Aprendizagem 2021/22

Homework II – Group 020

I. Pen-and-paper

1)

$$\phi_j(x) = \|x\|_2^j \quad , \quad j = 0, 1, 2, 3 \qquad \hat{z}(x) = \hat{z}\left(\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}\right) = w_0\|y\|_2^0 + w_1\|y\|_2^1 + w_2\|y\|_2^2 + w_3\|y\|_2^3$$

$$= w_0 + w_1\|y\|_2^1 + w_2\|y\|_2^2 + w_3\|y\|_2^3$$

$$\phi_j(x) = \phi\left(\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}\right) = \begin{bmatrix} \|y\|_2^0 \\ \|y\|_2^1 \\ \|y\|_2^2 \\ \|y\|_2^3 \end{bmatrix}$$

$$\phi = \begin{bmatrix} 1 & \sqrt{2} & 2 & 2^{3/2} \\ 1 & \sqrt{27} & 27 & 27^{3/2} \\ 1 & \sqrt{20} & 20 & 20^{3/2} \\ 1 & \sqrt{14} & 14 & 14^{3/2} \\ 1 & \sqrt{53} & 53 & 53^{3/2} \\ 1 & \sqrt{3} & 3 & 3^{3/2} \\ 1 & \sqrt{8} & 8 & 8^{3/2} \\ 1 & \sqrt{85} & 85 & 85^{3/2} \end{bmatrix} \qquad z = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 0 \\ 6 \\ 4 \\ 5 \\ 7 \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Função de erro:

$$E(w) = \sum_{n=1}^{8} \left[z_n - \hat{z}(x_n)\right]^2 = \sum_{n=1}^{8} \left[z_n - \phi_n^T w\right]^2$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j}\left(\sum_{n=1}^{8}\left[z_n - \phi_n^T w\right]^2\right) = \sum_{n=1}^{8} \frac{\partial}{\partial w_j}\left(z_n - \phi_n^T w\right)^2 = \sum_{n=1}^{8} 2\left(z_n - \phi_n^T w\right)\frac{\partial}{\partial w_j}\left(z_n - \phi_n^T w\right) =$$

$$= 2\sum_{n=1}^{8}\left(z_n - \phi_n^T w\right)\phi_n = 2\left[\sum_{n=1}^{8} \phi_n^T z_n - \sum_{n=1}^{8} \phi_n^T \phi_n w\right] = 2\left(\phi^T z - \phi^T \phi w\right)$$

$$\frac{\partial E}{\partial w_j} = 0 \iff 2\left(\phi^T z - \phi^T \phi w\right) = 0 \iff \phi^T z - \phi^T \phi w = 0 \iff w = \left(\phi^T \phi\right)^{-1} \phi^T z =$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \sqrt{2} & \sqrt{27} & \sqrt{20} & \sqrt{14} & \sqrt{53} & \sqrt{3} & \sqrt{8} & \sqrt{85} \\ 2 & 27 & 20 & 14 & 53 & 3 & 8 & 85 \\ 2^{3/2} & 27^{3/2} & 20^{3/2} & 14^{3/2} & 53^{3/2} & 3^{3/2} & 8^{3/2} & 85^{3/2} \end{bmatrix} \begin{bmatrix} 1 & \sqrt{2} & 2 & 2^{3/2} \\ 1 & \sqrt{27} & 27 & 27^{3/2} \\ 1 & \sqrt{20} & 20 & 20^{3/2} \\ 1 & \sqrt{14} & 14 & 14^{3/2} \\ 1 & \sqrt{53} & 53 & 53^{3/2} \\ 1 & \sqrt{3} & 3 & 3^{3/2} \\ 1 & \sqrt{8} & 8 & 8^{3/2} \\ 1 & \sqrt{85} & 85 & 85^{3/2} \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \sqrt{2} & \sqrt{27} & \sqrt{20} & \sqrt{14} & \sqrt{53} & \sqrt{3} & \sqrt{8} & \sqrt{85} \\ 2 & 27 & 20 & 14 & 53 & 3 & 8 & 85 \\ 2^{3/2} & 27^{3/2} & 20^{3/2} & 14^{3/2} & 53^{3/2} & 3^{3/2} & 8^{3/2} & 85^{3/2} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 2 \\ 0 \\ 6 \\ 4 \\ 5 \\ 7 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 35.884 & 212 & 1482.281 \\ 35.884 & 212 & 1482.281 & 11436 \\ 212 & 1482.281 & 11436 & 93573.516 \\ 1482.281 & 11436 & 93573.516 & 793976 \end{bmatrix}^{-1} \begin{bmatrix} 28 \\ 155.235 \\ 1088 \\ 8537.229 \end{bmatrix} = \begin{bmatrix} 8.196 & -6.231 & 1.305 & -0.079 \\ -6.231 & 5.078 & -1.104 & 0.069 \\ 1.305 & -1.104 & 0.247 & -0.016 \\ -0.079 & 0.069 & -0.016 & -0.007 \end{bmatrix}^{-1} \begin{bmatrix} 28 \\ 155.235 \\ 1088 \\ 8537.229 \end{bmatrix}$$

$$= \begin{bmatrix} 4.584 \\ -1.687 \\ 0.338 \\ -0.013 \end{bmatrix} \longrightarrow \hat{z}(x) = \hat{z}\left(\begin{bmatrix} 1 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}\right) = 4.584 - 1.687\|y\|_2^1 + 0.338\|y\|_2^2 - 0.013\|y\|_2^3$$

Aprendizagem 2021/22

Homework II – Group 020

2)

$$\hat{z}(x_9) = \hat{z}\left(\begin{bmatrix}1\\2\\0\\0\end{bmatrix}\right) = 4.584 - 1.687\sqrt{4} + 0.338\times 4 - 0.013\times 4^{3/2} = 2.458$$

$$\hat{z}(x_{10}) = \hat{z}\left(\begin{bmatrix}1\\1\\2\\1\end{bmatrix}\right) = 4.584 - 1.687\sqrt{6} + 0.338\times 6 - 0.013\times 6^{3/2} = 2.289$$

$$RMSE(z,\hat{z}) = \sqrt{\frac{(2-2.458)^2+(4-2.289)^2}{2}} = 1.252$$

3)

Binarização de y_3:

$y_3 \rightarrow$ 0 1 2 3 | 4 5 7 9

média = 3.5

$$y_3' = \begin{cases} 0, & y_3^i \leq 3.5 \\ 1, & \text{else} \end{cases}$$

$$t_i = \begin{cases} P, & \text{output}_i \geq 4 \\ N, & \text{else} \end{cases}$$

	y_1	y_2	y_3'	classe
x_1	1	1	0	N
x_2	1	1	1	N
x_3	0	2	1	N
x_4	1	2	0	N
x_5	2	0	1	P
x_6	1	1	0	P
x_7	2	0	0	P
x_8	0	2	1	P
x_9	2	0	0	N
x_{10}	1	2	0	P

$$H(z) = -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1 \text{ bit}$$

$$H(z|y_1) = \frac{2}{8}H(z|y_1=0) + \frac{4}{8}H(z|y_1=1) + \frac{2}{8}H(z|y_1=2)$$
$$= \frac{2}{8}\times -\left[\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{2}\log_2\frac{1}{2}\right] + \frac{4}{8}\times -\left[\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right] + \frac{2}{8}\times -\left[1\log_2 1\right]$$
$$= \frac{1}{4}\times 1 + \frac{1}{2}\times 0.811 + \frac{1}{4}\times 0 = 0.656 \text{ bits}$$

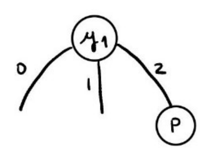
$$H(z|y_2) = \frac{2}{8}H(z|y_2=0) + \frac{3}{8}H(z|y_2=1) + \frac{3}{8}H(z|y_2=2)$$
$$= \frac{2}{8}\times -\left[1\log_2 1\right] + \frac{3}{8}\times -\left[\frac{2}{3}\log_2\frac{2}{3}+\frac{1}{3}\log_2\frac{1}{3}\right] + \frac{3}{8}\times -\left[\frac{2}{3}\log_2\frac{2}{3}+\frac{1}{3}\log_2\frac{1}{3}\right]$$
$$= \frac{1}{4}\times 0 + \frac{3}{8}\times 0.918 + \frac{3}{8}\times 0.918 = 0.689 \text{ bits}$$

$$H(z|y_3') = \frac{4}{8}H(z|y_3'=0) + \frac{4}{8}H(z|y_3'=1) = \frac{4}{8}\times -\left[\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{2}\log_2\frac{1}{2}\right] + \frac{4}{8}\times -\left[\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{2}\log_2\frac{1}{2}\right]$$
$$= \frac{4}{8} + \frac{4}{8} = 1 \text{ bit}$$

$IG(z|y_1) = H(z) - H(z|y_1) = 1 - 0.656 = 0.344$ bits $\longrightarrow$ maior ganho de informação $\rightarrow$ primeira feature na árvore de decisão

$IG(z|y_2) = H(z) - H(z|y_2) = 1 - 0.689 = 0.311$ bits

$IG(z|y_3') = H(z) - H(z|y_3') = 1 - 1 = 0$ bits



Aprendizagem 2021/22

Homework II – Group 020

Agora temos de ver entre y_2 e y_3' qual tem o maior ganho de informação para $y_1=0$ e $y_1=1$ para ver qual será a segunda feature na árvore de decisão. Para $y_1=2$ podemos ver diretamente que terá classe P.

for $y_1=0$: $H(z|y_1=0) = -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1\text{ bit}$

$$H(z|y_2, y_1=0) = 0\times H(z|y_2=0, y_1=0) + 0\times H(z|y_2=1, y_1=0) + \frac{2}{2}H(z|y_2=2, y_1=0) =$$
$$= 0 + 0 + \frac{2}{2}\times -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1\text{ bit}$$

$$H(z|y_3', y_1=0) = 0\times H(z|y_3'=0, y_1=0) + \frac{2}{2}H(z|y_3'=1, y_1=0) =$$
$$= 0 + \frac{2}{2}\times -\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1\text{ bit}$$

$IG(z|y_2, y_1=0) = H(z|y_1=0) - H(z|y_2, y_1=0) = 1 - 1 = 0\text{ bits}$
$IG(z|y_3', y_1=0) = H(z|y_1=0) - H(z|y_3', y_1=0) = 1 - 1 = 0\text{ bits}$
> Empate → escolhemos aleatoriamente y_2

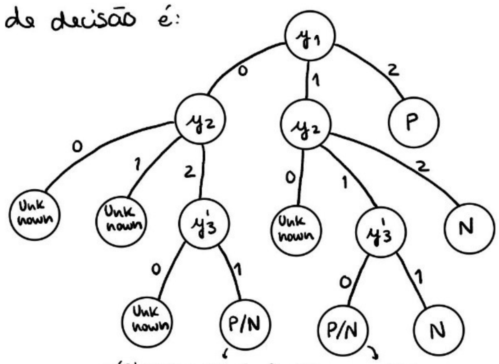
for $y_1=1$: $H(z|y_1=1) = -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] = 0.811\text{ bits}$

$$H(z|y_2, y_1=1) = 0\times H(z|y_2=0, y_1=1) + \frac{3}{4}H(z|y_2=1, y_1=1) + \frac{1}{4}H(z|y_2=2, y_1=1) =$$
$$= 0 + \frac{3}{4}\times -\left[\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right] + \frac{1}{4}\times -\left[1\log_2 1\right] = 0.786$$

$$H(z|y_3', y_1=1) = \frac{3}{4}H(z|y_3'=0, y_1=1) + \frac{1}{4}H(z|y_3'=1, y_1=1) =$$
$$= \frac{3}{4}\times -\left[\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right] + \frac{1}{4} - \left[1\log_2 1\right] = 0.786$$

$IG(z|y_2, y_1=1) = H(z|y_1=1) - H(z|y_2, y_1=1) = 0.811 - 0.786 = 0.025\text{ bits}$
$IG(z|y_3', y_1=1) = H(z|y_1=1) - H(z|y_3', y_1=1) = 0.811 - 0.786 = 0.025\text{ bits}$
> Empate → escolhemos aleatoriamente y_2

A árvore de decisão é:



$P(P|y_1=0, y_2=2, y_3'=1) = 0.5$
$P(N|y_1=0, y_2=2, y_3'=1) = 0.5$

$P(P|y_1=1, y_2=1, y_3'=0) = 0.5$
$P(N|y_1=1, y_2=1, y_3'=0) = 0.5$

4)

$\hat{z}(x_9) = \hat{z}\left(\begin{bmatrix}2\\0\\0\end{bmatrix}\right) = P$ ✓

$\hat{z}(x_{10}) = \hat{z}\left(\begin{bmatrix}1\\2\\1\end{bmatrix}\right) = N$ ✓

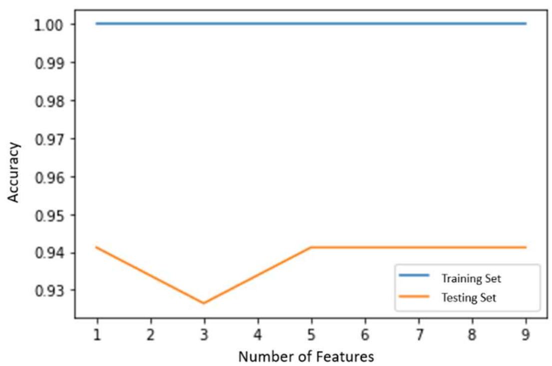
→ Ambos os dados de teste foram bem classificados por isso a accuracy no conjunto de teste é 100%.

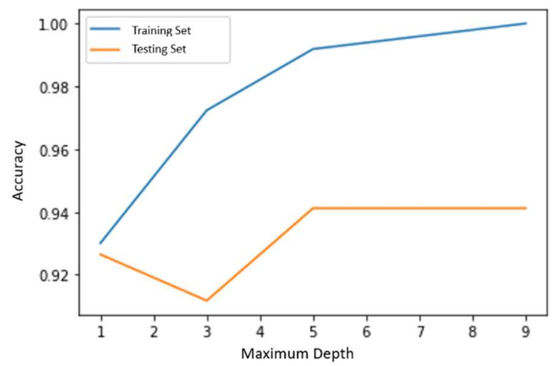
Aprendizagem 2021/22

Homework II – Group 020

II. Programming and critical analysis

5) Os gráficos abaixo representam a variação da *training* e *testing accuracy* variando o número de *features* (esquerda) e a profundidade máxima da árvore (direita).





6) Existe uma clara relação entre ambas as variáveis como se pode observar pelos gráficos acima tornando se mais evidente ao analisar a *accuracy* do conjunto de teste. Esta relação deriva do facto de que dada uma árvore com uma certa profundidade iremos também estar limitados na quantidade de *features* que podemos ter. Assim a profundidade da árvore tem uma relação direta com o número de features e ao variar uma variamos indiretamente a outra também.

7) Para identificarmos a melhor profundidade temos que ver em qual delas existe menor risco de *overfitting*, tal que a *accuracy* do conjunto de teste não desça significativamente em relação à *accuracy* do conjunto de treino. Analisando o gráfico, é de notar que a partir de uma profundidade 5 os valores para a *testing accuracy* estabilizam sendo este o máximo e a *training accuracy* sobe linearmente. Assim podemos concluir que a melhor profundidade é 5 pois tem a melhor accuracy no conjunto de teste e a menor diferença entre a *accuracy* de treino e de teste, estando portanto menos suscetível a *overfitting*.

Aprendizagem 2021/22

Homework II – Group 020

III. APPENDIX

```
from scipy.io import arff
import numpy as np
import pandas as pd
from sklearn import tree
import matplotlib.pyplot as plt
from sklearn.model_selection import KFold
from sklearn.feature_selection import
SelectKBest, mutual_info_regression
from sklearn.metrics import accuracy_score

data = arff.loadarff(r'/home/sara/apre/tpc-
2/breast.w.arff')
df = pd.DataFrame(data[0])

data_array = df.to_numpy()

x = np.empty((0,9))
y = np.empty((0,1))

for row in data_array:
    array_sum = np.sum(row[:-1])
    array_has_nan = np.isnan(array_sum)
    if(not array_has_nan):
        x = np.append(x, [row[:-1]], axis=0)
        y = np.append(y, row[-1])

y_aux = np.empty((0,1))

for idx, val in enumerate(y):
    if(val.decode('UTF-8') == 'benign'):
        y_aux = np.append(y_aux, 1)
    else:
        y_aux = np.append(y_aux, 0)

# ------------------ 5) i. -------------------
num_features = [1, 3, 5, 9]

train_acc_f = []
test_acc_f = []

for i in num_features:
    x_new =
SelectKBest(score_func=mutual_info_regression,
k=i).fit_transform(x, y_aux)

    kf = KFold(n_splits=10, random_state=20,
shuffle=True)
    dtc = tree.DecisionTreeClassifier()

    for train_index , test_index in
kf.split(x_new):
        x_train , x_test = x[train_index],
x[test_index]
        y_train , y_test = y[train_index],
y[test_index]
        dtc.fit(x_train, y_train)

        y_pred_test = dtc.predict(x_test)
        y_pred_train = dtc.predict(x_train)

    test_acc_f.append(accuracy_score(y_test,
y_pred_test))
    train_acc_f.append(accuracy_score(y_train,
y_pred_train))

plt.plot(num_features, train_acc_f,
label='Training Set')
plt.plot(num_features, test_acc_f,
label='Testing Set')
plt.xlabel('Number of Features')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

# ------------------ 5) ii. ------------------
max_depth = [1, 3, 5, 9]

train_acc_d = []
test_acc_d = []

for i in max_depth:
    x_new =
SelectKBest(k='all').fit_transform(x, y_aux)

    kf = KFold(n_splits=10, random_state=20,
shuffle=True)
    dtc =
tree.DecisionTreeClassifier(max_depth=i)

    for train_index , test_index in
kf.split(x_new):
        x_train , x_test = x[train_index],
x[test_index]
        y_train , y_test = y[train_index],
y[test_index]
        dtc.fit(x_train, y_train)

        y_pred_test = dtc.predict(x_test)
        y_pred_train = dtc.predict(x_train)

    test_acc_d.append(accuracy_score(y_test,
y_pred_test))
    train_acc_d.append(accuracy_score(y_train,
y_pred_train))

plt.plot(max_depth, train_acc_d,
label='Training Error')
plt.plot(max_depth, test_acc_d, label='Testing
Error')
plt.xlabel('Maximum Depth')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

END