Machine learning

Master 1 Physique Appliquée
2023-2024

Rapport du DM: Choix d'une méthode de machine learning et l'appliquer sur les donneées de Iris de Fischer

---

Idir DRICHE

Contexte général

De nombreuses choses dans notre environnement peuvent être regroupées en catégories telles que "ceci et cela". Pour être plus précis, nous utilisons des regroupements qui peuvent être binaires ou comporter plus de deux options, comme les différentes sortes de pizzas ou les modèles de voitures que vous pourriez envisager d'acheter.

Autrefois, les scientifiques classaient les populations en utilisant des étiquettes préétablies. Ainsi, tout individu partageant certaines caractéristiques était manuellement placé dans une classe avec d'autres individus partageant les mêmes caractéristiques.

Cette approche était également utilisée aux débuts d'Internet par Yahoo! pour classer les pages web.

Cependant, cette méthode a rapidement montré ses limites avec l'essor des volumes de données de plus en plus importants et provenant de sources diverses. Cela a conduit au développement de nouvelles techniques de classification automatique : l'apprentissage automatique supervisé et non supervisé. Parmi ces techniques, le clustering est l'une des plus couramment utilisées.

1 Introduction

Dans ce projet de machine learning, nous avions la possibilité de choisir une méthode et de l'appliquer aux données d'Iris de Fischer.

Pour ma part, j'ai opté pour la méthode des K-means. Dans ce devoir, j'ai cherché à approfondir ma compréhension du fonctionnement des K-means, ainsi que découvrir ses avantages et inconvénients.

Avant de commencer, la partie pratique, on doit faire un petit rappel sur la méthode du K-mean.

La méthode des K-moyennes (K-means) est l'un des algorithmes de clustering les plus populaires en machine learning non supervisé.

Le clustering consiste à regrouper selon un lien (critère) de similarité, une grande quantité de données en plusieurs sous-ensembles appelés clusters. Les éléments contenus dans un cluster sont similaires les uns aux autres, mais différents des éléments des autres clusters. Le but du clustering est de trouver la structure inhérente aux données et de l'exprimer sous la forme d'une collection de clusters.

En fait, lorsqu'il existe plusieurs modèles concurrents dans les données, il est difficile d'extraire un modèle spécifique, c'est pourquoi la création de clusters peut réduire la complexité en organisant les données en clusters

L'objectif principal de K-means est de partitionner un ensemble de données en un certain nombre (K) de groupes ou de clusters, où chaque point de données est attribué au cluster le plus proche en termes de similarité.

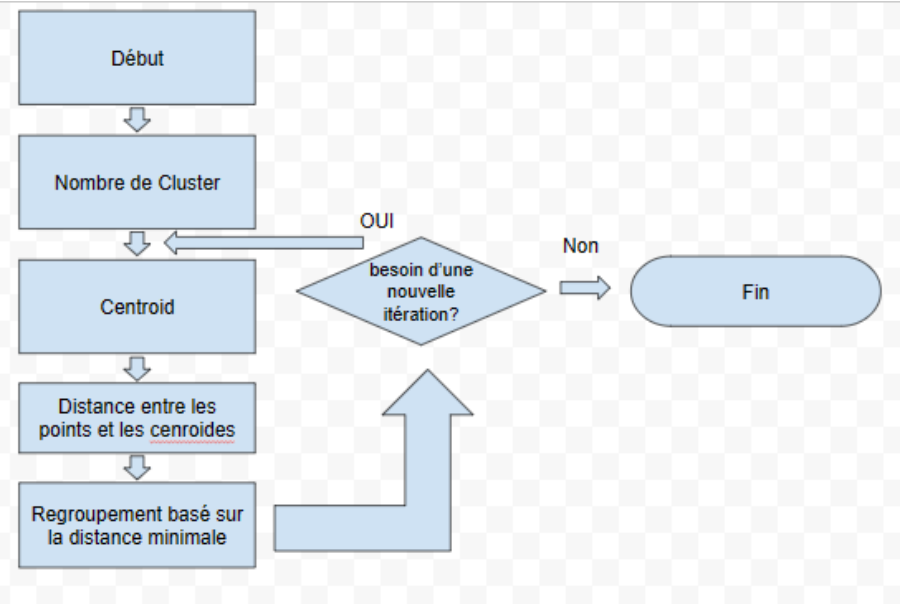


Figure 1: Principe de fonctionnement de l'algorithme de K-mean

Comme on l'a vu durant nos séances de TP; on doit préciser le nombre de cluster; après on choisit nos centroïdes (si on a pris deux Clusters donc on prend deux centroïdes), après le code calcul la distance entre chaque point et les centroïdes et les groupe suivant la distance minimale. Après on voit si on a besoin d'une nouvelle itération: on ajoute un autre centroides et le code refait le calul des distances, tandis-que si on a trouvé que les groupes sont bien distribué, alors c'est le fin, on a ce qu'on recherche !

La formule sur laquelle est basée la méthode du K-mean est celle de la distance entre deux points.

$$d(a,b) = \sqrt{\sum_i (a_i - b_i)^2} \tag{1}$$

Où a_i et b_i $(i = 1, n)$ sont les attributs des instances a et b respectivement. En pratique, on n'effectue pas la racine carrée, car on peut comparer directement le carré de la distance.

L'algorithme de clustering K-means est déployé pour découvrir des groupes qui n'ont pas été explicitement définis. Aujourd'hui, on l'utilise activement dans une grande variété d'applications commerciales, notamment :

Pourquoi le K-mean ?

- La segmentation de la clientèle
- Le regroupement de textes, de documents ou de résultats de recherche
- Le regroupement d'images ou compression d'images
- La détection d'anomalies
- L'apprentissage semi-supervisé

2 Code et résultats

Ja vais utiliser le jeu de données des plantes d'iris. Cet ensemble de données se compose de quatre champs, à savoir la longueur du sépale, la largeur du sépale, la longueur du pétale et la largeur du pétale.

Pourquoi utilisons-nous cet ensemble de données ? Nous savons au préalable (vu dans le TP avec le prof) que cet ensemble de données est divisé en trois classes différentes : Iris setosa, Iris versicolor et Iris virginica.

On va appliquer sur cet ensemble de données l'algorithme K-Means pour confirmer ou infirmer cette classification.

Tout abord, importons tous les bibliothèques dont nous aurons besoin dans ce tutoriel.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets
```

2.1 chargement l'ensemble de données

On a travaillé sur les iris. C'est un dataset déjà inclus dans la bibliothèque sklearn et très utilisé en clustering.

Nous pouvons utiliser la méthode `load_iris()` pour extraire les données et ensuite, nous les chargeons dans un dataframe pour mieux les visualiser :

```
iris = datasets.load_iris()
df = pd.DataFrame(iris.data)
df.columns=["Longueur_sepale", "Largeur_sepale", "Longueur_petale", "Largeur_petale"]
df
```

Ce bout de code va générer le tableau suivant:

	Longueur_sepale	Largeur_sepale	Longueur_petale	Largeur_petale
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

Figure 2: Tableau de valeurs

Maintenant, notre ensemble de données est prêt et nous pouvons passer au clustering de nos données.

2.2 Détermination du nombre de Clusters "K"

Selon l'algorithme de K-Means, on doit définir au préalable le nombre K de clusters. Le problème qui se pose est de trouver un K optimal. L'une des méthodes les plus populaires pour y arriver est la méthode d'Elbow.

L'idée est d'exécuter le clustering k-means pour une gamme de clusters k (disons de 1 à 10) et pour chaque valeur, nous calculons l'inertie intraclasse.

Lorsque l'on trace les distorsions et que le tracé ressemble à un bras, le "coude" (**le point d'inflexion de la courbe**) est la meilleure valeur de k.

Le bout de code qui nous permet de réaliser cette tache est le suivant:

```
# Détermination de la valeur optimale de K
tab=[]
for i in range(1,10):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df)
    tab.append(kmeans.inertia_)
plt.plot(range(1,10),tab)
plt.title("La méthode Elbow")
plt.xlabel("nombre de cluster")
plt.ylabel("Inertie intra-classe")
plt.show()
```

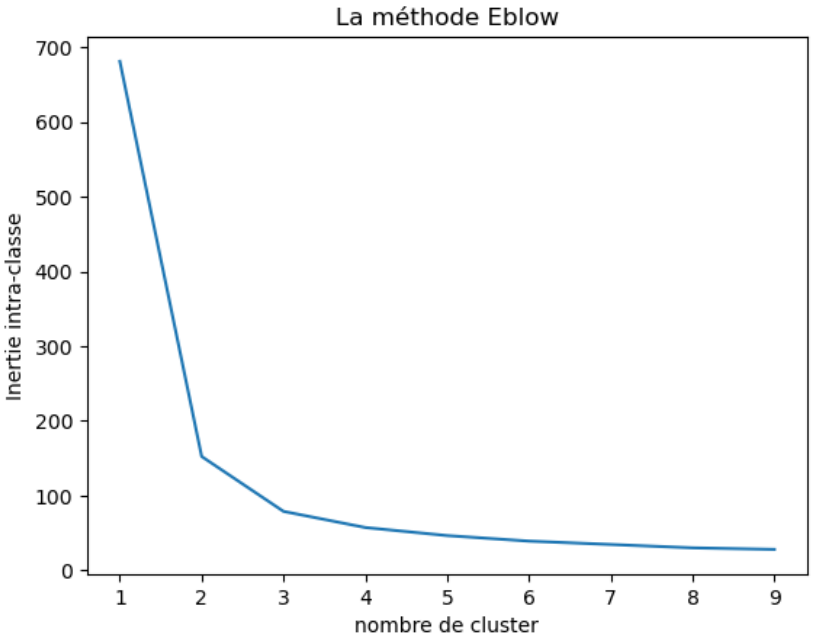


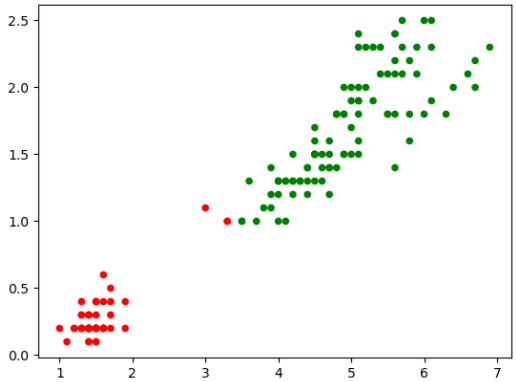
Figure 3: Tracé des distorsions

Sur ce graphique une courbe ayant la forme d'un bras. Selon la méthode d'Elbow, la valeur optimale de doit K=2 soit K est 3. C'est pour cela on doit vérifier, dans l'etape suivate quelle valeur de "K" on doit prendre.

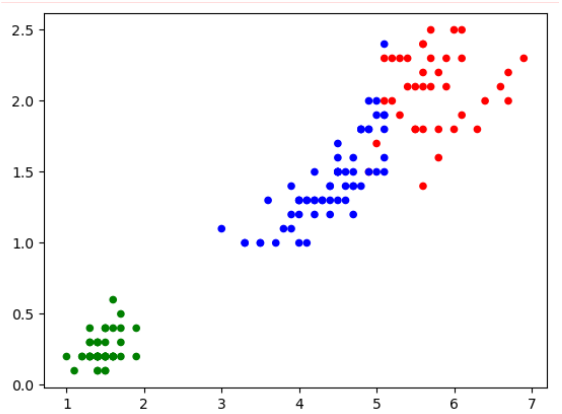
2.3 Application de l'algorithme de K Means

```
# Application de KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(df)

# Visualisation
colormap=np.array(["red", "green", "blue"])
plt.scatter(df.Longueur_petale, df.Largeur_petale, c=colormap[kmeans.labels_], s=20)
plt.show()
```



(a) K = 2



(b) K = 3

La figure (a) montre bien 2 classes d'observations respectivement en vert, rouge; tandis-que la figure (b) montre bien 3 classes d'observations respectivement en vert, rouge et bleue.

La figure (b) présente mieux notre distribution, ce qui est logique car on sait à l'avance qu'on a trois classes différentes

3 Avantages et inconvénients de la méthode

3.1 Avantages

- Relativement simple à mettre en œuvre
- S'adapte aux grands ensembles de données
- La convergence est garantie
- S'adapte facilement à de nouveaux exemples

3.2 Inconvénients

- Le choix du "K" se fait manuellement
- Cette méthode est dépendante des valeurs initiales
- Mise à l'échelle en fonction du nombre de dimensions
- L'utilisation de l'algorithme K-means est limitée car elle nécessite la pré-définition de la valeur moyenne du cluster, ce qui peut ne pas convenir à toutes les applications ;
- Dans l'algorithme K-means, le nombre de clusters K doit être défini à l'avance
- Avant d'appliquer l'algorithme K-means, une partition initiale basée sur le centre de regroupement initial doit être déterminée, puis cette partition doit être optimisée.
- L'algorithme doit constamment ajuster la classification des échantillons et calculer les nouveaux centres de clusters ajustés en permanence. Par conséquent, lorsque la quantité de données est très importante, le temps nécessaire pour exécuter l'algorithme peut être considérable ;
- Si un cluster contient des points aberrants, la valeur moyenne peut être sérieusement faussée ;
- L'algorithme K-means ne convient pas à la découverte de clusters de formes non convexes ou de clusters de tailles très différentes.

4 Conclusion

En conclusion, l'application de la méthode des K-means dans notre projet d'analyse des données d'Iris de Fischer s'est avérée être une approche fructueuse. En explorant en profondeur le fonctionnement de cette méthode, nous avons pu identifier ses avantages, tels que sa simplicité conceptuelle et sa capacité à partitionner efficacement les données en clusters distincts. Cependant, nous avons également pris note de ses inconvénients, notamment sa sensibilité aux points de départ initiaux et sa difficulté à gérer des formes de clusters complexes. Malgré ces limitations, les résultats obtenus ont été prometteurs, ouvrant la voie à de futures explorations et améliorations de cette technique dans des contextes plus vastes du machine learning. En somme, l'utilisation des K-means a été une étape cruciale dans notre analyse, démontrant son utilité et son potentiel pour des applications futures.

5 Annexes

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import datasets

# Chargement des données
iris = datasets.load_iris()
df = pd.DataFrame(iris.data)
df.columns=["Longueur_sepale", "Largeur_sepale", "Longueur_petale", "Largeur_petale"]
df

# Détermination de la valeur optimale de K
tab=[]
for i in range(1,10):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df)
    tab.append(kmeans.inertia_)
plt.plot(range(1,10),tab)
plt.title("La méthode Elbow")
plt.xlabel("Nombre de clusters")
plt.ylabel("Inertie intra-classe")
plt.show()

# Application de KMeans
kmeans = KMeans(n_clusters=2)
kmeans.fit(df)

# Visualisation
colormap=np.array(["red", "green", "blue"])
plt.scatter(df.Longueur_petale, df.Largeur_petale, c=colormap[kmeans.labels_], s=20)
plt.show()
```