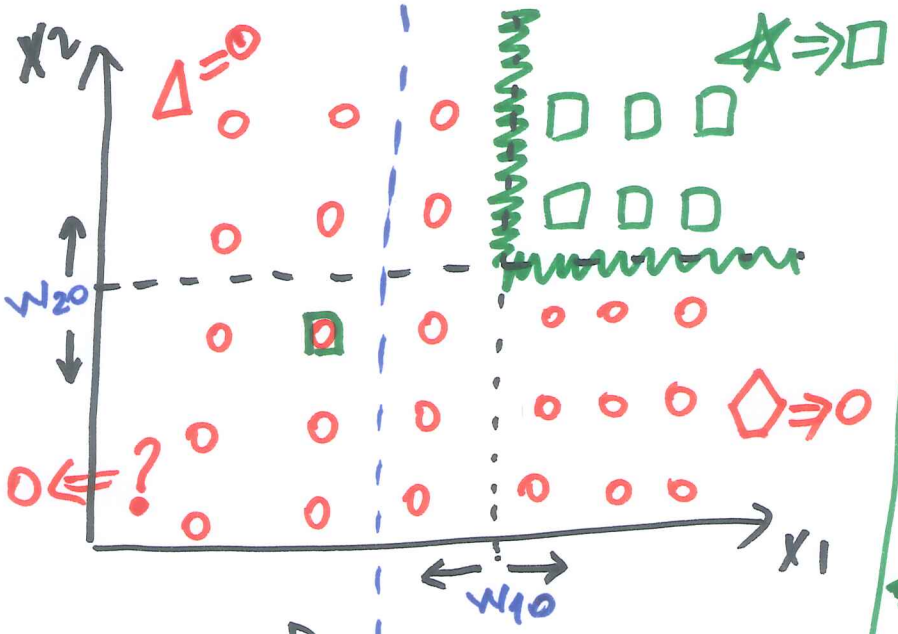
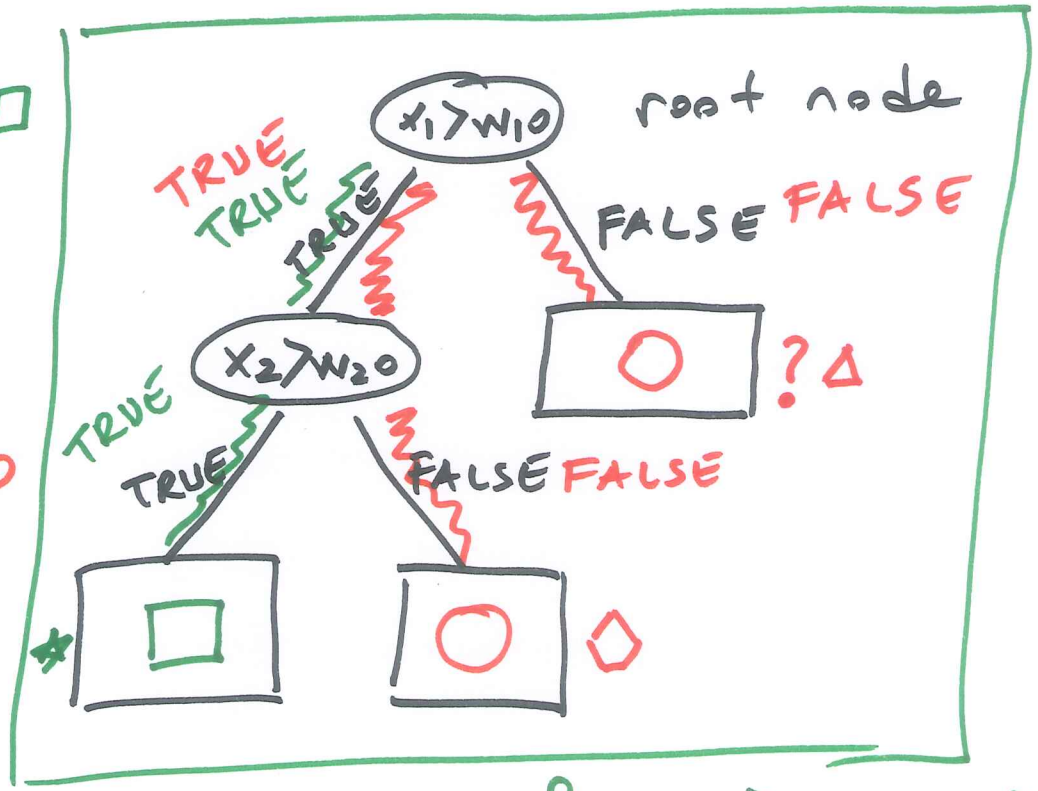
DECISION TREES



$x_i \in \mathbb{R}^D$

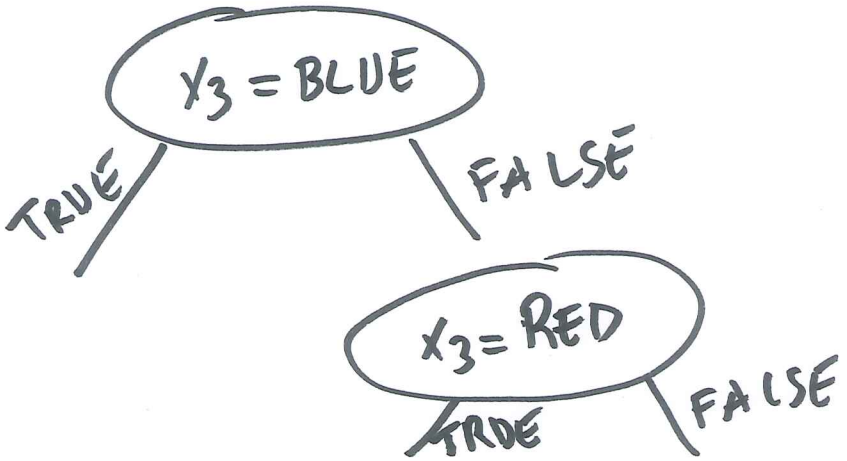
$x_3 = \{$ COLOR, RED, GREEN, BLUE $\}$

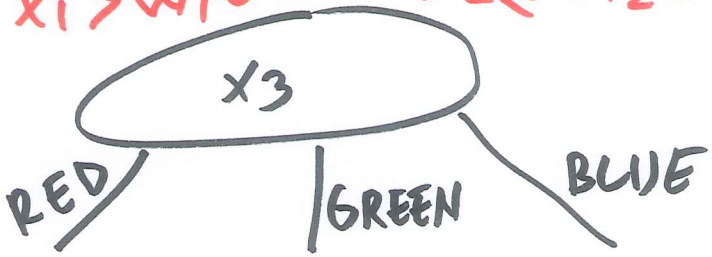


$x_1 > w_{10}$ & $x_2 > w_{20} \Rightarrow$ □

$x_1 \leq w_{10} \Rightarrow$ O

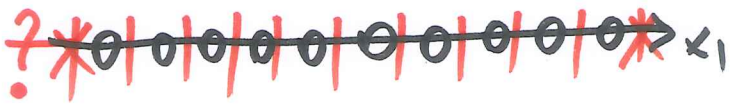
$x_1 > w_{10}$ & $x_2 \leq w_{20} \Rightarrow$ O

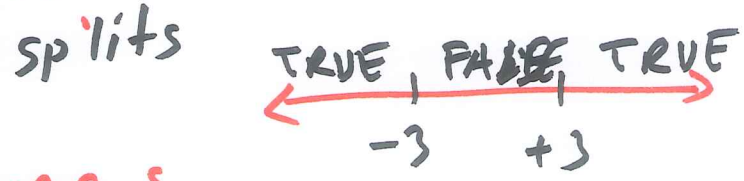


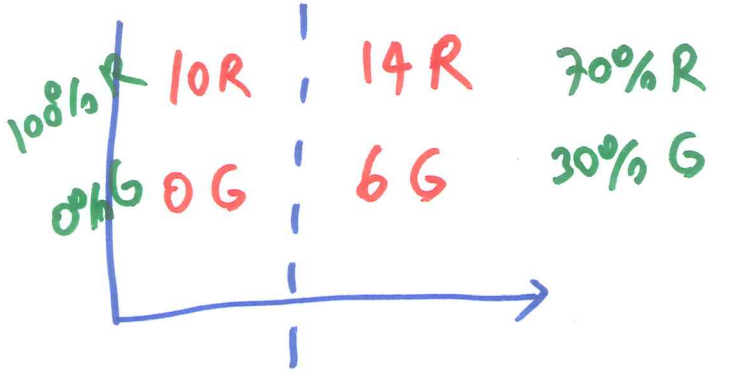


How can we learn w_{10} & w_{20} from data?

N data points $\Rightarrow$ N-1 splits







Univariate Trees

Each internal node $\Rightarrow$ we use only one feature

$$f_m(x): \quad x_j \overset{?}{>} w_{m0} \qquad [x_j = w_{m0}]$$

TRUE $\swarrow$ $\searrow$ FALSE

$$L_m = \{x \mid x_j > w_{m0}\} \qquad R_m = \{x \mid x_j \leq w_{m0}\}$$

$x_j = w_{m0}$

$x_j^2 > 9 \Rightarrow B1$

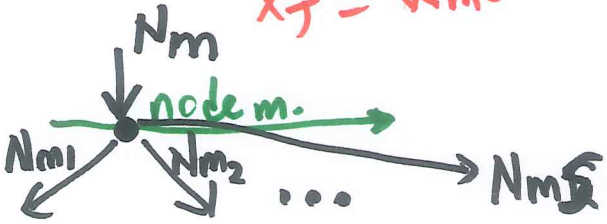
$x_j < -3 \Rightarrow B1$

$-3 \leq x_j \leq 3 \Rightarrow B2$

$x_j > 3 \Rightarrow B3$

$x_j \neq w_{m0}$

Goodness of a split:



N_m: # of data points that reach node m

S: # of splits.

K: # of classes.

$$N_m = N_{m1} + N_{m2} + \dots + N_{mS}$$

$$N_m = N_{m1} + N_{m2} + \dots + N_{mK}$$

$$P_{mc} = \hat{P}(y = c \mid x, m) = \frac{N_{mc}}{N_m}$$

$$I_m = -\sum_{c=1}^{K} p_{mc} \cdot \log_2(p_{mc})$$

node impurity

$$I_m = -\left[\overbrace{1 \cdot \log_2(1)}^{0} + \overbrace{0 \cdot \log_2(0)}^{0}\right] = 0 \qquad I_m = -\left[0.7 \log_2(0.7) + 0.3 \log_2(0.3)\right]$$

$$I_m = -\left[\overbrace{1 \cdot \log_2(1)}^{0} + \overbrace{0 \cdot \log_2(0)}^{0}\right] = 0 \qquad I_m = -\left[0.6 \log_2(0.6) + 0.4 \cdot \log_2(0.4)\right]$$

split impurity

$$I'_m = -\sum_{s=1}^{S} \frac{N_{ms}}{N_m} \sum_{c=1}^{K} p_{msc} \cdot \log_2(p_{msc})$$

- $\frac{N_{ms}}{N_m}$: weight of a child node
- p_{msc}: node index, split index, class index
- $\log_2(p_{msc})$: impurity of a child node

$$I'_m(\text{blue split}) = -\left[\frac{10}{30} \cdot 0 + \frac{20}{30} \cdot \right]$$

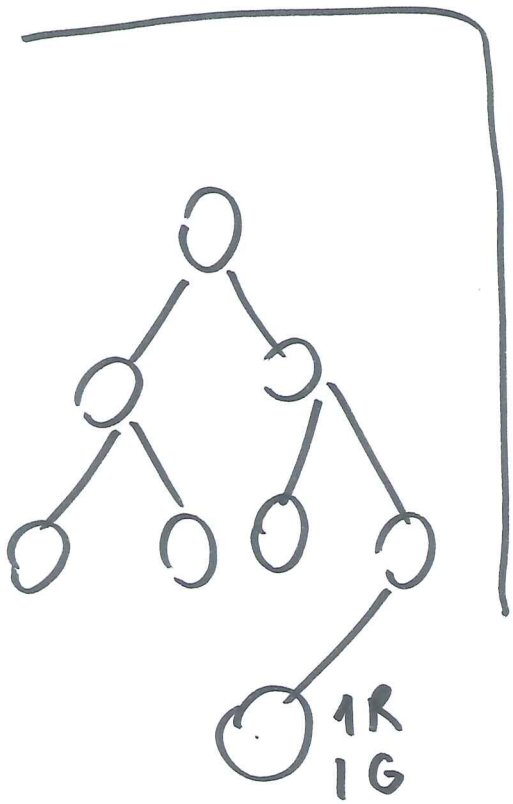
$$I'_m(\text{black split}) = -\left[\frac{15}{30} \cdot 0 + \frac{15}{30} \cdot \right]$$

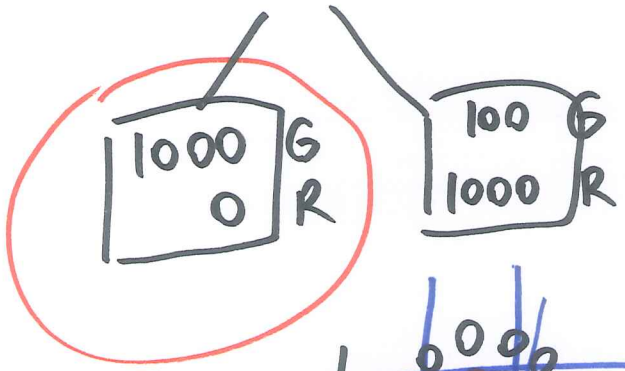
black split is better than blue split!!!

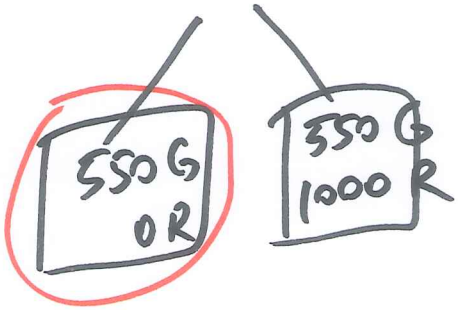
at each node ⇒

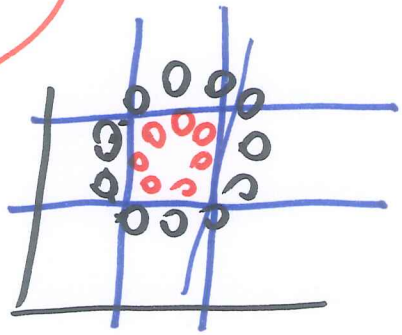
- for all features
  - for all possible splits
    - calculate the impurity
- pick the best split

Stop when all terminal nodes are "pure".



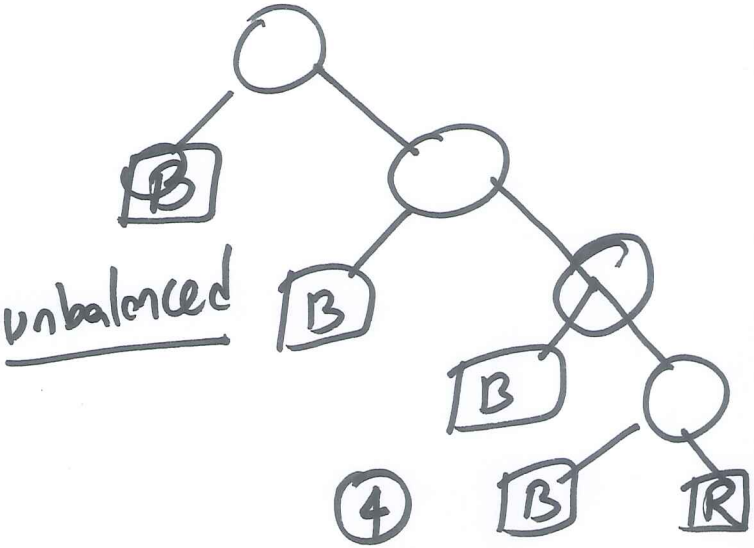






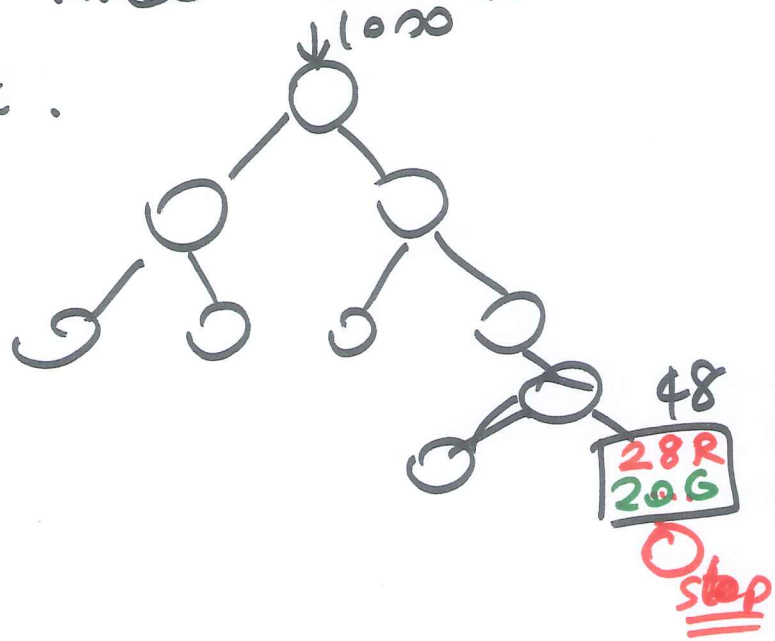
PROBLEM: (OVERFITTING)

pruning is the answer to this problem.

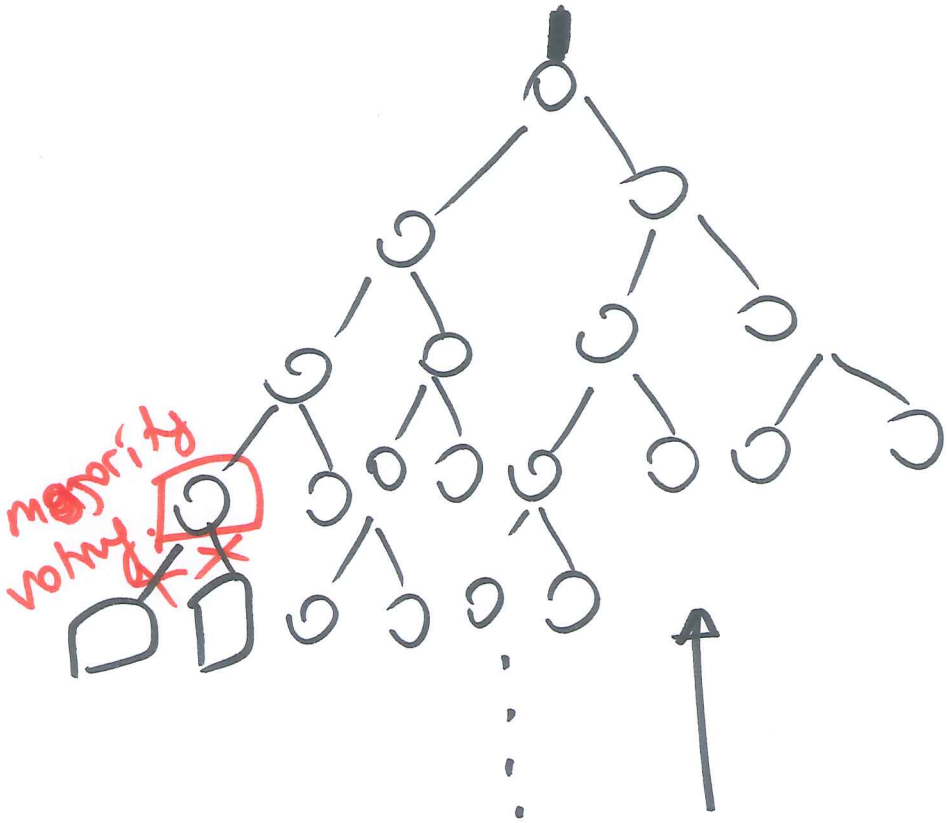


unbalanced

pre-pruning: you won't split if your node has at most 5% of the training set.



post-pruning: you grow your tree until it is completely pure.



⇐ pruning algorithm
training set] %80
pruning set] %20

if your misclassification error does not increase after cutting, CUT these branches.