TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 009

I. Pen-and-paper

1) Começamos por calcular a matriz de desenho (de tamanho 8x4 pois são 8 observações e M=3)

$$X = \begin{pmatrix} 1 & \phi_{11} & \phi_{12} & \phi_{13} \\ 1 & \phi_{12} & \phi_{22} & \phi_{23} \\ 1 & \phi_{13} & \phi_{23} & \phi_{33} \\ 1 & \phi_{14} & \phi_{24} & \phi_{43} \\ 1 & \phi_{15} & \phi_{25} & \phi_{53} \\ 1 & \phi_{16} & \phi_{26} & \phi_{63} \\ 1 & \phi_{17} & \phi_{27} & \phi_{73} \\ 1 & \phi_{18} & \phi_{28} & \phi_{83} \end{pmatrix} = \begin{pmatrix} 1 & 1.414 & 2 & 2.828 \\ 1 & 5.196 & 27 & 140.296 \\ 1 & 4.472 & 20 & 89.443 \\ 1 & 3.742 & 14 & 52.383 \\ 1 & 7.28 & 53 & 385.846 \\ 1 & 1.732 & 3 & 5.196 \\ 1 & 2.828 & 4 & 22.627 \\ 1 & 9.22 & 85 & 783.661 \end{pmatrix}$$

Exemplificação dos cálculos para as primeiras duas linhas da matriz:

$$\phi_{ij} = \|\boldsymbol{x_i}\|_2{}^{j} = \left(\sqrt{y_1^2 + y_2^2 + y_3^2}\right)^j$$

$\phi_{11} = \|\boldsymbol{x_1}\|_2{}^{1} = \left(\sqrt{1^2 + 1^2 + 0^2}\right)^1 = \sqrt{2} = 1.141$ $\quad$ $\phi_{21} = \|\boldsymbol{x_2}\|_2{}^{1} = \left(\sqrt{1^2 + 1^2 + 5^2}\right)^1 = \sqrt{27} = 5.196$

$\phi_{12} = \|\boldsymbol{x_1}\|_2{}^{2} = \left(\sqrt{1^2 + 1^2 + 0^2}\right)^2 = 2$ $\quad$ $\phi_{22} = \|\boldsymbol{x_2}\|_2{}^{2} = \left(\sqrt{1^2 + 1^2 + 5^2}\right)^2 = 27$

$\phi_{13} = \|\boldsymbol{x_1}\|_2{}^{3} = \left(\sqrt{1^2 + 1^2 + 0^2}\right)^3 = \left(\sqrt{2}\right)^3 = 2.828$ $\quad$ $\phi_{23} = \|\boldsymbol{x_2}\|_2{}^{3} = \left(\sqrt{1^2 + 1^2 + 5^2}\right)^3 = \left(\sqrt{27}\right)^3 = 140.296$

Para obter a função de aproximação $f(x, w) = \sum_{j=0}^{3} w_j \phi_j(x)$, iremos de seguida obter o vetor w que minimiza o erro da função em relação aos dados de treino. Seguidamente obtemos o vetor **w**, tal que $\nabla E(w) = 0$.

$$\nabla E(w) = 0 \Leftrightarrow w = (X^T \cdot X)^{-1} \cdot X^T . z$$

$$w = \left(\begin{pmatrix} 1 & 1.414 & 2 & 2.828 \\ 1 & 5.196 & 27 & 140.296 \\ 1 & 4.472 & 20 & 89.443 \\ 1 & 3.742 & 14 & 52.383 \\ 1 & 7.28 & 53 & 385.846 \\ 1 & 1.732 & 3 & 5.196 \\ 1 & 2.828 & 4 & 22.627 \\ 1 & 9.22 & 85 & 783.661 \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 1.414 & 2 & 2.828 \\ 1 & 5.196 & 27 & 140.296 \\ 1 & 4.472 & 20 & 89.443 \\ 1 & 3.742 & 14 & 52.383 \\ 1 & 7.28 & 53 & 385.846 \\ 1 & 1.732 & 3 & 5.196 \\ 1 & 2.828 & 4 & 22.627 \\ 1 & 9.22 & 85 & 783.661 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 1 & 1.414 & 2 & 2.828 \\ 1 & 5.196 & 27 & 140.296 \\ 1 & 4.472 & 20 & 89.443 \\ 1 & 3.742 & 14 & 52.383 \\ 1 & 7.28 & 53 & 385.846 \\ 1 & 1.732 & 3 & 5.196 \\ 1 & 2.828 & 4 & 22.627 \\ 1 & 9.22 & 85 & 783.661 \end{pmatrix}^T \cdot \begin{pmatrix} 1 \\ 3 \\ 2 \\ 0 \\ 6 \\ 4 \\ 5 \\ 7 \end{pmatrix}$$

$$. w = \begin{pmatrix} 4.5939 \\ -1.6959 \\ 0.3396 \\ -0.0134 \end{pmatrix}$$

Assim acabamos por obter a expressão para a função de aproximação

$$f(\boldsymbol{x}) = \sum_{j=0}^{3} w_j \phi_j(\boldsymbol{x}) = 4.539 - 1.6959 \, \|\boldsymbol{x}\|_2 + 0.3396 \|\boldsymbol{x}\|_2^2 - 0.0134 \|\boldsymbol{x}\|_2^3$$

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 009

2) Para calcular o RMSE começamos por determinar a previsão do modelo para todas as observações de teste e de seguida aplicamos a fórmula abaixo para cálculo do erro.

	z_i	$\widehat{z_i}$	$(z_i - \widehat{z_i})^2$
1	1	2.837	3.375
2	3	3.071	0.005
3	2	2.603	0.364
4	0	2.3	5.291
5	6	5.076	0.853
6	4	2.606	1.944
7	5	2.211	7.766
8	7	7.323	0.104

$$RMSE = \sqrt{\frac{1}{8}\sum_{i=1}^{8}(z_i - \widehat{z_i})^2} = 1.5697$$

Exemplificação do cálculo de $\widehat{z_i}$

$$\|x_1\|_2 = \phi_{11} = 1.414$$

$$\|x_1\|_2^2 = \phi_{12} = 2$$

$$\|x_1\|_2^3 = \phi_{13} = 2.828$$

$$\widehat{z_1} = f(x_1) = 4.539 - 1.6959 * 1.414 + 0.3396 * 2 - 0.0134 * 2.828 = 2.837$$

3) Para a binarização da variável y_3 começamos por ordenar os seus valores no conjunto de treino e obter uma mediana de 3.5. Assim, todos os valores menores que a mediana pertencerão a um dos *bins*, enquanto que os restantes valores farão parte do outro, garantindo assim uma separação equilibrada.

De seguida, prosseguimos com a escolha das variáveis para a construção da árvore de decisão. Começamos por obter o ganho de informação para cada uma das variáveis, escolhendo para raiz a variável y_1, uma vez que possui o maior ganho de informação.

$$E(z) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 1$$

$$E(z|y_1) = \frac{2}{8}\left(-\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right) + \frac{4}{8}\left(-\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right)\right) + \frac{2}{8}\left(-\frac{2}{2}\log_2\left(\frac{1}{2}\right)\right) = 0.656$$

$$E(z|y_2) = \frac{2}{8}\left(-\frac{2}{2}\log_2\left(\frac{1}{2}\right)\right) + \frac{3}{8}\left(-\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right) + \frac{3}{8}\left(-\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right) = 0.689$$

$$E(z|y_3) = \frac{4}{8}\left(-\left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right)\right) + \frac{4}{8}\left(-\left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right)\right) = 1$$

$$IG(z|y_1) = 1 - 0.656 = 0.344$$

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

$$IG(z|y_3) = 1 - 1 = 0$$

O processo de escolha de variáveis é repetido para cada nó, até ser obtida a árvore na sua totalidade.

Para $y_1 = 0$, nenhuma das variáveis apresenta ganho de informação pelo que o nó fica indeterminado, uma vez que existiria uma probabilidade de 50% de classificação como P e 50% de probabilidade de classificação como N.

$$E(z) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 1$$

$$E(z|y_2) = 1 * \left(-\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right) = 1 \qquad IG(z|y_2) = 1 - 1 = 0$$

$E(z|y_3) = 1 * \left(-\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right) = 1 \qquad IG(z|y_3) = 1 - 1 = 0$

Para $y_1 = 1$, ambas as variáveis possuem o mesmo valor de ganho de informação pelo que podemos optar por escolher qualquer uma das duas. Neste caso optamos por escolher y_2.

$E(z) = -\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.811$

$E(z|y_2) = \frac{3}{4}\left(-\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right) + \frac{1}{4}\left(-\frac{1}{1}\log_2\left(\frac{1}{1}\right)\right) = 0.689 \qquad IG(z|y_2) = 0.811 - 0.689 = 0.122$

$E(z|y_3) = \frac{3}{4}\left(-\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right)\right) + \frac{1}{4}\left(-\frac{1}{1}\log_2\left(\frac{1}{1}\right)\right) = 0.689 \qquad IG(z|y_3) = 0.811 - 0.689 = 0.122$

Para $y_1 = 2$, verificamos que a entropia da variável de output tem valor 0 pelo que este nó classifica a observação como P.

$E(z) = -\frac{2}{2}\log_2\frac{2}{2} = 0$

Para $y_1 = 1\ e\ y_2 = 2$, , dado que a entropia da variável de output é também 0, obtemos um nó que classifica a observação como N.

$E(z|y_1 = 1, y_2 = 2) = -\frac{1}{1}\log_2\frac{1}{1} = 0$

Para $y_1 = 1\ e\ y_2 = 1$, escolhemos a variável que resta uma vez que apresenta ganho de informação

$E(z) = -\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right) = 0.918$

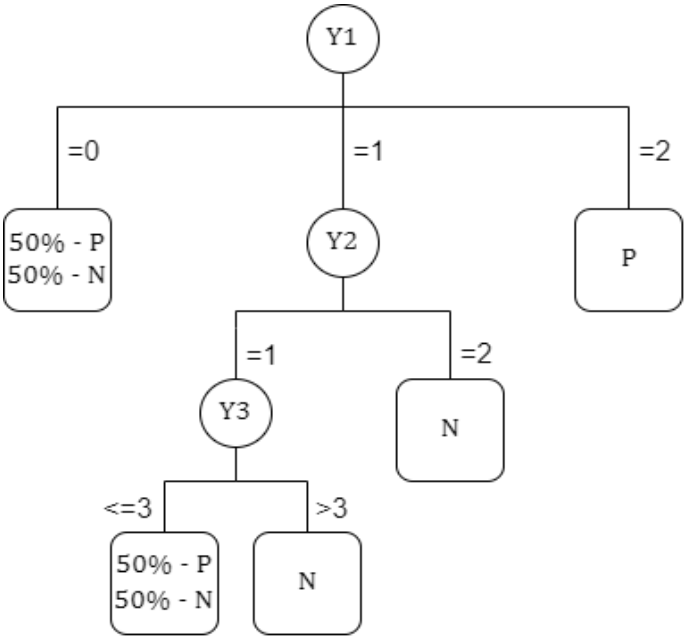
$E(z|y_3) = \frac{2}{3}\left(-\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)\right) + \frac{1}{3}\left(-\frac{1}{1}\log_2\left(\frac{1}{1}\right)\right) = 0.667 \qquad IG(z|y_3) = 0.918 - 0.667 = 0.251$

Para $y_1 = 1, y_2 = 1\ e\ y_3 \leq 3$, obtemos uma entropia de 1 para a variável de output, ficando o nó indeterminado e verificando-se igual probabilidade de ocorrência de as observações classificadas como P ou como N.

$E(z) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 1$

Para $y_1 = 1, y_2 = 1\ e\ y_3 > 3$, obtemos uma entropia de 0 para a variável de output, pelo que o nó é determinado e classifica as observações como N .

$E(z) = -\frac{1}{1}\log_2\frac{1}{1} = 0$



Árvore de decisão obtida

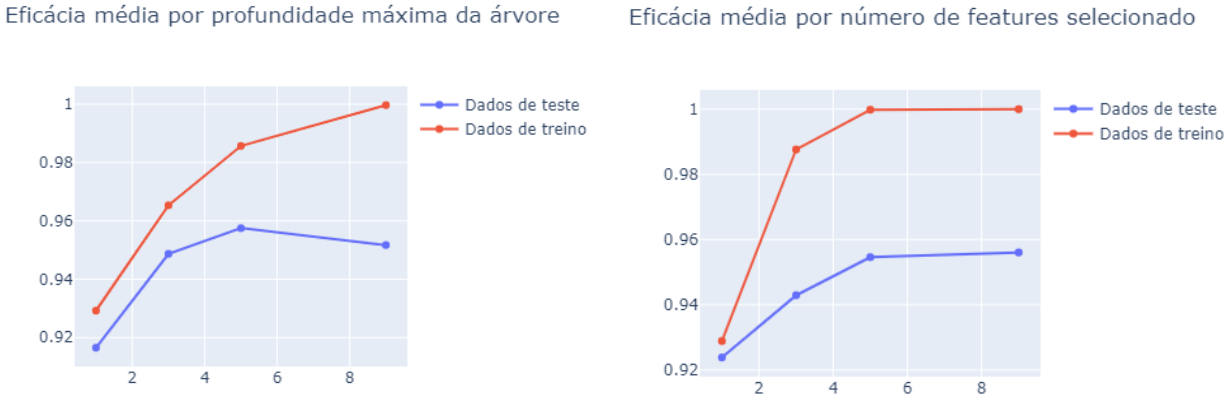
Aprendizagem 2021/22

Homework II – Group 009

4) De acordo com a árvore de decisão obtida, podemos perceber que as observações de teste x_9, x_{10} são classificadas como N e P respetivamente. Assim, podemos concluir que a eficácia do modelo nestas observações é de 0% uma vez que as classificações corretas para as observações seriam P para x_9 e N para x_{10}.

II. Programming and critical analysis

5)



6) A correlação observada entre os valores de eficácia do modelo para diferentes profundidades de árvore e para diferentes números de variáveis selecionadas deve-se sobretudo ao facto de ambos os híper-parâmetros controlarem a complexidade final da árvore de decisão construída. É de notar que estes híper-parâmetros se relacionam, uma vez que uma limitação no número de variáveis selecionadas influencia a profundidade da árvore e vice-versa.

Assim, a razão pela qual os valores da eficácia começam por aumentar tanto nos dados de treino como nos dados de teste deve-se ao facto que a árvore se ir tornando gradualmente mais complexa, abrangendo mais informação e aumentando a capacidade de previsão comparativamente a árvores com menos nós de decisão.

Por outro lado, a razão pela qual os valores de eficácia nos dados de teste deixam de aumentar a partir de determinado valor de profundidade máxima ou de número de variáveis selecionadas, deve-se ao facto de as árvores se tronarem demasiado complexas. Estas árvores acabam por se ajustar em demasia aos dados originando uma perda, ou um crescimento mais lento, da eficácia nos dados de teste e um aumento da eficácia nos dados de treino, o que mostra uma predisposição do modelo para *overfitting*.

7) Com base nos resultados obtidos, a profundidade ideal da árvore será a de valor 5 uma vez que para valores superiores a 5 se nota uma diminuição da eficácia do modelo em dados de teste.

Apesar da eficácia do modelo com profundidades maiores continuar a aumentar ao ser testado com os dados de treino. Este aumento deve-se sobretudo ao *overfitting* do modelo nestes casos, uma vez que o o aumento do valor da profundidade leva ao ajustamento excessivo aos dados de treino provocando uma queda na eficácia do modelo em valores externos ao conjunto de treino.

Ao escolher a profundidade máxima de 5 conseguimos obter um modelo que generalize melhor, não estando demasiado aproximado aos dados de treino.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 009

III. APPENDIX

```
import pandas as pd
import numpy as np
from scipy.io import arff
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import model_selection
import plotly.graph_objects as go
from sklearn.feature_selection import SelectKBest, mutual_info_classif

data = arff.loadarff('breast.w.arff')
df = pd.DataFrame(data[0])
df.dropna(inplace=True)
df.replace(b'benign', 0, inplace=True)
df.replace(b'malignant', 1, inplace=True)
data = df.drop(["Class"],axis=1).values
target = df["Class"].values
fold = model_selection.KFold(n_splits=10, shuffle=True, random_state=9)
features_mean_scores = [], []
max_depth_mean_scores = [], []

for n_features in (1, 3, 5, 9):
    scores = [], []
    for train_filter, test_filter in fold.split(data):
        X_train, X_test, y_train, y_test = data[train_filter], data[test_filter],
target[train_filter], target[test_filter]

        select = SelectKBest(mutual_info_classif, k=n_features).fit(X_train, y_train)
        X_train, X_test = select.transform(X_train), select.transform(X_test)

        clf = DecisionTreeClassifier(random_state=9)
        clf.fit(X_train, y_train)
        predictions = clf.predict(X_test)
        scores[0].append(accuracy_score(y_test, predictions))
        predictions = clf.predict(X_train)
        scores[1].append(accuracy_score(y_train, predictions))

    mean_score = np.array(scores[0]).mean()
    features_mean_scores[0].append(mean_score)
    print("Features =", n_features, "-", mean_score)
    mean_score = np.array(scores[1]).mean()
    features_mean_scores[1].append(mean_score)
    print("Features =", n_features, "-", mean_score)

for max_depth in (1, 3, 5, 9):
    scores = [], []
    for train_filter, test_filter in fold.split(data):
        X_train, X_test, y_train, y_test = data[train_filter], data[test_filter],
target[train_filter], target[test_filter]
        clf = DecisionTreeClassifier(max_depth=max_depth, random_state=9)
        clf.fit(X_train, y_train)
        predictions = clf.predict(X_test)
        scores[0].append(accuracy_score(y_test, predictions))
        predictions = clf.predict(X_train)
        scores[1].append(accuracy_score(y_train, predictions))
    max_depth_mean_scores[0].append(np.array(scores[0]).mean())
    print("Max depth =", max_depth, "-", np.array(scores[0]).mean())
    max_depth_mean_scores[1].append(np.array(scores[1]).mean())
    print("Max depth =", max_depth, "-", np.array(scores[1]).mean())
```

END