TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I I – Group 102

I. Pen-and-paper

f)

	y_1	y_2	Out
x_1	A	0	P
x_2	B	1	P
x_3	A	1	P
x_4	A	0	P
x_5	B	0	N
x_6	B	0	N
x_7	A	1	N
x_8	B	1	N

hamming	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	–	2,5	1,5	0,5	1,5	1,5	1,5	2,5
		t	t	t	0	0	0	0

$$\frac{1}{0,5} + \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} = 0,667 > 0$$

x_1 é um true positive

hamming	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_2	2,5	-	1,5	2,5	1,5	1,5	1,5	0,5

$$\frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{0,5} < 0 = \emptyset$$

$\Downarrow\Downarrow$

x_2 é um false positive

hamming	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_3	1,5	1,5	-	1,5	2,5	2,5	0,5	1,5

$$\frac{1}{1,5} + \frac{1}{1,5} + \frac{1}{1,5} - \frac{1}{0,5} - \frac{1}{6,5} = -\frac{1}{1,5} < 0$$

$\Downarrow\Downarrow$

x_3 é um false positive

hamming	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_4	0,5	2,5	1,5	-	1,5	1,5	1,5	2,5

$$\frac{1}{0,5} + \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} - \frac{1}{1,5} = \frac{1}{1,5} > 0$$

$\Downarrow\Downarrow$

x_4 é um true positive

$Recall = \frac{2}{4} = \frac{1}{2} = 0,5$

2)

	y_1	y_2	y_3	Out
x_1	A	0	1,2	P
x_2	B	1	0,8	P
x_3	A	1	0,5	P
x_4	A	0	0,9	P
x_5	B	0	0,8	P
x_6	B	0	1	N
x_7	b	0	0,9	N
x_8	A	1	1,2	N
x_9	B	1	0,8	N

$P(P) = \frac{5}{9}$ $P(N) = \frac{4}{9}$

$$P(Class \mid y_1, y_2, y_3) =$$

$$= \frac{P(Class) \times P(y_1, y_2, y_3 \mid Class)}{P(y_1, y_2, y_3)}$$

$$= \frac{P(Class) \times P(y_1, y_2 \mid Class) \times P(y_3 \mid Class)}{P(y_1, y_2, y_3)}$$

$$P(y_3 \mid Class) = N(y_3 \mid \sigma^2; \mu)$$

$\sigma^2(y_3) = 0,0475$ $\quad P(y_3) = N(y_3 \mid 0,9; 0,0475)$

$\mu(y_3) = 0,9$

$$\mu(y_3 \mid Class = P) = \frac{1,2 + 0,8 + 0,5 + 0,9 + 0,8}{5} = 0,84$$

$\mu(y_3 \mid Class = N) = 0,975$

$\sigma^2(y_3 \mid Class = P) = 0,063$

$\sigma^2(y_3 \mid Class = N) = 0,029$

$P(y_3 \mid Class = P) = N(y_3 \mid 0,84; 0,063)$

$P(y_3 \mid class = N) = N(y_3 \mid 0,975; 0,029)$

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I I – Group 102

$P(y_1 = A, y_2 = 1) = \frac{2}{9}$ $P(y_1 = A, y_2 = 0) = \frac{2}{9}$

$P(y_1 = B, y_2 = 1) = \frac{2}{9}$ $P(y_1 = B, y_2 = 0) = \frac{3}{9}$

$P(y_1 = A, y_2 = 1 | Class = P) = \frac{1}{5}$

$P(y_1 = A, y_2 = 0 | Class = P) = \frac{2}{5}$

$P(y_1 = B, y_2 = 1 | Class = P) = \frac{1}{5}$

$P(y_1 = B, y_2 = 0 | Class = P) = \frac{1}{5}$

$P(y_1 = A, y_2 = 1 | Class = N) = \frac{1}{4}$

$P(y_1 = A, y_2 = 0 | Class = N) = 0$

$P(y_1 = B, y_2 = 1 | Class = N) = \frac{1}{4}$

$P(y_1 = B, y_2 = 0 | Class = N) = \frac{2}{4}$

3) $P(Classe = P) = \frac{5}{9}$ $P(class = N) = \frac{4}{9}$

$P(Class | y_1, y_2, y_3) =$

$= \frac{P(Class) \times P(y_1, y_2, y_3 | class)}{P(y_1, y_2, y_3)}$

$= \frac{P(Class) \times P(y_1, y_2 | Class) \times P(y_3 | Class)}{P(y_1, y_2, y_3)}$

$N(X | \mu, \sigma) = \frac{e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2}}{\sigma\sqrt{2\pi}}$

$$P(Class = P \mid y_1 = A, y_2 = I, y_3 = 0,8) =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times N(0,8 \mid 0,84; 0,063)}{\frac{2}{9} \times N(0,8 \mid 0,9; 0,0475)}$$

$$= \frac{\frac{1}{9}}{\frac{2}{9}} \times \frac{1,569}{1,648} = \frac{1}{2} \times 0,952 = 0,476$$

$$P(Class = P \mid y_1 = B, y_2 = I, y_3 = I) =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times N(1 \mid 0,84; 0,063)}{\frac{2}{9} \times N(1 \mid 0,9; 0,0475)}$$

$$= \frac{1}{2} \times 0,787 = 0,394$$

$$P(Class = P \mid y_1 = B, y_2 = 0, y_3 = 0,9) =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times N(0,9 \mid 0,84; 0,063)}{\frac{3}{9} \times N(0,9 \mid 0,9; 0,0475)}$$

$$= \frac{1}{3} \times 0,844 = 0,281$$

4) If the decision Threshold is either 0,5 or 0,7, 2 out of 3 observations are incorrectly assumed.

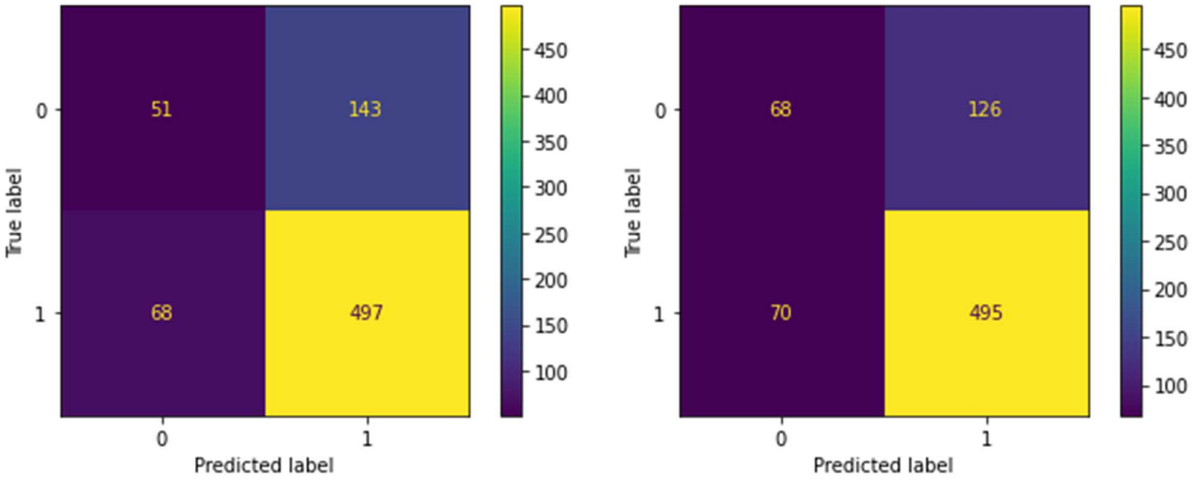
With it being 0,3, all of the observations are correctly assumed, therefore it optimizes the testing accuracy.

Aprendizagem 2022/23

Homework I I – Group 102

II. Programming and critical analysis

1.



KNN Naïve Bayes

2. Since the pvalue is 0.9104476998751558 which is higher than 0.05, there is no statistically significant difference between kNN and Naïve Bayes, not confirming the hypothesis.

3.

- While kNN classifier is affected by the quantity of data, the Naïve Bayes isn't
- While features independence is necessary for a Naïve Bayes classification, the kNN takes advantage of it by not depending on them.

III. APPENDIX

```
import pandas as pd, numpy as np
from scipy.io.arff import loadarff
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
from scipy import stats

#import dataset
data = loadarff('drive/MyDrive/ML/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']
```

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I I – Group 102

```
# folds dataset
skf = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 0)
# initialize main confusion matrixes
cm_knn = np.ndarray(shape = (2,2)).astype(int)
cm_gnb = np.ndarray(shape = (2,2)).astype(int)
# splits dataset
for train_index, test_index in skf.split(X, y):
  X_train, X_test = X.filter(items = train_index, axis = 0), X.filter(items = test_index, axis = 0)
  y_train, y_test = y.filter(items = train_index, axis = 0), y.filter(items = test_index, axis = 0)
# initialize classifiers
  knn = KNeighborsClassifier(n_neighbors = 5, weights = 'uniform', p = 2, metric = 'minkowski')
  gnb = GaussianNB()
  knn.fit(X_train, y_train)
  gnb.fit(X_train, y_train)
# train classifiers
  predictions_knn = knn.predict(X_test)
  predictions_gnb = gnb.predict(X_test)
# evaluate results
  tmp_knn = confusion_matrix(y_test, predictions_knn, labels = knn.classes_)
  tmp_gnb = confusion_matrix(y_test, predictions_gnb, labels = gnb.classes_)
# save results
  for i in range(2):
    for j in range(2):
      cm_knn[i][j] = cm_knn[i][j] + tmp_knn[i][j]
      cm_gnb[i][j] = cm_gnb[i][j] + tmp_gnb[i][j]

# display plots
disp = ConfusionMatrixDisplay(confusion_matrix = cm_knn)
disp.plot()
disp = ConfusionMatrixDisplay(confusion_matrix = cm_gnb)
disp.plot()
# compare KNN and GNB accuracies
result_knn = cross_val_score(KNeighborsClassifier(n_neighbors = 5, weights = 'uniform', p = 2, metri
c = 'minkowski'), X, y, cv = skf, scoring = 'accuracy')
result_gnb = cross_val_score(GaussianNB(), X, y, cv = skf, scoring = 'accuracy')
result = stats.ttest_rel(result_knn, result_gnb, alternative = 'greater')
print("pvalue =", result.pvalue)
```

END