Université Cadi Ayyad
École Supérieure De Technologie-Safi
Département : Informatique
Filière : LP Ingénierie des Systèmes d'information et Réseaux

Rapport du Projet

# Implémentation de l'algorithme DBSCAN en Python

***Réalisé par :***

**M. Yahya LEMKHARBECH**
**M. EL AOUMARI Abdelmoughith**

***Encadré par :***

**Mme. MOUNIR Ilham**

Année Universitaire : 2023/2024

# Introduction générale

L'évolution rapide des technologies de l'information a entraîné une explosion de données dans divers secteurs, créant un besoin impératif de méthodes intelligentes pour extraire des connaissances exploitables. Ce rapport se focalise sur l'implémentation de l'algorithme DBSCAN en Python, une contribution significative au domaine du data mining et de l'analyse de clusters.

Dans ce contexte, explorons les bases du data mining, de l'apprentissage automatique et de la science des données. Le data mining cherche à découvrir des motifs et des relations dans de vastes ensembles de données, tandis que l'apprentissage automatique permet aux machines d'apprendre des modèles sans intervention humaine constante. La science des données fusionne ces disciplines pour analyser, interpréter et extraire des informations significatives, offrant ainsi une vision holistique de l'analyse de données.

Ces approches distinctes offrent des avantages uniques : le data mining identifie des tendances cachées, l'apprentissage automatique crée des modèles prédictifs robustes, et la science des données offre une compréhension approfondie. Dans le domaine du data mining, les méthodes descriptives résument les caractéristiques importantes des données, tandis que les méthodes prédictives anticipent les futurs événements en se basant sur des modèles appris.

Focalisons notre attention sur le clustering, une technique cruciale du data mining visant à regrouper des données similaires pour identifier des structures sous-jacentes. L'algorithme DBSCAN, Density-Based Spatial Clustering of Applications with Noise, se distingue par sa capacité à identifier des clusters de formes complexes tout en détectant les points isolés. Pour illustrer concrètement son application, considérons un exemple d'ensemble de points représentant des positions géographiques, démontrant ainsi l'efficacité de DBSCAN dans des contextes spécifiques.

En abordant l'implémentation pratique, nous plongerons dans les technologies utilisées pour concrétiser notre projet. La présentation détaillée mettra en lumière les choix techniques et les étapes clés du processus de mise en œuvre. En conclusion, nous récapitulerons les points saillants du rapport et esquisserons des perspectives pour des développements futurs dans ce domaine dynamique.

# Table des matières

# Table des figures

# I- Généralités

## I.1 Introduction

Dans cette section, nous explorerons les fondements du data mining, de l'apprentissage automatique et de la science des données, trois domaines interconnectés qui jouent un rôle crucial dans l'analyse intelligente des données. Le data mining, en tant qu'approche clé, vise à extraire des informations utiles à partir de grands ensembles de données, révélant des tendances, des modèles et des relations cachées. Parallèlement, l'apprentissage automatique offre la capacité aux machines d'apprendre à partir de données, permettant ainsi la création de modèles prédictifs. La science des données fusionne ces disciplines pour fournir une compréhension globale des processus d'analyse de données.

Nous aborderons également les avantages distincts de chaque élément de ce triptyque. Le data mining, en identifiant des schémas cachés, ouvre la voie à une prise de décision informée. L'apprentissage automatique, quant à lui, s'attaque à la création de modèles capables de généraliser à partir de données, améliorant ainsi la capacité prédictive. Enfin, la science des données offre une approche holistique en combinant ces disciplines pour une compréhension approfondie des phénomènes étudiés.

Au sein du data mining, nous explorerons également les deux approches majeures : des méthodes descriptives qui cherchent à résumer les caractéristiques des données, et des méthodes prédictives qui visent à anticiper les futurs événements. Ces distinctions seront cruciales pour éclairer notre compréhension des technologies de data mining dans la section à venir, tout en jetant les bases nécessaires pour introduire le clustering et l'algorithme DBSCAN, au cœur de notre projet d'implémentation.

## I.2 Data Mining

### I.2.1 Définition de Data Mining

Le Data Mining, également connu sous le nom de fouille de données, est une discipline puissante qui repose sur l'exploration approfondie et l'analyse de vastes ensembles de données afin de

découvrir des modèles, des tendances et des relations significatives. Il s'agit d'un processus itératif qui utilise des méthodes statistiques, mathématiques et informatiques avancées pour extraire des informations utiles à partir de données brutes. Le Data Mining vise à révéler des connaissances enfouies dans les données, offrant ainsi des perspectives nouvelles et exploitables pour la prise de décision.

### I.2.2 Fondamentaux et Bases du Data Mining

Les fondamentaux du Data Mining reposent sur deux approches clés : les méthodes descriptives et les méthodes prédictives.

#### Méthodes Descriptives

Les méthodes descriptives sont axées sur la compréhension globale des données. Elles visent à résumer et à décrire les caractéristiques importantes du jeu de données. Voici quelques-unes des techniques descriptives les plus utilisées :

— **Clustering :** Classification des données en groupes homogènes, par exemple, regrouper des clients similaires basés sur leurs comportements d'achat.

— **Règles d'Association :** Identification de relations fréquentes entre les variables, par exemple, la corrélation entre l'achat de pain et de lait dans un supermarché.

#### Méthodes Prédictives

Les méthodes prédictives, d'autre part, sont orientées vers la modélisation des données pour anticiper les futurs événements. Elles utilisent des techniques telles que :

— **Régression :** Modélisation des relations fonctionnelles entre les variables pour faire des prédictions basées sur ces relations.

— **Arbres de décision :** Structures arborescentes qui prennent des décisions séquentielles basées sur les caractéristiques des données.

— **Réseaux de neurones :** Modèles inspirés du fonctionnement du cerveau humain, capables d'apprendre des modèles complexes à partir des données.

Cette sous-section jettera les bases nécessaires pour comprendre les concepts essentiels du Data Mining. Elle constituera un fondement solide pour la suite de notre exploration, notamment l'introduction au clustering et à l'algorithme DBSCAN.

## I.3 Algorithmes de Clustering

### I.3.1 Algorithmes de Partitionnement

Les algorithmes de partitionnement visent à diviser l'ensemble de données en plusieurs groupes distincts, appelés partitions. Voici quelques-uns des algorithmes de partitionnement les plus utilisés :

— **K-Means :** Partitionne les données en k clusters en minimisant la variance intra-cluster.

— **K-Medoids :** Similaire à K-Means, mais utilise des objets réels du cluster comme centres au lieu de moyennes.

### I.3.2 Algorithmes Hiérarchiques

Les algorithmes hiérarchiques construisent une hiérarchie de clusters, souvent sous la forme d'un dendrogramme. Voici quelques exemples d'algorithmes hiérarchiques :

— **CAH (Classification Ascendante Hiérarchique) :** Commence avec chaque point comme un cluster distinct et fusionne progressivement les clusters similaires.

— **CDH (Classification Descendante Hiérarchique) :** Commence avec un cluster global et divise récursivement les clusters en sous-clusters.

### I.3.3 Algorithmes basés sur la Densité

Les algorithmes basés sur la densité identifient des zones denses dans l'espace des données. Voici un exemple d'algorithme basé sur la densité :

— **DBSCAN (Density-Based Spatial Clustering of Applications with Noise) :** Identifie les clusters en se basant sur la densité des points.

### I.3.4 Comparaison des Algorithmes de Clustering

Dans cette sous-section, une petite comparaison entre les algorithmes de regroupement est présentée

TABLE I.1 – Comparaison des Catégories d'Algorithmes de Clustering

| **Critères** | **Algorithmes de Partitionnement** | **Algorithmes Hiérarchiques** | **Algorithmes basés sur la Densité** |
|---|---|---|---|
| Sensibilité à la forme des clusters | Variable | Peut varier | Robuste aux formes arbitraires |
| Gestion des bruits | Sensible aux points isolés | Peut être sensible | Gère efficacement les points isolés |
| Scalabilité | Bien pour un nombre modéré de clusters | Complexité élevée | Bien pour grandes bases de données |
| Interprétabilité | Centres de clusters faciles à interpréter | Structure hiérarchique | Structure basée sur la densité |

### I.3.5 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

#### Définition de DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering qui identifie les clusters dans un ensemble de données en se basant sur la densité des points. Contrairement aux méthodes de clustering traditionnelles, DBSCAN peut identifier des clusters de formes complexes et est résistant aux outliers. Il catégorise les points en trois types : les points noyaux (core points), les points frontières (border points), et les points de bruit (noise points). La densité d'un cluster est mesurée par le nombre de points à proximité d'un point donné, défini par un rayon spécifique.

### I.3.6 L'Algorithme Nearest Neighbors avec DBSCAN

#### Définition de l'Algorithme Nearest Neighbors

L'algorithme Nearest Neighbors (k-NN) est une méthode de classification et de régression basée sur la proximité des points dans l'espace des caractéristiques. Pour chaque point d'un ensemble de données, l'algorithme identifie les k voisins les plus proches en termes de distance. La classe ou la valeur cible du point est ensuite déterminée en fonction des classes ou des valeurs cibles majoritaires parmi ses voisins.

### Utilisation de l'Algorithme Nearest Neighbors avec DBSCAN

L'algorithme Nearest Neighbors peut être utilisé de manière synergique avec DBSCAN pour améliorer la robustesse du clustering. Après l'identification des clusters par DBSCAN, l'algorithme Nearest Neighbors peut être employé pour déterminer les relations de proximité entre les points au sein de chaque cluster.

Cela peut être utile dans les cas où la densité des points n'est pas uniforme à l'intérieur d'un cluster identifié par DBSCAN. En utilisant l'algorithme Nearest Neighbors, on peut affiner les relations de voisinage en considérant les k voisins les plus proches de chaque point dans un cluster spécifique.

Par exemple, pour chaque point d'un cluster identifié par DBSCAN, l'algorithme Nearest Neighbors peut être utilisé pour trouver les k points les plus proches à l'intérieur de ce cluster. Cela peut aider à mieux comprendre la structure interne du cluster et à identifier d'éventuelles sous-structures ou groupes plus denses.

L'utilisation de l'algorithme Nearest Neighbors avec DBSCAN offre ainsi une approche complémentaire pour explorer la proximité des points au sein des clusters identifiés, améliorant ainsi la précision et la granularité de l'analyse de clustering.

### Processus de l'Algorithme DBSCAN

Considérons l'ensemble de données suivant :

$$\{(1,2),(2,2),(2,3),(8,7),(8,8),(25,80)\}$$

Table I.2 – Ensemble de Données pour DBSCAN

| **Coordonnées (x, y)** |
|---|
| (1,2) |
| (2,2) |
| (2,3) |
| (8,7) |
| (8,8) |
| (25,80) |

1. **Sélection d'un point de départ non visité :** Choisissons le point $(1, 2)$ comme point de départ.
2. **Voisinage dans un rayon :** Les points dans le voisinage de $(1, 2)$ avec un rayon de $\epsilon = 3$ sont $(2, 2)$ et $(2, 3)$.
3. **Vérification de la densité :** Le nombre de points dans le voisinage est 2, ce qui dépasse le seuil. Ainsi, $(1, 2)$ est un point noyau et forme un cluster avec $(2, 2)$ et $(2, 3)$.
4. **Expansion du cluster :** Les points $(2, 2)$ et $(2, 3)$ deviennent également des points noyaux, et leur voisinage est exploré.
5. **Identification des points de bruit :** Les points $(8, 7)$, $(8, 8)$, et $(25, 80)$ sont considérés comme des points de bruit car ils ne sont pas densément connectés.
6. **Répétition :** Le processus est répété jusqu'à ce que tous les points aient été visités.

Après l'exécution de DBSCAN sur cet exemple, les clusters identifiés sont $\{(1, 2), (2, 2), (2, 3)\}$ et les points de bruit sont $\{(8, 7), (8, 8), (25, 80)\}$.

L'algorithme DBSCAN offre une approche robuste pour la détection de clusters en se basant sur la densité locale des points, ce qui le rend particulièrement adapté à des ensembles de données de formes et de densités variées.

### I.3.7 Exemple d'Application de DBSCAN

Considérons un ensemble de points dans un espace bidimensionnel, chacun représenté par des coordonnées $(x, y)$. Ces points forment un ensemble de données que nous allons utiliser comme exemple pour illustrer l'application de l'algorithme DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

— **Coordonnées des Points**

Les points de notre ensemble de données sont les suivants :

| **Point** | **Coordonnées** |
|---|---|
| $P1$ | $(3, 7)$ |
| $P2$ | $(4, 6)$ |
| $P3$ | $(5, 5)$ |
| $P4$ | $(6, 4)$ |
| $P5$ | $(7, 3)$ |
| $P6$ | $(6, 2)$ |
| $P7$ | $(7, 2)$ |
| $P8$ | $(8, 4)$ |
| $P9$ | $(3, 3)$ |
| $P10$ | $(2, 6)$ |
| $P11$ | $(3, 5)$ |
| $P12$ | $(2, 4)$ |

Ces points représentent notre ensemble de données spatial que nous allons analyser à l'aide de l'algorithme DBSCAN afin de détecter des clusters significatifs en fonction de la densité des points.

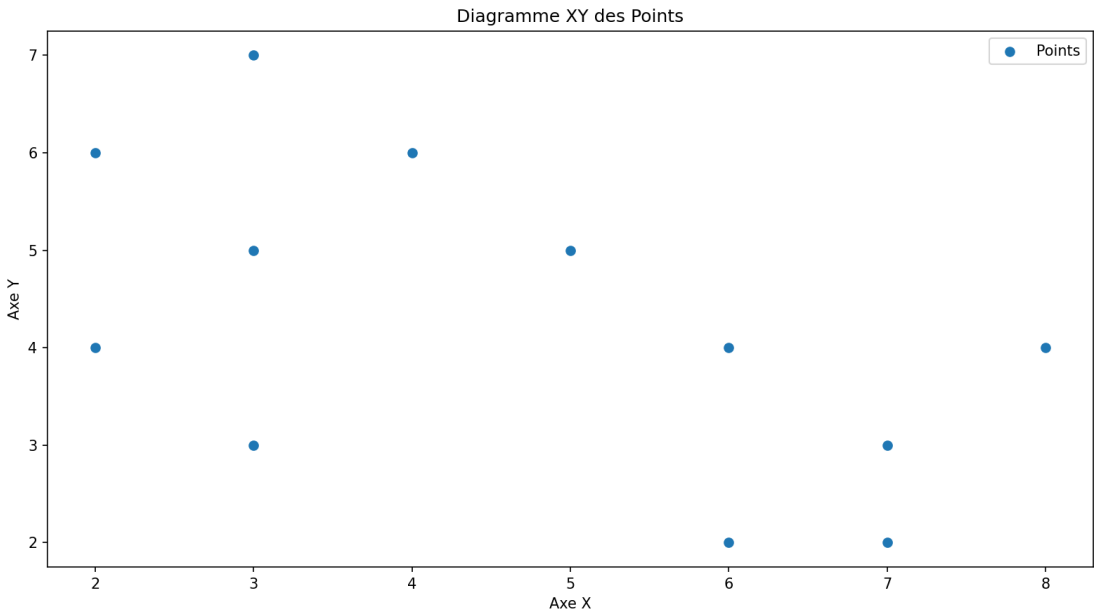


FIGURE I.1 – Representaion des points

— **Paramètres DBSCAN**

Pour cette application de DBSCAN, nous fixons les paramètres suivants :

— $\varepsilon = 1.9$ (Rayon de voisinage)

— **minPts** $= 4$ (Nombre minimum de points dans un voisinage pour former un cluster)

— **Calcul des Distances pour l'Exemple d'Application de DBSCAN**

**1. Formule Mathématique de Distance (Distance Euclidienne)**

La distance euclidienne entre deux points $(x_1, y_1)$ et $(x_2, y_2)$ dans un espace bidimensionnel est donnée par la formule :

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

**2. Matrice des Distances**

Calculons les distances entre les points de l'exemple d'application de DBSCAN :

| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | 0 | 1.41 | 2.83 | 4.24 | 5.66 | 5.1 | 5.1 | 3.61 | 4 | 3.61 | 2 | 3.16 |
| P2 | | 0 | 1.41 | 2.83 | 4.24 | 4.47 | 4.47 | 3.16 | 3.16 | 3.16 | 2.24 | 1.41 |
| P3 | | | 0 | 1.41 | 2.83 | 3.16 | 3.16 | 1.41 | 3.16 | 1.41 | 1.41 | 2.24 |
| P4 | | | | 0 | 1.41 | 2 | 2 | 1.41 | 3.61 | 1.41 | 3.16 | 1.41 |
| P5 | | | | | 0 | 3.16 | 3.16 | 3.61 | 5.66 | 4.24 | 5.1 | 4.24 |
| P6 | | | | | | 0 | 1 | 2.24 | 5.1 | 3.16 | 3.61 | 2.24 |
| P7 | | | | | | | 0 | 1.41 | 5.39 | 3.61 | 4.24 | 2.83 |
| P8 | | | | | | | | 0 | 4.47 | 2.24 | 3.16 | 1.41 |
| P9 | | | | | | | | | 0 | 2.24 | 1.41 | 0.71 |
| P10 | | | | | | | | | | 0 | 1.41 | 0.71 |
| P11 | | | | | | | | | | | 0 | 1.41 |
| P12 | | | | | | | | | | | | 0 |

La matrice des distances présente les distances euclidiennes entre tous les points de l'exemple d'application de DBSCAN avec des colonnes de largeur fixe et des lignes horizontales entre chaque ligne.

— **Construction des Clusters**

Pour construire les clusters à l'aide de l'algorithme DBSCAN, nous suivons les étapes suivantes :

1. **Identification des Voisins dans le Rayon de Voisinage**

   Pour chaque point $P_i$, identifions les points qui se trouvent dans son rayon de voisinage défini par le paramètre $\varepsilon$. Les voisins sont les points dont la distance par rapport à $P_i$ est inférieure à $\varepsilon$.

2. **Vérification du Nombre Minimum de Points (minPts)**

   Pour chaque point $P_i$, vérifions si le nombre de voisins est supérieur ou égal à minPts. Si c'est le cas, $P_i$ est considéré comme un point cœur.

3. **Extension des Clusters**

   Pour chaque point cœur $P_i$, étendons le cluster en ajoutant tous ses voisins (et les voisins de ses voisins) au cluster.

4. **Attribution des Labels aux Points**

   Chaque point est alors labellisé comme appartenant à un cluster s'il a été atteint lors de l'extension des clusters, sinon il est considéré comme un point de bruit.

**Tableau de Construction des Clusters**

| **Point** | **Voisins dans le Rayon ($\varepsilon$)** | **Cluster** |
|---|---|---|
| P1 | P2, P10 | Non |
| P2 | P1, P3, P11 | Oui |
| P3 | P2, P4 | Non |
| P4 | P3, P5 | Non |
| P5 | P4, P6, P7, P8 | Oui |
| P6 | P5, P7 | Non |
| P7 | P5, P6 | Non |
| P8 | P5 | Non |
| P9 | P12 | Non |
| P10 | P1, P11 | Non |
| P11 | P2, P10, P12 | Oui |
| P12 | P9, P11 | Non |

Ce tableau présente les voisins dans le rayon de voisinage ($\varepsilon$) pour chaque point ainsi que l'indication si le point appartient ou non à un cluster après l'application de l'algorithme DBSCAN.

— **Points Bruits, Bordures et Noyaux**

**Points Bruits :** P9

**Points Bordures :** P1, P3, P4, P6, P7, P8, P10, P12

**Points Noyaux :** P11, P5, P2

— **Représentation des clusters**

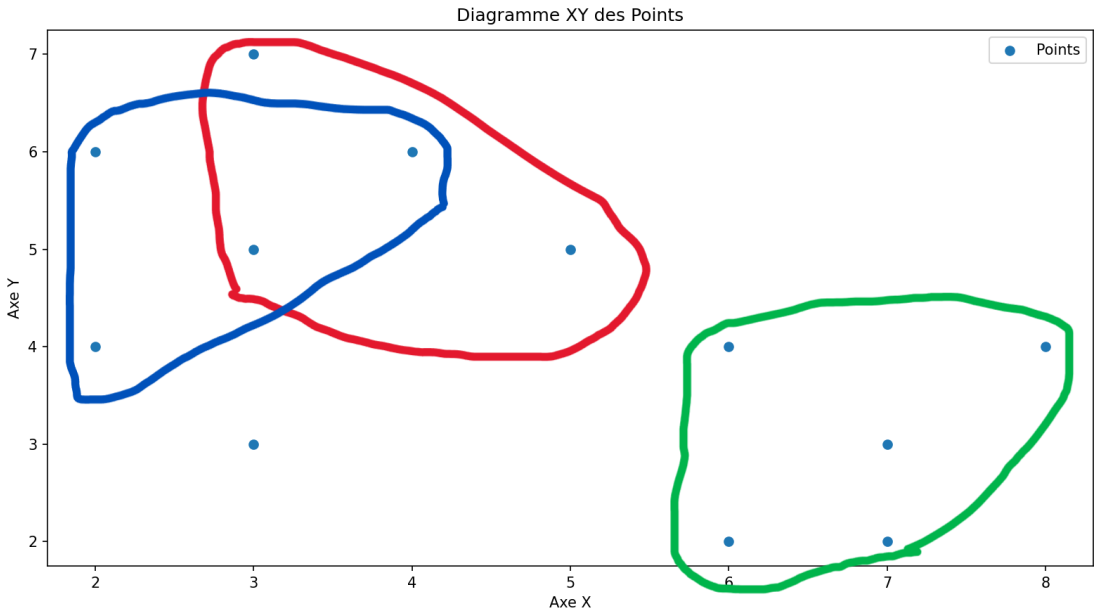


FIGURE I.2 – Representaion des clusters

## I.4 Conclusion

Le Data Mining, ou fouille de données, constitue une discipline essentielle pour extraire des informations significatives à partir de grandes quantités de données. Cette pratique, embrassant des domaines tels que l'apprentissage automatique et la science des données, permet de découvrir des motifs, des tendances et des relations cachées.

Dans le vaste domaine du Data Mining, on distingue des techniques descriptives et prédictives. Les méthodes descriptives se concentrent sur la caractérisation des données, tandis que les méthodes prédictives visent à anticiper les valeurs futures en se basant sur des modèles existants.

Parmi les technologies de Data Mining, le Clustering avec DBSCAN (Density-Based Spatial Clustering of Applications with Noise) mérite une attention particulière. DBSCAN est un algorithme de regroupement basé sur la densité qui permet d'identifier des zones de haute densité séparées par des zones de faible densité. Cette approche, reposant sur des paramètres tels que le rayon de voisinage ($\varepsilon$) et le nombre minimum de points (`minPts`), offre une classification des

données en noyaux, bordures et bruits.

En mettant en œuvre DBSCAN sur un ensemble de points bidimensionnels, représentés par des coordonnées $(x, y)$, nous pouvons observer comment les clusters se forment en fonction de la densité des points. Les paramètres $\varepsilon$ et `minPts` influent sur la classification des points en noyaux, bordures ou bruits, offrant ainsi une compréhension approfondie de la structure des données.

En conclusion, le Data Mining, avec des techniques telles que DBSCAN, offre des moyens puissants pour explorer, comprendre et exploiter la richesse des données, ouvrant la voie à des applications variées dans des domaines tels que la prise de décision, la prédiction et la découverte de connaissances cachées.

# II- Implémentation

## II.1 Introduction

Lorsque la théorie rencontre la pratique, c'est dans la section Implementation que la magie opère. Ici, nous plongeons au cœur de l'action, explorant les technologies qui ont alimenté notre implémentation et présentant le projet réalisé avec une perspective pratique.

Dans le vaste monde du Data Mining, les choix technologiques jouent un rôle crucial. Nous débuterons en examinant les technologies que nous avons délibérément sélectionnées pour façonner notre implémentation. Des langages de programmation aux bibliothèques spécialisées, chaque décision a été guidée par la volonté de transformer la théorie en application tangible.

Au-delà des choix technologiques, nous présenterons le projet que nous avons concrétisé. Les objectifs fixés, la portée du projet et son impact potentiel dans le domaine du Data Mining seront explorés. Cette partie du rapport offre un regard détaillé sur les résultats concrets de notre démarche.

En unissant ces deux aspects, nous construisons un pont entre la théorie et la pratique. L'implémentation devient ainsi une réalité palpable, intégrant les concepts de Data Mining dans un projet qui transcende les pages théoriques pour entrer dans le domaine de l'application concrète.

## II.2 Outils, langages et bibliothèques utilisés

### II.2.1 Outils utilisés

— **GitHub :** En tandem avec Git, nous avons exploité la plateforme GitHub pour la gestion collaborative de notre projet. Cette plateforme basée sur Git a favorisé le partage transparent du code source, facilitant la collaboration entre les membres de l'équipe et offrant une visibilité accrue sur l'évolution du projet.

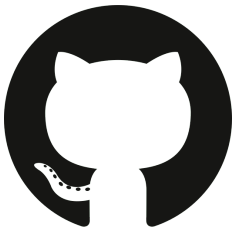

FIGURE II.1 – GitHub

— **Visual Studio Code (VSCode) :** En tant qu'éditeur de code source léger et puissant, VSCode a été l'outil de prédilection pour notre équipe de développement. Son interface intuitive et ses fonctionnalités avancées ont optimisé le processus de codage, améliorant ainsi l'efficacité et la qualité du travail.

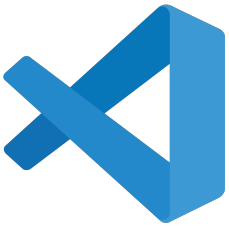

FIGURE II.2 – Visual Studio Code (VSCode)

### II.2.2 Langages de programmation

FIGURE II.3 – Logo de Python

Python est un langage de programmation polyvalent largement utilisé dans le domaine de l'informatique et de l'analyse de données. Connu pour sa simplicité syntaxique, sa flexibilité et sa communauté active, Python est devenu un choix populaire pour le développement d'applications, la création de scripts, et plus particulièrement, l'analyse de données. Il offre une variété de bibliothèques puissantes telles que NumPy, Pandas et Matplotlib, facilitant la manipulation, l'analyse et la visualisation de données.

### II.2.3 Bibliothèques utilisés

Dans cette figure, nous présentons les bibliothèques utilisées dans le cadre de notre projet

```
import pandas as pd
import numpy as np
from sklearn.neighbors import NearestNeighbors
from kneed import KneeLocator
from sklearn.cluster import DBSCAN
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter
```

FIGURE II.4 – Logo de Python

— **pandas (pd)** : pandas est une bibliothèque Python qui offre des structures de données flexibles et des outils de manipulation de données. Elle est largement utilisée pour l'analyse de données et la préparation des données.

— **numpy (np)** : NumPy est une bibliothèque fondamentale pour la manipulation de tableaux multidimensionnels et la réalisation de calculs mathématiques. Elle fournit des fonctions efficaces pour effectuer des opérations sur des tableaux.

— **sklearn.neighbors.NearestNeighbors** : NearestNeighbors est une classe dans la bibliothèque scikit-learn qui implémente la recherche des voisins les plus proches. Elle est utilisée pour trouver les voisins les plus proches d'un point donné dans un ensemble de données.

— **kneed (KneeLocator)** : KneeLocator est une fonctionnalité de la bibliothèque kneed qui aide à trouver le point de coude dans une courbe. C'est souvent utilisé pour déterminer le nombre optimal de clusters dans des méthodes de clustering.

— **sklearn.cluster.DBSCAN** : DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est une méthode de clustering dans scikit-learn qui identifie les clusters en fonction de la densité des points. Elle est particulièrement utile pour détecter des clusters de forme arbitraire.

— **seaborn (sns)** : seaborn est une bibliothèque de visualisation de données basée sur Matplotlib. Elle simplifie la création de graphiques informatifs et est souvent utilisée pour rendre les visualisations plus esthétiques.

— **matplotlib.pyplot (plt)** : Matplotlib est une bibliothèque de traçage de graphiques en 2D qui produit des graphiques de qualité. pyplot fournit une interface pour créer des tracés statiques, des tracés interactifs et des animations.

— **collections.Counter** : Counter est une sous-classe de dict dans le module collections. Il est utilisé pour compter le nombre d'occurrences des éléments dans une séquence, ce qui peut être utile pour l'analyse statistique des données.

## II.3 La présentation du projet

Pour le regroupement à l'aide de DBSCAN, nous utilisons un ensemble de données d'expression génique de cellules individuelles d'Arabidopsis thaliana, traité par un pipeline Cell Ranger de 10x genomics. Le jeu de données est soumis à une technique de réduction de dimension t-SNE préalable. À présent, nous utiliserons les vecteurs d'encastrement t-SNE pour identifier les groupes à l'aide de DBSCAN.

### II.3.1 Obtenir les données

```
import pandas as pd
# Charger le jeu de données
df = pd.read_csv("tsne_scores.csv")
print(df.shape)
✓ 0.0s
(4406, 2)
```

FIGURE II.5 – Lecture des données

Ce jeu de données compte 4406 lignes et deux caractéristiques. Il s'agit d'un jeu de données non étiqueté (sans information de cluster). Je vais identifier les informations de cluster sur ce jeu de données en utilisant DBSCAN.

### II.3.2 Calcul des Paramètres Requis pour le Regroupement avec DBSCAN

DBSCAN nécessite les paramètres $\varepsilon$ et minPts pour le regroupement. Le paramètre minPts est fixé à 4 pour un ensemble de données bidimensionnel. Dans le cas d'un ensemble de données multidimensionnel, minPts est défini à 2 fois le nombre de dimensions ; par exemple, minPts =

12 pour un ensemble de données avec 6 caractéristiques. Une expertise dans le domaine peut également être nécessaire pour ajuster ce paramètre.

Concernant la valeur optimale du paramètre $\varepsilon$, sa définition est délicate et dépend de la fonction de distance. Parfois, une connaissance spécialisée dans le domaine est essentielle pour déterminer un bon paramètre $\varepsilon$ . Idéalement, $\varepsilon$ devrait être aussi petit que possible.

Pour déterminer le paramètre optimal $\varepsilon$, le calcul des distances des k-nearest neighbors (distance moyenne de chaque point de données à ses k-voisins les plus proches) d'un ensemble de données sera effectué. La méthode des k-nearest neighbors sera utilisée, en exploitant la fonction `sklearn.neighbors.NearestNeighbors`.

La fonction NearestNeighbors requiert également le paramètre `n_neighbors` (nombre de voisins), qui peut être réglé sur la même valeur que minPts pour garantir la cohérence entre les paramètres.

```python
import numpy as np
from sklearn.neighbors import NearestNeighbors
# n_neighbors = 5 car la fonction kneighbors retourne la distance du point à lui-même (c'est-à-dire que la première colonne
nbrs = NearestNeighbors(n_neighbors=5).fit(df)

# Trouver les k-voisins d'un point
neigh_dist, neigh_ind = nbrs.kneighbors(df)

# Trier les distances des voisins (longueurs jusqu'aux points) par ordre croissant
# axis = 0 représente le tri le long du premier axe, c'est-à-dire le tri le long des lignes
sort_neigh_dist = np.sort(neigh_dist, axis=0)
```

✓ 0.0s Python

FIGURE II.6 – Parametres et regroupement

Maintenant, obtenez la colonne triée kth (distances avec les k-èmes voisins) et tracez le graphique des distances kNN.

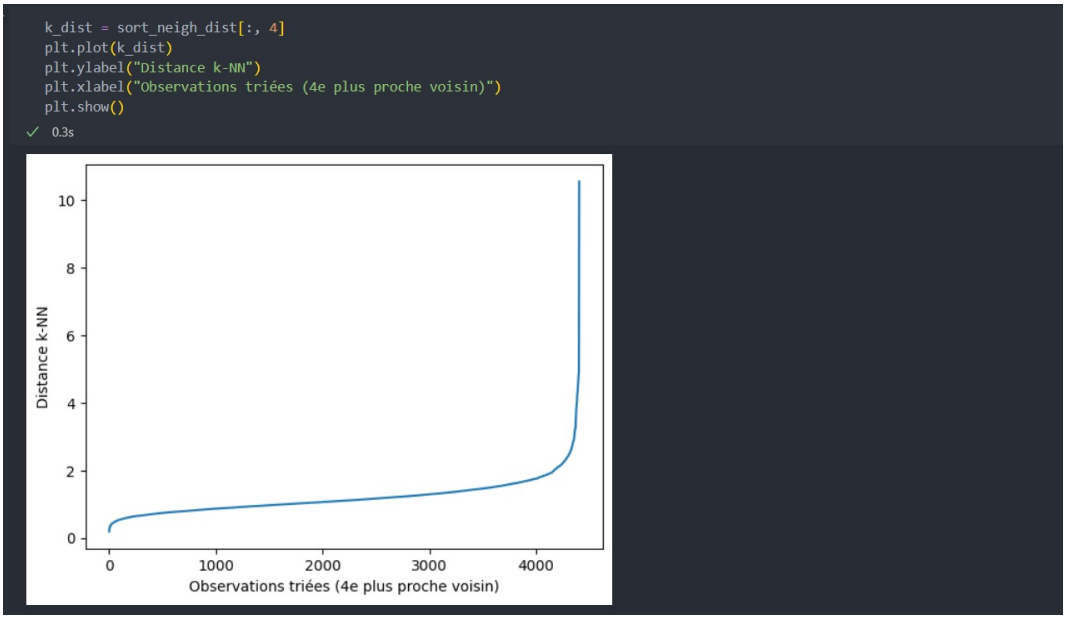


FIGURE II.7 – Diagramme des distances

Dans le graphique des distances k-NN, vous devriez rechercher le point du "knee" ou du "elbow" de la courbe pour trouver la valeur optimale de $\varepsilon$.

Identifier précisément le point du genou peut être difficile visuellement. Sur le graphique ci-dessous, le point du knee peut se situer n'importe où entre 2 et 5, c'est-à-dire que les points en dessous du point du knee appartiennent à un cluster, et les points au-dessus du point du knee sont du bruit ou des valeurs aberrantes (les points de bruit auront une distance kNN plus élevée). Vous devriez exécuter DBSCAN avec différentes valeurs de $\varepsilon$(entre 2 et 5) pour trouver le meilleur $\varepsilon$ qui donne le meilleur regroupement.

De plus, pour obtenir une estimation du point du genou, vous pouvez utiliser la fonction KneeLocator() du package kneed.

```
# Utiliser KneeLocator pour trouver le point de coude
kneedle = KneeLocator(x=range(1, len(neigh_dist)+1), y=k_dist, S=1.0, curve="concave", direction="increasing", online=True)
optimal_epsilon = kneedle.knee_y
✓ 0.3s

# Afficher le point de coude estimé
print(optimal_epsilon)
✓ 0.0s
4.5445133515748894
```

FIGURE II.8 – Calclule de $\varepsilon$

Tracé des distances

```
kneedle.plot_knee()
plt.show()
✓ 0.4s
```

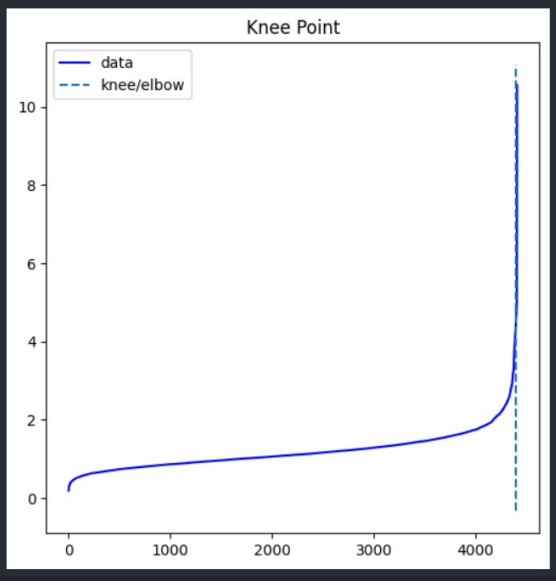


FIGURE II.9 – Projection du point $\varepsilon$

Nous utiliserons 4,54 comme valeur optimale de $\varepsilon$pour le clustering DBSCAN

### II.3.3 Effectuer le regroupement DBSCAN

Maintenant, nous avons calculé les paramètres $\varepsilon$ et minPts pour le regroupement DBSCAN. Nous allons transmettre ces paramètres à DBSCAN pour prédire les clusters en utilisant la classe sklearn.cluster.DBSCAN.

```
# Créer des clusters avec les paramètres eps=4.54 et min_samples=4
clusters = DBSCAN(eps=4.54, min_samples=4).fit(df)

# Obtenir les étiquettes des clusters
cluster_labels = clusters.labels_

# Résultat
print(cluster_labels)

# Vérifier les clusters uniques
unique_clusters = set(cluster_labels)
print(unique_clusters)
```

✓ 0.1s

```
[0 0 1 ... 1 1 1]
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -1}
```

FIGURE II.10 – Étiqueter les Clusters

### II.3.4 Visualisation du regroupement DBSCAN

Visualisez le regroupement sous forme de nuage de points (scatter plot) et colorez les clusters en utilisant les étiquettes de classe prédites.

```
p = sns.scatterplot(data = df, x = "t-SNE-1", y = "t-SNE-2", hue = clusters.labels_, legend = "full", palette = "deep")
sns.move_legend(p, "upper right", bbox_to_anchor = (1.17, 1.), title = 'Clusters')
plt.show()
```

✓ 0.7s

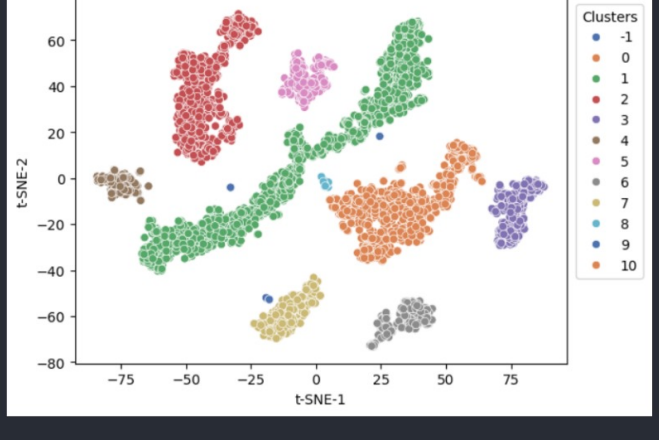


FIGURE II.11 – Diagramme des clusters

## II.4 Conclusion

En clôturant cette section d'implémentation, nous avons plongé dans le monde concret du Data Mining en appliquant l'algorithme DBSCAN à un ensemble de données d'expression génique. Les choix technologiques, les outils utilisés, et les langages de programmation ont été dévoilés, éclairant le chemin qui a conduit à la concrétisation de notre projet.

L'utilisation de Python, avec ses bibliothèques spécialisées telles que pandas, numpy, scikit-learn, et kneed, a permis une implémentation robuste et efficace. La visualisation des résultats à travers des graphiques a offert une compréhension visuelle du regroupement obtenu par DBSCAN.

Notre projet, axé sur l'analyse de l'expression génique, a servi d'exemple concret pour illustrer les étapes du processus, de l'obtention des données à l'étiquetage des clusters. En naviguant à travers ces étapes, nous avons établi un pont entre la théorie du Data Mining et sa mise en œuvre pratique.

Le diagramme des clusters a fourni une vue d'ensemble des résultats obtenus, mettant en lumière la capacité de DBSCAN à détecter des structures complexes au sein de l'ensemble de données. Cette réalisation s'inscrit dans une démarche plus large visant à exploiter les techniques de clustering pour extraire des informations significatives à partir de données brutes.

La section suivante poursuivra notre exploration en se concentrant sur le code Python, détaillant chaque étape de l'implémentation. En suivant ce cheminement, nous approfondirons notre compréhension technique de DBSCAN, consolidant ainsi les connaissances acquises dans cette section.

# Conclusion générale

Au terme de cette exploration du Data Mining, nous avons traversé les différentes facettes de cette discipline fascinante. Du contexte théorique aux applications concrètes, chaque chapitre a contribué à construire une compréhension approfondie du processus de fouille de données.

Le premier chapitre a jeté les bases, définissant le Data Mining et explorant ses objectifs, techniques, et domaines d'application. Nous avons abordé les deux grandes catégories de méthodes, descriptives et prédictives, soulignant l'importance de ces approches dans la transformation de données brutes en connaissances exploitables.

Le chapitre suivant a plongé dans les mécanismes du clustering, en mettant en évidence son rôle essentiel dans la découverte de structures intrinsèques au sein des données. L'introduction de DBSCAN comme algorithme de clustering basé sur la densité a ajouté une dimension pratique à notre compréhension, ouvrant la voie à une mise en œuvre concrète.

La mise en pratique de DBSCAN a constitué le cœur de notre troisième chapitre. De l'obtention des données d'expression génique à la détermination des paramètres optimaux pour le clustering, chaque étape a été détaillée. L'illustration de ces concepts à travers un projet concret a permis de consolider les connaissances acquises.

En conclusion, le Data Mining se révèle comme un outil puissant pour extraire des informations riches à partir de données complexes. Des techniques comme DBSCAN offrent des moyens sophistiqués de comprendre la structure sous-jacente des données, ouvrant ainsi des opportunités pour la prise de décision, la prédiction, et la révélation de connaissances enfouies.

Ce voyage à travers le Data Mining, de la théorie à la pratique, a démontré la pertinence et la puissance de cette discipline dans le contexte moderne de l'analyse de données. En continuant à explorer de nouvelles techniques, à affiner nos compétences, et à appliquer ces connaissances dans divers domaines, nous pourrons exploiter pleinement le potentiel du Data Mining pour façonner l'avenir de la compréhension des données.

# Références :

### II.4.1 Documentation Technique Générale


— **DBSCAN sur Wikipedia :** Consulté le 05/12/2023. `https://en.wikipedia.org/wiki/DBSCAN`

— **Documentation scikit-learn sur DBSCAN :** Consulté le 10/12/2023. `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN`

— **Article sur KDnuggets :** Consulté le 15/12/2023. `https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning`


### II.4.2 Documentation Personnalisée


— Mme. MOUNIR Ilham. (2023). "Outils d'aide à la décision : Fouille de Données" . Consulté le 27/12/2024.

— Mme. MOUNIR Ilham. (2023). "Clustering (regroupement ,segmentation)" . Consulté le 29/12/2024.