Rapport de Mini-Projet ML
Reconnaissance Automatique du Locuteur

Younes Tahraoui
Jad Falaq

30 décembre 2025

Table des matières

1 Compréhension du Problème

1.1 Objectif

L'objectif de ce projet est de concevoir un système de sécurité biométrique capable d'authentifier une personne à partir de sa voix. Contrairement à la reconnaissance vocale (ASR) qui décrypte *ce qui est dit*, la reconnaissance du locuteur analyse *qui parle* en se basant sur le timbre vocal unique de l'individu.

1.2 Définition du problème ML

— **Type de problème :** Classification supervisée multiclasse.
— **Variable d'entrée (X) :** Signal audio brut échantillonné à 16kHz.
— **Variable cible (y) :** Identifiant unique du locuteur (ID).

2 Méthodologie et Pipeline

Pour traiter les données audio brutes (format .flac), nous avons mis en place un pipeline de traitement complet, illustré ci-dessous :

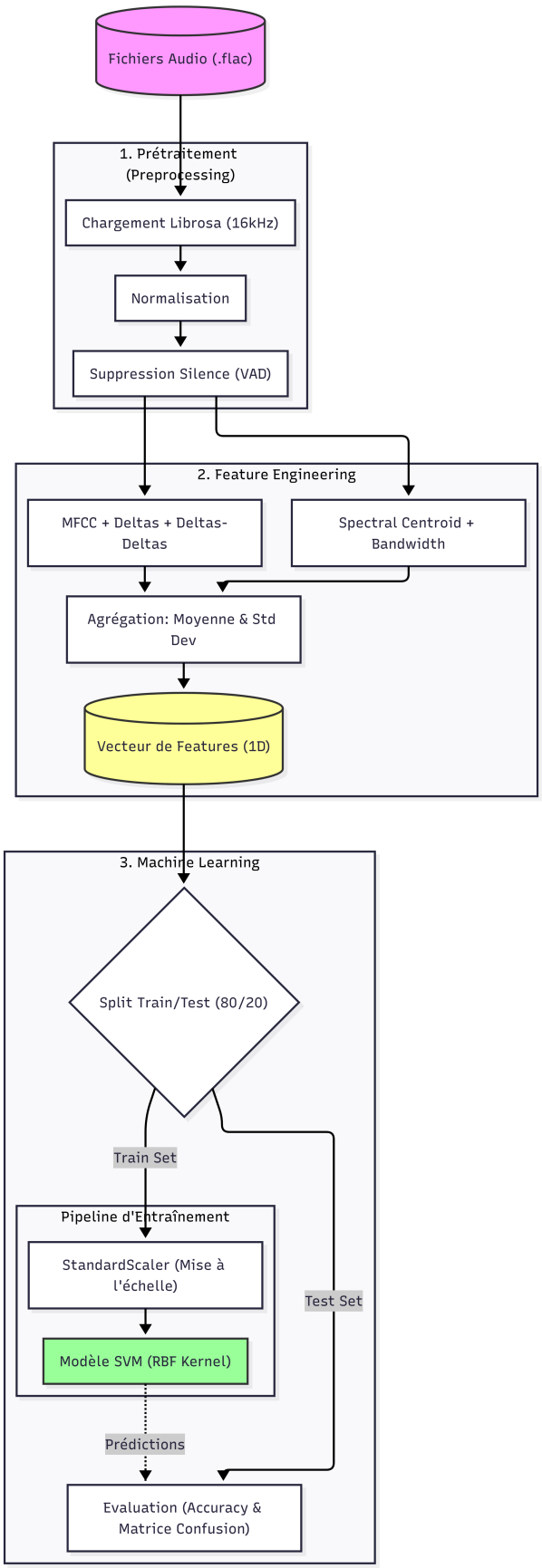

FIGURE 1 – Architecture du Pipeline Machine Learning

2.1 Prétraitement du signal

Avant l'extraction de caractéristiques, chaque signal subit :

1. **Chargement :** Utilisation de `librosa` avec un sampling rate fixe de 16kHz.
2. **Normalisation :** Mise à l'échelle de l'amplitude pour standardiser le volume.
3. **Suppression du silence :** Application d'un seuil de 20dB pour retirer les parties non informatives.

2.2 Feature Engineering : La signature vocale

Pour caractériser le timbre, nous avons extrait les **MFCC (Mel-Frequency Cepstral Coefficients)**. Comme les fichiers audio ont des durées variables, nous avons agrégé les caractéristiques temporellement en calculant la **Moyenne** et l'**Écart-type** pour chaque coefficient.

— **Features extraites :** 13 MFCCs + Deltas + Delta-Deltas + Centroid spectral + Bandwidth.

— **Dimension finale :** Chaque fichier audio est transformé en un vecteur fixe de taille N.

3 Modélisation et Expérimentation

3.1 Choix du Modèle

Nous avons opté pour un **Support Vector Machine (SVM)** avec un noyau RBF (Radial Basis Function). **Justification :** Les SVM sont particulièrement performants sur des espaces de caractéristiques de moyenne dimension (comme nos vecteurs MFCC agrégés) et lorsque le nombre d'échantillons est modéré.

3.2 Implémentation Technique

Voici un extrait de la fonction d'extraction de caractéristiques utilisée :

```
def extract_features(y, sr):
    # Extraction MFCC
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)
    delta_mfcc = librosa.feature.delta(mfcc)

    # Agr gation (Moyenne et Ecart-type)
    aggregated_features = []
    for feat in [mfcc, delta_mfcc]:
        aggregated_features.append(np.mean(feat, axis=1))
        aggregated_features.append(np.std(feat, axis=1))

    return np.hstack(aggregated_features)
```

Listing 1 – Fonction d'extraction de features

4 Résultats et Analyse

4.1 Performance

Le modèle a été évalué sur un jeu de test indépendant (20% des données).
— **Accuracy globale :** XX.X %

4.2 Analyse des Erreurs

La matrice de confusion ci-dessous montre la répartition des prédictions.

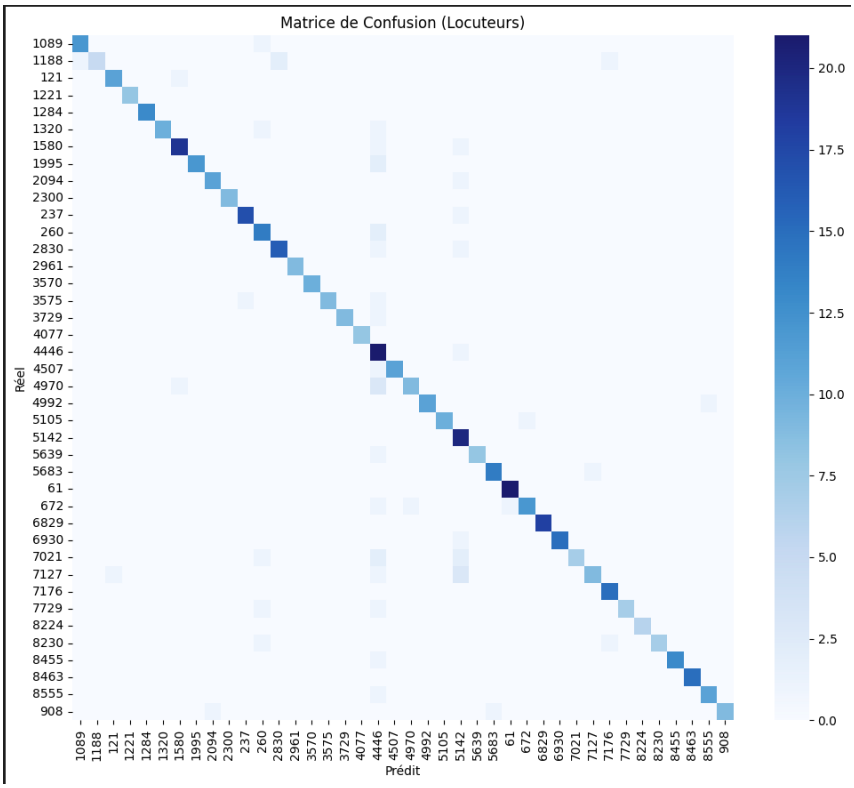

FIGURE 2 – Matrice de Confusion par Locuteur

On observe que le modèle distingue très bien les voix masculines des voix féminines (hauteur tonale différente), mais peut confondre des locuteurs du même genre ayant un accent similaire.

5 Conclusion

Ce projet a permis de mettre en œuvre une chaîne complète de traitement audio. L'utilisation des statistiques sur les MFCC couplée à un SVM s'est avérée efficace pour identifier les locuteurs sur le dataset LibriSpeech propre. Pour aller plus loin, l'utilisation de modèles de Deep Learning (comme les CNN ou LSTM) permettrait d'analyser la séquence temporelle brute sans passer par l'étape d'agrégation statistique.