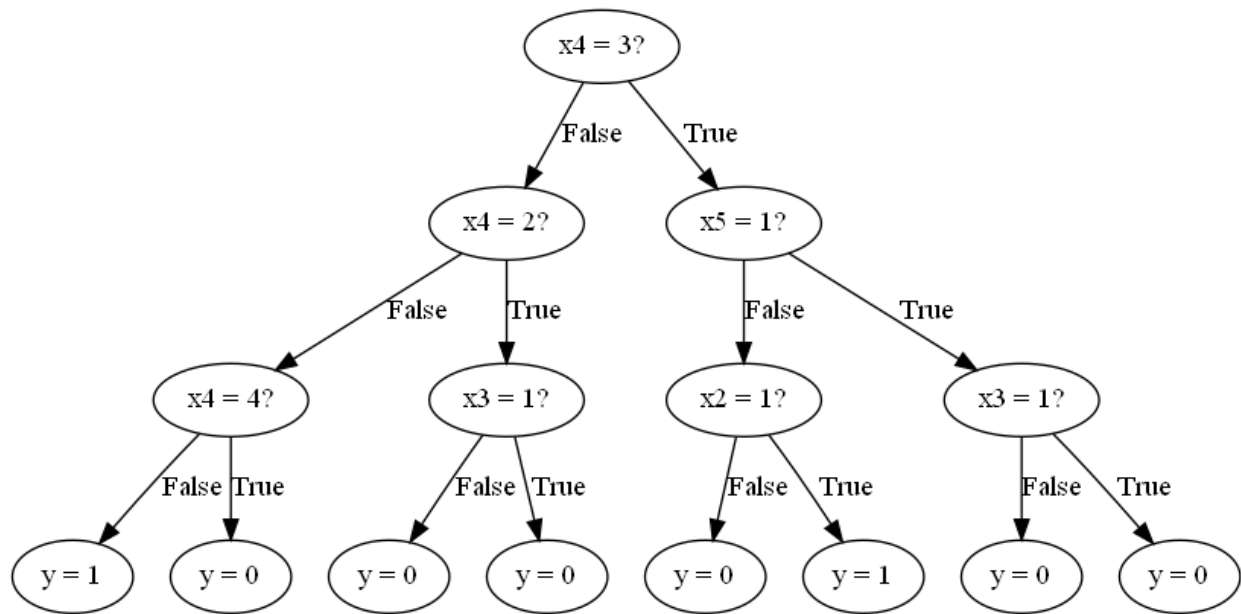
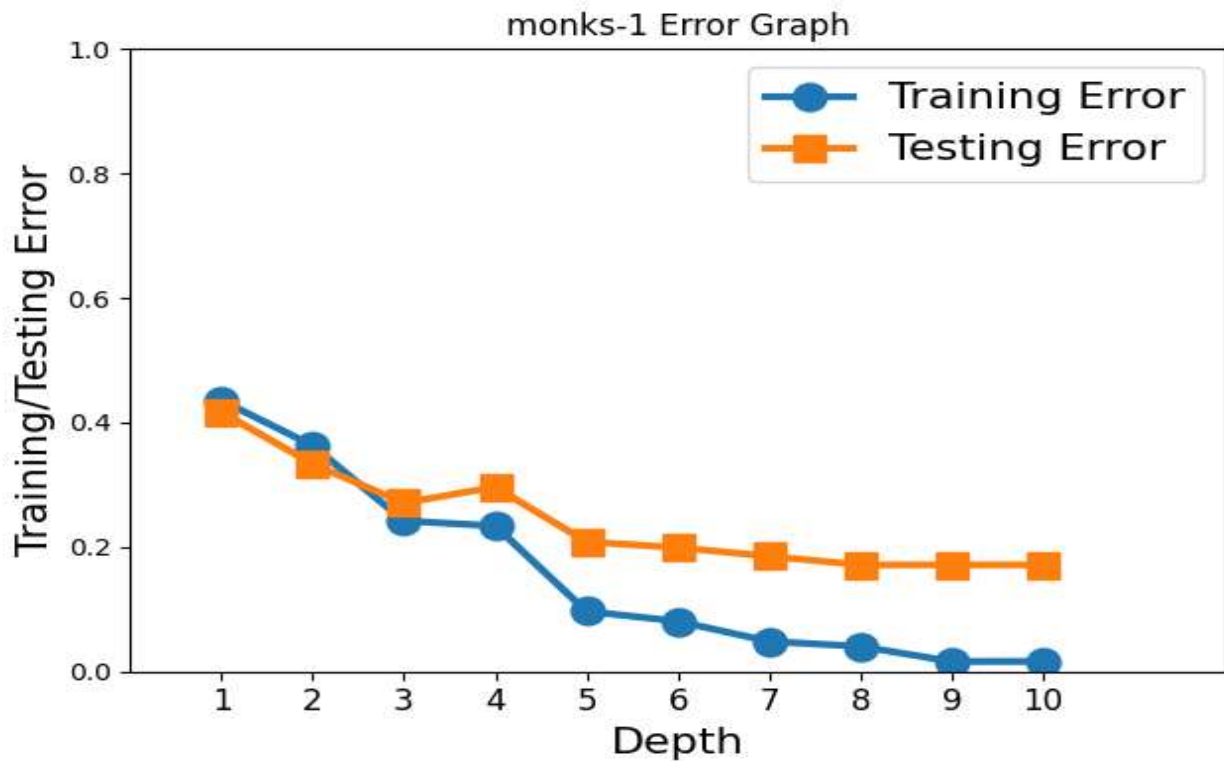


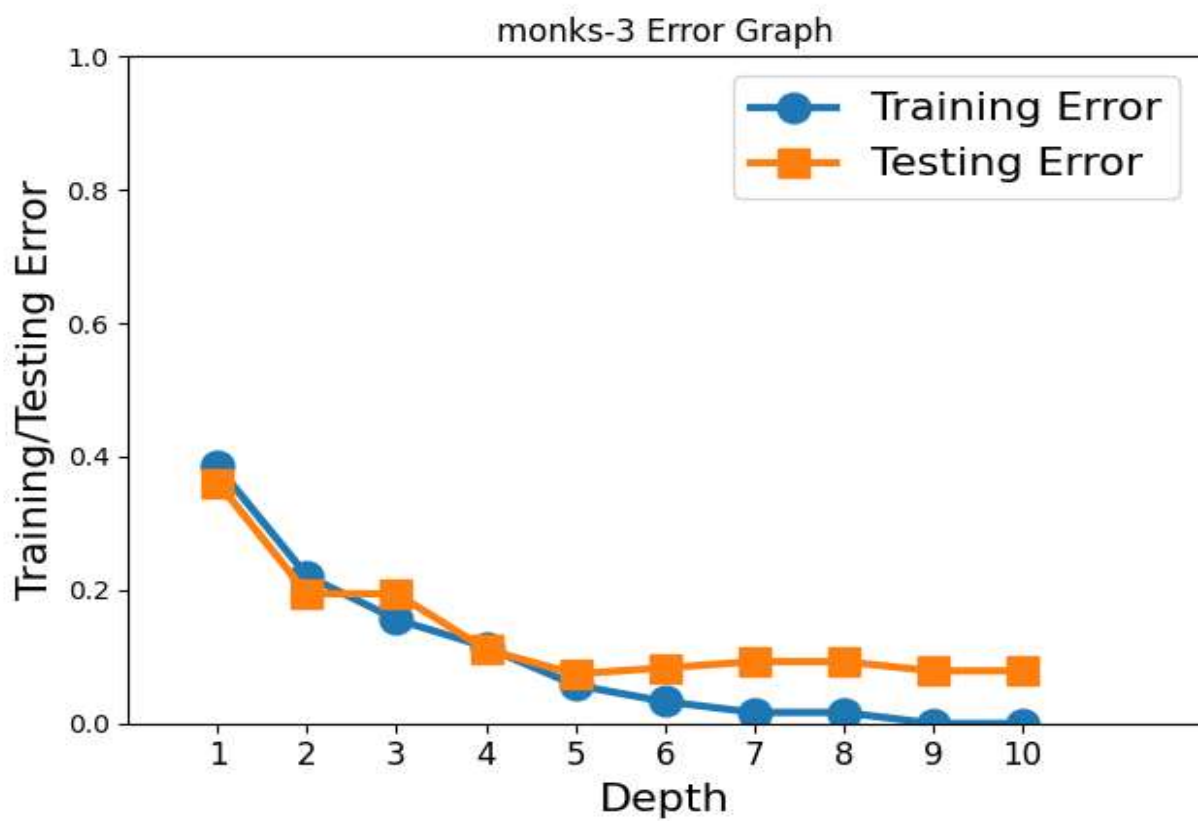
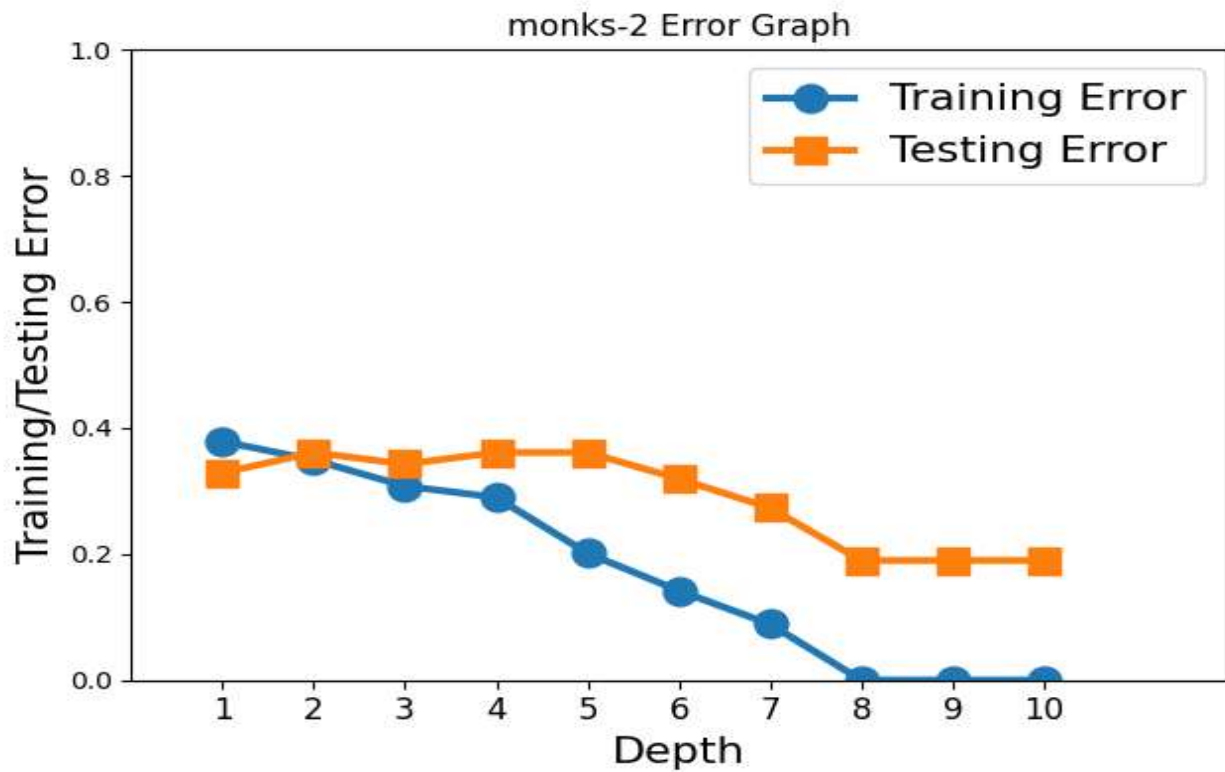
Assignment 2 Plotting Report

Decision Tree

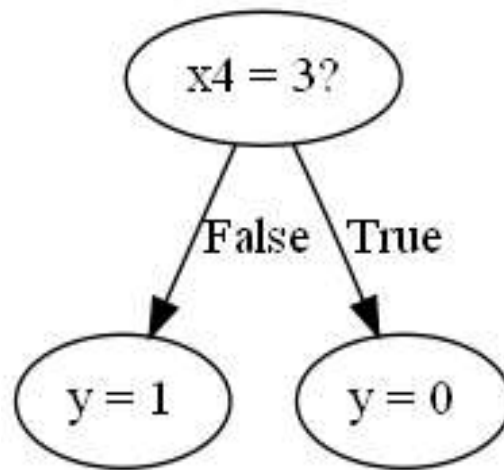


Part b – Plotting learned errors with depths of different monk data.





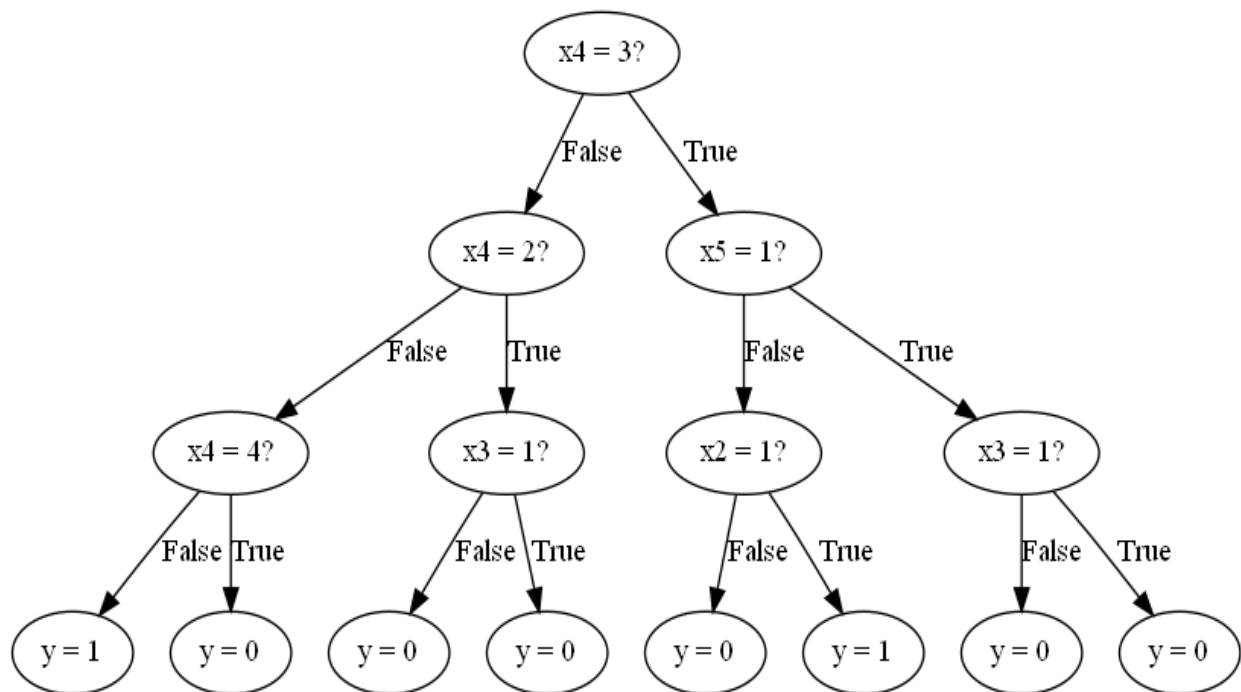
Part c Monk_1 Tree of depth 1,3,5



Monk1_depth:1 confusion matrix

[[72 144]

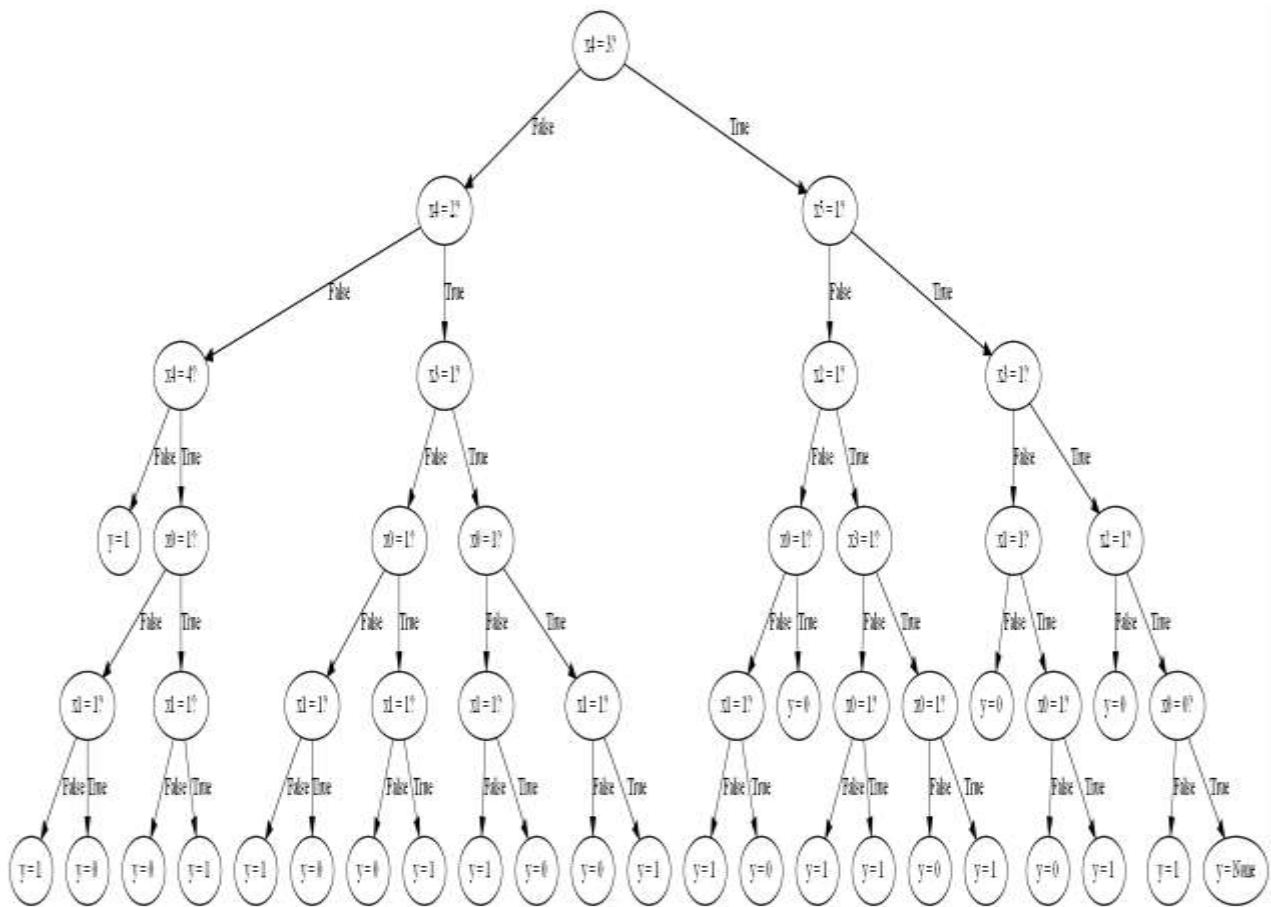
[36 180]]



Monk1_depth:3 confusion matrix

[[198 18]

[99 117]]

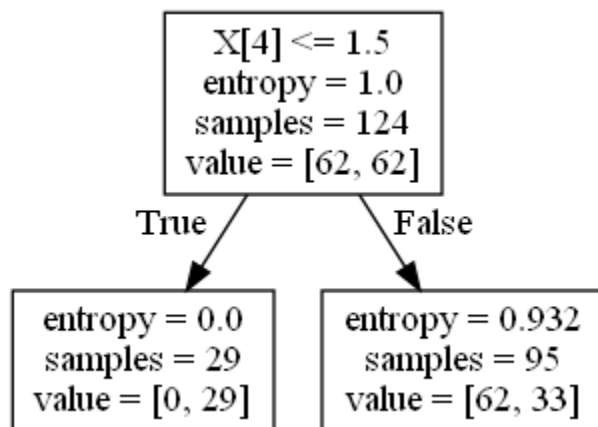


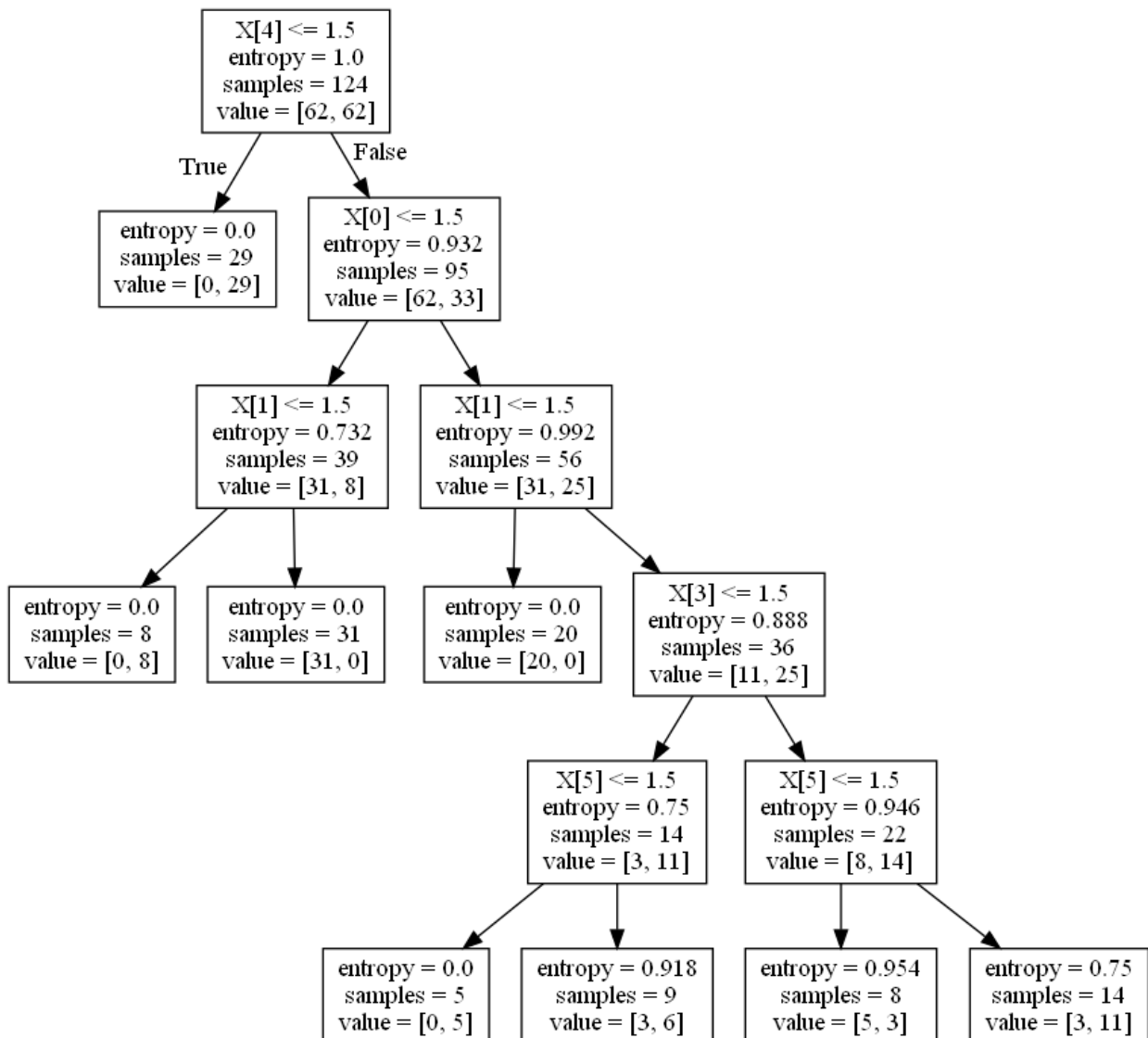
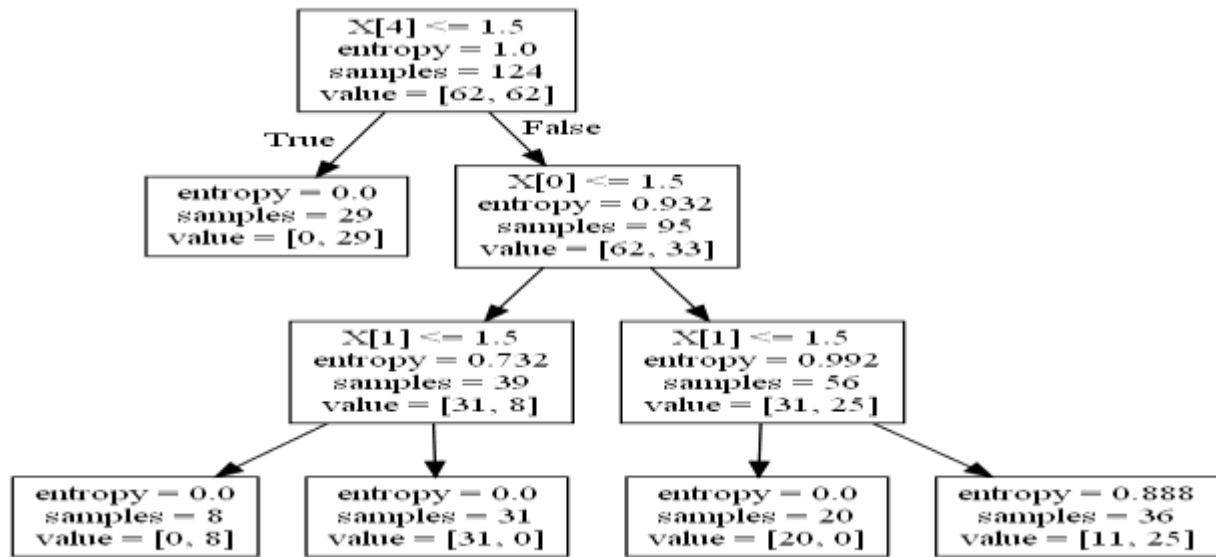
Monk1_depth:5 confusion_matrix

[[142 74]

[16 200]]

Part d Monk_1 Decision_Tree_Classifier to learn a Tree of depth 1,3,5





Along with the different confusion matrix:

Monk1_depth:1 confusion_matrix

```
[[216  0]
```

```
 [108 108]]
```

Monk1_depth:3 confusion_matrix

```
[[144  72]
```

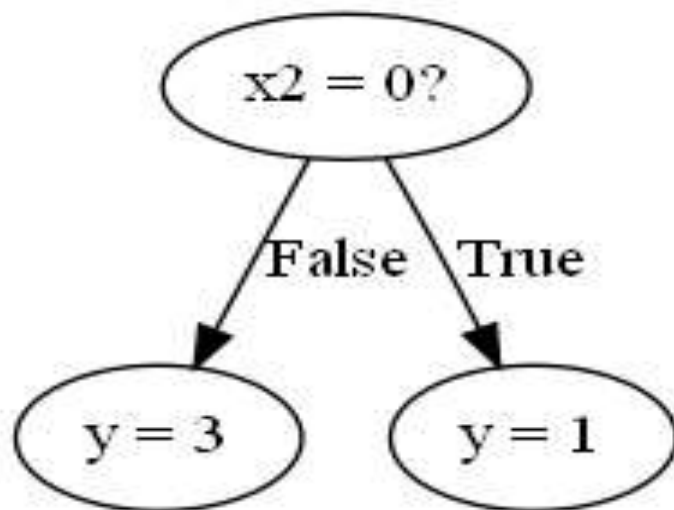
```
 [  0 216]]
```

Monk1_depth:5 confusion_matrix

```
[[168  48]
```

```
 [ 24 192]]
```

Part e(c) Self Data Iris with depth 1,3,5

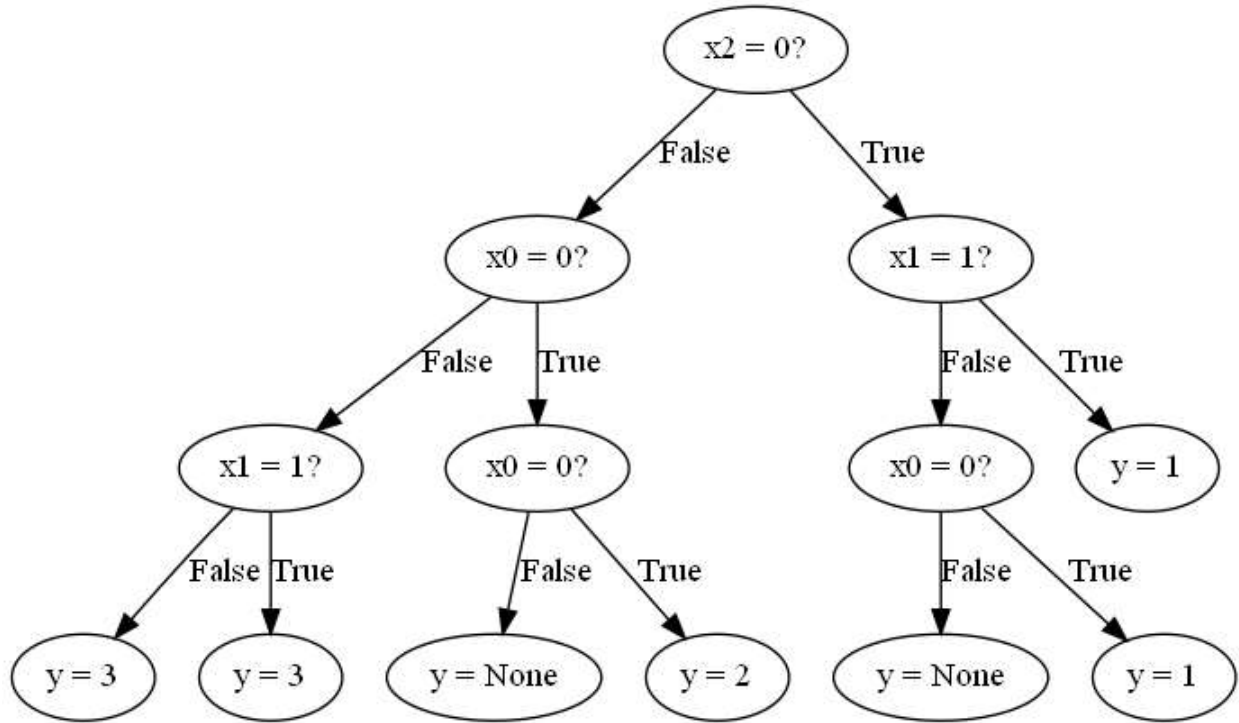


iris_depth:1 confusion_matrix

```
[[ 0  0  0]
```

```
 [ 2  0  8]
```

```
 [ 0  0 10]]
```

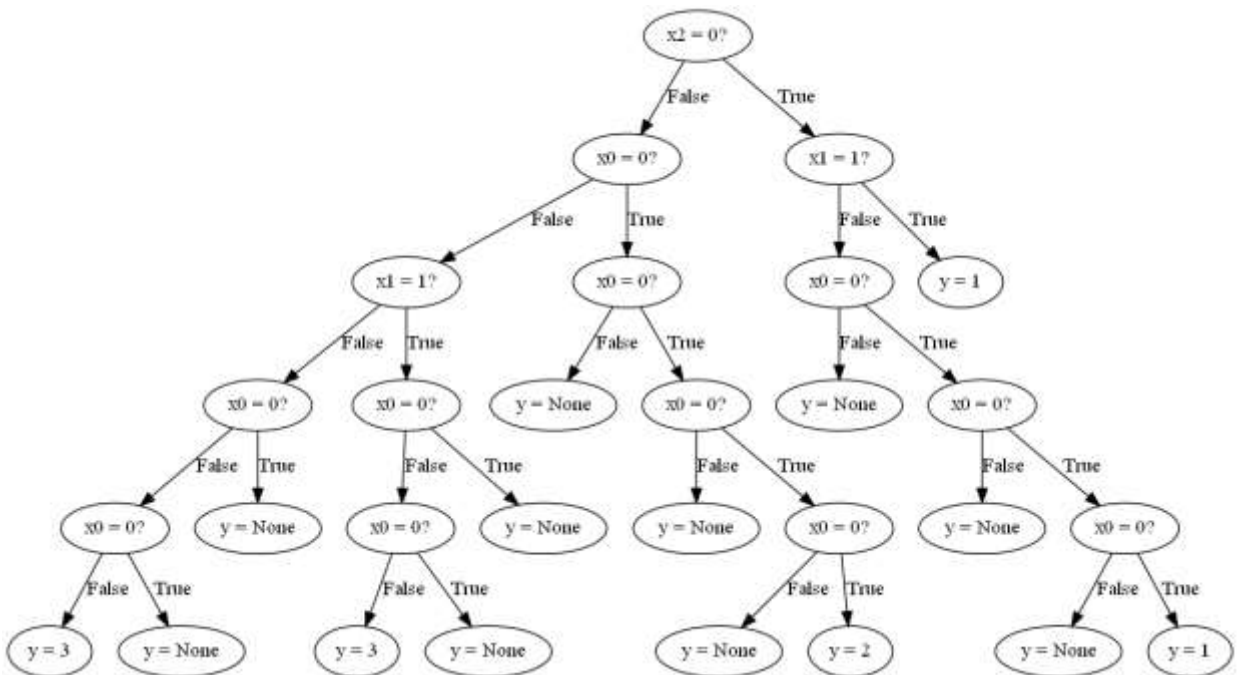


iris_depth:3 confusion_matrix

[[0 0 0]

[2 6 2]

[0 1 9]]



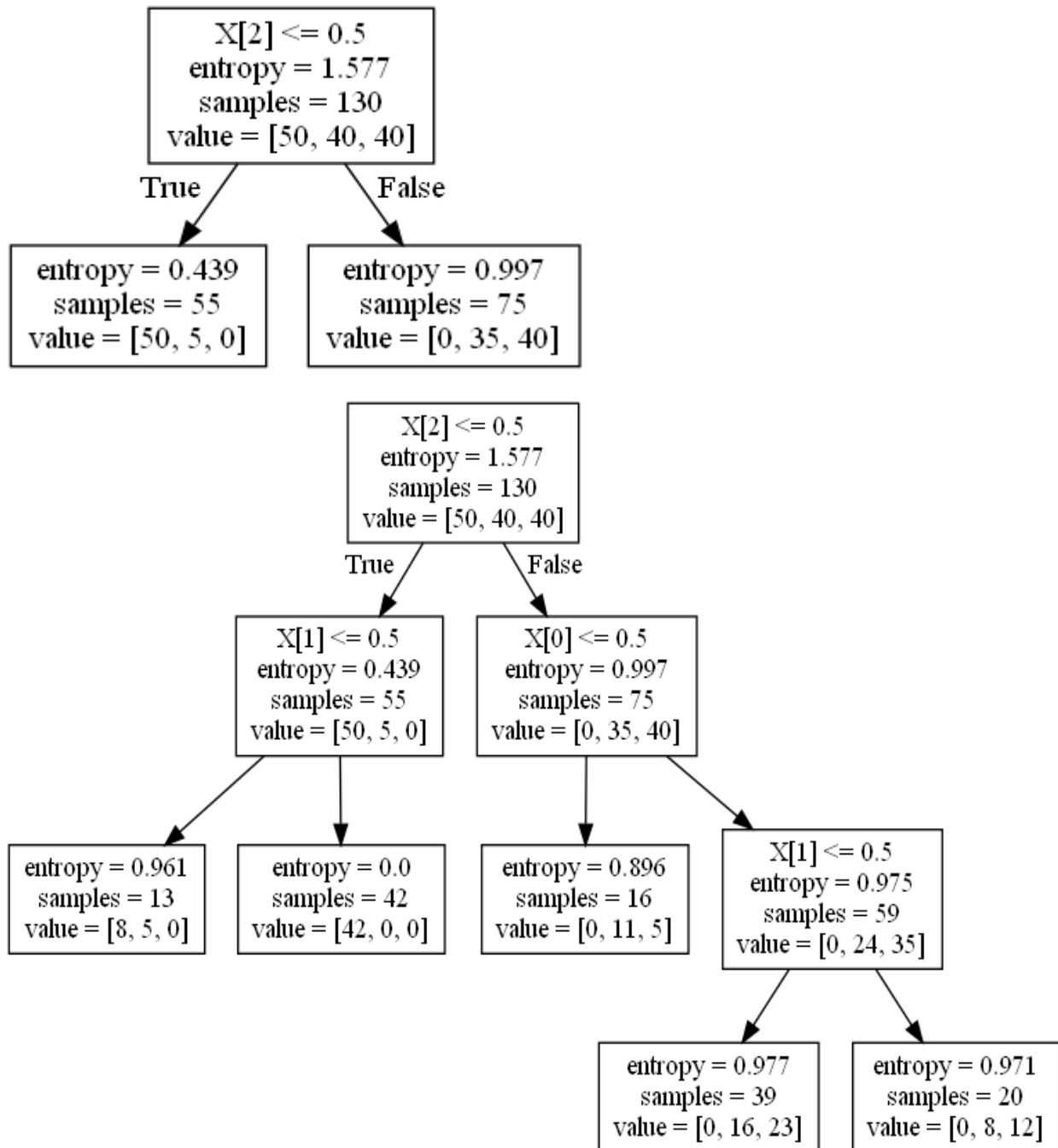
iris_depth:5 confusion_matrix

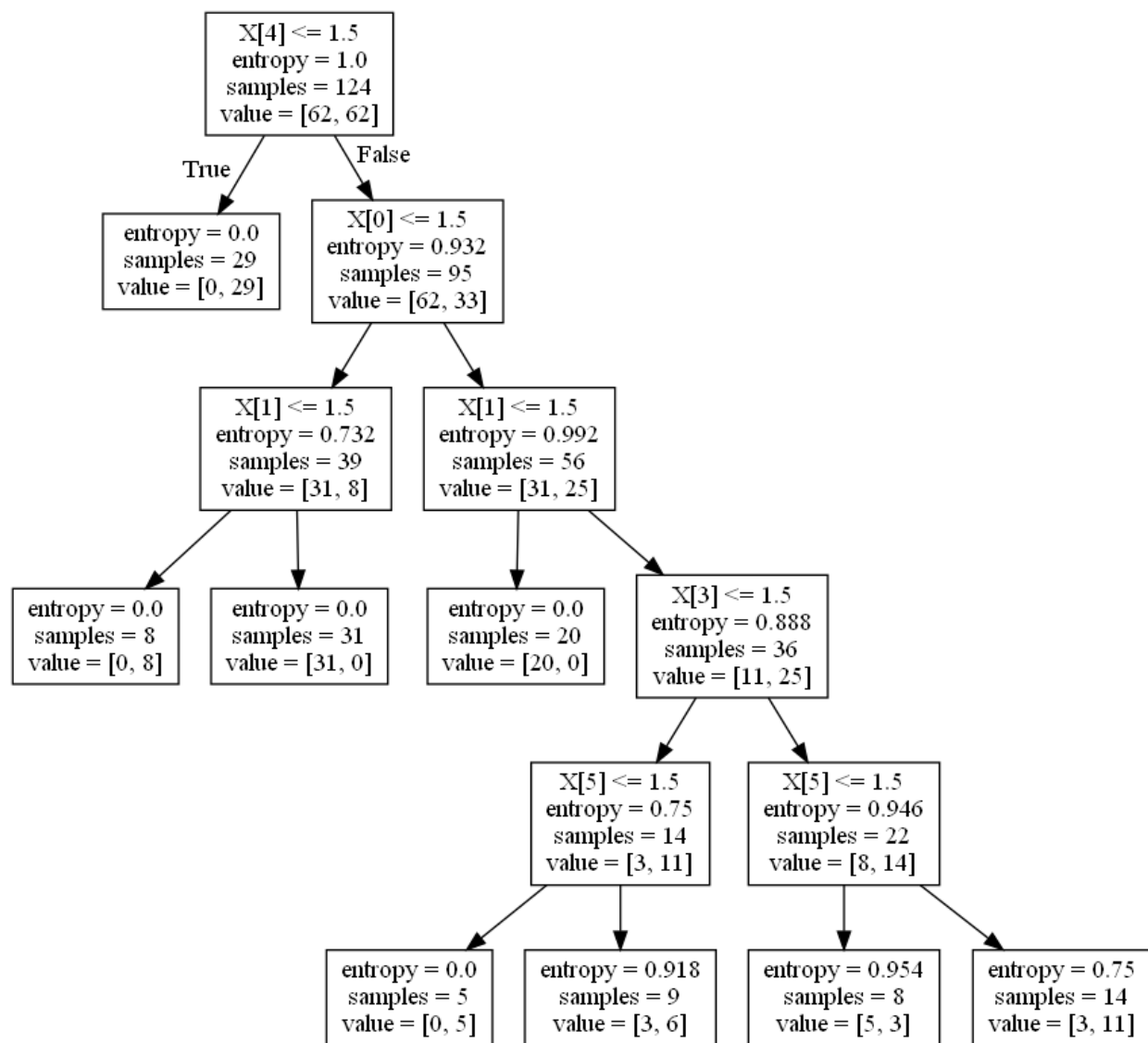
[[0 0 0]

[2 6 2]

[0 1 9]]

Part e(d) (Self Data) Decision_Tree_Classifier to learn a Iris with depth 1,3,5





Along with the confusion matrix:

iris_depth:1 confusion_matrix

```
[[ 0  0  0]
```

```
 [ 2  0  8]
```

```
 [ 0  0 10]]
```

iris_depth:3 confusion_matrix

```
[[0 0 0]
```

```
 [2 6 2]
```

```
[0 1 9]]
```

```
iris_depth:5 confusion_matrix
```

```
[[0 0 0]
```

```
 [2 6 2]
```

```
[0 1 9]]
```

From the above confusion matrix we could observe that ID3 algorithm that I had implement in this assignment is less efficient than the scikit-learn algorithm. For example, it can be tell that if the dataset is large and complicated (monk problem), then scikit-learn is way better than mine ID3 algorithm. But if the dataset is small like what I did for iris (150 datasets), the efficient is quit similar.