HW 4

28 de outubro de 2023 16:59

Observações: $(1, \underset{x_1}{0.6}, 0.1), (0, \underset{x_2}{-0.4}, 0.8), (0, \underset{x_3}{0.2}, 0.5), (1, \underset{x_4}{0.4}, -0.1)$ $\quad \{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$ (independentes)

Parâmetros iniciais: $\pi_1 = 0.5, \pi_2 = 0.5$; $p_1 = P(y_1 = 1) = 0.3$ (cluster 1), $p_2 = P(y_1 = 1) = 0.7$ (cluster 2) Logo, $P(y_1 = 0) = 0.7$ p/ cluster 1 e $P(y_1 = 0) = 0.3$ p/ cluster 2

$N_1(\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 2 \end{pmatrix})$, $N_2(\mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1.5 & 1 \\ 1 & 1.5 \end{pmatrix})$

(N_1: c_1; N_2: c_2)

1. • Expectations (E-step)

$\gamma_{ki} = P(c_k | x_i) = P(x_i | c_k) = P(y_1 | c_k) P(y_2, y_3 | c_k) P(c_k) = P(y_1 | c_k) N(y_2, y_3 | \mu_k, \Sigma_k) \pi_k$ (para cada cluster, $k \in \{1, 2\}$)

Utiliza-se o Python para este tipo de cálculos

x_1 c_1: $P(c=1) = \pi_1 = 0.5$ $\quad P(y_1 = 1 | c = 1) = p_1 = 0.3$ $\quad P(y_2 = 0.6, y_3 = 0.1 | c = 1) = 0.06658$

$P(c=1 | x_1) = P(y_1 = 1 | c=1) P(y_2 = 0.6, y_3 = 0.1 | c=1) P(c=1) = 0.5 \times 0.06658 \times 0.3 = 0.009986$

normalizado $\rightarrow \frac{0.009986}{0.009986 + 0.041866} = 0.192587$

c_2: $P(c=2) = \pi_2 = 0.5$ $\quad P(y_1 = 1 | c = 2) = p_2 = 0.7$ $\quad P(y_2 = 0.6, y_3 = 0.1 | c = 2) = 0.119618$

$P(c=2 | x_1) = P(y_1 = 1 | c=2) P(y_2 = 0.6, y_3 = 0.1 | c=2) P(c=2) = 0.7 \times 0.119618 \times 0.5 = 0.041866$

normalizado $\rightarrow \frac{0.041866}{0.009986 + 0.041866} = 0.809380$

x_2 c_1: $P(c=1) = \pi_1 = 0.5$ $\quad P(y_1 = 0 | c = 1) = p_1 = 0.7$ $\quad P(y_2 = -0.4, y_3 = 0.8 | c = 1) = 0.050049$

$P(c=1 | x_2) = P(y_1 = 0 | c=1) P(y_2 = -0.4, y_3 = 0.8 | c=1) P(c=1) = 0.7 \times 0.050049 \times 0.5 = 0.017517$

normalizado $\rightarrow \frac{0.017517}{0.017517 + 0.010229} = 0.631334$

c_2: $P(c=2) = \pi_2 = 0.5$ $\quad P(y_1 = 0 | c = 2) = p_2 = 0.3$ $\quad P(y_2 = -0.4, y_3 = 0.8 | c = 2) = 0.068191$

$P(c=2 | x_2) = P(y_1 = 0 | c=2) P(y_2 = -0.4, y_3 = 0.8 | c=2) P(c=2) = 0.3 \times 0.068191 \times 0.5 = 0.010229$

normalizado $\rightarrow \frac{0.010229}{0.017517 + 0.010229} = 0.368666$

x_3 c_1: $P(c=1) = \pi_1 = 0.5$ $\quad P(y_1 = 0 | c = 1) = p_1 = 0.7$ $\quad P(y_2 = 0.2, y_3 = 0.5 | c = 1) = 0.068374$

$P(c=1 | x_3) = P(y_1 = 0 | c=1) P(y_2 = 0.2, y_3 = 0.5 | c=1) P(c=1) = 0.7 \times 0.06658 \times 0.5 = 0.023931$

normalizado $\rightarrow \frac{0.023931}{0.023931 + 0.019437} = 0.551812$

c_2: $P(c=2) = \pi_2 = 0.5$ $\quad P(y_1 = 0 | c = 2) = p_2 = 0.3$ $\quad P(y_2 = 0.2, y_3 = 0.5 | c = 2) = 0.129581$

$P(c=2 | x_3) = P(y_1 = 0 | c=2) P(y_2 = 0.2, y_3 = 0.5 | c=2) P(c=2) = 0.3 \times 0.119618 \times 0.5 = 0.019437$

normalizado $\rightarrow \frac{0.019437}{0.023931 + 0.019437} = 0.448188$

x_4 c_1: $P(c=1) = \pi_1 = 0.5$ $\quad P(y_1=1 \mid c=1) = p_1 = 0.3$ $\quad P(y_2=0.4, y_3=-0.1 \mid c=1) = 0.059047$

$P(c=1 \mid x_4) = P(y_1=1 \mid c=1) P(y_2=0.4, y_3=-0.1 \mid c=1) P(c=1) = 0.3 \times 0.059047 \times 0.5 = 0.008857$

normalizada $\rightarrow \dfrac{0.008857}{0.008857 + 0.043575} = 0.168924$

c_2: $P(c=2) = \pi_2 = 0.5$ $\quad P(y_1=1 \mid c=2) = p_2 = 0.7$ $\quad P(y_2=0.4, y_3=-0.1 \mid c=2) = 0.124500$

$P(c=2 \mid x_4) = P(y_1=1 \mid c=2) P(y_2=0.4, y_3=-0.1 \mid c=2) P(c=2) = 0.7 \times 0.124500 \times 0.5 = 0.043575$

normalizada $\rightarrow \dfrac{0.043575}{0.008857 + 0.043575} = 0.831076$

- Maximization (M-Step)

$N_k = \sum \gamma_{ki} \rightarrow N_1 = 0.192587 + 0.631334 + 0.551812 + 0.168924 = 1.544657$

$N_2 = 0.809380 + 0.368666 + 0.448188 + 0.831076 = 2.457310$

- Atualizar priors

$\pi_k = \dfrac{N_k}{N} \rightarrow \pi_1 = \dfrac{1.542690}{1.544657 + 2.457310} = 0.385934 \qquad \pi_2 = \dfrac{2.457310}{1.544657 + 2.457310} = 0.614026$

- Atualização de $p(y_1 \mid c_k)$ — Bernoulli

$p(y_1 \mid c_k) = \dfrac{1}{N_k} \sum \gamma_{ki} \cdot x_i \rightarrow p_1 = \dfrac{0.190620 \times 1 + 0 + 0 + 1 \times 0.168924}{1.542690} = 0.234040$

$p_2 = \dfrac{0.809380 \times 1 + 0 + 0 + 0.831076}{2.457310} = 0.667582$

- Atualização de $p(y_2, y_3 \mid c_k)$ — Normal

⇒ Médias: $\mu_k = \dfrac{1}{N_k} \sum p(c_k \mid x_i) \begin{bmatrix} y_{2i} \\ y_{3i} \end{bmatrix}$

$$\mu_1 = \frac{1}{1.544657}\left(0.192587\begin{bmatrix}0.6\\0.1\end{bmatrix} + 0.631334\begin{bmatrix}-0.4\\0.8\end{bmatrix} + 0.551812\begin{bmatrix}0.2\\0.5\end{bmatrix} + 0.168924\begin{bmatrix}0.4\\-0.1\end{bmatrix}\right) = \begin{bmatrix}0.026511\\0.507128\end{bmatrix}$$

$$\mu_2 = \frac{1}{2.457310}\left(0.809380\begin{bmatrix}0.6\\0.1\end{bmatrix} + 0.368666\begin{bmatrix}-0.4\\0.8\end{bmatrix} + 0.448188\begin{bmatrix}0.2\\0.5\end{bmatrix} + 0.831076\begin{bmatrix}0.4\\-0.1\end{bmatrix}\right) = \begin{bmatrix}0.309375\\0.210335\end{bmatrix}$$

⇒ Matrizes de covariância: $\Sigma_k = \dfrac{1}{N_k} \sum_{i=1}^{q} p(c_k \mid x_i)(x_i - \mu_k)(x_i - \mu_k)^T$

$$\Sigma_1 = \frac{1}{1.544657}\Bigg(0.192587 \left(\begin{bmatrix}0.6\\0.1\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)\left(\begin{bmatrix}0.6\\0.1\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)^T \quad i=1$$

$$+ 0.631334 \left(\begin{bmatrix}-0.4\\0.8\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)^T \quad i=2$$

$$+ 0.551812 \left(\begin{bmatrix}0.2\\0.5\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)\left(\begin{bmatrix}0.2\\0.5\end{bmatrix} - \begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)^T \quad i=3$$

$$+ 0.168824\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.026511\\0.507128\end{bmatrix}\right)^T\Bigg) \quad i=4$$

$$= \frac{1}{1.549657}\left(\begin{bmatrix}0.063340 & -0.044966\\-0.044966 & 0.031922\end{bmatrix}+\begin{bmatrix}0.111841 & -0.078862\\-0.078862 & 0.054152\end{bmatrix}+\begin{bmatrix}0.016603 & -6.8200\times10^{-4}\\-6.8200\times10^{-4} & 2.8000\times10^{-5}\end{bmatrix}+\begin{bmatrix}0.023564 & -0.038304\\-0.038304 & 0.062266\end{bmatrix}\right)$$

$$= \begin{bmatrix}0.111365 & -0.105405\\-0.105405 & 0.104051\end{bmatrix}$$

$$\Sigma_2 = \frac{1}{2.457310}\Bigg(0.809380\left(\begin{bmatrix}0.6\\0.1\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)\left(\begin{bmatrix}0.6\\0.1\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)^T \quad i=1$$

$$+ 0.368666\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)\left(\begin{bmatrix}-0.4\\0.8\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)^T \quad i=2$$

$$+ 0.448188\left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)\left(\begin{bmatrix}0.2\\0.5\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)^T \quad i=3$$

$$+ 0.831076\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)\left(\begin{bmatrix}0.4\\-0.1\end{bmatrix}-\begin{bmatrix}0.309375\\0.210335\end{bmatrix}\right)^T\Bigg) \quad i=4$$

$$= \frac{1}{2.457310}\left(\begin{bmatrix}0.068363 & -0.025954\\-0.025954 & 0.009853\end{bmatrix}+\begin{bmatrix}0.185511 & -0.154211\\-0.154211 & 0.128187\end{bmatrix}+\begin{bmatrix}0.005362 & -0.014200\\-0.014200 & 0.037606\end{bmatrix}+\begin{bmatrix}0.006826 & -0.023373\\-0.023373 & 0.080039\end{bmatrix}\right)$$

$$= \begin{bmatrix}0.108276 & -0.088608\\-0.088608 & 0.104051\end{bmatrix}$$

Terminado o E-step e o M-step, concluímos uma epoch do algoritmo EM.

(2.) $x_{new} = \begin{pmatrix}1\\0.3\\0.7\end{pmatrix}$ Fazendo o E-step e M-step para a nova observação:

• E-step (segundo os novos parâmetros calculados acima)

c_1 $P(c=1) = \pi_1 = 0.385974$ $\quad P(y_1=1 \mid c=1) = 0.234040$ $\quad P(y_2=0.3, y_3=0.7 \mid c=1) = 0.027082$

$P(c=1 \mid x_{new}) = P(y_1=1 \mid c=1)\,P(y_2=0.3, y_3=0.7 \mid c=1)\,P(c=1) = 0.234040 \times 0.027082 \times 0.385974 = \underline{0.002446}$

normalizada $\rightarrow \frac{0.002446}{0.002446 + 0.028059} = \boxed{0.080196}$ (posterior for c_1)

c_2 $P(c=2) = \pi_2 = 0.614026$ $\quad P(y_1=1 \mid c=2) = p_2 = 0.667582$ $\quad P(y_2=0.3, y_3=0.7 \mid c=2) = 0.068451$

$P(c=2 \mid x_{new}) = P(y_1=1 \mid c=2)\,P(y_2=0.3, y_3=0.7 \mid c=2)\,P(c=2) = 0.667582 \times 0.068451 \times 0.614026 = \underline{0.028059}$

normalizada $\rightarrow \frac{0.028059}{0.002446 + 0.028059} = \boxed{0.919804}$ (posterior for c_2)

(3.) Calculam-se as likelihoods para cada variável utilizando a fórmula:

$P(c_i \mid x_j) = P(y_1 \mid c_i)\,P(y_2, y_3 \mid c_i)$ (ignoram-se os priors por causa da ML assumption)

x_1 $P(c=1 \mid x_1) = P(y_1=1 \mid c_1)\,P(y_2=0.6, y_3=0.1 \mid c_1) = 0.234040 \times 0.989026 = 0.231411$

$P(C=2|x_1) = P(y_1=1|C_2)\,P(y_2=0.6, y_3=0.1|C_2) = 0.667582 \times 1.415013 = 0.951313$

Normalizados: $P(C_1|x_1) = 0.195700$, $P(C_2|x_1) = \boxed{0.804300}$ x_1 atribuído a C_2

x_2 $P(C_1|x_2) = (1-0.234040) \times 1.653214 = 1.266296$ $P(C_2|x_2) = (1-0.667582) \times 0.266301 = 0.088523$

Normalizados: $P(C_1|x_2) = \boxed{0.934661}$, $P(C_2|x_2) = 0.065339$ x_2 atribuído a C_1

x_3 $P(C_1|x_3) = (1-0.234040) \times 1.872538 = 1.439120$ $P(C_2|x_3) = (1-0.667582) \times 1.365523 = 0.453924$

Normalizados: $P(C_1|x_3) = \boxed{0.760088}$, $P(C_2|x_3) = 0.239912$ x_3 atribuído a C_1

x_4 $P(C_1|x_4) = 0.234040 \times 0.088134 = 0.020167$ $P(C_2|x_4) = 0.667582 \times 1.082998 = 0.722980$

Normalizados: $P(C_1|x_4) = 0.027922$, $P(C_2|x_4) = \boxed{0.972078}$ x_4 atribuído a C_2

Atribuindo cada observação do cluster que apresenta maior probabilidade de pertencer:

$C_1 = \{x_2, x_3\}$, $C_2 = \{x_1, x_4\}$

→ distância de Manhattan, segundo o enunciado

- Para C_1: $a(x_2) = d(x_2, x_3) = |0-0| + |-0.4-0.2| + |0.8-0.5| = 0.9 = a(x_3)$

$d(x_2, x_1) = |1-0| + |0.6+0.4| + |0.8-0.1| = 2.7$
$d(x_2, x_4) = |0-1| + |0.4+0.4| + |0.8+0.1| = 2.7$
$\Big\} b(x_2) = \frac{2.7+2.7}{2} = 2.7$

$d(x_3, x_1) = |0-1| + |0.2-0.6| + |0.5-0.1| = 1.8$
$d(x_3, x_4) = |0-1| + |0.4-0.2| + |0.5+0.1| = 1.8$
$\Big\} b(x_3) = \frac{1.8+1.8}{2} = 1.8$

x_2: $S(x_2) = 1 - \frac{a(x_2)}{b(x_2)} = 1 - \frac{0.9}{2.7} = 2/3$ x_3: $S(x_3) = 1 - \frac{a(x_3)}{b(x_3)} = 1 - \frac{0.9}{1.8} = 0.5$

$S(C_1) = \frac{S(x_2)+S(x_3)}{2} = \frac{2/3+0.5}{2} = \boxed{0.5833}$

- Para C_2: $a(x_1) = d(x_1, x_4) = |1-1| + |0.6-0.4| + |0.1+0.1| = 0.4 = a(x_4)$

$d(x_1, x_2) = 2.7$
$d(x_1, x_3) = 1.8$
$\Big\} b(x_1) = \frac{2.7+1.8}{2} = 2.25$

$d(x_4, x_2) = 2.7$
$d(x_4, x_3) = 1.8$
$\Big\} b(x_4) = \frac{2.7+1.8}{2} = 2.25$

x_1: $S(x_1) = 1 - \frac{a(x_1)}{b(x_1)} = 1 - \frac{0.4}{2.25} = 0.8(2)$ x_4: $S(x_4) = 1 - \frac{a(x_4)}{b(x_4)} = 1 - \frac{0.4}{2.25} = 0.8(2)$

$S(C_2) = \frac{S(x_1)+S(x_4)}{2} = \boxed{0.8222}$

(4.) A purity de uma clustering solution é dada por $\frac{1}{n}\sum_{i=1}^{K} \max_j (|C_i \cap L_j|)$, em que L são as classes de referência e C são os clusters.

Como nos é dada a purity da solução de clustering (0.75) e se pede para identificar o nº de classes possíveis, o processo de raciocínio será o oposto (determinar classes para as quais a fórmula dê 0.75). Os clusters determinados anteriormente são: $C_1 = \{x_2, x_3\}$ e $C_2 = \{x_1, x_4\}$. Portanto:

• 1 classe

Para uma classe, só temos uma hipótese: $class1 = \{x_1, x_2, x_3, x_4\}$.

$Purity = \frac{1}{4} \max(c_1 \cap class1, c_2 \cap class1) = \frac{1}{4} \times 4 = 1$ ✗ > 0.75 de purity

• 2 classes

É possível, com duas classes, que a solução tenha 0.75 de purity.

$class1 = \{x_1, x_2, x_3\}$, $class2 = \{x_4\}$

$Purity = \frac{1}{4}(\max(c_1 \cap class1, c_1 \cap class2) + \max(c_2 \cap class1, c_2 \cap class2))$

$= \frac{1}{4}(2+1) = 0.75$ ✓

Para $class1 = \{x_2, x_3, x_4\}$ e $class2 = \{x_1\}$, a purity é: $\frac{1}{4}(2+1) = 0.75$ também, por exemplo.

Portanto, 2 classes é possível.

• 3 classes

$class1 = \{x_2, x_3\}$, $class2 = \{x_1\}$, $class3 = \{x_4\}$ (por ex.)

$Purity = \frac{1}{4}(\max(c_1 \cap \underset{2}{class1}, c_1 \cap \underset{0}{class2}, c_1 \cap \underset{0}{class3}) + \max(c_2 \cap \underset{0}{class1}, c_2 \cap \underset{1}{class2}, c_2 \cap \underset{1}{class3}))$

$= \frac{1}{4}(2+1) = 0.75$

Para: $class1 = \{x_2, x_4\}$, ou $class2 = \{x_3\}$, $class3 = \{x_2\}$ | $class2 = \{x_2\}$, $class3 = \{x_3\}$, temos:

$Purity = \frac{1}{4}(\max(c_1 \cap \underset{0}{class1}, c_1 \cap \underset{1}{class2}, c_1 \cap \underset{1}{class3}) + \max(c_2 \cap \underset{1}{class1}, c_2 \cap class2, c_2 \cap class3))$

$= \frac{1}{4}(1+2) = 0.75$

Portanto, 3 classes é possível.

• 4 classes

$class1 = \{x_1\}$, $class2 = \{x_2\}$, $class3 = \{x_3\}$, $class4 = \{x_4\}$.

$Purity = \frac{1}{4}(1+1) = 0.5$ ✗ Qualquer combinação com 4 classes não funcionará.

Não faz sentido considerar 5 classes ou superior.

Portanto, o número possível de classes é 2 ou 3.

Recall the column_diagnosis.arff dataset from previous homeworks. For the following exercises, normalize the data using sklearn's MinMaxScaler

```
import numpy as np
import pandas as pd
from scipy.io.arff import loadarff
from sklearn.preprocessing import MinMaxScaler

# reading the ARFF file
data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

X = df.drop('class', axis=1)
y = df['class']

scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
```

1.

[4v] Using sklearn, apply k-means clustering fully unsupervisedly on the normalized data with $k \in \{2,3,4,5\}$ (random=0 and remaining parameters as default). Assess the silhouette and purity of the produced solutions.

```
from sklearn.cluster import KMeans
from sklearn import metrics

k_val = [2, 3, 4, 5]

for k in k_val:
    # n_init = 10 just to supress the warnings, doesn't affect the
results
    kmeans_model = KMeans(n_clusters=k, random_state=0,
n_init=10).fit(X_scaled)
    y_pred = kmeans_model.labels_

    print(f'k = {k}:')
    # compute silhouette
    silhouette = metrics.silhouette_score(X_scaled, y_pred)
    print('Silhouette Score: ', silhouette)

    # compute contingency/confusion matrix + purity score
    confusion_matrix = metrics.cluster.contingency_matrix(y, y_pred)
    purity_score = (np.sum(np.amax(confusion_matrix, axis=0)) /
np.sum(confusion_matrix))
    print('Purity: ', purity_score)
```

```
k = 2:
Silhouette Score:  0.36044124340441114
Purity:  0.632258064516129
k = 3:
Silhouette Score:  0.29579055730002257
Purity:  0.667741935483871
k = 4:
Silhouette Score:  0.27442402122340176
Purity:  0.6612903225806451
k = 5:
Silhouette Score:  0.23823928397844843
Purity:  0.6774193548387096
```

2.

[2v] Consider the application of PCA after the data normalization:

- i. Identify the variability explained by the top two principal components.
- ii. For each one of these two components, sort the input variables by relevance by inspecting the absolute weights of the linear projection.

```python
from sklearn.decomposition import PCA

# learn the transformation (components as linear combination of features)
pca = PCA(n_components=2)
pca.fit(X_scaled)

# access the explained variance (using eigenvalues)
print("\nExplained Variance Ratio:")
print("PC1:", pca.explained_variance_ratio_[0])
print("PC2:", pca.explained_variance_ratio_[1])

X_pca = pca.transform(X_scaled)
# 1 scale principal components
weight1 = pca.components_[0]
weight2 = pca.components_[1]

weights = pca.components_
sorted_weight_pc1 = [sorted(enumerate(weight1), key=lambda x: abs(x[1]), reverse=True)]
sorted_weight_pc2 = [sorted(enumerate(weight2), key=lambda x: abs(x[1]), reverse=True)]
sorted_weights = sorted_weight_pc1 + sorted_weight_pc2

for i, component in enumerate(sorted_weights):
    print(f"\nTop variables for PC{i + 1}:")
```

```
    for idx, weight in component:
        variable_name = X.columns[idx]
        print(f"{variable_name}: {weight:5f}")


Explained Variance Ratio:
PC1: 0.5618144484299207
PC2: 0.20955952591361904

Top variables for PC1:
pelvic_incidence: 0.591621
lumbar_lordosis_angle: 0.515085
pelvic_tilt: 0.467039
sacral_slope: 0.325689
degree_spondylolisthesis: 0.216930
pelvic_radius: -0.115824

Top variables for PC2:
pelvic_tilt: -0.670373
pelvic_radius: -0.581074
sacral_slope: 0.443303
pelvic_incidence: 0.100037
lumbar_lordosis_angle: 0.080047
degree_spondylolisthesis: 0.004583
```

3.

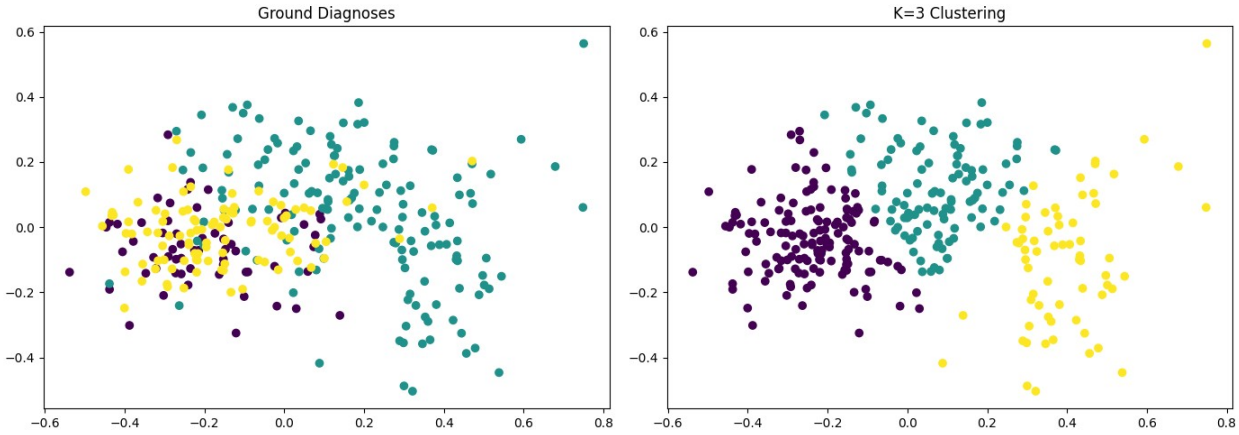
[2v] Visualize side-by-side the data using:

- 1) the ground diagnoses
- 2) the previously learned k =3 clustering solution.

To this end, projected the normalized data onto a 2-dimensional data space using PCA and then color observations using the reference and cluster annotations.

```
import matplotlib.pyplot as plt


kmeans_model = KMeans(n_clusters=3, random_state=0,
n_init=10).fit(X_scaled)
X_pca = pca.fit_transform(X_scaled)

temp = {"Hernia": 0, "Spondylolisthesis": 1, "Normal": 2}
y1 = [temp[name] for name in y]

# Visualize the data
plt.figure(figsize=(14,5))
```

```
plt.subplot(1, 2, 1)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y1)
plt.title('Ground Diagnoses')
plt.subplot(1, 2, 2)
plt.scatter(X_pca[:, 0], X_pca[:, 1],c=kmeans_model.labels_)
plt.title('K=3 Clustering')

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



4.

Considering the results from questions (1) and (3), identify two ways on how clustering can be used to characterize the population of ill and healthy individuals.

Com base nos resultados obtidos nas questões 1 e 3, verificamos que o silhouette score diminui progressivamente, iniciando em 0.36 para k=2 e terminando em 0.23 em k=5; a purity aumenta de 0.63 para 0.66 entre k=2 e k=3, e permanece praticamente inalterada até k=5. Os melhores valores de k aparentam, então, ser 2 e 3, pois consistem nos valores em que existe um melhor tradeoff entre o silhouette score e a purity, correspondendo à quantidade de clusters que melhor encapsulam grupos de indivíduos com caraterísticas idênticas; neste contexto, o clustering com k=2 pode identificar claramente grupos de indivíduos doentes e saudáveis; para k=3, esta solução de clustering pode identificar um grupo de indivíduos saudáveis, e dois grupos de indivíduos que partilham diferentes problemas de saúde ou fatores de risco. Recorrendo à visualização dos dados produzida no exercício 3, observando o ground diagnoses e posteriormente o clustering k=3 podemos reforçar a ideia de que para k=3, a solução de clustering poderá ser utilizada para caraterizar populações de indivíduos saudáveis e doentes, observando-se três grupos claramente distintos e bem identificados, sendo que um deles será de indivíduos saudáveis (a azul), e os outros de indivíduos que possivelmente partilhem caraterísticas como mencionado acima (a amarelo e roxo). Ainda, para novas observações, esta solução de clustering pode ser utilizada para avaliar a probabilidade de um indivíduo estar doente ou saudável, atribuindo-o ao cluster que melhor o caracteriza. Por exemplo, se um novo indivíduo for atribuído ao cluster azul, é provável que esteja saudável, enquanto que se for atribuído a um dos clusters amarelo ou roxo, provavelmente terá algum problema de saúde ou

fator de risco, dependendo do significado do cluster. Este tipo de classificação e a informação que providencia pode ser muito útil no contexto da avaliação inicial do estado de saúde de um indivíduo; deste modo, se forem necessárias avaliações médicas adicionais são realizadas mais rapidamente e são mais específicas.