TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 21

I. Pen-and-paper

1)

- p(class=0 | x) = $\frac{p(x|class=0) \times p(class=0)}{p(x)} = \frac{p(x|class=0) \times p(class=0)}{p(x|class=0) + p(x|class=1)}$

p(x | class = 0) = p(y1 | class = 0) x p(y2 | class = 0) x p(y3,y4 | class = 0)

p(y1 | class = 0):

$(y1 \mid class=0) \sim N(\mu \mid \delta^2)$

$$\mu = \frac{\sum_{i=1}^{4} y_{1i}}{4} = \frac{0.6 + 0.1 + 0.2 + 0.1}{4} = 0.25$$

$$\delta^2 = \frac{\sum_{i=1}^{4} (y_{1i} - \mu)^2}{4-1} = \frac{(0.6-0.25)^2 + (0.1-0.25)^2 + ...}{3} = 0.0567$$

$(y1 \mid class = 0) \sim N(0.25, 0.0567)$

p(y2 | class = 0):

$$p(y2 = A \mid class = 0) = \frac{2}{4} = 0.5$$

$$p(y2 = B \mid class = 0) = \frac{1}{4} = 0.25$$

$$p(y2 = C \mid class = 0) = \frac{1}{4} = 0.25$$

p(y3,y4 | class = 0):

$(y3,y4 \mid class = 0) \sim N(\mu, \Sigma)$

$$\mu_3 = \frac{\sum_{i=1}^{4} y_{3i}}{4} = \frac{0.2 - 0.1 - 0.1 + 0.8}{4} = 0.2$$

$$\mu_4 = \frac{\sum_{i=1}^{4} y_{4i}}{4} = \frac{0.4 - 0.4 - 0.4 + 0.8}{4} = 0.25$$

$$\delta^2_3 = \frac{\sum_{i=1}^{4} (y_{3i} - \mu)^2}{4-1} = \frac{(0.2-0.2)^2 + (-0.1-0.2)^2 + ...}{3} = 0.18$$

$$\delta^2_4 = \frac{\sum_{i=1}^{4} (y_{4i} - \mu)^2}{4-1} = \frac{(0.4-0.25)^2 + (-0.4-0.25)^2 + ...}{3} = 0.25$$

$$cov(y3, y4) = \frac{\sum_{i=1}^{4} (y_{3i} - \mu_{3i}) \times (y_{4i} - \mu_{4i})}{4-1} = \frac{(0.2-0.2) \times (0.4-0.25) + ...}{3} = 0.18$$

$$p(class = 0) = \frac{2}{5}$$

- p(class=1 | x) = $\frac{p(x|class=1) \times p(class=1)}{p(x)} = \frac{p(x|class=1) \times p(class=1)}{p(x|class=0) + p(x|class=1)}$

p(x | class = 1) = p(y1 | class = 1) x p(y2 | class = 1) x p(y3,y4 | class = 1)

p(y1 | class = 1):

$(y1 \mid class=1) \sim N(\mu \mid \delta^2)$

$$\mu = \frac{\sum_{i=5}^{10} y_{1i}}{6} = \frac{0.3 - 0.1 - 0.3 + 0.2 + 0.4 - 0.2}{6} = 0.05$$

$$\delta^2 = \frac{\sum_{i=5}^{10} (y_{1i} - \mu)^2}{6-1} = \frac{(0.3 - 0.05)^2 + (-0.1 - 0.05)^2 + \ldots}{5} = 0.083$$

$(y1 \mid class = 1) \sim N(0.05, 0.083)$

p(y2 | class = 1):

$$p(y2 = A \mid class = 1) = \frac{1}{6} = 0.17$$

$$p(y2 = B \mid class = 1) = \frac{2}{6} = 0.3(3)$$

$$p(y2 = C \mid class = 1) = \frac{3}{6} = 0.5$$

p(y3,y4 | class = 1):

$(y3,y4 \mid class = 1) \sim N(\mu, \Sigma)$

$$\mu_3 = \frac{\sum_{i=5}^{10} y_{3i}}{6} = \frac{0.1 + 0.2 + \ldots + 0.4}{6} = 0.1166$$

$$\mu_4 = \frac{\sum_{i=5}^{10} y_{4i}}{6} = \frac{0.3 - 0.2 + \ldots + 0.3}{6} = 0.083$$

$$\delta^2{}_3 = \frac{\sum_{i=5}^{10} (y_{3i} - \mu)^2}{6-1} = \frac{(0.1 - 0.1166)^2 + (0.2 - 0.1166)^2 + \ldots}{5} = 0.1097$$

$$\delta^2{}_4 = \frac{\sum_{i=5}^{10} (y_{4i} - \mu)^2}{6-1} = \frac{(0.3 - 0.083)^2 + (-0.2 - 0.083)^2 + \ldots}{5} = 0.214$$

$$cov(y3, y4) = \frac{\sum_{i=5}^{10} (y_{3i} - \mu_{3i}) \times (y_{4i} - \mu_{4i})}{6-1} = \frac{(0.1 - 0.1166) \times (0.3 - 0.083) + \ldots}{5} = 0.122$$

$$p(class = 1) = \frac{6}{10}$$

2)

p(class = 0 | x1) = 0.83502	p(class = 1 | x1) = 0.16498
p(class = 0 | x2) = 0.19507	p(class = 1 | x2) = 0.80493
p(class = 0 | x3) = 0.75914	p(class = 1 | x3) = 0.24086
p(class = 0 | x4) = 0.45872	p(class = 1 | x4) = 0.54128
p(class = 0 | x5) = 0.45625	p(class = 1 | x5) = 0.54375
p(class = 0 | x6) = 0.07245	p(class = 1 | x6) = 0.92755
p(class = 0 | x7) = 0.06366	p(class = 1 | x7) = 0.93634

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 21

p(class = 0 | x8) = 0.46651

p(class = 0 | x9) = 0.69934

p(class = 0 | x10) = 0.08637

p(class = 1 | x8) = 0.53349

p(class = 1 | x9) = 0.30066

p(class = 1 | x10) = 0.91362

Pred/Act	Class = 1	Class = 0
Class = 1	5	2
Class = 0	1	2

3) Precision $= \frac{TP}{TP+FP} = \frac{5}{6}$; Recall $= \frac{TP}{TP+FN} = \frac{5}{7}$; $\frac{1}{F1} = \frac{1}{2} \text{x} \left(\frac{1}{Precision} + \frac{1}{Recall}\right) \Leftrightarrow$ F1 = 0.76923

4) A: threshold = 0.3 when class = 1.

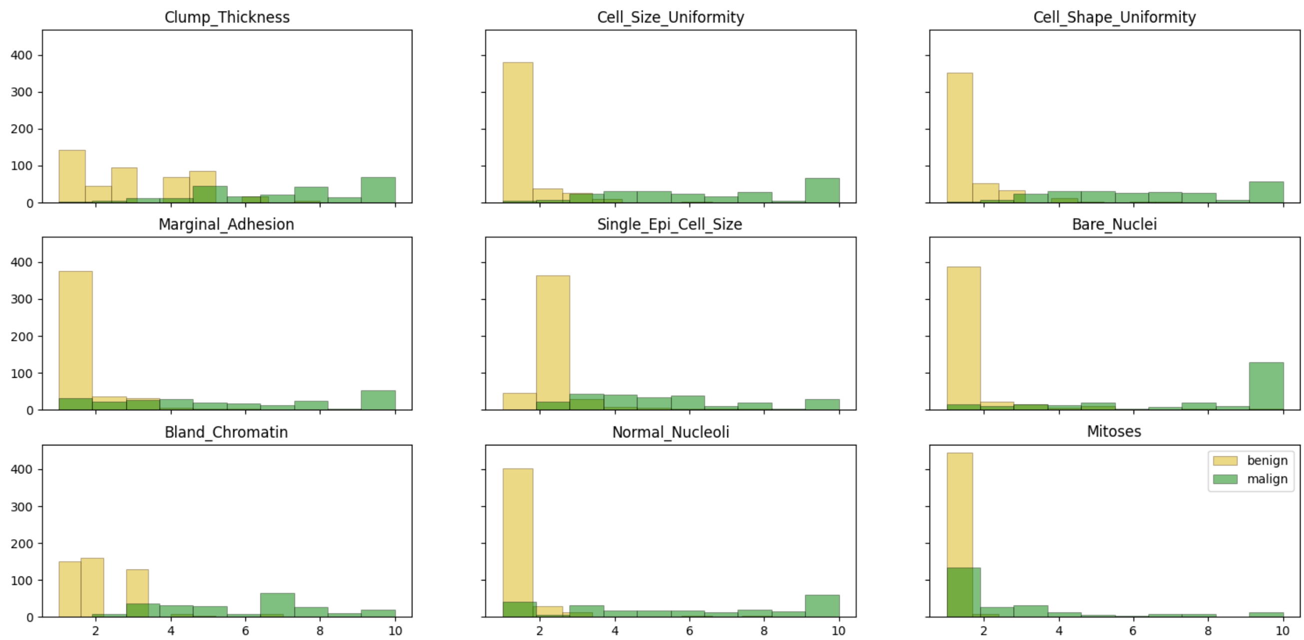
class =0	0.2	0.3	0.4	0.5	0.6	0.7	0.8
x1	1	1	1	1	1	1	1
x2	0	0	0	0	0	0	0
x3	1	1	1	1	1	1	0
x4	1	1	1	0	0	0	0
x5	1	1	1	0	0	0	0
x6	0	0	0	0	0	0	0
x7	0	0	0	0	0	0	0
x8	1	1	1	0	0	0	0
x9	1	1	1	1	1	0	0
x10	0	0	0	0	0	0	0
	$\frac{4}{10}$	$\frac{4}{10}$	$\frac{4}{10}$	$\frac{3}{10}$	$\frac{3}{10}$	$\frac{2}{10}$	$\frac{3}{10}$

class =1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
x1	0	0	0	0	0	0	0
x2	1	1	1	1	1	1	1
x3	1	0	0	0	0	0	0
x4	1	1	1	1	0	0	0
x5	1	1	1	1	0	0	0
x6	1	1	1	1	1	1	1
x7	1	1	1	1	1	1	1
x8	1	1	1	1	0	0	0
x9	1	1	0	0	0	0	0
x10	1	1	1	1	1	1	1
	$\frac{7}{10}$	$\frac{8}{10}$	$\frac{7}{10}$	$\frac{7}{10}$	$\frac{6}{10}$	$\frac{6}{10}$	$\frac{6}{10}$

Aprendizagem 2021/22
Homework I – Group 21

II. Programming and critical analysis

5)



6) [0.9692242114237001, 0.9736359761295823, 0.9706734867860188]

The K less sucesptible is k=5. Where its associated value is bigger than the others.

7) Pvalue = 9.91356e-06. Since the pvalue is 9.91356e-06, we must reject the null hypothesis (equal averages). Therefore, we can confirm the thesis that "“kNN is statistically superior to Naïve Bayes” 9,91356e-06.

8) Knn is better on finding similarity between observations; Knn is most likely to overfit. The decisions you get with K-NN are much more complex (also slower) compared to Naive Bayes, which lead to better results in terms of precision.

III. APPENDIX

Used Imports for exercises:

```
 from scipy.io import arff                 |  from sklearn.neighbors import KNeighborsClassifier
 import numpy as np                        |  from sklearn.metrics import accuracy_score
 import pandas as pd                       |  from sklearn.model_selection import KFold
 import matplotlib.pyplot as plt           |  from sklearn.naive_bayes import MultinomialNB
 from matplotlib import pyplot             |  from scipy import stats
```

Common Code:

```
 data = arff.loadarff('breast.w.arff') |  benign=df.loc[df['Class']== b'benign']
 df = pd.DataFrame(data[0])            |  malign=df.loc[df['Class']== b'malignant']
 df.head()                             |  X = df.iloc[:, :-1].values
 df = df.dropna()                      |  y = df.iloc[:, 9].values
```

5)

```
number_cols = ['Clump_Thickness','Cell_Size_Uniformity','Cell_Shape_Uniformity','Marginal_Adhesion',
'Single_Epi_Cell_Size', 'Bare_Nuclei','Bland_Chromatin','Normal_Nucleoli', 'Mitoses']
fig, ax = plt.subplots(3, 3, sharex=True, sharey=True)
for i in range(3):
    for j in range(3):
        ax[i,j].set_title(number_cols[i*3+j])
        column_name= number_cols[i*3+j]
        ax[i,j].hist(benign[column_name],label='benign', alpha=0.5,linewidth=0.5,
color='#DBB40C',edgecolor='#3d1c02')
        ax[i,j].hist(malign[column_name],label='malign',
alpha=0.5,linewidth=0.5,color='#008000',edgecolor='#000000')
pyplot.legend(loc='upper right')
plt.show()
```

6)

```
df['Class'] = df['Class'].replace([b'benign'],'0')      #common for exercise 6 and 7
df['Class'] = df['Class'].replace([b'malignant'],'1')   #common for exercise 6 and 7
list=[3,5,7]
lista=[]
for k in list:
    soma=0
    knn = KNeighborsClassifier(k)
    kf = KFold(n_splits=10,random_state=21,shuffle=True)
    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]
        y_train = y_train.astype('int')
        y_test = y_test.astype('int')
        knn.fit(X_train, y_train)
        pred = knn.predict(X_test)
        soma+=accuracy_score(pred,y_test)
    numero=soma/10
    lista.append(numero)
print(lista)
```

7)

```
knn = KNeighborsClassifier(3)
lista1=[]
lista2 = []
kf = KFold(n_splits=10,random_state=21,shuffle=True)
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]
    y_train = y_train.astype('int')
    y_test = y_test.astype('int')
    knn.fit(X_train, y_train)
    lista1.append(accuracy_score(knn.predict(X_test),y_test))
    clf = MultinomialNB().fit(X_train, y_train)
    lista2.append(accuracy_score(clf.predict(X_test),y_test))
print(stats.ttest_rel(lista1, lista2))
```