CS6375.003-Machine Learning
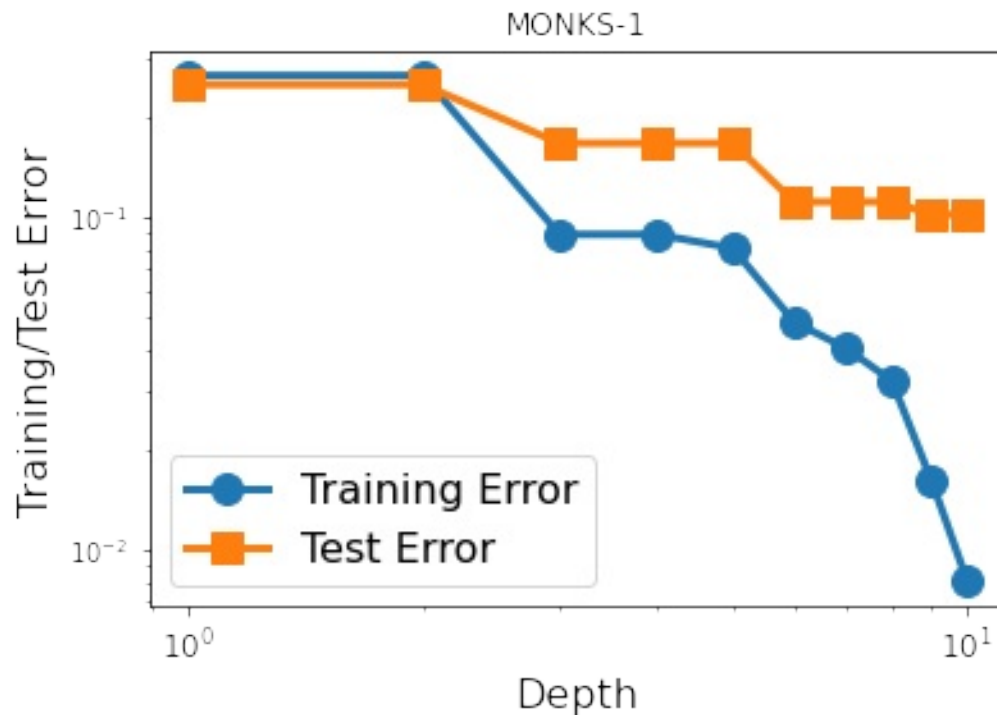
Assignment 2 Report – Fixed Length Decision Tree

By – Srivastchavan Rengarajan(sxr190067)

b)
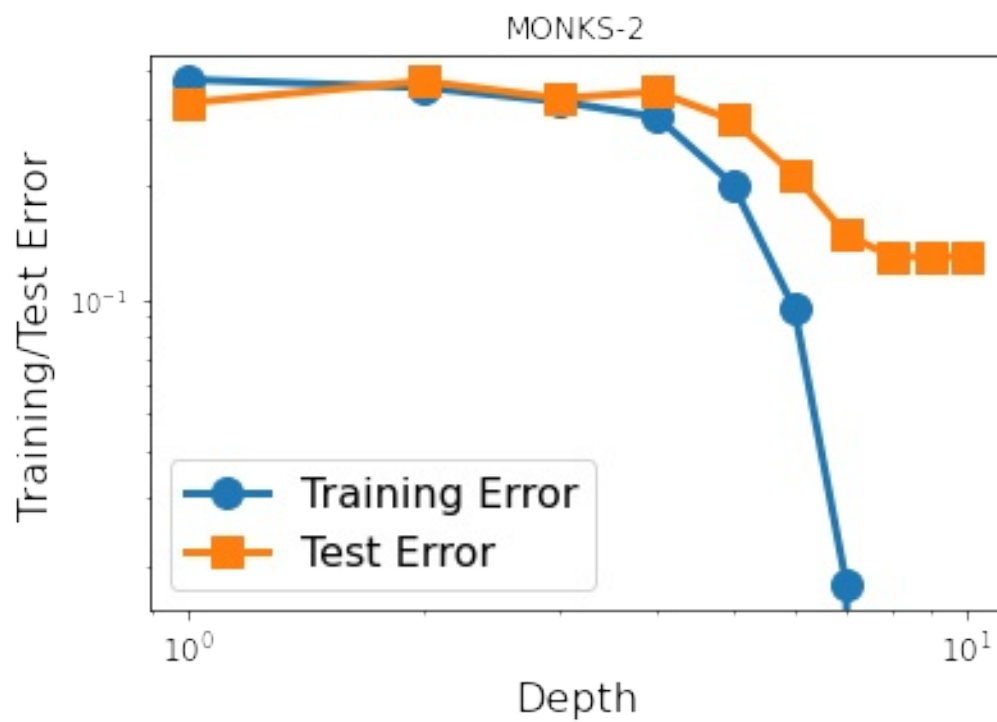
Graph for Monks-1 Dataset:

X- axis – Depth of Tree

Y-axis -Training/Test Error

Graph for Monks-2 Dataset:
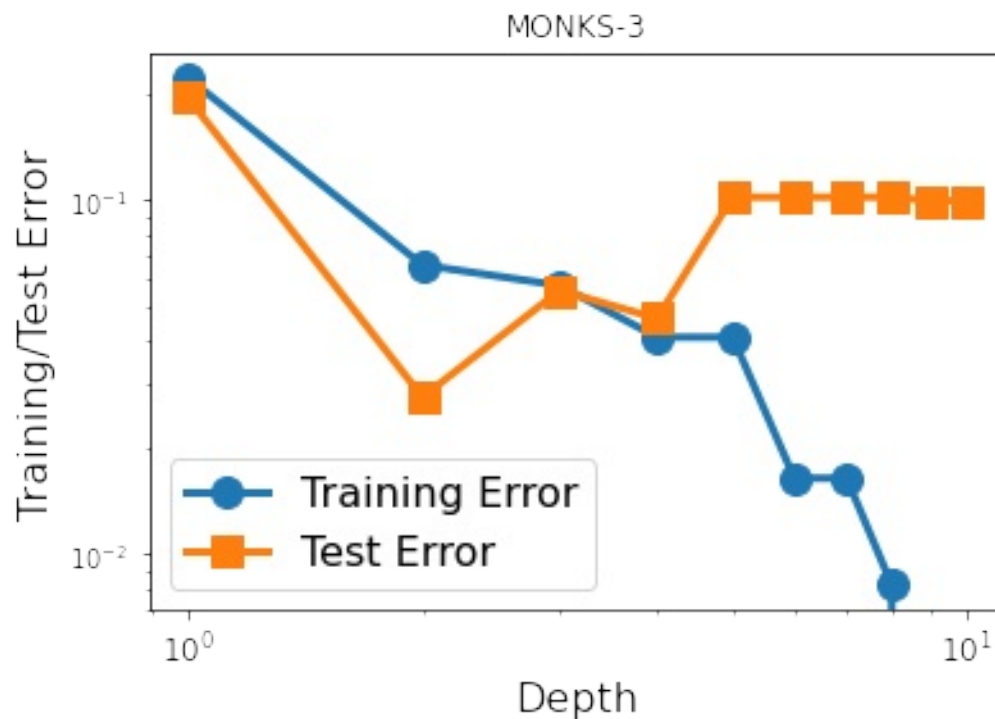
X- axis – Depth of Tree

Y-axis -Training/Test Error

Graph for Monks-3 Dataset:
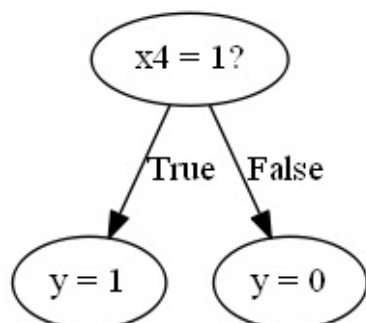
X- axis – Depth of Tree

Y-axis -Training/Test Error

c)

Decision Tree Monks-1 Dataset-

Depth=1:

```
TREE
+-- [SPLIT: x4 = 1 True]
|       +-- [LABEL = 1]
+-- [SPLIT: x4 = 1 False]
|       +-- [LABEL = 0]
```

Test Error = 25.00%.

MONKS Dataset: **Confusion matrix for depth 1**

	Predicted Positives	Predicted Negatives
True Positives	216	0
True Negatives	108	108

Depth=3:

```
TREE
+-- [SPLIT: x4 = 1 True]
|      +-- [LABEL = 1]
+-- [SPLIT: x4 = 1 False]
|      +-- [SPLIT: x0 = 1 True]
|      |      +-- [SPLIT: x1 = 1 True]
|      |      |      +-- [LABEL = 1]
|      |      +-- [SPLIT: x1 = 1 False]
|      |      |      +-- [LABEL = 0]
|      +-- [SPLIT: x0 = 1 False]
|      |      +-- [SPLIT: x1 = 1 True]
|      |      |      +-- [LABEL = 0]
|      |      +-- [SPLIT: x1 = 1 False]
|      |      |      +-- [LABEL = 1]
```

Test Error = 16.67%.

MONKS Dataset: **Confusion matrix for depth 3**

	Predicted Positives	Predicted Negatives
True Positives	144	72
True Negatives	0	216

Depth=5:

```
TREE
+-- [SPLIT: x4 = 1 True]
|       +-- [LABEL = 1]
+-- [SPLIT: x4 = 1 False]
|       +-- [SPLIT: x0 = 1 True]
```
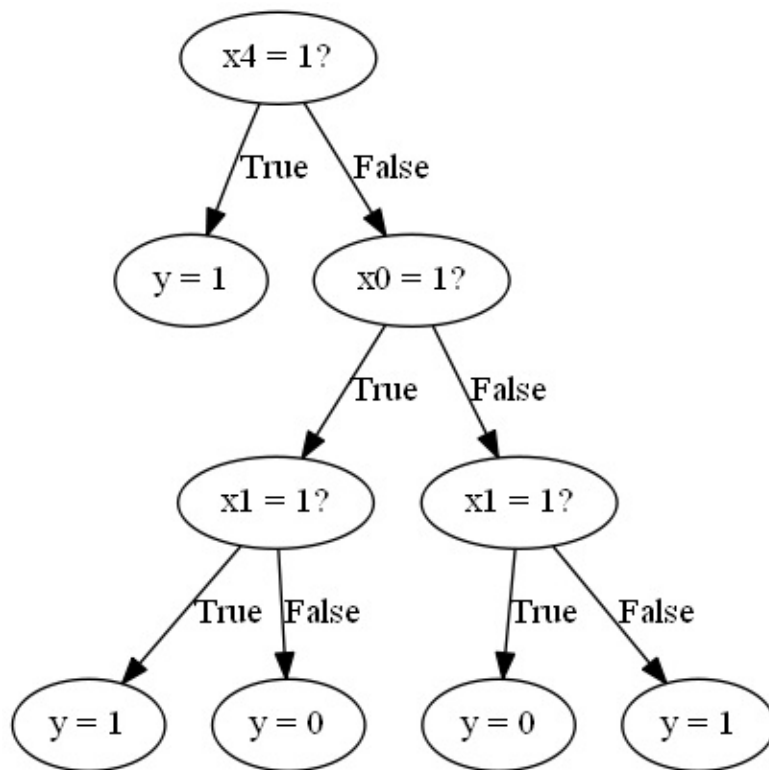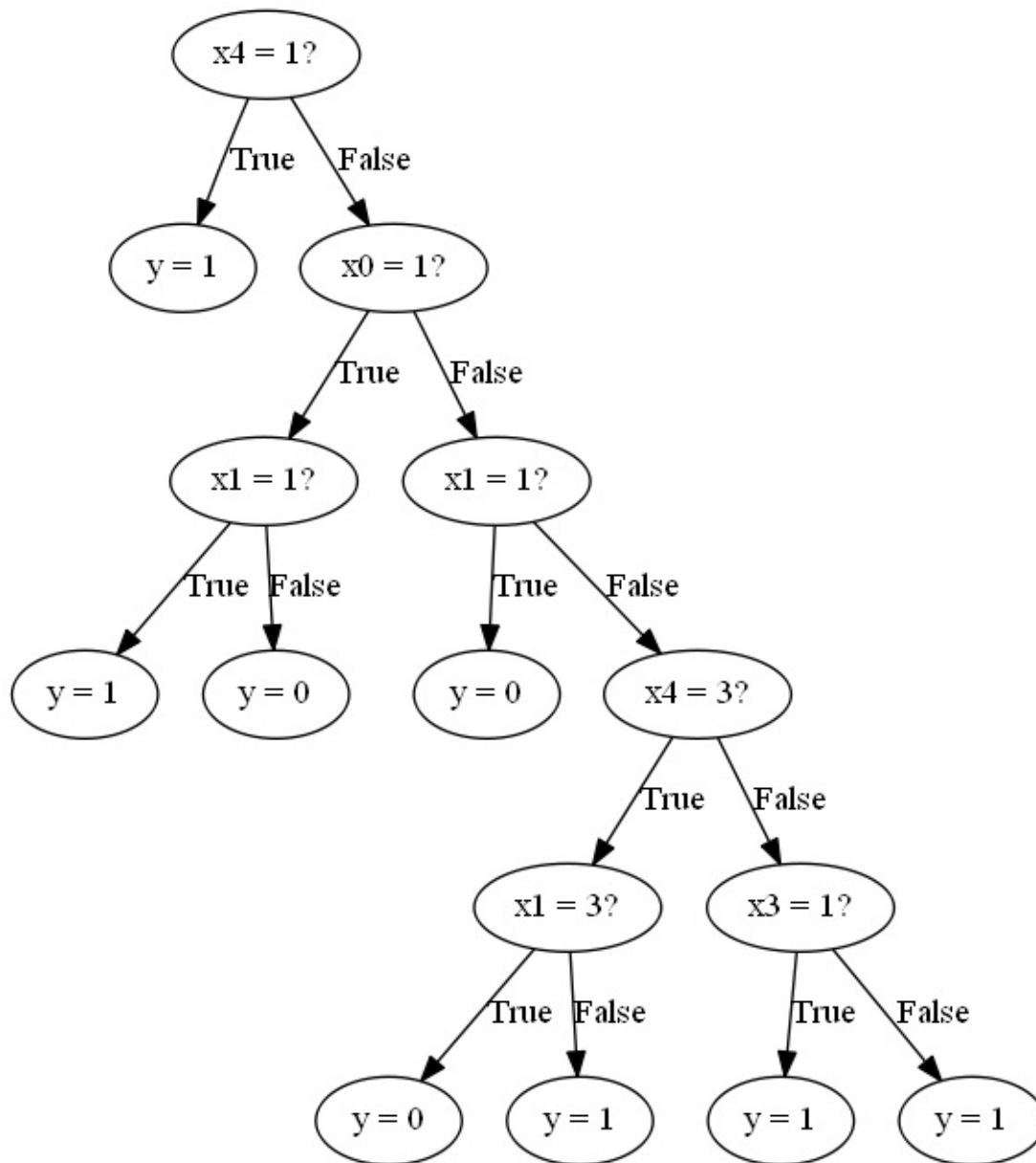
```
|          |              +-- [SPLIT: x1 = 1 True]
|          |              |        +-- [LABEL = 1]
|          |              +-- [SPLIT: x1 = 1 False]
|          |              |        +-- [LABEL = 0]
|          +-- [SPLIT: x0 = 1 False]
|          |              +-- [SPLIT: x1 = 1 True]
|          |              |        +-- [LABEL = 0]
|          |              +-- [SPLIT: x1 = 1 False]
|          |              |        +-- [SPLIT: x4 = 3 True]
|          |              |        |        +-- [SPLIT: x1 = 3 True]
|          |              |        |        |        +-- [LABEL = 0]
|          |              |        |        +-- [SPLIT: x1 = 3 False]
|          |              |        |        |        +-- [LABEL = 1]
|          |              |        +-- [SPLIT: x4 = 3 False]
|          |              |        |        +-- [SPLIT: x3 = 1 True]
|          |              |        |        |        +-- [LABEL = 1]
|          |              |        |        +-- [SPLIT: x3 = 1 False]
|          |              |        |        |        +-- [LABEL = 1]

Test Error = 16.67%.
```

MONKS Dataset: **Confusion matrix for depth 5**

	Predicted Positives	Predicted Negatives
True Positives	156	60
True Negatives	12	204

d)

Monk1 Dataset Decision Tree and Confusion matrix using scikit-learn

Depth=1:

```
Test Error = 25.00%.
MONKS Dataset: Confusion matrix for depth  1
                  Predicted Positives   Predicted
Negatives
True Positives                    216
0
True Negatives                    108
108
```

Depth=3:

```
Test Error = 16.67%.
MONKS Dataset: Confusion matrix for depth  3
                  Predicted Positives   Predicted
Negatives
True Positives                    144
72
True Negatives                      0
216
```

Depth=5:

```
Test Error = 16.67%.
MONKS Dataset: Confusion matrix for depth  5
                  Predicted Positives   Predicted
Negatives
True Positives                    168
48
True Negatives                     24
192
```

E)

Dataset used from UCI Repo - Dishonest Internet users dataset

URL-
https://archive.ics.uci.edu/ml/datasets/Dishonest+Inte

Attribute Information:

1) CT {CT_range_1, CT_range_2, CT_range_3, CT_range_4}

2) CU {CU_range_1, CU_range_2, CU_range_3, CU_range_4}

3) LT {LT_range_1, LT_range_2, LT_range_3, LT_range_4}

4) TC {sport, game, ECommerce, holiday}

5) TS {trustworthy, untrustworthy}

The numerical attributes (CT, CU, LT) was discretized.

Dishonest Internet users dataset- ID3 results

Depth=1:

```
TREE
+-- [SPLIT: x3 = 3 True]
|      +-- [LABEL = 0]
+-- [SPLIT: x3 = 3 False]
|      +-- [LABEL = 1]
```
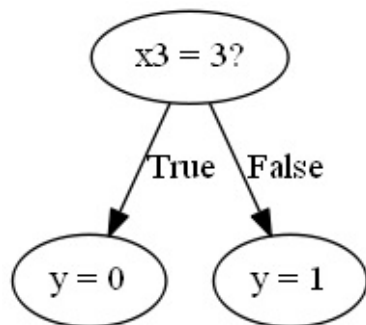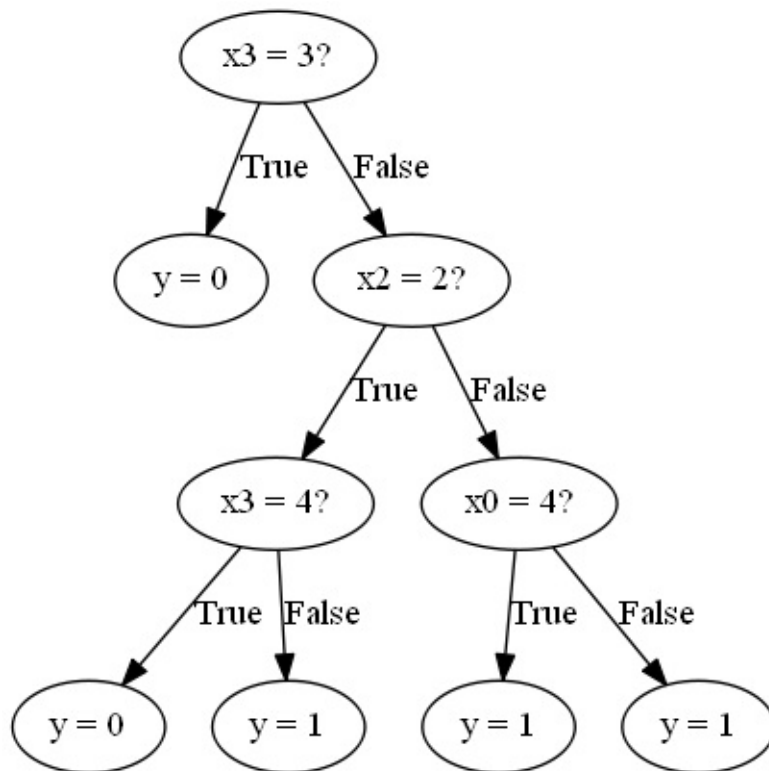
Dishonest Internet Users Dataset: **Confusion matrix for depth 1**

	Predicted Positives	Predicted Negatives
True Positives	11	21
True Negatives	0	75

Depth=3:

```
TREE
+-- [SPLIT: x3 = 3 True]
|      +-- [LABEL = 0]
+-- [SPLIT: x3 = 3 False]
|      +-- [SPLIT: x2 = 2 True]
|      |      +-- [SPLIT: x3 = 4 True]
|      |      |      +-- [LABEL = 0]
|      |      +-- [SPLIT: x3 = 4 False]
|      |      |      +-- [LABEL = 1]
|      +-- [SPLIT: x2 = 2 False]
|      |      +-- [SPLIT: x0 = 4 True]
|      |      |      +-- [LABEL = 1]
|      |      +-- [SPLIT: x0 = 4 False]
|      |      |      +-- [LABEL = 1]
```

```
            x3 = 3?
          /        \
     True          False
      /                \
   y = 0            x2 = 2?
                  /        \
              True          False
               /                \
          x3 = 4?              x0 = 4?
         /      \             /      \
     True      False      True      False
      /          \         /          \
   y = 0       y = 1    y = 1        y = 1
```

Dishonest Internet Users Dataset: **Confusion matrix for depth   3**

```
                    Predicted Positives   Predicted
Negatives
True Positives                      23
9
True Negatives                       0
75
```

Dishonest Internet users dataset- scikit-learn results

Depth=1:

Dishonest Internet Users Dataset: **Confusion matrix for depth   1**

```
                    Predicted Positives   Predicted
Negatives
True Positives                       0
```

```
32
True Negatives                          0
75
```

Depth=3:

```
Dishonest Internet Users Dataset: Confusion matrix
for depth  3
                    Predicted Positives   Predicted
Negatives
True Positives                     23
9
True Negatives                      0
75
```

Discussion –

As we can see that the results of tree constructed using ID3 algorithm and scikit-learn implementation are very similar, we can conclude that the scikit-learn uses the logic of id3 algorithm for the decision tree construction.

Also, we can see that the confusion matrices in both ID3 implementation and the scikit-learn implementation are also very similar if not the same.