MACHINE LEARNING - CS60050

ASSIGNMENT 1 - Decision trees

GROUP-DETAILS:
Group - 17
Jatoth Charan Sai - 19CS10035
Rahul Eaga - 19CS10029

CLASS's and FUNCTION'S USED

class **Node**:

1. It stores the information of a node in the tree
2. where it stores the values of data(splitting value), name, gini(gini index), gain(information gain), res(class of maximum probability).

Class **DTCla** (decision tree classifier class):

All below functions are included in the class

a. InfoGain
b. Gini
c. find_res
d. Predict
e. Split_by_gain
f. Split_by_gini
g. DecisionTree
h. Accuracy

Other Functions:

a. print_tree
b. prune

Functions of **DTCla**:

`infogain(x , y):`

It takes the two parameters x and y (where x = data except the selector column, y = data only containing the selector column)
Which calculates at which value(splitting value) and the **information gain** by splitting the data according to that value for each attribute in the x.
And considers that value of an attribute for which **information gain** is highest.

Example: let "abc" be a attribute, and val be the calculated value

Go left if x["abc"] <= val else go right

Returns a list containing [splitting value, gain, attribute name]
Where the gain is highest of all.

Gini(x , y):

It takes the two parameters x and y (where x = data except the selector column, y = data only containing the selector column)
Which calculates at which value(splitting value) and the **gini-index** by splitting the data according to that value for each attribute in the x.
And considers that value of an attribute for which the **gini-index** is least.

Example: let "abc" be a attribute, and val be the calculated value

Go left if x["abc"] <= val else go right

Returns a list containing [splitting value, gini, attribute name]
Where the gain is highest of all.

find_res(x, tree):

It gives a predicted output for the given insurance 'x' from the decision tree. It's a recursive function.

predict(x):

It returns the pandas data.frame of predicted outputs of a given data (input x).

It takes the help of `find_res` function to calculate the predicted value of each instance of the given data.

Split_by_gain(x, y, present_depth, Depth, Depth_limit):

`x` : data excluding selector column
`y` : data containing only selector column
`present _depth` : present depth of the node its been calculating
`Depth` : It's a binary true/false value that say's weather should we use depth limit or not
`Depth_limit` : It's a limiting value to the depth of a constructing tree

Makes a decision tree recursively.

Split_by_gini(x, y, present_depth, Depth, Depth_limit):

All the attributes mean the same as **Split_by_gain**(x, y, present_depth, Depth, Depth_limit).

Makes a decision tree recursively.

DecisionTree(x, y, Depth, Depth_limit, method):

`X, y, Depth, Depth_limit attributes` mean the same as above functions.
`Method ->` its value is 'gini' or 'gain', it says which impurity measure is to be used.

With the help of some above functions, it builds a decision tree by taking up the impurity type as gain or gini.

Accuracy(self, X, Y):

It returns the accuracy score for given test inputs.

Other Functions:

print_tree(Tree, i):

It's print's the tree with the help of a library named "**graphviz".** And saved in a pdf format or it can be printed.

prune(Tree, node,i, index):

`Node ->` node of a tree

`Tree ->` decision tree

`i` -> depth

Post prunes the given tree

PROCEDURE

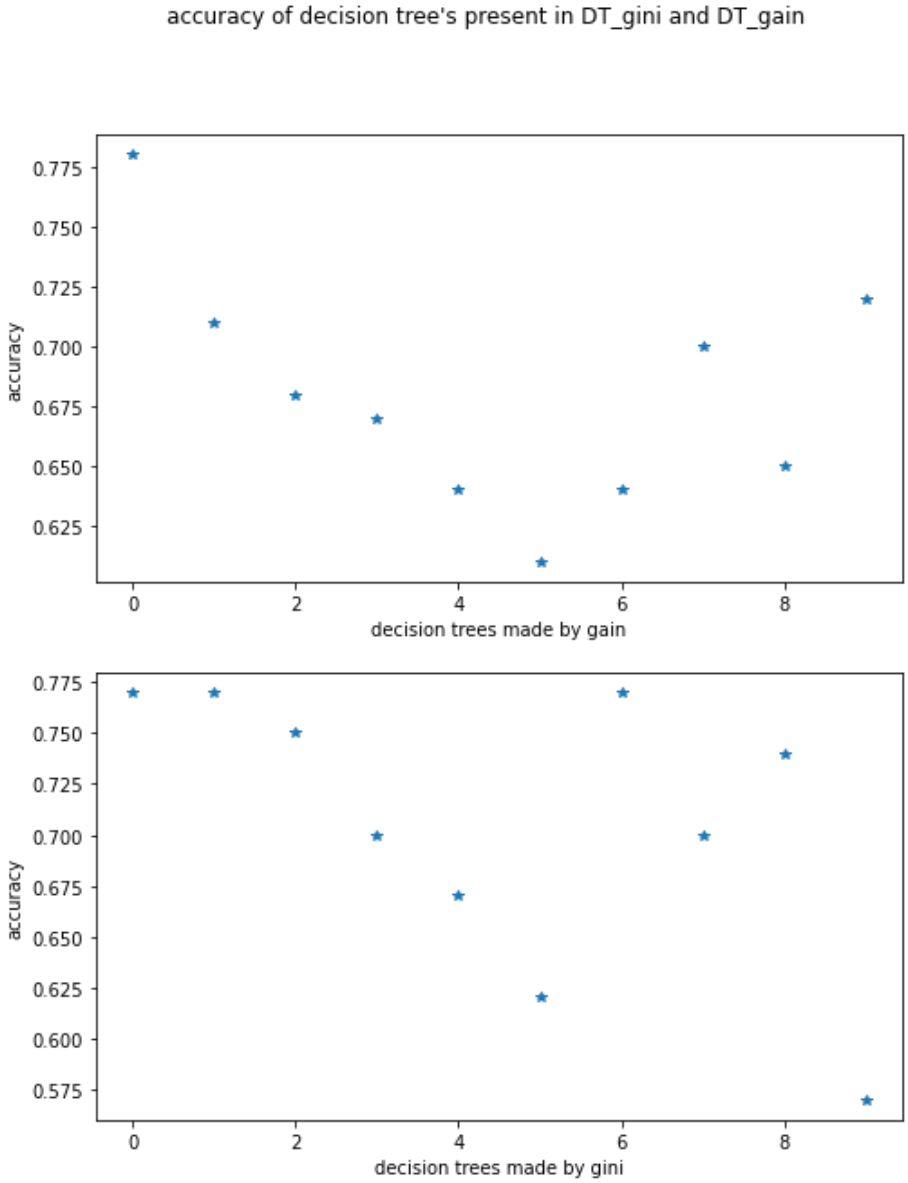
- Building a decision tree for the given data set provided in .csv format. The data is about the BUPA liver disorders. We had built a decision tree classifier. The data has the following attributes:
    a. mcv
    b. alkphos
    c. sgpt
    d. sgot
    e. gammagt
    f. Drinks (selector if >5) $(\text{class} = 1 \text{ if Drinks} > 5, \text{else class} = 0)$

- The data is divided into test and train data in the ratio of 20:80 respectively. 10 random splits of data are stored in the list.
- Making 2 lists of size 10 (contains **DTCla** class instances) for 2 impurities (gain & gini), and those 2 lists of 10 instances(total 20) are trained with those 10 randomly split data.
- Among those 20 decision trees, A decision tree is considered which accuracy is highest.
- Considered decision tree is pruned by the post-pruning process
- Find the best possible depth limit to be used for the dataset (by depth-limit vs accuracy plot)
- And analyze the data by the plot of test_accuracy vs the total number of nodes.

RESULTS

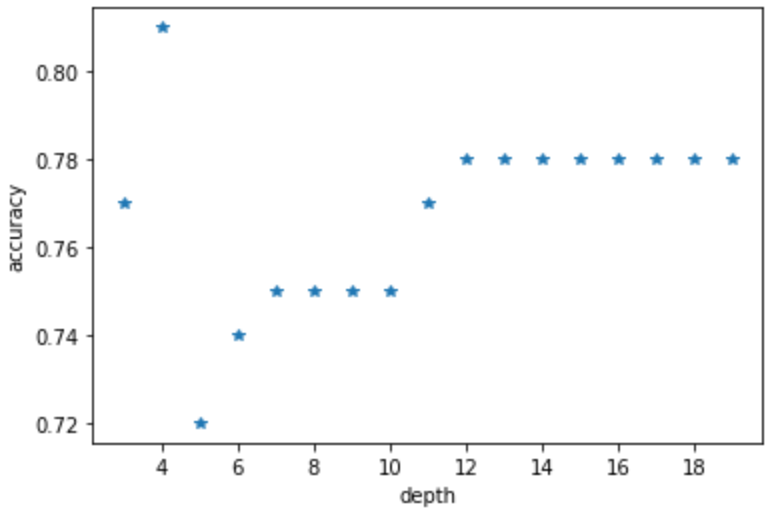
`gain ->  [0.78, 0.71, 0.68, 0.67, 0.64, 0.61, 0.64, 0.7, 0.65, 0.72]`

`gini ->  [0.77, 0.77, 0.75, 0.7, 0.67, 0.62, 0.77, 0.7, 0.74, 0.57]`




```
average accuracy by gain impurity ->  0.68
average accuracy by gini impurity ->  0.71
```

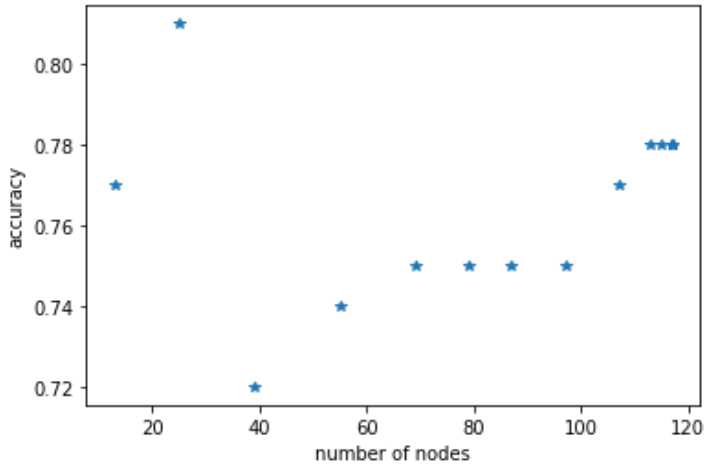
- 0th indexed tree classified on gain impurity(bolded in the list above) is considered for the following graphs



Accuracy -> [0.77, **0.81**, 0.72, 0.74, 0.75, 0.75, 0.75, 0.75, 0.77, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78]

depth -> [3, **4**, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

- By the above graph depth vs accuracy(test_accuracy)
- Depth of 4 is optimal for the considered decision tree.



Number of nodes -> [13, 25, 39, 55, 69, 79, 87, 97, 107, 113, 115, 117, 117, 117, 117, 117, 117]

accuracy -> [0.77, 0.81, 0.72, 0.74, 0.75, 0.75, 0.75, 0.75, 0.77, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78, 0.78]

- Above graph is number of nodes vs accuracy (test_accuracy)
- It is clear that as we overfit the data the accuracy is not optimal

★ After pruning the tree the test accuracy went from 78% to 85%

Final Decision Tree obtained from the question 2
its accuracy on test data = 0.78
classified by information gain

In figure:
Go left if attribute <= value
else go right