Homework I – Group 029

(ist1103000, ist1103179)

I. Pen-and-paper

1

D	y_1	y_2	y_3	y_4	y_{out}
x_1	0.24	1	1	0	A
x_2	0.06	2	0	0	B
x_3	0.04	0	0	0	B
x_4	0.36	0	2	1	C
x_5	0.32	0	0	2	C
x_6	0.68	2	2	1	A
x_7	0.9	0	1	2	A
x_8	0.76	2	2	0	A
x_9	0.46	1	1	1	B
x_{10}	0.62	0	0	1	B
x_{11}	0.44	1	2	2	C
x_{12}	0.52	0	2	0	C

$y_1 > 0.4$

class variable

To complete the given decision tree, we first need to:

I. compute the entropy of the class variable
II. compute the weighted entropy for each variable
III. compute the information gain
IV. Draw part of the tree and look for uncertainties — if found, repeat the process using a new table

$y_1 > 0.4$:

$$H(y_{out}) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right)\times 2 \simeq 1.557$$

$$H(y_{out}|y_2) = \frac{3}{7}\times\left(-\frac{1}{3}\log_2\left(\frac{1}{3}\right)\times 3\right) + \frac{2}{7}\times\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\times 2\right) + \underbrace{\frac{2}{7}\times(-\log_2 1)}_{0} \simeq 0.965$$

$$H(y_{out}|y_3) = \frac{1}{7}\times(-\log_2 1) + \frac{2}{7}\times\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\times 2\right) + \frac{4}{7}\times\left(-\frac{2}{4}\log_2\left(\frac{2}{4}\right)\times 2\right) \simeq 0.857$$

$$H(y_{out}|y_4) = \frac{2}{7}\times\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\times 2\right) + \frac{3}{7}\times\left(-\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{2}{7}\times\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\times 2\right) = 0.965$$

$IG(y_{out}|y_2) = 1.557 - 0.965 = 0.592$

$IG(y_{out}|y_3) = 1.557 - 0.857 = 0.7$ ← y_3 has the highest information gain and hence it is chosen

$IG(y_{out}|y_4) = 1.557 - 0.965 = 0.592$

Since there is some uncertainty regarding the 4 observations in which $y_3 = 2$, we repeat the process:

D	y_1	y_2	y_3	y_4	y_{out}
x_6	0.68	2	2	1	A
~~x_7~~	~~0.9~~	~~0~~	~~1~~	~~2~~	~~A~~
x_8	0.76	2	2	0	A
x_9	0.46	1	1	1	B
x_{10}	0.62	0	0	1	B
x_{11}	0.44	1	2	2	C
x_{12}	0.52	0	2	0	C

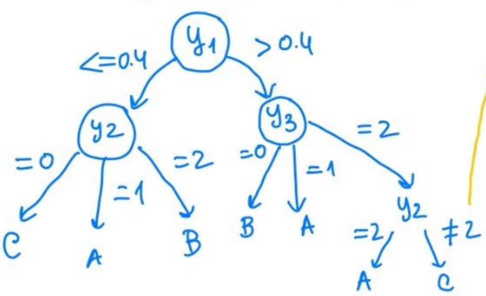
$$H(y_{out}|y_1>0.4 \wedge y_3=2 \wedge y_2) = \underbrace{\frac{1}{4}\times(-\log_2 1)\times 2}_{0} + \underbrace{\frac{2}{4}\times(-\log_2 1)}_{0} = 0$$

$$H(y_{out}|y_1>0.4 \wedge y_3=2 \wedge y_4) = \frac{2}{4}\times\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\times 2\right) + \underbrace{\frac{1}{4}\times(-\log_2 1)\times 2}_{0} = \frac{1}{2}$$

$$H(y_{out})' = -\frac{2}{4}\log_2\left(\frac{2}{4}\right)\times 2 = 1$$

$IG(y_{out}|y_1>0.4 \wedge y_3=2 \wedge y_2) = 1 - 0 = 1$ → y_2 has the highest information gain and hence it is chosen

$IG(y_{out}|y_1>0.4 \wedge y_3=2 \wedge y_4) = 1 - \frac{1}{2} = \frac{1}{2}$



Note: we get C when $y_1 > 0.4$, $y_3 = 2$ and $y_2 = 0$ or $y_1 > 0.4$, $y_3 = 2$ and $y_2 = 1$

Homework I – Group 029

(ist1103000, ist1103179)

2)

$\hat{y}_{out} = [A, B, C, C, C, A, A, A, A, B, C, C]$

$y_{out} = [A, B, B, C, C, A, A, A, B, B, C, C]$

$$\begin{array}{c|ccc} & & \text{expected} & \\ \text{predicted} & A & B & C \\ \hline A & 4 & 1 & 0 \\ B & 0 & 2 & 0 \\ C & 0 & 1 & 4 \end{array}$$

3)

$f_1\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$; $\text{precision} = \frac{TP}{TP + FP}$; $\text{recall} = \frac{TP}{TP + FN}$

class A : $\text{precision}_A = \frac{4}{5}$; $\text{recall}_A = \frac{4}{4} = 1$; $f_1\text{-score}_A = 2 \times \frac{\frac{4}{5} \times 1}{\frac{4}{5} + 1} = 2 \times \frac{4}{\cancel{5}} \times \frac{\cancel{5}}{9} = \frac{8}{9} \simeq 0.889$

class B : $\text{precision}_B = \frac{2}{2} = 1$; $\text{recall}_B = \frac{2}{4} = \frac{1}{2}$; $f_1\text{-score}_B = 2 \times \frac{1 \times \frac{1}{2}}{1 + \frac{1}{2}} = \frac{2}{3} \simeq 0.667$

class C : $\text{precision}_C = \frac{4}{5}$; $\text{recall}_C = 1$; $f_1\text{-score}_C = 2 \times \frac{1 \times \frac{4}{5}}{1 + \frac{4}{5}} = \frac{8}{\cancel{5}} \times \frac{\cancel{5}}{9} = \frac{8}{9} \simeq 0.889$

R: class B has the lowest f_1-score.

4)

y_1	y_2	Ry_1	Ry_2
0.24	1	3	8
0.06	2	2	11
0.04	0	1	3.5
0.36	0	5	3.5
0.32	0	4	3.5
0.68	2	10	11
0.9	0	12	3.5
0.76	2	11	11
0.46	1	7	8
0.62	0	9	3.5
0.44	1	6	8
0.52	0	8	3.5

$$\text{spearman}(y_1, y_2) = \overbrace{\frac{\sum_i (RY_{1i} - \mu_{RY1}) \times (RY_{2i} - \mu_{RY2})}{\sqrt{\sum_i (RY_{1i} - \mu_{RY1})^2} \times \sqrt{\sum_i (RY_{2i} - \mu_{RY2})^2}}}^{PCC(RY_1, RY_2)} =$$

$$= \frac{10.5}{\sqrt{143} \times \sqrt{121.5}} \simeq 0.08$$

very weak correlation

$\mu_{RY1} = 6.5$

$\mu_{RY2} = 6.5$

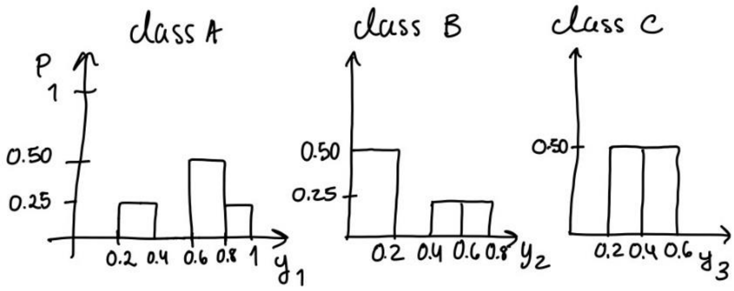
$\sum_i (RY_{1i} - \mu_{RY1}) \times (RY_{2i} - \mu_{RY2}) = 10.5$

$\sum_i (RY_{1i} - \mu_{RY1})^2 = 143$

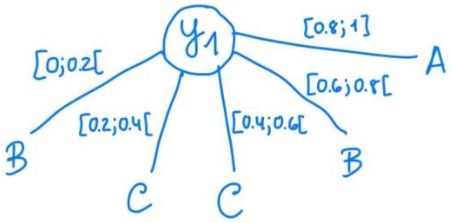
$\sum_i (RY_{2i} - \mu_{RY2})^2 = 121.5$

5)

	A	B	C
[0;0.2[	0	2	0
[0.2;0.4[	1	0	2
[0.4;0.6[	0	1	2
[0.6;0.8[	2	1	0
[0.8;1]	1	0	0



challenge :



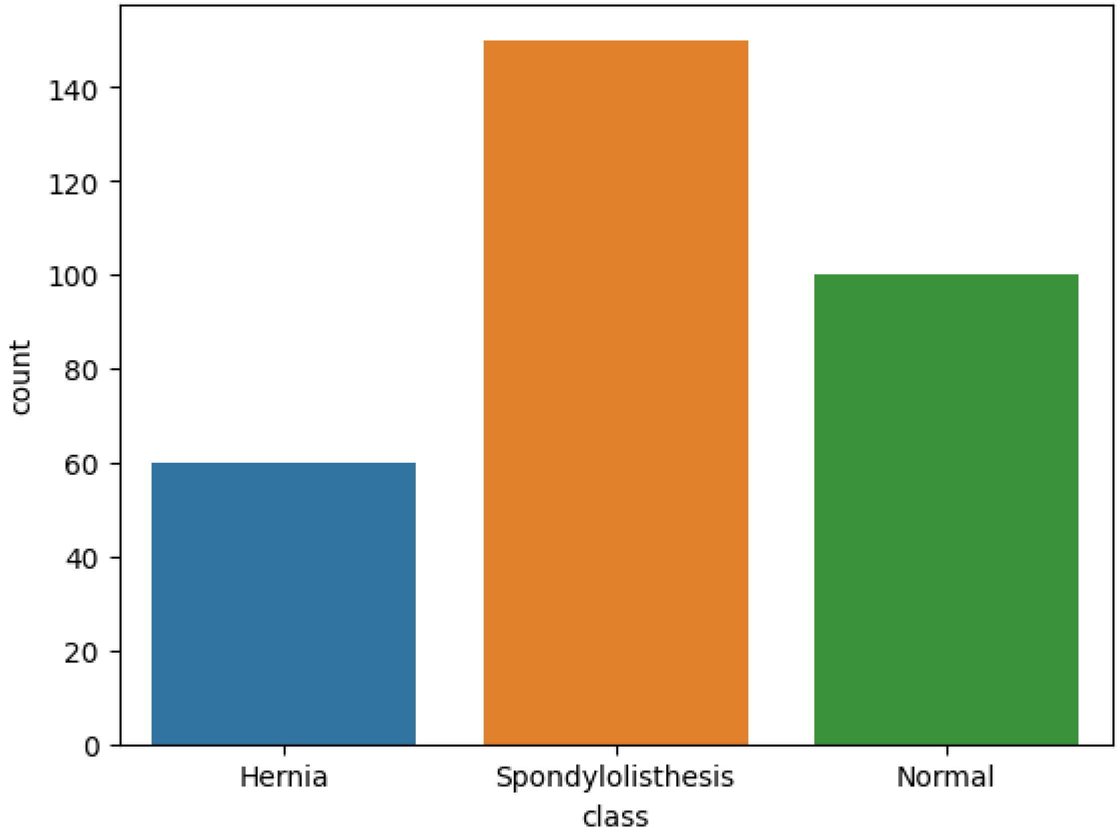
II. Programming and critical analysis

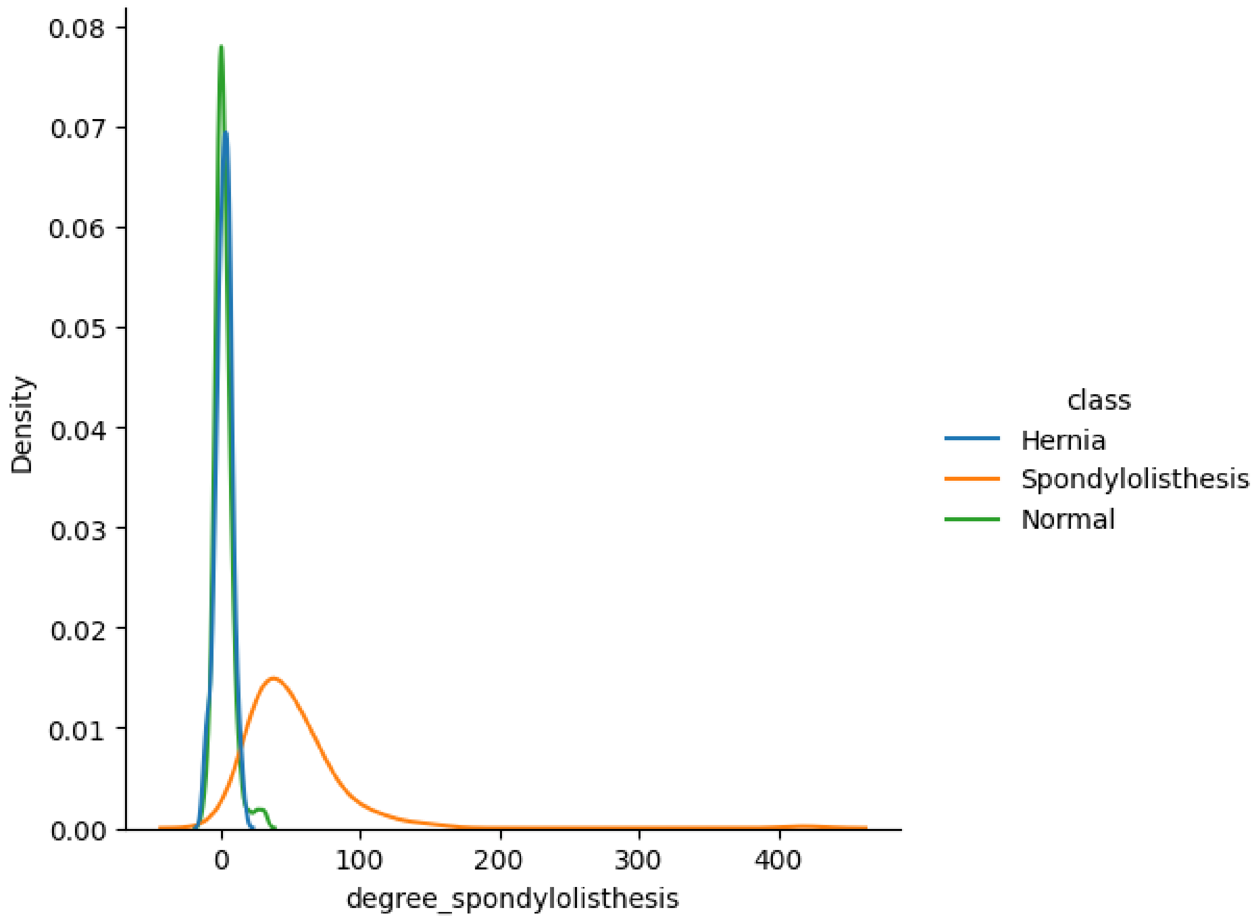
1) Code and graphs:

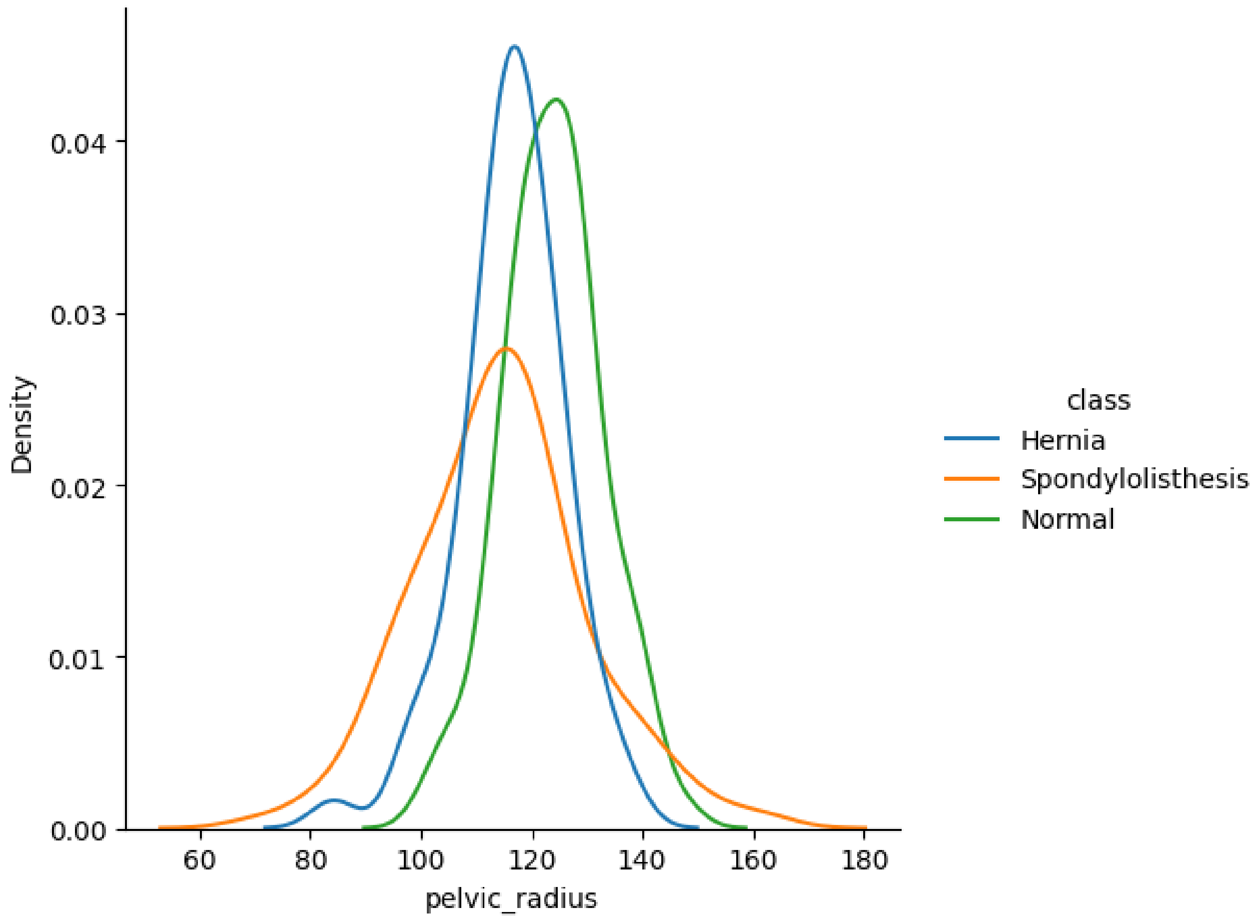
```
import pandas as pd
from scipy.io.arff import loadarff

data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df.head()

from sklearn.feature_selection import f_classif

input_vars = df.drop("class", axis=1)
output_vars = df["class"]

fimportance = f_classif(input_vars, output_vars)

print('features', input_vars.columns.values)
print('scores', fimportance[0])
print('pvalues', fimportance[1])


import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x='class', data=df)
plt.show()

sns.displot(df, x="degree_spondylolisthesis", hue="class", kind="kde", common_norm=False)

sns.displot(df, x="pelvic_radius", hue="class", kind="kde", common_norm=False)
```

Highest discriminative power: degree_spondylolisthesis.

Lowest discriminative power: pelvic_radius.


count
140
120
100
80
60
40
20
0
Hernia
Spondylolisthesis
Normal
class

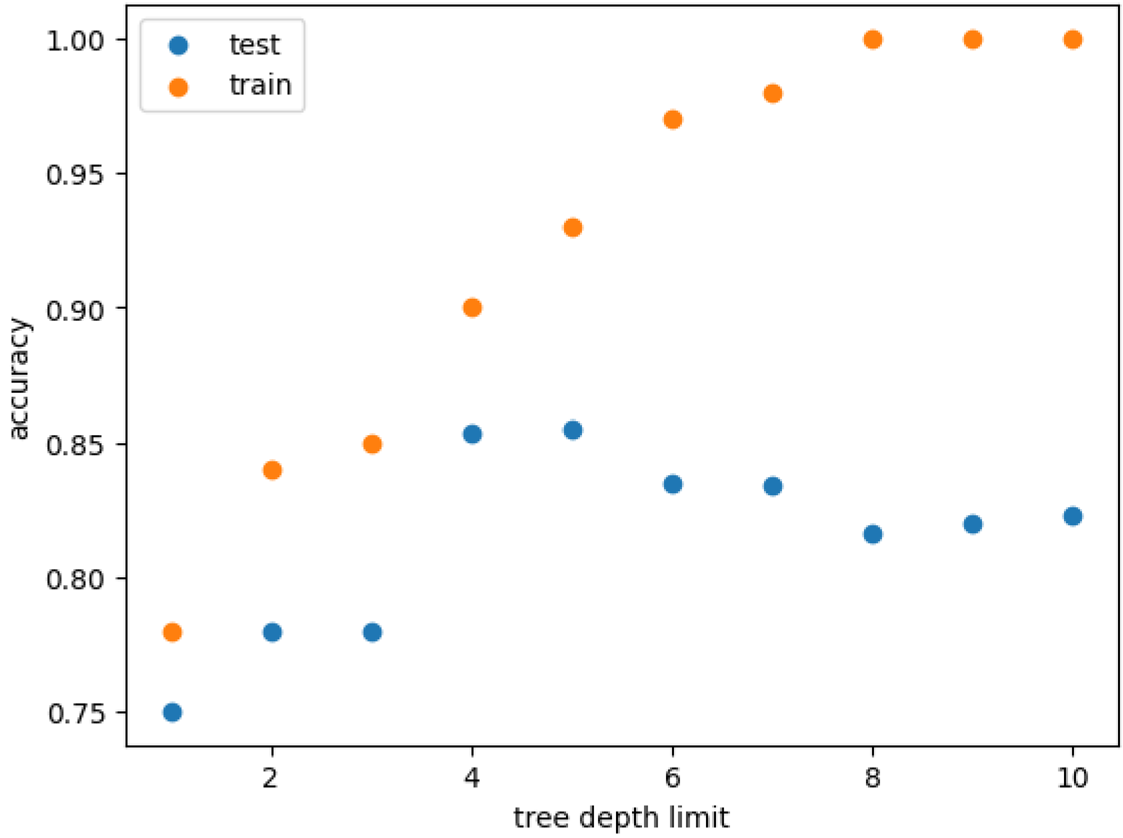

Density
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
degree_spondylolisthesis
class
Hernia
Spondylolisthesis
Normal



2) Code and graphs:

```
from sklearn import metrics, datasets, tree
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(input_vars, output_vars, train_size=0.7,
stratify=output_vars, random_state=0)
print("#training obs =",len(X_train),"\n#testing obs =",len(X_test))

depth_limits = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]     # tree depths
accuracies_test = []                                # list of accuracies of the test set for each tree depth
accuracies_train = []                               # list of accuracies of the training set for each tree depth

for depth in depth_limits:     # makes a decsion tree for every depth from 1 to 10
    test_sum, train_sum = 0, 0
    for _ in range(10):
        predictor = tree.DecisionTreeClassifier(max_depth=depth)
        predictor.fit(X_train, y_train)

        y_pred_test = predictor.predict(X_test)
        y_pred_train = predictor.predict(X_train)
```

```
        test_sum += round(metrics.accuracy_score(y_test, y_pred_test),2)
        train_sum += round(metrics.accuracy_score(y_train, y_pred_train),2)

    accuracies_test.append(test_sum/10)
    accuracies_train.append(train_sum/10)

plt.scatter(depth_limits, accuracies_test, label='test')
plt.scatter(depth_limits, accuracies_train, label='train')
plt.xlabel("tree depth limit")
plt.ylabel("accuracy")
plt.legend()
plt.show()
```



3) Considering the previous plot, we can assess that the deeper the tree reaches, the more accurate the classifications become as far as the accuracy of the training set is concerned. But in terms of the accuracy of the test set, the association between tree depth and accuracy is noticeably weaker, probably due to overfitting when the tree goes too deep, and we therefore observe that the results tend to be the most accurate around a maximum depth of 5. For that reason, the classification seems to have a better generalization capacity around that depth.

4)

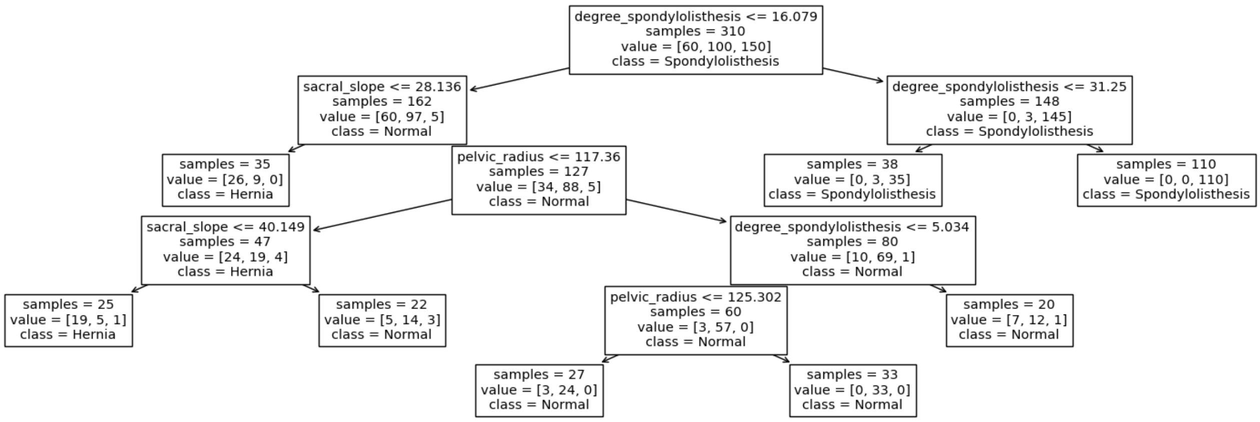
Aprendizagem 2023/24

Homework I – Group 029

(ist1103000, ist1103179)

i) Code and graphs:

```
new_predictor = tree.DecisionTreeClassifier(min_samples_leaf=20)
new_predictor.fit(input_vars, output_vars)

figure = plt.figure(figsize=(20, 6))
tree.plot_tree(new_predictor, feature_names=list(input_vars.columns.values),
class_names=["Hernia", "Normal", "Spondylolisthesis"], impurity=False)
plt.show()
```



ii) A hernia condition may be attributed if the following conditional associations are met:

a. degree_spondylolisthesis $\leq$ 16.079, sacral_slope $\leq$ 28.136
b. degree_spondylolisthesis $\leq$ 16.079, sacral_slope $>$ 28.136, pelvic_radius $\leq$ 117.36, sacral_slope $\leq$ 40.149