|  |  |
| --- | --- |
| **C:\Users\adilma\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3CC4D457.tmp** | **C:\Users\adilma\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\39D09393.tmp** |

|  |  |
| --- | --- |
| 5 étapes pour se préparer à enseigner avec et à propos de l'IA - École  branchée  **Unsupervised ML Clustering K-Means**  MASTER IT 2024 2025 | Réalisé par  EL MAJDOUBI Adil  OUAHMANE Abdallah    Pr .MAHMOUDI Abdelhak |



**Tables Des Matières**

[1. Introduction (1) 2](#_Toc195006910)

[2. Caractéristiques (2) 2](#_Toc195006911)

[3. Les avantages (3) 2](#_Toc195006912)

[4. Inconvénients (3) 2](#_Toc195006913)

[5. Clustering (2) 3](#_Toc195006914)

[6. K-Means (4) 3](#_Toc195006915)

[7. Implémentation du clustering K-Means en Python (5) 7](#_Toc195006916)

[8. Conclusion (3) 8](#_Toc195006917)

[Bibliographie 8](#_Toc195006918)

[Liste des tableaux 9](#_Toc195006919)

[Table des figures 9](#_Toc195006920)

# Introduction (1)

Unsupervised Leraning est un type d'apprentissage automatique où le modèle reçoit des données sans étiquettes ni supervision explicite. Contrairement à l'apprentissage supervisé, qui utilise des données étiquetées pour apprendre des modèles, l'apprentissage non supervisé explore les données pour en extraire des structures cachées et des motifs sous-jacents.

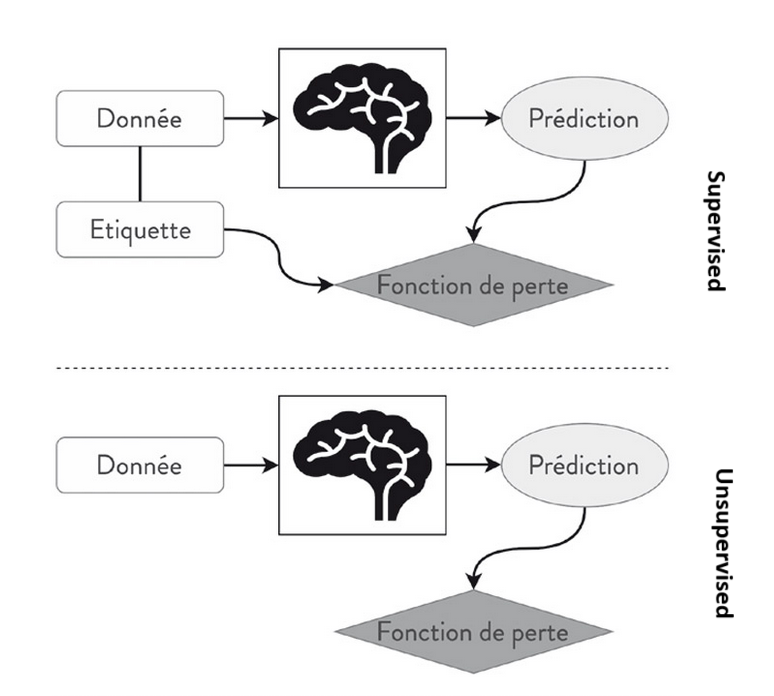


Figure : Apprentissage supervisé et non-supervisé

# Caractéristiques (2)

* Les données d'entraînement ne sont pas étiquetées (the training data is unlabeled).
* Le système essaie d'apprendre sans enseignant (the system tries to learn without a teacher).
* L’algorithme doit trouver une structure et des modèles dans les données (the algorithm must find structure and patterns in the data).
* Son propre, sans aucune indication sur ce qu'il faut rechercher (its own, without any guidance on what to look for).

# Les avantages (3)

* Pas besoin de données étiquetées : Contrairement à l’apprentissage supervisé, il ne nécessite pas d’annotations humaines, ce qui réduit le coût et le temps de préparation des données.
* Découverte de structures cachées : Il permet d’identifier des modèles et des structures cachées dans les données sans connaissances préalables.
* Adaptabilité : Utile dans des contextes où les règles sont inconnues ou en constante évolution.

# Inconvénients (3)

* Résultats moins interprétables : Les modèles peuvent produire des groupes ou des catégories difficiles à expliquer ou à valider.
* Manque de précision : Moins précis que l’apprentissage supervisé, car il n’y a pas de feedback direct pour corriger les erreurs.
* Difficulté d’évaluation : Il n’existe pas de métriques standards simples pour évaluer la qualité des résultats, contrairement aux modèles supervisés.
* Sensibilité aux paramètres : Le choix du nombre de clusters peut avoir un impact majeur sur les résultats.
* Overfitting : Les modèles peuvent capter du bruit inutile et créer des classifications inutiles.

# Clustering (2)

Dans ce system Unsupervised Learning, il y’a plusieurs modèles existent et peuvent être regroupés en cinq grandes catégories :

* Clustering.
* Dimensionality reduction.
* Anomaly detection.
* Novelty detection.
* Association Rule learning.

Clustering (classification non supervisée) est une technique clé de l'apprentissage non supervisé. Il consiste à regrouper des points de données en différentes classes ou clusters selon leurs similarités. L'objectif est que les données d'un même cluster soient plus semblables entre elles qu'avec celles des autres clusters.



Figure : Clustering (classification non supervisée)

* **Algorithmes de clustering les plus utilisés :**
* K-Means.
* Mean-Shift.
* DBSCAN.

# K-Means (4)

Cet algorithme, fondé sur le calcul des centres de gravité (moyennes), partitionne les données en K groupes.

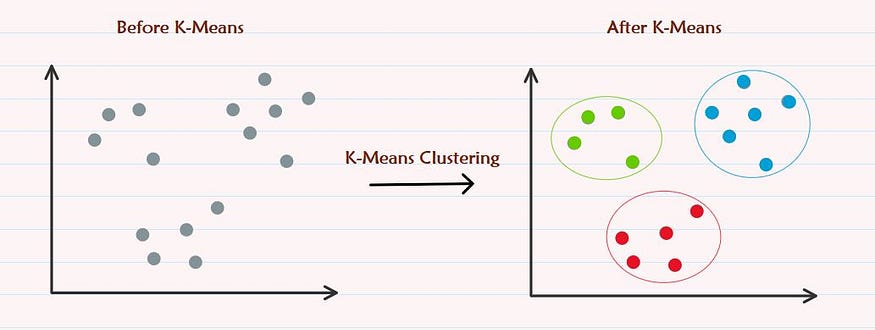


Figure : K-Means Algorithm

## Étapes du modèle K-Means



Figure : Points de données

* ***Étape 1 : Déterminer le nombre de clusters (K=?)***

La méthode du coude (Elbow method) est l'une des méthodes les plus courantes pour déterminer le nombre optimal de clusters. Elle utilise le concept de valeur WCSS (eng : Within Cluster Sum of Squares |fr : somme des carrés intra-cluster), qui définit les variations totales au sein d'un cluster. La formule de calcul de la valeur WCSS (pour 3 clusters) est donnée ci-dessous :

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2

∑Pi in Cluster1 distance(Pi C1)2: Il s'agit de la somme du carré des distances entre chaque point de données et son centroïde au sein d'un cluster1 et de la même chose pour les deux autres termes.

Pour mesurer la distance entre les points de données et le centroïde, nous pouvons utiliser n'importe quelle méthode, comme la distance euclidienne ou la distance de Manhattan.

Pour trouver la valeur optimale des clusters, la méthode du coude suit les étapes suivantes :

* Il exécute le clustering K-means sur un ensemble de données donné pour différentes valeurs K (de 1 à 10).
* Pour chaque valeur de K, calcule la valeur WCSS.
* Trace une courbe entre les valeurs WCSS calculées et le nombre de clusters K.
* Le point de courbure aigu ou un point du graphique ressemble à un bras ; ce point est alors considéré comme la meilleure valeur de K.

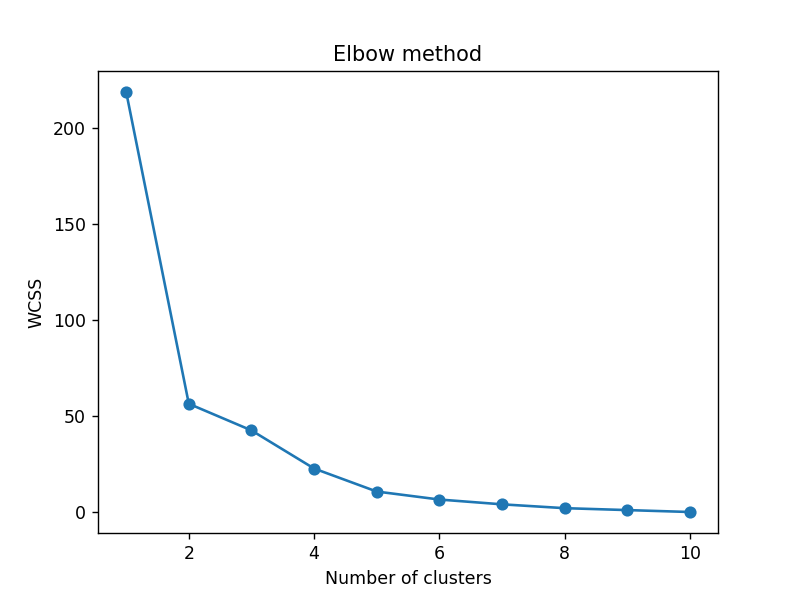


Figure : Elbow method

The Elbow method montre que 2 est une bonne valeur pour K.

* **Étape 2 : Sélectionnez k points aléatoires parmi les données comme centroïdes**

Ensuite, nous sélectionnons aléatoirement le centroïde de chaque cluster. Supposons que nous souhaitions avoir deux clusters, donc k est égal à 2. Nous sélectionnons ensuite aléatoirement le centroïde

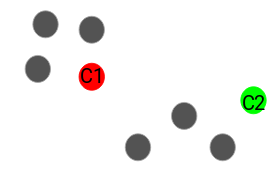


Figure : k centroïdes

Ici, les cercles rouges et verts représentent le centroïde de ces clusters.

* **Étape 3 : Attribuer tous les points au centroïde du cluster le plus proche :**

Une fois les centroïdes initialisés, nous attribuons chaque point au centroïde du cluster le plus proche.

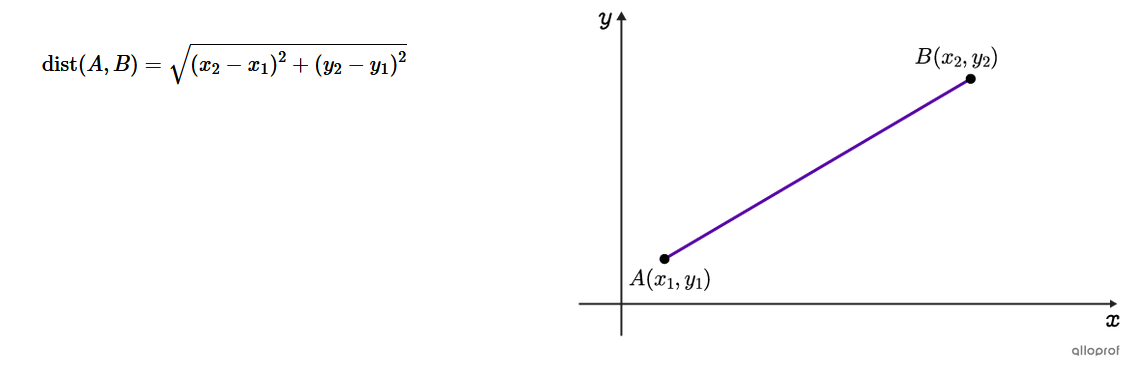


Figure : Distance entre deux points

Ici, vous pouvez voir que les points les plus proches du point rouge sont attribués au groupe rouge, tandis que les points les plus proches du point vert sont attribués au groupe vert.

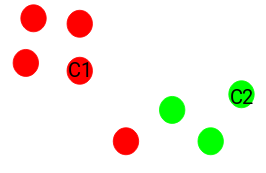


Figure : Attribuer les points au centroïde

* **Étape 4 : Recalculer les centroïdes ou la moyenne des clusters nouvellement formés**

Maintenant, une fois que nous avons attribué tous les points à l’un ou l’autre cluster, l’étape suivante consiste à calculer les centroïdes des clusters nouvellement formés.

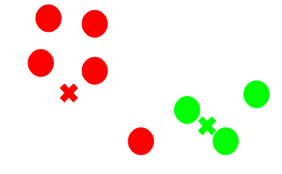


Figure : Recalculer les centroïdes

Ici, les croix rouges et vertes sont les nouveaux centroïdes.

* **Étape 5 : Répétez les étapes 3 et 4**

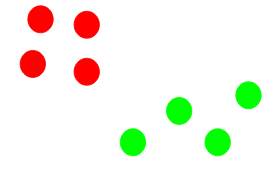


Figure : Affectation les points au cluster

L'étape de calcul du centroïde et d'affectation de tous les points au cluster en fonction de leur distance au centroïde est une seule itération.

* **Trois principaux critères d'arrêt pour l'algorithme K-means :**

Les centroïdes des clusters nouvellement formés ne changent pas ; Les points restent dans le même cluster ; Le nombre maximal d'itérations est atteint.

## Training & Test data (3)

La notion de données d'entraînement et de test n’est pas aussi rigide que dans l’apprentissage supervisé. Dans le cas du clustering K-Means, les données de test sont utilisées principalement pour évaluer la cohérence et la généralisation des clusters appris. Concrètement, chaque point du jeu de test est affecté à l’un des clusters en se basant sur les centroïdes calculés à partir des données d’entraînement. Cette assignation peut ensuite être évaluée à l’aide de différentes métriques, telles que la distance aux centroïdes et silhouette\_score.

# Implémentation du clustering K-Means en Python (5)

|  |  |
| --- | --- |
| Commencez par visualiser quelques points de données |  |
| Nous utilisons maintenant la méthode du coude pour visualiser l'inertie pour différentes valeurs de K |  |
|  |  |

Tableau :Implémentation K-Means en Python

# Conclusion (3)

L’apprentissage non supervisé est puissant pour analyser des données non étiquetées et identifier des tendances cachées, mais il reste plus difficile à interpréter et optimiser par rapport à l’apprentissage supervisé.

# Bibliographie

1. **Ascoli, Stéphane d'.** *Intelligence artificielle -en 5 minutes par jour.*

2. ***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (Aurélien Géron) (Z-Library).* 3.**

**3. ChatGPT. *https://chatgpt.com.* [En ligne]**

**4. tpointtech. [En ligne] https://www.tpointtech.com/k-means-clustering-algorithm-in-machine-learning. 4.**

**5. scikit-learn.org. [En ligne] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. 5.**

**6. w3schools. [En ligne] https://www.w3schools.com/python/python\_ml\_k-means.asp.**

# Liste des tableaux

[Tableau 1:Implémentation K-Means en Python 8](#_Toc195006864)

# Table des figures

[Figure 1: Apprentissage supervisé et non-supervisé 2](#_Toc195006878)

[Figure 2: Clustering (classification non supervisée) 3](#_Toc195006879)

[Figure 3: K-Means Algorithm 3](#_Toc195006880)

[Figure 4: Points de données 4](#_Toc195006881)

[Figure 5: Elbow method 5](#_Toc195006882)

[Figure 6: k centroïdes 5](#_Toc195006883)

[Figure 7: Distance entre deux points 5](#_Toc195006884)

[Figure 8: Attribuer les points au centroïde 6](#_Toc195006885)

[Figure 9: Recalculer les centroïdes 6](#_Toc195006886)

[Figure 10: Affectation les points au cluster 6](#_Toc195006887)