Ministère De L'enseignement Supérieur Et De La Recherche Scientifique

Université Abdelhamid Ibn Badis - Mostaganem

Faculté Des Sciences Exactes Et D'informatique

Département De Mathématiques Et Informatique

Filière : Informatique

RAPPORT DE MINI-PROJET

Option : Ingénierie des Systèmes d'Information

Rapport de Projet de Master en Informatique

Thème:

An Improved K-means Clustering Algorithm

Etudiant(e)s :

- Korghlou Meriem.
- Bennaceur Cérine-Maroua.

Encadrant(e) : Mme Meroufel Bakhta.

Année Universitaire 2021-2022

Remerciement

Nous remercions ALLAH qui nous a aidé et nous a donné la patience et le courage, la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et l'encadrement de Mme Meroufel Bakhta, on la remercie pour la qualité de son travail exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nous ne saurions jamais assez remercier la lumière de notre vie. Un immense merci à nos chers parents. Qui par leur encouragements et prières nous ont permis de surmonter tous les obstacles qui s'opposait à nous.

Pour conclure, nous souhaitons adresser nos remerciements à l'ensemble du personnel du département de Mathématique et Informatique de la faculté des sciences exactes et informatique, aussi tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Dédicace -Meriem-

Je dédie ce travail

À mon cher père ***Houari*** *ainsi qu'à ma tendre mère qui à eux deux se sont données tout le mal pour m'assurer la meilleure vie qu'il soit.*

À mon grand frère ***Ibrahim*** *qui m'a soutenu et orienté durant ma vie étudiante.*

À ma sœur et ma deuxième moitié ***Amina*** *qui a toujours été là pour moi dans le meilleur et dans le pire.*

À mon binôme ***Marwa*** *pour son aide, sa patience et sa compréhension tout au long de ce projet.*

À mon oncle ***Belkacem*** *et toute la famille* ***Mekerta****.*

À ma famille.

À l'équipe ***Elosys****.*

Dédicace -Marwa-

Je dédie ce travail à ma chère mère ***Djamila*** *et mon cher père* ***Hanifi*** *qui ont été toujours à mes côtés pour me soutenir et m'encourager et qui n'ont jamais cessé de formuler duaa à mon égard, que dieu les garde à jamais*

À ma sœur ***Nour*** *ma source de joie et de bonheur*

À mes frères ***Walid****,* ***Hichem***

Que dieu miséricordieux les protègent

À ma meilleure amie ***Yasmine*** *pour son soutien moral et ses conseils précieux*

À mon binôme et ma meilleure amie ***Meriem*** *qui a toujours été à mes côtés, avec qui j'ai partagé ces quatre dernières années d'études et avec qui j'ai eu l'honneur de les finir, que dieu lui donne du bonheur, santé et réussite.*

À ***Amina*** *et toute sa famille* ***Korghlou***

À toute ma famille.

À tous mes amis.

À tous ceux que j'aime.

À tous ceux qui m'aiment.

Résumé

La classification est une discipline particulière du Machine Learning qui ne cesse de s'améliorer en s'intégrant dans tous les domaines qui font appel à la statistique, son objectif est de séparer le jeu de donnée en groupes homogènes ayant des caractéristiques communes.

Dans ce projet nous nous intéressons à l'algorithme de k-means clustering.

Mots-clés : Intelligence Artificielle IA, Apprentissage Automatique AA, dataset, k-means, anomalies, centroïdes, cluster, chevauchement, densité.

Abstract

Classification is a particular discipline of Machine Learning which is constantly improving by integrating into all fields that use statistics, its objective is to separate the data set into homogeneous groups with common characteristics.

In this project we are interested in the k-means clustering algorithm.

Keywords: Artificial Intelligence AI, Machine Learning ML, dataset, k-means, outliers, centroids, cluster, density, overlap.

Table des figures

Figure 1 Les sous-ensembles de l'intelligence artificielle 14
Figure 2 les catégories de l'apprentissage automatique 16
Figure 3 Modèle du SVM 17
Figure 4 Les étapes de l'algorithme KNN 18
Figure 5 Organigramme de l'algorithme KNN 18
Figure 6 Différence entre la régression linéaire et la régression logistique 20
Figure 7 Différence entre la Classification et la Régression 20
Figure 8 Les types de clustering 21
Figure 9 Dataset avant et après l'application de k-means 22
Figure 10 Algorithme de k-médoïdes 23
Figure 11 Organigramme de l'algorithme du clustering hiérarchique 24
Figure 12 Dendrogramme du clustering hiérarchique 24
Figure 13 Dataset avant l'algorithme de PCA 25
Figure 14 Dataset après l'algorithme de PCA 25
Figure 15 Apprentissage par renforcement (agent et son environnement) 26
Figure 16 le fonctionnement du Q-learning 26
Figure 17 Le fonctionnement de DQN 27
Figure 18 Le fonctionnement de SARSA 27
Figure 19 Les étapes de l'apprentissage machine 29
Figure 20 La distance euclidienne 33
Figure 21 La distance de Manhattan 33
Figure 22 La similitude Cosine 33
Figure 23 Les centroïdes initiaux 34
Figure 24 distance entre les centroïdes et les autres points 34
Figure 25 les centres de gravités après l'optimisation 35
Figure 26 désigner les nouveau centroïdes 35
Figure 27 Le déroulement de l'algorithme K-means 36
Figure 28 Méthode de Elbow 37
Figure 29 Analyse de silhouette 38
Figure 30 Choix des centroïdes 40
Figure 31 Les anomalies 42
Figure 32 Avant et après l'application de k-means 43
Figure 33 Segmentation de la clientèle 43
Figure 34 application de k-means sur un dataset complexe 44
Figure 35 les anomalies dans l'algorithme de k-means 47
Figure 36 Anomalie 47
Figure 37 Anomalie externe 48
Figure 38 Algorithme Isolation Forest, Etape 1 49
Figure 39 Algorithme Isolation Forest, Etape 2 49
Figure 40 Algorithme Isolation Forest, Etape 3 49
Figure 41 Algorithme Isolation Forest, Etape 4 50

Figure 42 Calcul de diagonale 50
Figure 43 Calcul d'alpha 50
Figure 44 Calcul du des voisin 51
Figure 45 calcul de distances 51
Figure 46 Centroïdes éloignés les uns des autres 52
Figure 47 chevauchement 52
Figure 48 chevauchement de deux clusters 53
Figure 49 Clusters non chevauchés 54
Figure 50 Matrice de confusion 55
Figure 51 analyse de silhouette 56
Figure 52 Détection des anomalies 60
Figure 53 Calcul de la diagonale et la valeur alpha 60
Figure 54 Calcul de densité 61
Figure 55 Désignation des deux premiers centroïdes 62
Figure 56 Les trois centroïdes initiaux du dataset 62
Figure 57 Application de k-means 63
Figure 58 Visualisation de la nouvelle dataset avec des anomalies 63
Figure 59 K-means Classique face aux anomalies 64
Figure 60 La nouvelle approche face aux anomalies 64
Figure 61 application de la nouvelle méthode en laissant les anomalies 64
Figure 62 Comparaison entre les inerties 65
Figure 63 Comparaison de temps d'exécution 65

Table des matières

Table des figures 6
Table des matières 8
Introduction Générale 11
Chapitre 1 Machine Learning 13
1. Introduction 14
2. Intelligence artificielle IA 14
3. Apprentissage automatique 15
3.1 Catégorie de l'apprentissage automatique 15
I. Apprentissage supervisé 16
I.a Classification 16
I.a.1 Support Vector Machine (SVM) 17
I.a.2 K-plus proche voisins 17
I.b La Régression 18
I.b.1 Régression linéaire 18
I.b.2 Régression Logistique 19
II. Apprentissage non supervisé 20
II.a Clustering 21
II.a.1 Clustering partitionnel 21
II.a.2 Clustering hiérarchique 23
II.b Réduction de dimension 24
II.b.1 Principal component analysis (PCA) 24
III. Apprentissage par renforcement 25
III.a Les algorithmes d'apprentissage par renforcement 26
III.a.1 Q-Learning 26
III.a.2 DQN 27
III.a.3 SARSA 27
3.2 Les étapes du machine Learning 28
3.3 Limite de l'apprentissage automatique 29
4. Conclusion 30
Chapitre 2 k-means clustering 31
1. Introduction 32
2. Algorithme k-means 32

3. Métrique de distance 32
3.1 Distance euclidienne 33
3.2 Distance de Manhattan 33
3.3 Similitude cosinus 33
4. Les étapes de k-means 34
5. Limites du k-means 36
5.1 Comment choisir le nombre de cluster 36
5.1.1 Méthode de Elbow 37
5.1.2 Analyse de silhouette 37
5.2 Le choix des centroïdes initiaux 39
5.2.1 A base de tâtonnement 39
5.2.2 K-means ++ 40
5.2.3 A base de poids 41
5.3 Détection des anomalies 41
6. Domaine d'Applications de K-Means 43
7. Désavantages de K-means 44
8. Conclusion 44
Chapitre 3 Nouvelle approche de k-means 45
1. Introduction 46
2. Algorithme de la nouvelle approche 46
2.1 Désignation des centroïdes initiaux dans l'algorithme classique 46
2.2 Les étapes de DDK-means 47
I. Étape 1 : Détection d'anomalies 47
I.a Anomalie externe "global" 47
I.b Isolation Forest 48
II. Étape 2 : calcul de diagonale 50
III. Étape 3 : calcul d'alpha (α) 50
IV. Étape 4 : calcul de la densité (D) 51
V. Étape 5 : calcul de chevauchement intra cluster 51
VI. Étape 6 : étape finale 53
3. Algorithme de DDK-means 53
4. Matrice de confusion 54
5. Analyse de silhouette 56

6. Conclusion 56
Chapitre 4 L'implémentation 57
1. Introduction 58
2. Outils utilisés 58
3. Implémentation de DDK-meanse ..... **Error! Bookmark not defined.**
3.1 Détection des anomalies 58
3.2 Calcul de diagonale et alpha 59
3.3 Calcul de la densité 60
3.4 Choix des deux premiers points 60
3.5 Application de k-means 62
4. Analyse et comparaison 62
4.1 Comparaison inertie 64
4.2 Comparaison temps d'exécution 64
5. Conclusion 65
Conclusion Générale 66
Références 68

Introduction Générale

De nos jours, l'intelligence artificielle a pu se mettre au rang des indispensables et ceux grâce à ses algorithmes ces derniers sont plus communément appelés algorithmes d'apprentissage automatique.

En peu de temps l'IA est devenue le meilleur ami de l'homme, elle s'incruste dans notre vie et décide ainsi de nos actes les plus anodins jusqu'à nos réelles décisions.

Les algorithmes qu'englobe l'apprentissage automatique n'ont en aucun cas pour but d'acquérir des connaissances déjà formalisées cependant leur travail est essentiellement basé sur la compréhension de la structure des données pour ensuite les intégrer dans des modèles comme ceux de la classification, ces modèles permettent de créer des classes de données partageant les mêmes caractéristiques.

Les approches des méthodes de classifications des données sont multiples, on distingue généralement les approches supervisées et non supervisées, la désignation d'une démarche se fait selon la problématique qu'on a, ainsi que le type de données qui se trouvent à notre disposition.

Les algorithmes de classification non supervisées sont souvent utilisés pour étudier des données pour lesquelles on dispose de peu de renseignements. Il existe un large éventail de méthodes dédiées à la classification non supervisée, connue aussi comme des méthodes de regroupement (clustering). La technique qui prend largement le dessus dans le clustering est le K-Means et ceux grâce à sa remarquable commodité à comprendre et à manipuler, ce qui n'empêche pas la présence des différentes limites et problèmes. Le but de notre travail est de bien comprendre le k-means et de proposer une stratégie adaptative qui améliore le k-means classique en dépassant ses limites.

Ce projet a été répartie en quatre parties capitales, la première partie va être consacré aux différentes catégories de l'apprentissage automatique en citant leurs algorithmes les plus mentionnés dans la littérature, la seconde partie va être pleinement destiné à l'algorithme de clustering k-means, dans le troisième chapitre on va découvrir l'algorithme proposé en évoquant ses étapes en détails, pour finir on va montrer l'implémentation de la méthode dans le dernier chapitre.

Chapitre 1

Machine Learning

1. Introduction

Ce chapitre va être consacré à une des branches de l'intelligence artificielle et qui est l'apprentissage automatique, on va alors découvrir en premier lieu ses grandes catégories ainsi que leurs domaines d'applications, citant par la suite les étapes nécessaires à suivre pour obtenir un modèle efficace, à la fin on montrera les limites de l'apprentissage automatique.

2. Intelligence artificielle IA

L'intelligence artificielle (*Artificial Intelligence* en anglais AI), une nouvelle technologie qui permet à la machine d'imiter voir simuler son raisonnement et son comportement à celui d'un cerveau humain en facilitant ainsi de multiples tâches quotidiennes [1], l'IA a vu le jour dans les années 1950 avec le fameux turing test [2] là où ils se sont rendus compte qu'une machine a la capacité de penser, c'est de là que l'IA s'est fait une place prépondérante dans la nouvelle science suscitant un énorme engouement et chamboulant ainsi les anciens standards du domaine informatique.

Parmi les technologies qui sont apparus à travers l'IA (comme le montre la figure 1), on trouve l'apprentissage automatique (Machine Learning), l'Apprentissage approfondie (Deep Learning) et la science des données (data science) [3] [1] [2].

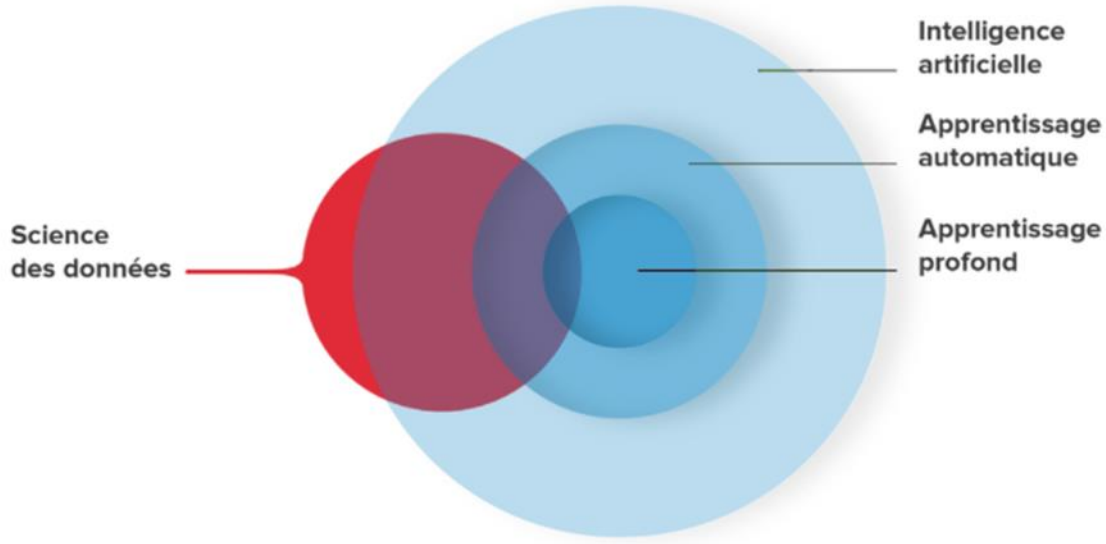

Figure 1 Les sous-ensembles de l'intelligence artificielle [4]

2.1 Domaine d'application de l'Intelligence Artificielle

- Sécurité : l'IA a été introduite dans le domaine de la sécurité là où on trouve la reconnaissance faciale, la sécurité contre les cyber-attaques, la détection de fraude … [2].

- Automobile : avec l'IA les voitures autonomes ont pu voir le jour dans un monde où la concurrence autour de l'industrie automobile bat son plein, les nouveautés sont la capacité de distinguer entre la forme d'un humain et celle d'une voiture avec la parfaite maîtrise des cartes routières [3] [2].

- Médecine : beaucoup l'ignore mais l'intelligence artificielle a permis de minimiser les représailles de la pandémie qui frappe actuellement le monde et qui est la covid-19, et ceux en comprenant ce qu'est ce nouveau virus pour ensuite réussir à le diagnostiquer, et d'arriver à prévoir son évolution et surtout de ralentir sa propagation tout en accélérant d'autres aspects de la recherche médicale.

- Toujours dans le domaine médical l'IA permet de détecter des tumeurs dans leurs stade initial, ou même de comprendre des mammographies avec une grande précision évitant ainsi le redoutable passage par la biopsie [5].

3. Apprentissage automatique

L'apprentissage automatique (machine-Learning en anglais ML) est une discipline scientifique qui représente une large poignet d'étude de l'intelligence artificielle, il offre ainsi une multitude d'algorithmes avec la possibilité de les appliquer ensuite sur des données spécifiques au domaine de l'étude concerner, ainsi ont permet à la machine d'apprendre et de prédire le futur avec l'habilitée de s'améliorer chaque jour .Par le biais de cette opération le dispositif aura la possibilité d'accomplir des tâches jusque-là impossibles à l'aide des moyens algorithmiques classiques [6] [1].

3.1 Catégorie de l'apprentissage automatique

En seulement quelques années le machine Learning a pu se faire une place prépondérante dans tous les domaines, de la transaction boursière au trafic aérien passant par la personnalisation du marketing, le ML est impérativement classé dans trois grandes catégories : Apprentissage supervisé, apprentissage non-supervisé et enfin apprentissage par renforcement.

Comme on peut le constater dans la figure 2 ci-dessous, l'apprentissage supervisé est constitué de deux grandes classes la régression et la classification, l'apprentissage non supervisé est aussi divisé en deux grandes parties le clustering (ou la classification non supervisé) et réduction de dimension, dans ce qui suit nous allons entamer ces points en détails [2].

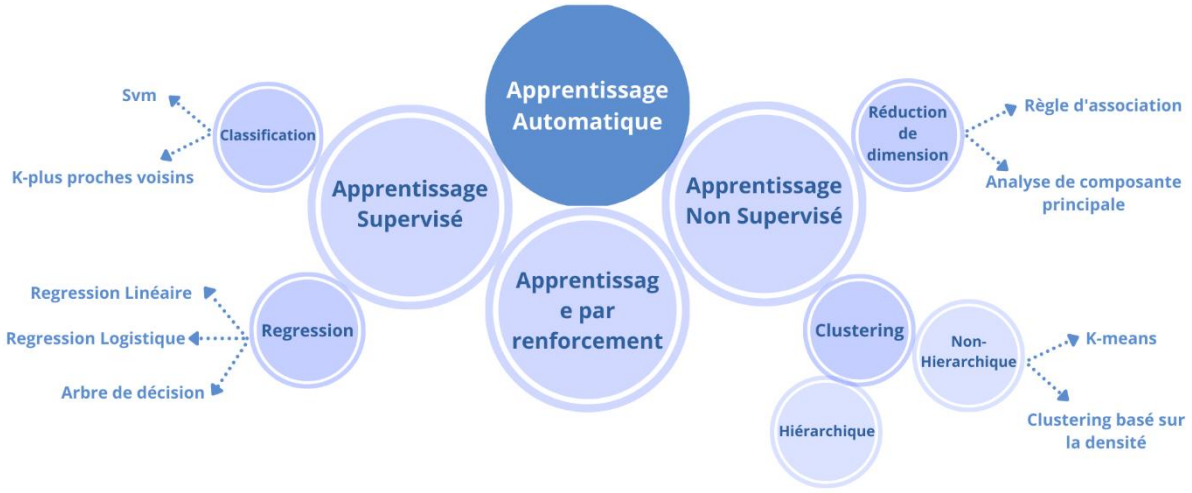

Figure 2 les catégories de l'apprentissage automatique [7]

I. Apprentissage supervisé

L'apprentissage supervisé est une approche d'apprentissage automatique utilisée dans le cas où les valeurs de sortie des échantillons étudiés sont connues au préalable, à l'aide des entrées et des sorties étiquetées, le modèle peut mesurer sa précision et apprendre au fil du temps, l'apprentissage supervisé peut se subdiviser en deux types, la classification et la régression.

I.a Classification

Comme son nom l'indique, la classification est une répartition du jeu de donnée par le modèle de classification et ceux en deux ou plusieurs classes, appliquée dans le cas où la valeur de sortie est discrète comme 'vélo', 'voiture'…, pour résoudre les problèmes de classification on dispose de plusieurs algorithmes comme le SVM mais aussi le KNN (KPP – K plus proches voisins), la classification est utilisée dans plusieurs domaines tels que :

- Domaine médical : Reconnaître si une tumeur est maligne ou bénigne.
- Sécurité : reconnaître si un mail est un spam ou pas.

I.a.1 Support Vector Machine (SVM)

Le SVM est un modèle appliqué face à un défi de classification et de régression. Cependant, dans la machine learning il est principalement utilisé dans la classification, ainsi que dans divers domaines vu la facilité à le comprendre et à l'implémenter, on site alors la bio-informatique [8], selon des recherches le svm a été utilisé lors de récente études là où ils ont découvert l'existence de plusieurs types de diabètes qui n'étaient pas connu jusqu'à là [9].

L'objectif de cet algorithme est de trouver un hyperplan optimal dans l'espace à N dimensions (N-le nombre d'entités) qui peut clairement séparer les points de données en classe distincte. Comme le montre la figure 3, on commence par identifier le bon hyperplan (la droite noire) qui sépare au mieux nos classes tout en maximisant la distance entre les points de données et l'hyperplan le plus proche [9] [10], à l'aide de ce qu'on appelle vecteur de support (les points entourés) on trace la droite bleu et rouge qui représente la marge de chaque classe.

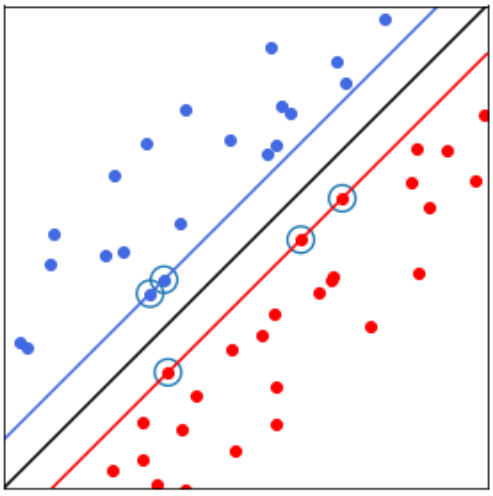

Figure 3 Modèle du SVM [11]

I.a.2 K-plus proche voisins

(K-nearest neighbors en anglais KNN) est un algorithme utilisé généralement pour résoudre les problèmes de la classification dans l'industrie en raison de sa facilité d'interpréter la variable cible mais aussi sa fiabilité de prédire le résultat avec un temps de calcul minime.

On commence par déterminer le nombre de k-voisins les plus proches qu'on doit prendre en considération, ensuite pour chaque nouveau point du dataset l'algorithme calcule la distance entre ce dernier et ses k points adjacents (voir figure 4), pour finalement le placer dans

la classe la plus fréquente parmi ces k-voisins, (la figure 5) montre le déroulement de l'algorithme [12].

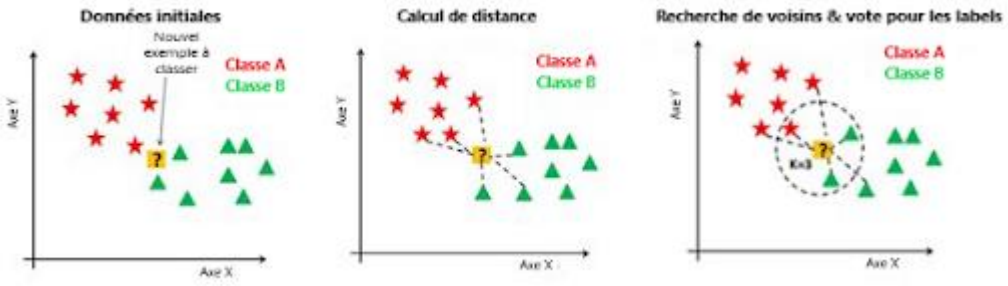

Figure 4 Les étapes de l'algorithme KNN [13]

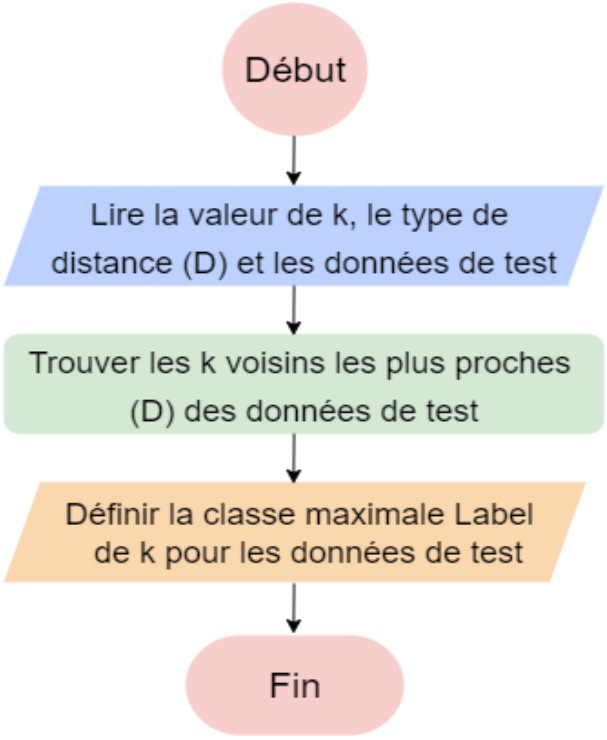

Figure 5 Organigramme de l'algorithme KNN [14]

I.b La Régression

Contrairement à la classification, la régression est utilisée lorsque la valeur de sortie est de type réel on peut la trouver dans différents secteurs comme la prédiction des biens ou de bourses, parmi ses algorithmes les plus fréquents on trouve le Régression linéaire et la Régression logistique.

I.b.1 Régression linéaire

Le modèle est adapté avec les meilleurs paramètres pour trouver la ligne la mieux ajustée entre la variable indépendante et dépendante (variable cible), la régression linéaire est employée lorsque la variable cible est de type continue (intervalle de valeur) cette dernière se distingue de deux types selon le nombre de variable indépendante, Régression linéaire simple et multiple [15] tel que :

- **La régression linéaire simple**

Dans ce modèle on trouve une seule variable indépendante et le modèle doit trouver une relation linéaire de cette dernière avec la variable cible [16].

- **La régression linéaire multiple**

Cette méthode est utilisée lorsque on a plusieurs variables indépendantes et une seule variable cible [16].

I.b.2 Régression Logistique

La régression logistique appartient à la famille d'apprentissage automatique supervisé, utilisée lorsque la variable indépendante est de nature discrète, autrement dit la régression logistique est utilisée lorsque on est face à une classification exemple on peut prédire s'il va pleuvoir ou non, cette dernière est divisée en deux clans [17].

- **Régression logistique simple**

Utilisé dans le cas où on a une seule variable indépendante [17].

- **Régression logistique multiple**

De multiples variables indépendantes sont utilisées pour prédire le résultat [17].

La différence majeure entre la régression linéaire et la régression logistique est le type de variable de sortie comme on peut l'apercevoir dans la figure 6, la régression logistique divise le résultat en deux catégories classe A ou B, tandis que la régression linéaire donne une sortie de type continue (donne une valeur réelle).

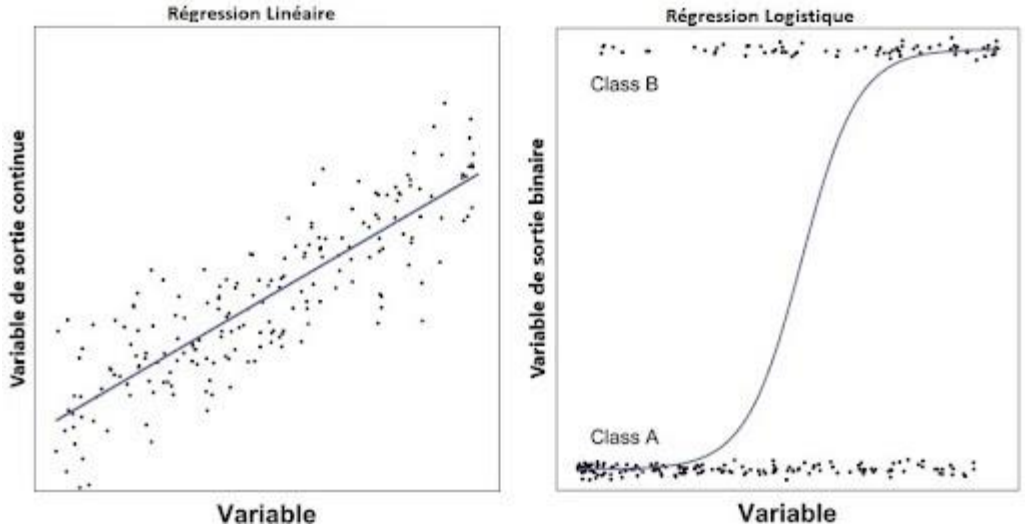

Figure 6 Différence entre la régression linéaire et la régression logistique [18]

Différence entre la régression et la classification

Prenant un exemple pour comprendre la dissimilarité entre la classification et la régression, si on souhaite prédire le prix d'une voiture, l'utilisation d'un modèle de classification nous permettrait de classifier cette voiture au sein d'une gamme de prix prédéterminés (cher, moyenne, pas cher). Par ailleurs l'utilisation d'un modèle de régression permettrait de prédire le prix exact de la voiture. La figure 7 indique que la classification divise le dataset en triangle et point (deux classes) tandis que la régression trace une droite et de là on peut prédire la valeur exacte de notre point [19].

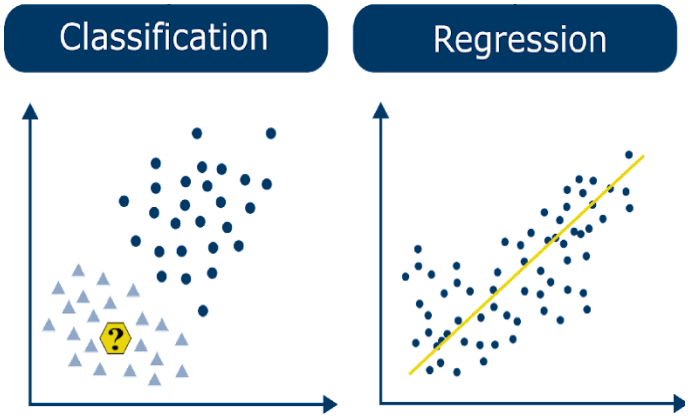

Figure 7 Différence entre la Classification et la Régression [20]

II. Apprentissage non supervisé

L'apprentissage non supervisé est appliqué dans le cas où les données ne sont pas étiquetées, dans cette démarche la machine a le pouvoir de déduire la donnée cible (target) des caractéristiques (features) en regroupant ou en schématisant les données selon leurs ressemblances avec très peu d'intervention humaine. Dans l'apprentissage non supervisé on distingue deux grands clans, le clustering et la réduction de dimension [6] [21].

II.a Clustering

C'est une technique très utilisée dans l'apprentissage non supervisé dans le dessein de regrouper les données non étiquetées d'un dataset selon leur ressemblance en classes homogènes. On retrouve alors le clustering hiérarchique et le Clustering partitionnel, qui est un partitionnement basé sur l'utilisation des centroïdes et l'exemple le plus connu est le k-means clustering dans cette même catégorie on a aussi le clustering basé sur la densité, algorithme de fuzzy…, (voir figure 8), dans ce qui suit nous allons voir tout ça en détails.

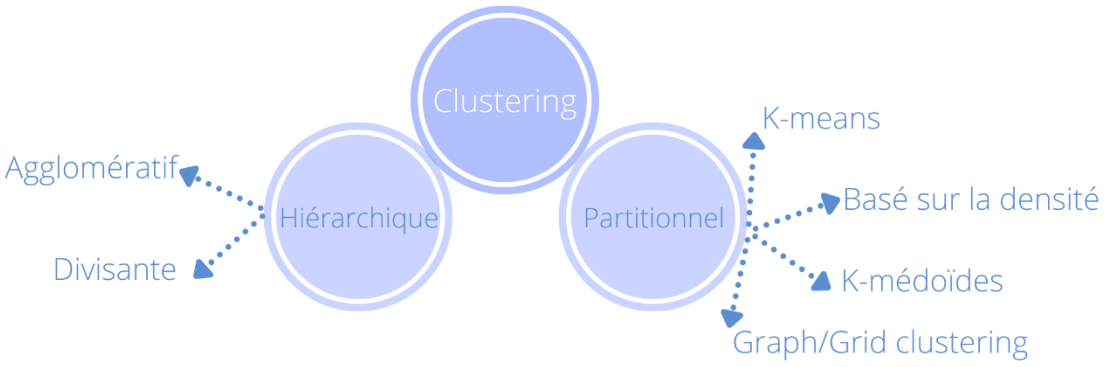

Figure 8 Les types de clustering [22]

II.a.1 Clustering partitionnel

- **K-means**

Le K-means est l'algorithme le plus connu dans l'apprentissage automatique non supervisé, il consiste à partitionner des données en classes homogènes, la figure 9 montre un dataset avant et après l'application du k-means.

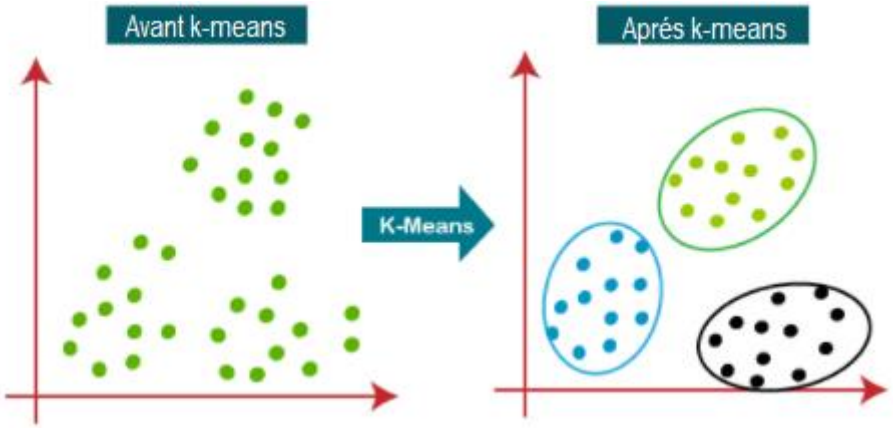

Figure 9 Dataset avant et après l'application de k-means [23]

Le k-means procède d'une manière récurrente dans chacune de ses étapes que ce soit dans le choix du nombre de cluster k, dans la désignation des centroïdes initiaux c_i, ou même dans l'élimination des anomalies du dataset, et ceux en essayant plusieurs possibilités à chaque itération pour avoir un résultat performant [12].

Le k-means a prouvé son efficacité dans le regroupement des données même faisant face à d'immense base de données (big data) [12], mais cela n'empêche ses multiples défaillances, le plus connu de tous est sa difficulté face au choix du nombre de clusters k et aussi sa sensibilité envers les variables aberrantes [12].

Le chapitre qui suit va être entièrement dédié à cet algorithme.

- **K-médoïde**

Comme nous l'avons déjà citer l'algorithme de k-means est très sensible aux valeurs aberrantes et aux valeurs nulles, c'est pourquoi il est considéré moins performant par rapport au k-médoïde, il a le même principe que le k-means à la seule différence on utilise des objets représentatifs situé au centre d'un cluster appelés médoïdes à la place des centroïdes pour minimiser la somme des dissemblances dans le but de réduire le bruit et les valeurs aberrantes [24].

Pour appliquer l'algorithme de k-médoïde il nous faut un dataset avec n éléments, et le nombre de cluster à former k, pour procéder comme suite [24] :

a. Sélectionné K points à partir du dataset comme des médoïdes initiaux.

b. Assigner chaque point de l'échantillon à un cluster selon le médoïde le plus proches.

c. Remplacer les médoïdes par d'autres points du dataset en minimisant le coût total (somme des distances au médoïdes le plus proches).

d. Répétez les étapes 2 et 3 jusqu'à ce qu'il n'y ait plus de changement dans les médoïdes.

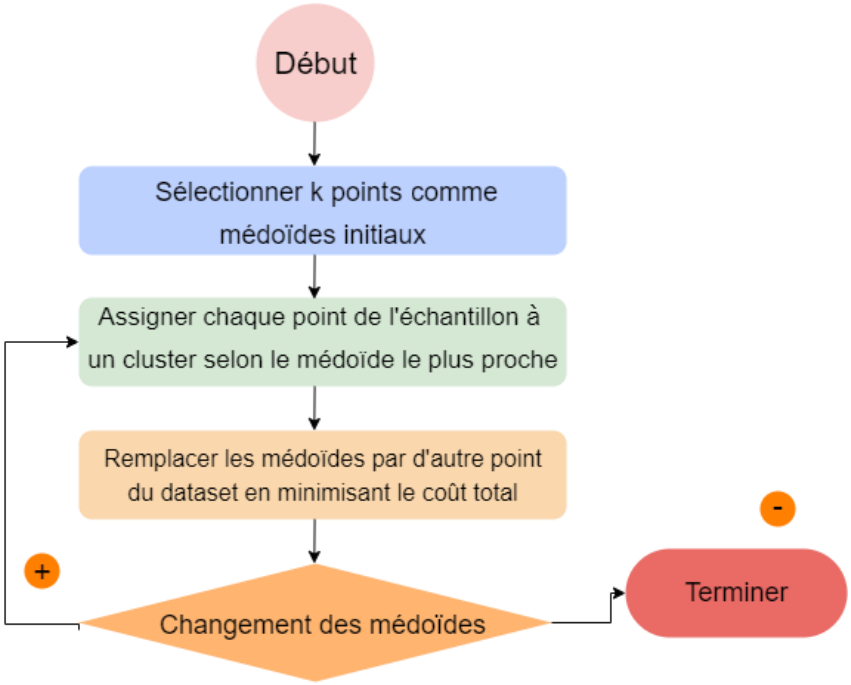

Figure 10 Algorithme de k-médoïdes

II.a.2 Clustering hiérarchique

C'est un algorithme qui construit une hiérarchie des clusters, comme montre la figure 11 au début de notre traitement chaque point du dataset représente un cluster en lui-même ensuite en calculant la distance (euclidienne ou la distance de Manhattan) entre chaque deux classes on parvient à fusionner les clusters les plus proches, on répète cette opération jusqu'à ce qu'il nous reste qu'un seul cluster [20] [25],contrairement au k-means dans le clustering hiérarchique il est impossible de connaître ou prévoir le nombre de clusters qu'on doit choisir et donc on est systématiquement obligé de le déduire depuis le dendrogramme (structure arborescente qui représente le clustering hiérarchique) comme le montre la figure 12.

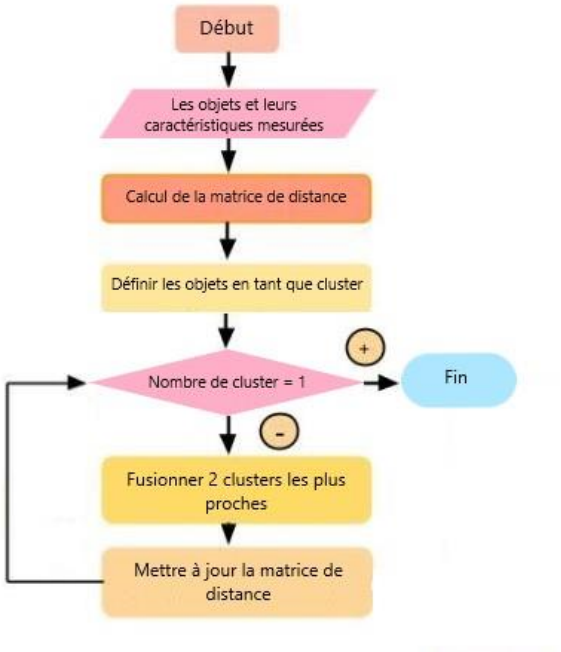

Figure 11 Organigramme de l'algorithme du clustering hiérarchique

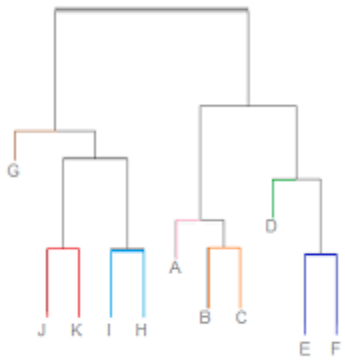

Figure 12 Dendrogramme du clustering hiérarchique

II.b Réduction de dimension

Dans le dessein d'avoir un algorithme robuste et sans erreurs, il est impératif d'avoir des données fiables mais avec l'accroissement de la taille des bases de données il est devenu difficile d'y procéder, pour cela on utilise des algorithmes pour récupérer les données d'un espace de dimension vaste et à les remplacer par des données dans un espace plus restreint cela limite le nombre de possibilités à tester, ce qui permet de traiter les données plus rapidement [12].

II.b.1 Principal component analysis (PCA)

Principal component analysis (PCA) est une technique utilisée pour mettre en évidence la variation dans un ensemble de données et faire ressortir des modèles forts. Utilisée pour faciliter l'exploration et la visualisation des données, son objectif est d'extraire des informations importantes du tableau, de les représenter sous la forme d'un ensemble de nouvelles variables orthogonales appelées composantes principales et d'afficher le modèle de similarité des observations et des variables sous forme de points de carte figure 13 [20].

Dans la figure 13 le dataset est constitué de plusieurs dimensions donnant un visuel complexe, la figure 14 montre l'évolution du dataset après l'application de l'algorithme de PCA.

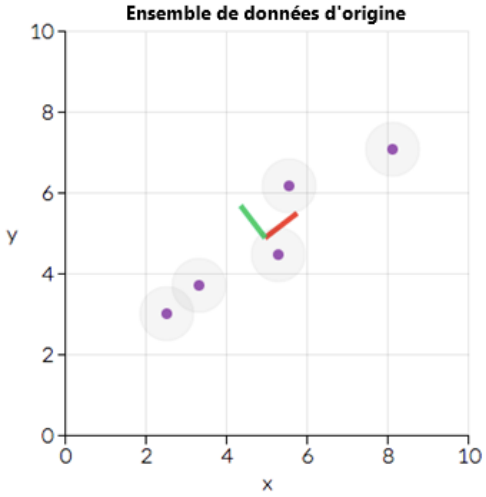

Figure 13 Dataset avant l'algorithme de PCA

Figure 14 Dataset après l'algorithme de PCA

III. Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de l'apprentissage automatique, appelé la science de la décision consiste à entraîner des modèles de l'intelligence artificielle d'une manière plus particulière dans le dessein de rendre la machine autonome, L'agent apprend à se comporter dans l'environnement en effectuant des actions selon le résultat, l'agent reçoit une récompense ou une punition (voir figure 15), avec cette technique l'agent cherche à maximiser ses récompenses en trouvant un modèle d'action plus approprié.

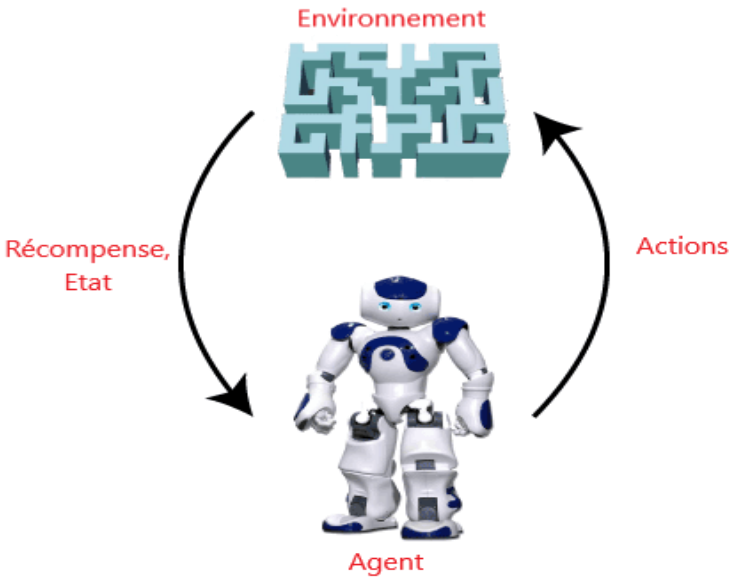

Figure 15 Apprentissage par renforcement (agent et son environnement) [26]

III.a Les algorithmes d'apprentissage par renforcement

III.a.1 Q-Learning

Q-learning est un algorithme d'apprentissage par renforcement off-policy, qui est utilisé pour l'apprentissage par différence temporelle. Il est considéré comme off-policy car la fonction Q-learning apprend à partir d'actions extérieures à l'algorithme policy. Cet algorithme trouvera la meilleure action à prendre compte tenu de l'état actuel.

L'organigramme ci-dessous (voir figure 16) explique le fonctionnement du Q-learning.

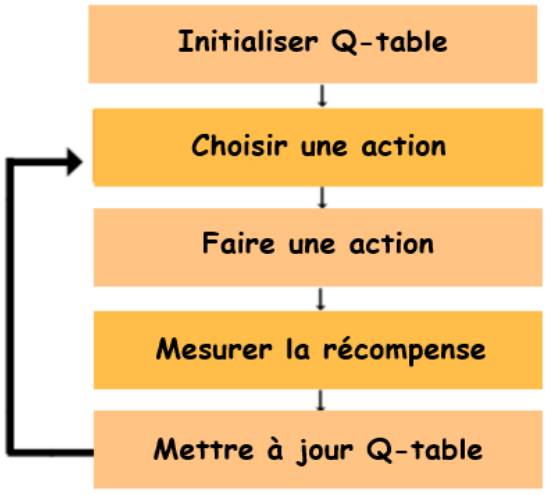

Figure 16 le fonctionnement du Q-learning

III.a.2 DQN

DQN est un Q-learning utilisant des réseaux de neurones. Cela implique principalement la mise en place et la formation d'un réseau de neurones capable d'estimer différentes valeurs Q pour chaque action dans un état particulier et de mettre à jour la table Q.

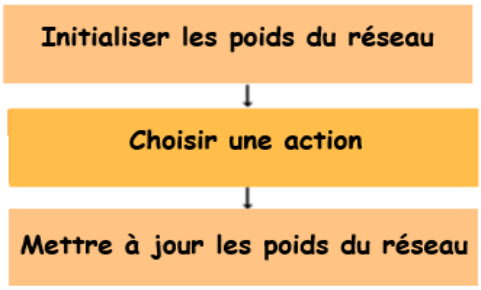

Figure 17 Le fonctionnement de DQN

III.a.3 SARSA

L'algorithme Sarsa signifie State Action Reward State action qui est un algorithme On-Policy pour TD-Learning (apprentissage par différence temporelle). Cet algorithme est une légère variation du populaire algorithme Q-Learning.

La principale différence entre celui-ci et Q-Learning est que, contrairement au Q-learning, la récompense maximale pour l'état suivant n'est pas nécessairement utilisée pour mettre à jour les valeurs Q. Au lieu de cela, une nouvelle action est choisie en utilisant la même stratégie, et donc une récompense qui détermine l'action initiale.

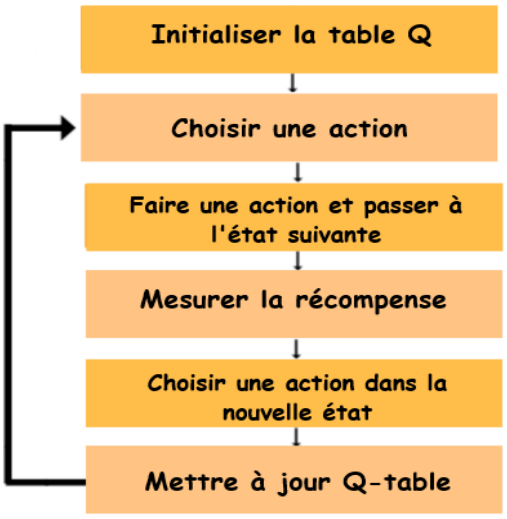

Figure 18 Le fonctionnement de SARSA

3.2 Les étapes du machine Learning

La procédure d'apprentissages automatiques passe par trois principales étapes :

1 Préparation des données

C'est une étape primordiale dans l'apprentissage machine au point que si cette étape n'est pas fiable l'acheminement de l'étude peut prendre une autre tournure. Elle est constituée de deux étapes principales :

a. Récolte de données : on commence par rassembler les données nécessaires adaptées à notre domaine d'étude.

b. Réconciliation : cette étape a deux phases :

- Mise à l'échelle : les données collectées ne sont pas toujours dans le bon format comme c'est le cas, par exemple des informations textuelles qu'on doit les échangées par des chiffres significatifs (0 pour male, 1 pour femelle) mais aussi les anomalies qu'il faut les éliminées dans le jeu de données avant de l'utiliser.

- Nettoyage des données : les valeurs manquantes ou dupliquées ont aussi un impact sur le modèle, c'est pourquoi il faut les éliminer ou les remplacer.

2 L'ingénierie des caractéristiques

On visualise nos données dans leurs ensembles pour voir s'il existe un lien entre les features, si ce dernier est imposant, il en résulte qu'elles soient directement dépendantes l'une de l'autre dans ce cas on peut se contenter d'une seule parmi les deux.

3 Le choix d'algorithme

a. Avant de choisir le modèle adéquat, il faut diviser le dataset en deux sous-ensembles :

- Les données d'entraînements : (training-set) qui servent à entraîner le modèle choisis.

- Les données de test : (test-set) celui-ci permet de vérifier la performance du modèle et voir s'il est capable de travailler avec d'autre données que celle du train-set.

b. Une fois que les données ont bien été séparées, on peut opter pour un modèle selon nos besoins et selon le domaine de recherche.

c. Après le choix du modèle, maintenant il faut l'entrainer sur sous-ensemble de donnée (train-set).

d. Maintenant nous examinons le modèle sur un autre groupe de donnée (test-set) pour s'assurer que le modèle en question marche à la perfection en déterminant un pourcentage de précision selon les exigences de nos attentes.

e. La dernière étape dans le cycle de vie du machine Learning est le déploiement du modèle sur le monde réel avec des données qui n'a jamais rencontré auparavant [1].

La figure 19 Montre les étapes du machine learning.

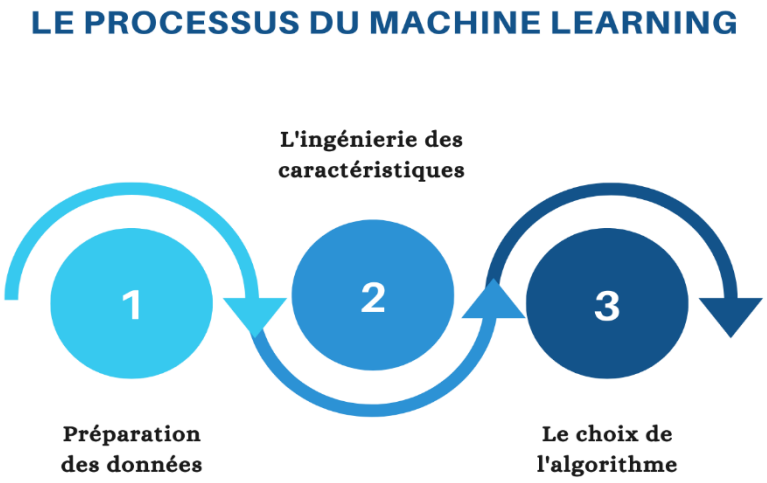

Figure 19 Les étapes de l'apprentissage machine

3.3 Limite de l'apprentissage automatique

Malgré les avantages d'apprentissage machine, il reste quelques limites qui constituent des défis pour chaque développeur de domaine.

- Pour obtenir un modèle fiable il est nécessaire de collecter une grande quantité de données et parfois il est difficile de les nettoyer et de les mettre au bon format.

- Sensibilité élevée aux erreurs et aux anomalies.

- Si le modèle est entraîné sur une base de données restreinte on risque de tomber dans le cas du sous-apprentissage, et dans le cas contraire il est probable d'être face à un problème de sur-apprentissage et dans les deux cas le modèle aura du mal à s'adapter avec des nouvelles données.

- Dans certaine situation le choix du modèle n'est pas toujours évident ce qu'il nous oblige d'essayer plusieurs modèles avec plusieurs paramètres.

4. Conclusion

Dans ce chapitre on a pu citer quelques algorithmes du machine learning de chaque catégorie en montrant leurs domaines d'application et leurs déroulement en bref.

Le chapitre qui suit va être consacré à un algorithme spécifique de l'apprentissage non supervisé qui est le k-means.

Chapitre 2
k-means clustering

1. Introduction

Les algorithmes de clustering ont pour but de classer un ensemble de données selon leurs similarités et leurs ressemblances.

Ce chapitre va être dédié à un des algorithmes de clustering qui est le k-means. Dans ce qui suit nous allons détailler les différentes définitions, concepts et techniques citées dans la littérature sur le k-means.

2. Algorithme k-means

K-means est un algorithme itératif très populaire dans l'apprentissage non supervisé consiste à regrouper les données homogènes dans un nombre prédéterminé de clusters 'K' distincts et qui ne chevauchent pas, autrement dit on ne peut pas avoir une instance qui appartient à deux classes au même temps, en calculant plusieurs fois les centroïdes (centre de gravité) jusqu'à ce que le barycentre optimal soit trouver [27].

L'algorithme de k-means clustering cherche à minimiser une fonction coût appelée *inertia* et qui représente la somme des distances entre les points d'un cluster x et le centroïde de ce dernier c [27].

La formule (1) représente la fonction coût de k-means (inertia) où :

n : le nombre de points du dataset

c : le centroïde de chaque cluster

i : fait référence à chaque point du dataset (i varie de 0 à n) [28].

$$inertia = \sum_{i=0}^{n} \min(\|x_i - c_i\|^2) \dots\dots\dots (1)$$

3. Métrique de distance

Pour que l'algorithme de k-means crée des clusters, il doit calculer la distance entre les points et leurs centroïdes. Il existe plusieurs façons pour calculer la distance, nous citerons comme exemple les trois techniques les plus utilisées dans la littérature.

3.1 Distance euclidienne

C'est la façon la plus évidente de représenter l'écart entre deux points [29] en calculant la longueur d'un segment qui les relient (voir figure 20), si on prend deux points tel que $P1(x_1, y_1)$, $P2(x_2, y_2)$ la distance euclidienne ***d*** entre P1et P2 sera (voir formule 2) :

$$d = (x_2 - x_1)^2 + (y_2 - y_1)^2 \ldots\ldots\ldots (2)$$

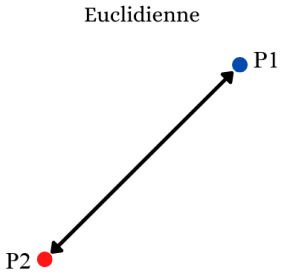

Figure 20 La distance euclidienne [24]

3.2 Distance de Manhattan

Dans le cas où on est face à une carte routière et on veut trouver la distance entre deux stations par exemple, la distance euclidienne n'est pas appropriée pour ce genre de défi et pour cela on a utilisé la distance de Manhattan.

La distance ***d*** entre deux points est la somme des valeurs absolues de la différence de leurs coordonnées cartésiennes [29] (voir figure 21). Autrement dit c'est la somme de la différence entre les coordonnées x et les coordonnées y (voir formule 3).

Tel que : $d = |x_2 - x_1| + |y_2 - y_1| \ldots\ldots\ldots (3)$

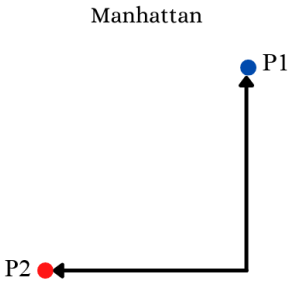

Figure 21 La distance de Manhattan

3.3 Similitude cosinus

La similitude cosinus est trop utilisée dans l'apprentissage automatique pour comparer des fichiers texte ou autre, elle mesure la similarité en utilisant le cosinus de l'angle θ entre deux vecteurs dans un espace multidimensionnel [30] (voir figure 22). Elle est donnée par la formule 4 :

$$similarity(x, y) = \cos\theta = \frac{x*y}{|x||y|} \ldots\ldots\ldots (4)$$

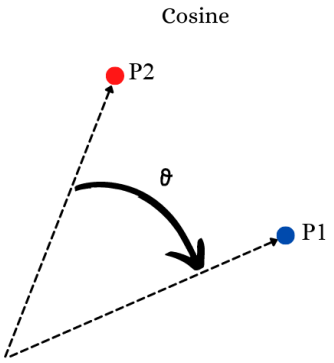

Figure 22 La similitude Cosine

4. Les étapes de k-means

Le k-means est un algorithme itératif qui peut être résumé en quatre étapes [31] :

- **Étape 1** : désigner le nombre de clusters ‘k’, généralement cette désignation sera soit : aléatoire ou par tâtonnement selon des connaissances précédentes sur la nature et la distribution du dataset, elle peut même être faite par l'utilisation des techniques d’apprentissages.

- **Étape 2** : initialiser k centroïdes d’une manière aléatoire (on en parlera plus en détail plus loin dans le document) dans la figure 23 Les points roses représentent les centroïdes initiaux, où k=3 et i =0...k.

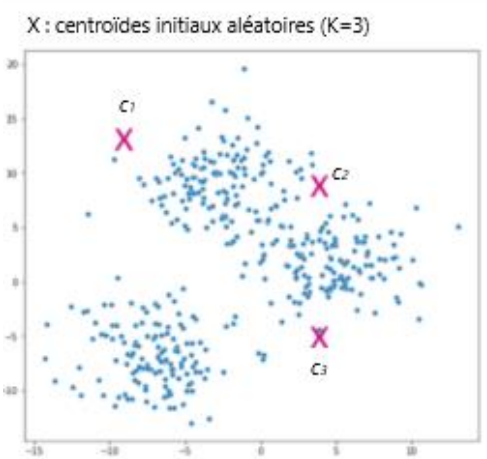

Figure 23 Les centroïdes initiaux [31]

- **Étape 3** : cette phase englobe trois étapes principales qui vont être réitérées jusqu’à ce que les centroïdes convergent vers une position d’équilibre.

  - La somme des distances ***d*** au carré (inertia) entre chaque point de données x_i et les ***k*** centroïdes c_i serait calculée en premier par la formule 5 (voir figure 24).

$$d(x_i, c_i) = \sqrt{\sum_{j=1}^{d}(x_{i1} - c_{i1})} \quad \ldots\ldots\ldots (5)$$

$$i=1...N,\ j=1...k$$

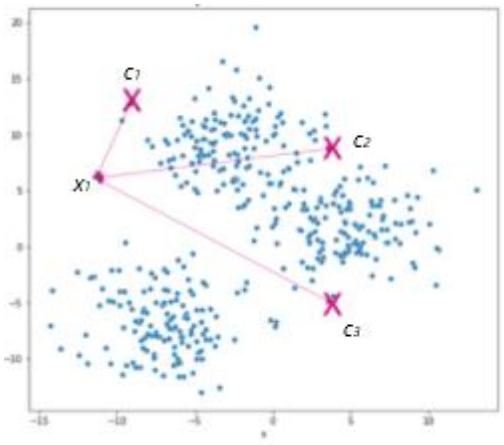

Figure 24 distance entre les centroïdes et les autres points

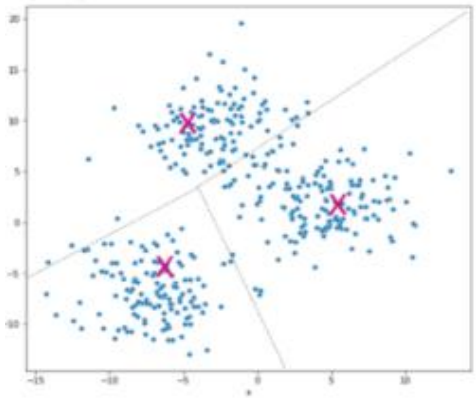

Figure 25 les centres de gravités après l'optimisation

- Afin d'optimiser la fonction coût (voir formule 1), le centre de gravité sera attribué au cluster le plus proche (voir figure 25).

- Enfin, Recalculer le nouveau centre de gravité $'c_i$ de chaque cluster *i* par la formule 6.

Figure 26 désigner les nouveau centroïdes

$$'c_i = \frac{1}{m_i}\sum_{j=1}^{Ni} d(x_i, c_i) \ldots\ldots\ldots (6)$$

Où :

d: la distances

c_i : le centroïde i

m_i : le nombre des points affecté au centroïde c_i

x_i : le point x affecté a ce centroïde

- **Etape 4 :** itérer l'algorithme (répéter l'étape 3) jusqu'à ce que les centroïdes ne changent plus de groupe

La figure 27 ci-dessous exprime le déroulement de l'algorithme de k-means.

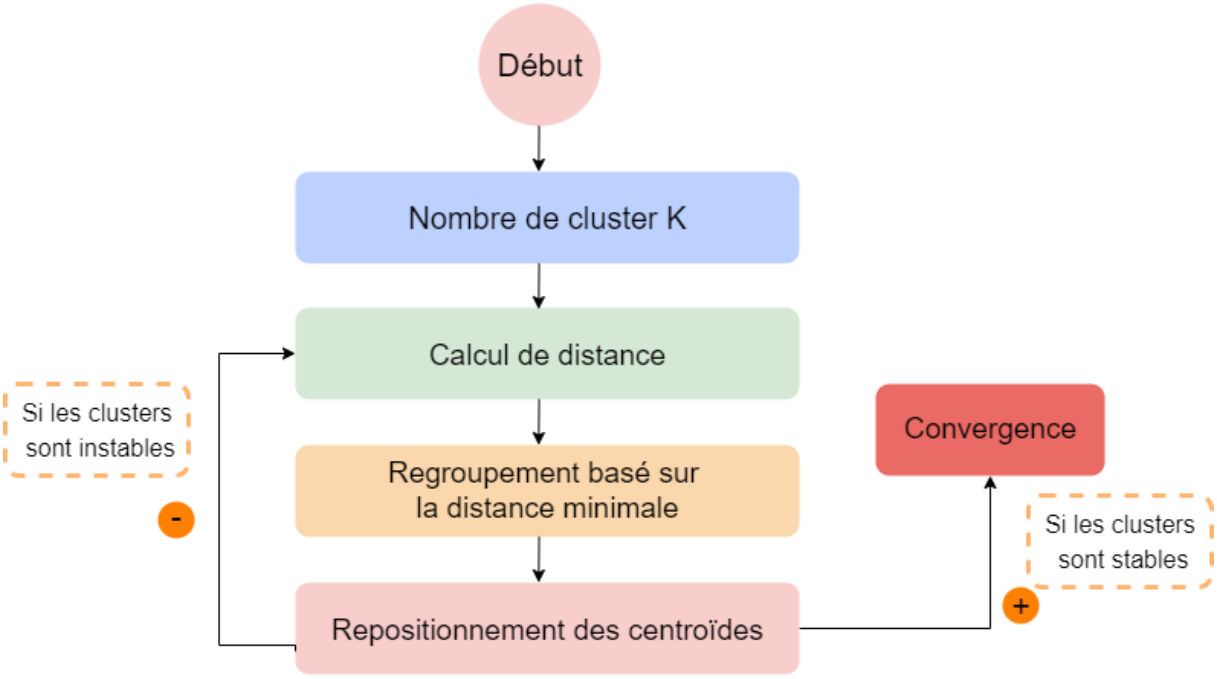

Figure 27 Le déroulement de l'algorithme K-means [32]

5. Limites du k-means

Les trois problèmes qui se répètent lors de l'application de k-means sont [32] :

a. Comment choisir le nombre de' k' autrement dit comment savoir combien de cluster faudra créer d'une façon où on n'aura pas de chevauchement (chaque point appartient à un seul cluster),

b. Comment initialiser les centroïdes de telle manière à ce qu'ils soient correctement dispersés dans le dataset.

c. Comment éliminer les outliers (anomalie) et ne pas les prendre comme centroïdes.

Dans la section suivante nous détaillerons chaque point et nous citerons quelques solutions existantes dans la littérature.

5.1 Comment choisir le nombre de cluster

La première préoccupation quand on veut utiliser le k-means est la façon dont le nombre de cluster 'k' est choisi, est ce qu'en fonction de la quantité des données ou par rapport à leur

répartition dans le nuage, l'algorithme de k-means clustering est naïf dans le sens où il exploite l'ensemble des k qu'on lui propose sans se soucier du fait que le nombre de k n'est pas adéquat, pour cela on compte deux procédés méthode de Elbow (la méthode du coude), et l'analyse de silhouette [33] [34].

5.1.1 Méthode de Elbow

Méthode de Elbow est une méthode itérative qui a pour objectif de trouver le k le plus optimal et ceux en faisant varier le nombre de cluster k dans un intervalle (de 1 à 10 par exemple) tout en calculant pour chaque **k** la somme de la distance au carré entre chaque point et le centroïde du cluster (Sum Squared Error) **SSE** [33], ensuite on fait une déduction visuelle de la meilleure valeur de k en traçant les métriques globales pour chaque valeur de k, dans la figure 28 On remarque un coude au niveau de k=3 ce qui représente le nombre de clusters qu'on doit créer.

SSE est calculée par la formule 7 : $SSE = \sum_{i=1}^{k} \sum_{x_j \in c_i} \|x_j - c_i\| \ldots\ldots\ldots (7)$

Tel que :

x_j : les points d'un cluster

c_i : les centroïdes d'un cluster [33].

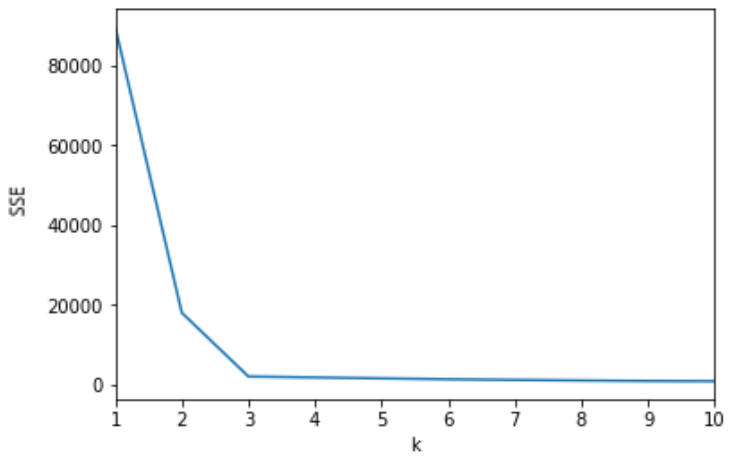

Figure 28 Méthode de Elbow

5.1.2 Analyse de silhouette

Une autre technique pour trouver le nombre de cluster k idéal en se basant sur le calcul de la distance qui sépare les clusters résultants (après l'application du k-means), en consultant le tracé de la silhouette on peut déduire la mesure de la proximité de chaque élément du cluster par rapport aux points des clusters voisins et celle de son propre cluster, cette dernière appartient à l'intervalle [-1, 1], quand cette valeur est proche de +1 cela montre que l'échantillon se tient éloigné des grappes voisines. Si cette valeur est nulle on en déduit que l'échantillon est sur ou très proche de la limite de décision entre deux clusters voisins et des valeurs négatives indiquent que ces échantillons peuvent avoir été affectés au mauvais cluster [35] [36].

Pour calculer le coefficient de silhouette il faut d'abord calculer la distance moyenne entre chaque point et les autres qui l'entourent dans le même cluster qu'on nomme a^i, à la suite de cela on enchaîne avec la mesure de la distance moyenne entre chaque point du cluster et tous les autres éléments du cluster voisin b^i [34]. Le coefficient est donné par la formule 8 :

$$S = \frac{b^i - a^i}{max(a^i, b^i)} \ldots\ldots\ldots (8)$$

i : les points du dataset

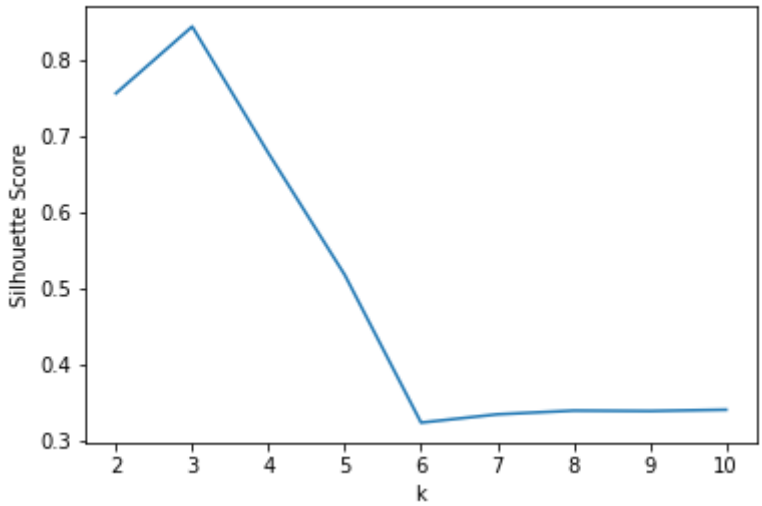

Figure 29 Analyse de silhouette

Depuis la figure 29 On voit très clairement l'existence d'un pique de s=0.8 autrement dit k=3 est le meilleur nombre de clusters.

Elbow (méthode de coude) est une règle de décision, tandis que la silhouette est une métrique appliquée dans le but de valider un certain regroupement de points. Étant donné que la méthode du coude n'est pas une alternative de la méthode de silhouette, pour avoir un résultat robuste il vaut mieux les combiner.

5.2 Le choix des centroïdes initiaux

Une fois que le nombre de cluster k est trouvé avec les méthodes déjà citées auparavant. On enchaîne avec la désignation des centres de gravité initiaux de notre dataset (centroïde), c'est une étape précieuse et primordiale dans l'algorithme de k-means étant donné que si le choix des centroïdes n'est pas approuvé, tout le reste du déroulement prend une autre tournure.

Prenant un exemple avec k=3, le premier centroïde est toujours sélectionné au hasard à partir du dataset le problème se pose lors de la sélection du deuxième et troisième centroïde, dans la méthode classique de k-means le choix des centroïdes initiaux est pris au hasard ce qui ne génère pas un résultat absolu autrement dit les centroïdes choisis risquent de ne pas être correctement positionnés dans l'espace de données. De nombreuses solutions sont proposées dans la littérature, nous citons quelques techniques populaires.

5.2.1 A base de tâtonnement

Une stratégie classique consiste à exécuter l'algorithme de k-means plusieurs fois d'affilée en modifiant à chaque fois la position initiale de nos centroïdes, pour chaque résultat donné on mesure la distance entre les points d'un cluster et les centroïdes et on retient la solution où la fonction coût est la plus petite possible (modèle b figure 30).

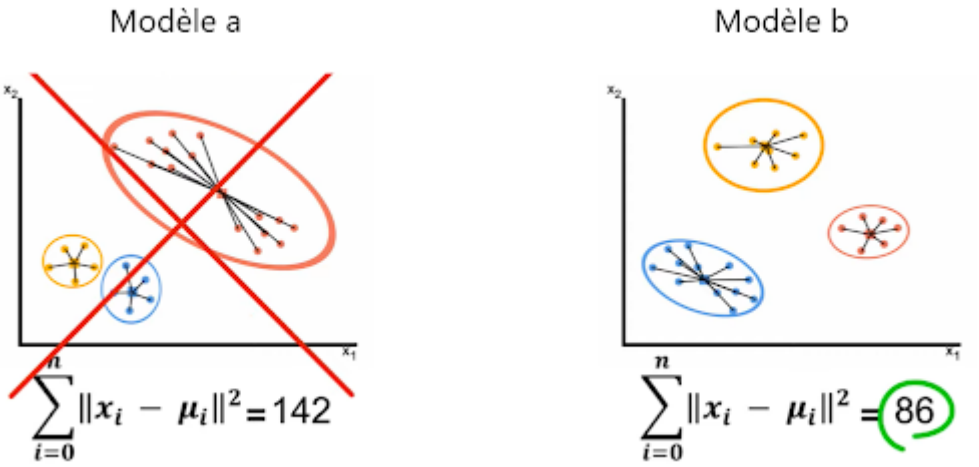

Figure 30 Choix des centroïdes

Cette méthode peut être efficace mais elle ne passe pas à l'échelle en termes de taille de dataset et le nombre de centroïdes

5.2.2 K-means ++

La technique la plus utilisée dans la sélection des centroïdes est le k-means++ qui est définit comme suit [37] :

1. Sélectionner au hasard le premier centroïde *c* à partir de notre jeu de données.

2. Calculer la distance *D* entre tous les points du dataset et le centroïde déjà sélectionné

$$D_i = max_{(j:1\to k)} \|x_i - c_j\|^2 \ldots\ldots\ldots (9)$$

c : centroïde de chaque cluster

x : les point d'un échantillon

i : varie de 1à n (taille du data set)

j : vari de 1 à k (nombre de classe)

3. Choisir le nouveau centre de cluster celui dont la distance $(D(x))^2$ est la plus éloignée du centroïde actuel.

4. Répéter les étapes 2 et 3 jusqu'à ce que k centroïde de chaque cluster soient trouvés.

5.2.3 A base de poids

Les stratégies à base de poids utilisent des métriques pour évaluer l'importance des points dans leurs dataset. L'idée générale de ces techniques est la suivante [6] :

1. Sélectionner une métrique d'évaluation d'importance des points.
2. Pour chaque point dans le dataset : calculer cette métrique.
3. Classer en ordre décroissant les points selon leurs valeurs de métrique d'évaluation.
4. Sélectionner comme centroïdes initiaux les k premiers points dans la liste ordonnée.

Les métriques peuvent être par exemple : la densité du point (voir formule 11), la dissimilarité du point ... [6].

5.3 Détection des anomalies

Il existe plusieurs interprétations d'une valeur aberrante, appelée aussi « outliers ». Selon les auteurs dans [49][1], une valeur aberrante est « une observation (ou sous-ensemble d'observations) qui semble être incompatible avec le reste de cet ensemble de données ». Cependant, la détection d'une anomalie nécessite une analyse et une évaluation concrète afin de s'assurer que la valeur identifiée représente bien un outlier.

Les valeurs aberrantes sont caractérisées par leurs grandes dissemblances par rapport aux autres observations de la même catégorie (point rouge dans la figure 31), qu'ils soient nombreux ou minimes, la présence des outliers dans un dataset a un impact significatif sur le regroupement des données lors de l'application de l'algorithme de k-means clustering et peut faire bifurquer l'étude.

Le problème majeur c'est lorsqu'une anomalie est choisie comme un centroïde initial et pour éviter cela l'idéal est d'éliminer les outliers au tout début de l'analyse [35].

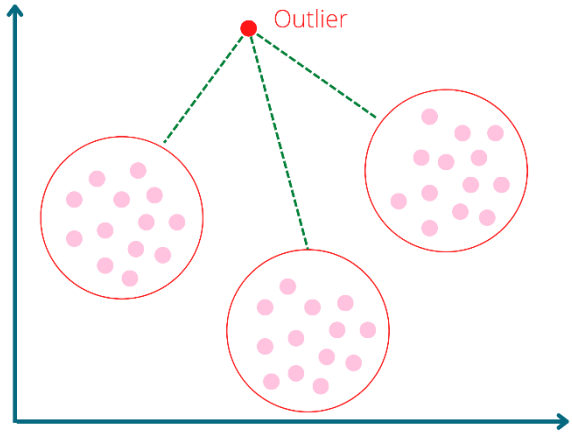

Figure 31 Les anomalies [38]

Il existe trois grandes catégories de techniques de détection d'anomalies :

- Les techniques de détection d'anomalies non supervisées détectent les anomalies dans un ensemble de données non étiquetées en supposant que la majorité des instances de l'ensemble de données sont normales et en recherchant les instances qui ne correspondent pas au reste des données ;

- Les techniques de détection d'anomalies supervisées nécessitent un ensemble de données où les données sont étiquetées normales ou anormales et impliquent l'entrainement d'un classificateur (la principale différence par rapport à de nombreux autres problèmes de classification statistique réside dans la nature déséquilibrée de la détection des valeurs aberrantes) ;

- Les techniques de détection d'anomalies semi-supervisées construisent un modèle représentant le comportement normal d'un ensemble de données normales, puis testent la probabilité qu'une instance de test soit compatible avec le modèle.

Dans le cas des k-means, les outliers sont généralement détectés par une technique non supervisée [39].

6. Domaine d'Applications de K-Means

L'utilisation de K-means clustering peut simplifier la complexité superflue que pourrait avoir un dataset mais pas que, car elle pourrait être utilisée dans des domaines extrêmement variés.

1. Segmentation d'image : étant donné que l'image est un ensemble de pixel, la segmentation d'image consiste à regrouper les pixels similaires, cette technique est ensuite utilisée dans le domaine médical, dans les voitures autonomes (segmenter les objets qui l'entourent pour ensuite les détecter) [36]. La figure 32 montre une image avant et après l'application de k-means avec k=3.

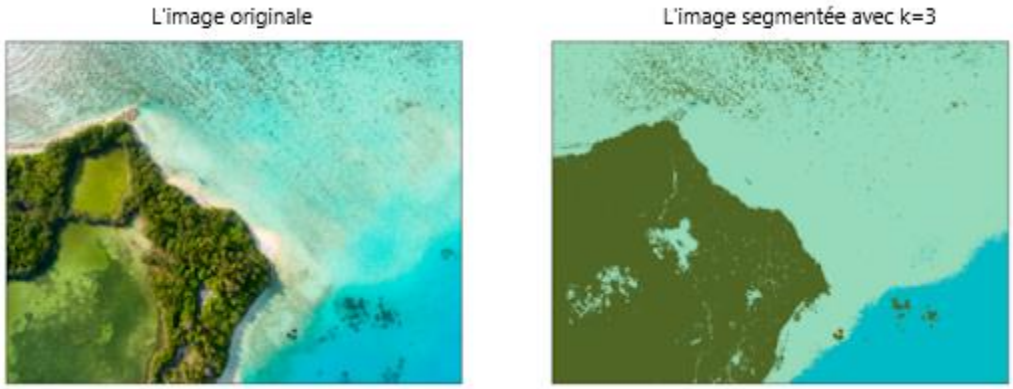

Figure 32 Avant et après l'application de k-means [40]

2. Segmentation de la clientèle : le clustering aide les spécialistes du marketing à segmenter les clients en fonction de l'historique de leurs achats pour ensuite les cibler à l'aide de publicité et autres. La figure ci-dessous montre un schéma représentatif de cette méthode.

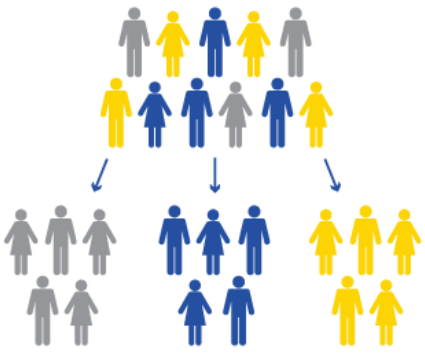

Figure 33 Segmentation de la clientèle [41]

3. Classification des documents : regrouper les documents de manière à ce que des documents similaires soient dans les mêmes clusters.

7. Désavantages de K-means

- Difficile de prédire le nombre de clusters (nombre de K).
- Si les clusters ont des formes géométriques complexes, k-means risque de ne pas diviser le dataset comme il le faut exemple figure 34.

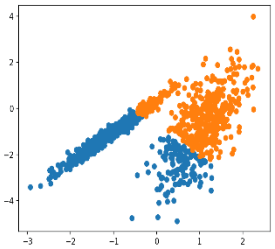

Figure 34 application de k-means sur un dataset complexe

- Dans le cas ou des points partagent les mêmes caractéristiques sont éloignés le k-means ne permet pas leurs regroupements.
- Si le choix des centroïdes initiaux n'est pas adéquat, les résultats peuvent prendre une autre tournure.
- Si des outliers sont présent dans un jeu de données il est probable que les résultats bifurquent en donnant des clusters vide de sens prenant en exemple le cas où un outliers est choisie au hasard comme un centroïde le reste des résultats prennent systématiquement une autre tournure.
- Sensible aux valeurs aberrantes.
- L'ordre des données a un impact sur le résultat final.

8. Conclusion

Ce chapitre ci-dessus a été entièrement consacré à l'algorithme de k-means clustering, on a d'abord abordé ses étapes pour ensuite mieux connaître ses limites et ses domaines d'applications. Afin d'améliorer les performances de k-means et dépasser ses limites, nous proposeront dans le prochain chapitre un algorithme de k-means plus performant.

Chapitre 3

La Nouvelle Approche « DDK-means »

1. Introduction

Dans le chapitre précédent on a vu les différentes limites de l'algorithme K-means, comme la présence des anomalies, désignation des centroïdes initiaux et le choix du nombre de clusters a réalisé tous cela a un impact sur l'efficacité de l'algorithme face à des grandes base de données.

Dans ce chapitre on va présenter une nouvelle approche DDK-means (**Density Distance K-means**), qui a était basé sur des études déjà faite auparavant, dans cette méthode on s'est focalisé sur la détection des valeurs aberrantes et la sélection des centroïdes initiaux.

2. Algorithme de la nouvelle approche

Dans notre approche DDK-means on a choisi de se focaliser sur les deux plus grandes faiblesses de l'algorithme de k-means et c'est : l'élimination des anomalies et la sélection des centroïdes initiaux qui représente une grande partie du flux de l'algorithme, Une meilleure sélection des centres de cluster initiaux pour le clustering k-means est toujours un problème de recherche intéressant en raison de l'importance du clustering k-means dans les applications du monde réel.

La performance de l'algorithme de k-means se dégrade face aux valeurs aberrantes provoquant des changements au niveau du flux de l'algorithme et donnant des statistiques moins bonnes.

2.1 Désignation des centroïdes initiaux dans l'algorithme classique

K-means sélectionne aléatoirement les centroïdes initiaux (étape 2 du k-means chapitre 2), pour ce fait la présence d'anomalie dans le jeu de donnée augmente la probabilité qu'une d'entre elles sera choisi comme centroïde initiale, dans ce cas le cluster le plus proches du centroïdes choisi sera migré vers ce point.

Dans la (figure 35- a), c1 est une anomalie qui a été choisie comme un centroïde initial et comme on peut voir dans la (figure 35-b) c1 attire tous les points du cluster le plus proche vers elle et cela donnera des mauvais résultats à la fin, on peut voir la différence avec c2 (figure

35-a) qui n'est pas une anomalie sa position comme centroïde n'a pas provoqué de changement au niveau de la position des points ce qui donne des résultats optimaux à la fin (figure 35-b).

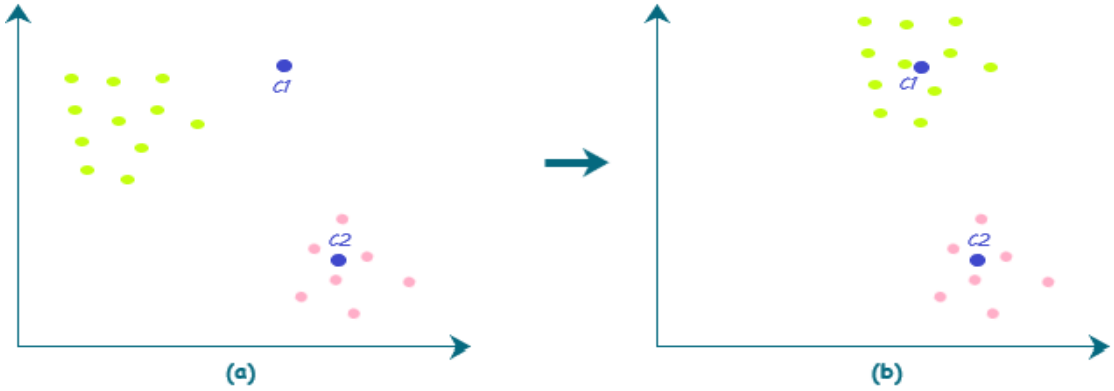

Figure 35 les anomalies dans l'algorithme de k-means

Dans ce qui suit nous allons présenter les étapes de l'approche DDK-means qui se base sur des études déjà faites auparavant [12].

2.2 Les étapes de DDK-means

I. Étape 1 : Détection d'anomalies

Une valeur aberrante est un point qui s'écarte tellement des autres observations qu'il aurait pu être généré par un mécanisme différent (voir figure36) [50]. Les anomalies sont générées suite à des erreurs de mesure ou de bruit, mais elles pourraient indiquer des changements de comportement ou un comportement aberrant dans le système observé, elles sont parfois utilisées dans un but précis comme par exemple dans le cas de la détection des fraudes [50]. On peut distinguer trois types d'outliers : Anomalie locale, globale et contextuelle [51].

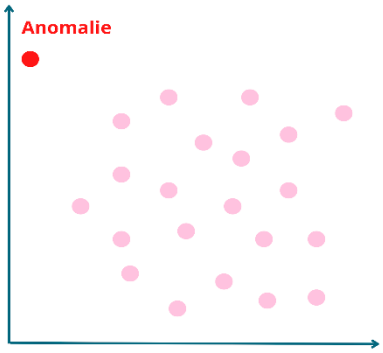

Figure 36 Anomalie

I.a Anomalie externe "global"

Une anomalie externe c'est la forme la plus simple à détecter puisqu'elle s'éclate fortement des autres observations, en générale toutes les méthodes de détection des outliers visent à détecter les valeurs aberrantes globales.

Dans la (figure 37) le point rouge représente une valeur aberrante externe.

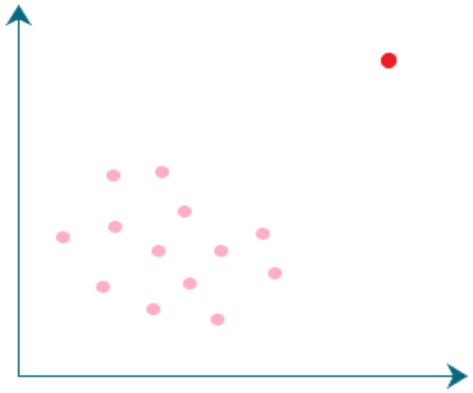

Figure 37 Anomalie externe

Pour éviter toute perturbation l'idée été d'isoler et éliminer les anomalies distantes et notre stratégie été de choisir l'algorithme d'**isolation forest** pour sa rapidité et son efficacité face aux anomalies global.

I.b Isolation Forest

Isolation Forest est un algorithme non-supervisé qui a pour but de détecter les valeurs aberrantes en les isolants d'une manière récursive il est très efficace quand il s'agit du repérage des outliers globales [52] [53].

L'idée est de faire plusieurs découpes au niveau de notre dataset et calculer le nombre de split qu'il faut faire avant d'isoler un échantillon, plus ce nombre est petit, plus il y'a de chance que cet élément soit un outlier [54].

1. On commence par choisir un point au hasard (voir figure 38).

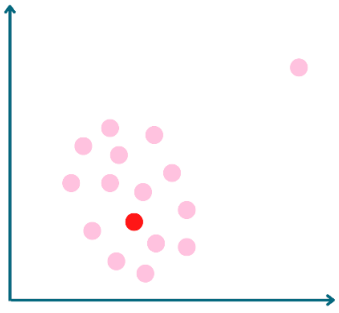

Figure 38 Algorithme Isolation Forest, Etape 1

2. On fait un split (découpe) au hasard (voir figure 39).

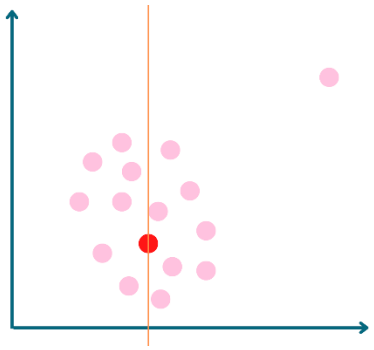

Figure 39 Algorithme Isolation Forest, Etape 2

3. Si aucun élément est isolé on refait cette étape plusieurs fois (figure 40).

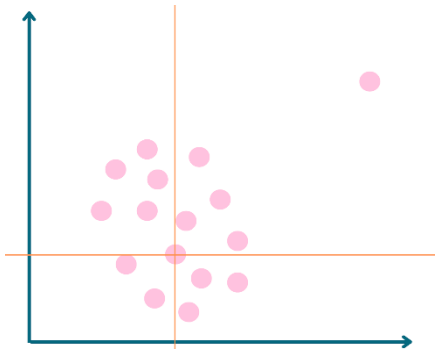

Figure 40 Algorithme Isolation Forest, Etape 3

4. Si un des échantillons est isolé il est désigné comme outlier ; (figure 41) le dernier split a révélé que le point encerclé est un outlier.

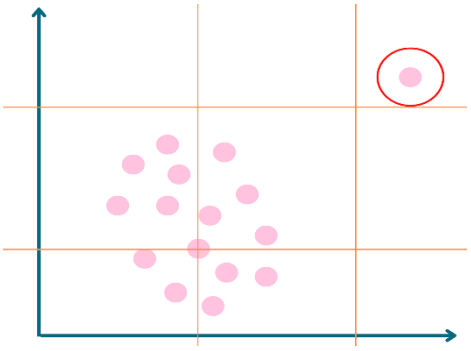

Figure 41 Algorithme Isolation Forest, Etape 4

II. Étape 2 : calcul de diagonale

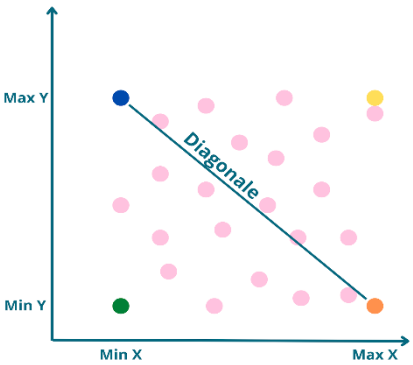

Figure 42 Calcul de diagonale

On calcule la diagonale de la base de données, puisqu'elle représente la plus grande distance et qui nous permet de mieux apercevoir la distribution des données (voir figure 42).

III. Étape 3 : calcul d'alpha (α)

Dans cette étape on calcule une valeur nommée alpha (α) qui représente la valeur de diagonale divisée par le nombre de cluster de la base de données, cette valeur sera utilisée par la suite comme u rayon de voisinage. Dans la (figure 43), on a choisi k=3, α est calculé par la formule suivante :

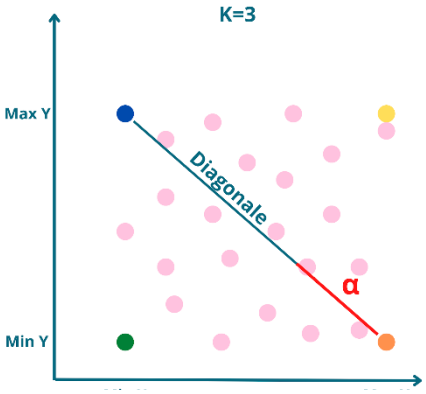

Figure 43 Calcul d'alpha

$$\alpha = \frac{diagonale}{k} \dots\dots\dots\dots (10)$$

Tel que :

α : rayon de voisinage

k : le nombre de cluster à former (définie au préalable).

$diagonale$: calculée dans l'étape précédente.

IV. Étape 4 : calcul de la densité (D)

Dans cette étape on calcule la densité de chaque point du dataset, le but derrière cela est de choisir à la fin les centroïdes qui ont une bonne densité autrement dit les points qui ne sont pas des outliers.

La densité de chaque point représente le nombre de voisin dans un rayon alpha (calculer dans l'étape 3 de l'algorithme) (voir figure 44).

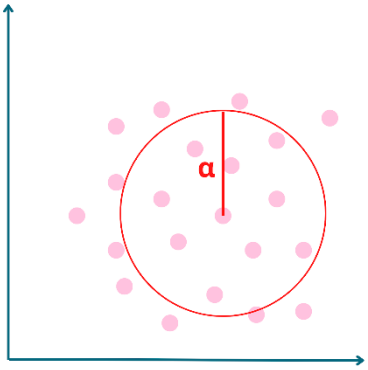

Figure 44 Calcul du des voisin

Pour chaque point on calcule l'écart entre cet échantillon et tous les autres points du dataset, et on garde que ceux qui ont une distance inférieure ou égale à (α), ces derniers seront considérés comme voisins (voir figure 45). Tel que :

Les lignes vertes : les distances avec les points qui vont être considérés comme voisins.

Les lignes rouges : les distances avec les points qui seront ignorés.

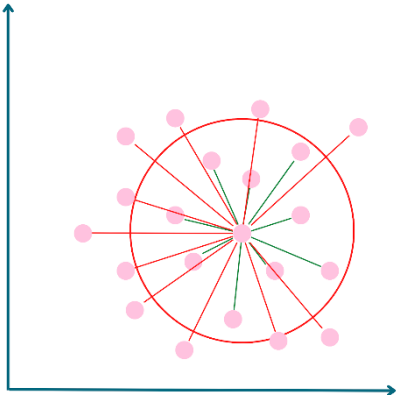

Figure 45 calcul de distances

V. Étape 5 : calcul de chevauchement intra cluster

Après le calcul de la densité de chaque point, l'idée est de prendre les centroïdes les plus éloigné les uns des autres (voir figure 46) les points rouges représentent des centroïdes initiaux.

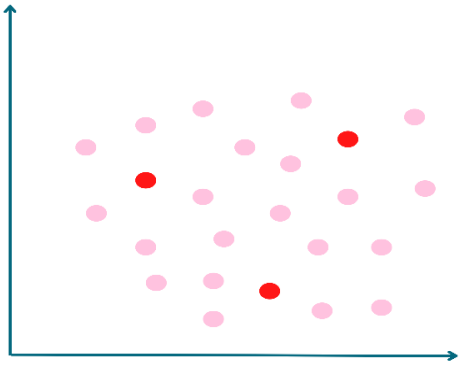

Figure 46 Centroïdes éloignés les uns des autres

L'idée consiste à étudier le chevauchement entre les clusters au premier lieu pour ensuite fusionner les groupes de données qui ont un grand nombre de points en commun (voir figure 47) dans la partie (a) on voit un très grand chevauchement entre les clusters ce qui va mener à joindre les clusters comme le montre la partie (b) de la (figure 47).

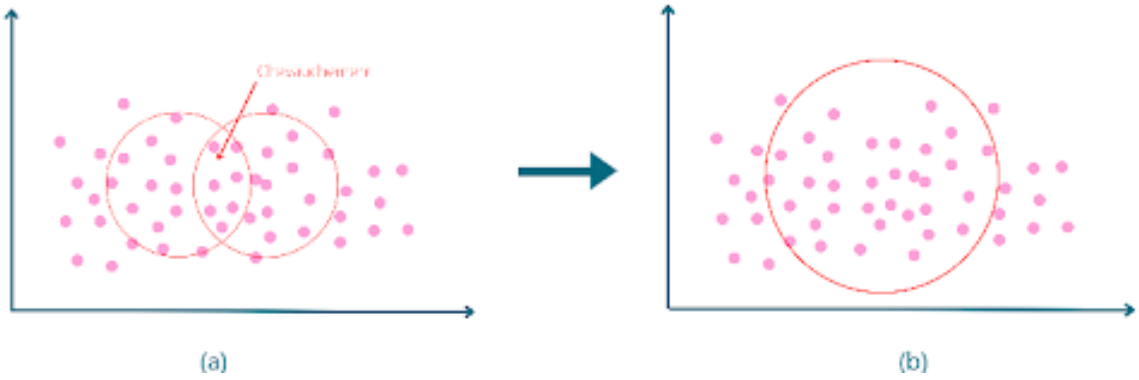

Figure 47 chevauchement

Les clusters seront formés selon la densité des points pour ensuite, on calcule un taux de densité et de chevauchement entre chaque deux cluster la formule sera donner comme suite : pour chaque cluster Ci :

$$C(c1, c2) = \frac{maxD}{pc} \dots\dots\dots\dots (11)$$

Où :

$C(c1, c2)$: *le chevauchement entre le cluster 1 et 2.*

$maxD$: *la densité la plus grande.*

pc : *nombre de point en commun.*

On commence par récupérer les voisins de chaque cluster et former des mini cluster, ensuite pour chaque deux clusters on compte le nombre de point en commun, les clusters qui ont un taux de (densité / chevauchement) élever et qui ont un nombre de point en commun très faible serons sélectionner, l'étape qui suit consiste à calculer les distances entre chaque deux points des deux ensembles, pour enfin sélectionner celle qui sont les plus loin possible (voir figure …).

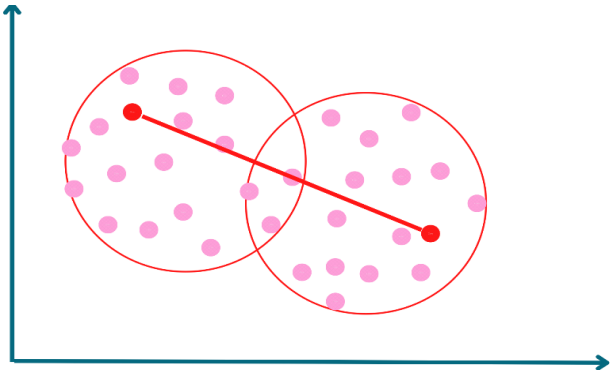

Figure 48 chevauchement de deux clusters

Cette technique nous permet de prendre les points qui ont une grande densité, un grand écart entre eux et leurs clusters ne chevauche pas.

On répète cette étape jusqu'à ce qu'on trouve les 'k' centroïdes pour les k cluster.

VI. Étape 6 : étape finale

Une fois que les centroïdes initiaux sont désignés, on applique l'algorithme de k-means classique (voir l'algorithme dans le chapitre 2 pour plus de détail) avec comme centroïdes initiaux les points sélectionner dans l'étape précédente. (Voir figure 48) le but est d'avoir des clusters bien définie et qui ne chevauche pas tout en gardant un bon nombre de cluster et en donnant des meilleurs résultats.

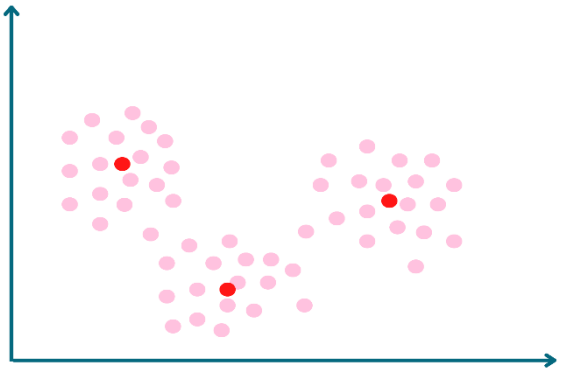

Figure 49 Clusters non chevauchés

3. Algorithme de DDK-means

Input : dataset, k // *k est le nombre de cluster*
Output : model de DDK-means // *DDk-means est le model optimisé de k-means*
Début :

1- $s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$ //*cette formule sert à détecter les anomalies*

2- $d' = \sqrt{d(minXminY, maxXminY)^2 + d(minXminY, minXmaxY)^2}$ //*d': la diagonale*

3- $\alpha = \frac{d'}{k}$ // α : *le rayon de voisinage*

4- //*calcul pour chaque élément le nombre de voisin*

def $calculD\ (a)$:

$d'' = d(a, x_i);$ //*a:un point donné*

$while(d'' < \alpha)$: //x_i *: tous les points du dataset*

$Voisin++;$

5- $d(a,b) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$ // d: *distance euclidienne entre deux points*

6- Calcul de nombre de voisin en commun entre les clusters

Tanque i < k :

7- $C(c1,c2) = \frac{maxD}{pc}$ //$C(c1,c2)$: *le taux de chevauchement entre cluster1 et cluster2*

//$maxD$*: la densité la plus grande.*

//pc *: nombre de point en commun.*

8- $max(d''(ai,bi) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$)

// d'' : *distance entre tous les points des deux clusters qui ont un chevauchement faible*

Fin tanque

9-application de k-means classique en utilisant centroïdes sélectionnés.

Fin

4. Matrice de confusion

La matrice de confusion représente une métrique de performance du modèle d'apprentissage automatique utilisée pour déterminer quel modèle est le plus efficace pour identifier les relations et les modèles entre les variables d'un ensemble de données en fonction des données d'entrées ou de formation. Il est défini comme le rapport des vrais positifs et des vrais négatifs à toutes les observations positives et négatives [55] [56].

		Classe Prédite	
		Classe = Oui	Classe = Non
Classe Réelle	Classe = Oui	Vrai positive	Faux Négative
	Classe = Non	Faux Positive	Vrai Négative

Figure 50 Matrice de confusion

• True Positive (TP) : les vrais positifs mesurent la précision avec laquelle le modèle prédit la classe des positifs.

• False Positive (FP) : un faux positif se produit lorsque le modèle prédit qu'une instance appartient à une classe à laquelle elle n'appartient pas réellement.

• True Negative (TN) : un vrai négatif est un résultat que le modèle prédit correctement comme étant négatif.

• False Negative (FN) : les faux négatifs se produisent lorsque le modèle prédit qu'une instance est négative alors qu'elle est en réalité positive.

Informellement, La matrice de confusion est la fraction des prédictions correctes de notre modèle.

Formellement, La matrice de confusion est définie par :

$$Accuracy = \frac{nombre\ de\ prédictions\ correctes}{nombre\ total\ de\ prédictions} \dots\dots\dots\dots (12)$$

Pour la classification binaire, La matrice de confusion peut également être calculée en termes de positifs et de négatifs comme suit :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots\dots (13)$$

L'idée est d'utiliser la matrice de confusion comme un moyen de comparaison entre le k-means clustering classique et la nouvelle approche.

5. Analyse de silhouette

La silhouette est une mesure des performances d'un algorithme de clustering qui étudie la distance de séparation entre les clusters résultants [34] [57]. Après avoir calculé le coefficient de silhouette (Chapitre 2 - analyse de silhouette) pour chaque point de l'ensemble de données on calcule la distance moyenne entre chaque point et tous les autres points du même cluster, ensuite on calcule la distance moyenne du même point avec tous les autres points des autres clusters (voir figure 50).

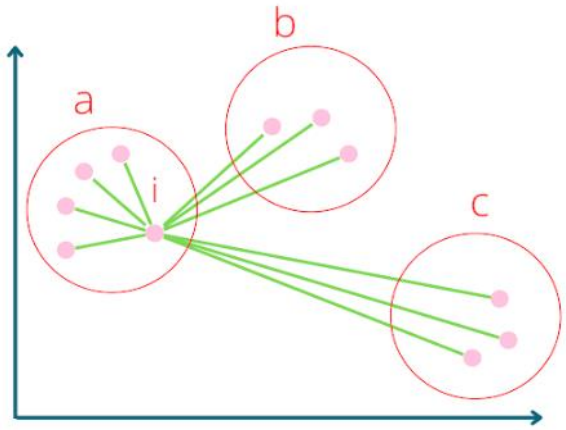

Figure 51 analyse de silhouette

Le tracé de silhouette affiche une mesure de la proximité de chaque point d'un cluster par rapport aux points des clusters voisins, offrant un moyen visuel d'évaluer des paramètres tels que la méthode du nombre de clusters. La métrique a une plage de [-1, 1].

Dans notre approche on va utiliser cette méthode après l'application de l'algorithme, pour vérifier que nos clusters sont bien séparés et il y'a pas de chevauchement entre eux, ce qui implique que les centroïdes initiaux ont étaient bien choisis [34] [57].

6. Conclusion

Dans ce chapitre on a présenté une nouvelle méthode pour pallier aux deux limites les plus importante de k-means clustering, on a vu un rappel sur les notions de base du k-means, et on a fini par aborder l'algorithme DDK-means.

Dans le dernier chapitre on va voir l'implémentation de cette nouvelle idée.

Chapitre 4

L'implémentation de DDK-means

1. Introduction

Ce dernier chapitre va être consacré à l'implémentation de notre approche tout en comparant à chaque étape nos résultats avec ceux trouvés dans l'algorithme de k-means classique.

2. Outils utilisés

Jupyter : JupyterLab est un environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données. Son interface flexible permet aux utilisateurs de configurer et d'organiser des flux de travail en science des données.

Python : Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique [58].

Base de données utilisé : dans l'implémentation de l'algorithme DDK-means on a opté pour une base de données qui a été conçu par 'numpy' spécialement pour la validation des clusters, elle contient deux caractéristiques (features), une target (valeur cible) a trois valeur (0, 1, 2), ce dataset contient en total 500 observations.

3. Implémentation de DDK-means

Notre approche DDK-means a été utilisé et tester sur une base de données à seulement deux caractéristiques, reste à l'améliorer pour le traitement des autres bases de données.

3.1 Détection des anomalies

On commence par détecter les valeurs aberrantes en utilisant l'algorithme d'isolation Forest (voir chapitre 3), dans la figure 51 les points 'violet' représentent des anomalies.

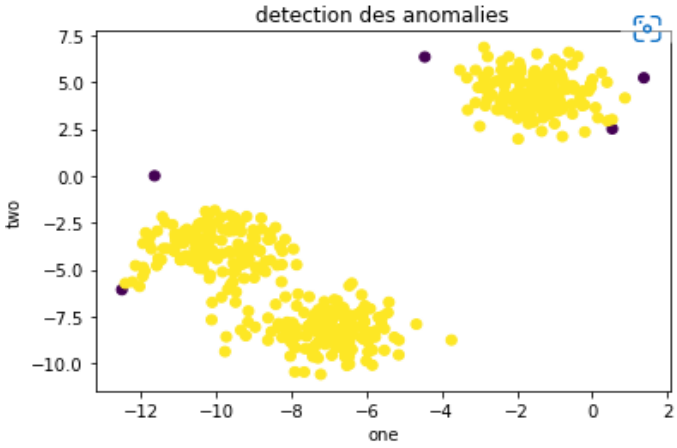

Figure 52 Détection des anomalies

3.2 Calcul de diagonale et alpha

Pour calculer la diagonale il faut d'abord déterminer les quatre coins du dataset pour pouvoir ensuite calculer la distance entre le point b et le point c (voir figure 52).

```
Calculer la diagonale (distance entre B et C) et alpha:
    La distance euclidienne entre b & c =  11.957033823694669
    alpha = 3.985677941231556

Mainteant on affiche les points en formant un rectangle
```

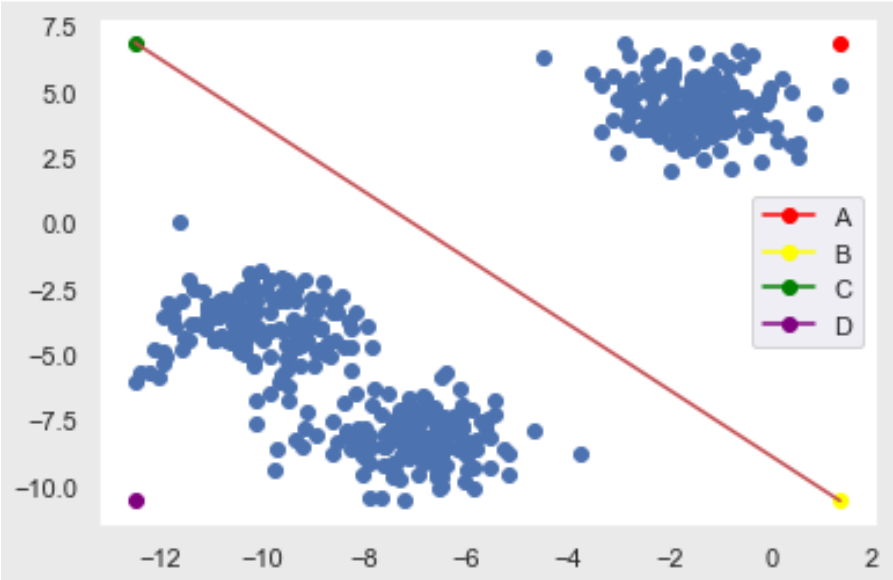

Figure 53 Calcul de la diagonale et la valeur alpha

3.3 Calcul de la densité

Pour calculer la densité d'un point donné, on compte le nombre de point qui se trouve dans un rayon alpha (voir figure 53), (on a pris une partie de la base des données juste pour visualiser ce qui va se passer avec tous les points du dataset).

Pour le calcul des distances on a utilisé la distance euclidienne (voir chapitre 2).

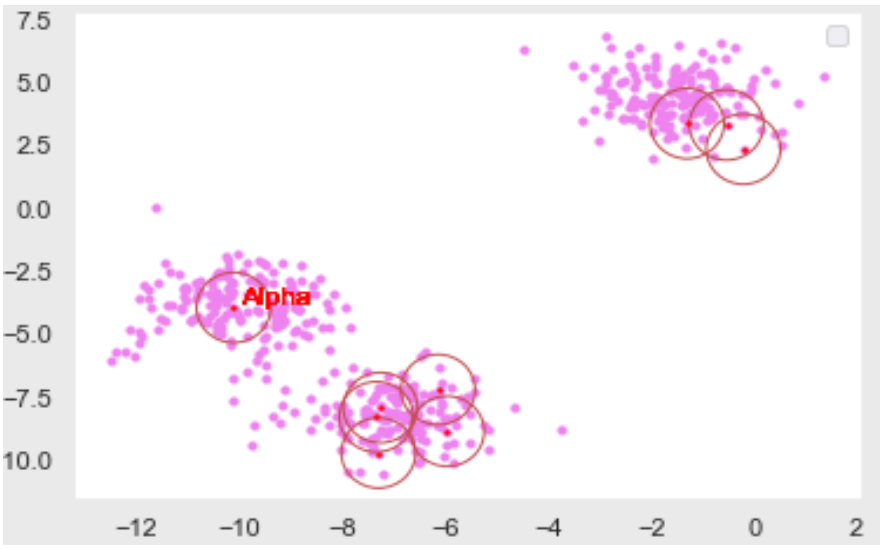

Figure 54 Calcul de densité

3.4 Choix des deux premiers points

Après le calcul de la densité on récupère la liste des voisins de chaque point pour ensuite construire des mini cluster avec ses derniers.

Pour chaque deux mini clusters, on calcule le nombre de point en commun entre eux (voir figure 54), pour ensuite évaluer le taux de chevauchement (voir chapitre 3), enfin on prend les deux clusters qui ont le moins de points en commun et on calcule la distance entre chaque deux points des deux ensembles, et on prend les deux points qui ont une grande distance entre eux.

En suivant cette méthode on aura choisi deux points qui ont une grande densité, et qui sont loin l'un de l'autre (voir figure 54) les points rouges représentent les deux premiers centroïdes.

```
le nombre de voisin du premier point est :  230
le nombre de voisin du premier point est:  295
le chevauchement est : 3
le taux de chevauchement entre les deux clusters est:  98.33333333333333
```

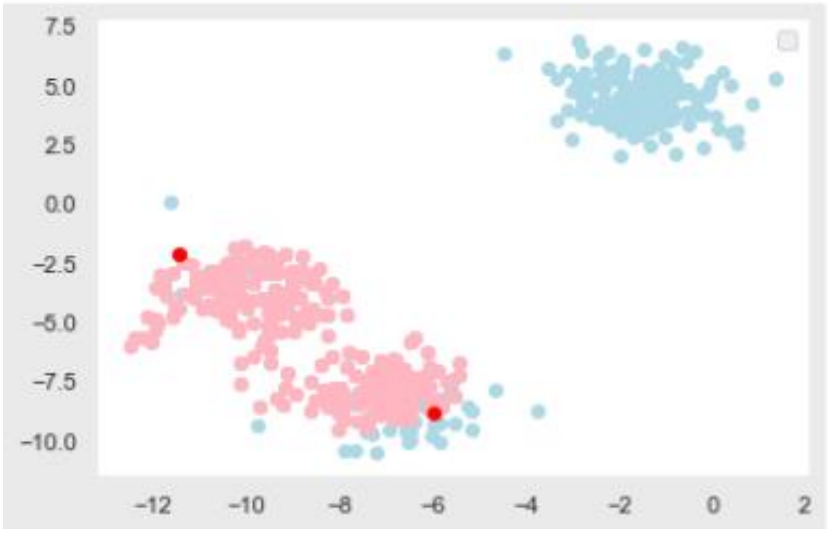

Figure 55 Désignation des deux premiers centroïdes

Une fois que les deux premiers centroïdes sont désignés on répète la même chose pour trouver le troisième centroïdes qui sera loin des deux premiers centroïdes, et qui n'a pas de chevauchement avec les deux clusters des deux autre centroïdes (dans ce cas on a trois clusters). (Voir figure 55) les trois points rouges représentent les centroïdes initiaux.

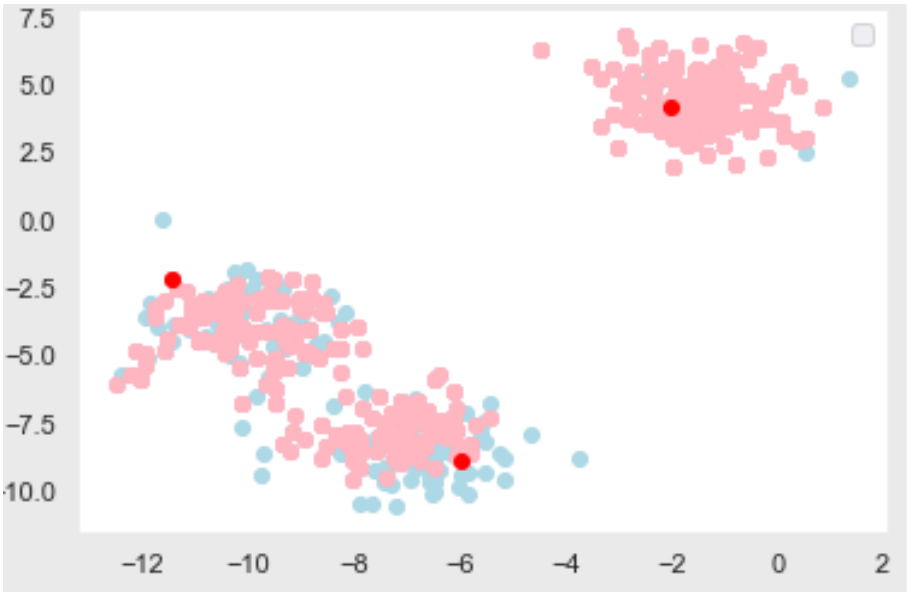

Figure 56 Les trois centroïdes initiaux du dataset

3.5 Application de k-means

Dans cette dernière étape on applique l'algorithme de k-means classique avec comme centroïdes initiaux les points qu'on a désigné dans l'étape précédente (voir figure 56).

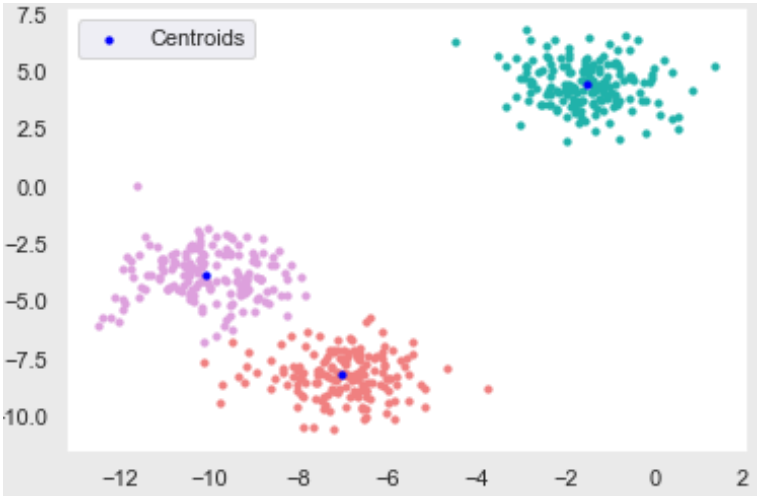

Figure 57 Application de k-means

4. Analyse et comparaison

L'idée est de comparer l'efficacité de la nouvelle méthode face aux limites de k-means classique.

On commence par générer des valeurs aléatoires comme anomalies, et on les injecte à notre base de données (voir figure 57).

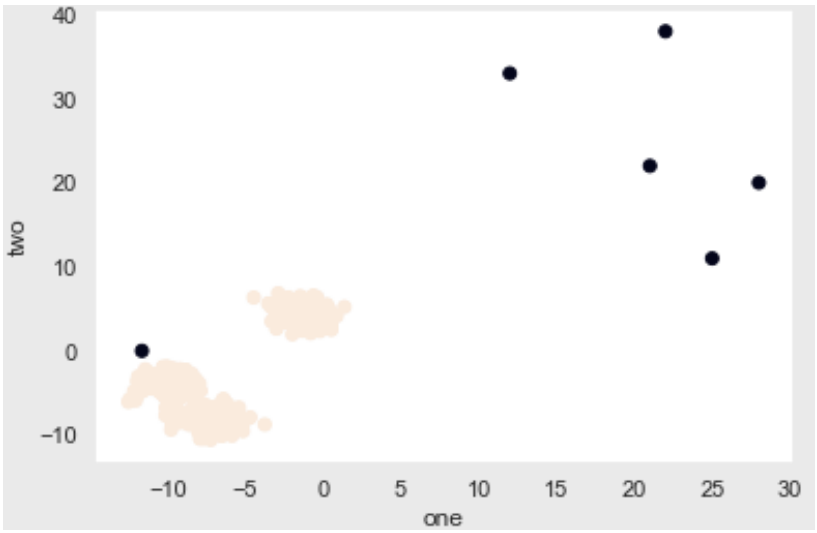

Figure 58 Visualisation de la nouvelle dataset avec des anomalies

Sur cette nouvelle base donnée on applique notre approche ainsi que l'algorithme de k-means classique et on compare les résultats.

Dans la figure (58), On voit que le k-means a ignoré les anomalies, ce qui a mené à former le troisième cluster qu'avec des valeurs aberrantes, chose qui est impossible avec notre approche puisque on détecte les outliers et on les supprime avant le choix des centroïdes et l'assemblement des clusters (voir figure 59).

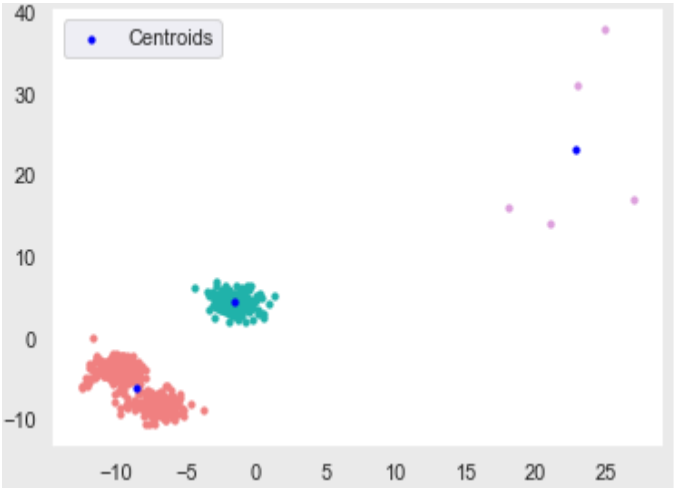

Figure 59 K-means Classique face aux anomalies

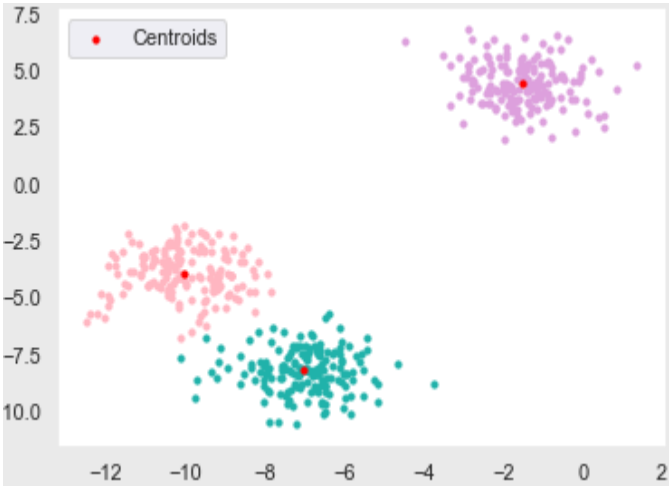

Figure 60 la nouvelle approche face aux anomalies

Pour voir l'efficacité de l'approche DDK-means dans le choix des centroïdes initiaux on a laissé les anomalies dans le dataset et appliqué DDK-means (voir figure 60), on voit bien qu'il a choisi les points (rouge) qui ont une grande densité et qui sont très loin les uns des autres comme centroïdes initiaux.

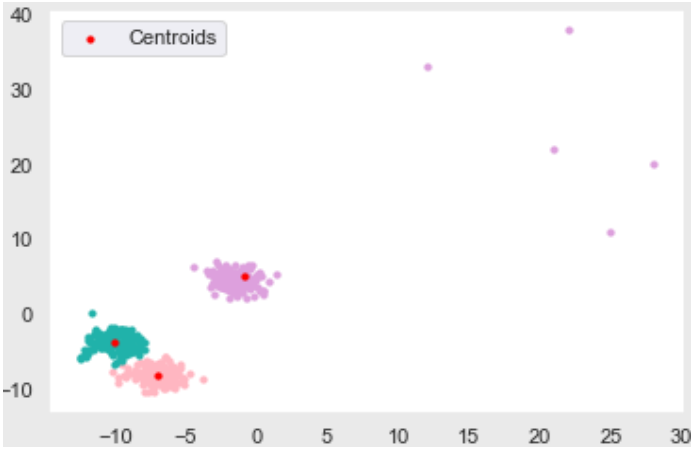

Figure 61 application de la nouvelle méthode en laissant les anomalies

4.1 Comparaison inertie

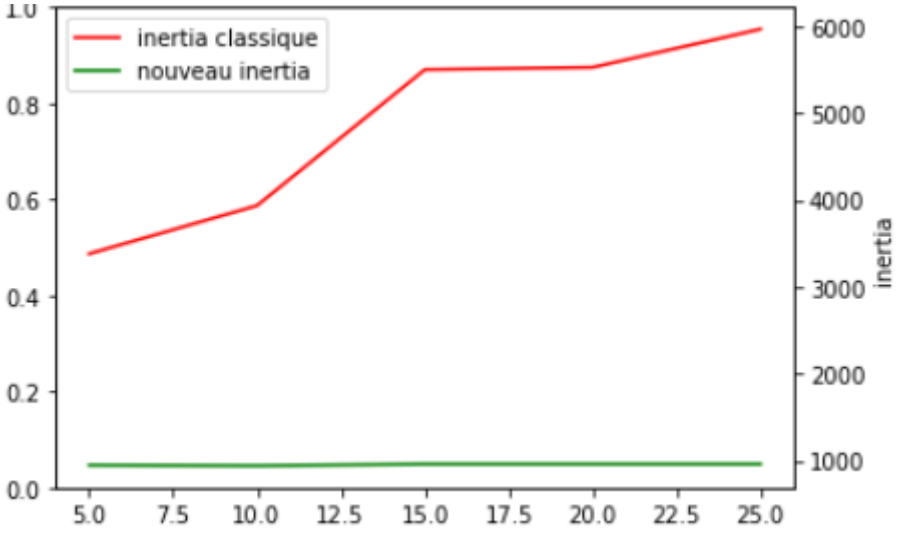

Figure 62 Comparaison entre les inerties

Dans la figure 61, on a fait plusieurs itérations en changeant à chaque fois le nombre d'anomalie dans le dataset pour visualiser et comparer la somme des distances aux carrés (inertia) (voir chapitre 2) de k-means classique avec celle obtenue par la nouvelle approche.

D'après le graphe (figure 61) on peut constater que les résultats donnés par la nouvelle méthode (schéma bleu) sont stables (entre 941 et 958) par rapport à celle donnée par le k-means classique qui ont doublés d'inertia entre (5 et 25) outliers chose qui n'arrive pas dans notre approche vue qu'on élimine les valeurs aberrantes au début de l'étude.

4.2 Comparaison temps d'exécution

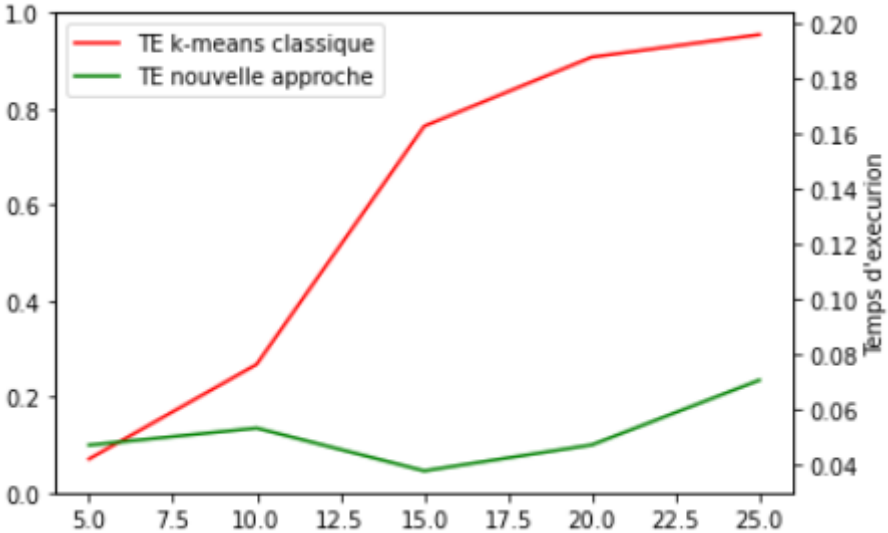

Figure 63 Comparaison de temps d'exécution

En observant la figure 62, on distingue clairement une hausse de temps d'exécution chaque fois que le nombre d'outliers augmente dans le k-means classique a l'inverse le temps d'exécution de la nouvelle approche reste stable même en ajoutant les nouveaux outliers.

5. Conclusion

Cette dernière section a été consacrée à l'implémentation de la nouvelle approche (DDK-means) en utilisant des bibliothèques python appropriées à la gestion et la visualisation des données.

En changeant plusieurs paramètres on a pu tracer des graphes pour comparer entre le k-means classique et la méthode DDK-means.

Conclusion Générale

L’intelligence artificielle est l’une des disciplines à avoir le plus évolué ces dernières années. Notre rapport lui a donc été entièrement dédié. Nous avons abordé les différents domaines dans lesquels cette évolution technologique a pris le dessus et nous avons focalisé notre travail sur un des algorithmes de la classification.

Le travail que nous avons effectué a été principalement centré sur l’algorithme de k-means, nous avons ainsi arborer les différentes caractéristiques de ce dernier passant des étapes et limites à ses performances.

Le dessein de cette étude a été de rendre le k-means plus performant et plus intelligent et ceux en partant d’un algorithme de k means classique, pour se faire nous avons utilisé de multiple méthode et effectué plusieurs tentatives défiant les deux plus grandes limites du k-means classique et qui sont la présence des anomalies dans la base de données et le choix des centroïdes initiaux, une tache assez sensible et qui demande beaucoup de précision dans l’unique but de proposer des résultats à la hauteur.

Dans le processus de rendre le k-means plus performant dans le choix des centroïdes et résistant face aux anomalies, nous avons calculé à chaque fois la densité le chevauchement ainsi que la densité, pour avoir enfin des résultats plutôt satisfaisants.

Les travaux menés dans cette thèse peuvent être poursuivis par une méthode pour choisir le bon nombre de cluster et comme ça on aura traité les trois limites majeures de k-means clustering.

L’algorithme va être tester sur plusieurs bases de données en changeant à chaque fois ces paramètres, pour ensuite faire un résumé général sur l’efficacité de la méthode face à différents dataset avec différents type et quantité de donnés.

Références

[1] RituSharma, ArpitKumar et CIndy chuah, «Turning the blackbox into a glassbox: An explainable machine learning approach for understanding hospitality customer, » *International Journal of Information Management Data Insights*, pp. 2-3, 2021.

[2] Z. Mohammed, Artificial inteligence Definition ethics and standard, the british university in egypt, 2018-2019.

[3] S. Ritu, k. Arpit et C. Cindy, «Turning the blackbox into a glassbox: An explainable machine learning approach for understanding hospitality customer, » *International Journal of Information Management Data Insights*, vol. 1, n° %1100050, 2021.

[4] Z. Messaoudi, « Spiria, » 22 Janvier 2020. [En ligne]. Available: https://www.spiria.com/fr/blogue/intelligence-artificielle/3-etapes-essentielles-apprentissage-automatique-machine-learning/.

[5] «Importance of AI in Healthcare Sector, » Data Flair, [En ligne]. Available: https://data-flair.training/blogs/ai-in-healthcare-sector/.

[6] Wei Yang, Hua Long, Lihua Ma et Huifang Sun, «Research on Clustering Method Based on Weighted Distance Density and K-Means, » *Procedia Computer Science,* 2020.

[7] « Quel algorithme d'apprentissage automatique devez-vous utiliser par type de problème ?» ichipro, [En ligne]. Available: https://ichi.pro/fr/quel-algorithme-d-apprentissage-automatique-devez-vous-utiliser-par-type-de-probleme-181665963074918.

[8] JyotismitaChaki, S.Thillai Ganesh, S.KCidham et S.Ananda Theertan, «Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review,» *Journal of King Saud University - Computer and Information Sciences,* 2020.

[9] J. cadima, « principal component analysis: a review and recent developments,» *philosophical transactions A,* 2015-2016.

[10] KRISTINA P. SINAGA et MIIN-SHEN YANG, «Unsupervised K-Means Clustering Algorith, » *IEEE Access,* pp. [1-11], 2020.

[11] « svm,» Data Analytics Post, [En ligne]. Available: https://dataanalyticspost.com/Lexique/svm/.

[12] Yunming Ye, Joshua Zhexue Huang, Xiaojun Chen, Shuigeng Zhou, Graham Williams et Xiaofei Xu1, «Neighborhood Density Method for Selecting Initial Cluster Centers in K-Means Clustering, » pp. [1-9], 2006.

[13] «K-Nearest neighbor clustering, » [En ligne]. Available: https://vatsalparsaniya.github.io/ML_Knowledge/Nearest_neighbours/Readme.html.

[14] M. Bazmara, «The flowchart of K nearest neighbor classifier procedure, » Research Gate, [En ligne]. Available: https://www.researchgate.net/figure/The-flowchart-of-K-nearest-neighbor-classifier-procedure_fig2_237080861.

[15] Deepanshi, «All you need to know about your first Machine Learning model – Linear Regression, » 25 May 2021. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/.

[16] P. Sharma, «A Beginner's Guide to Hierarchical Clustering and how to Perform it in Python, » 27 May 2019. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/.

[17] J. Admin, « Artificiel Intelligence Demystified, » 23 December 2016. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2016/12/artificial-intelligence-demystified/.

[18] K. P, D. M et D. A, Fundamentals of Clinical Data Science [Internet]., 2019.

[19] « RÉDUCTION DE DIMENSIONNALITÉ, » [En ligne]. Available: https://dataanalyticspost.com/Lexique/reduction-de-dimensionnalite/.

[20] D. Polzer, «7 of the most usd regression algorithms and how to choose the right one, » 21 July 2021. [En ligne]. Available: https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3.

[21] Jyotismita Chakia, S.Thillai Ganesh, S.KCidham et bS.Anand Theertan, «Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review,» *Journal of King Saud University - Computer and Information Sciences*, p. 11, 2020.

[22] s. Ranganathan, «Improvements to k-means clustering, » *Tampere university of technology*, p. 5, 2013.

[23] P. Sharma, «K Means Clustering Simplified in Python, » *Analytics Vidhya*, 2021.

[24] PreetiArora, D. Dr et S. Varshney, «Analysis of K-Means and K-Medoids Algorithm For Big Data, » *Procedia Computer Science*, vol. 78, pp. 507-512, 2016.

[25] A. Sharma, «How to Master the Popular DBSCAN Clustering Algorithm for Machine Learning, » 8 September 2020. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/.

[26] « Reinforcement Learning Tutorial,» Java T Point, [En ligne]. Available: https://www.javatpoint.com/reinforcement-learning.

[27] D. Bouchra et B. A. E. Kazi-Tani, *Une nouvelle approche de clustering pour améliorer les performances des réseaux de capteurs sans fil*, Tlemcen, Algérie : Université Abou Bakr Belkaid– Tlemcen, 2019.

[28] «Clustering, » Scikit Learn, [En ligne]. Available: https://scikit-learn.org/stable/modules/clustering.html.

[29] T. M. Ghazal, M. Z. Hussain, R. A. Said, A. Nadeem, M. K. Hasan, M. Ahmad, M. A. Khan et M. T. Naseem, «Performances of K-Means Clustering Algorithm with Different Distance Metrics, » *TechScience Press*, 2021.

[30] .. N. D. Vagisha Gupta, «Deep similarity learning for disease prediction, » *Trends in Deep Learning Methodologies*, 2021.

[31] F. Müller, «Simple Cluster Analysis using K-Means and Python, » relataly.com, 2021. [En ligne]. Available: https://www.relataly.com/simple-cluster-analysis-using-k-means-with-python/5070/.

[32] «K-means Clustering Algorithm: Applications, Types, and How Does It Work? » simplilearn, 24 december 2021. [En ligne]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm.

[33] E. Umargono, J. E. Suseno et V. G. S.K, «K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median, » 2019.

[34] P. J.Rousseeuw, «Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,» *Journal of Computational and Applied Mathematics,* vol. 20, p. 1987, 53-65.

[35] «Selecting the number of clusters with silhouette analysis on KMeans clustering, » Scikit-Learn, [En ligne]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py.

[36] K. Mahendru, «How to Determine the Optimal K for K-Means? » Analytics Vidhya, 17 june 2019. [En ligne]. Available: https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb.

[37] B. Aubaidan, M. Mohd et M. Albared, «COMPARATIVE STUDY OF K-MEANS AND K-MEANS++ CLUSTERING ALGORITHMS ON CRIME DOMAIN, » *Journal of Computer Science*, 2014.

[38] «Machine Learning | Outlier, » GeeksforGeeks, 2020. [En ligne]. Available: https://www.geeksforgeeks.org/machine-learning-outlier/.

[39] Q. Feng, Z. Zhang, Z. Huang, J. Xu, J. Wang, Improved algorithms for clustering with outliers, in: Proc. 30th International Symposium on Algorithms and Computation (ISAAC), 2019, pp. 61:1–61:12.

[40] N. S. Chauhan, «Introduction to Image Segmentation with K-Means clustering, » KDnuggets, 2019. [En ligne]. Available: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html.

[41] A. Gandotra, «Customer Segmentation using K-means, » medium, 17 Aug 2018. [En ligne]. Available: https://medium.com/@abhay.gandotra/customer-segmentation-using-k-means-d5cb17c2e7dd.

[42] « Fonctionnement de l'agrégation basée sur la densité, » ArcGIS Pro, [En ligne]. Available: https://pro.arcgis.com/fr/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm#.

[43] H. Bonthu, «An Introduction to Logistic Regression, » 11 July 2021. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2021/07/an-introduction-to-logistic-regression/.

[44] N. Dhanachandra, KhumanthemManglem et Y. JinaChanu, «Image Segmentation Using K - means Clustering Algorithm and Subtractive Clustering Algorithm, » *Procedia Computer Science*, vol. 54, pp. 764-771, 2015.

[45] R. DOMINGUES, «Machine Learning for Unsupervised Fraud Detection, » *ROYAL INSTITUTE OF TECHNOLOGY, SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION,* 2015.

[46] A. Lima, « Clustering DBSCAN dans ML | Clustering basé sur la densité, » Acervo Lima, [En ligne]. Available: https://fr.acervolima.com/clustering-dbscan-dans-ml-clustering-base-sur-la-densite/.

[47] S. Ray, «Understanding Support Vector Machine (SVM) algorithm from examples (along with code), » 13 September 2017. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[48] N. Sharma, «Importance of Distance Metrics in Machine Learning Modelling, » Towards Data Science, 13 january 2019. [En ligne]. Available: https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d

[49] Sinha, S. K. 1979. « Outliers in Statistical Data (Vic Barnett and Toby Lewis) ». SIAM Review, vol. 21, no 4, p. 576-577.

[50] B. Auffarth, Machine Learning for Time-Series with Python: Forecast, predict and detect anomalies with state-of-the-art machine learning, Packt Publishing - ebooks Account, 2021.

[51] «What is Outlier in data mining, » [En ligne]. Available: https://www.javatpoint.com/what-is-outlier-in-data-mining.

[52] F. T. Liu, K. M. Ting et Z.-H. Zhou, Isolation Forest, Eighth IEEE International Conference on Data Mining, 2008.

[53] M. A. I. Khan, Anomaly Detection with Isolation Forest and Kernel Density Estimation, Machine Learning Algorithms, 2022.

[54] Y. C. A. B. R. C. Maurras Togbe, Etude comparative des méthodes de détection, HAL open science, 2020.

[55] C. D. et J. G, «The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, » BMC Genomics, pp. 6-13, 2020.

[56] C. D., T. N. et J. G., «The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, » BioData Mining, pp. 1-22, 2021.

[57] W. A. M. A. A. B. Zahid Ansari, «Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions, » World of Computer Science and Information Technology Journal (WCSIT), pp. 217-226, 2011.

[58] «What is Python? Executive Summary, » python, [En ligne]. Available: https://www.python.org/doc/essays/blurb/.