Homework I – Group 047

I. Pen-and-paper

1)

	Class = 0	Class = 1
	$P(Class = 0) = \frac{4}{10} = 0.4$	$P(Class = 1) = \frac{6}{10} = 0.6$
y1	$\mu = \frac{1}{4} \times \sum_{i=1}^{4} x_i = 0.25$ $\sigma = \sqrt{\frac{1}{4} \times \sum_{i=1}^{4} (x_i - \mu)^2} = 0.2380$ $P(y1 \mid Class = 0) = N(\mu, \sigma^2)$	$\mu = \frac{1}{6} \times \sum_{i=5}^{10} x_i = 0.05$ $\sigma = \sqrt{\frac{1}{6} \times \sum_{i=5}^{10} (x_i - \mu)^2} = 0.2881$ $P(y1 \mid Class = 1) = N(\mu, \sigma^2)$
y2	$P(A \mid Class = 0) = \frac{2}{4} = 0.5$ $P(B \mid Class = 0) = \frac{1}{4} = 0.25$ $P(C \mid Class = 0) = \frac{1}{4} = 0.25$	$P(A \mid Class = 1) = \frac{1}{6} = 0.1667$ $P(B \mid Class = 1) = \frac{2}{6} = 0.3333$ $P(C \mid Class = 1) = \frac{3}{6} = 0.5$
y3/y4	$\mu = \frac{1}{4} \times \sum_{i=1}^{4} [x_{iy3}\ x_{iy4}] = [0.2\ 0.25]$ $\Sigma = \begin{bmatrix} cov(y_3, y_3) & cov(y_3, y_4) \\ cov(y_4, y_3) & cov(y_4, y_4) \end{bmatrix} = \begin{bmatrix} 0.1800 & 0.1800 \\ 0.1800 & 0.2500 \end{bmatrix}$ $\lvert\Sigma\rvert = cov(y_3, y_3) \times cov(y_3, y_4) - cov(y_4, y_3) \times cov(y_4, y_4) = 0.0126$ $\Sigma^{-1} = \begin{bmatrix} \frac{cov(y_4, y_4)}{\lvert\Sigma\rvert} & -\frac{cov(y_3, y_4)}{\lvert\Sigma\rvert} \\ -\frac{cov(y_4, y_3)}{\lvert\Sigma\rvert} & \frac{cov(y_3, y_3)}{\lvert\Sigma\rvert} \end{bmatrix} = \begin{bmatrix} 19.8413 & -14.2857 \\ -14.2857 & 14.2857 \end{bmatrix}$ $P(y3, y4 \mid Class = 0) = N(\mu, \Sigma)$	$\mu = \frac{1}{6} \times \sum_{i=5}^{10} [x_{iy3}\ x_{iy4}] = [0.1167\ 0.0833]$ $\Sigma = \begin{bmatrix} cov(y_3, y_3) & cov(y_3, y_4) \\ cov(y_4, y_3) & cov(y_4, y_4) \end{bmatrix} = \begin{bmatrix} 0.1097 & 0.1223 \\ 0.1223 & 0.2137 \end{bmatrix}$ $\lvert\Sigma\rvert = cov(y_3, y_3) \times cov(y_3, y_4) - cov(y_4, y_3) \times cov(y_4, y_4) = 0.0085$ $\Sigma^{-1} = \begin{bmatrix} \frac{cov(y_4, y_4)}{\lvert\Sigma\rvert} & -\frac{cov(y_3, y_4)}{\lvert\Sigma\rvert} \\ -\frac{cov(y_4, y_3)}{\lvert\Sigma\rvert} & \frac{cov(y_3, y_3)}{\lvert\Sigma\rvert} \end{bmatrix} = \begin{bmatrix} 25.2362 & -14.4488 \\ -14.4488 & 12.9528 \end{bmatrix}$ $P(y3, y4 \mid Class = 1) = N(\mu, \Sigma)$

2) $P(Class = c \mid x_i) = P(Class = c) \times P(y1_{x_i} \mid Class = c) \times P(y2_{x_i} \mid Class = c) \times P(y3_{x_i}, y4_{x_i} \mid Class = c), c = 0 \cup 1$

	Class(0)	Class(1)	
x1	$P(y1 = 0.6 \mid Class = 0) = 0.5686$ $P(y2 = A \mid Class = 0) = 0.5$ $P(y3 = 0.2, y4 = 0.4 \mid Class = 0) = 1.2074$ $P(Class = 0 \mid x1) = 0.13731$	$P(Y1 = 0.6 \mid Class = 1) = 0.2239$ $P(Y2 = A \mid Class = 1) = 0.1667$ $P(Y3 = 0.2, Y4 = 0.4 \mid Class = 1) = 1.2109$ $P(Class = 1 \mid x1) = 0.02711$	*0*
x2	$P(y1 = 0.1 \mid Class = 0) = 1.3741$ $P(y2 = B \mid Class = 0) = 0.25$ $P(y3 = -0.1, y4 = -0.4 \mid Class = 0) = 0.4603$ $P(Class = 0 \mid x2) = 0.06325$	$P(Y1 = 0.1 \mid Class = 1) = 1.3641$ $P(Y2 = B \mid Class = 1) = 0.3333$ $P(Y3 = -0.1, Y4 = -0.4 \mid Class = 1) = 0.9561$ $P(Class = 1 \mid x2) = 0.26082$	*1*

Homework I – Group 047

x3	$P(y1=0.2\|Class=0)=1.6393$ $P(y2=A\|Class=0)=0.5$ $P(y3=-0.1, y4=0.2 \mid Class=0)=0.7066$ $P(Class=0\|x3)=0.23167$	$P(Y1=0.2\|Class=1)=1.2092$ $P(Y2=A\|Class=1)=0.1667$ $P(Y3=-0.1, Y4=0.2\|Class=1)=0.6084$ $P(Class=1\|x3)=0.07356$	*0*
x4	$P(y1=0.1\|Class=0)=1.3741$ $P(y2=C\|Class=0)=0.25$ $P(y3=0.8, y4=0.8 \mid Class=0)=0.5124$ $P(Class=0\|x4)=0.07041$	$P(Y1=0.1\|Class=1)=1.3641$ $P(Y2=C\|Class=1)=0.5$ $P(Y3=0.8, Y4=0.8\|Class=1)=0.2030$ $P(Class=1\|x4)=0.08308$	*1*
x5	$P(y1=0.3\|Class=0)=1.6393$ $P(y2=B\|Class=0)=0.25$ $P(y3=0.1, y4=0.3 \mid Class=0)=1.1743$ $P(Class=0\|x5)=0.19250$	$P(Y1=0.3\|Class=1)=0.9503$ $P(Y2=B\|Class=1)=0.3333$ $P(Y3=0.1, Y4=0.3\|Class=1)=1.2064$ $P(Class=1\|x5)=0.22926$	*1*
x6	$P(y1=-0.1\|Class=0)=0.5686$ $P(y2=C\|Class=0)=0.25$ $P(y3=0.2, y4=-0.2 \mid Class=0)=0.3338$ $P(Class=0\|x6)=0.01898$	$P(Y1=-0.1\|Class=1)=1.2092$ $P(Y2=C\|Class=1)=0.5$ $P(Y3=0.2, Y4=-0.2\|Class=1)=0.6707$ $P(Class=1\|x6)=0.24330$	*1*
x7	$P(y1=-0.3\|Class=0)=0.1162$ $P(y2=C\|Class=0)=0.25$ $P(y3=-0.1, y4=0.2 \mid Class=0)=0.7066$ $P(Class=0\|x7)=0.00821$	$P(Y1=-0.3\|Class=1)=0.6620$ $P(Y2=C\|Class=1)=0.5$ $P(Y3=-0.1, Y4=0.2\|Class=1)=0.6084$ $P(Class=1\|x7)=0.12083$	*1*
x8	$P(y1=0.2\|Class=0)=1.6393$ $P(y2=B\|Class=0)=0.25$ $P(y3=0.5, y4=0.6 \mid Class=0)=1.0847$ $P(Class=0\|x8)=0.17782$	$P(Y1=0.2\|Class=1)=1.2092$ $P(Y2=B\|Class=1)=0.3333$ $P(Y3=0.5, Y4=0.6\|Class=1)=0.8399$ $P(Class=1\|x8)=0.20311$	*1*
x9	$P(y1=0.4\|Class=0)=1.3741$ $P(y2=A\|Class=0)=0.5$ $P(y3=-0.4, y4=-0.7 \mid Class=0)=0.2174$ $P(Class=0\|x9)=0.05976$	$P(Y1=0.4\|Class=1)=0.6620$ $P(Y2=A\|Class=1)=0.1667$ $P(Y3=-0.4, Y4=-0.7\|Class=1)=0.3876$ $P(Class=1\|x9)=0.02566$	*0*
x10	$P(y1=-0.2\|Class=0)=0.2807$ $P(y2=C\|Class=0)=0.25$ $P(y3=0.4, y4=0.3\|Class=0)=1.0804$ $P(Class=0\|x10)=0.03033$	$P(Y1=-0.2\|Class=1)=0.9503$ $P(Y2=C\|Class=1)=0.5$ $P(Y3=0.4, Y4=0.3\|Class=1)=1.1247$ $P(Class=1\|x10)=0.32062$	*1*

		Predicted	
		0	1
Real	0	2	2
	1	1	5

$$Precision(P)=\frac{TP}{TP+FP}=\frac{5}{5+2}=0.7143$$

$$Recall(R)=\frac{TP}{TP+FN}=\frac{5}{5+1}=0.8333$$

3) $F1=\frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R}=\frac{2\times 0.7143\times 0.8333}{0.7143+0.8333}=0.7692$

4) $P(Class=c\mid x_i)=\frac{P(Class=c\mid x_i)}{P(Class=0\mid x_i)+P(Class=1\mid x_i)}, c=0\cup 1$

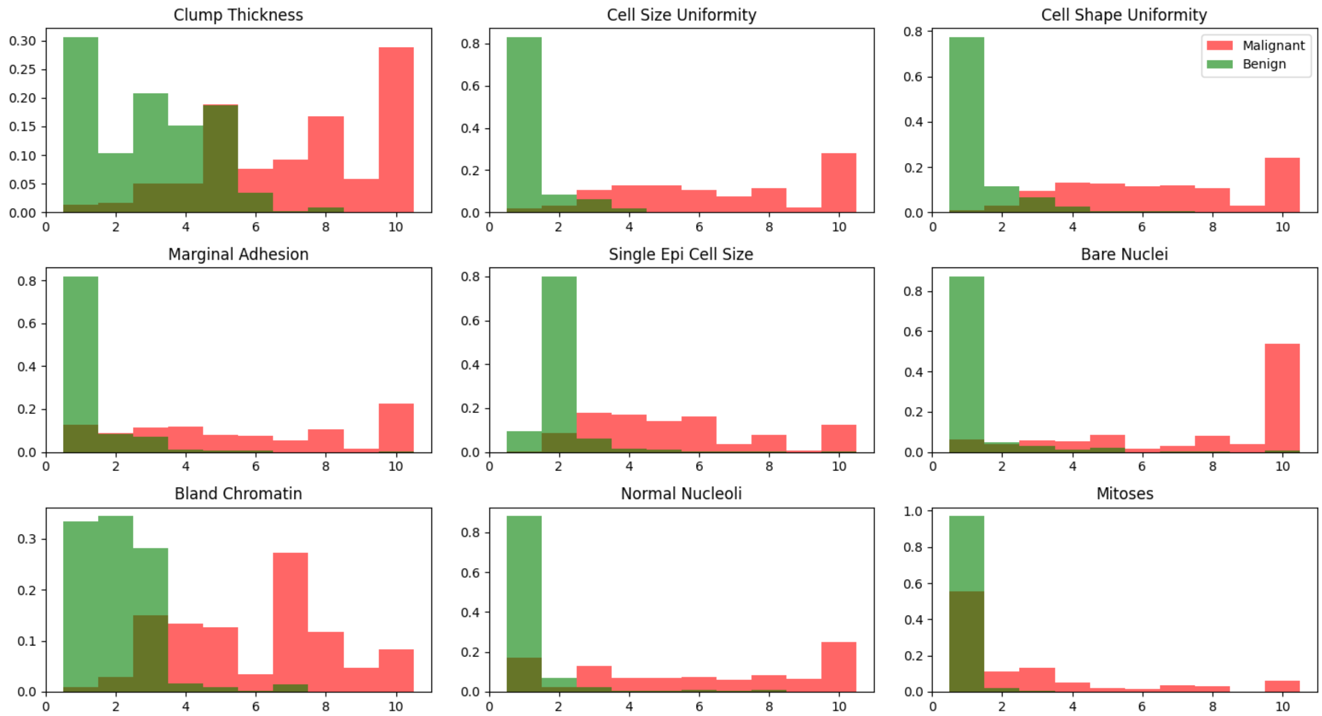
Probabilities	Real Class	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$P(Class=0\|x1)=0.8351$	0	0	0	0	0	0	0	1
$P(Class=0\|x2)=0.1952$	0	1	1	1	1	1	1	1
$P(Class=0\|x3)=0.7590$	0	0	0	0	0	0	1	1

$P(Class = 0 \mid x4) = 0.4587$	0	0	0	1	1	1	1	1
$P(Class = 0 \mid x5) = 0.4564$	1	0	0	1	1	1	1	1
$P(Class = 0 \mid x6) = 0.0724$	1	1	1	1	1	1	1	1
$P(Class = 0 \mid x7) = 0.0636$	1	1	1	1	1	1	1	1
$P(Class = 0 \mid x8) = 0.4668$	1	0	0	1	1	1	1	1
$P(Class = 0 \mid x9) = 0.6996$	1	0	0	0	0	1	1	1
$P(Class = 0 \mid x10) = 0.0864$	1	1	1	1	1	1	1	1
Threshold	**-**	**0.6**	**0.6**	**0.7**	**0.7**	**0.8**	**0.7**	**0.6**

From the table we can identify that the decision probability threshold that optimizes training accuracy is 0,7. This means that we can classify xi as being of Class 0 if $P(Class = 0 \mid xi) \geq 0{,}7$, or Class 1 otherwise.

II. Programming and critical analysis

5)



6) From this data we can see that K=5 has better accuracy, and therefore is less susceptible to overfitting.

K	*3*	*5*	*7*
Accuracy	0.9692668371696506	0.9721867007672635	0.9707161125319693

7) The hypotheses H0, “kNN is statistically inferior or equal to Naïve Bayes (multinomial assumption)”, returned a P-Value of 0.0003537432054576055. From this we can safely reject H0 for a significance level of 0.0004 and accept H1,” kNN is statistically superior to Naïve Bayes (multinomial assumption)”.

8) Two reasons that explain the difference in performance between kNN and Naïve Bayes are:

1. Naïve Bayes assumes that every variable is independent, but looking at **7)**, that doesn’t seem to be the case;
2. Not enough data to train Naïve Bayes enough to correctly predict the class.

III. APPENDIX

```
import matplotlib.pyplot as plt
import numpy as num
from scipy.io import arff
from scipy.stats import ttest_rel
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold, cross_val_score
from sklearn.naive_bayes import MultinomialNB

file = open("breast.w.arff", "r")
data, meta = arff.loadarff(file)
#5)
bins = [1,2,3,4,5,6,7,8,9,10,11]
for features, i in zip(meta.names()[:-1], range(1,10)):
    plt.subplot(3,3,i)
    dh = [data[(data["Class"]==b'benign')][features], data[(data["Class"]==b'malignant')][features]]
    plt.hist(dh, bins=bins, align='left', color=['g','r'], label=['Benign','Malignant'], alpha=0.6,
histtype='stepfilled', density=True)
    if i==3: plt.legend()
    plt.title(features.replace("_"," "))
plt.show()
#6)
input = data[meta.names()[:-1]].tolist()
output = data["Class"].tolist()
kFol = KFold(n_splits=10, shuffle=True, random_state=47)
crossKNN=[[],[],[]]
for i,k in zip(range(3,8,2),range(3)):
    classifier = KNeighborsClassifier(n_neighbors=i weights='uniform', metric='euclidean')
    crossKNN[k] = cross_val_score(classifier, input, output, scoring='accuracy', cv = kFol)
    kErr = num.average(crossKNN[k])
    print("Accuracy K={}: {}".format(i,kErr))
#7)
classifier = MultinomialNB()
crossNB = cross_val_score(classifier, input, output, scoring='accuracy', cv = kFol)
pval = ttest_rel(crossKNN[0], crossNB, alternative='greater').pvalue
print("p-value:",pval)
```

END