TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 010

I. Pen-and-paper

1) Assuming 1 is positive and 0 is negative

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

	y_1	y_2	Class
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

$$x_1 \to weight\left(0{:}\, 3 * \frac{1}{3/2}, 1{:}\, \frac{1}{3/2} + \frac{1}{1/2}\right) = weight\left(0{:}\, 2, 1{:}\, \frac{8}{3}\right) \to TP$$

$$x_2 \to weight\left(0{:}\, \frac{1}{1/2} + 3 * \frac{1}{3/2}, 1{:}\, \frac{1}{3/2}\right) = weight\left(0{:}\, 4, 1{:}\, \frac{2}{3}\right) \to FN$$

$$x_3 \to weight\left(0{:}\, \frac{1}{1/2} + \frac{1}{3/2}, 1{:}\, 3 * \frac{1}{3/2}\right) = weight\left(0{:}\, \frac{8}{3}, 1{:}\, 2\right) \to FN$$

$$x_4 \to weight\left(0{:}\, 3 * \frac{1}{3/2}, 1{:}\, \frac{1}{1/2} + \frac{1}{3/2}\right) = weight\left(0{:}\, 2, 1{:}\, \frac{8}{3}\right) \to TP$$

$$x_5 \to weight\left(0{:}\, \frac{1}{1/2} + \frac{1}{3/2}, 1{:}\, 3 * \frac{1}{3/2}\right) = weight\left(0{:}\, \frac{8}{3}, 1{:}\, 2\right) \to TN$$

$$x_6 \to weight\left(0{:}\, \frac{1}{1/2} + \frac{1}{3/2}, 1{:}\, 3 * \frac{1}{3/2}\right) = weight\left(0{:}\, \frac{8}{3}, 1{:}\, 2\right) \to TN$$

$$x_7 \to weight\left(0{:}\, \frac{1}{3/2}, 1{:}\, \frac{1}{1/2} + 3 * \frac{1}{3/2}\right) = weight\left(0{:}\, \frac{2}{3}, 1{:}\, 4\right) \to FP$$

$$x_8 \to weight\left(0{:}\, 3 * \frac{1}{3/2}, 1{:}\, \frac{1}{1/2} + \frac{1}{3/2}\right) = weight\left(0{:}\, 2, 1{:}\, \frac{8}{3}\right) \to FP$$

$$recall = \frac{TP}{TP + FN} = \frac{2}{2 + 2} = \frac{1}{2}$$

Aprendizagem 2022/23
Homework II – Group 010

2) Assuming 1 is positive and 0 is negative

	y_1	y_2	y_3	Class
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

$$p(class = 1) = \frac{5}{9}$$

$$p(class = 0) = \frac{4}{9}$$

$$p(y_1 = A) = \frac{4}{9}$$

$$p(y_1 = B) = \frac{5}{9}$$

$$p(y_2 = 0) = \frac{5}{9}$$

$$p(y_2 = 1) = \frac{4}{9}$$

$$p(y_1 = A, y_2 = 0) = \frac{2}{9};\ p(y_1 = A, y_2 = 0 \mid class = 0) = 0;\ p(y_1 = A, y_2 = 0 \mid class = 1) = \frac{2}{5}$$

$$p(y_1 = A, y_2 = 1) = \frac{2}{9};\ p(y_1 = A, y_2 = 1 \mid class = 0) = \frac{1}{4};\ p(y_1 = A, y_2 = 1 \mid class = 1) = \frac{1}{5}$$

$$p(y_1 = B, y_2 = 0) = \frac{3}{9} = \frac{1}{3};\ p(y_1 = B, y_2 = 0 \mid class = 0) = \frac{2}{4} = \frac{1}{2};\ p(y_1 = B, y_2 = 0 \mid class = 1) = \frac{1}{5}$$

$$p(y_1 = B, y_2 = 1) = \frac{2}{9};\ p(y_1 = B, y_2 = 1 \mid class = 0) = \frac{1}{4};\ p(y_1 = B, y_2 = 1 \mid class = 1) = \frac{1}{5}$$

Para class = 0:

$$\mu = 0.975;\ \sigma = \frac{\sqrt{105}}{60};\ p(y_3 = x \mid class = 0) = \frac{60 * e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sqrt{210\pi}}$$

$$p(class = 0 \mid y_1 = a, y_2 = b, y_3 = c) = \frac{p(y_1 = a, y_2 = b, y_3 = c \mid class = 0) * p(class = 0)}{p(y_1 = a, y_2 = b, y_3 = c)}$$

$$= \frac{p(y_1 = a, y_2 = b \mid class = 0) * p(y_3 = c \mid class = 0) * p(class = 0)}{p(y_1 = a, y_2 = b) * (p(y_3 = c \mid class = 0) * p(class = 0) + p(y_3 = c \mid class = 1) * p(class = 1))}$$

Para class = 1:

$$\mu = 0.84;\ \sigma = \sqrt{0.063};\ p(y_3 = x \mid class = 1) = \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sqrt{0.126\pi}}$$

$$p(class = 1 \mid y_1 = a, y_2 = b, y_3 = c) = \frac{p(y_1 = a, y_2 = b, y_3 = c \mid class = 1) * p(class = 1)}{p(y_1 = a, y_2 = b, y_3 = c)}$$

$$= \frac{p(y_1 = a, y_2 = b \mid class = 1) * p(y_3 = c \mid class = 1) * p(class = 1)}{p(y_1 = a, y_2 = b) * (p(y_3 = c \mid class = 0) * p(class = 0) + p(y_3 = c \mid class = 1) * p(class = 1))}$$

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 010

3) Assuming 1 is positive and 0 is negative

$$\begin{pmatrix} A \\ 1 \\ 0.8 \end{pmatrix}: \; p\left(class = 1 |\, y_1 = A, y_2 = 1, y_3 = 0.8\right) =$$

$$= \frac{p(y_1 = A, y_2 = 1 |\, class = 1) * p(y_3 = 0.8\, | class = 1) * p(class = 1)}{p(y_1 = A, y_2 = 1) * (p(y_3 = 0.8 |\, class = 0) * p(class = 0) + p(y_3 = 0.8 |\, class = 1) * p(class = 1))}$$

$$= \frac{\frac{1}{5} * 1.569 * \frac{5}{9}}{\frac{2}{9} * (1.382 * \frac{4}{9} + 1.569 * \frac{5}{9})} = 0.5280$$

$$\begin{pmatrix} B \\ 1 \\ 1 \end{pmatrix}: \; p\left(class = 1 |\, y_1 = B, y_2 = 1, y_3 = 1\right) =$$

$$= \frac{p(y_1 = B, y_2 = 1 |\, class = 1) * p(y_3 = 1\, | class = 1) * p(class = 1)}{p(y_1 = B, y_2 = 1) * (p(y_3 = 1 |\, class = 0) * p(class = 0) + p(y_3 = 1 |\, class = 1) * p(class = 1))}$$

$$= \frac{\frac{1}{5} * 1.297 * \frac{5}{9}}{\frac{2}{9} * (2.311 * \frac{4}{9} + 1.297 * \frac{5}{9})} = 0.3711$$

$$\begin{pmatrix} B \\ 0 \\ 0.9 \end{pmatrix}: \; p\left(class = 1 |\, y_1 = B, y_2 = 0, y_3 = 0.9\right) =$$

$$= \frac{p(y_1 = B, y_2 = 0 |\, class = 1) * p(y_3 = 0.9\, | class = 1) * p(class = 1)}{p(y_1 = B, y_2 = 0) * (p(y_3 = 0.9 |\, class = 0) * p(class = 0) + p(y_3 = 0.9 |\, class = 1) * p(class = 1))}$$

$$= \frac{\frac{1}{5} * 1.545 * \frac{5}{9}}{\frac{1}{3} * (2.121 * \frac{4}{9} + 1.545 * \frac{5}{9})} = 0.2850$$

4)

$$f\left(\begin{pmatrix} A \\ 1 \\ 0.8 \end{pmatrix}, 0.3\right) = Positive(0.5280 > 0.3)$$

$$f\left(\begin{pmatrix} B \\ 1 \\ 1 \end{pmatrix}, 0.3\right) = Positive(0.3711 > 0.3)$$

$$f\left(\begin{pmatrix} B \\ 0 \\ 0.9 \end{pmatrix}, 0.3\right) = Negative(0.2850 \leq 0.3)$$

$$Accuracy = \frac{3}{3} = 1$$

$$f\left(\begin{pmatrix} A \\ 1 \\ 0.8 \end{pmatrix}, 0.5\right) = Positive$$

$$f\left(\begin{pmatrix} B \\ 1 \\ 1 \end{pmatrix}, 0.5\right) = Negative$$

$$f\left(\begin{pmatrix} B \\ 0 \\ 0.9 \end{pmatrix}, 0.5\right) = Negative$$

$$Accuracy = \frac{2}{3}$$

$$f\left(\begin{pmatrix} A \\ 1 \\ 0.8 \end{pmatrix}, 0.7\right) = Negative$$

$$f\left(\begin{pmatrix} B \\ 1 \\ 1 \end{pmatrix}, 0.7\right) = Negative$$

$$f\left(\begin{pmatrix} B \\ 0 \\ 0.9 \end{pmatrix}, 0.7\right) = Negative$$

$$Accuracy = \frac{1}{3}$$

The decision threshold 0.3 is the one that optimizes testing accuracy.

II. Programming and critical analysis

5)

Confusion Matrix Naïve Bayes

	Predicted class=0	Predicted class=1
Real class=0	67	125
Real class=1	69	495

Confusion Matrix kNN

	Predicted class=0	Predicted class=1
Real class=0	50	142
Real class=1	67	497

6) p-value = 0.91 with H_0: Naïve Bayes better or equal to kNN (H_1: kNN better than Naïve Bayes). This means that this hypothesis H_0 is accepted for levels of significance equal or under 91% and is rejected for higher levels. For the usual levels of significance (0.01, 0.05 and 0.1) H_0 is accepted and H_1 (the one we wanted to classify) is rejected. We can conclude that, in this situation, the hypothesis "kNN is statistically superior to Naïve Bayes regarding accuracy" is not true.

7)

1. kNN is sensitive to outliers. In this case, kNN only works with five elements (increasing the risk of overfit), while Naïve Bayes works with all of them.
2. Also, kNN did not considerate the weight and the data was not normalized, which may have decreased the accuracy.

III. APPENDIX

```
import pandas as pd
import math
from scipy.io.arff import loadarff
from sklearn.feature_selection import SelectKBest

from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

from scipy import stats

data = loadarff('pd_speech.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

y = df['class']

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

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

naive_bayes_classifier = GaussianNB()
knn_classifier = KNeighborsClassifier(n_neighbors=5, weights='uniform',
metric='euclidean')

cm_kNN = [[0, 0], [0, 0]]
cm_NB = [[0, 0], [0, 0]]

accuracy_kNN = []
accuracy_NB = []


for train_index, test_index in cv.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    naive_bayes_classifier.fit(X_train, y_train)
    y_pred = naive_bayes_classifier.predict(X_test)
    cm = metrics.confusion_matrix(y_test, y_pred)
    cm_NB = [ (a + b) for a, b in zip(cm_NB, cm) ]

    accuracy_NB += [metrics.accuracy_score(y_test, y_pred)]
```

```
    knn_classifier.fit(X_train, y_train)
    y_pred = knn_classifier.predict(X_test)
    cm = metrics.confusion_matrix(y_test, y_pred)
    cm_kNN = [ (a + b) for a, b in zip(cm_kNN, cm) ]

    accuracy_kNN += [metrics.accuracy_score(y_test, y_pred)]

confusion_NB = pd.DataFrame(cm_NB, index=['Real class=0', 'class=1'], columns=['Predicted
class=0', 'Predicted class=1'])

confusion_kNN = pd.DataFrame(cm_kNN, index=['Real class=0', 'class=1'],
columns=['Predicted class=0', 'Predicted class=1'])

print("Naïve Bayes Confusion Matrix\n ", confusion_NB)
print("\n\nkNN Confusion Matrix\n", confusion_kNN)

res = stats.ttest_rel(accuracy_kNN, accuracy_NB, alternative='greater')
print(res.pvalue)
```

END