Aprendizagem 2022/23

**Homework I – Group 71**

(ist1103471, ist1104139)

# I. Pen-and-paper

1) z = y out

$H(z|y_1>0,4) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{2}{7}\log_2\frac{2}{7} - \frac{2}{7}\log_2\frac{2}{7} \approx 1,5567$

$y_2$

$H(z|y_1>0,4,\ y_2) = -\frac{3}{7}(3\times\frac{1}{3}\log_2\frac{1}{3}) - \frac{2}{7}(2\times\frac{1}{2}\log_2\frac{1}{2}) - \frac{2}{7}(1\log_2 1) \approx 0,965$

$IG(z|y_1>0,4,\ y_2) = 1,5567 - 0,9650 = 0,592$ //

$y_3$

$H(z|y_1>0,4,\ y_3) = -\frac{1}{7}(1\log_2 1) - \frac{2}{7}(2\times\frac{1}{2}\log_2\frac{1}{2}) - \frac{4}{7}(2\times\frac{1}{2}\log_2\frac{1}{2}) \approx \frac{6}{7}$

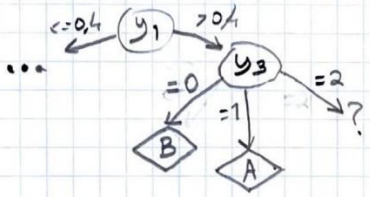
$IG(z|y_1>0,4,\ y_3) = 1,5567 - \frac{6}{7} = 0,700$ //

$y_4$

$H(z|y_1>0,4,\ y_4) = -\frac{2}{7}(2\times\frac{1}{2}\log_2\frac{1}{2}) - \frac{3}{7}(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}) - \frac{2}{7}(2\times\frac{1}{2}\log_2\frac{1}{2}) =$

$= \frac{2}{7} + \frac{3}{7}\log_2(3) \approx 0,965$

$IG(z|y_1>0,4, y_4) = 0,592$ //




$H(z|y_1>0,4,\ y_3=2) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$

$y_2$

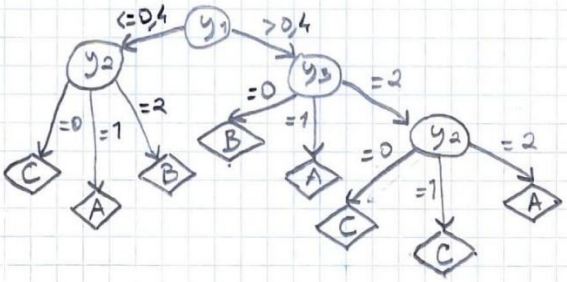
$H(z|y>0,4,\ y_3=2,\ y_2) = -\frac{1}{4}(1\log_2 1) - \frac{1}{4}(1\log_2 1) - \frac{2}{4}(1\log_2 1) = 0$ //

$IG(z|y>0,4,\ y_3=2,\ y_2) = 1 - 0 = 1$

$y_4$

$H(z|y>0,4,\ y_3=2,\ y_4) = -\frac{2}{4}(2\times\frac{1}{2}\log_2\frac{1}{2}) - \frac{1}{4}(1\log_2 1) - \frac{1}{4}(1\log_2 1) = \frac{1}{2}$ //

$IG(z|y>0,4, y_3=2,\ y_4) = 1 - \frac{1}{2} = \frac{1}{2}$



**Homework I – Group 71**

(ist1103471, ist1104139)

2)

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

$z = [A\ B\ B\ C\ C\ A\ A\ A\ B\ B\ C\ C]$

$\hat{z} = [A\ B\ C\ C\ C\ A\ A\ A\ A\ B\ C\ C]$

3)

$$F_1 = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)}$$

P: precision

R: recall

$P_A = \frac{4}{4+2} = \frac{2}{3}$ $\quad R_A = \frac{4}{4} = 1$

$P_B = \frac{1}{1} = 1$ $\quad R_B = \frac{2}{2+1+1} = \frac{1}{2}$

$P_C = \frac{4}{1+4} = \frac{4}{5}$ $\quad R_C = \frac{4}{4} = 1$

$$F_A = \frac{1}{\frac{1}{2}\left(\frac{1}{2/3} + \frac{1}{1}\right)} = \frac{4}{5} = 0,8$$

$$F_B = \frac{1}{\frac{1}{2}\left(\frac{1}{1} + \frac{1}{1/2}\right)} = \frac{2}{3} = (0,(6))$$ Class B has the lowest training $F_1$ score //

$$F_C = \frac{1}{\frac{1}{2}\left(\frac{1}{4/5} + \frac{1}{1}\right)} = \frac{8}{9} = 0,(8)$$

# Homework I – Group 71

(ist1103471, ist1104139)

4)

| $y_1$ | $y_1'$ | $y_2$ | $y_2'$ |
|---|---|---|---|
| 0,24 | 3 | 1 | 8 |
| 0,06 | 2 | 2 | 11 |
| 0,04 | 1 | 0 | 3,5 |
| 0,36 | 5 | 0 | 3,5 |
| 0,32 | 4 | 0 | 3,5 |
| 0,68 | 10 | 2 | 11 |
| 0,9 | 12 | 0 | 3,5 |
| 0,76 | 11 | 2 | 11 |
| 0,46 | 7 | 1 | 8 |
| 0,62 | 9 | 0 | 3,5 |
| 0,44 | 6 | 1 | 8 |
| 0,52 | 8 | 0 | 3,5 |

$\bar{y_1'} = 6,5$

$\bar{y_2'} = 6,5$

$\sum y_1 = \sum y_2 = 78$

$\sum (y_{1i}')^2 = 650$

$\sum (y_{2i}')^2 = 628,5$

$\sum y_{1i}' y_{2i}' = 517,5$

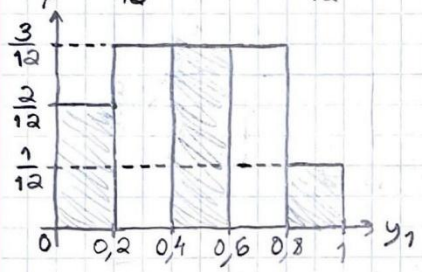
$$\text{Spearman}(y_1, y_2) = \text{Pearson}(y_1', y_2') = \frac{m\sum y_{1i}' y_{2i}' - \sum y_{1i}' \sum y_{2i}'}{\sqrt{(m(\sum y_{1i}'^2) - (\sum y_{1i}')^2) \times (m(\sum y_{2i}'^2) - (\sum y_{2i}')^2)}} =$$

$$= \frac{12 \times 517,5 - 78 \times 78}{\sqrt{(12 \times 650 - 78^2) \times (12 \times 628,5 - 78^2)}} \approx 0,0801$$

5)

$[0; 0,2[, [0,2; 0,4[, [0,4; 0,6[, [0,6; 0,8[, [0,8; 1]$

$\frac{2}{12}$ $\frac{3}{12}$ $\frac{3}{12}$ $\frac{3}{12}$ $\frac{1}{12}$



Aprendizagem 2022/23

**Homework I – Group 71**

(ist1103471, ist1104139)

## II. Programming and critical analysis

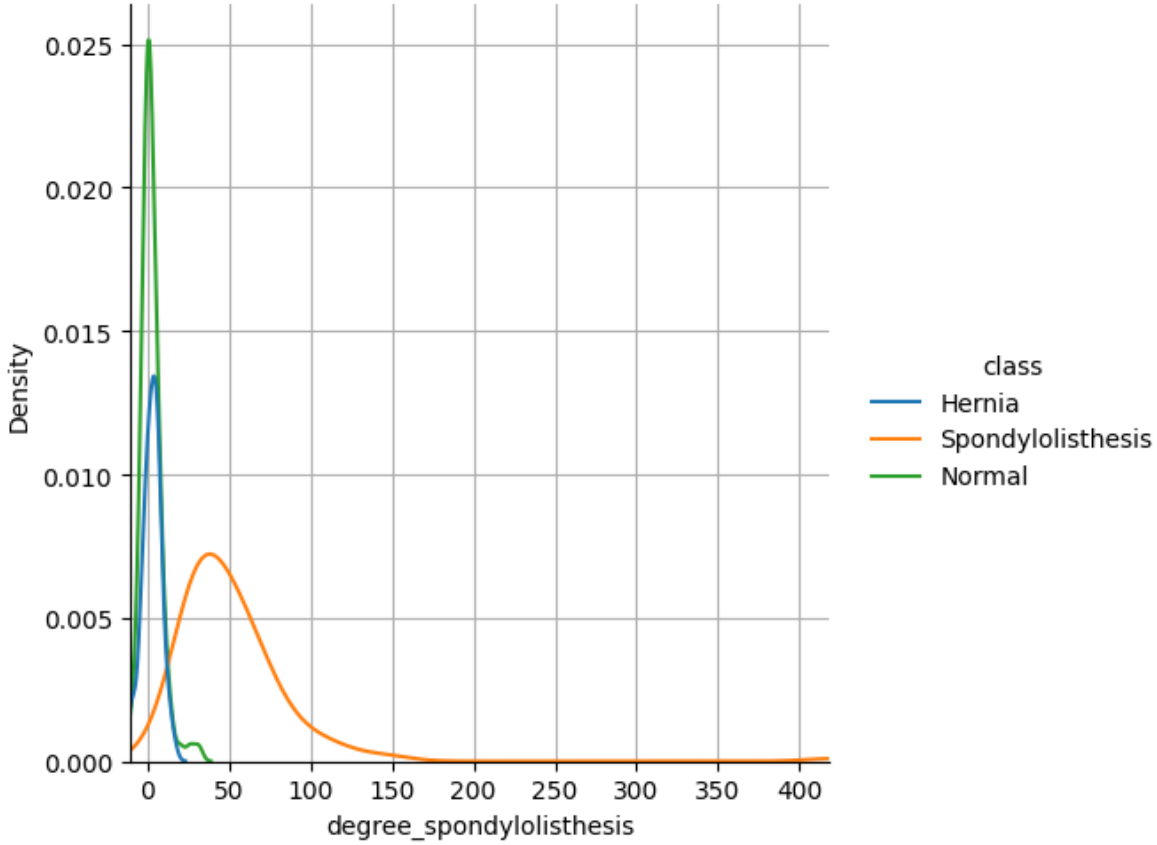
1)

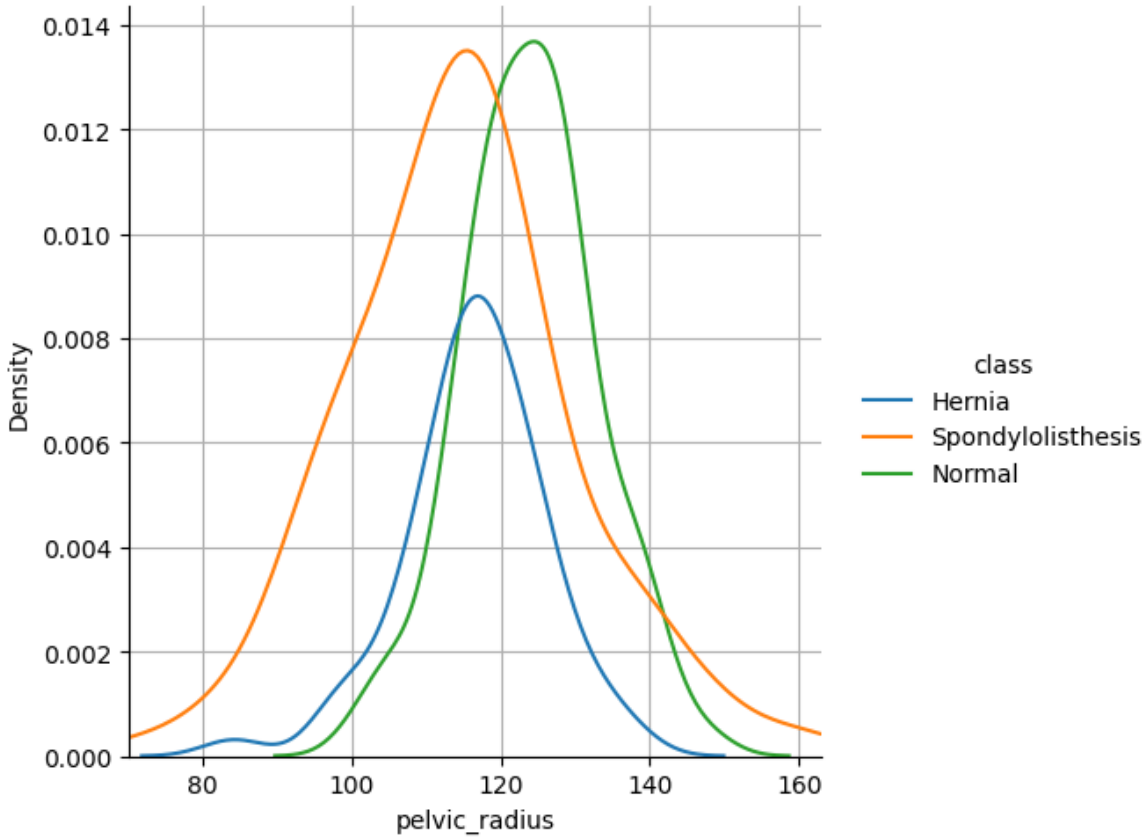
```
from scipy.io.arff import loadarff
from sklearn.feature_selection import f_classif
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.io.arff import loadarff
from sklearn.feature_selection import f_classif
import pandas as pd
from sklearn import metrics, tree
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import  mutual_info_classif

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

X = df.drop('class', axis=1)
y = df['class']

fimportance = f_classif(X, y)

variables = list(df.columns.values)
f_scores, p_values = f_classif(X, y)

highest_discriminative_power_index = np.argmax(f_scores)
lowest_discriminative_power_index = np.argmin(f_scores)
```

```
maxlim1 = df[variables[highest_discriminative_power_index]].max()
minlim1 = df[variables[highest_discriminative_power_index]].min()
sns.displot(df, x=variables[highest_discriminative_power_index], hue="class", kind="kde")
plt.xlim(minlim1, maxlim1)
plt.grid()

maxlim2 = df[variables[lowest_discriminative_power_index]].max()
minlim2 = df[variables[lowest_discriminative_power_index]].min()
sns.displot(df, x=variables[lowest_discriminative_power_index], hue="class", kind="kde")
plt.xlim(minlim2, maxlim2)
plt.grid()
```

**Homework I – Group 71**

(ist1103471, ist1104139)



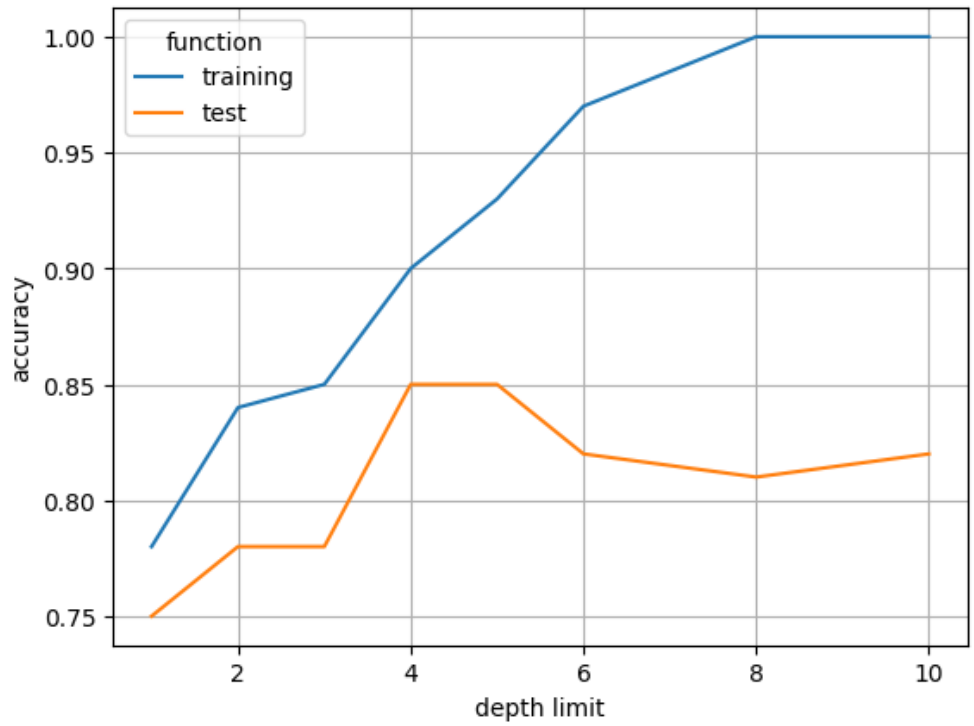



**Homework I – Group 71**

(ist1103471, ist1104139)

**2)**

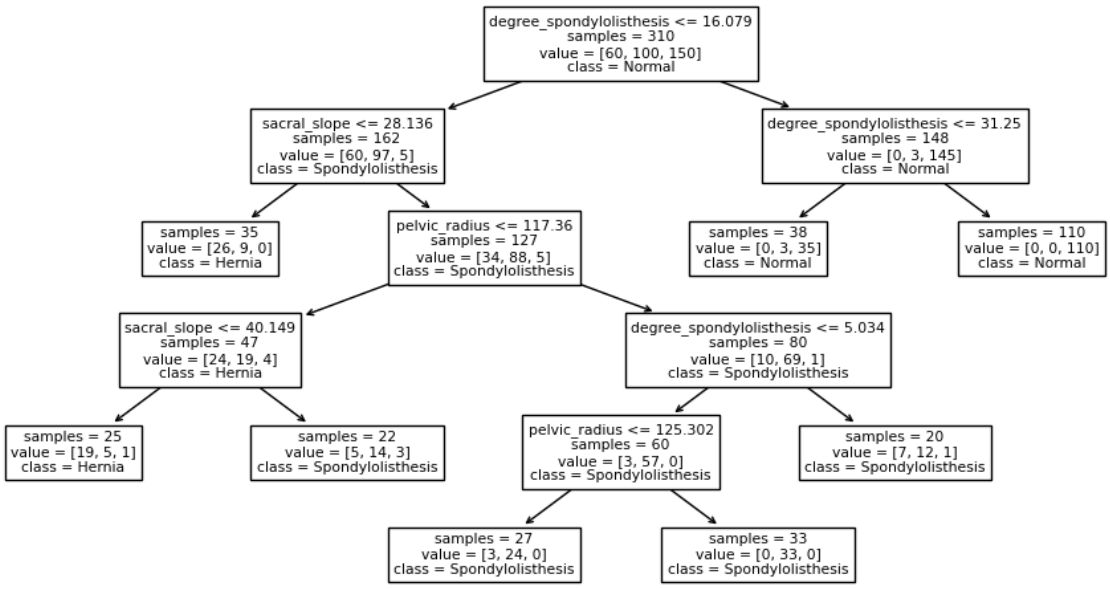
```
depth_limits = [1,2,3,4,5,6,8,10]
training_accs, test_accs = [], []
X_training, X_test, y_training, y_test = train_test_split(X, y, train_size=0.7, stratify=y, random_state=0)
f_scores = mutual_info_classif(X_training, y_training)

for d in depth_limits:
    avrge_train = 0
    avrge_test = 0
    for i in range (0, 10):
        predictor = tree.DecisionTreeClassifier(max_depth=d)
        predictor.fit(X_training, y_training)
        avrge_train = avrge_train + metrics.accuracy_score(y_training, predictor.predict(X_training))
        avrge_test = avrge_test + metrics.accuracy_score(y_test, predictor.predict(X_test))

    training_accs.append(round(avrge_train/10, 2))
    test_accs.append(round(avrge_test/10, 2))


df1 = pd.DataFrame({'depth limit': depth_limits, 'accuracy': training_accs, 'function': 'training'})
df2 = pd.DataFrame({'depth limit': depth_limits, 'accuracy': test_accs, 'function': 'test'})

df_final = pd.concat([df1, df2])

sns.lineplot(data=df_final, x='depth limit', y='accuracy', hue='function')
plt.grid()
```



**3)** According to the plot, previously made, we can see that the testing accuracy is always lower than the training accuracy, this happens because the model tries to recognize patterns present in the train sample, that might not be entirely present in the test sample. Following the increment of the depth limit, we can see that the testing accuracy reaches its peak at 4 and 5, meaning these are the values that make the model (decision tree) generalize better to a different sample than the one that was used to train the model itself. This happens because, the deeper the tree is (more extensive training), the better it gets at predicting outputs for the training sample, i.e. the more overfitted to the training data the model becomes.

**4) i)**

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

class_names = df['class'].unique()
feature_names = list(df.columns.values)

X = df.drop('class', axis=1)
y = df['class']

predictor = tree.DecisionTreeClassifier(min_samples_leaf=20)
predictor.fit(X, y)
figure = plt.figure(figsize=(12, 6))
tree.plot_tree(predictor, feature_names=feature_names, class_names=class_names, impurity=False)
plt.show()
```



**ii)** We can characterize a hernia condition by analyzing the decision tree made in the previous task. By doing so, it is visible that, there is a much higher probability of someone having a hernia condition, according to the training sample, if that same person has a degree of spondylolisthesis lower or equal to 16.079, because there is no leaf of the tree with the hernia class in the right side of the tree. Following down the tree throught the left branch, we can see that, it is almost certain that there is a hernia condition if the sacral slope is lower or equal to 28.136. But even if this feature is higher, there is still a probability of existing a hernia condition, which can be seen in the tree, if the pelvic radius and the sacral slope, are lower or equal to 117.36 and 40.139, respectively. Based on the results, we can conclude, that the most decisive features are the degree of spondylolisthesis and the sacral slope, accordingly.

**END**