HOMEWORK #2

APRENDIZAGEM
LEIC-A 2023/2024

Grupo #24
- Gonçalo Alves (Nº 103540)
- Daniel Nunes (Nº 103095)

I. Papel e Caneta

① a) De modo a treinar um classificador de Bayes, temos de calcular a fórmula resultante pelo estimador MAP.

Note-se que a fórmula para este estimador é:

$$\underset{c}{\text{argmax}}\ P(c|X) = \underset{c}{\text{argmax}}\ \frac{P(X|c)P(c)}{P(x)} =$$

$$= \underset{c}{\text{argmax}}\ P(x|c)P(c)\ .$$

Assim, teremos de calcular a probabilidade de cada prior ($P(c=?)$), a probabilidade de cada parâmetro de likelihood ($P(x|c=?)$) e a probabilidade de cada parâmetro de $P(x)$.
Para a classe de output $c=A$ (e usando apenas os dados x_1 a x_7 como dados de treino):

Prior
$$P(c=A) = \frac{3}{7}$$

Likelihood → $y_1, y_2 \perp y_3, y_4 \perp y_5$
$$P(X|c=A) = P(y_1, y_2 | c=A) \cdot P(y_3, y_4 | c=A) \cdot P(y_5 | c=A)$$

Como as variáveis y_1, y_2 são contínuas e $y_1 \times y_2 \in \mathbb{R}^2$ tem distribuição Normal, então temos de calcular os parâmetros da distribuição gaussiana multivariada de y_1 e y_2, no caso de a classe ser A

$$\vec{\mu}_{C=A} = \begin{bmatrix} \bar{y_1} \\ \bar{y_2} \end{bmatrix} = \begin{bmatrix} \frac{0.24+0.16+0.32}{3} \\ \frac{0.36+0.48+0.72}{3} \end{bmatrix} = \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}$$

$$\vec{\Sigma}_{C=A} = \begin{bmatrix} Var(y_1) & Cov(y_1,y_2) \\ Cov(y_1,y_2) & Var(y_2) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} \frac{(y_{1i}-\bar{y_1})^2}{6} & \sum_{1}^{3} \frac{(y_{1i}-\bar{y_1})(y_{2i}-\bar{y_2})}{6} \\ \sum_{1}^{3} \frac{(y_{1i}-\bar{y_1})(y_{2i}-\bar{y_2})}{6} & \sum_{i=1}^{3} \frac{(y_{2i}-\bar{y_2})^2}{6} \end{bmatrix} =$$

$$= \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix}$$

$$P(y_1, y_2 \mid C=A) = \mathcal{N}\left(y_1, y_2 \mid \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}, \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix}\right)$$

$$= \frac{1}{2\pi \times \sqrt{1.2288 \times 10^{-4}}} \exp\left\{-\frac{1}{2}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}\right)^T \times \right.$$

$$\left. \times \begin{bmatrix} 273.438 & -78.125 \\ -78.125 & 52.083 \end{bmatrix} \times \left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}\right)\right\}$$

Para y_3, y_4:

$$P(y_3=0, y_4=0 \mid C=A) = 0$$

$$P(y_3=0, y_4=1 \mid C=A) = \frac{1}{3}$$

$$P(y_3=1, y_4=0 \mid C=A) = \frac{1}{3}$$

$$P(y_3=1, y_4=1 \mid C=A) = \frac{1}{3}$$

Para y_5:

$P(y_5=0|C=A)=\frac{1}{3}$; $P(y_5=1|C=A)=\frac{1}{3}$; $P(y_5=2|C=A)=\frac{1}{3}$

Para a classe de output $C=B$ (e usando os mesmos dados de teste):

Prior

$$P(C=B)=\frac{4}{7}$$

Likelihood

$$P(X|C=B)=P(y_1,y_2|C=B)\cdot P(y_3,y_4|C=B)\cdot P(y_5|C=B)$$

$$P(y_1,y_2|C=B)=$$

$$=\mathcal{N}(y_1,y_2|\vec{\mu}_{C=B},\vec{\Sigma}_{C=B})=$$

$$=\mathcal{N}\left(y_1,y_2\left|\begin{bmatrix}0.5925\\0.3275\end{bmatrix},\begin{bmatrix}0.022892 & -0.009758\\-0.009758 & 0.031492\end{bmatrix}\right.\right)$$

$P(y_3=0,y_4=0|C=B)=\frac{1}{2}$

$P(y_3=0,y_4=1|C=B)=\frac{1}{4}$

$P(y_3=1,y_4=0|C=B)=\frac{1}{4}$

$P(y_3=1,y_4=1|C=B)=0$

$P(y_5=0|C=B)=\frac{1}{4}$; $P(y_5=1|C=B)=\frac{1}{2}$; $P(y_5=2|C=B)=\frac{1}{4}$

Denominador

Cálculo dos parâmetros da distribuição gaussiana multivariada de y_1 e y_2 (sem ser condicional a nenhuma classe):

$$\vec{\mu} = \begin{bmatrix} \frac{0.24+0.16+0.32+\dots}{7} \\ \frac{0.36+0.48+0.72+\dots}{7} \end{bmatrix} = \begin{bmatrix} 0.44143 \\ 0.41 \end{bmatrix}$$

$$\vec{\Sigma} = \begin{bmatrix} Var(y_1) & Cov(y_1,y_2) \\ Cov(y_1,y_2) & Var(y_2) \end{bmatrix} = \begin{bmatrix} 0.04908 & -0.02107 \\ -0.02107 & 0.03753 \end{bmatrix}$$

$$\Rightarrow P(y_1,y_2) = \mathcal{N}(y_1,y_2 | \vec{\mu} = \begin{bmatrix} 0.44143 \\ 0.41 \end{bmatrix}, \vec{\Sigma} = \begin{bmatrix} 0.04908 & -0.02107 \\ -0.02107 & 0.03753 \end{bmatrix})$$

$$= \frac{1}{2\pi \times \sqrt{1.3983 \times 10^{-3}}} \exp\left\{ -\frac{1}{2} \left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} 0.44143 \\ 0.41 \end{bmatrix} \right)^T \times \right.$$

$$\left. \times \begin{bmatrix} 26.84082 & 15.06519 \\ 15.06519 & 35.09876 \end{bmatrix} \times \left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} 0.44143 \\ 0.41 \end{bmatrix} \right) \right\}$$

- $P(y_3=0, y_4=0) = \frac{2}{7}$
- $P(y_3=0, y_4=1) = \frac{2}{7}$
- $P(y_3=1, y_4=0) = \frac{2}{7}$
- $P(y_3=1, y_4=1) = \frac{1}{7}$
- $P(y_5=0) = \frac{2}{7}$; $P(y_5=1) = \frac{3}{7}$; $P(y_5=2) = \frac{2}{7}$

$$\Rightarrow P(X_{novo}) = P(y_1,y_2) \cdot P(y_3,y_4) \cdot P(y_5)$$

(1)b) Para $x_8 = [0.38 \quad 0.52 \quad 0 \quad 1 \quad 0]^T$:

$$P(C=A \mid x_8) =$$

$$= \frac{P(x_8 \mid C=A) \cdot P(C=A)}{P(x_8)} =$$

$$= \frac{P(C=A) P(y_1=0.38, y_2=0.52 \mid C=A) \cdot P(y_3=0, y_4=1 \mid C=A) \cdot P(y_5=0 \mid C=A)}{P(y_1=0.38, y_2=0.52) \cdot P(y_3=0, y_4=1) \cdot P(y_5=0)} =$$ (X)

- $P(y_1=0.38, y_2=0.52 \mid C=A) = \mathcal{N}\left(x=\begin{bmatrix}0.38\\0.52\end{bmatrix} \mid \begin{bmatrix}0.24\\0.52\end{bmatrix}, \begin{bmatrix}0.0064 & 0.0096\\0.0096 & 0.0336\end{bmatrix}\right)$

$$= \frac{1}{2\pi\sqrt{1.288\times10^{-4}}} \exp\left\{-\frac{1}{2} \times \left(\begin{bmatrix}0.38\\0.52\end{bmatrix} - \begin{bmatrix}0.24\\0.52\end{bmatrix}\right)^T \times \right.$$

$$\left. \times \begin{bmatrix}273.438 & -78.125\\-78.125 & 52.083\end{bmatrix} \times \left(\begin{bmatrix}0.38\\0.52\end{bmatrix} - \begin{bmatrix}0.28\\0.52\end{bmatrix}\right)\right\}$$

$$= 14.3575 \times \exp\left\{-\frac{1}{2} \times [0.14 \quad 0] \begin{bmatrix}273.438 & -78.125\\-78.125 & 52.083\end{bmatrix} \begin{bmatrix}0.14\\0\end{bmatrix}\right\} =$$

$$= 14.3575 \times \exp\left\{-\frac{1}{2} \times 5.35738\right\} = 0.98470$$

- $P(y_3=0, y_4=1 \mid C=A) = \frac{1}{3}$
- $P(y_5=0 \mid C=A) = \frac{1}{3}$
- $P(C=A) = \frac{3}{7}$

- $P(y_1=0.38, y_2=0.52) = \mathcal{N}\left(x=\begin{bmatrix}0.38\\0.52\end{bmatrix} \mid \begin{bmatrix}0.44143\\0.41\end{bmatrix}, \begin{bmatrix}0.04908 & -0.02107\\-0.02107 & 0.03753\end{bmatrix}\right) =$

$$= 3.622472$$

- $P(y_3=0, y_4=1) = \frac{2}{7}$
- $P(y_5=0) = \frac{2}{7}$

Logo, $P(C=A \mid X_8 = [0.38 \quad 0.52 \quad 0 \quad 1 \quad 0]^T) =$

(*) $= \frac{\frac{3}{7} \times 0.98470 \times \frac{1}{3} \times \frac{1}{3}}{3.622472 \times \frac{2}{7} \times \frac{2}{7}} = \frac{0.04689}{0.29571} = \underline{0.15857}$

$$P(C=B \mid X_8) = \frac{P(X_8 \mid C=B)\, P(C=B)}{P(X_8)} =$$

$$= \frac{P(C=B) \cdot P(y_1=0.38, y_2=0.52 \mid C=B) \cdot P(y_3=0, y_4=1 \mid C=B) \cdot P(y_5=0 \mid C=B)}{P(X_8=[0.38 \quad 0.52 \quad 0 \quad 1 \quad 0]^T)}$$

$\cdot P(y_1=0.38, y_2=0.52 \mid C=B) = \mathcal{N}\left(X=\begin{bmatrix}0.38\\0.52\end{bmatrix} \middle| \begin{bmatrix}0.5925\\0.3275\end{bmatrix}, \begin{bmatrix}0.22892 & -0.009758\\-0.009758 & 0.031992\end{bmatrix}\right) =$

$= 1.96236$

$\cdot P(y_3=0, y_4=1 \mid C=B) = \frac{1}{4}$ $\quad \cdot P(y_5=0 \mid C=B) = \frac{1}{4}$

$\cdot P(C=B) = \frac{4}{7}$

$\Longrightarrow P(C=B \mid X_8) = \frac{\frac{4}{7} \times 1.96236 \times \frac{1}{4} \times \frac{1}{4}}{3.622472 \times \frac{2}{7} \times \frac{2}{7}} = \frac{0.07008}{0.29571} = \underline{0.237003}$

Como, para o classificador MAP, o critério é:

$$\underset{C}{\text{argmax}}\ P(C \mid X_{new}) = \underset{C}{\text{argmax}}\ \frac{P(X_{new} \mid C)\, P(C)}{P(X_{new})} =$$

$$= \underset{C}{\text{argmax}}\ P(X_{new} \mid C)\, P(C)$$

então o classificador de Bayes atribui a classe B ao conjunto de dados $X_8 = [0.38 \quad 0.52 \quad 0 \quad 1 \quad 0]^T$.

Para $x_q = [0.42 \quad 0.59 \quad 0 \quad 1 \quad 1]^T$:

- $P(y_1 = 0.42, y_2 = 0.59 \mid C = A) = \mathcal{N}\left(X = \begin{bmatrix} 0.42 \\ 0.59 \end{bmatrix} \middle| \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}, \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix}\right) =$
  $= 0.40307$

- $P(y_3 = 0, y_4 = 1 \mid C = A) = \frac{1}{3}$ $\quad$ - $P(y_5 = 1 \mid C = A) = \frac{1}{3}$
- $P(C = A) = \frac{3}{7}$

- $P(y_1 = 0.42, y_2 = 0.59) = \mathcal{N}\left(X = \begin{bmatrix} 0.42 \\ 0.59 \end{bmatrix} \middle| \begin{bmatrix} 0.44143 \\ 0.41 \end{bmatrix}, \begin{bmatrix} 0.04908 & -0.02107 \\ -0.02107 & 0.03753 \end{bmatrix}\right) =$
  $= 2.53881$

- $P(y_3 = 0, y_4 = 1) = \frac{2}{7}$ $\quad$ - $P(y_5 = 1) = \frac{3}{7}$

$\Rightarrow P(C = A \mid x_q = [0.42 \quad 0.59 \quad 0 \quad 1 \quad 1]^T) =$

$$= \frac{P(C=A)\,P(y_1 = 0.42, y_2 = 0.59 \mid C=A)\,P(y_3 = 0, y_4 = 1 \mid C=A)\,P(y_5 = 1 \mid C=A)}{P(y_1 = 0.42, y_2 = 0.59) \cdot P(y_3 = 0, y_4 = 1) \cdot P(y_5 = 1)}$$

$$= \frac{(3/7) \cdot 0.40307 \cdot (1/3) \cdot (1/3)}{2.53881 \cdot \frac{2}{7} \cdot \frac{3}{7}} = \frac{0.019194}{0.310874} = 0.061741$$

- $P(y_1 = 0.42, y_2 = 0.59 \mid C = B) = \mathcal{N}\left(X = \begin{bmatrix} 0.42 \\ 0.59 \end{bmatrix} \middle| \begin{bmatrix} 0.5925 \\ 0.3275 \end{bmatrix}, \begin{bmatrix} 0.22892 & -0.009758 \\ -0.009758 & 0.031992 \end{bmatrix}\right) =$
  $= 1.72857$

- $P(y_3 = 0, y_4 = 1 \mid C = B) = \frac{1}{4}$ $\quad$ - $P(y_5 = 1 \mid C = B) = \frac{1}{2}$
- $P(C = B) = \frac{4}{7}$

$\Rightarrow P(C = B \mid x_q = [0.42 \quad 0.59 \quad 0 \quad 1 \quad 1]^T) =$

$$= \frac{P(C=B)\,P(y_1 = 0.42, y_2 = 0.59 \mid C=B)\,P(y_3 = 0, y_4 = 1 \mid C=B)\,P(y_5 = 1 \mid C=B)}{P(y_1 = 0.42, y_2 = 0.59) \cdot P(y_3 = 0, y_4 = 1) \cdot P(y_5 = 1)} =$$

$$= \frac{(4/7)\cdot 1.72857\cdot(1/4)\cdot(1/2)}{P(x_q)} = \frac{0.123469}{0.310874} = 0.397168$$

Usando o mesmo critério de classificador MAP, conclui-se que o classificador de Bayes atribui a classe B ao conjunto de dados $x_q = [0.42 \;\; 0.59 \;\; 0 \;\; 1 \;\; 1]^T$.

(1)c) De modo a calcular o threshold que optimize a precisão dos testes, então temos de normalizar as probabilidades, de modo a que

$$P(A|x) + P(B|x) = 1.$$

Para x_8: $P(c=A|x_8) = 0.15857$
$P(c=B|x_8) = 0.237003$

Logo, após normalização:

$$\frac{P(c=A|x_8)}{P(c=A|x_8)+P(c=B|x_8)} = \frac{0.15857}{0.15857+0.237003} = 0.401$$

Para x_q: $P(c=A|x_q) = 0.061741$
$P(c=B|x_q) = 0.397168$

Logo, após normalização:

$$\frac{P(c=A|x_q)}{P(c=A|x_q)+P(c=B|x_q)} = \frac{0.061741}{0.061741+0.397168} = 0.134$$

Com o conjunto de dados de teste fornecidos no dataset, podemos então ajustar o threshold θ para um valor dentro do intervalo $]0.134, 0.401[$, fazendo com que a função de classificação seja do tipo:

$$f(x, \theta) = \begin{cases} A & \text{se } P(A|x) > \theta \\ B & \text{caso contrário} \end{cases}, \quad \text{com } \theta =]0.134, 0.401[$$

Deste modo, a entrada x_8 será classificada como A, visto que $P(C=A|x_8) > \theta$, e a entrada x_9 será classificada como B, pois $P(C=A|x_9) < \theta$.

(2a) • binarizar y_2 usando uma discretização de largura igual, sendo 1 var. numérica a dividir os valores de y_2 em três intervalos igualmente espaçados.

int 1 $\xrightarrow{\text{mapeado}}$ 0 : $[0, 0.5[\longrightarrow \{x_1, x_2, x_4, x_5, x_6\}$

int 2 $\xrightarrow{\text{mapeado}}$ 1 : $[0.5, 1] \longrightarrow \{x_3, x_7, x_8, x_9\}$

• atribuir aq valor de y_2 dentro do int 1 como 0 e do int 2 como 1. → Binarize

• dividir os dados em 3 folds, como não estamos a baralhar as observações, os folds serão apenas segmentos consecutivos.

Fold 1: x_1 (0, 1, 1, 0, A)
x_2 (0, 1, 0, 1, A)
x_3 (1, 0, 1, 2, A)

Fold 2: x_4 (0, 0, 0, 1, B)
x_5 (0, 0, 0, 0, B)
x_6 (0, 1, 0, 2, B)

Fold 3: x_7 (1, 0, 1, 1, B)
x_8 (1, 0, 1, 0, A)
x_9 (1, 0, 1, 1, B)

b) • Calcular distâncias de Hamming para as observações.

↳ # de características em que diferem

- Observações teste: x_7, x_8, x_9
- // treino: $x_1, x_2, x_3, x_4, x_5, x_6$

$d(x_7, x_1) = 4$	$d(x_8, x_1) = 2$	$d(x_9, x_1) = 4$
$d(x_7, x_2) = 4$	$d(x_8, x_2) = 4$	$d(x_9, x_2) = 4$
$d(x_7, x_3) = 2$	$d(x_8, x_3) = 1$	$d(x_9, x_3) = 2$
$d(x_7, x_4) = 2$	$d(x_8, x_4) = 4$	$d(x_9, x_4) = 2$
$d(x_7, x_5) = 3$	$d(x_8, x_5) = 3$	$d(x_9, x_5) = 3$
$d(x_7, x_6) = 4$	$d(x_8, x_6) = 5$	$d(x_9, x_6) = 4$

• De seguida classificamos as observações de treino com base na distânc de Haming em ordem crescente e selecionamos os $K = 3$ vizinhos mais próximos, para cada obs teste

$$\hat{z}_7 = \frac{\frac{1}{2} \times 0.32 + \frac{1}{2} \times 0.54 + \frac{1}{3} \times 0.66}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} \simeq 0.49$$

$$\hat{z}_8 = \frac{\frac{1}{2} \times 0.24 + 1 \times 0.32 + \frac{1}{3} \times 0.66}{\frac{1}{2} + 1 + \frac{1}{3}} = 0.36$$

$$\hat{z}_9 = \frac{\frac{1}{2} \times 0.32 + \frac{1}{2} \times 0.54 + \frac{1}{3} \times 0.66}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} \simeq 0.49$$

$$z = \begin{bmatrix} 0.41 \\ 0.38 \\ 0.42 \end{bmatrix} \quad \hat{z} = \begin{bmatrix} 0.49 \\ 0.36 \\ 0.49 \end{bmatrix}$$

$$MAE(\hat{z}, z) = \frac{1}{n} \sum_{i=1}^{n} |z_i - \hat{z}_i|$$

$$MAE = \frac{|0.41 - 0.49| + |0.38 - 0.36| + |0.42 - 0.49|}{3}$$

$$\simeq 0.055$$

$$MAE(\hat{z}, z) = 5.5\%$$

Homework 2 (Part II)

Aprendizagem 2023/2024 - LEIC @ IST

Group #24

- Daniel Nunes (Nº 103095)
- Gonçalo Alves (Nº 103540)

Data importing and preparation

In [ ]:
```
import pandas as pd, numpy as np
from scipy.io.arff import loadarff

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

df.head()
```

Out[ ]:

	pelvic_incidence	pelvic_tilt	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondyl
0	63.027817	22.552586	39.609117	40.475232	98.672917	-
1	39.056951	10.060991	25.015378	28.995960	114.405425	
2	68.832021	22.218482	50.092194	46.613539	105.985135	
3	69.297008	24.652878	44.311238	44.644130	101.868495	
4	49.712859	9.652075	28.317406	40.060784	108.168725	

In [ ]:
```
X = df.drop('class', axis=1);
Y = df['class'];
```

Using sklearn, apply a 10-fold stratified cross-validation with shuffling (`random_state=0`) for the assessment of predictive models along this section.

Exercise 1

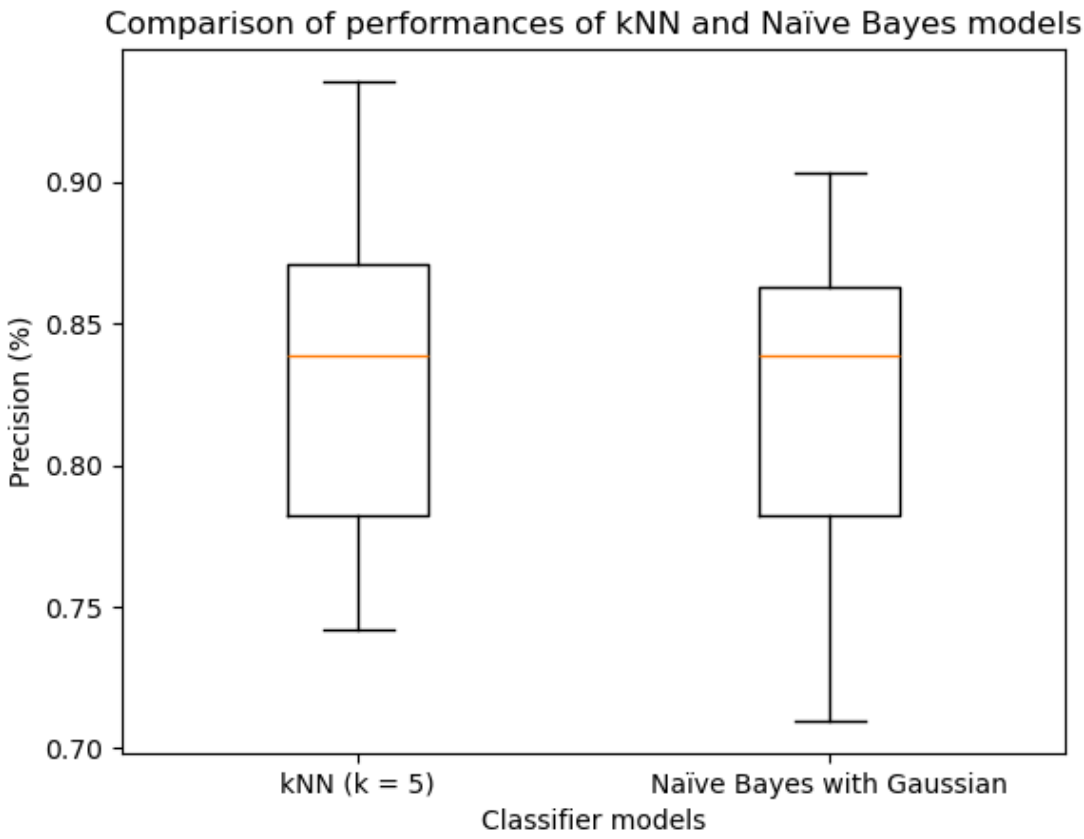
Compare the performance of kNN with k=5 and naïve Bayes with Gaussian assumption (consider all remaining parameters for each classifier as sklearn's default):

a. Plot two boxplots with the fold accuracies for each classifier.

In [ ]:
```
import numpy as np
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
```

```
knn_accuracy = []
nb_accuracy = []

skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

X_array = np.array(X)
y_array = np.array(Y)

for train_index, test_index in skf.split(X_array, y_array):
    X_train, X_test = X_array[train_index], X_array[test_index]
    Y_train, Y_test = y_array[train_index], y_array[test_index]

    # Treine e ajuste o classificador kNN
    knn = KNeighborsClassifier(n_neighbors=5)  # k=5
    knn.fit(X_train, Y_train)
    knn_pred = knn.predict(X_test)
    knn_acc = accuracy_score(Y_test, knn_pred)
    knn_accuracy.append(knn_acc)

    # Treine e ajuste o classificador Naïve Bayes Gaussiano
    nb = GaussianNB()
    nb.fit(X_train, Y_train)
    nb_pred = nb.predict(X_test)
    nb_acc = accuracy_score(Y_test, nb_pred)
    nb_accuracy.append(nb_acc)
```

In [ ]:
```
import matplotlib.pyplot as plt

plt.boxplot([knn_accuracy, nb_accuracy], labels=['kNN (k = 5)', \
    'Naïve Bayes with Gaussian'], widths=(0.3, 0.3))
plt.xlabel('Classifier models')
plt.ylabel('Precision (%)')
plt.title('Comparison of performances of kNN and Naïve Bayes models')
plt.show()
```



b. Using scipy, test the hypothesis "*k*NN is statistically superior to naïve Bayes regarding accuracy", asserting whether is true.

In [ ]:
```
from scipy import stats

p_value = stats.ttest_rel(knn_accuracy, nb_accuracy).pvalue
print(f"P-Value: {p_value}")
```

```
P-Value: 0.38085618124128184
```

Our answer

Since the P-Value generated by the t_test function is approximately equal to 0.38, then we cannot conclude if kNN is more accurate than the Naïve Bayes approach, since only a P-Value lower than 0.05 would indicate a discrepancy on the accuracies returned by both classifiers.

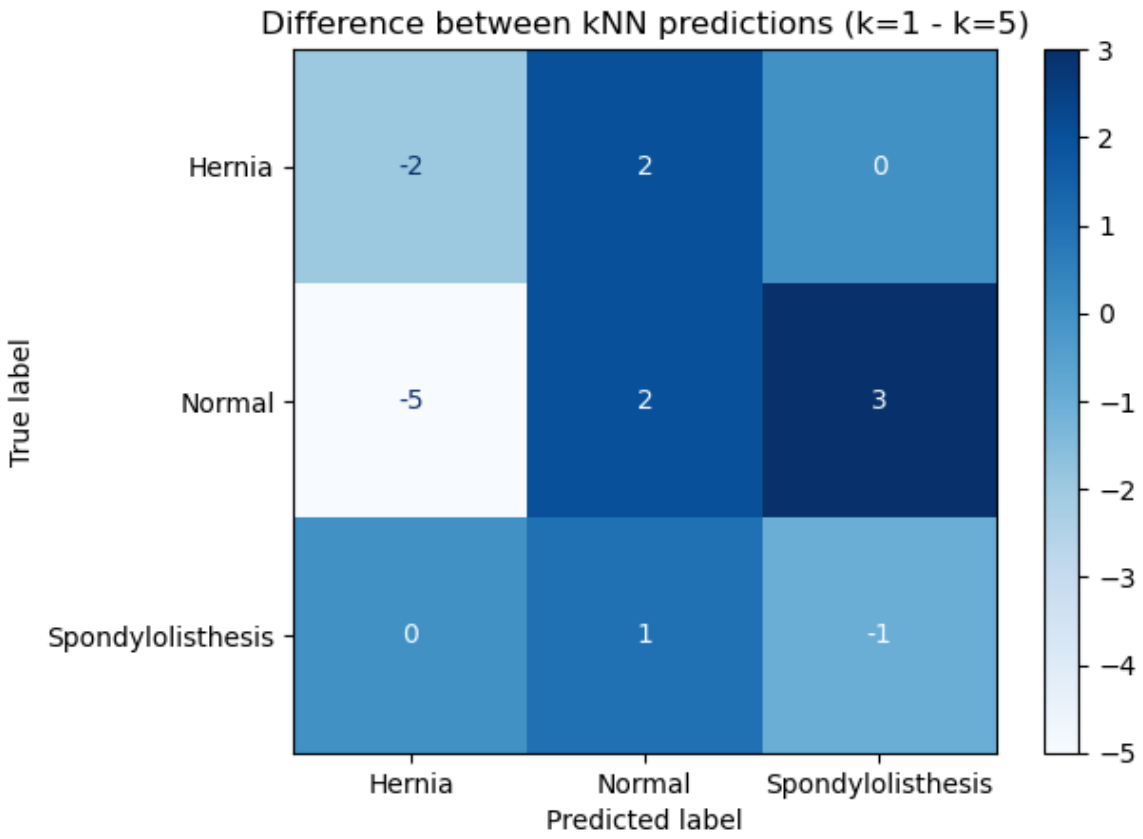
Exercise 2

Consider two kNN predictors with k=1 and k=5 (uniform weights, Euclidean distance, all remaining parameters as default). Plot the differences between the two cumulative confusion matrices of the predictors. Comment.

In [ ]:
```
from sklearn.metrics import confusion_matrix

kNN_1_confusion_matrices = []
kNN_5_confusion_matrices = []
```

```
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

X_array = np.array(X)
y_array = np.array(Y)

for train_index, test_index in skf.split(X_array, y_array):
    X_train, X_test = X_array[train_index], X_array[test_index]
    Y_train, Y_test = y_array[train_index], y_array[test_index]

    # k = 1
    kNN_1 = KNeighborsClassifier(n_neighbors=1)
    knn1_predictor = kNN_1.fit(X_train, Y_train)
    kNN_1_predictions = kNN_1.predict(X_test)
    kNN_1_confusion_matrix = confusion_matrix(Y_test, kNN_1_predictions)
    kNN_1_confusion_matrices.append(kNN_1_confusion_matrix)

    # k = 5
    kNN_5 = KNeighborsClassifier(n_neighbors=5)
    knn5_predictor = kNN_5.fit(X_train, Y_train)
    kNN_5_predictions = kNN_5.predict(X_test)
    kNN_5_confusion_matrix = confusion_matrix(Y_test, kNN_5_predictions)
    kNN_5_confusion_matrices.append(kNN_5_confusion_matrix)

kNN_1_cumulative_confusion = sum(kNN_1_confusion_matrices)
kNN_5_cumulative_confusion = sum(kNN_5_confusion_matrices)

kNN_confusion_difference = \
    kNN_1_cumulative_confusion - kNN_5_cumulative_confusion
```

In [ ]:
```
from sklearn.metrics import ConfusionMatrixDisplay

matrix_display = ConfusionMatrixDisplay( \
    confusion_matrix=kNN_confusion_difference, \
    display_labels=knn1_predictor.classes_ \
)
matrix_display.plot(cmap=plt.cm.Blues)
plt.title("Difference between kNN predictions (k=1 - k=5)")
plt.show()
```



Our answer

Given the result above, we can observe that the model with k=5 performs better in correctly classifying new items as Hernia and Spondylolisthesis, while the kNN model with k=1 performs better in classifying it as Normal. Because of this, the first model is probably less successful than the second at correctly identifying the various instances of each class.

Furthermore, the second model seems to have less false positives and, more importantly, less false negatives, which can be a more critical factor for diagnosing patients according to the properties used in the "column_diagnosis" dataset.

Therefore, by analysing the results of this confusion matrix, we can conclude that a bigger value of k in a kNN predictor can lead to a better impact in the model's performance.

Exercise 3

Considering the unique properties of `column_diagnosis`, identify three possible difficulties of naïve Bayes when learning from the given dataset.

Our answer

We can observe three possible problems of using a naïve Bayes classifier for learning the `column_diagnosis` dataset:

1. Almost all inputs of this dataset are continuous variables, which means that, when using a

Bayesian approach for creating a model from this data, it automatically assumes that each continuous variable follows a certain distribution (tipically a Gaussian distribution). However, if some of the variables of this dataset follow other distributions, using a naïve Bayes model might create some errors.

2. Naïve Bayes assumes that features are conditionally independent given the class label. This is done in order to try to achieve better optimization results while not sacrificing accuracy. However, we cannot fully confirm that the variables are completely independent from one another. If this is not the case, then the accuracy of the Bayesian model can decrease substantially.

3. After analysing this dataset, we concluded that there are 60 data points for the class "Hernia", 100 for the class "Normal" and 150 for a diagnosis of "Spondylolisthesis". Therefore, the classes in this dataset are imbalanced. This can introduce bias on the naïve model, meaning that for each new set of properties, the model can have a bigger chance of classifying it as one class, as it was trained with more data for that specific class.