Aprendizagem 2021/22

Homework I – Group XXX

Aprendizagem 2021/22

Homework I – Group 29

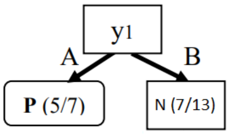
I. Pen-and-paper

1) #P = 5 + 3 + 3 = 11

#N = 2 + 5+ 2 = 9

	P	N
P	8	4
N	3	5

2)



	P	N
P	5	2
N	6	7

$$P = \frac{TP}{TP+FP} = \frac{5}{5+2} = \frac{5}{7} \sim 0.71428$$

$$R = \frac{TP}{TP+FN} = \frac{5}{5+6} = \frac{5}{11} \sim 0.45454$$

$$\frac{1}{F1} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right) = \frac{1}{2}\left(\frac{7}{5} + \frac{11}{5}\right) = \frac{9}{5}$$

$$F1 = \frac{5}{9} \sim 0.55556$$

3) The left tree path wasn't further decomposed in order to avoid overfitting and because further splitting would increase the impurity of the node.

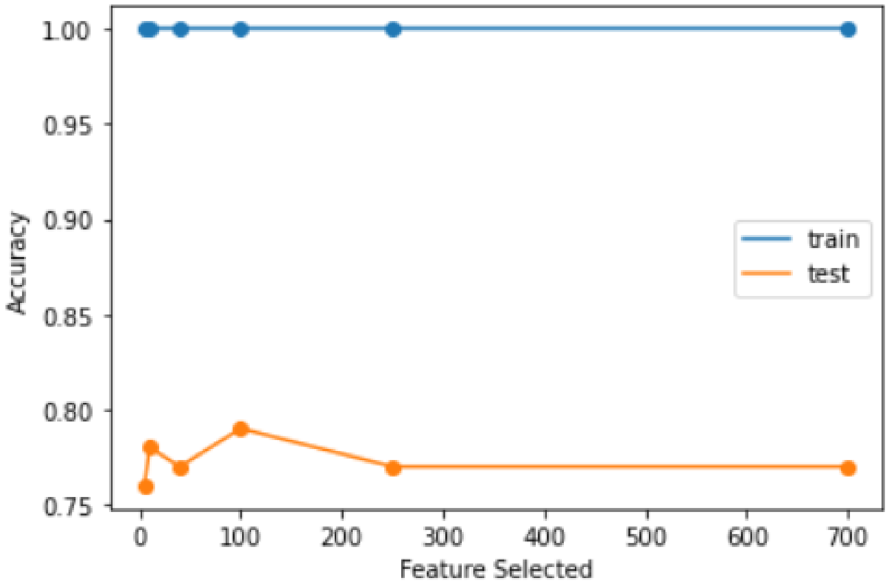
4) $IG(y_1) = H(X) - \sum_{i=1}^{k} \frac{|X_1|}{|X|} H(y_1)$ $\quad H(X) = -\frac{11}{20}\log_2\left(\frac{11}{20}\right) - \frac{9}{20}\log_2\left(\frac{9}{20}\right) \sim 0.99277$

$$\sum_{i=1}^{k} \frac{|X_i|}{|X|} H(y_1) = \frac{7}{20}\left[-\frac{5}{7}\log_2\left(\frac{5}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right)\right] + \frac{13}{20}\left[-\frac{6}{13}\log_2\left(\frac{6}{13}\right) - \frac{7}{13}\log_2\left(\frac{7}{13}\right)\right] \approx 0.94932$$

$IG(y_1) = 0.99277 - 0.94932 = 0.04345$

II. Programming and critical analysis

1. [1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
   [0.77, 0.76, 0.82, 0.81, 0.80, 0.81]



2. Training accuracy is consistently 1 because it is the set used to train the decision tree.

III. APPENDIX

```
import pandas as pd
import numpy as np
from sklearn import metrics, datasets, tree
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import SelectKBest
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from scipy.io.arff import loadarff
# Loading data from arff file
data = loadarff('pd_speech.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
# Splitting data into X and y
X = df.drop("class", axis=1)
y = df['class']
df.head()
feats = [5, 10, 40, 100, 250, 700]
train = np.zeros(len(feats))
```

```
test = np.zeros(len(feats))
# For loop to iterate through the different number of features
for index, el in enumerate(feats):
        # Feature selection using mutual information
        X_newbest = SelectKBest(score_func=mutual_info_classif, k=el).fit_transform(X, y)
        # Splitting the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split( X_newbest, y, train_size=0.3,
random_state=1)
        # Creating the decision tree
        D_Tree = DecisionTreeClassifier()
        # Training the decision tree
        predictor = D_Tree.fit(X_train, y_train)
        # Predicting the training and testing sets
        y_train_predict = predictor.predict(X_train)
        y_test_predict = predictor.predict(X_test)
        # Calculating the accuracy of the training and testing sets
        train[index] = round(metrics.accuracy_score(y_train, y_train_predict), 2)
        test[index] = round(metrics.accuracy_score(y_test, y_test_predict), 2)
# plot train and test
plt.plot(feats, train, label='train')
plt.plot(feats, test, label='test')
plt.xlabel('Feature Selected')
plt.ylabel('Accuracy')
# scatter train and test
plt.scatter(feats, train)
plt.scatter(feats, test)
plt.legend()
plt.show()
```

END