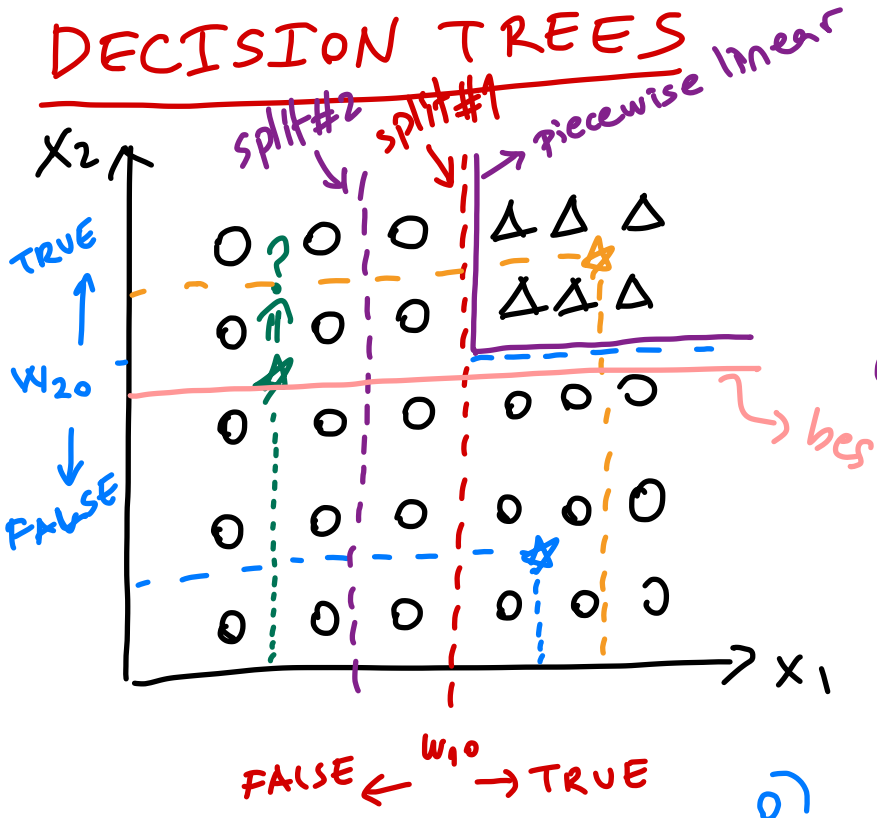
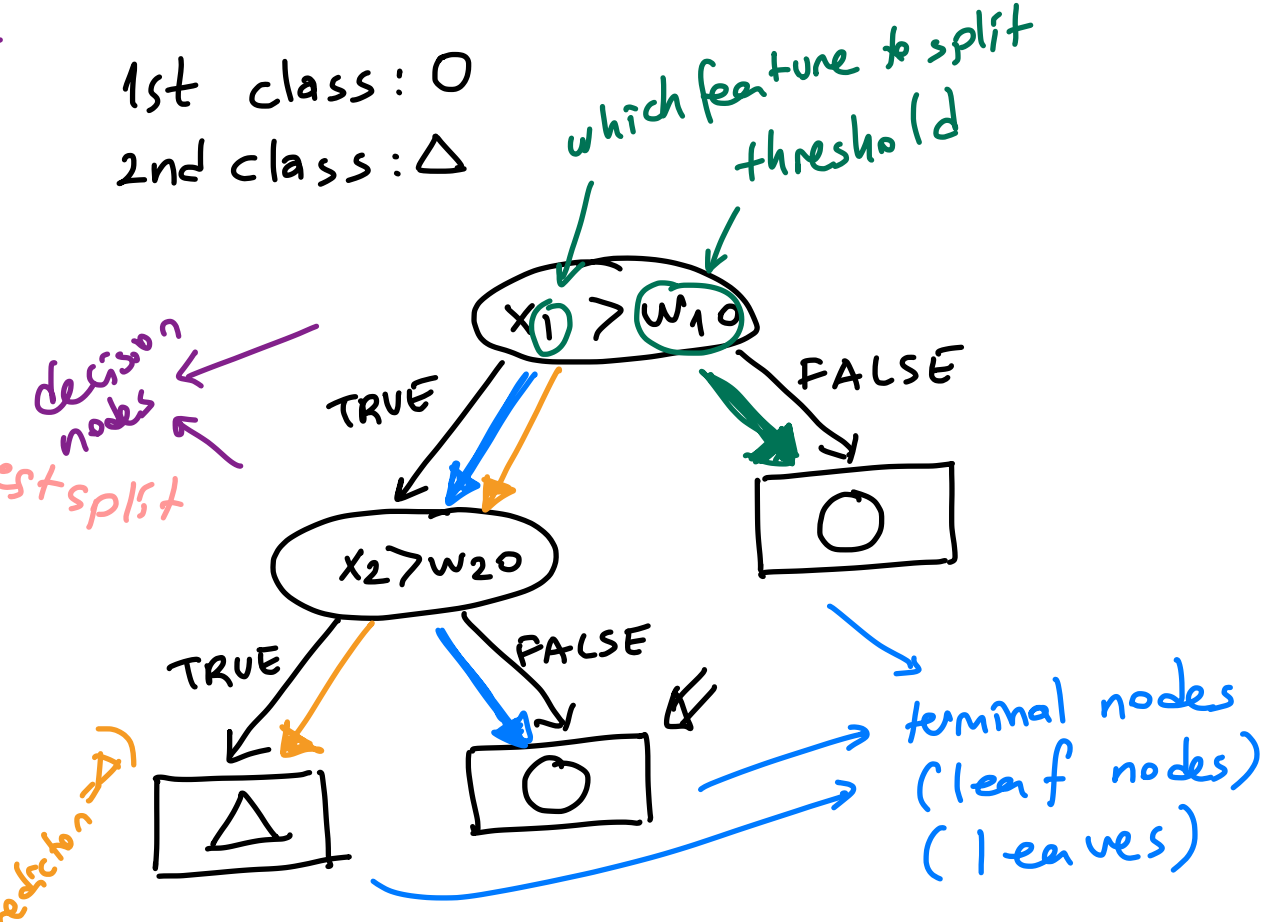
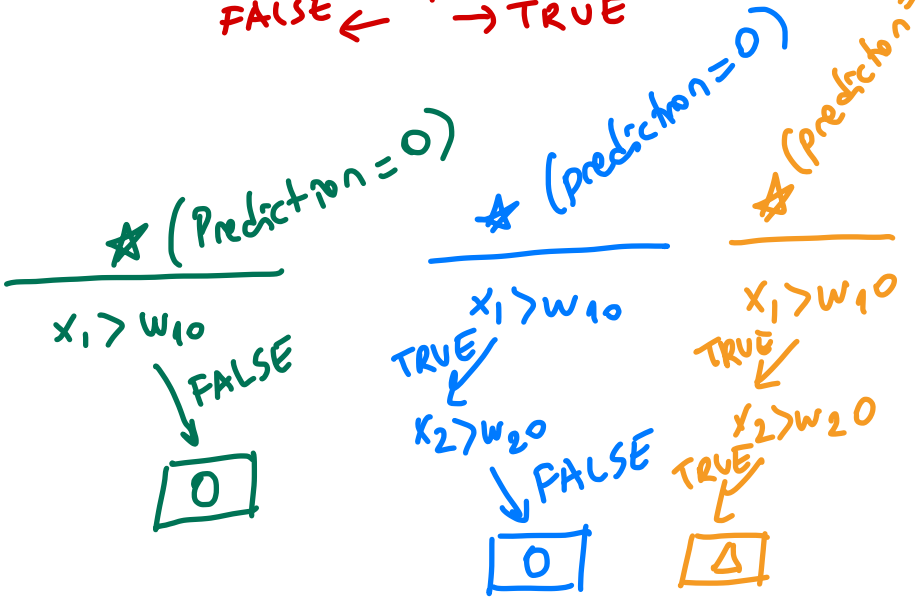
DECISION TREES



1st class : O
2nd class : △





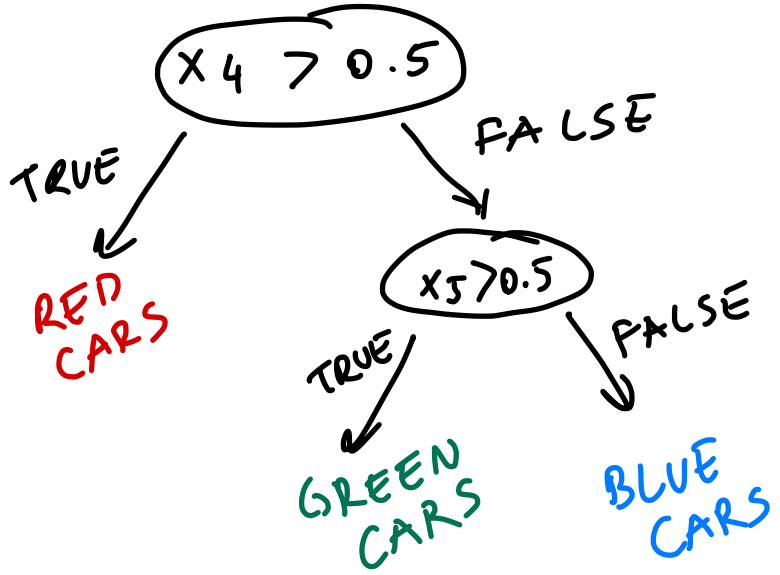
① if $x_1 \leq w_{10} \Rightarrow \hat{y} = O$

② if $x_1 > w_{10} \wedge x_2 \leq w_{20} \Rightarrow \hat{y} = O$

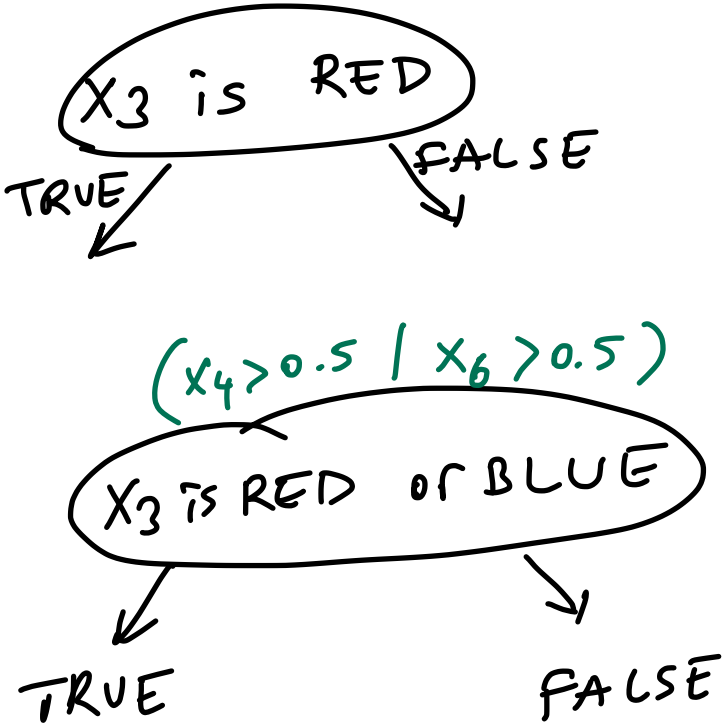
③ if $x_1 > w_{10} \wedge x_2 > w_{20} \Rightarrow \hat{y} = \triangle$

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

R	G	B
x_4	x_5	x_6
1	0	0
0	1	0
0	0	1

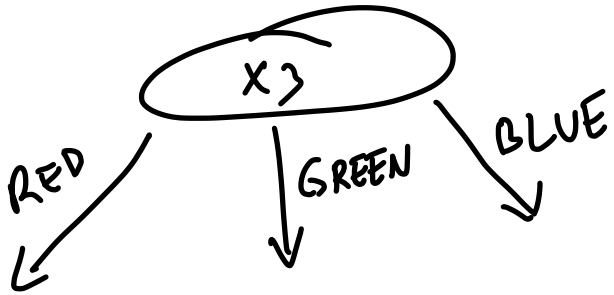


$\equiv$

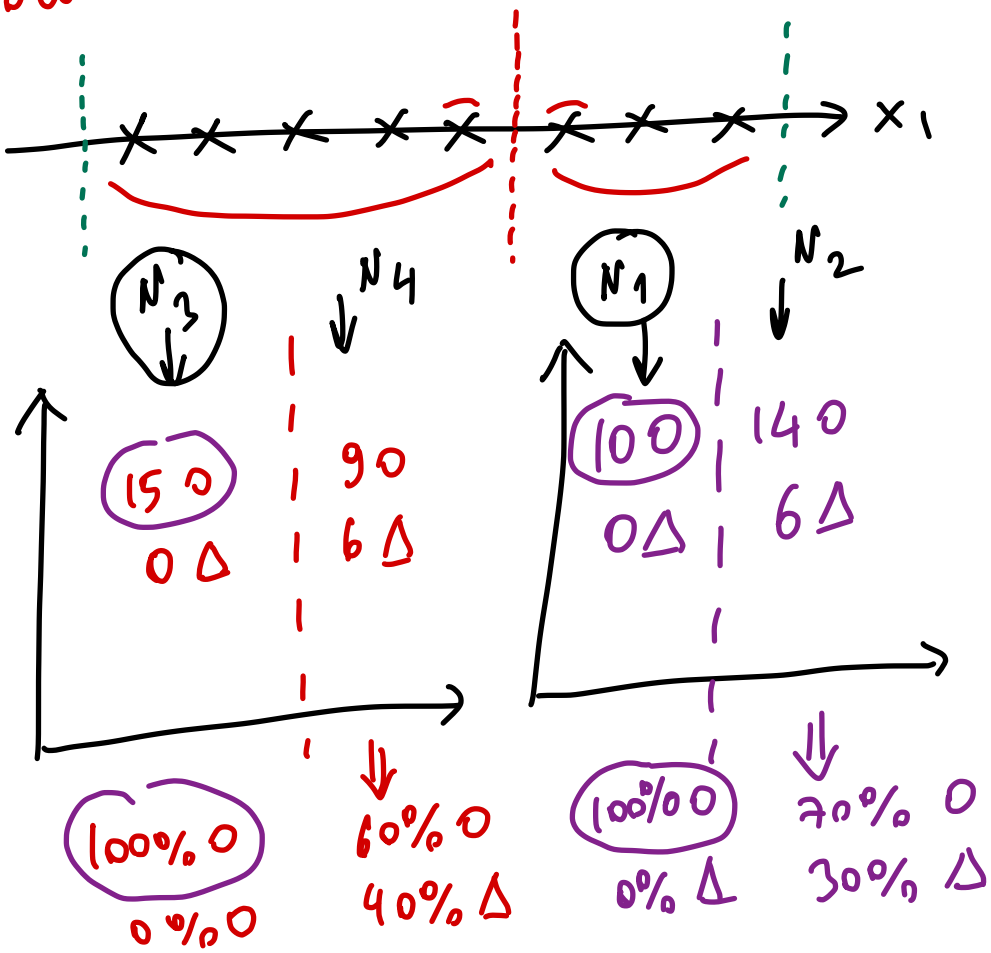


or

multiway split



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



N data points $\Rightarrow$ (N-1) possible splits

D features $\Rightarrow$ D(N-1) possible splits

Univariate Trees

Each decision node uses only one feature.

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

TRUE

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

Left child

$x_j = w_{m0}$

FALSE

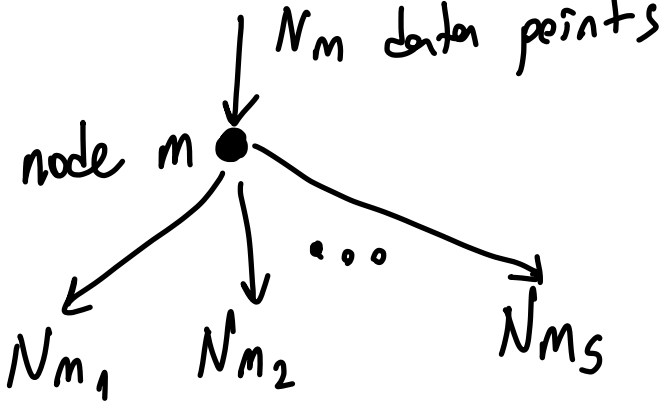
$R_m = \{x \mid x_j \leq w_{m0}\}$

Right child

$w_j \neq w_{m0}$

Goodness of a split

Is split #1 is better than Split#2?



S = # of splits (branches)

N_m = # of data points that reach node m

K = # of classes

$$N_m = N_{m_1} + N_{m_2} + \dots + N_{m_S} \qquad N_m = \sum_{s=1}^{S} N_{m_s} \quad \text{(splits)}$$

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

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

$0 \cdot \log_2(0) \equiv 0$

$$I_m = -\sum_{c=1}^{K} p_{mc} \log_2(p_{mc}) \quad \Leftrightarrow \quad 0 \cdot \log_2(0)$$

10 ○ 0 △

$$N_1 \Rightarrow I_m = -\left[\frac{10}{10}\cdot\log_2\left(\frac{10}{10}\right) + \frac{0}{10}\cdot\log_2\left(\frac{0}{10}\right)\right] = 0$$

14 ○ 6 △

$$N_2 \Rightarrow I_m = -\left[\frac{14}{20}\log_2\left(\frac{14}{20}\right) + \frac{6}{20}\log_2\left(\frac{6}{20}\right)\right] = 0.8813$$

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

9 (6, 3) $N_4 \Rightarrow I_m = -\left[\frac{9}{15}\log_2\left(\frac{9}{15}\right) + \frac{6}{15}\log_2\left(\frac{6}{15}\right)\right] = 0.9710$

$$I_m' = \sum_{s=1}^{S} \frac{N_{m,s}}{N_m}\left[-\sum_{c=1}^{K} P_{msc}\log_2\left(P_{msc}\right)\right]$$

I_m': impurity

$\Leftrightarrow$

$\frac{N_{m,s}}{N_m}$: weights.

$-\sum_{c=1}^{K} P_{msc}\log_2\left(P_{msc}\right)$: impurity of a child node

$\sum_{s=1}^{S}$: impurity of the split

P_{msc}: class index, split index, node index

$$I_m'(\text{split \#1}) = \left[\frac{15}{30}\cdot[0] + \frac{15}{30}\cdot[0.9710]\right] = 0.4855$$

minimum is better.

$$I_m'(\text{split \#2}) = \left[\frac{10}{30}[0] + \frac{20}{30}[0.8813]\right] = 0.5875$$

Split #1 is better than Split #2.

⇒ at each internal (decision) node

⇒
- for all features
  - for all possible splits
    - calculate impurity
- pick the best split among all possible splits (the one with minimum impurity)

⇒ Stop when all terminal nodes are "pure"

POSSIBLE PROBLEM ⇒ OVERFITTING
(Training accuracy is 100%)

PRUNING

① Prepruning

(A)
- fix the maximum depth
- if you reach this depth, stop

(B)
- you will not split if your node has a specified amount of your data set

② Post pruning

- grow your tree until it is completely pure.
- prune your tree step by step until your misclassification error starts increasing on a validation data set.