

Aprendizagem 2023/24

Homework II

I. Pen-and-paper [13v]

Consider the following dataset:

| D | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 |
|-----|-------|-------|-------|-------|-------|-------|
| X1 | 0.24 | 0.36 | 1 | 1 | 0 | A |
| X2 | 0.16 | 0.48 | 1 | 0 | 1 | A |
| X3 | 0.32 | 0.72 | 0 | 1 | 2 | A |
| X4 | 0.54 | 0.11 | 0 | 0 | 1 | B |
| X5 | 0.66 | 0.39 | 0 | 0 | 0 | B |
| X6 | 0.76 | 0.28 | 1 | 0 | 2 | B |
| X7 | 0.41 | 0.53 | 0 | 1 | 1 | B |
| X8 | 0.38 | 0.52 | 0 | 1 | 0 | A |
| X9 | 0.42 | 0.59 | 0 | 1 | 1 | B |

- Consider x_1-x_7 to be training observations, x_8-x_9 to be testing observations, y_1-y_5 to be input variables and y_6 to be the target variable.
Hint: you can use `scipy.stats.multivariate_normal` for multivariate distribution calculus
 - [3.5v] Learn a Bayesian classifier assuming: i) $\{y_1, y_2\}$, $\{y_3, y_4\}$ and $\{y_5\}$ sets of independent variables (e.g., $y_1 \perp y_3$ yet $y_1 \not\perp y_2$), and ii) $y_1 \times y_2 \in \mathbb{R}^2$ is normally distributed. Show all parameters (distributions and priors for subsequent testing).

Homework 2:
Bayesian and Lazy Learning
 I. Pen-and-Paper

Exercício 1)
Alínea a)

Bayes Rule: $p(h|D) = \frac{p(D|h) \cdot p(h)}{p(D)}$

$p(h)$: priors $\rightarrow p(A) = \frac{3}{7}$ $p(B) = \frac{4}{7}$

$p(D|h)$: y_3, y_4 and y_5 are sets of independent variables.
Probability Mass Functions of independent variables.

$p(y_3=0, y_4=0|A) = 0$ $p(y_3=0, y_4=0|B) = \frac{1}{2}$ $p(y_5=0|A) = \frac{1}{3}$
 $p(y_3=0, y_4=1|A) = \frac{1}{3}$ $p(y_3=0, y_4=1|B) = \frac{1}{4}$ $p(y_5=1|A) = \frac{1}{3}$
 $p(y_3=1, y_4=0|A) = \frac{1}{3}$ $p(y_3=1, y_4=0|B) = \frac{1}{4}$ $p(y_5=2|A) = \frac{1}{3}$
 $p(y_3=1, y_4=1|A) = \frac{1}{3}$ $p(y_3=1, y_4=1|B) = 0$ $p(y_5=0|B) = \frac{1}{4}$
 $p(y_5=1|B) = \frac{1}{2}$
 $p(y_5=2|B) = \frac{1}{4}$

Probability Density Functions y_1, y_2 are dependent variables and $y_1, y_2 \in \mathbb{R}^2$ is normally distributed.

$\mu_{y_1, y_2|A} = \frac{1}{3} \left(\begin{bmatrix} 0.24 \\ 0.36 \end{bmatrix} + \begin{bmatrix} 0.16 \\ 0.48 \end{bmatrix} + \begin{bmatrix} 0.32 \\ 0.72 \end{bmatrix} \right) = \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}$

$$\sum_{y_1, y_2|A} = \begin{bmatrix} \sum_{i=1}^n \sum_{j=1}^n \end{bmatrix}$$

$$\cdot \sum_{i=1}^n = \frac{1}{3-1} \left((0.24-0.24)^2 + (0.16-0.24)^2 + (0.32-0.24)^2 \right) = 0.0064$$

$$\sum_{i=1}^n = \sum_{j=1}^n = \frac{1}{3-1} \left((0.24-0.24)(0.36-0.52) + (0.16-0.24)(0.48-0.52) + (0.32-0.24)(0.72-0.52) \right) = 0.0096$$

$$\cdot \sum_{i=1}^n = \frac{1}{3-1} \left((0.38-0.52)^2 + (0.42-0.52)^2 + (0.41-0.52)^2 \right) = 0.0336$$

$$\sum_{y_1, y_2|A} = \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix}$$

$$\mu_{y_1, y_2|B} = \frac{1}{4} \left(\begin{bmatrix} 0.54 \\ 0.11 \end{bmatrix} + \begin{bmatrix} 0.66 \\ 0.39 \end{bmatrix} + \begin{bmatrix} 0.76 \\ 0.28 \end{bmatrix} + \begin{bmatrix} 0.41 \\ 0.53 \end{bmatrix} \right) = \begin{bmatrix} 0.5925 \\ 0.3275 \end{bmatrix}$$

$$\sum_{y_1, y_2|B} = \begin{bmatrix} \sum_{i=1}^n \sum_{j=1}^n \end{bmatrix}$$

$$\cdot \sum_{i=1}^n = \frac{1}{4-1} \left((0.54-0.5925)^2 + (0.66-0.5925)^2 + (0.76-0.5925)^2 + (0.41-0.5925)^2 \right) = 0.02289$$

$$\cdot \sum_{i=1}^n = \sum_{j=1}^n = \frac{1}{4-1} \left((0.54-0.5925)(0.11-0.3275) + (0.66-0.5925)(0.39-0.3275) + (0.76-0.5925)(0.28-0.3275) + (0.41-0.5925)(0.53-0.3275) \right) = -0.009758$$

$$\cdot \sum_{i=1}^n = \frac{1}{4-1} \left((0.11-0.3275)^2 + (0.39-0.3275)^2 + (0.28-0.3275)^2 + (0.53-0.3275)^2 \right) = 0.03149$$

Aprendizagem 2023/24
Homework II

$$\begin{aligned} & \mathcal{N}(\mu_{Y_1, Y_2, Y_3, Y_4, Y_5} | A = 1, \Sigma_{Y_1, Y_2, Y_3, Y_4, Y_5} | A = 1) \\ & \mathcal{N}(\mu_{Y_1, Y_2, Y_3, Y_4, Y_5} | B = 0, \Sigma_{Y_1, Y_2, Y_3, Y_4, Y_5} | B = 0) \\ & P(D): \\ & P(Y_3=0; Y_4=0) = \frac{2}{7} \quad P(Y_3=0; Y_4=1) = \frac{2}{7} \quad P(Y_3=1; Y_4=0) = \frac{2}{7} \quad P(Y_3=1; Y_4=1) = \frac{1}{7} \\ & P(Y_5=0) = \frac{2}{7} \quad P(Y_5=1) = \frac{3}{7} \quad P(Y_5=2) = \frac{2}{7} \\ & \mu_{Y_1, Y_2, Y_3, Y_4, Y_5} = \frac{1}{7} \left(\begin{bmatrix} 0,24 \\ 0,36 \end{bmatrix} + \begin{bmatrix} 0,16 \\ 0,48 \end{bmatrix} + \begin{bmatrix} 0,32 \\ 0,72 \end{bmatrix} + \begin{bmatrix} 0,54 \\ 0,81 \end{bmatrix} + \begin{bmatrix} 0,66 \\ 0,99 \end{bmatrix} + \begin{bmatrix} 0,76 \\ 0,91 \end{bmatrix} \right) = \begin{bmatrix} 0,4414 \\ 0,41 \end{bmatrix} \\ & \Sigma_{Y_1, Y_2, Y_3, Y_4, Y_5} = \begin{bmatrix} \Sigma_{00} & \Sigma_{10} \\ \Sigma_{01} & \Sigma_{11} \end{bmatrix} \\ & \Sigma_{00} = \frac{1}{7-1} \left((0,24-0,4414)^2 + (0,16-0,4414)^2 + (0,32-0,4414)^2 + (0,54-0,4414)^2 + (0,66-0,4414)^2 + (0,76-0,4414)^2 \right) = 0,04908 \\ & \Sigma_{10} = \Sigma_{01} = \frac{1}{7-1} \left((0,24-0,4414)(0,36-0,41) + (0,16-0,4414)(0,48-0,41) + (0,32-0,4414)(0,72-0,41) + (0,54-0,4414)(0,81-0,41) + (0,66-0,4414)(0,99-0,41) + (0,76-0,4414)(0,91-0,41) \right) = -0,02107 \\ & \Sigma_{11} = \frac{1}{7-1} \left((0,36-0,41)^2 + (0,48-0,41)^2 + (0,72-0,41)^2 + (0,81-0,41)^2 + (0,99-0,41)^2 + (0,91-0,41)^2 \right) = 0,03753 \end{aligned}$$

$$\begin{aligned} & \Sigma_{Y_1, Y_2, Y_3, Y_4, Y_5} = \begin{bmatrix} 0,04908 & -0,02107 \\ -0,02107 & 0,03753 \end{bmatrix} \\ & \mathcal{N}(\mu_{Y_1, Y_2, Y_3, Y_4, Y_5} = \begin{bmatrix} 0,4414 \\ 0,41 \end{bmatrix}, \Sigma_{Y_1, Y_2, Y_3, Y_4, Y_5} = \begin{bmatrix} 0,04908 & -0,02107 \\ -0,02107 & 0,03753 \end{bmatrix}) \end{aligned}$$

b. [2.5v] Under a MAP assumption, classify each testing observation showing all your calculus.

Alinea b)

$$\begin{aligned} & X_B: P(Y_1=0,38; Y_2=0,52; Y_3=0; Y_4=1; Y_5=0 | A) = \\ & = P(Y_1=0,38; Y_2=0,52 | A) \cdot P(Y_3=0; Y_4=1 | A) \cdot P(Y_5=0 | A) \cdot P(A), \\ & \cdot P(Y_3=0; Y_4=1 | A) = \frac{1}{3}, \quad P(Y_5=0 | A) = \frac{1}{3}, \quad P(A) = \frac{3}{7} \quad \text{and} \\ & P(Y_1=0,38; Y_2=0,52 | A) = \mathcal{N}\left(\begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix} \middle| \mu_{Y_1, Y_2} | A, \Sigma_{Y_1, Y_2} | A\right) \\ & \text{Using the calculations of the previous exercise:} \\ & \left| \Sigma_{Y_1, Y_2} | A \right| = 0,0064 \cdot 0,0336 - 0,0096^2 = 1,2288 \times 10^{-4} \\ & \Sigma_{Y_1, Y_2} | A = \frac{1}{1,2288 \times 10^{-4}} \begin{bmatrix} 0,0336 & -0,0096 \\ -0,0096 & 0,0064 \end{bmatrix} = \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,0833 \end{bmatrix} \\ & \text{Multivariate Gaussian distribution in m-dimensional spaces:} \\ & P(x | \mu, \Sigma) = \mathcal{N}(x | \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \\ & P(Y_1=0,38; Y_2=0,52 | A) = \frac{1}{(2\pi)^{\frac{2}{2}} \sqrt{1,2288 \times 10^{-4}}} e^{-\frac{1}{2} \begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix}^T \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,0833 \end{bmatrix}^{-1} \begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix}} \end{aligned}$$

$$\begin{aligned} & = \frac{1}{2\pi \sqrt{1,2288 \times 10^{-4}}} e^{-\frac{1}{2} \begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix}^T \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,0833 \end{bmatrix}^{-1} \begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix}} = \\ & = \frac{1}{2\pi \sqrt{1,2288 \times 10^{-4}}} e^{-0,07} \approx 0,989, \quad \text{therefore:} \\ & \cdot P(Y_1=0,38; Y_2=0,52; Y_3=0; Y_4=1; Y_5=0 | A) = 0,989 \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{3}{7} \approx 0,04689 \\ & \cdot P(Y_1=0,38; Y_2=0,52; Y_3=0; Y_4=1; Y_5=0 | B) = \\ & = P(Y_1=0,38; Y_2=0,52 | B) \cdot P(Y_3=0; Y_4=1 | B) \cdot P(Y_5=0 | B) \cdot P(B), \\ & \cdot P(Y_3=0; Y_4=1 | B) = \frac{1}{4}, \quad P(Y_5=0 | B) = \frac{1}{4}, \quad P(B) = \frac{4}{7} \quad \text{and} \\ & \cdot P(Y_1=0,38; Y_2=0,52 | B) = \mathcal{N}\left(\begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix} \middle| \mu_{Y_1, Y_2} | B, \Sigma_{Y_1, Y_2} | B\right) \\ & \text{Using the calculations of the previous exercise:} \\ & \left| \Sigma_{Y_1, Y_2} | B \right| = 0,02289 \cdot 0,03149 - (-0,009758)^2 = 6,2559 \times 10^{-4} \\ & \Sigma_{Y_1, Y_2} | B = \frac{1}{6,2559 \times 10^{-4}} \begin{bmatrix} 0,03149 & -0,009758 \\ -0,009758 & 0,02289 \end{bmatrix} = \begin{bmatrix} 50,3367 & -15,5981 \\ -15,5981 & 36,5816 \end{bmatrix} \\ & \text{Using scipy.stats.multivariate_normal:} \\ & P(Y_1=0,38; Y_2=0,52 | B) = \mathcal{N}\left(\begin{bmatrix} 0,38 \\ 0,52 \end{bmatrix} \middle| \begin{bmatrix} 0,5225 \\ 0,3275 \end{bmatrix}, \begin{bmatrix} 0,02289 & -0,009758 \\ -0,009758 & 0,03149 \end{bmatrix}\right) \approx 1,962, \\ & \text{therefore:} \\ & \cdot P(Y_1=0,38; Y_2=0,52; Y_3=0; Y_4=1; Y_5=0 | B) = 1,962 \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{4}{7} \approx 0,07008 \\ & \text{Therefore } X_B \text{ is classified as B.} \end{aligned}$$

Aprendizagem 2023/24

Homework II

$X_q: p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|A) =$
 $= p(y_1=0,42; y_2=0,59|A) \cdot p(y_3=0; y_4=1|A) \cdot p(y_5=1|A) \cdot p(A),$
 $\cdot p(y_3=0; y_4=1|A) = \frac{1}{3}, p(y_5=1|A) = \frac{1}{3}, p(A) = \frac{3}{7}$ and
 $\cdot p(y_1=0,42; y_2=0,59|A) = \mathcal{N}\left(\begin{bmatrix} 0,42 \\ 0,59 \end{bmatrix} \middle| \mu_{y_1,y_2|A}, \Sigma_{y_1,y_2|A}\right),$
 Using the calculations of the previous exercise:
 $\left| \Sigma_{y_1,y_2|A} \right| = 1,2288 \times 10^{-4} \sum_{y_1,y_2|A}^{-1} = \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,0833 \end{bmatrix}$
 Using scipy.stats.multivariate_normal:
 $p(y_1=0,42; y_2=0,59|A) = \mathcal{N}\left(\begin{bmatrix} 0,42 \\ 0,59 \end{bmatrix} \middle| \begin{bmatrix} 0,24 \\ 0,52 \end{bmatrix}, \begin{bmatrix} 0,0064 & 0,0096 \\ 0,0096 & 0,0336 \end{bmatrix}\right) \approx 0,409,$
 therefore:
 $\cdot p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|A) = 0,409 \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{3}{7} \approx 0,0919$
 $\cdot p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|B) =$
 $= p(y_1=0,42; y_2=0,59|B) \cdot p(y_3=0; y_4=1|B) \cdot p(y_5=1|B) \cdot p(B),$
 $p(y_3=0; y_4=1|B) = \frac{1}{4}, p(y_5=1|B) = \frac{1}{2}, p(B) = \frac{4}{7}$ and
 $\cdot p(y_1=0,42; y_2=0,59|B) = \mathcal{N}\left(\begin{bmatrix} 0,42 \\ 0,59 \end{bmatrix} \middle| \mu_{y_1,y_2|B}, \Sigma_{y_1,y_2|B}\right),$
 Using the calculations of the previous and this exercise:
 $\left| \Sigma_{y_1,y_2|B} \right| = 6,2559 \times 10^{-4} \sum_{y_1,y_2|B}^{-1} = \begin{bmatrix} 50,3367 & 15,5981 \\ 15,5981 & 36,5996 \end{bmatrix}$

→ Using scipy.stats.multivariate_normal:
 $p(y_1=0,42; y_2=0,59|B) = \mathcal{N}\left(\begin{bmatrix} 0,42 \\ 0,59 \end{bmatrix} \middle| \begin{bmatrix} 0,3275 \\ 0,3275 \end{bmatrix}, \begin{bmatrix} 0,0228 & -0,009758 \\ -0,009758 & 0,03149 \end{bmatrix}\right) \approx 1,728,$
 therefore:
 $\cdot p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|B) = 1,728 \cdot \frac{1}{4} \cdot \frac{1}{2} \cdot \frac{4}{7} \approx 0,1235$
 • Therefore X_q is classified as B.

c. [2v] Consider that the default decision threshold of $\theta = 0.5$ can be adjusted according to

$$f(x|\theta) = \begin{cases} A & P(A|x) > \theta \\ B & \text{otherwise} \end{cases}$$

Under a maximum likelihood assumption, what thresholds optimize testing accuracy?

AL'ica c)

$$P(A|x) = \frac{P(x|A) \cdot P(A)}{P(x|A) \cdot P(A) + P(x|B) \cdot P(B)} = \frac{P(x|A)}{P(x|A) + P(x|B)}$$

 Under ML assumption all priors are the same: $P(A) = P(B)$
 X_g normalization:
 $P(A|y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0) =$
 $= \frac{P(y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0|A)}{P(y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0|A) + P(y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0|B)}$
 • From the previous exercise:
 $\cdot p(y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0|A) \approx 0,04689$
 $\cdot p(y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0|B) \approx 0,07088$, therefore:
 $P(A|y_1=0,38; y_2=0,52; y_3=0; y_4=1; y_5=0) \approx \frac{0,04689}{0,04689 + 0,07088} \approx 0,40087$

→ X_q normalization:
 $P(A|y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1) =$
 $= \frac{P(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|A)}{P(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|A) + P(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|B)}$
 • From the previous exercise:
 $\cdot p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|A) \approx 0,0919$
 $\cdot p(y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1|B) \approx 0,1235$, therefore:
 $P(A|y_1=0,42; y_2=0,59; y_3=0; y_4=1; y_5=1) \approx \frac{0,0919}{0,0919 + 0,1235} \approx 0,43449$
 • Conclusion:
 The thresholds that maximize test accuracy are the thresholds in the range $[0,13449; 0,40087]$.

Aprendizagem 2023/24
Homework II

2. Let y_1 be the target numeric variable, y_2 - y_6 be the input variables where y_2 is binarized under an equal-width (equal-range) discretization. For the evaluation of regressors, consider a 3-fold cross-validation over the full dataset (x_1 - x_9) without shuffling the observations.
- a. [1v] Identify the observations and features per data fold after the binarization procedure.

Exercício 2)

Alínea a)

. Binarization of y_2 :

$$y_2 = \begin{cases} 0, & x_i < 0,5 \\ 1, & x_i \geq 0,5 \end{cases}, \text{ therefore: } y_2 = [0, 0, 1, 0, 0, 0, 1, 1, 1]$$

. On Fold 1 the training data subset is from x_1 to x_6 and the testing data subset is from x_7 to x_9 ;

. On Fold 2 the training data subset is from x_1 to x_3 and x_7 to x_9 . The testing data subset is from x_4 to x_6 ;

. On Fold 3 the training data subset is from x_4 to x_9 and the testing data subset is from x_1 to x_3 .

. Therefore:

| D | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 | Fold 1 | Fold 2 | Fold 3 |
|-------|-------|-------|-------|-------|-------|-------|---------------|---------------|---------------|
| x_1 | 0,24 | 0 | 1 | 1 | 0 | A | Training data | Training data | Testing data |
| x_2 | 0,16 | 0 | 1 | 0 | 1 | A | Training data | Training data | Testing data |
| x_3 | 0,32 | 1 | 0 | 1 | 2 | A | Training data | Training data | Testing data |
| x_4 | 0,54 | 0 | 0 | 0 | 1 | B | Testing data | Training data | Training data |
| x_5 | 0,66 | 0 | 0 | 0 | 0 | B | Testing data | Training data | Training data |
| x_6 | 0,76 | 0 | 1 | 0 | 2 | B | Testing data | Training data | Training data |
| x_7 | 0,41 | 1 | 0 | 1 | 1 | B | Testing data | Training data | Training data |
| x_8 | 0,38 | 1 | 0 | 1 | 0 | A | Testing data | Training data | Training data |
| x_9 | 0,42 | 1 | 0 | 1 | 1 | B | Testing data | Training data | Training data |

. The ~~features~~ features are the input variables. Therefore, features are from y_2 to y_6 .

Homework II

- b. [4v] Consider a distance-weighted k NN with $k = 3$, Hamming distance (d), and $1/d$ weighting. Compute the MAE of this k NN regressor for the 1st iteration of the cross-validation (i.e. train observations have the lower indices).

Answer b)

If we want to calculate the MAE of this KNN regressor for the 1st iteration of the cross-validation then, our training data subset is from X_1 to X_6 and, our testing data subset is from X_7 to X_8 .

$$H(X, Y) = \sum_{i=1}^n d(X_i, Y_i)$$

X_7 :

| $H(X_7, X_i)$ | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 |
|---------------|-------|-------|-------|-------|-------|-------|
| | 4 | 4 | 2 | 2 | 3 | 4 |

$$d(X_7, X_1) = d(1, 0) + d(0, 1) + d(1, 1) + d(1, 0) + d(B, A) = 4$$

$$d(X_7, X_2) = d(1, 0) + d(0, 1) + d(1, 0) + d(1, 1) + d(B, A) = 4$$

$$d(X_7, X_3) = d(1, 0) + d(0, 0) + d(1, 1) + d(1, 2) + d(B, A) = 3$$

$$d(X_7, X_4) = d(1, 0) + d(0, 0) + d(1, 0) + d(1, 1) + d(B, B) = 2$$

$$d(X_7, X_5) = d(1, 0) + d(0, 0) + d(1, 0) + d(1, 0) + d(B, B) = 3$$

$$d(X_7, X_6) = d(1, 0) + d(0, 1) + d(1, 0) + d(1, 2) + d(B, B) = 3$$

X_8 :

| $H(X_8, X_i)$ | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 |
|---------------|-------|-------|-------|-------|-------|-------|
| | 2 | 4 | 1 | 4 | 3 | 5 |

$$d(X_8, X_1) = d(1, 0) + d(0, 1) + d(1, 1) + d(0, 0) + d(A, A) = 2$$

$$d(X_8, X_2) = d(1, 0) + d(0, 1) + d(1, 0) + d(0, 1) + d(A, A) = 4$$

$$d(X_8, X_3) = d(1, 1) + d(0, 0) + d(1, 1) + d(0, 2) + d(A, A) = 1$$

$$d(X_8, X_4) = d(1, 0) + d(0, 0) + d(1, 0) + d(0, 1) + d(A, B) = 4$$

$$d(X_8, X_5) = d(1, 0) + d(0, 0) + d(1, 0) + d(0, 0) + d(A, B) = 3$$

$$d(X_8, X_6) = d(1, 0) + d(0, 1) + d(1, 0) + d(0, 2) + d(A, B) = 5$$

| X_i | $H(X_q, X_i)$ | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 |
|-------|---------------|-------|-------|-------|-------|-------|-------|
| X_q | | 4 | 4 | 2 | 2 | 3 | 4 |

$$d(X_q, X_1) = d(1, 0) + d(0, 1) + d(1, 1) + d(1, 0) + d(B, A) = 4$$

$$d(X_q, X_2) = d(1, 0) + d(0, 1) + d(1, 0) + d(1, 1) + d(B, A) = 4$$

$$d(X_q, X_3) = d(1, 1) + d(0, 0) + d(1, 1) + d(1, 2) + d(B, A) = 3$$

$$d(X_q, X_4) = d(1, 0) + d(0, 0) + d(1, 0) + d(1, 1) + d(B, B) = 2$$

$$d(X_q, X_5) = d(1, 0) + d(0, 0) + d(1, 0) + d(1, 0) + d(B, B) = 3$$

$$d(X_q, X_6) = d(1, 0) + d(0, 1) + d(1, 0) + d(1, 2) + d(B, B) = 4$$

For X_7 and X_q , the neighbors X_3, X_4 and X_5 were chosen.

For X_8 the neighbors X_1, X_3 and X_5 were chosen.

$$\hat{Z}_1(X_7) = \text{Weighted-mean} \left(\frac{1}{2} \cdot 0.32; \frac{1}{2} \cdot 0.54; \frac{1}{3} \cdot 0.66 \right) =$$

$$= \frac{\frac{1}{2} \cdot 0.32 + \frac{1}{2} \cdot 0.54 + \frac{1}{3} \cdot 0.66}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} \approx 0.4875$$

$$\hat{Z}_1(X_8) = \text{Weighted-mean} \left(\frac{1}{2} \cdot 0.24; \frac{1}{2} \cdot 0.32; \frac{1}{3} \cdot 0.66 \right) =$$

$$= \frac{\frac{1}{2} \cdot 0.24 + \frac{1}{2} \cdot 0.32 + \frac{1}{3} \cdot 0.66}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} \approx 0.36$$

$$\hat{Z}_1(X_q) = \text{Weighted-mean} \left(\frac{1}{2} \cdot 0.32; \frac{1}{2} \cdot 0.54; \frac{1}{3} \cdot 0.66 \right) =$$

$$= \frac{\frac{1}{2} \cdot 0.32 + \frac{1}{2} \cdot 0.54 + \frac{1}{3} \cdot 0.66}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} \approx 0.4875, \text{ therefore: } \frac{Z_1}{\hat{Z}_1} \begin{array}{c|c|c} 0.49 & 0.38 & 0.42 \\ \hline 0.4875 & 0.36 & 0.4875 \end{array}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Z_i - \hat{Z}_i|, \text{ therefore: } \text{MAE} = \frac{1}{3} (|0.49 - 0.4875| + |0.38 - 0.36| + |0.42 - 0.4875|) \approx 0.058$$

Aprendizagem 2023/24
Homework II

I. Programming and critical analysis [7v]

Considering the *column_diagnosis.arff* dataset available at the course webpage's homework tab. Using *sklearn*, apply a 10-fold stratified cross-validation with shuffling (*random_state=0*) for the assessment of predictive models along this section.

- 1) [3v] Compare the performance of *k*NN with $k = 5$ and naïve Bayes with Gaussian assumption (consider all remaining parameters for each classifier as *sklearn*'s default):

```
import pandas as pd
from scipy.io.arff import loadarff
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Read 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']

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

knn_accuracies, nb_accuracies = [], []
knn_classifier = KNeighborsClassifier(n_neighbors=5)
nb_classifier = GaussianNB()

# Iterate through each fold of stratified cross-validation
for train_index, test_index in stratified_cv.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Train both classifiers
    knn_classifier.fit(X_train, y_train)
    nb_classifier.fit(X_train, y_train)

    # Predict and evaluate kNN
    knn_pred = knn_classifier.predict(X_test)
    knn_accuracy = accuracy_score(y_test, knn_pred)
    knn_accuracies.append(knn_accuracy)

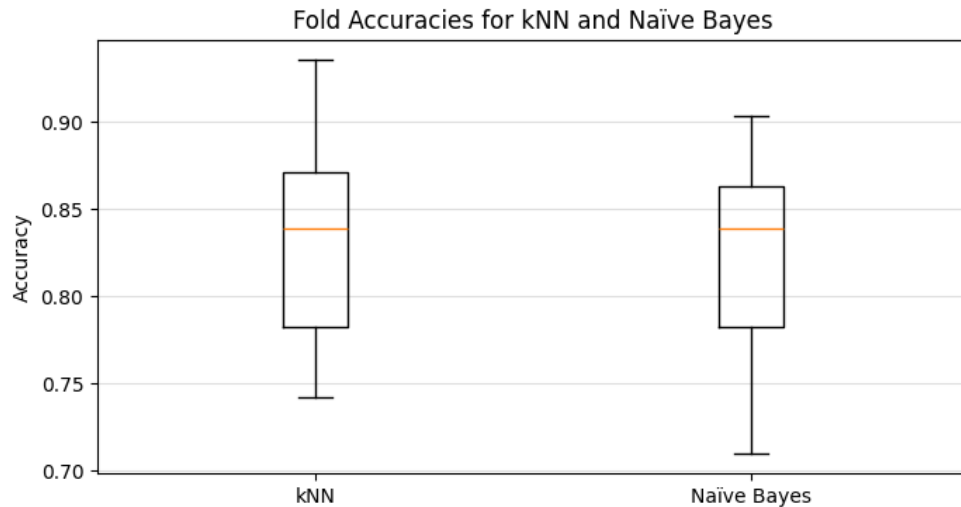
    # Predict and evaluate Gaussian Naïve Bayes
    nb_pred = nb_classifier.predict(X_test)
    nb_accuracy = accuracy_score(y_test, nb_pred)
    nb_accuracies.append(nb_accuracy)
```

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

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 4))
plt.boxplot([knn_accuracies, nb_accuracies], labels=['kNN', 'Naïve Bayes'])
plt.title('Fold Accuracies for kNN and Naïve Bayes')
plt.ylabel('Accuracy')
plt.grid(axis='y', alpha=0.4)
plt.show()
```

Aprendizagem 2023/24 Homework II



- b. Using *scipy*, test the hypothesis “kNN is statistically superior to naïve Bayes regarding accuracy”, asserting whether is true.

```
from scipy import stats

# H0: kNN is statistically equal to Naïve Bayes regarding accuracy
# H1: kNN is statistically superior to Naïve Bayes regarding accuracy
t_statistic, p_value = stats.ttest_rel(knn_accuracies, nb_accuracies, alternative='greater')

alpha = 0.05 # Significance level
if p_value < alpha:
    print('Para níveis de significância até 0.05, a hipótese nula (kNN é estatisticamente igual a Naïve Bayes em termos de precisão) \né rejeitada, ou seja, "kNN is statistically superior to Naïve Bayes regarding accuracy" confirma-se.')
else:
    print('Para níveis de significância até 0.05, a hipótese nula (kNN é estatisticamente igual a Naïve Bayes em termos de precisão), \nnão pode ser rejeitada, ou seja, "kNN is statistically superior to Naïve Bayes regarding accuracy" é falso.')
```

Output:

Para níveis de significância até 0.05, a hipótese nula (kNN é estatisticamente igual a Naïve Bayes em termos de precisão) não pode ser rejeitada, ou seja, "kNN is statistically superior to Naïve Bayes regarding accuracy" é falso.

Homework II

- 2) [2.5v] Consider two k NN 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.

```
import numpy as np
from sklearn.metrics import confusion_matrix

# Initialize confusion matrices
knn1_cumulative = np.array([[0,0,0], [0,0,0], [0,0,0]])
knn5_cumulative = np.array([[0,0,0], [0,0,0], [0,0,0]])
classes = ["Hernia", "Normal", "Spondylolisthesis"]

knn1 = KNeighborsClassifier(n_neighbors=1, weights='uniform', metric='euclidean')
knn5 = KNeighborsClassifier(n_neighbors=5, weights='uniform', metric='euclidean')

# Iterate through each fold of stratified cross-validation
for train_index, test_index in stratified_cv.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Train both classifiers
    knn1.fit(X_train, y_train)
    knn5.fit(X_train, y_train)

    # Make predictions and calculate confusion matrix for kNN1
    knn1_pred = knn1.predict(X_test)
    knn1_cm = confusion_matrix(y_test, knn1_pred, labels=classes)
    knn1_cumulative += np.array(knn1_cm)

    # Make predictions and calculate confusion matrix for kNN5
    knn5_pred = knn5.predict(X_test)
    knn5_cm = confusion_matrix(y_test, knn5_pred, labels=classes)
    knn5_cumulative += np.array(knn5_cm)

# Calculate the difference between cumulative confusion matrices
difference_matrix = knn1_cumulative - knn5_cumulative

print("Difference between cumulative confusion matrices (kNN1 - kNN5):\n\n",
      pd.DataFrame(difference_matrix, index=classes, columns=classes),
      "\n\nNote:\tColumns are predicted values\tLines are test values")
```

Output:

Difference between cumulative confusion matrices (kNN1 - kNN5):

| | Hernia | Normal | Spondylolisthesis |
|-------------------|--------|--------|-------------------|
| Hernia | -2 | 2 | 0 |
| Normal | -5 | 2 | 3 |
| Spondylolisthesis | 0 | 1 | -1 |

Note: Columns are predicted values
Lines are test values

Comentário sobre os resultados obtidos:

- Para a classe 'Hernia', o modelo com $k=5$ previu corretamente mais 2 instâncias, enquanto o modelo com $k=1$ previu incorretamente mais 2 instâncias de 'Hernia' como 'Normal'.
- Para a classe 'Normal', o modelo com $k=1$ previu corretamente mais 2 instâncias e previu incorretamente mais 3 instâncias de 'Normal' como 'Spondylolisthesis', no entanto, o modelo com $k=5$ previu incorretamente mais 5 instâncias de 'Normal' como 'Hernia'.
- Para a classe 'Spondylolisthesis', o modelo com $k=5$ previu corretamente mais 1 instância, enquanto o modelo com $k=1$ previu incorretamente mais 1 instância de 'Spondylolisthesis' como 'Normal'.

Assim, pode-se concluir que o modelo com $k=5$ teve melhor performance para as classes 'Hernia' e 'Spondylolisthesis', enquanto o modelo com $k=1$ teve melhor performance para a classe 'Normal'.

Aprendizagem 2023/24

Homework II

- 3) [1.5v] Considering the unique properties of *column_diagnosis*, identify three possible difficulties of naïve Bayes when learning from the given dataset.
- Assume independência condicional entre as variáveis. Essa suposição pode ser muito restritiva, pois nem sempre é verdadeira.
 - Assume que os dados seguem uma distribuição gaussiana (normal), o que pode não ser apropriado caso os dados não obedeçam a essa distribuição.
 - O uso de um conjunto de observações limitado (pode levar a estimativas inadequadas ou probabilidades nulas).

END