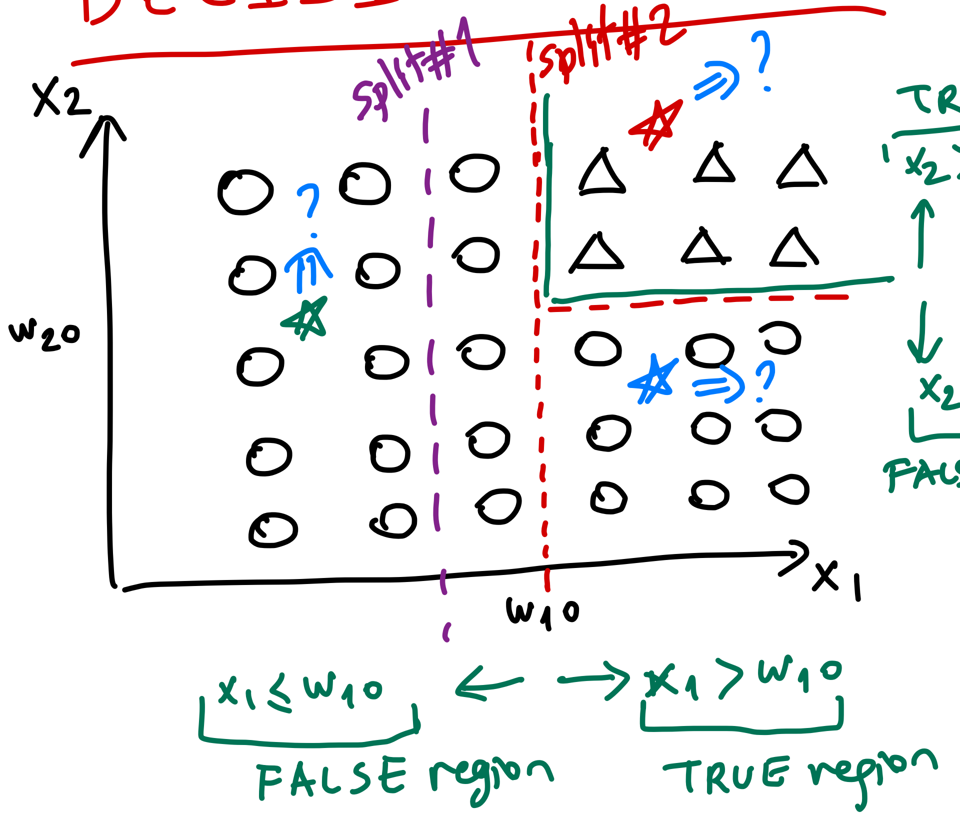
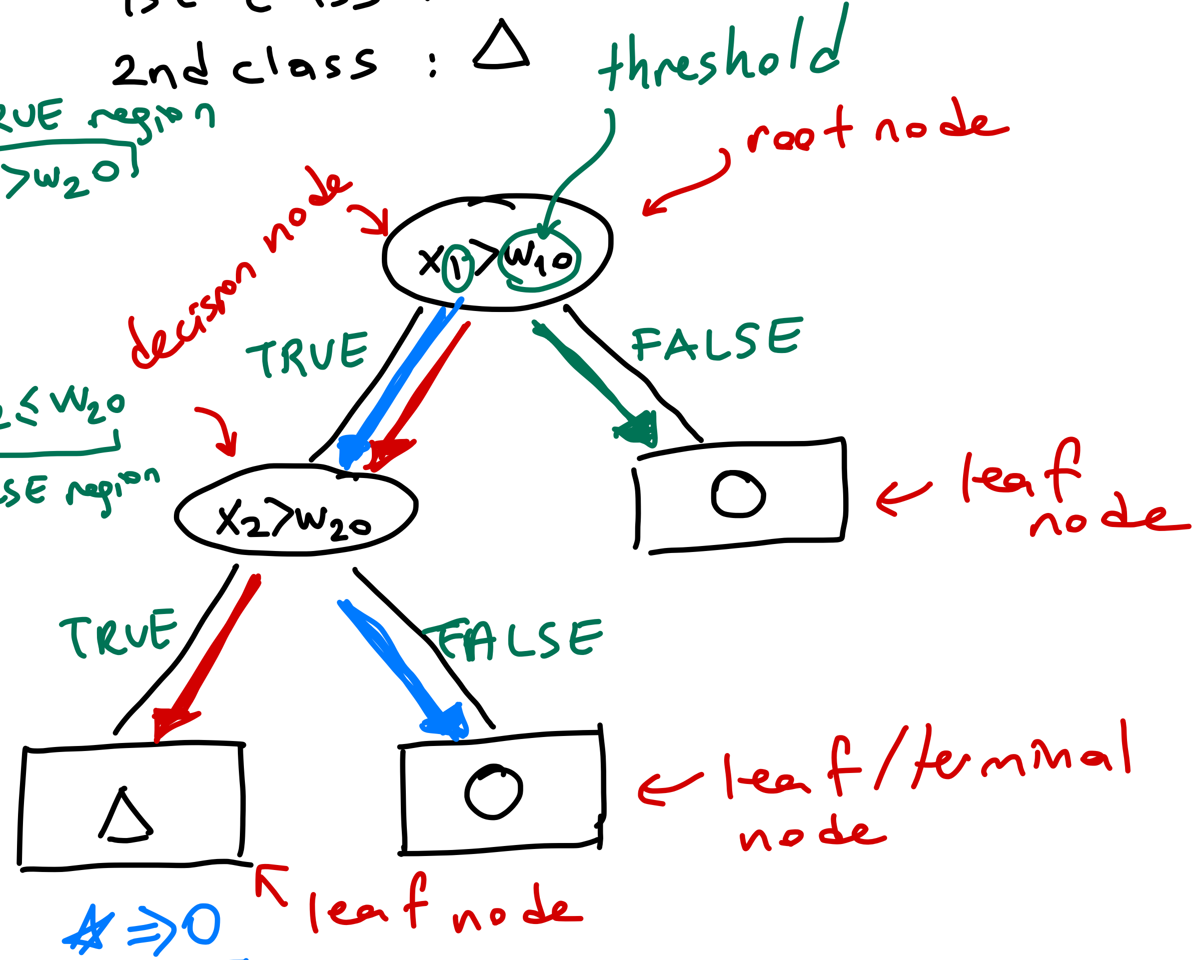
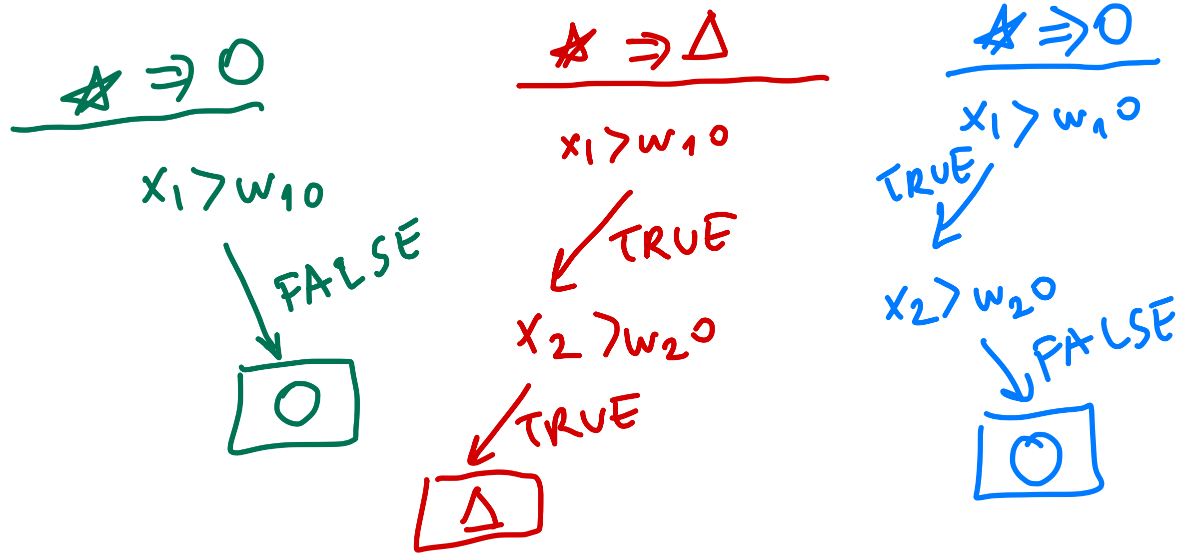
DECISION TREES

1st class : O

2nd class : Δ





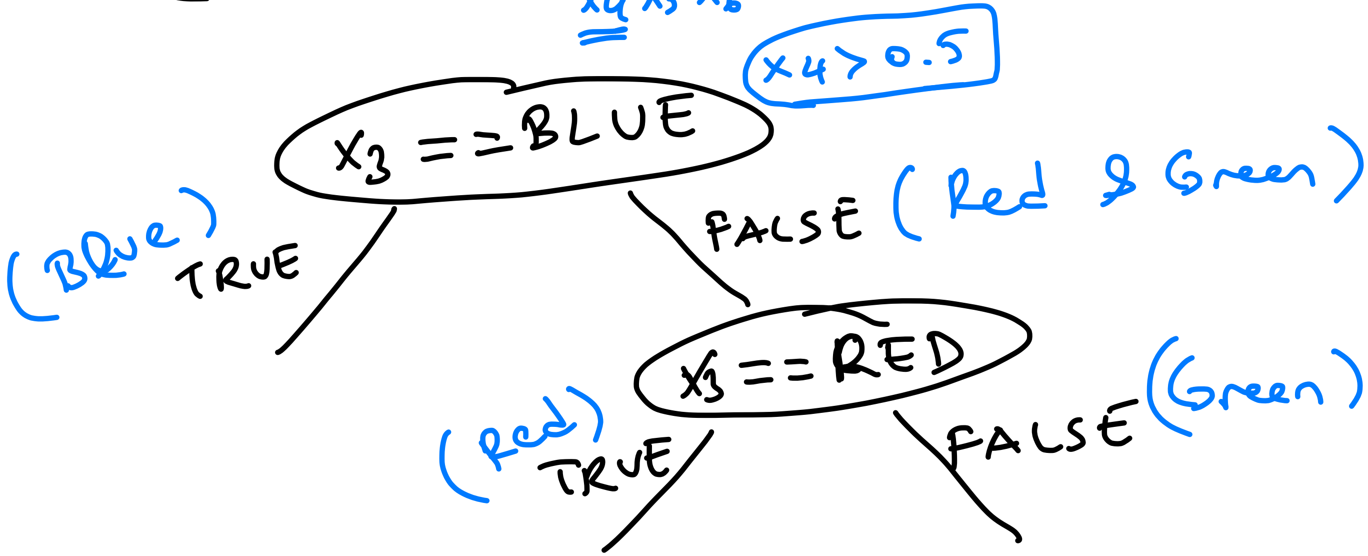


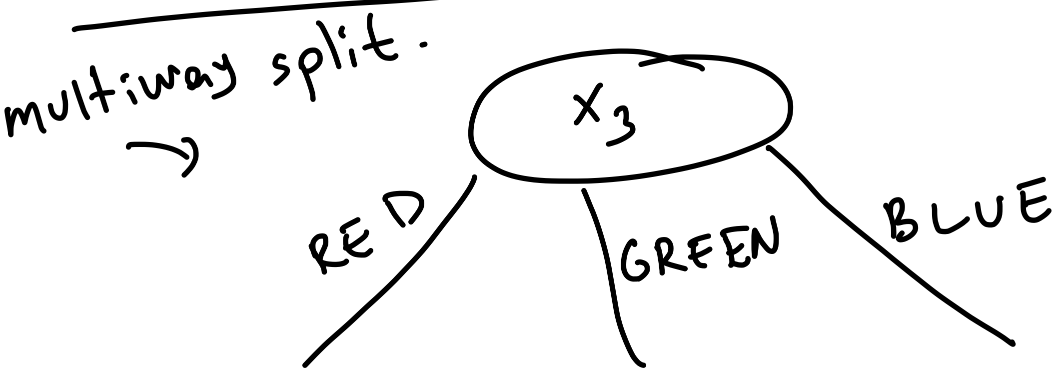
1. if $x_1 \le w_{10} \Rightarrow \hat{y} = O$
2. if $x_1 > w_{10}$ & $x_2 > w_{20} \Rightarrow \hat{y} = \Delta$
3. if $x_1 > w_{10}$ & $x_2 \le w_{20} \Rightarrow \hat{y} = O$

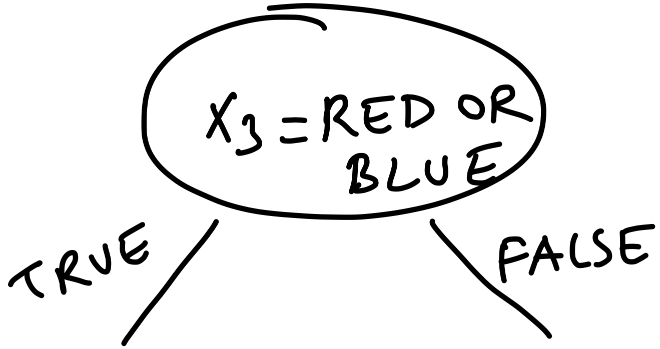
$X_3 = \begin{Bmatrix} RED \\ GREEN \\ BLUE \end{Bmatrix}$ $\begin{matrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ x_4\,x_5\,x_6 \end{matrix}$

$X_3 > GREEN$ (X)

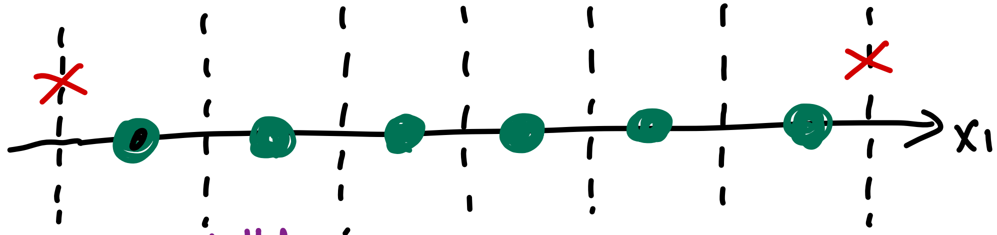
$x_3 \leq RED$ (X)

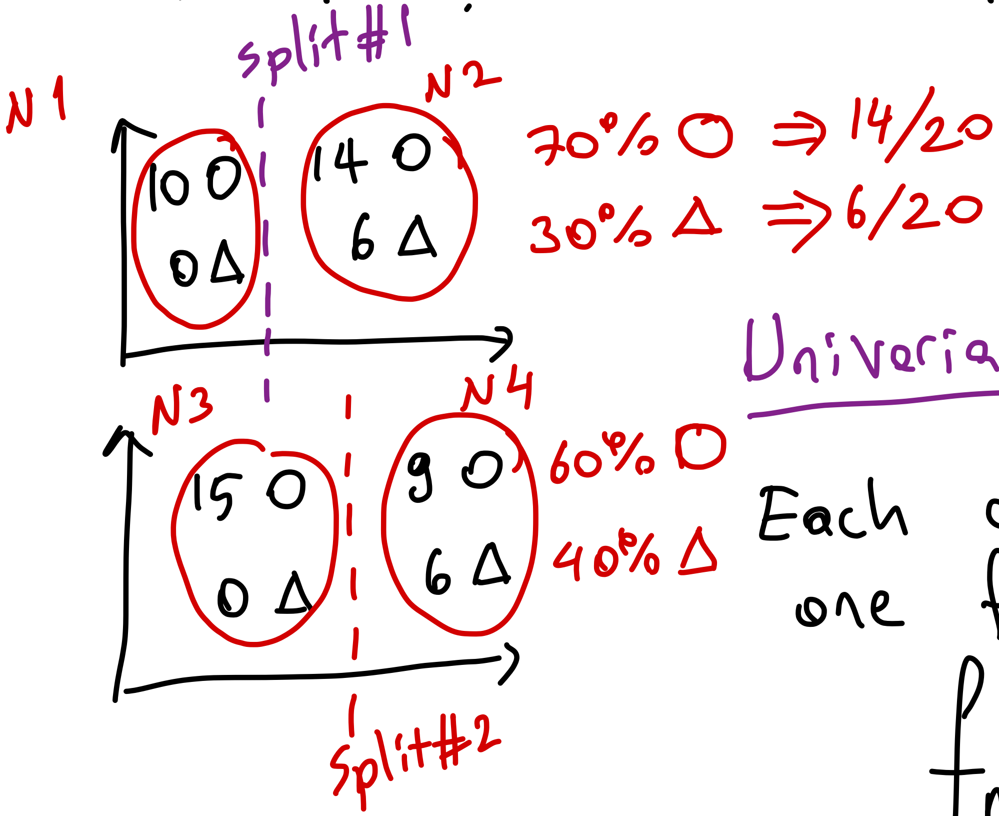






How can we learn on which feature and where to split? (learning)





N data points ⇒ (N−1) possible splits

D features ⇒ D × (N−1) possible splits in total.

Univariate Trees

Each decision (internal) node uses only one feature.

$$f_m(x) : x_j > w_{m0}$$

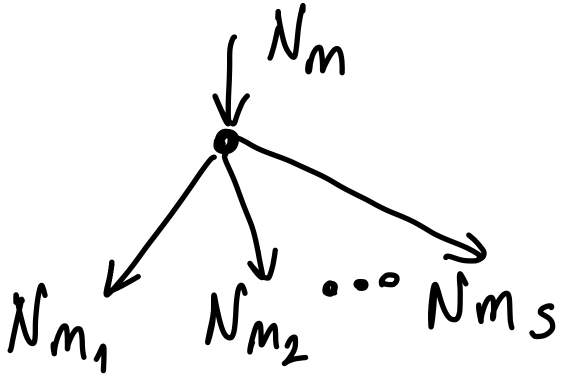
discrete $[x_j == w_{m0}]$

TRUE / FALSE

left child → $L_m = \{x \mid x_j > w_{m0}\}$ $[x_j == w_{m0}]$

right child $R_m = \{x \mid x_j \leq w_{m0}\}$ $[x_j \neq w_{m0}]$

Goodness of a split: Is split #2 better than split #1?



S = # of splits

N_m = # of data points that reach to node m

K = # of classes

$$\overbrace{N_m} = N_{m,1} + N_{m,2} + \cdots + N_{m,S} \qquad N_m = \sum_{s=1}^{S} N_{m,\underline{\underline{s}}} \quad \text{(splits)}$$

$$N_m = N_{m,1} + N_{m,2} + \cdots + N_{m,K} \qquad N_m = \sum_{c=1}^{K} N_{m,\underline{\underline{c}}} \quad \text{(classes)}$$

$$P_{mc} = \hat{P}(y = c \mid x_m) = \frac{N_{m,c}}{N_m} \qquad 0 \cdot \log_2(0) \equiv 0$$

$$I_m = -\sum_{c=1}^{K} P_{mc} \log\left[P_{mc}\right]$$

N1 $\Rightarrow I_m = -\left[\frac{10}{10} \cdot \underbrace{\log_2\left[\frac{10}{10}\right]}_{0} + \frac{0}{10} \cdot \underbrace{\log_2\left[\frac{0}{10}\right]}_{0}\right] = 0$

N2 $\Rightarrow I_m = -\left[\frac{14}{20} \log_2\left[\frac{14}{20}\right] + \frac{6}{20} \log_2\left[\frac{6}{20}\right]\right] = +0.8813$

$$N_3 \Rightarrow I_m = -\left[\frac{15}{15}\log_2\left(\frac{15}{15}\right) + \frac{0}{15}\log_2\left[\frac{0}{15}\right]\right] = 0$$

$$N_4 \Rightarrow I_m = -\left[\frac{9}{15}\log_2\left(\frac{9}{15}\right) + \frac{6}{15}\log_2\left(\frac{6}{15}\right)\right] = +0.9710$$

$$I'_m = \underbrace{-\sum_{s=1}^{S} \frac{N_{m,s}}{N_m} \underbrace{\left[\sum_{c=1}^{K} p_{msc}\log_2\left(p_{msc}\right)\right]}_{\text{impurity of a child node}}}_{\text{impurity of the split}}$$

class index

split index

node index

$$I'_m(\text{split }\#1) = -\left[\frac{10}{30}\cdot\overbrace{[0]}^{\text{impurity of node 1}} + \frac{20}{30}\cdot[-0.8813]\right] = 0.5875$$

$$I'_m(\text{split }\#2) = -\left[\frac{15}{30}\cdot[0] + \frac{15}{30}\cdot[-.9710]\right] = \underbrace{0.4855}_{\text{smaller}}$$

Split #2 is better than Split #1.

⇒ at each internal (decision) node ⇒

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

⇒ Stop when all terminal nodes are "pure"

POSSIBLE PROBLEM! (OVERFITTING)

Training accuracy is 100%!

PRUNING

① Prepruning

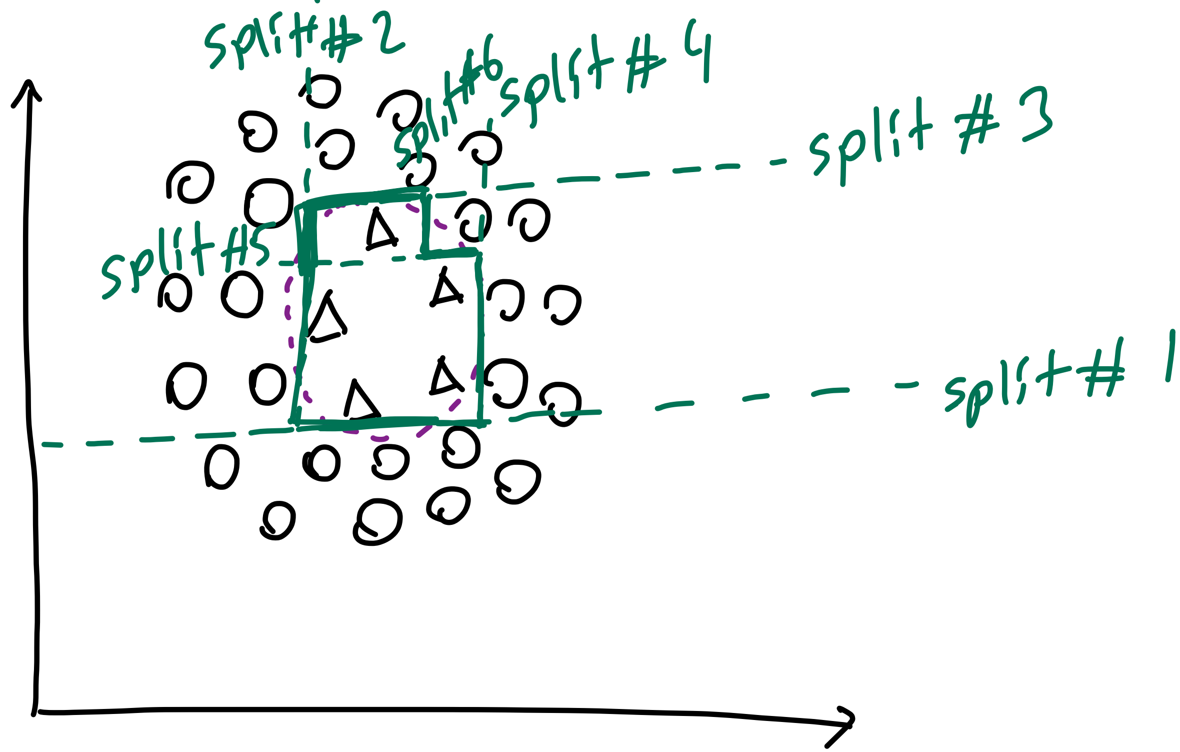
a1) - fix the maximum depth

a2) - if you reach this depth, stop

b) - You won't split if your node has a specified amount of your dataset.

② Postpruning

- grow your tree until it is completely pure
- prune your tree step by step until your misclass. error starts increasing on a validation dataset.


split # 2
split #6
split # 4
split # 3
split #5
split # 1