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

Part I: Pen and paper

We collected four positive (P) observations, {x1 = (A,0), x2 = (B, 1), x3 = (A, 1), x4 = (A, 0)} and four negative (N) observations, {x5 = (B, 0), x6 = (B, 0), x7 = (A, 1), x8 = (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: $\sum_{i=1}^{p} a_i \neq b_i$ — para features categóricas
$x_1 = [A\ A\ B]$
$x_2 = [B\ A\ B]$ Hamming $[x_1, x_2] = 1$

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
$\underbrace{x_3, x_4}_{P}, \underbrace{x_5, x_6, x_7}_{N}$

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: $\underbrace{x_3}_{P}, \underbrace{x_5, x_6, x_7, x_8}_{N}$ 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: $\underbrace{x_1, x_2, x_4}_{P}, \underbrace{x_7, x_8}_{N}$ 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: $\underbrace{x_1, x_3}_{P}, \underbrace{x_5, x_6, x_7}_{N}$ output (x_4) = N

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

5 mais próximos: $\underbrace{x_1, x_2, x_4}_{P}, \underbrace{x_6, x_8}_{N}$ Output $(x_5) = P$

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

5 mais próximos: $\underbrace{x_1, x_2, x_4}_{P}, \underbrace{x_5, x_8}_{N}$ output $(x_6) = P$

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

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

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

5 mais próximos: $\underbrace{x_2, x_3}_{P}, \underbrace{x_5, x_6, x_7}_{N}$ 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	Previsto N
Reais	P	1	3
	N	3	1

		Previsto P	Previsto N
Reais	P	TP	FN
	N	FP	TN

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

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

T → true
F → false

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

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 ← weighted distance K=3

escolhemos que o weight da primeira feature seja maior que o da segunda

weighted Hamming Distance $(x_i, x_j) = 2w_1 + 1w_2$

w_1: primeira característica (ou A ou B) — w_2: segunda característica (ou 1 ou 2)

x_1:

$d(x_1, x_2) = 2\times1 + 1\times1 = 3$ — $d(x_1, x_4) = 2\times0 + 1\times0 = 0$ — $d(x_1, x_6) = 2\times1 + 1\times0 = 2$ — $d(x_1, x_8) = 2\times1 + 1\times1 = 3$

$d(x_1, x_3) = 2\times0 + 1\times1 = 1$ — $d(x_1, x_5) = 2\times1 + 1\times0 = 2$ — $d(x_1, x_7) = 2\times0 + 1\times1 = 1$

3 mais próximos: x_3, x_4, x_7 — output $(x_1) = P$

x_2:

$d(x_2, x_1) = 2\times1 + 1\times1 = 3$ — $d(x_2, x_4) = 2\times1 + 1\times1 = 3$ — $d(x_2, x_6) = 2\times0 + 1\times1 = 1$ — $d(x_2, x_8) = 2\times0 + 1\times0 = 0$

$d(x_2, x_3) = 2\times1 + 1\times0 = 2$ — $d(x_2, x_5) = 2\times0 + 1\times1 = 1$ — $d(x_2, x_7) = 2\times1 + 1\times0 = 2$

3 mais próximos: x_5, x_6, x_8 — output $(x_2) = N$

x_3:

$d(x_3, x_1) = 2\times0 + 1\times1 = 1$ — $d(x_3, x_4) = 2\times0 + 1\times1 = 1$ — $d(x_3, x_6) = 2\times1 + 1\times1 = 3$ — $d(x_3, x_8) = 2\times1 + 1\times0 = 2$

$d(x_3, x_2) = 2\times1 + 1\times0 = 2$ — $d(x_3, x_5) = 2\times1 + 1\times1 = 3$ — $d(x_3, x_7) = 2\times0 + 1\times0 = 0$

3 mais próximos: x_1, x_4, x_7 — output $(x_3) = P$

x_4:

$d(x_4, x_1) = 2\times0 + 1\times0 = 0$ — $d(x_4, x_3) = 2\times0 + 1\times1 = 1$ — $d(x_4, x_6) = 2\times1 + 1\times0 = 2$ — $d(x_4, x_8) = 2\times1 + 1\times1 = 3$

$d(x_4, x_2) = 2\times1 + 1\times1 = 3$ — $d(x_4, x_5) = 2\times1 + 1\times0 = 2$ — $d(x_4, x_7) = 2\times0 + 1\times1 = 1$

3 mais próximos: x_1, x_3, x_7 — output $(x_4) = P$

x_5:

$d(x_5, x_1) = 2\times1 + 1\times0 = 2$ — $d(x_5, x_3) = 2\times1 + 1\times1 = 3$ — $d(x_5, x_6) = 2\times0 + 1\times0 = 0$ — $d(x_5, x_8) = 2\times0 + 1\times1 = 1$

$d(x_5, x_2) = 2\times0 + 1\times1 = 1$ — $d(x_5, x_4) = 2\times1 + 1\times0 = 2$ — $d(x_5, x_7) = 2\times1 + 1\times1 = 3$

3 mais próximos: x_2, x_6, x_8 — output $(x_5) = N$

x_6:

$d(x_6, x_1) = 2\times1 + 1\times0 = 2$ — $d(x_6, x_3) = 2\times1 + 1\times1 = 3$ — $d(x_6, x_5) = 2\times0 + 1\times0 = 0$ — $d(x_6, x_8) = 2\times0 + 1\times1 = 1$

$d(x_6, x_2) = 2\times0 + 1\times1 = 1$ — $d(x_6, x_4) = 2\times1 + 1\times0 = 2$ — $d(x_6, x_7) = 2\times1 + 1\times1 = 3$

3 mais próximos: x_2, x_5, x_8 — output $(x_6) = N$

x_7:

$d(x_7, x_1) = 2\times0 + 1\times1 = 1$ — $d(x_7, x_3) = 2\times0 + 1\times0 = 0$ — $d(x_7, x_5) = 2\times1 + 1\times1 = 3$ — $d(x_7, x_8) = 2\times1 + 1\times0 = 2$

$d(x_7, x_2) = 2\times1 + 1\times0 = 2$ — $d(x_7, x_4) = 2\times0 + 1\times1 = 1$ — $d(x_7, x_6) = 2\times1 + 1\times1 = 3$

3 mais próximos: x_1, x_3, x_4 — output $(x_7) = P$

x_8:

$d(x_8, x_1) = 2\times1 + 1\times1 = 3$ — $d(x_8, x_3) = 2\times1 + 1\times0 = 2$ — $d(x_8, x_5) = 2\times0 + 1\times1 = 1$ — $d(x_8, x_7) = 2\times1 + 1\times0 = 2$

$d(x_8, x_2) = 2\times0 + 1\times0 = 0$ — $d(x_8, x_4) = 2\times1 + 1\times1 = 3$ — $d(x_8, x_6) = 2\times0 + 1\times1 = 1$

3 mais próximos: x_2, x_5, x_6 — output $(x_8) = N$

	Real	Previsto
x_1	P	P
x_2	P	N
x_3	P	P
x_4	P	P
x_5	N	N
x_6	N	N
x_7	N	P
x_8	N	N

Reais \ Previsto	P	N
P	3	1
N	1	3

precisão $= \frac{3}{3+1} = 0,75$

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

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

K=3 distância: weighted hamming

K=5 distância: Hamming

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

aumentou 3 vezes

An additional positive observation was acquired, x9 = (B, 0), and a third variable y3 was independently monitored, yielding estimates, y3|P = {1.1, 0.8, 0.5, 0.9, 0.8} and y3|N = {1, 0.9, 1.2, 0.9}.

3. Considering the nine training observations, learn a Bayesian classifier assuming: i) y1 and y2 are dependent; ii) y1, y2 and y3 variable sets are independent and equally important; and iii) y3 is normally distributed. Show all parameters.

$x_9 = (B, 0)$ — P

$y_3|P = \{1,1 ; 0,8 ; 0,5 ; 0,9 ; 0,8\}$

$y_3|N = \{1 ; 0,9 ; 1,2 ; 0,9\}$

Assuming:

i) y_1 e y_2 are dependent

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

iii) y_3 is normally distribute

Teorema de Bayes

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

Distribuição normal

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

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

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

Para as 4 combinações possíveis de y_1 e y_2:

$P(y_1 = A, y_2 = 0) = 2/9$ $\quad P(y_1 = A, y_2 = 1) = 2/9$

$P(y_1 = B, y_2 = 0) = 3/9$ $\quad P(y_1 = B, y_2 = 1) = 2/9$

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

$P(y_1 = A, y_2 = 0 \mid P) = 2/5$
$P(y_1 = B, y_2 = 0 \mid P) = 1/5$

$P(y_1 = A, y_2 = 1 \mid P) = 1/5$
$P(y_1 = B, y_2 = 1 \mid P) = 1/5$

$P(y_1 = A, y_2 = 0 \mid N) = 0/4 = 0$
$P(y_1 = B, y_2 = 0 \mid N) = 2/4 = 1/2$

$P(y_1 = A, y_2 = 1 \mid N) = 1/4$
$P(y_1 = B, y_2 = 1 \mid N) = 1/4$

Para classes positivas:

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

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

$$P(y_3 \mid P) \sim \mathcal{N}(\mu = 0.82, \sigma^2 = 0.047)$$

Para classes negativas:

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

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

$$P(y_3 \mid N) \sim \mathcal{N}(\mu = 1, \sigma^2 = 0.02)$$

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.

MAP = $\arg\max \{ P(c) \times P(x|c) \}$ $\{y_1, y_2\}$ e $\{y_2\}$ são independentes

↳ maximum a posteriori

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

(A, 1, 0.8)

$P(A, 1, 0.8 | P) \cdot P(P) = P(y_1 = A, y_2 = 1 | P) \cdot P(y_3 = 0,8 | P) \cdot P(P) =$
$= 1/5 \times N(0,8 | \mu = 0,82, \sigma^2 = 0,047) \times 5/9 =$
$= 1/5 \times 1,832 \times 5/9 \approx 0,204$

$P(A, 1, 0.8 | N) \cdot P(N) = P(y_1 = A, y_2 = 1 | N) \cdot P(y_3 = 0,8 | N) \cdot P(N) =$
$= 1/4 \times N(0,8 | \mu = 1, \sigma^2 = 0,02) \times 4/9 =$
$= 1/4 \times 1,035 \times 4/9 \approx 0,115$

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

(B, 1, 1)

$P(B, 1, 1 | P) \cdot P(P) = P(y_1 = B, y_2 = 1 | P) \cdot P(y_3 = 1 | P) \cdot P(P) =$
$= 1/5 \times N(1 | \mu = 0,82, \sigma^2 = 0,047) \times 5/9 =$
$= 1/5 \times 1,305 \times 5/9 \approx 0,145$

$P(B, 1, 1 | N) \cdot P(N) = P(y_1 = B, y_2 = 1 | N) \cdot P(y_3 = 1 | N) \cdot P(N) =$
$= 1/4 \times N(1 | \mu = 1, \sigma^2 = 0,02) \times 4/9 =$
$= 1/4 \times 2,817 \times 4/3 \approx 0,313$

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

(B, 0, 0.9)

$P(B, 0, 0.9 | P) \cdot P(P) = P(y_1 = B, y_2 = 0 | P) \cdot P(y_3 = 0.9 | P) \cdot P(P) =$
$= 1/5 \times N(0.9 | \mu = 0,82, \sigma^2 = 0,047) \times 5/9 =$
$= 1/5 \times 1,719 \times 5/9 \approx 0,191$

$P(B, 0, 0.9 | N) \cdot P(N) = P(y_1 = B, y_2 = 0 | N) \cdot P(y_3 = 0.9 | N) \cdot P(N) =$
$= 1/2 \times N(0,9 | \mu = 1, \sigma^2 = 0,02) \times 4/9 =$
$= 1/2 \times 2,196 \times 4/9 \approx 0,488$

$P(B, 0, 0.9 | P) \cdot P(P) < P(B, 0, 0.9 | N) \cdot P(N)$, 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,

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

where ti represents a certain term i, V the number of unique terms in the vocabulary, and Nc 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 → certain term i

V → number of unique terms in the vocabulary

Nc → total number of terms in class c

V = 8

$N_P = 5$

$N_N = 4$

Classe P

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

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

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

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

Classe N

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

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

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

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

ML assumption ⇒ posterior ∝ likelihood x prior

maximum likelihood

$p(P|\text{sentence}) = \frac{2}{13} \times \frac{2}{13} \times \frac{1}{13} \times \frac{2}{13} = \frac{8}{28561}$ >

$p(N|\text{sentence}) = \frac{1}{12} \times \frac{1}{12} \times \frac{1}{12} \times \frac{1}{6} = \frac{1}{10368}$

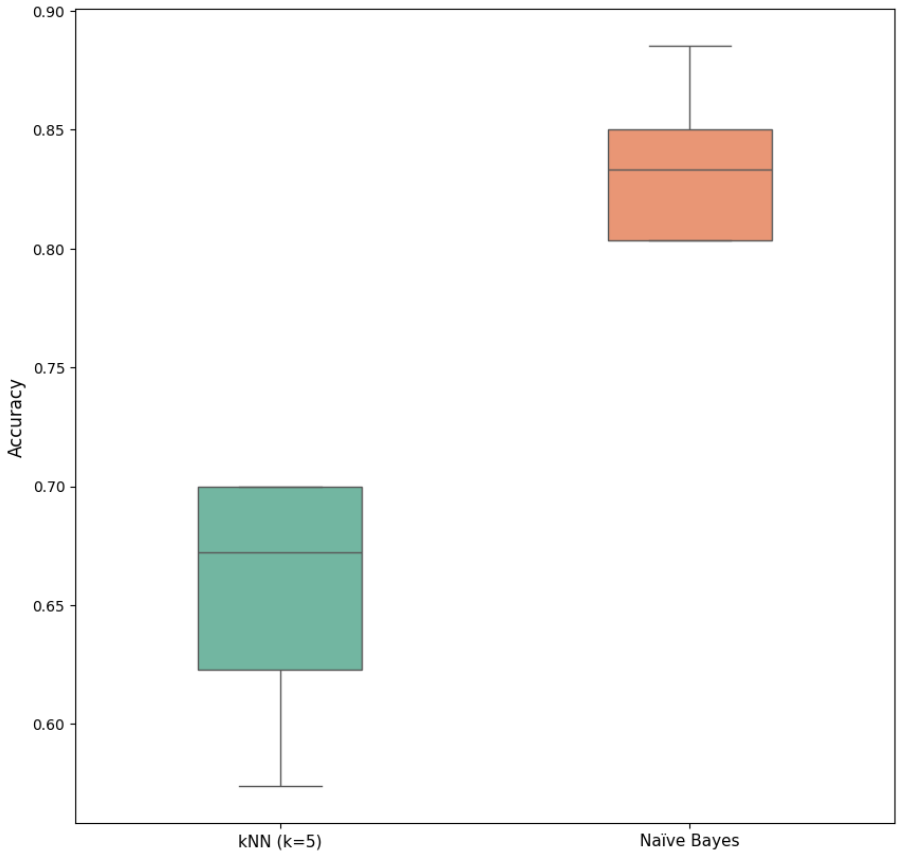
A frase é classificada como positiva (P)

Part II: Programming

Consider the heart-disease.csv data available at the course webpage's homework tab. Using skrlearn, apply a 5-fold stratified cross-validation with shuffling (random_state = 0) 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.

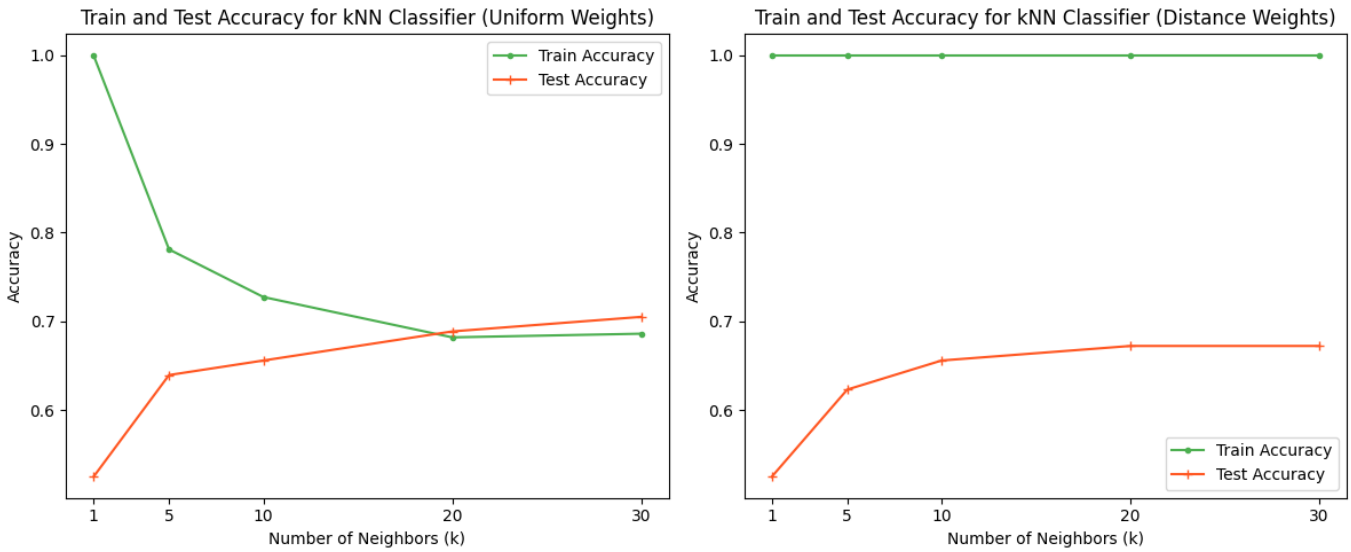
We will conduct a one-tailed test using the accuracy scores obtained from our previous analysis. We are considering the following hypotheses:

$$H_0 : accuracy_{kNN} = accuracy_{NaiveBayes}$$
$$H_1 : accuracy_{kNN} > accuracy_{NaiveBayes}$$

Using scipy, we calculated a p-value of ≈ 0.9987 for the non-scaled data and ≈ 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 the number of neighbors is small, the model tends to overfit the training data, leading to high training accuracy but lower test accuracy. However, as k grows, the test accuracy rises while training accuracy declines, indicating that the model is becoming less sensitive to noise and outliers. By incorporating more neighbors, it focuses less on individual data points and more on underlying patterns in the data, improving generalization. However, at higher values of k, the decline in training accuracy suggests that the model may begin to underfit.

In the distance weights case (right plot), there is less improvement in generalization as k increases. Training accuracy remains consistently high, while test accuracy only experiences slight gains. The persistent gap between training and test accuracy indicates that the model struggles to generalize effectively, as it gives more weight to closer neighbors. This leads to overfitting, particularly when the test data differs from the training set, since the model overly relies on local data points instead of capturing broader patterns.

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

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import StratifiedKFold, cross_val_score
from scipy import stats
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

df = pd.read_csv("./data/heart-disease.csv")

X = df.drop("target", axis = 1)
y = df["target"]

skf = StratifiedKFold(n_splits = 5, shuffle = True, random_state = 0)

knn_classifier = KNeighborsClassifier(n_neighbors = 5)
naive_bayes_classifier = GaussianNB()

knn_scores = cross_val_score(knn_classifier, X, y.tolist(), cv = skf, scoring='
    accuracy')
naive_bayes_scores = cross_val_score(naive_bayes_classifier, X, y.tolist(), cv =
    skf, scoring = 'accuracy')

plt.figure(figsize=(10, 10))

sns.boxplot(data = [knn_scores, naive_bayes_scores], palette = "Set2", width = 0.4)
plt.xticks([0, 1], ["kNN (k=5)", "Na ve Bayes"], fontsize = 11)
plt.ylabel("Accuracy", fontsize = 12)

plt.show()
```

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

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np

numeric_features = ["age", "trestbps", "chol", "thalach", "oldpeak"]

min_max_scaler = MinMaxScaler()
X_scaled_minmax = X.copy()
X_scaled_minmax[numeric_features] = min_max_scaler.fit_transform(X[numeric_features
    ])

knn_scores_minmax = cross_val_score(knn_classifier, X_scaled_minmax, y, cv = skf,
    scoring = "accuracy")
gnb_scores_minmax = cross_val_score(naive_bayes_classifier, X_scaled_minmax, y, cv
    = skf, scoring = "accuracy")

print(f"Na ve Bayes accuracy = {np.mean(gnb_scores_minmax):.2f}, {np.std(
    gnb_scores_minmax):.2f}")
print(f"kNN accuracy = {np.mean(knn_scores_minmax):.2f}, {np.std(knn_scores_minmax)
    :.2f}")
```

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

```
res = stats.ttest_rel(knn_scores , naive_bayes_scores , alternative = "greater")
print("Not scaled: knn > naive_bayes? pval=", res.pvalue)

res = stats.ttest_rel(knn_scores_minmax , gnb_scores_minmax , alternative = "
    greater")
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)

```
from sklearn import model_selection
from sklearn import metrics
from sklearn.model_selection import train_test_split

K_VALUES = [1, 5, 10, 20, 30]
weights = ["uniform", "distance"]

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.8,
    random_state = 0)

train_accuracies = [[] for _ in range(2)]
test_accuracies = [[] for _ in range(2)]

for k in K_VALUES:
    for i, weight in enumerate(weights):
        predictor = KNeighborsClassifier(n_neighbors = k, weights = weight)
        predictor.fit(X_train, y_train)

        train_accuracy = metrics.accuracy_score(y_train, predictor.predict(X_train)
    )
        test_accuracy = metrics.accuracy_score(y_test, predictor.predict(X_test))

        train_accuracies[i].append(train_accuracy)
        test_accuracies[i].append(test_accuracy)

```

```
plt.figure(figsize=(12, 5))

train_color = "#4caf50"
test_color = "#ff5722"

for i, weight in enumerate(weights):
    plt.subplot(1, 2, i + 1)
    plt.plot(K_VALUES, train_accuracies[i], label = "Train Accuracy", marker = ".",
     color = train_color)
    plt.plot(K_VALUES, test_accuracies[i], label = "Test Accuracy", marker = "+",
    color = test_color)
    plt.title(f"Train and Test Accuracy for kNN Classifier ({weight.capitalize()}
    Weights)")
    plt.xlabel("Number of Neighbors (k)")
    plt.ylabel("Accuracy")
    plt.legend()
    plt.xticks(K_VALUES)

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

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