Aprendizagem 2021/22

Homework I – Group XXX

I. Pen-and-paper

1)

	y_1	y_2	C
x_1	A	0	1
x_2	B	1	1
x_3	A	1	1
x_4	A	0	1
x_5	B	0	0
x_6	B	0	0
x_7	A	1	0
x_8	B	1	0

$$d(x_1, x_2) = \text{Hamming}(x_1, x_2) + \frac{1}{2}$$

$$= (\underbrace{1}_{A \neq B} + \underbrace{1}_{1 \neq 0}) + 1/2$$

$$= \frac{5}{2}$$

...

$$\text{Weighted mode} = \frac{1}{d_1} + \frac{1}{d_2} + \dots$$

$d(x_1, x_2)$	x_1 (+)	x_2 (+)	x_3 (+)	x_4 (+)	x_5 (−)	x_6 (−)	x_7 (−)	x_8 (−)	
x_1		5/2	3/2	1/2	3/2	3/2	3/2	5/2	+
x_2	5/2		3/2	5/2	3/2	3/2	3/2	1/2	−
x_3	3/2	3/2		3/2	5/2	5/2	1/2	3/2	−
x_4	1/2	5/2	3/2		3/2	3/2	3/2	5/2	+
x_5	3/2	3/2	5/2	3/2		1/2	5/2	3/2	−
x_6	3/2	3/2	5/2	3/2	1/2		5/2	3/2	−
x_7	3/2	3/2	1/2	3/2	5/2	5/2		3/2	+
x_8	5/2	1/2	3/2	5/2	3/2	3/2	3/2		+

$+ \leftarrow \boxed{x_1 +}\ \frac{2}{3} + \frac{2}{1} = \frac{8}{3}\quad \boxed{x_1 -}\ \frac{2}{3} + \frac{2}{3} + \frac{2}{3} = 2$

$- \leftarrow \boxed{x_2 +}\ \frac{2}{3}\quad \boxed{x_2 -}\ \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{1}{3} = \frac{7}{3}$

$- \leftarrow \boxed{x_3 +}\ \frac{2}{3} + \frac{2}{3} + \frac{2}{3} = 2\quad \boxed{x_3 -}\ 2 + 2/3 = 8/3$

$+ \leftarrow \boxed{x_4 +}\ 2 + 2/3 = 8/3\quad \boxed{x_4 -}\ 2/3 + 2/3 + 2/3 = 2$

$- \leftarrow \boxed{x_5 +}\ 2/3 + 2/3 + 2/3 = 2\quad \boxed{x_5 -}\ 2 + 2/3 = 8/3$

$- \leftarrow \boxed{x_6 +}\ 2/3 + 2/3 + 2/3 = 2\quad \boxed{x_6 -}\ 2 + 2/3 = 8/3$

$+ \leftarrow \boxed{x_7 +}\ 2/3 + 2/3 + 2 + 2/3 = 4\quad \boxed{x_7 -}\ 2/3$

$+ \leftarrow \boxed{x_8 +}\ 2 + 2/3 = 8/3\quad \boxed{x_8 -}\ 2/3 + 2/3 + 2/3 = 2$

$\Downarrow$

$$\text{recall} = \frac{TP}{TP + FN} = \frac{\overbrace{2}^{x_1, x_4}}{2 + \underbrace{2}_{x_7, x_8}} = \frac{1}{2}$$

2)

	y_1	y_2	y_3	C
x_1	A	0	1.2	1
x_2	B	1	0.8	1
x_3	A	1	0.5	1
x_4	A	0	0.9	1
x_5	B	0	1	0
x_6	B	0	0.9	0
x_7	A	1	1.2	0
x_8	B	1	0.8	0
x_9	B	0	0.8	1

- $\{y_1, y_2\} \perp\!\!\!\perp \{y_3\} \Rightarrow P(y_1, y_2, y_3 | C) = P(y_1, y_2 | C) \times P(y_3 | C)$
- Bayesian Theorem: $P(c|y) = \dfrac{P(y|c)\,P(c)}{P(y)}$

Learn a Bayesian Classifier $\Rightarrow$ Compute it's parameters

$P(C)$

$P(C=1) = 5/9 \quad P(C=0) = 4/9$

$P(y_1, y_2 | C)$

$P(y_1 = A, y_2 = 0 | C=1) = 2/5 = 0.4$

$P(y_1 = A, y_2 = 1 | C=1) = 1/5 = 0.2$

$P(y_1 = B, y_2 = 0 | C=1) = 1/5 = 0.2$

$P(y_1 = B, y_2 = 1 | C=1) = 1/5 = 0.2$

$P(y_1 = A, y_2 = 0 | C=0) = 0$

$P(y_1 = A, y_2 = 1 | C=0) = 1/4 = 0.25$

$P(y_1 = B, y_2 = 0 | C=0) = 2/4 = 0.5$

$P(y_1 = B, y_2 = 1 | C=0) = 1/4 = 0.25$

$P(y_3 | C)$

$\mu_1/\sigma_1 \rightarrow$ mean/standard deviation of observations where $C=1$

$\mu_0/\sigma_0 \rightarrow$ mean/standard deviation of observations where $C=0$

$$\mu_1 = \frac{1.2 + 0.8 + 0.5 + 0.9 + 0.8}{5} = 0.84 \qquad \sigma_1 = \sqrt{\frac{(1.2-0.84)^2 + (0.8-0.84)^2 + (0.5-0.84)^2 + (0.9-0.84)^2 + (0.8-0.84)^2}{5-1}} \simeq 0.251$$

$$\mu_0 = \frac{1 + 0.9 + 1.2 + 0.8}{4} = 0.975 \qquad \sigma_0 = \dots \simeq 0.171$$

$$P(y_3 | C=1) \sim N(y_3 | \mu_1, \sigma_1^2) = \frac{1}{0.251\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2 \times 0.063}(y_3 - 0.84)^2\right)$$

$$\simeq 1.5894\, e^{-7.9365(y_3 - 0.84)^2}$$

$$P(y_3 | C=0) \sim N(y_3 | \mu_0, \sigma_0^2) = \frac{1}{0.171\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2 \times 0.2924}(y_3 - 0.975)^2\right)$$

$$\simeq 2.333\, e^{-17.1(y_3 - 0.975)^2}$$

3) under MAP assumption, $P(c=1 \mid \theta_i) = \dfrac{P(\theta_i \mid c=1)P(c=1)}{P(\theta_i)}$

$\theta_1 = (A, 1, 0.8)$ $\theta_2 = (B\ 1\ 1)$ $\theta_3 = (B\ 0\ 0.9)$

$$P(\theta_1 \mid c=1) = P(A,1 \mid c=1)\ P(0.8 \mid c=1)$$
$$= 0.2 \times N(0.8 \mid \mu_1, \sigma_1^2)$$
$$= 0.2 \cdot 1.56934$$
$$= 0.31387$$

$$P(\theta_1 \mid c=0) = P(A,1 \mid c=0)\ P(0.8 \mid c=0)$$
$$= 0.25 \times N(0.8 \mid \mu_0, \sigma_0^2)$$
$$\simeq 0.25 \cdot 1.38191$$
$$= 0.34548$$

$$P(\theta_2 \mid c=1) = P(B,1 \mid c=1)\ P(1 \mid c=1)$$
$$= 0.2 \times N(1 \mid \mu_1, \sigma_1^2)$$
$$= 0.2 \cdot 1.29717$$
$$= 0.25943$$

$$P(\theta_2 \mid c=0) = P(B,1 \mid c=0)\ P(1 \mid c=0)$$
$$= 0.25 \times N(1 \mid \mu_0, \sigma_0^2)$$
$$\simeq 0.25 \cdot 2.3082$$
$$= 0.577$$

$$P(\theta_3 \mid c=1) = P(B,0 \mid c=1)\ P(0.9 \mid c=1)$$
$$= 0.2 \times N(0.9 \mid \mu_1, \sigma_1^2)$$
$$= 0.2 \cdot 1.54463$$
$$= 0.30893$$

$$P(\theta_3 \mid c=0) = P(B,0 \mid c=0)\ P(0.9 \mid c=0)$$
$$= 0.5 \times N(0.9 \mid \mu_0, \sigma_0^2)$$
$$\simeq 0.5 \cdot 2.1190$$
$$= 1.0595$$

$$P(c=1 \mid \theta_1) = \frac{P(\theta_1 \mid c=1)P(c=1)}{P(\theta_1)}$$
$$= \frac{0.31387 \cdot 5/9}{P(\theta_1)}$$
$$= \frac{0.17434}{P(\theta_1)}$$

$$P(c=0 \mid \theta_1) = \frac{P(\theta_1 \mid c=0)P(c=0)}{P(\theta_1)}$$
$$= \frac{0.34548 \cdot 4/9}{P(\theta_1)}$$
$$= \frac{0.15355}{P(\theta_1)}$$

$$P(c=1 \mid \theta_1) = \frac{0.17434}{0.17434 + 0.15355}$$
$$= 0.53170$$

$$P(c=1 \mid \theta_2) = \frac{P(\theta_2 \mid c=1)P(c=1)}{P(\theta_2)}$$

$$= \frac{0.25943 \times 5/9}{P(\theta_2)}$$

$$= \frac{0.14413}{P(\theta_2)}$$

$$P(c=0 \mid \theta_2) = \frac{P(\theta_2 \mid c=0)P(c=0)}{P(\theta)}$$

$$= \frac{0.577 \times 4/9}{P(\theta_2)}$$

$$= \frac{0.25644}{P(\theta_2)}$$

$$P(c=1 \mid \theta_2) = \frac{0.14413}{0.14413 + 0.25644}$$

$$= 0.35981$$

$$P(c=1 \mid \theta_3) = \frac{P(\theta_3 \mid c=1)P(c=1)}{P(\theta_3)}$$

$$= \frac{0.30893 \times 5/9}{P(\theta_3)}$$

$$= \frac{0.17163}{P(\theta_3)}$$

$$P(c=0 \mid \theta_3) = \frac{P(\theta_3 \mid c=0)P(c=0)}{P(\theta_3)}$$

$$= \frac{1.0595 \times 4/9}{P(\theta_3)}$$

$$= \frac{0.47089}{P(\theta_3)}$$

$$P(c=1 \mid \theta_3) = \frac{0.17163}{0.17163 + 0.47089}$$

$$= 0.26712$$

Therefore, the posterior probabilities of $C = \text{Positive}$ (given $\theta_1, \theta_2, \theta_3$) are:

$$P(c=1 \mid \theta_1) = 0.53170$$

$$P(c=1 \mid \theta_2) = 0.35981$$

$$P(c=1 \mid \theta_3) = 0.26712$$

4) $\theta = 0.5$

$P(c=1 \mid \theta_1) = 0.53170 > 0.5 \Rightarrow$ Classify as Positive ✓

$P(c=1 \mid \theta_2) = 0.35981 < 0.5 \Rightarrow$ Classify as Negative ✗

$P(c=1 \mid \theta_3) = 0.26712 < 0.5 \Rightarrow$ Classify as Negative ✓

Accuracy $= 2/3$

$\theta = 0.3$

$P(c=1 \mid \theta_1) = 0.53170 > 0.3 \Rightarrow$ Classify as Positive ✓

$P(c=1 \mid \theta_2) = 0.35981 > 0.3 \Rightarrow$ Classify as Positive ✓

$P(c=1 \mid \theta_3) = 0.26712 < 0.3 \Rightarrow$ Classify as Negative ✓

Accuracy $= 1$

$\theta = 0.7$

$P(c=1 \mid \theta_1) = 0.53170 < 0.7 \Rightarrow$ Classify as Negative ✗

$P(c=1 \mid \theta_2) = 0.35981 < 0.7 \Rightarrow$ Classify as Negative ✗

$P(c=1 \mid \theta_3) = 0.26712 < 0.7 \Rightarrow$ Classify as Negative ✓

Accuracy $= 1/3$

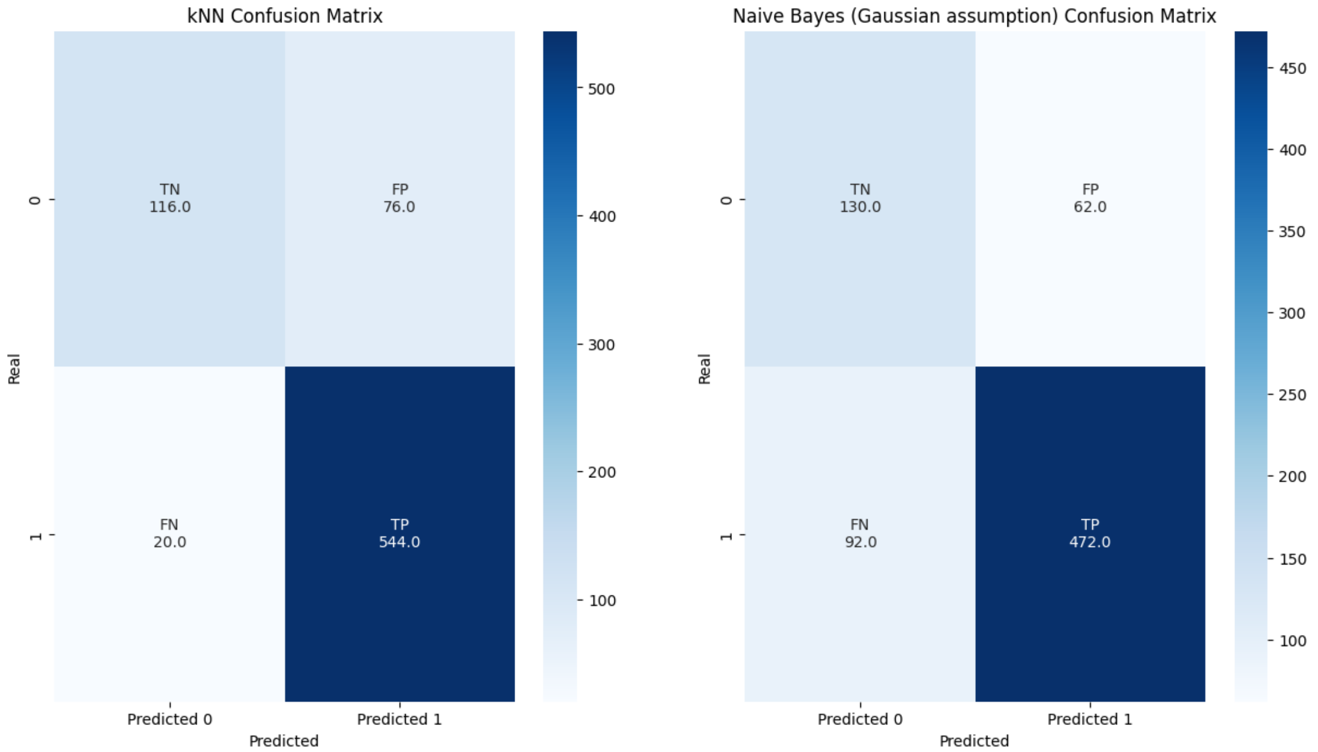
Therefore, the decision threshold of $\theta = 0.3$ optimizes test accuracy.

II. Programming and critical analysis

The code for the questions is in the Appendix of this document.

1)

Resulting Matrix:



2) H1: kNN < NB ? P-value = 0.9986831821715092.

   We fail to reject H0 and therefore kNN is statistically superior to Naive Bayes regarding accuracy.

3)

- The Naive Bayes' multinomial model assumes that all dataset's features are mutually independent. However, as it can be observed in the heatmap, the features are associated with each other, which can justify why the accuracy is smaller.

- The Naive Bayes' Gaussian assumption also assumes information about the features. In this case, it assumes the features follow a Normal distribution which can not be the case and therefore justify why the accuracy is smaller.

III. APPENDIX

Code used in question 1) of **II. Programming and critical analysis**:

Loads the Parkinson Disease's Data:

```
import numpy as np
import pandas as pd
from scipy.io.arff import loadarff
import seaborn as sns
import matplotlib.pyplot as plt

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']
```

Creates the 10-fold cross validator, the KNN and the Gaussian Naive Bayes. Caluculates the confusion matrix of kNN and Naive Bayes:

```
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.preprocessing import StandardScaler

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

knn = KNeighborsClassifier(n_neighbors=5, weights='uniform',
metric='euclidean')
nb = GaussianNB()

#function to calculate confusion matrix and accuracy so we dont repeat code
def calculate(x, X, Y):
    accuracy = []
    conf_mat = np.zeros((2, 2)) #inicializates the confusion matrix at 0's
    for train, test in skf.split(X, Y):
```

```
        X_train, X_test = X.iloc[train], X.iloc[test]
        Y_train, Y_test = Y.iloc[train], Y.iloc[test]
        #normalize data
        scaler = StandardScaler().fit(X_train)
        X_train = scaler.transform(X_train)
        X_test = scaler.transform(X_test)

        x.fit(X_train, Y_train)
        Y_pred = x.predict(X_test)

        accuracy.append(metrics.accuracy_score(Y_test, Y_pred))
        conf_mat += np.array(confusion_matrix(Y_test, Y_pred, labels=['0', '1']))
    return conf_mat, accuracy

nb_confusion = pd.DataFrame(calculate(nb, X, Y)[0], index=['0', '1'], columns=['Predicted 0', 'Predicted 1'])
knn_confusion = pd.DataFrame(calculate(knn, X, Y)[0], index=['0', '1'], columns=['Predicted 0', 'Predicted 1'])
```

Plots the accuracies in a graph:

```
titles = ["kNN Confusion Matrix", "Naive Bayes (Gaussian assumption) Confusion Matrix"]
matrices = [knn_confusion, nb_confusion]
plt.figure(figsize= (15, 8))
for i in range(len(matrices)):
    plt.subplot(1, 2, i+1)
    plt.title(titles[i])
    labels = np.array([["TN\n" + str(matrices[i].iat[0, 0]), "FP\n" + str(matrices[i].iat[0, 1])], ["FN\n" + str(matrices[i].iat[1, 0]), "TP\n" + str(matrices[i].iat[1, 1])]])
    sns.heatmap(matrices[i], annot=labels, fmt='', cmap="Blues")
    plt.xlabel("Predicted")
    plt.ylabel("Real")
```

Code used in question 2) of **II. Programming and critical analysis**:

```
#H0: kNN is statistically superior to Naive Bayes regarding accuracy
from scipy import stats
res = stats.ttest_rel(calculate(knn, X, Y)[1], calculate(nb, X, Y)[1], alternative='less')
print("kNN < nb? P-value =", res.pvalue)
```