

### Exercice 3 : Réduction de dimensionnalité (ACP)

L'objectif de cet exercice est d'utiliser l'analyse des composants de voisinage pour la réduction de la dimensionnalité. La base de données contient des images de chiffres de 0 à 9 avec environ 180 échantillons de chaque classe. Chaque image est de dimension  $8 \times 8 = 64$ , et est réduite à un point de données bidimensionnel. L'analyse en composantes principales (ACP) appliquée à ces données identifie la combinaison d'attributs (composantes principales ou directions dans l'espace des caractéristiques) qui représente le plus de variance dans les données.

1. Charger la base de données des digits.
2. Séparer les digits en 70% données d'apprentissage et 30% données de test.
3. Appliquer l'ACP pour réduire les dimensions à deux en utilisant la méthode **PCA()** de **sklearn.decomposition**.
4. Créer un modèle KNN avec un voisinage  $K=3$ , ensuite ajuster le modèle sur l'ensemble d'apprentissage intégré (**pca.transform()**) et calculer le score du KNN.
5. Exécuter ce bout de code pour afficher la figure des points projetés de toute la base de données initiale :

```
import matplotlib.pyplot as plt
X_embedded = pca.transform(digits.data)
plt.figure()
plt.scatter(X_embedded[:, 0], X_embedded[:, 1], c=y, s=30, cmap="Set1")
plt.title("{} KNN (k={})\nTest accuracy = {:.2f}".format('PCA', n_neighbors, acc_knn))
plt.show()
```

### Exercice 1 : Regroupement hiérarchique

2. Charger la base de données d'iris du module datasets de sklearn et attribuer à une variable X les données d'iris :

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
```

3. Dans une boucle for, appliquer le regroupement hiérarchique en utilisant AgglomerativeClustering() pour différents types de liens : moyen, complet et simple. Dans la même boucle, afficher le dendrogramme de chaque lien :

```
from sklearn.cluster import AgglomerativeClustering
for linkage in ("average", "complete", "single"):
    clustering = AgglomerativeClustering(distance_threshold=0, linkage=linkage,
n_clusters=None)
    clustering.fit(X)
    plt.title("Dendrogramme du regroupement hiérarchique")
    plot_dendrogram(clustering, truncate_mode="level", p=3)
    plt.xlabel("Nombre de points dans : %s linkage" % linkage)
    plt.show()
```