TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 010

I. Pen-and-paper

1. $\mathbf{w} = (\phi^T\phi)^{-1}\phi^T z$

$$\phi = \begin{bmatrix} 1 & \sqrt{2} & 2 & (\sqrt{2})^3 \\ 1 & \sqrt{27} & 27 & (\sqrt{27})^3 \\ 1 & \sqrt{20} & 20 & (\sqrt{20})^3 \\ 1 & \sqrt{14} & 14 & (\sqrt{14})^3 \\ 1 & \sqrt{53} & 53 & (\sqrt{53})^3 \\ 1 & \sqrt{3} & 3 & (\sqrt{3})^3 \\ 1 & \sqrt{8} & 8 & (\sqrt{8})^3 \\ 1 & \sqrt{85} & 85 & (\sqrt{85})^3 \end{bmatrix} \qquad (\phi^T\phi)^{-1} = \begin{bmatrix} 8.196 & -6.231 & 1.305 & -0.0793 \\ -6.231 & 5.078 & -1.104 & 0.0686 \\ 1.305 & -1.104 & 0.247 & -0.0157 \\ -0.0793 & 0.0686 & -0.0157 & 0.001 \end{bmatrix}$$

$(\phi^T\phi)^{-1}\phi^T =$ (Moore-Penrose Matrix)

$$\begin{bmatrix} 1.769 & -0.081 & -0.669 & -1.007 & 1.379 & 0.905 & -0.785 & -0.511 \\ 1.064 & -0.032 & 0.531 & 0.904 & -1.307 & -0.392 & 0.850 & 0.510 \\ 0.193 & 0.044 & -0.091 & -0.187 & 0.323 & 0.052 & -0.195 & -0.139 \\ 0.011 & -0.004 & 0.004 & 0.011 & -0.021 & -0.002 & 0.012 & 0.011 \end{bmatrix}$$

$$\mathbf{w}^T = \begin{bmatrix} 4.58352122 & -1.6872048 & 0.33773733 & -0.01330674 \end{bmatrix}$$

2. $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(output(x_i) - f(x_i, w))^2}$

$f(\mathbf{x}_9, \mathbf{w}) = 4.58352122 - 1.6872048 \times 2 + 0.33773733 \times 4 - 0.01330674 \times \sqrt{4}^3 = 2.453606971151089$

$f(\mathbf{x}_{10}, \mathbf{w}) = 4.58352122 - 1.6872048 \times \sqrt{6} + 0.33773733 \times 6 - 0.01330674 \times \sqrt{6}^3 = 2.2815859319033467$

$RMSE = \sqrt{\frac{1}{2}((2 - 2.453606971151089)^2 + (4 - 2.2815859319033467)^2)} = 1.2567231583983312$

3. To apply an equal depth binarization we calculated the median of y_3 which is 3.5 and all the values above were mapped to 1 whereas all the below are mapped to 0.

New dataset:

	y_1	y_2	y_3	*output*
$\mathbf{x}_1$	1	1	0	N
$\mathbf{x}_2$	1	1	1	N
$\mathbf{x}_3$	0	2	1	N
$\mathbf{x}_4$	1	2	0	N
$\mathbf{x}_5$	2	0	1	P
$\mathbf{x}_6$	1	1	0	P
$\mathbf{x}_7$	2	0	0	P
$\mathbf{x}_8$	0	2	1	P
$\mathbf{x}_9$	2	0	0	N
$\mathbf{x}_{10}$	1	2	0	P

Which variable has the highest IG?

$IG(out|y_i) = E(out) - E(out|y_i)$

$E(out) = -\sum p_i log_2(p_i) = -E(\frac{1}{2}, \frac{1}{2}) = 1$

$E(out|y_1) = -(\frac{1}{4}E(\frac{1}{2}, \frac{1}{2}) + \frac{1}{4}E(1) + \frac{1}{2}E(\frac{3}{4}, \frac{1}{4})) = 0.6556$

$E(out|y_2) = -(\frac{1}{4}E(1) + \frac{3}{8}E(\frac{2}{3}, \frac{1}{3}) + \frac{3}{8}E(\frac{2}{3}, \frac{1}{3})) = 0.6887$

$E(out|y_3) = -(\frac{1}{2}E(\frac{1}{2}, \frac{1}{2}) + \frac{1}{2}E(\frac{1}{2}, \frac{1}{2}) = 1$

$IG(out|y_1) = 0.34436 \quad IG(out|y_2) = 0.31128 \quad IG(out|y_3) = 0$

$\therefore y_1$ has the highest IG and it goes to the root of the tree

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 010

Which variable has the highest IG knowing that $y_1 = 0$?

$E_{y_1=0}(out) = -\sum p_i log_2(p_i) = -E(\frac{1}{2},\frac{1}{2}) = 1$

$E_{y_1=0}(out|y_2) = E(out|y_3 \wedge y_1 = 0) = -E(\frac{1}{2},\frac{1}{2}) = 1$

$IG_{y_1=0}(out|y_2) = IG_{y_1=0}(out|y_3) = 0$

We can not conclude anything by analysing y_2 and y_3 because they do not give any information about the output when $y_1 = 0$. Therefore, when $y_1 = 0$ there is an equal probability of the output being 0 or 1.

And knowing that $y_1 = 1$?

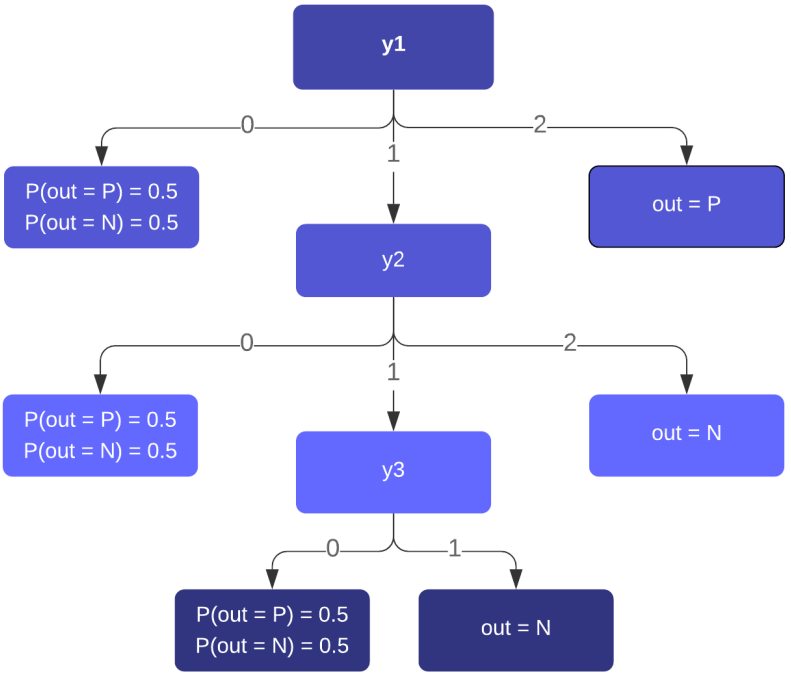
$E_{y_1=1}(out) = -\sum p_i log_2(p_i) = -E(\frac{3}{4},\frac{1}{4}) = 0.811278$

$E_{y_1=1}(out|y_2) = E_{y_1=1}(out|y_3) = -(\frac{3}{4}E(\frac{2}{3},\frac{1}{3}) + \frac{1}{4}E(1)) = 0.688722$

$IG_{y_1=1}(out|y_2) = IG_{y_1=1}(out|y_3)$ → We can choose one randomly. We chose y2 to be the root of the second level.

In all observations where $y_1 = 2$, $output = P$.

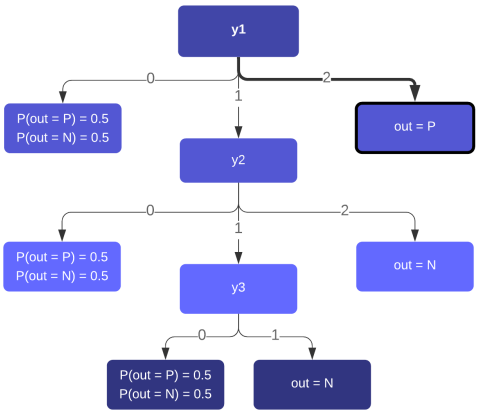
When we decide to look at y_2 after knowing that $y_1 = 1$, in all observations where $y_2 = 2$, $output = N$. There are no observations for $y_1 = 1 \wedge y_2 = 0$, so the probability of the output being 0 or 1 is the same. In the only observation where $y_1 = 1 \wedge y_2 = 0 \wedge y_3 = 1$, $output = N$ and in the same conditions but with $y_3 = 0$ there are no observations, so again the probability of the possible outputs is equal.

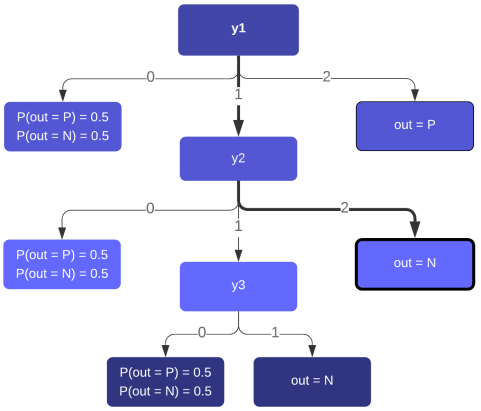


Aprendizagem 2021/22

Homework I – Group 010

4. $\mathbf{x}_9 = (y_1 = 2, y_2 = 0, y_3 = 0, output = N)$ $\mathbf{x}_{10} = (y_1 = 1, y_2 = 2, y_3 = 1, output = P)$

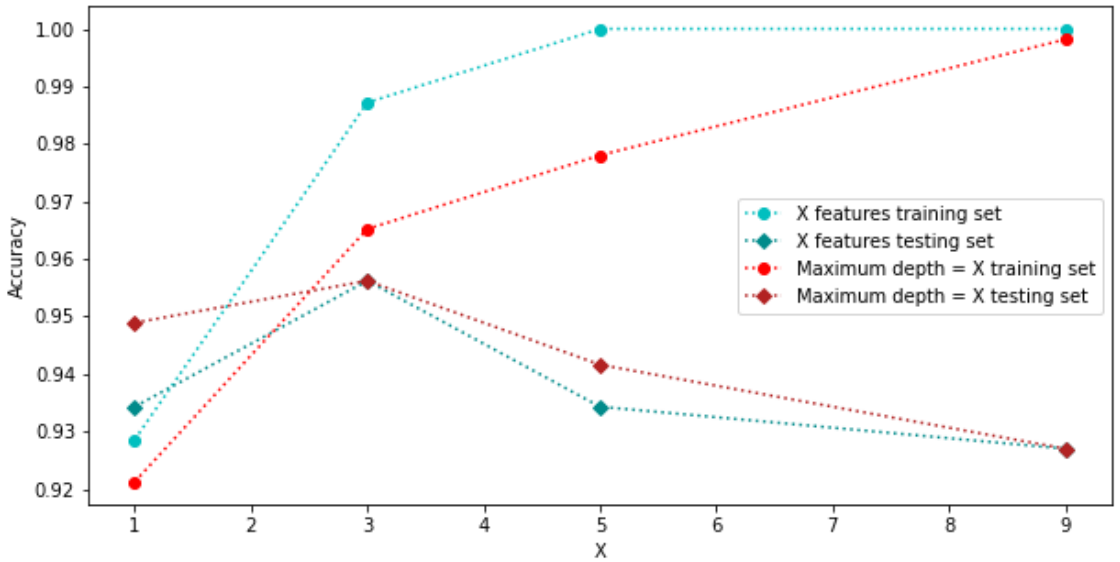




$\mathbf{x}_9$ is classified as P and the true value is N. $\mathbf{x}_{10}$ is classified as N and the true value is P. Accuracy = 0.

II. Programming and critical analysis

5.



6. The number of features considered at each split of a decision tree may reduce variance and limit overfitting. As we can see in the plot, as the number of features considered gets higher the accuracy in the train set also gets higher and, on the contrary, for the test set the accuracy gets lower when all the features are being considered. Here we can see that the model is getting too adapted to the train set (overfit) and the model underperforms on the test set. When only one feature is being considered, the accuracy is also lower, probably because the model is too simple and underfits the data.

The maximum tree depth also influences the accuracy of the decision tree. In general, the deeper the tree grows, the more complex and with more splits the model becomes, which means that it will fit perfectly for the training data and will not be able to generalize well on the testing set. The data on the plot confirms this. The accuracy of the train set gets higher with the increase of maximum depth, however, it gets lower on the test set.

We can also relate the number of features and the maximum depth of the tree and see how they influence the accuracy of the model simultaneously. When the features are discrete, the number of features limits the maximum depth of the tree. The contrary is not necessarily true because the maximum depth of the tree does not say much about the number of features being considered. Given what was concluded above, we can see that the higher the number of features and the maximum depth, the lower the accuracy on the testing set is because the model overfits the train data. We can also see this on the plot by observing the 'X features testing set' and 'Maximum depth = X testing set' lines as we can notice that these lines grow and decrease in a similar way. In conclusion, we may say that the accuracy varies similarly with the variation of these two characteristics of the model.

7. After using the Grid Search with cross validation, which is the process of performing hyperparameter tuning in order to determine the optimal values for a given model, we found that the best maximum depth value is 5. The parameters of the estimator used to apply the GridSearchCV (from sklearn) methods are optimized by cross-validated grid-search over a parameter grid, and in this case the parameter being optimized is the maximum depth of the tree. We splitted the data using the default 5-fold cross validation and after running the program several times, the best value for the maximum depth was either 5 or 9. We decided to choose 5 as the best value because if we select a value that is too high then the decision tree may overfit the training data which may cause the test error to increase, as seen in the above plot.

Note: Despite the fact that in the plot of question 5 the depth with the best accuracy is 3, we did not use cross validation to split that data. When we do that (like we are doing here with grid-search cv) we should get more accurate results to make conclusions about what is the best hyperparameters of a model.

TÉCNICO LISBOA

Aprendizagem 2021/22

Homework I – Group 010

III. APPENDIX

```
#---------------------------------------- Parse of input ------------------------------------#
import pandas as pd
from matplotlib import pyplot as plt
from scipy.io.arff import loadarff
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.model_selection import GridSearchCV

raw_data = loadarff('breast.w.arff')
df_data = pd.DataFrame(raw_data[0])  # converting data to a pandas DataFrame
df_data = df_data.dropna()  # all rows with Na values are dropped
df_data['Class'].replace({b'malignant': 1, b'benign': 0}, inplace=True)

#-------- Train and test of a decision tree classifier varying #features and max-depth ---------#
data, target = df_data.drop(columns='Class'), df_data['Class']
tra_acc_features, test_acc_features, tra_acc_depth, test_acc_depth = [], [], [], [] # to save
accuries to be plotted
values = [1,3,5,9] # number of the features and maximum depth to be looped through
x_train, x_test, y_train, y_test = train_test_split(data, target, test_size=0.3, random_state=10)
for v in values:
    kbest = SelectKBest(mutual_info_classif, k = v) # select k best features using MI value
    kbest.fit(data, target)
    cols = kbest.get_support(indices=True) # get the names of the best k features
    # gets only the columns of the featured selected
    x_train_features, x_test_features = x_train.iloc[:, cols], x_test.iloc[:, cols]
    clf_features = DecisionTreeClassifier(criterion= "entropy")
    clf_features.fit(x_train_features, y_train) # train max_features tree
    # max_depth is defined as parameter
    clf_depth = DecisionTreeClassifier(criterion = "entropy", max_depth = v)
    clf_depth.fit(x_train, y_train) #train max_depth tree
    tra_acc_features.append(clf_features.score(x_train_features, y_train)) # test on train set
    test_acc_features.append(clf_features.score(x_test_features, y_test)) # test on test set
    tra_acc_depth.append(clf_depth.score(x_train, y_train))
    test_acc_depth.append(clf_depth.score(x_test, y_test))

plt.figure(figsize=(10,5))
plt.plot(values, tra_acc_features, 'co:', label="X features training set")
plt.plot(values, test_acc_features, 'darkcyan',marker='D', linestyle=":", label="X features
testing set")
plt.plot(values, tra_acc_depth, 'ro:', label="Maximum depth = X training set")
plt.plot(values, test_acc_depth, color="firebrick", marker='D', linestyle=":", label="Maximum
depth = X testing set")
plt.xlabel('X')
plt.ylabel('Accuracy')
plt.legend()
plt.savefig('plots.png')

#------------------------- Grid-Search with CV to find the best max-depth ----------------------#
parameters = {'max_depth':[1,3,5,9]}
# decides which is the best hyperparameter for the decision tree
clf = GridSearchCV(DecisionTreeClassifier(), parameters)
clf.fit(data, target)
print ("Best score", clf.best_score_," Best depth", clf.best_params_)
```

END