TÉCNICO LISBOA

Aprendizagem 2022/23

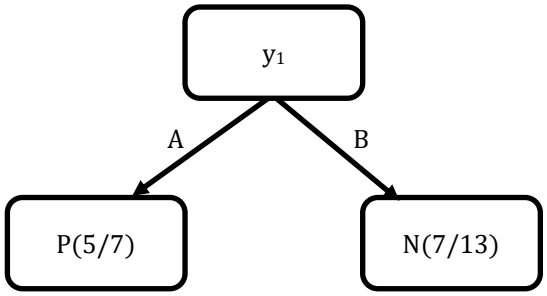
Homework I – Group 010

I. Pen-and-paper

1)

		Prediction	
		P	N
Real	P	8	3
	N	4	5

2)



$$recall = \frac{TP}{TP + FN} = \frac{5}{5+6} = \frac{5}{11}$$

$$\text{precision} = \frac{TP}{TP + FP} = \frac{5}{5+2} = \frac{5}{7}$$

$$F1 = \frac{2}{P^{-1} + R^{-1}} = \frac{2}{\frac{7}{5} + \frac{11}{5}} = \frac{5}{9}$$

3) The left tree path was not further decomposed because the data split is not statistically significant and the instances in the conditional dataset are correctly classified or no more variables available, decreasing the risk of overfitting.

4)

$$\text{E(class)} = -\sum \text{p(class} = x) * \log_2 \text{p(class} = x) = -\left(\frac{11}{20}\log_2\frac{11}{20} + \frac{9}{20}\log_2\frac{9}{20}\right) = 0.9928$$

$$\text{E(class} \mid y_1 = A) = -\sum_{x \in y_1} \text{p(class} = x \mid y_1 = A) * \log_2 \text{p(class} = x \mid y_1 = A)$$

$$= -\left(\frac{5}{7}\log_2\frac{5}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 0.8631$$

$$\text{E(class} \mid y_1 = B) = -\sum_{x \in y_1} \text{p(class} = x \mid y_1 = B) * \log_2 \text{p(class} = x \mid y_1 = B)$$

$$= -\left(\frac{5}{7}\log_2\frac{5}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 0.9957$$

$$\text{E(class} \mid y_1) = \sum_{x \in y_1} \text{p}(y_1 = k) * \text{E(class} \mid y_1 = k) = \frac{7}{20} * 0.8631 + \frac{13}{20} * 0.9957 = 0.94929$$

$$\text{IG(class} \mid y_1) = \text{E(class)} - \text{E(class} \mid y_1) = 0.9928 - 0.94929 = 0.04351$$

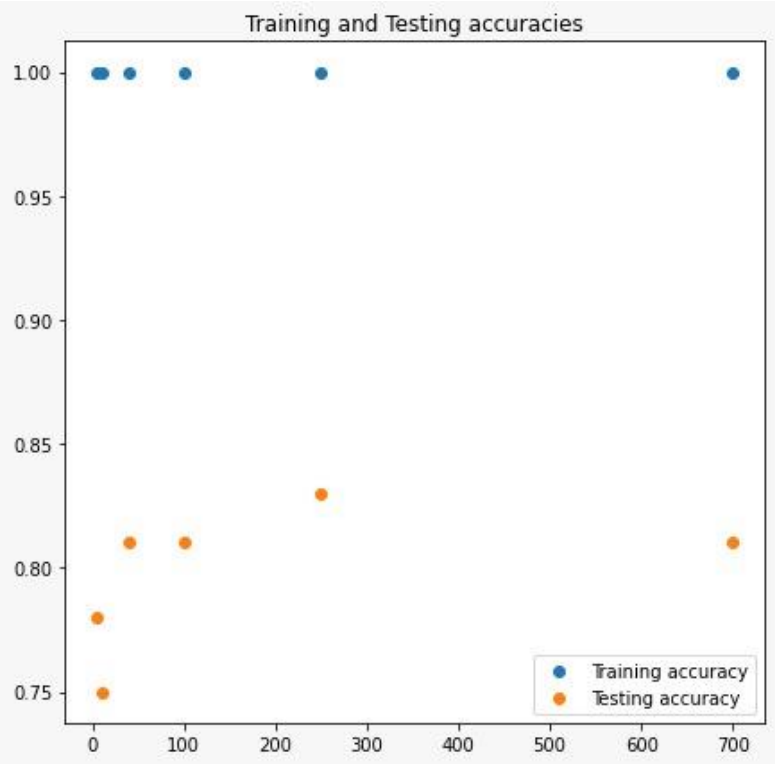
TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I – Group 010

II. Programming and critical analysis

5)



6) Because we did not limit the depth of the decision tree, the tree adapted perfectly to the data used for the training (x_train and y_train), which data used for the creation of the tree. Which consequently means that the tree will always give the correct answer according to the training data, making the accuracy always equal to 1.

III. APPENDIX

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

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework I – Group 010

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

target = df['class']

df.drop('class', axis=1)
features = df.columns[0:-1]
ig = {}
for v in features:
    ig[v] = mutual_info_classif(df[v].to_numpy().reshape(-1, 1), target, random_state=1)

sorted_igs = sorted(ig, key=ig.get, reverse=True)

accuracy_test, accuracy_train = [], []
n_features = (5, 10, 40, 100, 250, 700)
for i in n_features:
    aux = sorted_igs[:i]
    X_train, X_test, y_train, y_test = train_test_split(df[aux], target, test_size = 0.3,
random_state=1, stratify=target)

    predictor = tree.DecisionTreeClassifier(random_state=1)
    predictor.fit(X_train, y_train)

    y_pred = predictor.predict(X_train)
    accuracy_train += [round(metrics.accuracy_score(y_train, y_pred), 2)]

    y_pred = predictor.predict(X_test)
    accuracy_test += [round(metrics.accuracy_score(y_test, y_pred), 2)]


figure = plt.figure(figsize=(7, 7))
plt.title("Training and Testing accuracies")
plt.scatter(n_features, accuracy_train, label='Training accuracy')
plt.scatter(n_features, accuracy_test, label='Testing accuracy')
plt.legend(loc ="lower right")
```

END