Department of Computer Science & Engineering

Final Year B. Tech. (CSE) – I : 2021-22

4CS462 : PE2 - Data Mining Lab

Assignment No. 5

By DM21G03

(2018BTECS00082 : Hritik Belani , 2019BTECS00209 : Sailee Akim)

Date: 03/10/2021

- **Title**

  Data Analysis Tool

- **Objective**

Design the rule-based classifier: Extract the rules from decision tree build

- **Specification**

- Python 3.8.11

  - Dataset

- **Introduction and Theory**

  IF-THEN Rules
  Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from −

  IF condition THEN conclusion

  Let us consider a rule R1,

  R1: IF age = youth AND student = yes THEN buy_computer = yes

  Points to remember −

  - The IF part of the rule is called **rule antecedent** or **precondition**.
  - The THEN part of the rule is called **rule consequent**.
  - The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
  - The consequent part consists of class prediction.

  Note − We can also write rule R1 as follows −

R1: (age = youth) ^ (student = yes))(buys computer = yes)

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction
Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to remember −

To extract a rule from a decision tree −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm
Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

Note − The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

Algorithm: Sequential CoveringInput: D, a data set class-labeled tuples,Att_vals, the set of all attributes and their possible values.Output: A Set of IF-THEN rules.Method:Rule_set={ }; // initial set of rules learned is emptyfor each class c do repeat Rule = Learn_One_Rule(D, Att_valls, c); remove tuples covered by Rule form D; until termination condition; Rule_set=Rule_set+Rule; // add a new rule to rule-setend forreturn Rule_Set;

Rule Pruning
The rule is pruned is due to the following reason −

- The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
- The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

FOIL is one of the simple and effective method for rule pruning. For a given rule R,

FOIL_Prune = pos - neg / pos + neg

where pos and neg is the number of positive tuples covered by R, respectively.

Note − This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

❖ Procedure

1. Design the rule based classifier : Extract the rules from decision tree build in assignment no. 4.

2. Tabulate the results and evaluate the performance of rules generated using following metrics :

   a. Coverage

   b. Accuracy

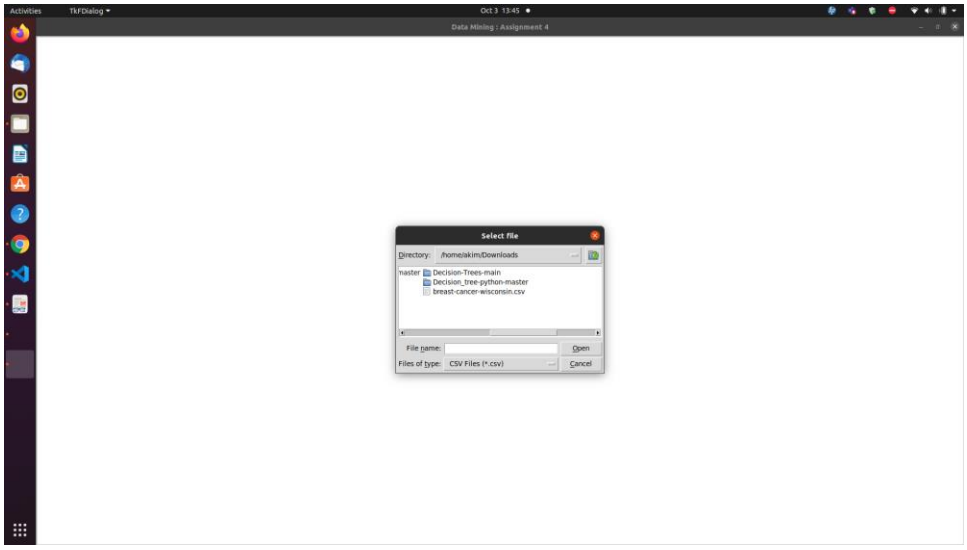
   c. Toughness (size)

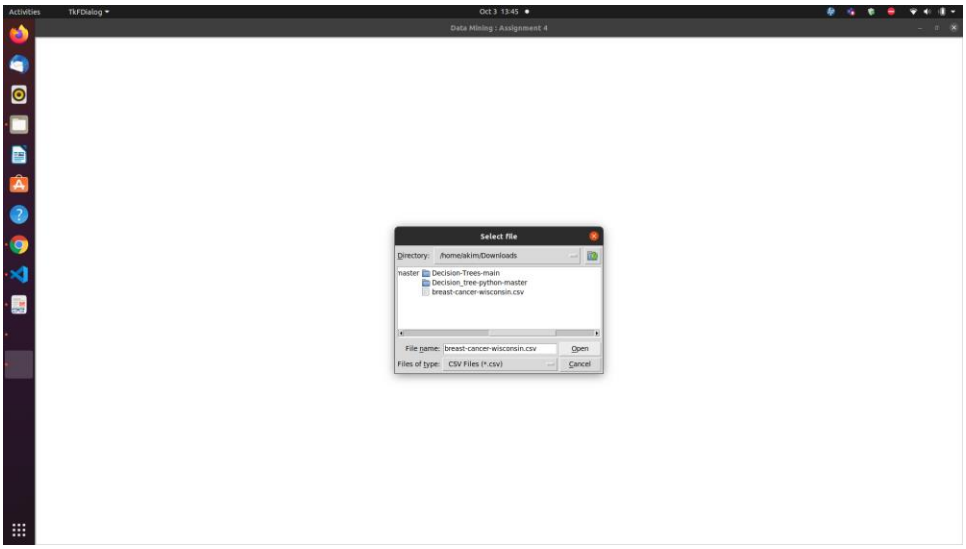
3. Use the following categorical data sets from UCI machine learning repository :

   a. Balance Scale data set

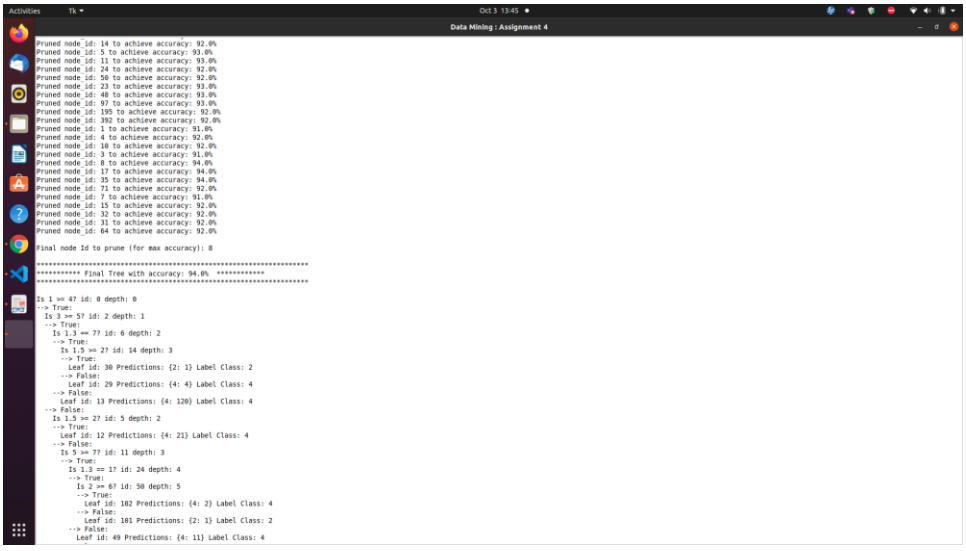
   b. Car evaluation data set

   c. Breast-cancer data set




Activities
TkFDialog
Oct 3 13:45
Data Mining : Assignment 4
Select file
Directory:
/home/akm/Downloads
master
Decision-Trees-main
Decision_tree-python-master
breast-cancer-wisconsin.csv
File name:
breast-cancer-wisconsin.csv
Open
Files of type:
CSV Files (*.csv)
Cancel


Activities
Tk
Oct 3 13:45
Data Mining : Assignment 4
Leaf nodes ****************
id = 30 depth =4
id = 29 depth =4
id = 13 depth =3
id = 12 depth =3
id = 102 depth =6
id = 101 depth =6
id = 49 depth =5
id = 98 depth =6
id = 196 depth =7
id = 786 depth =9
id = 785 depth =9
id = 391 depth =8
id = 47 depth =5
id = 22 depth =4
id = 21 depth =4
id = 9 depth =3
id = 18 depth =4
id = 36 depth =5
id = 72 depth =6
id = 144 depth =7
id = 143 depth =7
id = 16 depth =4
id = 66 depth =6
id = 65 depth =6
id = 130 depth =7
id = 129 depth =7
id = 63 depth =6
Non-leaf nodes ***************
id = 0 depth =0
id = 2 depth =1
id = 6 depth =2
id = 14 depth =3
id = 5 depth =2
id = 11 depth =3
id = 24 depth =4
id = 50 depth =5
id = 23 depth =4
id = 48 depth =5
id = 97 depth =6
id = 195 depth =7
id = 392 depth =8
id = 1 depth =1
id = 4 depth =2
id = 10 depth =3
id = 3 depth =2
id = 8 depth =3
id = 17 depth =4
id = 35 depth =5
id = 71 depth =6
id = 7 depth =3
id = 15 depth =4
id = 32 depth =5
id = 31 depth =5
id = 64 depth =6
Tree before pruning with accuracy: 92.0


Activities
Tk
Oct 3 13:45
Data Mining : Assignment 4
Pruned node_id: 14 to achieve accuracy: 92.0%
Pruned node_id: 5 to achieve accuracy: 93.0%
Pruned node_id: 11 to achieve accuracy: 93.0%
Pruned node_id: 24 to achieve accuracy: 92.0%
Pruned node_id: 50 to achieve accuracy: 92.0%
Pruned node_id: 23 to achieve accuracy: 93.0%
Pruned node_id: 48 to achieve accuracy: 93.0%
Pruned node_id: 97 to achieve accuracy: 93.0%
Pruned node_id: 195 to achieve accuracy: 92.0%
Pruned node_id: 392 to achieve accuracy: 92.0%
Pruned node_id: 1 to achieve accuracy: 91.0%
Pruned node_id: 4 to achieve accuracy: 92.0%
Pruned node_id: 10 to achieve accuracy: 92.0%
Pruned node_id: 3 to achieve accuracy: 91.0%
Pruned node_id: 8 to achieve accuracy: 94.0%
Pruned node_id: 17 to achieve accuracy: 94.0%
Pruned node_id: 35 to achieve accuracy: 94.0%
Pruned node_id: 71 to achieve accuracy: 92.0%
Pruned node_id: 7 to achieve accuracy: 91.0%
Pruned node_id: 15 to achieve accuracy: 92.0%
Pruned node_id: 32 to achieve accuracy: 92.0%
Pruned node_id: 31 to achieve accuracy: 92.0%
Pruned node_id: 64 to achieve accuracy: 92.0%
Final node Id to prune (for max accuracy): 8
********** Final Tree with accuracy: 94.0% **********
Is 1 >= 47 id: 0 depth: 0
--> True:
Is 3 >= 57 id: 2 depth: 1
--> True:
Is 1.3 == 77 id: 6 depth: 2
--> True:
Is 1.5 >= 27 id: 14 depth: 3
--> True:
Leaf id: 30 Predictions: {2: 1} Label Class: 2
--> False:
Leaf id: 29 Predictions: {4: 4} Label Class: 4
--> False:
Leaf id: 13 Predictions: {4: 120} Label Class: 4
--> False:
Is 1.5 >= 27 id: 5 depth: 2
--> True:
Leaf id: 12 Predictions: {4: 21} Label Class: 4
--> False:
Is 5 >= 77 id: 11 depth: 3
--> True:
Is 1.3 == 17 id: 24 depth: 4
--> True:
Is 2 >= 67 id: 50 depth: 5
--> True:
Leaf id: 102 Predictions: {4: 2} Label Class: 4
--> False:
Leaf id: 101 Predictions: {2: 1} Label Class: 2
--> False:
Leaf id: 49 Predictions: {4: 11} Label Class: 4

❖ Conclusion

Thus, implemented and discovered several ways to build decision trees along with rule- based Classification and their metrics such as recall, precision, support etc as well as coverage, accuracy and so on.

❖ References

https://www.tutorialspoint.com/data_mining/dm_rbc.htm

https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_559