

# Homework 2

Gonçalo Meneses 103401  
Tomás Aréde 103239

1)

a) Temos os conj. independentes:  $\{y_1, y_2\}$ ,  $\{y_3, y_4\}$  e  $\{y_5\}$ .

Temos um conjunto multivariável, logo, faremos matrizes.

Para cada conjunto de variáveis calcularemos os parâmetros tal como pedido.

Pelo T. de Bayes:

$$\text{posterior} \quad P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \rightarrow \text{Prior}$$

likelihood

Para o conjunto  $\{y_1, y_2\}$ , para o conjunto de teste indicado temos:

$$y = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_i^2 & \text{Cov}(y_i, y_j) \\ \text{Cov}(y_j, y_i) & \sigma_j^2 \end{bmatrix}$$

& Tendo em conta que:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{m}; \quad \sigma^2(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{m-1}$$

data frame é uma amostra  
& não é uma população

$$\text{Cov}(x_1, x_2) = \frac{1}{n-1} \sum_{i=1}^n (x_{1i} - \bar{x}_1) \cdot (x_{2i} - \bar{x}_2)$$

Tendo em conta isto calculou-se:

$$y|C=A = \begin{bmatrix} \widehat{y_1|C=A} \\ \widehat{y_2|C=A} \end{bmatrix} = \begin{bmatrix} 0,24 \\ 0,52 \end{bmatrix}$$

$$y|C=B = \begin{bmatrix} \widehat{y_1|C=B} \\ \widehat{y_2|C=B} \end{bmatrix} = \begin{bmatrix} 0,5925 \\ 0,3275 \end{bmatrix}$$

$$\Sigma|C=A = \begin{bmatrix} \sigma^2(y_1|C=A) & \text{Cov}(y_1|C=A, y_2|C=A) \\ \text{Cov}(y_2|C=A, y_1|C=A) & \sigma^2(y_2|C=A) \end{bmatrix} =$$

$$= \begin{bmatrix} 0,0064 & 0,0096 \\ 0,0096 & 0,0336 \end{bmatrix}$$

$$\Sigma|C=B = \begin{bmatrix} 0,02289 & -0,00976 \\ -0,00976 & 0,031492 \end{bmatrix}$$

(2)

Para  $\{y_3, y_4\} \in \{y_5\}$  não temos estes tipos de parâmetros.

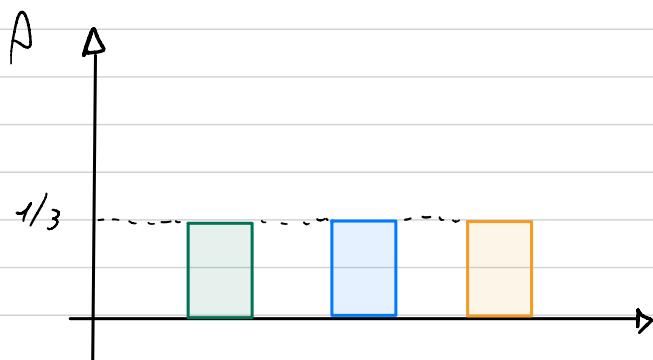
Função massa probabilidade:

$$P(y_3=1 \wedge y_4=1 | C=A) = 1/3 \quad (\text{green})$$

$$P(y_3=0 \wedge y_4=0 | C=A) = 0 \quad (\text{red})$$

$$P(y_3=1 \wedge y_4=0 | C=A) = 1/3 \quad (\text{blue})$$

$$P(y_3=0 \wedge y_4=1 | C=A) = 1/3 \quad (\text{orange})$$

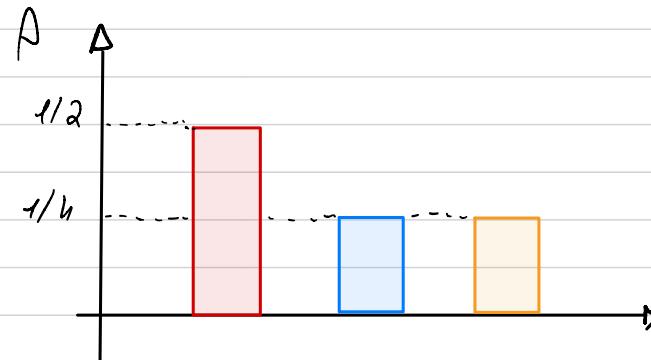


$$P(y_3=1 \wedge y_4=1 | C=B) = 0 \quad (\text{green})$$

$$P(y_3=0 \wedge y_4=0 | C=B) = 1/2 \quad (\text{red})$$

$$P(y_3=1 \wedge y_4=0 | C=B) = 1/4 \quad (\text{blue})$$

$$P(y_3=0 \wedge y_4=1 | C=B) = 1/4 \quad (\text{orange})$$

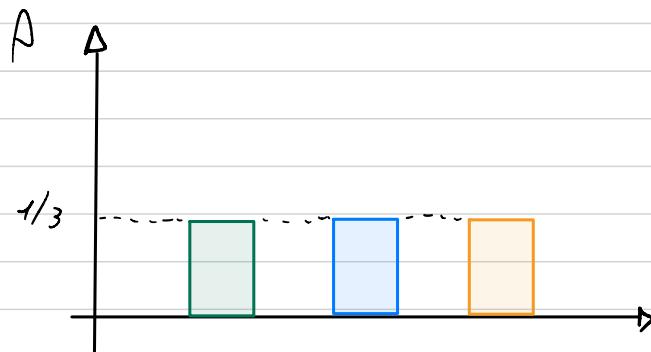


Da mesma forma  $\{y_5\}$  também é representado por uma FMP.

$$P(y_5=0 | C=A) = 1/3 \quad (\text{green})$$

$$P(y_5=1 | C=A) = 1/3 \quad (\text{blue})$$

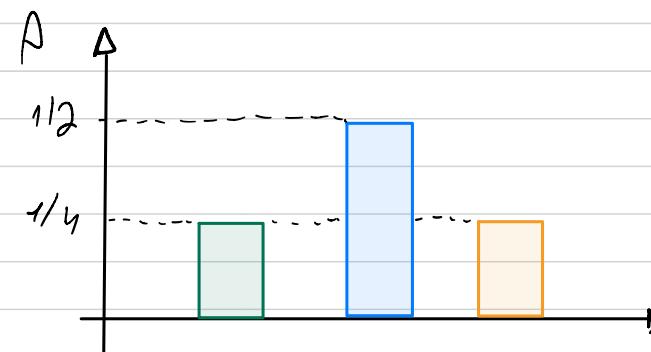
$$P(y_5=2 | C=A) = 1/3 \quad (\text{orange})$$



$$P(y_5=0 | C=B) = 1/4 \quad (\text{green})$$

$$P(y_5=1 | C=B) = 2/4 = 1/2 \quad (\text{blue})$$

$$P(y_5=2 | C=B) = 1/4 \quad (\text{orange})$$



Sabe-se por hipótese que  $y_1, y_2 \in \mathbb{R}^2$  tem uma distribuição normal. Portanto:

$$\{y_1, y_2\}_{C=A} \sim N(y|C=A, \Sigma|C=A)$$

Uma normal com parâmetros  $y$  e  $\Sigma$  é representada por:

$$f(x) = \frac{1}{(2\pi)^{n/2}} \cdot \frac{1}{\sqrt{\det(\Sigma)}} \cdot e^{-\frac{1}{2} \cdot (x-y)^T \Sigma^{-1} (x-y)}$$

conseguimos ver que temos ainda que calcular:

$$\det(\Sigma|C=A) = 0,0064 \cdot 0,0336 - 0,0096^2 = \\ = 1,2288 \cdot 10^{-4}$$

$$\det(\Sigma|C=B) = 6,2559 \cdot 10^{-4}$$

Para uma matriz  $2 \times 2$ , a matriz inversa calcula-se fazendo:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow A^{-1} = \frac{1}{\det(A)} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Calculando as inversas:

$$(\Sigma | C=A)^{-1} = \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,08(3) \end{bmatrix}$$

$$(\Sigma | C=B)^{-1} = \begin{bmatrix} 50,339 & 15,6012 \\ 15,6012 & 36,5892 \end{bmatrix}$$

Substituindo agora:

$$P(\{y_1, y_2\} | C=A) = \frac{1}{(2\pi)^{2/2}} \cdot \frac{1}{1.2288 \cdot 10^{-6}}$$

$$\circ \exp\left(-\frac{1}{2} \cdot (x_{\text{teste}} - y|C=A)^T \cdot (\Sigma | C=A)^{-1} \cdot (x_{\text{teste}} - y|C=A)\right)$$

$$P(\{y_1, y_2\} | C=B) = \frac{1}{(2\pi)^{2/2}} \cdot \frac{1}{1.7565 \cdot 10^{-3}} \cdot \exp\left(-\frac{1}{2} \cdot (x_{\text{teste}} - y|C=B)^T \cdot (\Sigma | C=B)^{-1} \cdot (x_{\text{teste}} - y|C=B)\right)$$

$$\circ (\Sigma | C=B)^{-1} \cdot (x_{\text{teste}} - y|C=B)$$

(5)

Para criar o modelo faltam-nos ainda as probabilidades a Priori:

- Priori:  $P(y_6 = A) = 3/7$

$$P(y_6 = B) = 4/7$$

Agora já temos tudo para criar um modelo!

b) MAP:  $y = \arg\max_{C_i} \{ P(C_i | x) \} = \arg\max_{C_i} \left\{ \frac{P(C_i) \cdot P(x|C_i)}{P(x)} \right\} =$   
 $= \arg\max_{C_i} \{ P(C_i) \cdot P(x|C_i) \}$

com x e comum  
pode-se ignorar

Tendo em conta as seguintes observações de testes

	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
$x_8$	0.38	0.52	0	1	0	A
$x_9$	0.42	0.59	0	1	1	B

Começando então para  $x_8$ :

$$P(x_8 | C = A) = P(y_1 = 0.38, y_2 = 0.52, y_3 = 0, y_4 = 1, y_5 = 0 | C = A)$$

↓ devido à independência entre conjuntos e à dependência dentro dos conjuntos

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

$$P(y_3 = 0, y_4 = 1 | C = A) \cdot P(y_5 = 0 | C = A)$$

$$P(y_6 = A)$$

↓

$$\phi = \frac{1}{(2\pi)^{2/2}} \cdot \frac{1}{1.2288 \cdot 10^{-4}} \cdot$$

$$0.2 \exp\left(-\frac{1}{2} \cdot (x_{\text{teste}} - y|C=A)^T \cdot (\Sigma|C=A)^{-1} \cdot (x_{\text{teste}} - y|C=A)\right)$$

$$= \frac{1}{2\pi \cdot \sqrt{1.2288 \cdot 10^{-4}}} \cdot \exp\left(-\frac{1}{2} \cdot [0,38 - 0,24 \quad 0,52 - 0,52]\right).$$

$$\cdot \begin{bmatrix} 273,4375 & -78,125 \\ -78,125 & 52,08(3) \end{bmatrix} \cdot \begin{bmatrix} 0,38 - 0,24 \\ 0,52 - 0,52 \end{bmatrix}$$

$$\approx 0,984705$$

Ficamos enfoc com:

$$p(x_8|C=A) \approx 0,984705 \cdot \frac{1}{3} \cdot \frac{1}{3} \approx 0,109412$$

(ignorando  
 $x_8$ ) me  
? não mincida!

$$p(C=A|x_8) = p(C=A) \cdot p(x_8|C=A) \approx 4,6891\%$$

Fazendo agora para  $P(C=B|x_8)$  e omitindo alguns passos por analogia:

$$P(x_8|C=B) = P(y_1=0,38; y_2=0,52|C=B) \cdot P(y_3=0, y_4=1|C=B)$$

$$\cdot P(y_5=0|C=B) \approx 0,12266$$

1/4

7

$$P(C=B|x_8) = P(C=B) \cdot P(x_8|C=B) \approx 7,0089\%$$

4/7

→ Portanto, como  $P(C=A|x_8) < P(C=B|x_8)$  então  
classificamos  $x_8$  como B.

Fazendo agora para  $x_9$ :  $\approx 0,40307$

$$P(x_9|C=A) = P(y_1=0,42, y_2=0,59|C=A).$$

$$P(y_3=0, y_4=1|C=A) \quad P(y_5=1|C=A) \approx 0,044786$$

1/3 1/3 3/7

$$P(C=A|x_9) = P(x_9|C=A) \cdot P(C=A) \approx 1,91939\%$$

$\approx 1,72875$

1/4

$$P(x_9|C=B) = P(y_1=0,42, y_2=0,59|C=B) \cdot P(y_3=0, y_4=1|C=B).$$

1/2 1/2

•  $P(y_5=1|C=B) \approx 11,2985$

$$P(C=B|x_g) = P(x_g|C=B) \cdot P(C=B) \approx 12,3482\%$$

4/7

→ Portanto, classificamos  $x_g$  como B pois  $P(C=A|x_g) < P(C=B|x_g)$

C)  $\theta = 0,5$  ML  $\rightarrow \operatorname{Argmax}_{c_i} P(\vec{x}|e_i)$

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

Como estamos a assumir ML podemos tomar:

$$P(A|\vec{x}) \approx P(\vec{x}|A)$$

Renormalizando os likelihoods:

$$\tilde{P}(x_g|C=A) = \frac{P(x_g|C=A)}{P(x_g|C=A) + P(x_g|C=B)} = 47,1465\%$$

$$\tilde{P}(x_g|C=B) = \frac{P(x_g|C=B)}{P(x_g|C=A) + P(x_g|C=B)} = 17,1672\%$$

Logo, os valores de  $\theta$  para os quais a "testing accuracy" é ótima são

$$\theta \in [17,1672\%; 47,1465\%]$$

(2)

a) Data fold (considerando  $z = y_1$ )

D	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$z$
$x_1$	0	1	1	0	A	0,24
$x_2$	0	1	0	1	A	0,16
$x_3$	1	0	1	2	A	0,32

D	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$z$
$x_4$	0	0	0	1	B	0,54
$x_5$	0	0	0	0	B	0,66
$x_6$	0	1	0	2	B	0,76

D	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$z$
$x_7$	1	0	1	1	B	0,41
$x_8$	1	0	1	0	A	0,38
$x_9$	1	0	1	1	B	0,42

Note: Tal como pedido no enunciado binarizamos  $y_2$  como:

Valores  $\in [0; 0,5[ \rightarrow 0$

Valores  $\in ]0,5; 1] \rightarrow 1$

b)

$$d(x_7, x_1) = 4 \quad d(x_7, x_2) = 4 \quad d(x_7, x_3) = 2$$

$$d(x_7, x_4) = 2 \quad d(x_7, x_5) = 3 \quad d(x_7, x_6) = 4$$

Logo,  $KNN_{K=3}(x_7) = \{x_3; x_4; x_5\}$

$$w_i = \frac{1}{d(x_7, x_i)}$$

↓

$$w_3 = \frac{1}{2}; w_4 = \frac{1}{2}; w_5 = \frac{1}{3}$$

$$d(x_8, x_1) = 2 \quad d(x_8, x_2) = 4 \quad d(x_8, x_3) = 1$$

$$d(x_8, x_4) = 4 \quad d(x_8, x_5) = 3 \quad d(x_8, x_6) = 5$$

(10)

Logo,  $KNN_{K=3}(x_8) = \{x_1, x_3, x_5\}$ ;  $w_1 = \frac{1}{2}; w_3 = 1; w_5 = \frac{1}{3}$

Analogamente,  $KNN_{K=3}(x_9) = \{x_3, x_4, x_5\}$ ;  $w_3 = \frac{1}{2}; w_4 = \frac{1}{2}; w_5 = \frac{1}{3}$

$$d(x_9, x_3) = d(x_9, x_4) = 2$$

$$d(x_9, x_5) = 3$$

$$\hat{z}_7 = \frac{\frac{1}{2} \cdot 0,32 + \frac{1}{2} \cdot 0,54 + \frac{1}{3} \cdot 0,60}{\frac{1}{2} + \frac{1}{2} + \frac{1}{3}} = 0,4875$$

$$\hat{z}_8 = \frac{\frac{1}{2} \cdot 0,24 + 0,32 + \frac{1}{3} \cdot 0,66}{\frac{1}{2} + 1 + \frac{1}{3}} = 0,36$$

$$\hat{z}_9 = 0,4875$$

$$MAE(\hat{z}, z) = \frac{1}{3} \sum_{i=7}^9 |z_i - \hat{z}_i| = 0,055$$

# G11\_notebook

October 9, 2023

## Homework 2

Gonçalo Meneses, 103401 e Tomás Arêde, 103239

Começamos por carregar o nosso ficheiro arff e criar o data frame que irá armazenar os nossos dados.

```
[ ]: from scipy.io import arff
import pandas as pd
import numpy as np

data, col_names = arff.loadarff('column_diagnosis.arff')

df = pd.DataFrame(data)

df['class'] = df['class'].str.decode('utf-8')

df.columns = col_names.names()
df.head()
```

```
[ ]:   pelvic_incidence  pelvic_tilt  lumbar_lordosis_angle  sacral_slope \
0          63.027817    22.552586            39.609117      40.475232
1          39.056951    10.060991            25.015378      28.995960
2          68.832021    22.218482            50.092194      46.613539
3          69.297008    24.652878            44.311238      44.644130
4          49.712859     9.652075            28.317406      40.060784

  pelvic_radius  degree_spondylolisthesis  class
0        98.672917                 -0.254400  Hernia
1       114.405425                  4.564259  Hernia
2       105.985135                 -3.530317  Hernia
3       101.868495                 11.211523  Hernia
4       108.168725                  7.918501  Hernia
```

Exercício 1 (a)

```
[ ]: from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
import matplotlib.pyplot as plt
```

```

import seaborn as sns

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

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

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

knn_score = cross_val_score(knn_classifier, X, y, cv=strat_cv)

naiveb_score = cross_val_score(naiveb_classifier, X, y, cv=strat_cv)

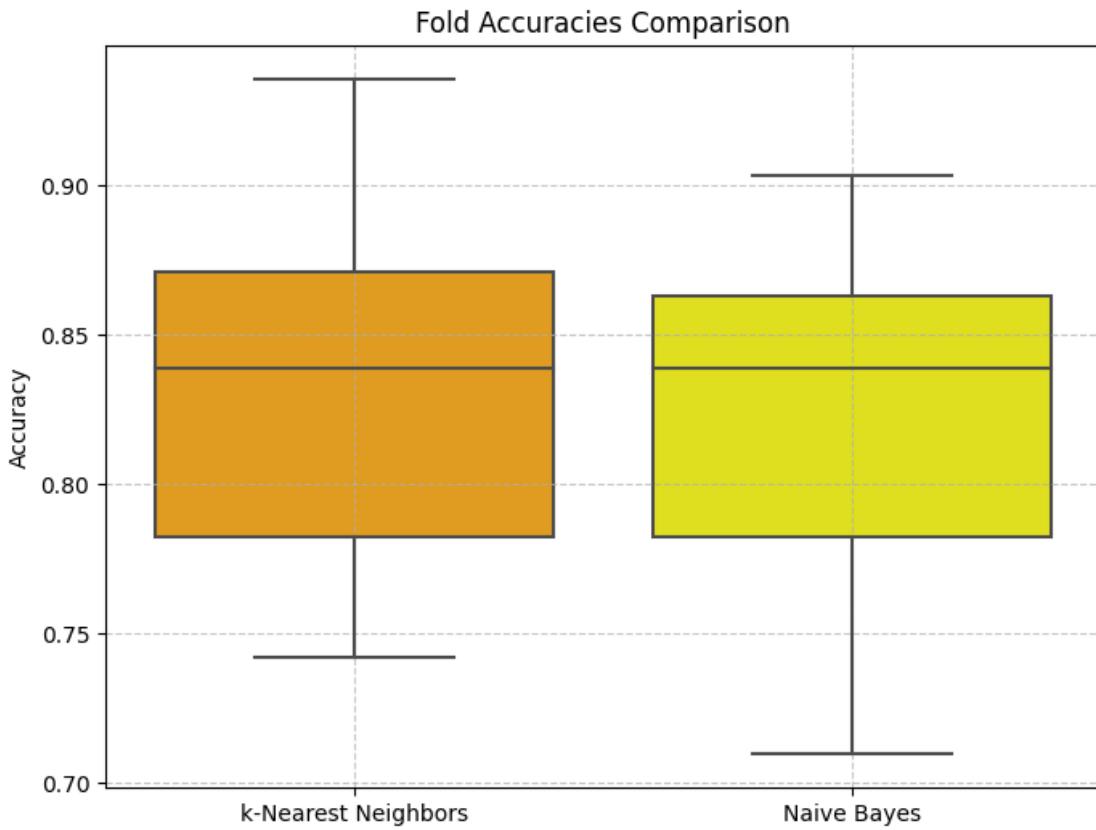
print("kNN Mean Accuracy:", np.mean(knn_score))
print("Naive Bayes Mean Accuracy:", np.mean(naiveb_score))

plt.figure(figsize=(8, 6))
sns.boxplot(data=[knn_score, naiveb_score], palette=['orange', 'yellow'])
plt.xticks([0, 1], ['k-Nearest Neighbors', 'Naive Bayes'])
plt.ylabel('Accuracy')
plt.title('Fold Accuracies Comparison')
plt.grid(True, linestyle='--', alpha=0.7)

plt.show()

```

kNN Mean Accuracy: 0.8387096774193548  
Naive Bayes Mean Accuracy: 0.8225806451612904



### Exercício 1 (b)

```
[ ]: from scipy import stats

t_stat, p_value = stats.ttest_rel(knn_score, naiveb_score, u
    ↪alternative="greater")

# Standard levels of significance
standard_alpha = [0.01, 0.05, 0.1]

print("Our p-value is:", p_value)

if any(p_value < alpha for alpha in standard_alpha):
    print("kNN is statistically superior to Naive Bayes regarding accuracy, for u
        ↪at least one of the standard levels of significance.")
else:
    print("There is no significant difference in accuracy between kNN and Naive u
        ↪Bayes, for the standard levels of significance.")
```

Our p-value is: 0.19042809062064092

There is no significant difference in accuracy between kNN and Naive Bayes, for

the standard levels of significance.

## Exercício 2

```
[ ]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

knn_1 = KNeighborsClassifier(n_neighbors=1, metric='euclidean', ↴
    ↪weights='uniform')
knn_5 = KNeighborsClassifier(n_neighbors=5, metric='euclidean', ↴
    ↪weights='uniform')

num_classes = len(np.unique(y))
class_names = df['class'].unique()

cumulative_conf_matrix_1 = np.zeros((num_classes, num_classes))
cumulative_conf_matrix_5 = np.zeros((num_classes, num_classes))

for train_ind, test_ind in strat_cv.split(X, y):
    X_train, X_test = X.iloc[train_ind], X.iloc[test_ind]
    y_train, y_test = y.iloc[train_ind], y.iloc[test_ind]

    knn_1.fit(X_train, y_train)
    knn_5.fit(X_train, y_train)

    y_pred_1 = knn_1.predict(X_test)
    y_pred_5 = knn_5.predict(X_test)

    conf_matrix_1 = confusion_matrix(y_test, y_pred_1)
    conf_matrix_5 = confusion_matrix(y_test, y_pred_5)

    cumulative_conf_matrix_1 += conf_matrix_1
    cumulative_conf_matrix_5 += conf_matrix_5

conf_matrix_diff = cumulative_conf_matrix_5 - cumulative_conf_matrix_1

disp_1 = ConfusionMatrixDisplay(confusion_matrix=cumulative_conf_matrix_1.
    ↪astype(int), display_labels=class_names)
disp_5 = ConfusionMatrixDisplay(confusion_matrix=cumulative_conf_matrix_5.
    ↪astype(int), display_labels=class_names)
disp_diff = ConfusionMatrixDisplay(confusion_matrix=conf_matrix_diff.
    ↪astype(int), display_labels=class_names)

fig, axes = plt.subplots(1, 3, figsize=(15,5))

disp_1.plot(ax=axes[0], cmap='coolwarm', xticks_rotation='vertical', ↴
    ↪values_format='d')
axes[0].set_title("Cumulative Confusion Matrix (k=1)")
```

```

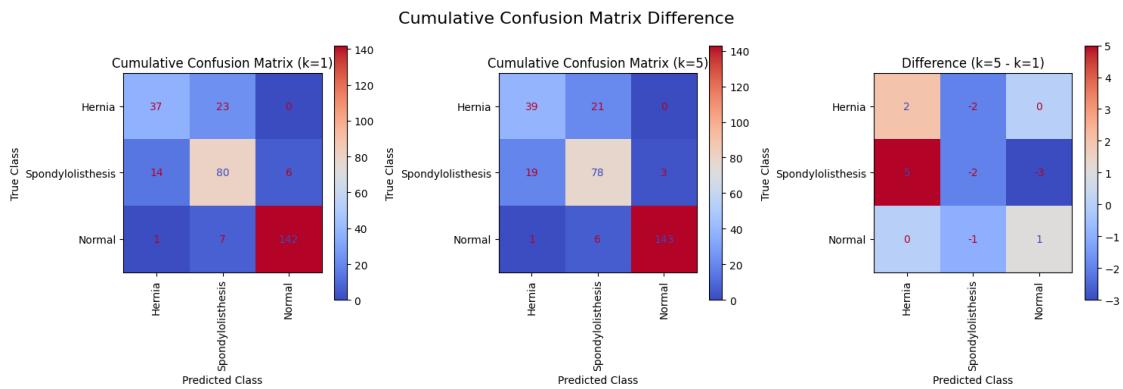
axes[0].set_xlabel("Predicted Class")
axes[0].set_ylabel("True Class")

disp_5.plot(ax=axes[1], cmap='coolwarm', xticks_rotation='vertical', □
    ↪values_format='d')
axes[1].set_title("Cumulative Confusion Matrix (k=5)")
axes[1].set_xlabel("Predicted Class")
axes[1].set_ylabel("True Class")

disp_diff.plot(ax=axes[2], cmap='coolwarm', xticks_rotation='vertical', □
    ↪values_format='d')
axes[2].set_title("Difference (k=5 - k=1)")
axes[2].set_xlabel("Predicted Class")
axes[2].set_ylabel("True Class")

plt.suptitle("Cumulative Confusion Matrix Difference", fontsize=16)
plt.tight_layout()
plt.show()

```



Como se pode ver no gráfico acima, as diferenças de desempenho entre os nossos dois predictors ( $k=1$  e  $k=5$ ) são relativamente pequenas, especialmente quando se considera a vasta gama de contagens (de 1 a 143). Especificamente, as diferenças registadas variam entre -5 e 3, o que representa apenas uma fração da variação global no conjunto de dados. Mesmo assim, é importante notar que o número de true positives aumenta para as classes Hernia (+2) e Normal (+1), diminuindo para a classe Spondylolisthesis (-2). Para além disso, é ainda observável uma diminuição do número de pessoas mal classificadas como Normal (-3) e Spondylolisthesis (-3), o que é compensado pelo aumento de pessoas mal classificadas como Hernia (+5). Por fim, podemos concluir que ambos os predictors apresentam uma precisão semelhante na previsão da classe do nosso conjunto de dados, devido à baixa dispersão dos mesmos.

### Exercício 3

Algumas dificuldades do Naive Bayes que podem surgir quando se aprende com o conjunto de dados column\_diagnosis começam com o pressuposto de independência, ou seja, todas as características

podem não ser condicionalmente independentes, dada a classe. Outro problema importante é o facto de que, ao executar o Naive Bayes, estamos a assumir que as nossas variáveis de entrada seguem uma distribuição Normal perfeita, o que não é o caso, como vimos no homework 1, tal poderá dever-se ao número insuficiente de amostras do nosso dataset. Isto conduzirá a resultados não óptimos com o nosso classificador Naive Bayes. Por fim, os nossos resultados podem estar enviesados devido à presença relativamente grande de indivíduos classificados como Spondylolisthesis em comparação com indivíduos classificados como Hernia e Normal.