Artificial Intelligence (AI)

Lec07: Decision Tree

충북대학교

문성태 (지능로봇공학과)

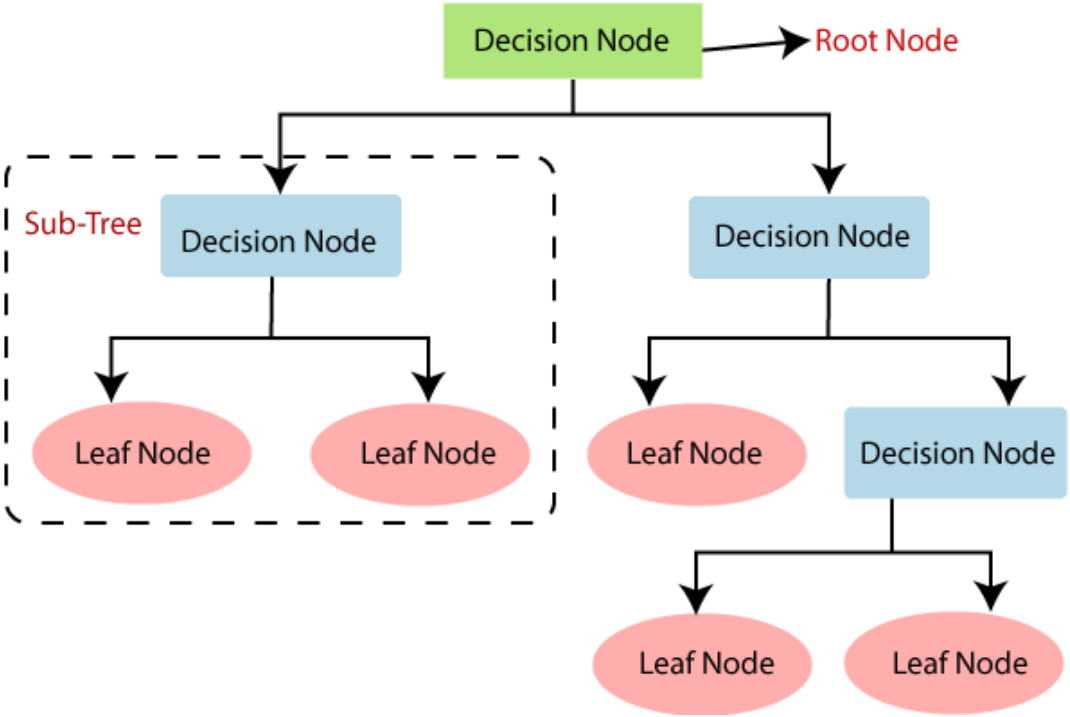
stmoon@cbnu.ac.kr

01

Introduction

What is Decision Tree

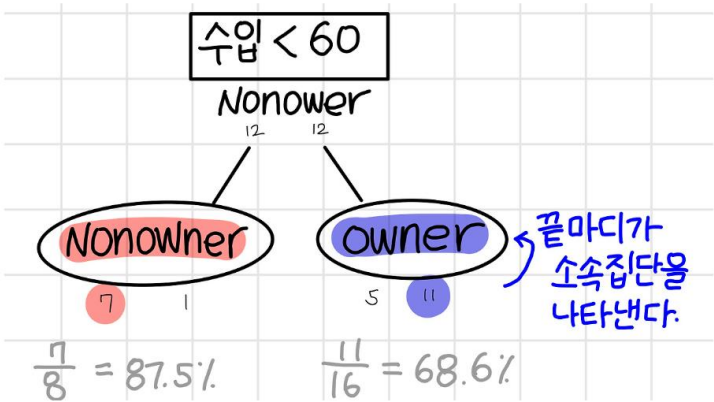
❖ Decision Tree = 스무 고개



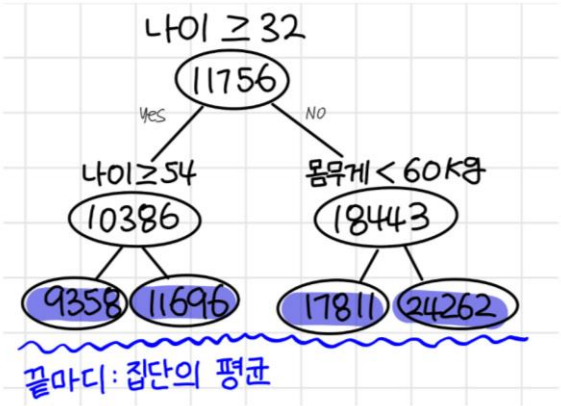
장단점

- Procs
  - 쉽고 직관적
  - No pre-processing (like scaling or normalization)
- Cons
  - overfitting

구분



범주형 변수

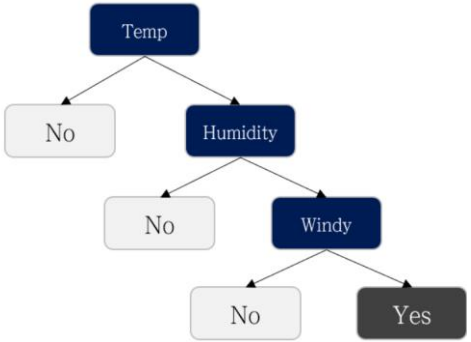


수치형 (연속형) 변수

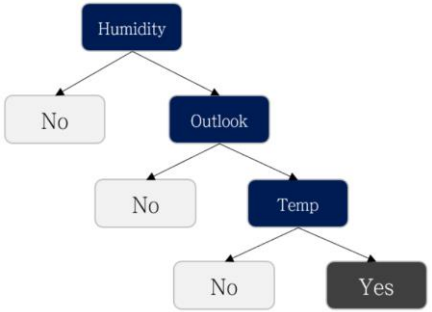
Which model is better?

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rain	mild	high	FALSE	Yes
rain	cool	normal	FALSE	Yes
rain	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rain	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rain	mild	high	TRUE	No

Which model is better?



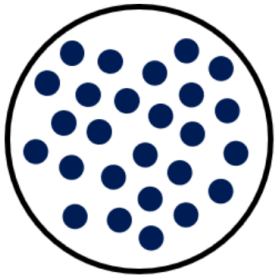
VS



변별력이 좋은 질문부터 하자

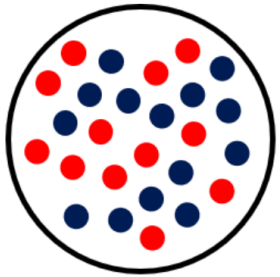
Impurity (불순도)

❖ 불순물이 포함된 정도 ➔ 변별력 구분 기준



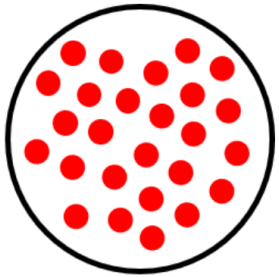
불순도 = 0

변별력이 좋다



불순도 〉 0

변별력이 좋지 않다



불순도 = 0

변별력이 좋다

➔ 최고의 성능을 보이는 Decision Tree를 만들기 위해서는 불순도가 가장 낮은 지표를 찾아 Tree를 구성하자

Decision Tree Algorithm

- Algorithm based on Entropy
  - ID3
  - C4.5

- Algorithm based on Gini index
  - CART (Classification And Regression Tree)

02

ID3 (Iterative Dichotomiser 3)

*) Dichotomiser는 "이분하다"라는 프랑스어

Impurity based on Entropy

- 정보를 정량화하는 단위
  - 정보가 클수록 엔트로피가 크고, 정보가 작으면 엔트로피가 작다.

$$H = \sum(\text{사건 발생확률}) \cdot \log_2\left(\frac{1}{\text{사건 발생확률}}\right)$$

*) unit of H: bit

$$= \sum_i p_i \log_2\left(\frac{1}{p_i}\right)$$

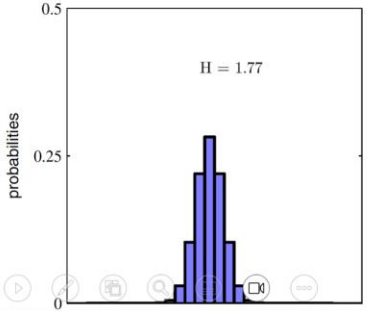
information

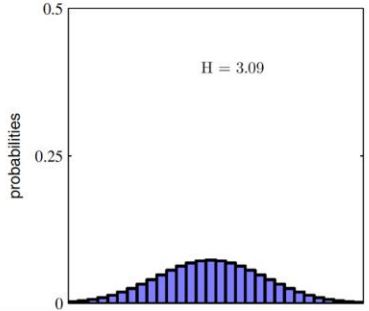
$$= -\sum_i p_i \log_2(p_i)$$

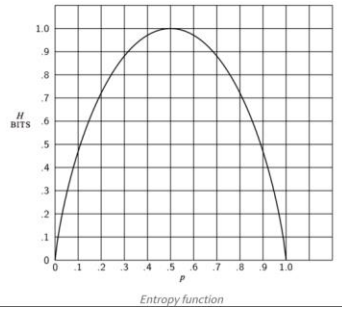
H는 "혼란" 또는 "무질서"와 관련된 단어인 "Havoc"에서 영감을 받았을 것으로 추정

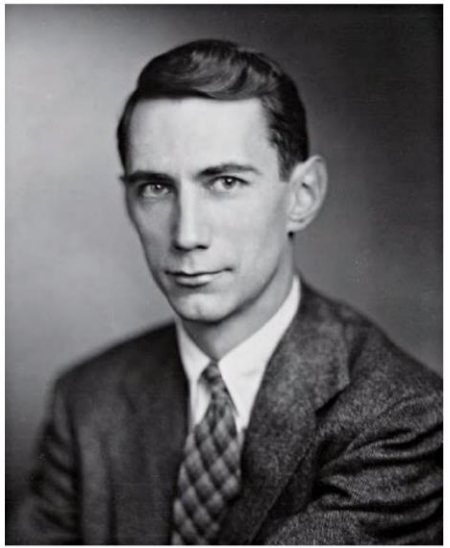
- Example: Probability of ball in the bag (q): red(80%), green(10%), blue(10%)

$$H(q) = -[0.8log(0.8) + 0.1log(0.1) + 0.1log(0.1)] = 0.63$$









Claude Shannon
(1961~2001)

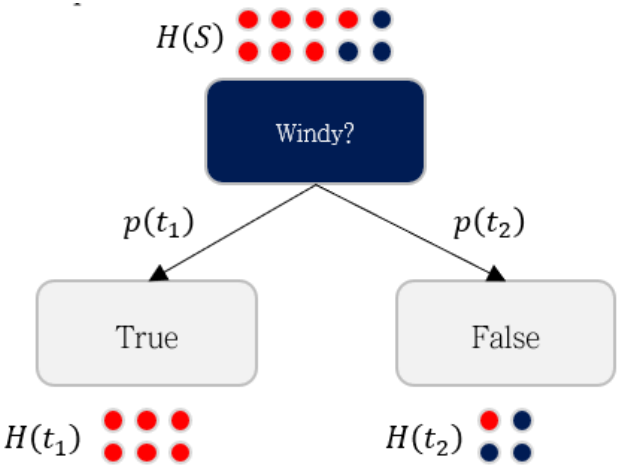
A Mathematical Theory of Communication

By C. E. SHANNON

Reprinted with corrections from *The Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, July, October, 1948.

Information Gain

❖ **분할 전 노드**의 엔트로피와 **분할 후 전체 노드**의 엔트로피의 차이



Entropy at parent level = $H(T)$

Entropy at children level = $H(T, a)$

Information Gain

❖ **분할 전 노드**의 엔트로피와 **분할 후 전체 노드**의 엔트로피의 차이

$$IG(T,a) = H(T) - H(T,a)$$

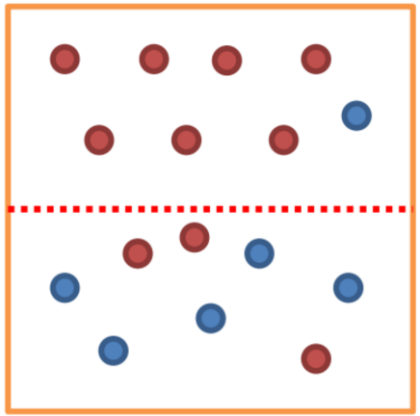
분할 전 Entropy　　분할 후 Entropy

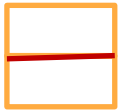
$$= H(T) - \sum p(t)H(t)$$

Information Gain

$$\mathrm{H}(T) = -\sum_{k=1}^{m} p_k \log_2 (p_k)$$

$$= -\frac{10}{16}\log_2 (\frac{10}{16}) - \frac{6}{16}\log_2 (\frac{6}{16}) \approx 0.95$$





$$H(T,a) = \sum_{i=1}^{d} R_i \left(-\sum_{k=1}^{m} p_k \log_2 (p_k) \right)$$

$$= 0.5 \times \left(-\frac{7}{8}\log_2 (\frac{7}{8}) - \frac{1}{8}\log_2 (\frac{1}{8}) \right) + 0.5 \times \left(-\frac{3}{8}\log_2 (\frac{3}{8}) - \frac{5}{8}\log_2 (\frac{5}{8}) \right) \approx 0.75$$

$$IG(T,a) = H(T) - H(T,a)$$

= 0.95 – 0.75

= 0.2

Example

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rain	mild	high	FALSE	Yes
rain	cool	normal	FALSE	Yes
rain	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rain	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rain	mild	high	TRUE	No

$$
\begin{aligned}
H(Play) &= -\sum_{i=1}^{c} p_i \log_2 p_i \\
&= -(\frac{5}{14} log_2 \frac{5}{14} + \frac{9}{14} log_2 \frac{9}{14}) \\
&= 0.94
\end{aligned}
$$

Example

		Play		
		Yes	No	Total
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rain	3	2	5

$$H(Play, Outlook) = p(sunny) \times H(3,2) + p(overcast) \times H(4,0) + p(rain) \times H(3,2)$$
$$= \frac{5}{14}(-\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5}) + \frac{4}{14}(-\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4}) + \frac{5}{14}(-\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5}$$
$$= 0.6935$$

		Play		
		Yes	No	Total
Humidity	High	3	4	7
	Normal	6	1	7

$$H(Play, Humidity) = p(High) \times H(3,4) + p(Normal) \times H(6,1)$$
$$= \frac{7}{14}(-\frac{3}{7}log_2\frac{3}{7} - \frac{4}{7}log_2\frac{4}{7}) + \frac{7}{14}(-\frac{6}{7}log_2\frac{6}{7} - \frac{1}{7}log_2\frac{1}{7})$$
$$= 0.7884$$

		Play		
		Yes	No	Total
Windy	True	3	3	6
	False	6	2	8

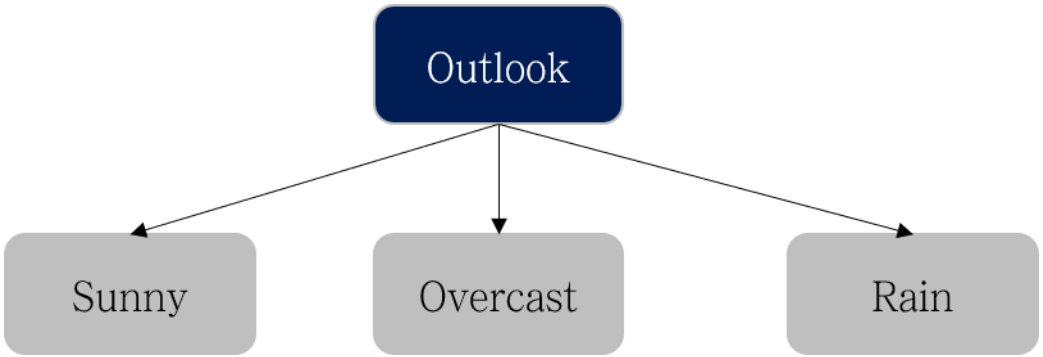
$$H(Play, Windy) = p(True) \times H(3,3) + p(False) \times H(6,2)$$
$$= \frac{6}{14}(-\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6}) + \frac{8}{14}(-\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8})$$
$$= 0.8921$$

		Play		
		Yes	No	Total
Temperature	Hot	2	2	4
	Mild	4	2	6
	Cool	3	1	4

$$H(Play, Temperature) = p(hot) \times H(2,2) + p(mild) \times H(4,2) + p(cool) \times H(3,1)$$
$$= \frac{4}{14}(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}) + \frac{6}{14}(-\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6}) + \frac{4}{14}(-\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4})$$
$$= 0.2857 + 0.3935 + 0.2317$$
$$= 0.911$$

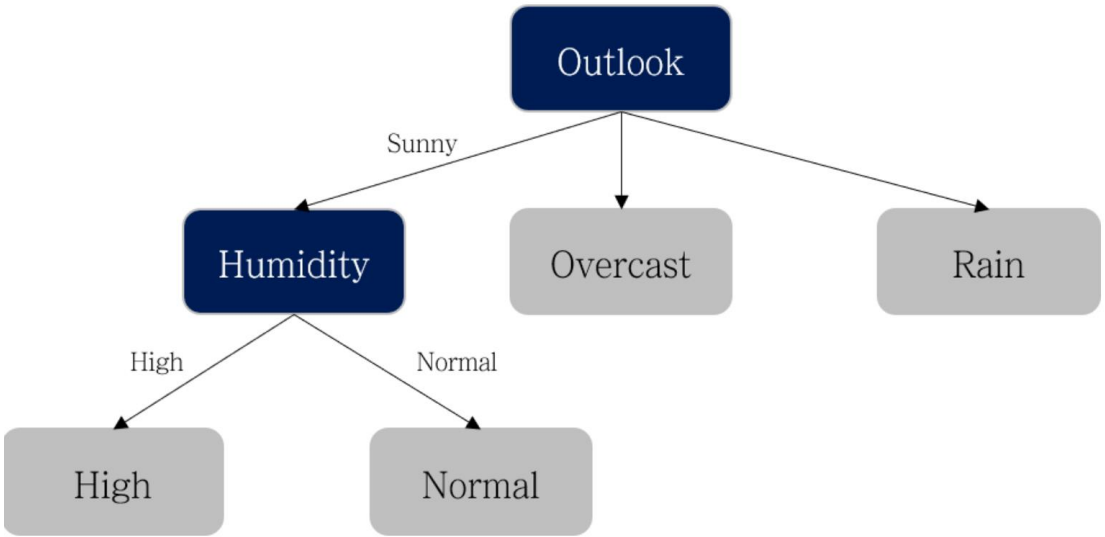
- $H(Play) - H(Play, Outlook) = 0.25$
- $H(Play) - H(Play, Temperature) = 0.02$
- $H(Play) - H(Play, Humidity) = 0.1514$
- $H(Play) - H(Play, Windy) = 0.047$

Example



- $H(Play) - H(Play, Outlook) = 0.25$
- $H(Play) - H(Play, Temperature) = 0.02$
- $H(Play) - H(Play, Humidity) = 0.1514$
- $H(Play) - H(Play, Windy) = 0.047$

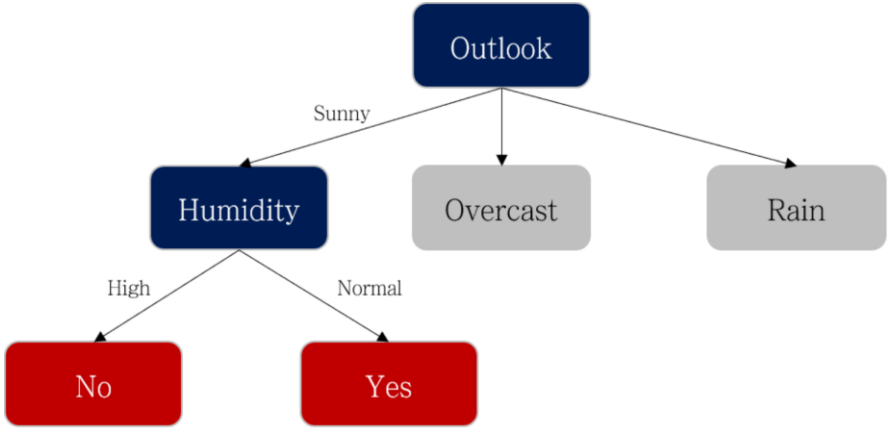
Example



Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes

- $H(Play) - H(Play, Temperature) = 0.571$
- $H(Play) - H(Play, Humidity) = 0.971$
- $H(Play) - H(Play, Windy) = 0.02$

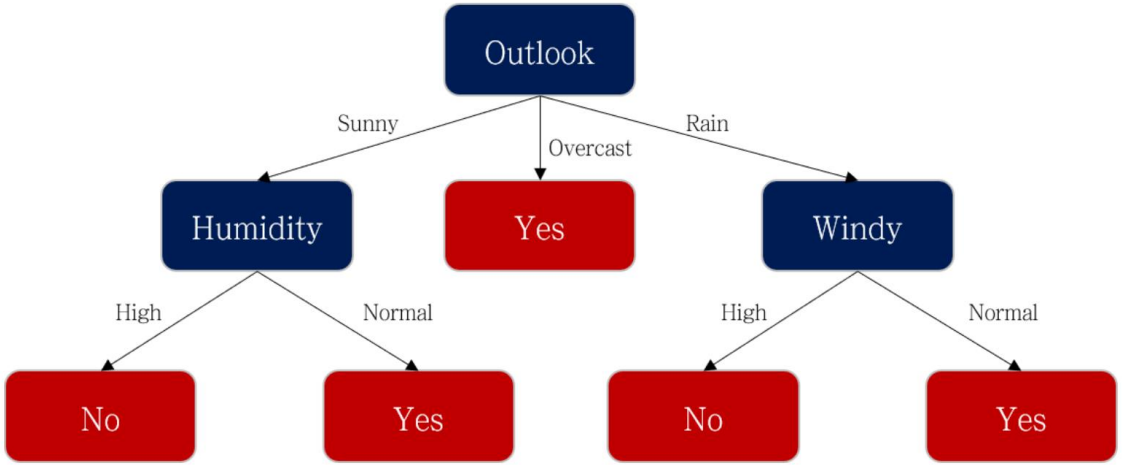
Example



Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes

Example

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rain	mild	high	FALSE	Yes
rain	cool	normal	FALSE	Yes
rain	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rain	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rain	mild	high	TRUE	No



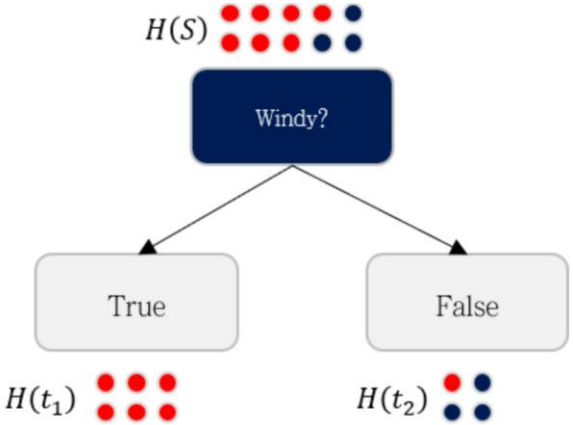
03

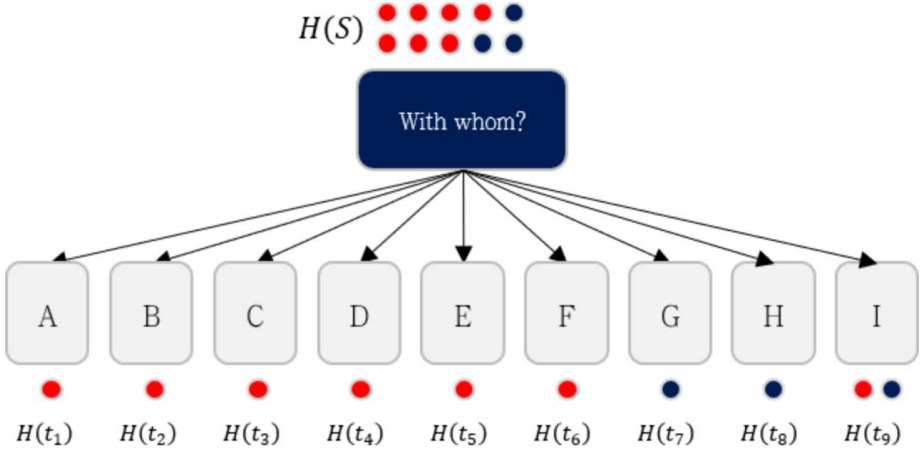
C4.5

Limitation of ID5

- Information gain
- Continuous variables
- Missing value
- Overfitting

Information Gain 한계점





Information Gain

$$= H(7,3) - (\frac{6}{10}H(6,0) + \frac{4}{10}H(1,3))$$
$$= 0.8813 - (\frac{6}{10} \times 0 + \frac{4}{10} \times 0.8113)$$
$$= 0.5568$$

<

Information Gain

$$= H(7,3) - (\frac{1}{10}H(1,0) + \frac{1}{10}H(1,0) + \ldots + \frac{1}{10}H(0,1) + \frac{2}{10}H(1,1))$$
$$= 0.8813 - (\frac{1}{10} \times 0 + \ldots + \frac{2}{10} \times 1)$$
$$= 0.6813$$

https://tyami.github.io/machine%20learning/decision-tree-3-c4_5/

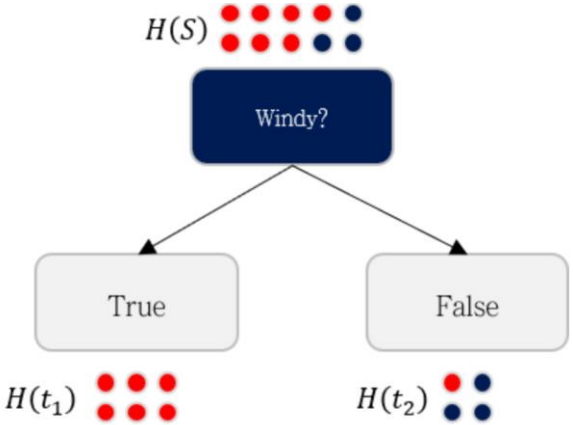
Information Gain Ratio (GR)

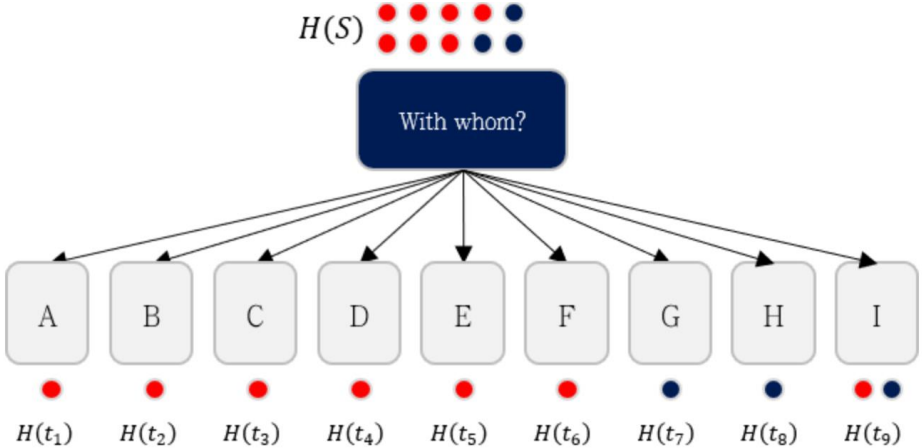
- Information gain (IG) 한계
  - 잘게 나눌 수록 균질할 (homogeneous) 가능성이 높다. ➔ 잘게 나눌수록 IG 증가
- 대안
  - Normalization by entropy (Intrinsic Value, IV)

$$Information\ Gain\ Ratio = \frac{IG}{IV}$$

$$Intrinsic\ Value = IV = -\sum_{i}^{N} p_i log_2 p_i$$

Information Gain Ratio (GR)





$$Information\ Gain\ Ratio = IG/IV$$
$$= \frac{0.5568}{-(\frac{6}{10}log_2\frac{6}{10} + \frac{4}{10}log_2\frac{4}{10})}$$
$$= \frac{0.5568}{0.9701}$$
$$= 0.5739$$

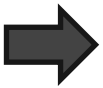
$>$

$$Information\ Gain\ Ratio = IG/IV$$
$$= \frac{0.6813}{3.1219}$$
$$= 0.2182$$

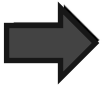
Continuous Variables

- Discrete variables ➔ Continuous variables

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rain	mild	high	FALSE	Yes
rain	cool	normal	FALSE	Yes
rain	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rain	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rain	mild	high	TRUE	No



Temperature	Play
36	No
35	No
22	Yes
24	Yes
20	Yes
35	No
31	Yes
30	No
29	Yes
28	Yes
25	Yes
26	Yes
29	Yes
27	No



Temperature	Play
20	Yes
22	Yes
24	Yes
25	Yes
26	Yes
27	No
28	Yes
29 29	Yes Yes
30	No
31	Yes
35 35	No No
36	No

Continuous Variables

- Method
  - 특정 Threshold를 만들어서 초과/이하 값으로 나누자 ➔ How?

Continuous Variables

- Method
  - 특정 Threshold를 만들어서 초과/이하 값으로 나누자

    ➔ IG를 breakpoint마다 나눈 후 가장 높은 곳을 threshold로 잡자 ➔ 너무 많은 계산량이 필요

	Temperature	Play
21	20	Yes
23	22	Yes
24.5	24	Yes
25.5	25	Yes
26.5	26	Yes
27.5	27	No
28.5	28	Yes
	29	Yes
	29	Yes
29.5	30	No
30.5	31	Yes
33	35	No
	35	No
35.5	36	No

$$IG(Play, Temperature(21)) = 0.04742$$
$$IG(Play, Temperature(23)) = 0.1001$$
$$IG(Play, Temperature(24.5)) = 0.1590$$
$$IG(Play, Temperature(25.5)) = 0.2257$$
$$IG(Play, Temperature(26.5)) = 0.3029$$
$$IG(Play, Temperature(27.5)) = 0.09000$$
$$IG(Play, Temperature(28.5)) = 0.1516$$
$$IG(Play, Temperature(29.5)) = 0.3586$$
$$IG(Play, Temperature(30.5)) = 0.1925$$
$$IG(Play, Temperature(33)) = 0.4009$$
$$IG(Play, Temperature(35.5)) = 0.2863$$

Continuous Variables

- Method
  - 특정 Threshold를 만들어서 초과/이하 값으로 나누자 ➔ 계산량을 줄이자!

Method 1
(all breakpoints)
11 Breakpoints

	Temperature	Play
21	20	Yes
23	22	Yes
24.5	24	Yes
25.5	25	Yes
26.5	26	Yes
27.5	27	No
28.5	28	Yes
29.5	29	Yes
	29	Yes
30.5	30	No
33	31	Yes
35.5	35	No
	35	No
	36	No

Method 2
(class 바뀔 때만)
5 Breakpoints

	Temperature	Play
	20	Yes
	22	Yes
	24	Yes
	25	Yes
26.5	26	Yes
27.5	27	No
	28	Yes
	29	Yes
	29	Yes
29.5	30	No
30.5	31	Yes
33	35	No
	35	No
	36	No

Method 3
(Q1, Median, Q3)
3 Breakpoints

	Temperature	Play
	20	Yes
	22	Yes
	24	Yes
25.25	25	Yes
	26	Yes
	27	No
28.5	28	Yes
	29	Yes
	29	Yes
30.75	30	No
	31	Yes
	35	No
	35	No
	36	No

Missing Value

- 결측치(missing value)가 발생 시 ID3는 활용 불가능
  - C4.5에서는 missing value를 고려
- Method
  - 1단계: Entropy는 Non-missing value로만 계산
  - 2단계: Information Gain는 Weighted Information Gain로 변경

$$Weighted\ Information\ Gain = F \times IG(S, A)$$
$$F = proportion\ of\ non\ missing\ value$$

  - 3단계: Intrinsic Value는 missing value를
    하나의 클래스로 보고 모든 데이터로 계산

Outlook	Play
Sunny	No
Missing value	No
	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
Missing value	No
	Yes
	Yes
	Yes
Overcast	Yes
Overcast	Yes
Rain	No

Missing Value

- 1단계: Entropy는 Non-missing value로만 계산 (데이터는 14개 중 8개 존재)

$$H(Play) = -\sum_{i=1}^{C} p_i log_2 p_i$$
$$= -(\frac{3}{8} log_2 \frac{3}{8} + \frac{5}{8} log_2 \frac{5}{8}) \quad = 0.9544$$

$$H(Play, Outlook) = \sum p(t) H(t)$$
$$= \frac{1}{8} H(1,0) + \frac{4}{8} H(2,2) + \frac{3}{8} H(0,3)$$
$$= 0.5$$

$$IG(Play, Outlook) = H(Play) - H(Play, Outlook)$$
$$= 0.4544$$

- 2단계: Information Gain는 Weighted Information Gain로 변경

$$F = \frac{8}{14} \qquad Weighted\ Information\ Gain = \frac{8}{14} \times 0.4544 \quad = 0.2597$$

- 3단계: Intrinsic Value는 missing value를 하나의 클래스로 보고 계산
  - sunny(n=1), rain(n=4), overcast(n=3), missing values(n=6))

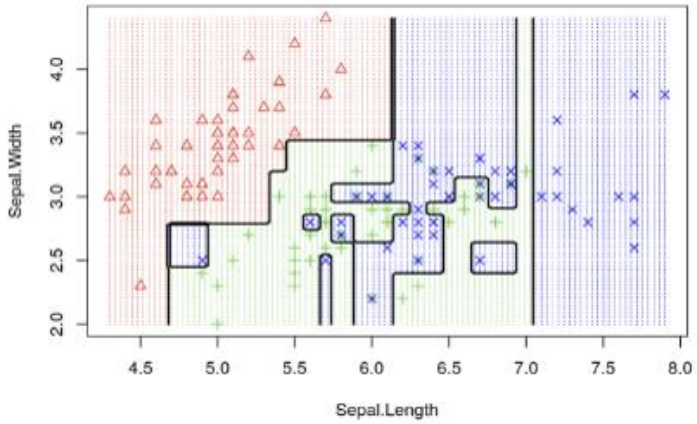
$$IV = -(\frac{1}{14} log_2 \frac{1}{14} + \frac{4}{14} log_2 \frac{4}{14} + \frac{3}{14} log_2 \frac{3}{14} + \frac{6}{14} log_2 \frac{6}{14}) \quad = 1.659$$

$$Information\ Gain\ Ratio = \frac{0.2597}{1.659} = 0.1565$$

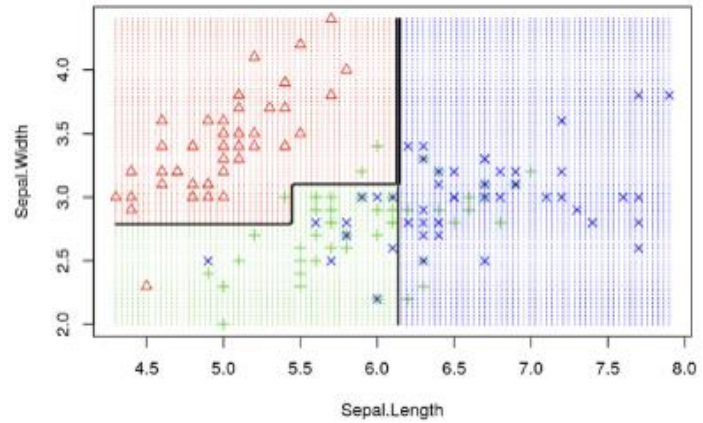
Outlook	Play
Sunny	No
	No
	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
	No
	Yes
	Yes
	Yes
Overcast	Yes
Overcast	Yes
Rain	No

Overfitting

- Pruning
  - Decision Tree에서 Overfitting을 방지하기 위해 적절한 수준에서 terminal node를 결합하는 방식



Full Tree (Overfitting)

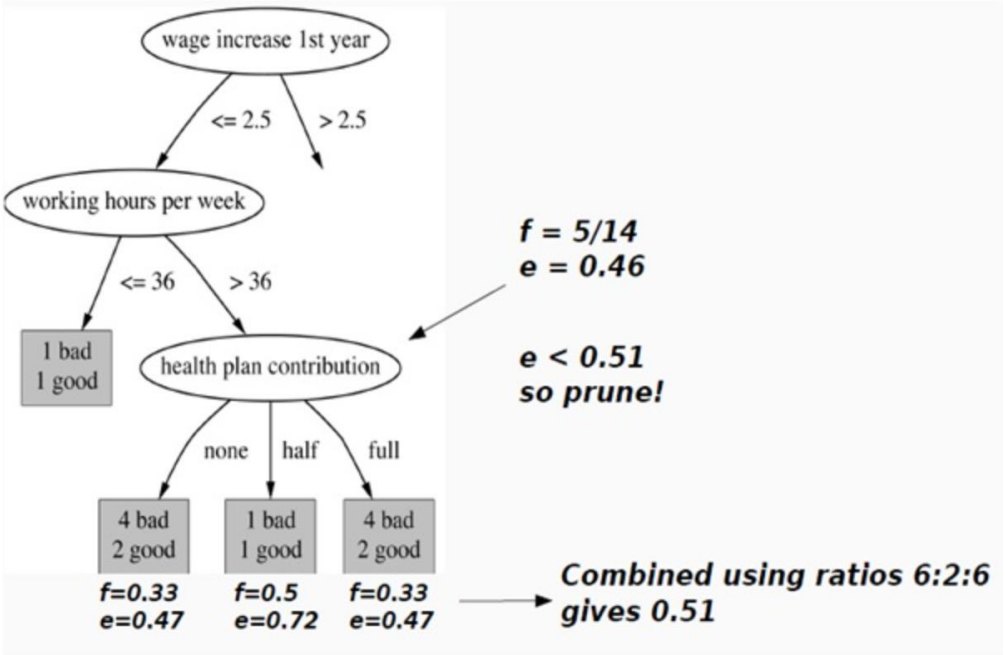


Tree Pruning

Pruning

- Method
  - Pre-Pruning: Decision tree 완성 전에 가지치기 수행
  - Post-Pruning: Decision tree 완성 후에 가지치기 수행 ➔C4.5 적용
- Post-Pruning
  - Step 1: 데이터를 training/test 데이터가 아니라 training/pruning/test 데이터 분리
  - Step 2: Training data만으로 의사결정 나무 생성.
  - Step 3: Pruning with test data Pruning data로 가지치기를 수행합니다.

Pruning Example



$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

$N = sample\ size$

$f = Error\ rate$

$z = z\ score\ (default\ z = 0.69)$

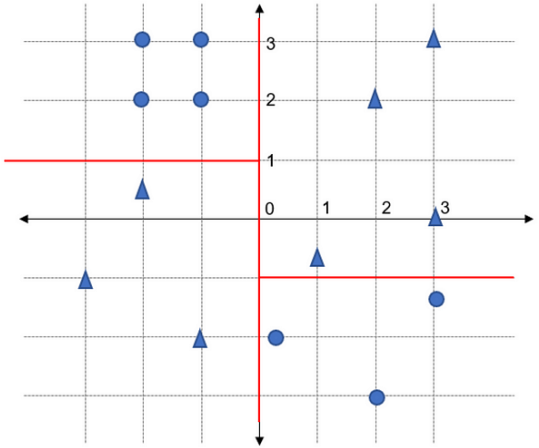
04

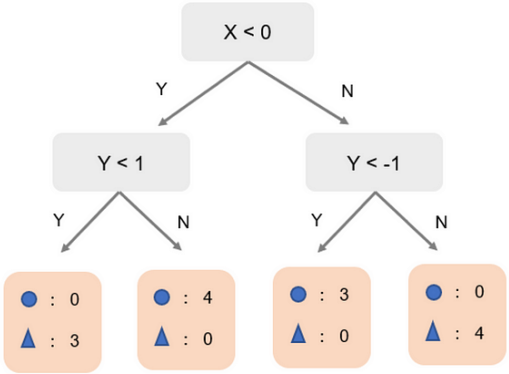
CART (Classification And Regression Tree)

CART (Classification And Regression Tree)

❖ 분류(classification)와 회귀(regression)가 가능한 Decision Tree

- 불순도: Gini index
- Binary tree
- Regression tree





GINI Index

- 불순도 (Impurity) 지표를 나타내는 통계학적 지수
  - 데이터를 특정 기준으로 분류 했을 때 여러 Class 데이터들이 섞여있는 정도(Impurity)를 나타내는 수치
  - 이탈리아의 통계학자인 코라도 지니(Corrado Gini)가 1912년 발표한 논문 "Variabilità e mutabilità"에 처음 소개

$$GI = 1 - \sum_j p_j^2$$

Where p_j is the probability of class j

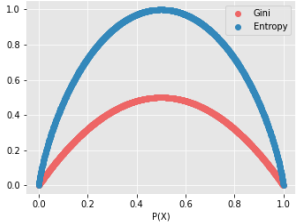
$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

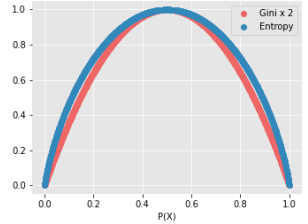
$$= 1 - \left(\frac{6}{16}\right)^2 - \left(\frac{10}{16}\right)^2$$

$$\approx 0.47$$

Gini vs Entropy

- Range
  - Gini [0, 0.5] vs Entropy [0,1]
- Computation Time
  - Gini가 Entropy에 비해 계산 속도가 빠르다
- Performance
  - 유사함
    - 하지만, entropy를 통해 얻은 결과가 조금 더 좋음





	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Gini	0.8619 ± 0.0088	0.8148 ± 0.0030	0.9350 ± 0.0146	0.8481± 0.0029
Entropy	0.8659 ± 0.0205	0.8119 ± 0.0223	0.9390 ± 0.0146	0.8583 ± 0.0067

➔ **데이터량이 많은 경우 Gini가 학습 시간에서 유리하다**

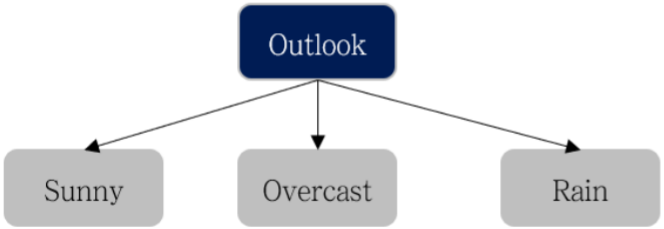
https://quantdare.com/decision-trees-gini-vs-entropy/

Binary Tree

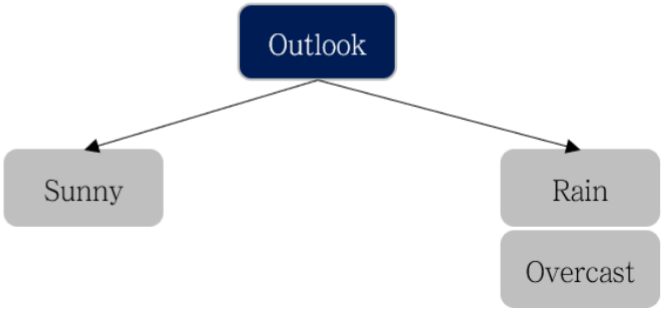
❖ CART

- Binary Tree 형태
  - 가지 분기 시, 여러 개의 자식 노드가 아닌 단 두 개의 노드로 분기

ID3

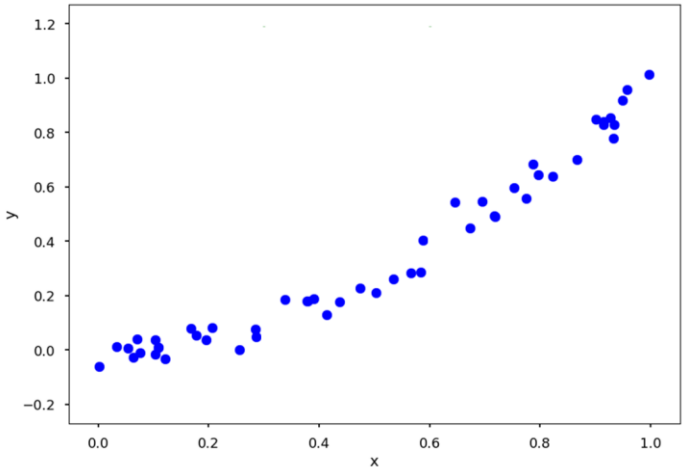


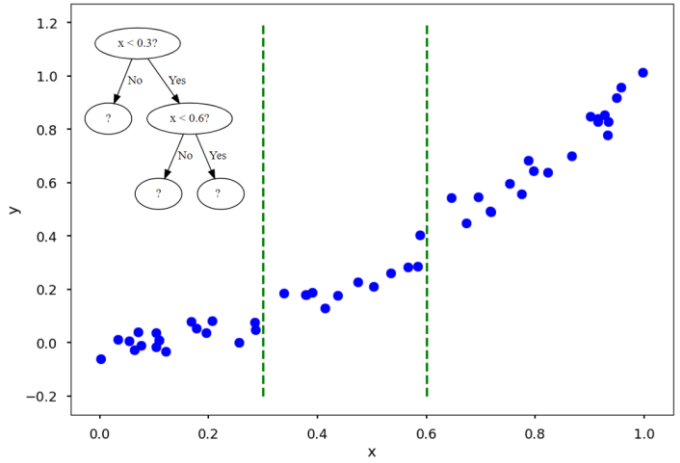
CART



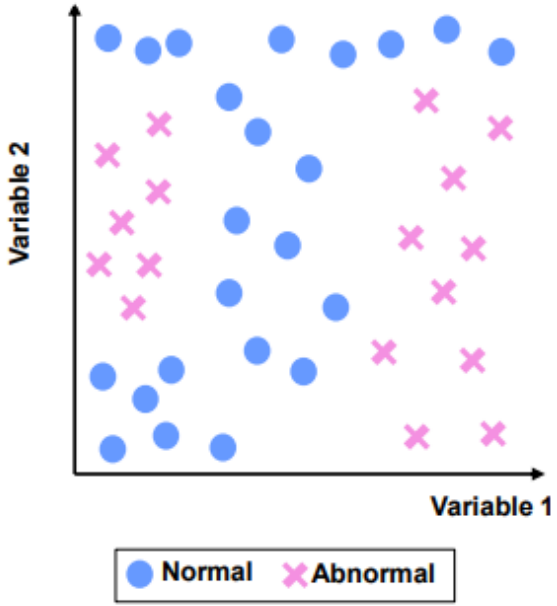
Regression Tree

- 분기 지표를 선택할 때 사용하는 index를 불순도 (Entropy, Gini index)가 아닌, 실제값과 예측값의 오차값을 사용



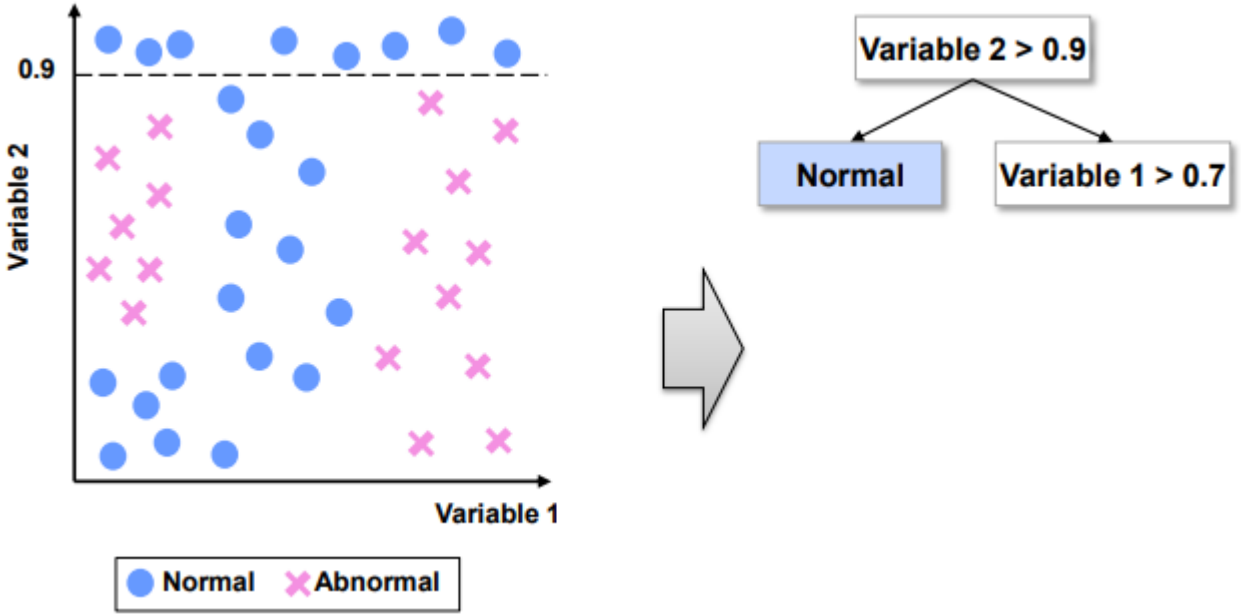


Example

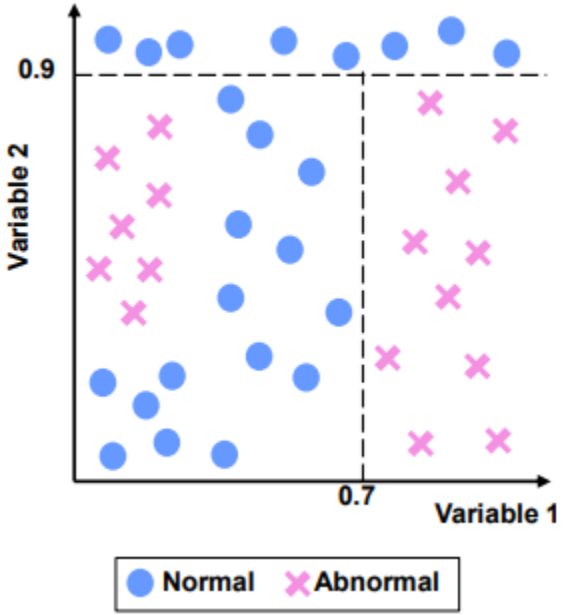


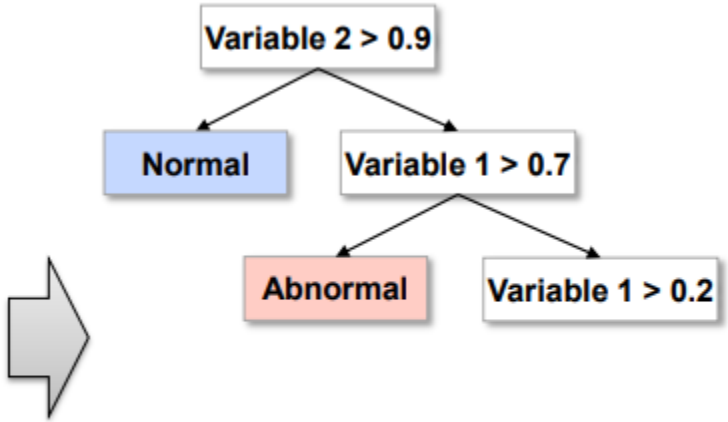
출처: https://velog.io/@barley_15/%EC%9D%98%EC%82%AC%EA%B2%B0%EC%A0%95%EB%82%98%EB%AC%B4-CART

Example

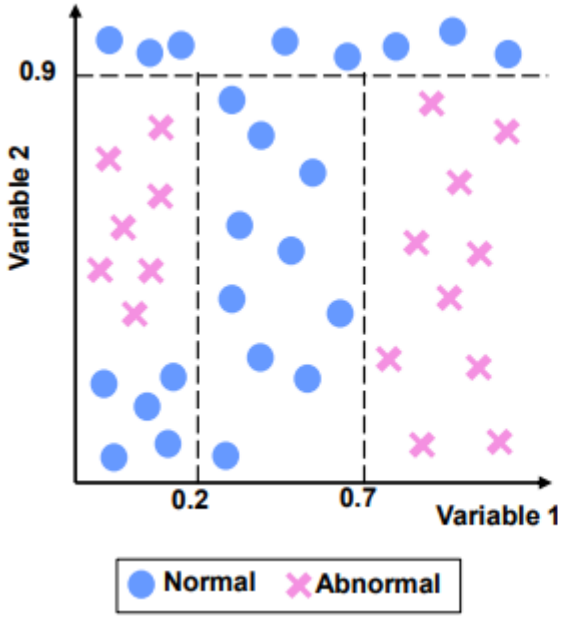

0.9
Variable 2
Variable 1
Normal
Abnormal
Variable 2 > 0.9
Normal
Variable 1 > 0.7

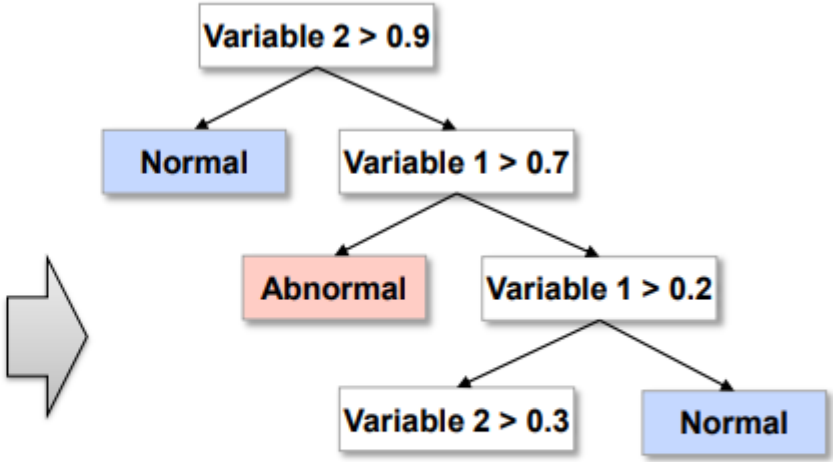
Example


0.9
Variable 2
0.7
Variable 1
Normal
Abnormal

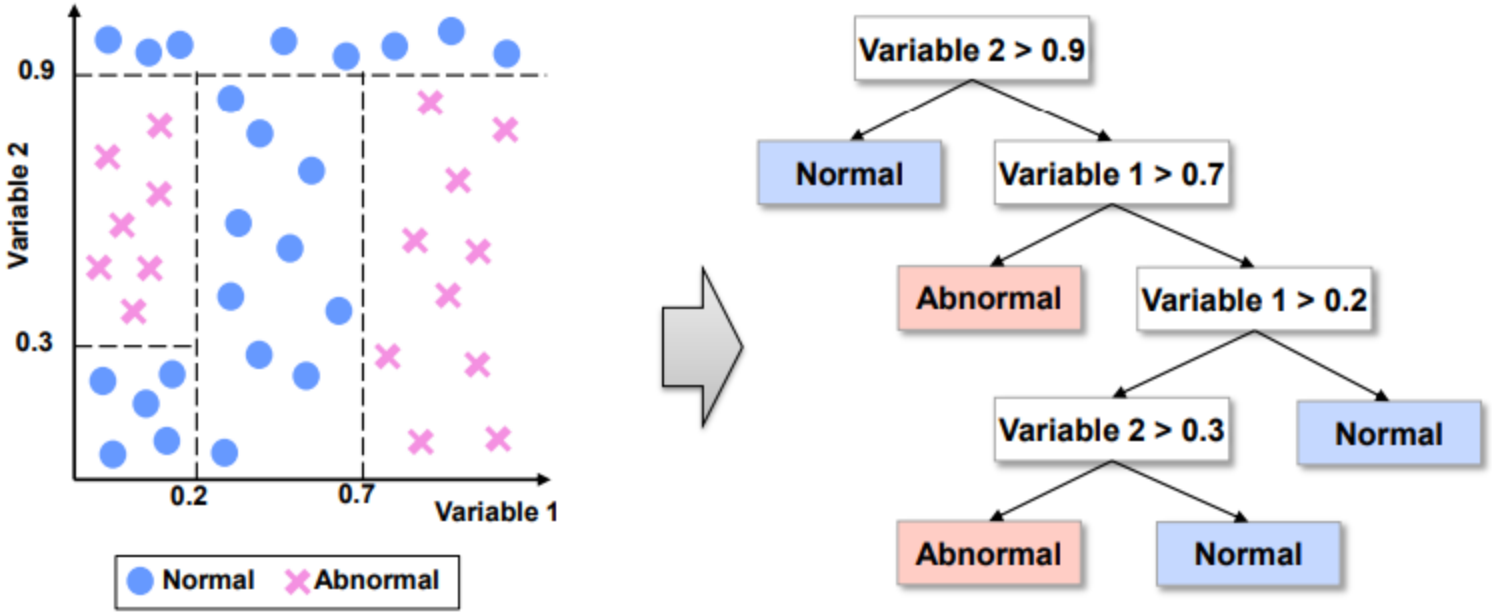

Variable 2 > 0.9
Normal
Variable 1 > 0.7
Abnormal
Variable 1 > 0.2

Example

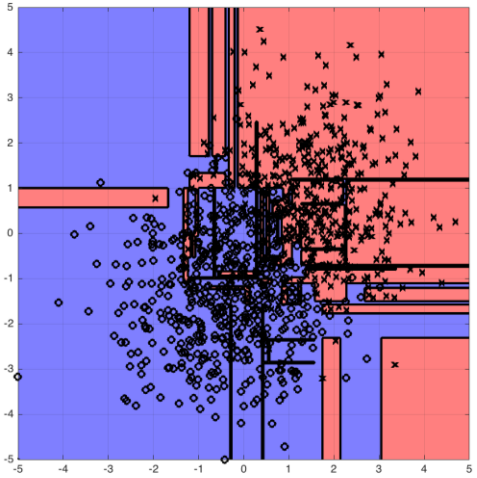

0.9
Variable 2
0.2
0.7
Variable 1
Normal
Abnormal

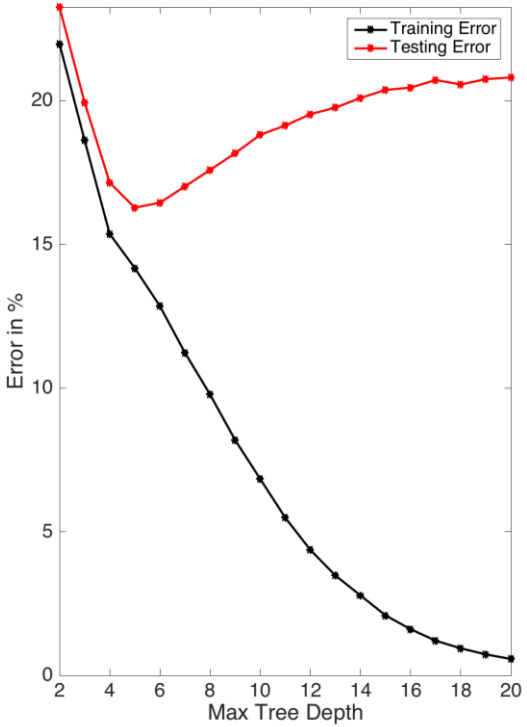

Variable 2 > 0.9
Normal
Variable 1 > 0.7
Abnormal
Variable 1 > 0.2
Variable 2 > 0.3
Normal

Example


0.9
0.3
Variable 2
0.2
0.7
Variable 1
Normal
Abnormal
Variable 2 > 0.9
Normal
Variable 1 > 0.7
Abnormal
Variable 1 > 0.2
Variable 2 > 0.3
Normal
Abnormal
Normal

Problem




Training Error
Testing Error
Error in %
Max Tree Depth
0
5
10
15
20
2
4
6
8
10
12
14
16
18
20

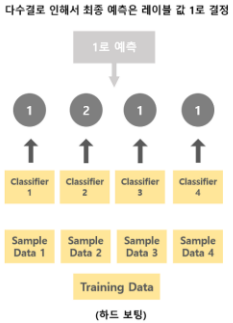
05

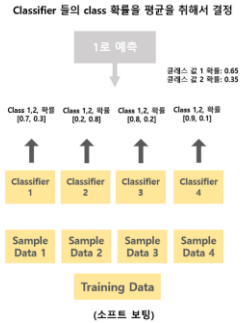
Random Forest

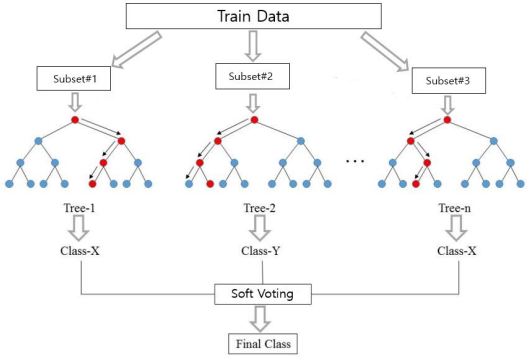
앙상블 학습 (ensemble learning)

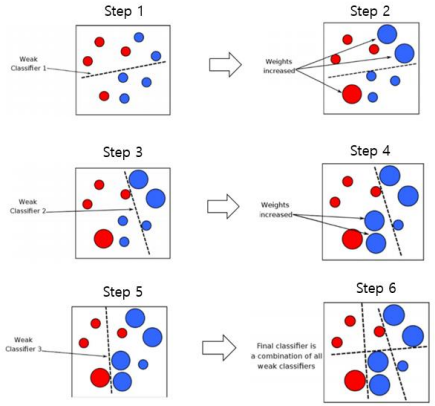
- 정의
  - 여러 개의 분류기를 생성하고 각 예측들을 결합함으로써 보다 정확한 예측을 도출하는 기법
- 방법
  - Voting
    - 서로 다른 알고리즘으로 예측하고 예측한 결과를 가지고 투표하듯 보팅을 통해 최종 예측 결과를 선정하는 방식
  - Bagging (bootstrap aggregation)
    - 같은 알고리즘으로 여러 개의 분류기를 만들어서 보팅으로 최종 결정하는 알고리즘
  - Boosting
    - 여러 개의 약한 학습기를 순차적으로 학습 - 예측 하면서 잘못 예측한 데이터에 가중치를 부여해서 오류를 개선해 나가면서 점진적으로 학습하는 방식

앙상블 학습 (ensemble learning)









Voting

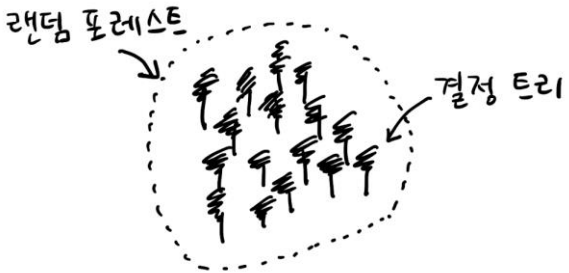
Bagging

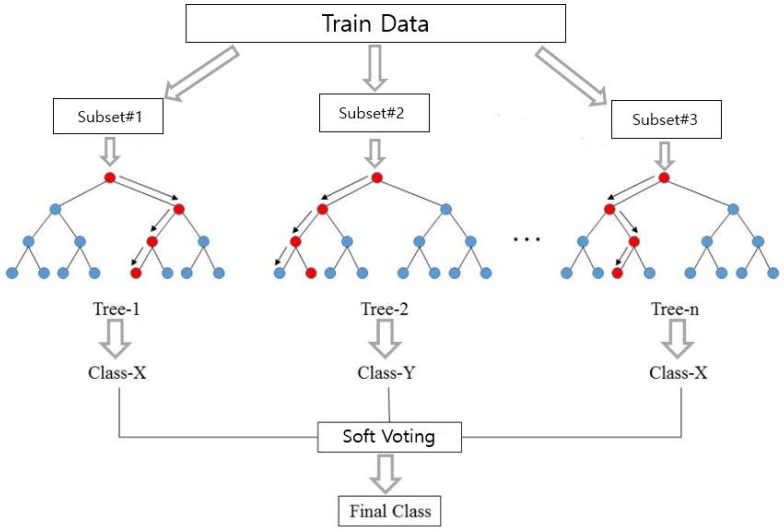
Boosting

ref: https://casa-de-feel.tistory.com/8

Random Forest

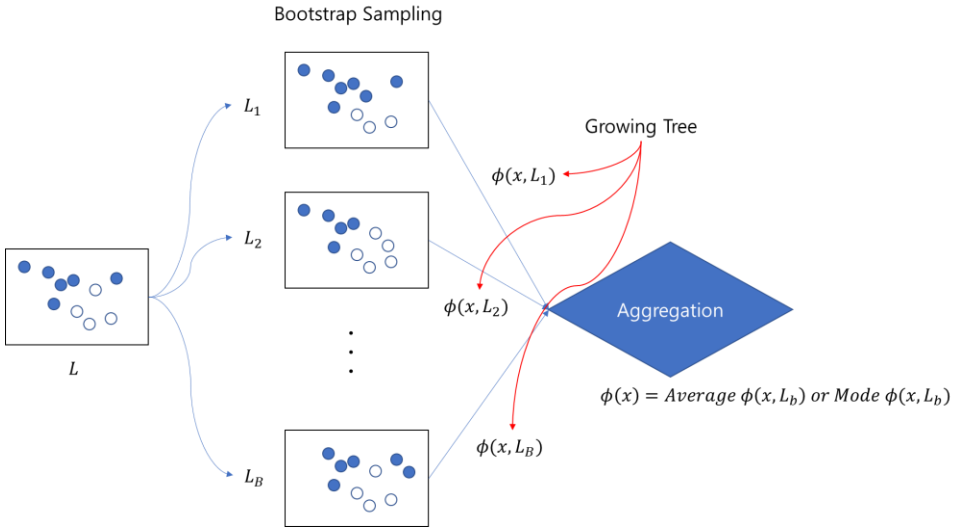
❖ 앙상블 학습 (ensemble learning)으로 결정트리를 랜덤하게 만들어 결정트리의 숲을 만드는 방법





Random Forest

- 여러 개의 결정 트리 분류기가 전체 학습 데이터에서 배깅 방식으로 각자의 데이터를 샘플링해 개별적으로 학습을 수행한 뒤 최종적으로 모든 분류기가 보팅을 통해 예측 결정
  - Bootstrap sampling (Bootstraping) : 학습 데이터를 여러 개의 데이터 세트로 중첩되게 분리



Random Forest

- 장점
  - 예측력이 좋다
  - 이상치(Outlier)에 강하다
- 단점
  - 모델의 해석이 어렵다
  - 계산량이 많아 학습에 소요되는 시간이 하나의 Decision Tree에 비해 많다