Problem 2:

b) $H(Y) = 1 \quad [4^+ 4^-]$

$H(Y|X_1) = p(X_1=1) \cdot H(Y|X_1=1) + p(X_1=0) \cdot H(Y|X_1=0)$

$H(Y|X_1=1) \quad [2^+ 2^-]$

$= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}$.

$= -\log\frac{1}{2} = 1$.

Similarly,

$H(Y|X_1=0) = 1$.

$\therefore H(Y|X) = 1 \times \frac{1}{2} + 1 \times \frac{1}{2} = 1$

$IG_1 = H(Y) - H(Y|X_1) = 1 - 1 = 0$.

$H(Y|X_2) = p(X_2=1) H(Y|X_2=1) + p(X_2=0) H(Y|X_2=0)$

$\therefore H(Y|X_2=0) = [2^+ 2^-]$

$= 1$.

$H(Y|X_2=1) = [2^+ 2^-]$

$= 1$.

$H(Y|X_2) = \frac{1}{2} \times 1 + \frac{1}{2} \times 1$

$= 1$.

$IG_2 = H(Y) - H(Y|X_2) = 1 - 1 = 0$.

$H(Y|X_3) = p(X_3=0) \cdot H(Y|X_3=0) + p(X_3=1) \cdot$
$H(Y|X_3=1)$.

$H(Y|X_3=0) = [1^+ \ 3^-]$.

$= -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}$.

$= -\frac{1}{4}(-2) - \frac{3}{4} \cdot \log\frac{3}{4}$.

$= 0.8113$.

Similarly, for $H(Y|X_3=1) = 0.8113$.

$\therefore H(Y|X_3) = \frac{1}{2} \cdot 0.8113 + \frac{1}{2} \cdot 0.8113$

$= 0.8113$.

$\therefore IG_3 = 1 - 0.8113$

$\therefore IG_3 = 0.1887$.

---

a) For class label Y, we have $[4^+ \ 4^-]$

$\therefore H(Y) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$.

---

c) As the Information gain for attribute X_3 is highest at the root node, we select attribute X_3 at the root node to split the data.

d) The criteria for stopping the growth of the decision tree is as follows:

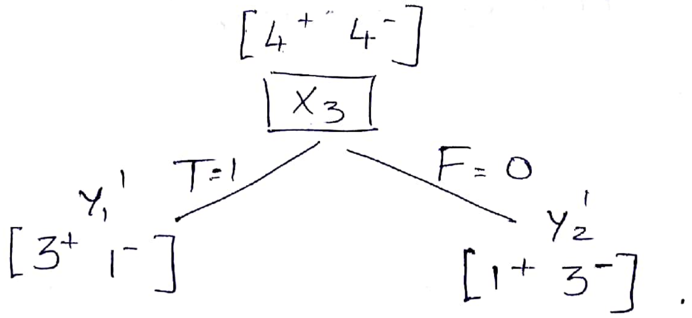
i) When we achieve a perfect split of data at the leaf nodes (Entropy is minimum & Information gain is maximum), we stop growing the tree further.

ii) If we run out of attributes to split on, we terminate the tree growth and compute the decision based on majority vote.

iii) If we do not ~~gain any information after splitting~~ have any data at the nodes, we stop the growth.

e) From part b) we know that attribute X_3 is the best for the root node.

$\therefore$ the tree is.



Consider X_1

<u>For $X_3 = 1$.</u>

$$\therefore H(Y_1') = \cancel{\tfrac{1}{2}} - \frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}$$

$$= 0.8113.$$

$$H(Y|X_1) = \frac{1}{2}\left[-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right] + \frac{1}{2}\left[-1\log_2 1 - 0\log_2 0\right]$$

$$= 0.5 \qquad IG(Y|X_1) = 0.3113.$$

$$H(Y|X_2) = \frac{3}{4}\left[-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right]$$

$$+ \frac{1}{4}\left[-\log_2 1 - 0\log_2 0\right]$$

$$= 0.6885.$$

$IG(Y|X_2) = 0.1228.$

Since $IG(\cancel{*}Y|X_1)$ is higher, we select X_1 as the splitting attribute.

<u>For $X_3 = 0$</u>

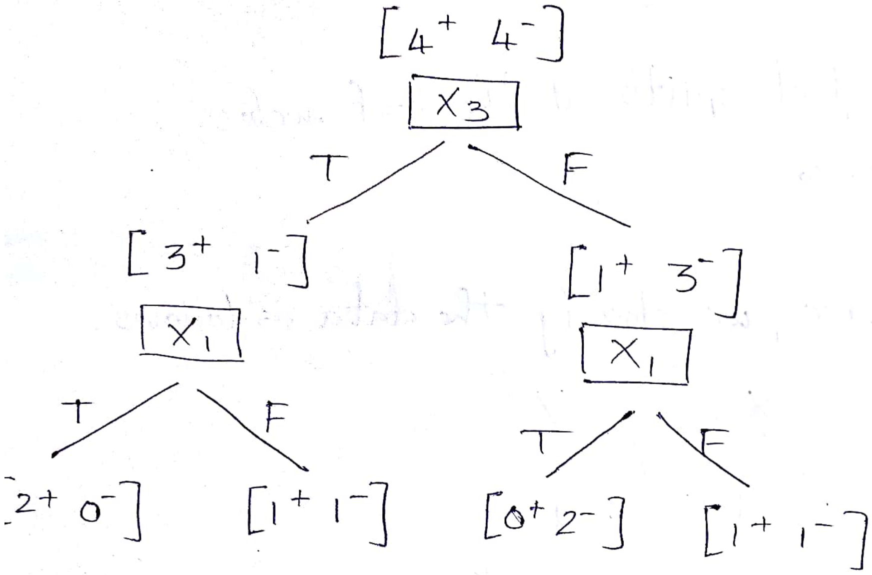
$$H(Y_2') = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}.$$

$$= 0.8113.$$

$H(Y|X_1) = 0.5$ $\qquad$ $IG(Y|X_1) = 0.3113$

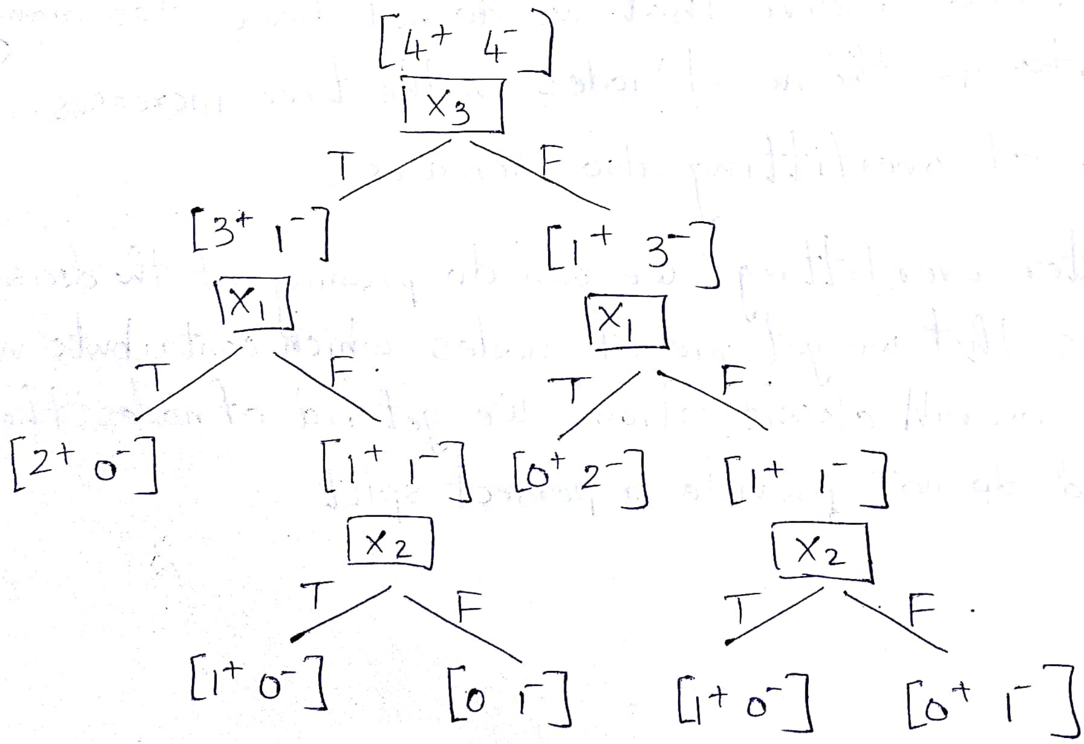
$H(Y|X_2) = 0.6885$ $\qquad$ $IG(Y|X_2) = 0.1228.$

Now, the tree looks as follows.



As we have only X_2 attribute left and we still don't have a perfect split at 2 leaf nodes, let's split them and hope that we achieve a perfect split.

∴ the final decision tree looks as follows:



As, we get perfect split at all the leaf nodes, we stop growing the tree.

f) As we don't have imperfect splits at the leaf nodes, the training error is 0 %.

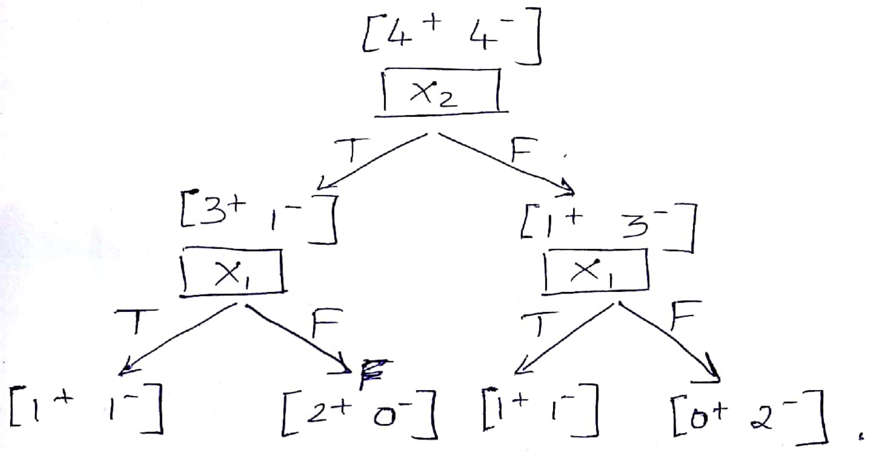
g) Using our decision tree, we classify the data as follows:

Instance	X_1	X_2	X_3	Y
9	1	1	1	1
10	1	0	0	0
11	0	1	1	1

h) In this case, we do not have a lot of samples to ensure that we keep some training data aside for validation. We also observe that we do not have too many attributes. ~~to~~ As the no of nodes in the tree increases the chances of overfitting also increases.

To counter overfitting, we can do pruning of the decision tree. Pruning means that we get rid of nodes which contribute very little to the overall classification. We get rid of nodes that ~~are~~ weigh less and do not provide a perfect split.

i) Yes the structure of the learned decision tree would change. It would look as follows.



Notice that if we further try to split the leaf nodes we do not gain any more information. We also have run out of ~~nodes~~ attributes to split with.

The leaf nodes have changed. Notice that we no longer have a perfect split at the leaf nodes. In this case, we flip a coin and assign a ~~random~~ class randomly as we have 50 - 50 % distribution at the leaf nodes. If we had an uneven distribution, we do a majority vote and assign a class