Aprendizagem 2022/23

Homework II – Group 001

(ist1102082, ist1103252)

I. Pen-and-paper

1)

a)

1) a) $\{y_1, y_2\}, \{y_3, y_4\}, \{y_5\}$ → sets of independent variables

$y_1 \times y_2 \in \mathbb{R}^2$ → These variables can be described by a Gaussian distribution.

→ For $y_6 = A$:

$\{y_1, y_2\}$ We will be calculating the Gaussian distribution parameters (μ, Σ)

• μ_1: $\bar{y_1} = \frac{0.24 + 0.16 + 0.32}{3} = 0.24$, $\bar{y_2} = \frac{0.36 + 0.48 + 0.72}{3} = 0.52$ $\Rightarrow \mu_1 = \begin{bmatrix} 0.24 \\ 0.52 \end{bmatrix}$

• $\Sigma_1 = \begin{pmatrix} \text{var}y_1 & \text{cov}y_1y_2 \\ \text{cov}y_1y_2 & \text{var}y_2 \end{pmatrix}$, with $\text{var} = \frac{\sum (x_i - \bar{x})^2}{n-1}$; $\text{cov}y_1y_2 = \frac{\sum (y_{1i} - \bar{y_1})(y_{2i} - \bar{y_2})}{n-1}$

• $\text{var}\, y_1 = \frac{(0.24 - 0.24)^2 + (0.16 - 0.24)^2 + (0.32 - 0.24)^2}{3-1} = 0.0064$

• $\text{var}\, y_2 = \frac{(0.36 - 0.52)^2 + (0.48 - 0.52)^2 + (0.72 - 0.52)^2}{3-1} = 0.0336$

• $\text{cov}y_1y_2 = \frac{(0.24 - 0.24)(0.36 - 0.52) + (0.16 - 0.24)(0.48 - 0.52) + (0.32 - 0.24)(0.72 - 0.32)}{3-1} = 0.0096$

Therefore, $\Sigma_1 = \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix}$

• $\det \Sigma = 0.0064 \times 0.0336 + 0.0096 \times 0.0096 = 0.000123$

• $\Sigma_1^{-1} = \frac{1}{\det_1} \Sigma = \frac{1}{0.000123} \begin{bmatrix} 0.0064 & 0.0096 \\ 0.0096 & 0.0336 \end{bmatrix} = \begin{bmatrix} 273.17 & 78.05 \\ 78.05 & 52.03 \end{bmatrix}$

• $P(y_1, y_2 \mid y_6 = A) = N(y_1, y_2 \mid \mu_1, \Sigma_1) =$

$= \frac{1}{(2\pi)^{\frac{d}{2}} \cdot |\Sigma_1|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}\left(\{y_1, y_2\} - \mu_1\right)^T \cdot \Sigma_1^{-1} \cdot \left(\{y_1, y_2\} - \mu_1\right)\right)$

Homework II – Group 001

(ist1102082, ist1103252)

$\{Y_3, Y_4\}$ $P(Y_3, Y_4 | Y_6 = A)$:

- $P(Y_3=0, Y_4=0 | Y_6=A) = 0$
- $P(Y_3=1, Y_4=0 | Y_6=A) = \frac{1}{3}$
- $P(Y_3=0, Y_4=1 | Y_6=A) = \frac{1}{3}$
- $P(Y_3=1, Y_4=2 | Y_6=A) = \frac{1}{3}$

$\{Y_5\}$ $P(Y_5 | Y_6 = A)$:

- $P(Y_5=0 | Y_6=A) = \frac{1}{3}$
- $P(Y_5=1 | Y_6=A) = \frac{1}{3}$
- $P(Y_5=2 | Y_6=A) = \frac{1}{3}$

Therefore:

$$P(Y_6=A | Y_1, Y_2, Y_3, Y_4, Y_5) = \frac{P(Y_1, Y_2, Y_3, Y_4, Y_5 | Y_6=A) \times P(Y_6=A)}{P(Y_1, Y_2, Y_3, Y_4, Y_5)} =$$

☆ We can conclude this since $\{Y_1, Y_2\}, \{Y_3, Y_4\}, \{Y_5\}$ are in dependent sets

$$\overset{☆}{=} \frac{P(Y_1, Y_2 | Y_6=A) \times P(Y_3, Y_4 | Y_6=A) \times P(Y_5 | Y_6=A) \times P(Y_6=A)}{P(Y_1, Y_2, Y_3, Y_4, Y_5)}$$

$$= \frac{N\left(\begin{bmatrix}0.24\\0.52\end{bmatrix}, \begin{bmatrix}0.0064 & 0.0096\\0.0096 & 0.0336\end{bmatrix}\right) \times P(Y_3, Y_4 | Y_6=A) \times P(Y_5 | Y_6=A) \times P(Y_6=A)}{P(Y_1, Y_2, Y_3, Y_4, Y_5)}$$

For $Y_6 = B$: (Note that we will be using the same logic used for $Y_6 = A$. Therefore, our calculations will be ommited)

$\{Y_1, Y_2\}$: $\mu_2 = \begin{bmatrix}0.5925\\0.3275\end{bmatrix}$ $\Sigma_2 \simeq \begin{bmatrix}0.0229 & -0.0098\\-0.0098 & 0.0375\end{bmatrix}$

$\det_2 = 0.000625$ $\Sigma_2^{-1} \simeq \begin{bmatrix}36.64 & 15.68\\15.68 & 50.4\end{bmatrix}$

$$P(Y_1, Y_2 | Y_6=B) = N(Y_1, Y_2 | \mu_2, \Sigma_2) =$$

$$= \frac{1}{(2\pi)^{\frac{d}{2}} \cdot |\Sigma_2|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}\left(\{Y_1, Y_2\} - \mu_2\right)^T \cdot \Sigma_2^{-1} \cdot \left(\{Y_1, Y_2\} - \mu_2\right)\right)$$

Homework II – Group 001

(ist1102082, ist1103252)

$\{y_1, y_2\}$ $P(y_3, y_4 | y_6 = B)$:

- $P(y_3 = 0, y_4 = 0 | y_6 = B) = \frac{1}{2}$
- $P(y_3 = 1, y_4 = 0 | y_6 = B) = \frac{1}{4}$
- $P(y_3 = 0, y_4 = 1 | y_6 = B) = \frac{1}{4}$
- $P(y_3 = 1, y_4 = 1 | y_6 = B) = 0$

$\{y_5\}$ $P(y_5 | y_6 = B)$:

- $P(y_5 = 0 | y_6 = B) = \frac{1}{4}$
- $P(y_5 = 1 | y_6 = B) = \frac{1}{2}$
- $P(y_5 = 2 | y_6 = B) = \frac{1}{4}$

Therefore:

$$P(y_6 = B | y_1, y_2, y_3, y_4, y_5) = \frac{P(y_1, y_2, y_3, y_4, y_5 | y_6 = B) \times P(y_6 = B)}{P(y_1, y_2, y_3, y_4, y_5)} =$$

☆ We can conclude this since $\{y_1, y_2\}, \{y_3, y_4\}, \{y_5\}$ are independent sets

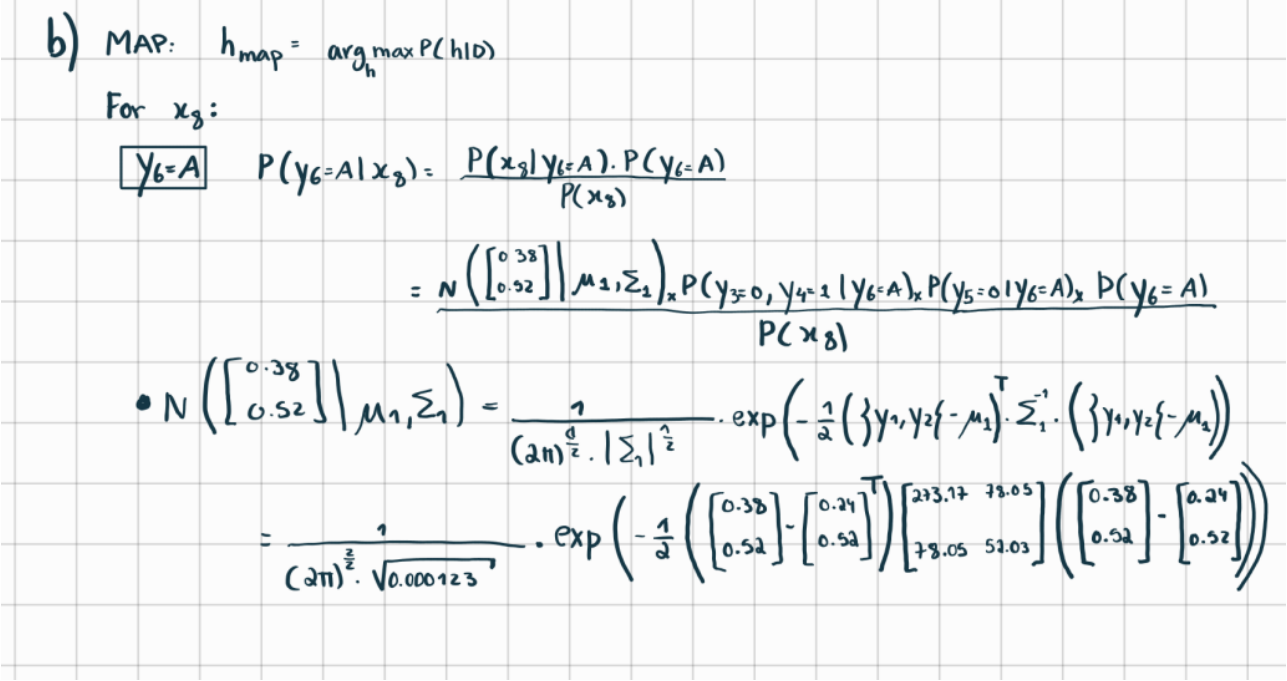
$$\overset{☆}{=} \frac{P(y_1, y_2 | y_6 = B) \times P(y_3, y_4 | y_6 = B) \times P(y_5 | y_6 = B) \times P(y_6 = B)}{P(y_1, y_2, y_3, y_4, y_5)}$$

$$= \frac{N\left(\begin{bmatrix} 0.5926 \\ 0.3275 \end{bmatrix}, \begin{bmatrix} 0.0229 & -0.0093 \\ -0.0093 & 0.0315 \end{bmatrix}\right) \times P(y_3, y_4 | y_6 = B) \times P(y_5 | y_6 = B) \times P(y_6 = B)}{P(y_1, y_2, y_3, y_4, y_5)}$$

Homework II – Group 001

(ist1102082, ist1103252)

b)



$$= 14.35 \times \exp\left(\begin{bmatrix}-0.07 & 0\end{bmatrix}\begin{bmatrix}273.17 & 78.05\\ 78.05 & 52.03\end{bmatrix}\begin{bmatrix}0.14\\ 0\end{bmatrix}\right)$$

$$= 14.35 \times \exp\left(\begin{bmatrix}-0.07 & 0\end{bmatrix}\begin{bmatrix}273.17 & 78.05\\ 78.05 & 52.03\end{bmatrix}\begin{bmatrix}0.14\\ 0\end{bmatrix}\right) = 0.9847$$

- $P(y_3=0, y_4=1 \mid y_6=A) = \frac{1}{3}$
- $P(y_5=0 \mid y_6=A) = \frac{1}{3}$
- $P(y_6=A) = \frac{3}{7}$

Therefore: $= \dfrac{N\left(\begin{bmatrix}0.38\\ 0.52\end{bmatrix} \middle| \mu_1, \Sigma_1\right) \times P(y_3=0, y_4=1 \mid y_6=A) \times P(y_5=0 \mid y_6=A) \times P(y_6=A)}{P(x_8)}$

$$= \frac{0.9839 \times \frac{1}{3} \times \frac{1}{3} \times \frac{3}{7}}{P(x_8)} = \frac{0.04685}{P(x_8)}$$

$\boxed{y_6=B}$ $\quad P(y_6=B \mid x_8) = \dfrac{P(x_8 \mid y_6=B) \cdot P(y_6=B)}{P(x_8)} =$

$$= \frac{N\left(\begin{bmatrix}0.38\\ 0.52\end{bmatrix} \middle| \mu_2, \Sigma_2\right) P(y_3=0, y_4=1 \mid y_6=B) P(y_5=0 \mid y_6=B) P(y_6=B)}{P(x_8)} =$$

$$= \frac{1.9624 \times \frac{1}{4} \times \frac{1}{4} \times \frac{4}{7}}{P(x_8)} = \frac{0.0701}{P(x_8)}$$

Note: Since $P(x_8)$ does not change depending if $y_6=A$ or $y_6=B$, it's value is constant and equal in both scenarios. Hence, we do not have to calculate it.

Therefore, after calculating MAP, $P(y_6=B \mid x_8) > P(y_6=A \mid x_8)$, we must conclude that x_8 should be classified has **B**.

For x_q:

Since the logic is the same used in x_8, Calculations will be ommited.

$\boxed{y_6 = A}$

$$\frac{P(y_6 = A \mid x_q) = P(y_6 = A)}{P(x_q)} =$$

$$= \frac{N\left(\begin{bmatrix}0.42\\0.59\end{bmatrix} \middle| \mu_2, \Sigma_2\right) \times P(y_3 = 0, y_4 = 1 \mid y_6 = A) \times P(y_5 = 1 \mid y_6 = A) \times P(y_6 = A)}{P(x_q)}$$

$$= \frac{0.4031 \times \frac{1}{3} \times \frac{1}{3} \times \frac{3}{7}}{P(x_q)} = \frac{0.0191}{P(x_q)}$$

$\boxed{y_6 = B}$

$$\frac{P(y_6 = B \mid x_q) = P(y_6 = B)}{P(x_q)} =$$

$$= \frac{N\left(\begin{bmatrix}0.42\\0.59\end{bmatrix} \middle| \mu_2, \Sigma_2\right) \times P(y_3 = 0, y_4 = 1 \mid y_6 = B) \times P(y_5 = 1 \mid y_6 = B) \times P(y_6 = B)}{P(x_q)}$$

$$= \frac{1.7236 \times \frac{1}{4} \times \frac{1}{2} \times \frac{4}{7}}{P(x_q)} = \frac{0.1233}{P(x_q)}$$

Therefore, after calculating MAP, $P(y_6 = B \mid x_q) > P(y_6 = A \mid x_q)$, we must conclude that x_q should be classified has **B**.

2)

Aprendizagem 2022/23

Homework II – Group 001

(ist1102082, ist1103252)

a)

2. a)

- First, we will apply the binarization procedure in y_2, under an equal-width discretization. Meaning, $y_2 < 0.5 = 0$, $y_2 \geq 0.5 = 1$.

$y_2 = [0.36 \quad 0.48 \quad 0.72 \quad 0.11 \quad 0.39 \quad 0.28 \quad 0.53 \quad 0.52 \quad 0.59]$

$y_2' = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1]$

- Using a 3-fold cross-validation over the full dataset, without shuffling the observations, we will end up with the following folds:

Fold 1: $\{x_1, x_2, x_3\}$ Fold 2: $\{x_4, x_5, x_6\}$ Fold 3: $\{x_7, x_8, x_9\}$

Therefore, here are the observations and features per data fold after the binarization procedure:

		y_2	y_3	y_4	y_5	y_6
Fold 1	x_1	0	1	1	0	A
	x_2	0	1	0	1	A
	x_3	1	0	1	2	A
Fold 2	x_4	0	0	0	1	B
	x_5	0	0	0	0	B
	x_6	0	1	0	2	B
Fold 3	x_7	1	0	1	1	B
	x_8	1	0	1	0	A
	x_9	1	0	1	1	B

b)

b)

- distance-weighted KNN, K=3
- Hamming distance, d
- weighting = $\frac{1}{d}$

- lower index → train:
  Fold 1, Fold 2 → train
  Fold 3 → test

- $MAE = \frac{1}{n}\sum_{i=1}^{n}|z-\hat{z}|$

	x_7	x_8	x_9
Hamming distances	$d(x_7,x_1)=4$ $d(x_7,x_2)=4$ $d(x_7,x_3)=2$ $d(x_7,x_4)=2$ $d(x_7,x_5)=3$ $d(x_7,x_6)=4$	$d(x_8,x_1)=2$ $d(x_8,x_2)=4$ $d(x_8,x_3)=1$ $d(x_8,x_4)=4$ $d(x_8,x_5)=3$ $d(x_8,x_6)=5$	$d(x_9,x_1)=4$ $d(x_9,x_2)=4$ $d(x_9,x_3)=2$ $d(x_9,x_4)=2$ $d(x_9,x_5)=3$ $d(x_9,x_6)=4$
Closest K=3 observations	x_3, x_4, x_5 (2, 2, 3)	x_1, x_3, x_5 (2, 1, 3)	x_3, x_4, x_5 (2, 2, 3)
weights	$w_{x_3}=\frac{1}{2}$ $w_{x_4}=\frac{1}{2}$ $w_{x_5}=\frac{1}{3}$	$w_{x_1}=\frac{1}{2}$ $w_{x_3}=1$ $w_{x_5}=\frac{1}{3}$	$w_{x_3}=\frac{1}{2}$ $w_{x_4}=\frac{1}{2}$ $w_{x_5}=\frac{1}{3}$
w_{AVG}	$\frac{\frac{1}{2}\times0.32+\frac{1}{2}\times0.54+\frac{1}{3}\times0.66}{\frac{1}{2}+\frac{1}{2}+\frac{1}{3}} = 0.4875$	$\frac{\frac{1}{2}\times0.24+1\times0.32+\frac{1}{3}\times0.66}{\frac{1}{2}+1+\frac{1}{3}} = 0.36$	$\frac{\frac{1}{2}\times0.32+\frac{1}{2}\times0.54+\frac{1}{3}\times0.66}{\frac{1}{2}+\frac{1}{2}+\frac{1}{3}} = 0.4875$

Therefore, we can know calculate MAE:

$$MAE = \frac{|0.41-0.4875|+|0.38-0.36|+|0.42-0.4875|}{3} = 0.055$$

Homework II – Group 001

(ist1102082, ist1103252)

c)

1.C)

- $P(A|x_8) \propto P(x_8|A) = 0.9839 \times \frac{1}{3} \times \frac{1}{3} = 0.1093$

Normalization: $P(x_8|A) = \frac{P(x_8|A)}{P(x_8|A)+P(x_8|B)} = \frac{0.1093}{0.1093+0.1227} = \underbrace{0.4712}_{\theta < 0.4712}$

- $P(A|x_9) \propto P(x_9|A) = 0.4031 \times \frac{1}{3} \times \frac{1}{3} = 0.04479$

Normalization: $P(x_9|A) = \frac{P(x_9|A)}{P(x_9|A)+P(x_9|B)} = \frac{0.04479}{0.04479+0.2161} = \underbrace{0.1717}_{\theta > 0.1717}$

therefore: R: $\theta \in]0.1717 ; 0.4712[$

II. Programming and critical analysis

1)

A)

Exercise 1 a.

+ Code + Markdown

Loading the data from the arff file and converting it into a dataframe

```
from scipy.io.arff import loadarff
import pandas as pd

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

✓ 0.0s Python

Separating the input data and the output data, required for some sklean functions

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

✓ 0.0s

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 001

(ist1102082, ist1103252)

First, we will apply a 10-fold stratified cross-validation with shuffling for the assessment of predictive models.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from scipy import stats

# Create 10-fold stratified cross-validation with shuffling
cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
```

✓ 0.0s

We will be comparing the performance of KNN with K =5 and naïve Bayes with Gaussian assumption.

```
# Create classifiers
knn_classifier = KNeighborsClassifier(n_neighbors=5)
nb_classifier = GaussianNB()
```

✓ 0.0s

Then, we will perform cross-validation for both the models.

```
# Perform cross-validation and get accuracies for k-NN and Naive Bayes
knn_scores = cross_val_score(knn_classifier, x, y, cv=cv)
print("knn_scores:", knn_scores)
nb_scores = cross_val_score(nb_classifier, x, y, cv=cv)
print("nb_scores", nb_scores)
```

✓ 0.0s

```
knn_scores: [0.93548387 0.80645161 0.87096774 0.93548387 0.74193548 0.87096774
 0.83870968 0.83870968 0.77419355 0.77419355]
nb_scores [0.83870968 0.87096774 0.83870968 0.87096774 0.77419355 0.83870968
 0.90322581 0.80645161 0.77419355 0.70967742]
```

Having the scores for both the models, we will compare them using two boxplots with the fold accuracies for each classifier.

+ Code + Markdown

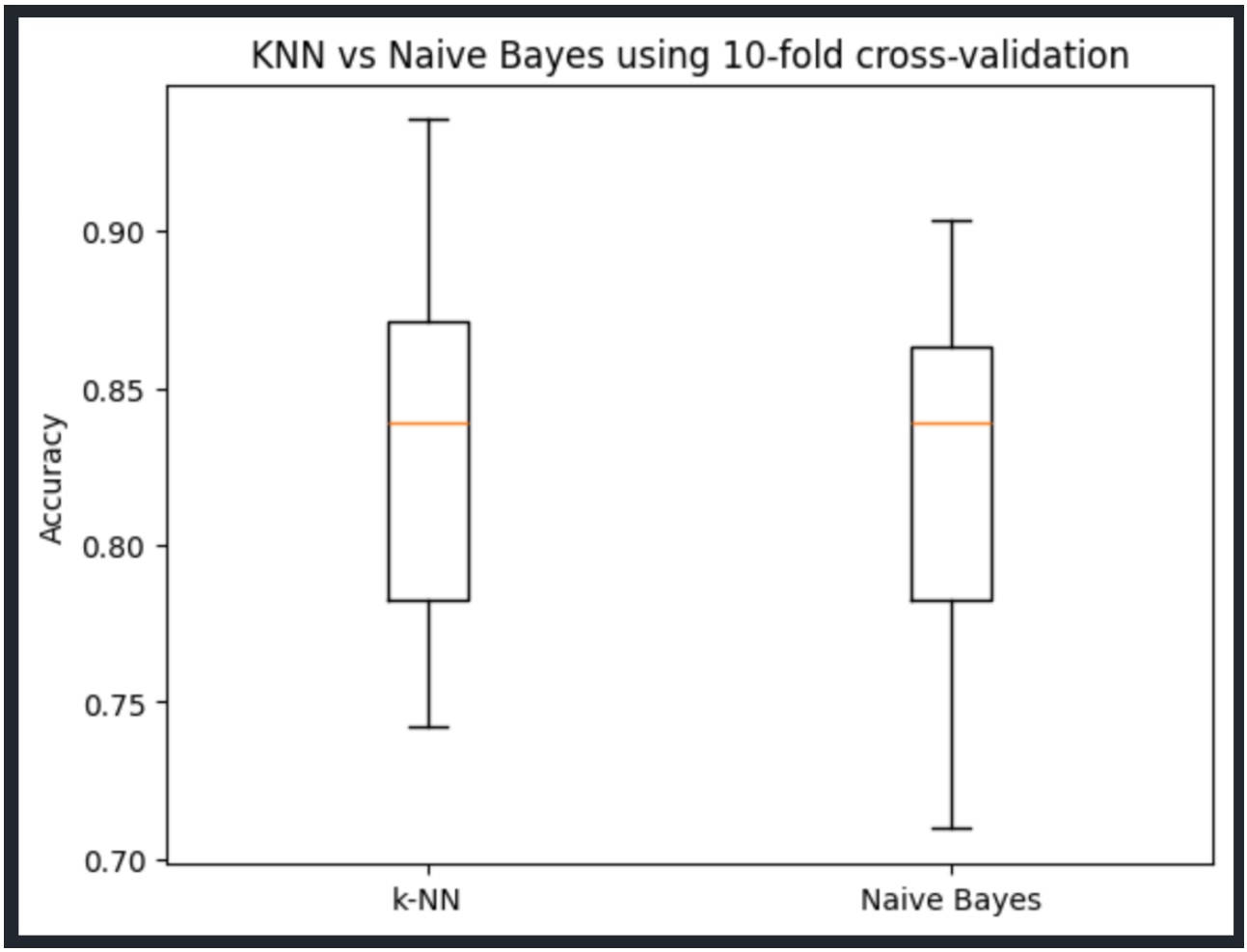
Add Markdown Cell

```
# Plot boxplots for fold accuracies
plt.boxplot([knn_scores, nb_scores], labels=['k-NN', 'Naive Bayes'])
plt.title('KNN vs Naive Bayes using 10-fold cross-validation')
plt.ylabel('Accuracy')
plt.show()
```

✓ 0.0s

Homework II – Group 001

(ist1102082, ist1103252)



B)

Homework II – Group 001

(ist1102082, ist1103252)

To compare the two classifiers, we will perform a statistical test.

```python
# Perform a statistical test to compare the two classifiers
t_statistic , p_value = stats.ttest_rel(knn_scores, nb_scores, alternative='greater')

# Define a significance level (default value of 0.05)
alpha = 0.05

# Check if k-NN is statistically superior to Naive Bayes
if p_value < alpha:
    print("Reject null hypothesis: k-NN is statistically superior to Naive Bayes regarding accuracy")
else:
    print("Fail to reject null hypothesis: No statistical difference between KNN and Naive Bayes")
```

9] ✓ 0.0s

```
t_statistic: 0.9214426752509264
p_value: 0.19042809062064092
Fail to reject null hypothesis: No statistical difference between KNN and Naive Bayes
```

2)

First, similar to the previous exercise, we will apply a 10-fold stratified cross-validation with shuffling for the assessment of predictive models. Also, we will be instantiating a KNN classifier with K = 5 and one with K=1.

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix

# Initialize stratified k-fold cross-validation with shuffling
kf = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

# Initialize k-NN classifiers with k=1 and k=5
knn1 = KNeighborsClassifier(n_neighbors=1)
knn5 = KNeighborsClassifier(n_neighbors=5)
```

✓ 0.0s Python

Then, we will initialize cumulative confusion matrices for both the classifiers, starting with all zeros for each cell.

```python
# Initialize cumulative confusion matrices for both classifiers
cumulative_cm1 = np.zeros((3, 3))
cumulative_cm5 = np.zeros((3, 3))
```

✓ 0.0s

Homework II – Group 001

(ist1102082, ist1103252)

Now, we will perform cross-validation for both the models and update the confusion matrices for each fold.

```
# Perform cross-validation
for train_index, test_index in kf.split(x, y):
    X_train, X_test = x.values[train_index], x.values[test_index]
    y_train, y_test = y.values[train_index], y.values[test_index]

    # Fit k-NN classifiers
    knn1.fit(X_train, y_train)
    knn5.fit(X_train, y_train)

    # Make predictions
    y_pred1 = knn1.predict(X_test)
    y_pred5 = knn5.predict(X_test)

    # Calculate confusion matrices
    cm1 = confusion_matrix(y_test, y_pred1)
    cm5 = confusion_matrix(y_test, y_pred5)

    # Update cumulative confusion matrices
    cumulative_cm1 += cm1
    cumulative_cm5 += cm5
```

✓ 0.0s

Plotting the difference between the two cumulative confusion matrices, we will be able to see the difference between the two classifiers.

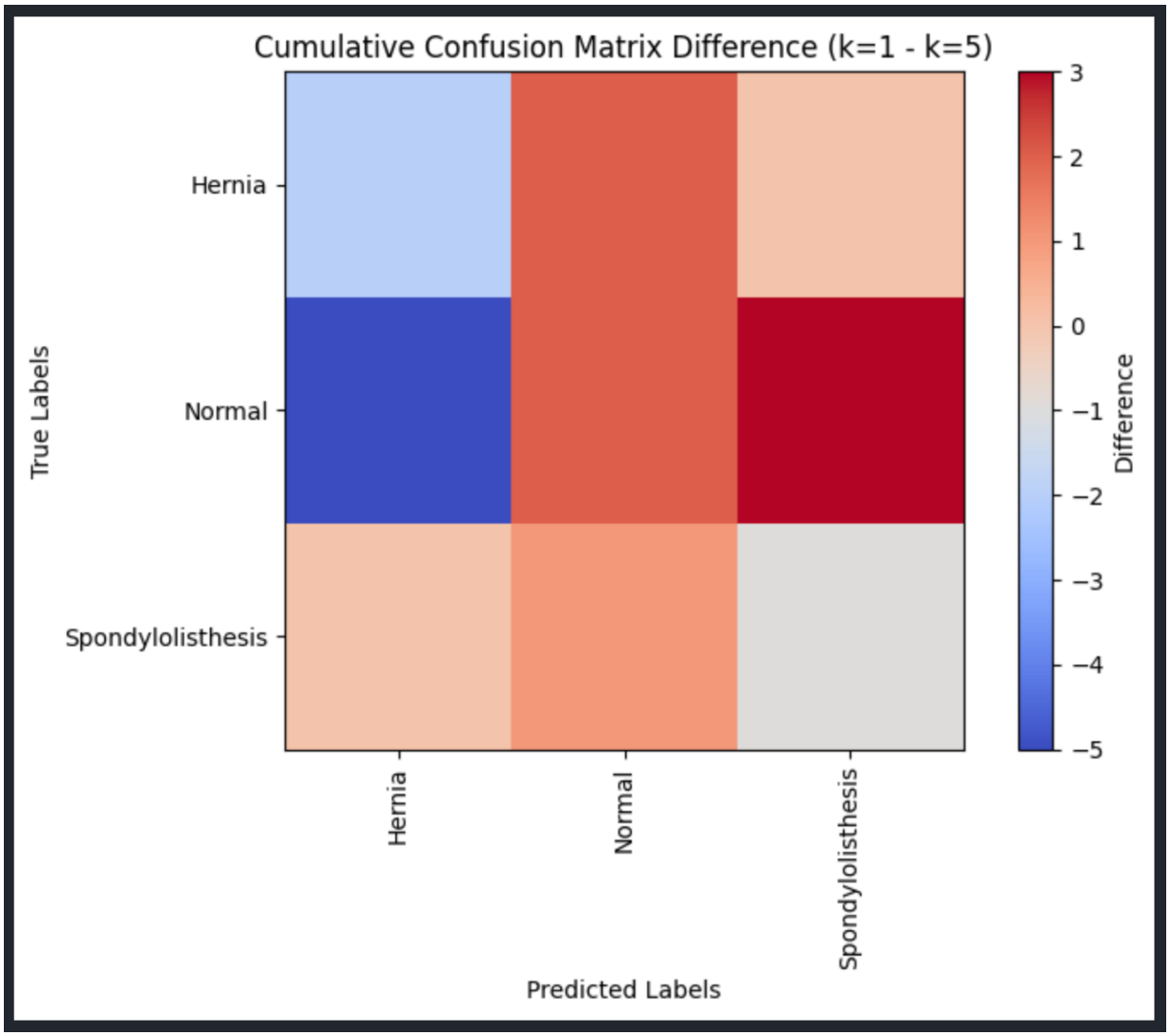
```
diff_cm = cumulative_cm1 - cumulative_cm5

plt.figure(figsize=(8, 6))
plt.imshow(diff_cm, cmap='coolwarm', interpolation='nearest')
plt.colorbar(label='Difference')
plt.title('Cumulative Confusion Matrix Difference (k=1 - k=5)')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')

classes = (df['class'].unique())
classes.sort()

plt.xticks(np.arange(len(classes)), classes, rotation=90)
plt.yticks(np.arange(len(classes)), classes)
plt.tight_layout()
plt.show()
```

✓ 0.0s

Aprendizagem 2022/23

Homework II – Group 001

(ist1102082, ist1103252)



Homework II – Group 001

(ist1102082, ist1103252)

For reference, these are the confusion matrices for K=1, K=5 and the difference respectively.

```
print("Cumulative Confusion Matrix (k=1):\n", cumulative_cm1)
print("\nCumulative Confusion Matrix (k=5):\n", cumulative_cm5)
print("\nCumulative Confusion Matrix Difference (k=1 - k=5):\n", diff_cm)
```

✓ 0.0s

```
Cumulative Confusion Matrix (k=1):
 [[ 37.  23.   0.]
 [ 14.  80.   6.]
 [  1.   7. 142.]]

Cumulative Confusion Matrix (k=5):
 [[ 39.  21.   0.]
 [ 19.  78.   3.]
 [  1.   6. 143.]]

Cumulative Confusion Matrix Difference (k=1 - k=5):
 [[-2.  2.  0.]
 [-5.  2.  3.]
 [ 0.  1. -1.]]
```

The model with K=1 is more sensitive to the training data, since it only uses the nearest neighbor to classify the data. Thus, it's also more prone to overfitting, which is why it is less accurate when it comes to the test data.

The model with K=5 is more robust to the training data, since it uses the 5 nearest neighbors to classify the data. Thus, it's also less prone to overfitting, which is why it is more accurate when it comes to the test data. We can also notice that this model is more accurate when it comes to diagnose patients with Spondylolisthesis class than the KNN with K=1.

Taking in consideration our dataset, in a medical context, the worst case scenario is when a patient is diagosed as Normal whilst carrying a disease. By analysing our difference confusion matrix, we can conclude that such scenario is more likely to happen with the KNN with K=1 than with the KNN with K=5. Thus we can conclude, that within this context, the KNN with K=5 is a more promising classifier than the KNN with K=1.

3)

- The ´column_diagnosis´ represents whether a patient has a medical condition (Spondylolisthesis or Hernia classes), or not (Normal class). In this case, the dataset has signs of some class imbalance, with only 60 observations of Hernia compared to 151 of Spondylolisthesis and 100 of Normal). Naïve Bayes might struggle to correctly classify positive cases due to the lack of sufficient training examples.
- The dataset includes various medical measurements. It can happen that some features might be strongly correlated (for example, three different features are pelvic related). Naïve Bayes assumes independence between features, so it might not capture this correlation effectively, potentially leading to suboptimal predictions, especially when both features are relevant to the diagnosis.
- Naïve Bayes is well-suited for categorical data, but it can struggle with continuous or numerical data. Since our dataset contains continuous values, this can cause some difficulties for Naïve Bayes. Other algorithms, like Gaussian Naïve Bayes (used in exercise 1.), can handle continuous data more easily.

END