Report:Decision trees

Name:Ruchita Deshmukh

NetId:rld170003

In collaboration with:Rinkle Seth(rcs170004)

a)Code file:

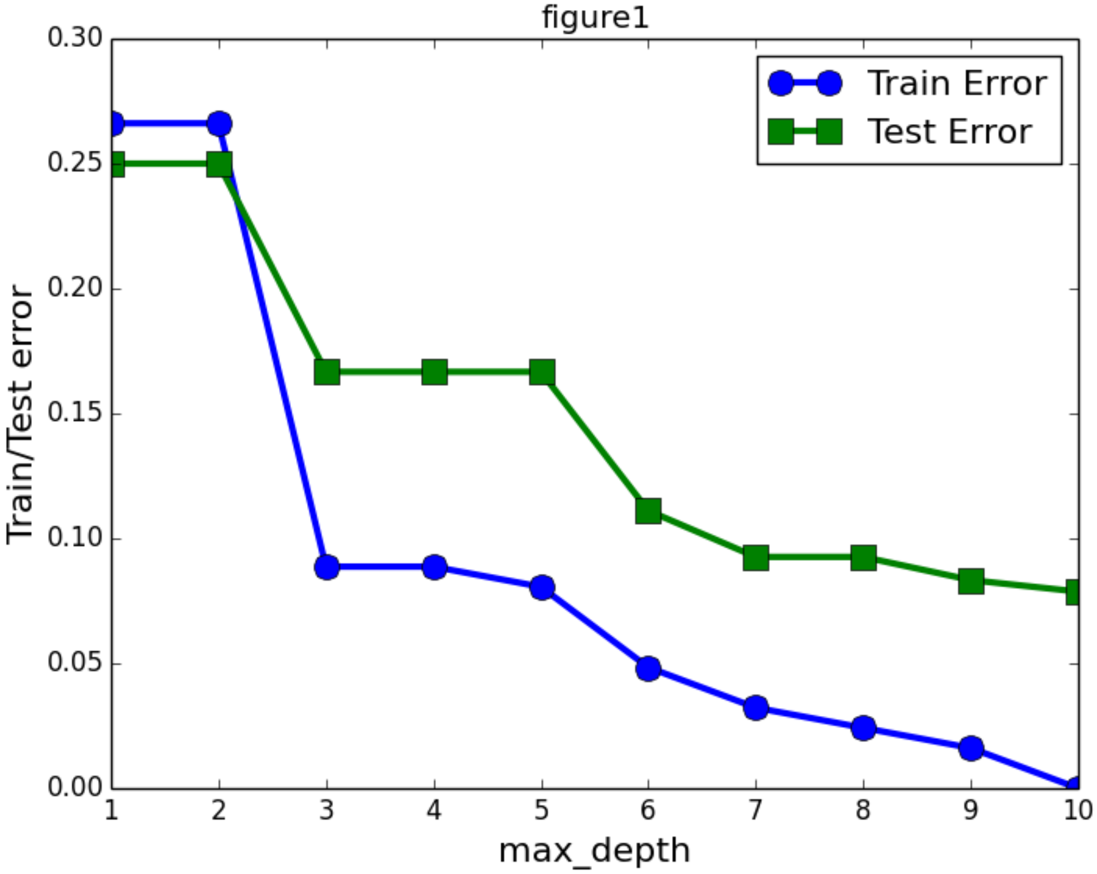
Report:

b)Learning curves

Monk1:

Plot-

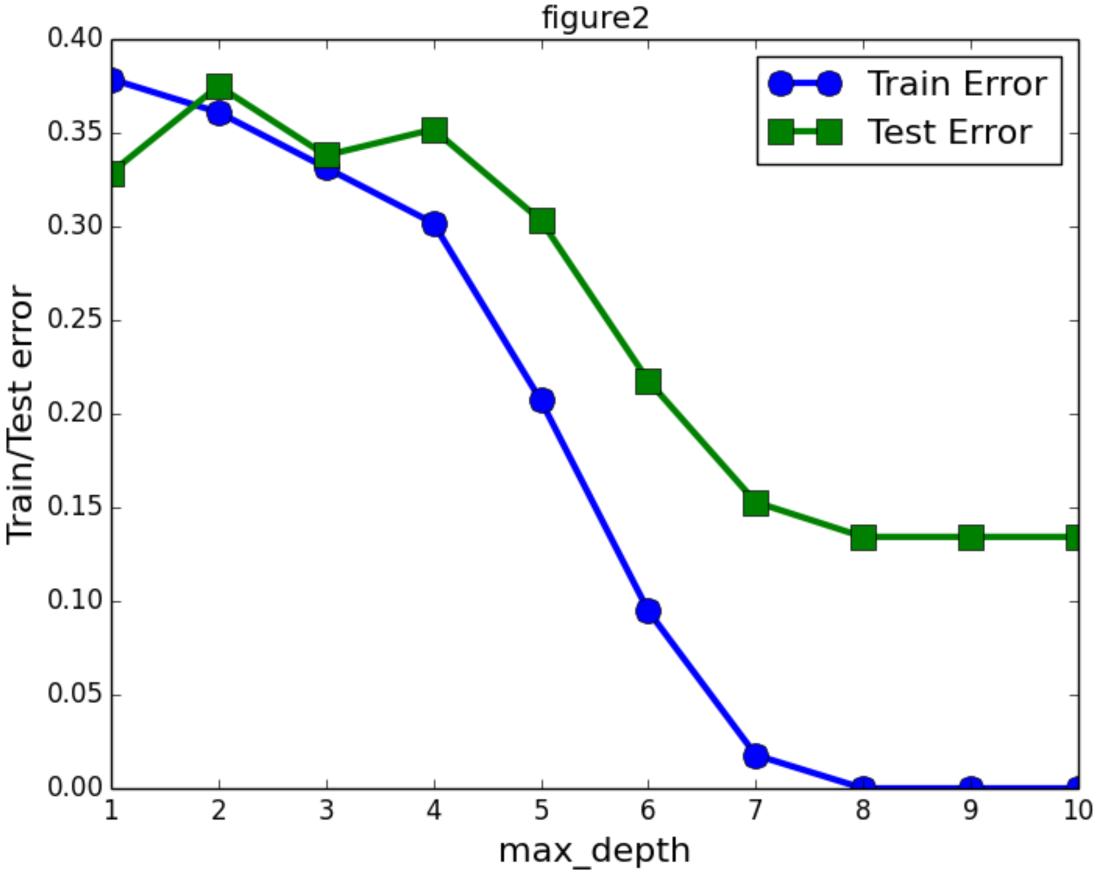
Test/train error. plot for depths(1 to 10 on monk1 dataset)



Monk2:

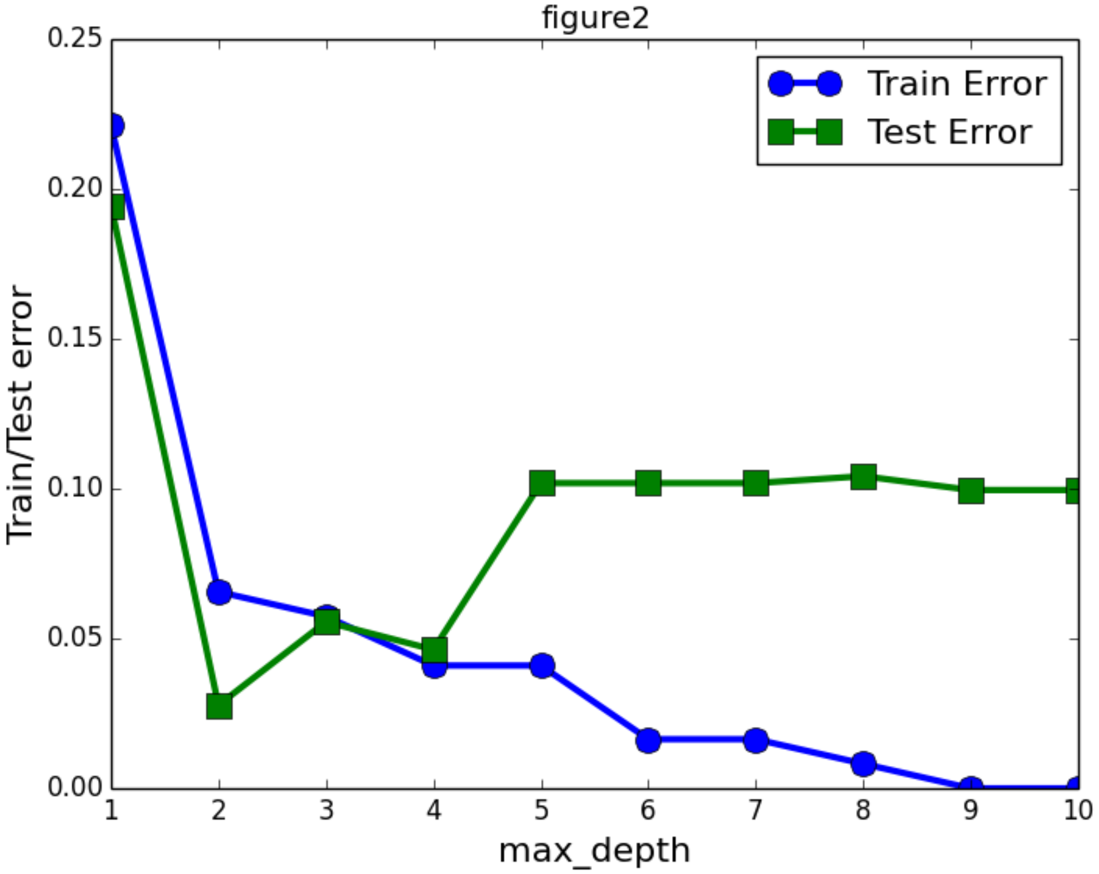
Plot:

Test/train error. plot for depths(1 to 10 on monk2 dataset)

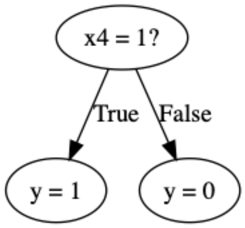


Monk3:
Plot:

Test/train error. plot for depths(1 to 10 on monk3 dataset)

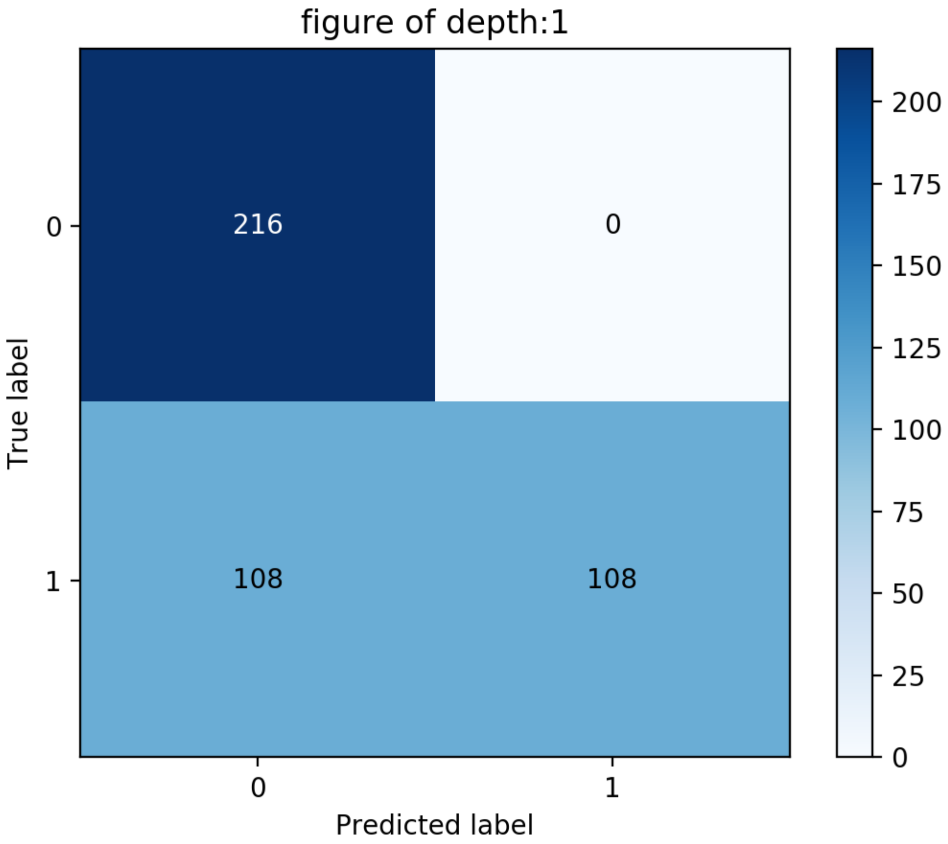


c)Weak Learners:
Monks1 dataset:
Depth:1
Tree of depth1 for monks:

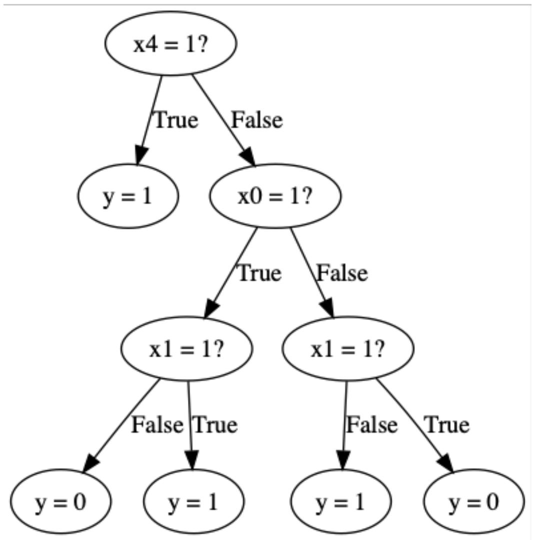


Confusion Matrix for monk1 for decision tree of depth1:

```
[[216,   0],
 [108, 108]]
```

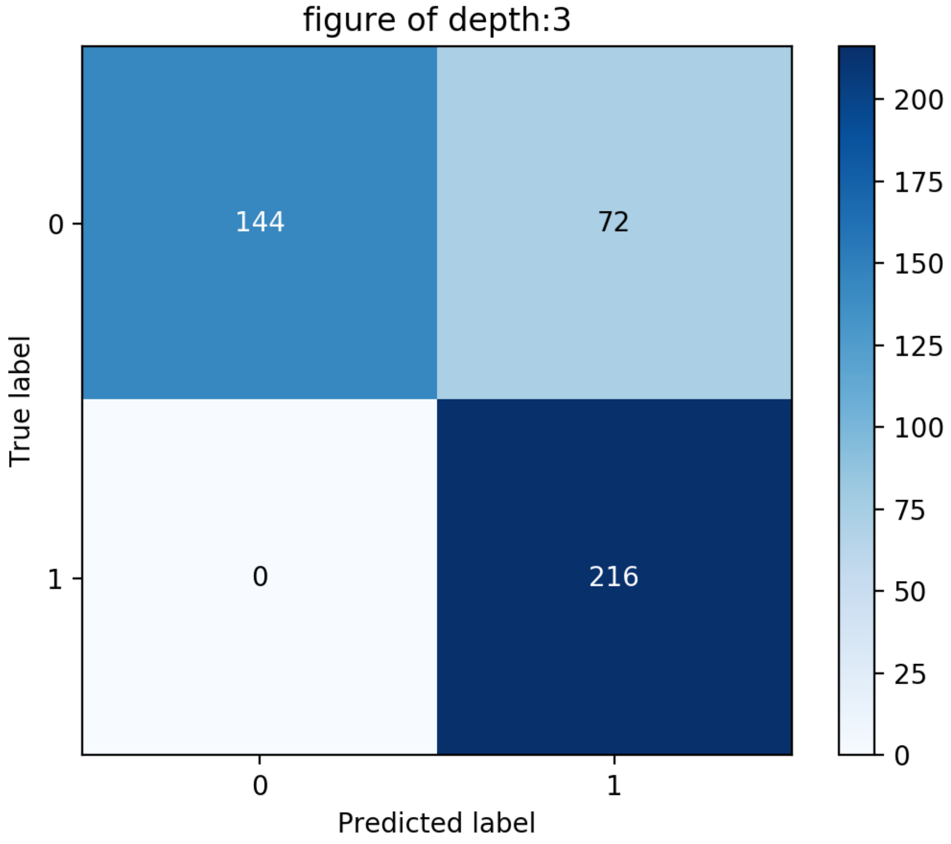


Tree of depth 3 for monks:
Tree:



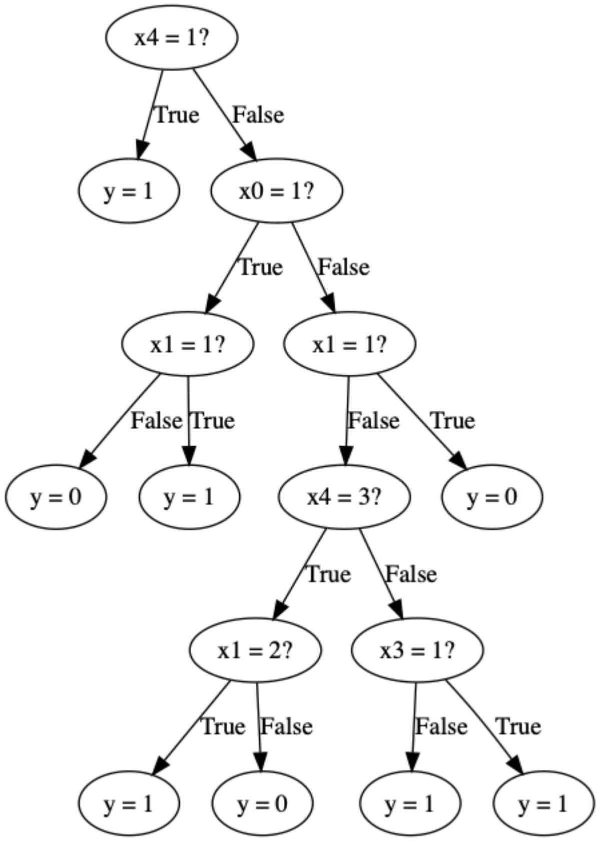
Confusion Matrix for monk1 for decision tree of depth 3:

```
array([            [144,  72],
[  0, 216]])
```



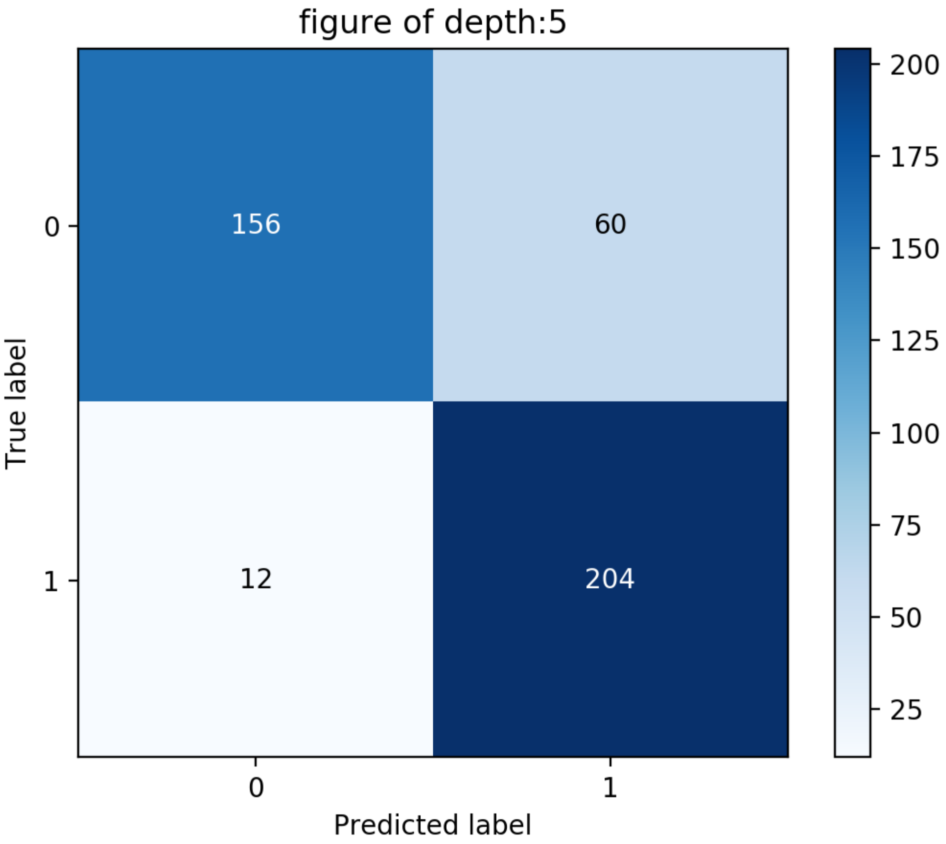
Depth:5

Depth:5 tree for monks1 dataset

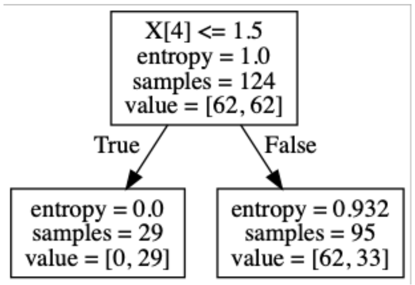


Confusion matrix for depth 5 tree for monks1 dataset:

```
array(   [[156,  60],
          [ 12, 204]])
```

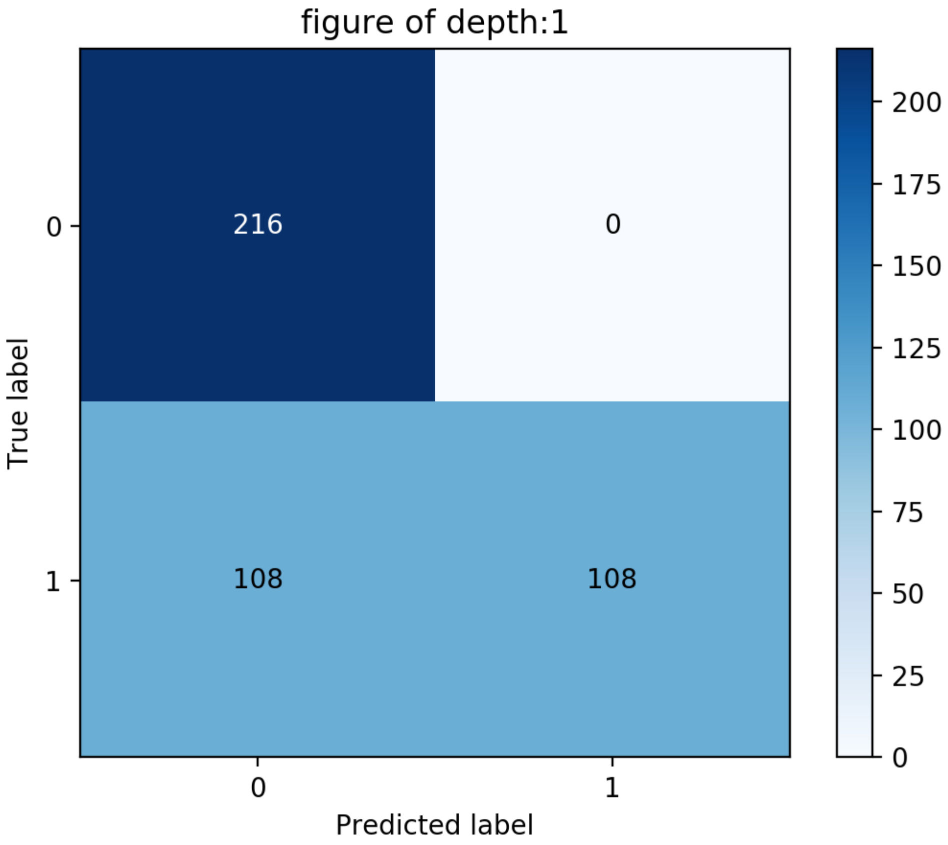


d)SciKitLearn:
Monks1:
Depth1:
Tree for monks1 depth 1 by sklearn:



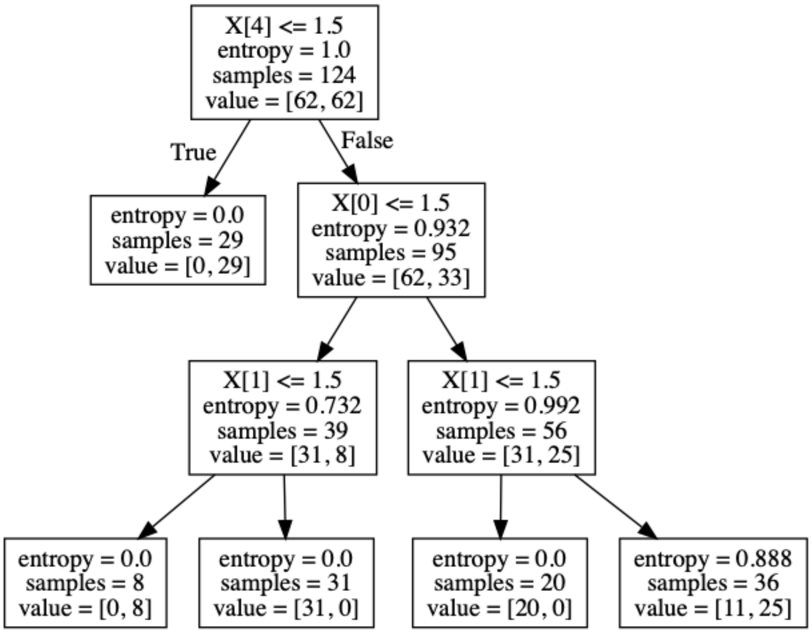
Confusion Matrix (sklearn):

```
[[216,   0],
  [108, 108]]
```



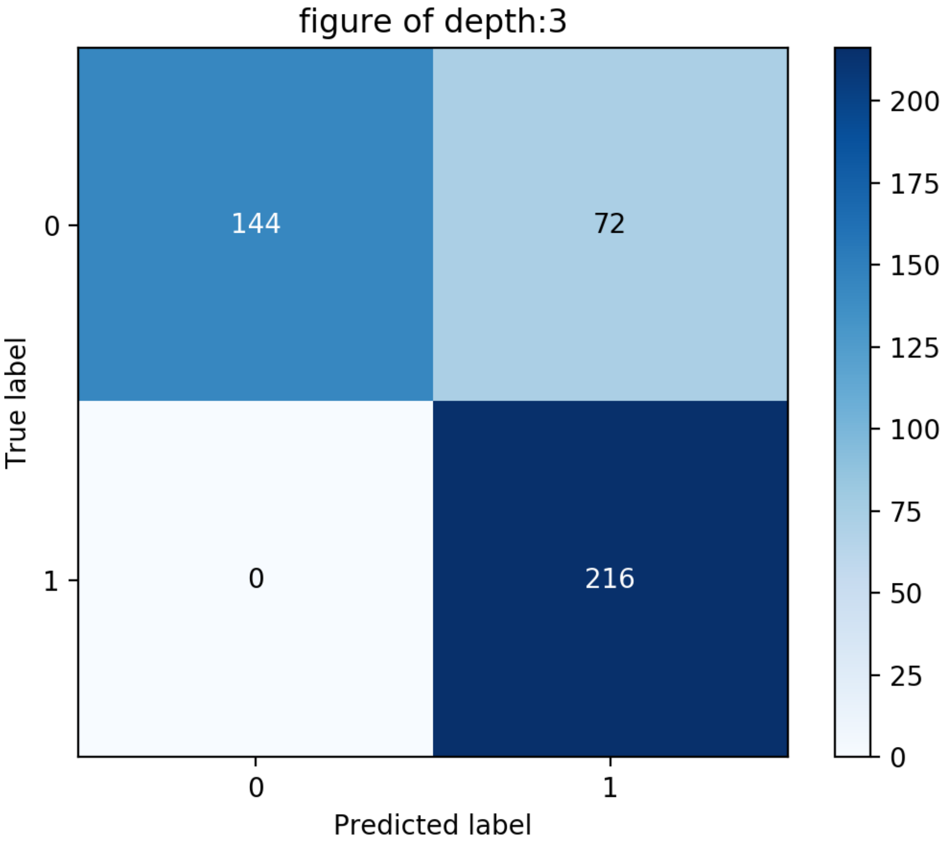
Depth:3

Tree of depth3(sklearn)::



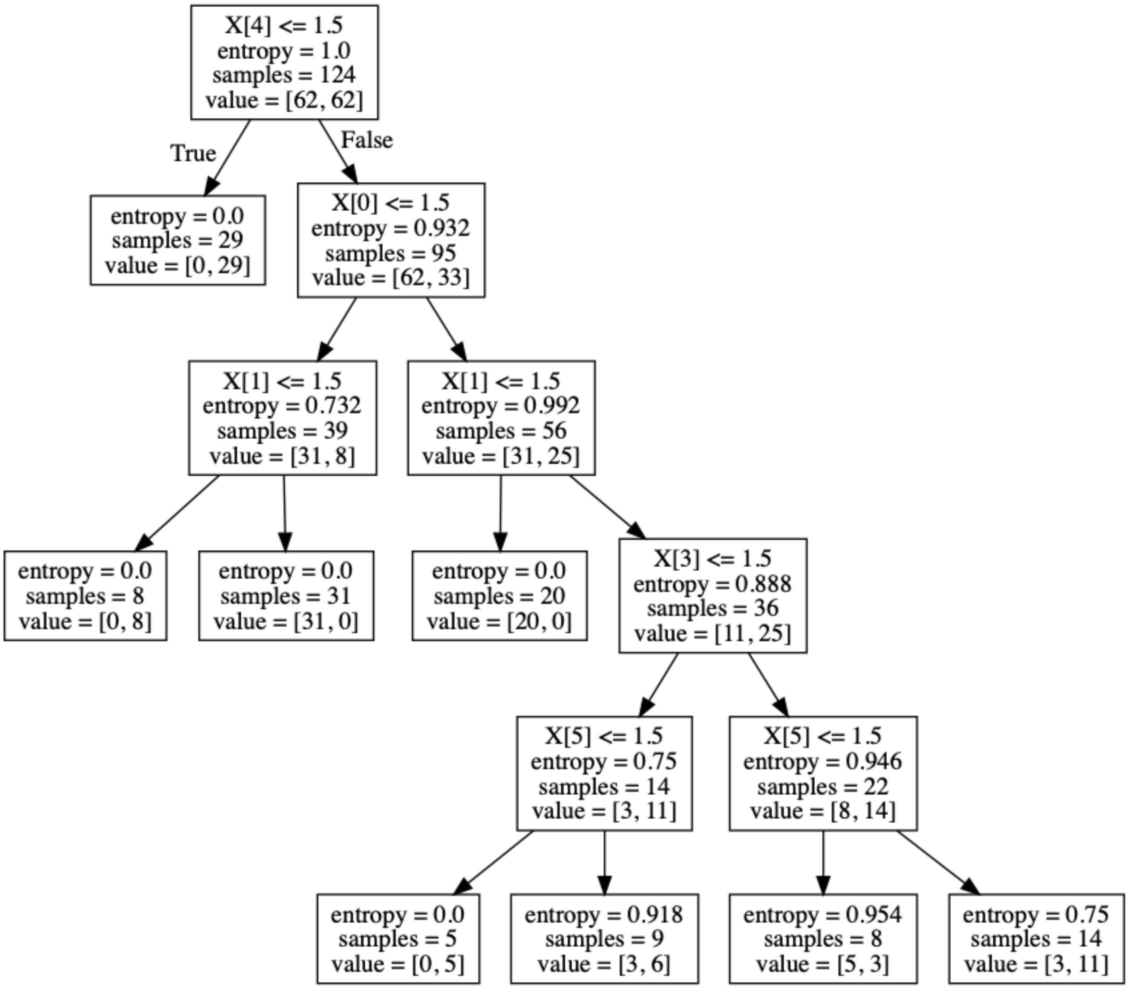
Confusion Matrix- depth 3(sklearn)::

```
        [144,  72],
          [  0, 216]]
```

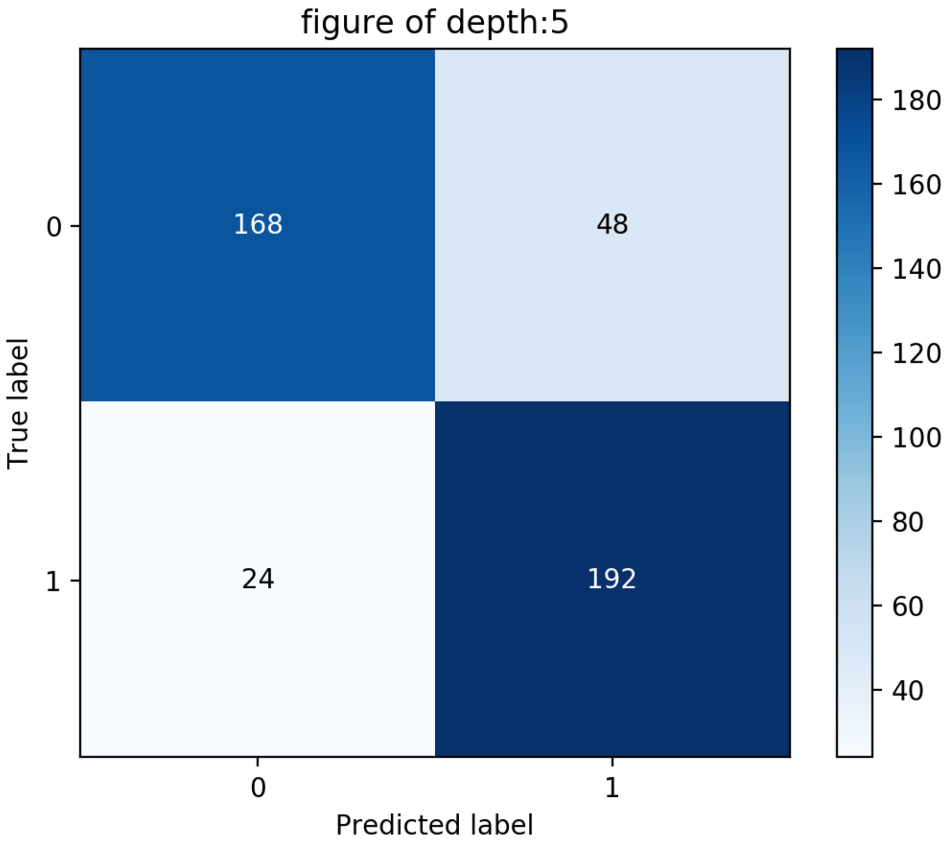


Depth 5:

Tree (sklearn)::



```
Confusion Matrix depth 5 (sklearn)::
[168,  48],
     [ 24, 192]]
```



e)Datasets:

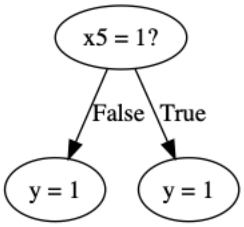
1)car evaluation dataset:

Src: UCI ML datasets

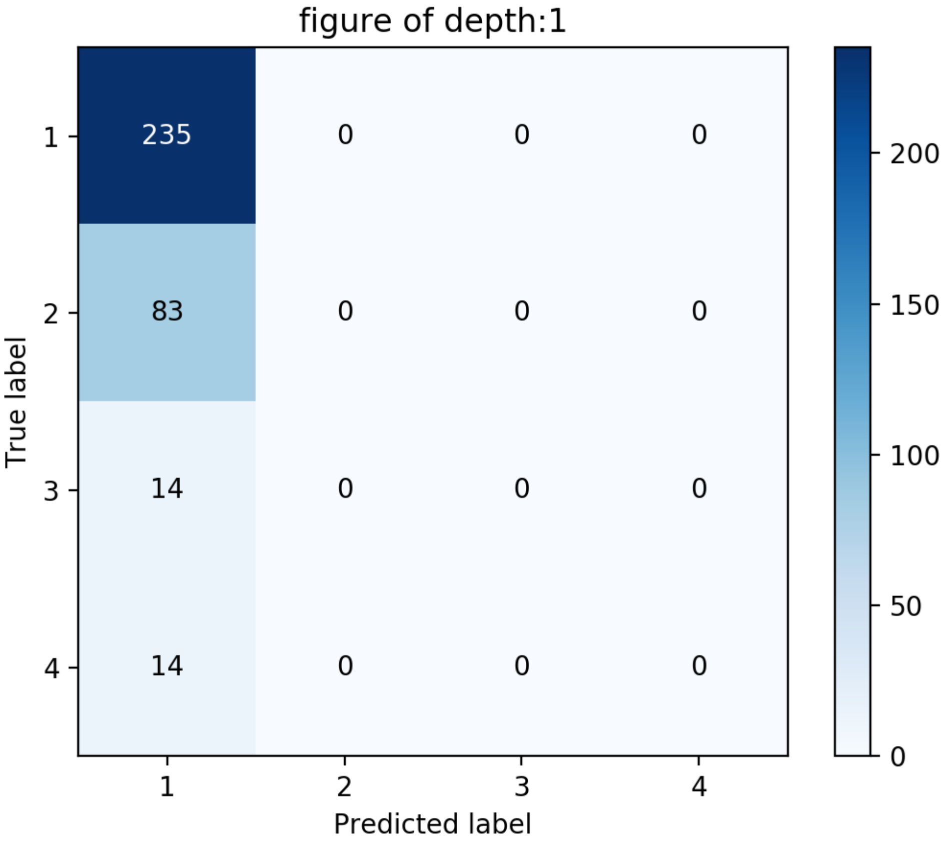
```
6. Number of Attributes: 6

7. Attribute Values:

   buying       v-high, high, med, low
   maint        v-high, high, med, low
   doors        2, 3, 4, 5-more
   persons      2, 4, more
   lug_boot     small, med, big
   safety       low, med, high

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

   class      N          N[%]
   -----------------------------
   unacc     1210     (70.023 %)
   acc        384     (22.222 %)
   good        69     ( 3.993 %)
   v-good       65      ( 3.762 %)
```

Preprocessing-converting non numerical values to numerical values:
i)Confusion matrix and trees at depth 1,3,5

depth 1:
tree of depth1 (my code)



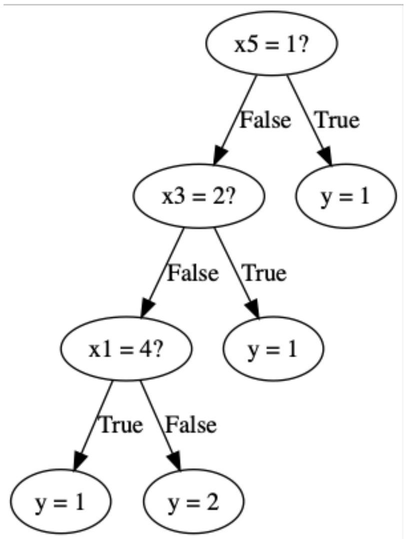
Confusion matrix depth 1 car dataset (my implementation:



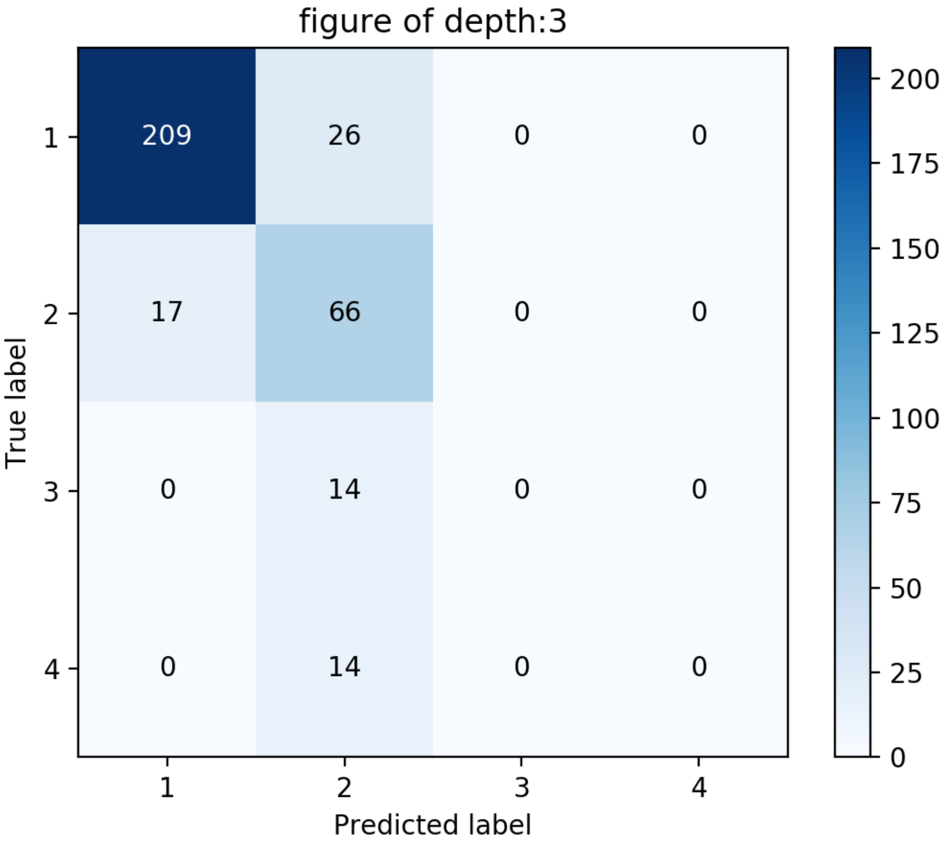
Best col,value: col 5 ==1

Since distribution of label 1 is 70%(about), best value classifies it as 1

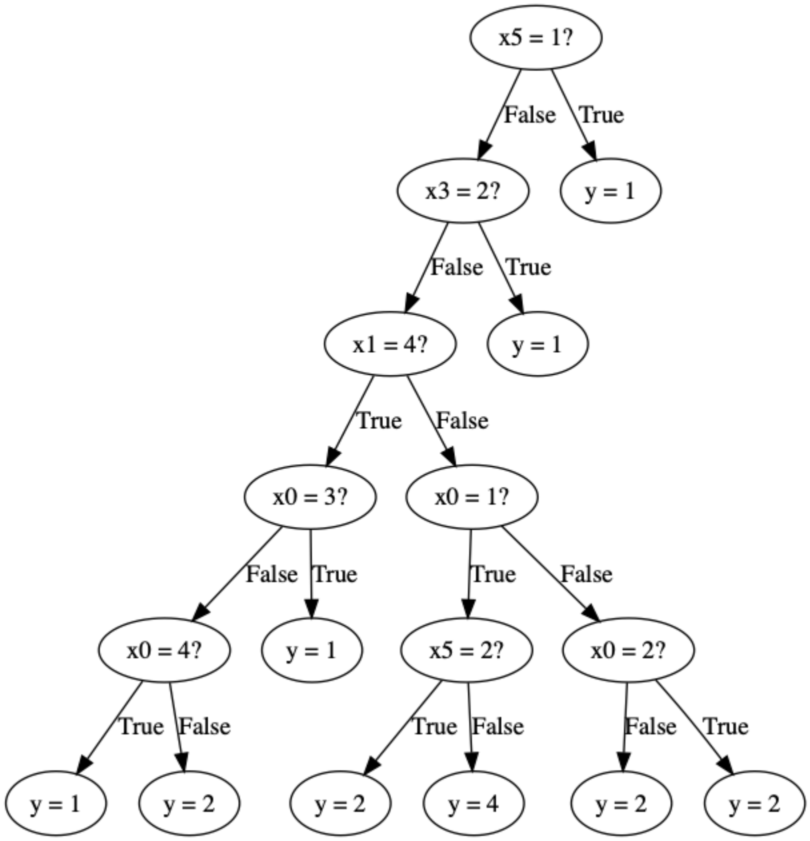
Depth 3:
Tree of depth 3(my implementation)



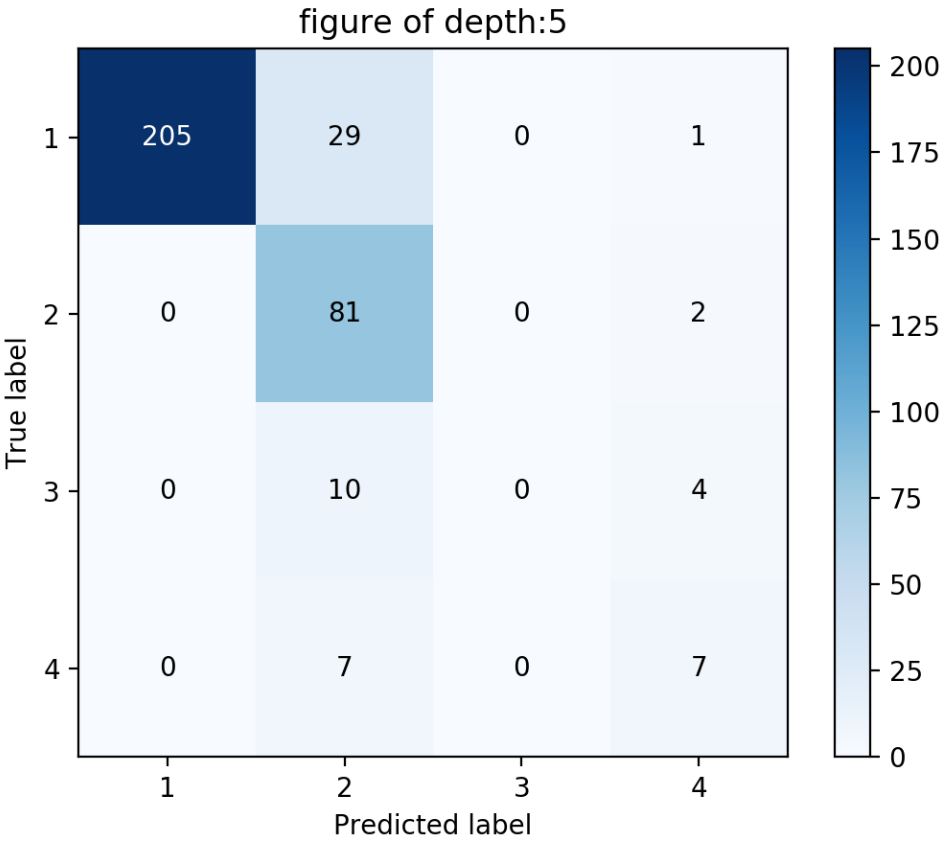
Confusion matrix depth 3-car dataset(my implementation:



Depth 5:
Tree of depth 5-car dataset(my implementation

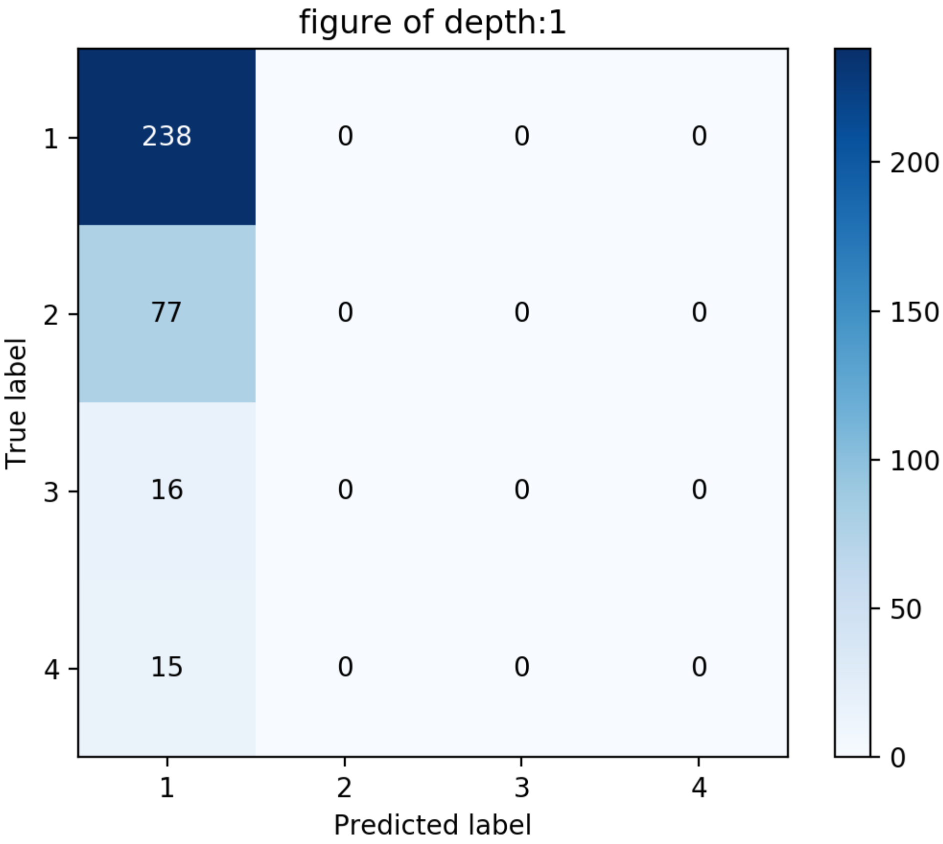


Confusion matrix for tree of depth 5 on car datasrt. For my implementation:

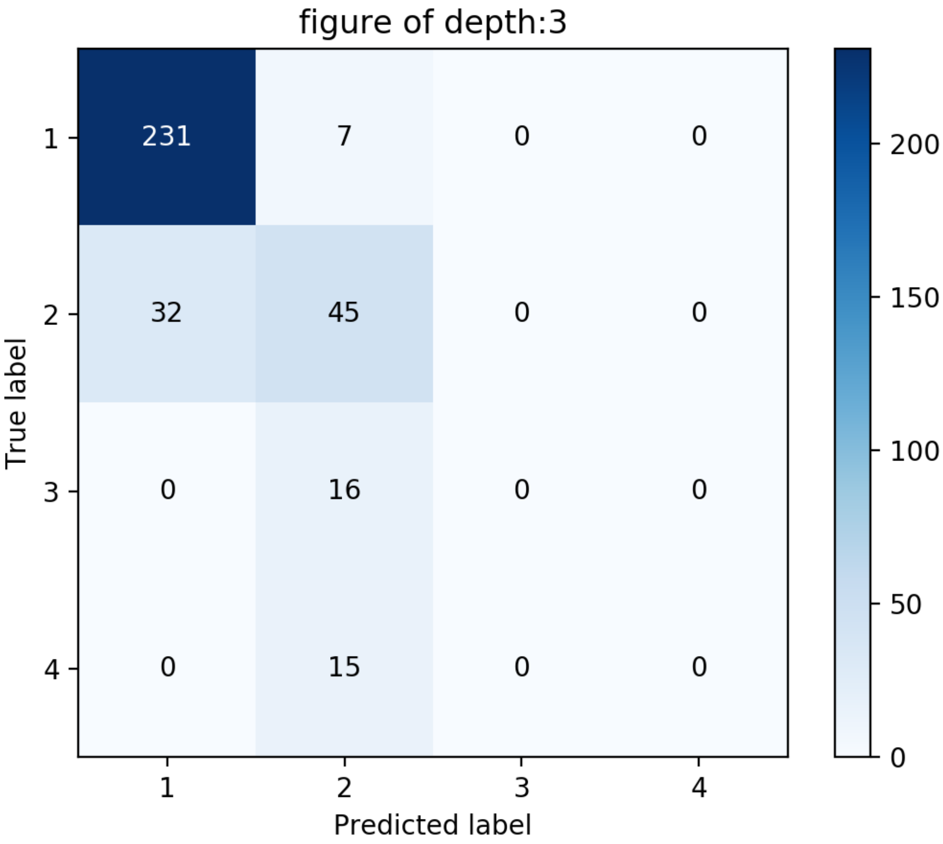


	1	2	3	4
1	205	29	0	1
2	0	81	0	2
3	0	10	0	4
4	0	7	0	7

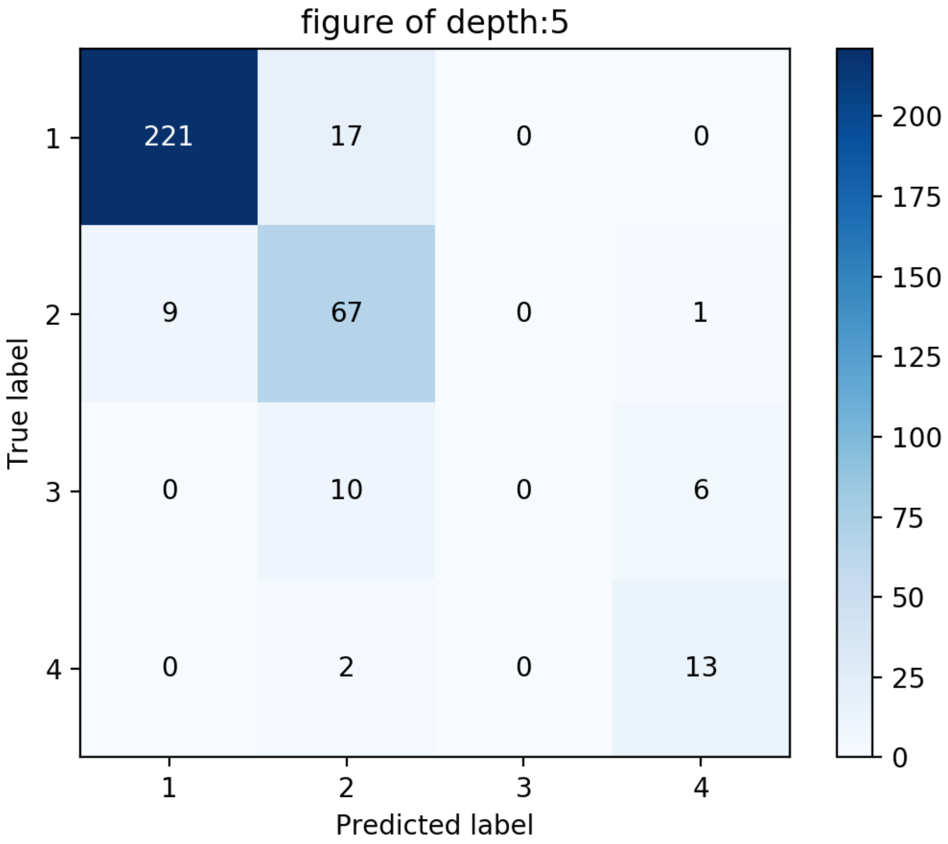
ii)Sklearn confusion matrices:
depth 1 (sklearn implementation):



Depth: 3(sklearn implementation):



Depth 5(sklearn implementation):



2)Hayes-roth dataset

Src-UCI ML datasets

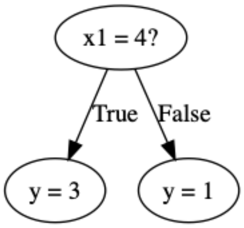
```
Number of Instances: 132 training instances, 28 test instances

6. Number of Attributes: 5 plus the class membership attribute.  3 concepts.

7. Attribute Information:
      -- 1. name: distinct for each instance and represented numerically
      -- 2. hobby: nominal values ranging between 1 and 3
      -- 3. age: nominal values ranging between 1 and 4
      -- 4. educational level: nominal values ranging between 1 and 4
      -- 5. marital status: nominal values ranging between 1 and 4
      -- 6. class: nominal value between 1 and 3

9. Missing Attribute Values: none

10. Class Distribution: see below

11. Detailed description of the experiment:
  1. 3 categories (1, 2, and neither -- which I call 3)
     -- some of the instances could be classified in either class 1 or 2, and
        they have been evenly distributed between the two classes
  2. 5 Attributes
     -- A. name (a randomly-generated number between 1 and 132)
```

```
      -- B. hobby (a randomly-generated number between 1 and 3)
      -- C. age (a number between 1 and 4)
      -- D. education level (a number between 1 and 4)
      -- E. marital status (a number between 1 and 4)
   3. Classification:
      -- only attributes C-E are diagnostic; values for A and B are ignored
      -- Class Neither: if a 4 occurs for any attribute C-E
      -- Class 1: Otherwise, if (# of 1's)>(# of 2's) for attributes C-E
      -- Class 2: Otherwise, if (# of 2's)>(# of 1's) for attributes C-E
      -- Either 1 or 2: Otherwise, if (# of 2's)=(# of 1's) for attributes C-E
   4. Prototypes:
      -- Class 1: 111
      -- Class 2: 222
      -- Class Either: 333
      -- Class Neither: 444
   5. Number of training instances: 132
      -- Each instance presented 0, 1, or 10 times
      -- None of the prototypes seen during training
      -- 3 instances from each of categories 1, 2, and either are repeated
         10 times each
      -- 3 additional instances from the Either category are shown during
         learning
   5. Number of test instances: 28
      -- All 9 class 1
      -- All 9 class 2
      -- All 6 class Either
      -- All 4 prototypes
      --------------------
      --    28 total
```

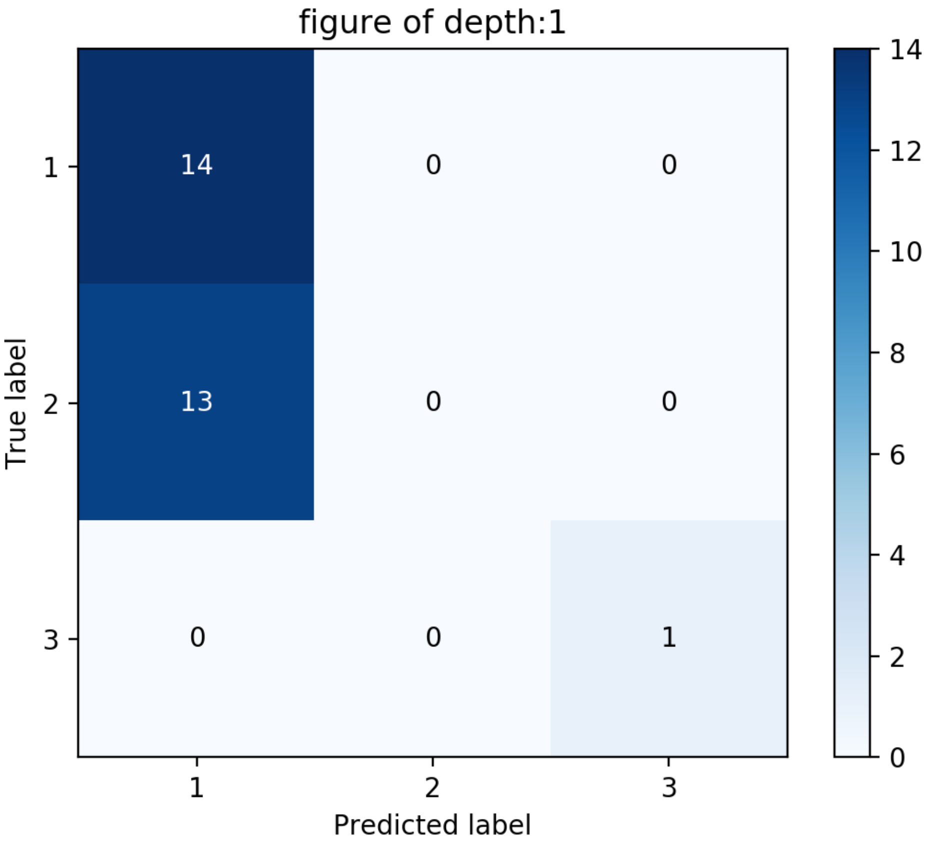
i)Trees and cm for depth 1, ,3, 5

Depth1(my implementation):

Tree of depth 1 :

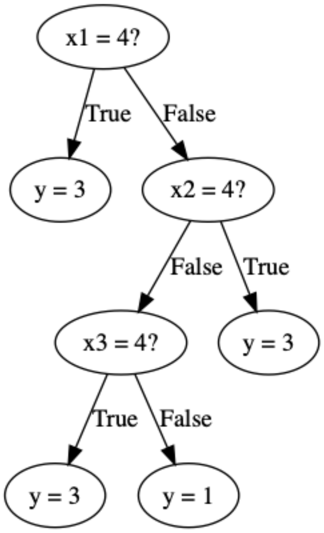


Confusion matrix(my implementation)::

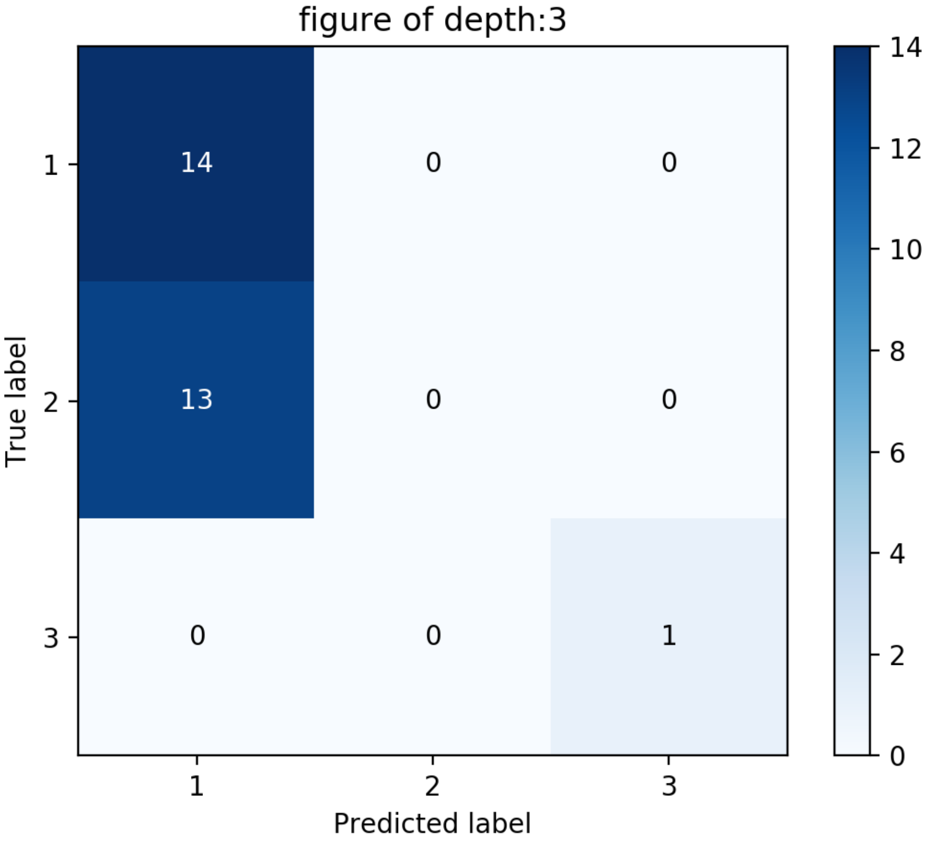


Depth 3:

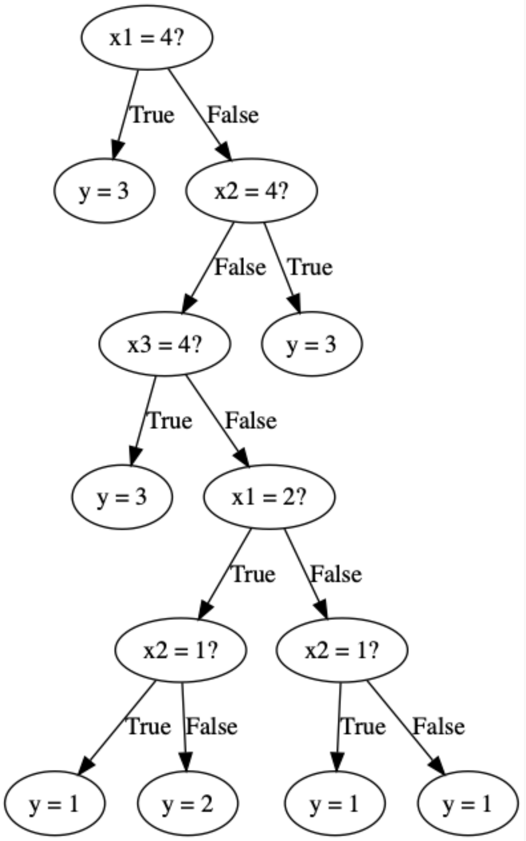
Tree: (my implementation):



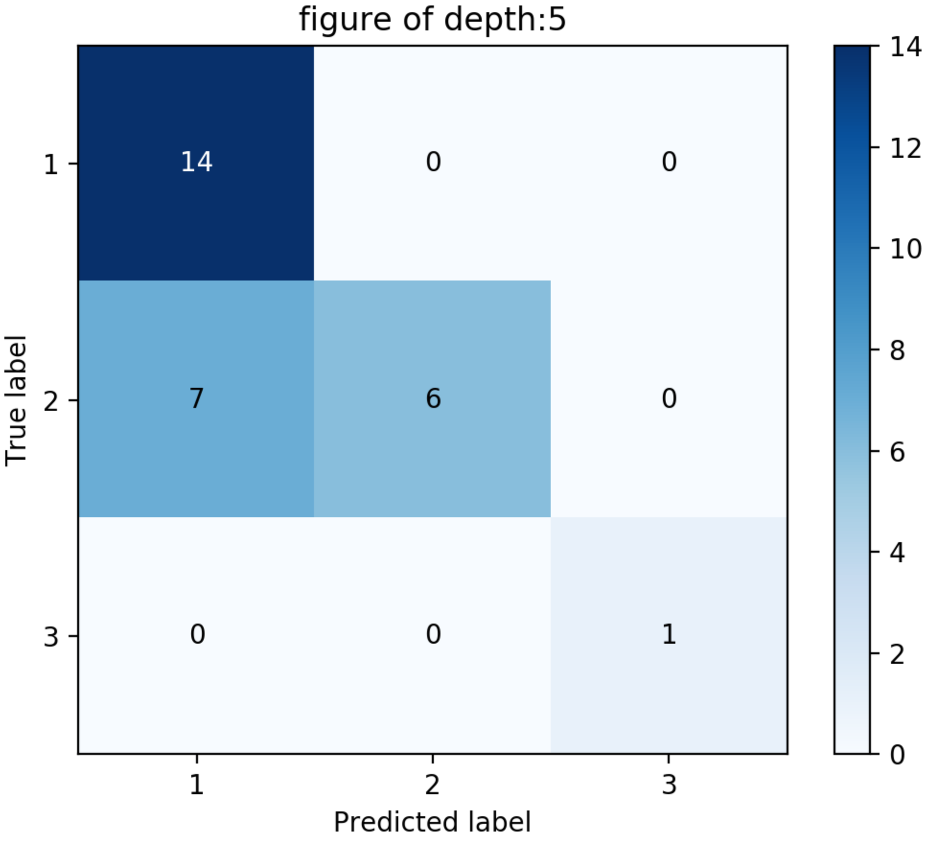
Confusion matrix: (my implementation):



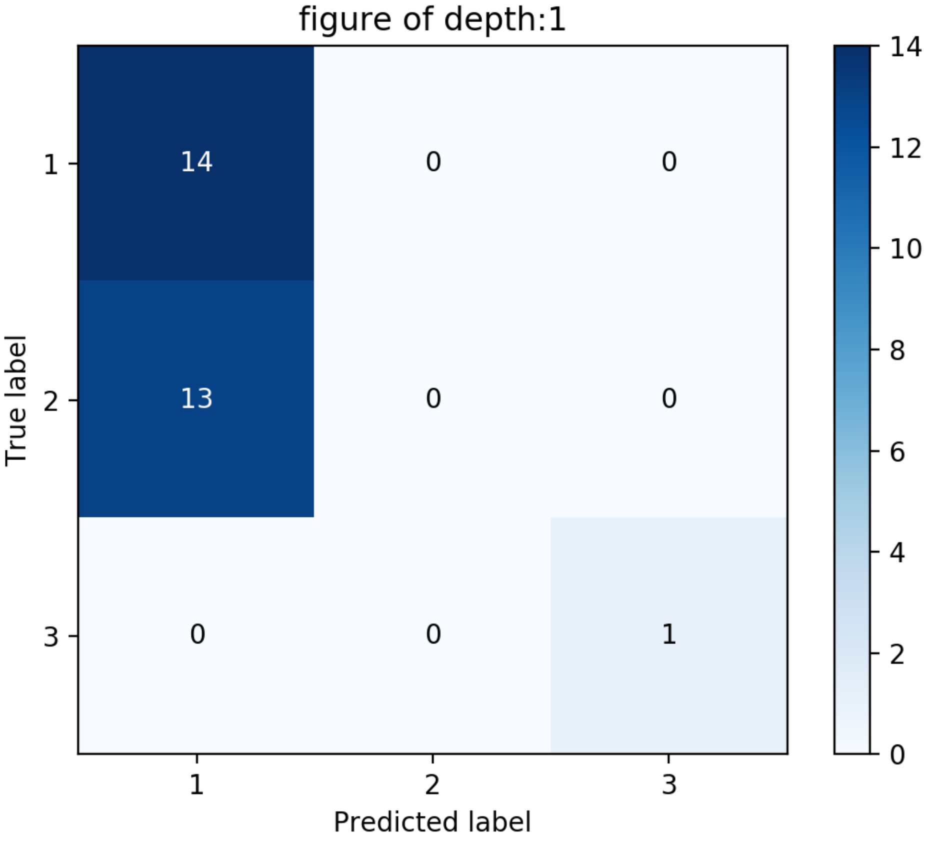
Depth 5:
Tree: (my implementation):



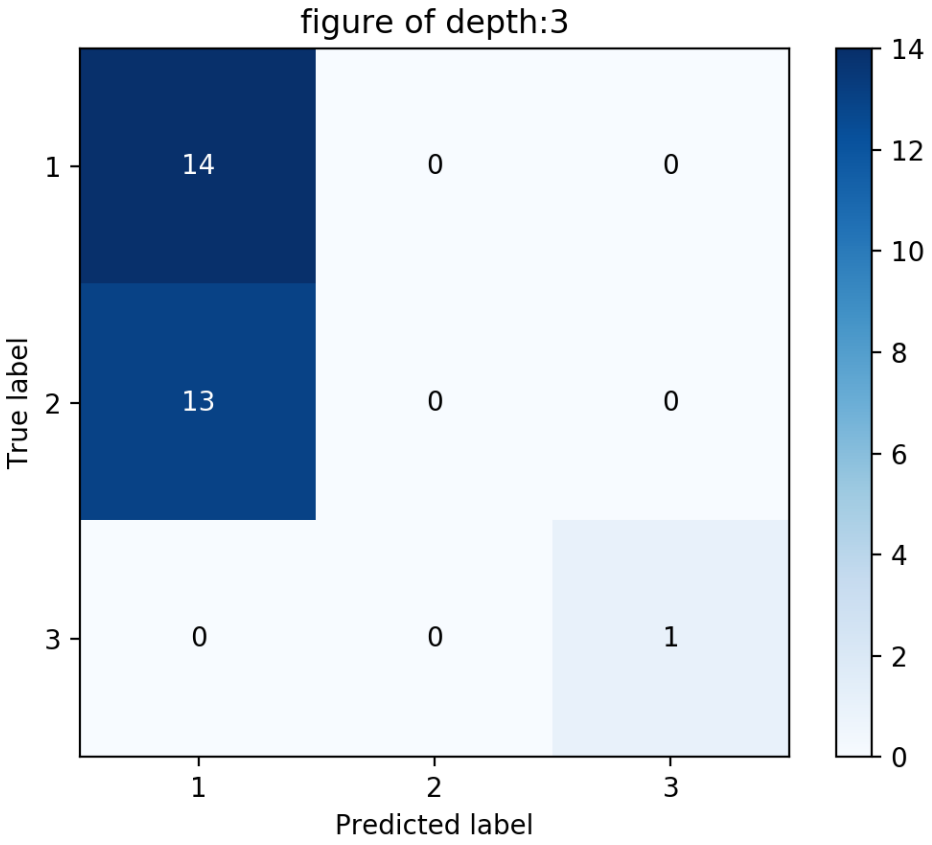
Confusion matrix; (my implementation):



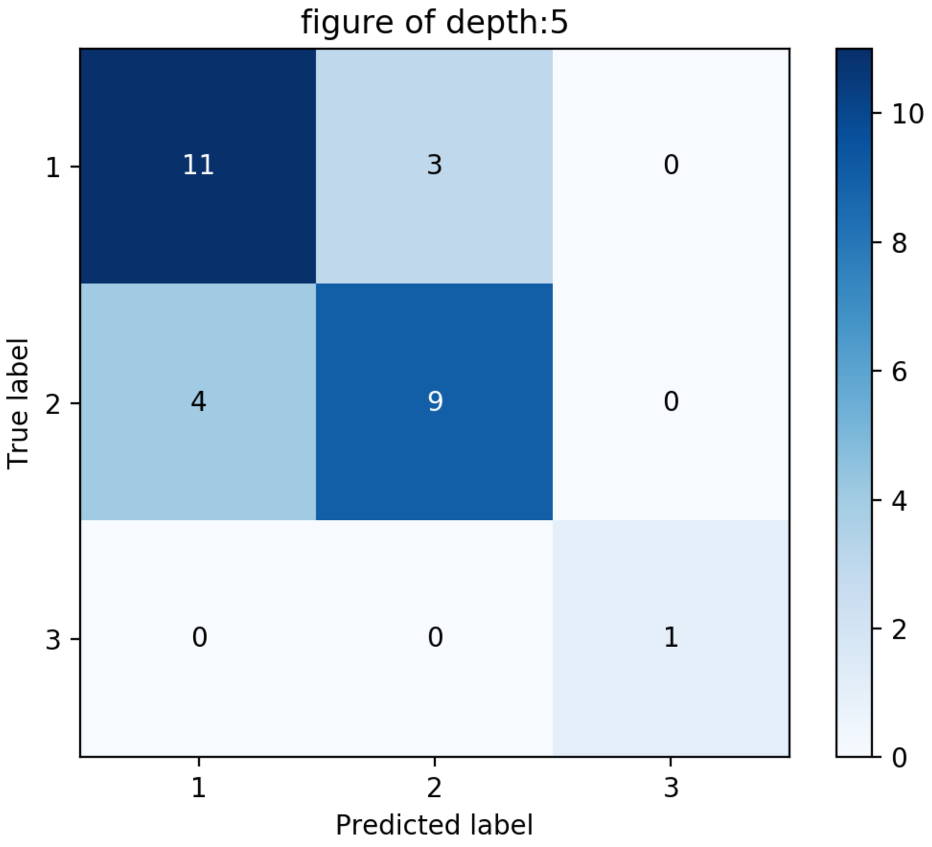
ii) Hayes-roth dataset
Sklearn confusion matrix:
depth 1(sklearn implementation)



Depth 3: (sklearn implementation)



Depth 5: (sklearn implementation)



Discussion:
We are implementing id3 algorithm for building decision trees.
When splitting data on the best pair of values, we get all rows having the best value for the best attribute in one subset and not having the value in one.
i.e:
e.g
col=best attribute
value=best value of x[col]
then,
x_left=>all rows where x[col]==value
x_right=> others
Whereas, in sklearns implementation,

col=best attribute
value=best value of x[col]

then,
x_left=>all rows where **x[col]>=value**
x_right=> all rows where **x[col]<value**

Also, while our algorithm finds the best value of the input attribute, sklearn's implementation focusses on finding a threshold

Our algorithm works well for categorical data

It might not work as well for non categorical data
For non categorical data, we might have to convert it to categories.
The outputs of sklearn's classifier and our classifier are similar for the very low depths.
Overall, our implementation of id3 algorithm for decision tree classifiers is comparable to sklearn's for categorical data.
Accuracy generally increases as max depth increases.