

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Chadli Bendjedid

Faculté des Sciences et de la Technologie

Département d'Informatique



الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة الشاذلي بن جديد

كلية العلوم والتكنولوجيا

قسم الآلة

MEMOIRE

Présentée par

REHAMNIA Isslam

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

**APPLICATION DES ALGORITHMES GENETIQUES POUR
LA CLASSIFICATION AUTOMATIQUE DE LA PAROLE**

Soutenue le : 27/10/2020

Devant le Jury composé de :

| Qualité | Nom et Prénom | Grade | Université |
|------------|-----------------------------|-------|--------------------------|
| Président | Mme. Bouguerne Imene | MCB | Chadli Bendjedid El-Tarf |
| Rapporteur | Mme. Makhlouf Amina | MCB | Chadli Bendjedid El-Tarf |
| Examineur | Mr. Benmachiche Abdelmadjid | MCB | Chadli Bendjedid El-Tarf |

Année Universitaire : 2019/2020

Remerciements

Ce travail est l'aboutissement d'un dur labeur et de beaucoup de sacrifices ; nos remerciements vont d'abord au Créateur de l'univers qui nous a doté d'intelligence, et nous a maintenu en santé pour mener à bien cette année d'étude. Je tiens aussi à adresser mes remerciements à ma famille, et plus précisément à mon frère et ma sœur qui m'ont toujours soutenus et poussés à continuer mes études. Ce présent travail a pu voir le jour grâce à leur soutien.

*Je souhaite également remercier les professeurs et tous mes amis, qui ont bien voulu répondre à mes différentes questions sur le domaine dans le but de permettre la facilité de l'élaboration de ce travail. Un remerciement spécial à le docteur **MAKHLOUF AMINA** qui m'a informé et mis sur le chemin de ce programme d'étude.*

Dédicace

*Merci **ALLAH** (mon dieu) de m'avoir donné la capacité d'écrire et de réfléchir.*

La force d'y croire, la patience d'aller jusqu'au bout de rêve

Je dédie ce travail

A ma chère mère

A mon cher père

Pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

*A mes sœurs **ROUMAÏSSA, NOUR EL HOUDA***

*A mon chère frère **YASSER***

Pour leurs encouragements permanents et leurs soutiens moral.

A mes chères ami (e)s :

OUSSAMA, HAKIM, ANTER, CHIHABE, MOHAMED, MOUSTAPHA, AMAR, SAMIR, SARRA.

A toute ma famille

A toute mes autres ami(e)s

A tous ceux que j'aime et ceux qui m'aiment

ISSLAM

Table des matières

| | |
|---|----|
| Remerciements..... | 2 |
| Dédicace..... | 3 |
| Table des matières | 4 |
| Liste des figures | 7 |
| Liste des acronymes | 8 |
| Introduction Générale..... | 9 |
| Objectifs..... | 9 |
| Contenu du mémoire | 9 |
| Chapitre 1 : Etat de l' Art..... | 10 |
| 1. Définition de la parole | 10 |
| 2. Définition de la Reconnaissance Vocale..... | 10 |
| 3. Historique..... | 10 |
| 4. Analyse du signal de parole et décodage..... | 11 |
| 5. Paramétrisation..... | 11 |
| 6. Différents niveaux de paramétrisation..... | 11 |
| 6.1. Niveau mot :..... | 11 |
| 6.2. Niveau phonétique :..... | 11 |
| 6.3. Niveau acoustique : | 12 |
| 7. les différentes phases de reconnaissance | 12 |
| 7.1 . Le capteur | 12 |
| 7.2 . Paramétrisation du signal..... | 12 |
| 7.3 . Prise de décision du choix du mot | 12 |
| 8. Les méthodes de classification..... | 13 |
| 9. Pourquoi la classification ? | 13 |
| 10. Les différentes méthodes de classification..... | 13 |
| 10.1. K-moyennes..... | 13 |

| | | |
|-------------------------------|--|----|
| 10.1.1 | Algorithme K-means | 15 |
| 10.1.2 | Avantages et Inconvénients : | 15 |
| 10.2. | La méthode de KNN : | 16 |
| 10.2.1 | Avantages et Inconvénient: | 17 |
| 10.3. | La méthode de l'arbre de décision : | 17 |
| 10.3.1 | Principe de la construction : | 18 |
| 10.3.2 | Avantages et Inconvénients : | 18 |
| .11 | L'algorithme génétique | 19 |
| 11.1. | Principes généraux | 19 |
| 11.2. | Population initiale | 20 |
| 11.3. | Fonction de Fitness | 21 |
| 11.4. | Sélection | 21 |
| 11.5. | croisement | 21 |
| 11.6. | Mutation | 22 |
| 11.7. | Conditions d'arrêt de l'algorithme | 23 |
| 12. | Avantages et désavantages des algorithmes génétiques | 23 |
| 13. | Réseau de neurones | 23 |
| 13.1. | Définition | 23 |
| 13.2. | Fonctionnement du réseau neuronal | 24 |
| 14.3. | Comment le réseau de neurones artificiels apprend | 24 |
| 13.4. | Les différents types de réseaux de neurones | 25 |
| 14. | classification ascendante hiérarchique | 25 |
| 14.1. | Définition | 25 |
| 14.2. | Principes de la classification ascendante hiérarchique | 25 |
| 14.3. | Avantages de la classification ascendante hiérarchique | 26 |
| 15. | Conclusion | 26 |
| Chapitre 2 : Conception | | 27 |
| 1. | Introduction | 27 |
| 2. | Reconnaissance vocal | 27 |
| 3. | Intelligence artificiel | 27 |

| | |
|--|----|
| 4. Apprentissage profond..... | 28 |
| .5 Fonctionnement du deep Learning | 28 |
| 6. Architecture du solution proposer | 29 |
| .7 Construction de la base de donnée | 30 |
| 8. L'extraction de caractéristique..... | 32 |
| 9. L'algorithme génétique proposer..... | 32 |
| 10. Test et résultat | 33 |
| 11. Conclusion..... | 34 |
| Chapitre 3 : Implémentation et Réalisation | 35 |
| 1. Introduction..... | 35 |
| 2. L'environnement matériel :..... | 35 |
| 3. Langage utiliser | 35 |
| 4. IDE..... | 35 |
| 5. Visual Studio Code..... | 36 |
| 6. Conversion analogique numérique | 36 |
| 7. Frame Work | 36 |
| 8. Présentation de l'application..... | 37 |
| 8.1. Le menu principal :..... | 38 |
| 8.2. La Sélection de la base des données..... | 39 |
| 8.3. La Sélection de vocal pour le test..... | 39 |
| 8.4. Diagramme d'enregistrement audio..... | 40 |
| 8.5. Affichage du résultat | 40 |
| 9. Conclusion | 41 |
| Conclusion et Perspectives | 42 |
| Références..... | 43 |

Liste des figures

| | |
|--|------------------------------------|
| Figure 1.1 : Schéma Synoptique d'un système de reconnaissance de parole selon une approche | 13 |
| Figure 1.2 : Exemple de k-means..... | 15 |
| Figure 1.3 : Fonctionnement de l'algorithme 'KNN' | 17 |
| Figure 1.4 : Exemple arbre de décision | 18 |
| Figure 1.5 : Principe général des algorithmes génétique..... | 20 |
| Figure 1.6 : Population, Chromosomes and Genes..... | 20 |
| Figure 1.7 : pointdecroisement..... | 21 |
| Figure 1.8 : Échange de gènes entre parents..... | Erreur ! Signet non défini. |
| Figure 1.9 :Nouvelle progéniture..... | 22 |
| Figure 1.10 : Mutation avant et après..... | 22 |
| Figure 1.11 : fonctionnement le réseau de neurones artificiels..... | Erreur ! Signet non défini. |
| Figure 2.1 : Architecture du système proposé..... | 29 |
| Figure 2. 2 : La base de données | Erreur ! Signet non défini. |
| Figure 2.3 : la coupure et la conversion de vocal..... | Erreur ! Signet non défini. |
| Figure 2.4 : deuxième base de données | Erreur ! Signet non défini. |
| Figure 2.5 : Initialisation de la population..... | Erreur ! Signet non défini. |
| Figure 2.6 : Croissement de chromosomes..... | Erreur ! Signet non défini. |
| Figure 2.7 : la mutation..... | Erreur ! Signet non défini. |
| Figure 2.8 : Diagramme d'évolution des résultats..... | Erreur ! Signet non défini. |
| Figure 3.1 : La fenêtre principale | Erreur ! Signet non défini. |
| Figure 3.2 : sélectionner les fichiers de la base des données..... | 39 |
| Figure 3.3 : Sélection des fichiers pour faire le test | 39 |
| Figure 3.4 : Diagramme d'enregistrement d'un fichier audio. | Erreur ! Signet non défini. |
| Figure 3.5 : résultat de la classification | Erreur ! Signet non défini. |

Liste des acronymes

AD. arber dication

ARPA. *Advanced Research Project Agency*

ASR. *Automatic Speech Recognition*

CAN. *Conversion analogique numérique*

IA. *intelligence artificiel*

IBM. *International Business Machines*

IDE. *Integrated Development Environment*

IHM. *interface homme machine*

k-means. *K-moyennes*

KNN. *k plus proches voisins*

LPC. *langage parlé complété*

LPCCs. *icensed professional counselors*

MFCCs. *mel-frequency cepstrum*

NTIC.,. *les nouvelles technologies de l'information et de la communication*

PLP. *Perceptual Linear Prediction*

RCGA. *Real Coded Genetic Algorithms*

Introduction Générale

Il est clair que la nature offre un grand nombre de population qui est souhaitable de répertorier en catégorie, et c'est ce que la classification offre, car l'objectif de la classification est de grouper un ensemble de données en classes de telle sorte que deux données appartenant à une même classe soit 12 plus similaire entre eux que deux données appartenant à des classes différentes, et ce relativement à certain critère fixé comme une mesure de similarité. Pour contourner ce choix, nous nous intéressons dans ce mémoire à la conception et à l'implémentation des d'une méthode classique et très utilisée Algorithmes Génétiques qui sont basés sur les concepts de la classification supervisée.

Objectifs

Dans cette étude, nous proposons un système de classification basé sur l'utilisation de l'algorithme Génétique. Le but est de classifier de parole. Nous allons traiter dans notre expérimentation un seul type de signal : la parole. Nos résultats obtenus pourraient être appliqués à des domaines tels que la gestion de documents sonores, la réalisation automatique de résumés de discours ou de réunions, la recherche d'informations, parce que les segments de parole autour d'une question contiennent généralement des informations pouvant s'avérer très utiles dans ces applications.

Contenu du mémoire

Pour réaliser notre objectif, nous avons allons organiser le mémoire comme suit : Une introduction générale, trois chapitres, le premier chapitre est un état de l'art sur les méthodes de classification, le principe de la reconnaissance de la parole et les méthodes de classification qui existent avec les détails des méthodes. Le deuxième chapitre est consacré pour la présentation de l'architecture de système et l'explication des paramètres de chaque méthode utilisée, ainsi la présentation des bases de données exploitées, ensuite la comparaison entre tous les résultats trouvés. Après dans le troisième chapitre, nous présentons les technologies et les outils utilisés dans la réalisation de notre projet. Finalement, nous terminons avec une conclusion générale.

1. Définition de la parole

On peut définir la parole comme :

- C'est un signal réel (4D). continu. d'énergie finie. non stationnaire
- Moyenne de communication par excellence.
- Information qui est véhiculée par les ondes sonores.
- C'est un bruit qui a un sens.
- C'est le vecteur d'information qui différencie l'homme de l'animal.
- Signal permettant la communication entre humains.

2. Définition de la Reconnaissance Vocale

La reconnaissance vocale ou reconnaissance automatique de la parole (Automatic Speech Recognition ASR) est une technique informatique qui permet d'analyser un mot ou une phrase captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

Technologie informatique qui permet à un logiciel d'interpréter la langue naturelle humaine.

Elle utilise le traitement du signal et l'intelligence artificielle.

3. Historique

Les travaux sur la reconnaissance de la parole datent du début du XXe siècle. Le premier système pouvant être considéré comme faisant de la reconnaissance de la parole date de 1952.

La reconnaissance de la parole est un domaine à forte croissance grâce à la déferlante des systèmes embarqués. Une évolution rapide :

1952 : Reconnaissance des 10 chiffres – Bell

1961 : Phonétographe – J.Dreyfus-Graf

1968 : Reconnaissance des mots isolés

1971 - 1976 : Projet ARPA – USA

1972 : Voice Command System – 1ère commercialisation (reconnaissance de mots)
1983 : Commande vocale dans avion de chasse – France 1ère mondiale!
1985 : Commercialisation des premiers systèmes de RV de plusieurs milliers de mots
1994 : Reconnaissance vocal sur PC – IBM
1997 : Dictée vocale en continu – IBM

4. Analyse du signal de parole et décodage

Une fois que le son a été émis par le locuteur, il est capté par un microphone. Le signal vocal est ensuite numérisé à l'aide d'un convertisseur analogique-numérique. Comme la voix humaine est constituée d'une multitude de sons, souvent répétitifs, le signal peut être compressé pour réduire le temps de traitement et l'encombrement en mémoire. L'analyse peut alors commencer.

5. Paramétrisation

La première étape consiste à paramétrer le signal vocal du locuteur. Cela permet d'obtenir une " empreinte " caractéristique du son, sur laquelle on pourra ensuite travailler pour la reconnaissance. Plusieurs méthodes existent.

Il y a deux types de paramétrisation

- Paramétrisation basée modèle de production
Exemple : LPC
- Paramétrisation basée modèle de perception
Exemple : PLP, PLP-RASTA

6. Différents niveaux de paramétrisation

6.1. Niveau mot :

- Durée du mot
- Energie du mot

6.2. Niveau phonétique :

- Durée du phonème
- Energie du phonème

- Taux de passage par zéro
- Fréquence fondamentale du phonème
- Formants

6.3. Niveau acoustique :

- MFCCs
- LPCCs
- Energie

7. les différentes phases de reconnaissance

7.1. Le capteur : un signal électrique est issu du microphone lorsque le locuteur parle

7.2. Paramétrisation du signal : cette étape consiste en un traitement mathématique du signal. Le signal suite à ce traitement est sous une forme :

- Temps
- Fréquence
- Intensité

Difficulté rencontrée : le bruit

Solution : soustraire la densité spectrale du bruit de fond de la densité spectrale du signal mesuré au microphone.

7.3. Prise de décision du choix du mot : Le signal vocal émis par l'utilisateur, une fois paramétré, va pouvoir être comparé aux mots du dictionnaire de référence en terme d'images acoustiques. L'algorithme de reconnaissance permet de choisir le mot le plus ressemblant, par calcul d'un taux de similitude - au sens d'une distance à définir - entre le mot prononcé et les diverses références.

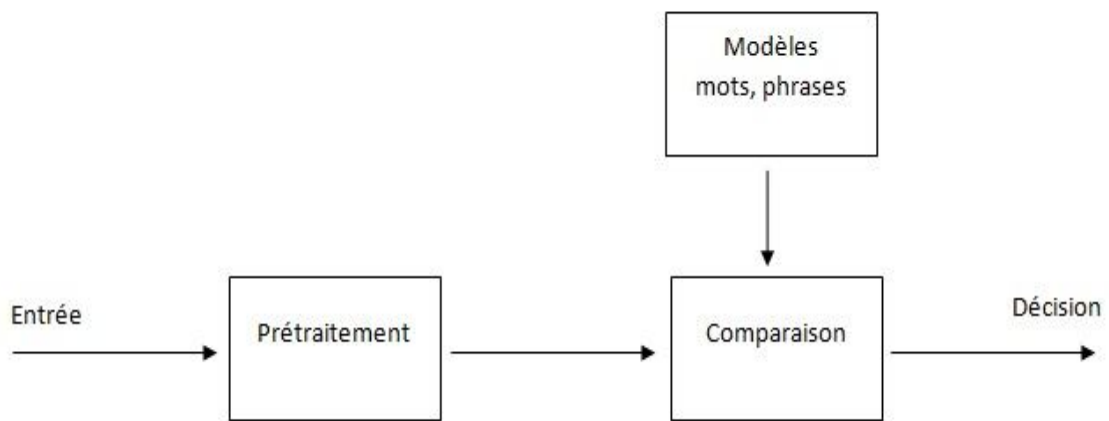


Figure 1.1 : Schéma Synoptique d'un système de reconnaissance de parole selon une approche Globale.

1. Les méthodes de classification

Les méthodes de classification aussi appelées de partition des données, permettent de grouper des objets (observations ou individus) dans des classes (clusters) de manière à ce que les objets appartenant à la même classe sont plus similaires entre eux qu'aux objets appartenant aux autres classes. Le calcul de la proximité entre objets se fait sur une série de variables mesurées sur tous les objets.

2. Pourquoi la classification ?

parmi les objectifs de la classification :

- Identifier des structures sous-jacentes dans les données
- Résumé des comportements
- Affecter de nouveaux individus à des catégories

3. Les différentes méthodes de classification

Il y a plusieurs méthodes de classification on propose quelques méthodes :

10.1. K-moyennes: Le partitionnement en k-moyennes (ou k-means en anglais) est une méthode de partitionnement de données. Étant donné des points et un entier k, le problème est de diviser les points en k groupes, souvent appelés clusters, de façon à minimiser une certaine

fonction. On considère la distance d'un point à la moyenne des points de son cluster ; la fonction à minimiser est la somme des carrés de ces distances.

Notion de similarité : Pour pouvoir regrouper un jeu de données en cluster distincts, l'algorithme K-Means a besoin d'un moyen de comparer le degré de similarité entre les différentes observations. Ainsi, deux données qui se ressemblent, auront une distance de dissimilarité réduite, alors que deux objets différents auront une distance de séparation plus grande.

Les littératures mathématiques et statistiques regorgent de définitions de distance, les plus connues pour les cas de clustering sont :

• **La distance Euclidienne** : C'est la distance géométrique qu'on apprend au collège. Soit une matrice à variables quantitatives. Dans l'espace vectoriel . La distance euclidienne entre deux observations et se calcule comme suit :

$$d(X1, X2) = \sqrt{\sum_j^n (X1n - X2n)^2} \quad (1)$$

• **La distance de Manhattan** (taxi-distance) : est la distance entre deux points parcourue par un taxi lorsqu'il se déplace dans une ville où les rues sont agencées selon un réseau ou un quadrillage. Un taxi-chemin est le trajet fait par un taxi lorsqu'il se déplace d'un nœud du réseau à un autre en utilisant les déplacements horizontaux et verticaux du réseau, et se calcule comme suit :

$$d(X1, X2) = \sum_{i=1}^n |Xi - Yi| \quad (2)$$

10.1.1 Algorithme K-means

Entrée :

- K le nombre de cluster à former
- Le Training Set (matrice de données)

DEBUT

Choisir aléatoirement K points (une ligne de la matrice de données). Ces points sont les centres des clusters (nommé centroïde).

REPETER

Affecter chaque point (élément de la matrice de donnée) au groupe dont il est le plus proche au son centre

Recalculer le centre de chaque cluster et modifier le centroïde

JUSQU'À CONVERGENCE

OU (stabilisation de l'inertie totale de la population)

FIN ALGORITHME

Remarque1 : Lors de la définition de l'algorithme, quand je parle de "point", c'est un point au sens "donnée/data" qui se trouve dans un espace vectoriel de dimension n . Avec n : le nombre de colonnes de la matrice de données.

Remarque2 : La convergence de l'algorithme K-Means peut être l'une des conditions suivantes :

- Un nombre d'itérations fixé à l'avance, dans ce cas, K-means effectuera les itérations et s'arrêtera peu importe la forme de clusters composés.
- Stabilisation des centres de clusters (les centroïdes ne bougent plus lors des itérations)

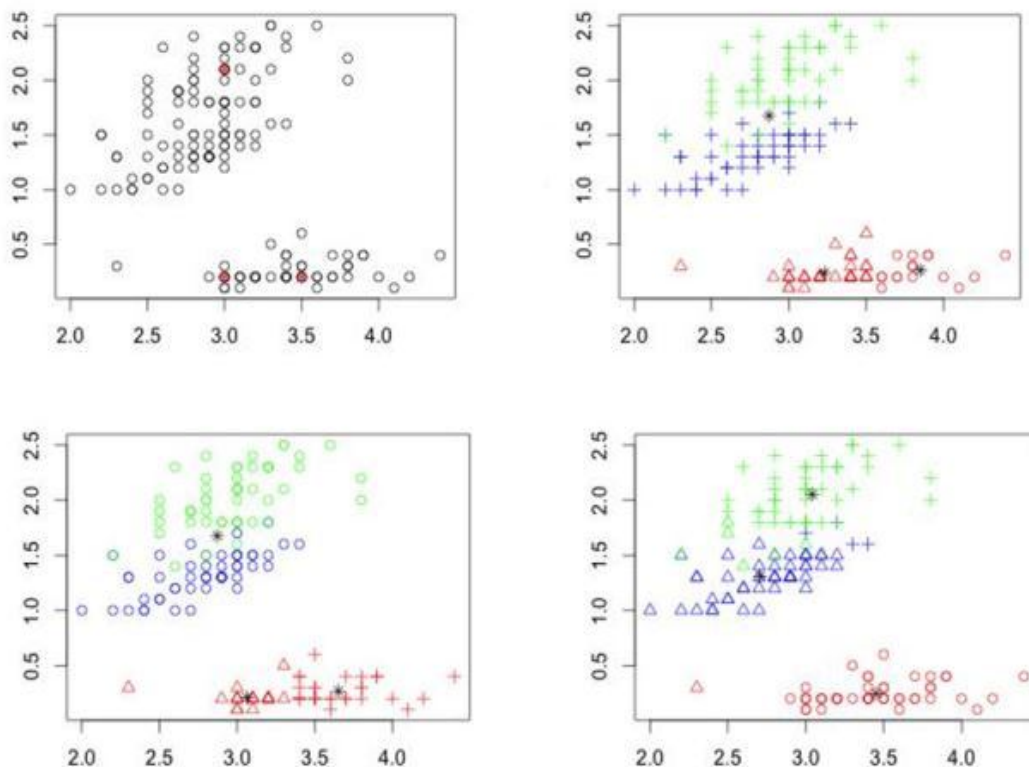


Figure 1.2 : Exemple de k-means (RAKOTOMALALA, 2018)

10.1.2 Avantages et Inconvénients :

On peut citer quelque avantage :

- L'algorithme de k-means est très populaire du fait qu'il est très facile à comprendre et à mettre en œuvre.
- Sa simplicité conceptuelle et sa rapidité
- Applicable à des données de grandes tailles, et aussi à tout type de données (mêmes textuelles)

K-moyennes comme tous les algorithmes a des inconvénients et parmi ces inconvénients :

- Le nombre de classe doit être fixé au départ
- Le résultat dépend de tirage initial des centres des classes
- Les clusters sont construits par rapports à des objets inexistantes (les milieux)

10.2. La méthode de KNN :

La méthode de KNN :ou k plus proches voisins fait partie des méthodes les plus simples d'apprentissage supervisé pouvant être utilisée pour les cas de classification , En effet, cet algorithme est qualifiée comme paresseux (Lazy Learning) car il n'apprend rien pendant la phase d'entraînement. Pour prédire la classe d'une nouvelle donnée d'entrée, il va chercher ses K voisins les plus proches (en utilisant la distance euclidienne, ou autres) et choisira la classe des voisins majoritaires.

Les étapes pour appliquer cette méthode :

- On fixe le nombre de voisins k.
- On détecte les k-voisins les plus proches des nouvelles données d'entrée que l'on veut classer.
- On attribue les classes correspondantes par vote majoritaire

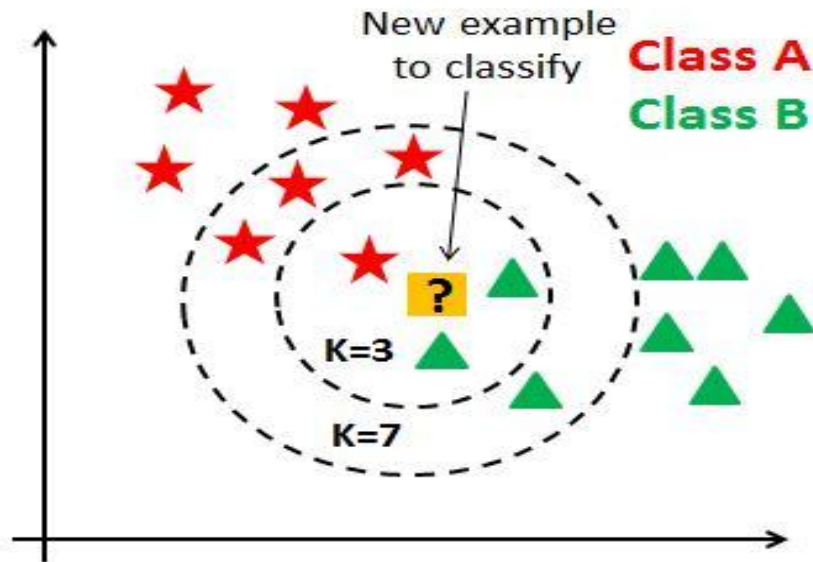


Figure 1.3 : Fonctionnement de l'algorithme 'KNN' (Harifi, 2019)

10.2.1 Avantages et Inconvénient:

On peut citer quelque avantage :

- simple à mettre en œuvre
- donne une probabilité d'erreur faible

KNN comme tous les algorithmes a des inconvénients et parmi ces inconvénients :

- Pas efficace pour des jeux de données larges.
- L'estimation de ce modèle devient de mauvaise qualité quand le nombre de variables explicatives est grand

10.3. La méthode de l'arbre de décision :

- Ensemble de règles de classification basant leur décision sur des tests associés aux attributs, organisés de manière arborescente
- Les arbres de décision (AD) sont une catégorie d'arbres utilisée dans l'exploration de données et en informatique décisionnelle. Ils emploient une représentation hiérarchique de la structure des données sous forme de séquences de décisions (tests) en vue de la prédiction d'un résultat ou d'une classe. Chaque individu (ou observation), qui doit être attribué(e) à une classe, est décrit(e) par un ensemble de variables qui sont testées dans les

nœuds de l'arbre. Les tests s'effectuent dans les nœuds internes et les décisions sont prise dans les nœuds feuille.

10.3.1 Principe de la construction :

Au départ, les points de la base d'apprentissage sont tous placés dans le nœud racine. Une des variables de description des points est la classe du point (la « vérité terrain ») ; cette variable est dite « variable cible ». La variable cible peut être catégorielle (problème de classement) ou valeur réelle (problème de régression). Chaque nœud est coupé (opération split) donnant naissance à plusieurs nœuds descendants. Un élément de la base d'apprentissage situé dans un nœud se retrouvera dans un seul de ses descendants.

- L'arbre est construit par partition récursive de chaque nœud en fonction de la valeur de l'attribut testé à chaque itération (top-down induction). Le critère optimisé est l'homogénéité des descendants par rapport à la variable cible. La variable qui est testée dans un nœud sera celle qui maximise cette homogénéité.
- Le processus s'arrête quand les éléments d'un nœud ont la même valeur pour la variable cible (homogénéité).

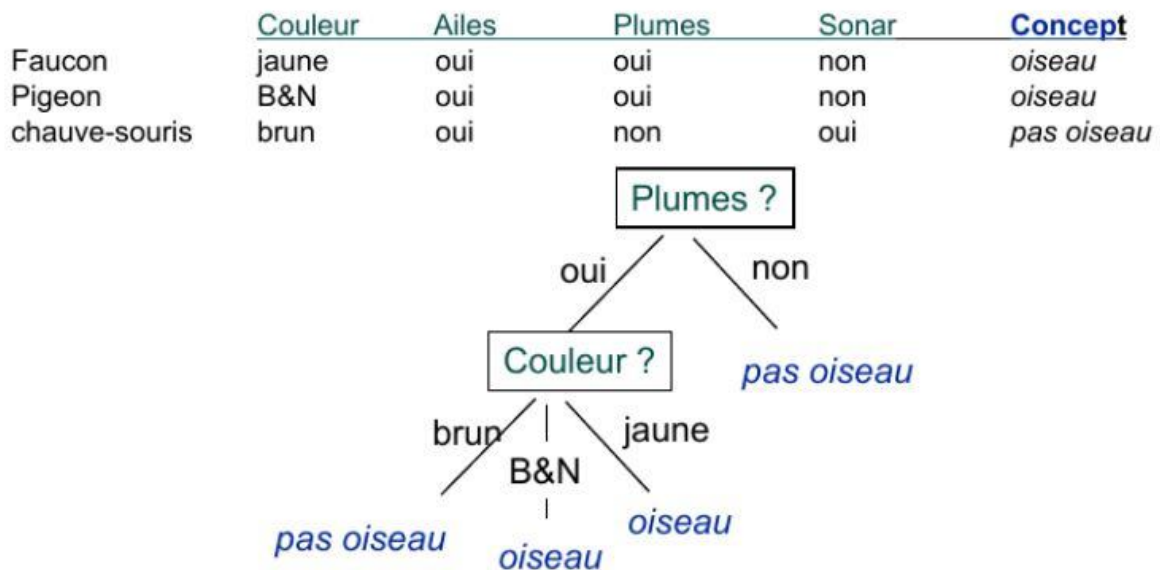


Figure 1.4 : Exemple arbre de décision(Michel Crucianu, 2020)

10.3.2 Avantages et Inconvénients :

L'utilisation des arbres de décision dans l'apprentissage automatique présente plusieurs avantages :

- Le coût d'utilisation de l'arbre pour prédire des données diminue à chaque point de donnée supplémentaire.
- Ils fonctionnent aussi bien pour les données de catégorie que numériques.
- La modélisation des problèmes est possible avec plusieurs données de sortie.
- La fiabilité d'un arbre peut être testée et quantifiée.
- Ils tendent à être précis, même si les hypothèses des données source ne sont pas respectées.

Mais ils présentent aussi quelques inconvénients :

- Lors de la gestion de données de catégorie comportant plusieurs niveaux, le gain d'information est biaisé en faveur des attributs disposant du plus de niveaux.
- Les calculs peuvent devenir compliqués lorsqu'une certaine incertitude est de mise et que de nombreux résultats sont liés entre eux.

4. L'algorithme génétique

11.1. Principes généraux

Comment fonctionne l'algorithme génétique

- 1- L'algorithme commence par créer une population initiale aléatoire.
- 2- L'algorithme crée ensuite une séquence de nouvelles populations. À chaque étape, l'algorithme utilise les individus de la génération actuelle pour créer la population suivante. Pour créer la nouvelle population, l'algorithme effectue les étapes suivantes:
 - Scores chaque membre de la population actuelle en calculant sa valeur de fitness. Ces valeurs sont appelées les scores bruts de fitness.
 - Met à l'échelle les scores de fitness bruts pour les convertir en une plage de valeurs plus utilisable. Ces valeurs mises à l'échelle sont appelées valeurs d'attente.
 - Sélectionne les membres, appelés parents, en fonction de leurs attentes.
 - Certains des individus de la population actuelle qui ont une condition physique inférieure sont choisis comme élite. Ces individus d'élite sont transmis à la population suivante.
 - Produit les enfants des parents. Les enfants sont produits soit en apportant des modifications aléatoires à un seul parent - mutation - ou en combinant les entrées vectorielles d'une paire de parents - croisement.
 - Remplace la population actuelle par les enfants pour former la prochaine génération.
- 3- L'algorithme s'arrête lorsque l'un des critères d'arrêt est satisfait. Voir Conditions d'arrêt de l'algorithme.

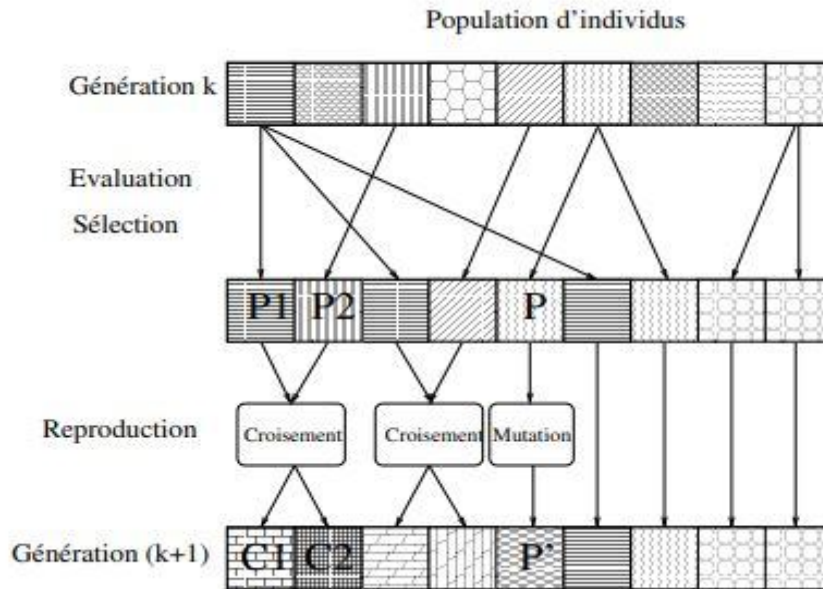


Figure 1.5 : Principe général des algorithmes génétique. (Durand, 2016)

11.2. Population initiale

Le processus commence par un ensemble d'individus que l'on appelle une population. Chaque individu est une solution au problème que vous souhaitez résoudre.

Un individu est caractérisé par un ensemble de paramètres (variables) appelés gènes. Les gènes sont joints en une chaîne pour former un chromosome (solution).

Dans un algorithme génétique, l'ensemble des gènes d'un individu est représenté à l'aide d'une chaîne, en termes d'alphabet. Habituellement, des valeurs binaires sont utilisées (chaîne de 1 et de 0). Nous disons que nous encodons les gènes dans un chromosome.

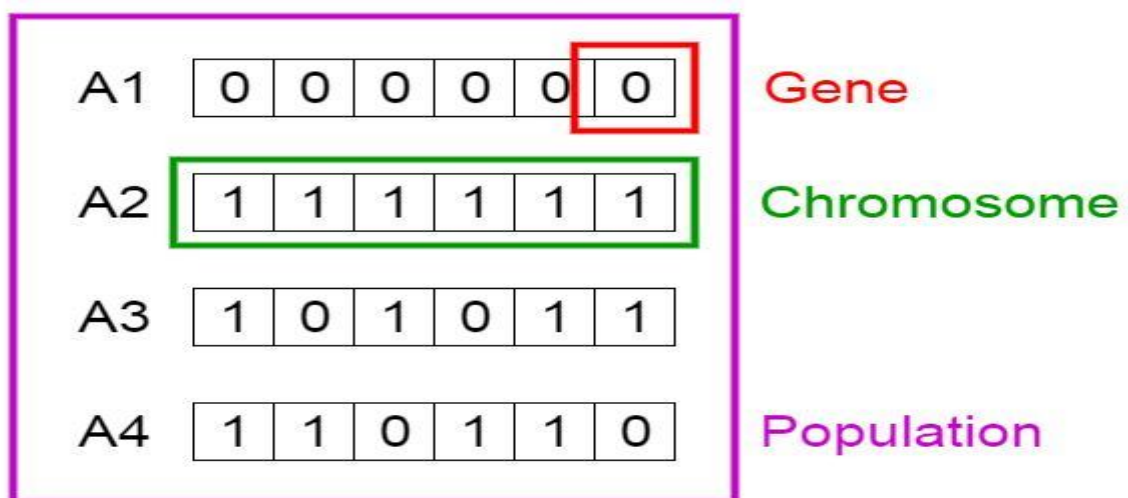


Figure 1.6: Population, Chromosomes and Genes

11.3. Fonction de Fitness

La fonction de Fitness détermine dans quelle mesure un individu est apte (la capacité d'un individu à rivaliser avec d'autres individus). Il donne un score de fitness à chaque individu. La probabilité qu'un individu soit sélectionné pour la reproduction est basée sur son score de fitness.

11.4. Sélection

L'idée de la phase de sélection est de sélectionner les individus les plus aptes et de les laisser transmettre leurs gènes à la génération suivante.

Deux paires d'individus (parents) sont sélectionnées en fonction de leurs scores de condition physique. Les personnes ayant une forme physique élevée ont plus de chances d'être sélectionnées pour la reproduction.

11.5. croisement

Le croisement est la phase la plus significative d'un algorithme génétique. Pour chaque paire de parents à accoupler, un point de croisement est choisi au hasard dans les gènes.

Par exemple, considérez que le point de croisement est 3 comme indiqué ci-dessous.

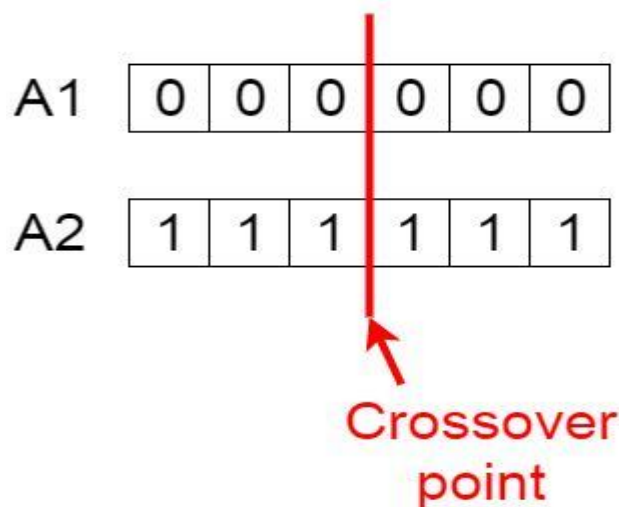


Figure 1.7 : point de croisement

La progéniture est créée en échangeant les gènes des parents entre eux jusqu'à ce que le point de croisement soit atteint.

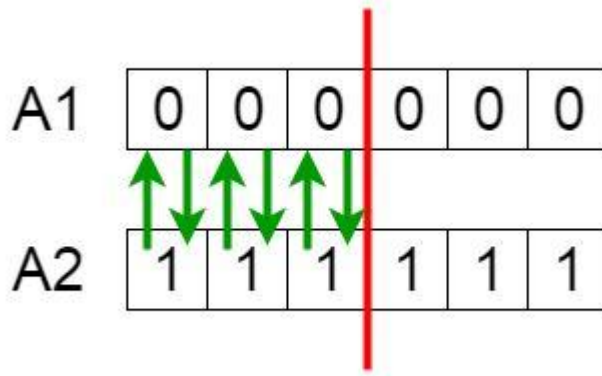


Figure 1.8: Échange de gènes entre parents

Les nouveaux descendants sont ajoutés à la population.

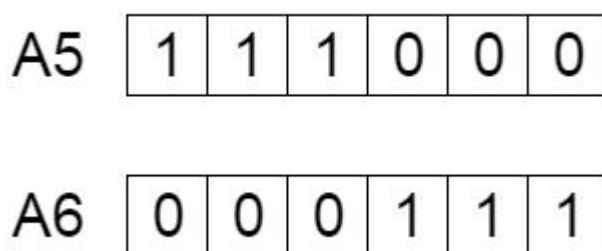
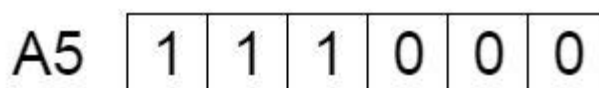


Figure 1.9 : Nouvelle progéniture

11.6. Mutation

Chez certains nouveaux descendants formés, certains de leurs gènes peuvent être soumis à une mutation avec une faible probabilité aléatoire. Cela implique que certains des bits de la chaîne de bits peuvent être retournés.

Before Mutation



After Mutation



Figure 1.10 : Mutation: avant et après

11.7. Conditions d'arrêt de l'algorithme

L'algorithme génétique utilise les options suivantes pour déterminer quand s'arrêter

- MaxGenerations - L'algorithme s'arrête lorsque le nombre de générations atteint MaxGenerations.
- MaxTime - L'algorithme s'arrête après avoir été exécuté pendant une durée en secondes égale à MaxTime.
- FitnessLimit - L'algorithme s'arrête lorsque la valeur de la fonction de fitness pour le meilleur point de la population actuelle est inférieure ou égale à FitnessLimit.
- MaxStallGenerations - L'algorithme s'arrête lorsque le changement relatif moyen de la valeur de la fonction MaxStallGenerations est inférieur à la tolérance Function.
- FunctionTolerance - L'algorithme s'exécute jusqu'à ce que le changement relatif moyen de la valeur de la fonction de fitness sur MaxStallGenerations soit inférieur à la tolérance Function.

L'algorithme s'arrête dès que l'une de ces conditions est remplie.

5. Avantages et désavantages des algorithmes génétiques

Les algorithmes génétiques présentent les avantages suivants : ce sont des méthodes robustes à l'initialisation (c'est-à-dire que leurs convergences ne dépendent pas de la valeur initiale), qui permettent de déterminer l'optimum global d'une fonctionnelle ou de s'en approcher, et qui sont parallélisables. En revanche leur inconvénient majeur réside dans le nombre important d'évaluations nécessaires et leur temps de convergence. En revanche, les méthodes déterministes convergent rapidement vers un optimum. Cependant, elles ne sont pas aussi robustes à l'initialisation que les algorithmes génétiques, ce qui n'assure pas que l'optimum déterminé est un optimum global.

6. Réseau de neurones

13.1. Définition

Un réseau de neurones artificiels ou réseau neuronal artificiel est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste, en particulier bayésien. Ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de créer des

classifications rapides , et d'autre part dans la famille des méthodes de l'intelligence artificielle auxquelles ils fournissent un mécanisme perceptif indépendant des idées propres de l'implémenter, et des informations d'entrée au raisonnement logique formel.

13.2. Fonctionnement du réseau neuronal

Un réseau neuronal s'inspire du fonctionnement des neurones biologiques et prend corps dans un ordinateur sous forme d'un algorithme. Le réseau neuronal peut se modifier lui-même en fonction des résultats de ses actions, ce qui permet l'apprentissage et la résolution de problèmes sans algorithme, donc sans programmation classique.

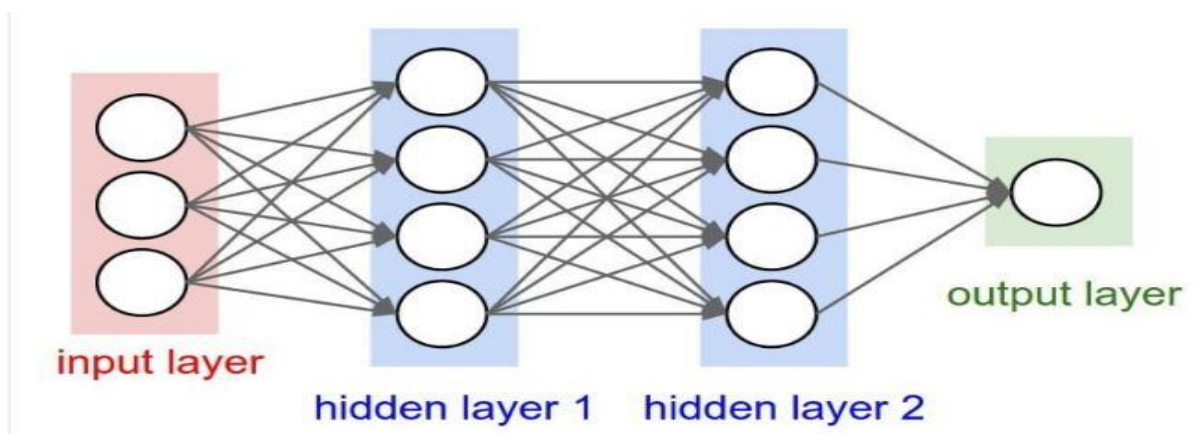


Figure 1.11 : fonctionnement le réseau de neurones artificiels(Bastien, 2019)

En règle générale, un réseau de neurones repose sur un grand nombre de processeurs opérant en parallèle et organisés en tiers. Le premier tiers reçoit les entrées d'informations brutes, un peu comme les nerfs optiques de l'être humain lorsqu'il traite des signaux visuels.

Par la suite, chaque tiers reçoit les sorties d'informations du tiers précédent. On retrouve le même processus chez l'Homme, lorsque les neurones reçoivent des signaux en provenance des neurones proches du nerf optique. Le dernier tiers, quant à lui, produit les résultats du système. (Bastien, 2019)

14.3. Comment le réseau de neurones artificiels apprend ?

Par le biais d'un algorithme, le réseau de neurones artificiels permet à l'ordinateur d'apprendre à partir de nouvelles données. L'ordinateur doté du réseau de neurones apprend à effectuer une

tâche en analysant des exemples pour s'entraîner. Ces exemples ont préalablement été étiquetés afin que le réseau puisse savoir ce dont il s'agit.

Par exemple, un réseau de neurones peut être utilisé pour apprendre à l'ordinateur à reconnaître des objets. Un grand nombre d'objets d'une même catégorie est présenté au réseau de neurones, et l'ordinateur apprend à reconnaître cet objet sur de nouvelles images en analysant les patterns récurrentes au sein des images d'exemple. Ainsi, en analysant des milliers de photos de chats, le Neural Network apprendra à reconnaître un chat sur n'importe quelle photo.

Contrairement à d'autres types d'algorithmes, les réseaux de neurones ne peuvent pas être programmés directement pour effectuer une tâche. A la manière du cerveau en développement d'un enfant, la seule instruction qu'ils ont est d'apprendre.

13.4. Les différents types de réseaux de neurones

- On distingue différents types de réseaux de neurones :
- Réseau de neurones Feedforward - Neurone artificiel: ...
- Fonction de base radiale Réseau neuronal: ...
- Réseau neuronal auto-organisateur de Kohonen: ...
- Réseau neuronal récurrent (RNN) - Mémoire à long et court terme: ...
- Réseau neuronal convolutif: ...
- Réseau neuronal modulaire

7. classification ascendante hiérarchique

14.1. Définition

est l'une d'entre elles. On cherche à ce que les individus regroupés au sein d'une même classe (homogénéité intra-classe) soient le plus semblables possibles tandis que les classes soient le plus dissemblables

14.2. Principes de la classification ascendante hiérarchique

- On commence par calculer la dissimilarité entre les N objets.
- Puis on regroupe les deux objets dont le regroupement minimise un critère d'agrégation donné, créant ainsi une classe comprenant ces deux objets.
- On calcule ensuite la dissimilarité entre cette classe et les N-2 autres objets en utilisant le critère d'agrégation. Puis on regroupe les deux objets ou classes d'objets dont le regroupement minimise le critère

On continue ainsi jusqu'à ce que tous les objets soient regroupés.

Ces regroupements successifs produisent un arbre binaire de classification (dendrogramme), dont la racine correspond à la classe regroupant l'ensemble des individus. Ce dendrogramme représente une hiérarchie de partitions. On peut alors choisir une partition en tronquant l'arbre à un niveau donné, le niveau dépendant soit des contraintes de l'utilisateur (l'utilisateur sait combien de classes il veut obtenir), soit de critères plus objectifs.

14.3. Avantages de la classification ascendante hiérarchique

La classification ascendante hiérarchique est une méthode de classification qui présente les avantages suivants :

- On travaille à partir des dissimilarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dissimilarité adapté au sujet étudié et à la nature des données.
- L'un des résultats est le dendrogramme, qui permet de visualiser le regroupement progressif des données. On peut alors se faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées.

8. Conclusion

Dans ce chapitre nous avons rappelé tous d'abord le but de la classification qui consiste à distribuer un ensemble potentiellement grand d'objets en différentes classes permettant de regrouper ces objets en sous-ensembles moins nombreux qu'eux.

Présenté ensuite les différents techniques de la classification publiées dans la littérature, puis nous avons conclu que la classification supervisée consiste à construire à partir d'une classification connue à priori d'un certain nombre d'individus, une fonction d'identification réalise un découpage de l'espace de représentation.

Comme nous avons montré que la classification non supervisée cherche à mettre en évidence des relations entre des objets en les regroupant en entités homogènes suivant une certaine mesure de similarité.

Et finalement, nous illustrons l'utilité et l'efficacité de certains algorithmes pour la segmentation et la classification automatique.

1. Introduction

Dans la première partie, nous rappelons quelques concepts de base sur l'intelligence artificielle et l'apprentissage profond.

Et nous fournirons la base de données ainsi que les différentes bibliothèques utilisées dans le développement de l'application

Dans le deuxième chapitre, nous présenterons une proposition de solution qui devrait répondre aux exigences de la classification de la parole en utilisant les étapes qu'elle comprend avec une brève explication des différentes étapes de la solution.

2. Reconnaissance vocal

La reconnaissance automatique de la parole (souvent improprement appelée reconnaissance vocale) est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

La reconnaissance de la parole, ainsi que la synthèse de la parole, l'identification du locuteur ou la vérification du locuteur, font partie des techniques de traitement de la parole. Ces techniques permettent notamment de réaliser des interfaces homme-machine (IHM) où une partie de l'interaction se fait à la voix : « interfaces vocales ».

3. Intelligence artificiel

L'intelligence artificielle (IA) est « l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence »

Elle correspond donc à un ensemble de concepts et de technologies plus qu'à une discipline autonome constituée. D'autres, remarquant la définition peu précise de l'IA, introduisent ce sujet comme « le grand mythe de notre temps »

Souvent classée dans le groupe des sciences cognitives, elle fait appel à la neurobiologie computationnelle (particulièrement aux réseaux neuronaux), à la logique mathématique (partie des mathématiques et de la philosophie) et à l'informatique. Elle recherche des méthodes de résolution de problèmes à forte complexité logique ou algorithmique. Par extension elle désigne,

dans le langage courant, les dispositifs imitant ou remplaçant l'homme dans certaines mises en œuvre de ses fonctions cognitives

4. Apprentissage profond

Le deep Learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine Learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

L'apprentissage profond s'applique à divers secteurs des NTIC, notamment :

- ✓ La reconnaissance visuelle par exemple d'un panneau de signalisation par un robot ou une voiture autonome
- ✓ La robotique
- ✓ La reconnaissance ou la comparaison de formes
- ✓ La sécurité
- ✓ La santé
- ✓ La pédagogie assistée par l'informatique
- ✓ L'art
- ✓ La traduction

5. Fonctionnement du deep Learning

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

6. Architecture du solution proposer

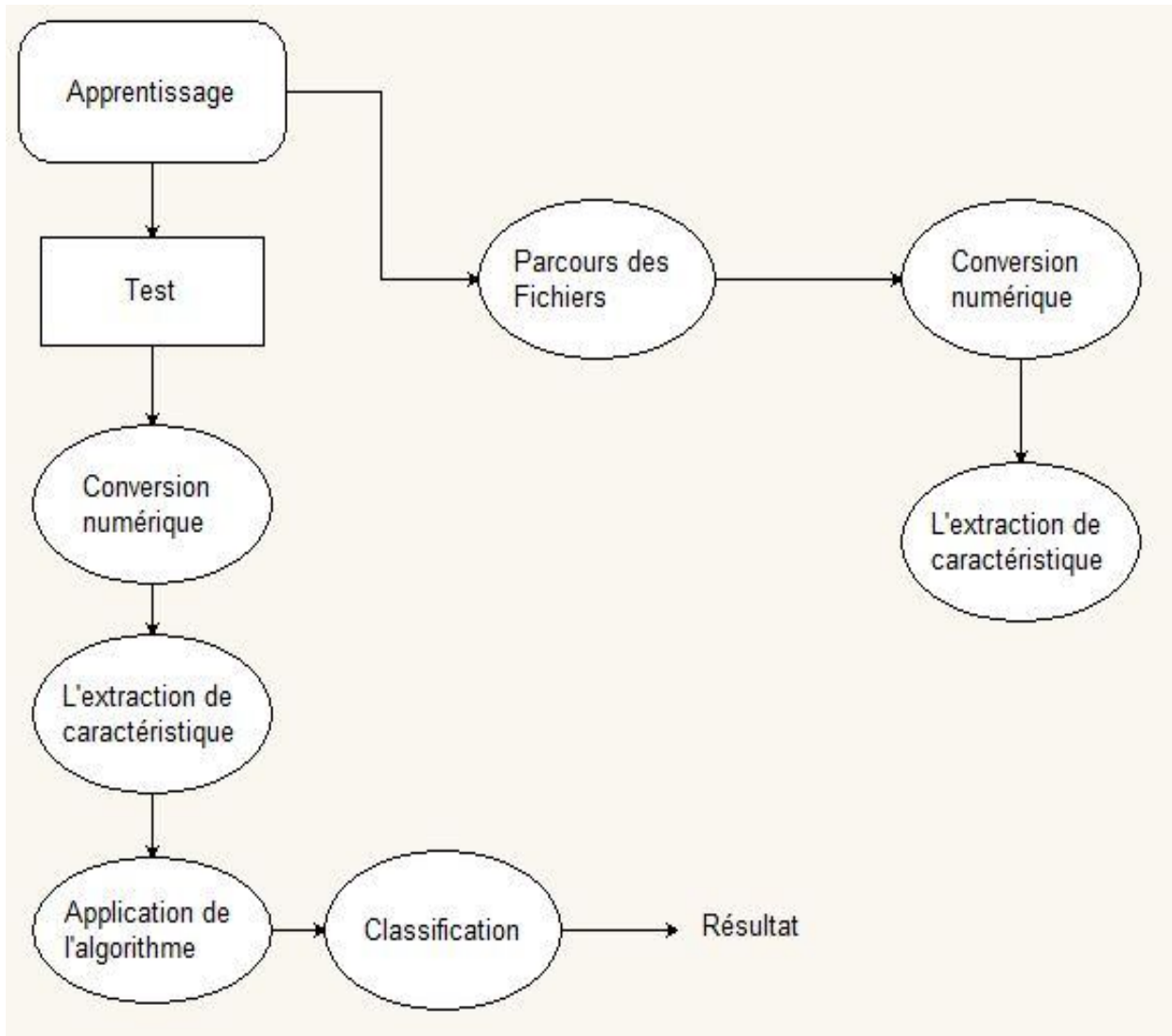


Figure 2.1 : Architecture du système proposé

Pour commencer, nous avons une base de données de fichiers : 0, 1, 2, 3 ...

Ensuite, l'extraction des caractéristiques de chacun des fichiers de la base de données et le résultat sera un ensemble de valeurs. Puis, les fichiers sont normalisés (toutes les valeurs sont comprises entre 0 et 1) et il existe une méthodologie qui effectue le processus de normalisation, ou par la formule suivante :

$$X_{normalisé} = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

Et puis toutes les anciennes valeurs (avant normalisation) sont remplacées par les nouvelles valeurs (après normalisation). Alors maintenant, nous avons une nouvelle base de données

Ce dernier est divisé en deux parties :

La première partie 30% : pour l'apprentissage

La deuxième partie 70 % : pour le test

Après l'apprentissage, l'application effectue le test (l'application de l'algorithme) puis calcule la fitness

7. Construction de la base de donnée

Le projet nécessite une de données et contient de nombreux échantillons, et pour créer la base de données nécessaire pour terminer le projet, nous avons recherché 10 personnes et demandé à chacun de lire les nombres de 0 à 9, et nous les avons enregistrés jusqu'à ce que nous obtenions 100 votes enregistrés

Ensuite, nous avons couper le temps supplémentaire (qui ne contient pas d'enregistrement) sur chaque enregistrement et converti le format de MP3 en WAV

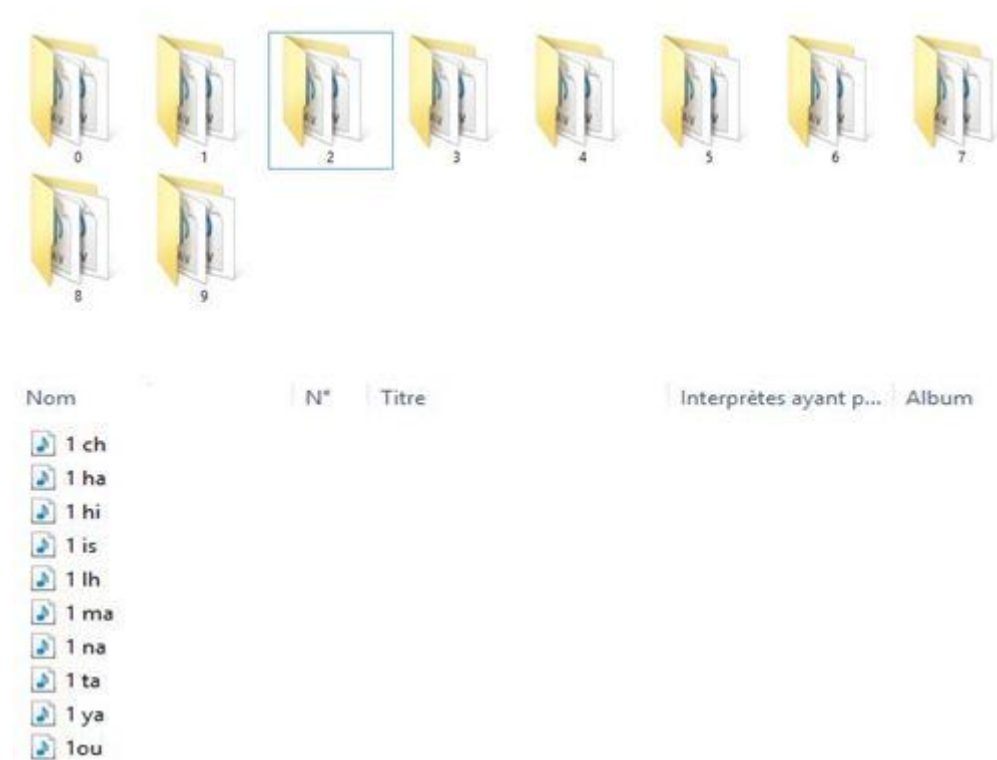


Figure 2. 2 : La base de données



Figure 2.3 : la coupure et la conversion de vocal

Mais pour développer notre programme et rendre le résultat plus précis, nous avons utilisé une deuxième base de données plus grande contenant 3000 fichiers audio : 6 personnes prononcent les nombres de 0 à 9, et chaque personne prononce chaque nombre de 49 manières différentes.

| | | | | | | | | |
|-------------|-------------|--------------|--------------|------------|-------------|--------------|--------------|-----------|
| 0_george_0 | 0_george_26 | 0_jackson_2 | 0_jackson_28 | 0_lucas_4 | 0_lucas_30 | 0_nicolas_6 | 0_nicolas_32 | 0_theo_8 |
| 0_george_1 | 0_george_27 | 0_jackson_3 | 0_jackson_29 | 0_lucas_5 | 0_lucas_31 | 0_nicolas_7 | 0_nicolas_33 | 0_theo_9 |
| 0_george_2 | 0_george_28 | 0_jackson_4 | 0_jackson_30 | 0_lucas_6 | 0_lucas_32 | 0_nicolas_8 | 0_nicolas_34 | 0_theo_10 |
| 0_george_3 | 0_george_29 | 0_jackson_5 | 0_jackson_31 | 0_lucas_7 | 0_lucas_33 | 0_nicolas_9 | 0_nicolas_35 | 0_theo_11 |
| 0_george_4 | 0_george_30 | 0_jackson_6 | 0_jackson_32 | 0_lucas_8 | 0_lucas_34 | 0_nicolas_10 | 0_nicolas_36 | 0_theo_12 |
| 0_george_5 | 0_george_31 | 0_jackson_7 | 0_jackson_33 | 0_lucas_9 | 0_lucas_35 | 0_nicolas_11 | 0_nicolas_37 | 0_theo_13 |
| 0_george_6 | 0_george_32 | 0_jackson_8 | 0_jackson_34 | 0_lucas_10 | 0_lucas_36 | 0_nicolas_12 | 0_nicolas_38 | 0_theo_14 |
| 0_george_7 | 0_george_33 | 0_jackson_9 | 0_jackson_35 | 0_lucas_11 | 0_lucas_37 | 0_nicolas_13 | 0_nicolas_39 | 0_theo_15 |
| 0_george_8 | 0_george_34 | 0_jackson_10 | 0_jackson_36 | 0_lucas_12 | 0_lucas_38 | 0_nicolas_14 | 0_nicolas_40 | 0_theo_16 |
| 0_george_9 | 0_george_35 | 0_jackson_11 | 0_jackson_37 | 0_lucas_13 | 0_lucas_39 | 0_nicolas_15 | 0_nicolas_41 | 0_theo_17 |
| 0_george_10 | 0_george_36 | 0_jackson_12 | 0_jackson_38 | 0_lucas_14 | 0_lucas_40 | 0_nicolas_16 | 0_nicolas_42 | 0_theo_18 |
| 0_george_11 | 0_george_37 | 0_jackson_13 | 0_jackson_39 | 0_lucas_15 | 0_lucas_41 | 0_nicolas_17 | 0_nicolas_43 | 0_theo_19 |
| 0_george_12 | 0_george_38 | 0_jackson_14 | 0_jackson_40 | 0_lucas_16 | 0_lucas_42 | 0_nicolas_18 | 0_nicolas_44 | 0_theo_20 |
| 0_george_13 | 0_george_39 | 0_jackson_15 | 0_jackson_41 | 0_lucas_17 | 0_lucas_43 | 0_nicolas_19 | 0_nicolas_45 | 0_theo_21 |
| 0_george_14 | 0_george_40 | 0_jackson_16 | 0_jackson_42 | 0_lucas_18 | 0_lucas_44 | 0_nicolas_20 | 0_nicolas_46 | 0_theo_22 |
| 0_george_15 | 0_george_41 | 0_jackson_17 | 0_jackson_43 | 0_lucas_19 | 0_lucas_45 | 0_nicolas_21 | 0_nicolas_47 | 0_theo_23 |
| 0_george_16 | 0_george_42 | 0_jackson_18 | 0_jackson_44 | 0_lucas_20 | 0_lucas_46 | 0_nicolas_22 | 0_nicolas_48 | 0_theo_24 |
| 0_george_17 | 0_george_43 | 0_jackson_19 | 0_jackson_45 | 0_lucas_21 | 0_lucas_47 | 0_nicolas_23 | 0_nicolas_49 | 0_theo_25 |
| 0_george_18 | 0_george_44 | 0_jackson_20 | 0_jackson_46 | 0_lucas_22 | 0_lucas_48 | 0_nicolas_24 | 0_theo_0 | 0_theo_26 |
| 0_george_19 | 0_george_45 | 0_jackson_21 | 0_jackson_47 | 0_lucas_23 | 0_lucas_49 | 0_nicolas_25 | 0_theo_1 | 0_theo_27 |
| 0_george_20 | 0_george_46 | 0_jackson_22 | 0_jackson_48 | 0_lucas_24 | 0_nicolas_0 | 0_nicolas_26 | 0_theo_2 | 0_theo_28 |
| 0_george_21 | 0_george_47 | 0_jackson_23 | 0_jackson_49 | 0_lucas_25 | 0_nicolas_1 | 0_nicolas_27 | 0_theo_3 | 0_theo_29 |
| 0_george_22 | 0_george_48 | 0_jackson_24 | 0_lucas_0 | 0_lucas_26 | 0_nicolas_2 | 0_nicolas_28 | 0_theo_4 | 0_theo_30 |
| 0_george_23 | 0_george_49 | 0_jackson_25 | 0_lucas_1 | 0_lucas_27 | 0_nicolas_3 | 0_nicolas_29 | 0_theo_5 | 0_theo_31 |
| 0_george_24 | 0_jackson_0 | 0_jackson_26 | 0_lucas_2 | 0_lucas_28 | 0_nicolas_4 | 0_nicolas_30 | 0_theo_6 | 0_theo_32 |
| 0_george_25 | 0_jackson_1 | 0_jackson_27 | 0_lucas_3 | 0_lucas_29 | 0_nicolas_5 | 0_nicolas_31 | 0_theo_7 | 0_theo_33 |

Figure 2.4 : deuxième base de données

8. L'extraction de caractéristique

L'extraction de caractéristique sonores est l'une des étapes préliminaires à la reconnaissance vocale. L'extraction de caractéristiques est une composante du traitement de l'audio qui va souvent de pair avec la classification. En effet, pour établir une règle de classification (supervisée ou non), on se base généralement sur un ensemble de critères numériques décrivant l'objet ou le phénomène observé (Zubiolo, 2015)

9. L'algorithme génétique proposer

La première étape de l'algorithme est l'initialisation des populations : on va donner comme paramètre : la taille de chaque chromosome, le nombre de chromosome de chaque population (20 chromosome), et le pourcentage de combien de Colonnes initialiser à faux (0.3 false).

L'algorithme va initialiser un tableau de taille fixe (taille de chromosome) et toutes les valeurs sont vraies. Ensuite, le tableau est divisé comme suit : de 0 jusqu'à [le pourcentage de combien de Colonnes initialiser faux (0.3 false) * la taille de chaque chromosome], par exemple : le pourcentage est 0.3 et la taille est 100 donc ($100 * 0.3 = 30$) alors les premiers 30 deviennent faux, ensuite ils sont disposés dans un ordre aléatoire. Après, il continue de créer des chromosomes jusqu'à ce qu'il atteigne 20 (la taille de la population).

```
def init_population(size, population_count=20, proba=.3):
    res = []
    for _ in range(population_count):
        chromo = np.ones(size, dtype=np.bool)
        chromo[:int(proba*size)] = False
        np.random.shuffle(chromo)
        res.append(chromo)
    return res
```

Figure 2.5 : Initialisation de la population

La deuxième étape est le croisement de chromosomes : on va donner comme paramètre : la population et meilleurs chromosome, Ensuite, deux chromosomes sont prises par exemple de population numéro 1.

On applique le croisement de 2 points : La position de croisement dans le premier chromosome commence du point numéro 1 au point numéro -2 et la coupure du deuxième chromosome commence de la position de croisement du premier chromosome jusqu'au point -1 puis il fait la mutation.

```

def cross(population,best_chromosome=None):
    population_count = len(population)
    res=[]
    if best_chromosome is not None:
        res.append(best_chromosome)
        population_count = population_count - 1
    for i in range(population_count):
        chromo = population[i]
        cross_position1 = random.randint(1,len(chromo)-2)
        cross_position2 = random.randint(cross_position1,len(chromo)-1)
        chromo[cross_position1:cross_position2] = population[(i+1)%population_count][cross_position1:cross_position2]
        res.append(mutate(chromo))
    return res

```

Figure 2.6 : Croisement de chromosomes

La troisième étape est la mutation : on va donner comme paramètre le chromosome et la probabilité de mutation 0.1(10%) ensuite, faire un parcours de toutes les cellules de chromosome et gérer un nombre aléatoire entre 0 et 1, après faire comparaison de tous les valeurs avec la probabilité de mutation, si est inférieur ou égale la probabilité il va les changer de 0 à 1 ou de 1 à 0.

```

def mutate(chromo,mutation_proba=.1):
    res = chromo
    for i in range(len(chromo)):
        r = random.random()
        if r <= mutation_proba:
            res[i] = not(res[i])
    return res

```

Figure 2.7 : la mutation

10. Test et résultat

Après avoir faire l'apprentissage et le test, L'application calcule la fitness : en fixant cette dernière avec une valeur de 0,9.Nous avons essayé de changer à chaque fois le nombre de générations (Chaque génération contient 20 chromosomes) et observer l'évolution des résultats :

| Nombre de Séquences | Nombre de Génération | meilleure Fitness |
|---------------------|----------------------|--------------------|
| 1 | 10 | 0.3662597114317425 |

| | | |
|---|----|--------------------|
| 2 | 20 | 0.3729189789123196 |
| 3 | 30 | 0.3729189789123196 |
| 4 | 40 | 0.3751387347391787 |
| 5 | 50 | 0.3751387347391787 |
| 6 | 60 | 0.3751387347391787 |
| 7 | 70 | 0.3751387347391787 |

Les résultats sont représentés dans le diagramme suivant :

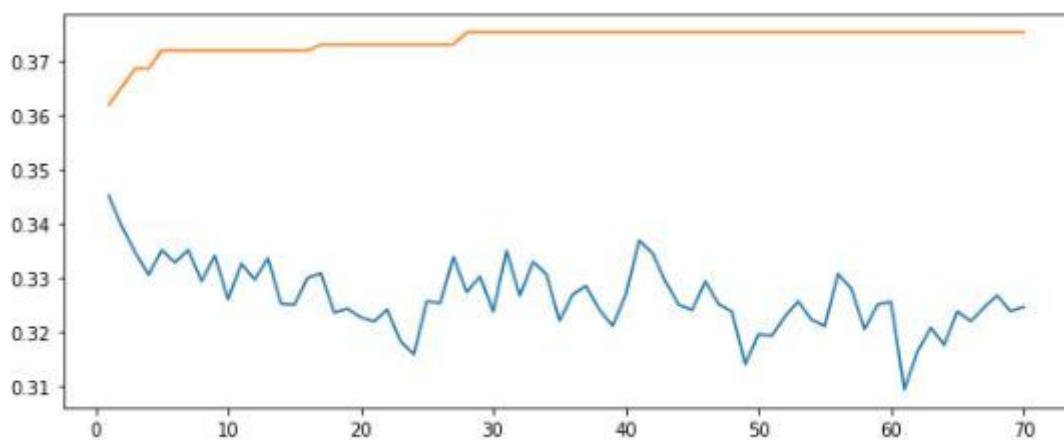


Figure 2.8 : Diagramme d'évolution des résultats

Cette figure présente le diagramme d'évolution des résultats de teste

- le diagramme en Blue est la moyenned'évolution des résultats de toutes les générations
- le diagramme en rouge est la moyenned'évolution de meilleure génération

11. Conclusion

Dans ce chapitre, la solution proposée et la méthode utilisée sont identifiées et modélisées au moyen d'un diagramme, et chaque étape a été identifiée et expliquée. Avec la présentation de l'algorithme génétique proposé et une explication simplifiée, Nous avons testé l'application et les résultats sont représentés dans un tableau.

Dans le chapitre suivant, nous présenterons l'implémentation de notre système.

Chapitre 3 : Implémentation et Réalisation

1. Introduction

Dans ce chapitre, on va présenter les outils utilisés pour construire notre application.

Le chapitre présent est composé de deux parties :

- La première partie est consacrée à la présentation de l'environnement de développement de l'application.
- La deuxième partie montre les principales interfaces graphiques de notre application.

2. L'environnement matériel :

Pour créer notre application, nous avons utilisé un PC de propriétés suivantes :

- ✓ Ordinateur Samsung
- ✓ Windows 8.1 professionnel
- ✓ Processeur : Intel Inside CORE i5
- ✓ RAM : 6 GO
- ✓ Système d'exploitation 64 bits

3. Langage utiliser

Python : est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Schème et Small talk

4. IDE

Un éditeur de code est un outil utilisé pour écrire et éditer du code. Ils sont généralement légers et peuvent être parfaits pour l'apprentissage. Cependant, une fois que votre programme s'agrandit, vous devez tester et déboguer votre code, c'est là que les IDE entrent en jeu.

Un IDE (Integrated Development Environment) comprend bien mieux votre code qu'un éditeur de texte. Il fournit généralement des fonctionnalités telles que l'automatisation de la construction, les tests et le débogage. Cela peut considérablement accélérer votre travail. L'inconvénient est que les IDE peuvent être compliqués à utiliser.

5. Visual Studio Code

C'est un éditeur qui ne vous gêne pas. Le cycle d'édition-construction-débugage délicieusement fluide signifie moins de temps à jouer avec votre environnement et plus de temps à exécuter vos idées. Visual Studio Code combine la simplicité d'un éditeur de code source avec des outils de développement puissants, tels que l'achèvement et le débogage de code IntelliSense.

6. Conversion analogique numérique

Un convertisseur analogique - numérique (CAN) est un dispositif électronique permettant la conversion d'un signal analogique en un signal numérique. Cette première définition pour être complète en deux autres, celles des signaux analogiques et numériques :

- Signal analogique : signal continu en temps et en amplitude.
- Signal numérique : signal échantillonné et quantifié, discret en temps et en amplitude.

7. Frame Work

Tkinter : Le paquet tkinter (interface Tk) est interface Python standard de la boîte à outils d'IUG Tk. Tk et tkinter sont disponibles sur la plupart des plates-formes Unix, ainsi que sur les systèmes Windows (Tk lui-même ne fait pas partie de Python ; il est soutenu par ActiveState).

Exécuter `python -m tkinter` depuis la ligne de commande ouvre une fenêtre de démonstration d'une interface Tk simple, vous indique que tkinter est correctement installé sur votre système et indique également quelle version de Tcl / Tk est installé ; vous pouvez donc lire la documentation Tcl / Tk spécifique à cette version.

La plupart du temps, tkinter est tout ce dont vous avez vraiment besoin, mais un certain nombre de modules supplémentaires sont également disponibles. L'interface Tk est située dans un module binaire nommé `_tkinter`. Ce module contient l'interface de bas niveau vers Tk et ne doit jamais être utilisé directement par les programmeurs d'application. Il s'agit généralement d'une bibliothèque partagée (ou DLL), mais peut dans certains cas être liée statiquement à l'interpréteur Python.

En plus du module d'interface Tk, tkinter inclut un certain nombre de modules Python, `tkinter.constants` étant l'un des plus importants. L'importation de `tkinter` importera automatiquement `tkinter.constants`, donc, généralement, pour utiliser Tkinter, tout ce dont vous avez besoin est une simple instruction d'importation:

« `import tkinter` » ou depuis « `tkinter import` » (2020)

Scikit-learn : est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python

Pandas : est une librairie python qui permet de manipuler facilement des données à analyser, manipuler des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes).ces tableaux sont appelés DataFrames, similaires aux dataframes sous R.

On peut facilement lire et écrire ces dataframes à partir ou vers un fichier tabulé.On peut également faciliter le traçage des graphes à partir de ces DataFrames grâce à matplotlib.

Pour utiliser pandas : « import pandas » (2020)

NumPy : est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

8. Présentation de l'application

Nous allons présenter dans cette section les principaux écrans de notre application.

8.1. Le menuprincipal :

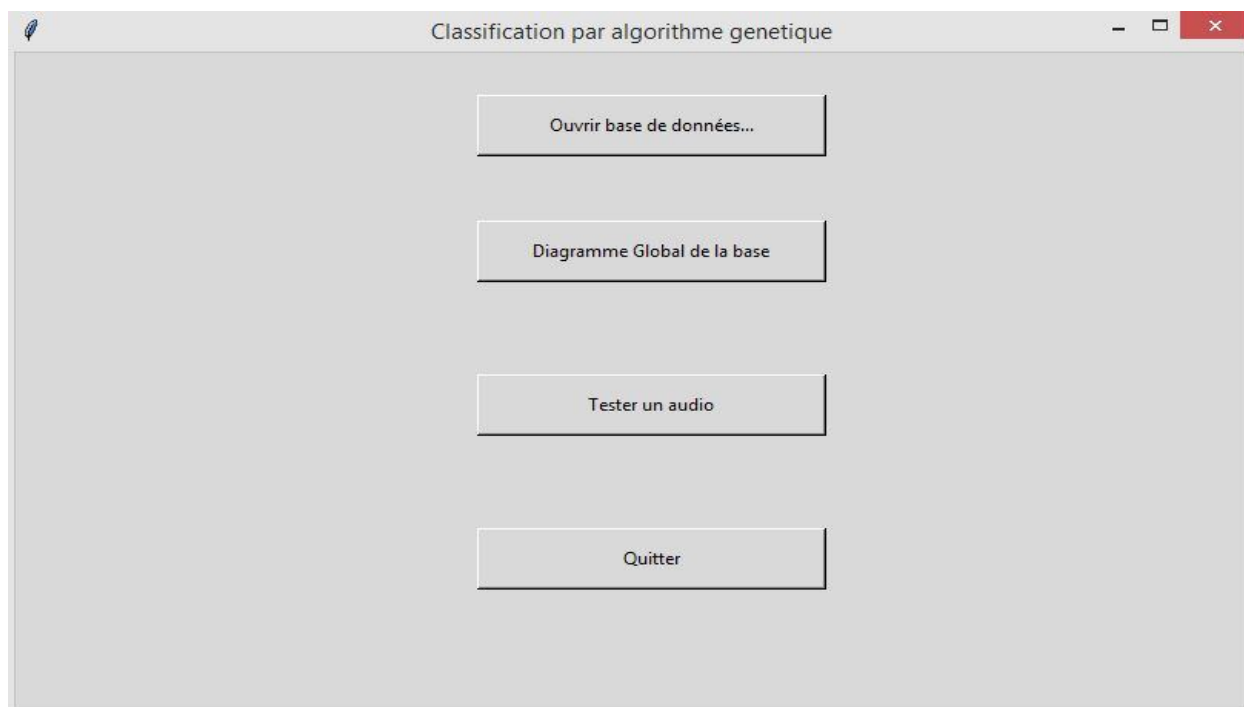


Figure 3.1: La fenêtre principale

La page principale est créé pour gérer l'application, contient 4 boutons permet à l'utilisateur d'utiliser l'application, elle lui permet l'accès à des différentes fonctionnalités comme : charger la base des données, dessiner le diagramme de la base des données, Tester un audio et un bouton pour quitter l'application.

8.2. La Sélection de la base des données

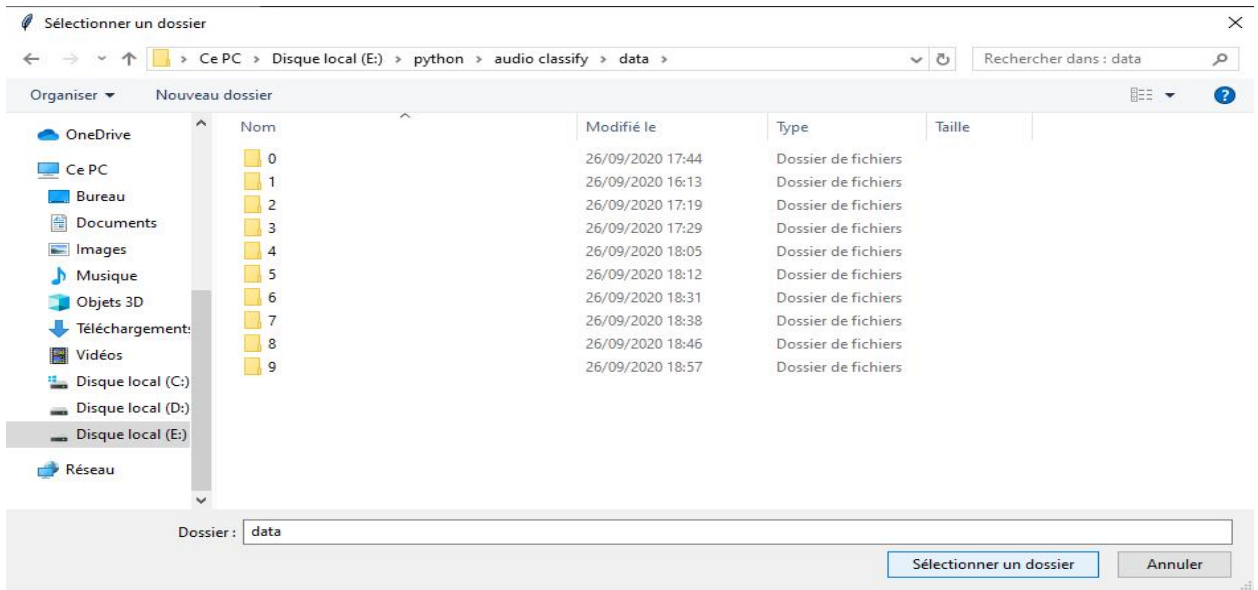


Figure 3.2: sélectionner les fichiers de la base des données.

Cette page permet d'ajouter une base de données, lorsque l'utilisateur termine la sélection de la base il clique sur le bouton terminer, la base sera charger automatiquement.

8.3. La Sélection de vocal pour le test

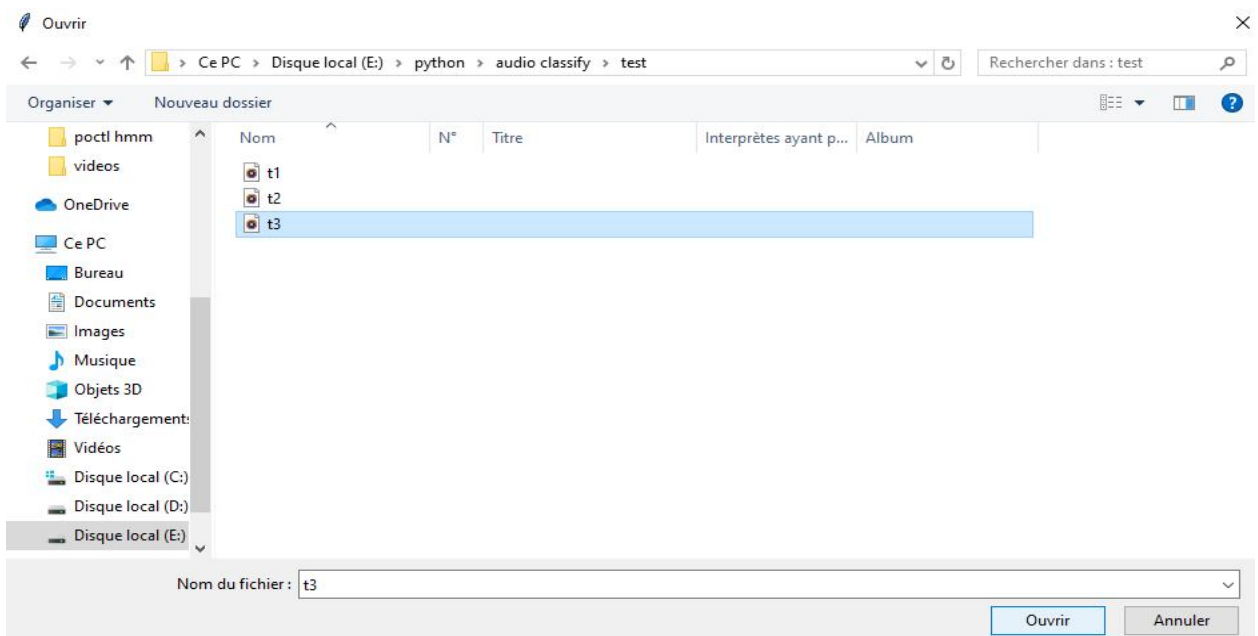


Figure 3.3 : Sélection des fichiers pour faire le test

Cette page permet d'insérer un vocal de test, lorsque l'utilisateur termine la sélection de vocal il clique sur le bouton ouvrir , le vocal sera insérer automatiquement pour faire le traitement.

8.4. Diagramme d'enregistrement audio

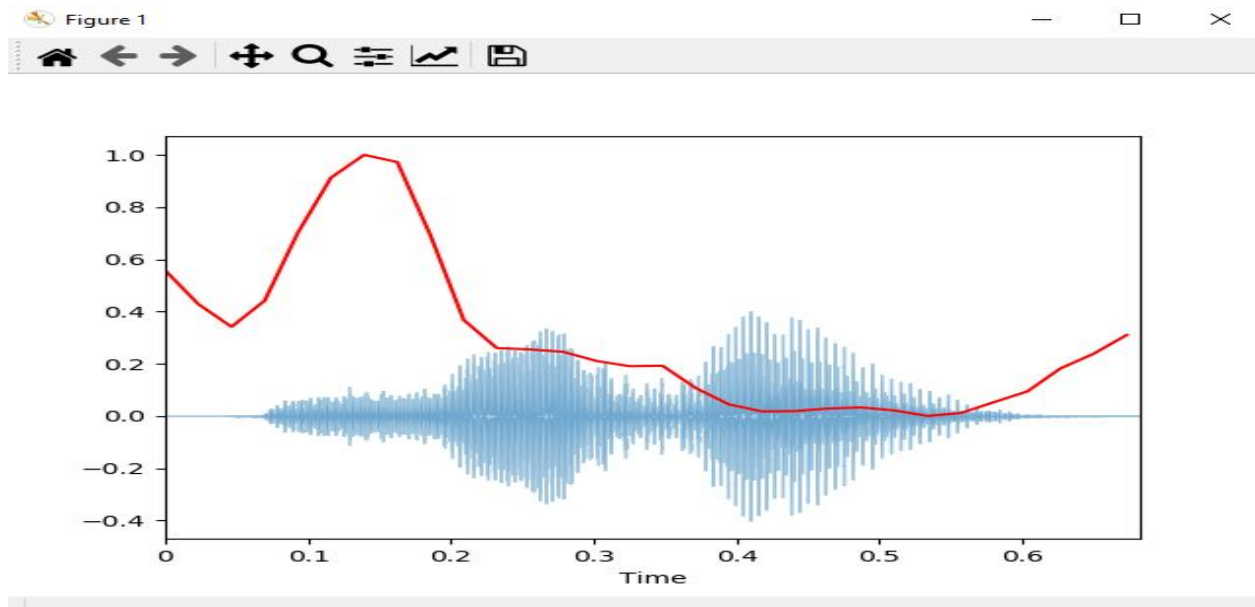


Figure 3.4 : Diagramme d'enregistrement d'un fichier audio.

Cette figure présente le diagramme de vocal avant et après l'extraction des caractéristiques

- le diagramme en bleu est avant l'extraction des caractéristiques
- le diagramme en rouge est après l'extraction des caractéristiques

8.5. Affichage du résultat

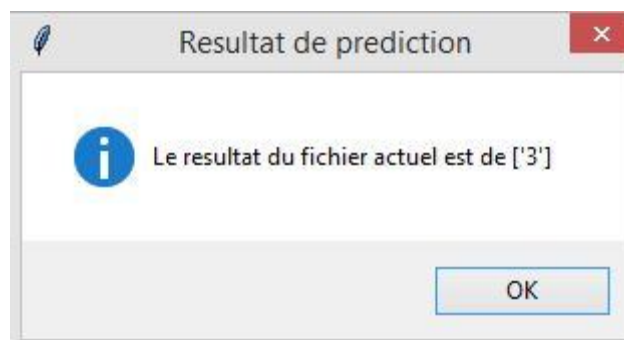


Figure 3.5 : résultat de la classification

Après le traitement du vocal et après l'application des algorithmes génétiques le résultat sera affiché comme suit :

9. Conclusion

Dans ce chapitre, nous avons expliqué notre application par des interfaces graphiques et donner une idée sur l'environnement de développement et les différents outils utilisés pour la mise en œuvre de travail.

Conclusion et Perspectives

L'objectif de ce projet de fin d'étude est le développement d'une application qui permet la classification automatique de la parole par l'application des algorithmes génétiques.

Nous avons présenté les différentes étapes de conception et de réalisation de l'application et l'outil utilisé pour la mise en œuvre de l'application (Python).

En effet, ce projet était une expérience intéressante qui nous a permis d'améliorer nos connaissances et nos compétences sur les outils nécessaires pour la conception et la programmation des applications informatiques en générale, et les algorithmes génétique en particulier.

Ces résultats seront le point de départ (les inputs) pour une prochaine amélioration de notre système de classification par exemple :

- Essaye d'utiliser des bases des données plus grandes.
- Essayez de travailler avec des méthodes de classification différentes et plus récentes afin d'améliorer la performance de notre système
- Essayez d'améliorer la performance de notre algorithme génétique proposé en l'hybridant avec d'autres techniques dans l'état de l'art.

Références

- ✓ (2020, sep 30). Retrieved from <https://docs.python.org>
- ✓ (2020, aout 21). Récupéré sur <http://python-simple.com>
- ✓ Durand, N. (2016, 03 25). Récupéré sur <https://tel.archives-ouvertes.fr/tel-01293722/document>
- ✓ L, B. (2019, 04 05). Récupéré sur <https://www.lebigdata.fr>
- ✓ RAKOTOMALALA. (2018, sep 02). Récupéré sur <http://eric.univ-lyon2.fr>
- ✓ Zubiolo, A. (2015).
- ✓ 21 aout 2020. <http://python-simple.com>.
- ✓ 30 sep 2020. <https://docs.python.org>.
- ✓ BENNANI, Younès. «Université Paris 13.» <https://lipn.univ-paris13.fr>. s.d. <https://lipn.univ-paris13.fr/~bennani/tmpc/TID/tid2.pdf>.
- ✓ Benzaki, Younes. *mrmint.fr*. 10 04 2018. <https://mrmint.fr/algorithmes-k-means>.
- ✓ CENTRE DE SUPPORT XLSTAT. s.d. <https://help.xlstat.com/s/article/choisir-une-methode-de-classification-avec-xlstat?language=fr>.
- ✓ Durand, Nicolas. 25 03 2016. <https://tel.archives-ouvertes.fr/tel-01293722/document>.
- ✓ Frederic P. Miller, Agnes F. Vandome, John McBrewster. «Reconnaissance Vocale.» Dans *Reconnaissance Vocale*, de Agnes F. Vandome, John McBrewster Frederic P. Miller, 84. Alphascript Publishing, 2010-08-17.
- ✓ Harifi, Kenza. 21 sep 2019. <https://medium.com>.
- ✓ —. *A Medium corporation [US]*. 21 11 2019. <https://medium.com/@kenzaharifi/bien-comprendre-lalgorithme-des-k-plus-proches-voisins-fonctionnement-et-implémentation-sur-r-et-a66d2d372679>.
- ✓ L, Bastien. 05 04 2019. <https://www.lebigdata.fr>.
- ✓ RAKOTOMALALA. 02 sep 2018. <http://eric.univ-lyon2.fr>.
- ✓ Sylvestre, HATCHI. *mathlasup5.fr*. 06 12 2018. <http://mathlasup5.fr/docs/manhattan.pdf>.
- ✓ *Synthèse et Reconnaissance vocale.* s.d. <https://synthesevoix.wordpress.com/reconnaissance-vocale/techniques/>.

- ✓ *Université Lumière Lyon 2 Ricco Rakotomalala.* s.d. http://eric.univ-lyon2.fr/~ricco/cours/slides/classif_centres_mobiles.pdf.
- ✓ Yassine, EL Amrani. «IEEE UAE.» *IEEE UAE STUDENT BRANCH.* s.d. <http://www.ieee.ma/uaesb/pdf/Algo-k-Moyennes.pdf>.
- ✓ Zubiolo, Alexis. 2015.

Résumé

Ce mémoire présente les différentes étapes de conception et de réalisation d'une application qui permet la classification automatique de la parole par l'application des algorithmes génétiques.

Après avoir terminé l'étude du projet, nous avons commencé avec la conception en utilisant l'algorithme génétique et la programmation en python, et enfin la concrétisation de l'application sous l'environnement de développement.

Ce projet a fait l'objet d'une expérience intéressante qui nous a permis d'améliorer nos connaissances et nos compétences dans la phase de la programmation sous le domaine de l'acoustique.

Mots clés : classification, algorithme génétique, python, parole

Abstract

This thesis presents the different stages of design and implementation of an application that allows the automatic classification of speech through the application of genetic algorithms.

After the study of the project was completed, we started with the conception using the genetic algorithm and programming in python, and finally the realization of the application under the development environment.

This project was an interesting experience that allowed us to improve our knowledge and skills in the programming phase under the acoustics field.

Keywords: classification, genetic algorithm, python, speech

ملخص

تقدم هذه الرسالة المراحل المختلفة لتصميم وتنفيذ تطبيق يسمح بالتصنيف التلقائي للكلام من خلال تطبيق الخوارزميات الجينية

بعد الانتهاء من دراسة المشروع، بدأنا بتصميم واستخدام الخوارزمية الجينية والبرمجة بلغة البايثون. وأخيرا تحقيق التطبيق في ظل بيئة التطوير.

كان هذا المشروع موضوع تجربة مثيرة للاهتمام، مما سمح لنا بتحسين معرفتنا ومهاراتنا في مجال البرمجة والجانب الصوتي.

الكلمات المفتاحية: التصنيف، الخوارزمية الجينية، البايثون، الكلام