

Aprendizagem 2024  
Homework II – Group 44  
(ist1106827, ist107245)

### Part I: Pen and paper

We collected four positive (P) observations,  $\{x_1 = (A, 0), x_2 = (B, 1), x_3 = (A, 1), x_4 = (A, 0)\}$  and four negative (N) observations,  $\{x_5 = (B, 0), x_6 = (B, 0), x_7 = (A, 1), x_8 = (B, 1)\}$ . Consider the problem of classifying observations as positive or negative.

1. Compute the F1-measure of a kNN with  $k = 5$  and Hamming distance using a leave-one-out evaluation schema. Show all calculus.

kNN with  $k=5$   
 Hamming distance:  $\sum_{i=1}^p a_i \neq b_i$   
 $x_1 = [A, 0]$   
 $x_2 = [B, 0]$       Hamming  $[x_1, x_2] = 1$       ] para features categóricas

1. Calcular a distância de  $x_i$  a todas as observações do dataset

Hamming  $[x_1, x_2] = 2$       Hamming  $[x_1, x_5] = 1$       Hamming  $[x_1, x_8] = 2$   
 Hamming  $[x_1, x_3] = 1$       Hamming  $[x_1, x_6] = 1$   
 Hamming  $[x_1, x_4] = 0$       Hamming  $[x_1, x_7] = 1$

2. Escolher os  $k=5$  mais próximos

$\underbrace{x_3, x_4, x_5, x_6, x_7}_P$        $\underbrace{x_2, x_8}_N$

3. O output de  $x_i$  é moda do output dos  $k$  vizinhos

output( $x_1$ ) = mode(P, P, N, N, N) = N

4. Repetir para todos

$d(x_2, x_1) = 2$        $d(x_2, x_5) = 1$        $d(x_2, x_8) = 0$   
 $d(x_2, x_3) = 1$        $d(x_2, x_6) = 1$   
 $d(x_2, x_4) = 2$        $d(x_2, x_7) = 1$

5 mais próximos:  $\underbrace{x_3, x_5, x_6, x_7, x_8}_P$       output( $x_2$ ) = N

$d(x_3, x_1) = 1$        $d(x_3, x_5) = 2$        $d(x_3, x_8) = 1$   
 $d(x_3, x_2) = 1$        $d(x_3, x_6) = 2$   
 $d(x_3, x_4) = 1$        $d(x_3, x_7) = 0$

5 mais próximos:  $\underbrace{x_1, x_2, x_4, x_7, x_8}_P$       output( $x_3$ ) = P

$d(x_4, x_1) = 0$        $d(x_4, x_5) = 1$        $d(x_4, x_8) = 2$   
 $d(x_4, x_2) = 2$        $d(x_4, x_6) = 1$   
 $d(x_4, x_3) = 1$        $d(x_4, x_7) = 1$

5 mais próximos:  $\underbrace{x_1, x_3, x_5, x_6, x_7}_P$       output( $x_4$ ) = N

$$\begin{aligned} d(x_5, x_1) &= 1 & d(x_5, x_4) &= 1 & d(x_5, x_8) &= 1 \\ d(x_5, x_2) &= 1 & d(x_5, x_6) &= 0 \\ d(x_5, x_3) &= 2 & d(x_5, x_7) &= 2 \end{aligned}$$

5 mais próximos:  $\underbrace{x_1, x_2, x_4}_P, \underbrace{x_6, x_8}_N$

$$\text{Output}(x_5) = P$$

$$\begin{aligned} d(x_6, x_1) &= 1 & d(x_6, x_4) &= 1 & d(x_6, x_8) &= 1 \\ d(x_6, x_2) &= 1 & d(x_6, x_5) &= 0 \\ d(x_6, x_3) &= 2 & d(x_6, x_7) &= 2 \end{aligned}$$

5 mais próximos:  $\underbrace{x_1, x_2, x_4}_P, \underbrace{x_5, x_8}_N$

$$\text{Output}(x_6) = P$$

$$\begin{aligned} d(x_7, x_1) &= 1 & d(x_7, x_4) &= 1 & d(x_7, x_8) &= 1 \\ d(x_7, x_2) &= 1 & d(x_7, x_5) &= 2 \\ d(x_7, x_3) &= 0 & d(x_7, x_6) &= 2 \end{aligned}$$

5 mais próximos:  $\underbrace{x_1, x_2, x_3, x_4}_P, \underbrace{x_8}_N$

$$\text{Output}(x_7) = P$$

$$\begin{aligned} d(x_8, x_1) &= 2 & d(x_8, x_4) &= 2 & d(x_8, x_7) &= 1 \\ d(x_8, x_2) &= 0 & d(x_8, x_5) &= 1 \\ d(x_8, x_3) &= 1 & d(x_8, x_6) &= 1 \end{aligned}$$

5 mais próximos:  $\underbrace{x_2, x_3}_P, \underbrace{x_5, x_6, x_7}_N$

$$\text{Output}(x_8) = N$$

	real	previsto
$x_1$	P	N
$x_2$	P	N
$x_3$	P	P
$x_4$	P	N
$x_5$	N	P
$x_6$	N	P
$x_7$	N	P
$x_8$	N	N

	previsto	
	P	N
reais P	1	3
reais N	3	1

	Previsto	
	P	N
P	TP	FN
N	FP	TN

$$\text{recall} = \frac{TP}{TP+FN} = \frac{1}{1+3} = \frac{1}{4}$$

$$\text{precision} = \frac{TP}{TP+FP} = \frac{1}{1+3} = \frac{1}{4}$$

T → true  
F → false

$$F1\text{-measure} = 2 \times \frac{\frac{1}{4} \times \frac{1}{4}}{\frac{1}{4} + \frac{1}{4}} = \frac{1}{4} = 0,25$$

2. Propose a new metric (distance) that improves the latter's performance (i.e., the F1-measure) by three fold.

improving the F1-measure by three fold  $\leftarrow$  weighted distance  $K=3$   
 escolhemos que o weight da primeira feature seja maior que o da segunda  
 weighted Hamming Distance  $d(x_i, x_j) = 2w_1 + 1w_2$   
 primeira característica (ou A ou B) segunda característica (ou 1 ou 2)

$x_1$ :

$$\begin{aligned} d(x_1, x_2) &= 2 \times 1 + 1 \times 1 = 3 & d(x_1, x_4) &= 2 \times 0 + 1 \times 0 = 0 & d(x_1, x_6) &= 2 \times 1 + 1 \times 0 = 2 & d(x_1, x_8) &= 2 \times 1 + 1 \times 1 = 3 \\ d(x_1, x_3) &= 2 \times 0 + 1 \times 1 = 1 & d(x_1, x_5) &= 2 \times 1 + 1 \times 0 = 2 & d(x_1, x_7) &= 2 \times 0 + 1 \times 1 = 1 & & = 3 \\ 3 \text{ mais próximos: } x_3, x_4, x_7 & & \text{output}(x_1) &= P & & & & \end{aligned}$$

$x_2$ :

$$\begin{aligned} d(x_2, x_1) &= 2 \times 1 + 1 \times 1 = 3 & d(x_2, x_4) &= 2 \times 1 + 1 \times 1 = 3 & d(x_2, x_6) &= 2 \times 0 + 1 \times 1 = 1 & d(x_2, x_8) &= 2 \times 0 + 1 \times 0 = 0 \\ d(x_2, x_3) &= 2 \times 1 + 1 \times 0 = 2 & d(x_2, x_5) &= 2 \times 0 + 1 \times 1 = 1 & d(x_2, x_7) &= 2 \times 1 + 1 \times 0 = 2 & & = 0 \\ 3 \text{ mais próximos: } x_5, x_6, x_8 & & \text{output}(x_2) &= N & & & & \end{aligned}$$

$x_3$ :

$$\begin{aligned} d(x_3, x_1) &= 2 \times 0 + 1 \times 1 = 1 & d(x_3, x_4) &= 2 \times 0 + 1 \times 1 = 1 & d(x_3, x_6) &= 2 \times 1 + 1 \times 1 = 3 & d(x_3, x_8) &= 2 \times 1 + 1 \times 0 = 2 \\ d(x_3, x_2) &= 2 \times 1 + 1 \times 0 = 2 & d(x_3, x_5) &= 2 \times 1 + 1 \times 1 = 3 & d(x_3, x_7) &= 2 \times 0 + 1 \times 0 = 0 & & = 2 \\ 3 \text{ mais próximos: } x_1, x_4, x_7 & & \text{output}(x_3) &= P & & & & \end{aligned}$$

$x_4$ :

$$\begin{aligned} d(x_4, x_1) &= 2 \times 0 + 1 \times 0 = 0 & d(x_4, x_3) &= 2 \times 0 + 1 \times 1 = 1 & d(x_4, x_6) &= 2 \times 1 + 1 \times 0 = 2 & d(x_4, x_8) &= 2 \times 1 + 1 \times 1 = 3 \\ d(x_4, x_2) &= 2 \times 1 + 1 \times 1 = 3 & d(x_4, x_5) &= 2 \times 1 + 1 \times 0 = 2 & d(x_4, x_7) &= 2 \times 0 + 1 \times 1 = 1 & & = 3 \\ 3 \text{ mais próximos: } x_1, x_3, x_7 & & \text{output}(x_4) &= P & & & & \end{aligned}$$

$x_5$ :

$$\begin{aligned} d(x_5, x_1) &= 2 \times 1 + 1 \times 0 = 2 & d(x_5, x_3) &= 2 \times 1 + 1 \times 1 = 3 & d(x_5, x_6) &= 2 \times 0 + 1 \times 0 = 0 & d(x_5, x_8) &= 2 \times 0 + 1 \times 1 = 1 \\ d(x_5, x_2) &= 2 \times 0 + 1 \times 1 = 1 & d(x_5, x_4) &= 2 \times 1 + 1 \times 0 = 2 & d(x_5, x_7) &= 2 \times 1 + 1 \times 1 = 3 & & = 1 \\ 3 \text{ mais próximos: } x_2, x_6, x_8 & & \text{output}(x_5) &= N & & & & \end{aligned}$$

$x_6$ :

$$\begin{aligned} d(x_6, x_1) &= 2 \times 1 + 1 \times 0 = 2 & d(x_6, x_3) &= 2 \times 1 + 1 \times 1 = 3 & d(x_6, x_5) &= 2 \times 0 + 1 \times 0 = 0 & d(x_6, x_8) &= 2 \times 0 + 1 \times 1 = 1 \\ d(x_6, x_2) &= 2 \times 0 + 1 \times 1 = 1 & d(x_6, x_4) &= 2 \times 1 + 1 \times 0 = 2 & d(x_6, x_7) &= 2 \times 1 + 1 \times 1 = 3 & & = 1 \\ 3 \text{ mais próximos: } x_2, x_5, x_8 & & \text{output}(x_6) &= N & & & & \end{aligned}$$

$x_7$ :

$$\begin{aligned} d(x_7, x_1) &= 2 \times 0 + 1 \times 1 = 1 & d(x_7, x_3) &= 2 \times 0 + 1 \times 0 = 0 & d(x_7, x_5) &= 2 \times 1 + 1 \times 1 = 3 & d(x_7, x_8) &= 2 \times 1 + 1 \times 0 = 2 \\ d(x_7, x_2) &= 2 \times 1 + 1 \times 0 = 2 & d(x_7, x_4) &= 2 \times 0 + 1 \times 1 = 1 & d(x_7, x_6) &= 2 \times 1 + 1 \times 1 = 3 & & = 2 \\ 3 \text{ mais próximos: } x_1, x_3, x_4 & & \text{output}(x_7) &= P & & & & \end{aligned}$$

$x_8$ :

$$\begin{aligned} d(x_8, x_1) &= 2 \times 1 + 1 \times 1 = 3 & d(x_8, x_3) &= 2 \times 1 + 1 \times 0 = 2 & d(x_8, x_5) &= 2 \times 0 + 1 \times 1 = 1 & d(x_8, x_7) &= 2 \times 1 + 1 \times 0 = 2 \\ d(x_8, x_2) &= 2 \times 0 + 1 \times 0 = 0 & d(x_8, x_4) &= 2 \times 1 + 1 \times 1 = 3 & d(x_8, x_6) &= 2 \times 0 + 1 \times 1 = 1 & & = 2 \\ 3 \text{ mais próximos: } x_2, x_5, x_6 & & \text{output}(x_8) &= N & & & & \end{aligned}$$

	REAL	PREVISTO			
x <sub>1</sub>	P	P			
x <sub>2</sub>	P	N			
x <sub>3</sub>	P	P			
x <sub>4</sub>	P	P			
x <sub>5</sub>	N	N			
x <sub>6</sub>	N	N			
x <sub>7</sub>	N	P			
x <sub>8</sub>	N	N			

	P	N
REALS	3	1
	1	3

$$\text{precisão} = \frac{3}{3+1} = 0,75$$

$$\text{recall} = \frac{3}{3+1} = 0,75$$

$$F1\text{-measure} = \frac{2 \times 0,75 \times 0,75}{0,75 + 0,75} = 0,75$$
  

$$\frac{F1\text{-measure} (ex2)}{F1\text{-measure} (ex1)} = \frac{0,75}{0,25} = 3$$

distância: weighted hamming (k=3)  
distância: Hamming (k=5)

↑ aumentou 3 vezes

An additional positive observation was acquired,  $x_9 = (B, 0)$ , and a third variable  $y_3$  was independently monitored, yielding estimates,  $y_3|P = \{1.1, 0.8, 0.5, 0.9, 0.8\}$  and  $y_3|N = \{1, 0.9, 1.2, 0.9\}$ .

3. 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; and iii)  $y_3$  is normally distributed. Show all parameters.

$$\frac{x_9 = (B, 0)}{P} \quad y_3|P = \{1.1; 0.8; 0.5; 0.9; 0.8\}$$

$$y_3|N = \{1; 0.9; 1.2; 0.9\}$$

Assuming:

- $y_1$  e  $y_2$  are dependent
- $\{y_1, y_2\}$  and  $\{y_3\}$  are independent and equally important
- $y_3$  is normally distribute

Teorema de Bayes

$$P(\text{class} = c | x) = \frac{P(x | \text{class} = c) \times P(\text{class} = c)}{P(x)}$$

Distribuição normal

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

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

$$\text{PRIORS: } P(\text{class} = P) = \frac{5}{9} \quad P(\text{class} = N) = \frac{4}{9}$$

Para as 4 combinações possíveis de  $y_1$  e  $y_2$ :

$$P(y_1 = A, y_2 = 0) = 2/9$$

$$P(y_1 = A, y_2 = 1) = 2/9$$

$$P(y_1 = B, y_2 = 0) = 3/9$$

$$P(y_1 = B, y_2 = 1) = 2/9$$

Considerando cada classe e as 4 combinações possíveis de  $y_1$  e  $y_2$ :

$$P(y_1=A, y_2=0 | P) = 2/5$$

$$P(y_1=B, y_2=0 | P) = 1/5$$

$$P(y_1=A, y_2=1 | P) = 1/5$$

$$P(y_1=B, y_2=1 | P) = 1/5$$

$$P(y_1=A, y_2=0 | N) = 0/4 = 0$$

$$P(y_1=B, y_2=0 | N) = 2/4 = 1/2$$

$$P(y_1=A, y_2=1 | N) = 1/4$$

$$P(y_1=B, y_2=1 | N) = 1/4$$

Para classes positivas:

$$\mu = \frac{\sum_{i=1}^5 y_{3i}}{5} = \frac{1.1+0.8+0.5+0.9+0.8}{5} = 0.82$$

$$\sigma^2 = \frac{1}{5-1} \sum_{i=1}^5 (y_{3i} - \mu)^2 = 0.047$$

$$P(y_3 | P) \sim N(\mu=0.82, \sigma^2=0.047)$$

Para classes negativas:

$$\mu = \frac{\sum_{i=1}^4 y_{3i}}{4} = \frac{1+0.9+1.2+0.9}{4} = 1$$

$$\sigma^2 = \frac{1}{4-1} \sum_{i=1}^4 (y_{3i} - \mu)^2 = 0.02$$

$$P(y_3 | N) \sim N(\mu=1, \sigma^2=0.02)$$

Consider now three testing observations,  $\{(A, 1, 0.8), (B, 1, 1), (B, 0, 0.9)\}$ .

4. Under a MAP assumption, classify each testing observation showing all your calculus.

$$\text{MAP} = \arg \max \{ P(c) \times P(x|c) \}$$

↳ maximum a posteriori

$\{y_1, y_2\}$  e  $\{y_3\}$  são independentes

$$\text{pdf} = \frac{1}{\sigma \times \sqrt{2\pi}} \times e^{-\frac{1}{2} \left( \frac{x-\mu}{\sigma} \right)^2}$$

$(A, 1, 0.8)$

$$\begin{aligned} P(A, 1, 0.8 | P) \cdot P(P) &= P(y_1=A, y_2=1) \cdot P(y_3=0.8) \cdot P(P) = \\ &= \frac{1}{5} \times N(0.8 | \mu=0.82, \sigma^2=0.047) \times \frac{5}{9} = \\ &= \frac{1}{5} \times 1.832 \times \frac{5}{9} = 0.204 \end{aligned}$$

$$\begin{aligned} P(A, 1, 0.8 | N) \cdot P(N) &= P(y_1=A, y_2=1) \cdot P(y_3=0.8) \cdot P(N) = \\ &= \frac{1}{4} \times N(0.8 | \mu=1, \sigma^2=0.02) \times \frac{4}{9} = \\ &= \frac{1}{4} \times 1.035 \times \frac{4}{9} = 0.115 \end{aligned}$$

$P(A, 1, 0.8 | P) \cdot P(P) > P(A, 1, 0.8 | N) \cdot P(N)$ , a observação  $(A, 1, 0.8)$  é classificada com **P**

$(B, 1, 1)$

$$\begin{aligned} P(B, 1, 1 | P) \cdot P(P) &= P(y_1=B, y_2=1) \cdot P(y_3=1) \cdot P(P) = \\ &= \frac{1}{5} \times N(1 | \mu=0.82, \sigma^2=0.047) \times \frac{5}{9} = \\ &= \frac{1}{5} \times 1.305 \times \frac{5}{9} = 0.145 \end{aligned}$$

$$\begin{aligned} P(B, 1, 1 | N) \cdot P(N) &= P(y_1=B, y_2=1) \cdot P(y_3=1) \cdot P(N) = \\ &= \frac{1}{4} \times N(1 | \mu=1, \sigma^2=0.02) \times \frac{4}{9} = \\ &= \frac{1}{4} \times 2.817 \times \frac{4}{9} = 0.313 \end{aligned}$$

$P(B, 1, 1 | P) \cdot P(P) < P(B, 1, 1 | N) \cdot P(N)$ , a observação  $(B, 1, 1)$  é classificada com **N**

$(B, 0, 0.9)$

$$\begin{aligned} P(B, 0, 0.9 | P) \cdot P(P) &= P(y_1=B, y_2=0) \cdot P(y_3=0.9) \cdot P(P) = \\ &= \frac{1}{5} \times N(0.9 | \mu=0.82, \sigma^2=0.047) \times \frac{5}{9} = \\ &= \frac{1}{5} \times 1.719 \times \frac{5}{9} = 0.191 \end{aligned}$$

$$\begin{aligned} P(B, 0, 0.9 | N) \cdot P(N) &= P(y_1=B, y_2=0) \cdot P(y_3=0.9) \cdot P(N) = \\ &= \frac{1}{4} \times N(0.9 | \mu=1, \sigma^2=0.02) \times \frac{4}{9} = \\ &= \frac{1}{4} \times 2.196 \times \frac{4}{9} = 0.244 \end{aligned}$$

$P(B, 0, 0.9 | P) \cdot P(P) < P(B, 0, 0.9 | N) \cdot P(N)$ , a observação  $(B, 0, 0.9)$  é classificada com **N**

At last, consider only the following sentences and their respective connotations,  
 {"Amazing run", P), ("I like it", P), ("Too tired", N), ("Bad run", N)}.

5. Using a naïve Bayes under a ML assumption, classify the new sentence "I like to run". For the likelihoods calculation consider the following formula,  $p(t_i|c) = (\text{freq}(t_i) + 1) / (N_c + V)$ , where  $t_i$  represents a certain term  $i$ ,  $V$  the number of unique terms in the vocabulary, and  $N_c$  the total number of terms in class  $c$ . Show all calculus.

{("Amazing run", P), ("I like it", P), ("Too tired", N), ("Bad run", N)}

⇒ "I like to run"

$$p(t_i|c) = \frac{\text{freq}(t_i) + 1}{N_c + V}$$

$t_i \rightarrow$  certain term  $i$

$V \rightarrow$  number of unique terms in the vocabulary

$N_c \rightarrow$  total number of terms in class  $c$

$$V = 8$$

$$N_P = 5$$

$$N_N = 4$$

classe P

$$p(I|P) = \frac{1+1}{5+8} = \frac{2}{13}$$

$$p(like|P) = \frac{1+1}{5+8} = \frac{2}{13}$$

$$p(to|P) = \frac{0+1}{5+8} = \frac{1}{13}$$

$$p(run|P) = \frac{1+1}{5+8} = \frac{2}{13}$$

classe N

$$p(I|N) = \frac{0+1}{4+8} = \frac{1}{12}$$

$$p(like|N) = \frac{0+1}{4+8} = \frac{1}{12}$$

$$p(to|N) = \frac{0+1}{4+8} = \frac{1}{12}$$

$$p(run|N) = \frac{1+1}{4+8} = \frac{2}{12} = \frac{1}{6}$$

ML assumption ⇒ posterior ∝ likelihood × prior

maximum likelihood

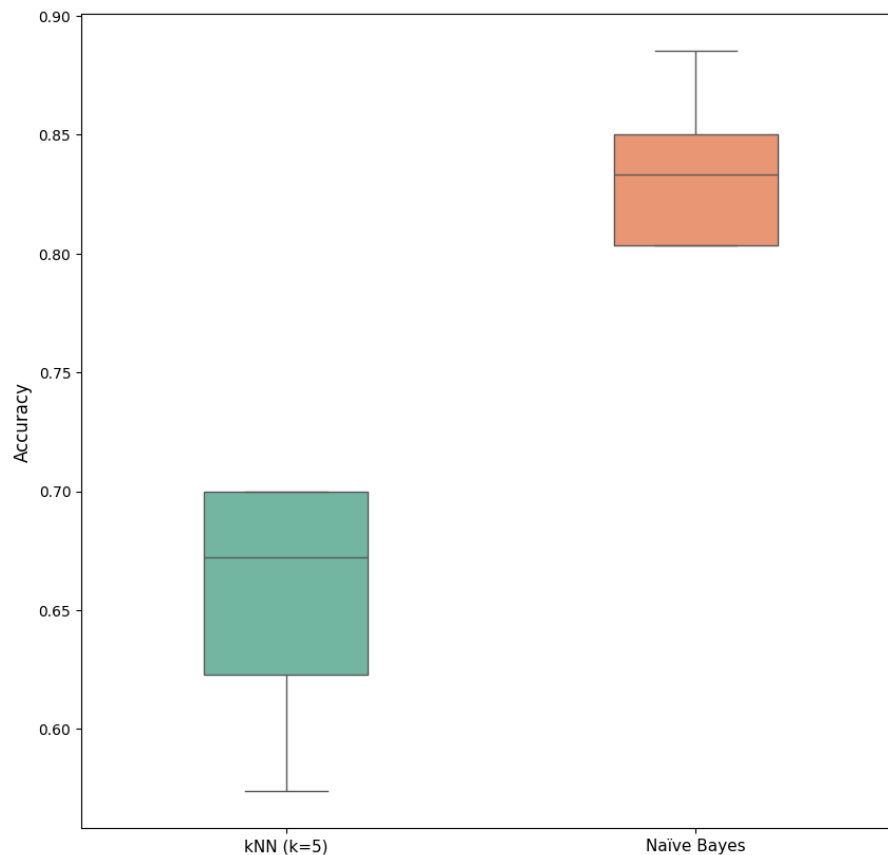
$$p(P|sentence) = \frac{2}{13} \times \frac{2}{13} \times \frac{1}{13} \times \frac{2}{13} = \frac{8}{28561} > p(N|sentence) = \frac{1}{12} \times \frac{1}{12} \times \frac{1}{12} \times \frac{1}{6} = \frac{1}{10368}$$

A frase é classificada como positiva (P)

## Part II: Programming

Consider the heart-disease.csv data available at the course webpage's homework tab. Using sklearn, apply a 5-fold stratified cross-validation with shuffling RANDOM FIXME for the assessment of predictive models along this selection.

1. Compare the performance of a kNN with  $k = 5$  and a naïve Bayes with Gaussian assumption (consider all remaining parameters as default):
  - a. Plot two boxplots with the fold accuracies for each classifier. Is there one more stable than the other regarding performance? Why do you think that is the case? Explain.



The Naïve Bayes classifier shows more consistent performance, with a narrower interquartile range (IQR) and less variability in accuracy. This stability likely arises from its assumption of feature independence and Gaussian distribution, which makes it less sensitive to noise and data imbalances. Since Naïve Bayes operates under these strong assumptions, it maintains consistent behavior across different datasets.

In contrast, the kNN classifier is more sensitive to the dataset's distribution and the selection of neighbors, which leads to greater fluctuations in accuracy. Its reliance on local relationships makes it more prone to changes, as these can shift based on the data. This sensitivity is reflected in the wider IQR and larger spread in kNN's accuracy, resulting in less stability compared to Naïve Bayes.



b. Report the accuracy of both models, this time scaling the data with a Min-Max scaler before training the models. Explain the impact that this preprocessing step has on the performance of each model, providing an explanation for the results.

$$accuracy_{NaiveBayes} = 0.84 \pm 0.03$$

$$accuracy_{kNN} = 0.83 \pm 0.02$$

Min-Max scaling has a minimal impact on Naïve Bayes. This can be attributed to the assumption that most (if not all) numeric variables exhibit a distribution close to Gaussian, which aligns with the assumptions made by the Naïve Bayes classifier. Since Naïve Bayes assumes feature independence and does not rely on distance metrics, its performance remains stable even when feature ranges vary.

In contrast, scaling significantly enhances kNN by improving accuracy and stability. Before scaling, features with larger ranges can disproportionately affect distance calculations, leading to inconsistent results. Normalizing these features ensures equal contribution, reducing sensitivity to data fluctuations.

Consequently, kNN benefits greatly from scaling, while Naïve Bayes experiences negligible changes.

c. Using scipy, test the hypothesis “the kNN model is statistically superior to naïve Bayes regarding accuracy”, asserting whether it is true.

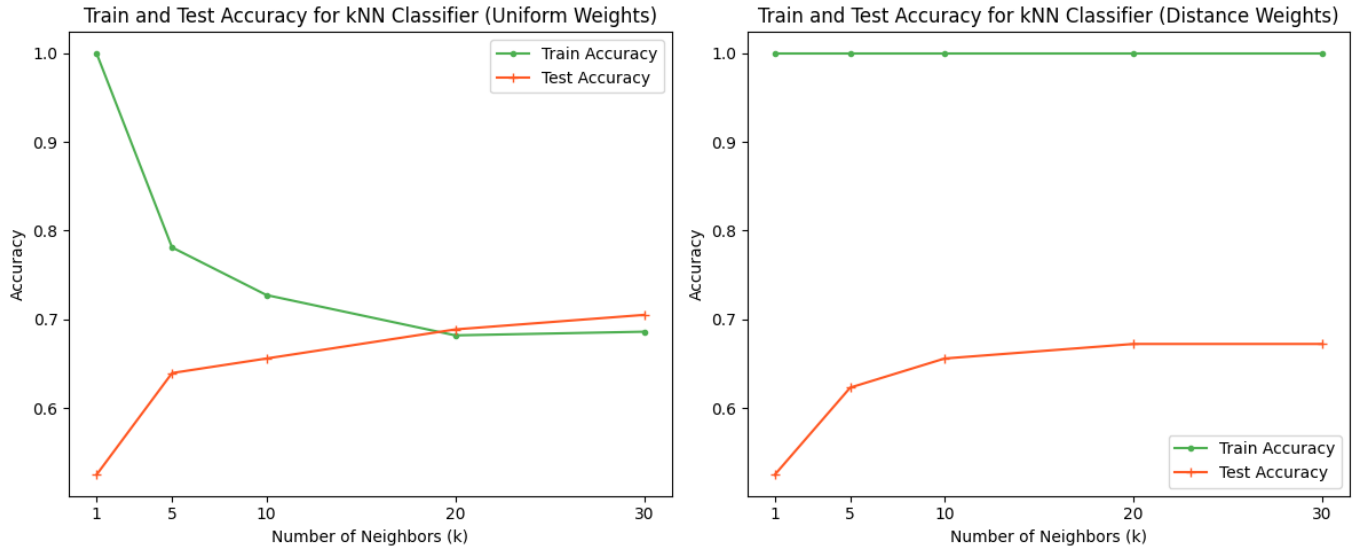
We will conduct a one-tailed test using the accuracy scores obtained from our previous analysis. We are considering the following hypotheses:

$$H_0 : accuracy_{kNN} = accuracy_{NaiveBayes}$$

$$H_1 : accuracy_{kNN} > accuracy_{NaiveBayes}$$

Using scipy, we calculated a p-value of  $\approx 0.9987$  for the non-scaled data and  $\approx 0.5037$  for the scaled data. Both values are well above conventional significance levels (1%, 5%, and 10%), indicating that we cannot reject the null hypothesis of equal model accuracies. Therefore, there is insufficient evidence to assert that the kNN model is statistically superior to Naïve Bayes in terms of accuracy.

2. Using a 80-20 train-test split, vary the number of neighbors of a kNN classifier using  $k = \{1, 5, 10, 20, 30\}$ . Additionally, for each  $k$ , train one classifier using uniform weights and distance weights.
- a. Plot the train and test accuracy for each model.



- b. Explain the impact of increasing the neighbors on the generalization ability of the models.

In the uniform weights case (left plot), when  $k = 1$ , the model overfits the training data, resulting in near-perfect training accuracy but lower test accuracy. As  $k$  increases, the test accuracy improves steadily and becomes close to the training accuracy around  $k = 20$ . However, after this point, the training accuracy continues to drop and becomes lower than the test accuracy, indicating that the model may be underfitting. This happens because with a large number of neighbors, the model smooths out the decision boundary too much, making it less effective at capturing details in the training data.

In the distance weights case (right plot), the training accuracy remains constant and very high across all values of  $k$ , indicating that the model fits the training data closely. The test accuracy steadily improves as  $k$  increases and plateaus around  $k = 20$ , where it reaches stable generalization performance. Beyond this point, further increases in  $k$  do not significantly change the test accuracy, as the model appropriately assigns more weight to closer neighbors, minimizing the influence of distant, less relevant points. This allows the model to generalize well even with larger values of  $k$ , without noticeable underfitting.

3. Considering the unique properties of the heart-disease.csv dataset, identify two possible difficulties of the naïve Bayes model used in the previous exercises when learning from the given dataset.

Naïve Bayes assumes that all features are conditionally independent given the class label. However, in the context of heart disease, many risk factors — such as cholesterol levels, blood pressure, and age — can be correlated. For example, high blood pressure often correlates with high cholesterol, and both are significant risk factors for heart disease. This violation of the independence assumption can lead to suboptimal model performance, as the Naïve Bayes model may fail to accurately capture the relationships among predictors, resulting in less effective classification of heart disease cases.

Additionally, the heart disease dataset includes both categorical variables (e.g., chest pain type) and continuous variables (e.g., age, cholesterol levels). Naïve Bayes is generally more effective with categorical data and operates under the assumption that continuous features follow a Gaussian distribution. However, if these continuous features deviate from normality — an occurrence that is common in medical datasets — the model's predictions may become inaccurate. This can adversely affect its performance in classifying heart disease cases, as the model may not adequately represent the underlying distributions of the continuous features.

## Appendix

```
1 from sklearn.neighbors import KNeighborsClassifier
2 from sklearn.naive_bayes import GaussianNB
3 from sklearn.model_selection import StratifiedKFold, cross_val_score
4 from scipy import stats
5 import matplotlib.pyplot as plt
6 import pandas as pd
7 import seaborn as sns
8
9 df = pd.read_csv("../data/heart-disease.csv")
10
11 X = df.drop("target", axis = 1)
12 y = df["target"]
13
14 skf = StratifiedKFold(n_splits = 5, shuffle = True, random_state = 0)
15
16 knn_classifier = KNeighborsClassifier(n_neighbors = 5)
17 naive_bayes_classifier = GaussianNB()
18
19 knn_scores = cross_val_score(knn_classifier, X, y.tolist(), cv = skf, scoring='
    accuracy')
20 naive_bayes_scores = cross_val_score(naive_bayes_classifier, X, y.tolist(), cv =
    skf, scoring = 'accuracy')
21
22 plt.figure(figsize=(10, 10))
23
24 sns.boxplot(data = [knn_scores, naive_bayes_scores], palette = "Set2", width = 0.4)
25 plt.xticks([0, 1], ["kNN (k=5)", "Na ve Bayes"], fontsize = 11)
26 plt.ylabel("Accuracy", fontsize = 12)
27
28 plt.show()
```

Listing 1: Accuracy Boxplot for k-Nearest Neighbors and Naïve Bayes Classifiers

```

1 from sklearn.preprocessing import MinMaxScaler
2 import numpy as np
3
4 numeric_features = ["age", "trestbps", "chol", "thalach", "oldpeak"]
5
6 min_max_scaler = MinMaxScaler()
7 X_scaled_minmax = X.copy()
8 X_scaled_minmax[numeric_features] = min_max_scaler.fit_transform(X[numeric_features
9 ])
10 knn_scores_minmax = cross_val_score(knn_classifier, X_scaled_minmax, y, cv = skf,
11     scoring = "accuracy")
12 gnb_scores_minmax = cross_val_score(naive_bayes_classifier, X_scaled_minmax, y, cv
13     = skf, scoring = "accuracy")
14
15 print(f"Na ve Bayes accuracy = {np.mean(gnb_scores_minmax):.2f}, {np.std(
16     gnb_scores_minmax):.2f}")
17 print(f"kNN accuracy = {np.mean(knn_scores_minmax):.2f}, {np.std(knn_scores_minmax)
18     :.2f}")

```

Listing 2: Scaled Accuracy Results for k-Nearest Neighbors and Naïve Bayes Classifiers

```

1 res = stats.ttest_rel(knn_scores , naive_bayes_scores , alternative = "greater")
2 print("Not scaled: knn > naive_bayes? pval=", res.pvalue)
3
4 res = stats.ttest_rel(knn_scores_minmax , gnb_scores_minmax , alternative = "
5     greater")
6 print("Scaled: knn > naive_bayes? pval=", res.pvalue)

```

Listing 3: Hypothesis Test Results for k-Nearest Neighbors vs. Naïve Bayes Classifiers (Not Scaled and Scaled Accuracies)

```

1 from sklearn import model_selection
2 from sklearn import metrics
3 from sklearn.model_selection import train_test_split
4
5 K_VALUES = [1, 5, 10, 20, 30]
6 weights = ["uniform", "distance"]
7
8 X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.8,
9     random_state = 0)
10
11 train_accuracies = [[] for _ in range(2)]
12 test_accuracies = [[] for _ in range(2)]
13
14 for k in K_VALUES:
15     for i, weight in enumerate(weights):
16         predictor = KNeighborsClassifier(n_neighbors = k, weights = weight)
17         predictor.fit(X_train, y_train)
18
19         train_accuracy = metrics.accuracy_score(y_train, predictor.predict(X_train))
20
21         test_accuracy = metrics.accuracy_score(y_test, predictor.predict(X_test))
22
23         train_accuracies[i].append(train_accuracy)
24         test_accuracies[i].append(test_accuracy)

```

```

24 plt.figure(figsize=(12, 5))
25
26 train_color = "#4caf50"
27 test_color = "#ff5722"
28
29 for i, weight in enumerate(weights):
30     plt.subplot(1, 2, i + 1)
31     plt.plot(K_VALUES, train_accuracies[i], label = "Train Accuracy", marker = ".",
32             color = train_color)
33     plt.plot(K_VALUES, test_accuracies[i], label = "Test Accuracy", marker = "+",
34             color = test_color)
35     plt.title(f"Train and Test Accuracy for kNN Classifier ({weight.capitalize()}
36             Weights)")
37     plt.xlabel("Number of Neighbors (k)")
38     plt.ylabel("Accuracy")
39     plt.legend()
40     plt.xticks(K_VALUES)
41
42 plt.tight_layout()
43 plt.show()

```

Listing 4: Accuracy Plot for k-Nearest Neighbors Classifier with Varying k Values and Weighting Schemes