Aprendizagem 2023
Homework I – Group 60
Maria João Rosa, Mariana Miranda
ist1102506, ist1102904

Part I: Pen and paper

1. Complete the given decision tree using Information gain with Shannon entropy (log_2). Consider that: i) a minimum of 4 observations is required to split an internal node, and ii) decisions by ascending alphabetic order should be placed in case of ties.

	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

Para a escolha da variável, utiliza-se o ganho de informação

$$IG(y_i) = H(y_{out}) - E(y_{out} \mid y_i, y_1 > 0.4)$$

$$H(y_{out}) = -\sum P_{y_{out}} \log_2 P_{y_{out}}$$
$$= -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{2}{7}\log_2\frac{2}{7} - \frac{3}{7}\log_2\frac{3}{7}\right)$$
$$\simeq 1,5567$$

$$H(y_{out} \mid y_2, y_1 > 0.4) = \frac{3}{7}\left(-\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{1}{3}\log_2\frac{1}{3} + \frac{1}{3}\log_2\frac{1}{3}\right)\right) + \frac{2}{7}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) + \frac{2}{7}\left(-\cancel{\log_2 1}^{\,0}\right) \simeq 0,9650$$

$$IG(y_{out} \mid y_2, y_1 > 0.4) = 1,5567 - 0,9650 = 0,5917$$

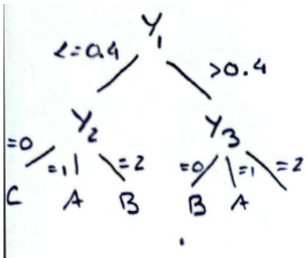
$$H(y_{out} \mid y_3, y_1 > 0.4) = \frac{1}{7}(-1\log_2 1) + \frac{2}{7}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) + \frac{4}{7}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) \simeq 0,8571$$

$$IG(y_{out} \mid y_3, y_1 > 0.4) = 1,5567 - 0,8571 = \boxed{0,6996}$$

$$H(y_{out} \mid y_4, y_1 > 0.4) = \frac{2}{7}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) + \frac{3}{7}\left(-\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right)\right) + \frac{2}{7}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) \simeq 0,9650$$

$$IG(y_{out} \mid y_4, y_1 > 0.4) = 1,5567 - 0,9650 = 0,5917$$

A variável y_3 é a que apresenta um maior ganho de informação



- Por observação da tabela, existe uma observação $Y_3 = 0$ onde $Y_{out} = B$

- Existem 2 observações com $Y_3 = 1$, que levam a $Y_{out} = A$ e $Y_{out} = B$, escolhendo por ordem alfabética a folha fica com A

- Existem 4 observações com $Y_3 = 2$, sendo este ramo o proximo a expandir

$$H(Y_{out}) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

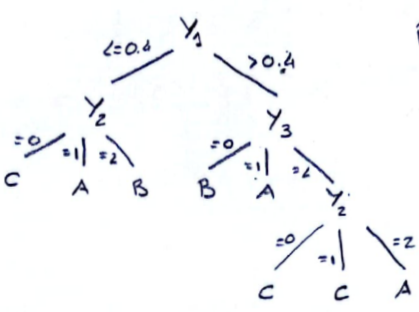
$$H(Y_{out} \mid Y_2, Y_1 > 0.4, Y_3 = 2) = \frac{1}{4}\left(-(1\log_2 1)\right) + \frac{1}{4}(-1\log_2 1) + \frac{1}{2}(1\log_2 1) = 0$$

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

$$H(Y_{out} \mid Y_4, Y_1 > 0.4, Y_3 = 2) = \frac{1}{2}\left(-\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right)\right) + \frac{1}{4}(-1\log_2 1) + \frac{1}{4}(-1\log_2 1) = 0.5$$

$$IG(Y_{out} \mid Y_4, Y_1 > 0.4, Y_3 = 2) = 1 - 0.5 = 0.5$$

Y_2 tem o maior ganho do informação



Por observação da tabela
para $Y_2 = 0$ o output é C
para $Y_2 = 1$ o output é C
para $Y_2 = 2$ o output é A

2. Draw the training confusion matrix for the learnt decision tree.

D	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
Real	A	B	B	C	C	A	A	A	B	B	C	C
Previsão	A	B	C	C	C	A	A	A	A	B	C	C

Previsão \ Real	A	B	C
A	4	1	0
B	0	2	0
C	0	1	4

Matriz de Confusão

3. Identify which class has the lowest training F1 score.

$$\text{Precision}_A \ (P_A) = \frac{TP_A}{TP_A + TF_A} = \frac{4}{4+1} = \frac{4}{5}$$

$$\text{Recall}_A \ (R_A) = \frac{TP_A}{TP_A + FN_A} = \frac{4}{4} = 1$$

$$\text{F1-score}_A = 2 \times \frac{4/5 \times 1}{4/5 + 1} = \frac{8}{9}$$

A fórmula do F1 score para $\beta = 1$ e $\alpha = 0,5$ é:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision}_B = \frac{TP_B}{TP_B + TF_B} = \frac{2}{2} = 1$$

$$\text{Recall}_B = \frac{TP_B}{TP_B + FN_B} = \frac{2}{1+2+1} = \frac{1}{2}$$

$$\text{F1-score}_B = 2 \times \frac{1 \times 1/2}{1 + 1/2} = \frac{2}{3}$$

$$\text{Precision}_C = \frac{TP_C}{TP_C + TF_C} = \frac{4}{1+4} = \frac{4}{5}$$

$$\text{Recall}_C = \frac{TP_C}{TP_C + FN_C} = \frac{4}{4} = 1$$

$$\text{F}_1\text{-score}_C = 2 \times \frac{4/5 \times 1}{4/5 + 1} = \frac{8}{9}$$

A classe B é a que apresenta menor F1 score

4. Considering y_2 to be ordinal, assess if y_1 and y_2 are correlated using the Spearman coefficient.

	Y_1	Y_2	$R(Y_1)$	$R(Y_2)$	(1) $R(Y_1)-\overline{R(Y_1)}$	(2) $R(Y_2)-\overline{R(Y_2)}$	(1) × (2)
x_1	0.24	1	3	8	-3,5	1,5	-5.25
x_2	0.06	2	2	11	-4,5	4,5	-20,25
x_3	0.04	0	1	3,5	-5,5	-3	16,50
x_4	0.36	0	5	3,5	-1,5	-3	4,50
x_5	0.32	0	4	3,5	-2,5	-3	7,50
x_6	0.68	2	10	11	3,5	4,5	15,75
x_7	0.9	0	12	3,5	5,5	-3	-16,50
x_8	0.76	2	11	11	4,5	4,5	20,25
x_9	0.46	1	7	8	0,5	1,5	0,75
x_{10}	0.62	0	9	3,5	2,5	-3	-7,50
x_{11}	0.44	1	6	8	-0,5	1,5	-0,75
x_{12}	0.52	0	8	3,5	1,5	-3	-4,50
							10,5

$\overline{R(Y_1)} = 6.5$

$\overline{R(Y_2)} = 6.5$

$$r_s(Y_1, Y_2) = \frac{cov(R(Y_1), R(Y_2))}{\sqrt{var(R(Y_1))}\sqrt{var(R(Y_2))}}$$

$$var(R(Y_1)) = \sum_{i=1}^{n} \frac{(R(Y_1) - \overline{R(Y_1)})^2}{n-1} = \frac{143}{11}$$

$$var(R(Y_2)) = \sum_{i=1}^{n} \frac{(R(Y_2) - \overline{R(Y_2)})^2}{n-1} = \frac{121,5}{11}$$

$$cov(R(Y_1), R(Y_2)) = \sum_{i=1}^{n} \frac{(R(Y_1) - \overline{R(Y_1)})(R(Y_2) - \overline{R(Y_2)})}{n-1}$$

$$= \frac{10,5}{11} = 0,95$$

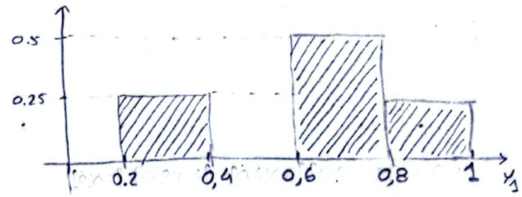
$$r_s(Y_1, Y_2) = \frac{0,95}{\sqrt{\frac{143}{11}} \times \sqrt{\frac{121,5}{11}}} = 0,0797$$

Como o valor de correlação entre as duas variáveis está muito próximo de 0, é possível concluir que as variáveis y_1 e y_2 estão pouco relacionadas.

5. Draw the class-conditional relative histograms of y_1 using 5 equally spaced bins in $[0, 1]$.

	y_1	Y_{out}
x_1	0.24	A
x_6	0.68	A
x_7	0.9	A
x_8	0.76	A

bins: [0; 0.2[[0.2; 0.4[[0.4; 0.6[[0.6; 0.8[[0.8; 1]



$h_{bin2} = \frac{1}{4} = 0.25$

$h_{bin4} = \frac{2}{4} = 0.5$

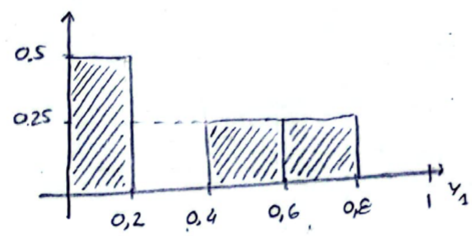
$h_{bin5} = \frac{1}{4} = 0.25$

	y_1	Y_{out}
x_2	0.06	B
x_3	0.04	B
x_9	0.46	B
x_{10}	0.62	B

$h_{bin1} = \frac{2}{4} = 0.5$

$h_{bin3} = \frac{1}{4} = 0.25$

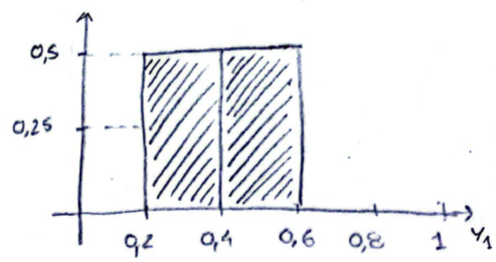
$h_{bin4} = \frac{1}{4} = 0.25$



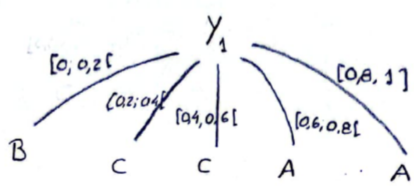
	y_1	Y_{out}
x_4	0.36	C
x_5	0.32	C
x_{11}	0.44	C
x_{12}	0.55	C

$h_{bin2} = \frac{2}{4} = 0.5$

$h_{bin3} = \frac{2}{4} = 0.5$



Challenge: find the root split using the discriminant rules from these empirical distributions.



Pela observação dos histogramas condicionados à classe é possível criar uma árvore de decisão de um só nível onde cada ramo corresponde ao intervalo de um bin (0,2) e a folha, à classe com maior proporção dos 3 histogramas.

Part II: Programming

1. Apply `f_classif` from `sklearn` to assess the discriminative power of the input variables. Identify the input variable with the highest and lowest discriminative power. Plot the class-conditional probability density functions of these two input variables.

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

data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])

f_values = f_classif(df.iloc[:, :-1], df.iloc[:, -1])[0]

# Variable with lowest discriminative power
lowest = df.columns[f_values.argmin()]

plt.figure(figsize=(10, 5))
plt.hist(df[lowest][df['class'] == b'Hernia'], alpha=0.2, color='blue', density=True)
plt.hist(df[lowest][df['class'] == b'Spondylolisthesis'], alpha=0.2, color='orange', density=True)
plt.hist(df[lowest][df['class'] == b'Normal'], alpha=0.2, color='green', density=True)
df[lowest][df['class'] == b'Hernia'].plot(kind='kde', label='Hernia', color='blue')
df[lowest][df['class'] == b'Spondylolisthesis'].plot(kind='kde', label='Spondylolisthesis', color='orange')
df[lowest][df['class'] == b'Normal'].plot(kind='kde', label='Normal', color='green')
plt.xlabel('Pelvic Radius')
plt.ylabel('Probability Density')
plt.xlim(50, 190)
plt.legend()
plt.show()

# Variable with highest discriminative power
highest = df.columns[f_values.argmax()]

plt.figure(figsize=(10, 5))
plt.hist(df[highest][df['class'] == b'Hernia'], alpha=0.2, color='blue', density=True)
plt.hist(df[highest][df['class'] == b'Spondylolisthesis'], alpha=0.2, color='orange', density=True)
plt.hist(df[highest][df['class'] == b'Normal'], alpha=0.2, color='green', density=True)
```

```
df[highest][df['class'] == b'Hernia'].plot(kind='kde', label='Hernia', color='blue')
df[highest][df['class'] == b'Spondylolisthesis'].plot(kind='kde', label='Spondylolisthesis', color='orange')
df[highest][df['class'] == b'Normal'].plot(kind='kde', label='Normal', color='green')
plt.xlabel('Degree of Spondylolisthesis')
plt.ylabel('Probability Density')
plt.xlim(-100, 500)
plt.legend()
plt.show()
```

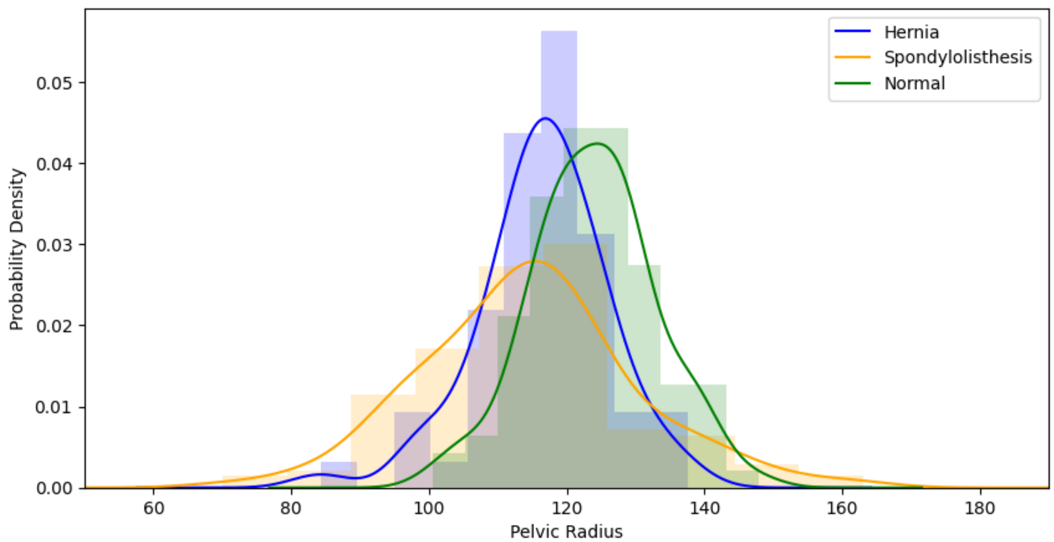


Figure 1: Variable with lowest discriminative power

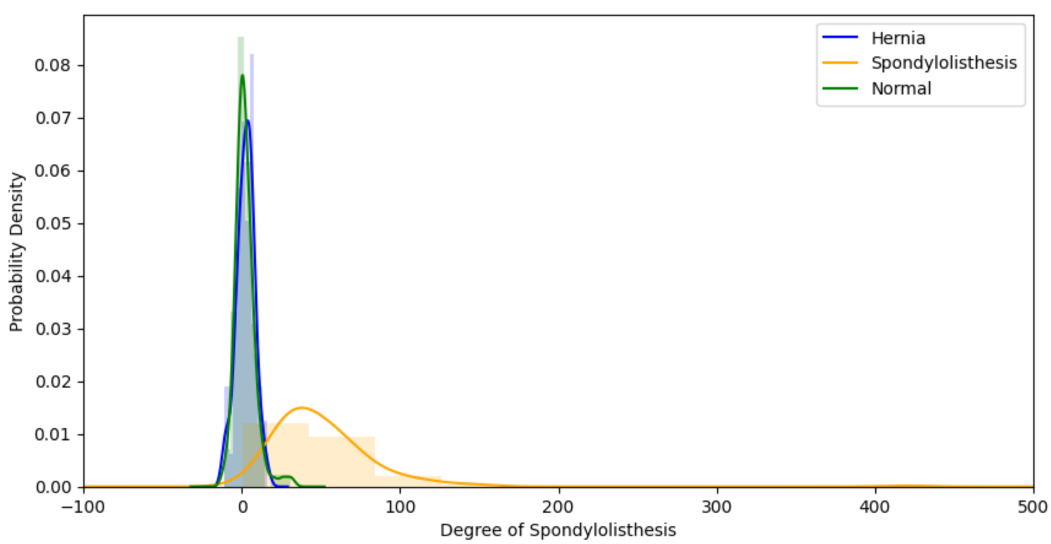


Figure 2: Variable with highest discriminative power

2. Using a stratified 70-30 training-testing split with a fixed seed (`random_state=0`), assess in a single plot both the training and testing accuracies of a decision tree with depth limits in $\{1, 2, 3, 4, 5, 6, 8, 10\}$ and the remaining parameters as default.
   [optional] Note that split thresholding of numeric variables in decision trees is non-deterministic in `sklearn`, hence you may opt to average the results using 10 runs per parameterization.

```python
import numpy as np
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

x = df.iloc[:, :-1]
y = df.iloc[:, -1]

x_train, x_test, y_train, y_test = train_test_split(x, y, stratify=y, test_size =
    0.3, random_state=0)
x_train = x_train.astype(np.float64)
y_train = y_train.astype(str)
x_test = x_test.astype(np.float64)
y_test = y_test.astype(str)

max_depth = (1, 2, 3, 4, 5, 6, 8, 10)
train_accuracy = []
test_accuracy = []

for depth in max_depth:
    train = []
    test = []
    for i in range(10):
        predictor = tree.DecisionTreeClassifier(max_depth=depth)
        predictor.fit(x_train, y_train)
        train_predictions = predictor.predict(x_train)
        test_predictions = predictor.predict(x_test)
        train.append(accuracy_score(y_train, train_predictions, normalize=True))
        test.append(accuracy_score(y_test, test_predictions, normalize=True))
    train_accuracy.append(np.mean(train))
    test_accuracy.append(np.mean(test))

plt.figure(figsize=(10, 5))
plt.plot(max_depth, train_accuracy, label='Train Accuracy', marker='x')
plt.plot(max_depth, test_accuracy, label='Test Accuracy', marker='x')
plt.xlabel('Max Depth')
plt.ylabel('Accuracy')
plt.grid()
plt.legend()
plt.show()
```

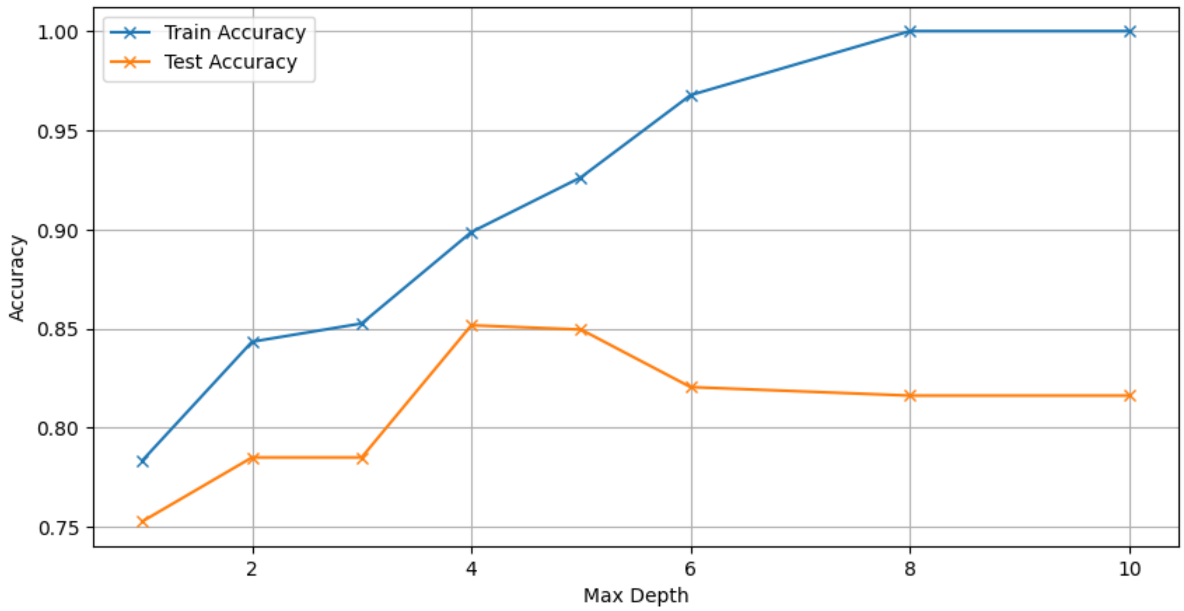


Figure 3: Training and Testing Accuracy

3. Comment on the results, including the generalization capacity across settings.

   Podemos observar que a exatidão aumenta com a profundidade máxima no grupo de treino, enquanto que no grupo de teste a exatidão é mais elevada com profundidade máxima 4 ou 5, diminuindo de seguida. Esta diminuição é um exemplo de *overfitting* do modelo, sendo que este modelo poderia beneficiar de uma profundidade máxima dentro das mencionadas acima, assim como de *pruning* (remoção dos ramos menos fiáveis da *decision tree*).

4. To deploy the predictor, a healthcare team opted to learn a single decision tree (`random_state=0`) using all available data as training data, and further ensuring that each leaf has a minimum of 20 individuals in order to avoid overfitting risks.

   i) Plot the decision tree.

```
a = df.iloc[:, :-1]
b = df.iloc[:, -1]
a = a.astype(np.float64)
b = b.astype(str)

predictor = tree.DecisionTreeClassifier(random_state=0, min_samples_leaf=20)
predictor.fit(a, b)
plt.figure(figsize=(20, 10))
tree.plot_tree(predictor, filled=True,
↪ feature_names=predictor.feature_names_in_,
↪ class_names=predictor.classes_)
plt.show()
```

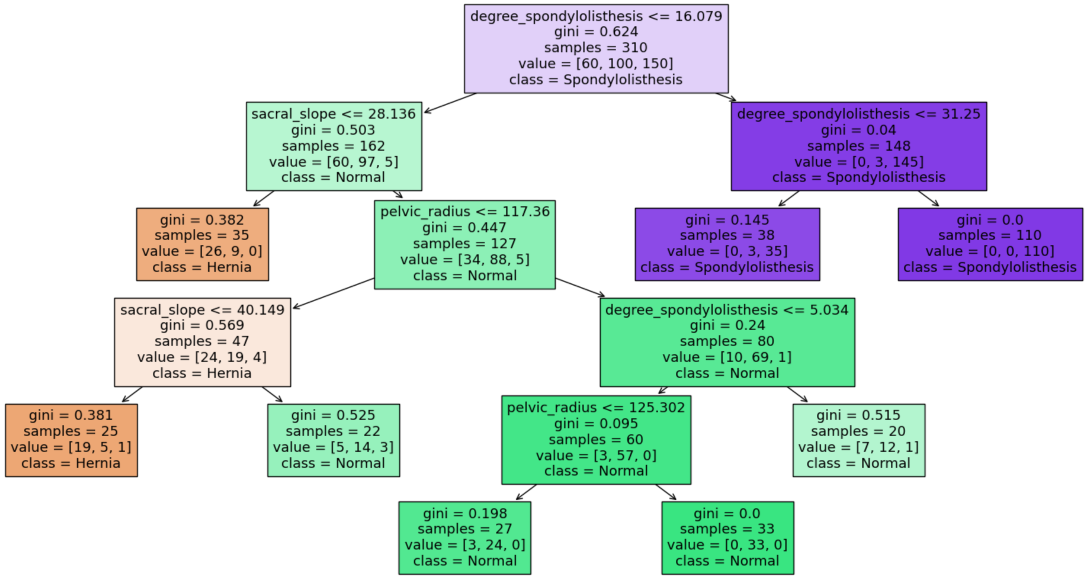


Figure 4: Decision Tree

ii) Characterize a hernia condition by identifying the hernia-conditional associations.

As condições que caraterizam uma hernia são:

$$\text{degree\_spondylolisthesis} \leq 16.079 \\ \wedge (\text{sacral\_slope} \leq 28.136 \vee (\text{pelvic\_radius} \leq 117.36 \wedge \text{sacral\_slope} \leq 40.149))$$