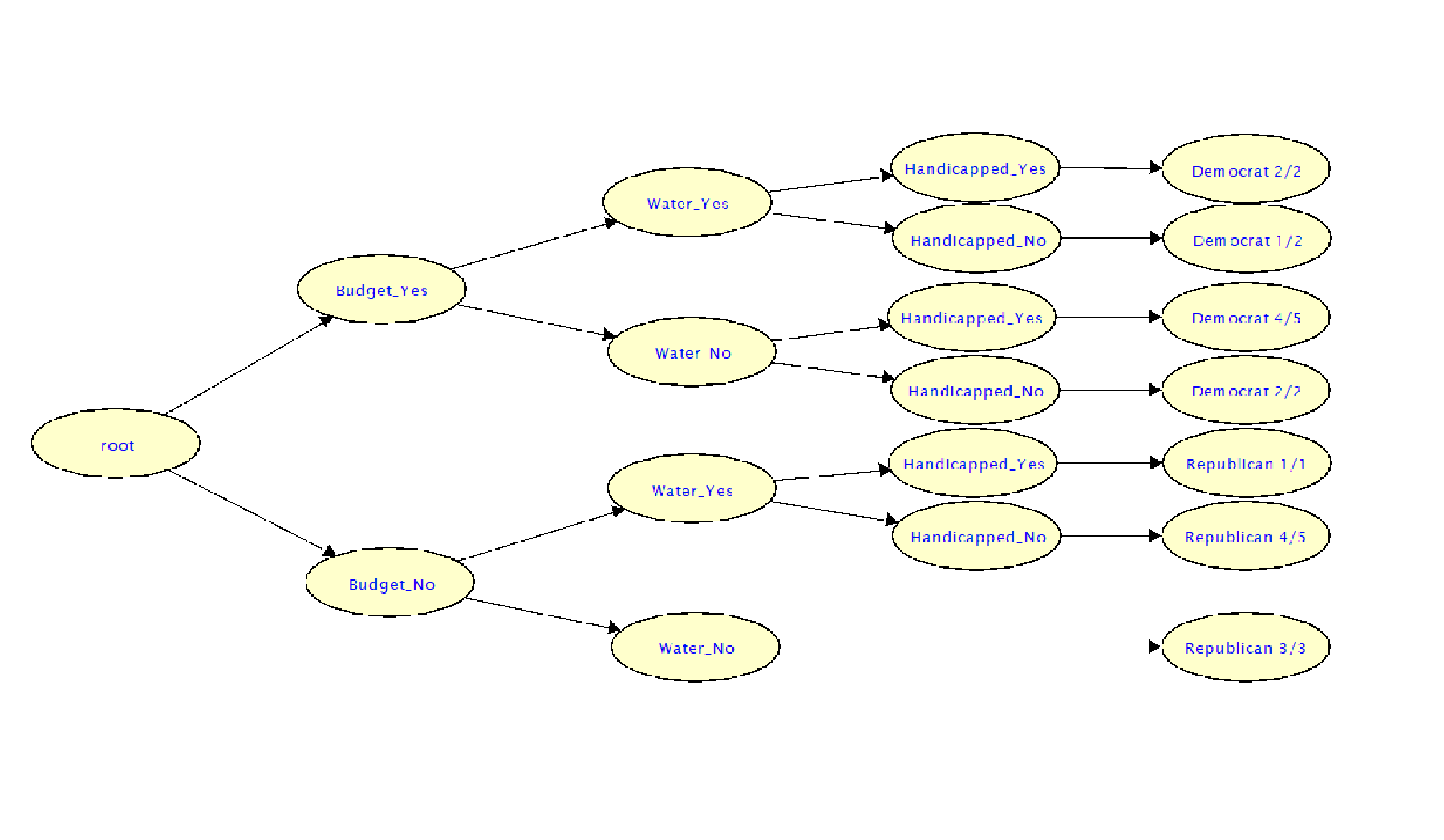
Michael Wang

CS 145 HW#2

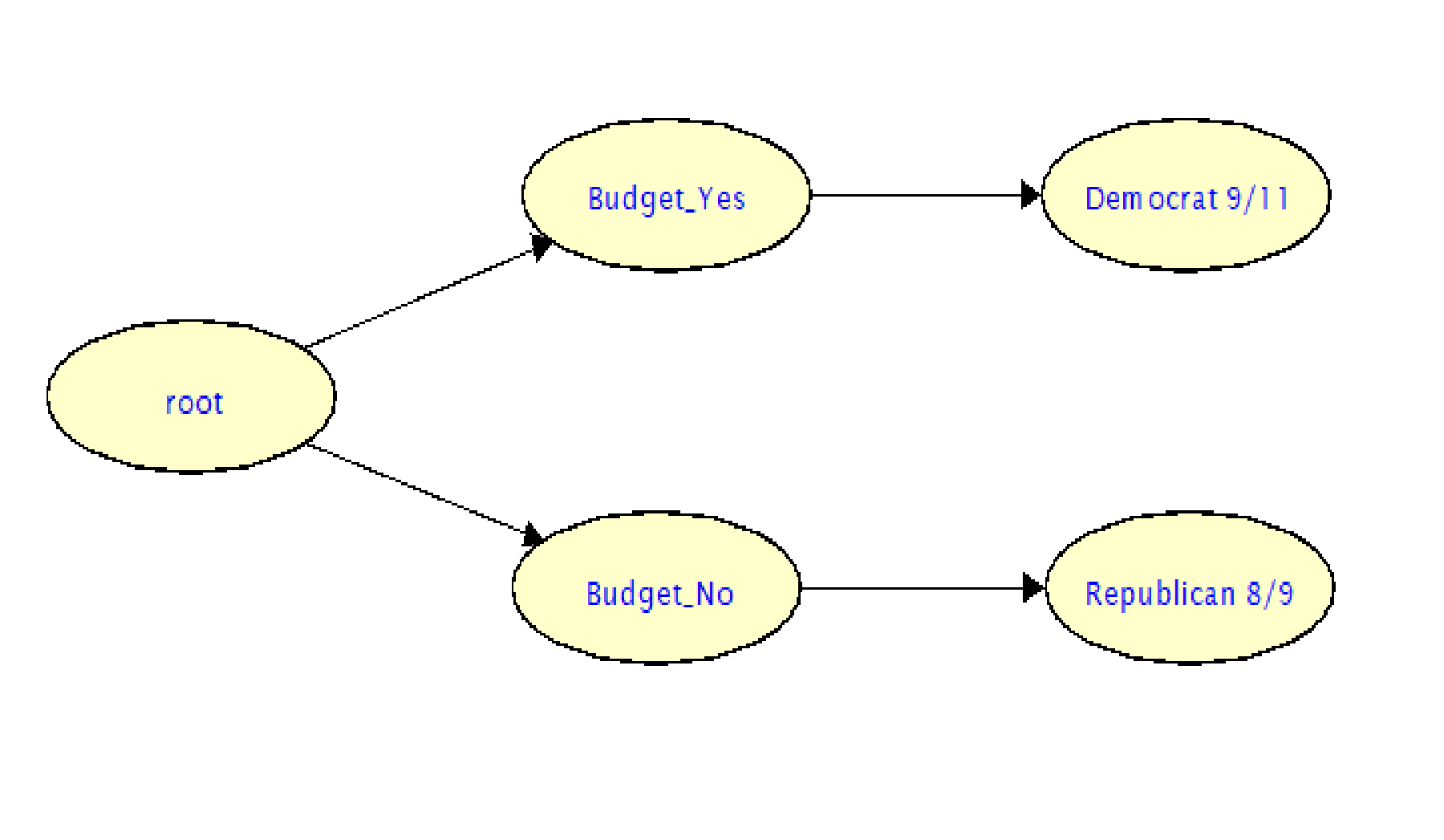
1.1) To find the attribute that results in the greatest information gain, it suffices to find the attribute with the lowest entropy:

Budget has the lowest entropy, so we branch the data along Y and N for that attribute. We now repeat the above for each of the two branches.

The ”yes-budget” branch has the same information gain for either attribute and the “no-budget” branch has a better information gain for the water attribute. I’ll branch the data on water for both budget branches for my convenience. We now have four branches and only one attribute remaining for each. We conclude by simply counting which party is most represented by a yes or no for handicapped infants and use that party as the leaf node. The resulting tree looks like:



If we take the Democratic ½ leaf as being Democratic, then we can simplify the tree to:



So much for all that work.

1.2)

Running the program for both algorithms produces

Attribute Selection Criterion: 0

best\_feature is: legs

best\_feature is: fins

best\_feature is: toothed

best\_feature is: eggs

best\_feature is: hair

best\_feature is: hair

best\_feature is: toothed

best\_feature is: aquatic

{'legs': {0: {'fins': {0.0: {'toothed': {0.0: 7.0, 1.0: 3.0}},

1.0: {'eggs': {0.0: 1.0, 1.0: 4.0}}}},

2: {'hair': {0.0: 2.0, 1.0: 1.0}},

4: {'hair': {0.0: {'toothed': {0.0: 7.0, 1.0: 5.0}}, 1.0: 1.0}},

6: {'aquatic': {0.0: 6.0, 1.0: 7.0}},

8: 7.0}}

**Test accuracy: 0.8571428571428571**

Attribute Selection Criterion: 1

best\_feature is: venomous

best\_feature is: backbone

best\_feature is: catsize

best\_feature is: hair

best\_feature is: airborne

best\_feature is: predator

best\_feature is: tail

best\_feature is: hair

best\_feature is: breathes

best\_feature is: fins

best\_feature is: aquatic

best\_feature is: airborne

best\_feature is: hair

best\_feature is: hair

best\_feature is: eggs

best\_feature is: toothed

best\_feature is: hair

best\_feature is: hair

best\_feature is: toothed

best\_feature is: legs

{'venomous': {0: {'backbone': {0.0: {'catsize': {0.0: {'hair': {0.0: {'airborne': {0.0: {'predator': {0.0: 6.0,

1.0: 7.0}},

1.0: 6.0}},

1.0: 6.0}},

1.0: 7.0}},

1.0: {'tail': {0.0: {'hair': {0.0: 5.0,

1.0: 1.0}},

1.0: {'breathes': {0.0: 4.0,

1.0: {'fins': {0.0: {'aquatic': {0.0: {'airborne': {0.0: {'hair': {0.0: 2.0,

1.0: 1.0}},

1.0: {'hair': {0.0: 2.0,

1.0: 1.0}}}},

1.0: {'eggs': {0.0: 1.0,

1.0: {'toothed': {0.0: {'hair': {0.0: 2.0,

1.0: 1.0}},

1.0: 5.0}}}}}},

1.0: 1.0}}}}}}}},

1: {'hair': {0.0: {'toothed': {0.0: 7.0,

1.0: {'legs': {0.0: 3.0,

4.0: 5.0}}}},

1.0: 6.0}}}}

**Test accuracy: 0.7142857142857143**

While the normalized gain ratio has a significantly lower accuracy than the unnormalized information gain algorithm, in general gain ratio should be preferable to information gain since it doesn’t give preference to attributes which have more data points. However, it is possible that the attributes that have more data points are coincidentally also stronger indicators of the final classification, which may be the case with this data causing information gain to be more accurate than gain ratio.

2.1)

a) Since only support vectors have non-zero , the support vectors are points 2, 6, and 18

b) The normal vector is given by [-1.338076, -0.3388998]

c) The bias is given by 03

d) The decision boundary function is given by

e) Plugging the point into the decision boundary function we get

Since the result is greater than 1, the boundary function classifies the point as +1.

f) Graph showing data points and decision boundary:

2.2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Kernel** | **Support Vectors** | **Accuracy** |
| Hard | Linear | (Terminated) 1048 | 0.5547445255474452 |
| Soft | Linear | 34 | 0.9890510948905109 |
| Soft | Polynomial | 19 | 0.9270072992700736 |
| Soft | Gaussian | 35 | 0.4452554744525547 |

A shown in the table above, using hard SVM results in a “Terminated (singular KKT matrix” error which I believe means that the data is not linearly separable and so Hard SVM is not possible. The resulting analysis for Hard SVM is thus meaningless and probably close what would be achieved with random classification. For Soft Linear SVM, the accuracy is extremely high at nearly 99%. The Polynomial kernel has a slightly worse accuracy and Gaussian a significantly worse accuracy. Clearly a gaussian separator is not well suited for this data set (in the not unlikely case that the low accuracy is a result of a bug in my code and not the Gaussian function, this analysis should be disregarded). Due to it having the highest accuracy and lowest complexity thus reducing overfitting and run time (the last of which is very noticeable especially compared with the Gaussian kernel), the linear kernel appears to be the best choice.