(a) The root attribute is Stripes. Below are the calculations of the entropy.

$$Entropy(S) = -\sum p_i \log_2 p_i = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.97095$$

$$Entropy(S, Color) = Entropy(S) - \frac{4}{5} Entropy(purple) - \frac{1}{5} Entropy(red)$$

$$Entropy(S, Color) = 0.97095 - \frac{4}{5} \left( -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} \right) - \frac{1}{5} \times 0 = 0.17095$$

$$Entropy(S, Stripes) = Entropy(S) - \frac{2}{5} Entropy(No) - \frac{3}{5} Entropy(Yes)$$

$$Entropy(S, Stripes) = 0.97095 - \frac{2}{5} \times 0 - \frac{3}{5} \left( -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \right) = 0.41997$$

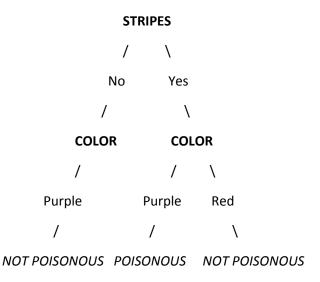
$$Entropy(S, Texture) = Entropy(S) - \frac{3}{5} Entropy(Smoth) - \frac{2}{5} Entropy(Rough)$$

$$Entropy(S, Texture) = 0.97095 - \frac{3}{5} \left( -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \right) - \frac{2}{5} \times \left( -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} \right)$$

$$= 0.01997$$

Because the entropy for Stripes is largest, that is Entropy(S, Stripes) > Entropy(S, Color) > Entropy(S, Texture), the root attribute will be Stripes to obtain maximum information gain.

(b) Below is the decision tree



For x=3.2 and the similarity measure sim(x,x')=|x-x'|, three nearest neighbor of x=3.2 would be :  $3NN=\{(3,5),(2,11),(3,8)\}$ . Therefore  $y=(5+11+8)\div 3=8$ .

The examples map from  $[x_1, x_2]$  to  $[x_1, x_1x_2]$  coordinates as follows:

From the mapping and results above it is obvious that XOR will be positive when  $x_1x_2=-1$  and negative when  $x_1x_2=+1$ . Therefore the maximum margin separator is the line  $x_1x_2=0$ , with the margin equals to 1. The separator corresponds to the  $x_1=0$  &  $x_2=0$  axes in the original space – this can be thought of as the limit of a hyperbolic separator with two branches.

To calculate the squared Euclidean distance in the projected space we have

$$dist(x_i, x_j) = (\phi(x_i) - \phi(x_j))^2$$

$$(\phi(x_i) - \phi(x_j))^2 = \phi(x_i)^2 - 2\phi(x_i)\phi(x_j) + \phi(x_j)^2$$

$$= \phi(x_i)\phi(x_i) - 2\phi(x_i)\phi(x_j) + \phi(x_j)\phi(x_j)$$

With the kernel trick

$$K(x_i, x_j) = \phi(x_i)\phi(x_j)$$

We have

$$\left(\phi(x_i) - \phi(x_j)\right)^2 = K(x_i, x_i) - 2K(x_i, x_j) + K(x_j, x_j)$$

That is, we can compute the squared Euclidean distance in feature space between any two points  $x_i, x_j$  in the original space without explicitly computing the  $\phi$  mapping but instead using the kernel function.

To implement  $XOR(AND(x_1, x_2), x_3)$ , we first consider  $AND(x_1, x_2)$ :

- 1.  $x_1 = +1$ 
  - $AND(x_1, x_2) = +1 \rightarrow x_2 = +1$
  - $AND(x_1, x_2) = -1 \rightarrow x_2 = -1$
- 2.  $x_2 = +1$ 
  - $AND(x_1, x_2) = +1 \rightarrow x_1 = +1$
  - $AND(x_1, x_2) = -1 \rightarrow x_1 = -1$

To consider  $XOR(AND(x_1, x_2), x_3)$ 

- 1.  $x_3 = -1$ :  $AND(x_1, x_2) = +1 \rightarrow x_1 = +1, x_2 = +1$
- 2.  $x_3 = +1$ :  $AND(x_1, x_2) = -1 \rightarrow x_1 = +1$ ,  $x_2 = -1$  or  $x_1 = -1$ ,  $x_2 = +1$  or  $x_1 = -1$ ,  $x_2 = -1$

That is, we will have  $(x_1 \land x_2 \land \neg x_3) \lor (\neg x_1 \land x_3) \lor (\neg x_2 \land x_3) = 1$  to implement a neural network for  $XOR(AND(x_1, x_2), x_3)$ .

