TÉCNICO LISBOA

Aprendizagem 2021/22

**Homework I – Group 114**

# I. Pen-and-paper

## 1)

1) Assumindo as variáveis/conjuntos de variáveis independentes, podemos utilizar um classificador "Naive Bayes" para responder ao problema.

$$class = \underset{k \in \{0,1\}}{\operatorname{argmax}} \; p(class_k) \prod_{i=1}^{n} P(Y_i \mid class_k)$$

- $P(Y_1 = v_1 \mid class = 0)$: $Y_1$ é distribuída normalmente ($N(\mu, \sigma^2)$)

$$\bar{x} = \mu = \frac{1}{N} \sum_{i=1}^{n} x_i = \frac{1}{4} \cdot (0,6 + 0,1 + 0,2 + 0,1) = 0,25$$

$$\sigma_{sample} = \sqrt{\frac{1}{n-1} \sum_{i=1} (x_i - \bar{x})^2} \approx 0,238$$

- $P(Y_1 = v_1 \mid class = 1)$: $(N(\mu, \sigma^2))$

$\mu = 0,05$ $\quad \sigma_{sample} = 0,288$

- $P(Y_2 = A \mid class = 0) = \frac{1}{2}$ ; $P(Y_2 = A \mid class = 1) = \frac{1}{6}$
- $P(Y_2 = B \mid class = 0) = \frac{1}{4}$ ; $P(Y_2 = B \mid class = 1) = \frac{2}{6} = \frac{1}{3}$
- $P(Y_2 = C \mid class = 0) = \frac{1}{4}$ ; $P(Y_2 = C \mid class = 1) = \frac{3}{6} = \frac{1}{2}$
- $P(Y_3 = v_3 ; Y_4 = v_4 \mid class = 0)$: $\{Y_3, Y_4\}$ tem distribuição Normal multivariada (2D) $N(\underline{\mu}, \Sigma)$

$$\underline{\mu} = [0,2 \; ; 0,25]^T = [\mu_1 ; \mu_2]^T = [\bar{x}_i \; ; \bar{x}_j]$$

$$\Sigma_{ij} = c_{ij} = \frac{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{N \in N - 1} = \begin{bmatrix} 0.18 & 0.18 \\ 0.18 & 0.25 \end{bmatrix}$$

- $P(Y_3 = v_3 ; Y_4 = v_4 \mid class = 1)$: $N(\underline{\mu}, \Sigma)$

$$\underline{\mu} = \left[\frac{7}{60} \; ; \frac{1}{12}\right]^T = [\mu_1 ; \mu_2]^T = [\bar{x}_i \; , \bar{x}_j]$$

$$\Sigma = \begin{bmatrix} 0,1097 & 0,1223 \\ 0,1223 & 0,2137 \end{bmatrix}$$

Fórmulas usadas nas distribuições normais:

Univariável: $\frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{1}{2 \cdot \sigma^2} (x - \mu)^2\right)$

2 variáveis: $\frac{1}{2\pi} \frac{1}{|\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \cdot (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu})\right)$

Quando queremos classificar um novo evento, por exemplo, $x_{new}$, calculamos a sua probabilidade condicionada à classificação = 0 e à classificação = 1. A probabilidade superior dará a sua respetiva classificação respetivamente, '0' ou '1'.

## 2) Answer 2

2) Usando ferramentas computacionais obtivemos os seguintes valores para os eventos $X_1, \ldots, X_{10}$

| | |
|---|---|
| $P(X_1 \mid class = 0) = 0,067\ldots$ | $P(X_1 \mid class = 1) = 0,017\ldots$ |
| $P(X_2 \mid class = 0) = 0,028\ldots$ | $P(X_2 \mid class = 1) = 0,170\ldots$ |
| $P(X_3 \mid class = 0) = 0,103\ldots$ | $P(X_3 \mid class = 1) = 0,048\ldots$ |
| $P(X_4 \mid class = 0) = 0,037\ldots$ | $P(X_4 \mid class = 1) = 0,0542\ldots$ |
| $P(X_5 \mid class = 0) = 0,085\ldots$ | $P(X_5 \mid class = 1) = 0,149\ldots$ |
| $P(X_6 \mid class = 0) = 0,008\ldots$ | $P(X_6 \mid class = 1) = 0,158\ldots$ |
| $P(X_7 \mid class = 0) = 0,003\ldots$ | $P(X_7 \mid class = 1) = 0,078\ldots$ |
| $P(X_8 \mid class = 0) = 0,079\ldots$ | $P(X_8 \mid class = 1) = 0,132\ldots$ |
| $P(X_9 \mid class = 0) = 0,026\ldots$ | $P(X_9 \mid class = 1) = 0,016\ldots$ |
| $P(X_{10} \mid class = 0) = 0,0735\ldots$ | $P(X_{10} \mid class = 1) = 0,209\ldots$ |

$\hat{y} = [0, 1, 0, 1, 1, 1, 1, 1, 0, 1]$ = predicted

$y = observed = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]$

| $y$ \ $\hat{y}$ | 1 | 0 |
|---|---|---|
| 1 | 5 | 1 |
| 0 | 2 | 2 |

TÉCNICO LISBOA

Aprendizagem 2021/22

**Homework I – Group 114**

**3)** Answer 3

**4)** Answer 4

3) F1:

$$\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$

$P$ = Precision $= \frac{TP}{TP+FP} = \frac{5}{5+2} = \frac{5}{7}$

$R$ = Recall $= \frac{TP}{P} = \frac{TP}{TP+FN} = \frac{5}{5+1} = \frac{5}{6}$

$$\frac{1}{F} = \frac{1}{2}\left(\frac{1}{\frac{5}{7}} + \frac{1}{\frac{5}{6}}\right) \Leftrightarrow \frac{1}{F} = \frac{1}{2}\left(\frac{7}{5} + \frac{6}{5}\right) \Leftrightarrow$$

$$\Leftrightarrow \frac{1}{F} = \frac{1}{2}\cdot\frac{13}{5} \Leftrightarrow \frac{1}{F} = \frac{13}{10} \Leftrightarrow F = \frac{10}{13} \tilde{\nu} 77\%$$

4) Usamos o ROC para analisar o nosso "Bayesian Classifier" e obtivemos o seguinte gráfico onde podemos observar que o "decision treshold" que otimiza o treino é no ponto (0.5; 0.83) pois assim temos um "True positive rate" elevado face ao "False Positive rate". A partir desse ponto o "False Positive rate" aumenta bastante face ao "True Positive Rate". O que vai piorar os resultados, pois iremos ter mais falsos positivos

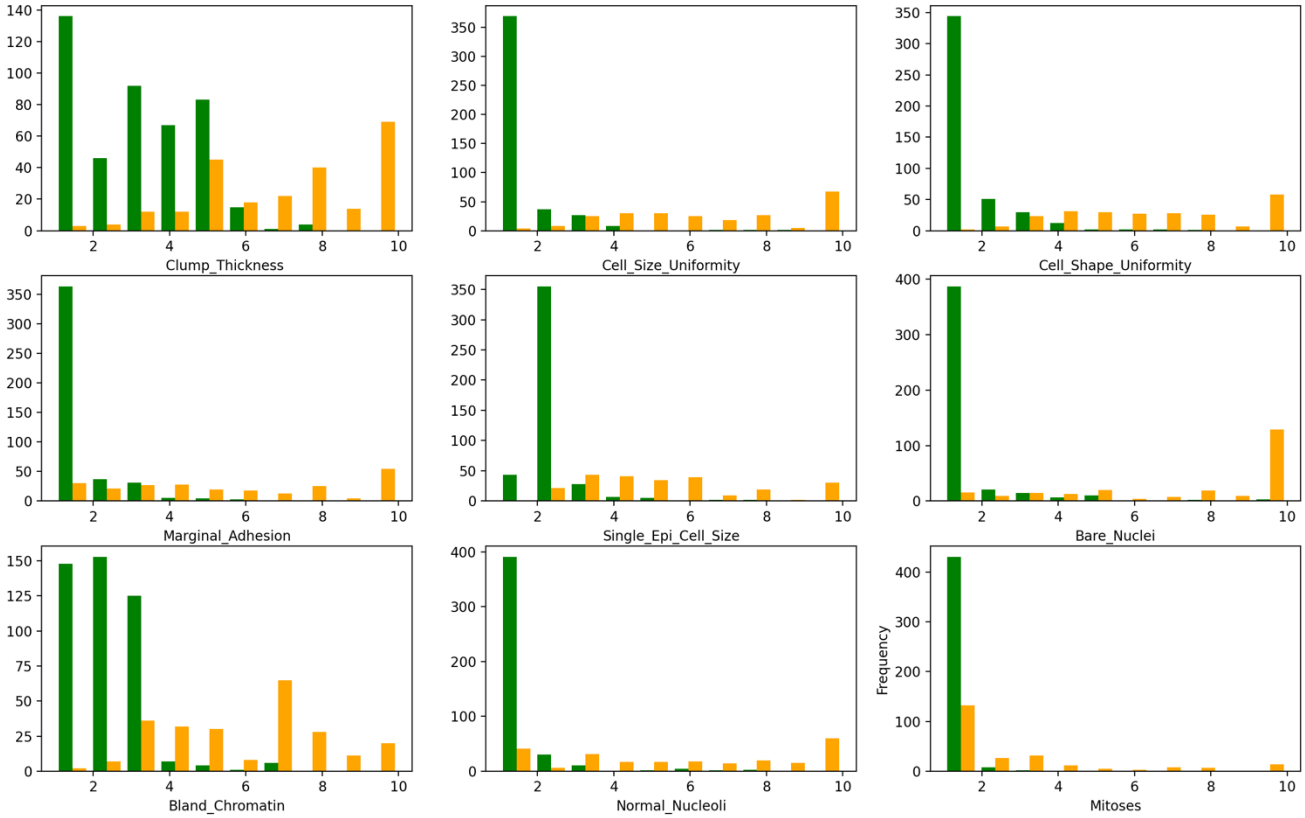
TÉCNICO LISBOA

Aprendizagem 2021/22

**Homework I – Group 114**

# II. Programming and critical analysis

## 5) Answer 5




Color Green = Benign Yellow = malignant

## 6) Answer 6

When k = 3 Distance Accuracy train : 0.9746 test : 0.9883
Uniform Accuracy train : 0.9785 test : 0.9825
When k = 5 Distance Accuracy train : 0.9746 test : 0.9649
Uniform Accuracy train : 0.9844 test : 0.9649
When k = 7 Distance Accuracy train : 0.9785 test : 0.9649
Uniforme Accuracy train : 0.9805 test : 0.9825

A melhor tentativa foi usando k = 3 através da distância Euclidiana. O overfitting é mais provável acontecer quando o conjunto de treino tem melhor rating que o conjunto de teste. Para os K = 3 é possível verificar que os score do teste > score de treino, sendo benéfico para a amostra, reduzindo o risco de Overfitting. Por outro lado quanto mais Accuracy, mais detalhada vai ser a regressão, o que não é ótimo para a amostra de dados(eventual perda de amostras significativa). Logo quanto menor for o Accuracy melhor será a regressão.

## 7) Answer 7

Multinominal NB score = 0.9048
K = 3 = 0.9746
A hipótese é verificável para esta experiência.

## 8) Answer 8

O classificador kNN em relação ao classificador de Naives Bayes é o mais fiável para a experiência. Quanto mais a taxa de acerto se aproxima de 1, mais detalhada vai ser a curva de regressão, o que não é fiável para os resultados, levando para situações de overfitting. Ainda assim o classificador kNN é o mais indicado para amostras com muitos atributos.

# III. APPENDIX

```
import numpy as np
import pandas as pd
from scipy.io.arff import loadarff
import codecs
import matplotlib.pyplot as plt
from scipy.io.arff.arffread import print_attribute
from sklearn.neighbors import DistanceMetric
from numpy.random.mtrand import RandomState, random_sample
from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import neighbors, datasets, preprocessing
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import MultinomialNB

if __name__ == '__main__':
    raw_data = loadarff('breast.w.arff')
    df_data = pd.DataFrame(raw_data[0])
    classe = df_data.pop('Class')
    df_data = df_data.astype(int)
    Y = classe.str.decode('utf-8')
def freq(n):  #getting frequency by symbol for benign and malignant
    a = []
    b = []
    for val in range(0,683):#iterating columns
        c = df_data.iloc[val,n]
        if Y[val] == 'benign':
            a.append(c)
        else:
            b.append(c)
    return [a,b]
fig, axs = plt.subplots(3,3)
names = ['Clump_Thickness '
,'Cell_Size_Uniformity','Cell_Shape_Uniformity','Marginal_Adhesion','Single_Epi_Cell_Size','Bare_Nuc
lei','Bland_Chromatin','Normal_Nucleoli','Mitoses'] #Column names (Attributes)
for i in range(1,10):# ploting by column (attributes)
    plt.subplot(3,3,i)
    plt.hist(freq(i-1),10,range=(1,10), align=('mid'), color=['green', 'orange'], label=['benign',
'malignant'])
    plt.xlabel(names[i-1])
plt.ylabel("Frequency")
plt.show()
#Exercice 6
X = df_data
kf = KFold(10,shuffle=True,random_state=114)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25)
knn = KNeighborsClassifier(n_neighbors = 3 ,weights='uniform')#uniform weights
knn = KNeighborsClassifier(n_neighbors = 3 ,metric='wminkowski', p=2,
                           metric_params={'w': np.random.random(X_train.shape[1])})#euclidean
knn.fit(X,Y)
a = knn.score(X_train, Y_train)
b = knn.score(X_test,Y_test)
print("When k = 3 uniform Accuracy train : %.4f test : %.4f" % (a,b))
#Exercise 7
clf = MultinomialNB()
clf.fit(X, Y)
print("Multinominal NB score = %0.4f " %clf.score(X,Y))
```

**END**