Homework I – due Mar 28th

SUBMITTED BY: Jyotsna Rajaraman

(Q1) // returns a tree

```
DECISION-TREE-LEARNING[examples, attributes, parent examples]:
    if examples is empty:
        return PLURALITY-VALUE(parent examples)
    else if all examples ∈ same class:
        return class
    else if attributes is empty:
        return PLURALITY-VALUE(examples)
    else:
        A, split ← BESTSPLIT(attributes, examples)
        tree ← new decision tree w root test A
        exs1 ← {e : e ∈ Examples and e.A ≤ split}
        exs2 ← examples − {exs1}
        subtree1 ← DECISION-TREE-LEARNING[exs1, attributes, examples]
        subtree2 ← DECISION-TREE-LEARNING[exs2, attributes, examples]
        Add 'A ≤ split' branch to tree w/ subtree 1
        Add 'A > split' branch to tree w/ subtree 2
        return tree
```

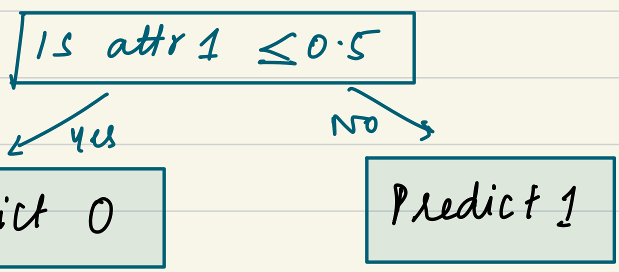
```
BEST SPLIT (attributes,examples):
            // returns attribute and value to split at
 max entropy = 0
  A = None ; split = None
  for  A  in  attribute:
       let vals = list of values of A
       sort vals in increasing order
       for  i  in  range (1, no.of values in vals-1):
          if  classlabel(vals(i)) not = classlabel(vals(i+1))
               split = vals(i) + vals(i+1) / 2
               calculalate ENTROPY(split, A, examples)
               if ENTROPY > max entropy:
                     maxentropy = ENTROPY
                      A = A
                      Split = split

      return  A, split.
```

(Q2) Early Stopping → may lead to less accurate or less expressive trees. Although it may be a solution to overfitting → it could result in underfitting

(a)

Attr 1	Attr 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

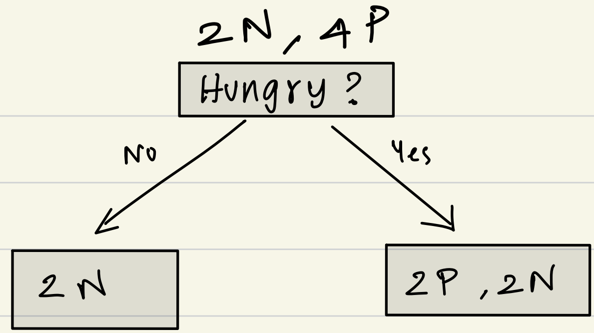
EARLY STOPPING TREE:



This results in a 50% chance for error!

∴ Early Stopping is problematic

(b)



$$\hat{p}_k = p \times \frac{p_k + n_k}{p+n} \qquad \hat{n}_k = n \times \frac{p_k + n_k}{p+n}$$

(i)

$$\hat{p}_k = 4 \times \frac{2}{6} = \frac{8}{6}$$

$$\hat{n}_k = 2 \times \frac{2}{6} = \frac{4}{6}$$

(ii)

$$\hat{p}_k = 4 \times \frac{4}{6} = \frac{16}{6}$$

$$\hat{n}_k = 2 \times \frac{4}{6} = \frac{8}{6}$$

$$\Delta = \left[\frac{(0 - 8/6)^2}{(8/6)^2} + \frac{(2 - 4/6)^2}{(4/6)^2}\right] + \left[\frac{(2 - 16/6)^2}{(16/6)^2} + \frac{(2 - 8/6)^2}{(8/6)^2}\right]$$

$$= \left[1 + \frac{1}{4}\right] + \left[\frac{1}{16} + \frac{1}{4}\right] = \frac{16}{16} + \frac{8}{16} + \frac{1}{16} = \frac{25}{16}$$

$$= 1.56$$

$\therefore$ We could prune this node $\because$ value of Δ is low & $\therefore$ null hyp is accepted

(Q3) x = 8 bit number

(a) Nothing is known

$\therefore$ x could be one of 2^8 options

Entropy = uncertainity. $\rightarrow -\sum_i p_i \log_2 p_i$

$-\sum_{i=1}^{2^8} p_i \log_2 p_i$ $\rightarrow$ prob of each num is equal $\because$ we know Nothing

$\therefore p_i = \frac{1}{2^8} \quad \forall i$

$-\cancel{2^8} \times \frac{1}{\cancel{2^8}} \cdot \log_2\left(\frac{1}{2^8}\right) = -1 \times -8 \log_2 2 = 8$ //

(b) Infogain : 8 - New entropy

Now only 2^7 options

$8 + \sum p_i \log_2 p_i = 8 + \cancel{2^7} \cdot \frac{1}{\cancel{2^7}} \log_2 2^{-7}$

$= 8 - 7 = 1$ //

(c) Now there are $\frac{2^8}{4} \times 3$ possibilities

$\therefore$ Info gain $= 8 -$ New entropy

$$= 8 + \sum p_i \log_2 p_i$$

$$= 8 + 3 \times 2^6 \times \frac{1}{3 \times 2^6} \log_2 (3 \times 2^6)^{-1}$$

$$= 8 - \log_2 (3 \times 2^6)$$

$$= 8 - (\log_2 3 + 6 \log_2 2)$$

$$= 8 - (1.585 + 6)$$

$$= 0.415$$

(d) Now there are only 8 possibilities.

$\therefore$ Info gain $= 8 -$ New entropy

$$= 8 + 8 \times \frac{1}{8} \log 2^{-3}$$

$$= 8 - 3 = 5$$

Q4. ML-Week1-Assignment.py is submitted

> It produces the validation curve and stores it as .png
> All the data is processed in ~10 minutes

Fun Facts:

Accuracy is higher for smaller depths! It appears that, atleast for this dataset we are better off with an decision tree with a depth 2-4 rather than a tree with depth 14-16