√1

a) $a = (0, 1)$, $b = (-2, -1)$, $c = (0.5, -0.5)$

$w = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ $w_0 = 0$

support vectors

$w^T x^a = -1$

$w^T x^b = -1$

$w^T x^c = 1$

b) $w = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

$$d = \frac{|g(x)|}{||w||} = \frac{|g(x)|}{\sqrt{2}}$$

$$d(-2, -1) = \frac{|-2 - (-1)|}{\sqrt{2}} = \frac{1}{\sqrt{2}}$$

$$d(-1, 1.5) = \frac{|-1 - 1.5|}{\sqrt{2}} = \frac{2.5}{\sqrt{2}}$$

$$d(1.5, -1) = \frac{|1.5 - (-1)|}{\sqrt{2}} = \frac{2.5}{\sqrt{2}}$$

i) Yes, the decision boundary will change. Because (0.5, -0.5) is the only support vector of negative labels

No, as (-1, -2,5) is not a support vector

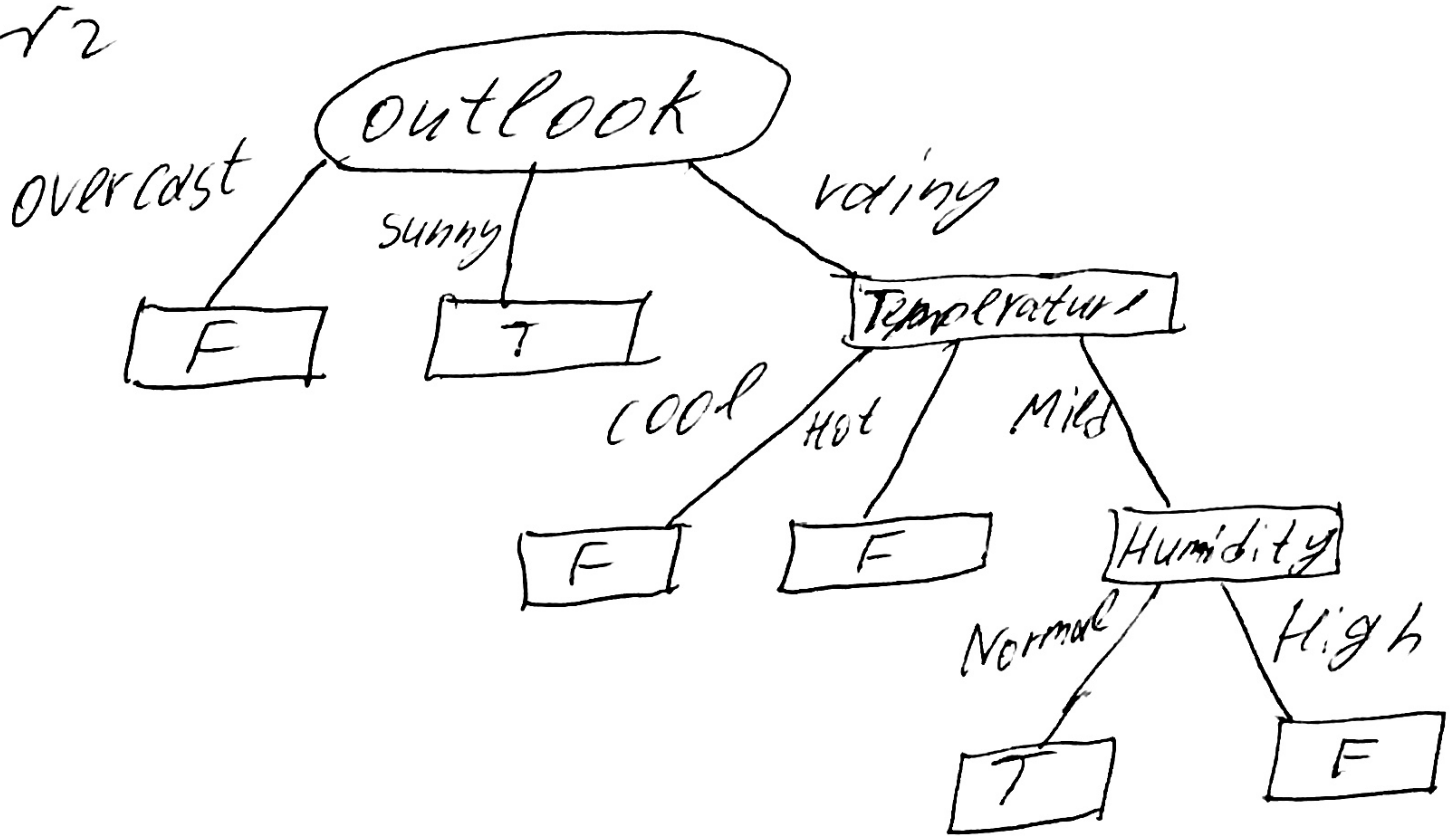
d) Yes, it will change. Soft margin will be used

e) large value of C: narrow margin

small value of C: wide margin

f) Even it data is linearly separable it is better to use soft margin. As we would have noisy data in real-world application. Soft margin perfectly handes noisy data

If the speed is the most important factor, linear should be used. In other cases it is better to use kernel. Since it will mostly have better results than linear r2


outlook
overcast
sunny
rainy
F
T
Temperature
cool
Hot
Mild
F
F
Humidity
Normal
High
T
F

~~before~~ before: entropy $= -\frac{9}{15}\log_2\left(\frac{9}{15}\right) - \frac{6}{15}\log_2\left(\frac{6}{15}\right) =$

$= -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.97095$

1st split:

for Outlook:

sunny:

entropy = 0

~~overcast = 0~~

overcast:

entropy = 0

rainy:

entropy $= -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right) =$

$= 0.72192$

Entropy(Outlook) $= \frac{5}{15} \cdot$ Entropy(rainy) $=$

$= 0.24064$

IG(Outlook) $= 0.73031$

2nd split: before: entropy $= -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\frac{4}{5} =$

$= 0.721928$

for Temperature:

cool:

entropy = 0

hot:

entropy = 0

mild:

entropy = 1

Entropy(Temperature) $= \frac{2}{5} \cdot 1 = 0.4$

IG(Temperature) = ~~[illegible]~~ 0.321928

3rd split:

entropy before = 1

for Humidity:

entropy(normal) = 0

entropy(high) = 0

entropy(humidity) = 0

IG(humidity) = 1

b) ~~False~~ No, No

Q3:

a)

```
(base) Sevas-MacBook-Pro:hw4_programming Seva$ python3 hw4.py
Training/validation accuracy for minimum node entropy 0.010000 is 1.000 / 0.863
Training/validation accuracy for minimum node entropy 0.050000 is 0.999 / 0.863
Training/validation accuracy for minimum node entropy 0.100000 is 0.997 / 0.865
Training/validation accuracy for minimum node entropy 0.200000 is 0.990 / 0.867
Training/validation accuracy for minimum node entropy 0.400000 is 0.979 / 0.861
Training/validation accuracy for minimum node entropy 0.800000 is 0.919 / 0.856
Training/validation accuracy for minimum node entropy 1.000000 is 0.871 / 0.840
Training/validation accuracy for minimum node entropy 2.000000 is 0.596 / 0.600
Test accuracy with minimum node entropy 0.200000 is 0.872
```

$\Theta = 0.2$ should be used. Training accuracy: 0.990. Validation accuracy: 0.867

b) For $\Theta <= 1$ the validation accuracy is very similar 84-86.3%. While for $\Theta = 2$ the tree has not grown enough. Underfitting of the training data clearly indicates this. And this also leads to much worse results for validation accuracy.