République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tunis El Manar
École Nationale d'Ingénieurs de Tunis

Département Technologies de l'Information et de la Communication

Projet de Fin d'Année 1

Identification automatique de la maladie de Parkinson à partir de la voix

Elaboré par :

Belsem JEMAI

Mouna REZGUI

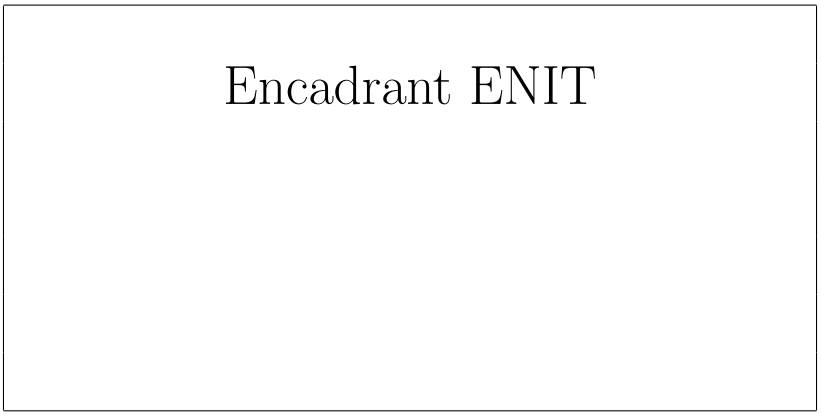
Classe : 1ère Année télécommunications

Encadré par :

Mme Sonia DJAZIRI-LARBI

Année Universitaire : 2024/2025

Signature



Remerciements

Nous tenons à exprimer notre profonde gratitude à tous ceux qui ont contribué, directement ou indirectement, à la réalisation de ce projet.
Nous souhaitons exprimer notre reconnaissance tout particulière à notre superviseuse Madame Sonia Djaziri-Larbi, pour ses précieux conseils, son soutien indéfectible et sa présence constante tout au long de cette mission. Son approche méthodique et ses conseils avisés ont grandement enrichi notre réflexion.
Nous souhaitons également exprimer notre gratitude envers l'ensemble du corps professoral du département des télécommunications de l'ENIT pour la qualité exceptionnelle de leur enseignement et les savoirs acquis qui ont grandement simplifié la mise en œuvre de ce projet.
Au final, nous souhaitons exprimer notre gratitude à nos amis et nos proches pour leur soutien moral et leurs encouragements tout au long de ce voyage.

Résumé

Ce projet vise à analyser les propriétés acoustiques de la voix dans le but d'identifier des biomarqueurs pertinents pour la détection de la maladie de Parkinson. Au cours de ce travail, nous avons extrait un grand ensemble de descripteurs acoustiques à partir d'une base de données d'enregistrements vocaux de 50 sujets témoins (HC) et de 50 sujets parkinsoniens (PD). Nous avons mené des études descriptives et statistiques de ces descripteurs, appliqué différentes techniques de sélection de descripteurs et évalué leur efficacité avec différents algorithmes d'apprentissage automatique. Le premier but est de tirer un vaste ensemble de descripteurs acoustiques à partir d'enregistrements de voix et d'examiner l'influence de diverses méthodes de sélection de caractéristiques sur la justesse des modèles de classification. Ce projet s'inscrit donc dans le domaine de l'intelligence artificielle appliquée au secteur de la santé, en exploitant l'apprentissage automatique afin d'optimiser la classification des patients.

Abstract

This project investigates the feasibility of using acoustic voice analysis for the diagnosis of Parkinson's Disease. In order to do this, a database has been prepared that consists of 50 control subjects with no health problems and 50 people identified as patients with Parkinson's disease. The next step consisted of extracting large numbers of acoustic features from these voice recordings. Descriptive statistical analyses were performed, followed by application of feature selection techniques to find the most informative descriptors, before determining the effectiveness of different machine learning algorithms for classifying the subjects based on the improvement in classifying their voices from the comprehensive set of extracted features. This work then falls into the area of artificial intelligence as applied to medicine with the aim of improving early detection of Parkinson's disease using noninvasive methods.

Table des matières

Liste des figures **6**

Liste des tables **7**

Liste des acronymes **8**

Introduction générale **9**

1 La voix comme biomarqueur de la maladie de Parkinson **11**

1.1 La voix humaine : production et caractéristiques générales 11

1.1.1 Physiologie de la production vocale 11

1.1.2 Caractéristiques acoustiques de la voix 12

1.2 La maladie de Parkinson 14

1.3 L'impact de maladie de Parkinson sur la voix 15

1.4 Conclusion 15

2 Extraction des descripteurs vocaux **16**

2.1 Les descripteurs acoustiques 16

2.1.1 Les descripteurs de la cavité bucco-nasale 16

2.1.2 Les descripteurs des cordes vocales 17

2.2 Outils et méthodes d'extraction 18

2.2.1 Environnement de développement 18

2.2.2 Bibliothèques utilisées pour l'extraction des caractéristiques vocales 18

2.3 Conclusion 19

3 Détection automatique de la maladie de Parkinson par l'analyse de la voix 20

3.1 L'apprentissage automatique 20

3.1.1 Définition de l'apprentissage automatique 21

3.1.2 Machine à vecteurs de support : SVM 23

3.1.3 Cross-validation 24

3.2 Bases de données 25

3.2.1 Description de la base de données audio 25

3.2.2 Base de données des descripteurs 25

3.3 Analyse statistique et classification des données 26

3.4 Conclusion 29

Conclusion générale 29

Bibliographie 30

A Script Python 34

Liste des figures

1.1 Système phonatoire [3] 12

1.2 Shimmer et Jitter [6] 13

1.3 Exemple d'audiogramme [3] 14

3.1 L'apprentissage supervisé [17] 21

3.2 L'apprentissage non supervisé [17] 22

3.3 L'apprentissage par renforcement [18] 22

3.4 SVM 23

3.5 Cross-validation [20] 24

3.6 Matrice de confusion 27

3.7 Les boxplots des descripteurs acoustiques 28

Liste des tableaux

3.1 les précisions moyennes de différents noyaux de la cross-validation 27

Liste des acronymes

PD Patient parkinsonien (Parkinson Disease)

HC sujet sain (Healthy control)

MP Maladie de Parkinson

MFCC Mel-Frequency Cepstral Coefficients

SVM Support Vector Machine

SVC Support Vector Classifier

Introduction générale

La maladie de Parkinson est un trouble neurodégénératif qui touche essentiellement la maîtrise musculaire, ce qui provoque des tremblements, une raideur musculaire et un ralentissement des gestes. En plus de ces troubles moteurs apparents, elle affecte aussi la voix, entraînant une réduction de l'intensité, un ton monotone et des modifications de l'articulation.

Pour améliorer la gestion des patients, il est crucial de détecter la maladie à un stade précoce. Toutefois, les diagnostics conventionnels s'appuient principalement sur l'examen clinique des symptômes moteurs, ce qui restreint leur efficacité aux premières étapes. L'extraction automatique de biomarqueurs via l'analyse vocale permet de détecter des indices subtils qui échappent aux médecins, proposant ainsi une méthode complémentaire pleine de promesses.

Dans le cadre de notre formation en ingénierie des télécommunications à l' Ecole Nationale d'Ingénieurs de Tunis (ENIT), nous avons réalisé un projet de fin d'année qui consiste à extraire des descripteurs vocaux et à effectuer une classification des patients parkinsoniens à l'aide d'une base de données comprenant 100 individus, répartis équitablement entre 50 patients atteints de Parkinson (PD) et 50 sujets sains (HC).

Chapitre 1

La voix comme biomarqueur de la maladie de Parkinson

Dans ce chapitre, on s'intéresse à la définition de la voix humaine en présentant brièvement la physiologie de sa production et ses propriétés acoustiques. On traite aussi la MP en étudiant ses origines, son évolution et les symptômes majeurs qui la définissent, puis on finit par une étude de l'effet de cette pathologie sur la voix.

1.1 La voix humaine : production et caractéristiques générales

La voix est le son produit par les vibrations des cordes vocales sous l'effet de l'air expiré, modulé par les cavités de résonance (pharynx, cavité buccale et nasale). Ce processus implique la coordination de plusieurs systèmes physiologiques, notamment le système respiratoire, le larynx et les articulateurs supraglottiques. La voix joue un rôle fondamental dans la communication humaine et peut être affectée par diverses pathologies, dont la MP [1].

1.1.1 Physiologie de la production vocale

La physiologie de la voix fait référence au mécanisme par lequel les cordes vocales, les poumons et les cavités bucco-nasales s'associent pour générer du son. Quand il est produit,

plusieurs processus physiologiques ont lieu. En effet, l'air est chassé des poumons vers le larynx, les cordes vocales s'écartent et se rejoignent sous l'influence du courant d'air, générant ainsi des vibrations qui sont altérées par le résonateur qui englobe la gorge, la bouche et les cavités bucco-nasales. La langue, les lèvres et les dents modifient le son pour constituer des mots. Ce mécanisme dépend d'une coordination musculaire précise et de la performance optimale des structures concernées [2].

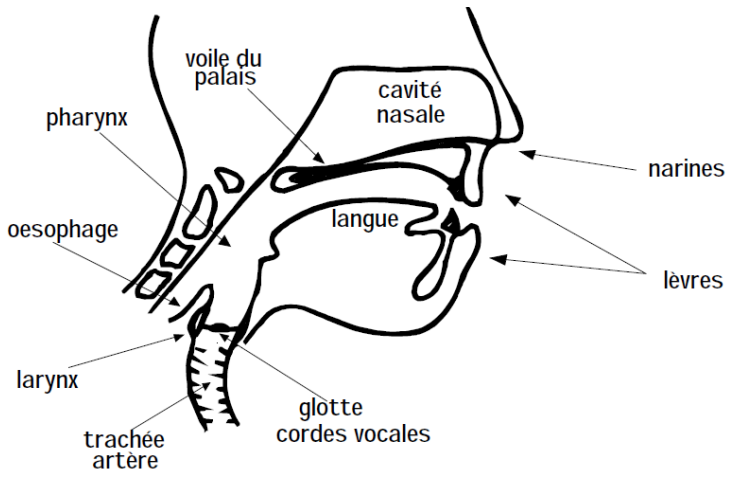


FIGURE 1.1 – Système phonatoire [3]

1.1.2 Caractéristiques acoustiques de la voix

L'analyse acoustique de la voix repose sur des paramètres mesurables qui permettent d'évaluer la qualité vocale et d'identifier d'éventuelles pathologies. Ces paramètres sont principalement issus du traitement du signal et permettent une évaluation objective des variations de fréquence, d'amplitude et de bruit dans la voix.

— **Fréquence Fondamentale F0 :** La fréquence fondamentale correspond à la fréquence de vibration des cordes vocales et est exprimée en Hertz (Hz). Elle détermine la hauteur de la voix : une F0 élevée est associée à une voix aiguë, tandis qu'une F0 faible correspond à une voix grave. Chez l'homme, elle varie en moyenne entre 85 et 180 Hz, tandis que chez la femme, elle est généralement comprise entre 165 et 255 Hz [1].

Lors de la phonation d'un son non voisé, les cordes vocales ne vibrent pas d'une manière périodique donc on peut pas mesurer F0.

— **Jitter :** C'est la moyenne des différences absolues entre les périodes consécutives, divisée par la période moyenne, exprimée en pourcentage.

— **Shimmer :** Le shimmer représente les variations d'amplitude du signal vocal d'une période à l'autre. Comme le jitter, il est un indicateur d'instabilité vocale.

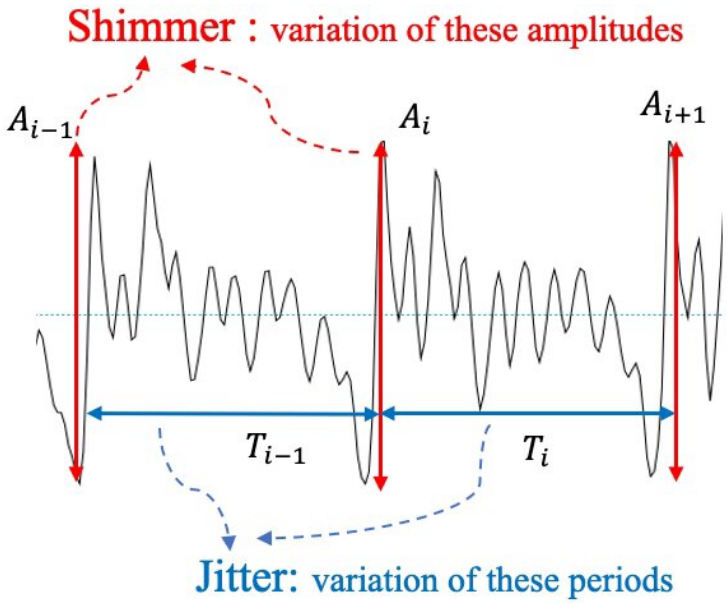


FIGURE 1.2 – Shimmer et Jitter [6]

Où Ti représente les longueurs des périodes glottiques extraites et Ai est l'amplitude extraite.

— **Rapport Harmonique-Bruit HNR :** Le HNR mesure le rapport entre l'énergie harmonique (signal périodique) et le bruit (composante aléatoire) dans la voix [7].

— **Mel-Frequency Cepstral Coefficients (MFCC) :** Les descripteurs extraits de l'analyse cepstrale de la parole permettent de caractériser la position des articulateurs dans la cavité bucco-nasale.

La parole est composée d'une succession de phonèmes voisés, de phonèmes non voisés et de silences, comme le montre la figure 1.3. Les phonèmes voisés sont caractérisés par une structure presque périodique, alors que les phonèmes non voisés ressemblent à du bruit. Dans la langue française, toutes les voyelles ainsi que certaines consonnes (R, L, V) sont voisées. Il est important de préciser que ces quatre paramètres, ainsi que le HNR, ne peuvent être mesurés que sur un son voisé, car leur calcul repose sur la présence de vibrations régulières des cordes vocales.

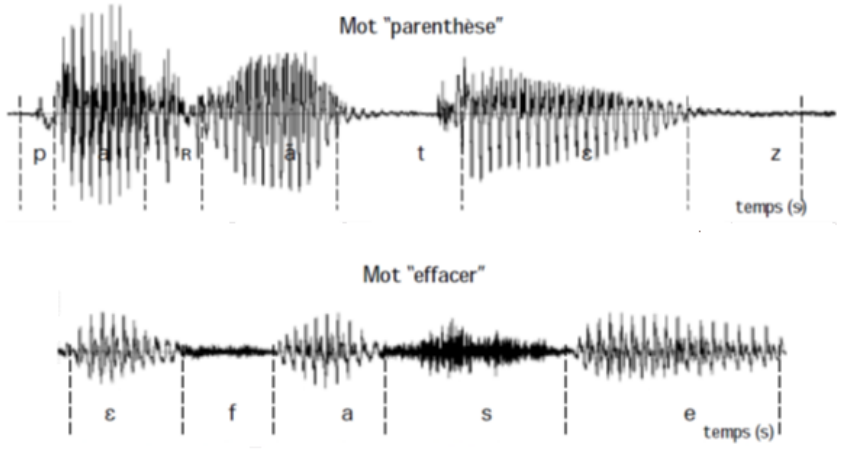


FIGURE 1.3 – Exemple d'audiogramme [3]

1.2 La maladie de Parkinson

La maladie de Parkinson est un trouble neurodégénératif, classé comme la deuxième affection la plus fréquente après la maladie d'Alzheimer, qui se caractérise principalement par une détérioration progressive des capacités motrices. Son incidence s'accroît avec l'âge, atteignant 1 à 2% chez les individus de plus de 65 ans. L'origine de la MP demeure mal comprise, mais il est établi qu'elle entraîne une diminution des neurones dopaminergiques dans la substances noire ce qui réduit la libération de dopamine au niveau de ganglions de la base. Actuellement, le diagnostic s'appuie majoritairement sur une évaluation motrice réalisée par un neurologue. En général, le diagnostic est établi si l'inspecteur remarque au moins deux des trois signes suivants : l'akinésie (la lenteur à commencer les mouvements), la rigidité et les tremblements au repos. Ces symptômes moteurs ne se révèlent hélas qu'après la destruction de 70% des neurones dopaminergiques [8]. Un défi crucial de la recherche est donc d'élaborer des méthodes pour détecter la maladie plus tôt, dans le but de pouvoir ralentir sa progression dès les premiers stades , lorsqu'on disposera de traitements appropriés.

1.3 L'impact de maladie de Parkinson sur la voix

Parmi les manifestations cliniques diverses de la MP, la modification de la voix des malades est un élément d'intérêt à plusieurs égards. Un grand nombre de publications [9] ont été consacrées à l'étude de la voix chez les PD, mettant en évidence des perturbations telles que la dysarthrie hypokinétique (qui correspond à une réduction de l'amplitude des mouvements des muscles responsables de l'articulation). Les divers éléments de la parole affectés par la dysarthrie parkinsonienne sont :

- La prosodie : une diminution des variations d'intensité et de hauteur rend la voix monotone, le rythme est perturbé et on note également des désordres de la fluidité (répétitions, bégaiements).
- L'articulation : la précision dans l'articulation des voyelles et des consonnes est affectée.
- La phonation : la force de la voix se réduit, la tonalité moyenne tend à descendre ou à monter, la fréquence et l'intensité deviennent fluctuantes, et le timbre prend une qualité soufflée ou voilée [10].
- Le rythme : l'aptitude à conserver un rythme verbal stable se dégrade [11].

Certaines perturbations de la voix caractéristiques de la MP sont déjà visibles plusieurs années avant le diagnostic clinique [12, 13].

Dans ce travail, nous nous intéresserons particulièrement aux marqueurs acoustiques les plus discriminants qui pourraient être utilisés pour établir un diagnostic précoce.

1.4 Conclusion

Ce premier chapitre nous a donné l'occasion de présenter les processus de production vocale et ses caractéristiques acoustiques majeurs. Nous avons aussi mis l'accent sur l'effet de la MP sur la voix, en mettant en lumière les modifications qu'elle peut provoquer. On peut analyser les variations vocales chez les patients atteints de Parkinson à l'aide de descripteurs spécifiques qui permettent de mesurer ces modifications.

Chapitre 2

Extraction des descripteurs vocaux

Ce chapitre mettra en lumière les catégories dominantes de descripteurs vocaux, en se focalisant particulièrement sur ceux associées à la cavité bucco-nasale et aux cordes vocales. Nous discuterons par la suite des outils et techniques employés pour leur extraction.

2.1 Les descripteurs acoustiques

L'analyse de la voix repose sur l'extraction de descripteurs vocaux qui permettent de quantifier ses caractéristiques acoustiques et d'identifier d'éventuelles altérations. Ces descripteurs, issus du traitement du signal, jouent un rôle essentiel dans l'évaluation objective des troubles de la voix, notamment chez les patients atteints de la MP.

2.1.1 Les descripteurs de la cavité bucco-nasale

La cavité bucco-nasale joue un rôle clé dans la résonance et la modulation du signal vocal. Son influence sur les propriétés spectrales de la voix permet d'extraire des descripteurs acoustiques pertinents, utilisés pour analyser et caractériser les variations de la parole. Parmi ces descripteurs, les MFCC sont particulièrement efficaces pour modéliser les caractéristiques spectrales du signal vocal [14].

Les MFCC sont obtenus en décomposant le spectre de la voix sur une échelle de fréquences perceptuelle, appelée échelle de Mel. Cette transformation permet d'extraire des caractéristiques robustes, qui capturent les variations des résonances vocales tout en réduisant l'influence des hautes fréquences. Grâce à cette approche, les MFCC sont couramment utilisés pour l'analyse des troubles vocaux, notamment dans le cadre de la MP, où la voix peut présenter des altérations subtiles mais significatives.
Dans cette etude on a utilisé les 13 premiers MFCC.

2.1.2 Les descripteurs des cordes vocales

Chez les PD de la maladie de Parkinson, la voix subit des altérations significatives dues aux troubles neuromusculaires affectant le contrôle des cordes vocales, ces changements se manifestent par une hypophonie, une réduction de l'intonation et une instabilité de la phonation, l'extraction et l'analyse des descripteurs vocaux permettent d'objectiver ces altérations et de suivre l'évolution de la maladie. Parmi les indicateurs les plus pertinents, on retrouve **la fréquence fondamentale F0**, dont la variabilité est souvent altérée chez les PD. Contrairement à une simple diminution de F0, c'est plutôt la réduction de sa variabilité qui est caractéristique de la maladie, entraînant une intonation plus monotone. **Le Jitter**, qui mesure les fluctuations aléatoires de la fréquence fondamentale et qui est représenté par l'équation suivante :

$$jitt = \frac{\frac{1}{N-1}\sum_{t=1}^{N-1}|T_i - T_{i-1}|}{\frac{1}{N}\sum_{i=1}^{N} T_i} \times 100 \qquad [4]$$

Une valeur plus élevée de jitter traduit un manque de stabilité dans la vibration des cordes vocales. **Le Shimmer**, qui reflète une variation irrégulière de l'amplitude vocale et qui se traduit par :

$$Shim = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}|A_i - A_{i+1}|}{\frac{1}{N}\sum_{i=1}^{N} A_i} \times 100 \qquad [5]$$

Une diminution du contrôle de l'intensité de la voix contribue ainsi à une qualité vocale altérée. Enfin le **rapport harmonique-bruit HNR** qui quantifie le rapport entre l'énergie harmonique et le bruit du signal vocal et qui tend à diminuer dans la MP, traduisant une augmentation du bruit vocal et une perte de clarté.

2.2 Outils et méthodes d'extraction

Cette section présente les outils et méthodes d'extraction des descripteurs vocaux, essentiels pour l'analyse et la détection automatique des patients parkinsoniens. Nous explorerons les techniques utilisées pour obtenir des caractéristiques pertinentes de la voix.

2.2.1 Environnement de développement

Pour l'implémentation des méthodes d'extraction des caractéristiques vocales, nous avons utilisé Python via la distribution Anaconda, qui facilite la gestion des bibliothèques scientifiques et l'exécution du code dans un environnement interactif. L'interface Jupyter Notebook a été choisie pour sa flexibilité et sa visualisation efficace des données.

2.2.2 Bibliothèques utilisées pour l'extraction des caractéristiques vocales

L'extraction des paramètres acoustiques de la voix a été réalisée à l'aide des bibliothèques suivantes :

- **Praat :** Un programme open source destiné à l'analyse, la manipulation et l'annotation de sons. Ces caractéristiques font de cet outil un instrument complet, notamment pour l'analyse de la parole. C'est aussi un outil qui permet de dessiner des graphiques, d'élaborer des grammaires fondées sur la théorie de l'optimalité, d'effectuer des synthèses articulatoires, de modéliser avec des réseaux neuronaux et de réaliser des analyses statistiques.
- **Parselmouth :** Une bibliothèque Python qui sert d'interface à Praat, un logiciel de référence en analyse acoustique et phonétique. Elle permet une interaction simplifiée avec Praat, en tirant parti de la flexibilité et de la puissance de Python. Grâce à Parselmouth, il est possible d'extraire divers descripteurs acoustiques utiles à l'analyse de la voix, tels que la fréquence fondamentale (F0), l'intensité, le rapport harmonique-bruit (HNR), le jitter et le shimmer, qui sont des indicateurs de stabilité vocale [15].

— **Librosa :** Une librairie dédiée à l'analyse du signal audio, souvent utilisée pour l'extraction de caractéristiques acoustiques. Elle permet particulièrement de réaliser des calculs de MFCCs, d'étudier les spectrogrammes et d'isoler différentes propriétés spectrales de la voix.

— **SciPy et NumPy :** Deux outils nécessaires pour le traitement numérique et statistique des signaux audio. Ils simplifient les tâches de filtrage, d'analyse fréquentielle, ainsi que le calcul de métriques utilisées dans la reconnaissance et la classification vocale.

2.3 Conclusion

L'extraction des descripteurs vocaux permet d'analyser objectivement les altérations de la voix chez les patients atteints de la maladie de Parkinson. Parmi les nombreux descripteurs existants, les MFCCs, F0, Jitter, Shimmer et HNR se distinguent par leur efficacité pour la détection de la maladie. L'utilisation d'outils comme Parselmouth, Praat et Librosa facilite cette analyse et constitue une base essentielle pour les approches de classification automatique.

Chapitre 3

Détection automatique de la maladie de Parkinson par l'analyse de la voix

Le diagnostic précoce de la maladie de Parkinson passe par l'analyse des altérations vocales, une méthode non invasive prometteuse. Grâce à l'apprentissage automatique, ces caractéristiques vocales permettent de détecter la MP à un stade précoce.

Ce chapitre explore les principes de l'apprentissage supervisé, en mettant particulièrement l'accent sur les machines à vecteurs de support (SVM) pour la classification. Nous présenterons la validation croisée à 5 folds afin d'évaluer la robustesse du modèle. Par la suite, nous détaillerons la base de données utilisée, les étapes de sa normalisation, ainsi que l'analyse statistique et la classification des données. Enfin, nous discuterons des performances du modèle et des résultats obtenus.

3.1 L'apprentissage automatique

Après avoir exploré les bases du traitement du signal et l'analyse de la voix, il devient crucial de se tourner vers des techniques avancées permettant de classer efficacement les descripteurs vocaux. L'apprentissage automatique offre des outils puissants pour améliorer la détection automatiques des PD en exploitant les données recueillies. Dans cette section, nous examinerons les méthodes et les modèles de classification adaptés à notre projet, afin d'optimiser la précision et l'efficacité du diagnostic.

3.1.1 Définition de l'apprentissage automatique

Le machine learning est un domaine de mathématiques appliquées et de l'informatique permettant aux ordinateurs d'apprendre et de s'améliorer sans programmation explicite, avec des applications importantes, notamment en médecine [16].
Ce domaine comprend plusieurs types d'apprentissage :

— **L'apprentissage supervisé :** C'est une tâche d'apprentissage automatique qui consiste à apprendre une fonction reliant une entrée à une sortie, basée sur des paires d'exemples d'entrée-sortie. Elle déduit une fonction à partir de données d'entraînement étiquetées composées d'un ensemble d'exemples d'apprentissage. L'ensemble de données d'entrée est scindé en un ensemble d'entraînement et un ensemble de test. Le jeu de données d'entraînement comprend une variable cible qui doit être prédite ou classée. Tous les algorithmes apprennent une sorte de modèle à partir des ensembles de données d'apprentissage et les appliquent aux ensembles de données de test pour la prédiction ou la détection automatique [17].

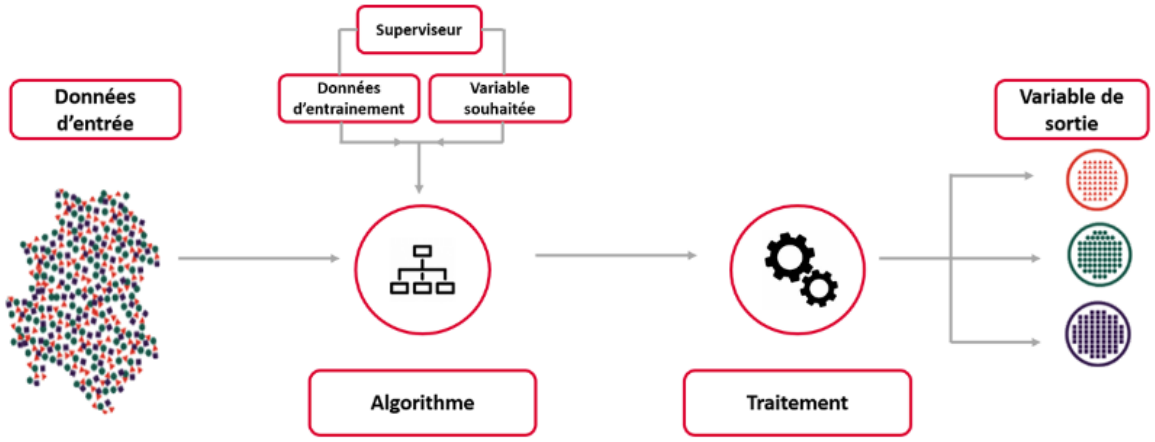


FIGURE 3.1 – L'apprentissage supervisé [17]

— **L'apprentissage non supervisé :** On parle d'apprentissage non supervisé car, contrairement à l'apprentissage supervisé, il n'y a pas de réponses correctes et pas d'enseignant. Les algorithmes sont laissés à leur propre initiative pour découvrir et présenter la structure intéressante dans les données. Les algorithmes d'apprentissage non supervisé apprennent quelques caractéristiques à partir des données. Lorsqu'on

introduit de nouvelles données, le modèle utilise les caractéristiques précédemment apprises pour identifier la classe de ces données. Il est principalement utilisé pour le regroupement et la réduction de caractéristiques [17].

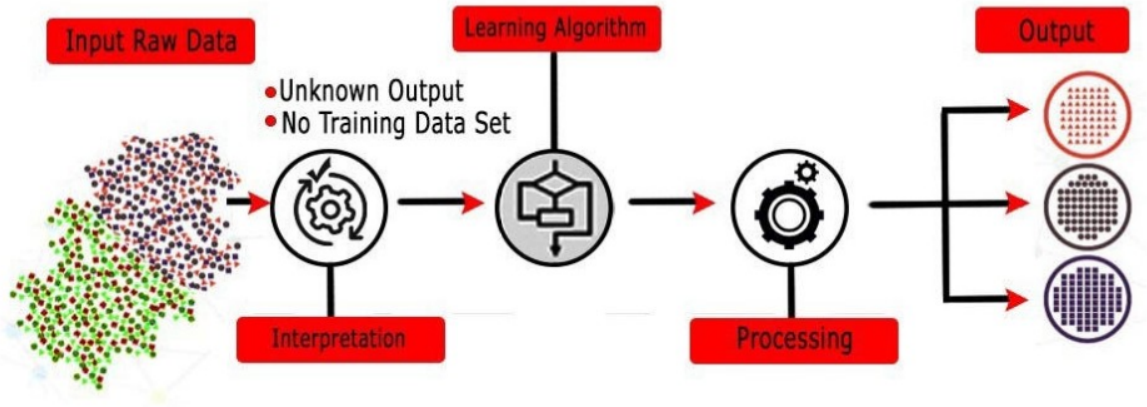


FIGURE 3.2 – L'apprentissage non supervisé [17]

— **L'apprentissage par renforcement :** De manière générale, l'apprentissage par renforcement est une structure formelle qui représente des problèmes décisionnels séquentiels. Dans ce contexte, un agent est entraîné à faire des choix optimaux en interagissant avec son environnement. Quand l'agent réalise une action, la condition du système se modifie et il reçoit une valeur unique, nommée récompense, qui intègre les renseignements concernant la qualité de la transition [18].

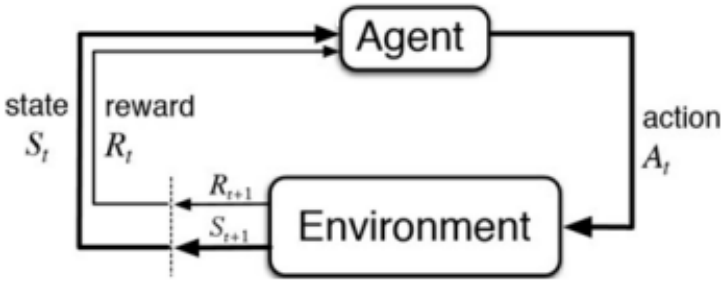


FIGURE 3.3 – L'apprentissage par renforcement [18]

Dans ce projet, nous avons utilisé l'apprentissage supervisé afin de détecter automatique-

ment les PD à partir de leurs descripteurs vocaux. Cette approche permet d'obtenir des résultats précis et fiables pour la classification.

3.1.2 Machine à vecteurs de support : SVM

Dans le domaine de l'apprentissage automatique, les SVM sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés, utilisés pour analyser les données dans le cadre de l'analyse de classification et de régression. Les SVMs, en plus d'assurer une classification linéaire, ont la capacité de réaliser efficacement une classification non linéaire grâce à ce qu'on appelle la technique du noyau, qui permet de projeter implicitement les données d'entrée dans des espaces de caractéristiques de haute dimension [17]. Il existe plusieurs types de noyaux utilisés dans les SVM, tels que le noyau linéaire, le noyau polynomial, le noyau Gaussien, et le noyau sigmoïde, chacun ayant des caractéristiques spécifiques adaptées à différents types de données. En gros, cette technique trace les frontières entre les différentes classes. Les marges sont tracées de telle manière que la distance entre celle-ci et les classes est maximisée, minimisant ainsi l'erreur de classification. Les bibliothèques populaires pour implémenter des SVM incluent scikit-learn (sKLearn), qui fournit des outils comme SVC pour réaliser ces classifications. Scikit-learn offre une interface simple et efficace pour appliquer ces modèles, avec la possibilité de sélectionner le type de noyau et d'optimiser les paramètres pour améliorer la performance du modèle.

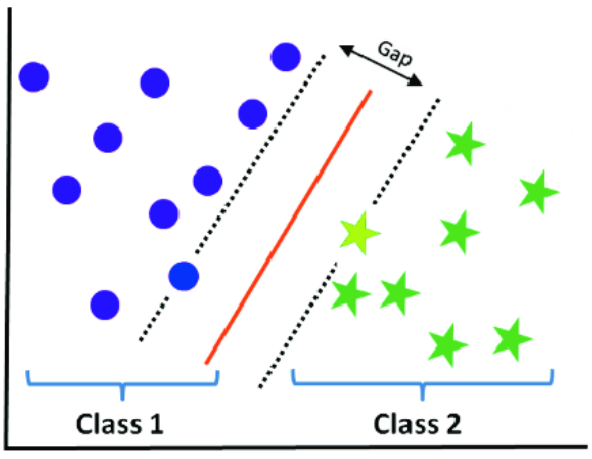


FIGURE 3.4 – SVM

3.1.3 Cross-validation

Dans notre travail, nous avons utilisé la validation croisée (cross-validation) pour évaluer et comparer les performances de nos modèles de classification. Cette méthode est essentielle pour garantir que le modèle ne se contente pas de mémoriser les données d'entraînement, mais qu'il peut bien généraliser à de nouvelles données. L'approche la plus basique, le Train-Test Split, consiste à diviser aléatoirement le dataset en deux parties : une pour l'entraînement (généralement 70-80%) et l'autre pour le test (20-30%). Cependant, cette méthode peut être biaisée si le dataset est limité. Pour pallier ces limites, nous avons appliqué la validation croisée K-Folds, qui divise les données en K sous-ensembles (folds). À chaque itération, un fold est utilisé pour tester le modèle tandis que les autres servent à l'entraînement. Cela permet d'obtenir une évaluation plus robuste et équilibrée des performances du modèle. Dans le cas où les données de sortie sont catégoriques, nous avons utilisé Stratified K-Folds, qui assure une répartition homogène des classes dans chaque fold, évitant ainsi un déséquilibre pouvant fausser l'apprentissage. Enfin, l'ensemble de ces techniques nous a permis de tester plusieurs modèles et de sélectionner celui offrant le meilleur compromis entre biais et variance, garantissant ainsi une bonne généralisation [19].

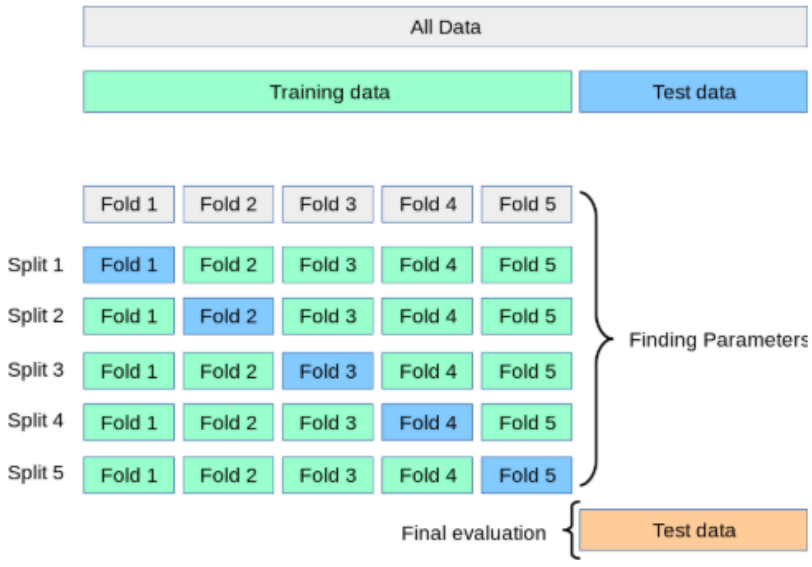


FIGURE 3.5 – Cross-validation [20]

3.2 Bases de données

Dans cette section, nous présentons les bases de données utilisée pour l'analyse et la classification des voix parkinsoniennes. La qualité et la structure des données sont essentielles pour assurer des résultats fiables et pertinents. Nous décrirons d'abord la nature de la base de données audio, en précisant ses caractéristiques et son origine. Ensuite, nous aborderons le processus de normalisation des valeurs des descripteurs extraites, une étape clé pour garantir une meilleure convergence des modèles.

3.2.1 Description de la base de données audio

La base de données décrite par [21] est un corpus de parole en espagnol conçu pour l'analyse des altérations vocales chez les patients atteints de la maladie de Parkinson. Elle comprend des enregistrements audio sous format wav échantionnés à 44,1KHz et codée à 16 bits/échatillon de 100 participants, répartis équitablement entre un groupe de PD et un groupe de HC, chacun réalisant trois répétitions de voyelles soutenues. Pour assurer une meilleure représentativité et minimiser les biais liés à la variabilité de la voix, seule la deuxième répétition a été sélectionnée, car elle est considérée comme la plus stable, réduisant ainsi l'impact de la fatigue vocale ou des fluctuations entre le premier et le dernier enregistrement. Captés dans des conditions contrôlées, ces enregistrements sont utilisés pour la détection automatique de la maladie de Parkinson, l'étude des biomarqueurs vocaux et l'entraînement d'algorithmes d'apprentissage automatique dédiés à l'analyse des troubles neurologiques.

3.2.2 Base de données des descripteurs

Dans le cadre de notre travail, après l'extraction des descripteurs vocaux, nous avons constitué une base de données à partir de ces derniers. Afin d'assurer une homogénéité entre les différentes caractéristiques et de réduire l'impact des différences d'échelle, nous avons appliqué une normalisation de type Min-Max sur les données d'entraînement. Cette méthode permet de transformer les valeurs de chaque descripteur pour qu'elles se situent

dans l'intervalle [0, 1], selon l'équation suivante :

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

Il est important de noter que cette normalisation est effectuée uniquement sur l'ensemble d'entraînement, sans tenir compte de l'ensemble de test. Cela garantit que les informations relatives aux données de test ne sont pas introduites dans le processus d'apprentissage, préservant ainsi l'intégrité du modèle. Lors de l'application de cette transformation sur l'ensemble de test, il est crucial de comparer les nouvelles valeurs normalisées avec celles de l'ensemble d'entraînement. Si une valeur de l'ensemble de test après normalisation dépasse 1, elle est contraint à 1, et si elle est inférieure à 0, elle est contrainte à 0. Cette approche permet de maintenir les valeurs dans la même plage que celles de l'ensemble d'entraînement, ce qui est essentiel pour une évaluation correcte du modèle. Cependant, cette méthode peut introduire une légère distorsion si les données de test présentent des valeurs extrêmes non observées dans les données d'entraînement, mais elle reste une solution pratique pour garantir la cohérence entre les ensembles tout en évitant un surapprentissage des données de test.

3.3 Analyse statistique et classification des données

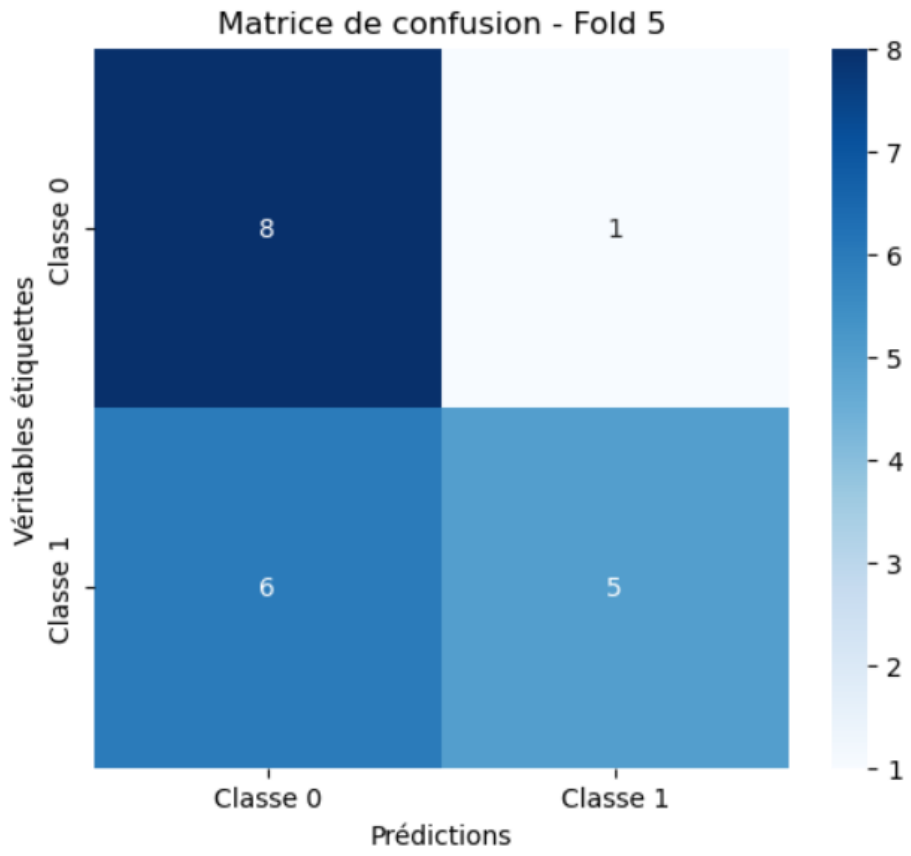
Le code effectue une validation croisée en 5 plis pour évaluer un modèle de classification SVM en suivant plusieurs étapes. Tout d'abord, les données sont converties en tableaux NumPy, puis la validation croisée est mise en place à l'aide de KFold, divisant les données en 5 sous-ensembles aléatoires. À chaque itération, un sous-ensemble est utilisé comme test et les 4 autres comme entraînement. Les données d'entraînement sont normalisées avec MinMaxScaler, et la même transformation est appliquée aux données de test pour garantir une cohérence. Ensuite, le modèle est entraîné sur les données normalisées d'entraînement, puis utilisé pour prédire les étiquettes du jeu de test. Plusieurs métriques sont calculées pour évaluer la performance du modèle : précision (accuracy), rappel (recall) et score F1. Une matrice de confusion est également affichée pour analyser les erreurs de classification. Enfin, après les 5 itérations, les scores moyens des métriques sont calculés et affichés, donnant une évaluation globale de la performance du modèle. L'utilisation de différents noyaux (linéaire, gaussien, polynomial) a permis d'obtenir des résultats variés en termes

de précision. Le table 3.1 résume les résultats de précision de différents noyaux :

Type de noyau	Précision moyenne
Linéaire	0.6
Polynomial	0.7
Gaussien (rbf)	0.67

TABLE 3.1 – les précisions moyennes de différents noyaux de la cross-validation

La figure ci-dessous représente les résultats du noyau polynomial




```
Score moyen de la cross-validation :
- Précision : 0.7000
- Recall : 0.7354
- F1-Score : 0.6851
```

FIGURE 3.6 – Matrice de confusion

Ces résultats soulignent l'importance cruciale du choix du noyau dans l'optimisation des performances du modèle de classification.

Afin d'analyser plus en détail les caractéristiques discriminantes entre les HC (groupe 0) et PD (groupe 1), nous avons tracé les boxplots de plusieurs descripteurs extraits des signaux vocaux.

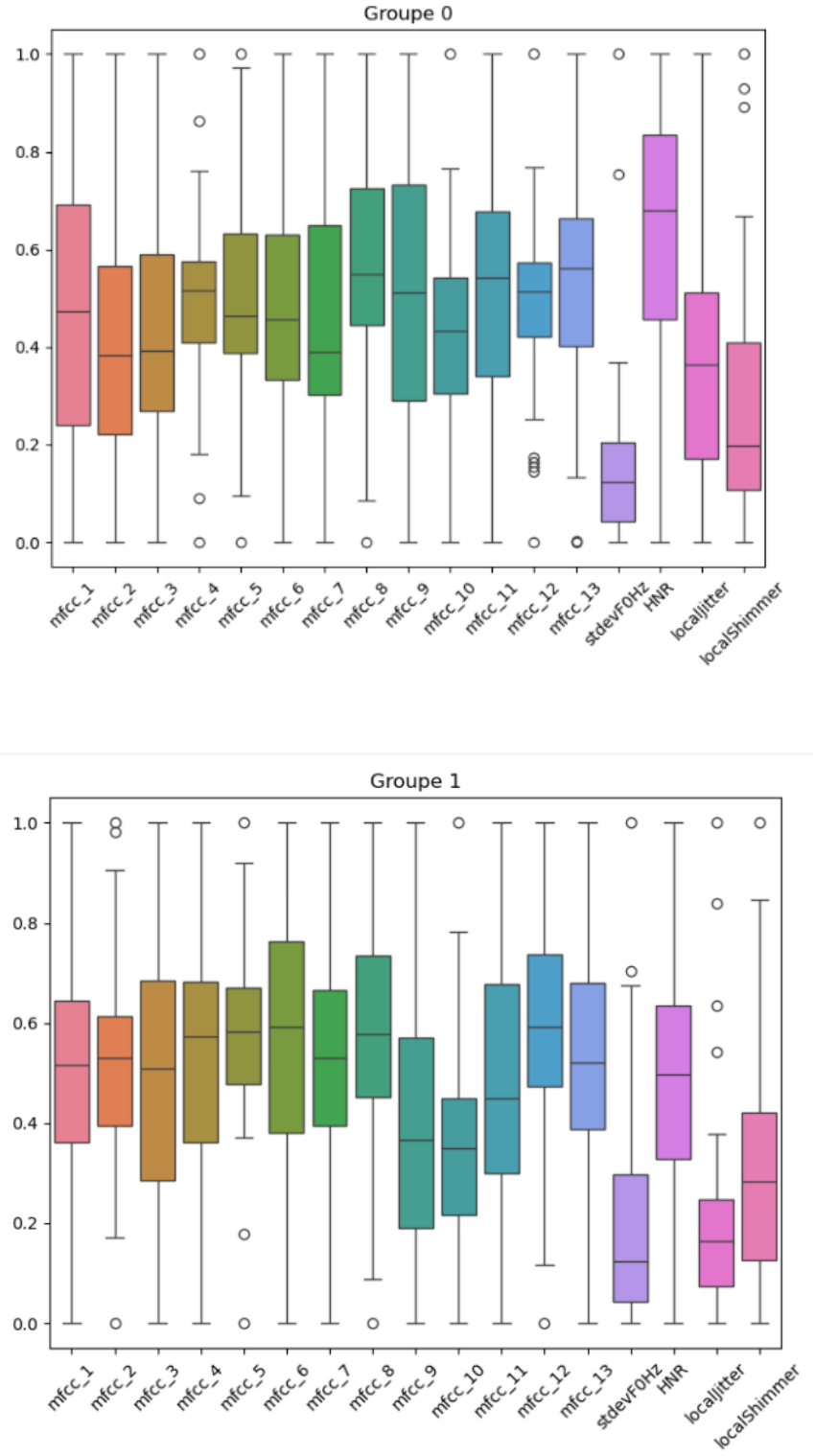


FIGURE 3.7 – Les boxplots des descripteurs acoustiques

Parmi ces descripteurs, le HNR (Harmonics-to-Noise Ratio), la fréquence fondamentale F0, le Jitter et le coefficient MFCC12 se sont révélés être les plus précis pour différencier les deux groupes. En effet, les différences entre les distributions des PD et HC sont plus marquées pour ces descripteurs, mettant en évidence leur potentiel pour la classification.

3.4 Conclusion

Ce chapitre a examiné la détection automatique de la maladie de Parkinson à travers l'analyse vocale, en recourant à l'apprentissage machine. Nous avons exposé l'algorithme SVM, la validation croisée, ainsi que la base de données et sa normalisation. L'analyse statistique a permis de déterminer les descripteurs les plus discriminants, et les résultats ont démontré l'influence du choix des noyaux SVM sur la classification. Cette étude met en évidence les apports du machine learning et présente plusieurs axe d'amélioration.

Conclusion générale

Cette étude a mis en lumière le potentiel de l'analyse vocale dans la détection précoce de la maladie de Parkinson. En faisant des recherches sur l'effet de la maladie sur la voix, nous avons identifié des altérations acoustiques importantes qui peuvent être utilisées comme indicateurs fiables. L'extraction et l'analyse de descripteurs vocaux, associées à des méthodes d'apprentissage automatique telles que les SVM, ont permis de classer les patients avec une grande précision.

L'un des des avantages principaux de cette approche est la possibilité de détecter la maladie à un stade précoce, avant l'apparition de symptômes moteurs plus évidents. Une telle avancée pourrait améliorer la prise en charge des patients en permettant une intervention rapide et un suivi médical adapté. Cette étude ouvre donc la voie à l'optimisation des modèles de classement et à l'intégration des nouvelles méthodes, afin de rendre encore plus précise la détection et d'utiliser-la comme un outil clinique efficace et accessible.

Bibliographie

[1] Titze, I. R. (1994). Principles of Voice Production. Prentice Hall.

[2] Giovanni, A., Lagier, A., & Henrich, N. (2017). Physiologie de la phonation. Consulté à l'adresse

[3] Sonia Djaziri-Larbi, "Traitement de la parole", cours de master Traitement de l'Information et Complexité du Vivant", Ecole Nationale d'Ingénieurs de Tunis, 2022.

[4] Brockmann M, Drinnan M J, Storck C, Carding P N. Reliable Jitter and Shimmer Measurements in Voice Clinics : The Relevance of Vowel, Gender, Vocal Intensity, and Fundamental Frequency Effects in a Typical Clinical Task. In Journal of Voice. Volume 25, Issue 1, January 2011.

[5] Teixeira, J. P.; Ferreira, D.; Carneiro, S.. Análise acústica vocal - determinação do Jitter e Shimmer para diagnóstico de patalogias da fala. In 6º Congresso Luso-Moçambicano de Engenharia. Maputo, Moçambique, 2011.

[6] chaiwongyen2024potential,Potential of Speech-pathological Features for Deepfake Speech Detection,Chaiwongyen, Anuwat and Duangpummet, Suradej and Karnjana, Jessada and Kongprawechnon, Waree and Unoki, Masashi,IEEE Access,2024,IEEE .

[7] Yumoto, E., Gould, W. J., & Baer, T. (1982). Harmonics-to-noise ratio as an index of the degree of hoarseness. Journal of Speech and Hearing Research, 25(2), 202-210.

[8] Maria C Rodriguez-Oroz, Marjan Jahanshahi, Paul Krack, Irene Litvan, Raúl Macias, Erwan Bezard,et José A Obeso. Initial clinical manifestations ofParkinson's disease : features and pathophysiological mechanisms. The Lancet Neurology, 8(12) :1128–1139, Décembre 2009.

[9] TROUBLES ET RÉÉDUCATION DE LA VOIX, DE LA PAROLE ET DE LA DEGLUTITION DANS LA MALADIE DE PARKINSON. Lorraine Olson Ramig, Stefanie Countryman, Cynthia Fox, Shimon Sapir.2001.

[10] Véronique Rolland-Monnoury. Les troubles de la parole dans la maladie de Parkinson. L'échos,(104) :17–18, Septembre 2010.

[11] Sabine Skodda, Julia Lorenz, et Uwe Schlegel. Instability of syllable repetition in Parkinson's disease—Impairment of automated speech performance ? Basal Ganglia, 3(1) :33–37, Mars 2013.

[12] Brian Harel, Michael Cannizzaro, et Peter J. Snyder.Variability in fundamental frequency during speech in prodromal and incipient Parkinson's disease : A longitudinal case study. Brain and Cognition, 56(1) :24–29, Octobre 2004.

[13] Jan Rusz, Jan Hlavnicka, Tereza Tykalova, Jitka Buskova, Olga Ulmanova, Evzen R uzicka, et KarelSonka. Quantitative assessment of motor speech abnormalities in idiopathic rapid eye movement sleep behaviour disorder. Sleep Medicine, Septembre 2015.

[14] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357-366.

[15] Jadoul, Y., Thompson, B., & de Boer, B. (2018). Introducing Parselmouth : A Python interface to Praat. Journal of Phonetics, 71, 1-15.

[16] An Introduction to Machine Learning, Vanessa Nilsen, May 22, 2018

[17] IJSR. (2020). Supervised learning in machine learning. International Journal of Science and Research, 9(1). https ://www.ijsr.net ; Licensed under Creative Commons Attribution CC BY.

[18] Apprentissage par renforcement développemental, Matthieu Zimmer, Université de Lorraine, 2018. Français

[19] Cross-Validation ou Validation croisée : définition et importance, Raphael Kassel, 3 Mai 2021, https ://datascientest.com/cross-validation

[20] Cross-validation :evaluating estimator performance. https ://scikit-learn.org/stable/modules/cross_validation.html

[21] Orozco-arroyave new 2014, New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease, Proceedings of the Ninth International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Orozco-Arroyave, Juan Rafael and Arias-Londoño, Julián David

and Vargas-Bonilla, Jesús Francisco and González-Rátiva, María Claudia and Nöth, Elmar, may 2014.

Annexe A

Script Python

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "6b40082d-cdc5-4b0e-adf0-4c5a8112d490",
   "metadata": {},
   "outputs": [],
   "source": [
    "#Installation du librosa\n",
    "pip install librosa\n",
    "#Installation du Parselmouth\n",
    "pip install praat-parselmouth\n",
    "#Audio Features Extraction\n",
    "##Importation des librairies\n",
    "import glob\n",
    "import numpy as np\n",
    "import pandas as pd\n",
    "import parselmouth\n",
    "from parselmouth.praat import call\n",
    "\n",
    "import librosa\n",
    "import os\n",
    "##Extraction de F0, Jitter, Shimmer, HNR\n",
    "def measurePitch(voiceID, f0min, f0max, unit):\n",
    "    sound = parselmouth.Sound(voiceID) # read the sound\n",
```

```
    "    pitch = call(sound, \"To Pitch\", 0.0, f0min, f0max) #create a praat pitch object\n",
    "    meanF0 = call(pitch, \"Get mean\", 0, 0, unit) # get mean pitch\n",
    "    stdevF0 = call(pitch, \"Get standard deviation\", 0 ,0, unit) # get standard deviation\n",
    "    harmonicity = call(sound, \"To Harmonicity (cc)\", 0.01, 75, 0.1, 1.0)\n",
    "    hnr = call(harmonicity, \"Get mean\", 0, 0)\n",
    "    pointProcess = call(sound, \"To PointProcess (periodic, cc)\", f0min, f0max)\n",
    "    localJitter = call(pointProcess, \"Get jitter (local)\", 0, 0, 0.0001, 0.02, 1.3)\n",
    "    localabsoluteJitter = call(pointProcess, \"Get jitter (local, absolute)\", 0, 0, 0.0001, 0.02, 1.3)\n",
    "    rapJitter = call(pointProcess, \"Get jitter (rap)\", 0, 0, 0.0001, 0.02, 1.3)\n",
    "    ppq5Jitter = call(pointProcess, \"Get jitter (ppq5)\", 0, 0, 0.0001, 0.02, 1.3)\n",
    "    ddpJitter = call(pointProcess, \"Get jitter (ddp)\", 0, 0, 0.0001, 0.02, 1.3)\n",
    "    localShimmer =  call([sound, pointProcess], \"Get shimmer (local)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "    localdbShimmer = call([sound, pointProcess], \"Get shimmer (local_dB)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "    apq3Shimmer = call([sound, pointProcess], \"Get shimmer (apq3)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "    aqpq5Shimmer = call([sound, pointProcess], \"Get shimmer (apq5)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "    apq11Shimmer =  call([sound, pointProcess], \"Get shimmer (apq11)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "    ddaShimmer = call([sound, pointProcess], \"Get shimmer (dda)\", 0, 0, 0.0001, 0.02, 1.3, 1.6)\n",
    "\n",
    "\n",
    "    return meanF0, stdevF0, hnr, localJitter, localabsoluteJitter, rapJitter, ppq5Jitter, ddpJitter, localShimmer,localdbShimmer, apq3Shimmer, aqpq5Shimmer, apq11Shimmer, ddaShimmer\n",
    "### create lists to put the results\n",
```

```
    "file_list = []\n",
    "mean_F0_list = []\n",
    "sd_F0_list = []\n",
    "hnr_list = []\n",
    "localJitter_list = []\n",
    "localabsoluteJitter_list = []\n",
    "rapJitter_list = []\n",
    "ppq5Jitter_list = []\n",
    "ddpJitter_list = []\n",
    "localShimmer_list = []\n",
    "localdbShimmer_list = []\n",
    "apq3Shimmer_list = []\n",
    "aqpq5Shimmer_list = []\n",
    "apq11Shimmer_list = []\n",
    "ddaShimmer_list = []\n",
    "### Go through all the wave files in the folder and measure pitch\n",
    "for wave_file in glob.glob(\"C:/Users/mouna/OneDrive/Desktop/HC/*.wav\"):\n",
    "    sound = parselmouth.Sound(wave_file)\n",
    "    (meanF0, stdevF0, hnr, localJitter, localabsoluteJitter, rapJitter, ppq5Jitter, ddpJitter, localShimmer, localdbShimmer, apq3Shimmer, aqpq5Shimmer, apq11Shimmer, ddaShimmer) = measurePitch(sound, 75, 500, \"Hertz\")\n",
    "    file_list.append(wave_file) # make an ID list\n",
    "    mean_F0_list.append(meanF0) # make a mean F0 list\n",
    "    sd_F0_list.append(stdevF0) # make a sd F0 list\n",
    "    hnr_list.append(hnr)\n",
    "    localJitter_list.append(localJitter)\n",
    "    localabsoluteJitter_list.append(localabsoluteJitter)\n",
    "    rapJitter_list.append(rapJitter)\n",
    "    ppq5Jitter_list.append(ppq5Jitter)\n",
    "    ddpJitter_list.append(ddpJitter)\n",
    "    localShimmer_list.append(localShimmer)\n",
    "    localdbShimmer_list.append(localdbShimmer)\n",
    "    apq3Shimmer_list.append(apq3Shimmer)\n",
    "    aqpq5Shimmer_list.append(aqpq5Shimmer)\n",
    "    apq11Shimmer_list.append(apq11Shimmer)\n",
    "    ddaShimmer_list.append(ddaShimmer)\n",
```

```
    "df = pd.DataFrame(np.column_stack([file_list, mean_F0_list,
       sd_F0_list, hnr_list, localJitter_list, localabsoluteJitter_list,
        rapJitter_list, ppq5Jitter_list, ddpJitter_list,
       localShimmer_list, localdbShimmer_list, apq3Shimmer_list,
       aqpq5Shimmer_list, apq11Shimmer_list, ddaShimmer_list]),\n",
    "                                columns=['voiceID', 'meanF0Hz', '
       stdevF0Hz', 'HNR', 'localJitter', 'localabsoluteJitter', '
       rapJitter',\n",
    "                                        'ppq5Jitter', 'ddpJitter',
       'localShimmer', 'localdbShimmer', 'apq3Shimmer', 'apq5Shimmer',\n
       ",
    "                                        'apq11Shimmer', 'ddaShimmer
       '])  #add these lists to pandas in the right order\n",
    "### Write out the updated dataframe\n",
    "df.to_csv(\"processed_results_PD.csv\", index=False)\n",
    "df.sample()\n",
    "## Extraction des MFCC\n",
    "folder_path = os.path.expanduser(\"C:/Users/mouna/OneDrive/Desktop/
       HC/\")\n",
    "\n",
    "audio_ext = (\".wav\")\n",
    "\n",
    "feature_names = [\"MFCC{}\".format(i) for i in range(1, 14)]\n",
    "data = pd.DataFrame(columns=[\"voiceID\", \"group\"] +
       feature_names)\n",
    "\n",
    "### Boucle chaque fichier audio dans le dossier.\n",
    "for filename in os.listdir(folder_path):\n",
    "    if filename.endswith(audio_ext):\n",
    "        file_path = os.path.join(folder_path, filename)\n",
    "        name = os.path.splitext(filename)[0]\n",
    "        status = 1 if name.startswith(\"P\") else 0  # Les fichiers
       commen ant par \"P\" sont consid r s comme appartenant\n",
    "        #   des patients atteints de la maladie de Parkinson, et
       tous les autres fichiers sont consid r s comme sains.\n",
    "\n",
    "        #### Extract MFCC features\n",
    "        signal, sr = librosa.load(file_path, sr=8000)\n",
    "\n",
```

```
    "        mfccs = librosa.feature.mfcc(y=signal, sr=sr, n_mfcc=13,
       hop_length=int(len(signal)/2),\n",
    "                                     window='hann', lifter=22,
       n_mels=22)\n",
    "        mfccs_mean = np.mean(mfccs.T, axis=0) #pour calculer la
       moyenne de chaque coefficient sur toutes les trames du signal
       audio.\n",
    "\n",
    "        #### Ajouter les donn es des caract ristiques au
       DataFrame\n",
    "        row = [name, status] + list(mfccs_mean)\n",
    "        data.loc[len(data)] = row\n",
    "\n",
    "### Save the DataFrame to a CSV file\n",
    "data.to_csv(\"MFCC_A_HC.csv\", index=False)\n",
    "mfcc_delta = librosa.feature.delta(mfcc, order=1)\n",
    "mfcc_delta2 = librosa.feature.delta(mfcc, order=2)\n",
    "## Extraction des MFCC\n",
    "import glob\n",
    "import librosa\n",
    "import numpy as np\n",
    "import pandas as pd\n",
    "\n",
    "audio_path = 'C:/Users/mouna/OneDrive/Desktop/HC/*.wav'\n",
    "window_duration = 0.020  # Taille de la fen tre en dur e de 20 ms
       \n",
    "output_csv = 'MFCCs_means_8KHz_HC_25_12.csv'  # Nom du fichier CSV
       de sortie\n",
    "\n",
    "### Initialisation d'une liste pour stocker les r sultats\n",
    "mfcc_results = []\n",
    "\n",
    "### Traitement de chaque fichier audio\n",
    "for file in glob.glob(audio_path):\n",
    "    #### Chargement du fichier audio\n",
    "    y, sr = librosa.load(file, sr=None)\n",
    "\n",
    "    #### Calcul des param tres\n",
```

```
    "    n_fft = int(sr * window_duration)  # Taille de la fen tre en nombre d'  chantillons  \n",
    "    hop_length = n_fft // 2  # 50 % de superposition\n",
    "\n",
    "    #### Extraction des MFCC\n",
    "    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13, window='hann', lifter=22, n_mels=22, hop_length=hop_length, n_fft=n_fft)\n",
    "\n",
    "    #### Calcul des delta MFCC et delta-delta MFCC\n",
    "    mfcc_delta = librosa.feature.delta(mfcc, order=1)  # Delta (premier ordre)\n",
    "    mfcc_delta2 = librosa.feature.delta(mfcc, order=2)  # Delta-Delta (second ordre)\n",
    "\n",
    "    #### Calcul des moyennes pour chaque fichier\n",
    "    mfcc_mean = np.mean(mfcc, axis=1)\n",
    "    mfcc_delta_mean = np.mean(mfcc_delta, axis=1)\n",
    "    mfcc_delta2_mean = np.mean(mfcc_delta2, axis=1)\n",
    "    #### Ajout des moyennes    la liste des r sultats (MFCC, Delta, Delta-Delta)\n",
    "    mfcc_results.append([file] + list(mfcc_mean) + list(mfcc_delta_mean) + list(mfcc_delta2_mean))\n",
    "\n",
    "### Cr ation d'un DataFrame pandas avec les r sultats\n",
    "columns = ['filename'] + [f'mfcc_{i}' for i in range(1, 14)] + \\\n",
    "          [f'mfcc_delta_{i}' for i in range(1, 14)] + \\\n",
    "          [f'mfcc_delta2_{i}' for i in range(1, 14)]\n",
    "\n",
    "df = pd.DataFrame(mfcc_results, columns=columns)\n",
    "\n",
    "### Sauvegarde des r sultats dans un fichier CSV\n",
    "df.to_csv(output_csv, index=False)\n",
    "\n",
    "print(f\"MFCC, Delta, and Delta-Delta averages saved to {output_csv}\")\n",
    "#Contruction des boxplots\n",
    "##Importation des librairies\n",
    "import numpy as np \n",
```

```
    "import pandas as pd \n",
    "import matplotlib.pyplot as plt \n",
    "import seaborn as sns \n",
    "%matplotlib inline\n",
    "import os\n",
    "import warnings\n",
    "\n",
    "warnings.filterwarnings('ignore')\n",
    "\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.preprocessing import MinMaxScaler\n",
    "##Importation du fichier excel contenant les descripteurs acoustiques\n",
    "data = 'C:/Users/mouna/OneDrive/Desktop/class.xlsx'\n",
    "df = pd.read_excel(data)\n",
    "df.shape\n",
    "df.head()\n",
    "##Boxplots en fonction du groupe\n",
    "fig, axes = plt.subplots(1, 3, figsize=(15, 6)) \n",
    "\n",
    "mfcc_features = [\"mfcc_1\", \"mfcc_2\", \"mfcc_3\"]\n",
    "palette = \"Paired\"\n",
    "\n",
    "for i, feature in enumerate(mfcc_features):\n",
    "    sns.boxplot(x=df[\"groupe\"], y=df[feature], ax=axes[i], palette=palette)\n",
    "    axes[i].set_title(f\"Boxplot de {feature} par groupe\")\n",
    "    axes[i].set_xlabel(\"Groupe\")\n",
    "    axes[i].set_ylabel(\"Valeur\")\n",
    "\n",
    "plt.tight_layout()  \n",
    "plt.show()\n",
    "df.info()\n",
    "df.isnull().sum()\n",
    "round(df.describe(),2)\n",
    "X = df.drop(['groupe'], axis=1)\n",
    "y = df['groupe']\n",
    "##Data split : train and test\n",
```

```
    "X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
        = 0.2, random_state = 0)\n",
    "X_train.shape, X_test.shape\n",
    "X_train_before = X_train.copy()\n",
    "cols = X_train.columns\n",
    "cols\n",
    "##Normalisation\n",
    "scaler = MinMaxScaler()\n",
    "X_train = scaler.fit_transform(X_train)\n",
    "X_test = scaler.transform(X_test)\n",
    "X_train = pd.DataFrame(X_train, columns=[cols])\n",
    "X_test = pd.DataFrame(X_test, columns=[cols])\n",
    "X_train.describe()\n",
    "X_test.describe()\n",
    "##Boxplots apr s normalisation\n",
    "plt.figure(figsize=(5, 4))\n",
    "sns.boxplot(data=X_train)\n",
    "plt.title(\"Boxplots Apr s Normalisation\")\n",
    "plt.xticks(rotation=45)  \n",
    "plt.show()\n",
    "##Comparaison des boxplots avant et apr s normalisation\n",
    "X_train_after = pd.DataFrame(scaler.fit_transform(X_train), columns
       =X_train.columns)\n",
    "fig, axes = plt.subplots(1, 2, figsize=(14, 6))\n",
    "\n",
    "sns.boxplot(data=X_train_before, ax=axes[0])\n",
    "axes[0].set_title(\"Avant Normalisation\")\n",
    "axes[0].set_xticklabels(axes[0].get_xticklabels(), rotation=45)\n",
    "\n",
    "sns.boxplot(data=X_train_after, ax=axes[1])\n",
    "axes[1].set_title(\"Groupe 1\")\n",
    "axes[1].set_xticklabels(axes[1].get_xticklabels(), rotation=45)\n",
    "\n",
    "plt.tight_layout()\n",
    "plt.show()\n",
    "#Classification\n",
    "##Importation des librairies\n",
    "import numpy as np \n",
    "import pandas as pd\n",
```

```
    "import matplotlib.pyplot as plt \n",
    "import seaborn as sns \n",
    "\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.svm import SVC\n",
    "from sklearn.metrics import confusion_matrix\n",
    "from sklearn.model_selection import KFold\n",
    "from sklearn.metrics import accuracy_score, recall_score, f1_score,
        confusion_matrix\n",
    "from sklearn.preprocessing import MinMaxScaler\n",
    "import warnings\n",
    "warnings.filterwarnings('ignore')\n",
    "##Importation du fichier excel contenant les descripteurs
       acoustiques\n",
    "data = 'C:/Users/mouna/OneDrive/Desktop/class.xlsx'\n",
    "df = pd.read_excel(data)\n",
    "df.head()\n",
    "df.shape\n",
    "##Classification des donn es\n",
    "X = df.drop(['filename'], axis=1)\n",
    "\n",
    "y = df['filename']\n",
    "def cross_validation(X, y, model, scaler=MinMaxScaler(), n_splits
       =5):\n",
    "    kf = KFold(n_splits=n_splits, shuffle=True, random_state=0)  #
       5 plis avec m lange des donn es\n",
    "    scores = []\n",
    "    recalls = []\n",
    "    f1_scores = []\n",
    "\n",
    "    X = np.array(X)\n",
    "    y = np.array(y)\n",
    "\n",
    "    ### Boucle pour chaque pli\n",
    "    for fold, (train_index, test_index) in enumerate(kf.split(X),
       1):\n",
    "        X_train, X_test = X[train_index], X[test_index]\n",
    "        y_train, y_test = y[train_index], y[test_index]\n",
    "        \n",
```

```
277     "        #### Normalisation (sur les donn es d'entra nement)\n",
278     "        X_train_scaled = scaler.fit_transform(X_train)\n",
279     "        X_test_scaled = scaler.transform(X_test)  # Utiliser les
        m mes  param tres de normalisation pour le test\n",
280     "        \n",
281     "        #### Entra ner le mod le\n",
282     "        model.fit(X_train_scaled, y_train)\n",
283     "        \n",
284     "        #### Pr dire sur les donn es de test\n",
285     "        y_pred = model.predict(X_test_scaled)\n",
286     "        \n",
287     "        #### Calculer les scores\n",
288     "        accuracy = accuracy_score(y_test, y_pred)\n",
289     "        recall = recall_score(y_test, y_pred)\n",
290     "        f1 = f1_score(y_test, y_pred)\n",
291     "        \n",
292     "        #### Ajouter les scores    la liste\n",
293     "        scores.append(accuracy)\n",
294     "        recalls.append(recall)\n",
295     "        f1_scores.append(f1)\n",
296     "        \n",
297     "        #### Afficher les scores de chaque it ration\n",
298     "        print(f\"Fold {fold} - Pr cision : {accuracy:.4f}, Recall
        : {recall:.4f}, F1-Score : {f1:.4f}\")\n",
299     "        \n",
300     "        #### Calculer la matrice de confusion\n",
301     "        cm = confusion_matrix(y_test, y_pred)\n",
302     "        \n",
303     "        #### Afficher la matrice de confusion avec seaborn\n",
304     "        plt.figure(figsize=(6, 5))\n",
305     "        sns.heatmap(cm, annot=True, fmt=\"d\", cmap=\"Blues\",
        xticklabels=[\"Classe 0\", \"Classe 1\"], yticklabels=[\"Classe
        0\", \"Classe 1\"])\n",
306     "        plt.title(f\"Matric de confusion - Fold {fold}\")\n",
307     "        plt.xlabel('Pr dictions')\n",
308     "        plt.ylabel('V ritables   tiquettes  ')\n",
309     "        plt.show()\n",
310     "\n",
311     "    ### Calculer la moyenne des scores\n",
```

```
    "    mean_accuracy = np.mean(scores)\n",
    "    mean_recall = np.mean(recalls)\n",
    "    mean_f1 = np.mean(f1_scores)\n",
    "    \n",
    "    print(f\"Score moyen de la cross-validation :\")\n",
    "    print(f\"- Pr cision : {mean_accuracy:.4f}\")\n",
    "    print(f\"- Recall : {mean_recall:.4f}\")\n",
    "    print(f\"- F1-Score : {mean_f1:.4f}\")\n",
    "    \n",
    "    return mean_accuracy, mean_recall, mean_f1\n",
    "from sklearn.svm import SVC\n",
    "\n",
    "model = SVC(kernel='polynomial')  \n",
    "\n",
    "mean_accuracy, mean_recall, mean_f1 = cross_validation(X, y, model,
        scaler=MinMaxScaler(), n_splits=5)"
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python [conda env:base] *",
   "language": "python",
   "name": "conda-base-py"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.12.7"
  }
 },
 "nbformat": 4,
```

```
 "nbformat_minor": 5
}
```

Listing A.1 – Code Jupyter Notebook - Extraction et analyse des caractéristiques acoustiques