Pr. I. HAMZAOUI / Filière : CCN-2
Module : Machine Learning
Année universitaire 2025/2026

Travaux Pratiques N°7

Nom et prénom : ……

Objectifs du TP : L'objectif de ce TP est de:

- Mettre en œuvre l'algorithme de clustering K-means.
- Étudier l'impact du nombre de clusters K et des hyperparamètres de K-means (init, n_init, max_iter…) sur la qualité du clustering.
- Utiliser plusieurs indicateurs de qualité (SSE/inertie, silhouette, Davies-Bouldin, Calinski-Harabasz) pour comparer les configurations.
- Appliquer la méthode du coude (Elbow) pour aider au choix de K.

Livrables à rendre : Complétez ce document manuscritement par le travail demandé.

Données : Vous disposez du fichier CSV 'DatasetTP7'.

- **Colonnes** : x1, x2,
- **Taille** : **2150** lignes.

	x1	x2
0	35.484907	114.369328
1	11.280286	102.558662
2	43.508122	61.869098
3	46.930165	42.136913
4	5.118633	5.293873

1) Sur un nouveau notebook Google Colab, exécutez le code ci-dessous :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score, calinski_harabasz_score

df = pd.read_csv("DatasetTP7.csv")
plt.figure()
plt.scatter(df["x1"], df["x2"], s=5)
plt.title("Données originales")
plt.xlabel("x1")
plt.ylabel("x2")
plt.show()
```

2) Appliquez une mise à l'échelle sur les colonnes x1 et x2 afin d'obtenir deux nouvelles variables dont les valeurs sont comprises strictement entre 0 et 1.

```
# Import du scaler

# On extrait les colonnes x1 et x2 sous forme de matrice numpy (n_lignes, 2)


# Créer un objet du scaler et appliquer les transformations sur les données



# Visualiser les données normalisées
```

---

3) Exécutez le code ci-dessous

```
configs1 = [
    {"name": "Kmeans1", "K": 3, "init": "k-means++", "n_init": 10, "max_iter": 400},
    {"name": "Kmeans2", "K": 2, "init": "random",    "n_init": 10, "max_iter": 450},
    {"name": "Kmeans3", "K": 4, "init": "k-means++", "n_init": 1,  "max_iter": 300},
    {"name": "Kmeans5", "K": 5, "init": "random", "n_init": 10, "max_iter": 200},
]
```

```
def evaluer_kmeans_configs(X, configs, random_state=0):
    resultats = []
    for cfg in configs:
        kmeans = KMeans(
            n_clusters=cfg["K"],
            init=cfg["init"],
            n_init=cfg["n_init"],
            max_iter=cfg["max_iter"],
            random_state=random_state
        )
        labels = kmeans.fit_predict(X)

        sse = kmeans.inertia_
        sil = silhouette_score(X, labels)
        dbi = davies_bouldin_score(X, labels)
        ch  = calinski_harabasz_score(X, labels)

        resultats.append({
            "Config": cfg["name"],
            "K": cfg["K"],
            "init": cfg["init"],
            "n_init": cfg["n_init"],
            "max_iter": cfg["max_iter"],
            "SSE": sse,
            "Silhouette": sil,
            "DBI": dbi,
            "CH": ch
        })
    return pd.DataFrame(resultats)
```

4) Appelez la fonction evaluer_kmeans_configs sur les données normalisées X_scaled en utilisant la liste configs1 définie à la question précédente et reprenez ci-dessous le tableau obtenu.

```
# Appelez la fonction evaluer_kmeans_configs et affichez le résultat
```

	K	init	n_init	max_iter	SSE	silhouette	DBI	CH
Kmeans1	3	k-means++	10	400				
Kmeans2	2	random	10	450				
Kmeans3	4	k-means++	1	300				
Kmeans4	5	random	10	200				

Lequel des modèles semble le meilleur ?

Réponse :

5) Exécutez le code de fonction de visualisation ci-dessous. Appelez cette fonction et visualisez le résultat des clustering sous les 4 configurations.

```
def visualiser_configs_2x2(X, configs, random_state=0):
    fig, axes = plt.subplots(2, 2, figsize=(10, 8))
    axes = axes.ravel()

    for ax, cfg in zip(axes, configs):
        kmeans = KMeans(
            n_clusters=cfg["K"],
            init=cfg["init"],
            n_init=cfg["n_init"],
            max_iter=cfg["max_iter"],
            random_state=random_state
        )
        labels = kmeans.fit_predict(X)
        centers = kmeans.cluster_centers_

        ax.scatter(X[:, 0], X[:, 1], c=labels, s=8)
        ax.scatter(centers[:, 0], centers[:, 1],
                   c="red", marker="X", s=80, edgecolors="black")
        ax.set_title(f"{cfg['name']}  (K={cfg['K']}, init={cfg['init']}, "
                     f"n_init={cfg['n_init']}, max_iter={cfg['max_iter']})")
        ax.set_xlabel("x1 (scaled)")
        ax.set_ylabel("x2 (scaled)")

    plt.tight_layout()
    plt.show()
```

6) En conservant exactement les hyperparamètres de la meilleure configuration précédente, faites varier uniquement le nombre de clusters K de 2 à 8, et évaluez chaque modèle.

```
# À compléter  d'après la config jugée "meilleure" à la Q4
init_best = "..."
n_init_best = ...
max_iter_best = ...

# Valeurs de K à tester
k_values = range(2, 9)   # 2,3,4,5,6,7,8

# Construction des configurations : on garde init, n_init, max_iter,
#    on ne change QUE le nombre de clusters K
configs_k = [
    {
        "name": f"Model_K{k}",
        "K": k,
        "init": init_best,
        "n_init": n_init_best,
        "max_iter": max_iter_best,
    }
    for k in k_values
]
```

7) Appelez la même fonction précédente evaluer_kmeans_configs sur les données normalisées X_scaled en utilisant configs_k

```
# Évaluation avec la même fonction que précédemment
```

K	SSE	silhouette	DBI	CH
2				
3				
4				
5				
6				
7				
8				

8) Exécutez le code demandé ci-dessous pour appliquer la méthode de coude (elbow).

```
# Crée une nouvelle figure (fenêtre de dessin) de taille 6x4 pouces


# Trace la courbe de l'inertie (SSE) en fonction de K
#  - en abscisse : la colonne "K"
#  - en ordonnée : la colonne "SSE"
#  - marker="o" : met un petit rond sur chaque point de la courbe


# Force les graduations de l'axe des x à être exactement les valeurs de K testées


# Légende de l'axe des x


# Légende de l'axe des y


# Titre du graphique


# Ajoute une grille légère en arrière-plan
#  True : on active la grille
#  linestyle="--" : tirets
#  alpha=0.5 : transparence (50 %)


# Affiche la figure à l'écran
```

9) En vous appuyant à la fois sur le tableau des résultats (SSE, Silhouette, DBI, CH) de la question 7 et sur la courbe du coude (SSE en fonction de K) (question 8), dites si la valeur de K que vous aviez jugée “meilleure” précédemment est confirmée ou non par la méthode du coude. Si ce n’est pas le cas, proposez une valeur de K et justifiez brièvement votre choix.

Réponse :