TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 020

I. Pen-and-paper

1)

y_1 $(C=0)$ $\mu = \frac{0.6+0.1+0.2+0.1}{4} = 0.25$ $(C=1)$ $\mu = \frac{0.3-0.1-0.3+0.2+0.4-0.2}{6} = 0.05$

$(C=0)$ $\sigma = \sqrt{\frac{(0.6-0.25)^2+(0.1-0.25)^2+(0.2-0.25)^2+(0.1-0.25)^2}{4-1}} = \sqrt{\frac{0.17}{3}} = 0.238$

$(C=1)$ $\sigma = \sqrt{\frac{(0.3-0.05)^2+(-0.1-0.05)^2+(-0.3-0.05)^2+(0.2-0.05)^2+(0.4-0.05)^2+(-0.2-0.05)^2}{6-1}} = \sqrt{\frac{0.415}{5}} = 0.288$

$p(y_1|C=0) = \frac{1}{\sqrt{2\pi}\sqrt{\frac{0.17}{3}}} \exp\left(-\frac{1}{2\times\frac{0.17}{3}}(y_1-0.25)^2\right)$

$p(y_2|C=1) = \frac{1}{\sqrt{2\pi}\sqrt{\frac{0.415}{5}}} \exp\left(-\frac{1}{2\times\frac{0.415}{5}}(y_1-0.05)^2\right)$

y_2

$p(y_2=A|C=0) = \frac{2}{4} = \frac{1}{2}$ $\quad p(y_2=A|C=1) = \frac{1}{6}$

$p(y_2=B|C=0) = \frac{1}{4}$ $\quad p(y_2=B|C=1) = \frac{2}{6} = \frac{1}{3}$

$p(y_2=C|C=0) = \frac{1}{4}$ $\quad p(y_2=C|C=1) = \frac{3}{6} = \frac{1}{2}$

$\{y_3, y_4\}$ $C=0$

$\mu = \frac{1}{4}\left(\begin{bmatrix}0.2\\0.4\end{bmatrix}+\begin{bmatrix}-0.1\\-0.4\end{bmatrix}+\begin{bmatrix}-0.1\\0.2\end{bmatrix}+\begin{bmatrix}0.8\\0.8\end{bmatrix}\right) = \begin{bmatrix}0.2\\0.25\end{bmatrix}$

$\Sigma_{00} = \frac{1}{4-1}\left[(0.2-0.2)^2+(-0.1-0.2)^2+(-0.1-0.2)^2+(0.8-0.2)^2\right] = \frac{0.54}{3} = 0.18$

$\Sigma_{11} = \frac{1}{4-1}\left[(0.4-0.25)^2+(-0.4-0.25)^2+(0.2-0.25)^2+(0.8-0.25)^2\right] = \frac{0.75}{3} = 0.25$

$\Sigma_{01} = \Sigma_{10} = \frac{1}{4-1}\left[(0.2-0.2)(0.4-0.25)+(-0.1-0.2)(-0.4-0.25)+\right.$
$\left.+(-0.1-0.2)(0.2-0.25)+(0.8-0.2)(0.8-0.25)\right] = \frac{0.54}{3} = 0.18$

$\Sigma = \begin{bmatrix}0.18 & 0.18\\0.18 & 0.25\end{bmatrix}$ $\quad |\Sigma| = 0.18\times0.25 - 0.18^2 = 0.0126$

$\Sigma^{-1} = \frac{1}{0.0126}\begin{bmatrix}0.25 & -0.18\\-0.18 & 0.18\end{bmatrix} = \begin{bmatrix}19.84 & -14.29\\-14.29 & 14.29\end{bmatrix}$

$C=1$

$\mu = \frac{1}{6}\left(\begin{bmatrix}0.1\\0.3\end{bmatrix}+\begin{bmatrix}0.2\\-0.2\end{bmatrix}+\begin{bmatrix}-0.1\\0.2\end{bmatrix}+\begin{bmatrix}0.5\\0.6\end{bmatrix}+\begin{bmatrix}-0.4\\-0.7\end{bmatrix}+\begin{bmatrix}0.4\\0.3\end{bmatrix}\right) = \begin{bmatrix}0.117\\0.083\end{bmatrix}$

$\Sigma_{00} = \frac{1}{6-1}\left[(0.1-0.117)^2+(0.2-0.117)^2+(-0.1-0.117)^2+(0.5-0.117)^2+\right.$
$\left.+(-0.4-0.117)^2+(0.4-0.117)^2\right] = \frac{0.548}{5} = 0.11$

$\Sigma_{11} = \frac{1}{6-1}\left[(0.3-0.083)^2+(-0.2-0.083)^2+(0.2-0.083)^2+(0.6-0.083)^2+\right.$
$\left.+(-0.7-0.083)^2+(0.3-0.083)^2\right] = \frac{1.068}{5} = 0.21$

$\Sigma_{01} = \Sigma_{10} = \frac{1}{6-1}\left[(0.1-0.117)(0.3-0.083)+(0.2-0.117)(-0.2-0.083)+\right.$
$\left.+(-0.1-0.117)(0.2-0.083)+(0.5-0.117)(0.6-0.083)+(-0.4-0.117)\times\right.$
$\left.\times(-0.7-0.083)+(0.4-0.117)(0.3-0.083)\right] = \frac{0.612}{5} = 0.12$

$\Sigma = \begin{bmatrix}0.11 & 0.12\\0.12 & 0.21\end{bmatrix}$ $\quad |\Sigma| = 0.11\times0.21 - 0.12^2 = 0.0087$

$\Sigma^{-1} = \frac{1}{0.0087}\begin{bmatrix}0.21 & -0.12\\-0.12 & 0.11\end{bmatrix} = \begin{bmatrix}24.14 & -13.79\\-13.79 & 12.64\end{bmatrix}$

$\{y_3, y_4\} = [a_0 \;\; a_1]^T$

$P(\{y_3,y_4\}|C=0) = \frac{1}{(2\pi)^{2/2}\sqrt{0.0126}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix}a_0\\a_1\end{bmatrix}-\begin{bmatrix}0.2\\0.25\end{bmatrix}\right)^T\begin{bmatrix}19.84 & -14.29\\-14.29 & 14.29\end{bmatrix}\left(\begin{bmatrix}a_0\\a_1\end{bmatrix}-\begin{bmatrix}0.2\\0.25\end{bmatrix}\right)\right)$

$P(\{y_3,y_4\}|C=1) = \frac{1}{(2\pi)^{2/2}\sqrt{0.0087}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix}a_0\\a_1\end{bmatrix}-\begin{bmatrix}0.117\\0.083\end{bmatrix}\right)^T\begin{bmatrix}24.14 & -13.79\\-13.79 & 12.64\end{bmatrix}\left(\begin{bmatrix}a_0\\a_1\end{bmatrix}-\begin{bmatrix}0.117\\0.083\end{bmatrix}\right)\right)$

2)

x_1

$p(C=0|y_1=0.6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\}) = \frac{p(C=0)\times p(y_1=6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\}|C=0)}{\propto}$

$= \frac{p(C=0)\times p(y_1=6|C=0)\times p(y_2=A|C=0)\times p(\{y_3,y_4\}=\{0.2,0.4\}|C=0)}{\propto}$

$= \frac{\frac{4}{10}\times\frac{1}{\sqrt{2\pi}\sqrt{\frac{0.17}{3}}}\exp\left(-\frac{1}{2\times\frac{0.17}{3}}(0.6-0.25)^2\right)\times\frac{1}{2}\times\frac{1}{2\pi\sqrt{0.0126}}\exp\left(-\frac{1}{2}\left(\begin{bmatrix}0.2\\0.4\end{bmatrix}-\begin{bmatrix}0.2\\0.25\end{bmatrix}\right)^T\begin{bmatrix}19.84 & -14.29\\-14.29 & 14.29\end{bmatrix}\left(\begin{bmatrix}0.2\\0.4\end{bmatrix}-\begin{bmatrix}0.2\\0.25\end{bmatrix}\right)\right)}{\propto}$

$= \frac{0.137300}{\propto}$

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 020

$$P(c=1 \mid y_1=0.6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\}) = \frac{P(c=1)\times P(y_1=6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\} \mid c=1)}{\propto}$$

$$= \frac{P(c=1)\times P(y_1=6 \mid c=1)\times P(y_2=A \mid c=1)\times P(\{y_3,y_4\}=\{0.2,0.4\} \mid c=1)}{\propto}$$

$$= \frac{\frac{6}{10}\times\frac{1}{\sqrt{2\pi}\sqrt{\frac{0.415}{5}}}\exp\left(-\frac{1}{2\times\frac{0.415}{5}}(0.6-0.05)^2\right)\times\frac{1}{6}\times\frac{1}{2\pi\sqrt{0.0087}}\exp\left(-\frac{1}{2}\left(\begin{bmatrix}0.2\\0.4\end{bmatrix}-\begin{bmatrix}0.117\\0.083\end{bmatrix}\right)^T\begin{bmatrix}24.14 & -13.79\\ -13.79 & 12.64\end{bmatrix}\left(\begin{bmatrix}0.2\\0.4\end{bmatrix}-\begin{bmatrix}0.117\\0.083\end{bmatrix}\right)\right)}{\propto}$$

$$= \frac{0.026771}{\propto}$$

$$\frac{0.137300+0.026771}{\propto} = 1 \iff \propto = 0.164071$$

→ $P(c=0 \mid y_1=0.6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\}) = \frac{0.137300}{0.164071} = 0.84$

→ $P(c=1 \mid y_1=0.6, y_2=A, \{y_3,y_4\}=\{0.2,0.4\}) = \frac{0.026771}{0.164071} = 0.16$ — True negative

(x2)

→ $P(c=0 \mid y_1=0.1, y_2=B, \{y_3,y_4\}=\{0.1,0.4\}) = \frac{0.063252}{0.319388} = 0.20$

→ $P(c=1 \mid y_1=0.1, y_2=B, \{y_3,y_4\}=\{0.1,0.4\}) = \frac{0.256136}{0.319388} = 0.80$ — False positive

(x3)

→ $P(c=0 \mid y_1=0.2, y_2=A, \{y_3,y_4\}=\{0.1,0.2\}) = \frac{0.231700}{0.307227} = 0.75$

→ $P(c=1 \mid y_1=0.2, y_2=A, \{y_3,y_4\}=\{0.1,0.2\}) = \frac{0.075527}{0.307227} = 0.25$ — True negative

(x4)

→ $P(c=0 \mid y_1=0.1, y_2=C, \{y_3,y_4\}=\{0.8,0.8\}) = \frac{0.070476}{0.153750} = 0.46$

→ $P(c=1 \mid y_1=0.1, y_2=C, \{y_3,y_4\}=\{0.8,0.8\}) = \frac{0.083273}{0.153750} = 0.54$ — False positive

(x5)

→ $P(c=0 \mid y_1=0.3, y_2=B, \{y_3,y_4\}=\{0.1,0.3\}) = \frac{0.192500}{0.420583} = 0.46$

→ $P(c=1 \mid y_1=0.3, y_2=B, \{y_3,y_4\}=\{0.1,0.3\}) = \frac{0.228082}{0420583} = 0.54$ — True positive

(x6)

→ $P(c=0 \mid y_1=-0.1, y_2=C, \{y_3,y_4\}=\{0.2,0.2\}) = \frac{0.018971}{0.267329} = 0.07$

→ $P(c=1 \mid y_1=-0.1, y_2=C, \{y_3,y_4\}=\{0.2,0.2\}) = \frac{0.248358}{0.267329} = 0.93$ — True positive

(x7)

→ $P(c=0 \mid y_1=-0.3, y_2=C, \{y_3,y_4\}=\{0.1,0.2\}) = \frac{0.008209}{0.132260} = 0.06$

→ $P(c=1 \mid y_1=-0.3, y_2=C, \{y_3,y_4\}=\{0.1,0.2\}) = \frac{0.124051}{0.132260} = 0.94$ — True positive

(x8)

→ $P(c=0 \mid y_1=0.2, y_2=B, \{y_3,y_4\}=\{0.5,0.6\}) = \frac{0.177860}{0.376879} = 0.47$

→ $P(c=1 \mid y_1=0.2, y_2=B, \{y_3,y_4\}=\{0.5,0.6\}) = \frac{0.199019}{0.376879} = 0.53$ — True positive

(x9)

→ $P(c=0 \mid y_1=0.4, y_2=A, \{y_3,y_4\}=\{0.4,0.7\}) = \frac{0.059801}{0.084544} = 0.71$

→ $P(c=1 \mid y_1=0.4, y_2=A, \{y_3,y_4\}=\{0.4,0.7\}) = \frac{0.024742}{0.084544} = 0.29$ — False negative

(x10)

→ $P(c=0 \mid y_1=-0.2, y_2=C, \{y_3,y_4\}=\{0.4,0.3\}) = \frac{0.030330}{0.350777} = 0.09$

→ $P(c=1 \mid y_1=-0.2, y_2=C, \{y_3,y_4\}=\{0.4,0.3\}) = \frac{0.320446}{0.350777} = 0.91$ — True positive

Confusion matrix:

Predicted \ True	N	P
N	2	1
P	2	5

Aprendizagem 2021/22

Homework I – Group 020

3)

$$Precision(N) = \frac{2}{3} \qquad Recall(N) = \frac{2}{4} = \frac{1}{2} \qquad F_1(N) = \left(\frac{\left(\frac{2}{3}\right)^{-1} + \left(\frac{1}{2}\right)^{-1}}{2}\right)^{-1} = \left(\frac{\frac{3}{2} + \frac{4}{2}}{2}\right)^{-1} = \left(\frac{7}{4}\right)^{-1} = \frac{4}{7}$$

$$Precision(P) = \frac{5}{7} \qquad Recall(P) = \frac{5}{6} \qquad F_1(P) = \left(\frac{\left(\frac{5}{7}\right)^{-1} + \left(\frac{5}{6}\right)^{-1}}{2}\right)^{-1} = \left(\frac{\frac{7}{5} + \frac{6}{5}}{2}\right) = \left(\frac{13}{10}\right)^{-1} = \frac{10}{13}$$

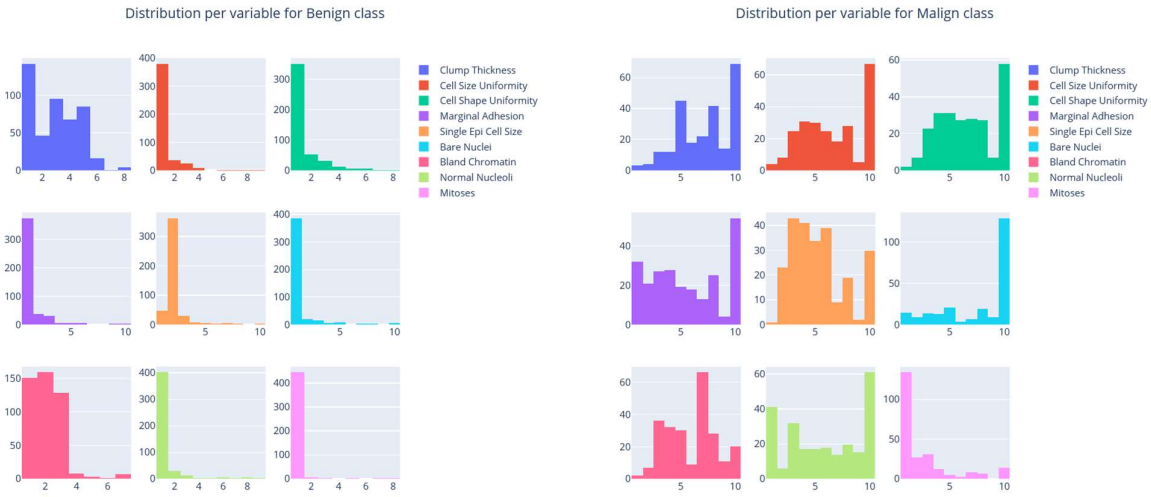
4) The table below shows the predicted class for each x_i, considering different thresholds. x_i is considered Negative (class = 0) if the probability of that x_i being class = 0 is greater or equal to the threshold.

			Tresholds										
	p(c=0|x_i)	p(c=1|x_i)	0.07	0.08	0.10	0.20	0.46	0.47	0.48	0.5	0.72	0.75	0.84
x_1	0.84	0.16	0	0	0	0	0	0	0	0	0	0	0
x_2	0.20	0.80	0	0	0	0	1	1	1	1	1	1	1
x_3	0.75	0.25	0	0	0	0	0	0	0	0	0	0	1
x_4	0.46	0.54	0	0	0	0	0	1	1	1	1	1	1
x_5	0.46	0.54	0	0	0	0	0	1	1	1	1	1	1
x_6	0.07	0.93	0	1	1	1	1	1	1	1	1	1	1
x_7	0.06	0.94	1	1	1	1	1	1	1	1	1	1	1
x_8	0.47	0.53	0	0	0	0	0	0	1	1	1	1	1
x_9	0.71	0.29	0	0	0	0	0	0	0	0	1	1	1
x_{10}	0.09	0.91	0	0	1	1	1	1	1	1	1	1	1
training accuracy			5/10	6/10	7/10	7/10	6/10	6/10	7/10	7/10	8/10	8/10	7/10

As shown above the thresholds that optimize training accuracy are 0.71 and 0.75 both having an accuracy of 80%. This means that it's harder for x_i to be considered Negative since that probability has to be above around 0.7 unlike it is to be considered Positive since that probability just has to be above around 0.3.

II. Programming and critical analysis

5)



TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 020

6) By running the program we get the following accuracy and standard deviation values of kNN under $k \in \{3, 5, 7\}$ using both the testing and the training set.

k	Average Test Accuracy	Average Train Accuracy	Standard deviation for each fold in testing set	Standard deviation for each fold in training set
3	0.956	0.982	0.02364	0.00389
5	0.956	0.979	0.01934	0.00300
7	0.956	0.982	0.01989	0.00217

Table 1: Test and train accuracy of *kNN* under $k \in \{3, 5, 7\}$

When analyzing this values we notice that for $k = 5$ the difference between average train accuracy and average test accuracy is a little smaller than for $k = 3$ and $k = 7$, with the average test accuracy being the same for the three of them. To check in more detail if there are significant variations when moving from the training set to the testing set we have to analyze the standard deviation for each fold. By doing this we see that $k = 5$ has the smaller standard deviation for the testing set and the second smaller for the training set, but still very similar to the other k values. This can mean that with $k = 5$, in spite of not having the best results in the training phase, it can adapt well in the testing phase and therefore is the less susceptible to the overfitting risk.

7) Using the T-test to determine if there is a significant difference between the results predicted with the kNN classifier and the Naïve Bayes classifier (multinomial assumption) we get the following values.

T-value	P-value	kNN accuracy	Multinomial Naïve Bayes accuracy
0.574	0.568	0.956	0.912

Table 2: Values obtained in T-test and accuracies for each classifier

With this p-value on a t-distribution table we get a t-value of approximately 0.675. The obtained t-value is lower than this one which means we can't reject the null hypothesis that there is no significant difference between this two groups. Since the accuracy with kNN is similar than with Multinomial Naïve Bayes we can't affirm that the first classifier is statistically superior to the last one.

8) Two reasons that can underlie the difference in performance between kNN and Naïve Bayes classifiers are for example the overfitting risk like we observed in kNN with $k = 3$ and the fact that the Naïve Bayes classifier assumes that all features are independent which rarely happens in real life, limiting the application of this algorithm in real-world use cases like this one.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 020

III. APPENDIX

```
import numpy as np
import pandas as pd
from scipy.io import arff
from plotly.subplots import make_subplots
import plotly.graph_objects as go
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
import statistics as st
from scipy import stats

data = arff.loadarff(r'/home/sara/apre/tpc-1/breast.w.arff')

# ------------------ EXERCISE 5 --------------------
b_y1, b_y2, b_y3, b_y4, b_y5, b_y6, b_y7, b_y8, b_y9 = ([] for i in range(9))
m_y1, m_y2, m_y3, m_y4, m_y5, m_y6, m_y7, m_y8, m_y9 = ([] for i in range(9))

for aux in data[0]:
    if(aux['Class'].decode('UTF-8') == 'benign'):
        b_y1.append(aux['Clump_Thickness'])
        b_y2.append(aux['Cell_Size_Uniformity'])
        b_y3.append(aux['Cell_Shape_Uniformity'])
        b_y4.append(aux['Marginal_Adhesion'])
        b_y5.append(aux['Single_Epi_Cell_Size'])
        b_y6.append(aux['Bare_Nuclei'])
        b_y7.append(aux['Bland_Chromatin'])
        b_y8.append(aux['Normal_Nucleoli'])
        b_y9.append(aux['Mitoses'])
    else:
        m_y1.append(aux['Clump_Thickness'])
        m_y2.append(aux['Cell_Size_Uniformity'])
        m_y3.append(aux['Cell_Shape_Uniformity'])
        m_y4.append(aux['Marginal_Adhesion'])
        m_y5.append(aux['Single_Epi_Cell_Size'])
        m_y6.append(aux['Bare_Nuclei'])
        m_y7.append(aux['Bland_Chromatin'])
        m_y8.append(aux['Normal_Nucleoli'])
        m_y9.append(aux['Mitoses'])

# benign class
fig = make_subplots(rows=3, cols=3)

trace1 = go.Histogram(x=b_y1, name="Clump Thickness")
trace2 = go.Histogram(x=b_y2, name="Cell Size Uniformity")
trace3 = go.Histogram(x=b_y3, name="Cell Shape Uniformity")
trace4 = go.Histogram(x=b_y4, name="Marginal Adhesion")
trace5 = go.Histogram(x=b_y5, name="Single Epi Cell Size")
trace6 = go.Histogram(x=b_y6, name="Bare Nuclei")
trace7 = go.Histogram(x=b_y7, name="Bland Chromatin")
trace8 = go.Histogram(x=b_y8, name="Normal Nucleoli")
trace9 = go.Histogram(x=b_y9, name="Mitoses")

fig.append_trace(trace1, 1, 1)
fig.append_trace(trace2, 1, 2)
fig.append_trace(trace3, 1, 3)
fig.append_trace(trace4, 2, 1)
fig.append_trace(trace5, 2, 2)
fig.append_trace(trace6, 2, 3)
fig.append_trace(trace7, 3, 1)
fig.append_trace(trace8, 3, 2)
```

```
fig.append_trace(trace9, 3, 3)
fig.update_layout(title_text='Benign Class Distribution Per Variable', title_x=0.5)
fig.show()

# malign class
fig = make_subplots(rows=3, cols=3)

trace1 = go.Histogram(x=m_y1, name="Clump Thickness")
trace2 = go.Histogram(x=m_y2, name="Cell Size Uniformity")
trace3 = go.Histogram(x=m_y3, name="Cell Shape Uniformity")
trace4 = go.Histogram(x=m_y4, name="Marginal Adhesion")
trace5 = go.Histogram(x=m_y5, name="Single Epi Cell Size")
trace6 = go.Histogram(x=m_y6, name="Bare Nuclei")
trace7 = go.Histogram(x=m_y7, name="Bland Chromatin")
trace8 = go.Histogram(x=m_y8, name="Normal Nucleoli")
trace9 = go.Histogram(x=m_y9, name="Mitoses")

fig.append_trace(trace1, 1, 1)
fig.append_trace(trace2, 1, 2)
fig.append_trace(trace3, 1, 3)
fig.append_trace(trace4, 2, 1)
fig.append_trace(trace5, 2, 2)
fig.append_trace(trace6, 2, 3)
fig.append_trace(trace7, 3, 1)
fig.append_trace(trace8, 3, 2)
fig.append_trace(trace9, 3, 3)

fig.update_layout(title_text='Malign Class Distribution Per Variable', title_x=0.5)
fig.show()


# ------------------ EXERCISE 6 --------------------
df = pd.DataFrame(data[0])
data_array = df.to_numpy()

x = np.empty((0,9))
y = np.empty((0,1))

for row in data_array:
    array_sum = np.sum(row[:-1])
    array_has_nan = np.isnan(array_sum)
    if(not array_has_nan):
        x = np.append(x, [row[:-1]], axis=0)
        y = np.append(y, row[-1])

kf = KFold(n_splits=10, random_state=20, shuffle=True)

knn = KNeighborsClassifier(n_neighbors=3)

acc_score_test = []
acc_score_train = []

for train_index , test_index in kf.split(x):
    x_train , x_test = x[train_index], x[test_index]
    y_train , y_test = y[train_index], y[test_index]

    knn.fit(x_train,y_train)
    y_pred_test = knn.predict(x_test)
    y_pred_train = knn.predict(x_train)
```

```
    acc_score_test.append(accuracy_score(y_pred_test, y_test))
    acc_score_train.append(accuracy_score(y_pred_train, y_train))
print("Accuracy testing set:", accuracy_score(y_test, y_pred_test))
print("Accuracy of each testing fold - {}".format(acc_score_test))
print("Standard deviation testing set:", st.stdev(acc_score_test))

print("Accuracy training set:", accuracy_score(y_train, y_pred_train))
print("Accuracy of each training fold - {}".format(acc_score_train))
print("Standard deviation training set:", st.stdev(acc_score_train))



# ------------------ EXERCISE 7 --------------------
mnb = MultinomialNB()

for train_index , test_index in kf.split(x):
    x_train , x_test = x[train_index], x[test_index]
    y_train , y_test = y[train_index], y[test_index]

    knn.fit(x_train,y_train)
    y_pred_knn = knn.predict(x_test)
    mnb.fit(x_train,y_train)
    y_pred_mnb = mnb.predict(x_test)

print("Accuracy kNN:", accuracy_score(y_test, y_pred_knn))
print("Accuracy MultinomialNB:", accuracy_score(y_test, y_pred_mnb))

y_aux_knn = np.empty((0,1))
y_aux_mnb = np.empty((0,1))

for idx, val in enumerate(y_pred_knn):
    if(val.decode('UTF-8') == 'benign'):
        y_aux_knn = np.append(y_aux_knn, 1)
    else:
        y_aux_knn = np.append(y_aux_knn, 0)

for idx, val in enumerate(y_pred_mnb):
    if(val.decode('UTF-8') == 'benign'):
        y_aux_mnb = np.append(y_aux_mnb, 1)
    else:
        y_aux_mnb = np.append(y_aux_mnb, 0)

print("T-test:", stats.ttest_rel(y_aux_knn, y_aux_mnb))
```

END