**Bagging**: For the mushroom dataset, used self-implemented bagging and boosting algorithm that uses id3 and computed the confusion matrix on the test set for tree depths 3, 5 and bag size 5, 10.

**Scikit Learner**: For mushroom, used scikit-learn’s BaggingClassifier and AdaBoostClassifier that use DecisionTreeClassifier to learn a tree using criterion=’entropy’ and computed the confusion matrix on the test set for stump depths 1, 2 and ensemble size 5, 10.

***Self-implementation on mushroom***

**Bagging**

tree depth = 3

bag size = 5

(tn, fp, fn, tp) = (216, 0, 108, 108)

[[216 0]

[108 108]]

***Scikit-learn's implementation on mushroom***

**Bagging**

tree depth = 3

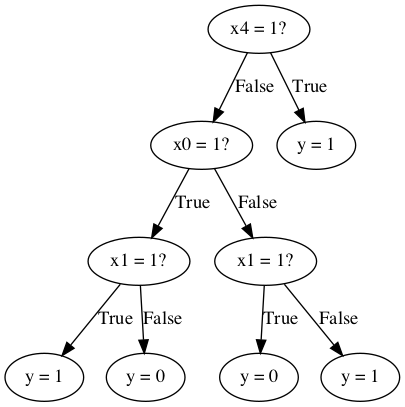
bag size = 5

(tn, fp, fn, tp) = (216, 0, 108, 108)

[[216 0]

[108 108]]

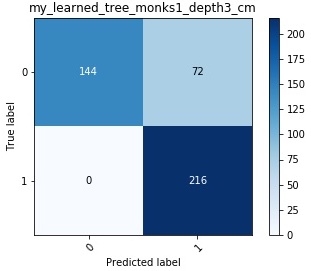
For depth = 3



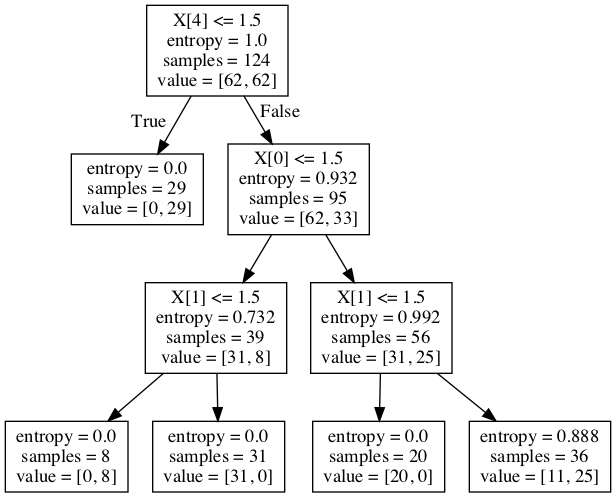
(tn, fp, fn, tp) = (144, 72, 0, 216)

[[144 72]

[ 0 216]]



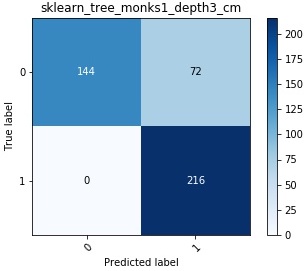
For depth = 3



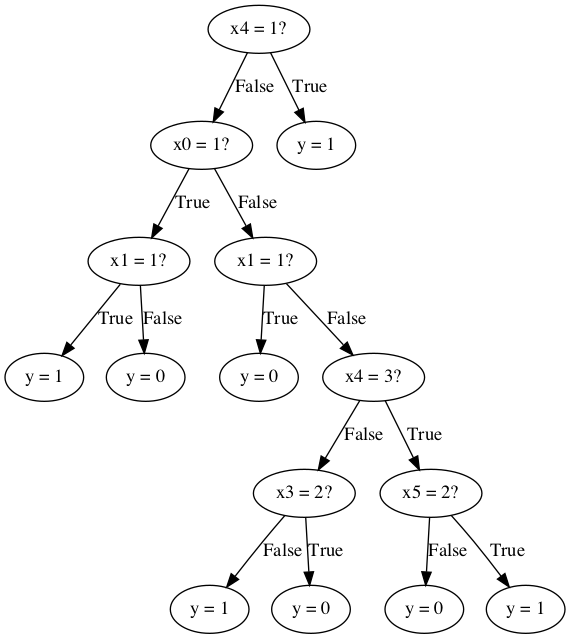
(tn, fp, fn, tp) = (144, 72, 0, 216)

[[144 72]

[ 0 216]]



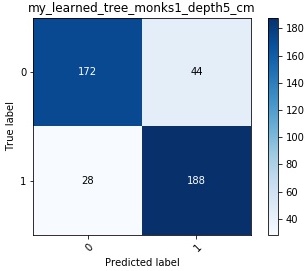
For depth = 5



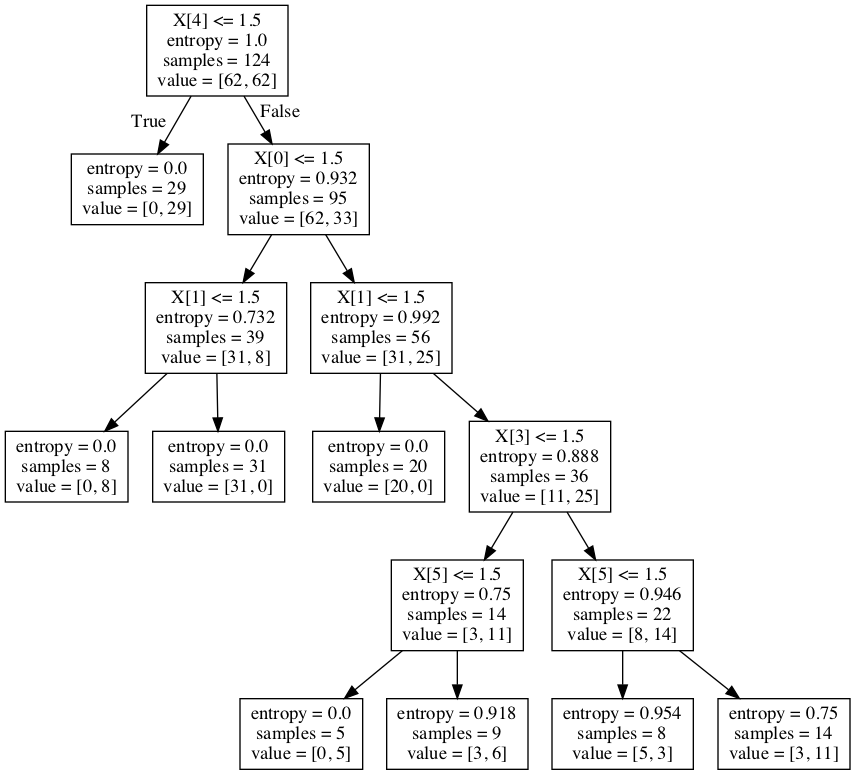
(tn, fp, fn, tp) = (172, 44, 28, 188)

[[172 44]

[ 28 188]]



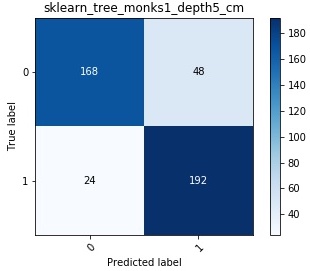
For depth = 5



(tn, fp, fn, tp) = (168, 48, 24, 192)

[[168 48]

[ 24 192]]

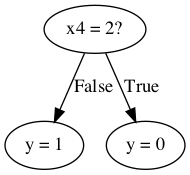


**Weak Learner**: For tic-tac-toe dataset, used self-implemented id3 algorithm to learn a decision tree and computed the confusion matrix on the test set for depths 1, 3, 5.

**Scikit Learner**: For tic-tac-toe dataset, used scikit-learn’s DecisionTreeClassifier to learn a decision tree using criterion=’entropy’ and computed the confusion matrix on the test set for depths 1, 3, 5.

***Self-implementation on tic-tac-toe data***

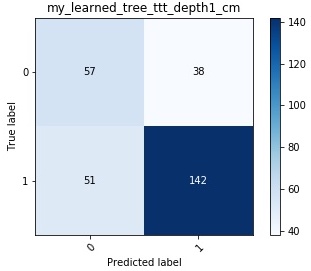
For depth = 1



(tn, fp, fn, tp) = (57, 38, 51, 142)

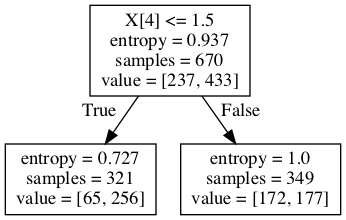
[[ 57 38]

[ 51 142]]



***Scikit-learn's impl on tic-tac-toe data***

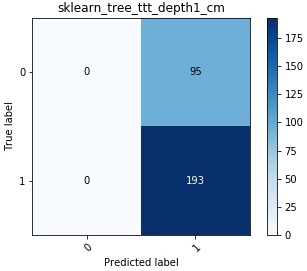
For depth = 1



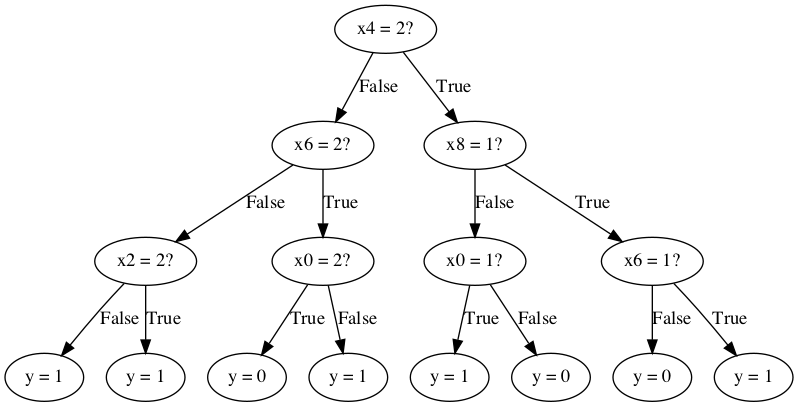
(tn, fp, fn, tp) = (0, 95, 0, 193)

[[ 0 95]

[ 0 193]]



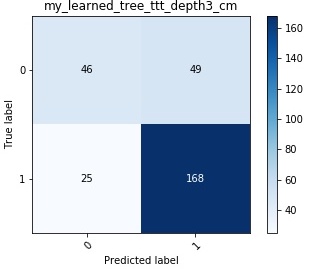
For depth = 3



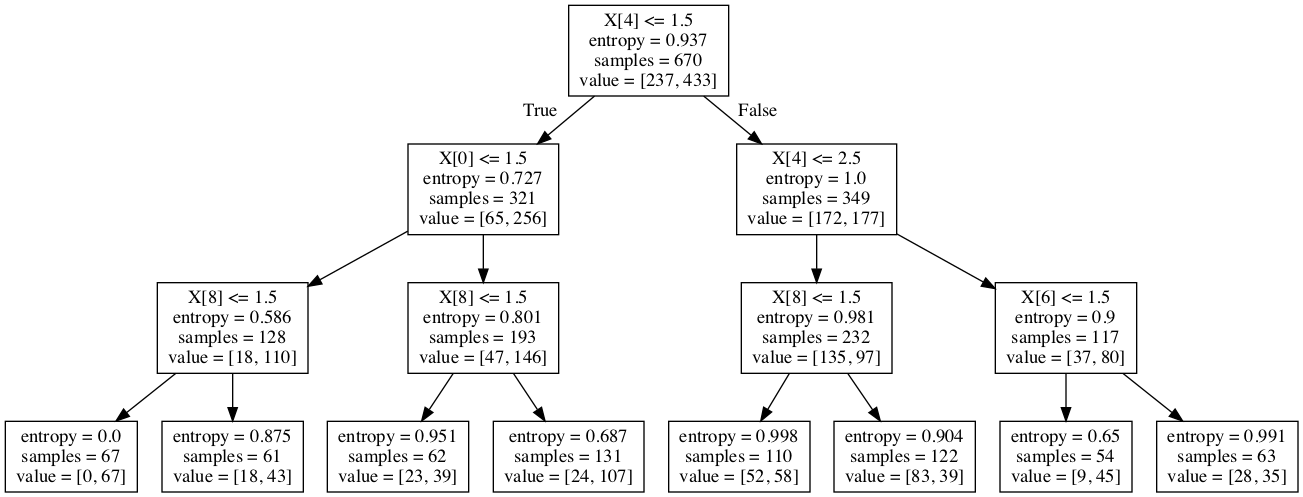
(tn, fp, fn, tp) = (46, 49, 25, 168)

[[ 46 49]

[ 25 168]]



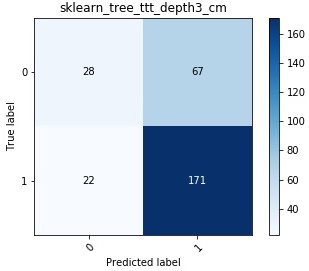
For depth = 3



(tn, fp, fn, tp) = (28, 67, 22, 171)

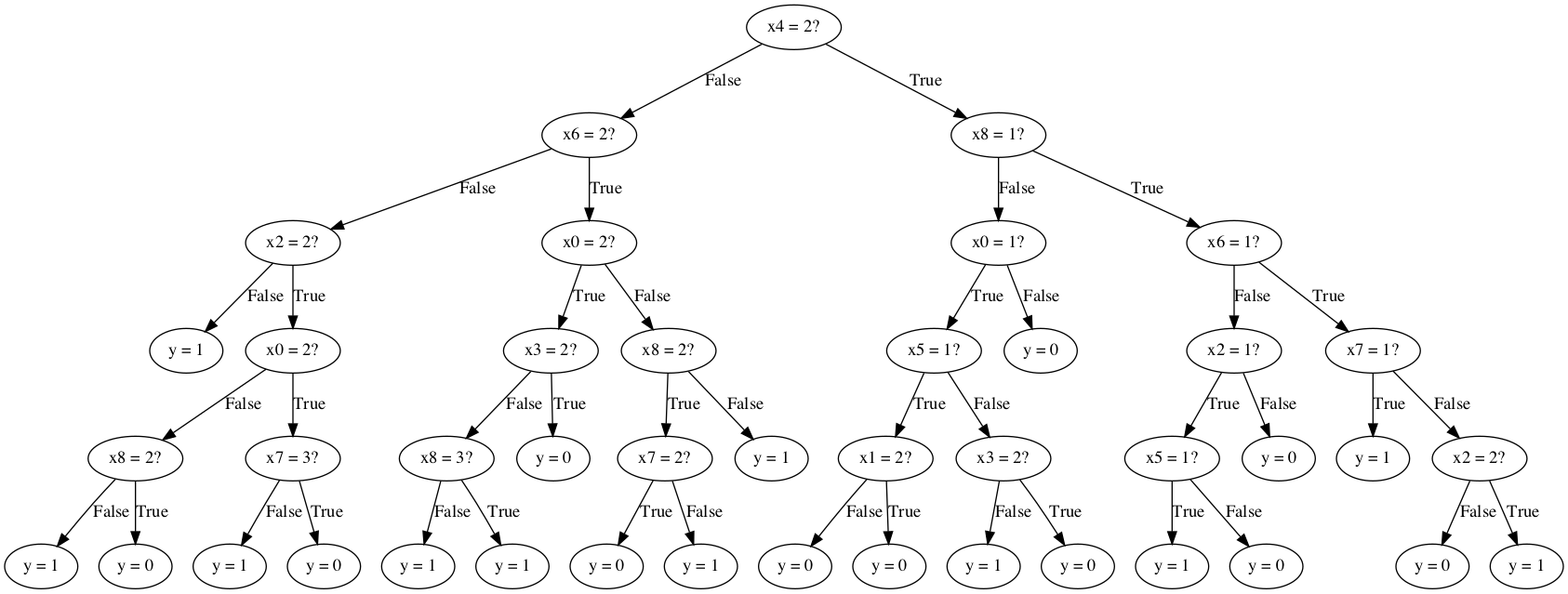
[[ 28 67]

[ 22 171]]



Self-implementation

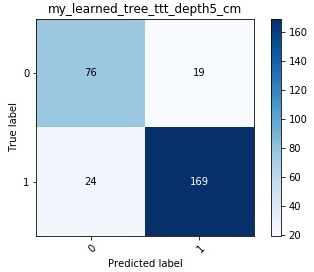
For depth = 5



(tn, fp, fn, tp) = (76, 19, 24, 169)

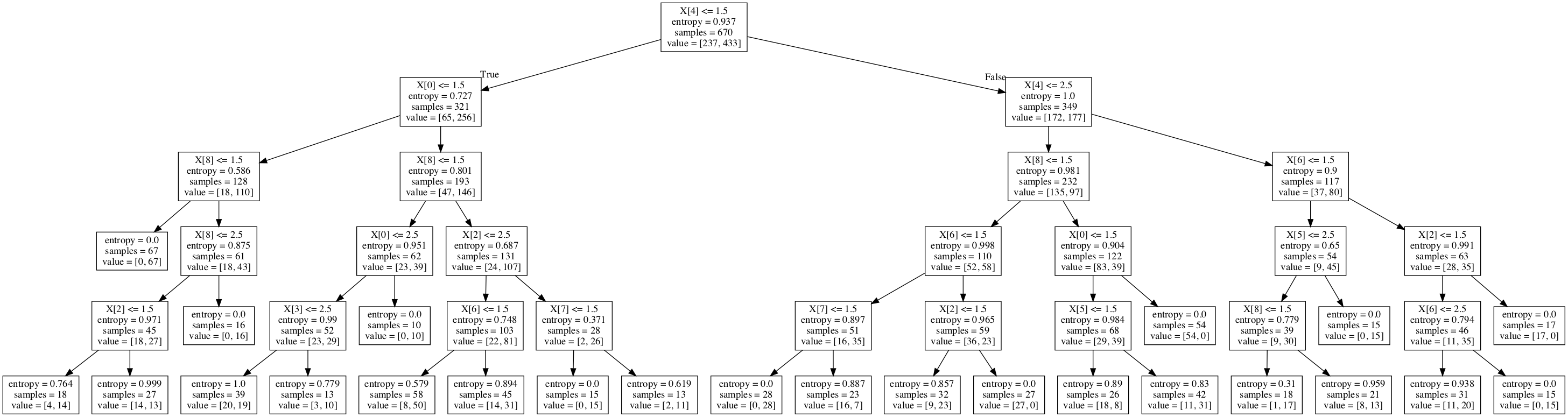
[[ 76 19]

[ 24 169]]



Scikit-learn's implementation

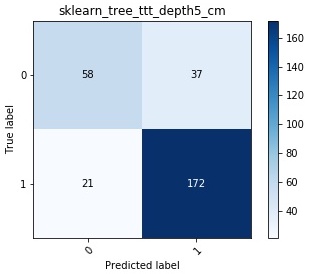
For depth = 5



(tn, fp, fn, tp) = (58, 37, 21, 172)

[[ 58 37]

[ 21 172]]



* In monks1 dataset

﻿For depth = 1, (tn, fp, fn, tp) = (216, 0, 108, 108) ﻿Self-implementation on monks-1

For depth = 1, (tn, fp, fn, tp) = (216, 0, 108, 108) Scikit-learn's implementation on monks-1

For depth = 3, (tn, fp, fn, tp) = (144, 72, 0, 216) ﻿Self-implementation on monks-1

For depth = 3, (tn, fp, fn, tp) = (144, 72, 0, 216) Scikit-learn's implementation on monks-1

For depth = 5, (tn, fp, fn, tp) = (172, 44, 28, 188) ﻿Self-implementation on monks-1

For depth = 5, (tn, fp, fn, tp) = (168, 48, 24, 192) Scikit-learn's implementation on monks-1

The self-implementation and Scikit implementation both produce the same tree and confusion matrix for depth 1 and 3. For depth 5, there is slight difference.

* In tic-tac-toe dataset

﻿For depth = 1, (tn, fp, fn, tp) = (57, 38, 51, 142) ﻿Self implementation on tic-tac-toe data

For depth = 1, (tn, fp, fn, tp) = (0, 95, 0, 193) Scikit-learn's implementation on tic-tac-toe data

For depth = 3, (tn, fp, fn, tp) = (46, 49, 25, 168) ﻿Self implementation on tic-tac-toe data

For depth = 3, (tn, fp, fn, tp) = (28, 67, 22, 171) Scikit-learn's implementation on tic-tac-toe data

For depth = 5, (tn, fp, fn, tp) = (76, 19, 24, 169) ﻿Self implementation on tic-tac-toe data

For depth = 5, (tn, fp, fn, tp) = (58, 37, 21, 172) Scikit-learn's implementation on tic-tac-toe data

The self-implementation and Scikit implementation both produce slightly different confusion matrix for depths 1, 3, and 5.