

17기 정규세션

ToBig's 16기 김송민

# Decision Tree

의사결정나무

# Contents

---

Unit 01 | 의사결정 나무란?

---

Unit 02 | ID3 알고리즘

---

Unit 03 | CART 알고리즘

---

Unit 04 | feature가 연속형이라면?

---

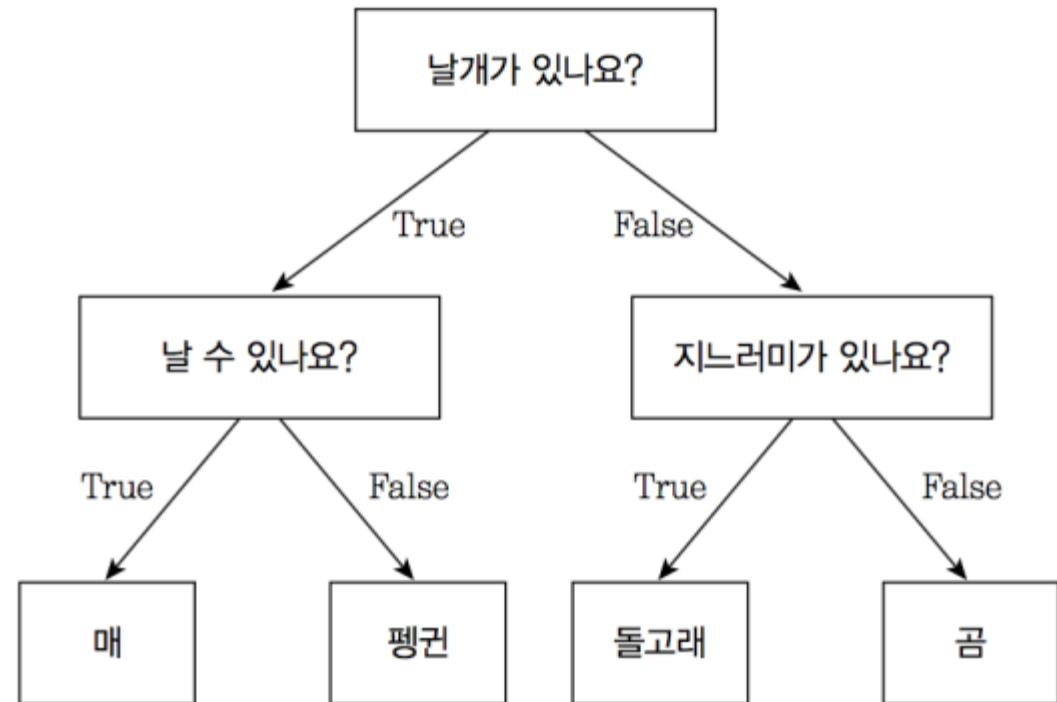
Unit 05 | 가지치기

---

## Unit 01 | 의사결정 나무란?

## Decision Tree

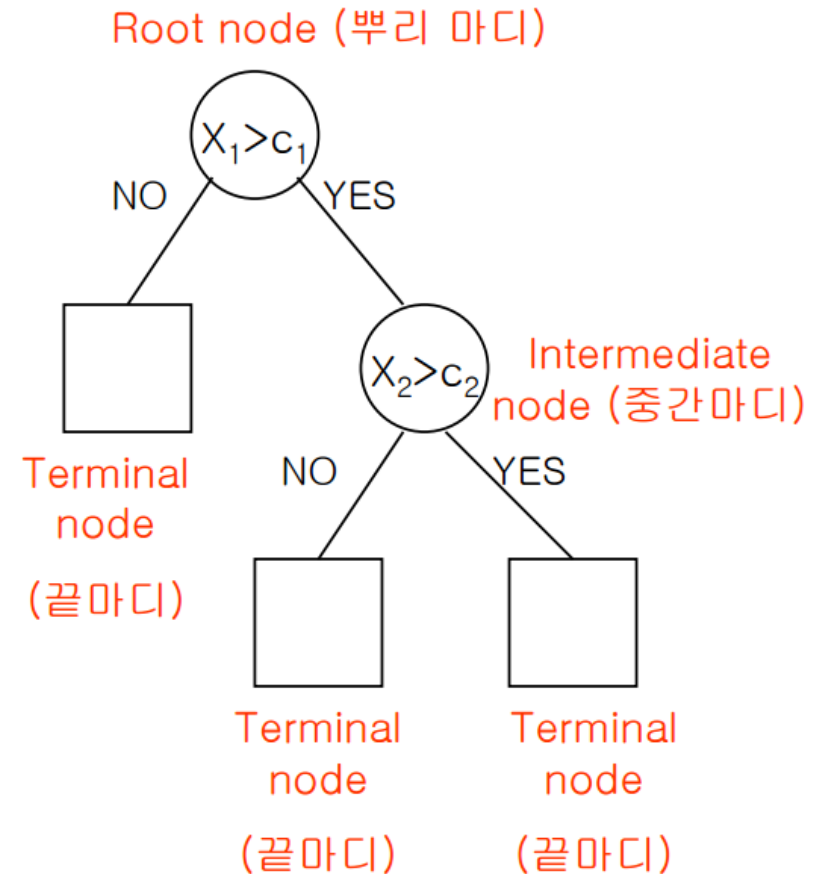
- 의사 결정 규칙을 나무구조로 나타내어 전체 데이터를 소집단으로 분류하거나 예측하는 방법
- 즉, 데이터 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내며, 이 모양이 나무와 비슷해붙인 이름.



## Unit 01 | 의사결정 나무란?

## Decision Tree

- 초기 지점은 root node이고, 이외의 node들은 intermediate node라고 한다. 통과하는 node들이 늘어날수록 조건에 부합하는 데이터의 수는 줄어든다.
- Terminal node의 데이터들의 합은 root node의 데이터와 동일함.

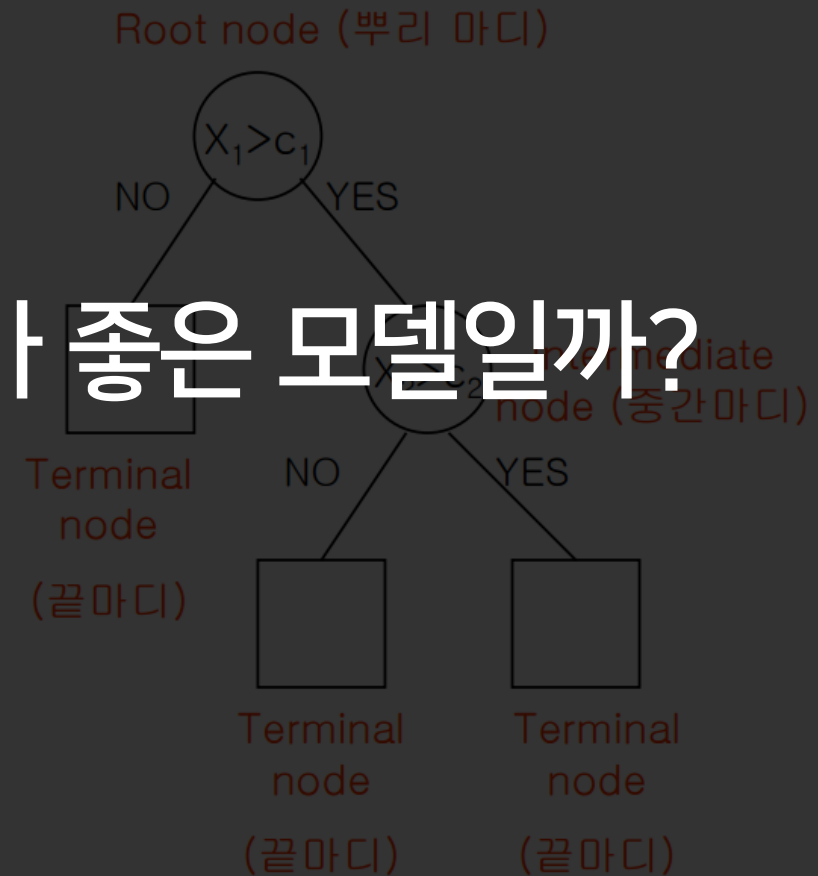


## Unit 01 | 의사결정 나무란?

## Decision Tree

- 초기 지점은 root node이고, 이외의 node들은 intermediate node이다. 즉, node들이 늘어날수록 조건에 부합하는 데이터의 수는 줄어든다.
- Terminal node의 데이터들의 합은 root node의 데이터와 동일함.

# 어떤 Decision Tree가 좋은 모델일까?



## Unit 01 | 의사결정 나무란?

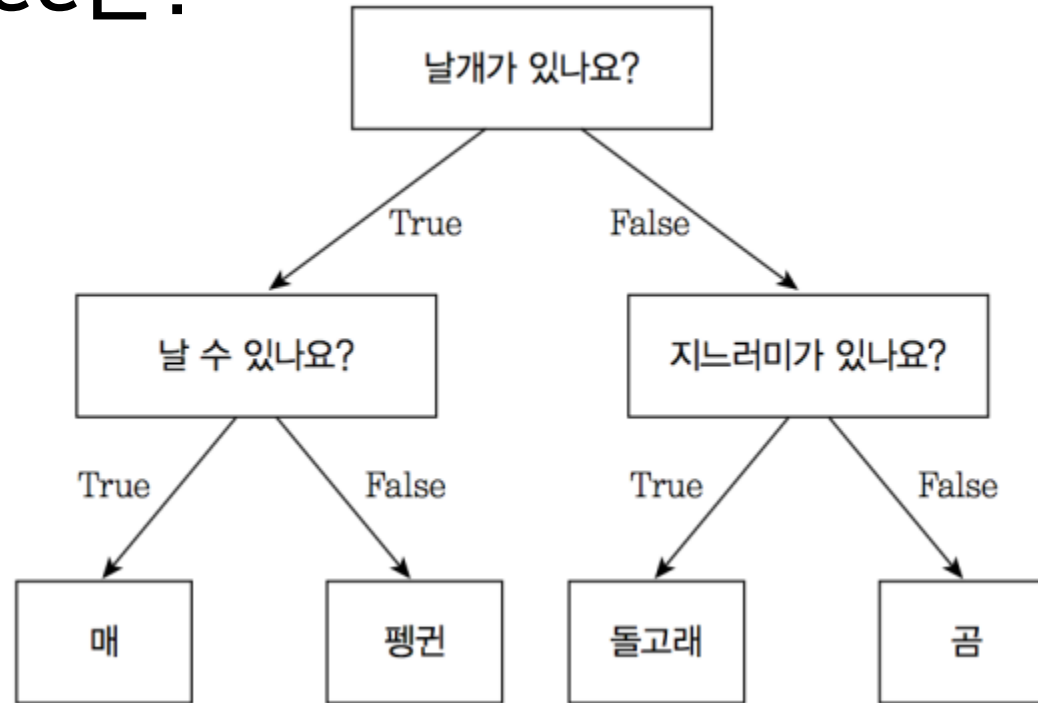
### 좋은 Decision Tree는?

똑같은 정확도를 내면서,  
**Simple한 것**을 선호한다!  
(= simple 할수록 일반화를 잘한다.)

**각각의 노드는 최대한 한가지 클래스만** 가지고 싶어한다.  
(= 한쪽에 몰려 있을수록 좋은 decision이다.)

## Unit 01 | 의사결정 나무란?

### 좋은 Decision Tree는?



Case1) 매 25  
          펭귄 5   👍

Case2) 매 18  
          펭귄 12

## Unit 01 | 의사결정 나무란?

### 좋은 Decision Tree는?

각 노드가 최대한 한 가지 클래스만 가지도록 하는 Decision Tree,

그러면서 최대한 간단한 Decision Tree를 만들 수 있도록

좋은 기준을 잡아야 한다!



## Unit 01 | 의사결정 나무란?

좋은 Decision Tree는?

좋은 노드들을 만들 수 있는  
좋은 기준을 어떻게 정할까?  
각 노드가 최대한 한 가지 클래스만을 포함하는 Decision Tree,  
그러면서 최대한 간단한 Decision Tree를 만들 수 있도록

**불순도!!**  
좋은 기준을 잡아야 한다!

## Unit 01 | 의사결정 나무란?

Q. 불순도를 측정하는 지표는?

A. Entropy, Gini index

Q. 어떤 기준으로 노드를 놓아야 하며, 어떤 노드를 가장 위에 놓아야 할까?

A. ID3 & CART 알고리즘

## Unit 02 | ID3 알고리즘

### 1. 순도 / 불순도 지표

#### ① Entropy (엔트로피)



High Entropy (messy)



Low Entropy (Clean)

## Unit 02 | ID3 알고리즘

- 순도 / 불순도 지표

### Entropy (엔트로피)

- 무질서도를 정량화 해서 표현한 값(열역학)
- 여기서는 데이터의 불확실성을 나타낸다.
- 어떤 집합의 entropy 가 높을수록 그 집단의 특징을 찾는것이 어렵다.
- 우리의 목적 : entropy를 감소시키는 방향으로 분류하기.

Entropy 감소 = 불순도 감소 = 순도 증가

$$Entropy(A) = - \sum_{k=1}^m p_k \log_2 (p_k)$$

## Unit 02 | ID3 알고리즘

Ex1)  $S = [+, +, +, +, +, +, +, +, +, +]$ Ex2)  $S = [+, +, +, +, +, +, -, -, -, -, -]$ Ex3)  $S = [-, -, -, -, -, -, -, -, -, -]$ 

- $S$  is a set of examples
- $p_{\oplus}$  is the proportion of examples in class  $\oplus$
- $p_{\ominus} = 1 - p_{\oplus}$  is the proportion of examples in class  $\ominus$

Entropy:

$$Entropy(A) = - \sum_{k=1}^m p_k \log_2 (p_k)$$

$$E(S) = -p_{\oplus} \cdot \log_2 p_{\oplus} - p_{\ominus} \cdot \log_2 p_{\ominus}$$

- Interpretation:
  - amount of unorderedness in the class distribution of  $S$

## Unit 02 | ID3 알고리즘



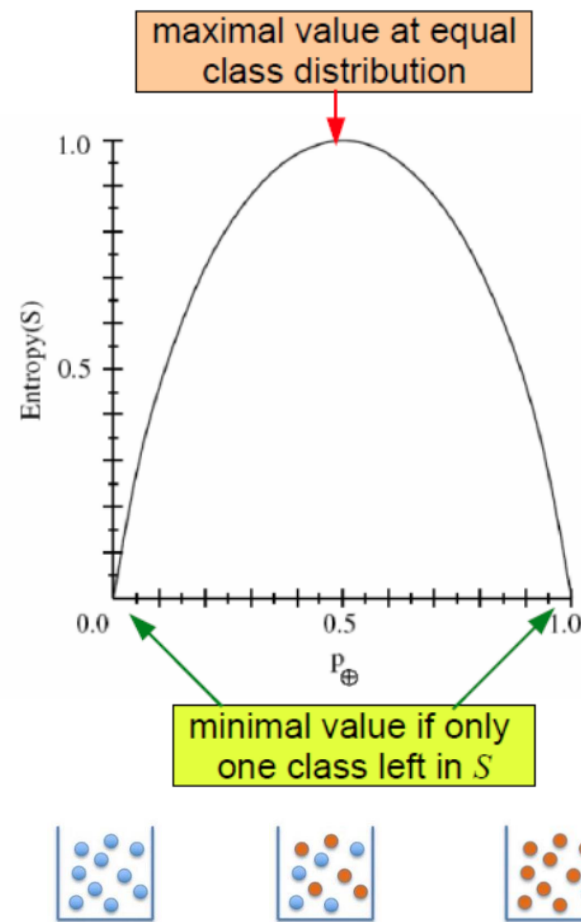
Ex1)  $S = [+, +, +, +, +, +, +, +, +, +]$



Ex2)  $S = [+, +, +, +, +, -, -, -, -, -]$



Ex3)  $S = [-, -, -, -, -, -, -, -, -, -]$



## Unit 02 | ID3 알고리즘



Ex1)  $S = [+, +, +, +, +, +, +, +, +, +]$

불확실성 최소 = 순도 최대 : 엔트로피 0

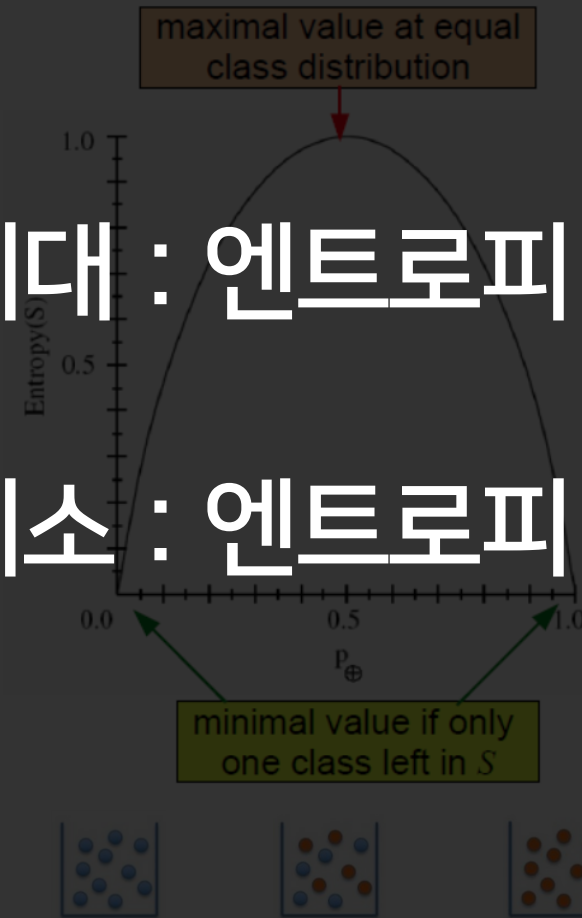


Ex2)  $S = [+, +, +, +, +, -, -, -, -, -]$

불확실성 최대 = 순도 최소 : 엔트로피 1



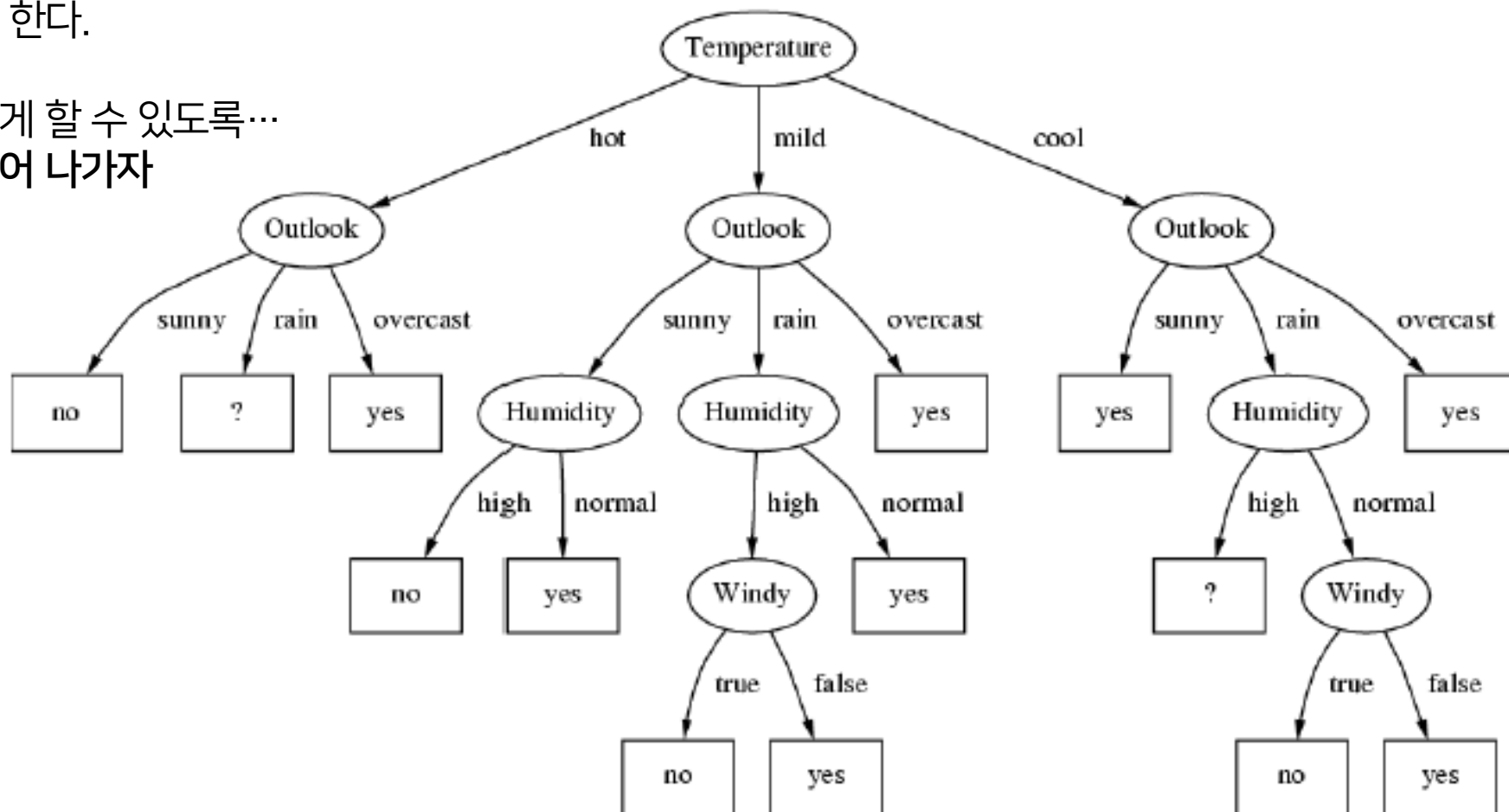
Ex3)  $S = [-, -, -, -, -, -, -, -, -, -]$



## Unit 02 | ID3 알고리즘

의사결정 나무를 만들기 위해서는  
그림과 같이 기준이 될 변수를 선택해야 한다.

새로운 데이터에 대해서 가장 잘 설명하게 할 수 있도록...  
불순도를 잘 줄여주는 기준을 찾아 나누어 나가자





## Unit 02 | ID3 알고리즘

- ID3 알고리즘

- Entropy 지수를 이용한 알고리즘
- Entropy 지수를 통해 Information Gain 도출
- Information Gain이 크게 나오는 변수 A를 기준으로 선택


Information Gain이란?

상위 노드의 Entropy에서 하위노드의 Entropy를 뺀 값!

즉, information Gain 값이 클수록, 엔트로피를 많이 줄였다는 의미.  
= 엔트로피가 작아졌다.

## Unit 02 | ID3 알고리즘

Information Gain for Attribute  $A$ 

  $Gain(S, A) = E(S) - I(S, A)$

$$I(S, A) = \sum_i \frac{|S_i|}{|S|} \cdot E(S_i) \quad \downarrow$$

$S$  : 주어진 데이터들의 집합  
 $|S|$  : 주어진 데이터들의 집합의 데이터 개수

$$Gain(S, A) = E(S) - I(S, A) = E(S) - \sum_i \frac{|S_i|}{|S|} \cdot E(S_i)$$

Unit 02 | ID3 알고리즘

예시 ) 야외 스포츠 경기가 열리기 좋은 날씨인가? 아닌가?

의사결정 나무 알고리즘을 활용해봅시다!

$Gain(S, A) = E(S) - I(S, A)$

$$I(S, A) = \sum_i \frac{|S_i|}{|S|} \cdot E(S_i)$$

S : 주어진 데이터들의 집합  
|S| : 주어진 데이터들의 집합  
의 데이터 개수

$E(S)$

$$E(Play) = (-\frac{5}{14} \log_2 \frac{5}{14}) + (-\frac{9}{14} \log_2 \frac{9}{14})$$
$$E(Play) = 0.94$$

$I(S, A)$

$$I(Outlook) = \frac{5}{14} \cdot 0.971 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.971 = 0.693$$

$Gain(S, A) =$   
 $E(S) - I(S, A)$

$$E(Play) - E(Play, Outlook) = 0.25$$

Outlook (날씨)	Temperature (온도)	Humidity (습도)	Windy (바람)	Play (경기 여부)
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

## Unit 02 | ID3 알고리즘

가장 먼저 라벨인 'Play' 변수에 대한 Entropy를 구합니다.

$$Entropy(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

$$E(Play) = \left(-\frac{5}{14} \log_2 \frac{5}{14}\right) + \left(-\frac{9}{14} \log_2 \frac{9}{14}\right)$$

$$E(Play) = 0.94$$

Outlook (날씨)	Temperature (온도)	Humidity (습도)	Windy (바람)	Play (참가여부)
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rainy	mild	high	FALSE	Yes
rainy	cool	normal	FALSE	Yes
rainy	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rainy	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rainy	mild	high	TRUE	No

## Unit 02 | ID3 알고리즘

이번엔 'Outlook' 변수의 각 고유값 구성을 살펴보고 각 class 집합에 대해 Entropy를 구해봅니다.

- Outlook = sunny: 2 examples yes, 3 examples no
- Outlook = overcast: 4 examples yes, 0 examples no
- Outlook = rainy : 3 examples yes, 2 examples no

Outlook (날씨)	Temperature (온도)	Humidity (습도)	Windy (바람)	Play (참가여부)
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rainy	mild	high	FALSE	Yes
rainy	cool	normal	FALSE	Yes
rainy	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rainy	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rainy	mild	high	TRUE	No

## Unit 02 | ID3 알고리즘

이번엔 'Outlook' 변수의 각 고유값 구성을 살펴보고 각 class 집합에 대해 Entropy를 구해봅시다.

- Outlook = sunny: 2 examples yes 3 examples no

$$E(\text{Outlook}=\text{sunny}) = -\frac{2}{5} \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) = 0.971$$

- Outlook = overcast: 4 examples yes, 0 examples no

$$E(\text{Outlook}=\text{overcast}) = -1 \log(1) - 0 \log(0) = 0$$

**Note:** this  
is normally  
undefined.  
Here: = 0

- Outlook = rainy : 3 examples yes 2 examples no

$$E(\text{Outlook}=\text{rainy}) = -\frac{3}{5} \log\left(\frac{3}{5}\right) - \frac{2}{5} \log\left(\frac{2}{5}\right) = 0.971$$

Outlook (날씨)	Temperature (온도)	Humidity (습도)	Windy (바람)	Play (참가여부)
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rainy	mild	high	FALSE	Yes
rainy	cool	normal	FALSE	Yes
rainy	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rainy	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rainy	mild	high	TRUE	No

$$\text{Entropy}(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

## Unit 02 | ID3 알고리즘

이번엔 'Outlook' 변수의 각 고유값 구성을 살펴보고 각 class 집합에 대해 Entropy를 구해봅시다.

엔트로피로 단일 집합의 품질을 계산했다.  
전체에서의 품질은 어떻게 계산할 수 있을까?

- Outlook = sunny: 2 examples yes, 3 examples no

$$E(\text{Outlook}=\text{sunny}) = -\frac{2}{5} \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) = 0.971$$

- Outlook = overcast: 4 examples yes, 0 examples no

$$E(\text{Outlook}=\text{overcast}) = -1 \log(1) - 0 \log(0) = 0$$

가중치를 고려한 평균을 이용하자!

How: this is normally undefined. Here: = 0

- Outlook = rainy: 3 examples yes, 2 examples no

$$E(\text{Outlook}=\text{rainy}) = -\frac{3}{5} \log\left(\frac{3}{5}\right) - \frac{2}{5} \log\left(\frac{2}{5}\right) = 0.971$$

Outlook (날씨)	Temp (온도)	Humidity (습도)	Windy (바람)	Play (참가여부)
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rainy	mild	high	FALSE	Yes
rainy	cool	normal	FALSE	Yes
rainy	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rainy	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rainy	mild	high	TRUE	No

$$Entropy(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

Unit 02 | ID3 알고리즘

이번엔 'Outlook' 변수의 각 고유값 구성을 살펴보고 각 class 집합에 대해 Entropy를 구해봅니다.

$$I(S, A) = \sum_i \frac{|S_i|}{|S|} \cdot E(S_i)$$

S : 주어진 데이터들의 집합

|S| : 주어진 데이터들의 집합의 데이터 개수

$$I(\text{Outlook}) = \frac{5}{14} \cdot 0.971 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.971 = 0.693$$

sunny

overcast

rain

Outlook (날씨)	Temperature (온도)	Humidity (습도)	Windy (바람)	Play (참가여부)
sunny	hot	high	FALSE	No
sunny	hot	high	TRUE	No
overcast	hot	high	FALSE	Yes
rainy	mild	high	FALSE	Yes
rainy	cool	normal	FALSE	Yes
rainy	cool	normal	TRUE	No
overcast	cool	normal	TRUE	Yes
sunny	mild	high	FALSE	No
sunny	cool	normal	FALSE	Yes
rainy	mild	normal	FALSE	Yes
sunny	mild	normal	TRUE	Yes
overcast	mild	high	TRUE	Yes
overcast	hot	normal	FALSE	Yes
rainy	mild	high	TRUE	No



## Unit 02 | ID3 알고리즘

=> Information Gain을 구해보자

상위 노드의 Entropy에서 하위노드의 Entropy를 뺀 값!

$$Gain(S, A) = E(S) - I(S, A) = E(S) - \sum_i \frac{|S_i|}{|S|} \cdot E(S_i)$$

$$E(\text{Play}, \text{Outlook}) = 0.6935$$

$$E(\text{Play}, \text{Temp}) = 0.911$$

$$E(\text{Play}, \text{Humidity}) = 0.7884$$

$$E(\text{Play}, \text{Windy}) = 0.8921$$



$$E(\text{Play}) - E(\text{Play}, \text{Outlook}) = 0.25$$

$$E(\text{Play}) - E(\text{Play}, \text{Temp}) = 0.02$$

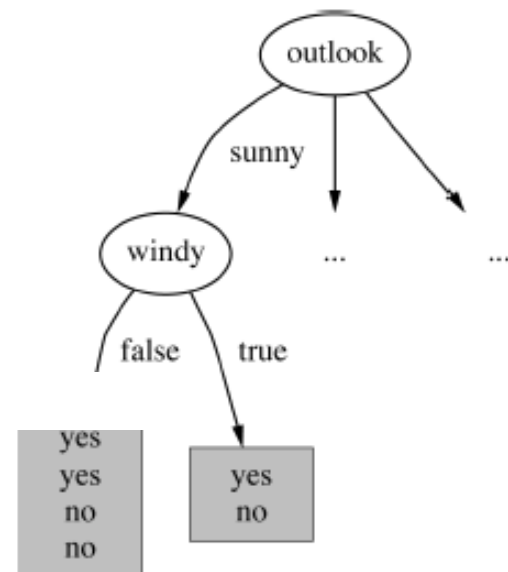
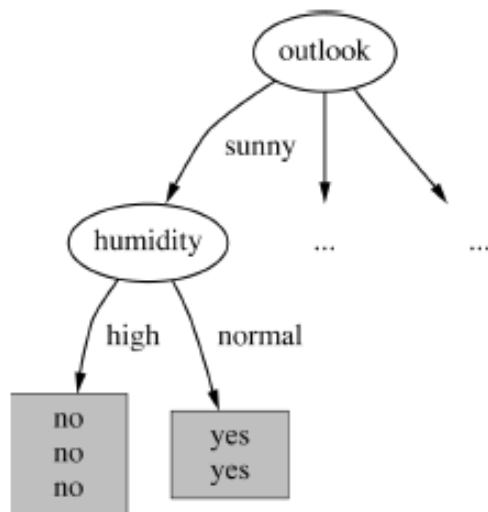
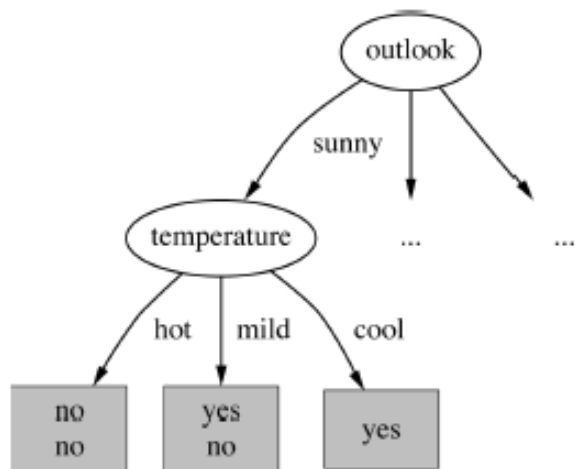
$$E(\text{Play}) - E(\text{Play}, \text{Humidity}) = 0.1514$$

$$E(\text{Play}) - E(\text{Play}, \text{Windy}) = 0.047$$

Information Gain 값이 가장 큰,  
즉, Entropy를 가장 많이 줄인 **Outlook** 변수를  
첫 번째 기준으로 **채택!!**

## Unit 02 | ID3 알고리즘

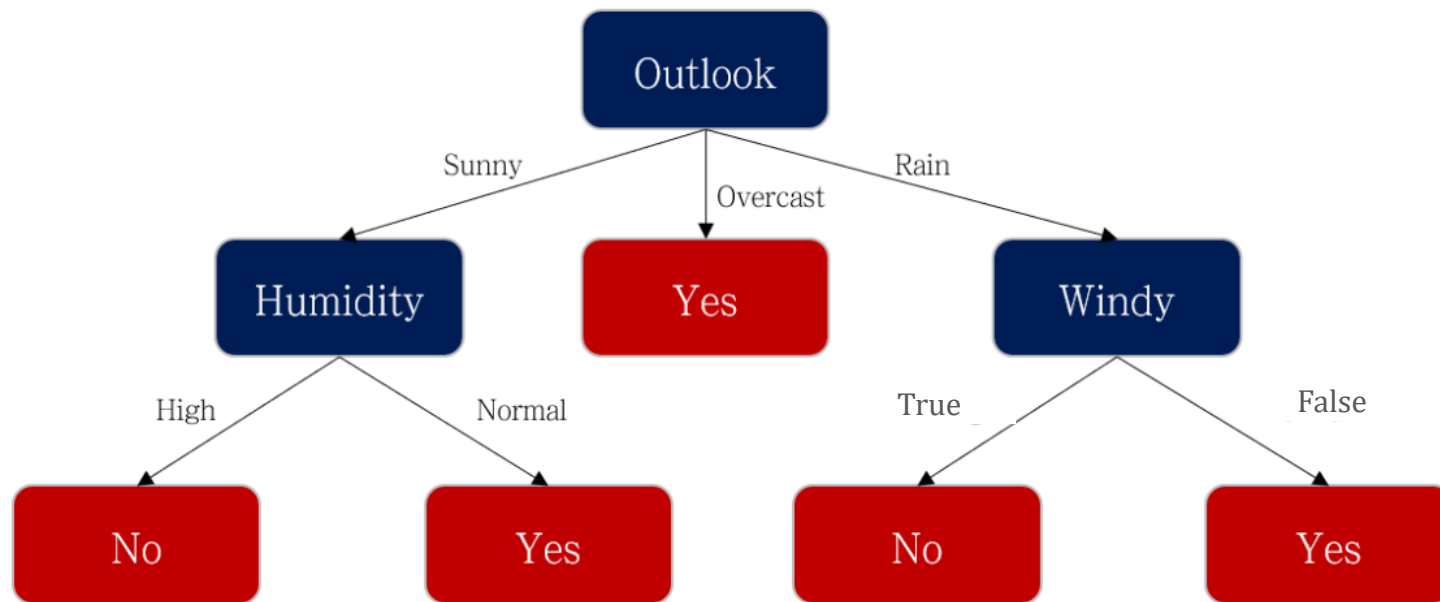
$$E(\text{Outlook}=\text{sunny}) = -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right) = 0.971$$



$\text{Gain}(\text{Temperature}) = 0.571 \text{ bits}$   
 $\text{Gain}(\text{Humidity}) = 0.971 \text{ bits}$   
 $\text{Gain}(\text{Windy}) = 0.020 \text{ bits}$

**Humidity is selected**

## Unit 02 | ID3 알고리즘



최종적으로 위와 같은 Decision Tree model이 만들어진다!!

## Unit 03 | CART 알고리즘

Q. 불순도를 측정하는 지표는?

A. Entropy, Gini index

Q. 어떤 기준으로 노드를 놓아야 하며, 어떤 노드를 가장 위에 놓아야 할까?

A. ID3 & CART 알고리즘

## Unit 03 | CART 알고리즘

## • 순도 / 불순도 지표

Gini index (지니지수)



- 데이터의 통계적 분산정도를 정량화 해서 표현한 값
- 어떤 집합의 gini index가 높을수록 그 집단의 데이터가 분산되어있다.
- 우리의 목적 : gini index를 감소시키는 방향으로 분류하기

Gini index 감소 = 불순도 감소 = 순도 증가

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

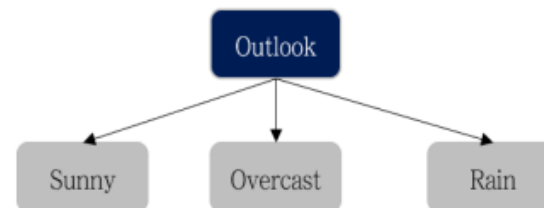
$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

## Unit 03 | CART 알고리즘

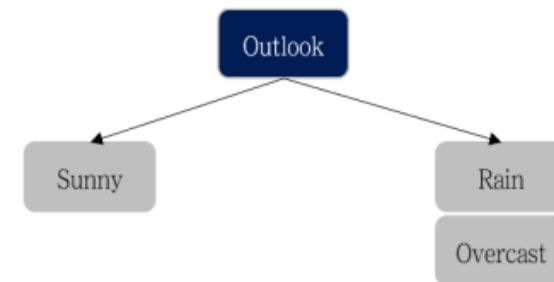
- CART 알고리즘

- Gini index 를 이용한 알고리즘
- 데이터를 split 했을 때의 불순한 정도
- **Binary split**을 전제로 분석함
- 데이터의 대상 속성을 얼마나 잘못 분류할지 계산

ID3



CART



## Unit 03 | CART 알고리즘

Gini index (지니지수) 계산

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

## Unit 03 | CART 알고리즘

## Gini index (지니지수) 계산

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

Ex. 컴퓨터를 살 것인가? (나이, 소득, 학생 여부, 신용등급에 의해 Yes or No 판단)

선택한 class의 Gini  
(여기선 age의 youth)

$$1 - P(\text{Yes} \mid \text{age}=\text{youth})^2 - P(\text{No} \mid \text{age}=\text{youth})^2$$

$$Gini_{\text{age}=\text{youth}} = \frac{5}{14} \left( 1 - \frac{2^2}{5} - \frac{3^2}{5} \right) + \frac{9}{14} \left( 1 - \frac{7^2}{9} - \frac{2^2}{9} \right) = 0.394$$

나머지 class의 Gini  
(youth 제외)

$$1 - P(\text{Yes} \mid \text{age} \neq \text{youth})^2 - P(\text{No} \mid \text{age} \neq \text{youth})^2$$



Unit 03 | CART 알고리즘

예시 )   컴퓨터를 살 것인가? (나이, 소득, 학생 여부, 신용등급에 의해 판단)

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

- Age

$\text{Gini}_{age}(D)$
- Credit

$\text{Gini}_{credit}(D)$
- Income

$\text{Gini}_{income}(D)$
- Student

$\text{Gini}_{student}(D)$

이중 가장 작은 Gini index 값을 가지는 변수로 최초 split 이 됨.

Unit 03 | CART 알고리즘

예시 )   컴퓨터를 살 것인가? (나이, 소득, 학생 여부, 신용등급에 의해 판단)

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Age

Gini<sub>age</sub>(D)

Credit

Gini<sub>credit</sub>(D)

Income

Gini<sub>income</sub>(D)

Student

Gini<sub>student</sub>(D)

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

# Unit 03 | CART 알고리즘

예시 )    컴퓨터를 살 것인가? (나이, 소득, 학생 여부, 신용등급에 의해 판단)

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Age

Gini<sub>age</sub>(D)

Gini<sub>age=youth</sub>

Gini<sub>age=middle\_aged</sub>

Gini<sub>age=senior</sub>

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

# Unit 03 | CART 알고리즘

예시 )   컴퓨터를 살 것인가? (나이, 소득, 학생 여부, 신용등급에 의해 판단)

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Age   Gini<sub>age</sub>(D)

$$Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

$$Gini_{age=youth} = \frac{5}{14} \left(1 - \frac{2^2}{5} - \frac{3^2}{5}\right) + \frac{9}{14} \left(1 - \frac{7^2}{9} - \frac{2^2}{9}\right) = 0.394$$

$$Gini_{age=middle\_aged} = \frac{4}{14} \left(1 - \frac{4^2}{4} - \frac{0^2}{4}\right) + \frac{10}{14} \left(1 - \frac{5^2}{10} - \frac{5^2}{10}\right) = \underline{0.357}$$

$$Gini_{age=senior} = \frac{5}{14} \left(1 - \frac{2^2}{5} - \frac{3^2}{5}\right) + \frac{9}{14} \left(1 - \frac{6^2}{9} - \frac{3^2}{9}\right) = 0.457$$

## Unit 03 | CART 알고리즘

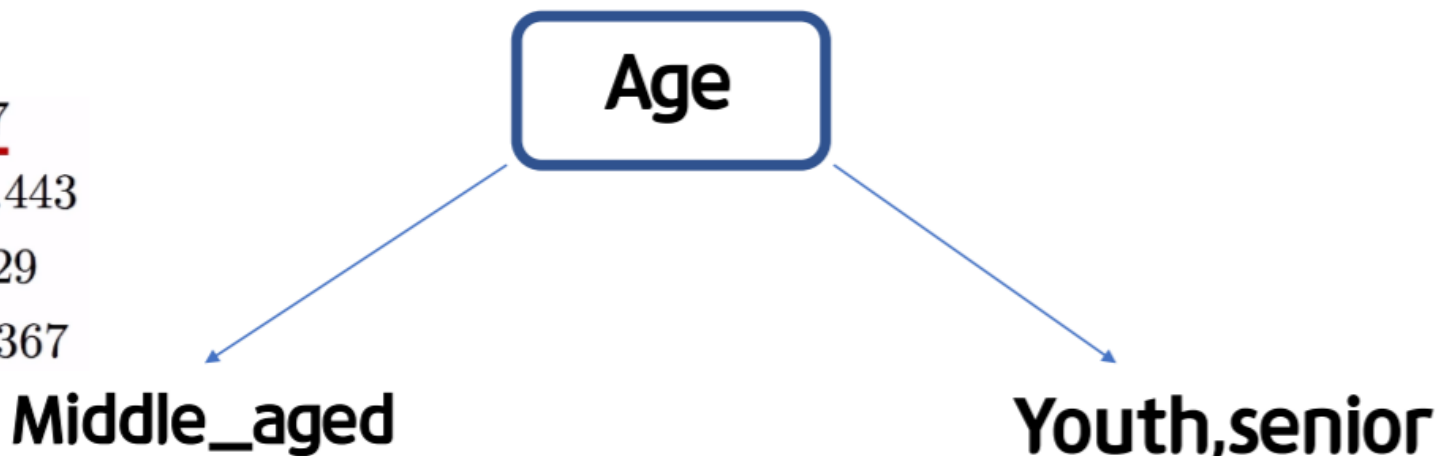
## Gini Index

$$\underline{\text{Min}(Gini_{age_i}) = 0.357}$$

$$\text{Min}(Gini_{income_i}) = 0.443$$

$$\text{Min}(Gini_{credit}) = 0.429$$

$$\text{Min}(Gini_{student}) = 0.367$$



	age	income	student	credit_rating	class_buys_computer
2	middle_aged	high	no	fair	yes
6	middle_aged	low	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
13	senior	medium	no	excellent	no

## Unit 04 | feature가 연속형이라면?

Q. 연속형 변수에 의사결정 나무를 사용하는 방법?

1. 전체 데이터를 모두 기준으로 분할 후 불순도를 계산한다.
2. 중위수, 사분위수를 기준으로 한다.
3. Label 의 class 가 바뀌는 수를 기준으로 한다.

## Unit 04 | feature가 연속형이라면?

STEP 1. 각 Feature에 대해 오름차순으로 정렬

INCOME	LOTSIZE	OWNERSHIP
43.2	17.2	X
49.2	17.6	X
52.8	19.6	X
59.4	17.6	X
60	18.4	O
61.5	21	O
64.8	21.6	O
65	20.8	X
84	20.4	X
85.8	16.8	O
87	23.6	O
110.1	19.2	O

STEP 2. Label의 class가 변하는 지점을 찾기

INCOME	LOTSIZE	OWNERSHIP
43.2	17.2	X
49.2	17.6	X
52.8	19.6	X
59.4	17.6	X
60	18.4	O
61.5	21	O
64.8	21.6	O
65	20.8	X
84	20.4	X
85.8	16.8	O
87	23.6	O
110.1	19.2	O

# Unit 04 | feature가 연속형이라면?

## STEP 3. 경계의 평균값을 기준값으로 잡기

INCOME	LOTSIZE	OWNERSHIP
43.2	17.2	X
49.2	17.6	X
52.8	19.6	X
59.4	17.6	X
60	18.4	O
61.5	21	O
64.8	21.6	O
65	20.8	X
84	20.4	X
85.8	16.8	O
87	23.6	O
110.1	19.2	O

59.7

64.9

84.9

## STEP 4. 각 기준점에 대해 분할 후, Gini index 혹은 Entropy 계산

$$\begin{aligned}
 G(\text{income}_{59.7}) &= \mathbf{0.25} & G(\text{lotsize}_{17}) &= 0.455 \\
 G(\text{income}_{64.9}) &= 0.486 & G(\text{lotsize}_{18}) &= 0.438 \\
 G(\text{income}_{84.9}) &= 0.333 & G(\text{lotsize}_{19.4}) &= 0.5 \\
 & & G(\text{lotsize}_{20.9}) &= 0.333
 \end{aligned}$$



## Unit 04 | feature가 연속형이라면?



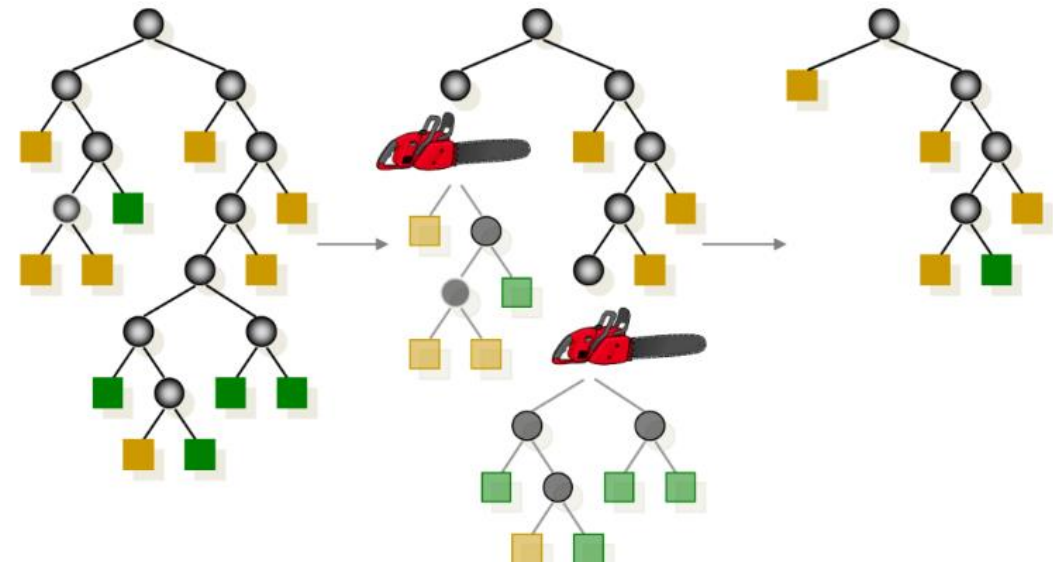
위와 같은 과정을 통해 첫 번째 기준점을 정한다.

또한 이 과정을 반복하면 최종 Decision Tree model을 만들 수 있다.

## Unit 05 | 가지치기

### 가지치기란?

- 모든 terminal node의 순도가 100% 인 상태를 Full tree 라고 한다. 이런 경우, 분기가 너무 많아 과적합 위험이 발생한다.
- 분기가 지나치게 증가할 경우 일반화 능력이 떨어지게 된다.
- 이를 방지하기 위해 의사결정나무에서 과적합을 방지하기 위해 적절한 수준에서 terminal node를 결합해 주는 것



## Unit 05 | 가지치기

### 가지치기의 종류?

- Pre pruning (사전 가지치기) |  
트리의 최대 depth 나 분기점의 최소 개수를 미리 지정
- Post pruning (사후 가지치기) |  
트리를 만든 후 데이터 포인트가 적은 노드를 삭제/병합

## 정리

### 장점

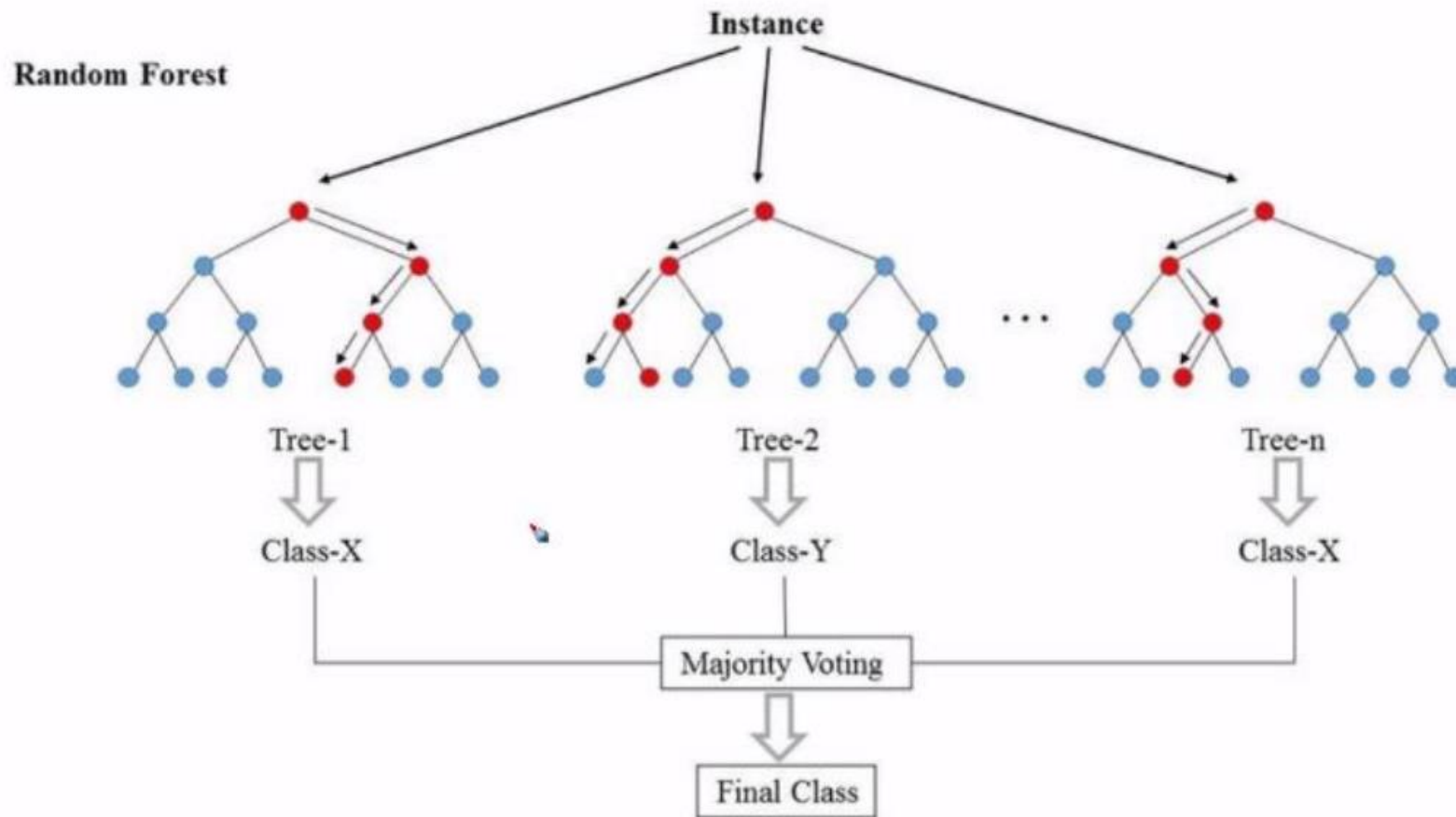
1. 결과를 해석하고 이해하기 용이하다.
2. 비모수적 모형이기 때문에 선형성, 정규성, 등분산성 등의 가정이 필요하지 않다.
3. 데이터를 가공할 필요가 거의 없다.

### 단점

1. 연속형 변수를 비연속적 값으로 취급하기 때문에 분리의 경계점 부근에서 예측오류가 클 가능성이 있다.
2. 데이터의 특성이 특정 변수에 수직/수평적으로 구분되지 못할 때 성능이 떨어지고 트리가 복잡해진다.
3. Overfitting 문제가 발생하기 쉽다.
4. 중간 단계에서 오류가 발생하면 다음 단계로 에러가 계속 전파된다.
5. 적은 개수의 노이즈에도 크게 영향을 받는다.

## 정리

단점 해결방안  
=> 앙상블 (Ensemble)



## 과제 및 데이터 설명

### 과제

week3\_DT\_assignment.ipynb를 완성해주세요!!

## Unit 06 | 과제

본인이 구현한 함수를 이용해 다음 문제를 풀어주세요.

1. 변수 income의 이진분류 결과를 보여주세요
2. 분류를 하는데 가장 중요한 변수를 선정하고, 해당 변수의 gini index를 제시해주세요.
3. 문제 2에서 제시한 feature로 dataframe을 split 하고, 나뉜 2개의 dataframe에서 각각 다음으로 중요한 변수를 선정하고 해당 변수의 gini index를 제시해주세요.  
(변수나 flow는 변경해도 무관합니다. 결과만 똑같이 나오면 됩니다.)

**\*\* 주의사항**

이 데이터셋 뿐만 아니라 변수의 class가 더 많은 데이터에도 상관없이 적용 가능하도록 구현해주세요.  
변수의 class 가 3개를 넘는 경우 모든 이진 분류의 경우의 수를 따져보아야 합니다.

## 과제 참고

$$\text{get\_gini(df, label)} \longrightarrow Gini(D_i) = 1 - \sum_{j=1}^x P_j^2$$

$$\text{get\_attribute\_gini\_index(df, attribute, label)} \longrightarrow Gini(A) = \sum_{j=1}^2 \frac{|D_j|}{|D|} * Gini(D_i)$$



## 참고자료

- <https://ratsgo.github.io/machine%20learning/2017/03/26/tree/>
- <https://tensorflow.blog/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/2-3-5-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%AC/>
- [https://yngie-c.github.io/machine%20learning/2020/04/06/decision\\_tree/](https://yngie-c.github.io/machine%20learning/2020/04/06/decision_tree/)
- <https://leedakyeong.tistory.com/category/%ED%86%B5%EA%B3%84%20%EC%A7%80%EC%8B%9D/Algorithm>
- <https://jihoonlee.tistory.com/16>
- <https://soobarkbar.tistory.com/17>
- Tobigs 13기 김미성님 강의자료
- Tobigs 15기 김동현님 강의자료



Q & A

들어주셔서 감사합니다.