LEIC-T 2023/2024
Aprendizagem - Machine Learning

Homework II - Group 007

Miguel Teixeira - 103449

Rodrigo Alves - 103299

a)

x_1	x_2	class
0,5	0,5	A
1	1,5	A
1,5	0,8	A
2	1,8	A
2	0	B
2	1	B
3	0	B
5	1,2	B

$P[C=A) = \frac{1}{2}$

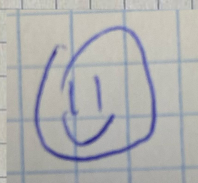
$P[C=B) = \frac{1}{2}$

$P(A) + P(B) = 1$

$P(X_1, X_2) = P(X_1, X_2, A) + P(X_1, X_2, B)$

	$P[x_1 \mid C=A)$	$p(x_1 \mid C=B)$
μ	1,25	3
σ	0,646	1,411

$P(A \mid X_1, X_2) + P(B \mid X_1, X_2) = 1$



	$P[x_2 \mid C=A)$	$p(x_2 \mid C=B)$
μ	1,15	0,55
σ	0,603	0,64

$$\frac{P(C=A \mid x_1=1, x_2=2)}{P(x_1=1, x_2=2)} = \frac{P(C=A)\, P(1 \mid \mu=1{,}25, \sigma=0{,}646)\, P(2 \mid \mu=1{,}15, \sigma=0{,}603)}{P(x_1=1, x_2=2)}$$

$$P(1 \mid \mu=1{,}25, \sigma=0{,}646) = \frac{1}{\sqrt{2\pi} \times 0{,}646} \times \exp\left(-\frac{1}{2 \cdot 0{,}646^2} \times (1-1{,}25)^2\right) =$$

$$= 0{,}573$$

$$P(2 \mid \mu=1{,}15, \sigma=0{,}603) = \frac{1}{\sqrt{2\pi} \times 0{,}603} \times \exp\left(-\frac{1}{2 \times 0{,}603^2} \times (2-1{,}15)^2\right) =$$

$$= 0{,}245$$

$$P(1 \mid \mu=3, \sigma=1{,}41) = \frac{1}{\sqrt{2\pi} \times 1{,}41} \times \exp\left(-\frac{1}{2 \times 1{,}41^2} \times (1-3)^2\right) =$$

$$= 0{,}103$$

$$P(2 \mid \mu=0{,}55, \sigma=0{,}64) = \frac{1}{\sqrt{2\pi} \times 0{,}64} \times \exp\left(-\frac{1}{2 \times 0{,}64} \times (2-0{,}55)^2\right) =$$

$$= 0{,}048$$

$$P(C=A \mid x_1=1, x_2=2) = \frac{\frac{1}{2} \times 0{,}140385}{\frac{1}{2} \times 0{,}140385 + \frac{1}{2} \times 0{,}004944} = 0{,}966$$

$$P(C=B \mid x_1=1, x_2=2) = \frac{\frac{1}{2} \times 0{,}004944}{\frac{1}{2} \times 0{,}140385 + \frac{1}{2} \times 0{,}004944} = 0{,}034$$

Comparing the numerators, we find that A is the most probable class for the query vector

b) $P(C=A) = \frac{1}{2}$

$P(C=B) = \frac{1}{2}$

$P(x_1,x_2|C=A) \sim \begin{bmatrix} 1,25 \\ 1,15 \end{bmatrix}$ $\quad P(x_1,x_2|C=B) \begin{bmatrix} 3 \\ 0,55 \end{bmatrix}$

Para $C=A$

$$C_{11} = \frac{(0,5-1,25)^2 + (1-1,25)^2 + (1,5-1,25)^2 + (2-1,25)^2}{4-1} = \frac{5}{12} = 0,4167$$

$$C_{22} = \frac{(0,5-1,15)^2 + (1,5-1,15)^2 + (0,8-1,15)^2 + (1,8-1,15)^2}{4-1} = 0,363$$

$$C_{21} = C_{12} = \frac{((0,5-1,25)\cdot(0,5-1,15)) + ((1-1,25)\cdot(1,5-1,15)) + ((1,5-1,25)\cdot(0,8-1,15)) + ((2-1,25)\cdot(1,8-1,15))}{3}$$

$$= \frac{\frac{39}{80} - \frac{7}{80} - \frac{7}{80} + \frac{39}{80}}{3} = \frac{4}{15} = 0,267$$

$$\begin{bmatrix} 0,4167 & 0,267 \\ 0,267 & 0,363 \end{bmatrix}$$

Para $C=B$

$$C_{11} = \frac{(2-3)^2 + (2-3)^2 + (3-3)^2 + (5-3)^2}{4-1} = 2$$

$$C_{22} = \frac{(0-0,55)^2 + (1-0,55)^2 + (0-0,55)^2 + (1,2-0,55)^2}{4-1} = 0,41$$

$$C_{12} = C_{21} = \frac{((2-3)\cdot(0-0,55)) + ((2-3)\cdot(1-0,55)) + ((3-3)\cdot(0-0,55)) + ((5-3)\cdot(1,2-0,55))}{4-1}$$

$$= 0,467$$

$$\begin{bmatrix} 2 & 0,467 \\ 0,467 & 0,41 \end{bmatrix}$$

$$P(C=A \mid x_1=1, x_2=2) = \frac{P(C=A)\, P(x_1=1, x_2=2 \mid C=A)}{P(x_1=1, x_2=2)}$$

$$\Sigma = \begin{bmatrix} 0,4167 & 0,267 \\ 0,267 & 0,363 \end{bmatrix}$$

$$\det = (0,363 + 0,4167) - (0,267)^2 = 0,08$$

$$\Sigma^{-1} = \frac{1}{0{,}08} \times \begin{bmatrix} 0{,}363 & -0{,}267 \\ -0{,}267 & 0{,}4107 \end{bmatrix} = \begin{bmatrix} 4{,}54 & -3{,}34 \\ -3{,}34 & 5{,}21 \end{bmatrix}$$

$$N(\vec{x} \mid \vec{y}, \Sigma) = \frac{1}{(2\pi)^{\frac{2}{2}} \sqrt{0{,}08}} \cdot \exp\left(-\frac{1}{2}\left(\left(\begin{bmatrix}1\\2\end{bmatrix} - \begin{bmatrix}1{,}25\\1{,}15\end{bmatrix}\right)^T \cdot \begin{bmatrix} 4{,}54 & -3{,}34 \\ -3{,}34 & 5{,}21 \end{bmatrix} \cdot \left(\begin{bmatrix}1\\2\end{bmatrix} - \begin{bmatrix}1{,}25\\1{,}15\end{bmatrix}\right)\right)\right)$$

$$= \frac{1}{2\pi \sqrt{0{,}08}} \cdot \exp\left(-\frac{1}{2}\left(\begin{bmatrix}-0{,}25 & 0{,}85\end{bmatrix} \cdot \begin{bmatrix} 4{,}54 & -3{,}34 \\ -3{,}34 & 5{,}21 \end{bmatrix} \cdot \begin{bmatrix}-0{,}25\\0{,}85\end{bmatrix}\right)\right) =$$

$$= \frac{1}{2\pi \sqrt{0{,}08}} \cdot \exp\left(-\frac{1}{2}\left(\begin{bmatrix}-0{,}25 & 0{,}85\end{bmatrix} \cdot \begin{bmatrix}-3{,}974\\5{,}2635\end{bmatrix}\right)\right) =$$

$$= \frac{1}{2\pi\sqrt{0{,}08}} \cdot \exp\left(-\frac{1}{2} \times 5{,}47\right) = 0{,}0366$$

$$P(C=B \mid x_1 = 1, x_2 = 2) = \frac{P(C=B)\, P(x_1 = 1, x_2 = 2 \mid C = B)}{P(x_1 = 1, x_2 = 2)}$$

$$\Sigma = \begin{bmatrix} 2 & 0{,}467 \\ 0{,}467 & 0{,}41 \end{bmatrix}$$

$$\det = (2 \times 0{,}41) - (0{,}407)^2 = 0{,}602$$

$$\Sigma^{-1} = \begin{bmatrix} 0{,}41 & -0{,}407 \\ -0{,}467 & 2 \end{bmatrix} \times \frac{1}{0{,}602} = \begin{bmatrix} 0{,}68 & -0{,}78 \\ -0{,}78 & 3{,}323 \end{bmatrix}$$

$$N(\vec{x} \mid \vec{y}, \Sigma) = \frac{1}{(2\pi)^{2/2}\sqrt{0{,}602}} \times \exp\left(-\frac{1}{2}\left(\begin{bmatrix}1\\2\end{bmatrix} - \begin{bmatrix}3\\0{,}55\end{bmatrix}\right)^T \cdot \begin{bmatrix} 0{,}68 & -0{,}78 \\ -0{,}78 & 3{,}323 \end{bmatrix} \cdot \left(\begin{bmatrix}1\\2\end{bmatrix} - \begin{bmatrix}3\\0{,}55\end{bmatrix}\right)\right) =$$

$$= \frac{1}{2\pi \times \sqrt{0{,}602}} \times \exp\left(-\frac{1}{2}\begin{bmatrix}-2 & 1{,}45\end{bmatrix} \cdot \begin{bmatrix} 0{,}68 & -0{,}78 \\ -0{,}78 & 3{,}323 \end{bmatrix} \cdot \begin{bmatrix}-2 & 1{,}45\end{bmatrix}\right) =$$

$$= \frac{1}{2\pi \times \sqrt{0{,}602}} \times \exp\left(-\frac{1}{2}\left(\begin{bmatrix}-2 & 1{,}45\end{bmatrix} \cdot \begin{bmatrix}-2{,}491\\6{,}378\end{bmatrix}\right)\right) =$$

$$= \frac{1}{2\pi \times \sqrt{0{,}602}} \times \exp(-7{,}11) = 0{,}00017$$

b)

$$P(x_1=1, x_2=2) = 0{,}00017 + 0{,}0366 = 0{,}03677$$

$$P(C=A \mid x_1=1, x_2=2) = \frac{0{,}0366 \times \frac{1}{2}}{0{,}03677} = \frac{0{,}0995}{2} = 0{,}0448$$

$$P(C=B \mid x_1=1, x=2) = \frac{0{,}00017 \times \frac{1}{2}}{0{,}03677} = \frac{0{,}000462}{2} = 0{,}000231$$

Comparing the numerators, we find that A is the most probable class for the query vector

c)

x_3	class
0	A
1	A
1	A
0	A
1	B
1	B
0	B
1	B

$P(A) = \frac{1}{2}$ $\quad P(x_3=1) = \frac{5}{8}$

$P(B) = \frac{1}{2}$ $\quad P(x_3=0) = \frac{3}{8}$

	$P(x_3 \mid A)$	$P(x_3 \mid B)$
1	$\frac{1}{2}$	$\frac{3}{4}$
0	$\frac{1}{2}$	$\frac{1}{4}$

$$\cancel{P(A \mid x_3=1) = \frac{P(x_3 \mid A)\,P(A)}{P(x_3)}} \qquad \cancel{P(x_3) = P(x_3 \mid A) + P(x_3 \mid B)}$$

$$\cancel{P(B \mid x_3=1) = \frac{P(x_3 \mid B)\,P(B)}{P(x_3)}}$$

$$P(A \mid x_3=1) = \frac{P(x_3=1 \mid A)\,P(A)}{P(x_3=1)} = \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{5}{8}} = 0{,}4$$

$$P(B \mid x_3=1) = \frac{P(x_3=1 \mid B)\,P(B)}{P(x_3=1)} = \frac{\frac{3}{4} \times \frac{1}{2}}{\frac{5}{8}} = 0{,}6$$

The most probable class is class = B

d) $P(x_1, x_2, x_3, A) = \underline{P(x_1, x_2|A)P(x_3|A)} \cdot P(A)$

$P(x_1, x_2, x_3, B) = P(x_1, x_2|B)P(x_3|B) \cdot P(B)$

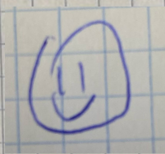
From b):

$\cancel{P(x_1, x_2|B) = 0}$

$P(x_1=1, x_2=2|A) = 0,0366$

$P(x_1=1, x_2=2|B) = 0,00017$

$P(A) = P(B) = \frac{1}{2}$



From c)

$P(x_3=1|A) = \frac{1}{2}$

$P(x_3=1|B) = \frac{3}{4}$

$P(x_1, x_2, x_3, A) = 0,0366 \times \frac{1}{2} \times \frac{1}{2} = 0,00915$

$P(x_1, x_2, x_3, B) = 0,00017 \times \frac{3}{4} \times \frac{1}{2} = 0,00006375$

$P(x_1, x_2, x_3) = P(x_1, x_2, x_3, A) + P(x_1, x_2, x_3, B) = 0,00915 + 0,00006375$

$= \cancel{0,00917125}\ 0,00921375$

the most probable class is A, ~~...~~

$P(x_1, x_2, C=A) = P(x_1=1, x_2=2|A)P(A) = 0,0183$

$P(x_1, x_2, C=B) = P(x_1=1, x_2=2|B)P(B) = 0,000085$

~~P(x_1 x~~

$P(A|x_1, x_2, x_3) = \frac{P(x_1, x_2, x_3, A)}{P(x_1, x_2, x_3)} = \cancel{0,99168}\ 0,993$

$P(B|x_1, x_2, x_3) = \frac{P(x_1, x_2, x_3, B)}{P(x_1, x_2, x_3)} = \cancel{0,00832}\ 0,007$

III Software Experiments

```python
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn import metrics, datasets

# Load the digits dataset
digits = datasets.load_digits()
X, y = digits.data, digits.target

# Split the digits dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, stratify=y, random_state=7)

# Print the sizes of the training and testing sets
print("Train size:", len(X_train), "\nTest size:", len(X_test))

# 1. Learn classifiers
knn1 = KNeighborsClassifier(n_neighbors=1)
knn3 = KNeighborsClassifier(n_neighbors=3)
gauss = GaussianNB()

knn1.fit(X_train, y_train)
knn3.fit(X_train, y_train)
gauss.fit(X_train, y_train)

# 2. Test classifiers on digits dataset
y_pred1 = knn1.predict(X_test)
y_pred3 = knn3.predict(X_test)
y_predg = gauss.predict(X_test)

print("kNN (k=1) accuracy on the testing set:", round(metrics.accuracy_score(y_test, y_pred1), 2))
print("kNN (k=3) accuracy on the testing set:", round(metrics.accuracy_score(y_test, y_pred3), 2))
print("Gaussian Naive Bayes accuracy on the testing set:", round(metrics.accuracy_score(y_test, y_predg), 2))

# Load the wine dataset
wine = datasets.load_wine()
X_wine, y_wine = wine.data, wine.target

# Split the wine dataset
X_train_wine, X_test_wine, y_train_wine, y_test_wine = train_test_split(X_wine, y_wine, train_size=0.7, stratify=y_wine, random_state=7)

# 1. Learn classifiers on wine dataset
knn1_wine = KNeighborsClassifier(n_neighbors=1)
knn3_wine = KNeighborsClassifier(n_neighbors=3)
gauss_wine = GaussianNB()

knn1_wine.fit(X_train_wine, y_train_wine)
knn3_wine.fit(X_train_wine, y_train_wine)
gauss_wine.fit(X_train_wine, y_train_wine)

# 2. Test classifiers on wine dataset
y_pred1_wine = knn1_wine.predict(X_test_wine)
y_pred3_wine = knn3_wine.predict(X_test_wine)
y_predg_wine = gauss_wine.predict(X_test_wine)

print("\nkNN (k=1) accuracy on the wine dataset:", round(metrics.accuracy_score(y_test_wine, y_pred1_wine), 2))
print("kNN (k=3) accuracy on the wine dataset:", round(metrics.accuracy_score(y_test_wine, y_pred3_wine), 2))
print("Gaussian Naive Bayes accuracy on the wine dataset:", round(metrics.accuracy_score(y_test_wine, y_predg_wine), 2))

```

```
Train size: 1257
Test size: 540
kNN (k=1) accuracy on the testing set: 0.99
kNN (k=3) accuracy on the testing set: 0.99
Gaussian Naive Bayes accuracy on the testing set: 0.85

kNN (k=1) accuracy on the wine dataset: 0.72
kNN (k=3) accuracy on the wine dataset: 0.67
Gaussian Naive Bayes accuracy on the wine dataset: 0.96
```

For the digits dataset, kNN with k=1 and k=3 works better because it's good at understanding how nearby things are connected, which is important for recognising patterns in pictures. Gaussian Naive Bayes doesn't work as well because it assumes things are not related and doesn't understand how things are arranged in pictures.

For the wine dataset, Gaussian Naive Bayes is better because it's good at handling data that has probabilities and assumes that different characteristics of wine are not connected, which is true for this dataset.