TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 018

I. Pen-and-paper

1) [4v] Compute the recall of a distance-weighted kNN with $k = 5$ and distance $d(\mathbf{x}1, \mathbf{x}2) = Hamming(\mathbf{x}1, \mathbf{x}2) + ½$ using leave-one-out evaluation schema (i.e., when classifying one observation, use all remaining ones).

$$Hamming(x_1, x_2) = \sum_{i=1}^{n} T(a_{1i}, a_{2i}) \qquad recall/sensitivity = \frac{TP}{TP + FN} \qquad w_{i,j} = \frac{1}{d(x_i, x_j)}$$

$$d(x_1, x_2) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_1, x_3) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_1, x_4) = (0 + 0) + \frac{1}{2} = \frac{1}{2}$$
$$d(x_1, x_5) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_1, x_6) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_1, x_7) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_1, x_8) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$

$$d(x_6, x_7) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_6, x_8) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$

$$d(x_2, x_3) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_2, x_4) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_2, x_5) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_2, x_6) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_2, x_7) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_2, x_8) = (0 + 0) + \frac{1}{2} = \frac{1}{2}$$

$$d(x_5, x_6) = (0 + 0) + \frac{1}{2} = \frac{1}{2}$$
$$d(x_5, x_7) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_5, x_8) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_7, x_8) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$

$$d(x_3, x_4) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_3, x_5) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_3, x_6) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$
$$d(x_3, x_7) = (0 + 0) + \frac{1}{2} = \frac{1}{2}$$
$$d(x_3, x_8) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$

$$d(x_4, x_5) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_4, x_6) = (1 + 0) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_4, x_7) = (0 + 1) + \frac{1}{2} = \frac{3}{2}$$
$$d(x_4, x_8) = (1 + 1) + \frac{1}{2} = \frac{5}{2}$$

$$kNN(x_1)_{k=5} = \{x_3, x_4, x_5, x_6, x_7\} \rightarrow \hat{z}_1 = moda[(\tfrac{2}{3} + 2) \times P, (\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3}) \times N] = moda(\tfrac{8}{3}\, P,\ 2\, N) = P$$
$$kNN(x_2)_{k=5} = \{x_3, x_5, x_6, x_7, x_8\} \rightarrow \hat{z}_2 = moda[(\tfrac{2}{3}) \times P, (\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3} + 2) \times N] = moda(\tfrac{2}{3}\, P,\ 4\, N) = N$$
$$kNN(x_3)_{k=5} = \{x_1, x_2, x_4, x_7, x_8\} \rightarrow \hat{z}_3 = moda[(\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{5}) \times P, (\tfrac{2}{3} + 2) \times N] = moda(\tfrac{26}{15}\, P,\ \tfrac{8}{3}\, N) = N$$
$$kNN(x_4)_{k=5} = \{x_1, x_3, x_5, x_6, x_7\} \rightarrow \hat{z}_4 = moda[(2 + \tfrac{2}{3}) \times P, (\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3}) \times N] = moda(\tfrac{8}{3}\, P,\ 2\, N) = P$$
$$kNN(x_5)_{k=5} = \{x_1, x_2, x_4, x_6, x_8\} \rightarrow \hat{z}_5 = moda[(\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3}) \times P, (2 + \tfrac{2}{3}) \times N] = moda(2\, P,\ \tfrac{8}{3}\, N) = N$$
$$kNN(x_6)_{k=5} = \{x_1, x_2, x_4, x_5, x_8\} \rightarrow \hat{z}_6 = moda[(\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3}) \times P, (2 + \tfrac{2}{3}) \times N] = moda(2\, P,\ \tfrac{8}{3}\, N) = N$$
$$kNN(x_7)_{k=5} = \{x_1, x_2, x_3, x_4, x_8\} \rightarrow \hat{z}_7 = moda[(\tfrac{2}{3} + \tfrac{2}{3} + 2 + \tfrac{2}{3}) \times P, (\tfrac{2}{3}) \times N] = moda(4\, P,\ \tfrac{2}{3}\, N) = P$$
$$kNN(x_8)_{k=5} = \{x_2, x_3, x_5, x_6, x_7\} \rightarrow \hat{z}_8 = moda[(\tfrac{2}{3} + 2) \times P, (\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{2}{3}) \times N] = moda(\tfrac{8}{3}\, P,\ 2\, N) = P$$

	y_1	y_2	z
x_1	A	0	P
x_2	B	1	P
x_3	A	1	P
x_4	A	0	P
x_5	B	0	N
x_6	B	0	N
x_7	A	1	N
x_8	B	1	N

Confusion matrix:

Previstos

Reais	P	N
P	2	2
N	2	2

Logo, $Recall_P = \frac{2}{2+2} = \frac{1}{2}$, $Recall_N = \frac{2}{2+2} = \frac{1}{2}$ ❒

Aprendizagem 2022/23

Homework II – Group 018

2) [4v] Considering the nine training observations, learn a Bayesian classifier assuming:

i) y_1 and y_2 are dependent;

ii) $\{y_1, y_2\}$ and $\{y_3\}$ variable sets are independent and equally important;

iii) y_3 is normally distributed. Show all parameters.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i;$$

$$\sigma_k^2 = \frac{1}{n-1}\sum_{i=1}^{n} (y_{ki} - \mu)^2;$$

$$p(y_k \mid z = P) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_k-\mu)^2};$$

$$p(y \mid \mu, \Sigma) = N(y \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \times exp(-\frac{1}{2}(y - \mu)^T \Sigma^{-1}(y - \mu)).$$

$p(y_1, y_2 \mid z = P)$:

- Assumindo $A = 0$ e $B = 1$, temos que:

$$\mu = \frac{1}{5}\left(\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}0.4\\0.4\end{bmatrix}, \qquad \Sigma = \begin{bmatrix} var(y_1) & cov(y_1, y_2) \\ cov(y_1, y_2) & var(y_2)\end{bmatrix}$$

$$var(y_1) = \frac{1}{5-1}\sum_{i=1}^{4}(y_{1i} - \mu_1)^2 = \frac{1}{4}((0-0.4)^2 + (1-0.4)^2 + (0-0.4)^2 + (0-0.4)^2 + (1-0.4)^2) = 0.3$$

$$var(y_2) = \frac{1}{5-1}\sum_{i=1}^{4}(y_{2i} - \mu_2)^2 = \frac{1}{4}((0-0.4)^2 + (1-0.4)^2 + (1-0.4)^2 + (0-0.4)^2 + (0-0.4)^2) = 0.3$$

$$cov(y_1, y_2) = \frac{1}{5-1}\sum_{i=1}^{4}(y_{1i} - \mu_1)(y_{2i} - \mu_2) =$$
$$= \frac{1}{4}((0-0.4)(0-0.4) + (1-0.4)(1-0.4) + (0-0.4)(1-0.4) +$$
$$(0-0.4)(0-0.4) + (1-0.4)(0-0.4)) = 0.05$$

- $\mu = \begin{bmatrix}0.4\\0.4\end{bmatrix}, \qquad \Sigma = \begin{bmatrix}0.3 & 0.05\\0.05 & 0.3\end{bmatrix}$

$$|\Sigma| = 0.3 \times 0.3 - 0.05 \times 0.05 = 0.0875$$

$$\Sigma^{-1} = \frac{1}{0.0875}\begin{bmatrix}0.3 & -0.05\\-0.05 & 0.3\end{bmatrix} \simeq \begin{bmatrix}3.429 & -0.571\\-0.571 & 3.429\end{bmatrix}$$

Logo, temos que:

- $p(y_1, y_2 \mid z = P) = \frac{1}{2\pi\sqrt{0.0875}} \exp\left(-\frac{1}{2}\begin{bmatrix}y_1 - 0.4 & y_2 - 0.4\end{bmatrix}\begin{bmatrix}3.429 & -0.571\\-0.571 & 3.429\end{bmatrix}\begin{bmatrix}y_1 - 0.4\\y_2 - 0.4\end{bmatrix}\right)$

Aprendizagem 2022/23
Homework II – Group 018

$p(y_1, y_2 \mid z = N)$:

- Assumindo $A = 0$ e $B = 1$, temos que:

$$\mu = \frac{1}{4}\left(\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix}\right) = \begin{bmatrix}0.75\\0.5\end{bmatrix}, \qquad \Sigma = \begin{bmatrix} var(y_1) & cov(y_1, y_2) \\ cov(y_1, y_2) & var(y_2) \end{bmatrix}$$

$$var(y_1) = \frac{1}{4-1}\sum_{i=1}^{4}(y_{1i} - \mu_1)^2 = \frac{1}{3}((1-0.75)^2 + (1-0.75)^2 + (0-0.75)^2 + (1-0.75)^2) = 0.25$$

$$var(y_2) = \frac{1}{4-1}\sum_{i=1}^{4}(y_{2i} - \mu_2)^2 = \frac{1}{3}((0-0.5)^2 + (0-0.5)^2 + (1-0.5)^2 + (1-0.5)^2) = \frac{1}{3}$$

$$cov(y_1, y_2) = \frac{1}{4-1}\sum_{i=1}^{4}(y_{1i} - \mu_1)(y_{2i} - \mu_2) =$$
$$= \frac{1}{3}\big((1-0.75)(0-0.5) + (1-0.75)(0-0.5) + (0-0.75)(1-0.5) + (1-0.75)(1-0.5)\big) = -\frac{1}{6}$$

- $\mu = \begin{bmatrix}0.75\\0.5\end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 0.25 & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{3} \end{bmatrix}$

$$|\Sigma| = 0.25 \times \frac{1}{3} - (-\frac{1}{6} \times (-\frac{1}{6})) = \frac{1}{18}$$

$$\Sigma^{-1} = 18 \begin{bmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & 0.25 \end{bmatrix} = \begin{bmatrix} 6 & 3 \\ 3 & 4.5 \end{bmatrix}$$

Logo, temos que:

- $p(y_1, y_2 \mid z = N) = \frac{1}{2\pi\sqrt{\frac{1}{18}}} \exp\left(-\frac{1}{2}\begin{bmatrix} y_1 - 0.75 & y_2 - 0.5 \end{bmatrix} \begin{bmatrix} 6 & 3 \\ 3 & 4.5 \end{bmatrix} \begin{bmatrix} y_1 - 0.75 \\ y_2 - 0.5 \end{bmatrix}\right)$

$p(y_3 \mid z = P)$:

$$\mu = \frac{1}{5}(1.2 + 0.8 + 0.5 + 0.9 + 0.8) = 0.84$$

$$\sigma^2 = \frac{1}{5-1}\sum_{i=1}^{5}(y_{3i} - \mu)^2 =$$
$$= \frac{1}{4}((1.2-0.84)^2 + (0.8-0.84)^2 + (0.5-0.84)^2 + (0.9-0.84)^2 + (0.8-0.84)^2) = 0.063$$

Logo, temos que:

- $p(y_3 \mid z = P) = \frac{1}{\sqrt{2\pi\, 0.063}} \exp(-\frac{1}{2 \times 0.063}(y_3 - 0.84)^2)$

Aprendizagem 2022/23

Homework II – Group 018

$p(y_3 \mid z = N)$:

$$\mu = \frac{1}{4}(1 + 0.9 + 1.2 + 0.8) = 0.975$$

$$\sigma^2 = \frac{1}{4-1}\sum_{i=1}^{5}(y_{3i} - \mu)^2 =$$

$$= \frac{1}{3}((1-0.975)^2 + (0.9-0.975)^2 + (1.2-0.975)^2 + (0.8-0.975)^2) \simeq 0.029$$

Logo, temos que:

- $p(y_3 \mid z = N) = \frac{1}{\sqrt{2\pi \times 0.029}} \exp(-\frac{1}{2 \times 0.029}(y_3 - 0.975)^2)$

E então, para concluir, temos que:

- $p(z = P \mid x_{new}) = \frac{p(x_{new} \mid z=P)\, p(z=P)}{p(x_{new})} = \frac{p(x_{new} \mid z=P)\, p(z=P)}{p(x_{new} \mid z=P)\, p(z=P) + p(x_{new} \mid z=N)\, p(z=N)}$;
- $p(z = N \mid x_{new}) = 1 - p(z = P \mid x_{new})$;
- $p(x_{new} \mid z = P) = p(y_1, y_2 \mid z = P)\, p(y_3 \mid z = P)$, $\quad p(z = P) = \frac{5}{9}$;
- $p(x_{new} \mid z = N) = p(y_1, y_2 \mid z = N)\, p(y_3 \mid z = N)$, $\quad p(z = N) = \frac{4}{9}$.

Onde:

- $p(y_1, y_2 \mid z = P) = \frac{1}{2\pi\sqrt{0.0875}} \exp\left(-\frac{1}{2}\begin{bmatrix} y_1 - 0.4 & y_2 - 0.4\end{bmatrix} \begin{bmatrix} 3.429 & -0.571 \\ -0.571 & 3.429 \end{bmatrix} \begin{bmatrix} y_1 - 0.4 \\ y_2 - 0.4 \end{bmatrix}\right)$;
- $p(y_1, y_2 \mid z = N) = \frac{1}{2\pi\sqrt{\frac{1}{18}}} \exp\left(-\frac{1}{2}\begin{bmatrix} y_1 - 0.75 & y_2 - 0.5\end{bmatrix} \begin{bmatrix} 6 & 3 \\ 3 & 4.5 \end{bmatrix} \begin{bmatrix} y_1 - 0.75 \\ y_2 - 0.5 \end{bmatrix}\right)$;
- $p(y_3 \mid z = P) = \frac{1}{\sqrt{2\pi\, 0.063}} \exp(-\frac{1}{2 \times 0.063}(y_3 - 0.84)^2)$;
- $p(y_3 \mid z = N) = \frac{1}{\sqrt{2\pi \times 0.029}} \exp(-\frac{1}{2 \times 0.029}(y_3 - 0.975)^2)$.

	y_1	y_2	y_3	z
x_1	A	0	1.2	P
x_2	B	1	0.8	P
x_3	A	1	0.5	P
x_4	A	0	0.9	P
x_{new}	B	0	0.8	P
x_5	B	0	1	N
x_6	B	0	0.9	N
x_7	A	1	1.2	N
x_8	B	1	0.8	N

	y_1	y_2	y_3	z
x_1	0	0	1.2	P
x_2	1	1	0.8	P
x_3	0	1	0.5	P
x_4	0	0	0.9	P
x_{new}	1	0	0.8	P
x_5	1	0	1	N
x_6	1	0	0.9	N
x_7	0	1	1.2	N
x_8	1	1	0.8	N

Aprendizagem 2022/23

Homework II – Group 018

3) [3v] Under a MAP assumption, compute $P(\text{Positive}|\mathbf{x})$ of each testing observation.

	y_1	y_2	y_3	z
x_{new1}	A	1	0.8	P
x_{new2}	B	1	1	P
x_{new3}	B	0	0.9	N

- $MAP = argmax_c\ p(c = P\,|\,x) \Rightarrow argmax_c\ p(x\,|\,c = P)\,p(c = P)$
- $p(c = P) = \frac{5}{9}, \quad p(c = N) = \frac{4}{9}$

Sabemos que:

$$p(z = C\,|\,x_{new}) = p(z = C\,|\,\{y_1, y_2, y_3\}) =$$
$$= \frac{p(\{y_1, y_2\}\,|\,z=C)\times p(\{y_3\}\,|\,z=C)\times p(z=C)}{p(\{y_1, y_2\}\,|\,z=C)\times p(\{y_3\}\,|\,z=C)\times p(z=C) + p(\{y_1, y_2\}\,|\,z=\neg C)\times p(\{y_3\}\,|\,z=\neg C)\times p(z=\neg C)}$$

Tendo em conta os resultados da alínea anterior, temos que:

$p(z = P|\,x_{new1})$:

$$p(y_1 = A, y_2 = 1\,|\,z = P) = \frac{1}{2\pi\sqrt{0.0875}}\exp\left(-\frac{1}{2}\begin{bmatrix}0-0.4 & 1-0.4\end{bmatrix}\begin{bmatrix}3.429 & -0.571\\ -0.571 & 3.429\end{bmatrix}\begin{bmatrix}0-0.4\\ 1-0.4\end{bmatrix}\right) \simeq 0.1924$$
$$p(y_3 = 0.8|\,z = P) = \frac{1}{\sqrt{2\pi\,0.063}}\exp(-\frac{1}{2\times 0.063}(0.8-0.84)^2) \simeq 1.5694$$

$$p(y_1 = A, y_2 = 1\,|\,z = N) = \frac{1}{2\pi\sqrt{\frac{1}{18}}}\exp\left(-\frac{1}{2}\begin{bmatrix}0-0.75 & 1-0.5\end{bmatrix}\begin{bmatrix}6 & 3\\ 3 & 4.5\end{bmatrix}\begin{bmatrix}0-0.75\\ 1-0.5\end{bmatrix}\right) \simeq 0.2192$$
$$p(y_3 = 0.8|\,z = N) = \frac{1}{\sqrt{2\pi\times 0.029}}\exp(-\frac{1}{2\times 0.029}(0.8-0.975)^2) \simeq 1.3816$$

Logo,

- $p(z = P\,|\,x_{new1}) = \frac{0.1924\times 1.5694\times\frac{5}{9}}{(0.1924\times 1.5694\times\frac{5}{9}) + (0.2192\times 1.3816\times\frac{4}{9})} \simeq 0.5548$ ❒

$p(z = P|\, x_{new2})$:

$$p(y_1 = B, y_2 = 1 \mid z = P) = \frac{1}{2\pi\sqrt{0.0875}} \exp\left(-\frac{1}{2}[1-0.4 \quad 1-0.4]\begin{bmatrix} 3.429 & -0.571 \\ -0.571 & 3.429 \end{bmatrix}\begin{bmatrix} 1-0.4 \\ 1-0.4 \end{bmatrix}\right) \simeq 0.1923$$

$$p(y_3 = 1 |\, z = P) = \frac{1}{\sqrt{2\pi\, 0.063}} \exp(-\frac{1}{2\, \times 0.063}(1-0.84)^2) \simeq 1.2972$$

$$p(y_1 = B, y_2 = 1 \mid z = N) = \frac{1}{2\pi\sqrt{\frac{1}{18}}} \exp\left(-\frac{1}{2}[1-0.75 \quad 1-0.5]\begin{bmatrix} 6 & 3 \\ 3 & 4.5 \end{bmatrix}\begin{bmatrix} 1-0.75 \\ 1-0.5 \end{bmatrix}\right) \simeq 0.2192$$

$$p(y_3 = 1 |\, z = N) = \frac{1}{\sqrt{2\pi\, \times\, 0.029}} \exp(-\frac{1}{2\, \times\, 0.029}(1-0.975)^2) \simeq 2.3176$$

Logo,

- $p(z = P \mid x_{new2}) = \frac{0.1923 \times 1.2972 \times \frac{5}{9}}{(0.1923 \times 1.2972 \times \frac{5}{9}) + (0.2192 \times 2.3176 \times \frac{4}{9})} \simeq 0.3803$ ❒

$p(z = P|\, x_{new3})$:

$$p(y_1 = B, y_2 = 0 \mid z = P) = \frac{1}{2\pi\sqrt{0.0875}} \exp\left(-\frac{1}{2}[1-0.4 \quad 0-0.4]\begin{bmatrix} 3.429 & -0.571 \\ -0.571 & 3.429 \end{bmatrix}\begin{bmatrix} 1-0.4 \\ 0-0.4 \end{bmatrix}\right) \simeq 0.1924$$

$$p(y_3 = 0.9 |\, z = P) = \frac{1}{\sqrt{2\pi\, 0.063}} \exp(-\frac{1}{2\, \times 0.063}(0.9-0.84)^2) \simeq 1.5447$$

$$p(y_1 = B, y_2 = 0 \mid z = N) = \frac{1}{2\pi\sqrt{\frac{1}{18}}} \exp\left(-\frac{1}{2}[1-0.75 \quad 0-0.5]\begin{bmatrix} 6 & 3 \\ 3 & 4.5 \end{bmatrix}\begin{bmatrix} 1-0.75 \\ 0-0.5 \end{bmatrix}\right) \simeq 0.4641$$

$$p(y_3 = 0.9 |\, z = N) = \frac{1}{\sqrt{2\pi\, \times\, 0.029}} \exp(-\frac{1}{2\, \times\, 0.029}(0.9-0.975)^2) \simeq 2.1261$$

Logo,

- $p(z = P \mid x_{new3}) = \frac{0.1924 \times 1.5447 \times \frac{5}{9}}{(0.1924 \times 1.5447 \times \frac{5}{9}) + (0.4641 \times 2.1261 \times \frac{4}{9})} \simeq 0.2735$ ❒

4) [2v] Given a binary class variable, the default decision threshold of $\theta = 0.5$,

$$f(\mathbf{x}|\theta) = \begin{cases} \text{Positive} & P(\text{Positive}|\mathbf{x}) > \theta \\ \text{Negative} & \text{otherwise} \end{cases}$$

can be adjusted. Which decision threshold – 0.3, 0.5 or 0.7 – optimizes testing accuracy?

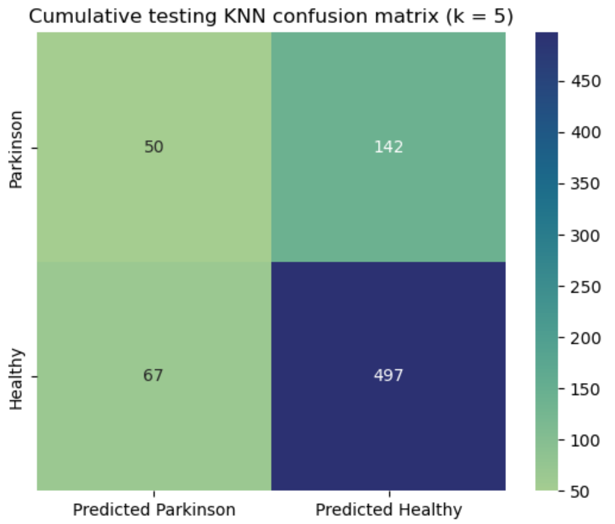
Dos resultados da alínea anterior temos que:

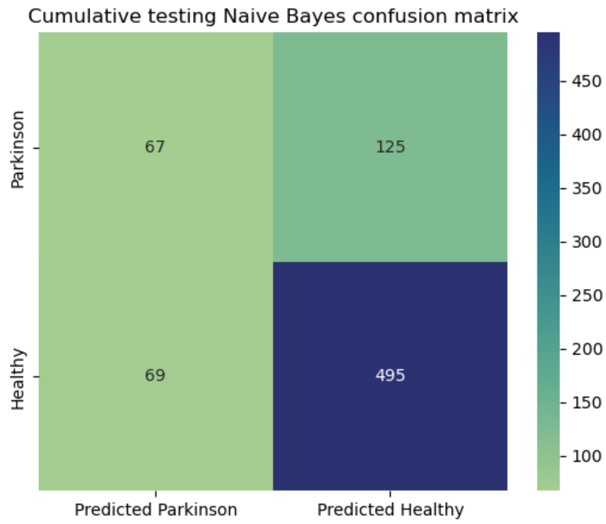
	$p(P \mid x_i)$	0.3	0.5	0.7
x_1	0.5548	P	P	N
x_2	0.3803	P	N	N
x_3	0.2735	N	N	N
	Accuracy:	1	$\frac{2}{3}$	$\frac{1}{3}$

Pela análise da tabela obtida, podemos claramente verificar que $\theta = 0.3$ é o threshold mais indicado para o classificador Bayesiano aprendido, visto que, apresenta a maior accuracy de entre os três thresholds (accuracy = 1).

II. Programming and critical analysis

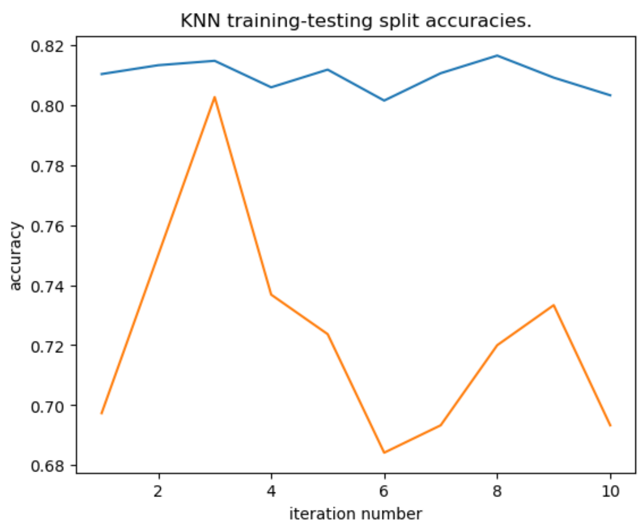
5) [3v] Using sklearn, considering a 10-fold stratified cross validation (random=0), plot the cumulative testing confusion matrices of kNN (uniform weights, k = 5, Euclidean distance) and Naïve Bayes (Gaussian assumption). Use all remaining classifier parameters as default.

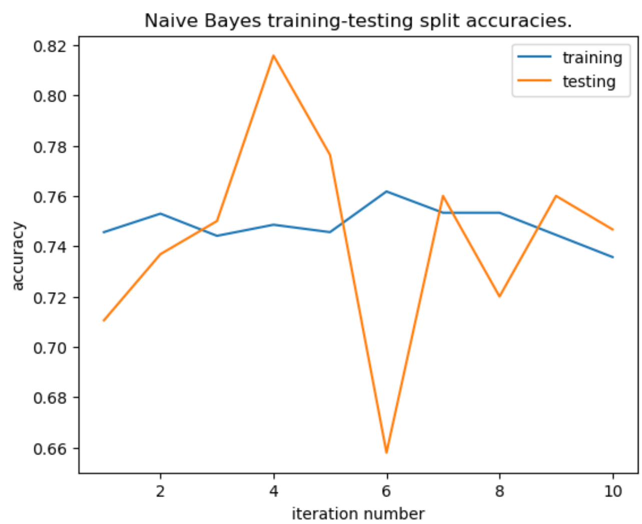




6) [2v] Using scipy, test the hypothesis "kNN is statistically superior to Naïve Bayes regarding accuracy", asserting whether is true.

Non-Normalized



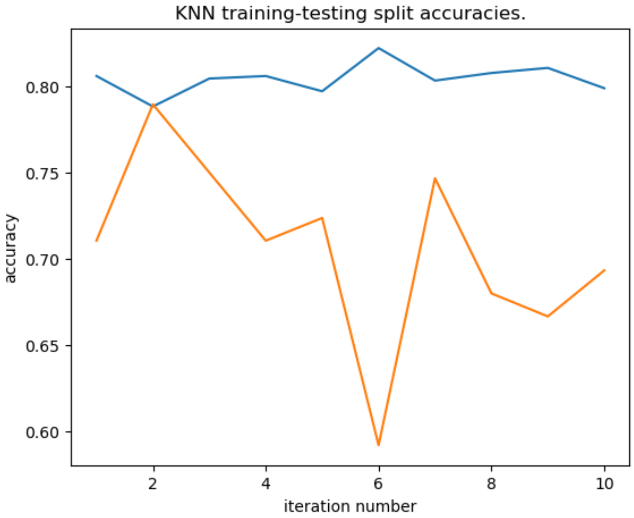


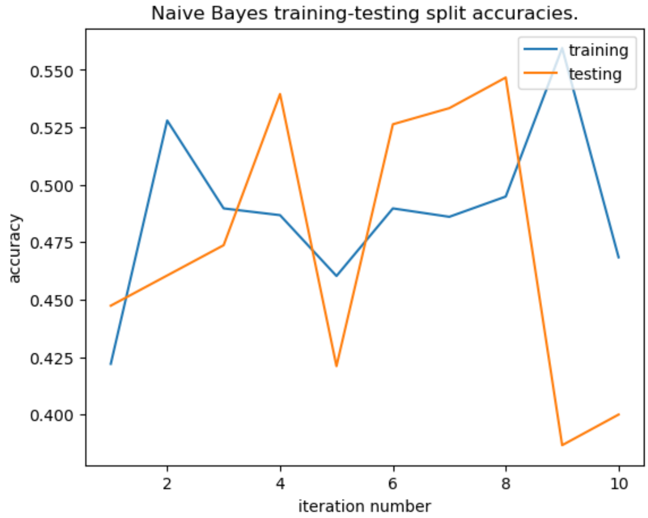

```
KNN average train accuracy: 0.8096706832512741;
KNN average test accuracy: 0.7234736842105263;
Naive Bayes average train accuracy: 0.7485313552733869;
Naive Bayes average test accuracy: 0.7434035087719298;
pvalue_train: 3.411262553853029e-09,    pvalue_test: 0.17910460024968827.
```

Aprendizagem 2022/23

Homework II – Group 018

Normalized*






```
KNN average train accuracy: 0.8043793728945323;
KNN average test accuracy: 0.7062982456140351;
Naive Bayes average train accuracy: 0.4885281160922519;
Naive Bayes average test accuracy: 0.4735087719298246;
pvalue_train: 7.653005756230356e-10,    pvalue_test: 1.1521851431962293e-05.
```

Assumindo a hipótese:

- $H_0 =$ "$as\ average\ accuracies\ para\ ambos\ os\ modelos\ de\ classificação\ são\ iguais.$"
- $H_1 = \neg H_0$

Para valores tanto não normalizados como normalizados, podemos ver que ambos os $pvalue\_train$ são inferiores a 0.05 ($pvalue\_train_{non-norm} \simeq 3.41 \times 10^{-9}$ e $pvalue\_train_{norm} \simeq 7.65 \times 10^{-10}$), rejeitamos a hipótese nula (H_0), e, analisando os valores de average training accuracy obtidos para ambos os modelos, podemos então concluir que em termos de accuracy de treino, KNN é estatisticamente superior a Naive Bayes (norm: $0.81 > 0.74$; non-norm: $0.80 > 0.49$).

Por outro lado, para o $pvalue\_test$ já temos conclusões diferentes dependendo da normalização ou não dos valores: para valores normalizados, rejeitamos também a hipótese nula (H_0) ($pvalue\_test_{norm} \simeq 1.15 \times 10^{-5} < 0.05$); já para não normalizados, não podemos concluir nada sobre a accuracy de teste (uma vez que $pvalue\_test_{non-norm} \simeq 0.179 > 0.05$).

Logo, pela análise dos valores obtidos para a average test accuracy, reparamos que kNN é ligeiramente melhor do que Naive Bayes quando os valores são normalizados. Quando estes não são normalizados, notamos o contrário, Naive Bayes torna-se ligeiramente melhor do que KNN em termos de accuracy de teste.

Note-se que a normalização beneficia a classificação por kNN, uma vez que estamos a retirar a dominância que certas features (que teriam distâncias muito elevadas) poderão ter sobre os resultados finais, tornando-os assim mais equitativos. Sem ter os dados normalizados, podemos claramente observar esta fraqueza que o kNN apresenta.

7) [2v] Enumerate three possible reasons that could underlie the observed differences in predictive accuracy between *k*NN and Naïve Bayes.

Considerando a hipótese como não rejeitada:

- Independência entre variáveis: o Naïve Bayes assume que todas as *features* são independentes entre si, algo que já não acontece com kNN, que acaba por ser mais favorável nos casos com muitas variáveis, entre as quais podem existir dependências e associações (como aparenta ser o caso).
- Número de variáveis [≈750]: a ideia anterior tem ainda mais relevância tendo em conta o elevado número de features deste *dataset*, o que pode justificar as *accuracies* mais baixas do Naïve Bayes.
- Número de vizinhos [k=5]: a escolha do número de vizinhos mais próximos a comparar é bastante importante para a *performance* do kNN (para além das funções de distância e de peso usadas): se for demasiado pequeno, pode causar overfitting; se for demasiado grande, pode eliminar detalhes importantes e exceder na *smoothness* dos resultados. Com isto, k=5 parece ter sido ideal e contribuído para uma boa *overall performance* do modelo.

Aprendizagem 2022/23
Homework II – Group 018

III. APPENDIX

* No código descrito em "Appendix", colocou-se em comentário (a verde) as alterações com valores normalizados.

```
# Import wall
from scipy.io.arff import loadarff

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score
#from sklearn.preprocessing import normalize

from scipy.stats import ttest_rel

# Load and prepare data.
data = loadarff('pd_speech.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

x = df.drop("class", axis=1)
y = np.ravel(df['class'])

# -------------------------------------------------------- #

# Creates the cross-validation object we'll be using:
# - Stratified K fold cross-validation, k = 5.
folds = 10
skf_cv = StratifiedKFold(n_splits = folds, random_state = 0, shuffle = True)

# Creates the classifier objects we'll be using:
# - KNN classifier, k = 5 and using euclidean distance (p = 2);
# - Naive Bayes classifier (Gaussian assumption).
k = 5
knn_clf = KNeighborsClassifier(n_neighbors = k, weights = "uniform", p = 2)
nb_clf = GaussianNB()
```

```
# Question 7:
# Create the cumulative confusion matrices.
cumulative_knn_cf_matrix = np.zeros(shape = (2, 2))
cumulative_nb_cf_matrix = np.zeros(shape = (2, 2))


# Question 6:
# training and testing accuracies for both classifiers.
knn_acc_train, knn_acc_test = [], []
nb_acc_train, nb_acc_test = [], []


# Generate indices to split data into training and test sets.
for train_index, test_index in skf_cv.split(x, y):
    # Generate the train and test splits for our data.
    # x_train, x_test = normalize(x.iloc[train_index]), normalize(x.iloc[test_index])
    x_train , x_test = x.iloc[train_index], x.iloc[test_index]
    y_train , y_test = y[train_index], y[test_index]


    # Fit both knn and NB Gaussian classifiers according to x_train and y_train.
    knn = knn_clf.fit(x_train, y_train)
    nb = nb_clf.fit(x_train, y_train)

    # Perform classification on the array of test values.
    knn_pred = knn_clf.predict(x_test)
    nb_pred = nb_clf.predict(x_test)

    # Generate the confusion matrices for knn and NB predictions.
    knn_cf_matrix = confusion_matrix(y_test, knn_pred)
    nb_cf_matrix = confusion_matrix(y_test, nb_pred)

    # Question 5:
    # Add this iteration's confusion matrices to the cumulative ones.
    cumulative_knn_cf_matrix = np.add(cumulative_knn_cf_matrix, knn_cf_matrix)
    cumulative_nb_cf_matrix = np.add(cumulative_nb_cf_matrix, nb_cf_matrix)

    # Question 6:
    # Perform classification on the array of train values.
    knn_pred_train = knn_clf.predict(x_train)
    nb_pred_train = nb_clf.predict(x_train)
```

```
    # Get training and test accuracies for both classifiers.
    knn_acc_train.append(accuracy_score(y_train, knn_pred_train))
    knn_acc_test.append(accuracy_score(y_test, knn_pred))

    nb_acc_train.append(accuracy_score(y_train, nb_pred_train))
    nb_acc_test.append(accuracy_score(y_test, nb_pred))


# ------------------------------------------------------------------------------ #
# Question 5 plot:
# Plot the confusion matrices.
knn_df = pd.DataFrame(cumulative_knn_cf_matrix,
                      index = ['Parkinson', 'Healthy'],
                      columns = ['Predicted Parkinson', 'Predicted Healthy'])

nb_df = pd.DataFrame(cumulative_nb_cf_matrix,
                     index = ['Parkinson', 'Healthy'],
                     columns = ['Predicted Parkinson', 'Predicted Healthy'])


plt.figure(figsize=(14, 5))

sns.color_palette("crest", as_cmap=True)

plt.subplot(121)
sns.heatmap(knn_df, annot=True, fmt='g', cmap = "crest")
plt.title("Cumulative testing KNN confusion matrix (k = 5)")

plt.subplot(122)
sns.heatmap(nb_df, annot=True, fmt='g', cmap = "crest")
plt.title("Cumulative testing Naive Bayes confusion matrix")
# ------------------------------------------------------------------------------ #
```

```python
# Question 6:
# Using the following results from the previous code:
# - knn_acc_train, knn_acc_test
# - nb_acc_train, nb_acc_test

print("KNN average train accuracy: " + str(np.average(knn_acc_train)) + "\n" + \
      "KNN average test accuracy: " + str(np.average(knn_acc_test)) + "\n" + \
      "Naive Bayes average train accuracy: " + str(np.average(nb_acc_train)) + "\n" + \
      "Naive Bayes average test accuracy: " + str(np.average(nb_acc_test)))

pvalue_train = ttest_rel(knn_acc_train, nb_acc_train)
pvalue_test = ttest_rel(knn_acc_test, nb_acc_test)

print("pvalue_train: " + str(pvalue_train) + "\n" + \
      "pvalue_test: " + str(pvalue_test))

# ---------------------------------------------------------------------------- #
# Question 6 plot:
# Plot the accuracies.
x = np.arange(1, folds + 1)

plt.figure(figsize=(14, 5))

plt.subplot(121)
plt.plot(x, knn_acc_train, label = "training")
plt.plot(x, knn_acc_test, label = "testing")
plt.title("KNN training-testing split accuracies.")
plt.ylabel("accuracy")
plt.xlabel("iteration number")

plt.subplot(122)
plt.plot(x, nb_acc_train, label = "training")
plt.plot(x, nb_acc_test, label = "testing")
plt.title("Naive Bayes training-testing split accuracies.")
plt.ylabel("accuracy")
plt.xlabel("iteration number")

plt.legend(loc = "upper right")

plt.show()
# ---------------------------------------------------------------------------- #
```