#Name: Hun Lee

#UID: 604958834

1.1

10 dem, 10 rep

Info(D) = I(10,10) = -10/20log(10/20)-10/20log(10/20) = 1

8 yes: 6dem 2rep, 12 no: 4dem 8rep

gain(infant)= 1 - .87548 = .12452

I(6,2) =-6/8log(6/8)-2/8log(2/8) = .31127+0.5 = .81127

I(4,8)=-4/12log(4/12)-8/12log(8/12) =.52832+.38997=.91829

Info(infant) = 8/20\*.81127 + 12/20\*.91829 = .32451+.55097 =.87548

10 yes: 4dem 6rep, 10 no: 6dem 4rep

gain(water)= 1 - .971 = .029

I(4,6) =-4/10log(4/10) - 6/10log(6/10) = .52877+.44218 = .971

I(6,4)=- -6/10log(6/10) - 4/10log(4/10) =.52877+.44218 = .971

Info(water)= .971

11 yes: 9dem 2rep, 9 no: 8rep 1dem

gain(budget)= 1 - .44419 = .55581

I(9,2) =-9/11log(9/11) - 2/11log(2/11) = .23687+.44717 = .68404

I(1,8)= -1/9log(1/9) - 8/9log(8/9) =.35221+.15104 = .50325

Info(budget)= 11/20\*.68404 + 9/20\*.15104 = .44419

Yes side:

|  |  |  |  |
| --- | --- | --- | --- |
| class | infants | water | Budget |
| D | y | n | y |
| D | y | y | y |
| D | N | y | y |
| D | Y | n | y |
| D | Y | y | y |
| R | N | y | y |
| D | N | n | y |
| D | N | n | y |
| D | Y | n | y |
| D | Y | n | y |
| R | Y | n | y |

Info(budget|yes)-9/11log(9/11) - 2/11log(2/11) = .23687+.44717 = .68404

4 yes: 3D 1R, 7no: 6D 1R

gain|bud(water) =.68404-.67153 = 0.01251

I(3,1) =-3/4log(3/4) - 1/4log(1/4) =.31128+.5 = .81128

I(6,1) =-6/7log(6/7) - 1/7log(1/7) = .19062+.40105= .59167

INFO|bud(water) =4/11\*.81128 + 7/11\*.59167=.67153

7 yes: 6D 1R, 4 no: 3D, 1R

gain|bud(infant) =.68404-.67153 = 0.01251

I(3,1) =-3/4log(3/4) - 1/4log(1/4) =.31128+.5 = .81128

I(6,1) =-6/7log(6/7) - 1/7log(1/7) = .19062+.40105= .59167

INFO|bud(infant) = 4/11\*.81128 + 7/11\*.59167=.67153

Their gain is the same so I can choose any one. I chose to go with water.

Yes water:

|  |  |  |
| --- | --- | --- |
| Class | Infants | water |
| D | y | y |
| D | n | y |
| D | y | y |
| R | n | y |

No water:

|  |  |  |
| --- | --- | --- |
| class | infants | water |
| D | y | n |
| D | y | n |
| D | n | n |
| D | n | n |
| D | y | n |
| D | y | n |
| R | y | n |

No Budget:

|  |  |  |  |
| --- | --- | --- | --- |
| class | infants | water | budget |
| R | n | y | n |
| R | n | y | n |
| D | n | y | n |
| R | y | y | n |
| R | n | y | n |
| R | n | n | n |
| R | n | n | n |
| R | n | y | n |
| R | n | n | n |

Info(budget|no)- -1/9log(1/9) - 8/9log(8/9) =.35221+.15104 = .50325

6 yes: 1D 5R, 3 no: 0D 3R

gain|bud(water) =.50325-.43334 = .06991

I(3,0) = 0

I(1,5) =-1/6log(1/6) - 5/6log(5/6) = .43083+.21919 = .65002

INFO|bud(water) = .43334

1 yes: 0D 1R, 8 no: 7R, 1D

gain|bud(infant) =.50325 -.48316 = .0201

I(0,1) = 0

I(1,7) =-1/8log(1/8) - 7/8log(7/8) = .375+.16856 = .54356

INFO|bud(infant) = 8/9\*.54356 = .48316

Their gain is the same so I can choose any one. I chose to go with water.

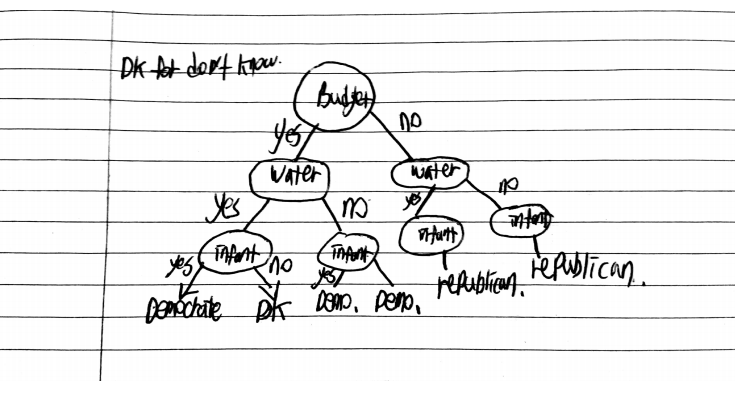
Water has higher gain here so we go with water.

Yes water:

|  |  |  |
| --- | --- | --- |
| **class** | **infants** | Water |
| **R** | **n** | **y** |
| **R** | **n** | **y** |
| **D** | **n** | **y** |
| **R** | **y** | **y** |
| **R** | **n** | **y** |
| **R** | **n** | **y** |

**No water:**

|  |  |  |
| --- | --- | --- |
| **class** | **infants** | **Water** |
| **R** | **n** | **n** |
| **R** | **n** | **n** |
| **R** | **n** | **n** |



1.2

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: array([7.]), 1.0: array([3.])}}, 1.0: {'eggs': {0.0: array([1.]), 1.0: array([4.])}}}}, 2: {'hair': {0.0: array([2.]), 1.0: array([1.])}}, 4: {'hair': {0.0: {'toothed': {0.0: array([7.]), 1.0: array([5.])}}, 1.0: array([1.])}}, 6: {'aquatic': {0.0: array([6.]), 1.0: array([7.])}}, 8: array([7.])}}

Test accuracy: 0.8571428571428571

Attribute Selection Criterion: 1

best\_feature is: feathers

best\_feature is: backbone

best\_feature is: airborne

best\_feature is: predator

best\_feature is: milk

best\_feature is: fins

best\_feature is: legs

{'feathers': {0: {'backbone': {0.0: {'airborne': {0.0: {'predator': {0.0: array([6.]),1.0: array([7.])}},

1.0: array([6.])}},1.0: {'milk': {0.0: {'fins': {0.0: {'legs': {0.0: array([3.]),4.0: array([5.])}},1.0: array([4.])}},

1.0: array([1.])}}}},1: array([2.])}}

Test accuracy: 0.8095238095238095

I would choose to use information gain method to predict my result because the test accuracy for it is higher than the other one. I expect that ratio method performs not as well because, it is mean to be used on a large data set. Looking at the number of data, it might not satisfy that requirement.

2.1

\alpha\_2 = 0.5084

\alpha\_6= 0.4625

\alpha\_18 = 0.9709

others = 0

a.

support vectors are 2, 6, 18

b.

w=∑αiyixi.

w1 = (.5084\* 0.91\* 1) + (0.4625\* 0.41\*1) + (.9709\* 2.05\* -1) = 0.4626+ 0.1896 - 1.9903 = -1.3381

w2 = (.5084\* 0.32\* 1) + (0.4625\* 2.04\*1) + (.9709\* 1.54\* -1) = 0.1627 + 0.9435 -1.4952 = -0.389

w = [-1.3381,-0.389]

c.

x2=[0.91,0.32] x6=[0.41,2.04] x18=[2.05,1.54] 1.4301 .2072 +

wx2=(-1.2176 -0.1245) = -1.3422, (y-wx2)/N =1 + 1.3422 = 2.3422/3 = 0.7807

wx6=(-0.5486 - 0.7935) = -1.3421,(y-wx6)/N =1 + 1.3421 = 2.3421/3 = 0.7807

wx18=(-2.743 -0.5990) = -3.342, (y-wx18)/N =-1 + 3.342 = 2.342/3 = 0.7806

sum of all products = 2.342 = B

d.

f(x) = wTx+b

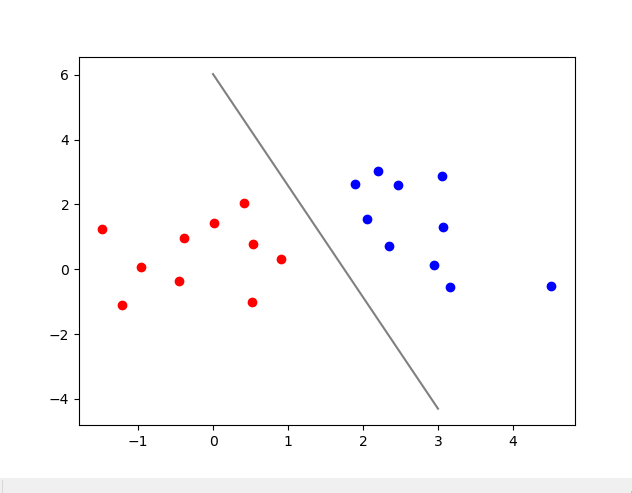
f(x) = -1.3381x1 + -0.389x2 + 2.342

e.

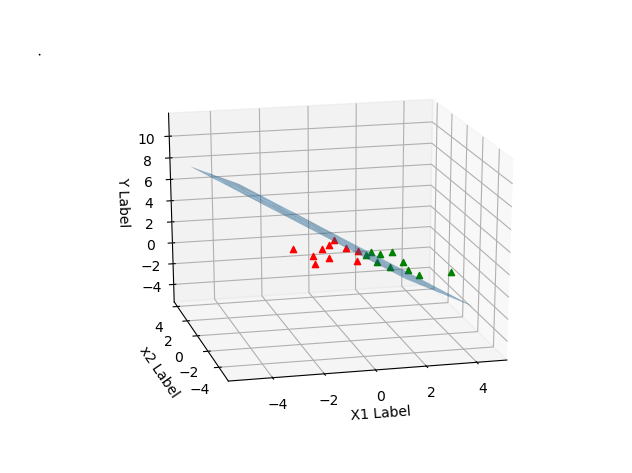
f(1,2) = 1.3381 -.778 + 2.342 = 2.9021

since the value is greater than 1, it belongs to class 1

2D graph



3D graph



2.2

a.

Margin: 0

kernel Type: 0

Terminated (singular KKT matrix).

1098 support vectors out of 1098 points

Test accuracy: 0.5547445255474452

b.

Margin: 1

kernel Type: 0

Optimal solution found.

34 support vectors out of 1098 points

Test accuracy: 0.9890510948905109

c.

Margin: 1

kernel Type: 1

Optimal solution found.

19 support vectors out of 1098 points

Test accuracy: 0.927007299270073

d.

Margin: 1

kernel Type: 2

Gaussian Kernel computing: I am not efficiently implemented. Please consider smarter implementation

Optimal solution found.

35 support vectors out of 1098 points

Test accuracy: 1.0

**What function Would you choose?**

Looking at the result and accuracy, it seems that soft margin and gaussian model is the best choice. However, if you consider the cost that comes with lifting vectors using kernel functions and cost you have to pay for the soft margin, it is best to choose soft margin with linear kernel. While we do not need to play with entries which helps us to saves some computations latency, accuracy of the test is not too much different from more expensive models. In fact this model has .99 accuracy. While gaussian function has 100 percent accuracy, the difference can be over-looked. For these reasons, I would use soft margin with linear kernel.