Documentation for Decision Trees

Q1.Start from depth = 1 and go to different depths (2,4,6,8...,16). For each depth, compute the error (the number of misclassifications) on the test set. Plot a learning curve with the depth of the tree on the x-axis and the accuracy on the y-axis.

ANSWER:

FOR MONKS-1:

a) To compute the error (the number of misclassifications) we will run the following code:

```
C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-1.train monks-1.test 1
 None a5 {'0': 62, '1': 62} 1
|        1 a6 {'1': 29} 1
|        2 a4 {'0': 20, '1': 11} 0
|        4 a1 {'0': 23, '1': 11} 0
|        3 a6 {'0': 19, '1': 11} 0
,0,1
0,216,0
1,108,108
```

The command implies that the decision tree has been trained on the monks-1.train data, and later tested on the test data(monks-1.test) at a given depth of 1.

The result is a confusion matrix from which we can calculate the number of misclassifications in the following way:

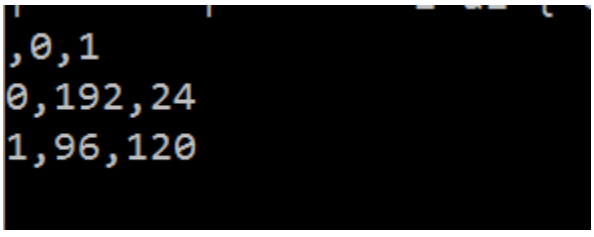
TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 216	FP = 0
ACTUAL: YES	FN = 108	TP = 108

The number of misclassifications at depth 1:

FP+FN = 108

Similarly, if we perform the operation at various depths of 2, 4, 6, 8, 10, 12, 14, 16 we will get the following confusion matrices and the error:

The number of misclassifications at depth 2:

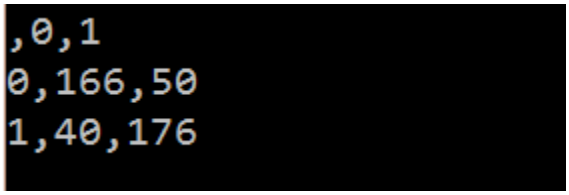


FP+FN = 24 +96 = 120

The number of misclassifications at depth 4:

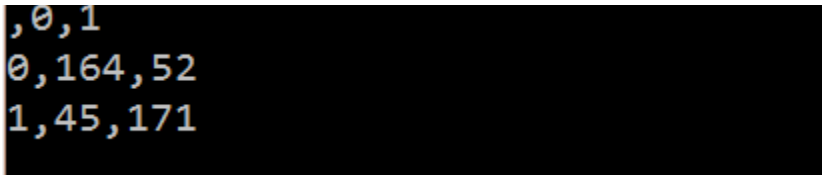
FP +FN = 50+48 = 98

The number of misclassifications at depth 6:



FP + FN= 90

The number of misclassifications at depth 8:

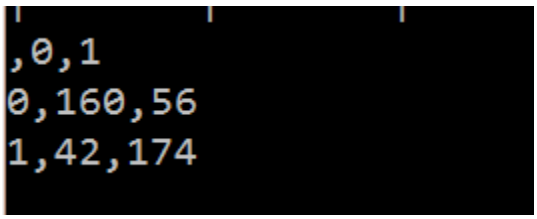


FP +FN = 97

The number of misclassifications at depth 10:

FP + FN = 97

The number of misclassifications at depth 12:



FP + FN = 98

The number of misclassifications at depth 14:

FP + FN = 99

The number of misclassifications at depth 16:

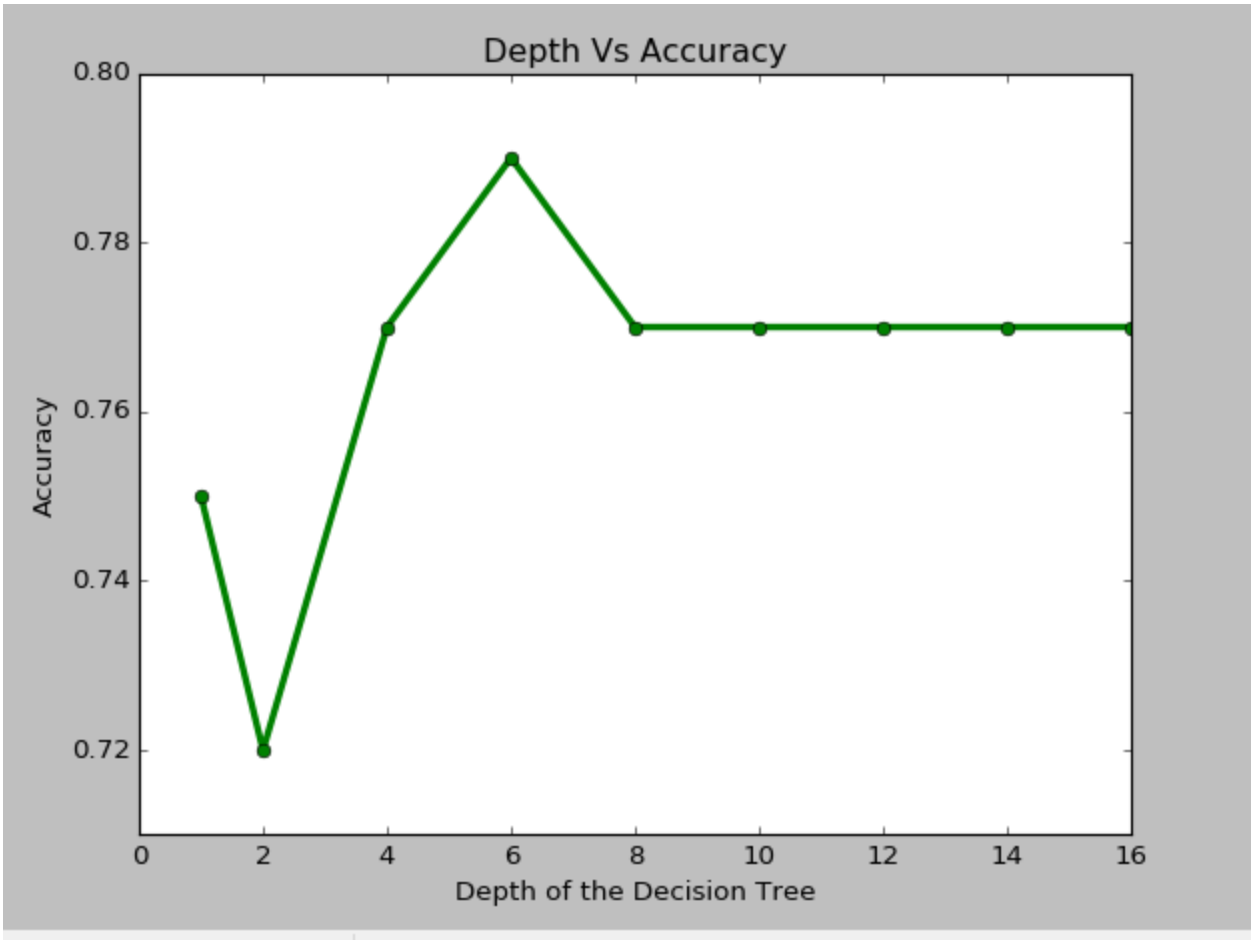
FP + FN = 97

b) To Plot the curve with depth of a tree and accuracy, we need the values of accuracy at different depths which can be calculated similarly from the confusion matrix in the following way:

Accuracy = (TP + TN) / TOTAL POPULATION

DEPTH	ACCURACY
1	(216+108)/432 = .75
2	.72
4	.77
6	.79
8	.77
10	.77
12	.77
14	.77
16	.77

The plot obtained is:



It has been obtained in the following way:

```
import matplotlib.pyplot as plt
depth = [1 , 2 , 4 , 6 , 8 , 10, 12, 14, 16]
accuracy = [.75, .72, .77,.79, .77, .77, .77, .77,.77]
plt.title("Depth Vs Accuracy")
plt.xlabel("Depth of the Decision Tree")
plt.ylabel("Accuracy")
plt.plot(depth, accuracy, color= "g" , linewidth = 3.0 , marker ="o")
plt.show()
```

FOR MONKS-2:

a)If we perform the similar operation on monks-2 dataset at depth 1 we get the following confusion matrix:

```
C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-2.train monks-2.test 1
 None a5 {'0': 105, '1': 64} 0
|        1 a3 {'0': 29, '1': 14} 0
|        2 a3 {'0': 20, '1': 20} 1
|        4 a2 {'0': 26, '1': 11} 0
|        3 a3 {'0': 30, '1': 19} 0
,0,1
0,220,70
1,104,38
```

The number of misclassifications at depth 1:

FP+FN = 174

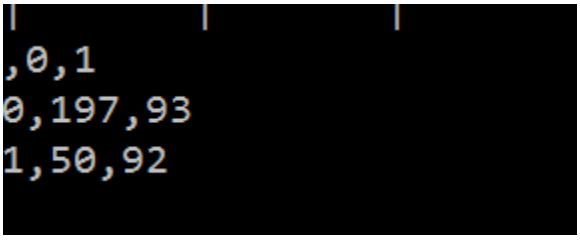
The number of misclassifications at depth 2:

FP +FN = 170

The number of misclassifications at depth 4:

FP +FN = 156

The number of misclassifications at depth 6:

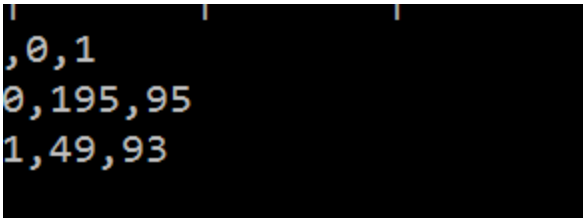


FP +FN = 143

The number of misclassifications at depth 8:

FP +FN = 143

The number of misclassifications at depth 10:

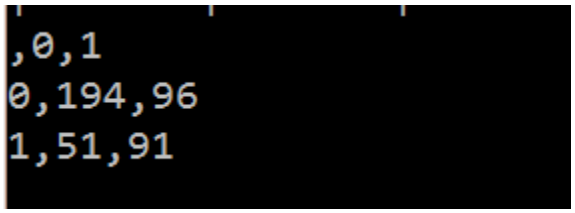


FP +FN = 144

The number of misclassifications at depth 12:

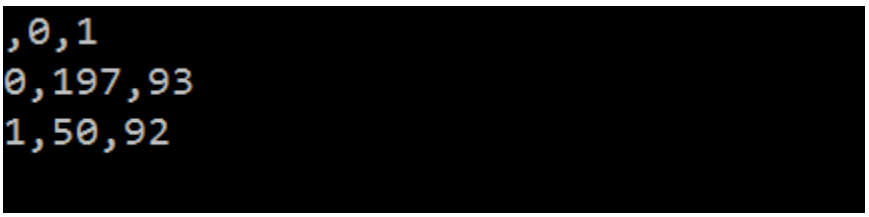
FP +FN = 143

The number of misclassifications at depth 14:



FP + FN = 147

The number of misclassifications at depth 16:



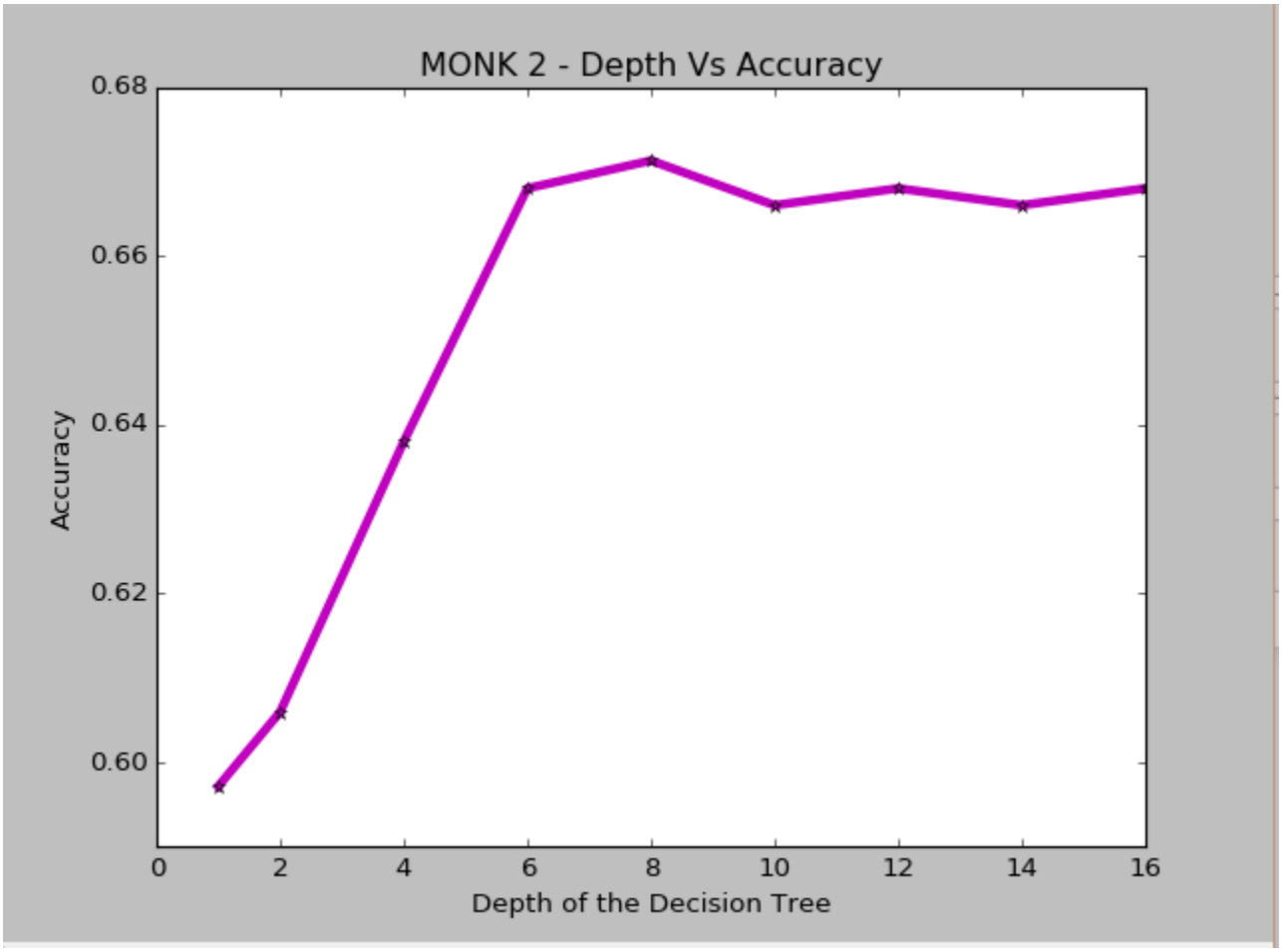
FP +FN = 143

b) To Plot the curve with depth of a tree and accuracy, we need the values of accuracy at different depths which can be calculated similarly from the confusion matrix in the following way:

Accuracy = (TP + TN) / TOTAL POPULATION

DEPTH	ACCURACY
1	(220+38)/ 432=.5972
2	.606
4	.638
6	.668

8	.67129
10	.666
12	.668
14	.666
16	.668



The code for the above plot is:

```
import matplotlib.pyplot as plt
depth = [1 , 2 , 4 , 6 , 8 , 10, 12, 14, 16]
accuracy = [.5972, .606, .638,.668, .67129, .666, .668, .666,.668]
plt.title("MONK 2 - Depth Vs Accuracy")
plt.xlabel("Depth of the Decision Tree")
plt.ylabel("Accuracy")
plt.plot(depth, accuracy, color= "m" , linewidth = 4.0 , marker ="*")
plt.show()
```

FOR MONKS-3:

a)If we perform the similar operation on monks-3 dataset at depth 1 we get the following confusion matrix:

```
C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-3.test 1
 None a2 {'0': 62, '1': 60} 0
|       2 a5 {'0': 11, '1': 31} 1
|       1 a5 {'0': 13, '1': 26} 1
|       3 a4 {'0': 38, '1': 3} 0
,0,1
0,132,72
1,12,216
```

The number of misclassifications at depth 1:

FP +FN = 84

The number of misclassifications at depth 2:

FP +FN = 12

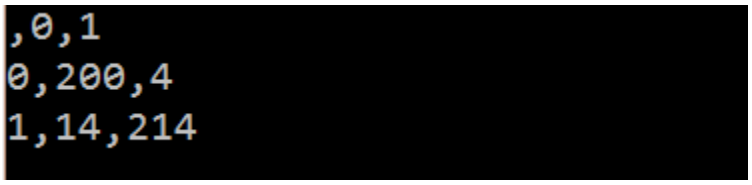
The number of misclassifications at depth 4:

FP +FN = 18

The number of misclassifications at depth 6:

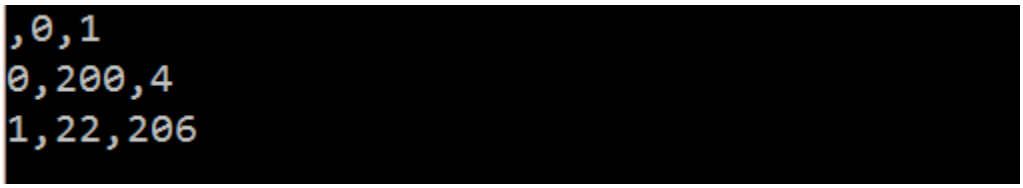
FP +FN = 20

The number of misclassifications at depth 8:



FP + FN = 18

The number of misclassifications at depth 10:



FP +FN = 26

The number of misclassifications at depth 12:

FP + FN = 20

The number of misclassifications at depth 14:

FP + FN = 20

The number of misclassifications at depth 16:

FP +FN = 18

b) To Plot the curve with depth of a tree and accuracy, we need the values of accuracy at different depths which can be calculated similarly from the confusion matrix in the following way:

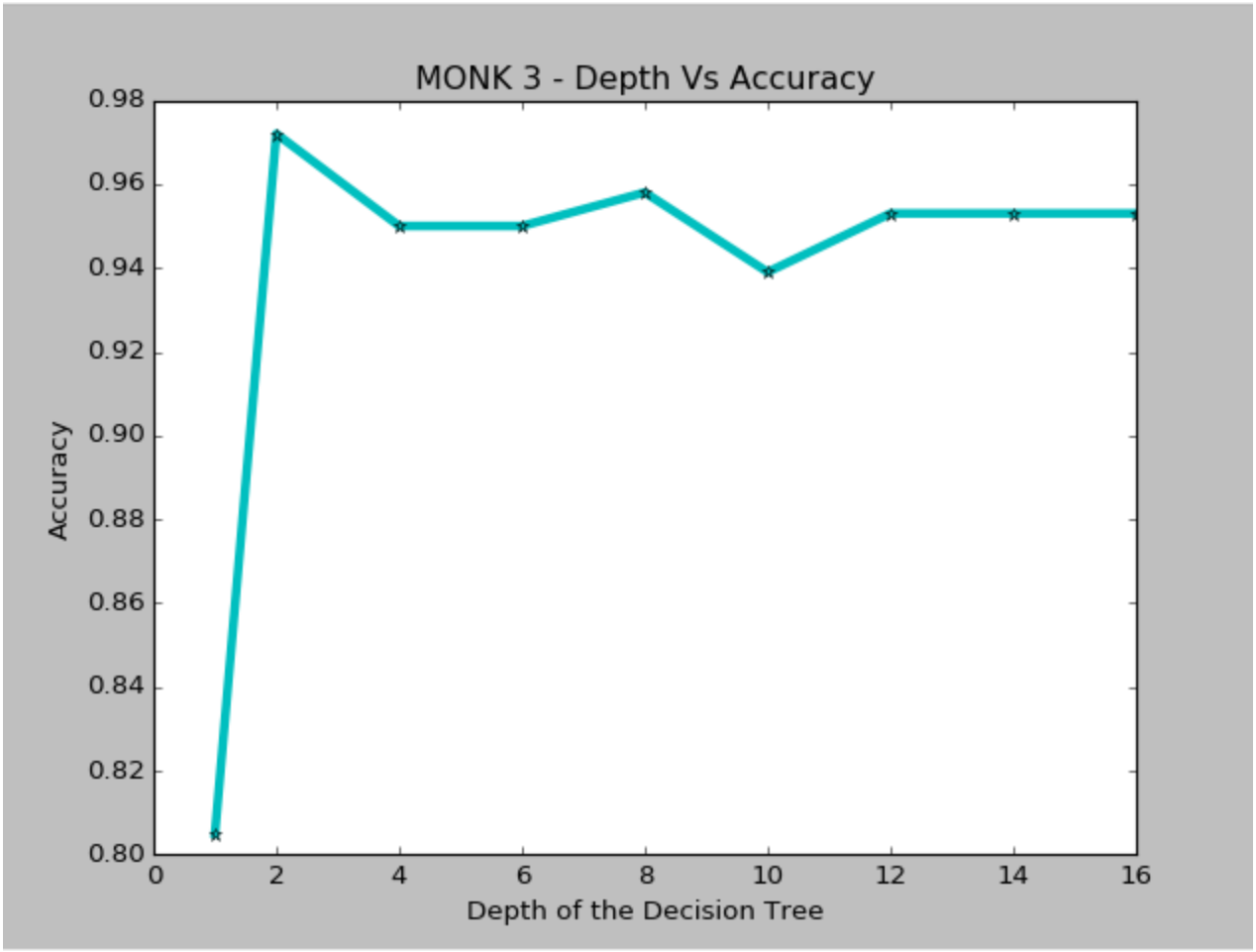
Accuracy = (TP + TN) / TOTAL POPULATION

DEPTH	ACCURACY
1	348/432=.805
2	.972
4	.95
6	.95
8	.958
10	.939
12	.953
14	.953
16	.953

To get the plot between depth of the decision tree and accuracy we will execute the following code:

```
import matplotlib.pyplot as plt
depth = [1 , 2 , 4 , 6 , 8 , 10, 12, 14, 16]
accuracy = [.805, .972, .95,.95, .958, .939, .953, .953,.953]
plt.title("MONK 3 - Depth Vs Accuracy")
plt.xlabel("Depth of the Decision Tree")
plt.ylabel("Accuracy")
plt.plot(depth, accuracy, color= "c" , linewidth = 4.0 , marker ="*")
plt.show()
```

The plot obtained is as follows:



So, the average confusion matrix for depth 1, number of misclassifications and the accuracy are:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 189	FP = 47.3
ACTUAL: YES	FN = 74.66	TP = 120

Number of Misclassifications: FP+FN = 121.96

Accuracy: .71

For depth 2:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 206	FP = 30
ACTUAL: YES	FN = 70	TP = 126.6

Number of Misclassifications: FP+FN = 100

Accuracy: .76

For depth 4:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 194	FP = 42.66
ACTUAL: YES	FN = 48	TP = 147.33

Number of Misclassifications: FP+FN = 90.66

Accuracy: .79

For depth 6:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 187	FP = 49
ACTUAL: YES	FN = 35.3	TP = 160

Number of Misclassifications: FP+FN = 84.3

Accuracy: .805

For depth 8:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 187	FP = 49.6
ACTUAL: YES	FN = 36.3	TP = 159

Number of Misclassifications: FP+FN = 85.9

Accuracy: .80

For depth 10:

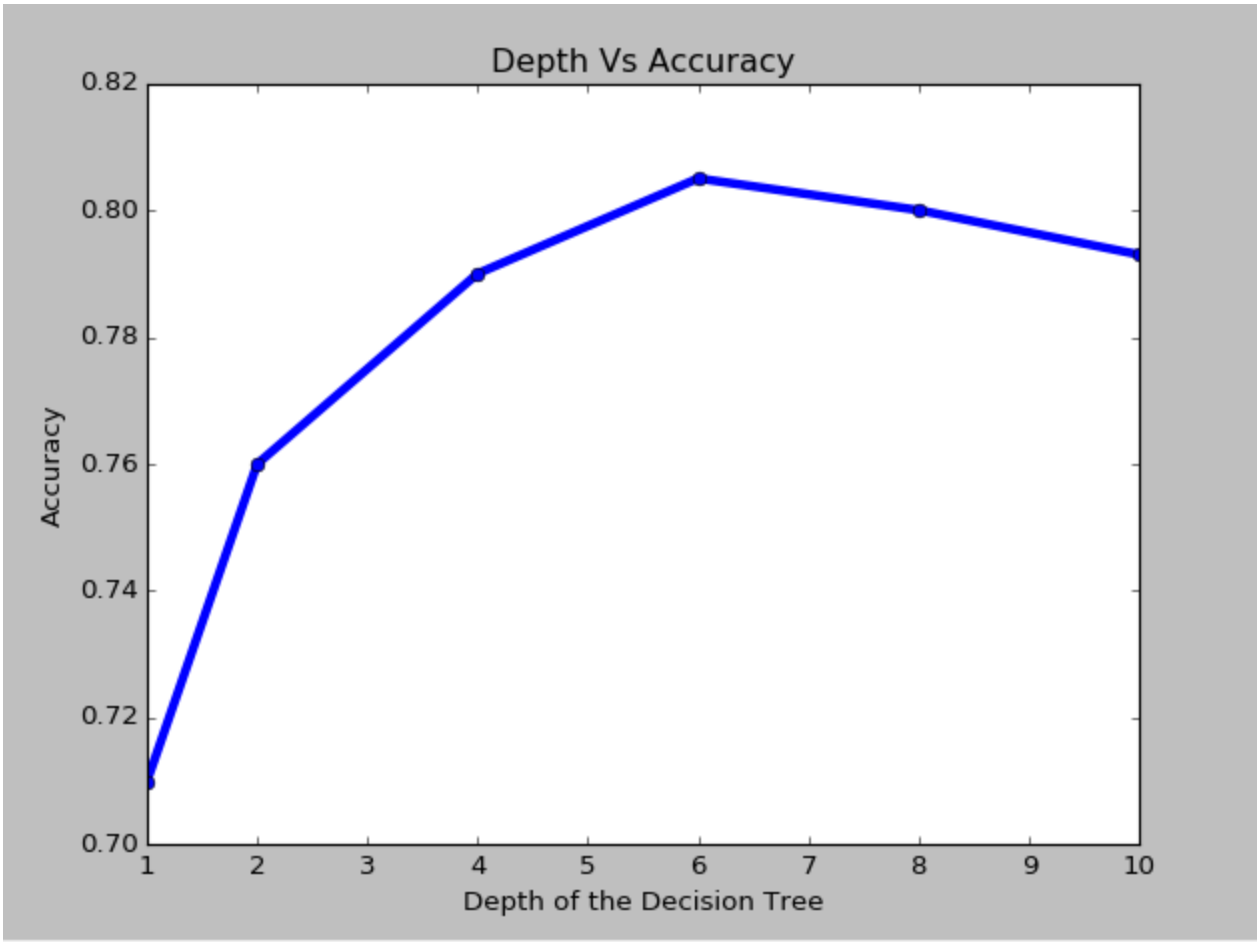
TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 186	FP = 50.3
ACTUAL: YES	FN = 38.66	TP = 156.6

Number of Misclassifications: FP+FN = 88.96

Accuracy: .793

The Plot between the depth of the decision tree and the accuracy will be:

```
import matplotlib.pyplot as plt
depth = [1 , 2 , 4 , 6 , 8 , 10]
accuracy = [.71, .76,.79, .805,.80, .793]
plt.title("Depth Vs Accuracy")
plt.xlabel("Depth of the Decision Tree")
plt.ylabel("Accuracy")
plt.plot(depth, accuracy, color= "b" , linewidth = 4.0 , marker ="o")
plt.show()
```


Depth Vs Accuracy
0.82
0.80
0.78
0.76
0.74
0.72
0.70
Accuracy
1
2
3
4
5
6
7
8
9
10
Depth of the Decision Tree

Report the learned decision tree (depth 1 and depth 2) and report the confusion matrix for these two depths (a confusion matrix has the true label as rows and predicted labels in the columns. Each entry of the matrix is the number of examples. In a binary case, the top left corner is the number of negative examples correctly classified and the bottom right is the number of positives correctly classified).

ANSWER:

For the following problem we will first train the data on Monks-1 train dataset at depths 1 and 2 and then test it on the Monks-1 test, Monks-2 test and Monks-3 test dataset. The confusion matrices obtained from each of the test results will be then used to build the final confusion matrix (which will be the average of the three confusion matrices obtained.)

Same will be done after we train on Monks-2 train dataset and test on Monks-1 test, Monks-2 test and Monks-3 test.

Finally, we will train on Monks-3 train dataset and test on Monks-1 test, Monks-2 test and Monks-test.

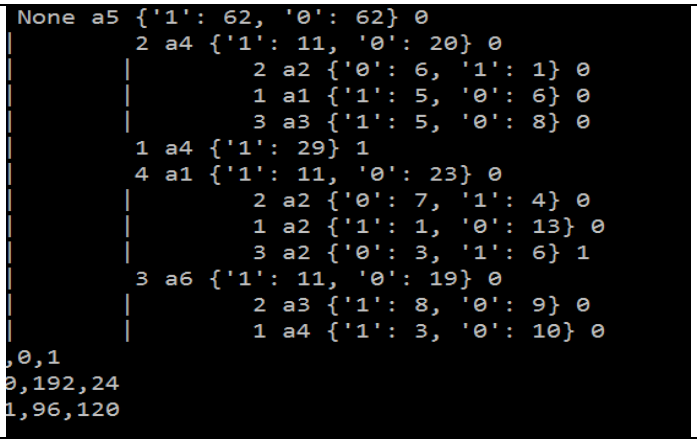
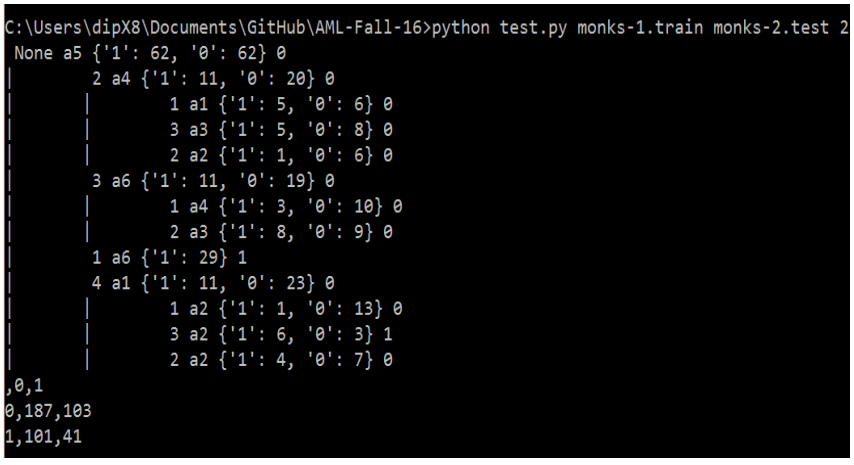
A) DEPTH 1 : MONKS-1 TRAIN

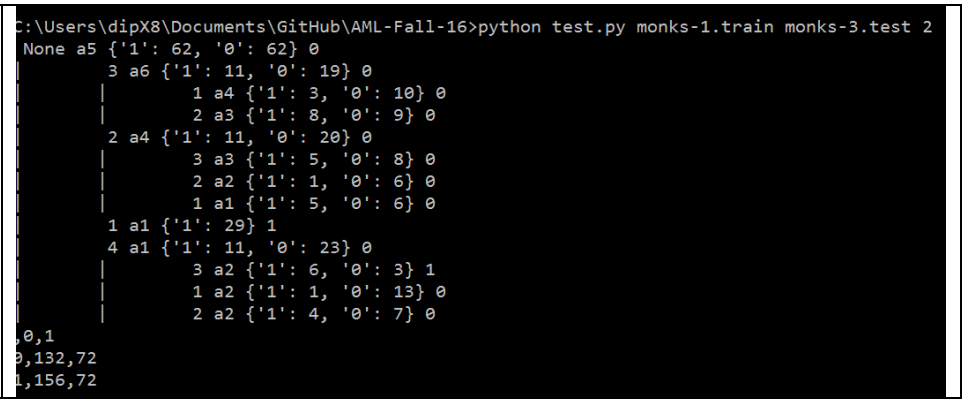
TRAIN	TEST	CONFUSION MATRIX
Monks-1.train	Monks-1.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-1.train monks-1.test 1 None a5 {'0': 62, '1': 62} 1 | 1 a6 {'1': 29} 1 | 2 a4 {'0': 20, '1': 11} 0 | 4 a1 {'0': 23, '1': 11} 0 | 3 a6 {'0': 19, '1': 11} 0 ,0,1 0,216,0 1,108,108
Monks-1.train	Monks-2.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-1.train monks-2.test 1 None a5 {'1': 62, '0': 62} 0 | 3 a6 {'1': 11, '0': 19} 0 | 4 a1 {'1': 11, '0': 23} 0 | 1 a4 {'1': 29} 1 | 2 a4 {'1': 11, '0': 20} 0 ,0,1 0,210,80 1,114,28
Monks-1.train	Monks-3.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-1.train monks-3.test 1 None a5 {'1': 62, '0': 62} 0 | 1 a6 {'1': 29} 1 | 3 a6 {'1': 11, '0': 19} 0 | 2 a4 {'1': 11, '0': 20} 0 | 4 a1 {'1': 11, '0': 23} 0 ,0,1 0,168,36 1,156,72

The final confusion matrix when the model is trained on monks-1 training dataset at depth 1 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 198	FP = 38.6
ACTUAL: YES	FN = 126	TP = 69.3

B) DEPTH 2 :MONKS-1 TRAIN

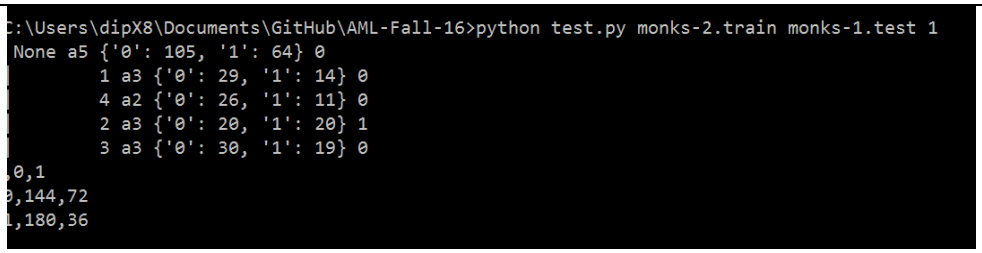
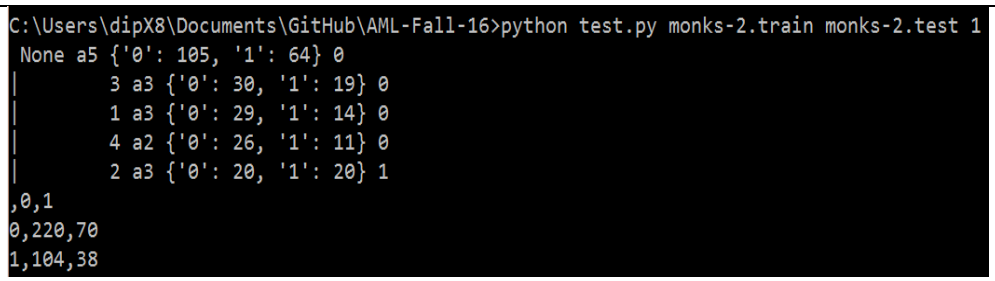
TRAIN	TEST	CONFUSION MATRIX
Monks-1.train	Monks-1.test	
Monks-1.train	Monks-2.test	

Monks-1.train	Monks-3.test	

The final confusion matrix when the model is trained on monks-1 training dataset at depth 2 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 170	FP = 66.3
ACTUAL: YES	FN = 117	TP = 77.6

A) DEPTH 1 : MONKS-2 TRAIN

TRAIN	TEST	CONFUSION MATRIX
Monks-2.train	Monks-1.test	
Monks-2.train	Monks-2.test	

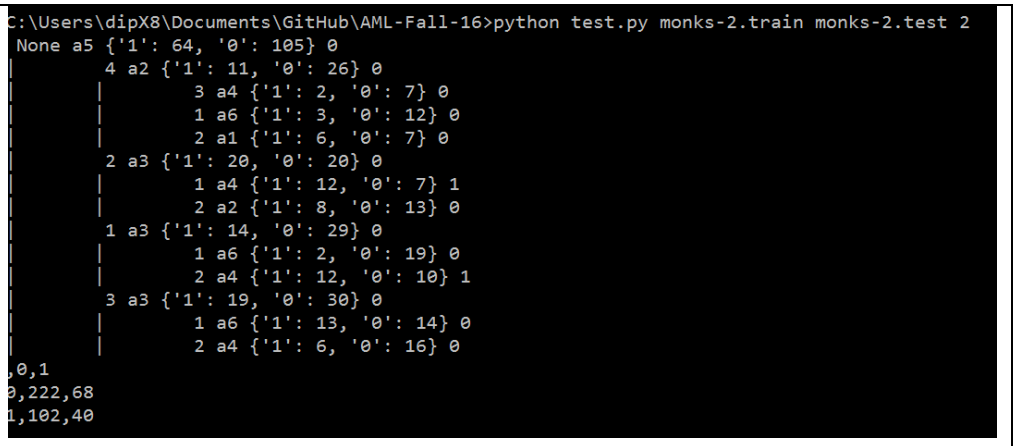
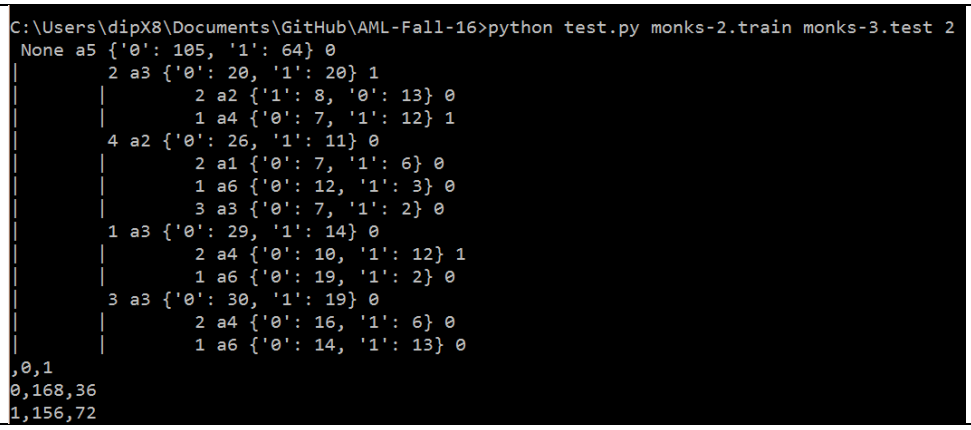
Monks-2.train	Monks-3.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-2.train monks-3.test 1 None a5 {'0': 105, '1': 64} 0 | 3 a3 {'0': 30, '1': 19} 0 | 1 a3 {'0': 29, '1': 14} 0 | 2 a3 {'0': 20, '1': 20} 1 | 4 a2 {'0': 26, '1': 11} 0 ,0,1 0,168,36 1,156,72

The final confusion matrix when the model is trained on monks-2 training dataset at depth 1 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 177	FP = 59.3
ACTUAL: YES	FN = 146.6	TP = 48.66

B) DEPTH 2 :MONKS-2 TRAIN

TRAIN	TEST	CONFUSION MATRIX
Monks-2.train	Monks-1.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-2.train monks-1.test 2 None a5 {'1': 64, '0': 105} 0 | 1 a3 {'1': 14, '0': 29} 0 | | 1 a6 {'1': 2, '0': 19} 0 | | 2 a4 {'1': 12, '0': 10} 1 | 3 a3 {'1': 19, '0': 30} 0 | | 1 a6 {'1': 13, '0': 14} 0 | | 2 a4 {'1': 6, '0': 16} 0 | 4 a2 {'1': 11, '0': 26} 0 | | 1 a6 {'1': 3, '0': 12} 0 | | 3 a3 {'1': 2, '0': 7} 0 | | 2 a1 {'1': 6, '0': 7} 0 | 2 a3 {'1': 20, '0': 20} 0 | | 1 a4 {'1': 12, '0': 7} 1 | | 2 a2 {'1': 8, '0': 13} 0 ,0,1 0,180,36 1,144,72

Monks-2.train	Monks-2.test	
Monks-2.train	Monks-3.test	

The final confusion matrix when the model is trained on monks-1 training dataset at depth 2 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 190	FP =46.6
ACTUAL: YES	FN = 134	TP =61.33

A) DEPTH 1 : MONKS-3 TRAIN

TRAIN	TEST	CONFUSION MATRIX
Monks -3.train	Monks -1.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-1.test 1 None a2 {'0': 62, '1': 60} 0 | 2 a5 {'0': 11, '1': 31} 1 | 3 a4 {'0': 38, '1': 3} 0 | 1 a5 {'0': 13, '1': 26} 1 ,0,1 0,72,144 1,72,144
Monks -3.train	Monks -2.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-2.test 1 None a2 {'1': 60, '0': 62} 0 | 1 a5 {'1': 26, '0': 13} 1 | 2 a5 {'1': 31, '0': 11} 1 | 3 a4 {'1': 3, '0': 38} 0 ,0,1 0,93,197 1,51,91
Monks -3.train	Monks -3.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-3.test 1 None a2 {'0': 62, '1': 60} 0 | 2 a5 {'0': 11, '1': 31} 1 | 3 a4 {'0': 38, '1': 3} 0 | 1 a5 {'0': 13, '1': 26} 1 ,0,1 0,132,72 1,12,216

The final confusion matrix when the model is trained on monks-3 training dataset at depth 1 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 99	FP = 137.66
ACTUAL: YES	FN = 45	TP = 150.33

B) DEPTH 2 :MONKS-3 TRAIN

TRAIN	TEST	CONFUSION MATRIX
Monks-3.train	Monks-1.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-1.test 2 None a2 {'0': 62, '1': 60} 0 | 3 a4 {'0': 38, '1': 3} 0 | | 3 a1 {'0': 14} 0 | | 1 a5 {'0': 11, '1': 3} 0 | | 2 a1 {'0': 13} 0 | 1 a5 {'0': 13, '1': 26} 1 | | 3 a4 {'0': 1, '1': 5} 1 | | 4 a1 {'0': 12} 0 | | 1 a1 {'1': 12} 1 | | 2 a1 {'1': 9} 1 | 2 a5 {'0': 11, '1': 31} 1 | | 3 a3 {'0': 3, '1': 9} 1 | | 4 a1 {'0': 7} 0 | | 1 a1 {'1': 10} 1 | | 2 a1 {'0': 1, '1': 12} 1 ,0,1 0,120,96 1,96,120
Monks-3.train	Monks-2.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-2.test 2 None a2 {'1': 60, '0': 62} 0 | 2 a5 {'1': 31, '0': 11} 1 | | 2 a1 {'1': 12, '0': 1} 1 | | 1 a1 {'1': 10} 1 | | 4 a1 {'0': 7} 0 | | 3 a3 {'1': 9, '0': 3} 1 | 3 a4 {'1': 3, '0': 38} 0 | | 2 a1 {'0': 13} 0 | | 3 a1 {'0': 14} 0 | | 1 a5 {'1': 3, '0': 11} 0 | 1 a5 {'1': 26, '0': 13} 1 | | 2 a1 {'1': 9} 1 | | 3 a4 {'1': 5, '0': 1} 1 | | 4 a1 {'0': 12} 0 | | 1 a1 {'1': 12} 1 ,0,1 0,140,150 1,76,66
Monks-3.train	Monks-3.test	C:\Users\dipX8\Documents\GitHub\AML-Fall-16>python test.py monks-3.train monks-3.test 2 None a2 {'1': 60, '0': 62} 0 | 1 a5 {'1': 26, '0': 13} 1 | | 4 a6 {'0': 12} 0 | | 2 a6 {'1': 9} 1 | | 3 a4 {'1': 5, '0': 1} 1 | | 1 a6 {'1': 12} 1 | 2 a5 {'1': 31, '0': 11} 1 | | 4 a6 {'0': 7} 0 | | 3 a3 {'1': 9, '0': 3} 1 | | 1 a6 {'1': 10} 1 | | 2 a1 {'1': 12, '0': 1} 1 | 3 a4 {'1': 3, '0': 38} 0 | | 1 a5 {'1': 3, '0': 11} 0 | | 2 a5 {'0': 13} 0 | | 3 a5 {'0': 14} 0 ,0,1 0,204,0 1,12,216

The final confusion matrix when the model is trained on monks-3 training dataset at depth 2 is:

TOTAL POPULATION 432	PRED.: NO	PRED.: YES
ACTUAL : NO	TN = 154.66	FP =82
ACTUAL: YES	FN = 61.33	TP =134

3. Now, use Weka's default decision tree (J48) algorithm on this training set to learn a decision tree. Report the tree and the confusion matrix on the test set. Do not change the default parameters of Weka

In Weka, the Monks dataset was discretized, i.e the numbers are made as levels before running the dataset to ensure the uniformity of the decision tree.

Train: Monks-1.train Test: Monks2.test

```
a5 = 1: 1 (29.0)
a5 = 2: 0 (31.0/11.0)
a5 = 3
|   a6 = 1: 0 (13.0/3.0)
|   a6 = 2
|   |   a3 = 1: 1 (7.0/2.0)
|   |   a3 = 2: 0 (10.0/3.0)
a5 = 4
|   a1 = 1: 0 (14.0/1.0)
|   a1 = 2
|   |   a2 = 1: 0 (6.0)
|   |   a2 = 2: 1 (4.0)
|   |   a2 = 3: 0 (1.0)
|   a1 = 3
|   |   a2 = 1: 1 (0.0)
|   |   a2 = 2: 0 (3.0)
|   |   a2 = 3: 1 (6.0)


=== Confusion Matrix ===

   a   b   <-- classified as
 172 118 |   a = 0
  89  53 |   b = 1
```

Train: Monks-1.train Test: Monks3.test

Tree as above, Confusion Matrix as below.

```
=== Confusion Matrix ===

   a   b   <-- classified as
 126  78 |   a = 0
 135  93 |   b = 1
```

Train: Monks-2.train Test: Monks1.test

```
a4 = 1: 0 (54.0/15.0)
a4 = 2
|   a5 = 1
|   |   a3 = 1: 0 (7.0/1.0)
|   |   a3 = 2: 1 (5.0)
|   a5 = 2
|   |   a3 = 1
|   |   |   a6 = 1: 0 (3.0/1.0)
|   |   |   a6 = 2: 1 (4.0)
|   |   a3 = 2
|   |   |   a2 = 1: 1 (2.0)
|   |   |   a2 = 2: 0 (3.0)
|   |   |   a2 = 3: 0 (2.0)
|   a5 = 3: 0 (17.0/6.0)
|   a5 = 4: 0 (11.0/3.0)
a4 = 3
|   a3 = 1
|   |   a5 = 1: 0 (7.0/1.0)
|   |   a5 = 2: 1 (7.0/1.0)
|   |   a5 = 3: 1 (9.0/4.0)
|   |   a5 = 4
|   |   |   a2 = 1: 0 (2.0)
|   |   |   a2 = 2: 1 (3.0/1.0)
|   |   |   a2 = 3: 1 (2.0)
|   a3 = 2
|   |   a6 = 1
|   |   |   a1 = 1: 1 (4.0/1.0)
|   |   |   a1 = 2: 0 (4.0/1.0)
|   |   |   a1 = 3: 1 (4.0/1.0)
|   |   a6 = 2: 0 (19.0/4.0)


=== Confusion Matrix ===

   a   b   <-- classified as
 162  54 |   a = 0
 165  51 |   b = 1
```

Monks-2 train vs Monks 3 test

```
=== Confusion Matrix ===

   a   b   <-- classified as
 159  45 |   a = 0
 168  60 |   b = 1
```

Same dtree as above.

Monks-3 train vs Monks 1 test

```
a2 = 1
|   a5 = 1: 1 (12.0)
|   a5 = 2: 1 (9.0)
|   a5 = 3: 1 (6.0/1.0)
|   a5 = 4: 0 (12.0)
a2 = 2
|   a5 = 1: 1 (10.0)
|   a5 = 2: 1 (13.0/1.0)
|   a5 = 3: 1 (12.0/3.0)
|   a5 = 4: 0 (7.0)
a2 = 3: 0 (41.0/3.0)
```

```
  a   b   <-- classified as
120  96 |   a = 0
 96 120 |   b = 1
```

Monks-3 train vs Monks 2 test

Same dree as above

```
=== Confusion Matrix ===

   a   b   <-- classified as
 140 150 |   a = 0
  76  66 |   b = 1
```

4. In Own Dataset

The dataset is a customer churn dataset with a binary outcome yes or no.

The data is split into training and test at 70% as training and 30% as test.

The following is the output of the program

Depth : 1

```
C:\Users\Ganesh\Documents\GitHub\AML>python churn.py churn_dataset_discretize.csv 2
Test - Train Split Ratio 0.3
# of Rows in Original Dataset 1000
# of Rows in Train Dataset 700
# of Rows in Test Dataset 300
 None Contract {'Yes': 344, 'No': 356} No
|        Month-to-month InternetService {'Yes': 311, 'No': 135} Yes
|       |        Fiber optic tenure {'Yes': 215, 'No': 53} Yes
|       |        DSL tenure {'Yes': 78, 'No': 53} Yes
|       |        No tenure {'Yes': 18, 'No': 29} No
|        One year StreamingMovies {'Yes': 24, 'No': 103} No
|       |        No tenure {'Yes': 2, 'No': 30} No
|       |        Yes OnlineBackup {'Yes': 20, 'No': 43} No
|       |        No internet service PaymentMethod {'Yes': 2, 'No': 30} No
|        Two year tenure {'Yes': 9, 'No': 118} No
|       |        (22.5-66.5] SeniorCitizen {'Yes': 8, 'No': 47} No
|       |        (-inf-9.5] StreamingTV {'No': 2} No
|       |        (66.5-inf) PaymentMethod {'Yes': 1, 'No': 63} No
|       |        (9.5-22.5] StreamingTV {'No': 6} No
,No,Yes
No,90,54
Yes,31,125
```

Depth : 2

```
C:\Users\Ganesh\Documents\GitHub\AML>python churn.py churn_dataset_discretize.csv 3
Test - Train Split Ratio 0.3
# of Rows in Original Dataset 1000
# of Rows in Train Dataset 700
# of Rows in Test Dataset 300
```

```
None Contract {'Yes': 354, 'No': 346} Yes
|        Month-to-month InternetService {'Yes': 314, 'No': 139} Yes
|        |        Fiber optic tenure {'Yes': 216, 'No': 58} Yes
|        |        |        (-inf-9.5] PaymentMethod {'Yes': 106, 'No': 13} Yes
|        |        |        (9.5-22.5] PaperlessBilling {'Yes': 45, 'No': 14} Yes
|        |        |        (22.5-66.5] SeniorCitizen {'Yes': 64, 'No': 29} Yes
|        |        |        (66.5-inf) StreamingTV {'Yes': 1, 'No': 2} No
|        |        No tenure {'Yes': 19, 'No': 32} No
|        |        |        (-inf-9.5] PaperlessBilling {'Yes': 16, 'No': 19} No
|        |        |        (9.5-22.5] Partner {'Yes': 2, 'No': 8} No
|        |        |        (22.5-66.5] Dependents {'Yes': 1, 'No': 5} No
|        |        DSL tenure {'Yes': 79, 'No': 49} Yes
|        |        |        (-inf-9.5] PaymentMethod {'Yes': 57, 'No': 13} Yes
|        |        |        (9.5-22.5] PaymentMethod {'Yes': 14, 'No': 18} No
|        |        |        (22.5-66.5] StreamingMovies {'Yes': 8, 'No': 17} No
|        |        |        (66.5-inf) PhoneService {'No': 1} No
|        Two year PaymentMethod {'Yes': 8, 'No': 110} No
|        |        Electronic check StreamingMovies {'Yes': 4, 'No': 8} No
|        |        |        Yes Partner {'Yes': 3, 'No': 6} No
|        |        |        No PhoneService {'Yes': 1} Yes
|        |        |        No internet service PhoneService {'No': 2} No
|        |        Credit card (automatic) StreamingMovies {'Yes': 3, 'No': 39} No
|        |        |        Yes OnlineBackup {'Yes': 1, 'No': 20} No
|        |        |        No Partner {'Yes': 2, 'No': 3} No
|        |        |        No internet service PhoneService {'No': 16} No
|        |        Mailed check PhoneService {'No': 28} No
|        |        Bank transfer (automatic) SeniorCitizen {'Yes': 1, 'No': 35} No
|        |        |        (0.5-inf) OnlineSecurity {'Yes': 1, 'No': 1} No
|        |        |        (-inf-0.5] PhoneService {'No': 34} No
|        One year StreamingTV {'Yes': 32, 'No': 97} No
|        |        Yes tenure {'Yes': 23, 'No': 36} No
|        |        |        (9.5-22.5] MultipleLines {'Yes': 7, 'No': 4} Yes
|        |        |        (22.5-66.5] StreamingMovies {'Yes': 15, 'No': 28} No
|        |        |        (66.5-inf) OnlineSecurity {'Yes': 1, 'No': 4} No
|        |        No tenure {'Yes': 5, 'No': 37} No
|        |        |        (-inf-9.5] PhoneService {'Yes': 1} Yes
|        |        |        (9.5-22.5] OnlineBackup {'Yes': 2, 'No': 5} No
|        |        |        (22.5-66.5] PaymentMethod {'Yes': 1, 'No': 30} No
|        |        |        (66.5-inf) PaperlessBilling {'Yes': 1, 'No': 2} No
|        |        No internet service PaymentMethod {'Yes': 4, 'No': 24} No
|        |        |        Electronic check tenure {'Yes': 1, 'No': 3} No
|        |        |        Mailed check PhoneService {'No': 9} No
|        |        |        Bank transfer (automatic) PaperlessBilling {'Yes': 3, 'No': 6} No
```

```
,No,Yes
No,117,37
Yes,31,115
```

<u>Weka Output:</u>

```
Contract = Month-to-month
|  InternetService = Fiber optic
|  |  TotalCharges = '(-inf-375.15]': Yes (121.0/12.0)
|  |  TotalCharges = '(375.15-inf)'
|  |  |  SeniorCitizen = '(-inf-0.5]'
|  |  |  |  PaperlessBilling = Yes
|  |  |  |  |  tenure = '(-inf-9.5]': Yes (27.0/5.0)
|  |  |  |  |  tenure = '(9.5-22.5]': Yes (51.0/7.0)
|  |  |  |  |  tenure = '(22.5-66.5]'
|  |  |  |  |  |  PaymentMethod = Electronic check
|  |  |  |  |  |  |  DeviceProtection = Yes
```

```
|  |  |  |  |  |  |  |  MultipleLines = Yes
|  |  |  |  |  |  |  |  |  Dependents = No
|  |  |  |  |  |  |  |  |  |  OnlineBackup = No: No (5.0/1.0)
|  |  |  |  |  |  |  |  |  |  OnlineBackup = No internet service: Yes (0.0)
|  |  |  |  |  |  |  |  |  |  OnlineBackup = Yes: Yes (10.0/2.0)
|  |  |  |  |  |  |  |  |  Dependents = Yes: Yes (5.0)
|  |  |  |  |  |  |  |  MultipleLines = No phone service: Yes (0.0)
|  |  |  |  |  |  |  |  MultipleLines = No: No (6.0/1.0)
|  |  |  |  |  |  |  DeviceProtection = No: Yes (17.0/2.0)
|  |  |  |  |  |  |  DeviceProtection = No internet service: Yes (0.0)
|  |  |  |  |  |  PaymentMethod = Mailed check
|  |  |  |  |  |  |  gender = Female: Yes (2.0)
|  |  |  |  |  |  |  gender = Male: No (2.0)
|  |  |  |  |  |  PaymentMethod = Credit card (automatic)
|  |  |  |  |  |  |  DeviceProtection = Yes: Yes (2.0)
|  |  |  |  |  |  |  DeviceProtection = No: No (8.0/2.0)
|  |  |  |  |  |  |  DeviceProtection = No internet service: No (0.0)
|  |  |  |  |  |  PaymentMethod = Bank transfer (automatic)
|  |  |  |  |  |  |  OnlineSecurity = No
|  |  |  |  |  |  |  |  Dependents = No: Yes (7.0/2.0)
|  |  |  |  |  |  |  |  Dependents = Yes
|  |  |  |  |  |  |  |  |  StreamingMovies = Yes: No (3.0)
|  |  |  |  |  |  |  |  |  StreamingMovies = No: Yes (3.0)
|  |  |  |  |  |  |  |  |  StreamingMovies = No internet service: Yes (0.0)
|  |  |  |  |  |  |  OnlineSecurity = No internet service: No (0.0)
|  |  |  |  |  |  |  OnlineSecurity = Yes: No (4.0)
|  |  |  |  |  tenure = '(66.5-inf)': No (1.0)
|  |  |  |  PaperlessBilling = No
|  |  |  |  |  OnlineBackup = No
|  |  |  |  |  |  MultipleLines = Yes: Yes (12.0/3.0)
|  |  |  |  |  |  MultipleLines = No phone service: Yes (0.0)
|  |  |  |  |  |  MultipleLines = No: No (11.0/4.0)
|  |  |  |  |  OnlineBackup = No internet service: No (0.0)
|  |  |  |  |  OnlineBackup = Yes: No (6.0/1.0)
|  |  |  SeniorCitizen = '(0.5-inf)': Yes (84.0/13.0)
|  InternetService = DSL
|  |  tenure = '(-inf-9.5]': Yes (104.0/22.0)
|  |  tenure = '(9.5-22.5]'
|  |  |  OnlineBackup = No
|  |  |  |  Dependents = No
|  |  |  |  |  PaymentMethod = Electronic check: Yes (13.0/3.0)
|  |  |  |  |  PaymentMethod = Mailed check: Yes (3.0)
|  |  |  |  |  PaymentMethod = Credit card (automatic)
|  |  |  |  |  |  gender = Female: Yes (3.0)
|  |  |  |  |  |  gender = Male
|  |  |  |  |  |  |  OnlineSecurity = No: Yes (3.0/1.0)
|  |  |  |  |  |  |  OnlineSecurity = No internet service: No (0.0)
|  |  |  |  |  |  |  OnlineSecurity = Yes: No (2.0)
|  |  |  |  |  PaymentMethod = Bank transfer (automatic): No (3.0)
|  |  |  |  Dependents = Yes: No (7.0/1.0)
|  |  |  OnlineBackup = No internet service: No (0.0)
|  |  |  OnlineBackup = Yes: No (11.0/1.0)
```

```
|  |  tenure = '(22.5-66.5]': No (41.0/12.0)
|  |  tenure = '(66.5-inf)': No (1.0)
|  InternetService = No
|  |  tenure = '(-inf-9.5]'
|  |  |  Dependents = No
|  |  |  |  PaperlessBilling = Yes: Yes (12.0/3.0)
|  |  |  |  PaperlessBilling = No
|  |  |  |  |  Partner = No: No (27.0/11.0)
|  |  |  |  |  Partner = Yes: Yes (2.0)
|  |  |  Dependents = Yes
|  |  |  |  gender = Female: Yes (4.0/1.0)
|  |  |  |  gender = Male: No (6.0)
|  |  tenure = '(9.5-22.5]': No (11.0/2.0)
|  |  tenure = '(22.5-66.5]': No (8.0/1.0)
|  |  tenure = '(66.5-inf)': No (0.0)
Contract = One year
|  StreamingTV = No
|  |  TotalCharges = '(-inf-375.15]': Yes (2.0)
|  |  TotalCharges = '(375.15-inf)': No (56.0/6.0)
|  StreamingTV = Yes
|  |  OnlineSecurity = No: No (48.0/15.0)
|  |  OnlineSecurity = No internet service: No (0.0)
|  |  OnlineSecurity = Yes
|  |  |  PaymentMethod = Electronic check
|  |  |  |  Partner = No: No (3.0/1.0)
|  |  |  |  Partner = Yes: Yes (5.0/1.0)
|  |  |  PaymentMethod = Mailed check: Yes (3.0)
|  |  |  PaymentMethod = Credit card (automatic): No (6.0/1.0)
|  |  |  PaymentMethod = Bank transfer (automatic)
|  |  |  |  OnlineBackup = No: No (4.0/1.0)
|  |  |  |  OnlineBackup = No internet service: Yes (0.0)
|  |  |  |  OnlineBackup = Yes: Yes (5.0/1.0)
|  StreamingTV = No internet service: No (41.0/4.0)
Contract = Two year: No (179.0/13.0)
```

```
=== Confusion Matrix ===

   a   b   <-- classified as
 110  29 |   a = Yes
  40 121 |   b = No
```