TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 039

I. Pen-and-paper

1)

$$X = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 5 \\ 1 & 0 & 2 & 4 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 0 & 7 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 2 \\ 1 & 0 & 2 & 9 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 0 \\ 6 \\ 4 \\ 5 \\ 7 \end{bmatrix}$$

$$\Phi = \left(\Phi^{\dagger}\Phi\right)^{-1}\Phi^{\dagger}$$

$$\Phi = \begin{bmatrix} 1 & 1.414214 & 2 & 2.828427 \\ 1 & 5.196152 & 27 & 140.296115 \\ 1 & 4.472136 & 20 & 89.442719 \\ 1 & 3.741657 & 14 & 52.383203 \\ 1 & 7.280110 & 53 & 385.845824 \\ 1 & 1.732051 & 3 & 5.196152 \\ 1 & 2.828427 & 8 & 22.627417 \\ 1 & 9.219544 & 85 & 783.661279 \end{bmatrix}$$ (Ivertible Matrix)

$$\Phi^{\dagger} = \begin{bmatrix} 1.000000 & 1.000000 & 1.000000 & 1.000000 & 1.000000 & 1.000000 & 1.000000 & 1.000000 \\ 1.414214 & 5.196152 & 4.472136 & 3.741657 & 7.280110 & 1.732051 & 2.828427 & 9.219544 \\ 2.000000 & 27.000000 & 20.000000 & 14.000000 & 53.000000 & 3.000000 & 8.000000 & 85.000000 \\ 2.828427 & 140.296115 & 89.442719 & 52.383203 & 385.845824 & 5.196152 & 22.627417 & 783.661279 \end{bmatrix}$$

$$\left(\Phi^{\dagger}\Phi\right)^{-1} = \begin{bmatrix} 1.768589 & -0.081148 & -0.669408 & -1.006877 & 1.379417 & 0.905129 & -0.785045 & -0.510657 \\ -1.064351 & -0.031859 & 0.531217 & 0.903994 & -1.306988 & -0.392170 & 0.850102 & 0.510055 \\ 0.193258 & 0.043578 & -0.090735 & -0.186777 & 0.323205 & 0.052375 & -0.195406 & -0.139498 \\ -0.010744 & -0.004349 & 0.004405 & 0.010938 & -0.021355 & -0.002207 & 0.012279 & 0.011034 \end{bmatrix}$$

$$\Phi\, Z = W = \begin{bmatrix} 4.583521 \\ -1.687205 \\ 0.337737 \\ -0.013307 \end{bmatrix} \qquad f(x, w) = 4.583521 - 1.687205x + 0.337737x^2 - 0.013307x^3$$

2)

	Y1	Y2	Y3	Output	Predicted
X9	2	0	0	2	2.4536069711
X10	1	2	1	4	2.2815859319

RMSE=1.25672316

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 039

3)

IG=[0.23869281 0.21576155 0.]

Chosen var -> Y1

If y1=2 -> 1

If y1=1 -> 0

If Y1=0; Decision made by Y3 (tied between Y2 and Y3) (IG(Y1=0) = [0. 0. 0.])

If Y3=0 -> 0

If Y3=1 -> 1

4)

$Accuracy(polinomial) = \frac{TP+TN}{All} = \frac{0+1}{2} = 50\%$

$Accuracy(tree) = \frac{TP+TN}{All} = \frac{0+0}{2} = 0\%$

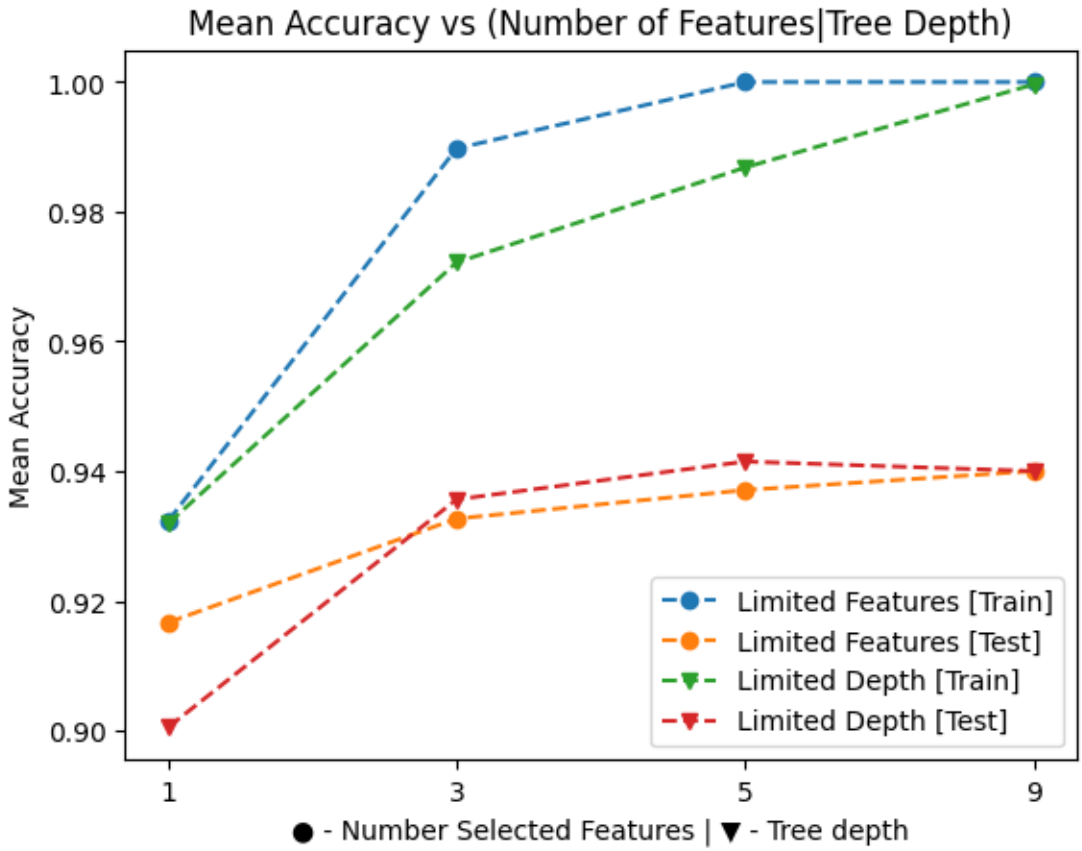
II. Programming and critical analysis

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 039

5)



6) The observed correlation could be caused by:

- The fact that limiting the tree depth limits the number of features used to classify.
- The inverse, limiting the number of features (by selecting the k best) is already part of the constructing the decision tree model.

We can then conclude that, limiting a decision tree by either its depth or by the number of features gives, approximately the same results since, either one limits the other due to the fact that when limiting the depth to 1, it will be most likely to use the feature with the biggest information gain, the same way we end up choosing the best feature on the same metric to build the tree. This happen when k is 2 as well, if we limit by depth 2 it will most likely recure to the best feature on the first layer and, one the second choose the second best. Again, the same way we would choose the 2 best feature and then build the tree, ending up with a tree with the most valuable feature on the first layer and the second most valuable on the second. This is true for bigger k values as well.

7) We can see the accuracy of the prediction on the test subjects stabilize and even decrease slightly for k bigger than 5, showing overfitting to the training data, meaning we would need to either need to increase the amount of training data if possible, or less optimally select a simpler model. So, in this case the, the chosen depth is 5.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 039

III. APPENDIX

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.io import arff
from scipy.sparse.construct import random
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_validate
from sklearn.model_selection import StratifiedKFold
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import SelectKBest


GROUPN = 95

# Extract Data
data = pd.DataFrame( arff.loadarff( "breast.w.arff" )[0] )
# Elements array
X = data.drop(columns=data.columns[-1]).to_numpy().astype(int)
# Results array binarized
Y = data[data.columns[-1]].replace(b'benign', 0).replace(b'malignant', 1)

knn = StratifiedKFold(n_splits=10, random_state=GROUPN, shuffle=True)

def q5_1():

    train_accuracy_list, test_accuracy_list= list(), list()

    for k_val in [1, 3, 5, 9]:

        KBest = SelectKBest(mutual_info_classif, k=k_val).fit_transform(X, Y)

        clf = DecisionTreeClassifier(random_state=GROUPN)

        accuracy = cross_validate(clf, KBest, Y, scoring="accuracy", return_train_score=True)

        train_accuracy_list.append(sum(accuracy["train_score"])/len(accuracy["train_score"]))
        test_accuracy_list.append(sum(accuracy["test_score"])/len(accuracy["test_score"]))

    plt.plot(range(4), train_accuracy_list,label="Limited Features [Train]", linestyle="--",
marker="o")
    plt.plot(range(4), test_accuracy_list,label="Limited Features [Test]", linestyle="--",
marker="o")

def q5_2():

    train_accuracy_list, test_accuracy_list= list(), list()

    for k_val in [1, 3, 5, 9]:

        clf = DecisionTreeClassifier(max_depth=k_val, random_state=GROUPN)

        accuracy = cross_validate(clf, X, Y, scoring="accuracy", return_train_score=True)
```

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework II – Group 039

```
        train_accuracy_list.append(sum(accuracy["train_score"])/len(accuracy["train_score"]))
        test_accuracy_list.append(sum(accuracy["test_score"])/len(accuracy["test_score"]))


    plt.plot(range(4), train_accuracy_list,label="Limited Depth [Train]", linestyle="--",
marker="v")
    plt.plot(range(4), test_accuracy_list,label="Limited Depth [Test]", linestyle="--", marker="v")

q5_1()
q5_2()

plt.title('Mean Accuracy vs (Number of Features|Tree Depth)')
plt.xlabel('● - Number Selected Features | ▼ - Tree depth')
plt.xticks(range(4), [1, 3, 5, 9])
plt.ylabel('Mean Accuracy')
plt.savefig("graph.png")
plt.show()
```

END