TÉCNICO LISBOA

Aprendizagem 2023/24

**Homework I – Group 075**

(ist1103860, ist1103883)

## I. Pen-and-paper

1) • $IG(Y_{out} | Y_1 > 0.4, Y_i) = E(Y_{out} | Y_1 > 0.4) - E(Y_{out} | Y_1 > 0.4, Y_i)$

• $E(Y_{out} | Y_1 > 0.4) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right) \simeq 1.557$

• $E(Y_{out} | Y_1 > 0.4, Y_2) = \frac{3}{7}E(Y_{out} | Y_1 > 0.4, Y_2 = 0) + \frac{2}{7}E(Y_{out} | Y_1 > 0.4, Y_2 = 1) +$

$+ \frac{2}{7}E(Y_{out} | Y_1 > 0.4, Y_2 = 2) = \frac{3}{7}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3}\right) +$

$+ \frac{2}{7}\left(-0\log_2 0 - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right) + \frac{2}{7}\left(-1\log_2 1 - 0\log_2 0 - 0\log_2 0\right) =$

$= \frac{3}{7} \times 3\left(-\frac{1}{3}\log_2\frac{1}{3}\right) + \frac{2}{7} \times 1 + \frac{2}{7} \times 0 \simeq 0.965$

• $E(Y_{out} | Y_1 > 0.4, Y_3) = \frac{1}{7}E(Y_{out} | Y_1 > 0.4, Y_3 = 0) + \frac{2}{7}E(Y_{out} | Y_1 > 0.4, Y_3 = 1) +$

$+ \frac{4}{7}E(Y_{out} | Y_1 > 0.4, Y_3 = 2) = \frac{1}{7}\left(-0\log_2 0 - 1\log_2 1 - 0\log_2 0\right) + \frac{2}{7}\left(-\frac{1}{2}\log_2\frac{1}{2}\right.$

$\left. - \frac{1}{2}\log_2\frac{1}{2} - 0\log_2 0\right) + \frac{4}{7}\left(-\frac{1}{2}\log_2 - \frac{1}{2}\log_2\frac{1}{2} - 0\right) = \frac{1}{7} \times 0 + \frac{2}{7} \times 1 + \frac{4}{7} \times 1 =$

$= 6/7 \simeq 0.857$

• $E(Y_{out} | Y_1 > 0.4, Y_4) = \frac{2}{7}E(Y_{out} | Y_1 > 0.4, Y_4 = 0) + \frac{3}{7}E(Y_{out} | Y_1 > 0.4, Y_4 = 1) +$

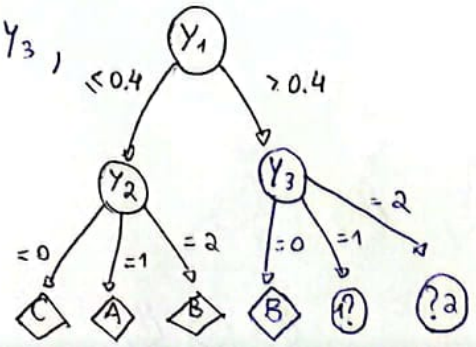
$+ \frac{2}{7}E(Y_{out} | Y_1 > 0.4, Y_4 = 2) = \frac{2}{7}\left(-\frac{1}{2}\log_2\frac{1}{2} - 0\log_2 0 - \frac{1}{2}\log_2\frac{1}{2}\right) + \frac{3}{7}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{1}{3}\right.$

$\left. - 0\log_2 0\right) + \frac{2}{7}\left(-\frac{1}{2}\log_2\frac{1}{2} - 0\log_2 0 - \frac{1}{2}\log_2\frac{1}{2}\right) \simeq 0.965$

• $IG(Y_{out} | Y_1 > 0.4, Y_2) = 1.557 - 0.965 = 0.592 = IG(Y_{out} | Y_1 > 0.4, Y_4)$

• $IG(Y_{out} | Y_1 > 0.4, Y_3) = 1.557 - 0.857 = 0.7$

Optamos por $Y_3$,



(?1) → Não podemos continuar a expandir este ramo, visto que só temos 2 observações, para continuar a expandi-lo, sendo que no mínimo precisamos de 4. Atendendo, a lei), (?1) deverá ficar com o valor A.

(?2) → Temos o número mínimo de observações necessárias para continuar a expandir este nó:

- $IG(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_i) = E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2) - E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_i)$
- $E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2) = \left(-\frac{1}{2}\log_2\frac{1}{2} - 0\log_2 0 - \frac{1}{2}\log_2\frac{1}{2}\right) = 1$
- $E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_2) = \frac{1}{4}E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_2 = 0) + \frac{1}{4}E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_2 = 1) + \frac{2}{4}E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_2 = 2) =$

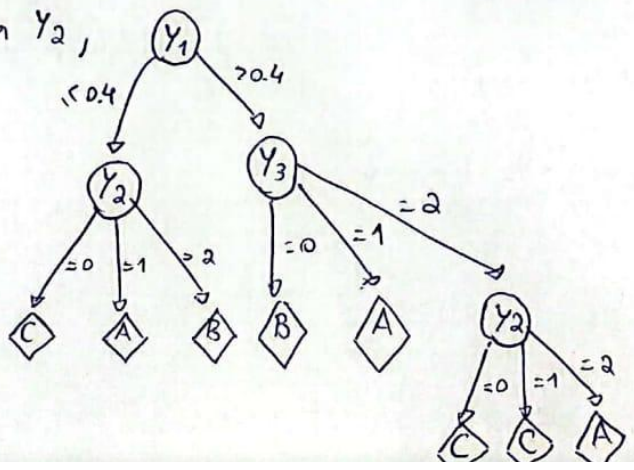
$$= \frac{1}{4}\left(-0\log_2 0 - 0\log_2 0 - 1\log_2 1\right) + \frac{1}{4}\left(-0\log_2 0 - 0\log_2 0 - 1\log_2 1\right) + \frac{1}{2}\left(-1\log_2 1 - 0\log_2 0 - 0\log_2 0\right) = 0$$

- $E(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_4) = \frac{2}{4}E(Y_{out} \mid Y_1 > 0, Y_3 = 2, Y_4 = 0) + \frac{1}{4}E(Y_{out} \mid Y_1 > 0, Y_3 = 2, Y_4 = 1) + \frac{1}{4}E(Y_{out} \mid Y_1 > 0, Y_3 = 2, Y_4 = 2) = \frac{1}{2}\left(-\frac{1}{2}\log_2\frac{1}{2} * 0\log_2 0 - \frac{1}{2}\log_2\frac{1}{2}\right) +$

$$+ \frac{1}{4}\left(-1\log_2 1 - 0\log_2 0 - 0\log_2 0\right) + \frac{1}{4}\left(-0\log_2 0 - 0\log_2 0 - 1\log_2 1\right) = \frac{1}{2}$$

- $IG(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_2) = 1 - 0 = 1$
- $IG(Y_{out} \mid Y_1 > 0.4, Y_3 = 2, Y_4) = 1 - \frac{1}{2} = \frac{1}{2}$

Optamos por $Y_2$,



Aprendizagem 2023/24

# Homework I – Group 075

(ist1103860, ist1103883)

2)

| predicted \ true | A | B | C |
|---|---|---|---|
| A | 4 | 1 | 0 |
| B | 0 | 2 | 0 |
| C | 0 | 1 | 4 |

3) $\frac{1}{F_1} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right) \Leftrightarrow F_1 = \frac{2 \times P \times R}{R + P}$ $\quad P \to$ precision, $R \to$ recall

- $precision_A = \frac{TP_A}{TP_A + FP_A} = \frac{4}{4+1} = \frac{4}{5}$ $\quad recall_A = \frac{TP_A}{P_A} = \frac{TP_A}{FN_A + TP_A} = \frac{4}{0+4} = 1$
- $precision_B = \frac{TP_B}{TP_B + FP_B} = \frac{2}{2+0} = 1$ $\quad recall_B = \frac{TP_B}{P_B} = \frac{TP_B}{TP_B + FN_B} = \frac{2}{2+2} = \frac{2}{4} = \frac{1}{2}$
- $precision_C = \frac{TP_C}{TP_C + FP_C} = \frac{4}{4+1} = \frac{4}{5}$ $\quad recall_C = \frac{TP_C}{P_C} = \frac{TP_C}{TP_C + FN_C} = \frac{4}{4+0} = 1$
- $F_{1A} = F_{1C} = \frac{2 \times \frac{4}{5} \times 1}{1 + \frac{4}{5}} = \frac{\frac{8}{5}}{\frac{9}{5}} = \frac{8}{9}$ $\quad F_{1B} = \frac{2 \times 1 \times \frac{1}{2}}{\frac{1}{2} + 1} = \frac{\frac{2}{2}}{\frac{3}{2}} = \frac{2}{3} \overset{\times 3}{=} \frac{6}{9}$

Tem-se que $F_{1A} = F_{1C} > F_{1B}$, logo a classe com menor valor de $F_1$, é a classe B.

4) Spearman $(Y_1, Y_2) = ?$

| D | $Y_1$ | $Y_2$ | rank $Y_1$ | rank $Y_2$ |
|---|---|---|---|---|
| $X_1$ | 0.24 | 1 | 10 | 5 |
| $X_2$ | 0.06 | 2 | 11 | 2 |
| $X_3$ | 0.04 | 0 | 12 | 9.5 |
| $X_4$ | 0.36 | 0 | 8 | 9.5 |
| $X_5$ | 0.32 | 0 | 9 | 9.5 |
| $X_6$ | 0.68 | 2 | 3 | 2 |
| $X_7$ | 0.9 | 0 | 1 | 9.5 |
| $X_8$ | 0.76 | 2 | 2 | 2 |
| $X_9$ | 0.46 | 1 | 7 | 5 |
| $X_{10}$ | 0.62 | 0 | 4 | 9.5 |
| $X_{11}$ | 0.44 | 1 | 6 | 5 |
| $X_{12}$ | 0.52 | 0 | 5 | 9.5 |

- Spearman $(Y_1, Y_2)$ = Pearson ([10, 11, 12, 8, 9, 3, 1, 2, 7, 4, 6, 5], [5, 2, 9.5, 9.5, 9.5, 2, 9.5, 2, 5, 9.5, 5, 9.5]) = ⊛
- $Pearson(Y_1, Y_2) = \frac{Cov(Y_1, Y_2)}{\sqrt{Var(Y_1) Var(Y_2)}}$ $\quad \bar{r}_{Y_1} = \frac{\sum_{i=1}^{12} i}{12} = 6.5$
- $Var(rank\ Y_1) = \frac{\sum_{i=1}^{m} (Y_i - \bar{Y})^2}{m-1} = \frac{\sum_{i=1}^{12} (Y_i)^2 - 12 \times \bar{Y}_1^2}{11} = \frac{143}{11} = 13$ $\quad \bar{r}_{Y_2} = \frac{5 \times 3 + 9.5 \times 6}{12} + 2 \times 3 \simeq 6.5$

Aprendizagem 2023/24

**Homework I – Group 075**

(ist1103860, ist1103883)

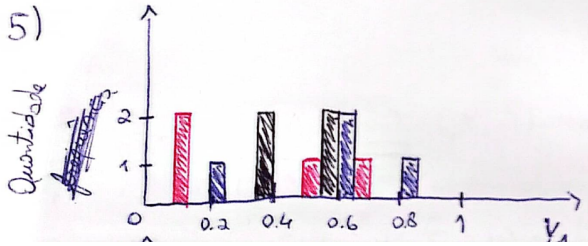
- $Var(rank_{Y_2}) = \frac{628.5 - \frac{507}{11}}{11} \simeq 11.045$

- $Cov(rank_{Y_1}, rank_{Y_2}) = \frac{\sum_{i=1}^{12} (Y_{1i} - \bar{Y}_1)(Y_{2i} - \bar{Y}_2)}{11} = \frac{(10-6.5)(5-6.5) + (11-6.5)(2-6.5) + (12-6.5)(9.5-6.5) + (8-6.5)(9.5-6.5) + (9-6.5)(9.5-6.5) + (3-6.5)(2-6.5) + (1-6.5)(9.5-6.5) + (2-6.5)(2-6.5) + (7-6.5)(5-6.5) + (4-6.5)(9.5-6.5) + (6-6.5)(5-6.5) + (5-6.5)(9.5-6.5)}{11} \simeq 9.955$

$$Spearman(Y_1, Y_2) = \frac{9.955}{\sqrt{13 \times 11.045}} \simeq 0.0797$$

Logo, podemos dizer que existe uma correlação forte entre $Y_1$ e $Y_2$.

5)




Legenda: A, B, C

]0;0,2[ → B

]0.2;0,6[ → C

]0.6;1[ → A

Divisão da Roda

**Homework I – Group 075**

(ist1103860, ist1103883)

## II. Programming and critical analysis

**1)**

```
from sklearn.feature_selection import f_classif
from scipy.io.arff import loadarff
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Reading the ARFF file
data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

# Separate features from the outcome (class)
X = df.drop('class', axis=1)
y = df['class']

fimportance = f_classif(X, y)  #0 - fvalue, 1 - pvalue

highest_index = fimportance[0].argmax()
lowest_index = fimportance[0].argmin()

highest_power, lowest_power = X.columns[highest_index], X.columns[lowest_index]

print('Input variable with the highest discriminative power:',highest_power)
print('Input variable with the lowest discriminative power:', lowest_power)

# Plot class-conditional probability density functions
# Plot highest discriminative power feature
classes = df['class'].unique()

plt.figure(figsize=(10, 6))
for target_class in classes:
        subset = df[df['class'] == target_class]
        sns.kdeplot(subset[highest_power], label=f'Class {target_class}')
plt.xlabel(highest_power)
plt.ylabel('Density')
plt.title(f'Class-Conditional Probability Density for {highest_power}')
plt.legend(title='Class')
plt.show()

# Plot lowest discriminative power feature
```

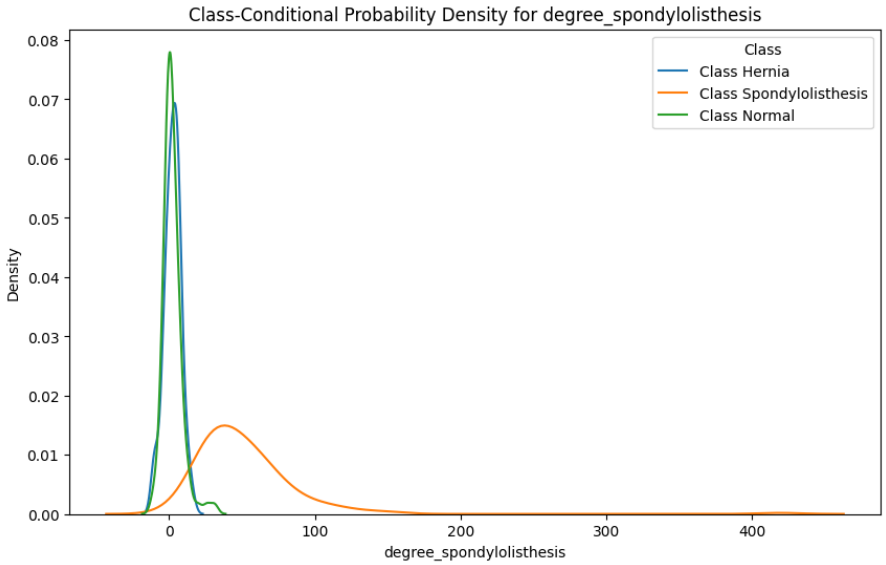
**Homework I – Group 075**

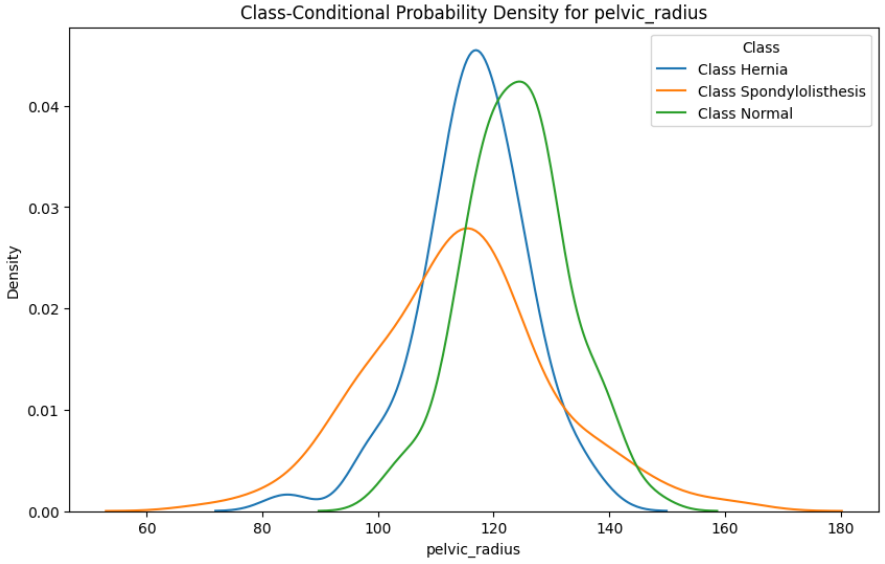
(ist1103860, ist1103883)

```
plt.figure(figsize=(10, 6))
for target_class in classes:
        subset = df[df['class'] == target_class]
        sns.kdeplot(subset[lowest_power], label=f'Class {target_class}')
plt.xlabel(lowest_power)
plt.ylabel('Density')
plt.title(f'Class-Conditional Probability Density for {lowest_power}')
plt.legend(title='Class')
plt.show()
```

Input variable with the highest discriminative power: degree_spondylolisthesis

Input variable with the lowest discriminative power: pelvic_radius



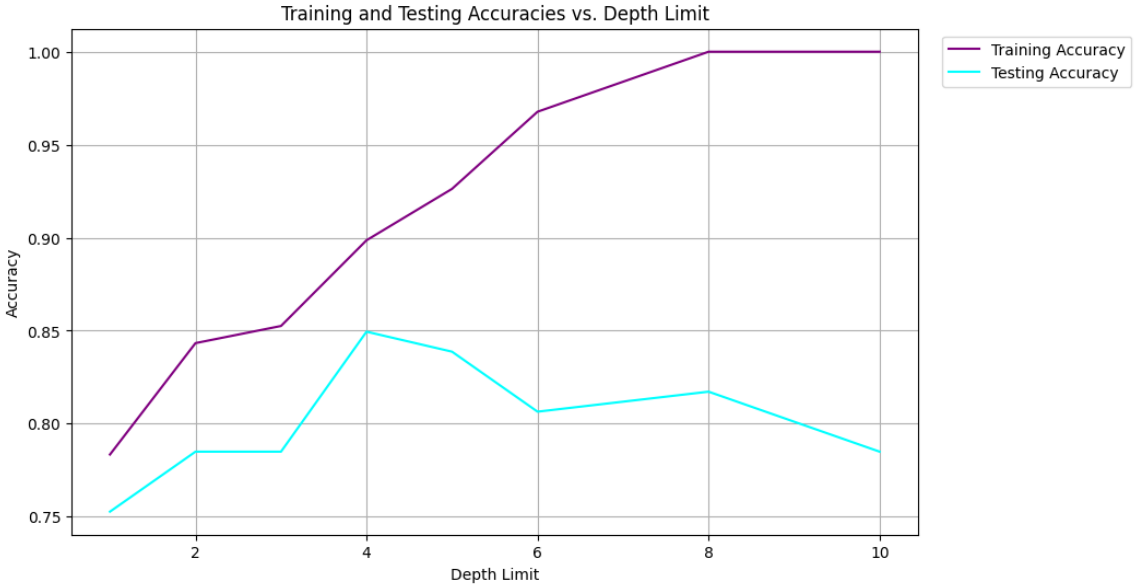
**Homework I – Group 075**

(ist1103860, ist1103883)

**2)**

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from scipy.io.arff import loadarff
import pandas as pd
# Reading the ARFF file
data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

# Separate features from the outcome (class)
X = df.drop('class', axis=1)
y = df['class']

depth_limits = [1, 2, 3, 4, 5, 6, 8, 10]

# Split the dataset into a training set (70%) and a testing set (30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state = 0, stratify=y)

training_accuracies, testing_accuracies = [], []

for depth in depth_limits:
        # learn classifier using hold-out partitioning
        predictor = DecisionTreeClassifier(max_depth=depth)
        predictor.fit(X_train, y_train)

        # Predictions on training and testing sets
        y_train_pred = predictor.predict(X_train)
        y_test_pred = predictor.predict(X_test)

        # Calculate training and testing accuracies and append them to the respective lists
        training_accuracies.append(accuracy_score(y_train, y_train_pred))
        testing_accuracies.append(accuracy_score(y_test, y_test_pred))

#draw plot
plt.figure(figsize=(10, 6))
plt.plot(depth_limits, training_accuracies, label='Training Accuracy', color= 'purple')
plt.plot(depth_limits, testing_accuracies, label='Testing Accuracy', color = 'cyan')
plt.xlabel('Depth Limit')
plt.ylabel('Accuracy')
```

```
plt.title('Training and Testing Accuracies vs. Depth Limit')
plt.legend(loc='upper left', bbox_to_anchor=(1.02, 1))
plt.grid(True)
plt.show()
```




**3)**

Analisando a **training accuracy**, podemos ver que ela aumenta à medida que o limite de profundidade da árvore aumenta. O que demonstra, que a nossa árvore de decisão, tem um bom desempenho nos treinos que efetuou.

Em relação à **testing accuracy**, ela acaba também por aumentar à medida que o limite de profundidade aumenta, que indica que o modelo generaliza melhor mesmo com o aumento da complexidade do modelo. Contudo, por volta do limite de profundidade 4, a **testing accuracy** fica constante, sendo que depois até diminui.

Esta descida na **testing accuracy**, indica-nos que estamos com o problema de **overfitting**. O modelo tem um bom desempenho nos treinos, mas não consegue generalizar bem para dados não vistos, acabando por reduzir a **testing accuracy**.

O pico da **testing accuracy** (com limite de profundidade 4) representa a capacidade de generalização óptima do modelo, pois para além deste ponto começa a ocorrer overfitting e a complexidade do modelo aumenta provocando uma diminuição da **testing accuracy.**

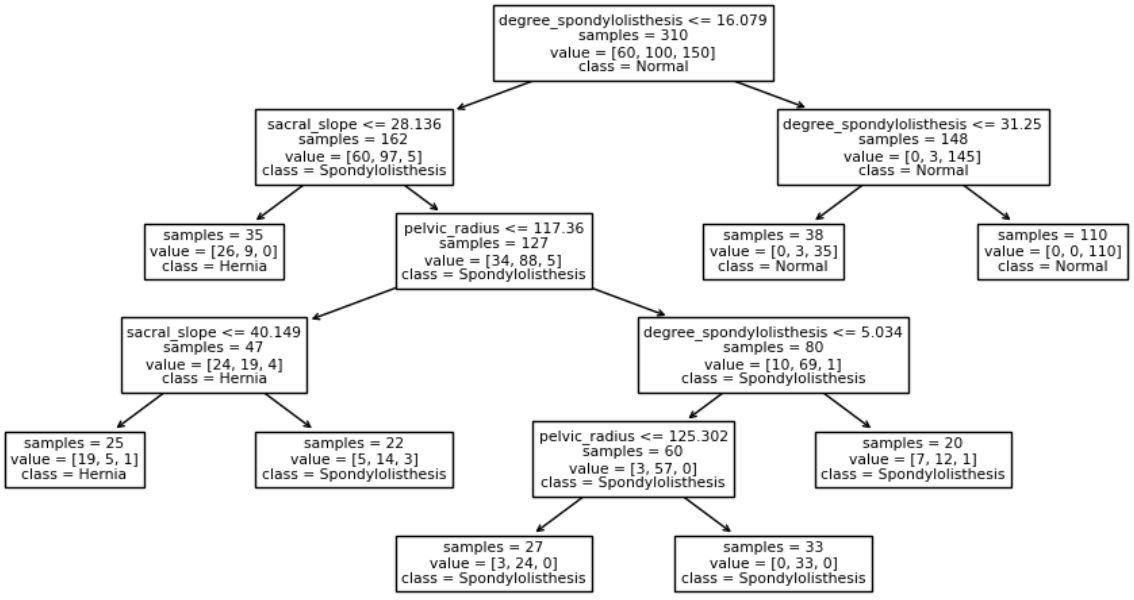
Assim, para obter a melhor capacidade de generalização, é essencial selecionar um limite de profundidade adequado que equilibre a complexidade do modelo e o desempenho em dados não vistos. Um limite de profundidade de cerca de 4 parece ser ideal, resultando no maior equilíbrio entre precisão de teste e generalização para dados não vistos.

**Homework I – Group 075**

(ist1103860, ist1103883)

**4) i)**

```
from scipy.io.arff import loadarff
import pandas as pd
from sklearn.tree import plot_tree, DecisionTreeClassifier
import matplotlib.pyplot as plt

# Reading the ARFF file
data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')

# Separate features from the outcome (class)
X = df.drop('class', axis=1)
y = df['class']

# learn classifier
predictor = DecisionTreeClassifier(min_samples_leaf=20, random_state=0)
predictor.fit(X, y)

class_names = df['class'].unique()

figure = plt.figure(figsize=(12, 6))
plot_tree(predictor, feature_names=X.columns,
class_names=class_names, impurity=False)
plt.show()
```



**ii)**

- Se o grau de espondilolistese for menor ou igual que 16.079, e o declive sacral for menor ou igual que 28.136, então o indivíduo deve ter uma hérnia.
- Caso o grau de espondilolistese seja menor ou igual que 16.079, e o declive sacral esteja entre 28.136 e 40.149(inclusivé), e o seu raio da bacia seja inferior ou igual a 117.36, o indivíduo deve uma hérnia.