

Aprendizagem 2021/22  
Homework IV – Group 96

I. Pen-and-paper

1)

HW4

1-  $P(C=1) = 0,7$ ,  $P(C=2) = 0,3$ ,  $P(x|C=1) = N(\mu_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$ ,  $P(x|C=2) = N(\mu_2 = \begin{bmatrix} -1 \\ -4 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix})$

Para  $x_1$ :  $P(C=1|x_1) = P(C=1)P(x_1|C=1) = 0,7 \times \frac{1}{2\pi \det(\Sigma_1)} \exp\left(-\frac{1}{2} \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right)^T \Sigma_1^{-1} \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \end{bmatrix}\right)\right) = 0,11141$

$P(C=2|x_1) = P(C=2)P(x_1|C=2) = 0,3 \times \frac{1}{2\pi \det(\Sigma_2)} \exp\left(-\frac{1}{2} \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} - \begin{bmatrix} -1 \\ -4 \end{bmatrix}\right)^T \Sigma_2^{-1} \left(\begin{bmatrix} 2 \\ 4 \end{bmatrix} - \begin{bmatrix} -1 \\ -4 \end{bmatrix}\right)\right) = 2,8316 \times 10^{-10}$

Normalizar:  $P(C=1|x_1) = \frac{P(C=1|x_1)}{P(C=1|x_1) + P(C=2|x_1)} = 0,999999999975$ ,  $P(C=2|x_1) = 1 - P(C=1|x_1) = 2,5 \times 10^{-9}$

Para  $x_2$ :  $P(C=1|x_2) = 7 \times 10^{-16}$ ,  $P(C=2|x_2) = 0,9999999999999993$

"  $x_3$ :  $P(C=1|x_3) = 0,9827$ ,  $P(C=2|x_3) = 0,0173 = B_3$

"  $x_4$ :  $P(C=1|x_4) = 0,8570$ ,  $P(C=2|x_4) = 0,1430 = B_4$

$\mu_1 = \frac{A_1 \begin{pmatrix} 2 \\ 4 \end{pmatrix} + A_2 \begin{pmatrix} -1 \\ -4 \end{pmatrix} + A_3 \begin{pmatrix} -1 \\ 2 \end{pmatrix} + A_4 \begin{pmatrix} 4 \\ 0 \end{pmatrix}}{A_1 + A_2 + A_3 + A_4} = \begin{pmatrix} 1,565 \\ 2,101 \end{pmatrix}$ ,  $\Sigma_1 = \begin{pmatrix} 4,133 & -1,163 \\ -1,163 & 2,606 \end{pmatrix}$

$\Sigma_{00} = \frac{A_1 (2-1,565)^2 + A_2 (-1-1,565)^2 + A_3 (-1-1,565)^2 + A_4 (4-1,565)^2}{A_1 + A_2 + A_3 + A_4} = 4,133$ ,  $\Sigma_{10} = \Sigma_{01} = -1,163$ ,  $\Sigma_{11} = 2,606$

$\mu_2 = \begin{pmatrix} -0,384 \\ -3,418 \end{pmatrix}$ ,  $\Sigma_2 = \begin{pmatrix} 2,702 & 2,106 \\ 2,106 & 2,169 \end{pmatrix}$   $\rightarrow$  tudo em  $P_{\text{norm}}$

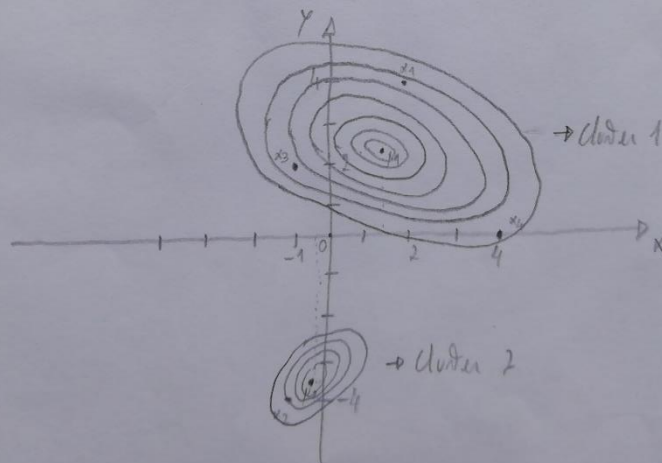
Para as próximas iterações, consideramos:

$P(C=1) = \frac{A_1 + A_2 + A_3 + A_4}{A_1 + A_2 + A_3 + A_4 + B_1 + B_2 + B_3 + B_4} = 0,71$

$P(C=2) = 1 - P(C=1) = 0,29$

$P(x|C=1) = N(\mu_1 = \begin{bmatrix} 1,565 \\ 2,101 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 4,133 & -1,163 \\ -1,163 & 2,606 \end{bmatrix})$

$P(x|C=2) = N(\mu_2 = \begin{bmatrix} -0,384 \\ -3,418 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2,702 & 2,106 \\ 2,106 & 2,169 \end{bmatrix})$



Aprendizagem 2021/22  
**Homework IV – Group 96**

- 2) A silhueta associada ao *cluster*/centroide 2 é maior, logo, é melhor, já que possui apenas um único ponto. O *cluster* 1, tendo 3 pontos, tem uma silhueta menor e acaba por ser menos compacto e ter mais dispersão.

2-  $a[x_i]$  → medida das dist. dos pontos do *cluster* 1,  $b[x_i]$  → medida das dist. a partir do *cluster* 2

$$j(x_1) = 1 - a[x_1]/b[x_1] = 1 - 4,039/8,544 = 0,527, \quad a[x_1] = \frac{3,606 + 4,472}{2} = 4,039, \quad b[x_1] = 8,544$$

$$j(x_2) = 1 - a[x_2]/b[x_2] = 1, \quad j(x_3) = 1 - a[x_3]/b[x_3] = 0,251, \quad j(x_4) = 1 - a[x_4]/b[x_4] = 0,23$$

$$j(c_1) = \frac{0,527 + 0,251 + 0,23}{3} = 0,336, \quad j(c_2) = 1$$

3)

a.  $MLP = n \times n \times 3 + n \times 1 \times 3 + 2 \times n + 2 \times 1$

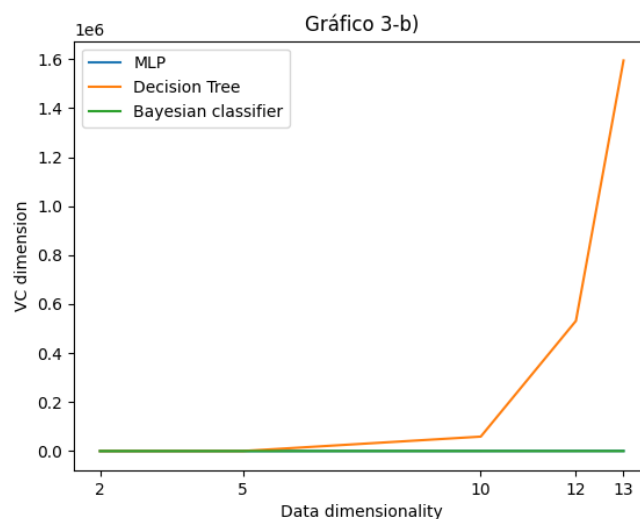
$Decision Tree = 3^n$

$Bayesian Classifier = 1 + 2 \times (n + \frac{n \times n - n}{2})$

Assim, para  $n = 5$ , temos  $d_{vc} = 102$  para o *MLP*,  $d_{vc} = 243$  para a *Decision Tree* e  $d_{vc} = 31$  para o *Bayesian Classifier*.

- b. Usando as equações acima, obtemos:

	2	5	10	12	13
i.	24	102	352	494	574
ii.	9	243	59049	531441	1594323
iii.	7	31	111	157	183

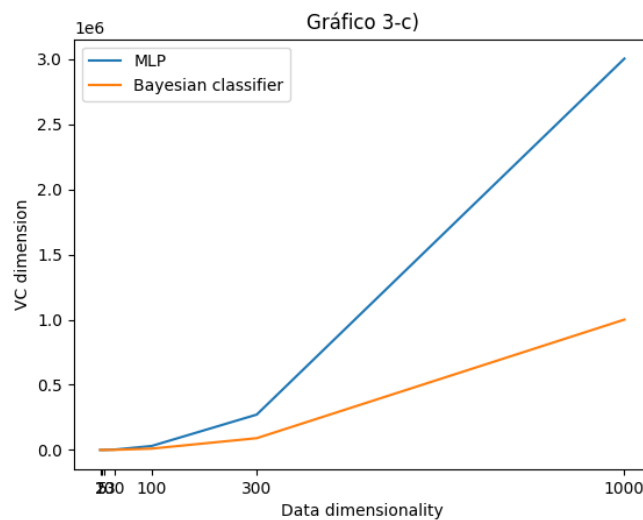


Concluimos que a *VC dimension* da *Decision Tree* aumenta de modo exponencial, ou seja, de modo muito superior aos outros dois classificadores (polinomiais).

Aprendizagem 2021/22  
 Homework IV – Group 96

- c. Além dos dados da alínea anterior para  $m = 2, 5$  e  $10$  para i. e iii., usando as equações acima, obtemos:

	30	100	300	1000
i.	2852	30502	271502	3005002
iii.	931	10101	90301	1001001

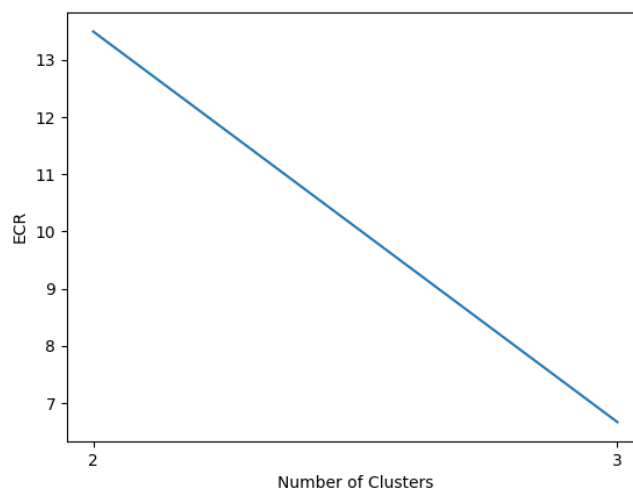


Concluimos que a *VC dimension* do *MLP* e do *Bayesian Classifier* aumenta de forma polinomial e é mais acentuada no *MLP* devido ao termo  $3n^2$  em comparação com o termo  $n^2$  do *Bayesian classifier*.

## II. Programming and critical analysis

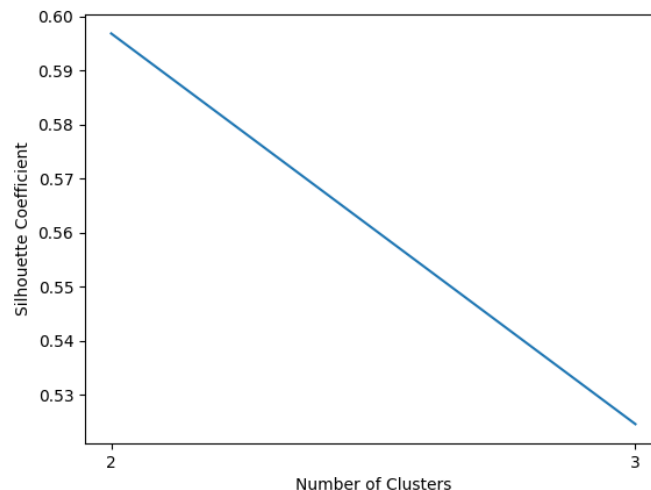
4)

- a. Com 3 *clusters*, o *ECR* é menor, o que significa que há menos classificações erradas.



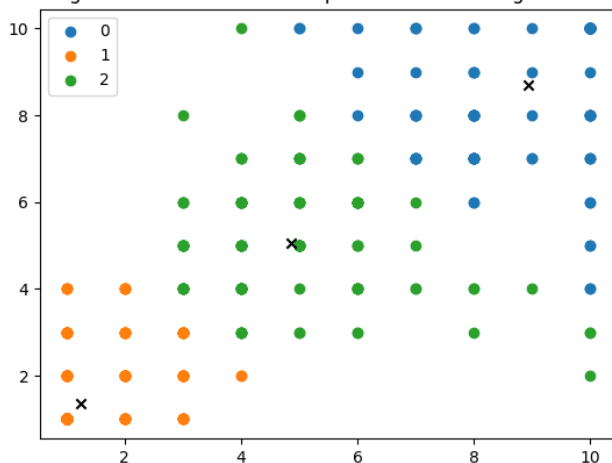
Aprendizagem 2021/22  
 Homework IV – Group 96

- b. Com 3 *clusters*, a *silhouette* é menor, o que significa que os *clusters* estão mais compactos, minimizando a distância entre pontos do mesmo *cluster* e maximizando a distância entre pontos de *clusters* diferentes.



5)

Clustering solution with k=3 and top 2 features with higher mutual info



Clustering solution with k=2 and top 2 features with higher mutual info

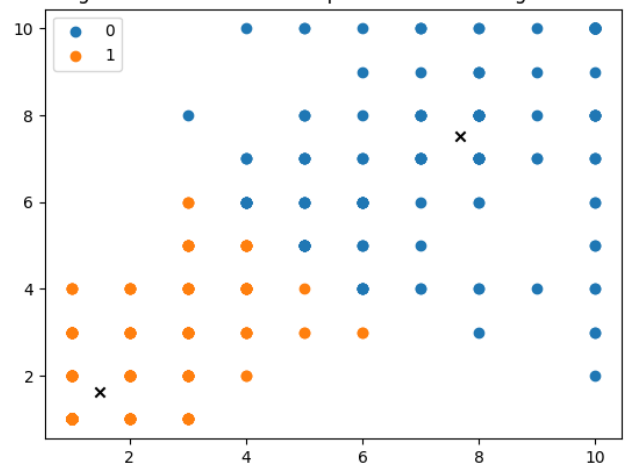


Figura 1 Gráfico extra para análise na pergunta 6

- 6) Com base no gráfico da pergunta anterior, concluímos que 3 *clusters* é uma boa solução, já que diferencia bem as classes reais, *malign* e *benign*. O cluster do “meio”, a verde, corresponde à classe *malign* e, apesar de conter algumas amostras da classe *benign*, o erro não é assim muito grande (11,6). Se compararmos com o uso de apenas 2 *clusters* na mesma situação (gráfico extra na pergunta 5), concluímos que há muito mais classificações erradas, com um *ECR* de 32,0, levando a uma solução pior do que com 3 *clusters*.

### III. APPENDIX

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
```

Aprendizagem 2021/22  
**Homework IV – Group 96**

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_classif

df = pd.read_csv("test.csv", dtype={"Clump_Thickness": int,
                                     "Cell_Size_Uniformity": int,
                                     "Cell_Shape_Uniformity": int,
                                     "Marginal_Adhesion": int,
                                     "Single_Epi_Cell_Size": int,
                                     "Bare_Nuclei": float,
                                     "Bland_Chromatin": int,
                                     "Normal_Nucleoli": int,
                                     "Mitoses": int,
                                     "Class": str}, na_values=["?"])

df = df.dropna()

columns = ["Clump_Thickness", "Cell_Size_Uniformity", "Cell_Shape_Uniformity",
           "Marginal_Adhesion", "Single_Epi_Cell_Size", "Bare_Nuclei", "Bland_Chromatin",
           "Normal_Nucleoli", "Mitoses", "Class"]
features = columns[:-1]
classes = ["benign", "malign"]

Y = df["Class"]
X = df.drop(columns=["Class"])

Y = [0 if x == "benign" else 1 for x in Y]
Y = np.ravel(pd.DataFrame(Y))

# Pergunta 4

def ecr(true_labels, clf_labels, n_clusters):
    cluster0, cluster1, cluster2 = [], [], [] # indices dos pontos pertencentes aos
    cluster K

    for i in range(0, len(true_labels)):
        if clf_labels[i] == 0:
            cluster0 += [i]
        elif clf_labels[i] == 1:
            cluster1 += [i]
        else:
            cluster2 += [i]

    # cK indica a classe real maioritaria no cluster K
    l0 = list(true_labels[cluster0])
    c0 = max(l0, key=l0.count) if l0 != [] else 0
    l1 = list(true_labels[cluster1])
    c1 = max(l1, key=l1.count) if l1 != [] else 0
    l2 = list(true_labels[cluster2])
    c2 = max(l2, key=l2.count) if l2 != [] else 0

    print(c0, c1, c2)

    errors = 0
    l = [[cluster0, c0], [cluster1, c1], [cluster2, c2]]
    for c in l:
        for i in c[0]:
            if true_labels[i] != c[1]: # comparar classe real com a classe desse
cluster e quantas classificacoes erradas
```

```
        errors += 1

    return (1/n_clusters) * errors

silhouette_coefficients = []
ecrs = []
n_clusters = (2, 3)
for k in n_clusters:
    clf = KMeans(n_clusters=k, random_state=0)
    clf.fit(X)
    ecrs.append(ecr(Y, clf.labels_, k))
    score = silhouette_score(X, clf.labels_)
    silhouette_coefficients.append(score)

plt.plot(n_clusters, ecrs)
plt.xticks(n_clusters)
plt.xlabel("Number of Clusters")
plt.ylabel("ECR")
plt.show()

plt.plot(n_clusters, silhouette_coefficients)
plt.xticks(n_clusters)
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Coefficient")
plt.show()

# Pergunta 5

ecrs2 = []
silhouette_coefficients2 = []

select = SelectKBest(score_func=mutual_info_classif, k=2)
X_new = select.fit_transform(X, Y)

for k in n_clusters:
    clf = KMeans(n_clusters=k, random_state=0)
    clf.fit(X_new)

    label = clf.labels_
    u_labels = np.unique(label)
    centroids = clf.cluster_centers_

    for i in u_labels:
        plt.scatter(X_new[label == i, 0], X_new[label == i, 1], label=i)
    plt.scatter(centroids[:, 0], centroids[:, 1], marker="x", color='k')
    plt.legend()
    plt.title("Clustering solution with k=" + str(k) + "and top 2 features with\nhigher mutual info")
    plt.show()

    print(ecr(Y, clf.labels_, k))
```

END