

# Homework 4

Tomas Andrade 103739  
Gonzalo Meneses 103401

17

	$y_1$	$y_2$	$y_3$
$x_1$	1	0,6	0,1
$x_2$	0	-0,4	0,8
$x_3$	0	0,2	0,5
$x_4$	1	0,4	-0,1

$\{y_1\} \perp\!\!\!\perp \{y_2; y_3\}$   
 $\downarrow$   
bermoulli       $\hookrightarrow$  Gaussian

$$\tilde{\pi}_1 = p(K=1) = 0,5 / \tilde{\pi}_2 = p(K=2) = 0,5$$

$$\rho_1 = p(y_1 = 1 | K=1) = 0,3$$

$$\rho_2 = p(y_1 = 1 | K=2) = 0,7$$

$$N_1 \left( \mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \Sigma_1 = \begin{pmatrix} 2 & 0,5 \\ 0,5 & 2 \end{pmatrix} \right)$$

$$N_2 \left( \mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \Sigma_2 = \begin{pmatrix} 1,5 & 1 \\ 1 & 1,5 \end{pmatrix} \right)$$

E-STEP:

$$x_1 = (1; 0,6; 0,1)$$

$$p(K=1 | x_1) = \frac{p(x_1 | K=1) \cdot p(K=1)}{p(x_1)} =$$

$$= \frac{p(x_1 | K=1) \cdot p(K=1)}{p(x_1 | K=1) \cdot p(K=1) + p(x_1 | K=2) \cdot p(K=2)}$$

$$= \frac{p(x_1 | K=1)}{p(x_1 | K=1) + p(x_1 | K=2)} =$$

$$= \frac{p(x_1 | K=1)}{p(x_1 | K=1) + p(x_1 | K=2)} =$$

①

$$= \frac{P(y_1 = 1 | K=1) \cdot P(y_2 = 0,6; y_3 = 0,1 | K=1)}{P_1}$$

$$P(y_1 = 1 | K=1) \cdot P(y_2 = 0,6; y_3 = 0,1 | K=1) + P(y_1 = 1 | K=2) \cdot P(y_2 = 0,6; y_3 = 0,1 | K=2)$$

$$= \frac{P_1 \cdot P(y_2 = 0,6; y_3 = 0,1 | K=1)}{P_1 \cdot P(y_2 = 0,6; y_3 = 0,1 | K=1) + P_2 \cdot P(y_2 = 0,6; y_3 = 0,1 | K=2)}$$

$$P_1 \cdot P(y_2 = 0,6; y_3 = 0,1 | K=1) + P_2 \cdot P(y_2 = 0,6; y_3 = 0,1 | K=2)$$

C. Aux:

$$P(y_2 = 0,6; y_3 = 0,1 | K=1) = \frac{1}{(2\pi)^{n/2}} \cdot \frac{1}{\sqrt{\det(\Sigma_1)}} \cdot e^{-\frac{1}{2} \cdot 0,421(3)}$$

(utilizando  
numpy)

$$= \frac{1}{2^{\frac{n}{2}}} \cdot \frac{1}{\sqrt{3,75}} \cdot e \approx 0,0665753$$

$$-\frac{1}{2} \cdot (x - \mu_2)^T \cdot \Sigma_2^{-1} \cdot (x - \mu_2)$$

$$P(y_2 = 0,6; y_3 = 0,1 | K=2) = \frac{1}{(2\pi)^{n/2}} \cdot \frac{1}{\sqrt{\det(\Sigma_2)}} \cdot e$$

$$= \frac{1}{2^{\frac{n}{2}}} \cdot \frac{1}{\sqrt{0,348}} \cdot e \approx 0,1196184$$

(utilizando  
numpy)

(2)

log., ficamos com:

$$P(K=1 | x_1) \approx \frac{0,3 \cdot 0,0665753}{0,3 \cdot 0,0665753 + 0,7 \cdot 0,1196184} \approx \\ \approx 0,19258959 = 19,258959\%$$

$$P(K=2 | x_1) = 1 - P(K=1 | x_1) \approx 0,80741041 = \\ = 80,741041\%$$

→ Aplicando o mesmo raciocínio para  $x_2 = (0; -0,4; 0,8)$   
ficamos com:

$$P(K=1 | x_2) = \frac{P(x_2 | K=1) \cdot P(K=1)}{P(x_2)} = \text{(utilizando numpy)}$$

$$= \frac{P(y_1=0 | K=1) \cdot P(y_2=-0,4; y_3=0,8 | K=1) \cdot P(K=1)}{P(y_1=0 | K=1) \cdot P(y_2=-0,4; y_3=0,8 | K=1) \cdot P(K=1) + P(y_1=0 | K=2) \cdot P(y_2=-0,4; y_3=0,8 | K=2) \cdot P(K=2)}$$
$$= \frac{1 - P_1 = 0,7}{0,7} \cdot \frac{\approx 0,050048888}{\approx 0,050048888} \cdot \frac{P(K=1)}{1 - P_2 = 0,3} \approx \frac{0,7 \cdot 0,050048888}{0,7 \cdot 0,050048888 + 0,3 \cdot 0,06819058} \approx 0,6819058$$

$$\Rightarrow \frac{0,7 \cdot 0,050048888}{0,7 \cdot 0,050048888 + 0,3 \cdot 0,06819058} \approx 63,1345116\%$$

$$P(K=2 | X_2) = 1 - P(K=1 | X_2) \approx 36,8654883\%.$$

→ Aplicando o mesmo raciocínio para  $X_3 = (0; 0,2; 0,5)$   
ficamos com:

$$P(K=1 | X_3) = \frac{P(X_3 | K=1) \cdot P(K=1)}{P(X_3)} = \text{(utilizando numpy)}$$

$$= \frac{P(y_1=0 | K=1) \cdot P(y_2=0,2; y_3=0,5 | K=1) \cdot P(K=1)}{P(y_1=0 | K=1) \cdot P(y_2=0,2; y_3=0,5 | K=1) + P(y_1=0 | K=2) \cdot P(y_2=0,2; y_3=0,5 | K=2)} =$$

$$= \frac{1 - P_1 = 0,7 \quad \approx 0,06837452}{1 - P_2 = 0,3 \quad \approx 0,12958103}$$

$$\approx 55,1811281\%.$$

$$P(K=2 | X_3) = 1 - P(K=1 | X_3) \approx 44,81887185\%.$$

→ Aplicando o mesmo raciocínio para  $X_4 = (1; 0,4; -0,1)$   
ficamos com:

$$\begin{aligned}
 p(K=1 | X_4) &= \frac{p(X_4 | K=1) \cdot p(K=1)}{p(X_4)} = && \text{(utilizando numpy)} \\
 &= \frac{0,3 \cdot p(y_1=1 | K=1) \cdot p(y_2=0,4 | K=1) \cdot p(y_3=-0,1 | K=1) \cdot p(K=1)}{p(y_1=1 | K=1) + p(y_1=2 | K=1) + p(y_1=3 | K=1)} = \\
 &\approx 0,059046993 && \\
 &\approx 0,059046993 && \\
 &\approx 0,124520089706 &&
 \end{aligned}$$

$\approx 16,892423385\%$ .

$$p(K=2 | X_4) = 1 - p(K=1 | X_4) \approx 83,1075766\%$$

Feito o E-STEP, podemos passar para o M-STEP onde precisamos de calcular os novos parâmetros das novas distribuições.

M-STEP:

$$\mu_j^1 = \frac{\sum_i p(K_j | x_i) \cdot x_i}{\sum_i p(K_j | x_i)}$$

utilizando os valores anteriormente calculados e utilizando a biblioteca numpy

$$\hookrightarrow \mu_1^1 = \begin{bmatrix} 0,026509 \\ 0,50712978 \end{bmatrix}; \mu_2^1 = \begin{bmatrix} 0,30914476 \\ 0,2104205 \end{bmatrix}$$

(5)

Por outro lado:

$$\sum_j = \frac{\sum_i P(K_j | X_i) (X_i - \mu_j) (X_i - \mu_j)^T}{\sum_i P(K_j | X_i)}$$

utilizando os valores anteriormente calculados e utilizando a biblioteca numpy

$$\sum_1 = \begin{bmatrix} 0,14136501 & -0,10540546 \\ -0,10540546 & 0,0960526 \end{bmatrix}$$

$$\sum_2 = \begin{bmatrix} 0,10829305 & -0,08865175 \\ -0,08865175 & 0,1041233 \end{bmatrix}$$

Para atualizar os priors: numpy

$$P(K=1) = \frac{\sum_i P(K=1 | X_i)}{\sum_j \sum_i P(K_j | X_i)} \approx 38,61675547\%$$

$$P(K=2) = \frac{\sum_i P(K=2 | X_i)}{\sum_j \sum_i P(K_j | X_i)} \approx 61,38324453\%$$

⑥

Para atualizar as Probabilidades obtidas através da distribuição de Bernoulli podemos utilizar a fórmula da média/centroide mas utilizando para a variável  $y_1$ :

$$P_1 = \frac{\sum_i P(K_1 | y_{1i}) \cdot y_{1i}}{\sum_i P(K_1 | y_{1i})} \approx 23,4039484\%$$

$$P_2 \approx 66,731817\%$$

2) Para calcular a qual cluster a observação  $x_{\text{new}} = \begin{pmatrix} 1 \\ 0,3 \\ 0,7 \end{pmatrix}$  realizaremos de novo o E-step mas c/ o  $x_{\text{new}}$  utilizando os parâmetros atualizados calculados em 1).

Ficamos então com:

$$\begin{aligned} P(K=1 | X_{\text{new}}) &= \frac{P(X_{\text{new}} | K=1) \cdot P(K=1)}{P(X_{\text{new}})} = \\ &\approx 0,234039484 \quad \approx 0,020755851 \quad \approx 0,3861675547 \\ &= P(y_1=1 | K=1) \cdot P(y_2=0,3; y_3=0,7 | K=1) - P(K=1) \\ &= P(y_1=1 | K=1) \cdot P(y_2=0,3; y_3=0,7 | K=1) + P(y_1=1 | K=2) \cdot P(y_2=0,3; y_3=0,7 | K=2) \\ &\approx 0,234039484 \quad \approx 0,3861675547 \quad \approx 0,66731817 \quad \approx 0,6138324453 \\ &\approx 0,020755851 \quad \approx 0,06843082147 \end{aligned}$$

$$\approx 8,0289604\%$$

$$P(K=2 | X_{\text{new}}) = 1 - P(K_1 | X_{\text{new}}) = 91,9710396\%$$

∴ Logo,  $X_{new}$  tem 8,0289604% de probabilidade de pertencer ao cluster  $k=1$  e 91,9710396% de probabilidade de pertencer ao cluster  $k=2$ , concluindo então que é mais provável  $X_{new}$  pertencer a  $k=2$ .

3) A silhueta de uma observação pode ser descrita por:

$$S(x) = 1 - \frac{a(x)}{b(x)}$$

→ distância  
 ao seu  
 cluster  
 → distância  
 aos outros clusters

Se  $a(x) > b(x)$ :  $S(x) = \frac{b(x)}{a(x)} - 1$

$$\text{ML} \rightarrow P(k=j | x_i) \sim P(x_i | k=j) \cdot P(K=j)$$

Relembrando que temos o data set:

	$y_1$	$y_2$	$y_3$	cluster k
$x_1$	1	0,6	0,1	$k_2$
$x_2$	0	-0,4	0,8	$k_1$
$x_3$	0	0,2	0,5	$k_1$
$x_4$	1	0,4	-0,1	$k_2$

← temos que definir segundo  
ML em qual cluster se insere  
cada observação

$$P(k=1) \approx 0,234039484 \quad \approx 0,98903972$$

$$P(k=1) \approx 0,234039484 \quad \approx 0,98903972$$

$$P(k=1) \approx 0,234039484 \quad \approx 0,98903972$$

$$P(k=2 | x_1) \sim P(x_1 | k=2) \cdot P(k=2) = P(y_1=1 | k=2) \cdot P(y_2=0,6; y_3=0,1 | k=2) \approx 0,234039484 \quad \approx 0,98903972$$

$$\approx 0,234039484 \quad \approx 0,98903972$$

Como  $P(k_1 | x_1) < P(k_2 | x_1) \Rightarrow x_1 \in k_2$

ML para  $X_2$ :

$$P(K=1 | X_2) \sim P(X_2 | K=1) \cdot P(K=1) = P(y_1=0 | K=1) \cdot P(y_2=0.4; y_3=0.8 | K=1) \approx 1,2663$$

$\approx 1,65326075$

$$P(K=2 | X_2) \sim P(X_2 | K=2) \cdot P(K=2) = P(y_1=0 | K=2) \cdot P(y_2=0.4; y_3=0.8 | K=2) \approx 0,0887$$

$\approx 0,2667315436$

Como  $P(K_2 | X_2) < P(K_1 | X_2) \Rightarrow X_2 \in K_1$

ML para  $X_3$ :

$$P(K=1 | X_3) \sim P(X_3 | K=1) \cdot P(K=1) = P(y_1=0 | K=1) \cdot P(y_2=0.9; y_3=0.5 | K=1) \approx 1,4391$$

$\approx 1,877525603$

$$P(K=2 | X_3) \sim P(X_3 | K=2) \cdot P(K=2) = P(y_1=0 | K=2) \cdot P(y_2=0.1; y_3=0.5 | K=2) \approx 0,4542$$

$\approx 1,365191669$

Como  $P(K_2 | X_3) < P(K_1 | X_3) \Rightarrow X_3 \in K_1$

ML para  $X_4$ :

$$P(K=1 | X_4) \sim P(X_4 | K=1) \cdot P(K=1) = P(y_1 = 1 | K=1) \cdot P(y_2 = 0,4; y_3 = -0,1 | K=1) \approx$$

$\approx 0,08872530$

$$\approx 0,0208$$

$$P(K=2 | X_4) \sim P(X_4 | K=2) \cdot P(K=2) = P(y_1 = 1 | K=2) \cdot P(y_2 = 0,4; y_3 = -0,1 | K=2) \approx$$

$\approx 1,08390867$

$$\approx 0,7233$$

Como  $P(K_2 | X_4) > P(K_1 | X_4) \Rightarrow X_4 \in K_2$

Calculando agora a silhueta:

→ A silhueta pode ser calculada ainda da seguinte forma:

$$S(x_j) = 1 - \frac{\frac{1}{|C_i|-1} \cdot \sum_{k \neq j}^{|C_i|} \|x_j - x_k\|_1}{\min_e \left( \frac{1}{|C_e|} \cdot \sum_k^{|C_e|} \|x_j - x_k\|_1 \right)}$$

$\left. \begin{array}{l} a(x_j) \\ b(x_j) \end{array} \right.$

## Distâncias

	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	0	2,7	1,8	0,4
$x_2$	2,7	0	0,9	2,7
$x_3$	1,8	0,9	0	1,8
$x_4$	0,4	2,7	1,8	0

$$d_{12} = d_{21} = |1-0| + |0,6+0,4| + |0,1-0,8| = 2,7$$

$$d_{13} = d_{31} = |1+0,4| + |0,4| = 1,8$$

$$d_{14} = d_{41} = |0+0,2| + |0,2| = 0,4$$

$$d_{32} = d_{23} = |0+0,6| + |0,3| = 0,9$$

$$d_{24} = d_{42} = |1+0,8| + |0,9| = 2,7$$

$$d_{34} = d_{43} = |1+0,2| + |0,6| = 1,8$$

$$\alpha(x_1) = \frac{1}{2-1} d_{14} = 0,4 = \alpha(x_4)$$

$$\alpha(x_2) = d_{23} = 0,9 = \alpha(x_3)$$

$$b(x_1) = \frac{1}{2} (d_{12} + d_{13}) = \frac{4,5}{2} = 2,25$$

$$b(x_2) = \frac{d_{21} + d_{24}}{2} = 2,7$$

$$b(x_3) = \frac{d_{31} + d_{34}}{2} = 1,8$$

$$b(x_4) = \frac{d_{42} + d_{43}}{2} = \frac{4,5}{2} = 2,25$$

$$S(x_1) = 1 - \frac{0,4}{2,25} = 0,8(2)$$

$$S(x_2) = 1 - \frac{0,9}{2,25} = 0,6$$

$$S(x_3) = 1 - \frac{0,9}{1,8} = 0,5$$

$$S(x_4) = 1 - \frac{0,4}{2,25} = 0,8(2)$$

$$S(C_1) = \frac{S(x_1) + S(x_4)}{2} = 0,8(2)$$

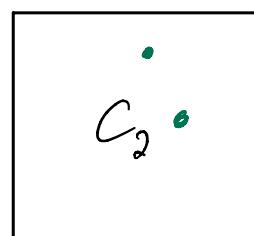
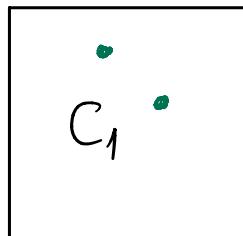
$$S(C_2) = \frac{S(x_2) + S(x_3)}{2} = 0,58(3)$$

4) purity = 0,75 =  $\frac{1}{N} \sum_{K=1}^2 \max_j (|C_K \cap l_j|)$

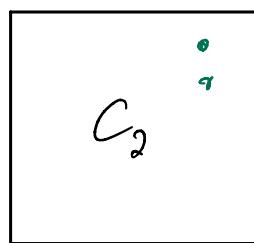
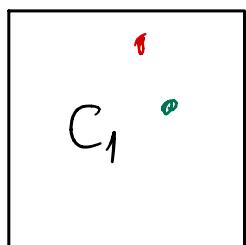
$$\Rightarrow \max(|C_1 \cap l_j|) + \max(|C_2 \cap l_j|) = 3$$

fara:

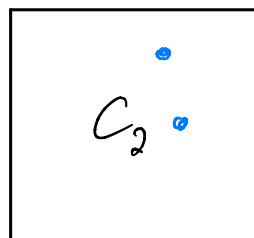
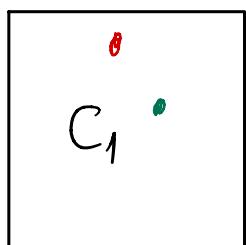
$$j=1 \rightarrow \max(2) + \max(2) = 4 \quad (=) \quad \text{purity} = 1 \quad \text{Impossible}$$



$$j=2 \rightarrow \text{p-ex: } \max(1,1) + \max(2,0) = 3 \quad \text{Possível}$$



$$j=3 \rightarrow \text{p-ex: } \max(1,0,1) + \max(0,1,0) = 3 \quad \text{Possível}$$



$j=4 \rightarrow \text{Impossível}$ , uma vez que o número de "labels" é igual aos de observações, ou seja, a purity é obrigatoriamente  $1/4$ .

$j > 5 \rightarrow \text{Impossível}$ , pois teríamos mais "labels" que observações.

Logo, o nº de classes possíveis neste caso para uma purity de 0,75 é 2 ou 3 classes.

# G11\_notebook

October 30, 2023

## 1 Homework 4

Gonçalo Meneses, 103401, e Tomás Arêde, 103239.

Antes de começar, vamos remover os avisos que surgem ao longo deste notebook.

```
[ ]: import warnings  
warnings.filterwarnings('ignore')
```

Começamos por carregar o nosso ficheiro *arff* e criar o data frame que irá armazenar os nossos dados.

```
[ ]: from scipy.io import arff  
import pandas as pd  
import numpy as np  
  
data, col_names = arff.loadarff('column_diagnosis.arff')  
df = pd.DataFrame(data)  
df['class'] = df['class'].str.decode('utf-8')  
df.columns = col_names.names()
```

De seguida, procedemos à normalização das features do nosso dataset.

```
[ ]: from sklearn.preprocessing import MinMaxScaler  
  
X = df.drop('class', axis=1)  
y = df['class']  
  
scaler = MinMaxScaler()  
X_norm = scaler.fit_transform(X)  
  
df_norm = pd.DataFrame(X_norm, columns=X.columns)  
df_norm = pd.concat([df_norm, y], axis=1)  
df_norm.head()
```

```
[ ]:   pelvic_incidence  pelvic_tilt  lumbar_lordosis_angle  sacral_slope  \n0      0.355688      0.519900          0.229180      0.250857\n1      0.124501      0.296783          0.098578      0.144629\n2      0.411666      0.513932          0.322995      0.307661
```

3	0.416151	0.557414	0.271260	0.289436
4	0.227272	0.289479	0.128129	0.247022

	pelvic_radius	degree_spondylolisthesis	class	
0	0.307461	0.025148	Hernia	
1	0.476649	0.036365	Hernia	
2	0.386097	0.017523	Hernia	
3	0.341826	0.051838	Hernia	
4	0.409579	0.044173	Hernia	

### 1.0.1 Exercício 1

```
[ ]: from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics.cluster import contingency_matrix
import matplotlib.pyplot as plt
%matplotlib inline
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('svg', 'pdf')

def purity_score(y_true, y_pred):
    confusion_matrix = contingency_matrix(y_true, y_pred)
    return np.sum(np.amax(confusion_matrix, axis=0)) / np.sum(confusion_matrix)

clustered_data = df_norm.copy()

silhouette_scores = []
purity_scores = []

k_values = range(2, 6)

for k in k_values:
    k_means = KMeans(n_clusters=k, random_state=0)
    k_means.fit(X_norm)

    cluster_labels = k_means.labels_
    clustered_data[f'Cluster_{k}'] = cluster_labels

    silhouette = silhouette_score(X_norm, cluster_labels)
    silhouette_scores.append(silhouette)

    purity = purity_score(y, cluster_labels)
    purity_scores.append(purity)

print("Purity and Silhouette scores for different values of k:")
for k, purity, silhouette in zip(k_values, purity_scores, silhouette_scores):
    print(f"k={k}: Purity={purity:.5f}, Silhouette={silhouette:.5f}")
```

```

plt.figure(figsize=(10, 6))
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score', color='tab:blue')
plt.plot(k_values, silhouette_scores, marker='o', color='tab:blue', label='Silhouette Score')
plt.xticks(k_values)

plt.twinx()
plt.ylabel('Purity', color='tab:red')
plt.plot(k_values, purity_scores, marker='s', color='tab:red', label='Purity')

plt.title('Silhouette and Purity Scores vs. Number of Clusters (k)')
plt.show()

```

Purity and Silhouette scores for different values of k:

k=2: Purity=0.63226, Silhouette=0.36044

k=3: Purity=0.66774, Silhouette=0.29579

k=4: Purity=0.66129, Silhouette=0.27442

k=5: Purity=0.67742, Silhouette=0.23824



### 1.0.2 Exercício 2

Exercício 2 (a)

```
[ ]: from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_norm)

explained_variance = pca.explained_variance_ratio_
print("Explained variabilities for the top two principal components:", ↴
      explained_variance)

total_explained_variance = np.sum(explained_variance)
print(f"Total explained variability by the top two principal components: ↴
      {total_explained_variance * 100:.5f}%)
```

Explained variabilities for the top two principal components: [0.56181445  
0.20955953]

Total explained variability by the top two principal components: 77.13740%

### Exercício 2 (b)

```
[ ]: def sort_by_weight(weight, features):
    return sorted(zip(abs(weight), features), reverse=True)

component_1_loadings = pca.components_[0]
component_2_loadings = pca.components_[1]

print("Sorted input variables by absolute weight for the first principal component:")
for weight, variable in sort_by_weight(component_1_loadings, X.columns):
    print(f"{variable}: {weight:.5f}")

print("\nSorted input variables by absolute weight for the second principal component:")
for weight, variable in sort_by_weight(component_2_loadings, X.columns):
    print(f"{variable}: {weight:.5f})
```

Sorted input variables by absolute weight for the first principal component:  
pelvic\_incidence: 0.59162  
lumbar\_lordosis\_angle: 0.51508  
pelvic\_tilt: 0.46704  
sacral\_slope: 0.32569  
degree\_spondylolisthesis: 0.21693  
pelvic\_radius: 0.11582

Sorted input variables by absolute weight for the second principal component:  
pelvic\_tilt: 0.67037  
pelvic\_radius: 0.58107  
sacral\_slope: 0.44330  
pelvic\_incidence: 0.10004  
lumbar\_lordosis\_angle: 0.08005

```
degree_spondylolisthesis: 0.00458
```

### 1.0.3 Exercício 3

```
[ ]: from sklearn.preprocessing import LabelEncoder
from matplotlib.colors import ListedColormap

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

cmap = ListedColormap(['#1f77b4', '#ff7f0e', '#2ca02c'])

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

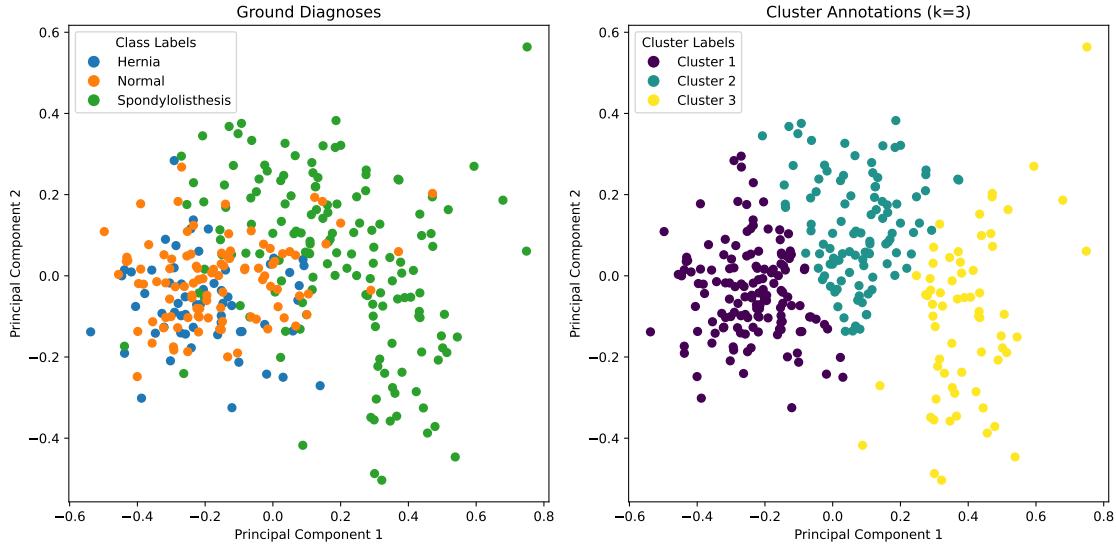
scatter1 = ax1.scatter(X_pca[:, 0], X_pca[:, 1], c=y_encoded, cmap=cmap)
ax1.set_title('Ground Diagnoses')
ax1.set_xlabel('Principal Component 1')
ax1.set_ylabel('Principal Component 2')

scatter2 = ax2.scatter(X_pca[:, 0], X_pca[:, 1], c=clustered_data['Cluster_3'],
                      cmap='viridis')
ax2.set_title('Cluster Annotations (k=3)')
ax2.set_xlabel('Principal Component 1')
ax2.set_ylabel('Principal Component 2')

labels = label_encoder.classes_
handles1 = [plt.Line2D([0], [0], marker='o', color='w',
                     markerfacecolor=cmap(i), markersize=10, label=label) for i, label in
            enumerate(labels)]
ax1.legend(handles=handles1, title='Class Labels')

cluster_colors = ['#440154', '#21908C', '#FDE724']
custom_cmap = ListedColormap(cluster_colors)
handles2 = [plt.Line2D([0], [0], marker='o', color='w',
                     markerfacecolor=custom_cmap(i), markersize=10, label=f'Cluster {i + 1}') for
            i in range(3)]
ax2.legend(handles=handles2, title='Cluster Labels')

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



#### 1.0.4 Exercício 4

A técnica de aprendizagem por *clustering* pode ser útil no diagnóstico de um indivíduo de duas formas distintas: criação de perfis de risco para avaliação da saúde e integração de informações clínicas adicionais.

No primeiro caso, mesmo quando observamos que dois grupos distintos de indivíduos podem ser agrupados no mesmo *cluster*, ainda é possível obter informações valiosas por meio do *clustering*. Por exemplo, se um novo indivíduo for atribuído ao segundo ou terceiro *cluster*, é provável que ele esteja em risco de desenvolver *Spondylolisthesis*. Porém, caso seja atribuído ao primeiro *cluster* podemos inferir que a pessoa tem uma baixa probabilidade de ser diagnosticada com *Spondylolisthesis* e, portanto, pode ser *Hernia* ou *Normal*.

Relativamente ao nosso segundo ponto, um profissional de saúde poderá decidir o diagnóstico através da combinação deste resultado do *clustering* com outras informações de diagnóstico ou características do indivíduo, como histórico médico, resultados de exames ou sintomas específicos. Assim, os prestadores de saúde poderão, então, fazer uma avaliação precisa do paciente, diagnosticando-o, finalmente, como *Hernia* ou *Normal*.

Em resumo, o *clustering* pode ser uma ferramenta valiosa para ajudar a avaliar se um indivíduo está doente ou não, fornecendo uma avaliação preliminar com base na associação a grupos de perfil de saúde semelhante, permitindo o aperfeiçoamento dessa avaliação com informações clínicas específicas do paciente.