

1. a) Observações de freino:  $x_1, x_2$

**Priors:**  $P(y_0 = A) = \frac{3}{7}$   $P(y_0 = B) = \frac{4}{7}$

Para o conjunto  $\{y_1, y_2\}$ :

• Para a classe A

$$\mu(y_1 | y_0 = A) = \frac{0.24 + 0.16 + 0.22}{3} = 0.24$$

$$\mu(y_2 | y_0 = A) = \frac{0.36 + 0.48 + 0.42}{3} = 0.52$$

$$\Sigma = \begin{bmatrix} \text{cov}(y_1, y_1) & \text{cov}(y_1, y_2) \\ \text{cov}(y_2, y_1) & \text{cov}(y_2, y_2) \end{bmatrix} \quad \text{cov}(y_1, y_1) = \sigma_{y_1 | A}^2 = \frac{\sum_i (y_{1i})^2 - n\bar{y}_1^2}{2} = 0.0064 \quad \text{cov}(y_2, y_2) = \sigma_{y_2 | A}^2 = \frac{\sum_i (y_{2i})^2 - n\bar{y}_2^2}{2} = 0.0236$$

$$\text{cov}(y_1, y_2 | A) = \frac{\sum_i (y_{1i} - \bar{y}_1)(y_{2i} - \bar{y}_2)}{2} = \frac{0.0192}{2} = 0.0096 = \text{cov}(y_2, y_1 | A)$$

$$\Sigma_A = \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0236 \end{bmatrix}$$

$$\vec{\mu}_A = [0.24, 0.52]$$

$$(y_1, y_2 | A) \sim N(\vec{\mu}_A, \Sigma_A)$$

$$\sigma(x) = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n-1}}$$

$$\text{cov}(x_i, y_i) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

• Para a classe B

$$\mu(y_1 | y_0 = B) = \frac{0.54 + 0.66 + 0.16 + 0.41}{4} = 0.5925$$

$$\mu(y_2 | y_0 = B) = \frac{0.11 + 0.39 + 0.28 + 0.53}{4} = 0.3245$$

De modo semelhante a acima:

$$\text{cov}(y_1, y_1 | B) = \sigma_{y_1 | B}^2 = 0.0229 \quad \text{cov}(y_2, y_2 | B) = \sigma_{y_2 | B}^2 = 0.0315 \quad \text{cov}(y_1, y_2 | B) = \text{cov}(y_2, y_1 | B) = -0.0096$$

$$\Sigma_B = \begin{bmatrix} 0.0229 & -0.0096 \\ -0.0096 & 0.0315 \end{bmatrix} \quad \vec{\mu}_B = [0.5925, 0.3245] \quad (y_1, y_2 | B) \sim N(\vec{\mu}_B, \Sigma_B)$$

Para o conjunto  $\{y_3, y_4\}$ :

$$P(X | y_0): \quad P(y_3 = 0, y_4 = 0 | A) = 0 \quad P(y_3 = 1, y_4 = 0 | A) = P(y_3 = 0, y_4 = 1 | A) = P(y_3 = 1, y_4 = 1 | A) = \frac{1}{3}$$

$$P(y_3 = 0, y_4 = 0 | B) = \frac{1}{2} \quad P(y_3 = 1, y_4 = 0 | B) = P(y_3 = 0, y_4 = 1 | B) = \frac{1}{4} \quad P(y_3 = 1, y_4 = 1 | B) = 0$$

$$P(y_3 = 0, y_4 = 0) = 0 \times \frac{3}{4} + \frac{1}{2} \times \frac{4}{4} = \frac{1}{2} \quad P(y_3 = 1, y_4 = 0) = P(y_3 = 0, y_4 = 0) = \frac{1}{3} \times \frac{3}{7} + \frac{1}{4} \times \frac{4}{7} = \frac{2}{7}$$

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

Para o conjunto  $\{y_5\}$ :

$$P(X | y_0): \quad P(y_5 = 0 | A) = P(y_5 = 1 | A) = P(y_5 = 2 | A) = \frac{1}{3}$$

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

$$P(y_5 = 0) = \frac{1}{3} \times \frac{3}{7} + \frac{1}{4} \times \frac{4}{7} = \frac{2}{7} \quad P(y_5 = 1) = \frac{1}{3} \times \frac{2}{7} + \frac{1}{2} \times \frac{4}{7} = \frac{3}{7}$$

$$P(y_5 = 2) = 1 - (\frac{2}{7} + \frac{3}{7}) = \frac{2}{7}$$

b) **Para  $x_8$ :**  $P(A | x_8) = P(y_1, y_2 | A) \times P(y_3, y_4 | A) \times P(y_5 | A) \times P(A)$

$$P(A) = \frac{3}{7}, \quad P(B) = \frac{4}{7} \quad (\text{Prior scimo})$$

$$P(B | x_8) = P(y_1, y_2 | B) \times P(y_3, y_4 | B) \times P(y_5 | B) \times P(B)$$

**$P(A | y_8)$**

$$P(y_3, y_4 | A) = P(0, 1 | A) = \frac{1}{3} \quad P(y_5 | A) = P(0 | A) = \frac{1}{3}$$

$$P(y_1, y_2 | A) = P(0.38, 0.52 | A) \quad \checkmark$$

$$P(y_3, y_4 | A) = P(0, 1 | A) = \frac{1}{3} \quad P(y_5 | A) = P(0 | A) = \frac{1}{3}$$

$$P(y_1, y_2 | A) = P(0.38, 0.52 | A)$$

utilizando o Python com a função multivariate\_normal sugerida com os parâmetros determinados acima ( $(y_1, y_2 | A) \sim N(\vec{\mu}_A, \Sigma_A)$ ) para o vetor  $[0.38, 0.52]$ , obtém-se:

$$\text{PDF}_A([0.38, 0.52]) = P(0.38, 0.52 | A) = 0.9847$$

$$\bullet P(A|x_8) = P(y_1, y_2 | A) \times P(y_3, y_4 | A) \times P(y_5 | A) \times P(A) = 0.9847 \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \approx 0.0469$$

$$P(B|x_8)$$

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

$$P(y_1, y_2 | B) = P(0.42, 0.59 | B) \quad (\text{do mesmo modo que acima, mas } (y_1, y_2 | B) \sim N(\vec{\mu}_B, \Sigma_B))$$

$$\Rightarrow P(0.38, 0.52 | A) = \text{PDF}_B([0.38, 0.52]) = 1.9659$$

$$\bullet P(B|x_8) = P(y_1, y_2 | B) \times P(y_3, y_4 | B) \times P(y_5 | B) \times P(B) = 1.9659 \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} = 0.0702$$

$$\boxed{\text{Para } x_9} \quad P(A|x_9) = P(y_1, y_2 | A) \times P(y_3, y_4 | A) \times P(y_5 | A) \times P(A) = P(0.42, 0.59 | A) P(0, 1 | A) P(1 | A) P(A)$$

$$P(B|x_9) = P(y_1, y_2 | B) \times P(y_3, y_4 | B) \times P(y_5 | B) \times P(B) = P(0.42, 0.59 | B) P(0, 1 | B) P(1 | B) P(B)$$

$$P(A|x_9)$$

$$P(y_3, y_4 | A) = P(0, 1 | A) = \frac{1}{3} \quad P(y_5 | A) = P(1 | A) = \frac{1}{3}$$

$$P(y_1, y_2 | A) = P(0.42, 0.59 | A) \stackrel{?}{=} \text{PDF}_A([0.42, 0.59]) = 0.4031$$

do mesmo modo que acima ( $(y_1, y_2 | A) \sim N(\vec{\mu}_A, \Sigma_A)$ )

$$\bullet P(A|x_9) = P(0.42, 0.59 | A) P(0, 1 | A) P(1 | A) P(A) = 0.4031 \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \approx 0.0192$$

$$P(B|x_9)$$

$$P(y_3, y_4 | B) = P(0, 1 | B) = \frac{1}{4} \quad P(y_5 | B) = P(1 | B) = \frac{1}{2}$$

$$P(y_1, y_2 | B) = P(0.42, 0.59 | B) \stackrel{?}{=} \text{PDF}_B([0.42, 0.59]) = 1.7318$$

$(y_1, y_2 | B) \sim N(\vec{\mu}_B, \Sigma_B)$

$$\bullet P(B|x_9) = P(0.42, 0.59 | B) P(0, 1 | B) P(1 | B) P(B) = 1.7318 \times \frac{1}{4} \times \frac{1}{2} \times \frac{1}{4} = 0.127$$

Com base nos resultados obtidos, pode-se concluir que a classificação segundo MAP

para  $x_8$  será  $\max\{P(A|x_8), P(B|x_8)\} = \max\{0.0469, 0.0702\} = 0.0702 \rightarrow$  classifica-se  $x_8$  como B,

e para  $x_9$  será  $\max\{P(A|x_9), P(B|x_9)\} = \max\{0.0192, 0.127\} = 0.127 \rightarrow$  classifica-se  $x_9$  como B.

$$\text{c)} \quad \boxed{x_8} \quad P(x_8 | A) = \frac{P(A|x_8) P(x_8)}{P(A)} = \frac{P(y_1, y_2 | A) \times P(y_3, y_4 | A) \times P(y_5 | A) \times P(A) \times P(x_8)}{P(A)}$$

$$= P(x_8) \times 0.9847 \times \frac{1}{3} \times \frac{1}{3} = 0.1094 P(x_8)$$

$$P(x_8 | B) = \frac{P(B|x_8) P(x_8)}{P(B)} = \frac{P(y_1, y_2 | B) \times P(y_3, y_4 | B) \times P(y_5 | B) \times P(B) \times P(x_8)}{P(B)} = 1.9659 \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times P(x_8)$$

$$= 0.1229 P(x_8)$$

Vamos aplicar ML a  $x_8$ : ao comparar, ignorar-se  $P(x_8)$ , e fazermos  $\max \{0.1084; 0.1229\} = 0.1229$ ; logo, classifica-se  $x_8$  como B segundo ML.

x<sub>8</sub> De modo semelhante a  $x_8$ :  $P(x_9|A) = 0.4031 \times \frac{1}{3} \times \frac{1}{3} \times P(x_9) = 0.0448 P(x_9)$

$$P(x_9|B) = 1.7318 \times \frac{1}{4} \times \frac{1}{2} P(x_9) = 0.2165 P(x_9)$$

Novamente, aplicando ML apenas estamos interessados em comparar as probabilidades, logo:

$$\max \{0.0448; 0.2165\} = 0.2165 \rightarrow \text{ML classifica } x_9 \text{ como B.}$$

$$\underline{P(A) = P(B) = 0.5 \text{ (novos priors)}}$$

$$P(A|x_9) = 0.9847 \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{2} \approx 0.0597 \quad P(A|x_9) = 0.4031 \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{2} \approx 0.0224$$

$$P(B|x_9) = 1.9659 \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{2} \approx 0.0614 \quad P(B|x_9) = 1.7318 \times \frac{1}{4} \times \frac{1}{2} \times \frac{1}{2} \approx 0.1082$$

### • Normalizações de $P(A|x_9)$ e $P(B|x_9)$

$$P(A|x_9) = \frac{0.0597}{0.0597 + 0.0614} \approx 0.4711 \quad P(A|x_9) = \frac{0.0224}{0.0224 + 0.1082} \approx 0.1715$$

O algoritmo ML identificou  $x_9$  e  $x_8$  como B. logo,  $\theta$  tem de pertencer ao intervalo  $[0.47; 1]$  para que  $f(x|\theta)$  decida  $x_8$  e  $x_9$  ambos como B (cumprindo a condição  $P(A|x) < \theta$ ). Conclui-se que o preцiso de teste é entro maximizado para  $\theta \in [0.47; 1]$ .

### (2.) a) Binarizaçao de $y_2$ (aqui usamos $x_8$ e $x_9$ tambem):

Vamos binarizar os valores de  $y_2$  com base em dois bins (queremos que assumam um de dois valores)

c/ o mesmo range: escolhem-se os bins  $[0; 0.5[$  e  $[0.5; 1]$  (domínio  $[0; 1]$ ), com base nas  $F(x_i)$ .

$$\{x_1, x_2, x_4, x_5, x_6\} \in [0; 0.5[ \text{ (ficam iguais a 0)}$$

$$\{x_3, x_7, x_8, x_9\} \in [0.5; 1] \text{ (ficam iguais a 1)}$$

Ap s binarizaçao →

D	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	
X1	0.24	0.56	0	1	1	0	A
X2	0.16	0.48	0	1	0	1	A
X3	0.32	0.52	1	0	1	2	A
X4	0.54	0.51	0	0	0	1	B
X5	0.66	0.59	0	0	0	0	B
X6	0.76	0.56	0	1	0	2	B
X7	0.41	0.53	1	0	1	1	B
X8	0.38	0.52	1	0	1	0	A
X9	0.42	0.59	1	0	1	1	B

Identificaçao dos 3 folds (sem shuffle) ap s binarizaçao:

$$\text{Fold 1: } \begin{cases} x_1 = (0.24, 0, 1, 1, 0, A) \\ x_2 = (0.16, 0, 1, 0, 1, A) \\ x_3 = (0.32, 1, 0, 1, 2, A) \end{cases} \quad \text{Fold 2: } \begin{cases} x_4 = (0.54, 0, 0, 0, 1, B) \\ x_5 = (0.66, 0, 0, 0, 0, B) \\ x_6 = (0.76, 0, 1, 0, 2, B) \end{cases} \quad \text{Fold 3: } \begin{cases} x_7 = (0.41, 1, 0, 1, 1, B) \\ x_8 = (0.38, 1, 0, 1, 0, A) \\ x_9 = (0.42, 1, 0, 1, 1, B) \end{cases}$$

b) Pelo enunciado, vamos utilizar as observaçoes dos folds 1 e 2 para treino e do fold 3 para teste. ( $x_1 - x_6$ )

• Distâncias de Hamming (qtd. de vars. diferentes de  $y_2 - y_6$  para cada par)

x<sub>1</sub>  $d(x_1, x_1) = 4 \quad d(x_1, x_2) = 4 \quad d(x_1, x_3) = 2 \quad d(x_1, x_4) = 2 \quad d(x_1, x_5) = 3 \quad d(x_1, x_6) = 4$

KNN,  $k = 3 \rightarrow$  escolhe-se  $d(x_1, x_3)$ ,  $d(x_1, x_4)$  e  $d(x_1, x_5)$ .

x<sub>2</sub>  $d(x_2, x_1) = 2 \quad d(x_2, x_2) = 4 \quad d(x_2, x_3) = 1 \quad d(x_2, x_4) = 4 \quad d(x_2, x_5) = 3 \quad d(x_2, x_6) = 5$

KNN,  $k = 3 \rightarrow$  escolhe-se  $d(x_2, x_1)$ ,  $d(x_2, x_3)$  e  $d(x_2, x_5)$ .

$$x_3 \quad d(x_3, x_1) = 4 \quad d(x_3, x_2) = 4 \quad d(x_3, x_3) = 2 \quad d(x_3, x_4) = 2 \quad d(x_3, x_5) = 3 \quad d(x_3, x_6) = 4$$

KNN, k=3 → escolhe-se  $d(x_3, x_3)$ ,  $d(x_3, x_4)$  e  $d(x_3, x_5)$ .

### • Cálculo do MAE

Target variable é  $y_1$  e o peso é  $1/d$ , logo:

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

• Cálculo de  $\hat{z}$  (para observações de teste, logo  $\hat{z} = \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \hat{z}_3 \end{bmatrix}$ )

$$\hat{z}_1 = \frac{1}{d(x_3, x_3)} y_1(x_3) + \frac{1}{d(x_3, x_4)} y_1(x_4) + \frac{1}{d(x_3, x_5)} y_1(x_5) = \frac{0.32}{2} + \frac{0.54}{2} + \frac{0.66}{3} = 0.65$$

$$\hat{z}_2 = \frac{y_1(x_1)}{d(x_2, x_1)} + \frac{y_1(x_3)}{d(x_2, x_3)} + \frac{y_1(x_5)}{d(x_2, x_5)} = \frac{0.24}{2} + \frac{0.32}{1} + \frac{0.66}{3} = 0.66$$

$$\hat{z}_3 = \frac{y_1(x_2)}{d(x_3, x_2)} + \frac{y_1(x_4)}{d(x_3, x_4)} + \frac{y_1(x_5)}{d(x_3, x_5)} = \frac{0.32}{2} + \frac{0.54}{2} + \frac{0.66}{3} = 0.65$$

$$\therefore \hat{z} = \begin{bmatrix} 0.65 \\ 0.66 \\ 0.65 \end{bmatrix} \quad \text{Por observação direta, } z = \begin{bmatrix} 0.41 \\ 0.39 \\ 0.42 \end{bmatrix}$$

$$MAE(\hat{z}, z) = \frac{1}{n} \sum_{i=1}^n |z_i - \hat{z}_i| = \frac{1}{3} (|0.65 - 0.41| + |0.66 - 0.39| + |0.65 - 0.42|) = 0.25$$

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

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.

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

- Plot two boxplots with the fold accuracies for each predictor.

```
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 import metrics
import matplotlib.pyplot as plt

# Reading the 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']

knn_predictor = KNeighborsClassifier(n_neighbors=5)
nb_predictor = GaussianNB()

# 10-fold stratified cross-validator with shuffling
folds = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

knn_accuracies = []
nb_accuracies = []

# iterate per fold
for train_k, test_k in folds.split(X, y):
    X_train, X_test = X.iloc[train_k], X.iloc[test_k]
    y_train, y_test = y.iloc[train_k], y.iloc[test_k]

    # train and assess k-NN
    knn_predictor.fit(X_train, y_train)
    y_pred = knn_predictor.predict(X_test)
    knn_accuracies.append(round(metrics.accuracy_score(y_test, y_pred), 2))

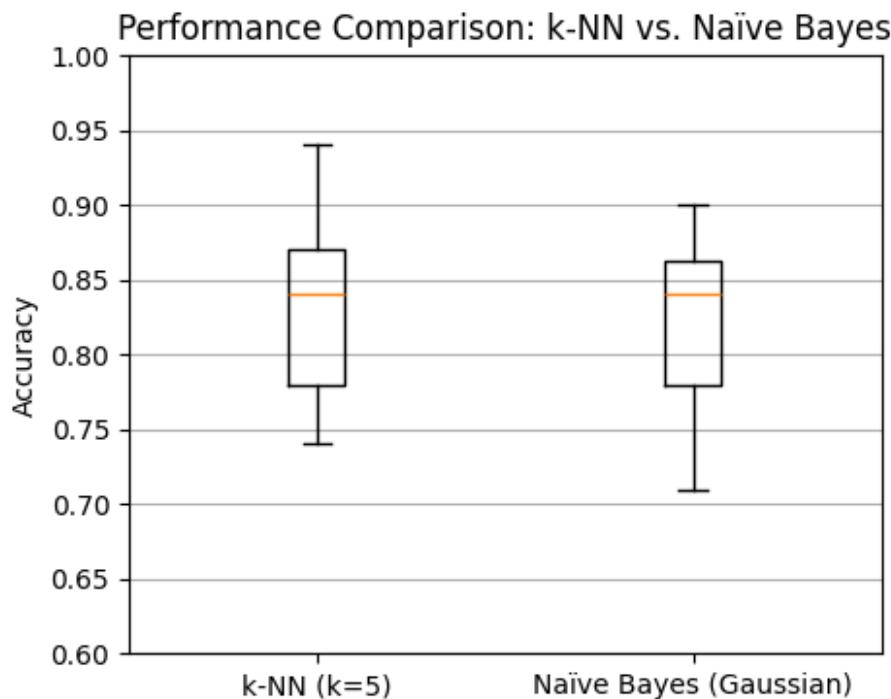
    nb_accuracies.append(round(metrics.accuracy_score(y_test, nb_predictor.predict(X_test)), 2))
```

```

# train and assess Naïve Bayes with Gaussian assumption
nb_predictor.fit(X_train, y_train)
y_pred = nb_predictor.predict(X_test)
nb_accuracies.append(round(metrics.accuracy_score(y_test, y_pred), 2))

# plots
plt.figure(figsize=(5, 4))
plt.boxplot([knn_accuracies, nb_accuracies], labels=['k-NN (k=5)', 'Naïve Bayes (Gaussian)'])
plt.title('Performance Comparison: k-NN vs. Naïve Bayes')
plt.ylabel('Accuracy')
plt.ylim(0.6, 1)
plt.grid(axis='y')
plt.show()

```



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

```

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 import metrics

```

```

from scipy import stats

# Reading the ARFF file
data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df.head()

X = df.drop('class', axis=1)
y = df['class']

knn_predictor = KNeighborsClassifier(n_neighbors=5)
nb_predictor = GaussianNB()

# 10-fold stratified cross-validator with shuffling
folds = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

knn_accuracies = []
nb_accuracies = []

# iterate per fold
for train_k, test_k in folds.split(X, y):
    X_train, X_test = X.iloc[train_k], X.iloc[test_k]
    y_train, y_test = y.iloc[train_k], y.iloc[test_k]

    # train and assess k-NN
    knn_predictor.fit(X_train, y_train)
    y_pred = knn_predictor.predict(X_test)
    knn_accuracies.append(round(metrics.accuracy_score(y_test, y_pred), 2))

    # train and assess Naïve Bayes with Gaussian assumption
    nb_predictor.fit(X_train, y_train)
    y_pred = nb_predictor.predict(X_test)
    nb_accuracies.append(round(metrics.accuracy_score(y_test, y_pred), 2))

# kNN is better than naïve Bayes?
res = stats.ttest_rel(knn_accuracies, nb_accuracies,
alternative='greater')
print("kNN > naïve Bayes? pval=", res.pvalue)

kNN > naïve Bayes? pval= 0.1734666237861796

```

## Comentário:

- $k\text{NN} > \text{náive Bayes? } p\text{val}= 0.1734666237861796$
- como  $p\text{val} >$  valores de significância usuais (ou seja,  $p\text{val}$  é superior a 0.01, 0.05 e 0.1), não é possível rejeitar a hipótese nula e afirmar que o  $k\text{NN}$  é estatisticamente superior ao Naive Bayes.

## 2.

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
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.io import arff
import pandas as pd
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix

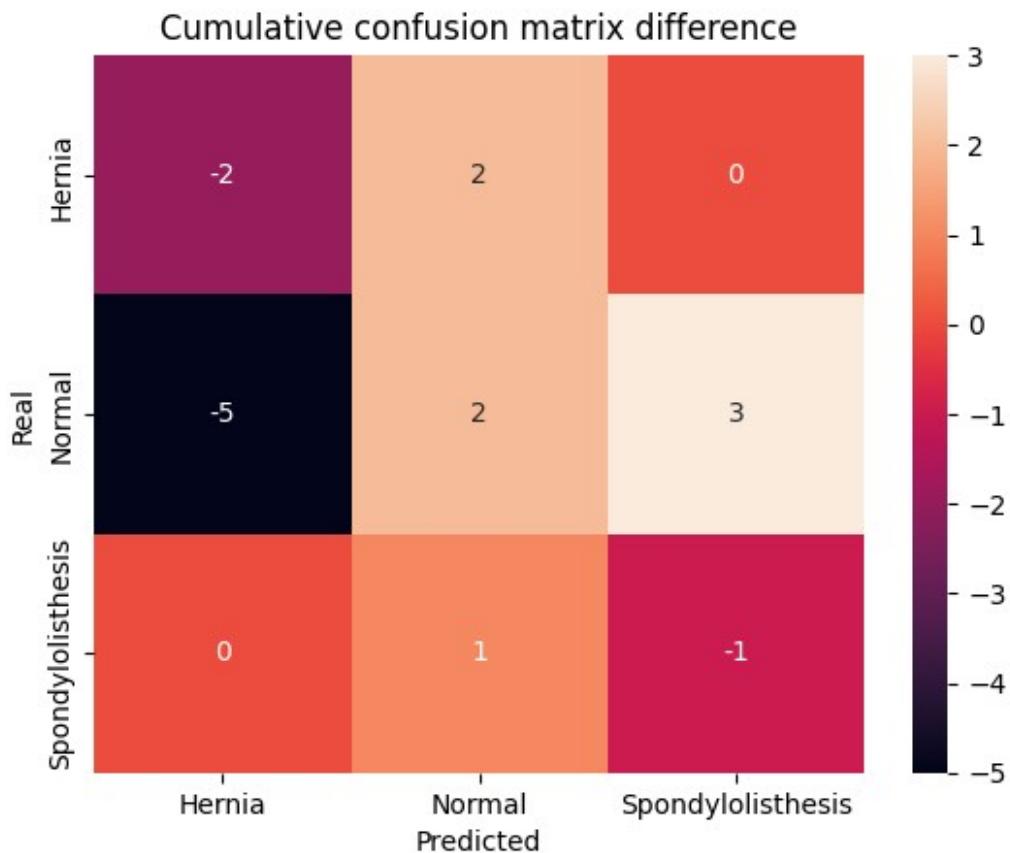
# reads file, extracts relevant data and initializes relevant
variables
data = arff.loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
X = df.drop('class', axis=1)
y = df['class']

knn1, knn5 = KNeighborsClassifier(n_neighbors=1),
KNeighborsClassifier(n_neighbors=5)
kfolds = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
cumul_cm_k1, cumul_cm_k5 = np.zeros((3, 3), dtype=int), np.zeros((3,
3), dtype=int)

# performs cross-validation and calculates cumulative confusion
matrices
for train_index, test_index in kfolds.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]
    # k=1
    knn1.fit(X_train, y_train)
    ypred_k1 = knn1.predict(X_test)
    cm_k1 = confusion_matrix(y_test, ypred_k1)
    cumul_cm_k1 += cm_k1
    # k=5
    knn5.fit(X_train, y_train)
    ypred_k5 = knn5.predict(X_test)
    cm_k5 = confusion_matrix(y_test, ypred_k5)
    cumul_cm_k5 += cm_k5

# calculates matrix of difference between both matrixes and labels it
difference = cumul_cm_k1 - cumul_cm_k5
difference = pd.DataFrame(difference, index=['Hernia', 'Normal',
'Spondylolisthesis'], columns=['Hernia', 'Normal'],
```

```
'Spondylolisthesis'])
# plots the heatmap
sns.heatmap(difference, annot=True, fmt='g')
plt.title('Cumulative confusion matrix difference')
plt.xlabel('Predicted')
plt.ylabel('Real')
plt.show()
```



### Comentário dos resultados obtidos:

Relativamente aos resultados obtidos, podemos observar que, apesar da diferença não ser muito significativa, o preditor kNN com  $k=5$  apresenta maior precisão a identificar hérnias e casos de "spondylolisthesis" do que o preditor kNN com  $k=1$ , sendo que este último é apenas mais preciso a identificar casos normais. O classificador com  $k=1$  apresenta mais falsos positivos e falsos negativos para o caso de uma hérnia e para o caso de "spondylolisthesis" do que o classificador com  $k=5$ , e este último ( $k=5$ ) apresenta mais falsos positivos mas menos falsos negativos para casos normais (do que o classificador  $k=1$ ). Globalmente, e como já comentado, os resultados obtidos não diferem muito, mas o classificador  $k=5$  apresenta, expectavelmente, um melhor desempenho e maior precisão do que o classificador  $k=1$ .

### 3.

Considering the unique properties of column\_diagnosis, identify three possible difficulties of naïve Bayes when learning from the given dataset.

Com base no dataset fornecido, três possíveis dificuldades do algoritmo de Naive Bayes em aprender com base no mesmo são, primeiramente, a suposição de independência condicional entre as variáveis que caracterizam os dados; neste caso, é bastante possível que existam dependências entre as variáveis, uma vez que três delas incidem sobre a pélvis, por exemplo, e existe possivelmente mais dependências entre estas variáveis - logo, daqui pode advir uma perda de precisão. De seguida, outro problema que pode enfrentar é desequilíbrio entre classes, uma vez que, por exemplo, a classe Spondylolisthesis apresenta maior frequência relativa (mais do dobro do que a classe hérnia, e 50% mais do que a classe normal), a classe normal apresenta maior frequência relativa que a classe hérnia mas menos que a Spondylolisthesis, e a classe hérnia apresenta menos frequência relativa que ambas (existem 150 registos da classe Spondylolisthesis, 100 da classe normal e 60 da classe hérnia), o que pode levar o modelo a ter menor precisão que o desejado para classes com menor frequência, e potencialmente a ter um viés para classes com maior frequência. Por fim (e apesar desta dificuldade parecer ser talvez a menos preocupante ou não tão relevante), a abordagem de Naive Bayes também é suscetível a outliers, que podem influenciar o cálculo das estimativas de probabilidade e consequentemente a precisão do modelo para cada classe e em geral.