< S- algorithm >

(y=1)

x_1 = < Sunny, warm, normal, Strong, warm, same >

x_2 = < Sunny, warm, normal, light, warm, same >

x_3 = < Sunny, warm, normal, Strong, warm, change >

h_0 = < ∅, ∅, ∅, ∅, ∅, ∅ >

h_1 = < Sunny, warm, normal, Strong, warm, same >

h_{12} = < Sunny, warm, normal, ?, warm, same >

x_1, x_2 제일 U

⇓

Strong, light 중 하나 O

h_{123} = < Sunny, warm, normal, ?, warm, ? >

$x_1 x_2 x_3$

same, change

< Candidate elimination Algorithm >

S: specific, G: generally S = < ∅ ... ∅ >

G = < ? ... ? >

Set G 작아 ↓ error 안나올까

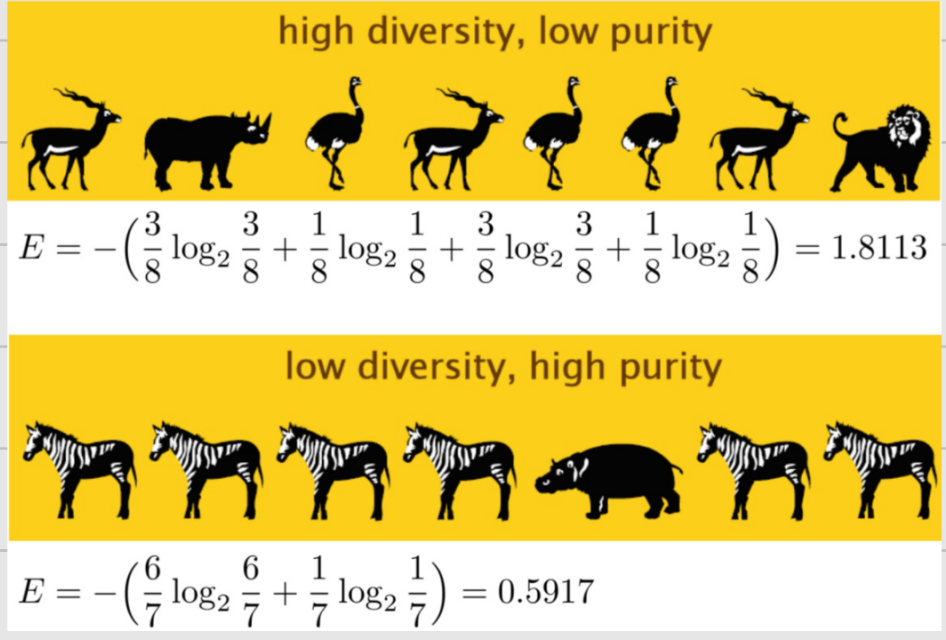
<Decision Tree>

나누는 기준 → 자식노드 가 pure 하게

즉 자식노드의 impurity는 부모노드보다 작음

Impurity Measure

↳ $E = -\sum_{j=1}^{J} p_j \log_2 p_j$

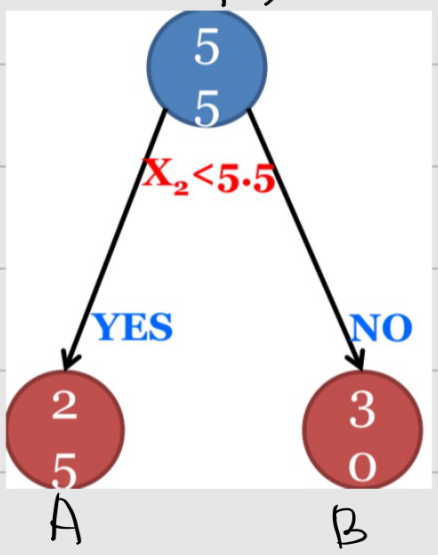


X1	X2	Class
1	4	1
2	6	1
2	5	1
2	4	2
2	3	2
3	6	1
4	6	1
4	5	2
4	4	2
5	4	2

Class 1 : 5개

Class 2 : 5개

<1>



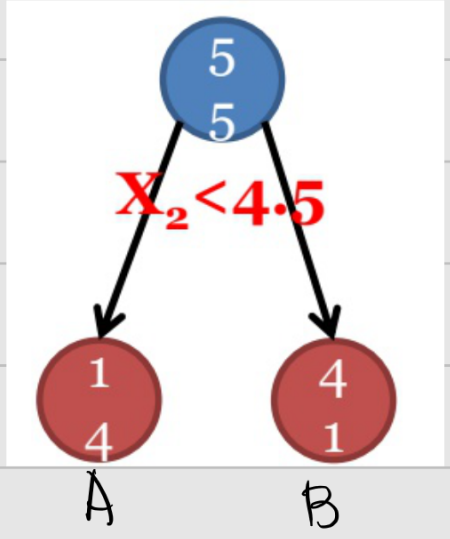
초기 $E = -\left(\frac{5}{10}\log_2\frac{5}{10} + \frac{5}{10}\log_2\frac{5}{10}\right) = 1$

A의 $E = -\left(\frac{5}{7}\log_2\frac{5}{7} + \frac{2}{7}\log_2\frac{2}{7}\right) = 0.86$

B의 $E = -\left(\frac{3}{3}\log_2 1\right) = 0$

이후 가중평균 $\Rightarrow \frac{7}{10} \times 0.8631 + \frac{3}{10} \times 0 = 0.604$

<2>



A의 E : 0.7219

B의 E : 0.7219

$E = \frac{1}{2} \times 2 \times 0.7219 = 0.7219$

1번의 E가 더 작음 → impurity ↓

이처럼 계속 반복 leaf node가 pure 할때까지

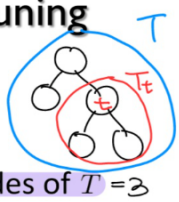
↓

fully grown tree

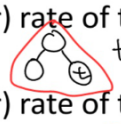
Pruning 이 필요한 이유

↳ Why? Overfitting

Minimal Cost-Complexity Pruning



- T : a final tree with leaf nodes
- T_t : a subtree which root node is the node *t*
- $|T|$: complexity of tree T = number of leaf nodes of T =3
- $|T_t|$: complexity of subtree T_t = number of leaf nodes of T_t =2
- $R(t)$: misclassification (error) rate of the tree T, when the node *t* is claimed as a leaf node.



t 밑 2개 버리기

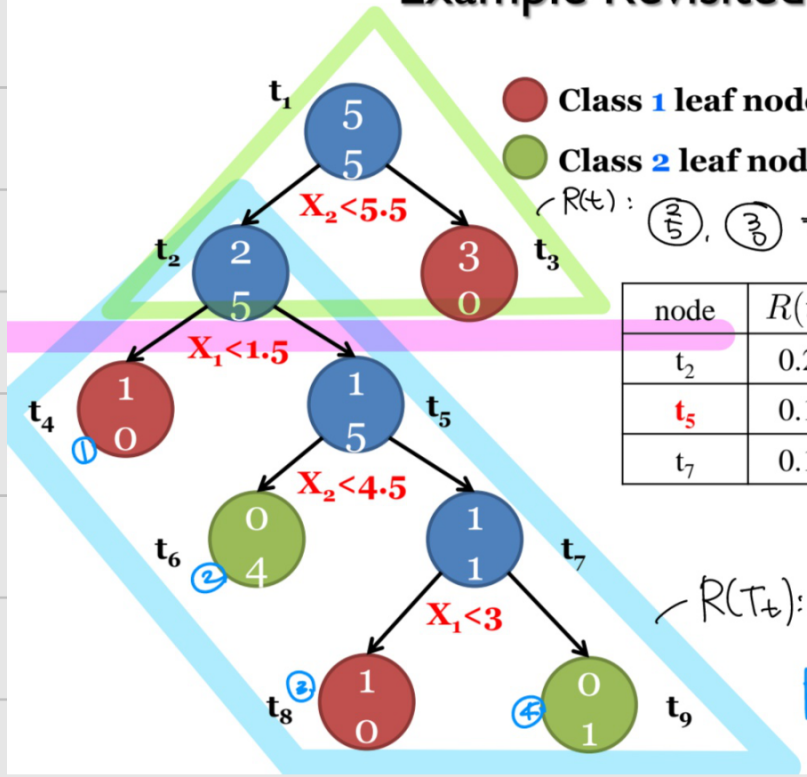
- $R(T_t)$: misclassification (error) rate of the subtree T_t itself

- Prune the tree at the node *t* when $\alpha(t)$ is minimal.

$$\alpha(t) = \frac{R(t) - R(T_t)}{|T_t| - 1}$$

↓ ↑

Example Revisited



Class 1 leaf node

Class 2 leaf node

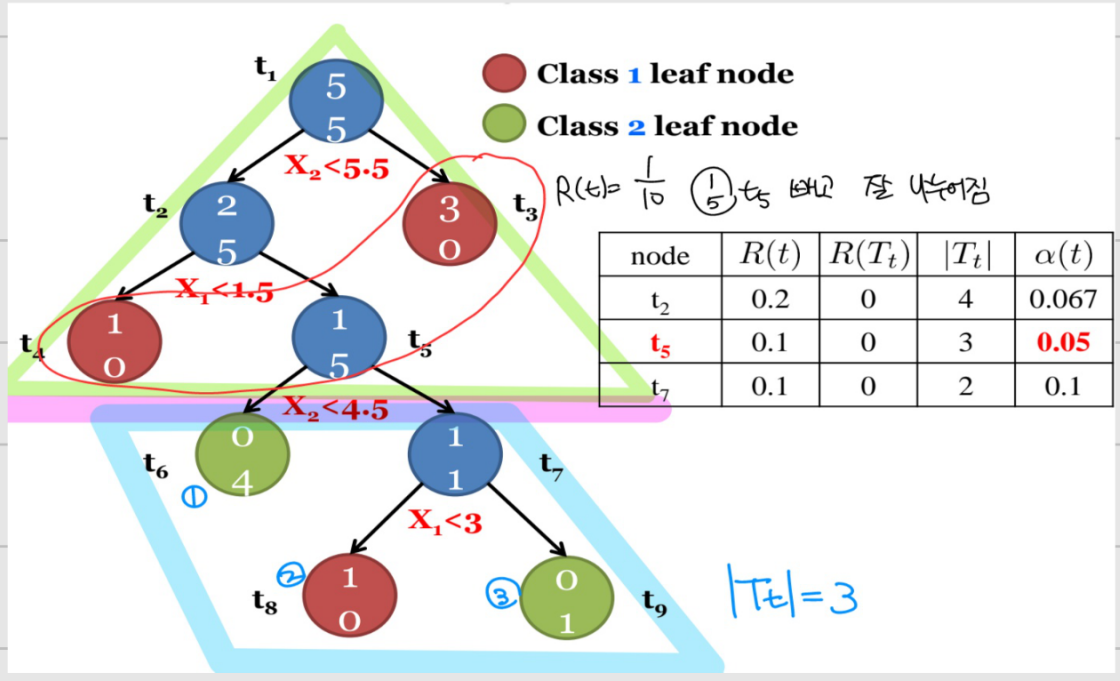
R(t) : (2/5), (3/0) → 2개 틀림 ∴ 2/10 = 0.2

node	$R(t)$	$R(T_t)$	$\lvert T_t \rvert$	$\alpha(t)$
t_2	0.2	0	4	0.067
t_5	0.1	0	3	**0.05**
t_7	0.1	0	2	0.1

$R(T_t)$: leaf 노드들 모두 틀리지 X

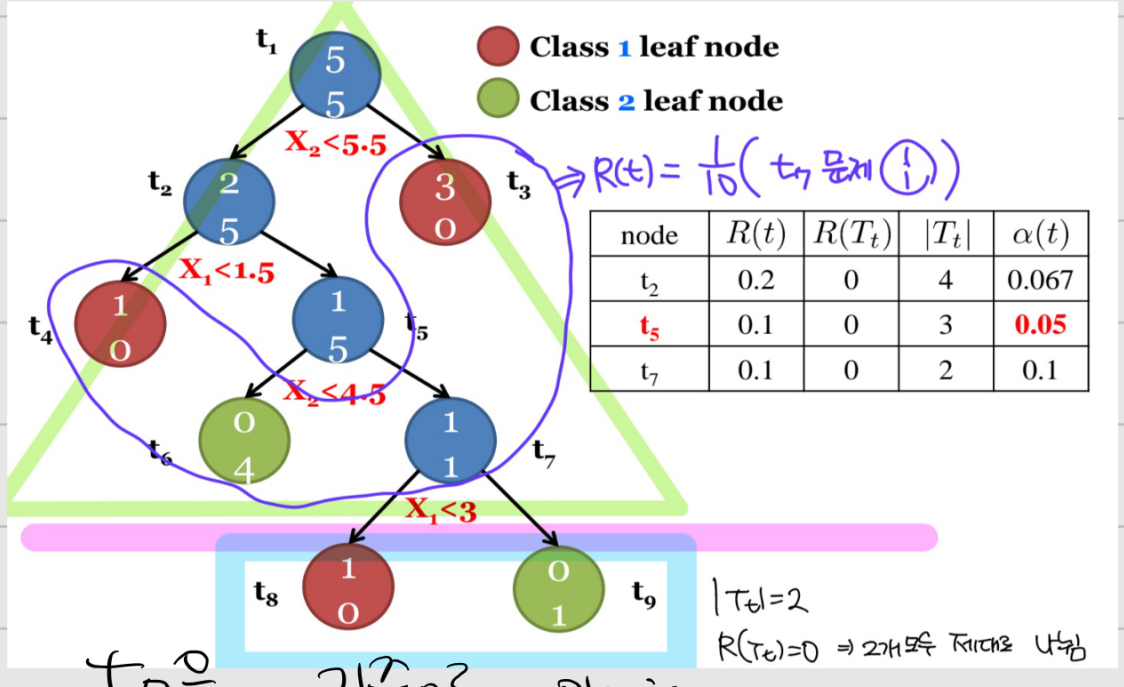
$|T_t| = 4$

t_2를 기준으로 pruning



node	$R(t)$	$R(T_t)$	$\lvert T_t\rvert$	$\alpha(t)$
t_2	0.2	0	4	0.067
t_5	0.1	0	3	**0.05**
t_7	0.1	0	2	0.1

t_5를 기준으로 pruning



node	$R(t)$	$R(T_t)$	$\lvert T_t\rvert$	$\alpha(t)$
t_2	0.2	0	4	0.067
t_5	0.1	0	3	**0.05**
t_7	0.1	0	2	0.1

t_7을 기준으로 pruning

α가 가장 작은 지점으로 pruning

언제까지 pruning?