Aprendizagem 2023/2024

Homework I – Group 015

(ist1102763, ist1103493)

I. Pen-and-paper

1)

There are 7 observations such that $y_1 > 0.4$.

We have:

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

$$H(y_{out} \mid y_1 > 0.4) = -\sum_{v \in y_{out}} P(y_{out} = v \mid y_1 > 0.4)\ log_2 P(y_{out} = v \mid y_1 > 0.4) = -\left(\tfrac{3}{7}\ log_2\tfrac{3}{7} + \tfrac{2}{7}\ log_2\tfrac{2}{7} + \tfrac{2}{7}\ log_2\tfrac{2}{7}\right) = 1.5567$$

$$H(y_{out} \mid y_1 > 0.4,\ y_i) = \sum_{v \in y_i} P(y_i = v \mid y_1 > 0.4)\ H(y_{out} \mid y_1 > 0.4,\ y_i = v)$$

$$H(y_{out} \mid y_1 > 0.4,\ y_i = z) = -\sum_{v \in y_{out}} P(y_{out} = v \mid y_1 > 0.4,\ y_i = z)\ log_2 P(y_{out} = v \mid y_1 > 0.4,\ y_i = z)$$

Therefore, for y_2:

$$H(y_{out} \mid y_1 > 0.4,\ y_2 = 0) = -\left(\tfrac{1}{3}\ log_2\tfrac{1}{3} + \tfrac{1}{3}\ log_2\tfrac{1}{3} + \tfrac{1}{3}\ log_2\tfrac{1}{3}\right) = 1.5850$$

$$H(y_{out} \mid y_1 > 0.4,\ y_2 = 1) = -\left(0 + \tfrac{1}{2}\ log_2\tfrac{1}{2} + \tfrac{1}{2}\ log_2\tfrac{1}{2}\right) = 1$$

$$H(y_{out} \mid y_1 > 0.4,\ y_2 = 2) = -(1\ log_2 1 + 0 + 0) = 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_2) = \tfrac{3}{7} \cdot 1.5850 + \tfrac{2}{7} \cdot 1 + \tfrac{2}{7} \cdot 0 = 0.9650$$

$$IG(y_{out} \mid y_1 > 0.4,\ y_2) = 1.5567 - 0.9650 = 0.5917$$

For y_3:

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 0) = -(1\ log_2 1 + 0 + 0) = 0$$

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

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

$$H(y_{out} \mid y_1 > 0.4,\ y_3) = \tfrac{1}{7} \cdot 0 + \tfrac{2}{7} \cdot 1 + \tfrac{4}{7} \cdot 1 = 0.8571$$

$$IG(y_{out} \mid y_1 > 0.4,\ y_3) = 1.5567 - 0.8571 = 0.6996$$

And for y_4:

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

$$H(y_{out} \mid y_1 > 0.4,\ y_4 = 1) = -\left(\tfrac{1}{3}\ log_2\tfrac{1}{3} + \tfrac{2}{3}\ log_2\tfrac{2}{3} + 0\right) = 0.9183$$

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

$$H(y_{out} \mid y_1 > 0.4,\ y_4) = \tfrac{2}{7} \cdot 1 + \tfrac{3}{7} \cdot 0.9183 + \tfrac{2}{7} \cdot 1 = 0.9650$$

$$IG(y_{out} \mid y_1 > 0.4,\ y_4) = 1.5567 - 0.9650 = 0.5917$$

The variable with the highest Information Gain is y_3, so it is the one chosen to do the split.

For $y_3 = 0$, we have $y_{out} = B$.

For $y_3 = 1$, we have two observations, one with $y_{out} = A$ and another one with $y_{out} = B$, so we follow the alphabetical order and decide $y_{out} = A$.

For $y_3 = 2$, we have four observations, two with $y_{out} = A$ and two with $y_{out} = C$, so we do another split:

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2) \ = \ -(\tfrac{2}{4}\ log_2\tfrac{2}{4} + 0 + \tfrac{2}{4}\ log_2\tfrac{2}{4}) \ = \ 1$$

Therefore, for y_2:

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_2 = 0) \ = \ -(0 + 0 + 1\ log_2 1) \ = \ 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_2 = 1) \ = \ -(0 + 0 + 1\ log_2 1) \ = \ 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_2 = 2) \ = \ -(\tfrac{2}{2}\ log_2\tfrac{2}{2} + 0 + 0) \ = \ 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_2) \ = \ \tfrac{1}{4}\cdot 0 + \tfrac{1}{4}\cdot 0 + \tfrac{2}{4}\cdot 0 \ = \ 0$$

$$IG(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_2) \ = \ 1 - 0 \ = \ 1$$

And for y_4:

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_4 = 0) \ = \ -(\tfrac{1}{2}\ log_2\tfrac{1}{2} + 0 + \tfrac{1}{2}\ log_2\tfrac{1}{2}) \ = \ 1$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_4 = 1) \ = \ -(1\ log_2 1 + 0 + 0) \ = \ 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_4 = 2) \ = \ -(0 + 0 + 1\ log_2 1) \ = \ 0$$

$$H(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_4) \ = \ \tfrac{2}{4}\cdot 1 + \tfrac{1}{4}\cdot 0 + \tfrac{1}{4}\cdot 0 \ = \ 0.5$$

$$IG(y_{out} \mid y_1 > 0.4,\ y_3 = 2,\ y_4) \ = \ 1 - 0.5 \ = \ 0.5$$

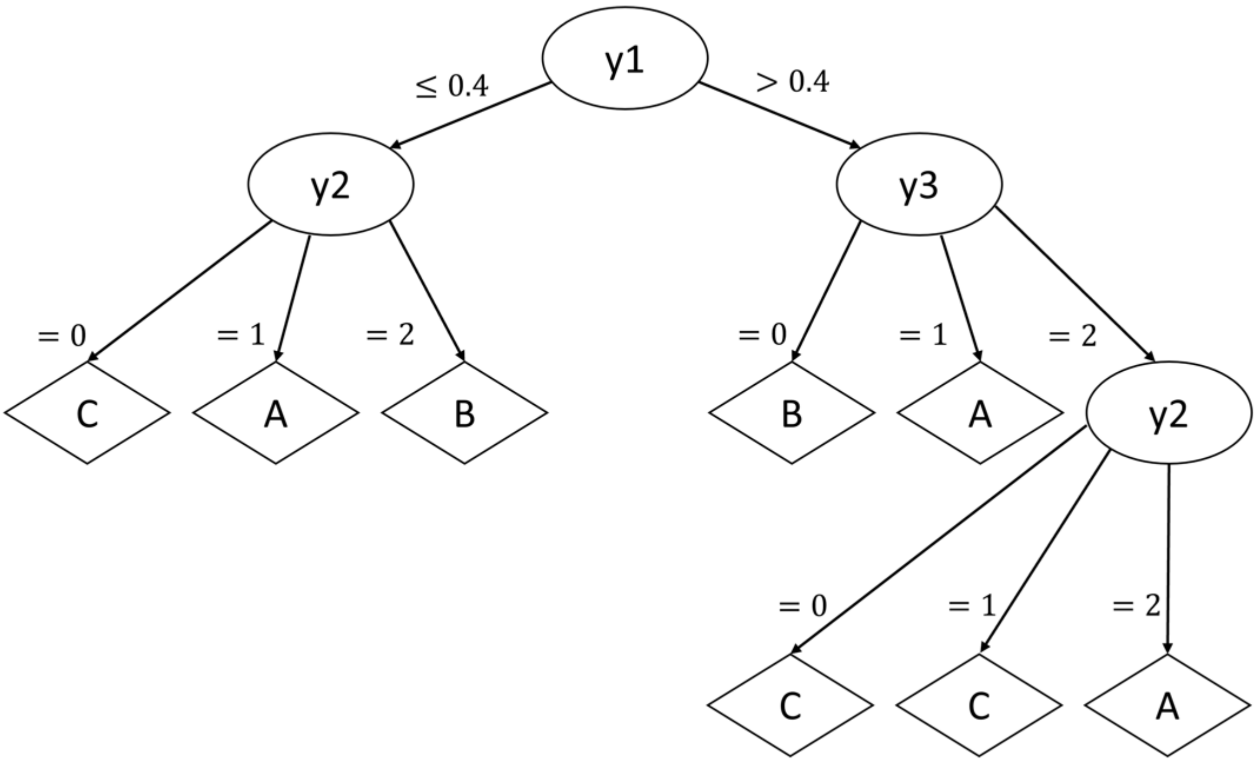
The variable with the highest Information Gain is y_2, so it is the one chosen to do the split.

For y_2 = 0, we have y_{out} = C.

For y_2 = 1, we have y_{out} = C.

For y_2 = 2, we have y_{out} = A.

The complete decision tree is the following:



Aprendizagem 2023/2024

Homework I – Group 015

(ist1102763, ist1103493)

2)

		Ground Truth			
		A	**B**	**C**	
Prediction	**A**	4	1	0	5
	B	0	2	0	2
	C	0	1	4	5
		4	4	4	12

3)

$\beta = 1 \qquad \alpha = \frac{1}{1+\beta^2} = 0.5$

$F_1 = \frac{1}{\alpha \cdot \frac{1}{p} + (1-\alpha) \cdot \frac{1}{p}} = \frac{2}{\frac{1}{p} + \frac{1}{p}}$

$precision_{class} = \frac{TP_{class}}{TP_{class} + FP_{class}}$

$recall_{class} = \frac{TP_{class}}{TP_{class} + FN_{class}}$

Class A:

$precision_A = \frac{4}{4+1} = 0.8$

$recall_A = \frac{4}{4+0} = 1$

$F_{1_A} = \frac{2}{\frac{1}{0.8} + \frac{1}{1}} = \frac{8}{9}$

Class B:

$precision_B = \frac{2}{2+0} = 1$

$recall_B = \frac{2}{2+2} = 0.5$

$F_{1_B} = \frac{2}{\frac{1}{1} + \frac{1}{0.5}} = \frac{2}{3}$

Class C:

$precision_C = \frac{4}{4+1} = 0.8$

$recall_C = \frac{4}{4+0} = 1$

$F_{1_C} = \frac{2}{\frac{1}{0.8} + \frac{1}{1}} = \frac{8}{9}$

The class with the lowest training F1 score is class B.

4)

$$Spearman(y_1, y_2) = Pearson(rank\, y_1, rank\, y_2) = \frac{Cov(rank\, y_1, rank\, y_2)}{\sqrt{Var(rank\, y_1)\, Var(rank\, y_2)}}$$

D	y_1	y_2	rank y_1	rank y_2
x_1	0.24	1	3	8
x_2	0.06	2	2	11
x_3	0.04	0	1	3.5
x_4	0.36	0	5	3.5
x_5	0.32	0	4	3.5
x_6	0.68	2	10	11
x_7	0.90	0	12	3.5
x_8	0.76	2	11	11
x_9	0.46	1	7	8
x_{10}	0.62	0	9	3.5
x_{11}	0.44	1	6	8
x_{12}	0.52	0	8	3.5

$$Cov(rank\, y_1, rank\, y_2) = \frac{1}{12-1} \sum_{i=1}^{12} (rank\, y_{1_i} - \overline{rank\, y_1}) \cdot (rank\, y_{2_i} - \overline{rank\, y_2}) = 0.9545455$$

$$Var(rank\, y_1) = \frac{1}{12-1} \sum_{i=1}^{12} (rank\, y_{1_i} - \overline{rank\, y_1})^2 = 13$$

$$Var(rank\, y_2) = \frac{1}{12-1} \sum_{i=1}^{12} (rank\, y_{2_i} - \overline{rank\, y_2})^2 = 11.0454545$$

$$Spearman(y_1, y_2) = \frac{0.9545455}{\sqrt{13 \cdot 11.0454545}} = 0.0796587$$

Therefore, y_1 and y_2 are not correlated.

Aprendizagem 2023/2024

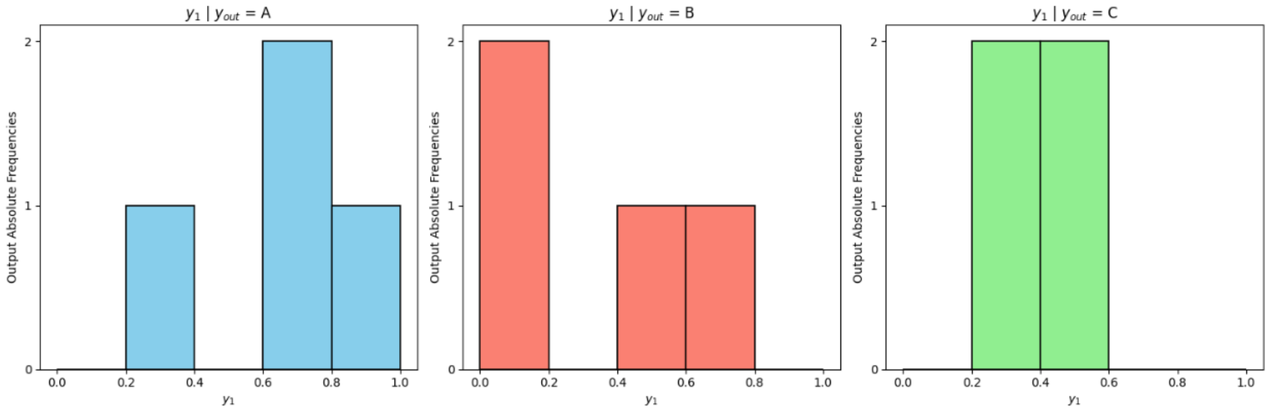
Homework I – Group 015

(ist1102763, ist1103493)

5)

```python
import numpy as np
import matplotlib.pyplot as plt

# Define the bin edges for all graphics
bin_edges = [0, 0.2, 0.4, 0.6, 0.8, 1]

# Data for graphic A
data_a_values = [0, 1, 0, 2, 1] # Corresponding values for each bin

# Data for graphic B
data_b_values = [2, 0, 1, 1, 0] # Corresponding values for each bin

# Data for graphic C
data_c_values = [0, 2, 2, 0, 0] # Corresponding values for each bin

# Create subplots
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Plot graphic A
axes[0].bar(bin_edges[:-1], data_a_values, width=0.2, color='skyblue', align='edge', edgecolor='black', linewidth=1.2)
axes[0].set_title('$y_1$ | $y_{out}$ = A')
axes[0].set_xlabel('$y_1$')
axes[0].set_ylabel('Output Absolute Frequencies')
axes[0].set_yticks(np.arange(0, max(data_a_values) + 1, 1))

# Plot graphic B
axes[1].bar(bin_edges[:-1], data_b_values, width=0.2, color='salmon', align='edge', edgecolor='black', linewidth=1.2)
axes[1].set_title('$y_1$ | $y_{out}$ = B')
axes[1].set_xlabel('$y_1$')
axes[1].set_ylabel('Output Absolute Frequencies')
axes[1].set_yticks(np.arange(0, max(data_b_values) + 1, 1))
```

```python
# Plot graphic C
axes[2].bar(bin_edges[:-1], data_c_values, width=0.2, color='lightgreen', align='edge', edgecolor='black', linewidth=1.2)
axes[2].set_title('$y_1$ | $y_{out}$ = C')
axes[2].set_xlabel('$y_1$')
axes[2].set_ylabel('Output Absolute Frequencies')
axes[2].set_yticks(np.arange(0, max(data_c_values) + 1, 1))

# Adjust layout
plt.tight_layout()

# Show the plots
plt.show()
```



According to the obtained class-conditional distribution, the root split of y_1 would result in the thresholds $[0;0.2[$, $[0.2;0.6[$ and $[0.6;1.0[$, as these are the intervals in which the highest frequencies of each output class are observed.

Aprendizagem 2023/2024

Homework I – Group 015

(ist1102763, ist1103493)

II. Programming and critical analysis

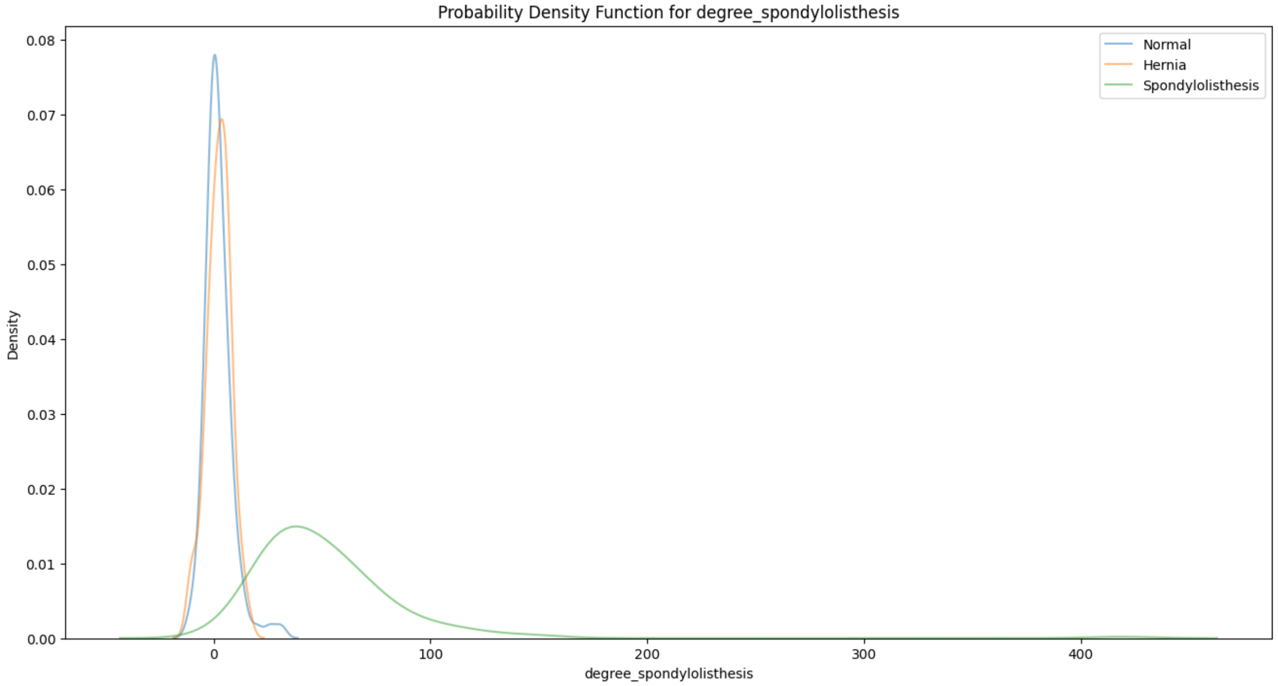
```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import seaborn as sns
from scipy.io.arff import loadarff
from sklearn import feature_selection, model_selection, tree, metrics
```

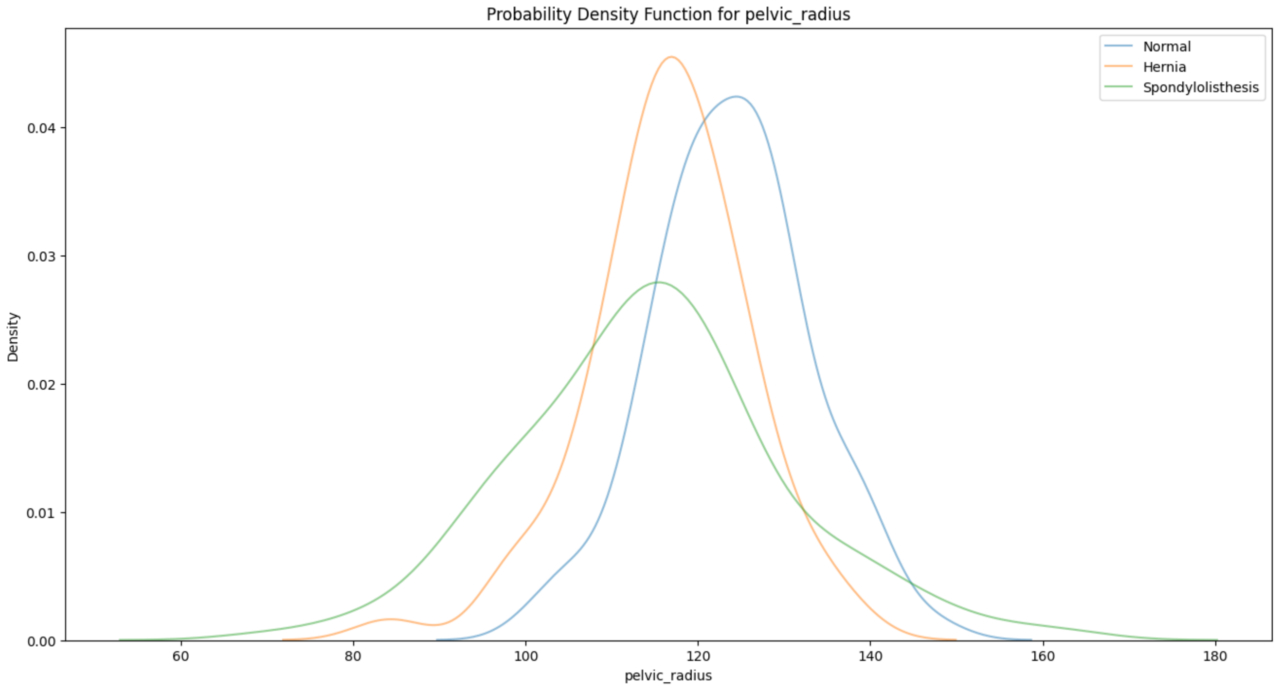
```
# Data Loading
data = loadarff("./column_diagnosis.arff")
df = pd.DataFrame(data[0])
df["class"] = df["class"].str.decode("utf-8")

# Data Preprocessing
X = df.drop("class", axis=1)
y = df["class"]
```

1)

```
# Calculate F-values using f_classif
F_values = feature_selection.f_classif(X, y)[0]

# Identify the input variable with the highest and lowest discriminative power
max_F_index = np.argmax(F_values)
print("Variable with the Highest Discriminative Power:\t", X.columns.values[max_F_index])
min_F_index = np.argmin(F_values)
print("Variable with the Lowest Discriminative Power:\t", X.columns.values[min_F_index])

# Create subplots
fig, axes = plt.subplots(2, 1, figsize=(16, 18))

# Plot class-conditional probability density functions
sns.kdeplot(X.iloc[:, max_F_index][y == "Normal"], label="Normal", ax=axes[0], alpha=0.5)
sns.kdeplot(X.iloc[:, max_F_index][y == "Hernia"], label="Hernia", ax=axes[0], alpha=0.5)
sns.kdeplot(X.iloc[:, max_F_index][y == "Spondylolisthesis"], label="Spondylolisthesis", ax=axes[0], alpha=0.5)
axes[0].set_title("Probability Density Function for degree_spondylolisthesis")

sns.kdeplot(X.iloc[:, min_F_index][y == "Normal"], label="Normal", ax=axes[1], alpha=0.5)
sns.kdeplot(X.iloc[:, min_F_index][y == "Hernia"], label="Hernia", ax=axes[1], alpha=0.5)
sns.kdeplot(X.iloc[:, min_F_index][y == "Spondylolisthesis"], label="Spondylolisthesis", ax=axes[1], alpha=0.5)
axes[1].set_title("Probability Density Function for pelvic_radius")

# Add legends
axes[0].legend()
axes[1].legend()

# Show the plots
plt.show()
```

```
Variable with the Highest Discriminative Power:   degree_spondylolisthesis
Variable with the Lowest Discriminative Power:    pelvic_radius
```


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


Probability Density Function for pelvic_radius
Normal
Hernia
Spondylolisthesis
Density
pelvic_radius
0.00
0.01
0.02
0.03
0.04
60
80
100
120
140
160
180

Aprendizagem 2023/2024

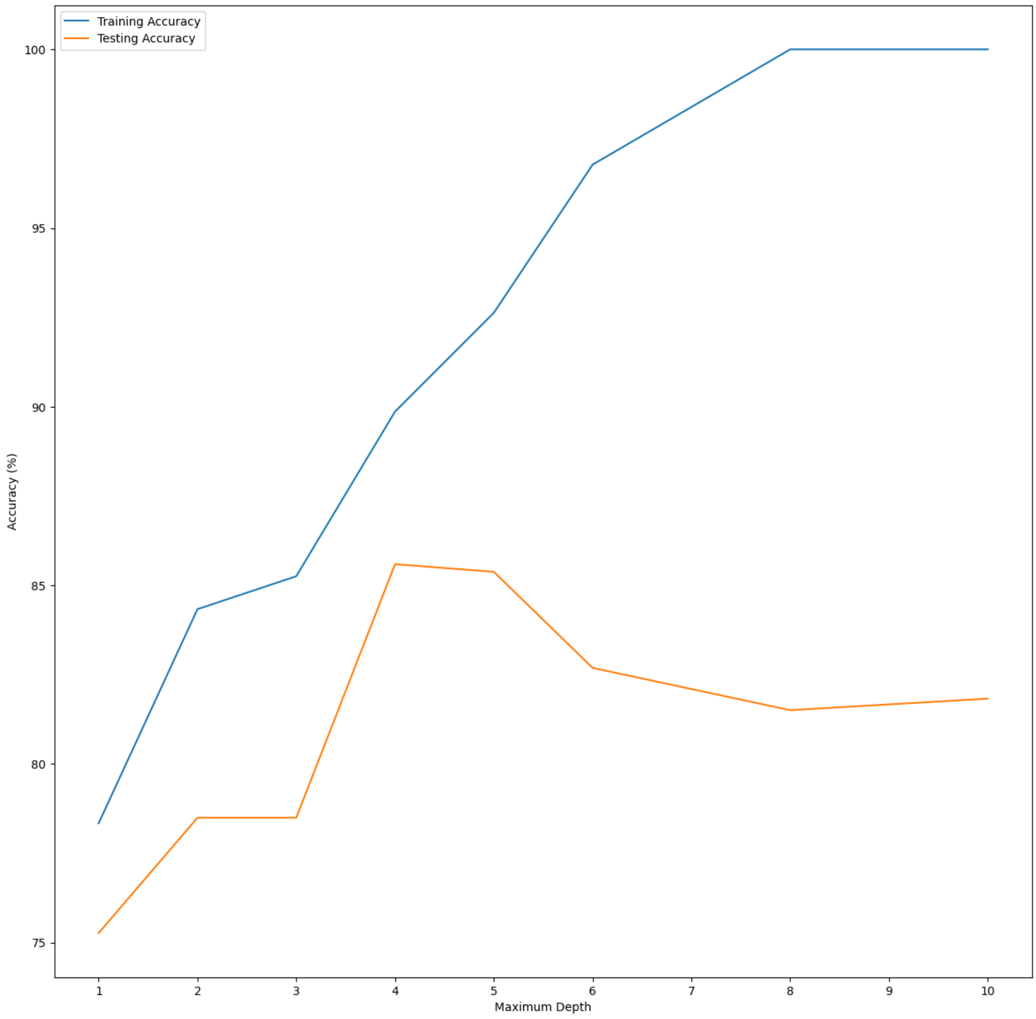
Homework I – Group 015

(ist1102763, ist1103493)

2)

```python
# Split the dataset into a training set (70%) and a testing set (30%),
# using stratified sampling
X_train, X_test, y_train, y_test = \
    model_selection.train_test_split(X, y, train_size=0.7, \
    stratify=y, random_state=0)

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

training_acc = []
testing_acc = []

# Averaging results over 10 runs per parameterization
num_runs = 10

for i in depth_limits:
    train_acc_sum = 0
    test_acc_sum = 0
    for _ in range(num_runs):
        clf = tree.DecisionTreeClassifier(max_depth=i)
        clf.fit(X_train, y_train)
        y_pred_train = clf.predict(X_train)
        y_pred_test = clf.predict(X_test)
        train_acc_sum += metrics.accuracy_score(y_train, y_pred_train)
        test_acc_sum += metrics.accuracy_score(y_test, y_pred_test)
    training_acc.append(train_acc_sum / num_runs)
    testing_acc.append(test_acc_sum / num_runs)

# Multiply the accuracy scores by 100 to convert them to percentages
training_acc = [acc * 100 for acc in training_acc]
testing_acc = [acc * 100 for acc in testing_acc]
```

```python
plt.figure(figsize = (15, 15))
plt.plot(depth_limits, training_acc, label="Training Accuracy")
plt.plot(depth_limits, testing_acc, label="Testing Accuracy")
plt.xticks(range(1, 11))
plt.xlabel("Maximum Depth")
plt.ylabel("Accuracy (%)")
plt.legend()
plt.show()
```



3)

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

The results show that as the maximum depth of the decision tree increases, training accuracy improves significantly.

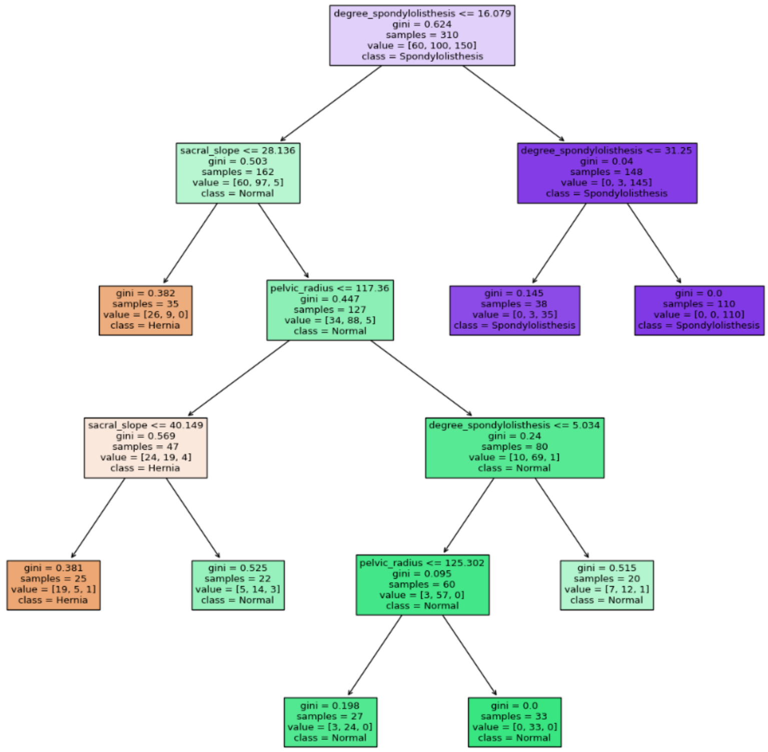
However, testing accuracy plateaus and then decreases beyond a certain depth. This suggests that the model has high capacity to fit the training data but lacks generalization capacity, performing worse on unseen data with overly deep trees. The optimal maximum depth balances model complexity and generalization.

The training accuracy consistently increases as the maximum depth of the decision tree grows. This occurs because deeper trees can capture more details and noise within the training data, resulting in a better fit. Training accuracy reaches 100% with high maximum depths, indicating that the model effectively memorizes the training data. However, extremely high training accuracy doesn't necessarily translate to good generalization, as the model may struggle to perform well on new, unseen data due to overfitting.

4)

i.

```
clf = tree.DecisionTreeClassifier(random_state = 0, min_samples_leaf = 20)
clf.fit(X, y)

plt.figure(figsize = (15, 15))
tree.plot_tree(clf, feature_names = X.columns.values, class_names = clf.classes_, filled = True)
plt.show()

# gini: measure of impurity or disorder in a node.
# It quantifies the likelihood of misclassifying a randomly chosen element if it
# was randomly classified according to the distribution of samples in the node.
```



ii.

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

Based on the plotted decision tree, a hernia condition is identified with the highest probability when one of the following sets of predicates is true:

- degree_spondylolisthesis $\leq$ 16.079 $\wedge$ sacral_slope $\leq$ 28.136
- degree_spondylolisthesis $\leq$ 16.079 $\wedge$ 28.136 $<$ sacral_slope $\leq$ 40.149 $\wedge$ pelvic_radius $\leq$ 117.36

END