Aprendizagem 2022/23

Homework II – Group 009

I. Pen-and-paper

1) Começamos por calcular as distâncias para todas as observações, tendo obtido a seguinte tabela:

	y_1	y_2	z	$d(x_1,x_j)$	$d(x_2,x_j)$	$d(x_3,x_j)$	$d(x_4,x_j)$	$d(x_5,x_j)$	$d(x_6,x_j)$	$d(x_7,x_j)$	$d(x_8,x_j)$
x_1	A	0	P	-	2,5	1,5	0,5	1,5	1,5	1,5	2,5
x_2	B	1	P	2,5	-	1,5	2,5	1,5	1,5	1,5	0,5
x_3	A	1	P	1,5	1,5	-	1,5	2,5	2,5	0,5	1,5
x_4	A	0	P	0,5	2,5	1,5	-	1,5	1,5	1,5	2,5
x_5	B	0	N	1,5	1,5	2,5	1,5	-	0,5	2,5	1,5
x_6	B	0	N	1,5	1,5	2,5	1,5	0,5	-	2,5	1,5
x_7	A	1	N	1,5	1,5	0,5	1,5	2,5	2,5	-	1,5
x_8	B	1	N	2,5	0,5	1,5	2,5	1,5	1,5	1,5	-

De seguida, usando distance-weighted kNN com k=5, obtivemos as seguintes previsões para cada observação:

$$\hat{z}/x_1 = weighted\ mode\left(\frac{1}{1.5}P + \frac{1}{0.5}P, \frac{3}{1.5}N\right) = P$$

$$\hat{z}/x_2 = weighted\ mode\left(\frac{1}{1.5}P, \frac{3}{1.5}N + \frac{1}{0.5}N\right) = N$$

$$\hat{z}/x_3 = weighted\ mode\left(\frac{3}{1.5}P, \frac{1}{0.5}N + \frac{1}{1.5}N\right) = N$$

$$\hat{z}/x_4 = weighted\ mode\left(\frac{1}{1.5}P + \frac{1}{0.5}P, \frac{3}{1.5}N\right) = P$$

$$\hat{z}/x_5 = weighted\ mode\left(\frac{3}{1.5}P, \frac{1}{0.5}N + \frac{1}{1.5}N\right) = N$$

$$\hat{z}/x_6 = weighted\ mode\left(\frac{3}{1.5}P, \frac{1}{0.5}N + \frac{1}{1.5}N\right) = N$$

$$\hat{z}/x_7 = weighted\ mode\left(\frac{3}{1.5}P + \frac{1}{0.5}P, \frac{1}{1.5}N\right) = P$$

$$\hat{z}/x_8 = weighted\ mode\left(\frac{1}{1.5}P + \frac{1}{0.5}P, \frac{3}{1.5}N\right) = P$$

Com base nestas previsões conseguimos obter a seguinte matriz de confusão:

		Real	
		P	N
Previsão	P	2	2
	N	2	2

Por fim, com base nos resultados da matriz de confusão, calculamos o seguinte recall:

$$Recall = \frac{TP}{TP+FN} = \frac{2}{4}$$

Aprendizagem 2022/23

Homework II – Group 009

2) Aprender um classificador Bayesiano, significa conseguir calcular as seguintes probabilidades:

$$P(Positivo/x)=\frac{P(x/Positivo)\,x\,P(Positivo)}{P(x)}$$

$$P(Negativo/x)=\frac{P(x/Negativo)\,x\,P(Negativo)}{P(x)}$$

Como P(Positivo/x)=1 - P(Negativo/x) temos que:

$$P(Positivo/x)=1-P(Negativo/x)\Leftrightarrow\frac{P(x/Positivo)\,x\,P(Positivo)}{P(x)}=1-\frac{P(x/Negativo)\,x\,P(Negativo)}{P(x)}$$

$$P(x)=P(x/Positivo)\,x\,P(Positivo)+P(x/Negativo)\,x\,P(Negativo)$$

Portanto ficamos com:

$$P(Positivo/x)=\frac{P(x/Positivo)\,x\,P(Positivo)}{P(x/Positivo)\,x\,P(Positivo)+P(x/Negativo)\,x\,P(Negativo)}$$

$$P(Negativo/x)=\frac{P(x/Negativo)\,x\,P(Negativo)}{P(x/Positivo)\,x\,P(Positivo)+P(x/Negativo)\,x\,P(Negativo)}$$

Logo para conseguirmos calcular tanto P(Positivo/x) como P(Negativo/x) vamos precisar de calcular P(x/Positivo), P(Positivo), P(x/Negativo) e P(Negativo).

<u>Para P(x/Positivo) temos:</u>

$$P(x/Positivo)=P(y_1=a_1,y_2=a_2,y_3=a_3/Positivo)$$

Mas como assumimos que i) y_1 e y_2 são dependentes e ii) $\{y_1,y_2\}$ e $\{y_3\}$ são conjuntos de variáveis independentes e igualmente importantes, temos que:

$$P(x/Positivo)=P(y_1=a_1,y_2=a_2/Positivo)\,x\,P(y_3=a_3/Positivo)$$

Onde $P(y_1=a_1,y_2=a_2/Positivo)$ é dado por:

$$P(y_1=a_1,y_2=a_2/Positivo)=\begin{cases}\frac{2}{5},\ se\ a_1=A\ e\ a_2=0\\ \frac{1}{5},\ se\ a_1=A\ e\ a_2=1\\ \frac{1}{5},\ se\ a_1=B\ e\ a_2=0\\ \frac{1}{5},\ se\ a_1=B\ e\ a_2=1\end{cases}$$

E devido a iii) y_3 é distribuído normalmente, temos que $P(y_3{=}a_3/Positivo)$ é dado por:

$$P(y_3=a_3/Positivo)=N(a_3/\mu=0.84,\sigma=0.251)=\frac{1}{0.251\,x\sqrt{2\pi}}e^{-\frac{1}{2x0.251^2}x(a_3-0.84)^2}$$

Para P(Positivo) temos:

$$P(Positivo)=\frac{observações\ positivas}{total\ de\ observações}=\frac{5}{9}$$

Para P(x/Negativo) temos:

$$P(x/Negativo)=P(y_1=a_1,y_2=a_2,y_3=a_3/Negativo)$$

Mas como assumimos que i) y_1 e y_2 são dependentes e ii) $\{y_1,y_2\}$ e $\{y_3\}$ são conjuntos de variáveis independentes e igualmente importantes, temos que:

$$P(x/Negativo)=P(y_1=a_1,y_2=a_2/Negativo)\,x\,P(y_3=a_3/Negativo)$$

Onde $P(y_1{=}a_1,y_2{=}a_2/Negativo)$ é dado por:

$$P(y_1=a_1,y_2=a_2/Negativo)=\begin{cases}0,\ se\ a_1=A\ e\ a_2=0\\ \frac{1}{4},\ se\ a_1=A\ e\ a_2=1\\ \frac{2}{4},\ se\ a_1=B\ e\ a_2=0\\ \frac{1}{4},\ se\ a_1=B\ e\ a_2=1\end{cases}$$

E devido a iii) y_3 é distribuído normalmente, temos que $P(y_3{=}a_3/Positivo)$ é dado por:

$$P(y_3=a_3/Negativo)=N(a_3/\mu=0.975,\sigma=0.1708)=\frac{1}{0.1708\,x\sqrt{2\pi}}e^{-\frac{1}{2x0.1708^2}x(a_3-0.975)^2}$$

Para P(Negativo) temos:

$$P(Negativo)=\frac{observações\ negativas}{total\ de\ observações}=\frac{4}{9}$$

3) Denominando cada observação de teste da seguinte forma:

$$x_{t1}=\{y_1=A,y_2=1,y_3=0.8\}$$

$$x_{t2}=\{y_1=B,y_2=1,y_3=1\}$$

$$x_{t3}=\{y_1=B,y_2=0,y_3=0.9\}$$

Temos para x_{t1}:

$$P(x_{t1}/Positivo)=\frac{1}{5}x1.5694=0.3139$$

$$P(x_{t1}/Negativo)=\frac{1}{4}x\,1.3819=0.3455$$

$$P(Positivo/x_{t1})=\frac{0.3139\,x\,\frac{5}{9}}{0.3139\,x\,\frac{5}{9}+0.3455\,x\,\frac{4}{5}}=0.5318$$

Temos para x_{t2}:

$$P(x_{t2}/Positivo)=\frac{1}{5}x\,1.2972=0.2594$$

$$P(x_{t2}/Negativo)=\frac{1}{4}x\,2.3111=0.5778$$

$$P(Positivo/x_{t2})=\frac{0.2594\,x\,\frac{5}{9}}{0.2594\,x\,\frac{5}{9}+0.5778\,x\,\frac{4}{5}}=0.3595$$

Temos para x_{t3}:

$$P(x_{t3}/Positivo)=\frac{1}{5}x\,1.5447=0.3089$$

$$P(x_{t3}/Negativo)=\frac{2}{4}x\,2.1212=1.0606$$

$$P(Positivo/x_{t3})=\frac{0.3089\,x\,\frac{5}{9}}{0.3089\,x\,\frac{5}{9}+1.0606\,x\,\frac{4}{5}}=0.2669$$

4) Considerando θ=0.3 temos:

$$z_{t1}=Positivo\,,\ P(Positivo/x_{t1})=0.5318>\theta=0.3\Rightarrow\hat{z}_{t1}=Positivo$$

$$z_{t2}=Positivo\,,\ P(Positivo/x_{t2})=0.3595>\theta=0.3\Rightarrow\hat{z}_{t2}=Positivo$$

$$z_{t3}=Negativo\,,\ P(Positivo/x_{t3})=0.2669<\theta=0.3\Rightarrow\hat{z}_{t3}=Negativo$$

$$testing\ accuracy_{\theta=0.3}=\frac{previsões\ corretas}{total\ de\ previsões}=\frac{3}{3}$$

<u>Considerando θ=0.5 temos:</u>

$$z_{t1} = Positivo\,,\ P(Positivo/x_{t1}) = 0.5318 > \theta = 0.5 \Rightarrow \hat{z}_{t1} = Positivo$$

$$z_{t2} = Positivo\,,\ P(Positivo/x_{t2}) = 0.3595 < \theta = 0.5 \Rightarrow \hat{z}_{t2} = Negativo$$

$$z_{t3} = Negativo\,,\ P(Positivo/x_{t3}) = 0.2669 < \theta = 0.5 \Rightarrow \hat{z}_{t3} = Negativo$$

$$testing\ accuracy_{\theta=0.5} = \frac{previsões\ corretas}{total\ de\ previsões} = \frac{2}{3}$$

<u>Considerando θ=0.7 temos:</u>

$$z_{t1} = Positivo\,,\ P(Positivo/x_{t1}) = 0.5318 < \theta = 0.7 \Rightarrow \hat{z}_{t1} = Negativo$$

$$z_{t2} = Positivo\,,\ P(Positivo/x_{t2}) = 0.3595 < \theta = 0.7 \Rightarrow \hat{z}_{t2} = Negativo$$

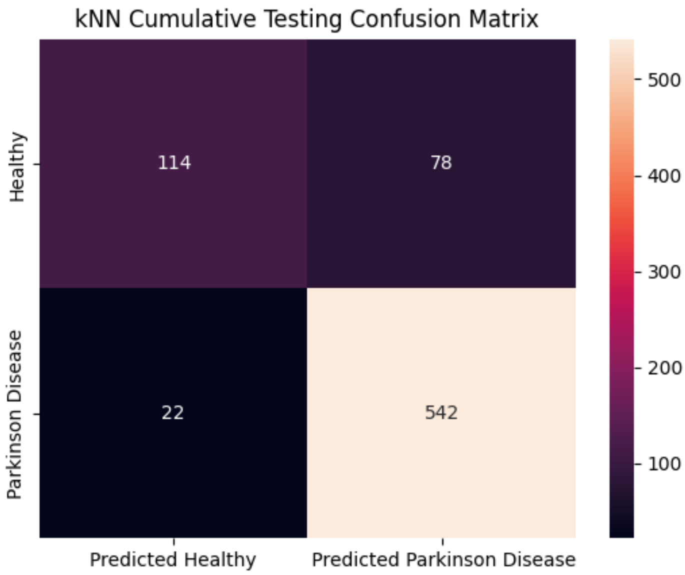
$$z_{t3} = Negativo\,,\ P(Positivo/x_{t3}) = 0.2669 < \theta = 0.7 \Rightarrow \hat{z}_{t3} = Negativo$$

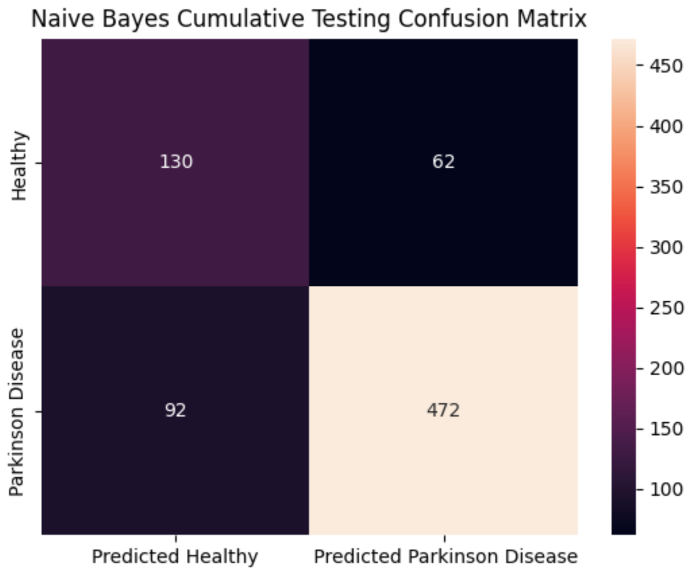
$$testing\ accuracy_{\theta=0.7} = \frac{previsões\ corretas}{total\ de\ previsões} = \frac{1}{3}$$

A maior testing accuracy acontece quando temos θ=0.3, portanto podemos concluir, com base neste conjunto de teste, que vai ser esse o valor do decision threshold que maximiza a testing accuracy.

II. Programming and critical analysis

5) Importa referir que antes de aplicar tanto o kNN como o Naive Bayes, procedemos à normalização dos dados, pois verificamos que as características do conjunto de dados de entrada diferem muito entre os seus intervalos e são medidos em unidades de medida diferentes.





6) Os resultados que obtivemos para a accuracy, aplicando 10-fold stratified cross validation foram os seguintes:

Para o kNN:

$$testing\ accuracy_{kNN} = 0.87 \pm 0.04$$

Para o Naive Bayes:

$$testing\ accuracy_{Naive\ Bayes} = 0.8 \pm 0.05$$

As hipóteses que permitem testar, neste caso, se em termos de accuracy o kNN é estatisticamente superior ao Naive Bayes são as seguintes:

$$\begin{cases} H_0: testing\ accuracy_{kNN} \leq testing\ accuracy_{Naive\ Bayes} \\ H_1: testing\ accuracy_{kNN} > testing\ accuracy_{Naive\ Bayes} \end{cases}$$

Através do teste-t obtivemos o seguinte p-value:

$$p-value = 0.00247 < \alpha = 0.05$$

Portanto rejeitamos H_0, logo existe evidência estatística ao nível de 5% de que o kNN é superior ao Naive Bayes, em termos de accuracy.

7) Duas razões possíveis para as diferenças na predictive accuracy entre o kNN e o Naive Bayes são:

- O Naive Bayes assume que as características do conjunto de dados são independentes, portanto se existir dependências entre as características, a accuracy no Naive Bayes pode ser afetada negativamente;
- O Naive Bayes só consegue criar fronteiras de decisão lineares, elípticas ou parabólicas, enquanto que o kNN não sofre desta limitação. Portanto, se a fronteira de decisão do conjunto de dados não é linear, elíptica ou parabólica, a accuracy no Naive Bayes pode ser afetada negativamente.

III. APPENDIX

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.io.arff import loadarff
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler

data = loadarff('pd_speech.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
X = df.drop('class',axis=1)
y = df['class']
scaler = StandardScaler().fit(X)

knn_predictor = KNeighborsClassifier()
knn_cm_sum = np.array([[0,0],[0,0]])
knn_acc_scores = []

nb_predictor = GaussianNB()
nb_cm_sum = np.array([[0,0],[0,0]])
nb_acc_scores = []

folds = StratifiedKFold(n_splits=10, random_state=0, shuffle=True)
for train_k, test_k in folds.split(X,y):
	X_train, X_test = X.iloc[train_k], X.iloc[test_k]
	X_train, X_test = scaler.transform(X_train), scaler.transform(X_test)
	y_train, y_test = y.iloc[train_k], y.iloc[test_k]

	knn_predictor.fit(X_train,y_train)
	knn_y_pred = knn_predictor.predict(X_test)
	knn_cm = np.array(confusion_matrix(y_test,knn_y_pred,labels=['0','1']))
	knn_cm_sum = np.add(knn_cm_sum,knn_cm)
	knn_acc_scores.append(accuracy_score(y_test,knn_y_pred))

	nb_predictor.fit(X_train, y_train)
	nb_y_pred = nb_predictor.predict(X_test)
	nb_cm = np.array(confusion_matrix(y_test,nb_y_pred))
	nb_cm_sum = np.add(nb_cm_sum,nb_cm)
	nb_acc_scores.append(accuracy_score(y_test,nb_y_pred))

knn_cm_sum_dt = pd.DataFrame(knn_cm_sum, index=['Healthy', 'Parkinson Disease'], columns=['Predicted 
Healthy', ' Predicted Parkinson Disease'])
sns.heatmap(knn_cm_sum_dt,annot=True,fmt='g')
plt.title('kNN Cumulative Testing Confusion Matrix')
plt.show()
nb_cm_sum_dt = pd.DataFrame(nb_cm_sum, index=['Healthy', 'Parkinson Disease'], columns=['Predicted 
Healthy', ' Predicted Parkinson Disease'])
sns.heatmap(nb_cm_sum_dt,annot=True,fmt='g')
plt.title('Naive Bayes Cumulative Testing Confusion Matrix')
plt.show()
print('knn accuracy',round(np.mean(knn_acc_scores),2),'±',round(np.std(knn_acc_scores),2))
print('nb accuracy',round(np.mean(nb_acc_scores),2),'±',round(np.std(nb_acc_scores),2))
res = stats.ttest_rel(knn_acc_scores,nb_acc_scores,alternative='greater')
print(res.pvalue)
```

END