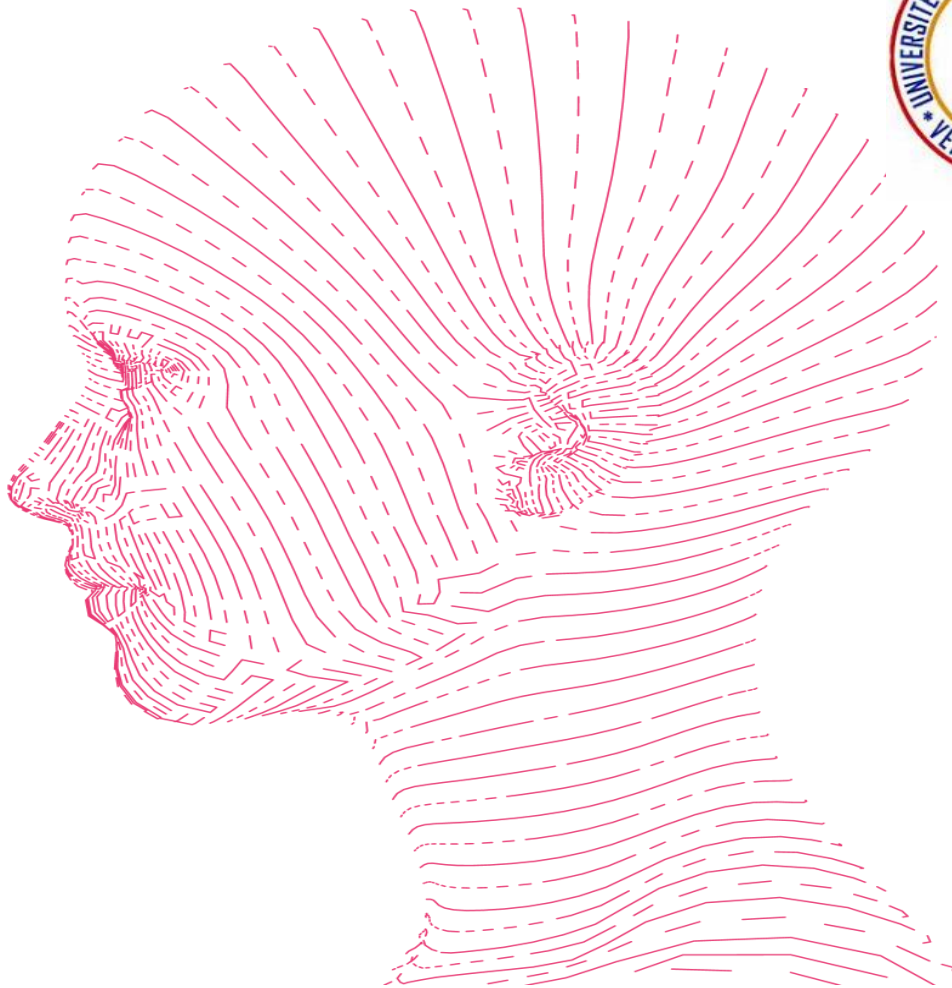
Machine Learning



- Dispensé par **MWAMBA KASONGO Dahouda**
- Docteur en génie logiciel et systèmes d'information
- Machine and Deep Learning Engineer

Assisté par Ass. **Daniel MBAYA**

- E-mail : dahouda37@gmail.com
- Tel.: **+243 99 66 55 265**

Heure : 08H00 – 12H00

PLAN DU COURS

CHAPITRE 4 Apprentissage non supervisé : Clustering

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.1 K-Means Clustering Algorithm

Le clustering K-Means est un algorithme d'apprentissage non supervisé utilisé pour résoudre les problèmes de clustering dans le machine learning ou la science des données. Dans ce chapitre, nous apprendrons ce qu'est l'algorithme de clustering K-means, comment l'algorithme fonctionne, ainsi que l'implémentation Python du clustering K-means.

4.2 Définition de Clustering k-Means

❖ Le clustering K-Means est un algorithme d'apprentissage non supervisé qui regroupe l'ensemble de données non étiqueté en différents clusters. Et K définit le nombre de clusters prédéfinis qui doivent être créés dans le processus, si K=2, il y aura deux clusters, et pour K=3, il y aura trois clusters, et ainsi de suite.

❖ Il s'agit d'un algorithme itératif qui divise l'ensemble de données non étiqueté en k clusters différents de telle sorte que chaque ensemble de données n'appartienne qu'à un seul groupe ayant des **propriétés similaires**.

Il nous permet de regrouper les données en différents groupes et constitue un moyen pratique de découvrir les catégories de groupes dans l' ensemble de données non étiqueté sans avoir besoin d'aucun entrainement.

❖ Il s'agit d'un algorithme basé sur le **centroïde**, où chaque cluster est associé à un centroïde.
❖ L'objectif principal de cet algorithme est de minimiser la somme des distances entre le point de données et leurs clusters correspondants.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

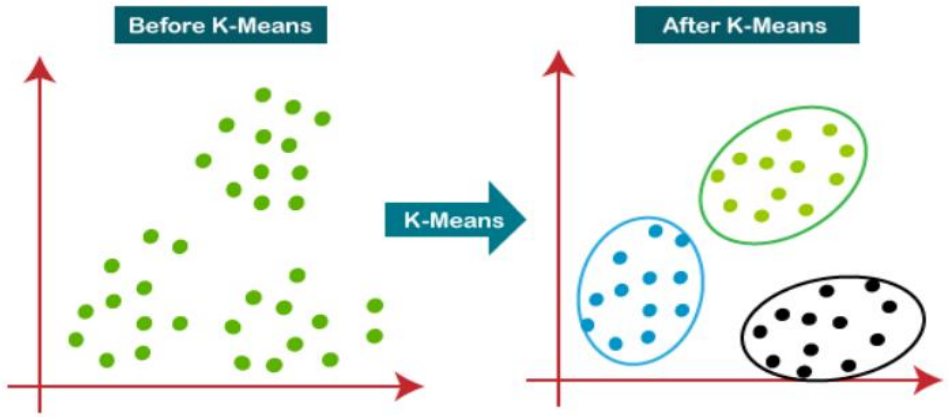
CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

- ❑ L'algorithme prend l'ensemble de données non étiqueté comme entrée, divise l'ensemble de données en k-nombre de clusters et répète le processus jusqu'à ce qu'il ne trouve pas les meilleurs clusters. La valeur de k doit être prédéterminée dans cet algorithme.
- ❑ L'algorithme de clustering k-means effectue principalement deux tâches :

- ✓ Détermine la meilleure valeur pour K points centraux ou centroïdes par un processus itératif.
- ✓ Affecte chaque point de données à son centre k le plus proche. Les points de données qui sont proches du centre k particulier créent un cluster.

- ❑ Le diagramme ci-dessous explique le fonctionnement de l'algorithme de clustering K-means :



CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

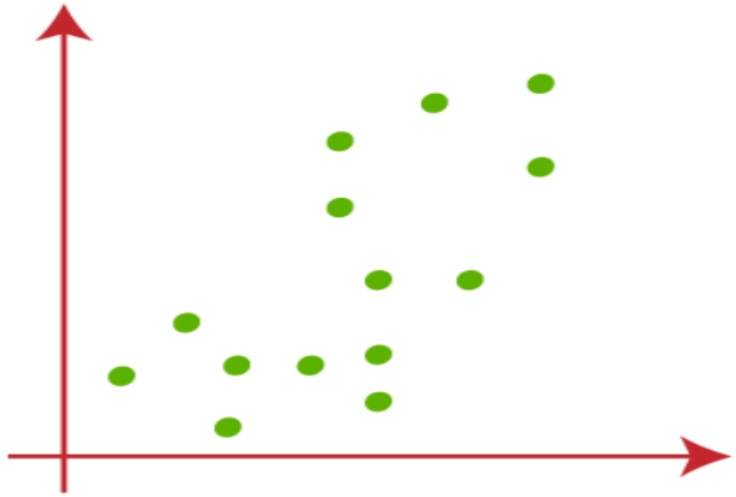
Le fonctionnement de l'algorithme K-Means est expliqué dans les étapes ci-dessous :

Étape 1 : sélectionnez le nombre K pour déterminer le nombre de clusters.
Étape 2 : sélectionnez des points K ou des centroïdes aléatoires. (Il peut s'agir d'autres éléments de l'ensemble de données d'entrée).
Étape 3 : attribuez chaque point de données à son centroïde le plus proche, qui formera les clusters K prédéfinis.
Étape 4 : calculez la variance et placez un nouveau centroïde de chaque cluster.
Étape 5 : répétez les troisièmes étapes, ce qui signifie réaffecter chaque point de données au nouveau centroïde le plus proche de chaque cluster.
Étape 6 : si une réaffectation se produit, passez à l'étape 4, sinon passez à TERMINER.
Étape 7 : le modèle est prêt.

Comprenons les étapes ci-dessus en considérant les tracés visuels :

- Supposons que nous ayons deux variables M1 et M2.

Le diagramme de scatter de l'axe x-y de ces deux variables est le suivant:



CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

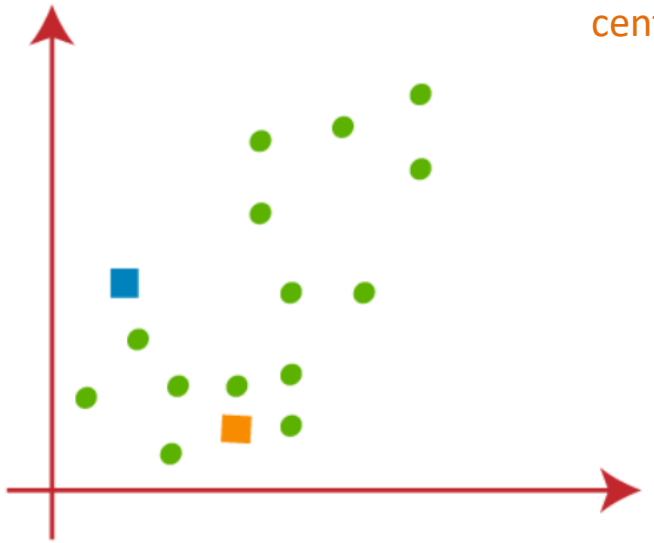
4.3 Fonctionnement de K-Means Clustering Algorithm

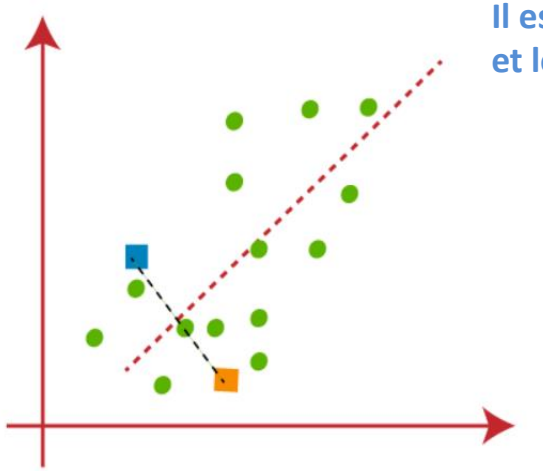
❖ Prenons un nombre k de clusters, c'est-à-dire K=2, pour identifier l'ensemble de données et les placer dans différents clusters.
Cela signifie qu'ici nous allons essayer de regrouper ces ensembles de données en deux clusters différents.

❖ Nous devons choisir des points k aléatoires ou un centroïde pour former le cluster.
Ces points peuvent être soit les points de l'ensemble de données, soit tout autre point.

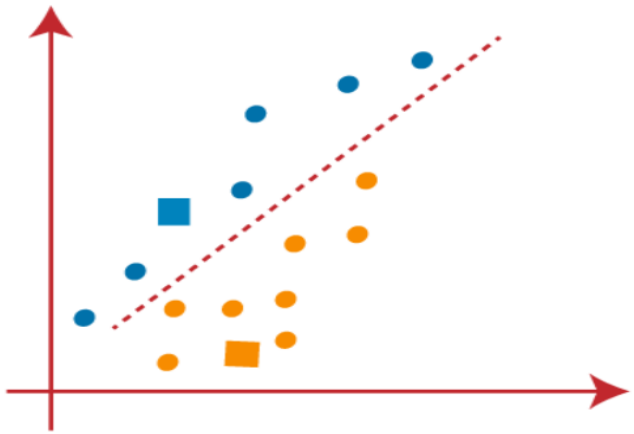
❖ Nous sélectionnons donc ici les deux points ci-dessous comme points k, qui ne font pas partie de notre ensemble de données.

Nous allons maintenant attribuer chaque point de données du nuage de points à son point K ou centroïde le plus proche.





Il est clair que les points à gauche de la ligne sont proches du centroïde K1 ou bleu, et les points à droite de la ligne sont proches du centroïde jaune.

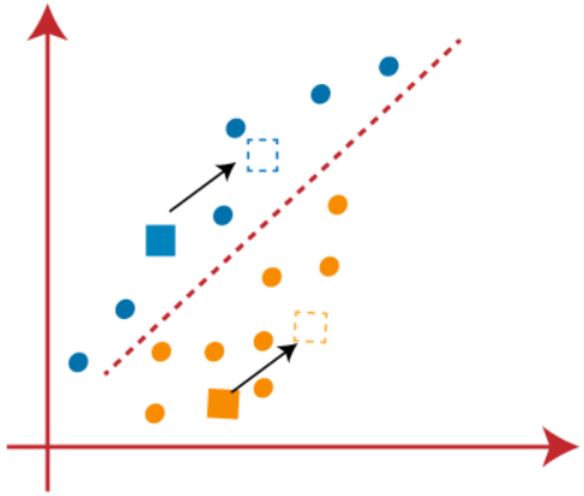


CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

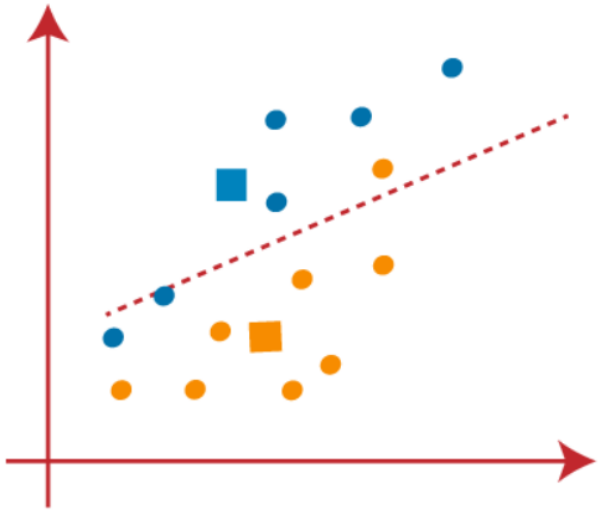
CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

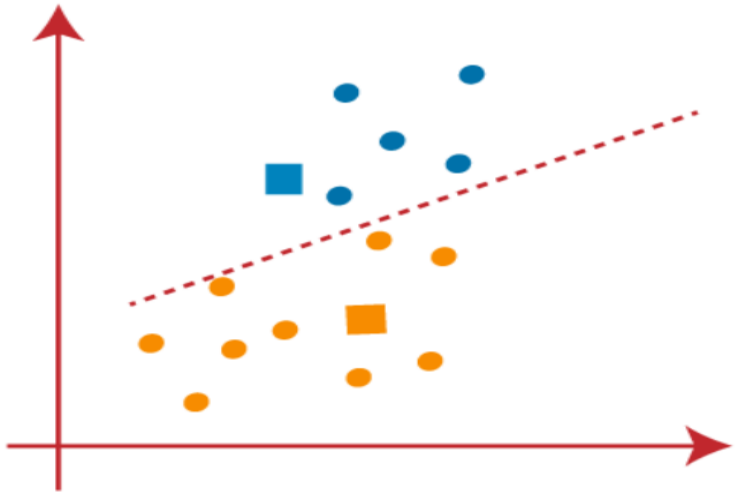
Comme nous devons trouver le cluster le plus proche, nous allons répéter le processus en choisissant un nouveau centroïde.



Ensuite, il va réaffecter chaque point de données au nouveau centroïde.
Pour cela, il va répéter le même processus de recherche d'une ligne médiane.
La médiane sera comme dans l'image ci-dessous:



Sur l'image du milieu, nous pouvons voir qu'un point jaune se trouve sur le côté gauche de la ligne et que deux points bleus se trouvent à droite de la ligne.
Ces trois points seront donc attribués à de nouveaux centroïdes.

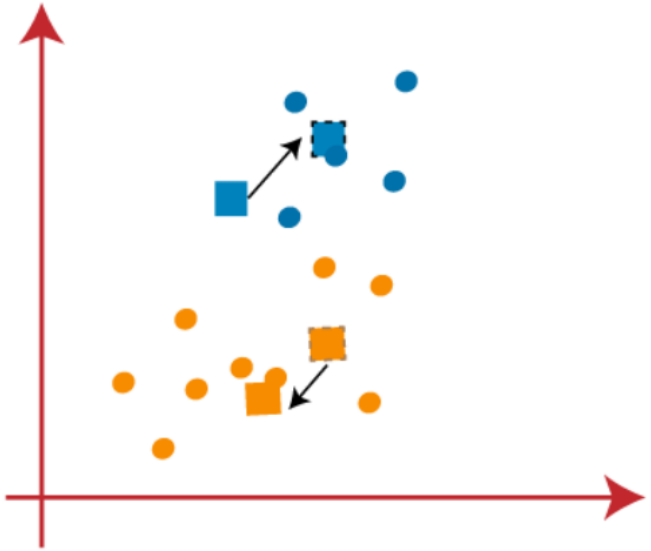


CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

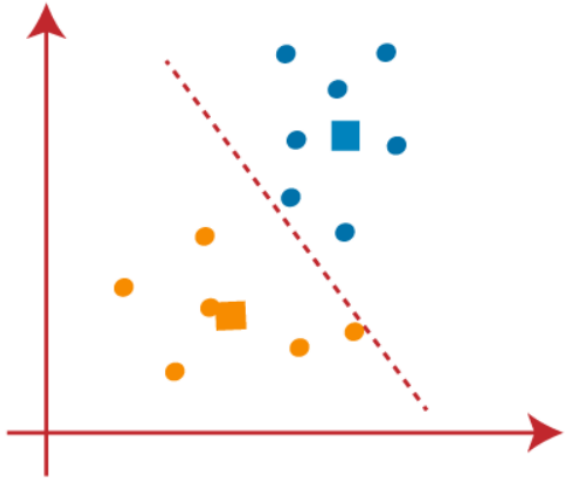
CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

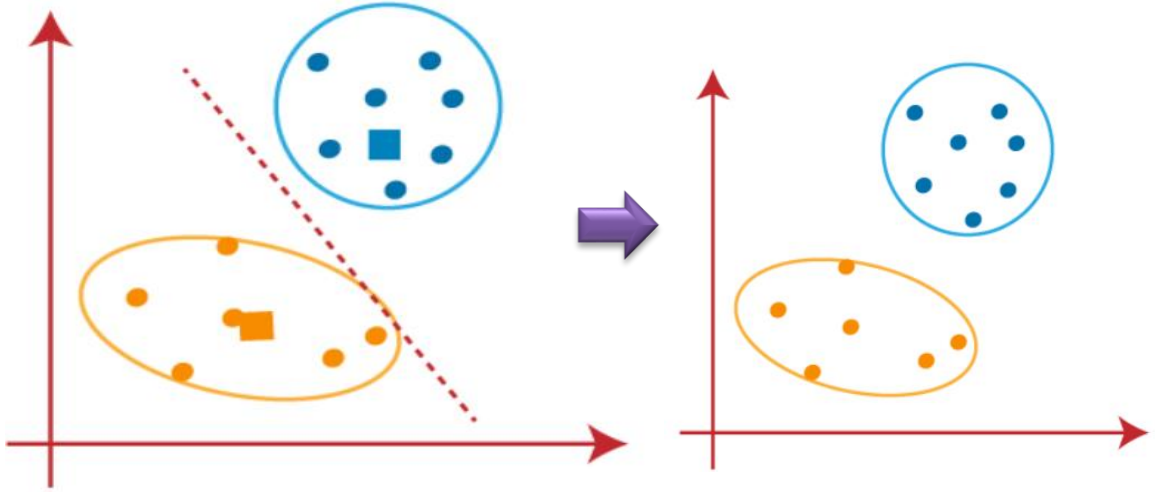
❖ La réaffectation ayant eu lieu, nous allons donc passer à l'étape 4, qui consiste à trouver de nouveaux centroïdes ou points K. Nous allons répéter le processus en trouvant le centre de gravité des centroïdes, de sorte que les nouveaux centroïdes seront tels qu'illustrés dans l'image ci-dessous :



Comme nous avons obtenu les nouveaux centroïdes, nous allons à nouveau tracer la ligne médiane et réaffecter les points de données. L'image sera donc :



Nous pouvons voir dans l'image ci-dessus qu'il n'y a pas de points de données différents de chaque côté de la ligne, ce qui signifie que notre modèle est formé. Considérez l'image ci-dessous.



CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

4.3.1 Comment choisir la valeur de « nombre K de clusters » ?

- Les performances de l'algorithme de clustering K-means dépendent des clusters hautement efficaces qu'il forme.

Mais choisir le nombre optimal de clusters est une tâche difficile.

- Il existe différentes manières de trouver le nombre optimal de clusters, mais nous allons ici discuter de la méthode la plus appropriée pour trouver le nombre de clusters ou la valeur de K. La méthode est Elbow Method.

4.3.2 Elbow Method

- La méthode Elbow (méthode du coude) est une technique utilisée en apprentissage non supervisé, notamment dans l'algorithme de k-means clustering, pour déterminer le nombre optimal de clusters k.
- Elle vise à trouver le bon compromis entre le nombre de clusters et la variance expliquée (ou compacité des clusters).
- Concepts clés :

✓ **Within-Cluster Sum of Squares** (WCSS) [Somme des carrés intra-cluster] : La somme des distances au carré entre chaque point et le centroïde de son cluster. L'objectif est de minimiser cette valeur.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

4.3.3 Étapes de la méthode Elbow

- ✓ **Étapes 1: Exécuter k-means pour une plage de valeurs de k** : commencez par un petit nombre de clusters (par exemple,k=1) et augmentez progressivement k (par exemple, jusqu'à k=10).

- ✓ **Étapes 2: Calculer la somme des carrés intra-cluster WCSS pour chaque k** : Pour chaque nombre de clusters, calculez la somme des carrés intra-cluster. Plus le nombre de clusters augmente, plus WCSS diminue car les points de données sont plus proches de leur centroïde respectif.

- ❑ La formule du WCSS est : $$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

où C_i représente le i-ème cluster,
x est un point de données, et
μ_i est le centroïde du cluster i.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

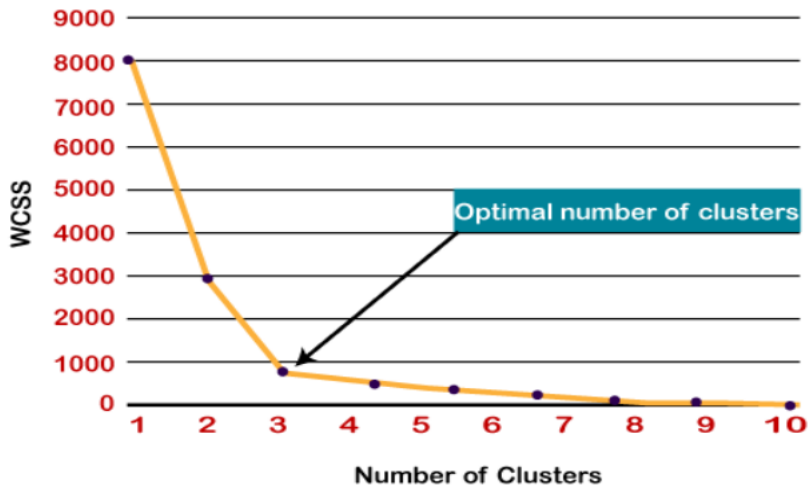
CHAPITRE 4 Apprentissage non supervisé : Clustering

4.3 Fonctionnement de K-Means Clustering Algorithm

4.3.3 Étapes de la méthode Elbow

- ✓ **Étapes 3 : Tracer WCSS contre k** : Créez un graphique en mettant k sur l'axe des abscisses et WCSS sur l'axe des ordonnées.
  En général, la somme des carrés diminue rapidement au début, puis la diminution devient plus lente à mesure que k augmente.

- ✓ **Étapes 4 : Identifier le "coude" (elbow)** : Le point où la diminution de WCSS devient moins significative est appelé le "coude".
  C'est l'endroit où l'ajout de clusters supplémentaires n'améliore plus beaucoup la qualité de la partition.
  Ce point correspond au nombre optimal de clusters.



CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.1 Comprendre le problème

Nous devons comprendre quel type de problème nous allons résoudre ici. Nous disposons donc d'un ensemble de données (Mall_Customers dataset), qui contient les données des clients qui visitent le centre commercial et y dépensent.

✓ Dans l'ensemble de données, nous avons des attributs suivants :

- Customer_Id [numero du client],
- Gender [Genre],
- Age,
- Annual Income [Revenu annuel] ($) et
- Spending Score (qui est la valeur calculée du montant dépensé par un client dans le centre commercial, plus la valeur est élevée, plus il a dépensé).

✓ Il s'agit d'une méthode non supervisée, nous ne savons donc pas exactement quelles sont les catégories.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.2 segmentation de la clientèle avec l'algorithme de clustering K-means

Étape 1 : Importer les packages nécessaires et de la dataset

```
# importation de bibliothèques
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
# Importer la dataset
dataset = pd.read_csv('Mall_Customers.csv')
dataset
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Code Python où on utilises les bibliothèques importées pour appliquer la méthode Elbow avec l'algorithme k-means clustering.

- ✓ **pd.read_csv('Mall_Customers.csv')** indique à pandas de lire le fichier 'Mall_Customers.csv' qui se trouve dans le même répertoire que le script Python.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.2 segmentation de la clientèle avec l'algorithme de clustering K-means

Étape 2 : Étape de prétraitement des données

```
1 dataset['Genre'] = np.where(dataset['Genre'] == 'Male', 1, 0)
```

❖ Le résultat de np.where() remplace la colonne Genre dans le DataFrame.
Ainsi, la colonne Genre qui contenait des chaînes de caractères ('Male', 'Female')
est maintenant transformée en valeurs numériques :1 pour 'Male' , 0 pour 'Female'.

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

```
1 # Extraction de variables indépendantes
2
3 x = dataset.iloc[:, 3:5].values
```

❖ L'instruction **x = dataset.iloc[:, 3:5].values** va créer un tableau NumPy avec les valeurs des colonnes "Annual Income" et "Spending Score" comme ceci :

```
[ 76,  40],
[ 76,  87],
[ 77,  12],
[ 77,  97],
[ 77,  36],
```

✓ **.values** : Cette méthode extrait les valeurs du DataFrame sous forme de tableau NumPy. Cela est pratique pour utiliser les données avec des algorithmes de machine learning.

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

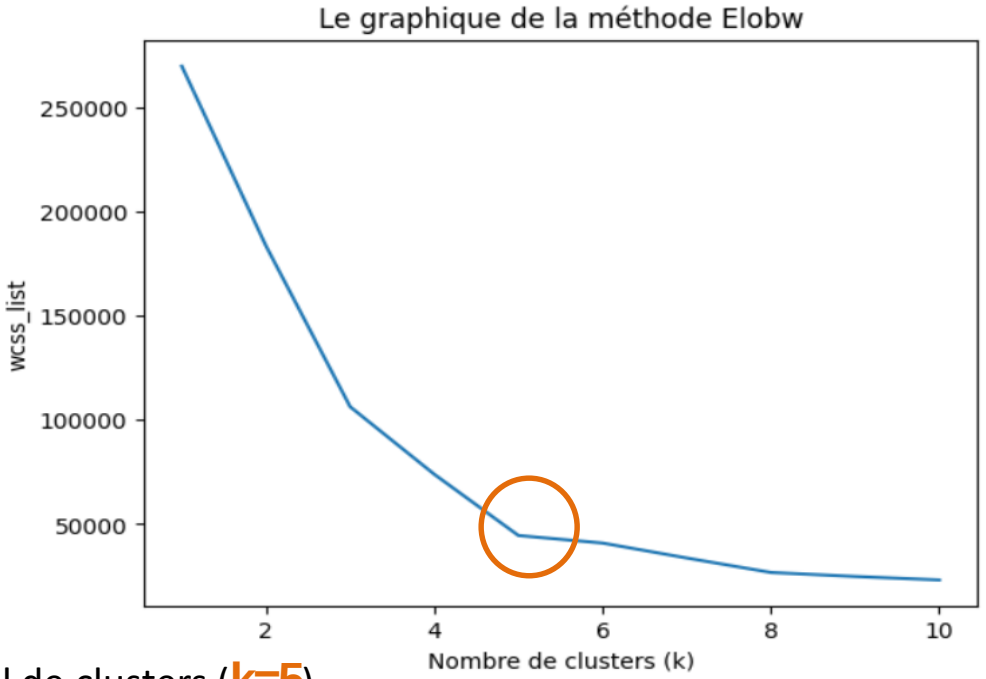
4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.2 segmentation de la clientèle avec l'algorithme de clustering K-means

Étape 3 : Trouver le nombre optimal de clusters à l'aide de la méthode du coude

```python
#trouver le nombre optimal de clusters en utilisant la méthode du coude
from sklearn.cluster import KMeans
wcss_list= []  #Initialisation de la liste des valeurs de WCSS

#Utilisation de la boucle for pour les itérations de 1 à 10.
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss_list)
plt.title('Le graphique de la méthode Elobw')
plt.xlabel('Nombre de clusters (k)')
plt.ylabel('wcss_list')
plt.show()
```



- **Identification du coude (elbow)** : Ce point correspond au nombre optimal de clusters (**k=5**).

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

CHAPITRE 4 Apprentissage non supervisé : Clustering

4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.2 segmentation de la clientèle avec l'algorithme de clustering K-means

Étape 4 : Entraînement de l'algorithme K-means sur les données d'entraînement

```
1 #Entraînement du modèle K-means sur un ensemble de données
2 kmeans = KMeans(n_clusters=5, init='k-means++', random_state= 42)
3
4 # Prediction de cluster
5 y_predict= kmeans.fit_predict(x)
```

- ✓ **KMeans(n_clusters=5)** : Crée un objet de l'algorithme K-means avec 5 clusters.
- ✓ **init='k-means++'** : k-means++ est une méthode d'initialisation améliorée pour choisir les centroïdes initiaux.

- ➢ **fit_predict()** combine deux étapes :
- ✓ **fit** : Entraîne le modèle k-means sur les données x.
- ✓ **predict** : Après avoir trouvé les clusters, il attribue chaque point de données à un cluster.

```
1 y_predict
```

```
array([4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 0,
       4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 1, 3, 1, 3, 1,
       0, 1, 3, 1, 3, 1, 3, 1, 3, 1, 0, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1], dtype=int32)
```

- ✓ **y_predict** : Cette variable contient les étiquettes de cluster pour chaque point de données.

C'est un tableau 1D où chaque élément est un entier représentant le numéro du cluster auquel un point a été attribué (de 0 à 4, car il y a 5 clusters).

CHAPITRE 3 APPRENTISSAGE NON SUPERVISE

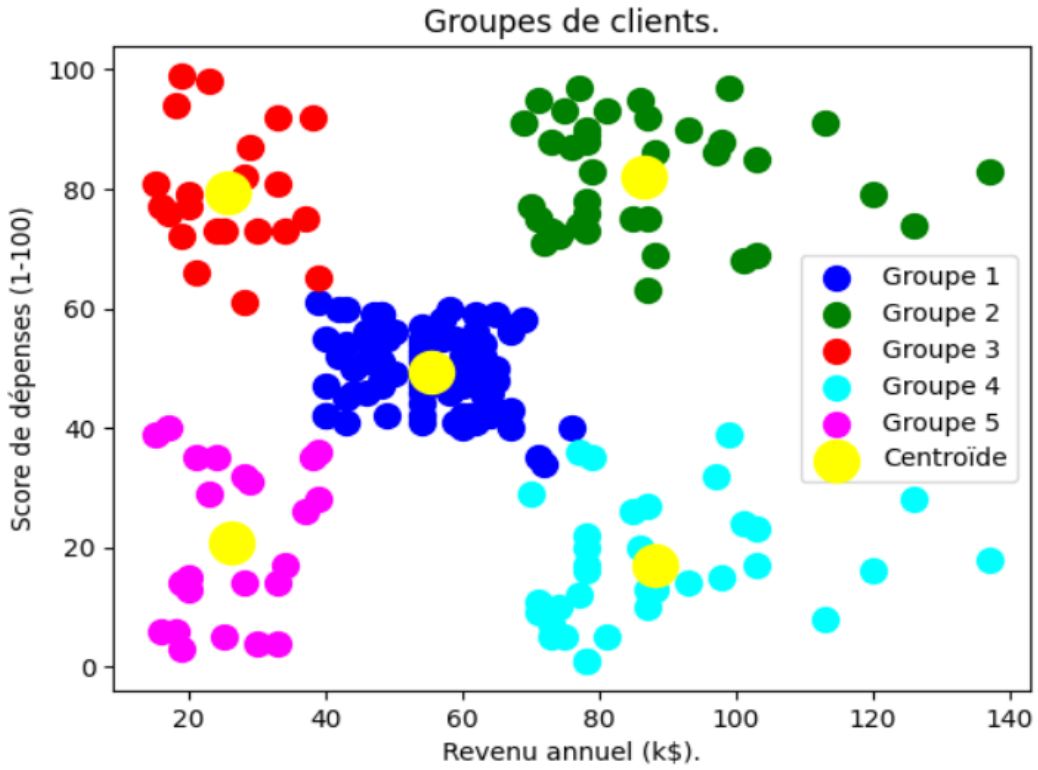
CHAPITRE 4 Apprentissage non supervisé : Clustering

4.4 Implémentation Python de l'algorithme de clustering K-means

4.4.2 segmentation de la clientèle avec l'algorithme de clustering K-means

Étape 5 : Visualisation des clusters

```
#Visualisation des clusters
plt.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Groupe 1') # Pour le premier groupe.
plt.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Groupe 2') # Pour le deuxième groupe.
plt.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label = 'Groupe 3') # Pour le troisième groupe.
plt.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c = 'cyan', label = 'Groupe 4') # Pour le quatrième groupe.
plt.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c = 'magenta', label = 'Groupe 5') # Pour le cinquième groupe
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroïde')
plt.title('Groupes de clients.')
plt.xlabel('Revenu annuel (k$).')
plt.ylabel('Score de dépenses (1-100)')
plt.legend()
plt.show()
```



PLAN DU COURS

CHAPITRE 5 Projet pratiques

- Projet 1 : Sur la Régression
  - Utiliser un ensemble de données, appliquer une régression linéaire, évaluer le modèle
- Projet 2 : Sur la Classification
  - Utiliser un ensemble de données, appliquer une régression logistique ou autres algorithmes
- Projet 3 : Machine Learning avec AWS

References

[1]. "Python for Data Analysis" by Wes McKinney (2nd Edition, 2017)
[2]. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett (1st Edition, 2013)
[3]. "Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani (1st Edition, 2013)
[4]. "Data Science from Scratch: First Principles with Python" by Joel Grus (1st Edition, 2015)
[5]. "Machine Learning Yearning" by Andrew Ng (1st Edition, 2018)
[6]. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2nd Edition, 2009)
[7]. "Storytelling with Data: A Data Visualization Guide for Business Professionals" by Cole Nussbaumer Knaflic (1st Edition, 2015)
[8]. "Data Science for Dummies" by Lillian Pierson (1st Edition, 2015)
[9]. "Practical Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce (1st Edition, 2017)
[11]. "Python Data Science Handbook" by Jake VanderPlas (1st Edition, 2016)
[12]. "Big Data: A Revolution That Will Transform How We Live, Work, and Think" by Viktor Mayer-Schönberger and Kenneth Cukier (1st Edition, 2013)
[13]. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei (3rd Edition, 2011)
[15]. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (1st Edition, 2016)
[14]. "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die" by Eric Siegel (1st Edition, 2013)
[15]. "Data Science for Social Good: Building AI Applications Sustainably" by Rayid Ghani, David Uminsky, and Dennis Wei (1st Edition, 2018)
[16]. "R for Data Science" by Hadley Wickham and Garrett Grolemund (1st Edition, 2017)
[17]. "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing, and Presenting Data" by EMC Education Services (1st Edition, 2015)
[18]. "Data Analysis with Open Source Tools" by Philipp K. Janert (1st Edition, 2010)
[19]. "Mining of Massive Datasets" by Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman (1st Edition, 2014)
[20]. "Data Science at the Command Line: Facing the Future with Time-Tested Tools" by Jeroen Janssens (1st Edition, 2014)

References

[21]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. ISBN: 978-0262035613.
[22]. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer. ISBN: 978-0387310732.
[23]. Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press. ISBN: 978-0262018029.
[24]. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.).
[25]. Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media. ISBN: 978-1492032649.
[26]. Russell, S. J., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson. ISBN: 978-0134610993.
[27]. Chollet, F. (2018). Deep Learning with Python. Manning Publications. ISBN: 978-1617294433.
[28]. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill. ISBN: 978-0070428072.
[29]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
[30]. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects.