

Aprendizagem 2022
Homework I – Group 58

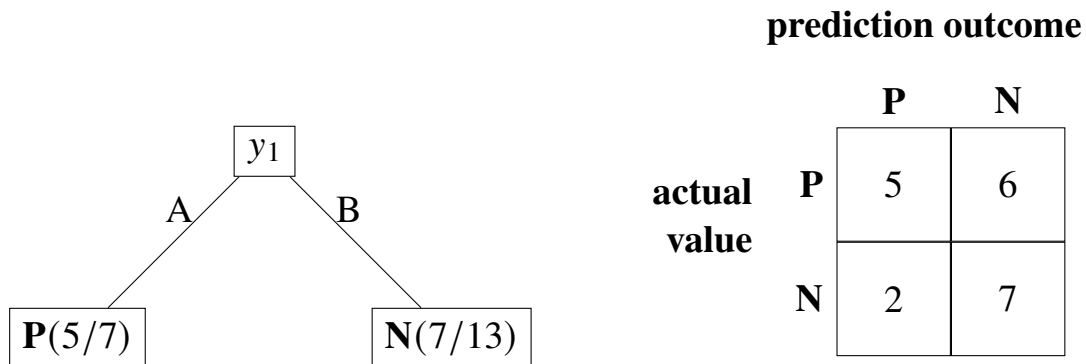
Part I: Pen and paper

1.

prediction outcome

		P	N
actual value	P	8	3
	N	4	5

2. Post-pruning of the given tree under a maximum depth of 1:



$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

$$F1 = \frac{5}{5 + \frac{1}{2}(2 + 6)} = \frac{5}{9}$$

3. One reason why the left tree path was not further decomposed is because in the data set, whenever the variable y_1 had the value A , the class was always P , so there was no need to have more nodes.

Another reason is due to reducing the tree complexity and increasing its efficiency and speed and, therefore, predictive power.

4. $IG(class \mid y_1) = E(class) - E(class \mid y_1)$

$$E(class) = - \sum_{x \in class} P(class = x) * \log_2[P(class = x)]$$

$$= -(\frac{11}{20} * \log_2 \frac{11}{20} + \frac{9}{20} * \log_2 \frac{9}{20})$$

$$\simeq 0.992774$$

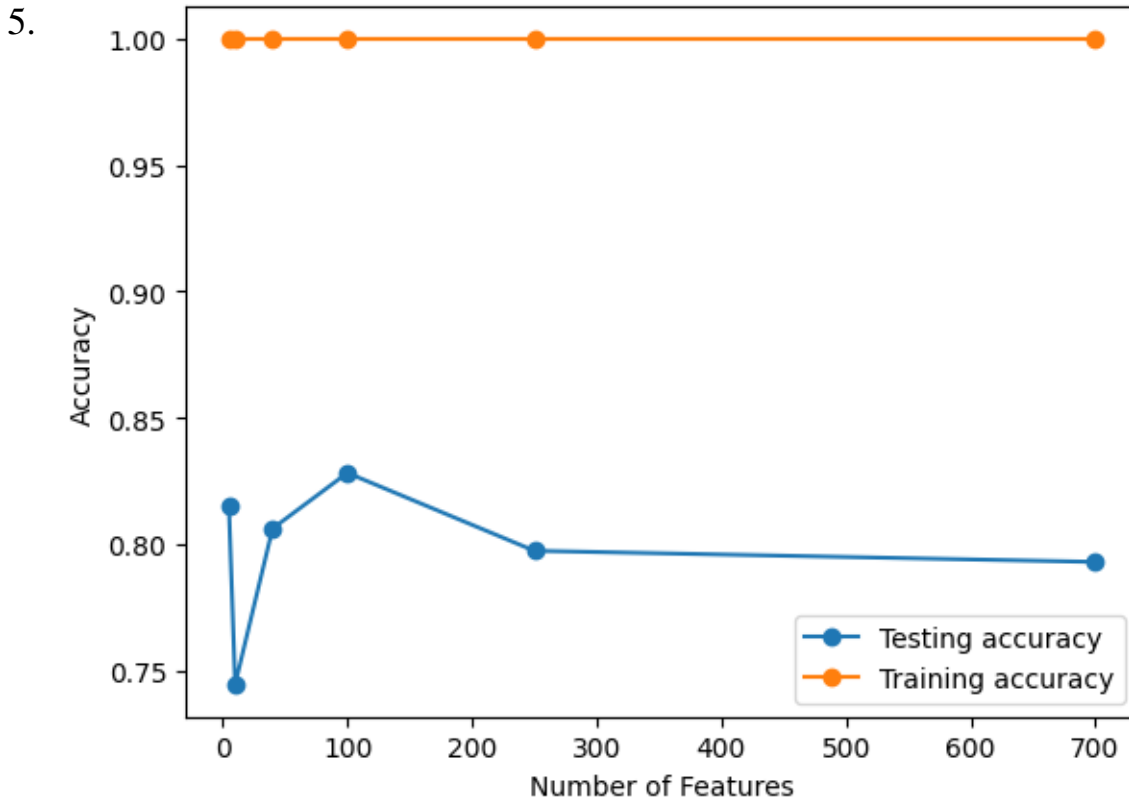
$$E(class|y_1) = \sum_{x \in y_1} P(y_1 = x) * E(class|y_1 = x)$$

$$= \frac{7}{20}(-[\frac{5}{7} * \log_2 \frac{5}{7} + \frac{2}{7} * \log_2 \frac{2}{7}]) + \frac{13}{20}(-[\frac{6}{13} * \log_2 \frac{6}{13} + \frac{7}{13} * \log_2 \frac{7}{13}])$$

$$\simeq 0.949315$$

$$IG(class|y_1) = 0.992774 - 0.949315 \simeq 0.042459$$

Part II: Programming



6. The training accuracy is persistently 1 because the data model is trained too well on our training data set, therefore, it predicts the correct values 100% of the time on that data set.

Appendix

```
1 from matplotlib import markers
2 import pandas as pd
3 from scipy.io.arff import loadarff
4 from sklearn.feature_selection import SelectKBest, mutual_info_classif
5 from sklearn.model_selection import train_test_split
6 from sklearn.tree import DecisionTreeClassifier
7 from sklearn import metrics
8 import matplotlib.pyplot as plt
9 import seaborn as sns
10 num_feat = [5, 10, 40, 100, 250, 700]
11 test_acc = []
12 train_acc = []
13
14 data = loadarff('pd_speech.arff')
15 df = pd.DataFrame(data[0])
16 df['class'] = df['class'].str.decode('utf-8')
17 X = df.drop('class', axis=1)
18 y = df['class']
19
20 for i in num_feat:
21     X_new = SelectKBest(mutual_info_classif, k=i).fit_transform(X, y)
22     X_train, X_test, y_train, y_test = train_test_split(X_new, y, train_size = 0.7,
23     stratify = y, random_state = 1)
24     classifier = DecisionTreeClassifier(random_state=1)
25     classifier = classifier.fit(X_train, y_train)
26
27     pred = classifier.predict(X_test)
28     test_acc.append(metrics.accuracy_score(y_test, pred))
29     train_acc.append(classifier.score(X_train, y_train))
30
31 fig, ax = plt.subplots()
32 ax.plot(num_feat, test_acc, label='Testing accuracy', marker='o')
33 ax.plot(num_feat, train_acc, label='Training accuracy', marker='o')
34 ax.set_xlabel("Number of Features")
35 ax.set_ylabel("Accuracy")
36 ax.legend()
37
38 plt.show()
```