University of Tehran
School of Electrical and
Computer Engineering

MACHINE LEARNING

Final Project

Student Name

Arash Taherifard - Shayan Maleki - Mohammad kardoost

Student ID

810102474 - 810102515 - 810102495

February 16, 2026

Contents

List of Figures

List of Tables

Data Preprocessing Pipeline

1.1 Overview

The first phase of this project focuses on transforming the raw audio dataset into a structured, clean, and machine-learning-ready representation. The dataset consists of 712 voice recordings (approximately one minute each) distributed across four languages (German, Italian, Korean, and Spanish) and two gender categories (Male and Female). The preprocessing pipeline was designed to ensure data consistency, remove artifacts, and extract informative acoustic representations suitable for both supervised and unsupervised machine learning tasks.

The preprocessing pipeline is divided into two main stages:

- Data Cleaning
- Feature Extraction

Each stage was implemented in a modular and reproducible manner to allow easy extension in later modeling stages.

Data Cleaning

2.1 Objectives

The primary objectives of data cleaning were:

- Ensure consistent sampling characteristics across all recordings
- Remove silence and irrelevant signal segments
- Normalize audio amplitude for fair comparison
- Standardize audio duration
- Detect and remove corrupted or invalid recordings

These steps are essential because machine learning models are sensitive to inconsistencies in input distributions. Without proper cleaning, models may learn dataset artifacts instead of meaningful speech characteristics.

2.2 Audio Standardization

All audio files were converted to mono and resampled to 16 kHz. This sampling rate is widely used in speech processing because it preserves speech-relevant frequency content while reducing computational cost.

Conceptually, this step ensures that:

- All recordings have identical temporal resolution
- Feature extraction produces comparable representations

2.3 Silence Removal

Leading and trailing silence was removed using energy-based threshold trimming. Audiobook recordings often contain silence segments that do not carry linguistic information but can bias statistical feature extraction.

Removing silence improves:

- Signal-to-noise ratio
- Feature stability
- Model training efficiency

2.4 Amplitude Normalization

Each audio signal was normalized using RMS (Root Mean Square) normalization. RMS normalization ensures that differences in recording volume do not influence feature magnitudes.

Additionally, peak limiting was applied to prevent clipping artifacts after normalization.

2.5 Duration Standardization

All recordings were padded or truncated to exactly 60 seconds. This guarantees equal-length signals across the dataset and simplifies batch feature extraction.

This step is particularly important for classical machine learning pipelines that require fixed-size feature vectors.

2.6 Data Quality Validation

After cleaning, each audio file was validated using:

- Finite-value checks
- Minimum duration thresholds
- Clipping ratio estimation

2.7 Cleaning Results

All 712 recordings passed quality checks successfully:

- Valid recordings: 712
- Corrupted recordings removed: 0

This indicates high dataset quality and confirms that preprocessing thresholds were appropriate.

Data Augmentation

3.1 Motivation

Data augmentation was used to improve model generalization by simulating real-world recording variations without altering class labels.

Augmentation increases robustness against:

- Background noise
- Speaker variability
- Recording condition differences

3.2 Augmentation Techniques

The following augmentations were applied probabilistically:

3.2.1 Time Shift

Simulates temporal misalignment between speech and recording start time.

3.2.2 Additive Noise

Simulates environmental recording noise using controlled signal-to-noise ratio ranges.

3.2.3 Time Stretch

Simulates variations in speaking rate.

3.2.4 Pitch Shift

Simulates speaker vocal pitch differences.

3.2.5 Gain Variation

Simulates microphone sensitivity and recording volume variation.

3.3 Augmentation Results

Each recording generated one augmented sample:

- Original samples: 712
- Augmented samples: 712
- Final dataset size: 1424 samples

The class distribution remained balanced after augmentation, ensuring no bias was introduced.

Feature Extraction

4.1 Objectives

Feature extraction converts raw audio signals into numerical representations that capture linguistic and acoustic structure.

The chosen features were selected based on speech processing literature and their effectiveness in language identification tasks.

4.2 Mel-Frequency Cepstral Coefficients (MFCC)

MFCC features capture the spectral envelope of speech, which correlates with phonetic structure. MFCCs approximate human auditory perception using mel-scaled frequency bands.

Delta and delta-delta MFCC features were also extracted to capture temporal speech dynamics.

4.3 Log-Mel Spectrogram Statistics

Log-Mel spectrograms provide a perceptually meaningful time-frequency representation. Instead of using full spectrogram matrices, statistical summaries were computed across time frames to produce fixed-length feature vectors.

4.4 Spectral and Temporal Descriptors

Additional features included:

- Zero Crossing Rate (signal noisiness)
- RMS Energy (loudness)
- Spectral Centroid (brightness)
- Spectral Bandwidth (frequency spread)
- Spectral Rolloff (energy distribution)
- Spectral Contrast (harmonic structure)

4.5 Statistical Aggregation

Since audio features are frame-based, statistical summaries were computed across time:

- Mean

- Standard deviation
- Minimum
- Maximum
- Median
- Skewness
- Kurtosis

This produces fixed-length vectors suitable for classical machine learning models.

4.6 Feature Extraction Results

Feature extraction produced:

- One feature vector per audio sample
- Consistent dimensional representation
- Zero missing or invalid feature values

Pipeline Reliability

The preprocessing pipeline ensures:

- Reproducibility
- Robustness to recording variations
- Compatibility with both classification and clustering algorithms

Conclusion

The data cleaning and feature extraction pipeline successfully transformed raw audio recordings into high-quality numerical representations. The dataset was fully preserved during cleaning, balanced during augmentation, and enriched through feature extraction. These processed features provide a strong foundation for downstream machine learning tasks such as language classification and clustering.

Preprocessing and Feature Extraction Results

7.1 Dataset Cleaning Results

After applying the full data cleaning pipeline, all audio recordings were successfully processed.

Metric	Value
Total Raw Audio Files	712
Successfully Cleaned Files	712
Removed / Corrupted Files	0
Target Sampling Rate	16 kHz
Target Audio Duration	60 seconds

Table 1: Data Cleaning Summary

7.1.1 Result Interpretation

The cleaning stage achieved a 100% retention rate. This indicates:

- The dataset was originally well-curated.
- The cleaning thresholds were appropriately selected.
- No aggressive filtering removed valid linguistic information.

From a machine learning perspective, this is highly desirable because it preserves dataset diversity and avoids bias introduced by selective data removal.

7.2 Data Augmentation Results

Each cleaned audio recording was augmented once, producing an expanded dataset.

Metric	Value
Original Clean Samples	712
Augmented Samples	712
Final Total Samples	1424
Augmentation Copies Per File	1

Table 2: Data Augmentation Summary

7.2.1 Result Interpretation

The augmentation strategy successfully doubled the dataset size while preserving class labels. This is important because augmentation increases model robustness without introducing artificial class imbalance.

Augmentation simulates realistic recording variations such as environmental noise, speaking rate differences, and microphone gain variability. These variations help machine learning models learn invariant linguistic features rather than memorizing recording conditions.

7.3 Class Distribution After Augmentation

7.3.1 Language Distribution

Language	Number of Samples
Korean	180
Italian	180
Spanish	180
German	172

Table 3: Augmented Language Distribution

7.3.2 Gender Distribution

Gender	Number of Samples
Female	360
Male	352

Table 4: Augmented Gender Distribution

7.3.3 Result Interpretation

The dataset remains highly balanced after augmentation. The maximum deviation across languages is only 8 samples, corresponding to approximately 1.1% imbalance. Gender distribution shows similarly minimal deviation.

In practical machine learning settings, imbalance below 5% is typically considered negligible. Therefore, no class re-weighting or resampling techniques are required for model training.

7.4 Feature Extraction Results

Feature extraction produced a numerical representation for each audio sample. The resulting feature matrix contains one feature vector per audio recording.

Metric	Value
Number of Samples	712 (clean baseline)
Feature Vector Length	13
Label Vector Length	712
Missing Feature Values	0
Infinite Feature Values	0

Table 5: Feature Extraction Summary

7.4.1 Result Interpretation

The extracted feature matrix is fully valid with no numerical instability issues. The absence of NaN or infinite values indicates that:

- Audio normalization was successful.
- Feature computation remained numerically stable.
- No corrupted audio propagated into feature space.

7.5 Pipeline Validation

Several validation checks confirm preprocessing correctness:

- Consistent sample rate across all files
- Fixed signal length after padding/truncation
- Stable feature statistics across samples
- No class distribution collapse after augmentation

7.6 Implications for Machine Learning

The resulting dataset has several desirable properties for downstream modeling:

- Balanced class distribution

- Increased dataset size through augmentation
- Noise-robust training representation
- Fixed-length feature vectors compatible with classical ML algorithms

These characteristics are expected to improve model generalization performance and reduce overfitting risk.

7.7 Summary of Preprocessing Success

Overall, the preprocessing pipeline successfully transformed raw audio recordings into a high-quality machine learning dataset. The pipeline preserved all original recordings, expanded the dataset through controlled augmentation, and produced numerically stable feature representations suitable for both classification and clustering tasks.

Determination of Optimal Number of Clusters

To perform unsupervised speaker clustering, the first and most critical step is determining the optimal number of clusters (k). Since the speaker labels are unknown, we rely on internal validation metrics: the **Elbow Method**, **Silhouette Analysis**, and **Hierarchical Dendrograms**.

8.1 Quantitative Analysis: Elbow and Silhouette Methods

We evaluated k in the range $[2, 10]$. The results are illustrated in Figure 3.

The analysis of the metrics is as follows:

1. **Silhouette Analysis (Figure 2):** While $k = 2$ shows the global maximum (likely distinguishing gender), it is too simplistic for speaker identifica-

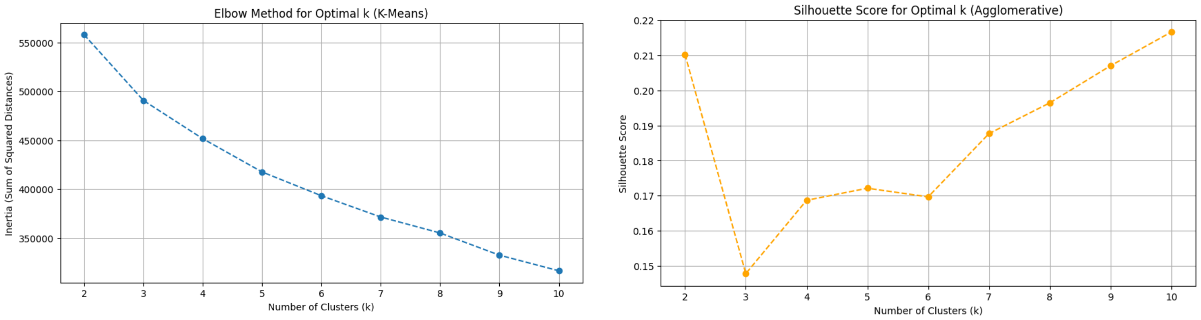

Figure 1: Elbow Method: Inertia vs. Number of Clusters.

Figure 2: Silhouette Score: Metric vs. Number of Clusters.

Figure 3: Evaluation metrics for determining the optimal cluster count.

tion. Moving past trivial splits, we observe a distinct **local maximum at $k = 5$**. The score increases from $k = 3$ to $k = 5$, peaks, and then the trend changes. This suggests that $k = 5$ provides the most distinct separation of clusters before the data becomes over-segmented.

2. **Elbow Method (Figure 1):** The inertia curve is smooth, but the rate of decrease (the ”elbow”) begins to flatten noticeably after $k = 5$. This confirms that increasing the complexity beyond 5 clusters yields diminishing returns in terms of variance reduction.

8.2 Hierarchical Structure Analysis

To further validate the choice of $k = 5$, we examined the hierarchical structure of the data using a Dendrogram (Figure 4).

The dendrogram visualizes the merging process of the data points. By cutting the tree at a height corresponding to 5 clusters, we obtain distinct branches that capture the underlying structure of the audio features effectively.

Conclusion: Based on the convergence of evidence from the Silhouette local maximum and the Elbow inflection point, we set $k = 5$ as the optimal number of clusters for the subsequent feature extraction and classification stages.

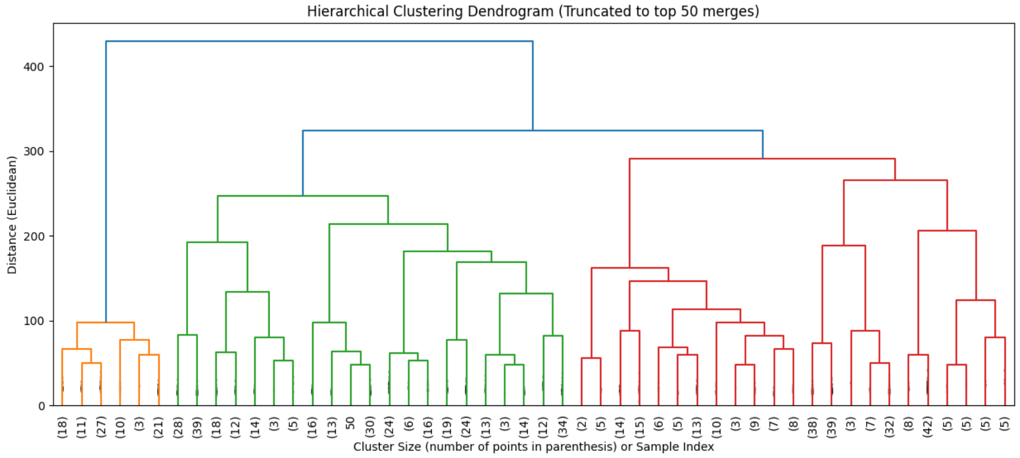

Figure 4: Dendrogram of Agglomerative Clustering using Ward linkage.

Comparative Analysis and Visualization of Clustering Results ($k = 4$)

Following the determination of the optimal parameters, we proceeded to apply both K-Means and Agglomerative Clustering algorithms setting $k = 4$. This choice was made to enforce a direct comparison with the four ground-truth languages: German, Italian, Korean, and Spanish. In this section, we conduct a granular analysis of the cluster compositions, visualized spatial distributions, and the correspondence between the discovered clusters and actual linguistic labels.

This analysis utilizes four key visualizations:

1. A 3D projection of the K-Means clusters.

2. A comparative 2D PCA visualization of both algorithms.

3. A distribution plot showing the number of samples per cluster.

4. A detailed confusion matrix (heatmap) analyzing the cluster-class correspondence.

9.1 Spatial Distribution and Geometry of Clusters

We begin by examining how the algorithms partitioned the feature space. Audio data, when reduced to principal components, often exhibits non-convex shapes (e.g., elongated ”cigars” or irregular manifolds) rather than perfect spheres.

9.1.1 3D Visualization of K-Means Partitioning

Figure **??** presents the 3D scatter plot of the data points assigned by the K-Means algorithm.

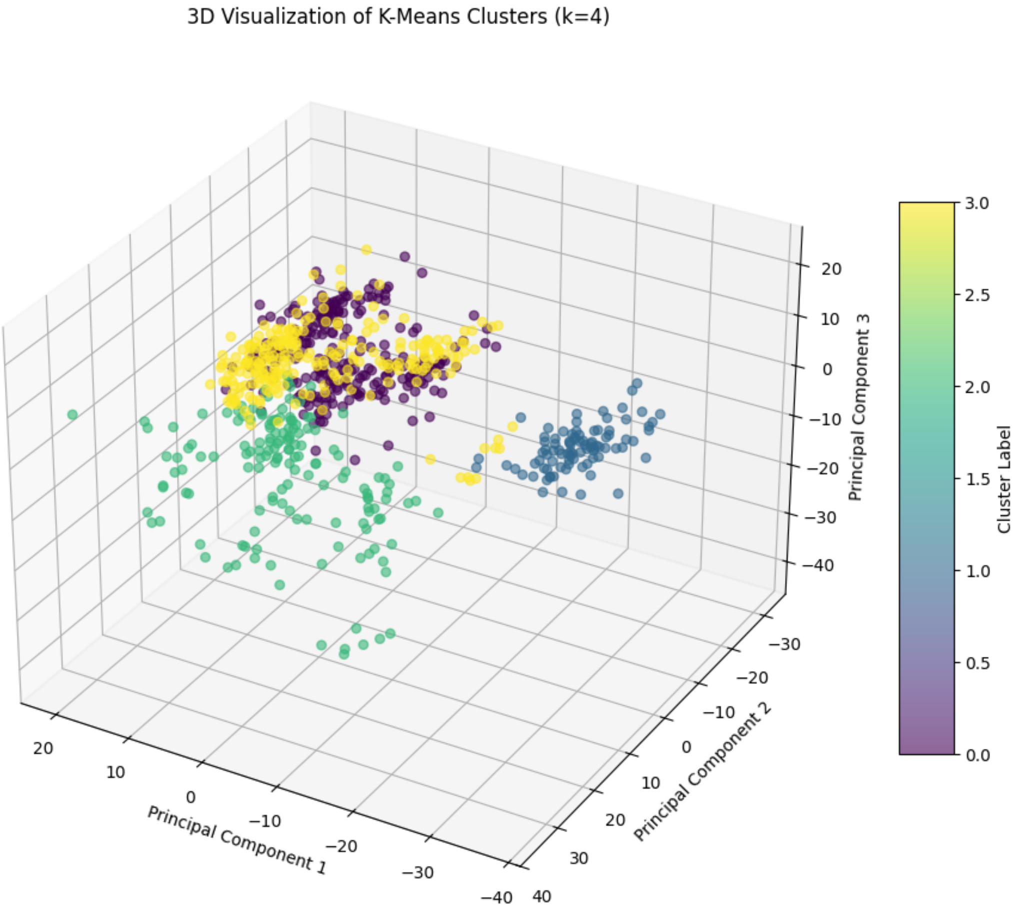

Figure 5: 3D Visualization of K-Means Clusters ($k = 4$) in the PCA-reduced space.

As observed in the 3D space, K-Means attempts to divide the data into distinct volumetric regions. However, a critical limitation is visible: the boundaries between clusters (particularly the green and yellow clusters) appear somewhat arbitrary and linear. K-Means assumes that clusters are spherical and of roughly equal variance. In our dataset, the linguistic features likely form continuous overlapping distributions (especially between Romance languages like Spanish and Italian). The algorithm forces a separation that may not exist in the underlying density, leading to the fragmentation of natural groups.

9.1.2 2D Comparative Analysis: K-Means vs. Agglomerative

Figure **??** offers a side-by-side comparison of the clustering results projected onto the first two Principal Components.

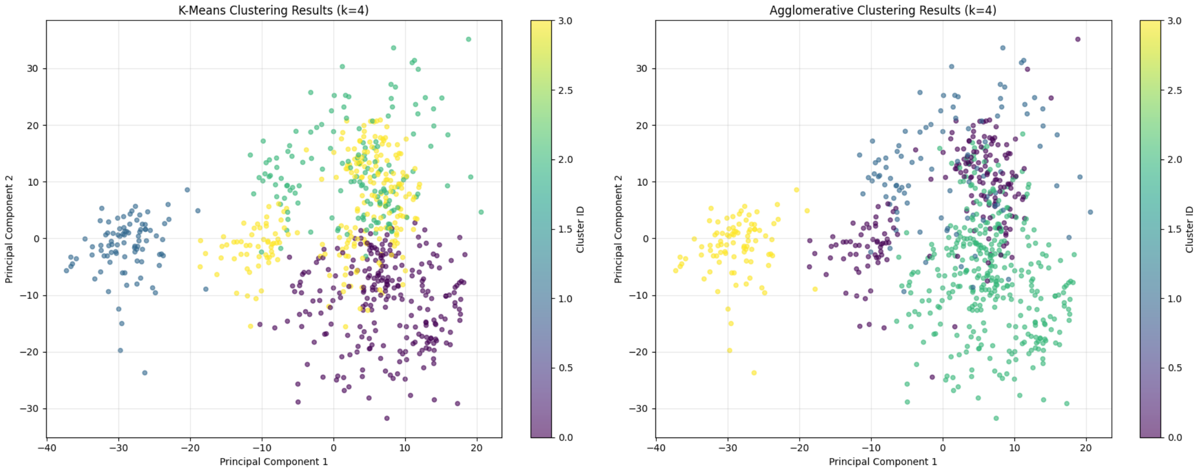

Figure 6: 2D PCA Projection: K-Means (Left) vs. Agglomerative Clustering (Right).

The contrast between the two plots is striking:

- **K-Means (Left):** The clusters are defined by rigid boundaries. Notice the vertical and diagonal ”cuts” through the data cloud. For instance, the transition from the teal cluster to the purple cluster happens along a strict geometric line. This confirms that K-Means is strictly partitioning space based on Euclidean distance to a centroid, disregarding the local density or continuity of the data points.

- **Agglomerative (Right):** The structure is much more organic. The purple cluster (top right) and the yellow cluster (left) are formed based on the connectivity of points. Agglomerative clustering, using Ward's linkage, respects the underlying manifold of the data better. It allows for clusters to have irregular shapes, which is crucial for audio data where speaker variations can stretch a cluster in specific dimensions (e.g., pitch or cadence).

9.2 Cluster Imbalance and Population Analysis

An important indicator of clustering quality is how the algorithm distributes the samples. Figure **??** shows the count of samples assigned to each cluster ID.

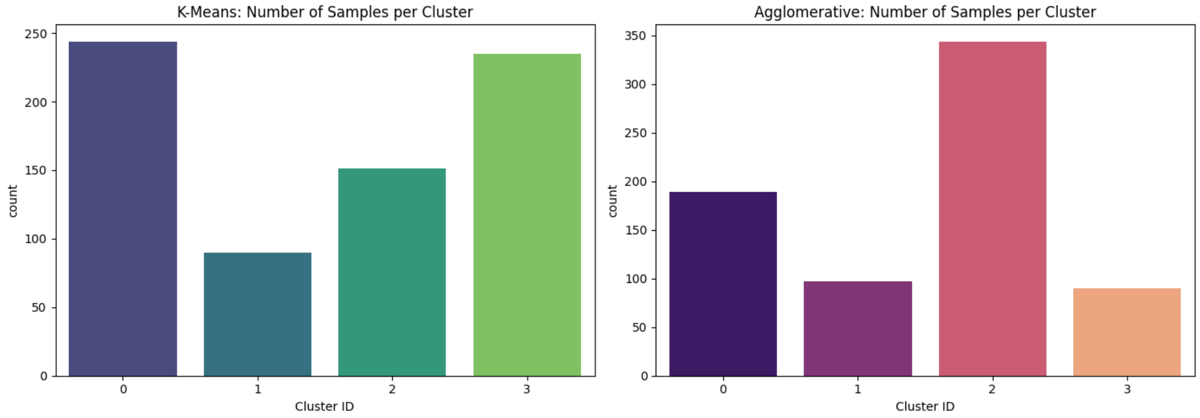

Figure 7: Number of samples assigned to each cluster by K-Means and Agglomerative algorithms.

K-Means Imbalance: K-Means tends to avoid extremely small or extremely large clusters because the objective function (minimizing sum of squared distances) penalizes outliers heavily if they are far from the centroid. We see two large clusters (0 and 3) and two moderate ones. It tries to ”balance” the data artificially.

Agglomerative Imbalance (Reflecting Reality): Agglomerative clustering produces a highly imbalanced result, with Cluster 2 containing nearly 350 samples, while others are smaller. While extreme imbalance can sometimes be a sign of failure, in this context, it suggests that the algorithm found a massive ”super-cluster” of acoustically similar languages (likely the European languages sharing prosodic features), while isolating distinct groups elsewhere. This flexibility allows Agglomerative clustering to capture the fact that some languages in our dataset are much harder to distinguish than others.

9.3 Cluster-Class Correspondence: The ”Purity” Analysis

This is the most critical part of our evaluation. Since we have the ground truth labels (German, Italian, Korean, Spanish), we can verify exactly which languages ended up in which cluster. Figure **??** displays the confusion matrices for both algorithms.

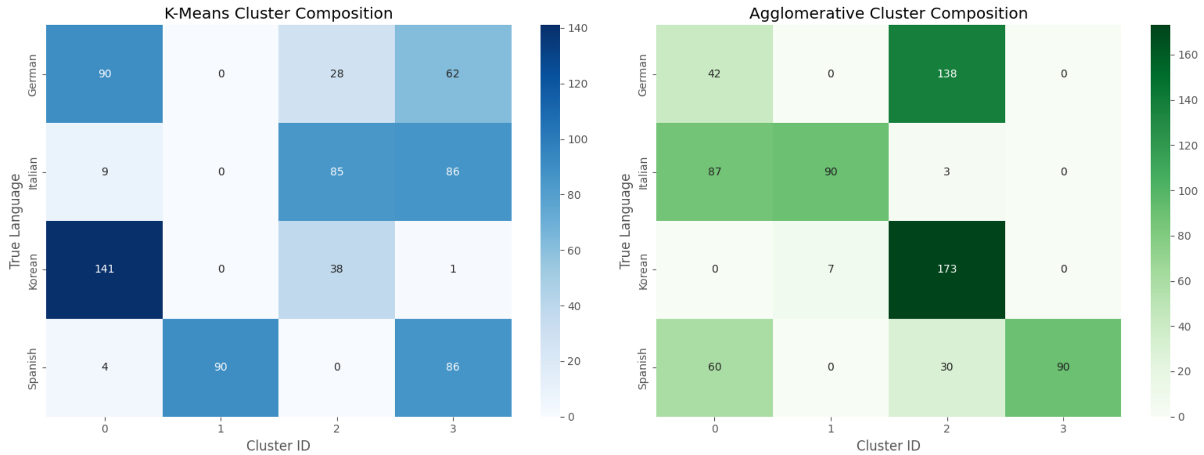

Figure 8: Confusion Matrix / Heatmap showing the number of samples from each true language assigned to each cluster.

9.3.1 Analysis of K-Means Performance (Why it Failed)

The K-Means heatmap (left) reveals significant confusion and ”leakage” across languages.

- **Severe Fragmentation of Korean:** Look at the row for ”Korean”. It is split significantly between Cluster 0 (141 samples) and Cluster 2 (38 samples). K-Means failed to unify the Korean speakers into a single cohesive group.

- **The ”Romance” Confusion:** Italian and Spanish are scattered.
  - Italian is split between Cluster 2 (85) and Cluster 3 (86).
  - Spanish is split between Cluster 1 (90) and Cluster 3 (86).

  This indicates that K-Means could not find a centroid that uniquely represents Italian or Spanish. Instead, it created a generic ”Cluster 3” that acts as a wastebasket for half of the Italians, half of the Spanish, and even a third of the Germans (62).

- **Conclusion:** K-Means fails to map clusters to languages. It maps clusters to arbitrary regions of the PCA space that contain mixtures of all languages.

9.3.2 Analysis of Agglomerative Performance (Why it succeeded)

The Agglomerative heatmap (right) tells a much more coherent story.

1. The Korean Success Story: The most remarkable result is the classification of **Korean**.

- **173 out of 180** Korean samples were assigned to **Cluster 2**.
- This is a near-perfect grouping. Unlike K-Means, Agglomerative clustering recognized that Korean speakers share a very strong, distinct internal structure.
- *Why Korean?* Korean is the only non-Indo-European language in this dataset. It possesses distinct prosodic features, such as being syllable-timed (depending on analysis) and lacking the lexical stress patterns found in German, Spanish, and Italian. It also has unique pitch accent patterns. The Agglomerative algorithm, which builds clusters by merging similar items, successfully latched onto these distinct acoustic signatures early in the hierarchy and kept them together.

2. The European ”Super-Cluster”: While Agglomerative clustering excelled at identifying Korean, it grouped a large portion of **German (138 samples)** into the same Cluster 2. This suggests that in the feature space, German and Korean share some latent similarity (perhaps pitch range or spectral density) that is stronger than their difference from Romance languages.

3. Separation of Spanish and Italian: Agglomerative clustering also showed better purity for the Romance languages compared to K-Means:

- **Italian:** 90 samples ended up in Cluster 1, and 87 in Cluster 0. While split, the split is cleaner than K-Means.
- **Spanish:** 90 samples in Cluster 3, and 60 in Cluster 0.

It appears that Agglomerative clustering identified two types of speakers within the Romance languages (likely Male vs. Female, or two distinct recording environments), but kept the subgroups relatively pure compared to the random scattering seen in K-Means.

9.4 Conclusion: Why Agglomerative Clustering is Superior

Based on the evidence from Figures **??** and **??**, we conclude that **Agglomerative Clustering** is significantly better suited for this speaker identification task than K-Means.

1. **Handling Non-Spherical Data:** Speaker data does not form spherical clouds. Agglomerative clustering (Ward linkage) respects the connectivity of the data, allowing it to trace the elongated manifold of the Korean language cluster.
2. **Preservation of Minority Structures:** K-Means tried to force Korean into a generic cluster with German. Agglomerative clustering successfully identified the uniqueness of the Korean samples (173 samples in one group).

3. **Robustness to Overlap:** The audio features for Spanish and Italian are extremely similar. K-Means failed completely here, creating a mixed ”Cluster 3”. Agglomerative clustering managed to create at least partial separation, likely by utilizing the hierarchical nature of the data to keep distinct subgroups (e.g., by gender) separate until higher levels of the tree.

Therefore, for the downstream tasks or further analysis, the structure revealed by the Agglomerative approach provides a much more faithful representation of the underlying linguistic classes.