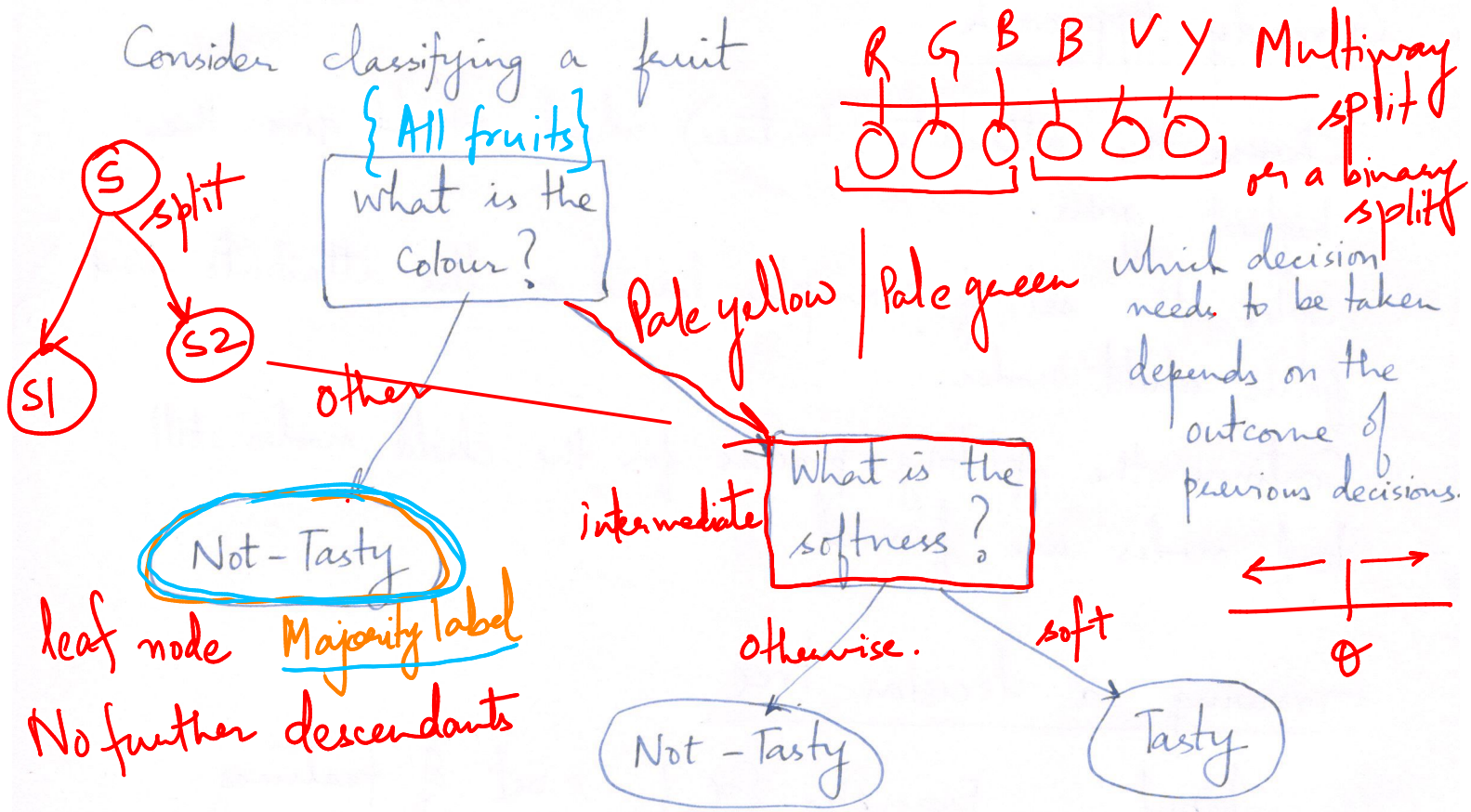
Decision Tree

A decision tree is a predictor which predicts the label for a given instance $x \in X$.

It makes a series of decisions for classifying a given instance x.

Consider classifying a fruit



which decision needs to be taken depends on the outcome of previous decisions.

Decision is taken at intermediate nodes.

Decision is based on examining the values of chosen attributes (features)

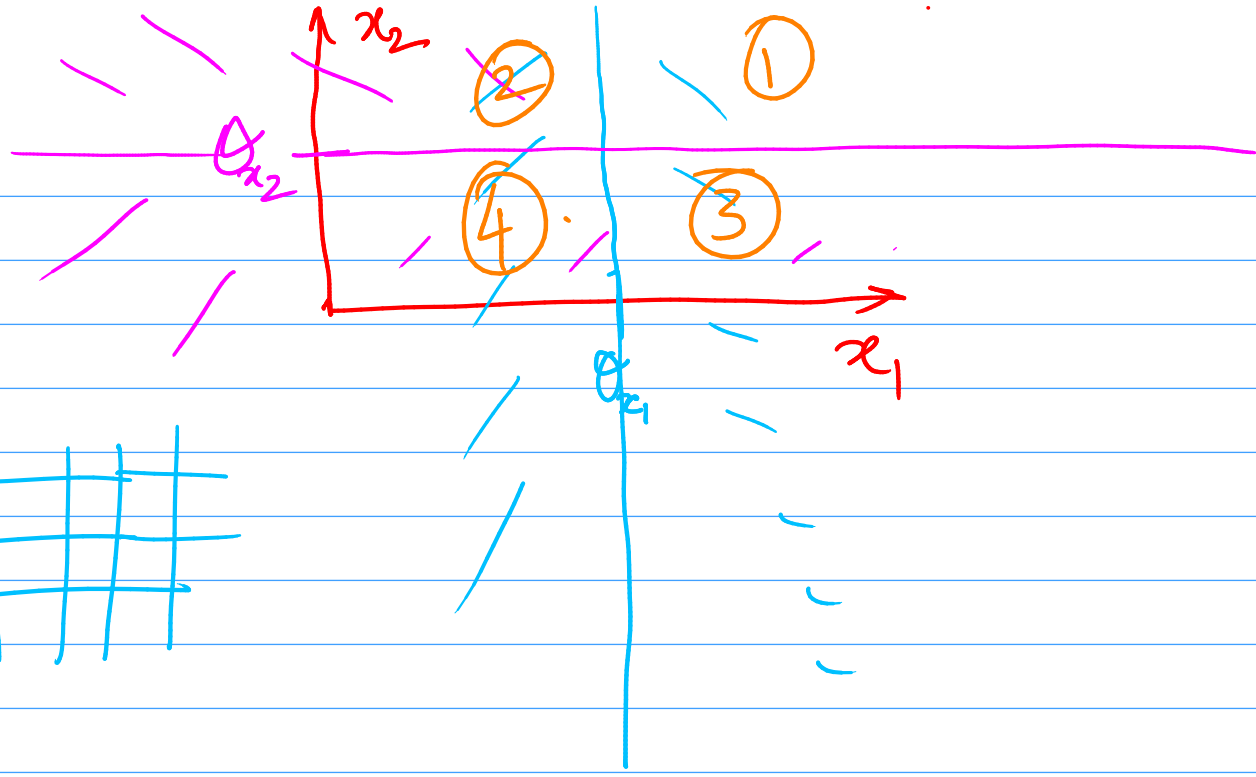
design

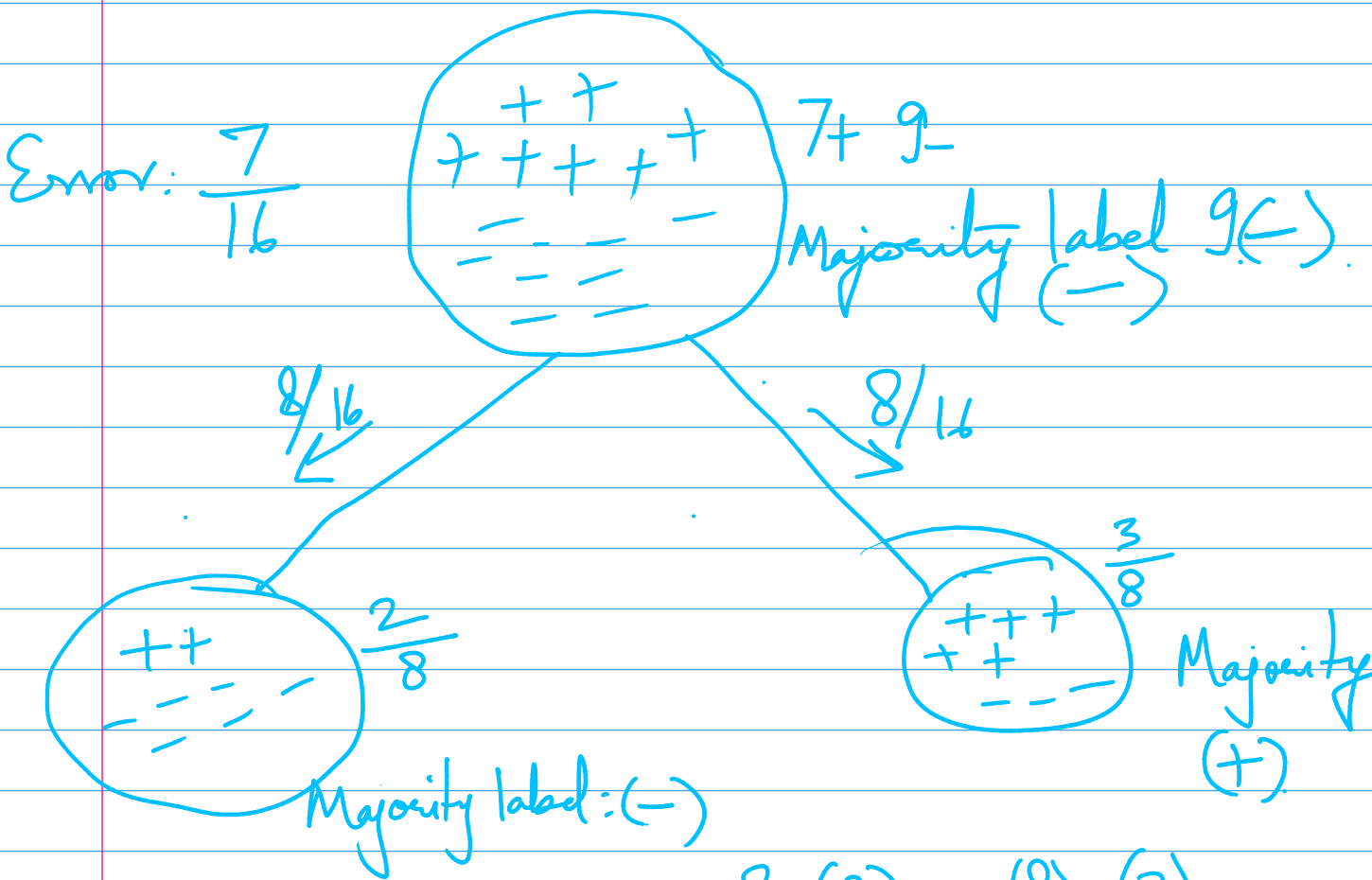
Every leaf node has a target label value associated with it.

The decision tree needs to be constructed in some optimised sense.

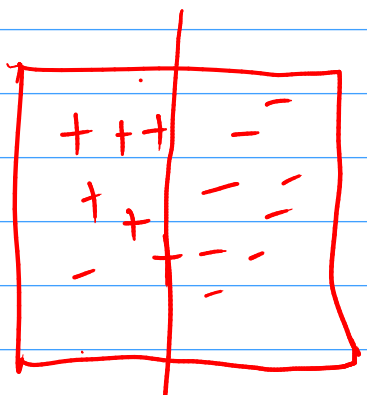
Considerations while constructing a decision tree:

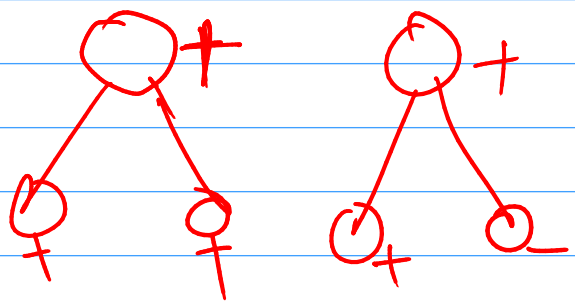
What split decision should be taken at the root and

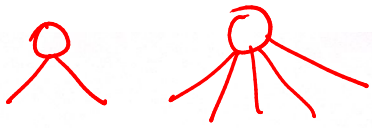




$$\frac{7}{16} - \frac{8}{16}\left(\frac{2}{8}\right) - \left(\frac{8}{16}\right)\left(\frac{3}{8}\right)$$







intermediate nodes?
- Which attribute to use?
- Which specific value or range of values to look for?

When to declare a node as a leaf node?

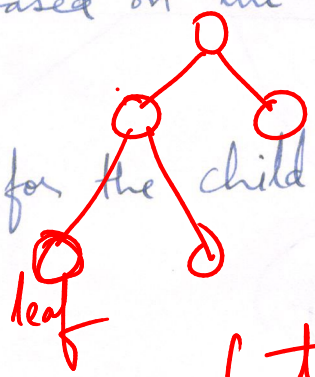
What should be the size (depth) of the tree?
→ how many decisions?

Greedy Approach

Choose the attribute (feature) which would give the highest gain.

Split the set of examples based on the attribute and form child nodes.

Continue the splitting process for the child nodes till leaf nodes are formed.



Growing (a) decision tree

Decision Tree

Input: Examples $\{x\}$, Set of features

$\{x\}$ → set of examples

features available for making a split.

base case of recursion:

If the target label of all examples is 1, make a leaf L with label 1.

If the target label of all examples is 0, make a leaf L with label 0.

No split is possible:

If the set of features is empty, make a leaf L, and assign the majority label.

procedure contd.

split is possible

Choose the feature with the best gain. greedy

Let's say the j-th attribute gives the best gain. choose

Split the examples into two subsets: jth attribute is binary

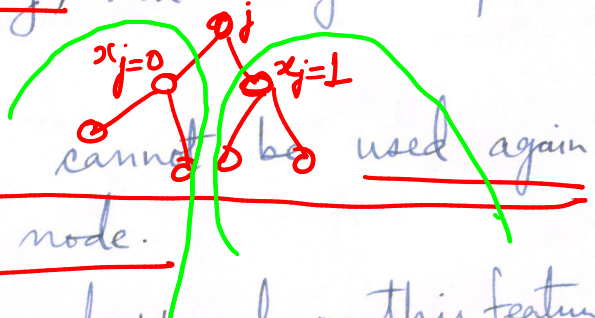
$I_1 \equiv \{x : x_j = 1\}$ jth feature

$I_2 \equiv \{x : x_j = 0\}$ if the feature x_j is binary.

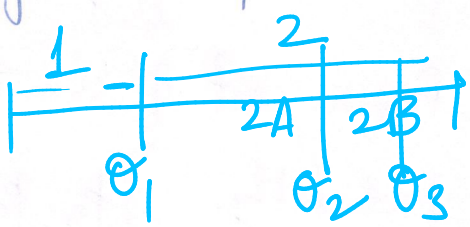
x_j: jth feature of x

Repeat the procedure for the subsets I_1 and I_2. recursive call.

If the split feature is binary, then we get a split into two subsets.



- In this case the feature j cannot be used again down the sub-tree after the node. why?? binary features. That is, after the split has happened on this feature, we cannot use this feature again as split feature in any descendant node.



Gain Measures

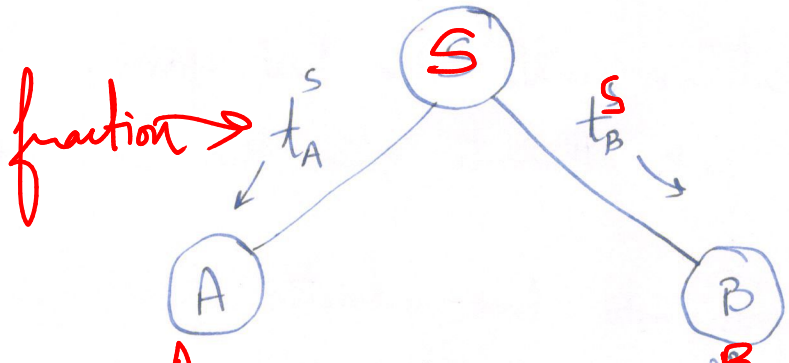
What should improve if we split a node?

1) Train error 2) Information Gain 3) Information Gain Ratio

4) Gini Index.

Gain as a reduction in training error.

How much the error decreases if the node is split?



fraction → t_A^S t_B^S

t_A^S: fraction of examples of S descending to node A.

$f_{\text{minority label}}^A$ $f_{\text{minority label}}^B$

leading to training error leading to training error.

Reduction in training error:

$$f_{\text{minority label}}^S - \left(t_A^S \cdot f_{\text{minority label}}^A + t_B^S \cdot f_{\text{minority label}}^B \right)$$

error rate after the split.

Gain as "Information Gain"

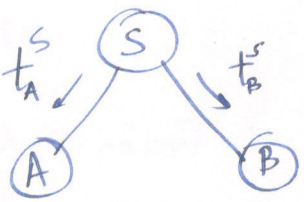
How much information gain (reduction in entropy) happens if the node is split?

Entropy: Measure of randomness (Uncertainty)

Highly uncertain events carry more information

For a set of events $\{e_1\ e_2 \cdots e_N\}$ which occur with probability $\{p_{e_1}\ p_{e_2} \cdots p_{e_N}\}$ the entropy is given as

$$\text{Entropy} = \sum_i p_{e_i} \log \frac{1}{p_{e_i}}$$



Information Gain: $\text{Entropy}(S) - t_A^S \cdot \text{Entropy}(A) - t_B^S \cdot \text{Entropy}(B)$

Probability is computed as the frequency of target labels for the examples in the set.

$x \in \mathbb{R}^d$