Kenneth Eversole
11463589
CPTS440
HW11

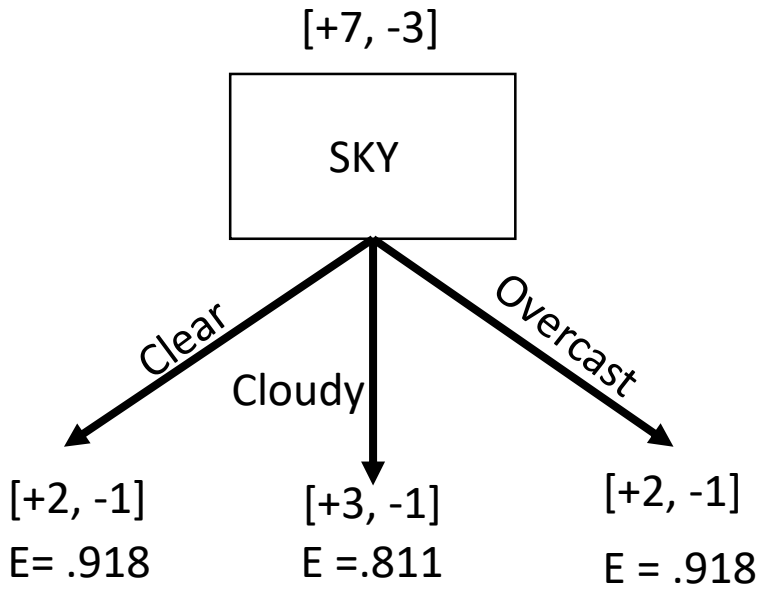
1 Decision trees)

1A)

YES = 7
No = 0

Entropy(s)$= -\frac{7}{10}\log_2\frac{7}{10} -\frac{3}{10}\log_2\frac{3}{10} =.\mathbf{882}$

1B)



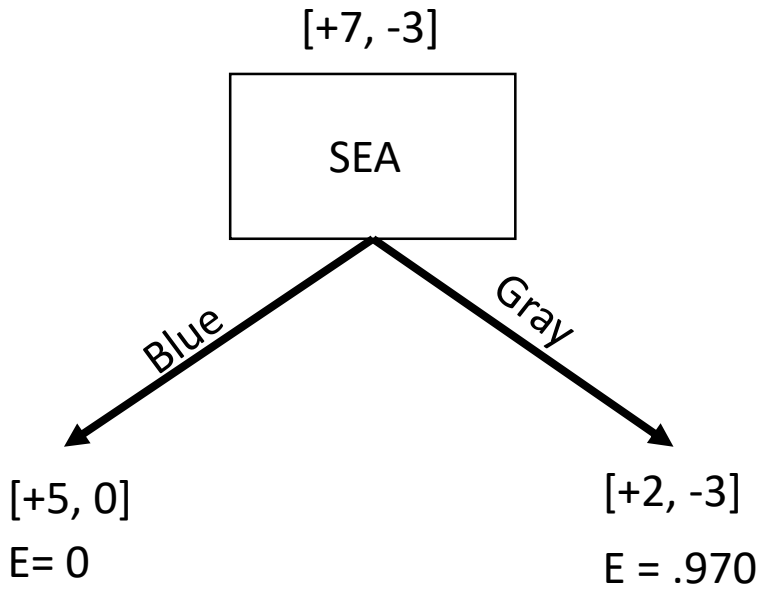
Entropy(S, Sky) = $.882 \quad -\left[\frac{3}{10}(.918) + \frac{4}{10}(.811) + \frac{3}{10}(.918)\right]$

= .882 - .8752

= .0068

Information gain = .0068

1C)



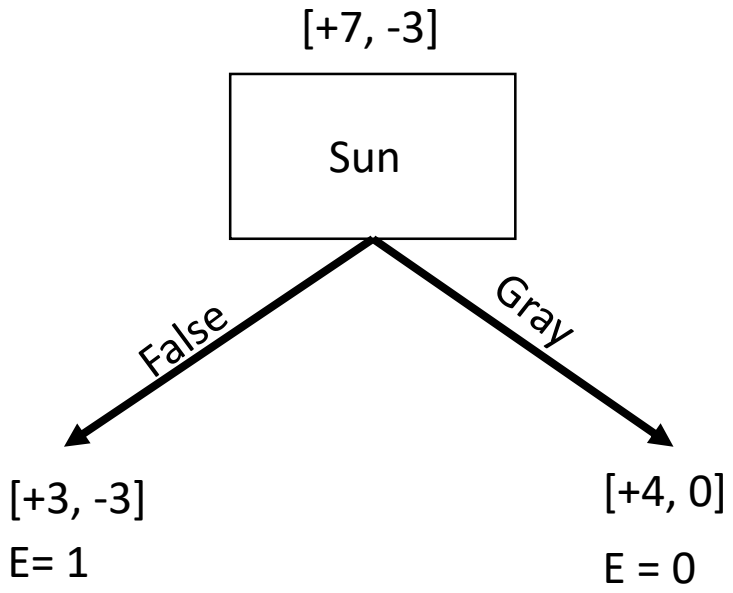
Entropy(S, Sea) = $.882 \quad -\left[\frac{5}{10}0 + \frac{5}{10}(.970)\right]$

= .882 - .450

= .3965

Information gain = .3965

1D)



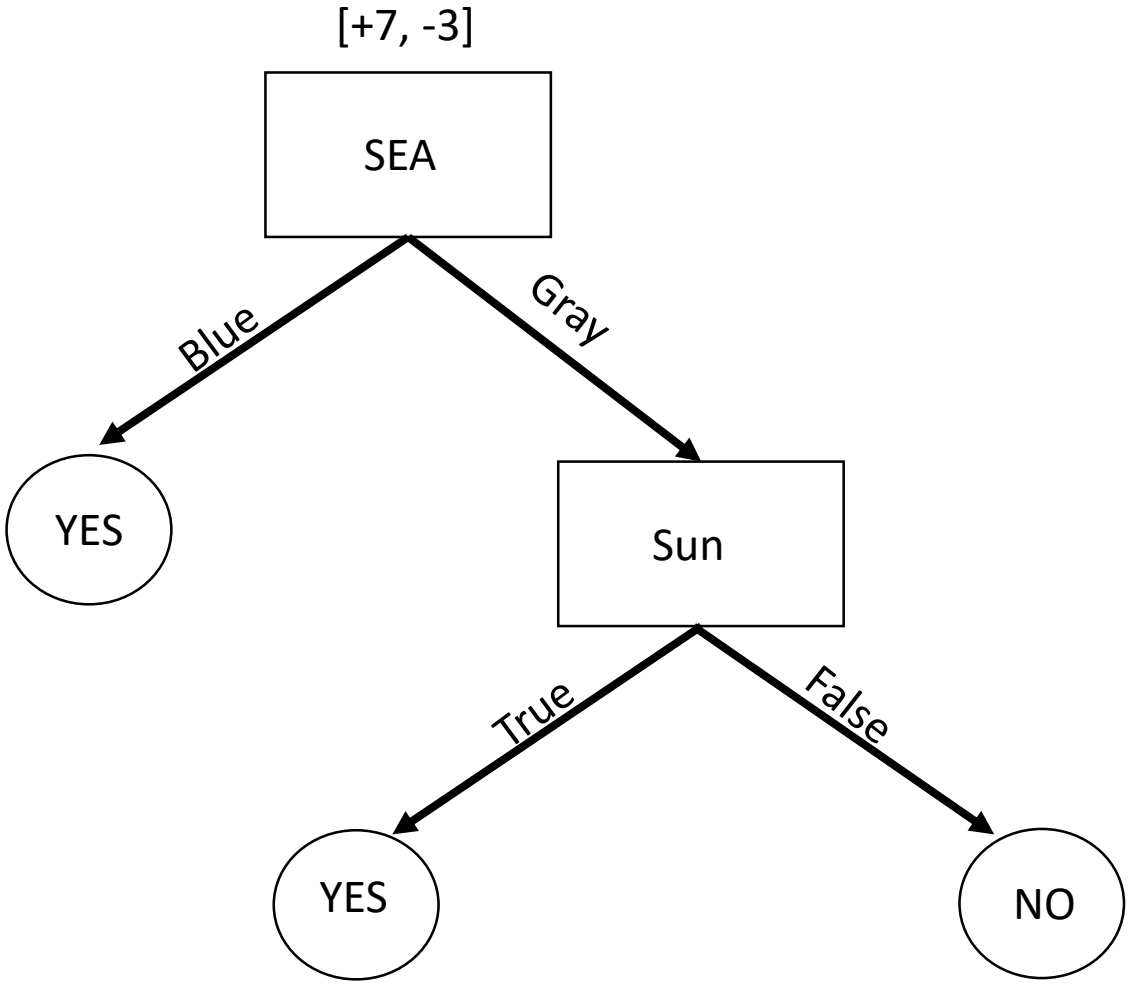
Entropy(S, Sea) = .882 $-\left[\frac{6}{10}1+\frac{4}{10}(0)\right]$

= .882 - .6

= .282

Information gain = .282

1E)



WORK:

Entropy(gray, sun) = .970 $-$
$[\frac{2}{5}$ entropy(true,sun) + $\frac{3}{5}$entropy(false,sun)]

= .970 –[0 + 0]
IG = .970

Entropy(gray, sun) = .970 $-$
$[\frac{2}{5}$ entropy(clear,sky) + $\frac{2}{5}$entropy(cloudy,sun) +
$\frac{2}{5}$entropy(overcast,sky)]

= .970 – $[\frac{2}{5}$ (1) + $\frac{2}{5}$ (1)+ $\frac{2}{5}$(0)]
=.970 - .8 = .17
IG = .17

1F)

How would the learned Decision Tree classify the new instance <Sky=overcast,Sea=gray,Sun=true>

It would classify it as **Sail= yes**

2 perceptron)

2A)

Sky	Sea	Sun	Sail
1	1	1	1
1	2	2	1
1	2	1	0
2	1	2	1
2	1	1	1
2	2	2	1
2	2	1	0
3	1	2	1
3	1	1	1
3	2	1	0

2 B)

Rule: if W * X is greater than 0 then 1(yes) else 0(no)

I decided to code this instead of having the TA's try to decipher my hand writing. (the data in results of each pass is on the next page)

Code ->

```
import numpy as np
import itertools
def perceptron():
    output = open("output.txt", "w")
    sail_data = list()
    sail_data_labels = list()
    for data, label in zip(open("data.txt", "r"), open("datalabels.txt", "r")):
        temp = list(map(float, data.split(",")))
        temp2 = np.array(temp)
        sail_data.append(temp2)
        sail_data_labels.append(int(label.strip('\n')))
    weight = np.zeros(4, np.float) #weight
    for i in range(len(weight)):
        weight[i] = 1.0
    for k in range(3):
        tempstr = "[pass " + str(k + 1) + "]\n"
        output.write(tempstr)
        for j in range(len(sail_data)):
            x = sail_data[j]
            yHat_t = np.dot(x, weight)
            correct_label = sail_data_labels[j]


            if yHat_t > 0:
                yHat_t = 1
            else:
                yHat_t = 0

            if yHat_t != correct_label:
                weight = weight + (.5 *(correct_label-yHat_t)*x )
                tempstr = str(weight.tolist())  + "\n"
                output.write(tempstr)
            else:
                output.write("Guess Correct\n")
    tempstr ="Trained weight: " + str(weight.tolist())  + "\n"
    output.write(tempstr)
    output.close()
if __name__ == "__main__":
    perceptron()
```

2 B continued)

Pass 1	Pass 2	Pass 3
Guess Correct	[0.5, 0.0, -1.0, 1.0]	Guess Correct
Guess Correct	Guess Correct	Guess Correct
[0.5, 0.5, 0.0, 0.5]	Guess Correct	Guess Correct
Guess Correct	Guess Correct	Guess Correct
Guess Correct	Guess Correct	Guess Correct
Guess Correct	Guess Correct	Guess Correct
[0.0, -0.5, -1.0, 0.0]	Guess Correct	Guess Correct
[0.5, 1.0, -0.5, 1.0]	Guess Correct	Guess Correct
Guess Correct	Guess Correct	Guess Correct
[0.0, -0.5, -1.5, 0.5]	Guess Correct	Guess Correct

Trained weight: [0.5, 0.0, -1.0, 1.0]

2 c How would the learnedperceptronclassify the new instance <Sky=overcast,Sea=gray,Sun=true>?Show your work.) [3, 2,1]

X = [1,3,2,1]
Weight = [0.5, 0.0, -1.0, 1.0]

Y_hat = Weight * X

Y_hat = -0.5 < 0 so Y_hat = no
Guess: NO