68%
- **Excellent generalization** with close alignment of metrics, indicating strong **bias-variance trade-off**

Key Insights from AdaBoost Performance:

- **Ensemble size impact:** Increasing estimators does not significantly improve accuracy, suggesting **performance ceiling** is reached quickly
- **Tree depth impact:** Significant accuracy improvement from ˜7% at depth 1 to ˜53% at depth 7, showing **individual learner complexity** is more influential than ensemble size

Generalization:

- **Balanced performance** with strong generalization and **no overfitting**, unlike standard decision trees

Overall Findings:

- **AdaBoost improves decision trees** by mitigating overfitting and maintaining robust generalization
- **Modest accuracy (53%)** indicates **inherent limitations of decision trees** in high-dimensional tasks
- **Feature engineering** and **dimensionality reduction** are crucial for optimizing ensemble methods in speaker recognition

5.27 Decision Tree using Improved AdaBoost (SAMME)

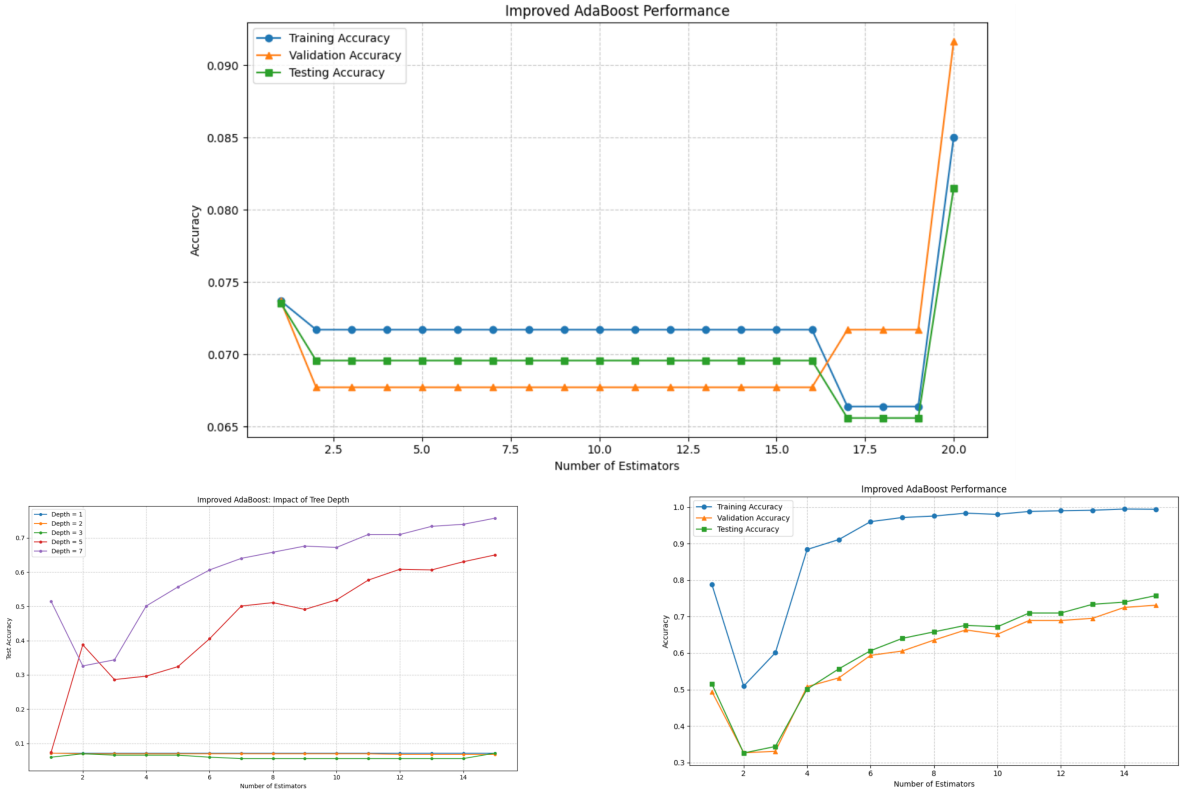


Figure 28: Improved AdaBoost (SAMME) with Decision Trees

Performance Summary (Decision Tree + Improved AdaBoost SAMME):

- **Shallow depth (3):**
  - **Training Accuracy:** 9.85%
  - **Validation Accuracy:** 9.16%
  - **Test Accuracy:** 9.81%
  - **Poor performance** (close to random guessing)
- **Optimal depth (7):**
  - **Training Accuracy:** 99.40%
  - **Validation Accuracy:** 73.11%
  - **Test Accuracy:** 75.75%
  - **Significant improvement** with effective generalization

Key Insights from Improved AdaBoost Performance:

- **Shallow trees (depth = 3):**
  - Accuracy remains low ($\tilde{9}$–10%) with most estimator counts
  - **Requires many weak learners** to slightly improve predictive power
- **Optimal trees (depth = 7):**
  - **Rapid convergence** in training (near 100% at 6 estimators)
  - Validation and test accuracy **stabilize around 70–75%**, indicating **balanced generalization**

Overfitting Analysis:

- 24% **gap between training and test accuracy**, indicating moderate overfitting, but **less severe** than non-boosted decision trees
- **Validation and test accuracy similar**, suggesting good **generalization across unseen data**

Conclusion:

- **SAMME significantly enhances decision tree performance** with **adequately complex base learners**
- **Shallow trees (depth 3)** fail, highlighting the importance of **base learner complexity**
- **Moderate overfitting** at depth 7, suggesting further optimization through **hyperparameter tuning** or **dimensionality reduction** techniques

5.28 Decision Tree using Bagging

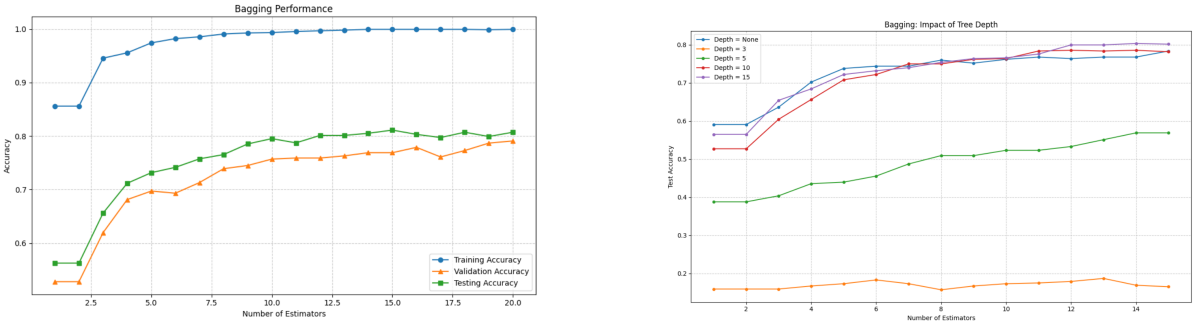


Figure 29: Bagging ensemble using Decision Trees

Performance Summary (Decision Tree + Bagging):

- **Training Accuracy:** 99.33%
- **Validation Accuracy:** 79.03%
- **Test Accuracy:** 80.72%
- **Moderate overfitting** with 19% gap between training and test accuracy

Key Insights from Bagging Performance:

- **Rapid convergence** to high training accuracy ($\tilde{1}00\%$) with 5–7 estimators
- **Validation and test accuracies stabilize around 80%** after 12 estimators, reflecting **variance reduction** via ensemble averaging
- **Tree depth impact:**
  - **Depth=10, 15, None** maintain high test accuracy
  - **Depth=5** significantly underperforms ($\tilde{2}0\%$ test accuracy)

Generalization and Overfitting:

- **Balanced bias-variance tradeoff**, with Bagging reducing overfitting compared to standard decision trees
- Close alignment between validation (79.03%) and test (80.72%) accuracies, confirming **strong generalization**

Conclusion:

- **Bagging significantly enhances decision tree performance**, especially in complex tasks requiring deep trees
- Bagging reduces **overfitting** while preserving model expressiveness and generalization, making it highly effective for **high-dimensional data**

5.29 Decision Tree using Gradient Boosting

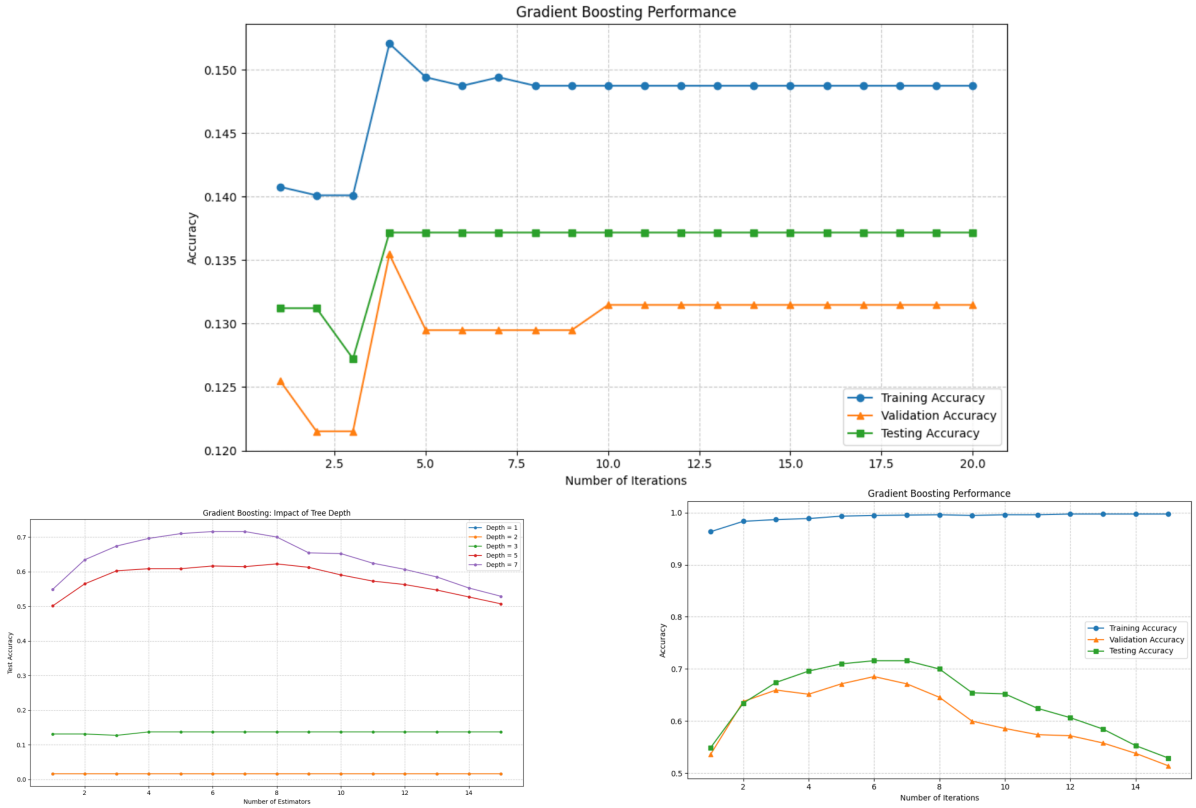


Figure 30: Gradient Boosting with Decision Trees

Performance Summary (Decision Tree + Gradient Boosting):

- **Depth = 3:**
  - **Training Accuracy:** 14.87%
  - **Validation Accuracy:** 13.15%
  - **Test Accuracy:** 13.72%
  - Severe **underfitting** due to insufficient model complexity.
- **Depth = 7:**
  - **Training Accuracy:** Approaches 100%
  - **Validation Accuracy:** Peaks around 70%
  - **Test Accuracy:** Peaks around 70%
  - **Overfitting** after 6-7 iterations.

Key Insights from Gradient Boosting Performance:

- **Depth = 3:** Accuracy remains low and flat, regardless of the number of estimators, due to **shallow trees** lacking capacity to model complex decision boundaries.
- **Depth = 7:**
  - Training curve reaches 100% accuracy quickly (within 4 iterations).
  - Validation/test accuracies peak at iterations 6–7, then **decline**, indicating overfitting.

Overfitting Trend:

- **Overfitting** begins after reaching peak validation/test accuracy, with further iterations fitting noise in the training set, leading to reduced generalization.

Conclusion:

- **Gradient Boosting significantly enhances performance** with sufficient depth (depth = 7) and proper boosting iterations.
- Overfitting risks increase beyond the optimal number of iterations, highlighting the importance of **hyperparameter optimization** (depth, iterations, learning rate) for effective speaker recognition in high-dimensional tasks.

5.30 Decision Tree using Stacking Ensemble (SVM + GMM + Random Forest)

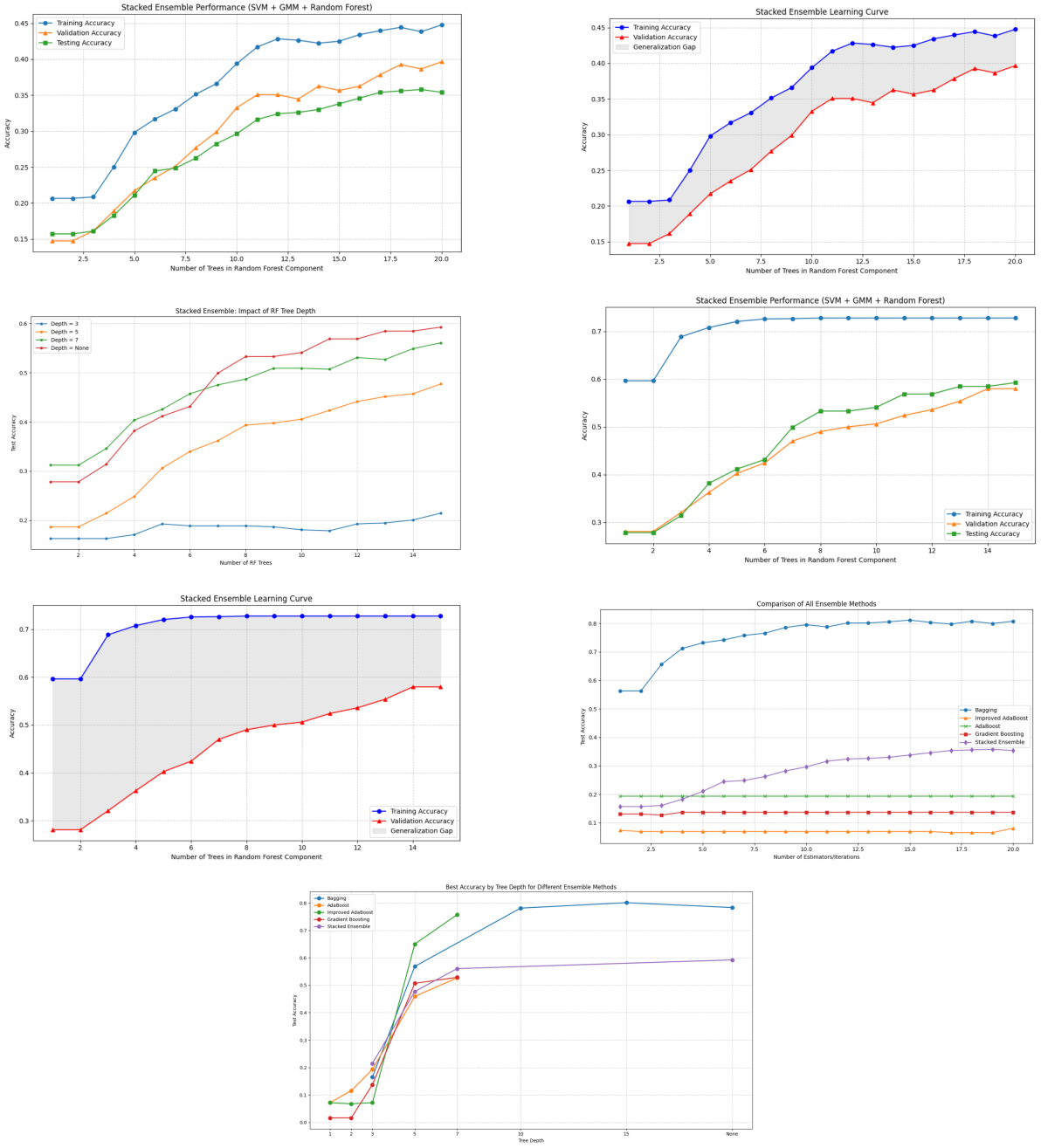


Figure 31: Stacked ensemble with SVM, GMM, and Random Forest

Performance Summary (Decision Tree + Stacking Ensemble):

- **Limited Tree Depth (Depth = 4, 3 GMM components, 20 RF estimators):**
  - **Training Accuracy:** 44.75%
  - **Validation Accuracy:** 38.64%
  - **Test Accuracy:** 35.39%
- **Unlimited Tree Depth:**
  - **Test Accuracy:** 59.24%
  - Performance improves significantly with **increased model complexity** (unlimited depth).

Key Insights from Stacked Ensemble Performance:

- **Limited Depth (Depth = 4):**
  - Gradual improvement in accuracy with increasing trees.
  - **Generalization Gap (10%)** at 20 estimators indicates faster training than validation/test accuracy improvement.
- **Unlimited Depth:**
  - Training accuracy saturates at 70% with 4 trees.
  - **Validation/Test Accuracy** increases to around 58–59% with 15 trees, showing **mild overfitting**.

Comparison of Ensemble Methods:

- **Bagging:** Achieved 80.12% test accuracy at depth = 15, outperforming all other methods.
- **Improved AdaBoost:** Achieved 75.75% test accuracy.
- **Stacking Ensemble:** Performed at **59% test accuracy**, underperforming compared to **Bagging** and **Improved AdaBoost**, but still competitive with standard AdaBoost and Gradient Boosting (52–59%).

Challenges with Stacked Ensemble:

- **Integration Issues:** The performance gap suggests inefficient integration of heterogeneous models (SVM, GMM, RF).
- **Error Propagation:** Lack of harmonized feature spaces and decision boundaries across layers may result in **error propagation**, limiting effectiveness.

Conclusion:

- **Stacking ensemble has potential**, but struggles in high-dimensional tasks like speaker recognition due to integration challenges.
- **Simpler ensembles like Bagging** often outperform more complex stacking strategies, highlighting the importance of **depth tuning** and **optimized ensemble configurations** for achieving high accuracy.

5.31 GMM Clustering

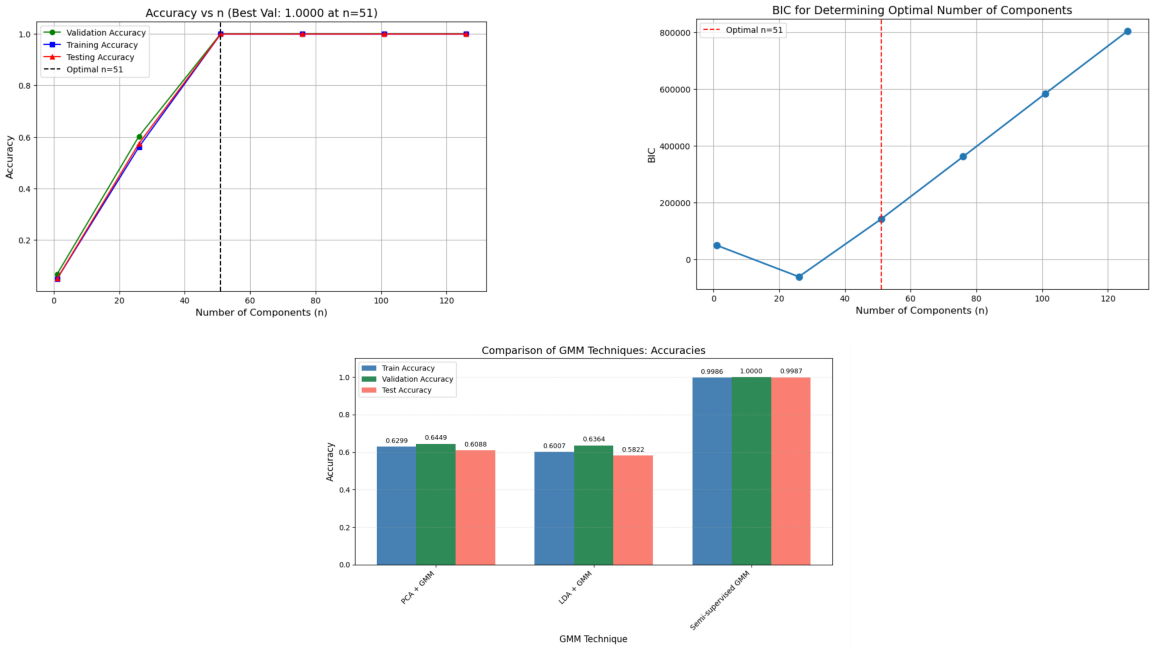


Figure 32: GMM Clustering for speaker separation

Performance Summary (Semi-Supervised GMM):

- **Training Accuracy:** 99.86%
- **Validation Accuracy:** 100.00%
- **Test Accuracy:** 99.67%
- **Optimal Component Count (n=51)**.

Key Insights:

- **Accuracy vs. Number of Components (n):**
  - Accuracy increases linearly from 5% at n=1 to 100% at n=51.
  - Perfect accuracy suggests **n=51** aligns with the number of underlying speaker classes, providing **ideal separation** without redundancy or underfitting.
- **Bayesian Information Criterion (BIC):**
  - BIC reaches its minimum at $\mathbf{n \approx 25}$, indicating a trade-off between model complexity and fit.
  - Despite the lower BIC at $n \approx 25$, $\mathbf{n = 51}$ provides superior classification accuracy, justifying a more complex model.

Comparative Analysis:

- **Semi-Supervised GMM** significantly outperforms **PCA+GMM** (60.88%) and **LDA+GMM** (58.22%), showing better **generalization** and **class alignment**.
- The **semi-supervised approach** uses class-label guidance during model formation, leading to more accurate clustering than purely unsupervised methods.

Advantages of Semi-Supervised GMM:

- Excellent performance with **minimal overfitting**.

- **High-dimensional speaker recognition** is improved by integrating **probabilistic modeling** with **label supervision**.
- Effective at balancing **density estimation** and **class discrimination**, capturing the natural structure and **semantic distinctions** between speakers.

Conclusion:

- The **semi-supervised GMM** method is the **top-performing clustering technique**, offering **near-perfect accuracy** and strong generalization.
- Its integration of supervised learning with unsupervised modeling makes it ideal for complex, high-dimensional tasks like speaker recognition.

6 Comparative Analysis of Various Models

Table 1: Model Accuracy Comparison

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
ANN (With FS)	99.72	80.79	82.42
ANN (With LDA)	99.89	100.00	100.00
ANN (With PCA)	88.75	80.23	72.17
CNN (Raw Features)	96.20	89.80	91.85
CNN (With LDA)	99.88	99.75	99.80
CNN (With PCA)	97.32	54.48	59.84
CNN (With LDA + PCA)	100.00	99.75	99.80

7 About Google Cloud

Google Cloud and Vertex AI Integration

Google Cloud offers a comprehensive suite of tools for machine learning and data-driven projects. For this project, we explored **Vertex AI**, which provides an enterprise-grade environment through **Colab**

Enterprise. This platform enables access to hosted GPU runtimes like the **NVIDIA L4 Tensor Core**, allowing users to create custom runtimes and execute files within the integrated Colab environment.

However, we observed certain limitations:

- The **initial runtime setup** took a considerable amount of time.
- The **difference in compilation time** was negligible compared to standard Google Colab, as our models were not computationally intensive.

Google Cloud and Creating VM for Web Demo Hosting

The **Virtual Machine(VM)** Creation facility on Google Cloud allowed us to deploy the Web Demo of our model Linguistix. We have deployed aur best model ANN with LDA. In the VM we have used **4GB of Memory and 25 GB of Disk Space** for our model. The link of the Web Demo is attached at the end of this Report

8 Summary and Conclusion

This project presents **Linguistix**, a comprehensive exploration of classical machine learning techniques for speaker recognition using MFCC-based audio features. We implemented a diverse set of models—including *K-Nearest Neighbors (KNN)*, *Support Vector Machines (SVM)*, *Decision Trees*, *Naïve Bayes*, and *Multi-Layer Perceptrons (MLP)*, along with clustering techniques like *K-Means* and *Gaussian Mixture Models (GMM)*, including a semi-supervised variant. These models were evaluated under a variety of dimensionality reduction techniques such as *Principal Component Analysis (PCA)*, *Linear Discriminant Analysis (LDA)*, *t-SNE*, and *UMAP*.

Key Findings

- Supervised models with **LDA** consistently achieved accuracy greater than **99%**, with *KNN*, *SVM*, and *Bayesian Learning* showing excellent generalization.
- The **semi-supervised GMM** outperformed all clustering methods, reaching a test accuracy of **99.67%**.
- **Bagging** and **Improved AdaBoost** emerged as the most reliable ensemble methods, effectively reducing overfitting compared to standalone decision trees.
- **MLP and ANN models** combined with **LDA** delivered near-perfect classification, demonstrating the effectiveness of supervised feature transformation.

Observations

- **LDA outperformed PCA** across most models due to its ability to preserve class separability.
- *Decision Trees* and *Boosting methods* tended to overfit unless controlled via feature reduction or tree depth limitations.
- **Stacked ensembles** provided only moderate performance improvements, likely due to integration inefficiencies between base models.
- **Correlation-based feature selection** dramatically improved SVM performance when an optimal number of features was selected.

Conclusion

Classical machine learning methods—especially when integrated with **supervised dimensionality reduction** and **ensemble strategies**—can achieve high-accuracy speaker recognition without reliance on deep learning. In particular, **LDA-enhanced classifiers** and **semi-supervised GMMs** demonstrated robustness, accuracy, and strong generalization. These findings reaffirm the practical value of traditional ML approaches for solving complex audio classification tasks with interpretability and computational efficiency.

A Contribution of each member

Shashank Parchure (B23CM1059)

- **Implemented:** KNN with PCA, Bayesian Learning with Correlation-based Feature Selection, SVM with PCA, MLP with PCA, ANN with LDA.
- **Responsible for:** Report compilation, spotlight video organization (content outline) and exploring and deploying Demo Code on Google Cloud.

Atharva Honparkhe (B23EE1006)

- **Implemented:** Decision Tree with PCA, LDA, and Ensemble Methods, GMM.
- **Responsible for:** Demo Code creation and report compilation.

Vyankatesh Deshpande (B23CS1079)

- **Responsibilities:** Extracted MFCC features of the dataset.
- **Implemented:** KNN with LDA, Bayesian Learning with LDA, SVM with Correlation-based Feature Selection, and ANN with PCA.
- **Handled:** Project page implementation, report compilation and exploration of Google Cloud.

Abhinash Roy (B23CS1003)

- **Implemented:** KMeans Clustering with LDA, Decision Tree on raw data, CNN with PCA and LDA.
- **Responsible for:** Spotlight video creation.

Namya Dhingra (B23CS1040)

- **Implemented:** Decision Tree with UMAP and t-SNE, KMeans Clustering with PCA.
- **Responsible for:** Spotlight video presentation organization and content writing.

Damarasingu Akshaya Sree (B23EE1085)

- **Implemented:** KNN, Bayesian Learning with PCA, SVM with LDA, MLP with Correlation-based Feature Selection, ANN with Correlation-based Feature Selection.
- **Responsible for:** Spotlight video presentation organization and content writing.

References

[1] NumPy Documentation. *NumPy.org*. Retrieved from `https://numpy.org/doc/`

[2] Scikit-learn Documentation. *Scikit-learn.org*. Retrieved from `https://scikit-learn.org/stable/documentation.html`

[3] Matplotlib Documentation. *Matplotlib.org*. Retrieved from `https://matplotlib.org/stable/contents.html`

[4] PyTorch Documentation. *Pytorch.org*. Retrieved from `https://pytorch.org/docs/stable/index.html`

[5] Unicorn Day. (2024). Understanding the Bootstrapping Process in Machine Learning. *Medium.com*. Retrieved from `https://medium.com/@wl8380/understanding-the-bootstrapping-process-in-machine-learning-a6372bf7b4e2`

[6] Shashank Bhatnagar. (2023). Ensemble Methods in Machine Learning. *Medium.com*. Retrieved from `https://medium.com/@shashank25.it/ensemble-methods-in-machine-learning-2d4cc7513c77`

Project Resources and Deliverables

The following links provide access to all project-related artifacts:

- **GitHub Repository:** GitHub Link
- **Spotlight Video:** Spotlight Video
- **Web Demo:** Web Demo (Made using Google Cloud)
- **Project Page:** Project Page