Aprendizagem 2021/22

Homework I – Group 081

I. Pen-and-paper

Primeira pergunta:

Four positive observations, $\left\{\binom{A}{0},\binom{B}{1},\binom{A}{1},\binom{A}{0}\right\}$, and four negative observations, $\left\{\binom{B}{0},\binom{B}{0},\binom{A}{1},\binom{B}{1}\right\}$, were collected. Consider the problem of classifying observations as positive or negative.

1) [4v] Compute the recall of a distance-weighted kNN with $k = 5$ and distance $d(\mathbf{x}_1, \mathbf{x}_2) = Hamming(\mathbf{x}_1, \mathbf{x}_2) + \frac{1}{2}$ using leave-one-out evaluation schema (i.e., when classifying one observation, use all remaining ones).

Resposta:

$$Recall = \frac{true\ positive}{true\ positive + false\ negative}.$$

Observações positivas:

$$\left\{\binom{A}{0},\binom{B}{1},\binom{A}{1},\binom{A}{0}\right\}$$

Observações negativas:

$$\left\{\binom{B}{0},\binom{B}{0},\binom{A}{1},\binom{B}{1}\right\}$$

Conjunto total das observações: {x1,x2 , … , x8}

Número de features: 2 {y1,y2}

Class output: 2 {P,N}

Nota:

	y1	y2	out
x1	A	0	P
x2	B	1	P
x3	A	1	P
x4	A	0	P
x5	B	0	N
x6	B	0	N
x7	A	1	N
x8	B	1	N

Def:

Hamming(X1,X2) representa o número de entradas distintas entre X1 e X2

Exemplo:

X1 = (A,0)

X2 = (B,1)

Hamming(X1,X2) = 2

Aprendizagem 2021/22

Homework I – Group 081

Parametros de avaliação;

Distance-weighted,kNN with k = 5 and distance(xi,xj) = Hamming(xi,cj) + ½

Nota: O ½ permite o calculo do peso no fim das contas evitando a divisão por 0.

Weight = 1/distance

Avaliar distância x1:

<u>Positivos:</u>

Dis(x1,x2) = 2 + ½

Dis(x1,x3) = 1 + ½

Dis(x1,x4) = 0 + ½

<u>Negativos:</u>

Dis(x1,x5) = 1 + ½

Dis(x1,x6) = 1 + ½

Dis(x1,x7) = 1 + ½

Dis(x1,x8) = 2 + ½

Avaliar peso x1 para os 5 com menos distância:

<u>Positivos:</u>

w(x1,x3) = 1 / (1 + ½)

w(x1,x4) = 1 / (0 + ½)

<u>Negativos:</u>

w(x1,x5) = 1 / (1 + ½)

w(x1,x6) = 1 / (1 + ½)

w(x1,x7) = 1 / (1 + ½)

P: (2/3 + 2) = 8/3

N: (3 * 2/3) = 6/3

Concluimos que segundo estes parametros de avaliação x1 terá classe P (o que vai de acordo com as observações)

Como isto se verifica P será considerado um "True positive"

Iremos agora repetir o mesmo para todos os x`s

Chegaremos às seguintes conclusões:

	kNN out	Obsevado	Classificação
x1	P	P	TP
x2	N	P	FN
x3	N	P	FN
x4	P	P	TP
x5	N	N	TN
x6	N	N	TN
x7	N	N	TN
x8	N	N	TN

Conclusões:

Recall = TP / (TP + FN)

Recall = 2 / 4 = 0.5

Aprendizagem 2021/22
Homework I – Group 081

Segunda pergunta:

An additional positive observation was acquired, $\binom{B}{0}$, and a third variable y_3 was independently monitored, yielding estimates $y_3|P = \{1.2, 0.8, 0.5, 0.9, 0.8\}$ and $y_3|N = \{1, 0.9, 1.2, 0.8\}$.

2) [4v] Considering the nine training observations, learn a Bayesian classifier assuming: i) y_1 and y_2 are dependent, ii) $\{y_1, y_2\}$ and $\{y_3\}$ variable sets are independent and equally important, and ii) y_3 is normally distributed. Show all parameters.

Resposta:

Objetivo: Mostar todos os parametros necessários para aprender o "Bayesian classifier"

$$P(P \mid y1,y2,y3) = \frac{P(P) * P(y1,y2,y3 \mid P)}{P(y1,y2,y3)}$$

Trivial:

P(P) = 5/9

Menos simples:

>>> P(y1,y2,y3 | P)

Dado que os subconjuntos {y1,y2} e {y3} são independentes entre si podemos dizer que:

P(y1,y2,y3 | P) = P(y1,y2|P) * P(y3|P)

>> P(y1,y2|P)

Conjunto discreto [ALL COMBINATIONS]:

P(A,0|P) = 2/5
P(A,1|P) = 1/5
P(B,0|P) = 1/5
P(B,1|P) = 1/5

(estes valores derivam de observação direta)

>> P(y3|P)

Conjunto continuo [NORMAL DISTRIBUTION]

P(y3|P) = N(y3| mean,DP^2)

$$\text{Mean} = \frac{1.2+0.8+0.5+0.9+0.8}{5} = 0.84$$

DP = [...] = sqrt(0.063) (calculated using rstudio)

>>> P(y1,y2,y3): Este valor terá o mesmo valor tanto em $P(P \mid y1,y2,y3)$ como em $P(N \mid y1,y2,y3)$ por isso não terá importância para o objetivo pedido.

Precisamos ainda de parametros semelhantes mas agora para:

P(N | y1,y2,y3) = = $\frac{P(N)*P(y1,y2,y3 \mid N)}{P(y1,y2,y3)}$

Trivial:

P(N) = 4/9

Menos simples:

>>> P(y1,y2,y3 | N)

Dado que os subconjuntos {y1,y2} e {y3} são independentes entre si podemos dizer que:

P(y1,y2,y3 | N) = P(y1,y2|N) * P(y3|N)

>> P(y1,y2|N)

Conjunto discreto [ALL COMBINATIONS]:

P(A,0|N) = 0
P(A,1|N) = 1/4
P(B,0|N) = 1/2
P(B,1|N) = 1/4

(estes valores derivam de observação direta)

>> P(y3|N)

Conjunto continuo [NORMAL DISTRIBUTION]

P(y3|N) = N(y3| mean,DP^2)

Mean = $\frac{1+0.9+1.2+0.8}{4}$ = sqrt(0.975)

DP = [...] = 0.02917 (calculated using rstudio)

>>> P(y1,y2,y3): Este valor terá o mesmo valor tanto em P(P | y1,y2,y3) como em P(N | y1,y2,y3) por isso não terá importância para o objetivo pedido

Todos os parametros que precisamos foram devidamente calculados.

Terceira pergunta:

Considering three testing observations, $\left\{\left(\begin{pmatrix}A\\1\\0.8\end{pmatrix}, \text{Positive}\right), \left(\begin{pmatrix}B\\1\\1\end{pmatrix}, \text{Positive}\right), \left(\begin{pmatrix}B\\0\\0.9\end{pmatrix}, \text{Negative}\right)\right\}$.

3) [3v] Under a MAP assumption, compute $P(\text{Positive}|\mathbf{x})$ of each testing observation.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 081

Resposta:

Geral:

$$\frac{P(P) * P(y1, y2, y3 \mid P)}{P(y1, y2, y3)}$$

Nota: Média das observações:

0.9

Desvio padrão das observações:

0.2179

Parte 1: P(y1=A,y2=1,y3=0.8 | Positive)

$$\frac{P(P)*P(y1=A\ ,y2=1\ ,y3=0.8 \mid P)}{P(y1=A,y2=1,y3=0.8)} = \frac{\left(\frac{5}{9}\right)*P(y1 = A, y2 = 1|P)*P(y3=0.8|P)}{P(y1=A,y2=1)*P(y3=0.8)} =$$

$$= \frac{\left(\frac{5}{9}\right)*\frac{1}{5}*N(0.8|0.84,sqrt(0.063))}{\frac{2}{9}*N(0.8|0.9,0.2179)} = [...] = 0.4763 \text{ (calculado usando o rstudio)}$$

Parte 2: P(y1=B,y2=1,y3=1 | Positive)

$$\frac{P(P)*P(y1=B\ ,y2=1\ ,y3=1 \mid P)}{P(y1=B,y2=1,y3=1)} = \frac{\left(\frac{5}{9}\right)*P(y1 = B, y2 = 1|P)*P(y3=1|P)}{P(y1=B,y2=1)*P(y3=1)} =$$

$$= \frac{\left(\frac{5}{9}\right)*\frac{1}{5}*N(1|0.84,sqrt(0.063))}{\frac{2}{9}*N(1|0.9,0.2179)} = [...] = 0.3937 \text{ (calculado usando o rstudio)}$$

Parte 3: P(y1=B,y2=0,y3=0.9 | Positive)

$$\frac{P(P)*P(y1=B\ ,y2=0\ ,y3=0.9 \mid P)}{P(y1=B,y2=0,y3=0.9)} = \frac{\left(\frac{5}{9}\right)*P(y1 = B, y2 = 0|P)*P(y3=0.9|P)}{P(y1=B,y2=0)*P(y3=0.9)} =$$

$$= \frac{\left(\frac{5}{9}\right)*\frac{1}{5}*N(0.9|0.84,sqrt(0.063))}{\frac{3}{9}*N(0.9|0.9,0.2179)} = [...] = 0.2813 \text{ (calculado usando o rstudio)}$$

Conclusão:

Todas as probabilidas P(Positive|x) foram calculadas com sucesso

Aprendizagem 2021/22

Homework I – Group 081

Quarta pergunta:

4) [2v] Given a binary class variable, the default decision threshold of $\theta = 0.5$,

$$f(\mathbf{x}|\theta) = \begin{cases} \text{Positive} & P(\text{Positive}|\mathbf{x}) > \theta \\ \text{Negative} & \text{otherwise} \end{cases}$$

can be adjusted. Which decision threshold – 0.3, 0.5 or 0.7 – optimizes testing accuracy?

Resposta:

Accuracy = (TP + TN) / ALL

X1 = (A,1,0.8) X2 = (B,1,1) X3 = (B,0,0.9)

Threshold of 0.3

1. P(P|X1) = 0.476, segundo este threshold dá Positivo, o seu valor também é positivo ou seja é um TP
2. P(P|X2) = 0.39, segundo este threshold dá Positivo, o seu valor também é positivo ou seja é um TP
3. P(P|X3) = 0.28, segundo este threshold dá Negativo, o seu valor também é negativo ou seja é um TN

Accucary = 3/3 = 100%

Threshold of 0.5 e 0.7

1. P(P|X1) = 0.476, segundo este threshold dá Negativo, o seu valor é positivo ou seja é um FP
2. P(P|X2) = 0.39, segundo este threshold dá Negativo, o seu valor é positivo ou seja é um FP
3. P(P|X3) = 0.28, segundo este threshold dá Nesitivo, o seu valor também é negativo ou seja é um TN

Accucary = 1/3 = 33.(3)%

Conclusão:

Neste contexto o Threshold que maximizava os nossos resultados era o de 0.3

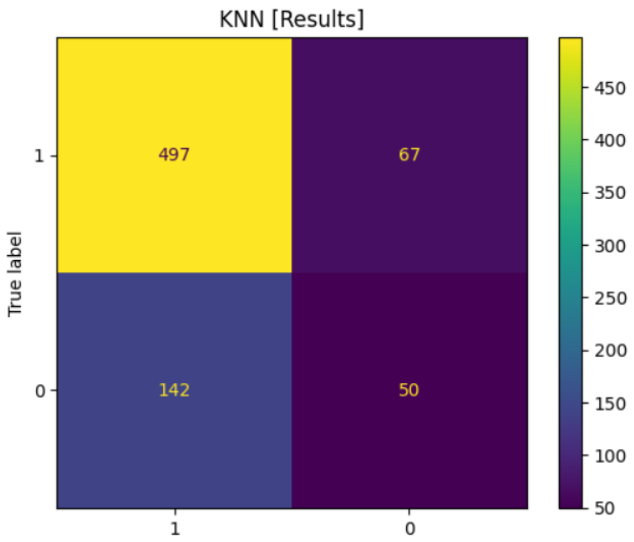
TÉCNICO LISBOA

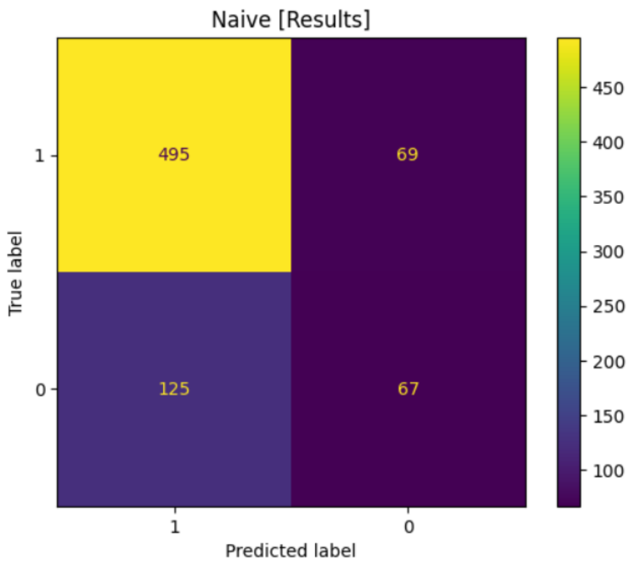
Aprendizagem 2021/22

Homework I – Group 081

II. Programming and critical analysis

5) [3v] Using `sklearn`, considering a 10-fold stratified cross validation (`random=0`), plot the cumulative testing confusion matrices of kNN (uniform weights, $k = 5$, Euclidean distance) and Naïve Bayes (Gaussian assumption). Use all remaining classifier parameters as default.





6) [2v] Using `scipy`, test the hypothesis "kNN is statistically superior to Naïve Bayes regarding accuracy", asserting whether is true.

```
0.9104476998751558
```

Usando estatistica (ttest)

Chegamos ao valor p: 0.91

0.91 > 0.05

Desta forma concluimos que a afirmação "kNN is statistically superior to Naive Bayes regarding accuracy" não pode ser demostrada usando os nossos dados.

7) [2v] Enumerate three possible reasons that could underlie the observed differences in predictive accuracy between kNN and Naïve Bayes.

As diferenças na precisão dos metodos poderá ser causada por:

1) As features poderão ser pouco dependentes favorecendo bastante a precisão do naïve bayes.

2) No geral o Naïve bayes é muito preciso quando trabalhamos com big data, o tamanho de pd_speecb.arff poderá ter interferido nos resultados.

3) O valor de k escolhido para o kNN poderá não ter sido o mais indicado para o problema

III. APPENDIX

Pergunta 5

```
from matplotlib import pyplot as plt
from scipy.io.arff import loadarff
from sklearn import metrics, tree
from sklearn.model_selection import PredefinedSplit, train_test_split
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import seaborn as sns

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

# Split in data and targets
X = df.drop('class', axis=1)
y = df['class']

# Define labels
labels = list(set(y))

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

# KNN (5,uniform,euclidean)
neigh = KNeighborsClassifier(n_neighbors=5)
df = pd.DataFrame()
df_compare = pd.DataFrame()

# NAIVE
gnb = GaussianNB()
naive_df = pd.DataFrame()



for train_index,test_index in skf.split(X,y):
    neigh.fit(X.iloc[train_index],y.iloc[train_index])
    predictor = neigh.predict(X.iloc[test_index])
    df = pd.concat([df,pd.DataFrame(predictor)])
    df_compare = pd.concat([df_compare,pd.DataFrame(y.iloc[test_index])])
```

```
    naive_df = pd.concat([naive_df,pd.DataFrame(gnb.fit(X.iloc[train_index],
y.iloc[train_index]).predict(X.iloc[test_index]))])


cm = metrics.confusion_matrix(df_compare,df,labels=labels)
cm_display =
metrics.ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=labels)
cm_display.plot()
cm_display.ax_.set_title("KNN [Results]")


cm = metrics.confusion_matrix(df_compare,naive_df,labels=labels)
cm_display =
metrics.ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=labels)
cm_display.plot()
cm_display.ax_.set_title("Naive [Results]")
```

Pergunta 6

```
import imp
from re import A
from matplotlib import pyplot as plt
from scipy.io.arff import loadarff
from sklearn import metrics, tree
from sklearn.model_selection import PredefinedSplit, train_test_split
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import GaussianNB
from scipy import stats
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import seaborn as sns

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

# Split in data and targets
X = df.drop('class', axis=1)
y = df['class']

# Define labels
labels = list(set(y))

# FOLDS: 10
```

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


# BASE
df_compare = pd.DataFrame()

# KNN (5,uniform,euclidean)
neigh = KNeighborsClassifier(n_neighbors=5)
knn_df = pd.DataFrame()

# NAIVE
gnb = GaussianNB()
naive_df = pd.DataFrame()

knn_accs = []
naive_accs = []


for train_index,test_index in skf.split(X,y):
    neigh.fit(X.iloc[train_index],y.iloc[train_index])
    predictor = neigh.predict(X.iloc[test_index])
    knn_df = pd.concat([knn_df,pd.DataFrame(predictor)])
    df_compare = pd.concat([df_compare,pd.DataFrame(y.iloc[test_index])])


    naive_predictor = gnb.fit(X.iloc[train_index],
y.iloc[train_index]).predict(X.iloc[test_index])
    naive_df = pd.concat([naive_df,pd.DataFrame(naive_predictor)])


    knn_acc = metrics.accuracy_score(y.iloc[test_index],predictor)
    naive_acc = metrics.accuracy_score(y.iloc[test_index],naive_predictor)

    knn_accs.append(knn_acc)
    naive_accs.append(naive_acc)


var = stats.ttest_rel(knn_accs,naive_accs,alternative="greater")
print(var.pvalue)

# Pvalue = 0.91
# 0.91 > 0.05 negamos a afirmação (a accuracy de knn nao é maior)
```

END