Aprendizagem 2022/23

Homework I – Group 35

(ist1100032, ist1100070)

I. Pen-and-paper

1)

1) $E(\text{yout} \mid y_1 > 0.4) = -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{2}{7}\log_2\frac{2}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 1.56$

b) ??

- $IG(\text{yout} \mid y_1 > 0.4 \wedge y_2) = E(\text{yout} \mid y_1 > 0.4) - E(\text{yout} \mid y_1 > 0.4 \wedge y_2) = 1.56 - 0.965 = \boxed{0.595}$

  $\rightarrow E(\text{yout} \mid y_1 > 0.4 \wedge y_2) = E(\text{yout} \mid y_1 > 0.4 \wedge y_2 = 0) + E(\text{yout} \mid y_1 > 0.4 \wedge y_2 = 1) + E(\text{yout} \mid y_1 > 0.4 \wedge y_2 = 2)$

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

  $= \frac{3}{7}\cdot\log_2(3) + \frac{2}{7} + 0 = 0.965$

- $IG(\text{yout} \mid y_1 > 0.4 \wedge y_3) = E(\text{yout} \mid y_1 > 0.4) - E(\text{yout} \mid y_1 > 0.4 \wedge y_2) = 1.56 - \frac{6}{7} = \boxed{0.703}$ ← y_3

  $\rightarrow E(\text{yout} \mid y_1 > 0.4 \wedge y_3) = E(\text{yout} \mid y_1 > 0.4 \wedge y_3 = 0) + E(\text{yout} \mid y_1 > 0.4 \wedge y_3 = 1) + E(\text{yout} \mid y_1 > 0.4 \wedge y_3 = 2)$

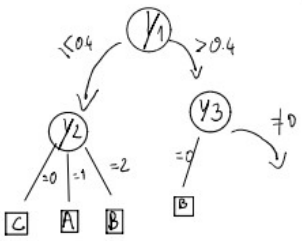
  $= \frac{1}{7}\times(0) + \frac{2}{7}\times(1) + \frac{4}{7}(1) = \frac{6}{7}$

- $IG(\text{yout} \mid y_1 > 0.4 \wedge y_4) = E(\text{yout} \mid y_1 > 0.4) - E(\text{yout} \mid y_1 > 0.4 \wedge y_2) = 1.56 - 0.965 = \boxed{0.595}$

  $\rightarrow E(\text{yout} \mid y_1 > 0.4 \wedge y_4) = E(\text{yout} \mid y_1 > 0.4 \wedge y_4 = 0) + E(\text{yout} \mid y_1 > 0.4 \wedge y_4 = 1) + E(\text{yout} \mid y_1 > 0.4 \wedge y_4 = 2)$

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

  $= \frac{2}{7} + 0.394 + \frac{2}{7} = 0.965$



Homework I – Group 35

(ist1100032, ist1100070)

1) Continuação

$E(y_{out} \mid y_1 > 0,4 \wedge y_3 \neq 0 \wedge y_2) =$

$= \underbrace{\frac{1}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}_{y_2 = 0} + \underbrace{\frac{1}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}_{y_2 = 1} +$

$+ \underbrace{\frac{1}{3}\left(-1\log_2 1\right)}_{y_2 = 2} =$

$= \frac{2}{3}$

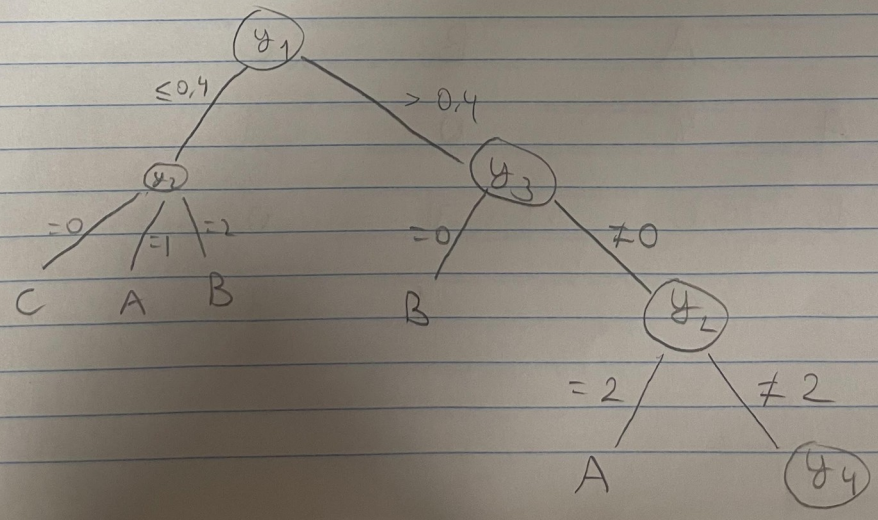
$IG(y_{out} \mid y_1 > 0,4 \wedge y_3 \neq 0 \wedge y_2) = 1,459 - \frac{2}{3} = 0,7923(3)$ ← y_2

$E(y_{out} \mid y_1 > 0,4 \wedge y_3 \neq 0 \wedge y_4) =$

$= \underbrace{\frac{1}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}_{y_4 = 0} + \underbrace{\frac{1}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}_{y_4 = 1} +$

$= \underbrace{\frac{1}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}_{y_4 = 2} = 1$

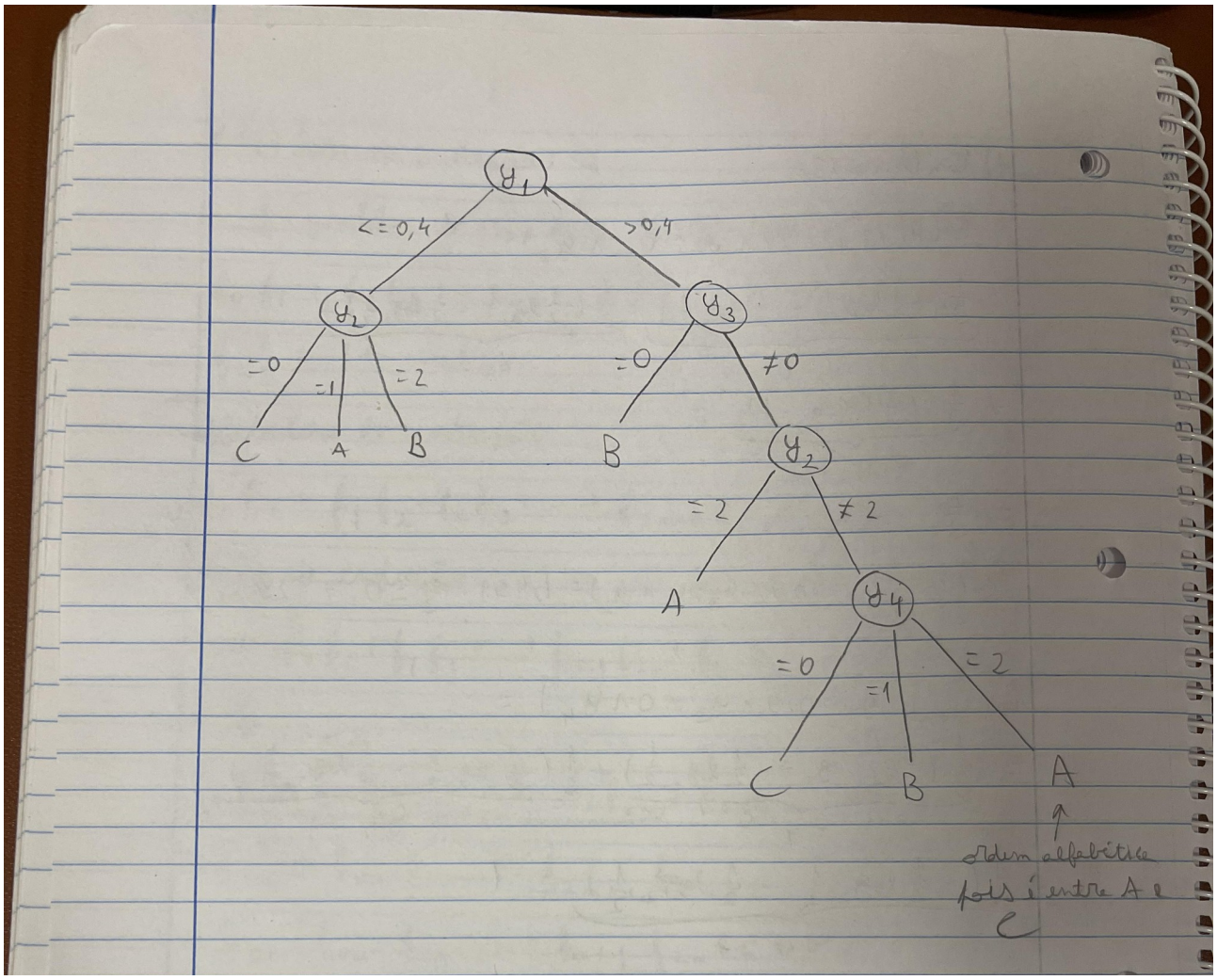
$IG(y_{out} \mid y_1 > 0,4 \wedge y_3 \neq 0 \wedge y_4) = 1,459 - 1 = 0,459$



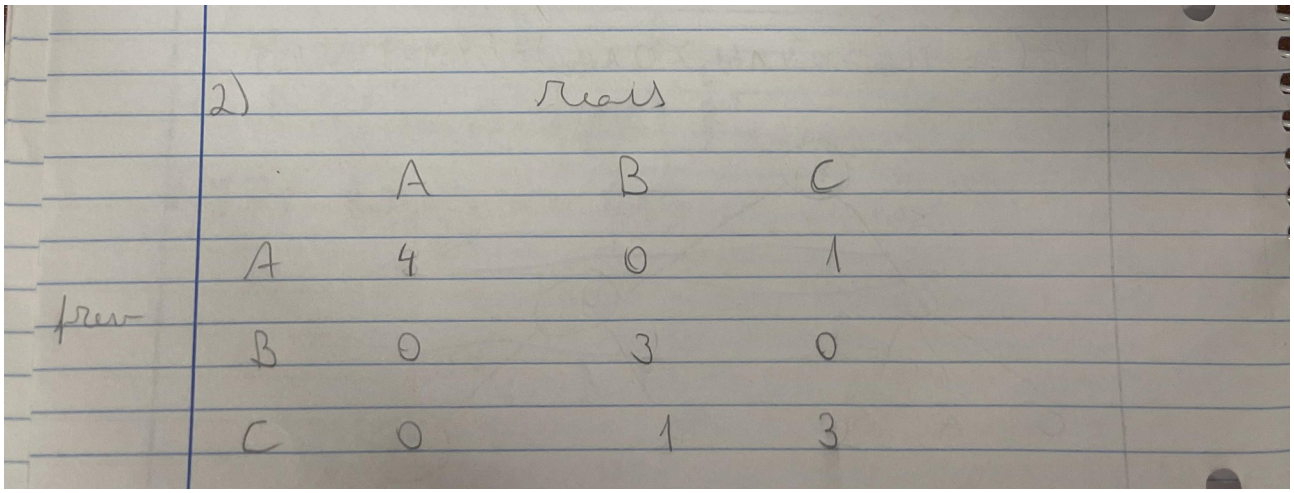
Aprendizagem 2022/23

Homework I – Group 35

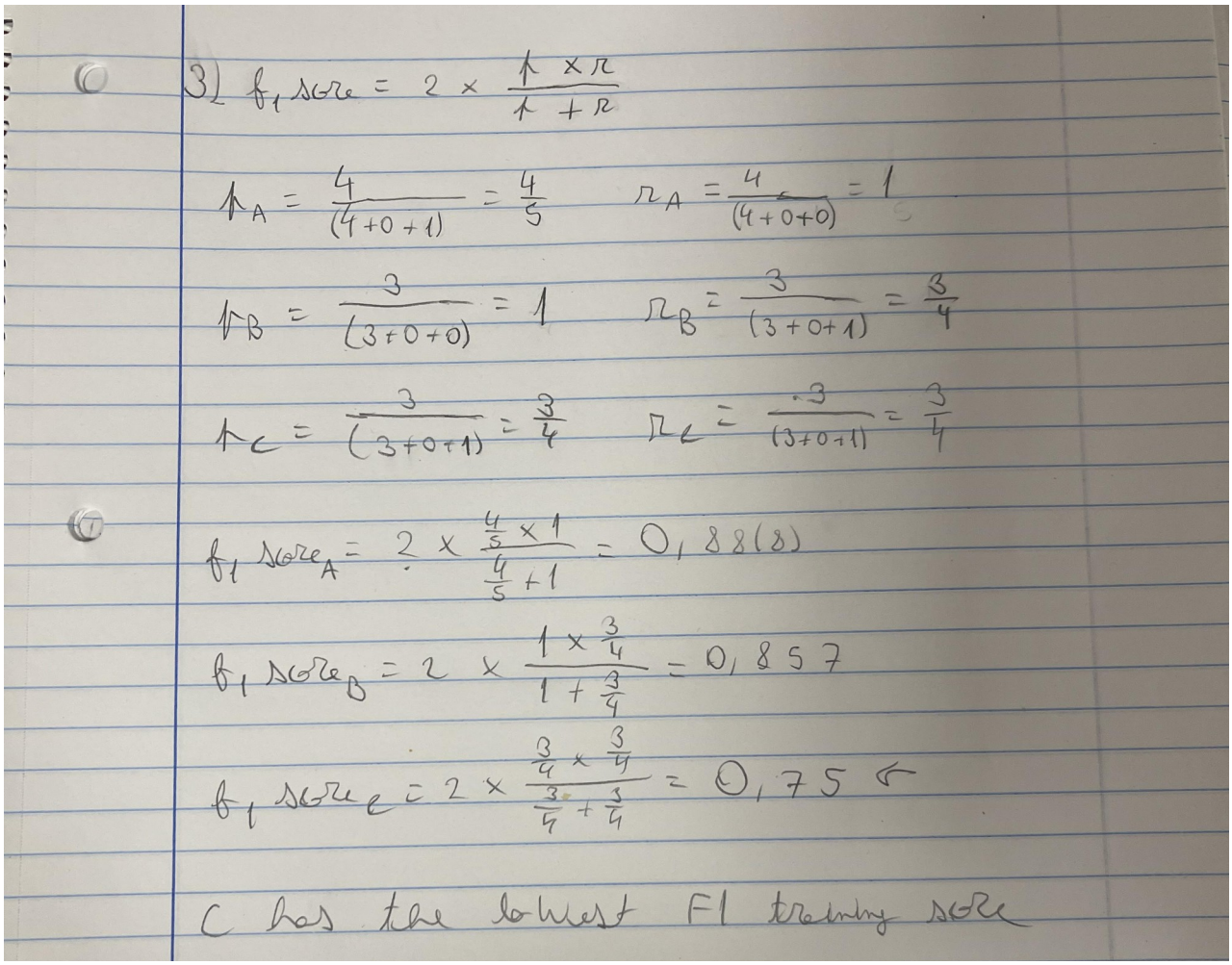
(ist1100032, ist1100070)



2)



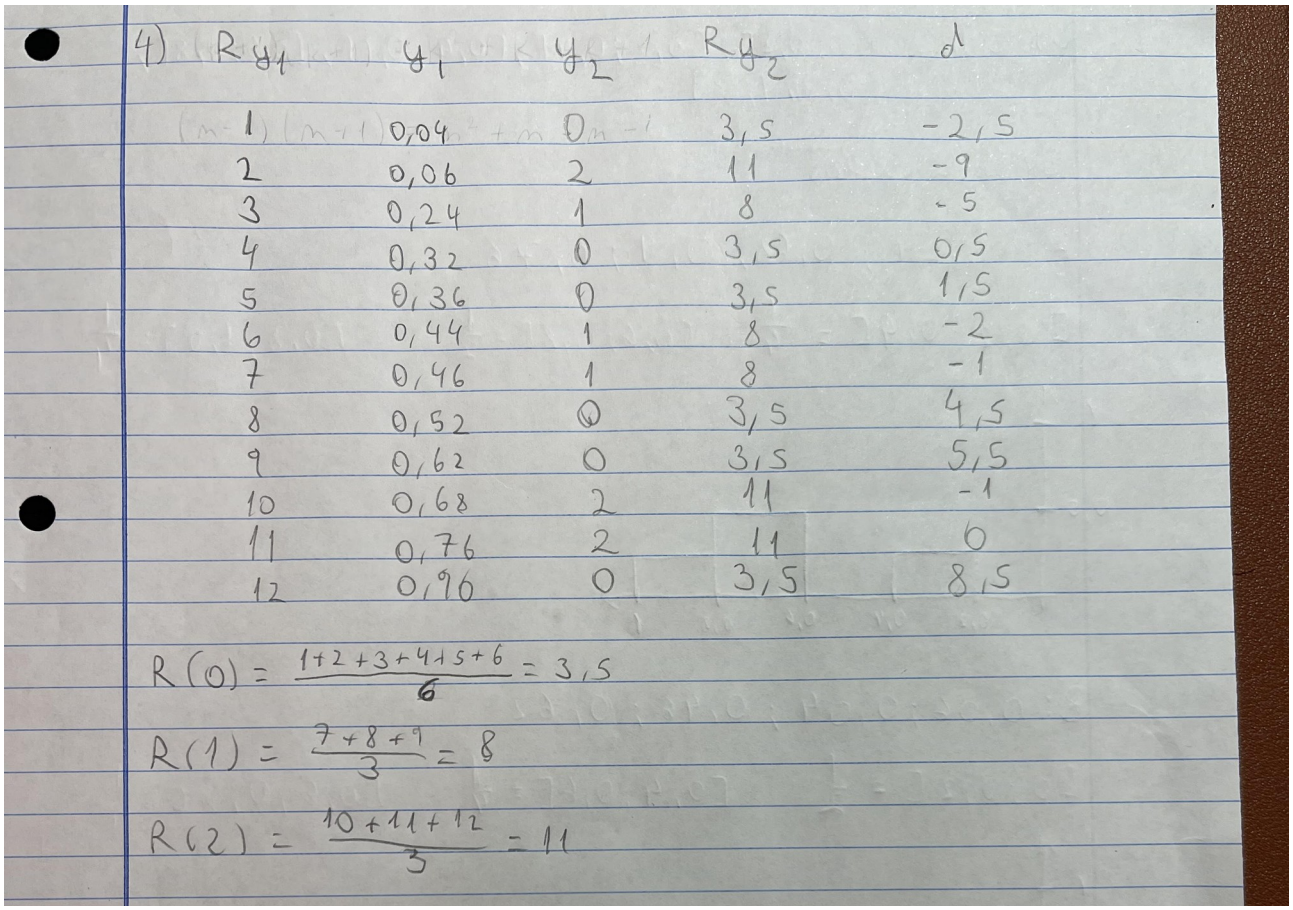
3)


3) $f_1\,score = 2 \times \frac{\uparrow \times \pi}{\uparrow + \pi}$
$\uparrow_A = \frac{4}{(4+0+1)} = \frac{4}{5}$ $\pi_A = \frac{4}{(4+0+0)} = 1$
$\uparrow_B = \frac{3}{(3+0+0)} = 1$ $\pi_B = \frac{3}{(3+0+1)} = \frac{3}{4}$
$\uparrow_C = \frac{3}{(3+0+1)} = \frac{3}{4}$ $\pi_C = \frac{3}{(3+0+1)} = \frac{3}{4}$
$f_1\,score_A = 2 \times \frac{\frac{4}{5} \times 1}{\frac{4}{5}+1} = 0,888$
$f_1\,score_B = 2 \times \frac{1 \times \frac{3}{4}}{1+\frac{3}{4}} = 0,857$
$f_1\,score_C = 2 \times \frac{\frac{3}{4} \times \frac{3}{4}}{\frac{3}{4}+\frac{3}{4}} = 0,75$
C has the lowest F1 training score

Homework I – Group 35

(ist1100032, ist1100070)

4)



4)

	R_{y_1}	y_1	y_2	R_{y_2}	d
1		0,04		3,5	-2,5
2		0,06	2	11	-9
3		0,24	1	8	-5
4		0,32	0	3,5	0,5
5		0,36	0	3,5	1,5
6		0,44	1	8	-2
7		0,46	1	8	-1
8		0,52	0	3,5	4,5
9		0,62	0	3,5	5,5
10		0,68	2	11	-1
11		0,76	2	11	0
12		0,96	0	3,5	8,5

$$R(0) = \frac{1+2+3+4+5+6}{6} = 3,5$$

$$R(1) = \frac{7+8+9}{3} = 8$$

$$R(2) = \frac{10+11+12}{3} = 11$$

Homework I – Group 35

(ist1100032, ist1100070)

$$\sigma = \sqrt{\frac{\Sigma (x_i)^2 - n\bar{x}^2}{n-1}} =$$

$$\sigma(y_1) = \sqrt{\frac{\Sigma (y_{1i})^2 - 12\bar{y}_1^2}{11}} = \sqrt{\frac{650 - 12 \times 6{,}5^2}{11}} = 3{,}61$$

$$\sigma(y_2) = \sqrt{\frac{\Sigma (y_{2i})^2 - 12\bar{y}_2}{11}} = \sqrt{\frac{628{,}5 - 12 \times 6{,}5^2}{11}} = 3{,}32$$

$$Cov(y_1, y_2) = \frac{\sum_{i=1}^{n} y_{1i} y_{2i} - n\bar{y}_1\bar{y}_2}{n-1} =$$

$$= \frac{517{,}5 - 12 \times 6{,}5^2}{11} =$$

$$= \frac{10{,}5}{11} = 0{,}955$$

$$r = \frac{Cov(y_1, y_2)}{\sigma(y_1) \times \sigma(y_2)} = \frac{0{,}955}{3{,}61 \times 3{,}32} = 0{,}0797$$

The Spearman coefficient is 0,0797 which indicates that y_1 and y_2 aren't correlated

Homework I – Group 35

(ist1100032, ist1100070)

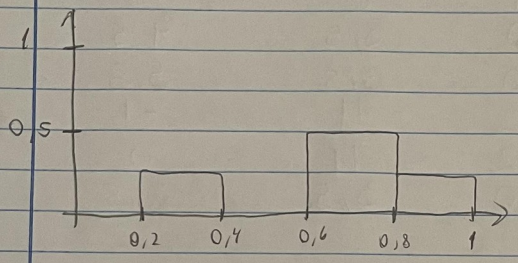
5)

5) 5 bins: $[0; 0,2[$, $[0,2; 0,4[$, $[0,4; 0,6[$, $[0,6; 0,8[$, $[0,8; 1,0]$

For each class

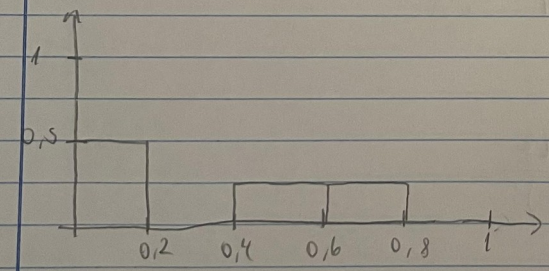
A: 0,24 ; 0,68 ; 0,9 ; 0,76

$[0,2; 0,4[= \frac{1}{4}$ $[0,6; 0,8[= \frac{1}{2}$ $[0,8; 1,0] = \frac{1}{4}$



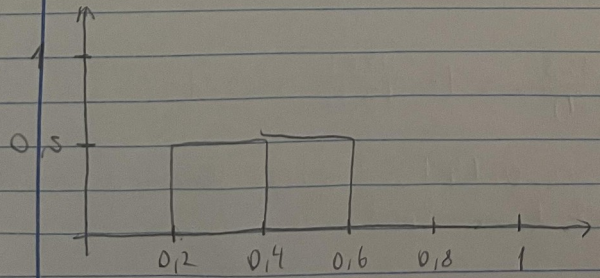
B: 0,06 ; 0,04 ; 0,46 ; 0,62

$[0; 0,2[= \frac{1}{2}$ $[0,4; 0,6[= \frac{1}{4}$ $[0,6; 0,8[= \frac{1}{4}$



C: 0,36 ; 0,32 ; 0,44 ; 0,52

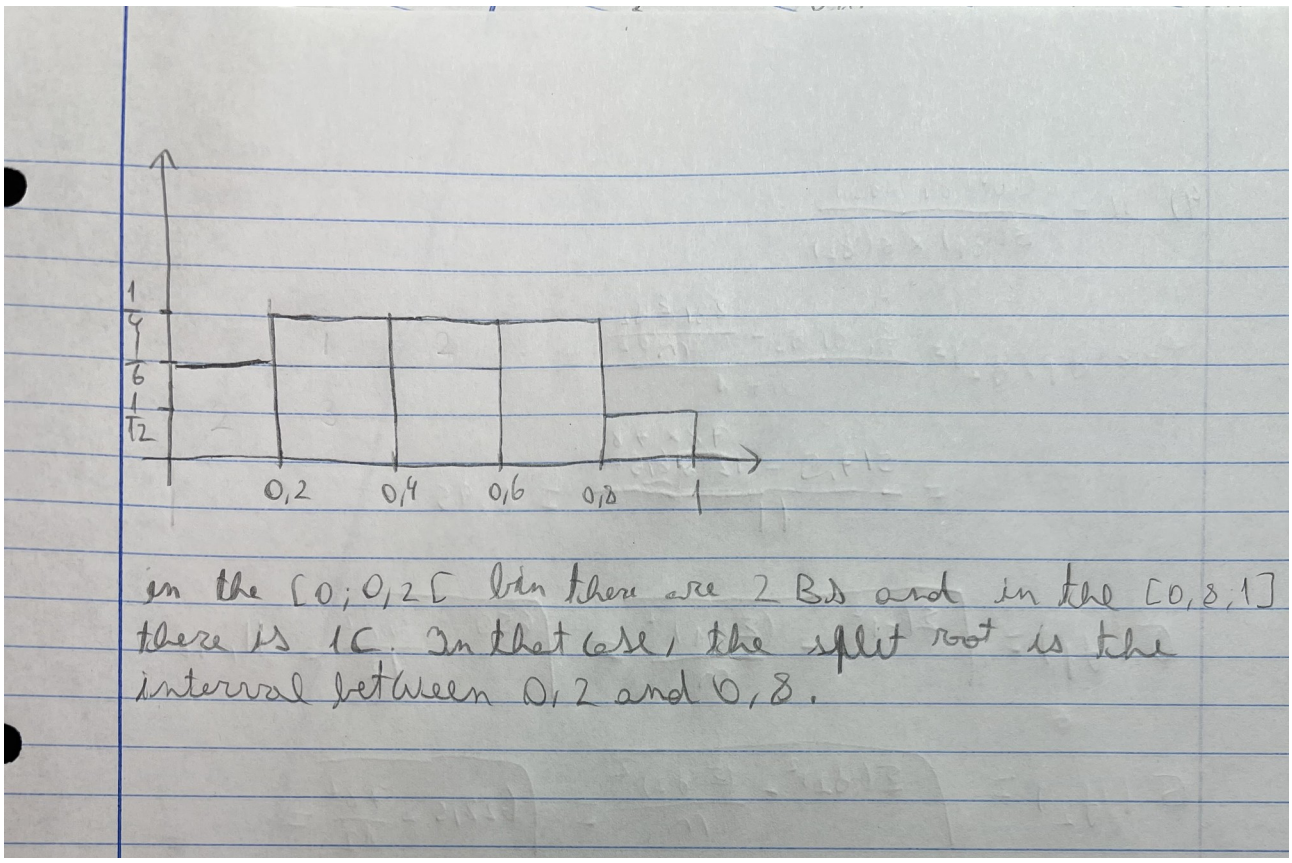
$[0,2; 0,4[= \frac{1}{2}$ $[0,4; 0,6[= \frac{1}{2}$



Aprendizagem 2022/23

Homework I – Group 35

(ist1100032, ist1100070)



in the [0; 0,2[bin there are 2 Bs and in the [0,8,1] there is 1C. In that case, the split root is the interval between 0,2 and 0,8.

Homework I – Group 35

(ist1100032, ist1100070)

II. Programming and critical analysis

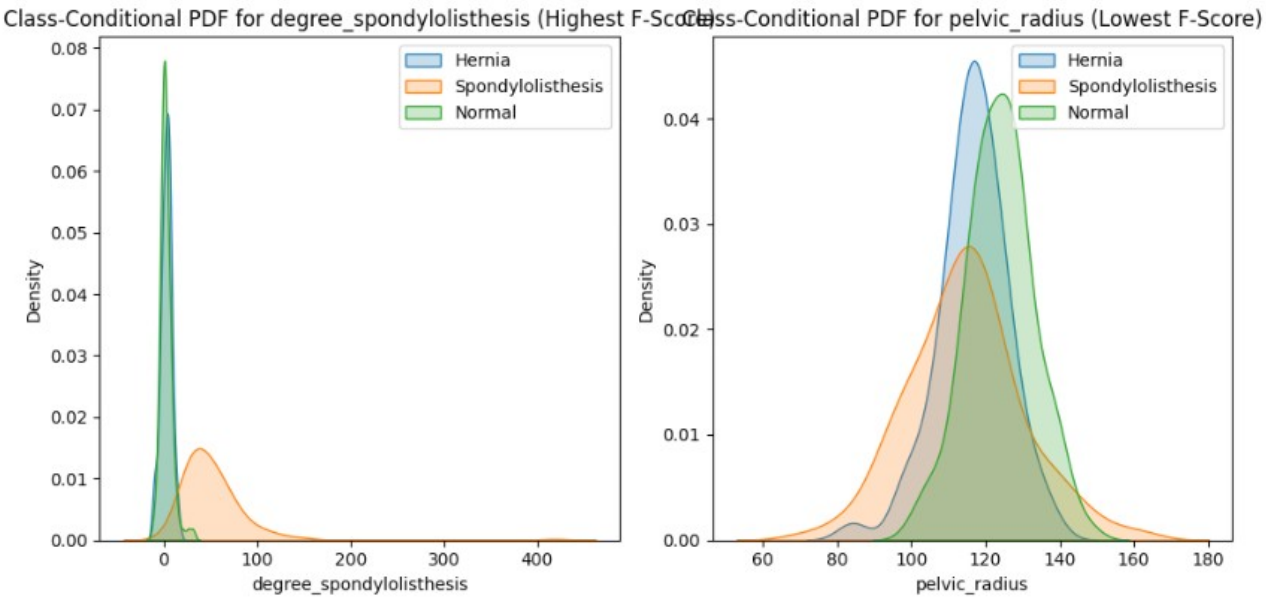
1)

```
import warnings
import pandas as pd
from scipy.io.arff import loadarff
from sklearn.feature_selection import f_classif

warnings.filterwarnings("ignore", category=FutureWarning) # Ignore FutureWarnings

# Load data from 'column_diagnosis.arff' into a DataFrame

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

# Separate features (X) and target (y)
X = df.drop('class', axis=1)
y = df['class']
y = y.astype(str)

# Compute feature importance using F-Score
fimportance = f_classif(X, y)

feature_importance_df = pd.DataFrame({'Feature': X.columns, 'F-Score': fimportance[0]})

index_of_highest_fscore = feature_importance_df['F-Score'].idxmax()
index_of_lowest_fscore = feature_importance_df['F-Score'].idxmin()

highest_feature = feature_importance_df.iloc[index_of_highest_fscore]
lowest_feature = feature_importance_df.iloc[index_of_lowest_fscore]

# Print the feature with the highest and lowest F-Score
print("Feature with Highest F-Score:")
print(highest_feature)

print("\nFeature with Lowest F-Score:")
print(lowest_feature)
```

```
Feature with Highest F-Score:
Feature    degree_spondylolisthesis
F-Score                  119.122881
Name: 5, dtype: object
```

```
Feature with Lowest F-Score:
Feature    pelvic_radius
F-Score        16.866935
Name: 4, dtype: object
```

```
import matplotlib.pyplot as plt
import seaborn as sns

highest_feature_name = highest_feature['Feature']
lowest_feature_name = lowest_feature['Feature']

plt.figure(figsize=(10, 5))

# Plot for the feature with the highest F-Score
plt.subplot(1, 2, 1)
for class_label in df['class'].unique():
    sns.kdeplot(data=df[df['class'] == class_label], x=highest_feature_name, label=class_label.decode('utf-8'),  fill = True)
plt.title(f'Class-Conditional PDF for {highest_feature_name} (Highest F-Score)')
plt.legend()

# Plot for the feature with the lowest F-Score
plt.subplot(1, 2, 2)
for class_label in df['class'].unique():
    sns.kdeplot(data=df[df['class'] == class_label], x=lowest_feature_name, label=class_label.decode('utf-8'), fill = True)
plt.title(f'Class-Conditional PDF for {lowest_feature_name} (Lowest F-Score)')
plt.legend()

plt.tight_layout()
plt.show()
```



2)

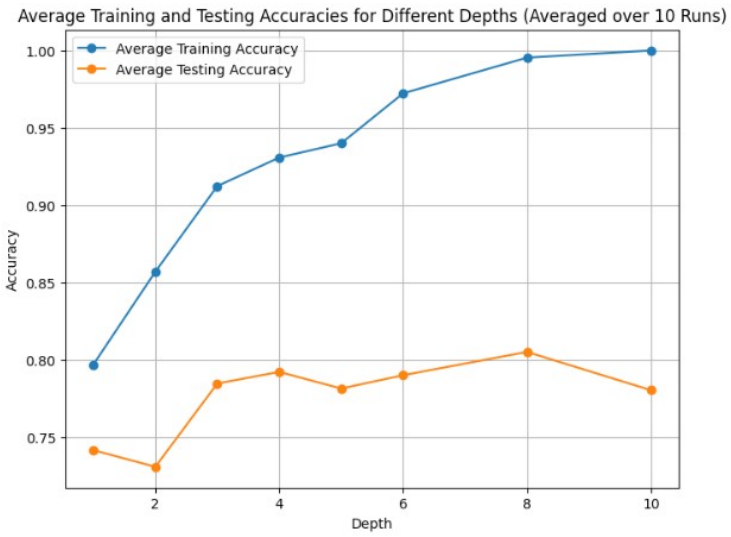
TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I – Group 35

(ist1100032, ist1100070)

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

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

average_train_accuracies = []
average_test_accuracies = []

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=0)

for depth in depths:
    # List to store  accuracies for each run
    train_accuracies = []
    test_accuracies = []
    for _ in range(num_runs):
        # Create a decision tree classifier with the specified depth
        predictor = tree.DecisionTreeClassifier(max_depth=depth, random_state=_)
        predictor.fit(X_train, y_train)

        # Training Accuracy
        y_train_pred = predictor.predict(X_train)
        train_accuracy = metrics.accuracy_score(y_train, y_train_pred)
        train_accuracies.append(train_accuracy)

        # Testing Accuracy
        y_test_pred = predictor.predict(X_test)
        test_accuracy = metrics.accuracy_score(y_test, y_test_pred)
        test_accuracies.append(test_accuracy)

    # Calculate and store the average accuracies for the current depth
    average_train_accuracy = sum(train_accuracies) / num_runs
    average_train_accuracies.append(average_train_accuracy)

    average_test_accuracy = sum(test_accuracies) / num_runs
    average_test_accuracies.append(average_test_accuracy)

# Create the line plot
plt.figure(figsize=(8, 6))
plt.plot(depths, average_train_accuracies, label='Average Training Accuracy', marker='o')
plt.plot(depths, average_test_accuracies, label='Average Testing Accuracy', marker='o')

plt.title('Average Training and Testing Accuracies for Different Depths (Averaged over 10 Runs)')
plt.xlabel('Depth')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```



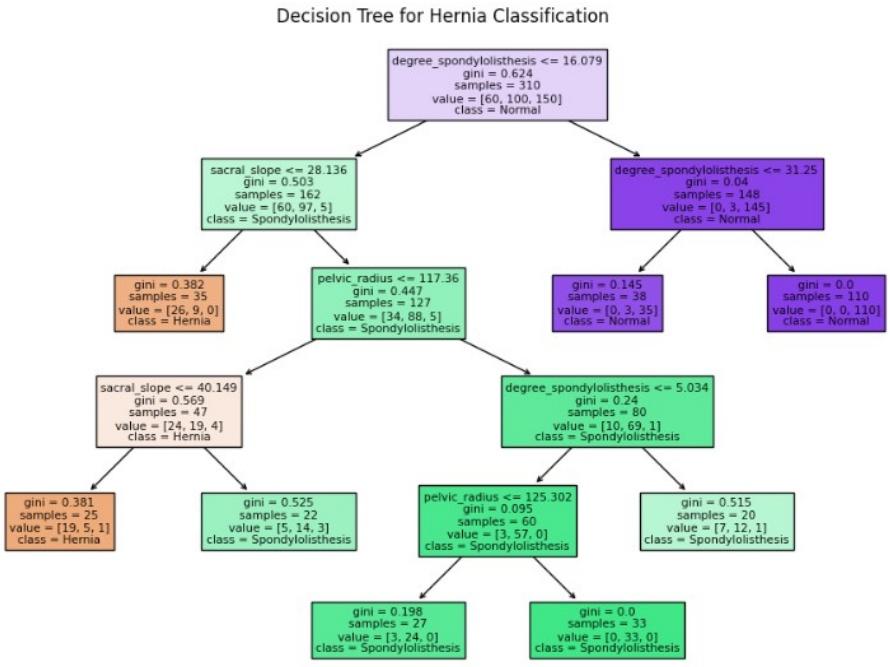
3)

Generalization capacity across different settings is poor, as the average testing accuracy is lower than the training accuracy. Overfitting becomes more pronounced with increasing maximum depth, as evidenced by the widening gap between the training and testing accuracy lines. This suggests that the model is learning complex patterns in the training data that do not generalize well to new data. Despite having lower training accuracy, shallower models produce the best testing results.

4)

i)

```
from sklearn.tree import plot_tree

decision_tree = tree.DecisionTreeClassifier(random_state=0, min_samples_leaf=20)

decision_tree.fit(X, y)

# Plot the decision tree
plt.figure(figsize=(12, 8))
plot_tree(decision_tree, filled=True, feature_names=X.columns, class_names=y.unique())
plt.title("Decision Tree for Hernia Classification")
plt.show()
```



ii)

By analysing the Decision Tree, we can see that the biomechanical features that best predict the condition of hernia are:

1. Degree of spondylolisthesis
2. Sacral slope
3. Pelvic radius

Homework I – Group 35

(ist1100032, ist1100070)

Starting from the root node we see that a degree of spondylolisthesis greater than 16 will classify the patient as normal.
If lower, we look at the Sacral slope that if lesser than 28 also classifies the patient as having the hernia condition.
Only then will we look at the Pelvic radius, that if lesser than 117 and in combination with a sacral slope lesser than 440 leads to a hernia condition.

Therefore, the conditional associations are the following:
- Degree of spondylolisthesis <= 16.079 -> sacral slope <= 28.136
- Degree of spondylolisthesis <= 16.079 -> sacral slope > 28.136 -> pelvic radius <= 117.36 -> sacral slope <= 20.149

We can conclude that the empirical probability of being diagnosed with a hernia condition is [(25+35)/310]x100 = 19%.

END