ENSA
École nationale des sciences appliquées
KOURIBGA

Master Big Data et Aide à la Décision

Clustering : DBSCAN Algorithm

Préparé par :

AIT AISSA Iman
BAHADDOU Youness
SAFINY Yazid

Sous la supervision de :

Prof OURDOU Amal

Table des matières

Table des figures

Introduction

Depuis son évolution, l'être humain avait employé plusieurs types des outils pour accomplir divers charges, comme les machines qui ont rendu la vie humaine facile.

Afin que les machines puissent imiter la pensée et les actions humaines , en mi-vingtième siècle le terme d'intelligence artificielle apparu qui met en oeuvre certain nombre de techniques qui sont largement utilisées dans de nombreux. Domaines tels que la régulation de processus industriels, le traitement d'images, le diagnostic, la médecine, la technologie spatiale et les systèmes de gestion de données informatiques.

Cette réflexion va donner naissance à l'apprentissage automatique (ou machine learning) consiste en effet à utiliser conjointement des quantités massives de data et d'algorithmes ce qui rend possible la résolution de plusieurs problèmes du monde réel. Parmi les exemples d'applications célèbres : Recommandation de produits, Détection des SPAM ,Reconnaissance vocale, Chat Bots. Apprentissage supervisé et apprentissage non supervisé sont deux catégories principales de machine learning : L'apprentissage supervisé lorsque nous enseignons ou formons la machine en utilisant des données bien étiquetées.
En revanche l'apprentissage non supervisé permet de : classer, étiqueter et regrouper les points de données dans les ensembles de données sans avoir aucune orientation externe dans l'exécution de cette tâche. Il existe quatre types de tâches non supervisées : clustering, analyse en composantes principales, détection d'anomalies et encodeurs automatiques. Cependant, chaque tâche d'apprentissage non supervisé est une variété d'algorithmes.

Ce travail examine l'apprentissage non supervisé en mettant le point sur l'algorithme du clustering DBSCAN et en offrant une évaluation approfondie le principes, les étapes, des mesures de performance, les avantages et inconvénients et l'efficacité de cet algorithme.

Chapitre 1

Initiation à l'apprentissage automatique et Clustering

1.1 Intelligence Artificielle

L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains en se basant sur des systèmes informatiques , des données avec des systèmes de gestion et des algorithmes d'IA avancés (code).

1.2 Machine Learning

1.2.1 Définition

L'apprentissage automatique est un système d'algorithmes informatiques qui peut apprendre à partir d'exemples grâce à l'auto-amélioration sans être explicitement codé par un programmeur. L'apprentissage automatique fait partie de l'intelligence artificielle qui combine des données avec des outils statistiques pour prédire un résultat qui peut être utilisé pour faire des informations exploitables.

1.2.2 Besoin de l'apprentissage automatique

L'apprentissage automatique est important car il permet aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sur des tâches spécifiques sans être explicitement programmés. Cette capacité à apprendre à partir des données et à s'adapter à de nouvelles situations rend l'apprentissage automatique particulièrement utile pour les tâches impliquant de grandes quantités de données, la prise de décision complexe et les environnements dynamiques. Voici quelques domaines spécifiques dans lesquels l'apprentissage automatique est utilisé :

— **Modélisation prédictive** : L'apprentissage automatique peut être utilisé pour créer des modèles prédictifs qui peuvent aider les entreprises à prendre de meilleures décisions.

— **Traitement du langage naturel** : L'apprentissage automatique est utilisé pour construire des systèmes capables de comprendre et d'interpréter le langage humain.

— **Vision par ordinateur** : L'apprentissage automatique est utilisé pour construire des systèmes capables de reconnaître et d'interpréter des images et des vidéos.

— **Détection des fraudes** : L'apprentissage automatique peut être utilisé pour détecter les comportements frauduleux dans les transactions financières, la publicité en ligne et d'autres domaines.

— **Systèmes de recommandation** : L'apprentissage automatique peut être utilisé pour créer des systèmes de recommandation qui suggèrent des produits, des services ou du contenu aux utilisateurs en fonction de leur comportement et de leurs préférences passées.

Types d'apprentissage automatique

— **ML supervisé**

— **ML non supervisé**

]

1.3 Unsupervised Learning

1.3.1 Définition

Unsupervised Learning (Apprentissage non supervisé) appelé aussi Unsupervised Machine Learning (Apprentissage Automatique non supervisé) est un type de Machine Learning qui utilise des algorithmes pour analyser, regrouper, et qui identifie des modèles dans les datasets (ensembles de données) contenant des individus, ou des data points qui ne sont ni classés ni étiquetés. Ces algorithmes classe et regroupe les individus lui-même sans aucune orientation externe ou sans intervention humaine dans l'exécution. La capacité des algorithmes de l'apprentissage non supervisé à découvrir les similitudes et les différences entre les individus (data points) sans avoir des catégories ou des classes attribué à ces individus est la solution pour plusieurs problèmes comme l'analyse exploratoire des données (Exploratory Data Analysis), et la segmentation des client (Client Segmentation), etc.

1.3.2 Types de l'apprentissage non supervisé

Les problèmes d'apprentissage non supervisé (Unsupervised Learning) sont divisés en deux catégories :

— **Clustering** : Le clustering est une technique d'apprentissage non supervisée, qui regroupe des points de données non étiquetés en fonction de leur similitude et de leurs différences.

— **Association Rules** : est une autre forme ou type d'apprentissage non supervisé, qui trouve des relations entre les individus (data points). Ces algorithmes trouvent les individus qui apparaissent ensemble dans une datasets, ils sont souvent utilisés pour l'analyse des achats (Market Basket Analysis), ce qui permet aux entreprises de comprendre la relation entre l'achat de différents produits. Par exemple, les clients qui achètent le produit A ont aussi tendance à acheter le produit B. Ils existent plusieurs algorithmes qui sont utilisés dans ce type comme l'algorithme Apriori, l'algorithme Eclat, l'algorithme FP-growth, etc.

1.3.3 Application de l'apprentissage non supervisé

On peut utiliser l'apprentissage non supervisé dans plusieurs cas ou applications, comme :

— **Products Segmentation**

— **Customer Segmentation**

— **Systèmes de recommandation**

— **Détection des similarités**

— **News Sections**

— **Détection des anomalies**

1.3.4 Avantages de l'apprentissage non supervisé

L'utilisation de l'apprentissage non supervisé à plusieurs avantages comme :

— Il peut voir ce que les être humaine ne peuvent pas visualiser

— Facilité d'obtention des données non étiquetées

— Nécessite moins de préparation manuelle des données que l'apprentissage supervisé (Supervised Learning)

1.3.5 Défis de l'apprentissage non supervisé

Il existe quelques défis qui peuvent survenir lors de l'exécution d'un algorithme de l'apprentissage automatique non supervisé, et parmi ces défis :

— Temps d'entrainement plus long dans le cas de Big Data (volume élevé de données)

— Risque de résultats inexactes

— Intervention et vérification humaine pour valider les résultats obtenus

1.4 Clustering

1.4.1 Qu'est-ce que le clustering ?

Il s'agit essentiellement d'un type de méthode d'apprentissage non supervisé. Généralement, il est utilisé comme un processus pour trouver une structure significative, des processus sous-jacents explicatifs, des caractéristiques génératives et des regroupements inhérents à un ensemble d'exemples.

Le regroupement consiste à diviser la population ou les points de données en un certain nombre de groupes de sorte que les points de données des mêmes groupes soient plus similaires aux autres points de données du même groupe et différents des points de données des autres groupes. Il s'agit essentiellement d'une collection d'objets sur la base de la similitude et de la dissemblance entre eux.

1.4.2 Pourquoi le clustering ?

Le regroupement est très important car il détermine le regroupement intrinsèque parmi les données non étiquetées présentes. Il n'y a pas de critères pour un bon regroupement. Cela dépend de l'utilisateur, quels sont les critères qu'il peut utiliser pour satisfaire son besoin. Par exemple, nous pourrions être intéressés à trouver des représentants pour des groupes homogènes (réduction de données), à trouver des « amas naturels » et à décrire leurs propriétés inconnues (types de données « naturels »), à trouver des regroupements utiles et appropriés (classes de données « utiles ») ou à trouver des objets de données inhabituels (détection de valeurs aberrantes). Cet algorithme doit faire des hypothèses qui constituent la similitude des points et chaque hypothèse fait des clusters différents et également valides.

1.4.3 Méthodes de clustering

Le clustering a une myriade d'utilisations dans une variété d'industries. Quelques points communs Les applications du clustering sont les suivantes :

— Méthodes basées sur la densité : Ces méthodes considèrent les amas comme la région dense présentant certaines similitudes et différences par rapport à la région dense inférieure de l'espace. Ces méthodes ont une bonne précision et la capacité de fusionner deux clusters. Exemple DBSCAN (Density-Based Spatial Clustering of Applications with Noise)), OPTICS (Ordering Points to Identify Clustering Structure), etc.

— Méthodes hiérarchiques : Les clusters formés dans cette méthode forment une structure arborescente basée sur la hiérarchie. De nouveaux clusters sont formés à l'aide du cluster précédemment formé. Il est divisé en deux catégories :
  — Approche agglomérative (approche ascendante)
  — Clivage (approche descendante)

— Méthodes de partitionnement : Ces méthodes partitionnent les objets en k clusters et chaque partition forme un cluster. Cette méthode est utilisée pour optimiser une fonction de similarité de critère objectif par exemple lorsque la distance est un paramètre majeur exemple K-means, CLARANS (Clustering Large Applications based on Randomized Search), etc.

— Méthodes basées sur la grille : Dans cette méthode, l'espace de données est formulé en un nombre fini de cellules qui forment une structure en forme de grille. Toutes les opérations de clustering effectuées sur ces grilles sont rapides et indépendantes du nombre d'objets de données par exemple STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest), etc.

1.4.4 Applications du clustering dans différents domaines

— Marketing : Il peut être utilisé pour caractériser et découvrir des segments de clientèle à des fins de marketing.

— Biologie : Il peut être utilisé pour la classification parmi différentes espèces de plantes et d'animaux.

— Bibliothèques : Il est utilisé pour regrouper différents livres sur la base de sujets et d'informations.

— Assurance : Il est utilisé pour reconnaître les clients, leurs politiques et identifier les fraudes.

— Urbanisme : Il est utilisé pour faire des groupes de maisons et pour étudier leurs valeurs en fonction de leur emplacement géographique et d'autres facteurs présents.

— Études sismiques : En apprenant les zones touchées par le tremblement de terre, nous pouvons déterminer les zones dangereuses.

Chapitre 2

DBSCAN

Il existe plusieurs algorithmes qui peuvent être appliquer sur les problèmes du Clustering, et on peut diviser ces algorithmes en trois sous catégories :

— Partition-based clustering (Clustering basé sur des partitions) : K-means

— Hierarchical clustering (Clustering hiérarchique) : Agglomerative

— Density-based clustering (Clustering basé sur la densité) : DBSCAN

2.1 Density-based Clustering

Le clustering basé sur la densité fait référence à des méthodes d'apprentissage non supervisées qui identifient des clusters distinctifs dans les données, sur la base de l'idée qu'un cluster dans une dataset est une région contiguë à haute densité de points, séparée des autres clusters par des régions clairsemées. Les points de données dans les régions clairsemées de séparation sont généralement considérés comme du bruit/des valeurs aberrantes (outliers). Les techniques de partition-based et hierarchical clustering sont très efficaces avec des clusters de forme normale. Cependant, lorsqu'il s'agit de clusters de forme arbitraire ou de détection de valeurs aberrantes, les techniques basées sur la densité sont plus efficaces. Les algorithmes de clustering basés sur la densité sont très efficaces pour trouver des régions à haute densité et des valeurs aberrantes. Il est très important de détecter les valeurs aberrantes pour certaines tâches, par ex. Détection des anomalies. La figure représente une dataset de clustering basé sur la densité, avec 3 Clusters, et des Outlier

2.2 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est une méthode d'apprentissage automatique non supervisée populaire pour séparer les groupes (Clusters) de haute densité des groupes de faible densité. DBSCAN est capable de trouver des clusters de forme arbitraire et des clusters avec des valeurs aberrantes.

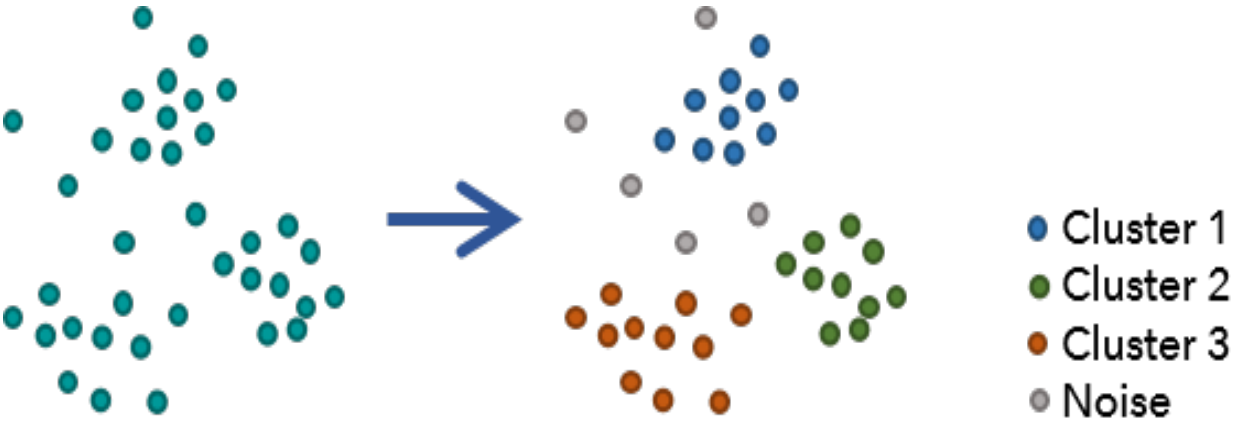


FIGURE 2.1 – Exemple d'une dataset de clustering basé sur la densité

2.3 Principe du DBSCAN

L'idée principale du DBSCAN est qu'un individu (data point) appartient à un cluster s'il est proche de plusieurs individus (data points) de ce cluster. Il existe deux principaux paramètres de DBSCAN :

- **Eps** : La distance qui spécifie les voisinages. Deux points sont considérés comme voisins si la distance qui les sépare est inférieure ou égale à eps.
- **minPts** : nombre minimal de points de données pour définir un core points.

En se basant sur ces deux parametres les individus (data points) sont classé comme Core points, Border points, ou outliers.

- **Core point** : un individu (data point) est un Core point s'il y a au moins un nombre minPts de points dans sa zone avec un rayon eps.
- **Border point** : un individu (data point) est un Border point s'il est accessible à partir d'un Core point et qu'il y a moins de minPts de points dans sa zone.
- **Outlier** : un individu (data point) est un Outlier s'il ne s'agit pas d'un Core point ni d'un Border point et il n'est pas accessible à partir d'aucun Core Point.

La figure représente la différence entre Core points, Border points et Outliers dans une dataset.
Les Core points sont les points représenté en rouge, avec 4 est le minimum nombre de voisinage.
Les Border points sont les points B et C en jaune, puisqu'il n'a pas 4 voisin dans un rayon (eps) qui est spécifier, mais il est accessible à travers d'autre Core points.
L' Outlier est le point N en bleu, puisqu'ila moins de 4 voisins dans le rayon, et il n'est pas accessible à partir aucun autre point.

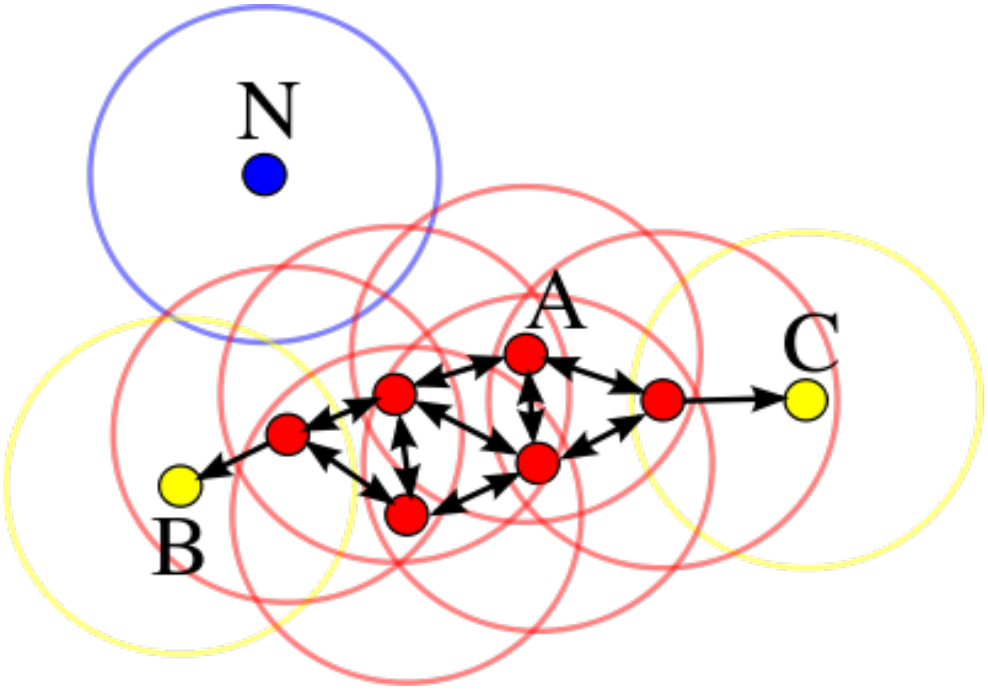


FIGURE 2.2 – Différence entre Core, Border points et Outlier

2.3.1 Étapes de l'algorithme du DBSCAN

— minPts et eps doivent être choisit.

— Un point de départ est sélectionné au hasard et sa zone de voisinage est déterminée à l'aide du rayon eps. S'il y a au moins un nombre minPts de points dans le voisinage, le point est marqué comme Core point et une formation de cluster commence. Sinon, le point est marqué comme Outlier. Une fois qu'une formation de cluster commence, tous les points dans le voisinage du point initial deviennent une partie du cluster. Si ces nouveaux points sont également des Core points, les points qui se trouvent dans leur voisinage sont également ajoutés à ce cluster.

— On choisit au hasard un autre point parmi les points qui n'ont pas été visités lors des étapes précédentes puis on répéte les deux étapes précédentes jusqu'à tous les points sont visités.

En appliquant ces étapes, l'algorithme DBSCAN est capable de trouver des régions à haute densité et de les séparer des régions à faible densité. Un cluster comprend des Core points voisins (c'est-à-dire accessibles les uns des autres) et tous les Border points de ces Core points. La condition requise pour former un cluster est d'avoir au moins un Core point.

2.3.2 Avantages et Inconvénients du DBSCAN

Avantages

— Ne nécessite pas de spécifier le nombre de clusters au préalable.

— Fonctionne bien avec des clusters de formes arbitraires.

— DBSCAN est robuste aux valeurs aberrantes et capable de détecter les valeurs aberrantes.

Inconvénients

— Dans certains cas, déterminer une distance appropriée de voisinage (eps) n'est pas facile et nécessite une connaissance du domaine.

2.4 Méthodes d'évaluation

Lorsque vous parlez de validation d'un modèle d'apprentissage automatique, il est important de savoir que les techniques de validation utilisées aident non seulement à mesurer les performances, mais vous aident également à comprendre votre modèle à un niveau plus profond. C'est la raison pour laquelle un temps important est consacré au processus de validation des résultats lors de la construction d'un modèle d'apprentissage automatique.

La validation des résultats est une étape cruciale car elle garantit que notre modèle donne de bons résultats non seulement sur les données d'entraînement, mais surtout sur les données en direct ou de test. Dans le cas de l'apprentissage supervisé, cela se fait principalement en mesurant les mesures de performance telles que l'exactitude, la précision, le rappel, l'ASC, etc. sur l'ensemble d'entraînement et les ensembles de retrait. Ces mesures de performance aident à décider de la viabilité du modèle.

Cependant, si ce n'est pas le cas, nous pouvons ajuster les hyperparamètres et répéter le même processus jusqu'à ce que nous obtenions les performances souhaitées. Cependant, dans le cas d'un apprentissage non supervisé, le processus n'est pas très simple car nous n'avons pas la vérité sur le terrain. En l'absence d'étiquettes, il est très difficile d'identifier les KPI qui peuvent être utilisés pour valider les résultats.

Il existe deux classes de techniques statistiques pour valider les résultats de l'apprentissage par grappes. Il s'agit de :

— Validation interne.

— Validation externe.

La majeure partie de la littérature relative à la validation interne pour l'apprentissage par grappes s'articule autour des deux types de mesures suivants :

— Cohésion au sein de chaque cluster.

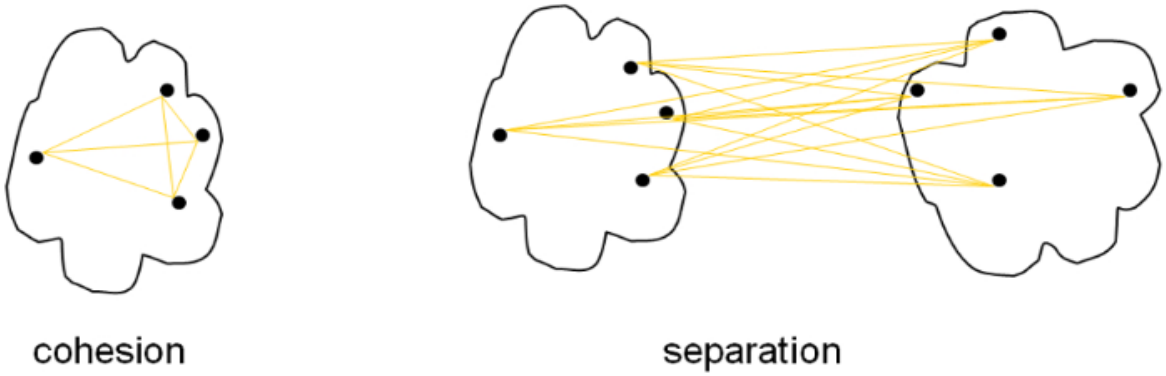
— Séparation entre les différents groupes.

La validation métier/utilisateur, comme son nom l'indique, nécessite des entrées externes aux données. L'idée est de générer des grappes sur la base des connaissances d'experts en la matière, puis d'évaluer la similitude entre les deux ensembles de grappes, c'est-à-dire les grappes générées par le blanchiment d'argent et les grappes générées à la suite d'entrées humaines. Cependant, dans la plupart des cas, ces connaissances ne sont pas facilement disponibles. De plus, cette approche n'est pas très évolutive. Par conséquent, dans la pratique, la validation externe est généralement ignorée.

Proposons maintenant la validation par double échantillon comme méthodologie pour valider les résultats de l'apprentissage non supervisé en plus de la validation interne, qui est très similaire à la validation externe, mais sans avoir besoin d'apports humains. Dans les sections suivantes, nous expliquons brièvement différentes mesures pour effectuer des validations internes et externes. Il sera suivi d'une explication de la façon d'effectuer une validation à double échantillon en cas de clustering non supervisé et de ses avantages.

Validation interne

La plupart des méthodes de validation interne combinent cohésion et séparation pour estimer le score de validation.



L'approche consiste à calculer le score de validation de chaque cluster, puis à les combiner de manière pondérée pour arriver au score final pour l'ensemble des clusters. Soit S un ensemble de clusters C1 , C2 , C3 ,....., Cn , alors la validité de S sera calculée comme suit :

$$Validity(S) = \sum_{k=1}^{n} Wk * Validity(Ck) \tag{2.1}$$

La cohésion d'un cluster peut être calculée en additionnant la similitude entre chaque paire d'enregistrements contenus dans ce cluster.

$$Cohesion(Ck) = \sum_{x \in Ck; y \in Ck} similarity(x, y) \tag{2.2}$$

La séparation entre deux groupes peut être calculée en additionnant la distance entre chaque paire d'enregistrements appartenant aux deux groupes et les deux enregistrements proviennent de groupes différents.

$$Separation(Cj, Ck) = \sum_{x \in Cj; y \in Ck} distance(x, y) \tag{2.3}$$

Un ensemble de grappes ayant une cohésion élevée au sein des grappes et une séparation élevée entre les grappes est considérée comme bonne.

En pratique, au lieu de traiter de deux paramètres, plusieurs mesures sont disponibles qui combinent les deux mesures ci-dessus en une seule mesure. Voici quelques exemples de telles mesures :

— **Coefficient de silhouette :**

Le score de silhouette et le diagramme de silhouette sont utilisés pour mesurer la distance de séparation entre les grappes. Il affiche une mesure de la proximité de chaque point d'un cluster par rapport aux points des clusters voisins. Cette mesure a une plage de [-1, 1] et est un excellent outil pour inspecter visuellement les similitudes au sein des clusters et les différences entre les clusters.

Le score de silhouette est calculé en utilisant la distance moyenne intra-grappe () et la distance moyenne la plus proche de la grappe () pour chaque échantillon. Le coefficient de silhouette d'un échantillon est :

$$i * n * (n - i)/max(i, n) \tag{2.4}$$

n est la distance entre chaque échantillon et le groupe le plus proche dont l'échantillon ne fait pas partie, tandis que **i** est la distance moyenne au sein de chaque groupe.

Les diagrammes de silhouette typiques représentent l'étiquette de cluster sur l'axe y, tandis que le score de silhouette réel sur l'axe des x. La taille/épaisseur des silhouettes est également proportionnelle au nombre d'échantillons à l'intérieur de cette grappe.

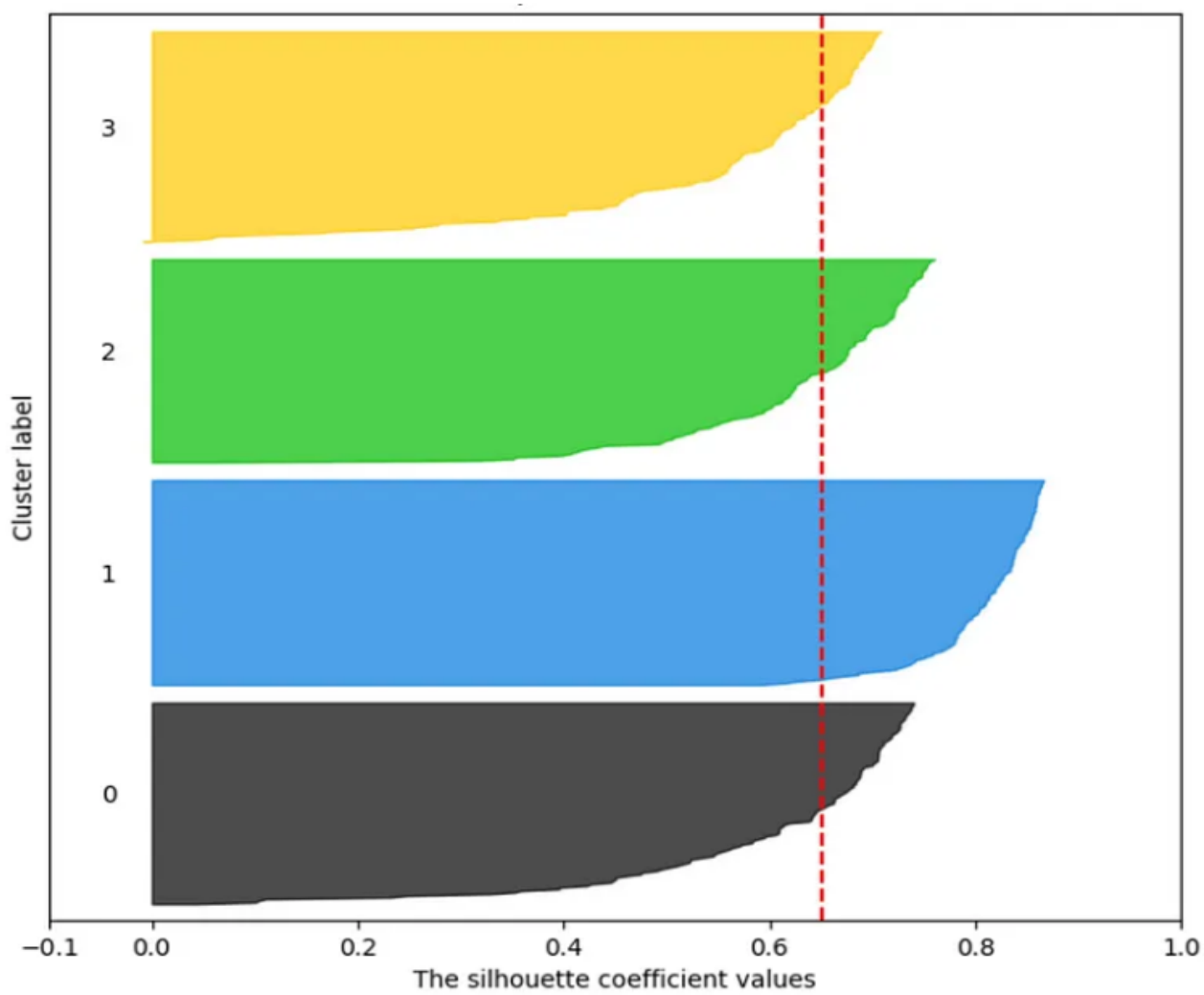


FIGURE 2.3 – Un exemple de tracé de silhouette

Plus les coefficients de silhouette sont élevés (plus ils se rapprochent de +1), plus les échantillons de la grappe sont éloignés des échantillons des grappes voisines. Une valeur de 0 indique que l'échantillon se trouve sur ou très près de la limite de décision entre deux grappes voisines. Les valeurs négatives, au contraire, indiquent que ces échantillons ont peut-être été affectés au mauvais cluster. En faisant la moyenne des coefficients de silhouette, nous pouvons obtenir un score de silhouette global qui peut être utilisé pour décrire la performance de l'ensemble de la population avec une seule valeur.

La meilleure valeur est 1 et la pire valeur est -1. Les valeurs proches de 0 indiquent clusters qui se chevauchent. Les valeurs négatives indiquent généralement qu'un échantillon a a été affecté au mauvais cluster, car un cluster différent est plus similaire.

Pour calculer le score de silhouette en Python, vous pouvez simplement utiliser Sklearn et faire :

```
sklearn.metrics.silhouette_score(X, labels, *, metric='euclidean',
sample_size=None, random_state=None, **kwds)
```

FIGURE 2.4 – Score de silhouette en Python

La fonction prend comme entrée :

— X : tableau de distances par paires entre les échantillons, ou tableau d'entités, si le paramètre « precalculé » est défini sur False.
— étiquettes : ensemble d'étiquettes représentant l'étiquette à laquelle chaque échantillon est affecté.

— **Coefficient de Calisnki-Harabasz :**

L'indice de Calinski-Harabasz est également connu sous le nom de critère du ratio de variance. Le score est défini comme le rapport entre la dispersion intra-cluster et la dispersion entre clusters. L'indice C-H est un excellent moyen d'évaluer les performances d'un algorithme de clustering car il ne nécessite pas d'informations sur les étiquettes de réalité terrain.

Plus l'indice est élevé, meilleure est la performance.

La formule est la suivante :

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

où tr(Bk) est la trace de la matrice de dispersion entre groupes et tr(Wk) est la trace de la matrice de dispersion intra-cluster définie par :

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

Un score Calinski-Harabasz plus élevé correspond à un modèle avec des clusters mieux définis.

Pour le calculer avec Python :

```
sklearn.metrics.calinski_harabasz_score(X, labels)
```

— Indice de Dunn.

— Score Xie-Beni.

— Indice Hartigan.

2.4.1 Validation externe

Ce type de validation des résultats peut être effectué si de véritables étiquettes de cluster sont disponibles. Les étiquettes générées par les PME peuvent également être utilisées pour créer de véritables étiquettes. Dans cette approche, nous aurons un ensemble de clusters S= C1, C2, C3,....., Cn qui ont été générés à la suite d'un algorithme de clustering. Nous aurons un autre ensemble de clusters P = D1, D2, D3, ..., Dm qui représentent les vraies étiquettes de cluster sur les mêmes données. L'idée est de mesurer la similitude statistique entre les deux ensembles. Un jeu de clusters est considéré comme bon s'il est très similaire au véritable jeu de clusters.

Afin de mesurer la similitude entre S et P, nous étiquetons chaque paire d'enregistrements de données comme positive si les paires appartiennent au même groupe dans P sinon négatif. Un exercice similaire est également effectué pour S. Nous calculons ensuite une matrice de confusion entre les étiquettes de paires de S et P qui peuvent être utilisées pour mesurer la similitude.

TP : Nombre de paires d'enregistrements qui se trouvent dans le même cluster, pour S et P.

FP : Nombre de paires d'enregistrements qui se trouvent dans le même cluster en S mais pas en P.

FN : Nombre de paires d'enregistrements qui se trouvent dans le même cluster en P mais pas en S.

TN : Nombre de paires d'enregistrements qui ne sont pas dans le même groupe S et P.

Sur les 4 indicateurs ci-dessus, nous pouvons calculer différentes métriques pour obtenir une estimation de la similitude entre S (étiquettes de cluster générées par une méthode non supervisée) et P (vraies étiquettes de cluster). Voici quelques exemples de mesures qui pourraient être utilisées :

— La précision : mesure le rapport entre les vrais positifs et le total des positifs prédits.

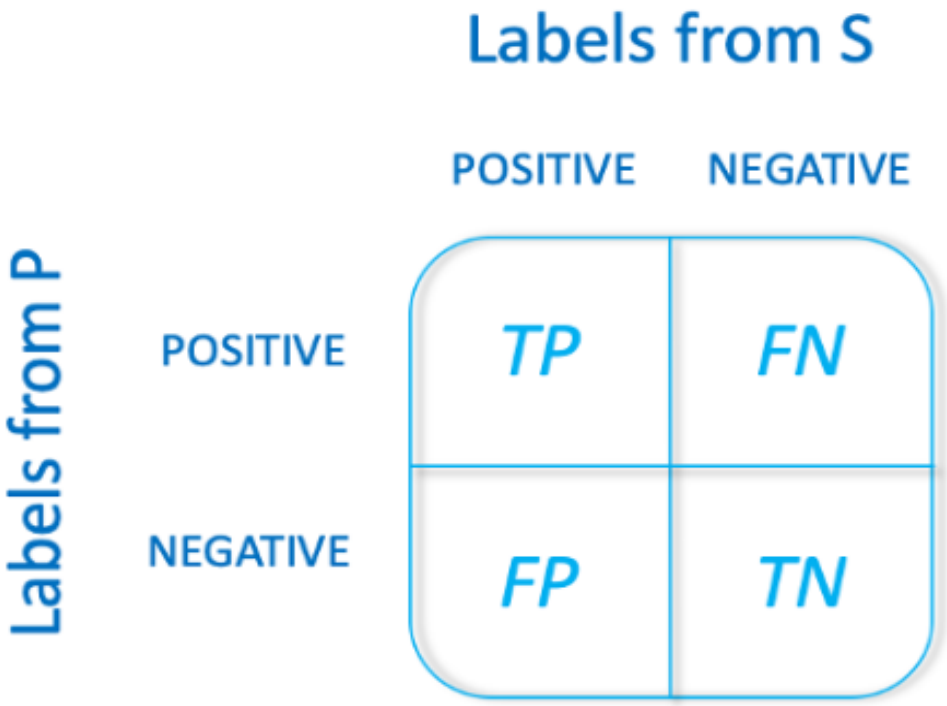
$$Pr = \frac{TP}{TP + FP} \tag{2.5}$$

— Le rappel : mesure le ratio de positifs capturés sur le total des vrais positifs.

$$R = \frac{TP}{TP + FN} \tag{2.6}$$

— F1-measure : combine précision et rappel en une seule métrique.

$$F1 = 2 * \frac{Pr * R}{Pr + R} \tag{2.7}$$



— Similitude Jaccard.

— Information mutuelle.

— Indice Fowlkes-Mallows.

Chapitre 3

Implémentation

Objectif

Regroupement des clients idéaux ayant des besoins, les comportements et les préoccupations similaires en utilisant l'algorithme de clustering DBSCAN afin de pouvoir réponde aux leurs exigences.

3.1 les bibliothèques de python

Matplotlib : capable de produire et de dessiner des graphes de qualité pour visualiser les données.

Pandas : offre notamment la possibilité de lire des données en provenance de nombreuses sources.

Seaborn : est une librairie qui vient s'ajouter à Matplotlib, remplace certains réglages par défaut et fonctions, et lui ajoute de nouvelles fonctionnalités.

Sickitlearn : La principale bibliothèque d'outils dédiée à l'apprentissage automatique et la science des données. elle est utile pour les algorithmes de clustering.

3.2 Mise en oeuvre avec python

3.2.1 La base de données

L'ensemble de données "customer personality analysis "est fourni par le Dr Omar Romero-Hernandez téléchargée d'aprés kaggle.
Ces données comprennent les caractéristiques différentes des 2240 exemples de clients décrivent par 29 attribues (année de naissance, niveau d'éducation, état civil, revenu annuel nombre d'enfants du client, Montant dépensé en poisson, sucreries au cours des 2 dernières années, Nombre d'achats effectués avec une remise, nombre d'achats effectués directement en magasin, nombre de visites sur le site Web etc)

Les attribues de la base de données :

```
'ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
'AcceptedCmp2', 'Complain', 'Z_CostContact', 'Z_Revenue', 'Response'],
```

3.2.2 Pré-traitement des données

1- supprimer les valeurs dupliquées par la fonction : drop_duplicates()

2- Fusionner les cinq attributs de Promotion d'un client :
'AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3' 'AcceptedCmp4' et 'AcceptedCmp5'
en un seul attribut et les supprimer pour réduire le nombre de colonnes.

3- Calculer l'âge des clients à partir de la colonne 'Year_Birth' et la durée d'inscription du client par la colonne : 'Dt_Customer' **de date d'inscription en utilisant la fonction pandas** datetime **et les remplacer dans la base de données**

4- Visualiser toutes les colonnes de l'ensemble de données en utilisant les bibliothèque Seaborn et Matplotlib pour une exploration rapide et efficace des relations entre toutes les variables

5- Convertir les deux attributs : 'Marital_Status' et 'Education' **qui comprennent des variables catégorielles en variables numériques binaires en utilisant OneHotEncoder une fonctionnalité de la bibliothèque scikitlearn.**

6- Normalisation dse données par la fonction 'StandardScaler' **du module** 'sklearn.preprocessing' **du package** 'Sickit_learn'

7- Les données d'entrée sont les features de la base de donnnées après les modification précédents et les réduire à deux dimensions par la fonction 'PCA' **du module** 'sklearn.decomposition' afin de réduire la complexité du modèle.

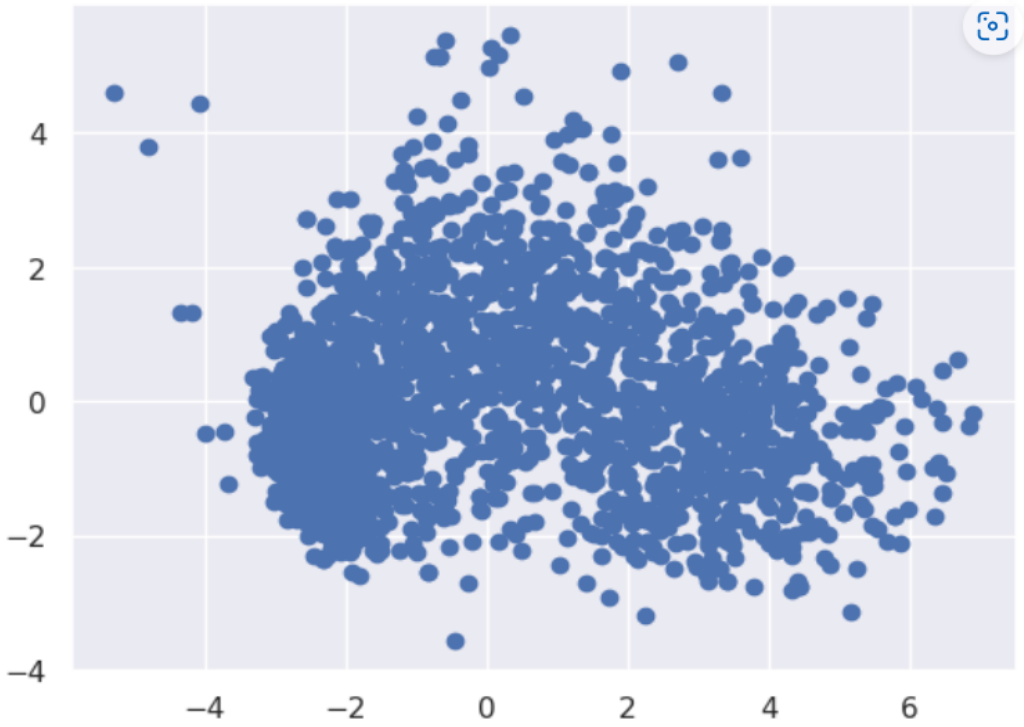


FIGURE 3.1 – Visualisation des donées après le prétraitement

3.2.3 Étude du corrélation entre des attributs

La corrélation est un terme statistique décrivant le degré auquel deux attributs évoluent en coordination l'une avec l'autre et afficher la matrice de corrélation à l'aide de la foncionnalité 'heatmap' de seaborn.

3.2.4 Application de l'algorithme

Nous construisons le modèle à laide de scikit-learn. Nous avons utilisé l'algorithme de clustering DBSCAN en ajustant les deux paramètres de cet l'algorithme : le rayon de voisinage 'eps' et le nombre minimum de points requis pour former un cluster 'minPts'.

notre modèle avec eps=1.298 et min_samples=256

3.3 Validation des modèles et les prédictions

3.3.1 Validation interne

Le silhouette score égale 0.445 on peut l'améliorer pour qu'il soit proche de 1 par modification des paramètres.

3.3.2 Résultat

Nous avons trouvé comme résultat de clustering par algorithme DBSCAN des clients deux clusters :
Cluster 0 : 1463
Cluster 1 : 518
avec une identififcation 'n_noise=259' des points isolés ou de faible densité comme des points de bruit dans le modèle en noir dans la figure ci-dessous.

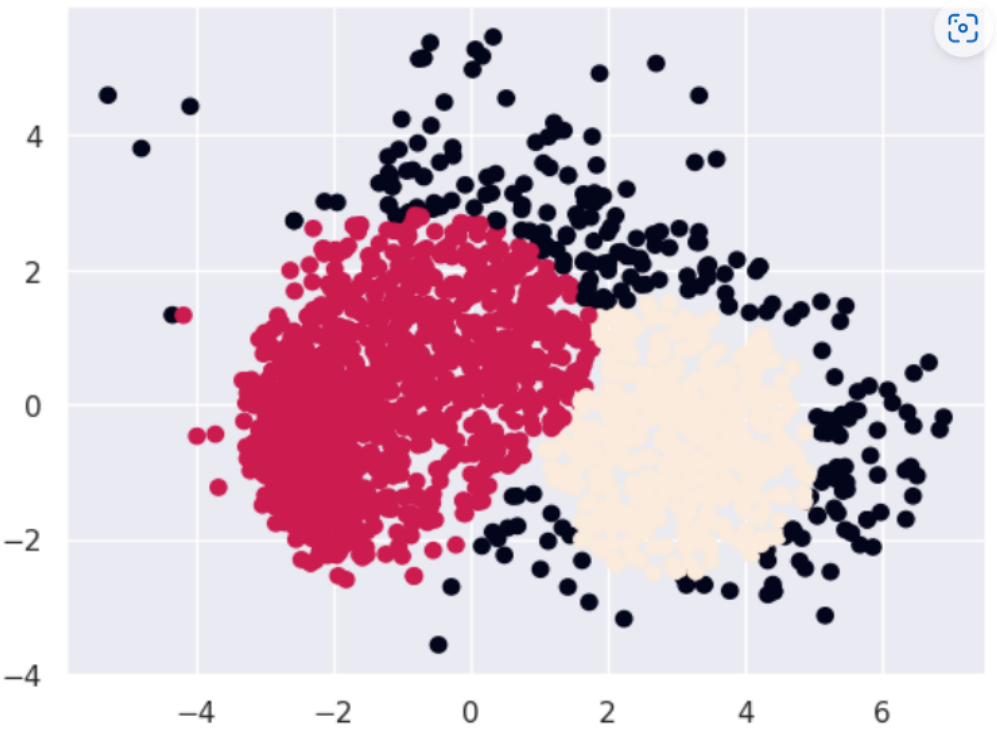


FIGURE 3.2 – Shéma des clusters

Conclusion

En guise de conclusion, notre projet avait pour ambition l'application d'algorithme DBSCAN à l'ensemble de données "Customer Personality Analysis". Nous avons commencé par prétraiter les données en effectuant une analyse exploratoire pour mieux comprendre la distribution des variables.

Ensuite, nous avons appliqué l'algorithme DBSCAN à l'ensemble de données pour détecter les segments de clients similaires en termes de personnalité et comportement d'achat. Nous avons utlisé les paramètres epsilon et minPts pour déterminer les clusters.

Notre analyse des résultats a montré que l'algorithme DBSCAN a identifié deux segments de clients différents avec des profils d'achat distincts. Ces segments étaient caractérisés par des niveaux caractéristiques similaires entre les clients. Nous avons également observé que les segments étaient stables par rapport à différents choix de paramètres.

Finalement, nous pouvons conclure que l'algorithme DBSCAN est un outil utile pour l'analyse de segments de clients dans des ensembles de données de taille modérée.

Bibliographie

[1] https ://www.guavus.com/technical-blog/unsupervised-machine-learning-validation-techniques/

[2] https ://towardsdatascience.com/performance-metrics-in-machine-learning-part-3-clustering-d69550662dc6

[3] https ://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

[4] https ://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/