

Correction Activité 7 : Réduction de dimension ACP

```

import matplotlib.pyplot as plt
from sklearn import decomposition
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
iris = datasets.load_iris()#Q1: Charger la base
X = iris.data # récupérer les valeurs des attributs pour toutes les observations
print(X.shape) # Q2 : afficher la dimension de l'ensemble de données (150 observations, 4 variables)
print(iris.feature_names) #Q3: afficher les noms des variables
# Q4: normalisation : Centrer et réduire les données
scaler=StandardScaler().fit(X)
X_CR=scaler.transform(X)
# Q5: Effectuer l'acp
pca = decomposition.PCA()
pca.fit(X_CR) #déterminer les nouveaux axes
decomposition.PCA()
ComposantesPrincipales = pca.transform(X_CR) #Q6: Projection des données sur les nouveaux axes
# Q7: affichage des inerties expliquées pour les différents nouveaux axes
print(pca.explained_variance_ratio_)
fig = plt.figure()
plt.plot(pca.explained_variance_ratio_','.')
plt.show()
#Conclusion : la 1ère et 2ème composantes représentent 0.73+0.23=0.96 =96% de la variance
# Deux axes sont suffisants pour représenter la variabilité des données.
# Q8: affichage des données après projection sur les TROIS premiers axes principaux
y = iris.target # récupérer les classes des observations==> utiles seulement pour l'affichage
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(ComposantesPrincipales[:, 0], ComposantesPrincipales[:, 1], ComposantesPrincipales[:, 2], c=y)
ax.set_xlabel('1ère CP') #1ère composante principale
ax.set_ylabel('2ème CP') #2ème composante principale
ax.set_zlabel('3ème CP') #3ème composante principale
plt.show()
# Q8: affichage des données après projection sur les DEUX premiers axes principaux
fig = plt.figure()
ax = fig.add_subplot(111)
ax.scatter(ComposantesPrincipales[:, 0], ComposantesPrincipales[:, 1], c=y)
ax.set_xlabel('1ère CP') #1ère composante principale
ax.set_ylabel('2ème CP') #2ème composante principale
plt.show()

```

#Q8: Conclusion : Les 3 classes sont parfaitement discernables dans un espace de représentation 2D.

ACP nous a permis de réduire le nombre de variables tout en concevant la variabilité des données