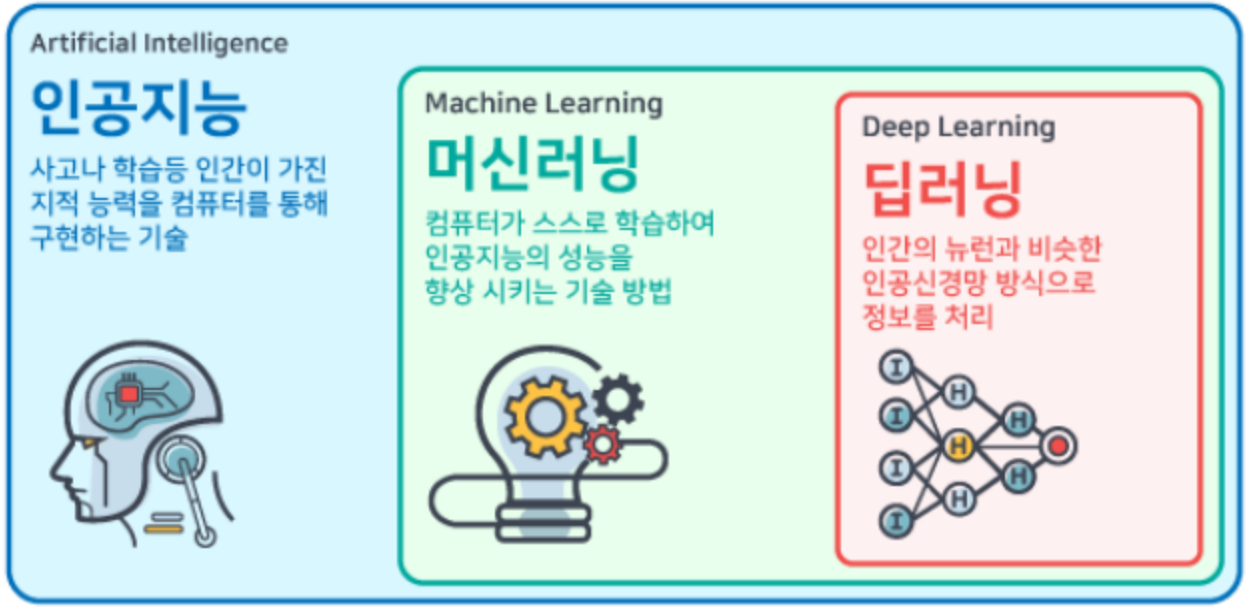
DSL 5기 박채은

Decision Tree

Machine learning

머신러닝(기계학습)

인공지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야(위키피디아)



Machine learning에 대한 아주 기본적인 설명

- 대표적인 과제 종류(데이터 제공 방식)에 따른 머신러닝의 종류

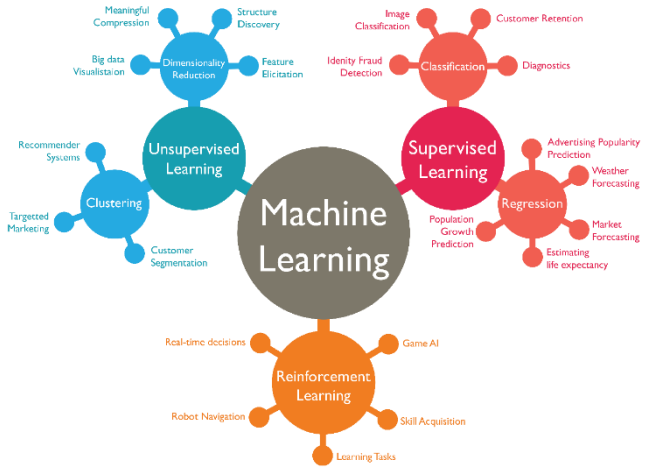
지도학습 | 비지도학습 | 강화학습

1. 지도학습

Y = f(X)에 대하여 입력 변수(X)와 출력 변수(Y)의 관계에 대하여 모델링하는 것

- 회귀(regression): 입력 변수 X에 대하여 연속형 출력 변수 Y를 예측
- 분류(classification): 입력 변수 X에 대하여 이산형 출력 변수 Y(class)를 분류

예) 숫자 손글씨 데이터와 숫자 라벨로 숫자 이미지 분류 모델 만들기



2. 비지도학습

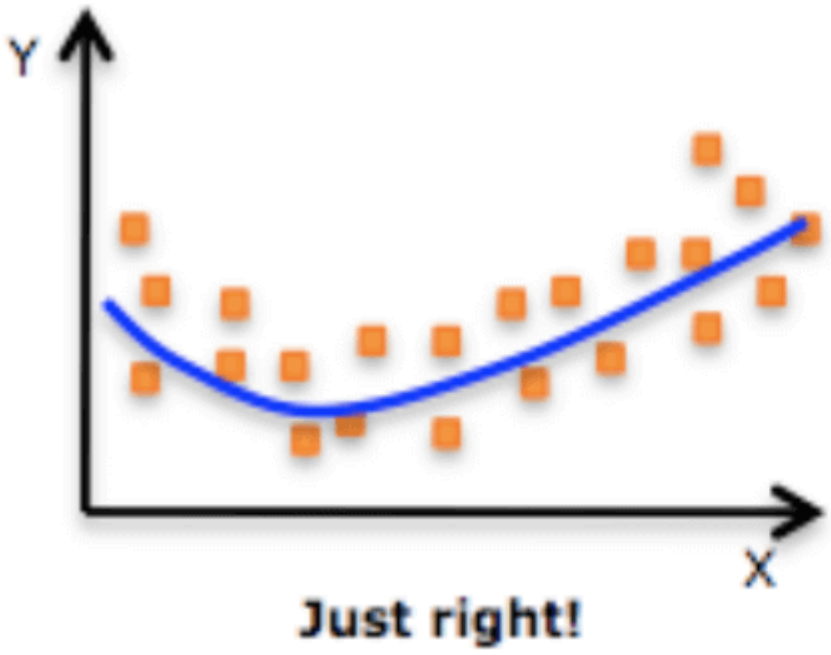
출력 변수 Y가 존재하지 않고, 입력 변수 X만을 가지고 X 내에서 데이터의 관계와 특징을 찾아내는 것

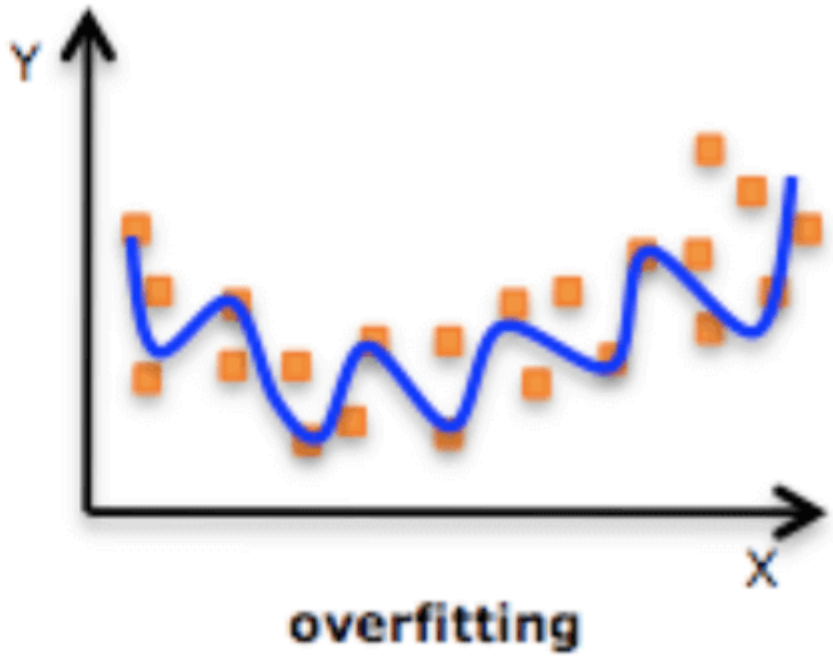
예) 군집 분석: 유사한 데이터끼리 그룹화

PCA: 독립 변수들의 차원을 축소화

- 오버피팅(overfitting)

학습 데이터의 지엽적인 특성까지 과하게 학습하여 새로운 데이터를 잘 예측하지 못하는 것.

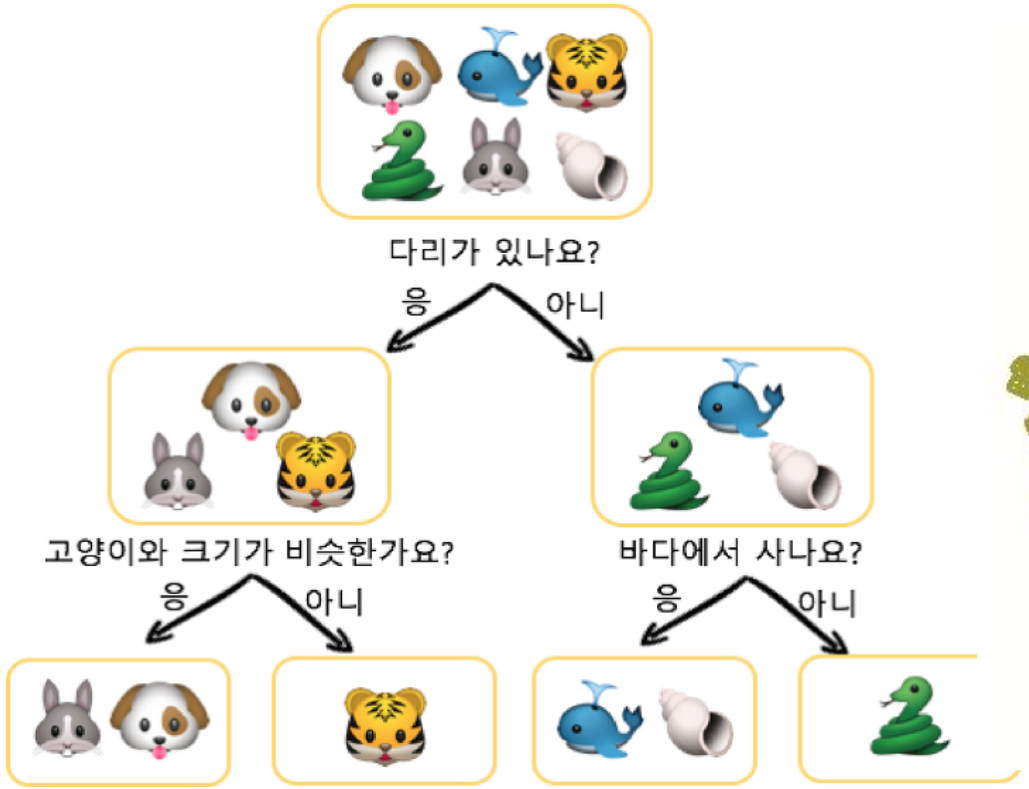


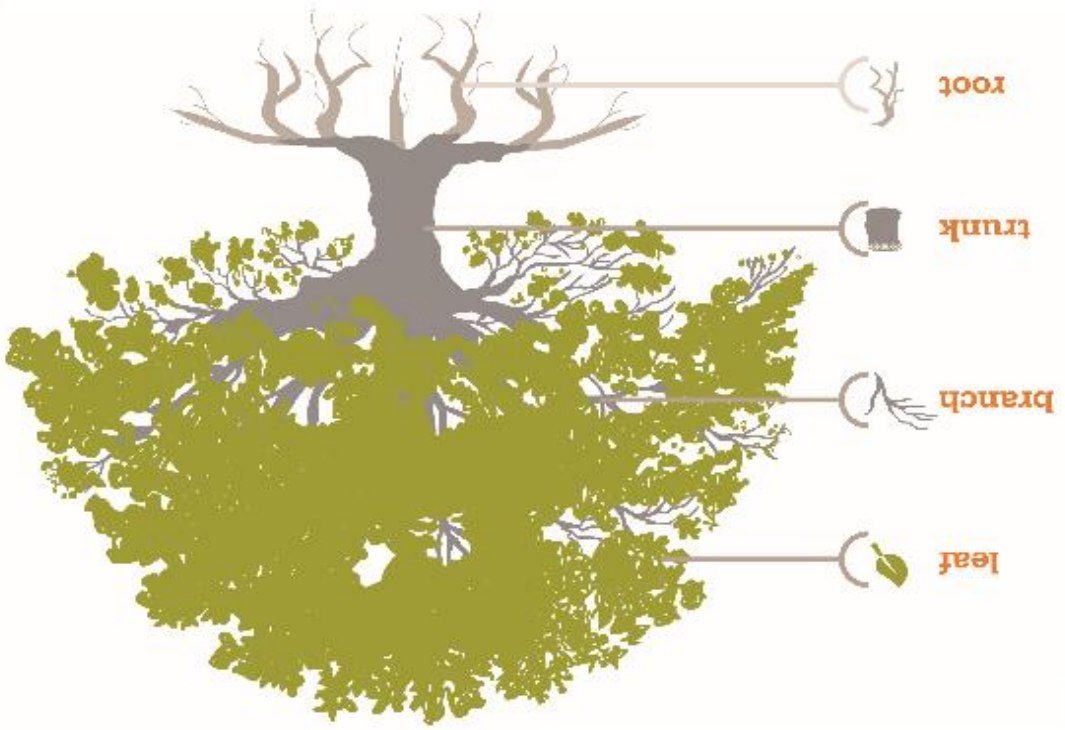


Decision tree

Decision tree

- 스무고개처럼 질문을 통해 대상을 좁혀 나가며 학습



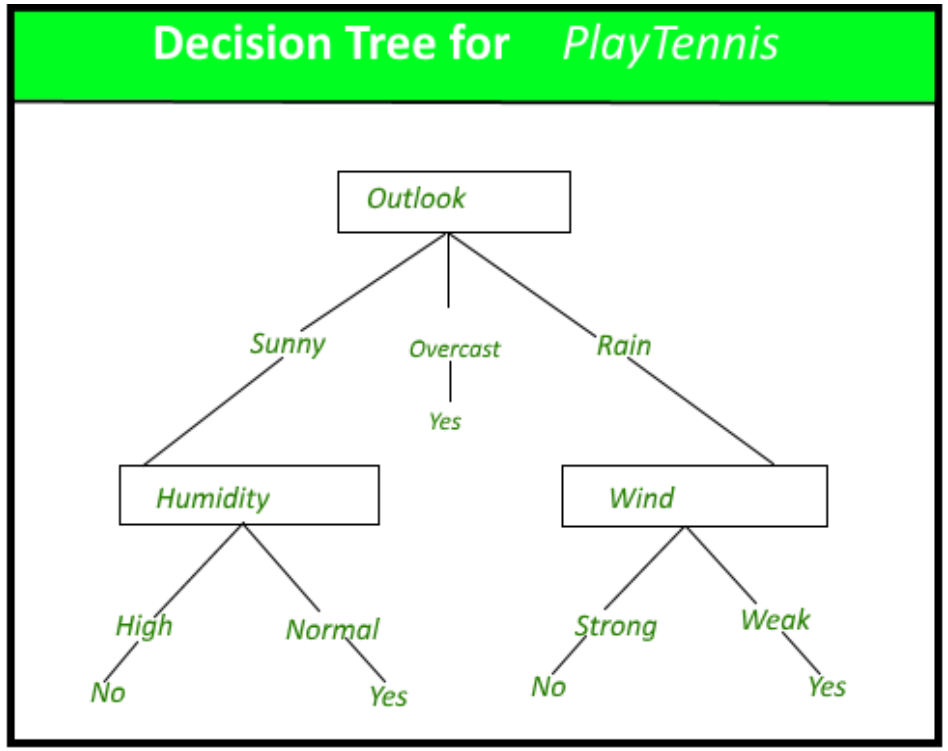


출처: 브런치

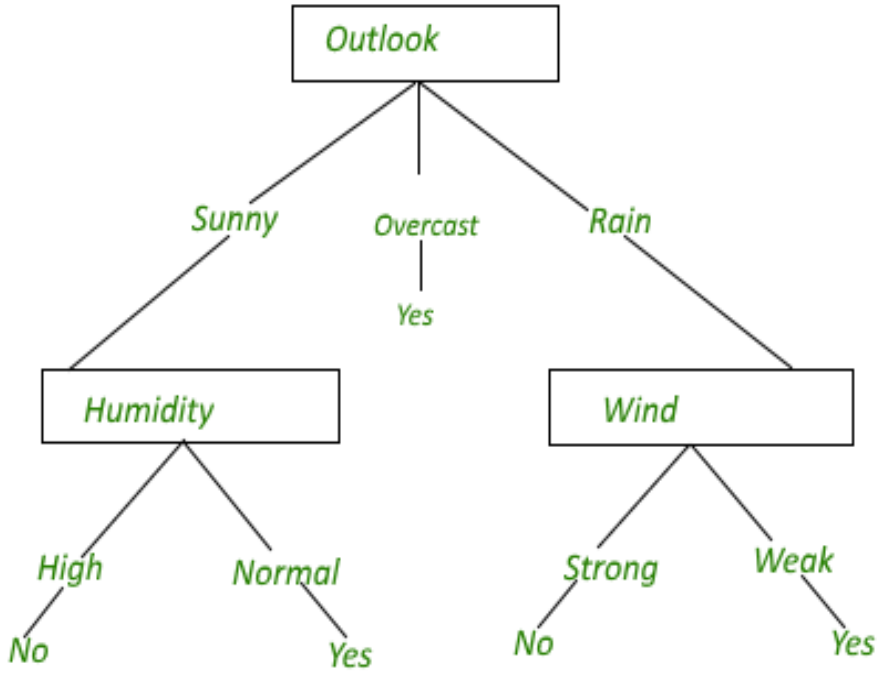
Decision tree(=결정트리, 의사결정트리, 의사결정나무)

- 분류와 회귀 모두 가능한 지도학습 모델 중 하나
- 변수들로 기준을 만들고, 기준을 통하여 샘플을 분류 -> 분류된 집단의 성질을 통하여 종속변수 Y를 예측하는 모형

Day	Outlook	Temperature	Humidity	Windy	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

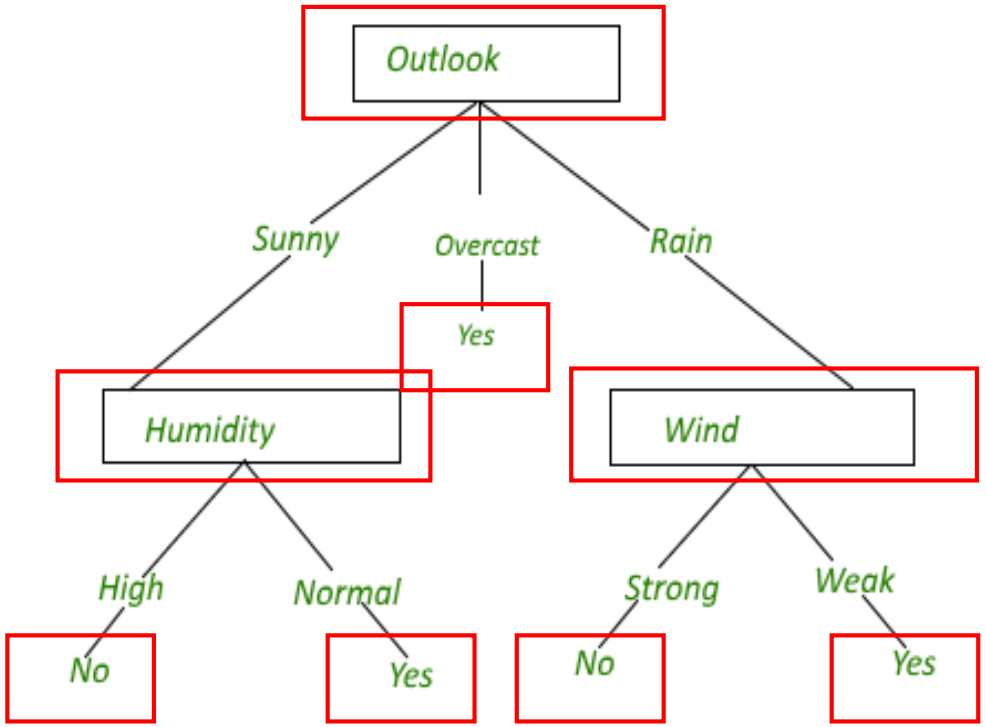


Decision tree 용어



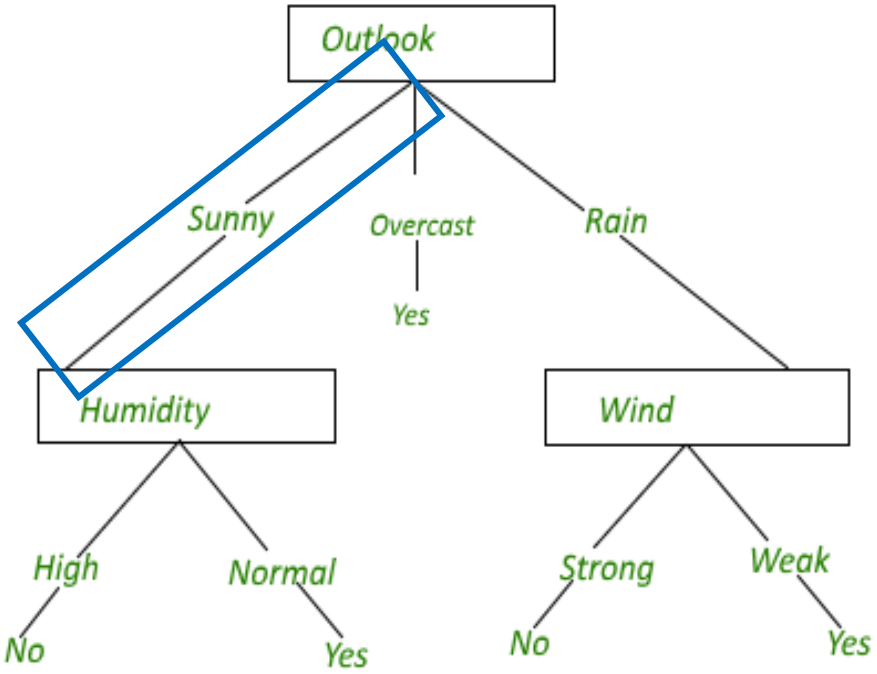
- Node: 질문이나 정답이 위치
  - Parent node: child node의 상위 노드
  - Child node: parent node의 하위 노드
  - Root node: 가장 위의 노드
  - Leaf node(terminal node): 하위 노드가 없는 가장 아래의 노드
- Edge: 샘플을 분류하는 조건이 위치
- Depth: root node에서 특정 노드까지 도달하기 위해 거쳐야하는 edge의 수

Decision tree 용어



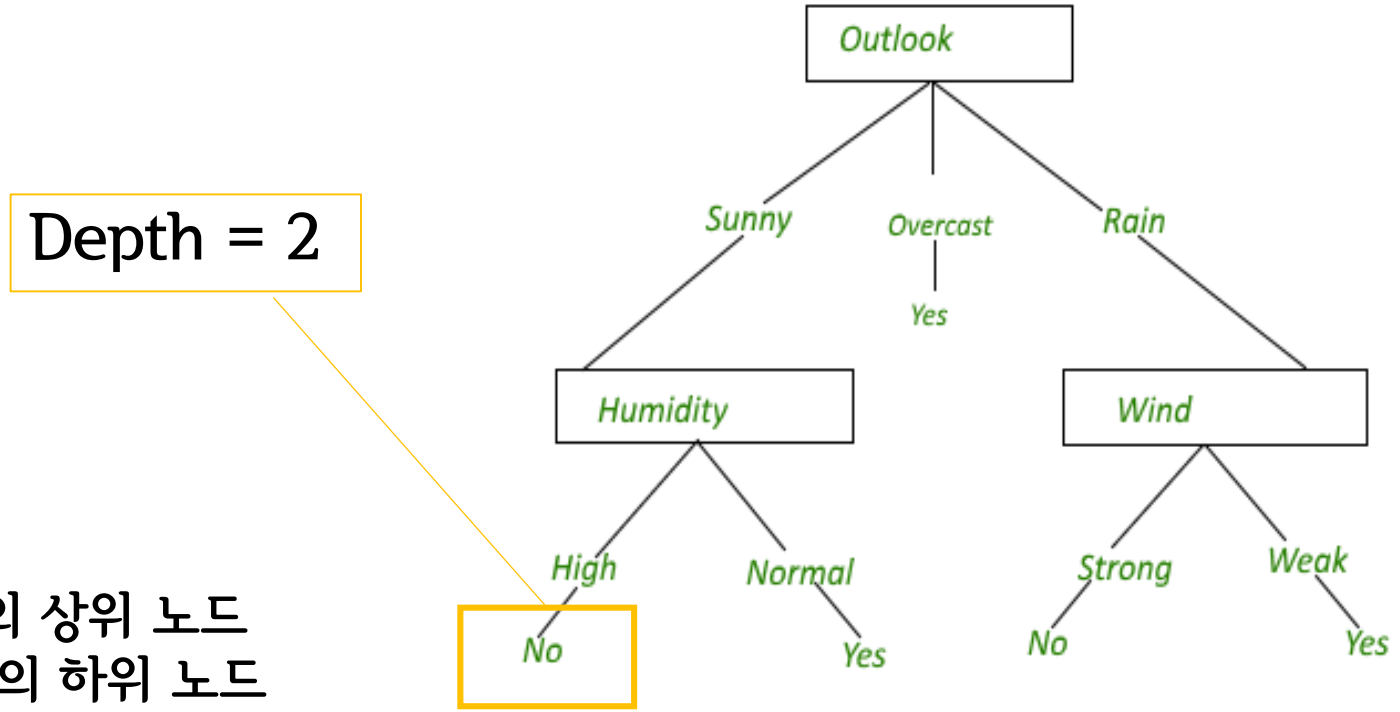
- Node: 질문이나 정답이 위치
  - Parent node: child node의 상위 노드
  - Child node: parent node의 하위 노드
  - Root node: 가장 위의 노드
  - Leaf node(terminal node): 하위 노드가 없는 가장 아래의 노드
- Edge: 샘플을 분류하는 조건이 위치
- Depth: root node에서 특정 노드까지 도달하기 위해 거쳐야하는 edge의 수

Decision tree 용어



- Node: 질문이나 정답이 위치
  - Parent node: child node의 상위 노드
  - Child node: parent node의 하위 노드
  - Root node: 가장 위의 노드
  - Leaf node(terminal node): 하위 노드가 없는 가장 아래의 노드
- Edge: 샘플을 분류하는 조건이 위치
- Depth: root node에서 특정 노드까지 도달하기 위해 거쳐야하는 edge의 수

Decision tree 용어



- Node: 질문이나 정답이 위치
  - Parent node: child node의 상위 노드
  - Child node: parent node의 하위 노드
  - Root node: 가장 위의 노드
  - Leaf node(terminal node): 하위 노드가 없는 가장 아래의 노드
- Edge: 샘플을 분류하는 조건이 위치
- Depth: root node에서 특정 노드까지 도달하기 위해 거쳐야하는 edge의 수

CART(Classification and Regression Tree)

- 트리 알고리즘은 여러 방식이 존재(CART, C4.5 등)

- CART는 의사결정나무분석을 형성하는데 있어 가장 보편적인 알고리즘

- scikit-learn에서도 CART 알고리즘 사용함.

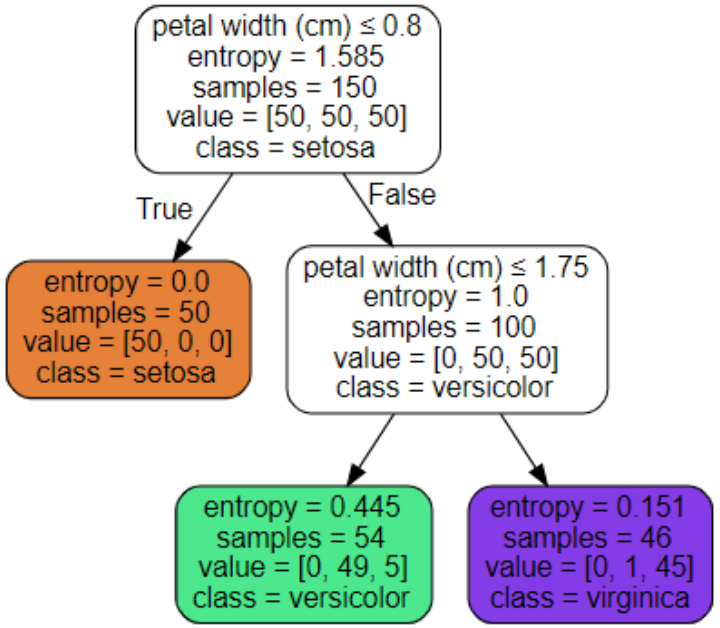
<특성>
- 분류와 회귀 모델 모두 사용 가능
- Binary tree로 모형 형성

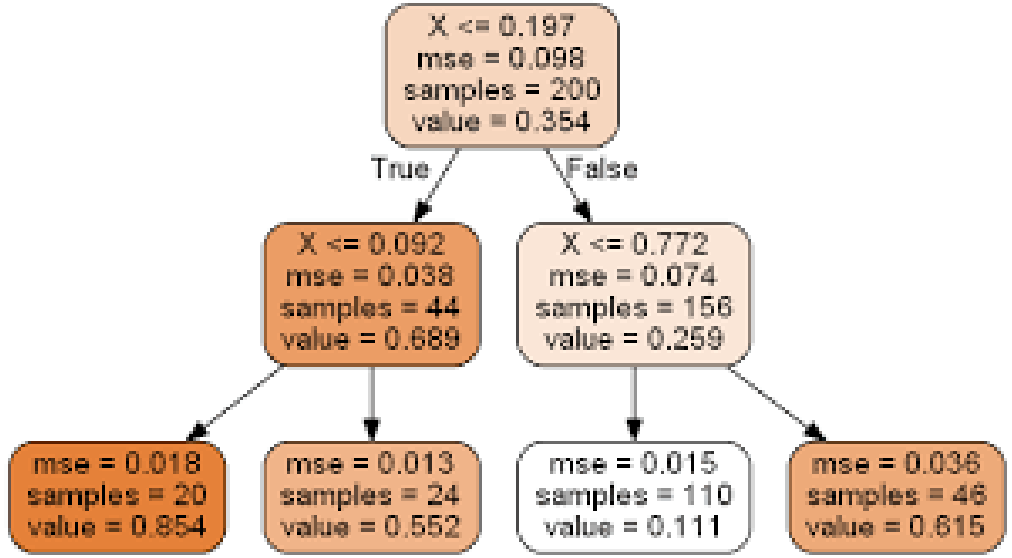
(이 외에도 알고리즘 별로 가지치기 방식 등도 달라짐.)

Decision tree 구분

- 종속변수에 따라 구분

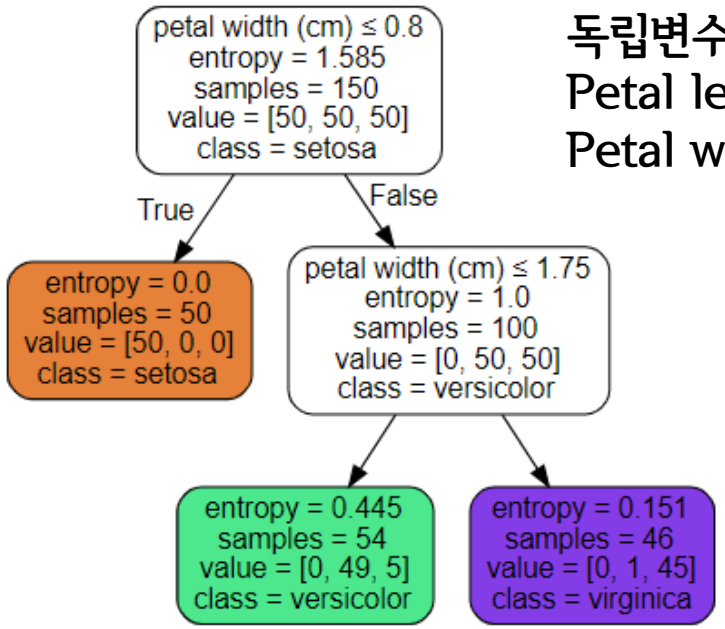
종속변수가
- 범주형 변수: 분류 트리(classification tree)
- 연속형 변수: 회귀 트리(regression tree)



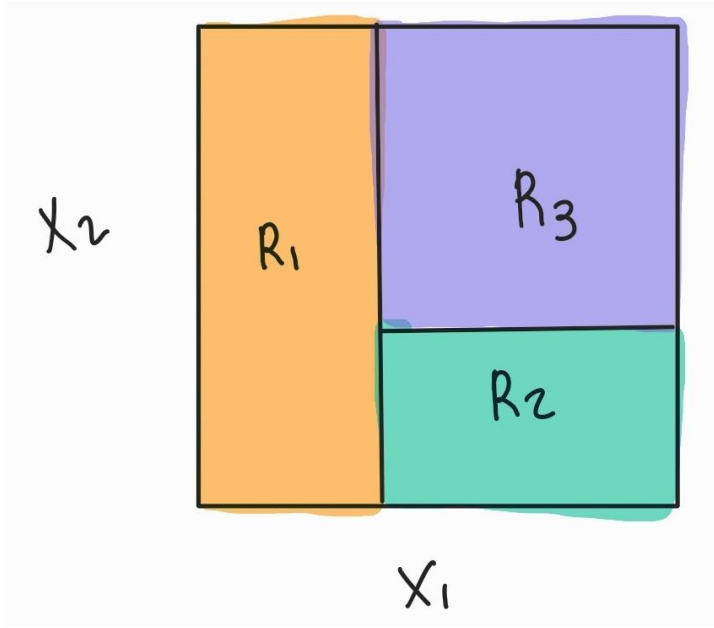


classification tree

Classification tree



독립변수:
Petal length(X1)
Petal width(X2)



Tree 조건에 따라 독립변수의 공간(X가 가질 수 있는 영역)을 block으로 나누는 개념
-> 나누어진 영역 안에 속하는 샘플의 특성을 통하여 Y를 예측

영역 나누기 위한 Tree 조건 어떻게 설정할까?

- 독립변수와 기준을 정해야 함. 예) petal length <= 2.45

변수 기준

- 독립변수는 범주형일 수도, 연속형일 수도 있음.
- 독립변수가 범주형인 경우
  각 범주에 따라 영역 나눔.
- 독립변수가 연속형인 경우
  어떤 분기점에 의해 영역 나눔.

어떤 독립변수와 기준을 조건으로 먼저 설정할까?

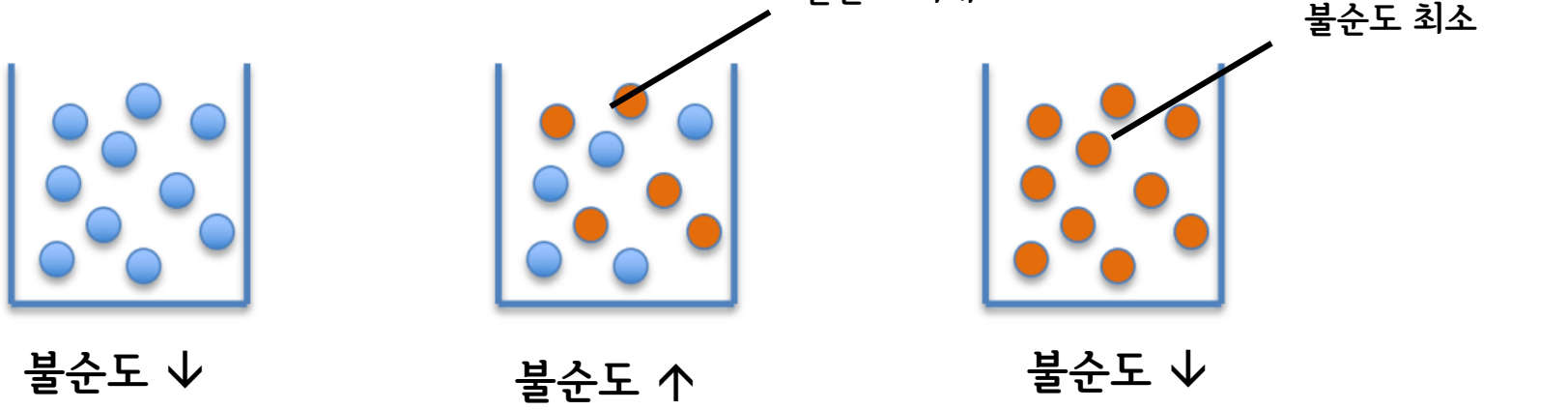
Classification tree

어떤 독립변수와 기준을 조건으로 먼저 설정할까?

데이터를 가장 잘 구분할 수 있는 조건을 설정해야 함.
불순도 낮추는 방향으로 설정!!

불순도(Impurity)

해당 영역 안에 다양한 범주의 개체들이 얼마나 포함되어 있는가



불순도를 수치화한 척도: entropy, Gini index 등

결정트리는 불순도를 최소화하는 방향으로 학습 진행 -> 즉 entropy와 Gini index를 최소화하는 방향으로 학습 진행

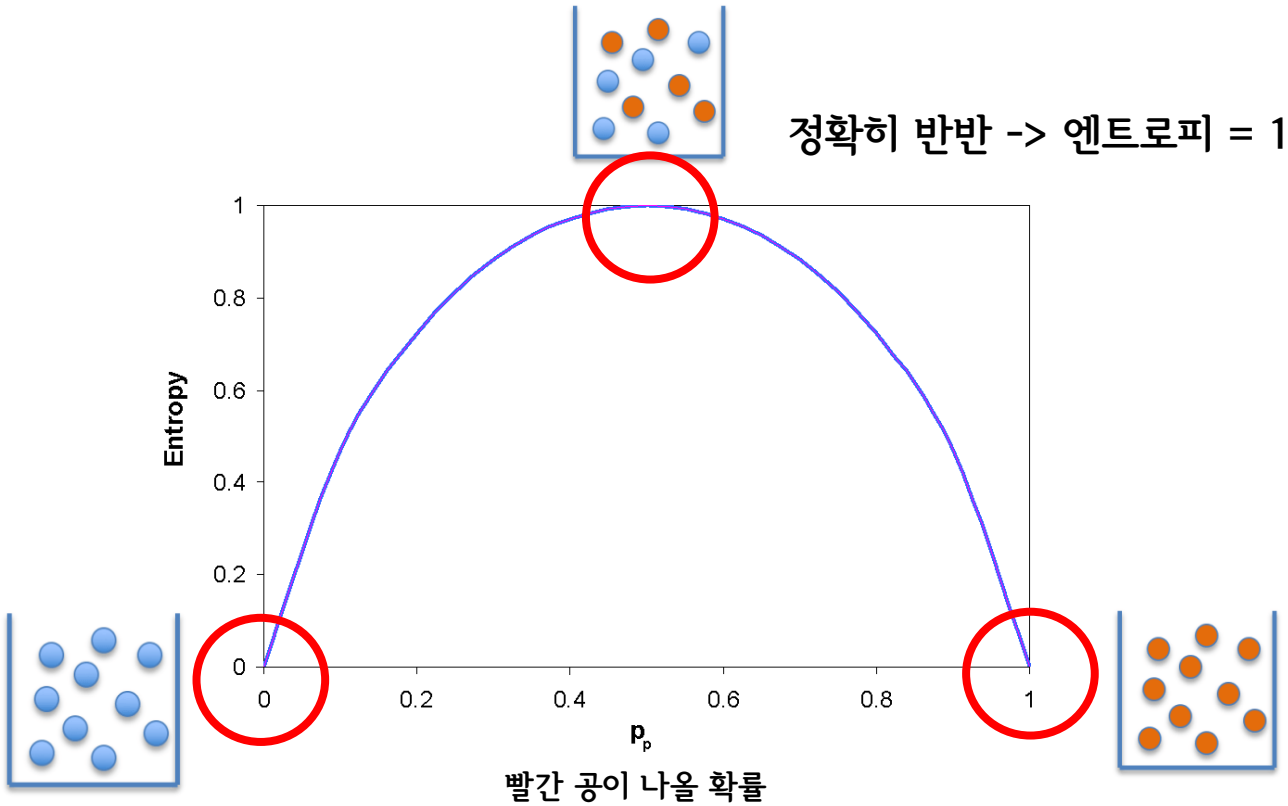
Classification tree

불순도를 수치화한 척도(1)

엔트로피(entropy)

$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$

p_i: 한 영역 안에 존재하는 데이터 가운데 범주 i에 속하는 데이터의 비율



Classification tree

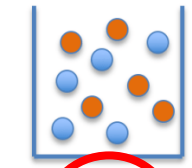
불순도를 수치화한 척도(1)

엔트로피(entropy)

$$\text{Entropy} = -\sum_i (p_i)\log_2(p_i)$$

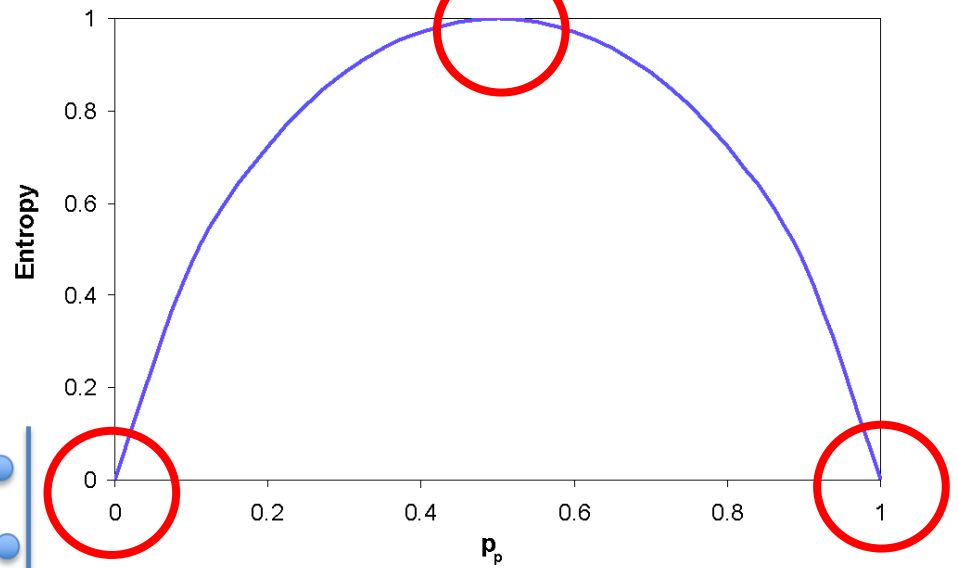
p_i: 한 영역 안에 존재하는 데이터 가운데 범주 i에 속하는 데이터의 비율

$p_{\text{빨간공}} = 0.5$
$p_{\text{파란공}} = 0.5$



$\text{Entropy} = -0.5 * \log_2(0.5) - 0.5 * \log_2(0.5) = 1$

정확히 반반 -> 엔트로피 = 1



$\text{Entropy} = -1 * \log_2 1 = 0$

$p_{\text{파란공}} = 1$

$\text{Entropy} = -1 * \log_2 1 = 0$

$p_{\text{빨간공}} = 1$

엔트로피(entropy)

- 불순도를 수치화한 척도

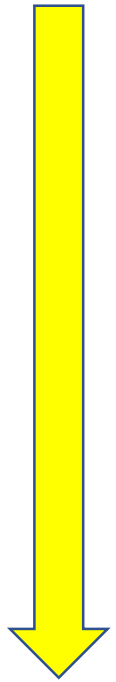
$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$

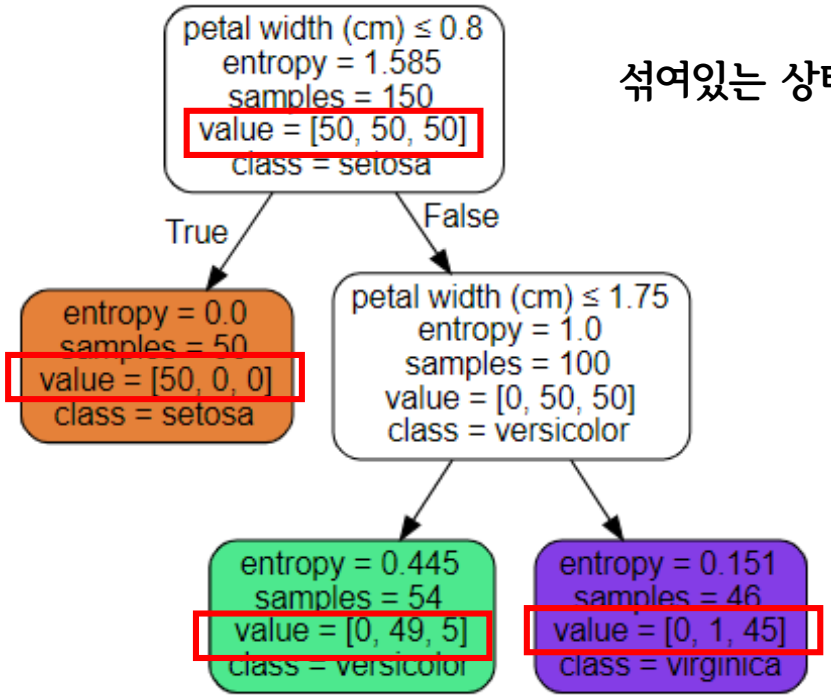
p_i: 한 영역 안에 존재하는 데이터 가운데 범주 i에 속하는 데이터의 비율

High Entropy

엔트로피 낮아짐

Low Entropy





섞여있는 상태

분류된 상태

Information Gain(정보 획득량)

불순도 ↓
= 엔트로피 ↓
= IG ↑

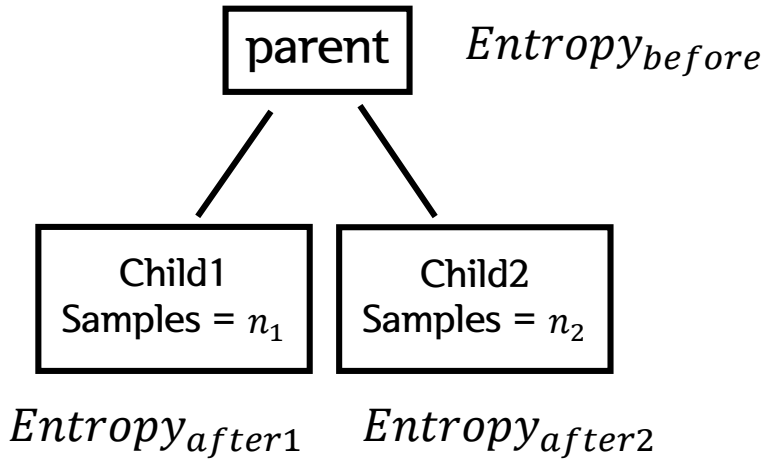
Information Gain(IG) = $Entropy_{before} - Entropy_{after}$
= Entropy(parent) - [weighted average]Entropy(children)

- 분기 이전과 분기 이후의 엔트로피의 차이

- Information Gain을 최대화하는 방향으로 학습이 진행됨.
- 가능한 모든 변수와 기준에서 IG를 계산해 가장 큰 IG 값을 보이는 변수와 기준을 선택함.

- 분기된 가지에서의 entropy 요약($Entropy_{after}$ 구하기)

$Entropy_{after}$ = [weighted average]Entropy(children)
$= \frac{n_1}{n_1+n_2} Entropy_{after1} + \frac{n_2}{n_1+n_2} Entropy_{after2}$



Classification tree

예) 독립변수가 범주형 변수인 경우

속도 slow인지 fast인지 예측

- node로 사용될 변수 구하기(독립변수가 범주형 변수인 경우)

1. Root node의 엔트로피 구하기
2. 각 독립변수에 대해 트리 분할 후 자식 노드의 엔트로피 계산
3. 각 독립변수에 대한 IG 계산 후 IG가 최대가 되는 분기 조건 찾기
4. 모든 leaf node의 엔트로피가 0이 될 때까지 2, 3을 반복 수행

이렇게 되면 leaf node는 반드시
한 가지 범주의 데이터만을 가지게 됨.
이렇게 만든 트리를 full tree라고 함.
오버피팅 문제 생길 수 있음. Pruning 필요

경사	표면	속도제한	속도
Steep	Bumpy	Yes	Slow
Steep	Smooth	Yes	Slow
Flat	Bumpy	No	Fast
steep	smooth	no	fast

$p\_slow = 0.5$
$p\_fast = 0.5$

$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$

1 현재의 엔트로피 = 1 ←정확히 반반

(식으로 계산)

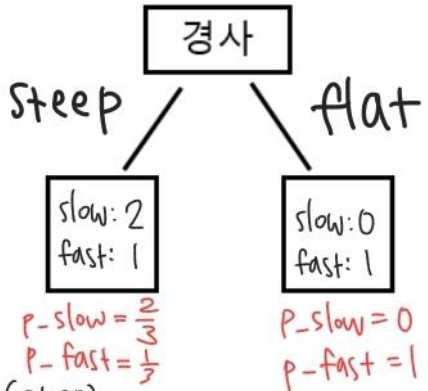
$\text{Entropy} = -0.5 * \log_2(0.5) - 0.5 * \log_2(0.5) = 1$

Classification tree

경사	표면	속도제한	속도
Steep	Bumpy	Yes	Slow
Steep	Smooth	Yes	Slow
Flat	Bumpy	No	Fast
steep	smooth	no	fast

- 가능한 <u>모든 변수와 기준</u>에서 IG를 계산해 가장 큰 IG 값을 보이는 변수와 기준을 선택함.

(1) 경사를 기준으로 분기



$P\_slow = \frac{2}{3}$, $P\_fast = \frac{1}{3}$

$P\_slow = 0$, $P\_fast = 1$

entropy (steep)

$= -\frac{2}{3}\cdot\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)$

$= 0.9183$

entropy (flat) $= 0$

2 entropy (속도|경사)

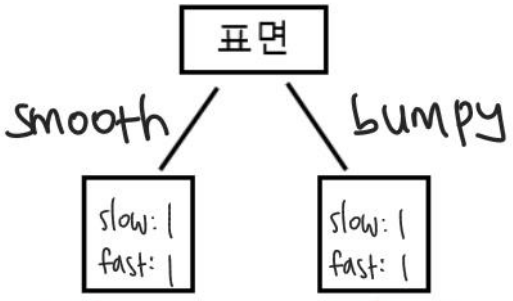
= [weighted average] entropy (children)

$= \frac{3}{4}\cdot 0.9183 + \frac{1}{4}\cdot 0$

$= 0.6888$

3 IG (속도|경사) $= 1 - 0.6888 = 0.3112$

(2) 표면을 기준으로 분기



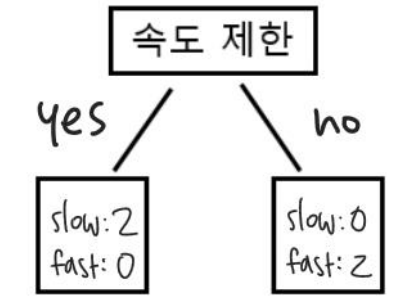
$P\_slow = \frac{1}{2}$, $P\_fast = \frac{1}{2}$

$P\_slow = \frac{1}{2}$, $P\_fast = \frac{1}{2}$

entropy (속도|표면) = 1

IG (속도, 표면) = 0

(3) 속도 제한을 기준으로 분기



$P\_slow = 1$, $P\_fast = 0$

$P\_slow = 0$, $P\_fast = 1$

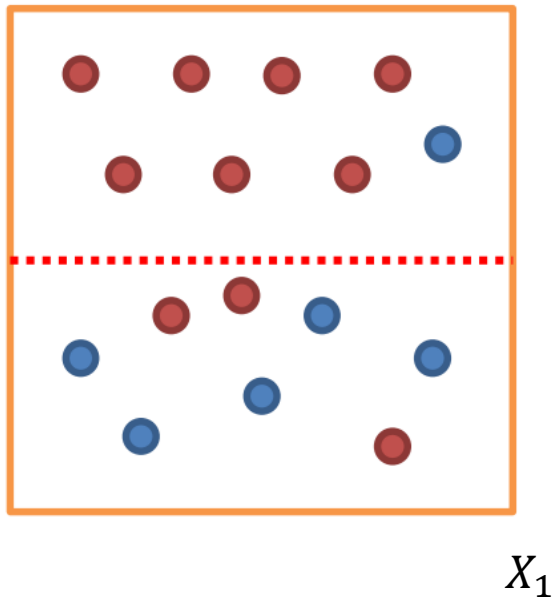
entropy (속도|속도제한) = 0

IG (속도, 속도제한) = 1

IG 가장 큰 방향으로 학습이 진행됨.
첫 node를 속도 제한으로 잡기

Classification tree

예) 독립변수가 연속형 변수인 경우



- 분기 이전 :

$$Entropy(A) = -\frac{10}{16}\log_2\left(\frac{10}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) \approx 0.95$$

- 분기 이후 :

$$Entropy(A) = \sum_{i=1}^{d} R_i \left(-\sum_{k=1}^{m} p_k \log_2(p_k)\right)$$

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

$$\triangle(j, s) = \text{분기이전} - \text{분기이후} = 0.95 - 0.75 = 0.20$$

모든 변수 j와 기준 s의 조합들 중 IG가 가장 큰 조합 선택

변수: d개, 개체: n개 → 1회 분기 위해 계산해야 하는 경우의 수: d(n - 1)

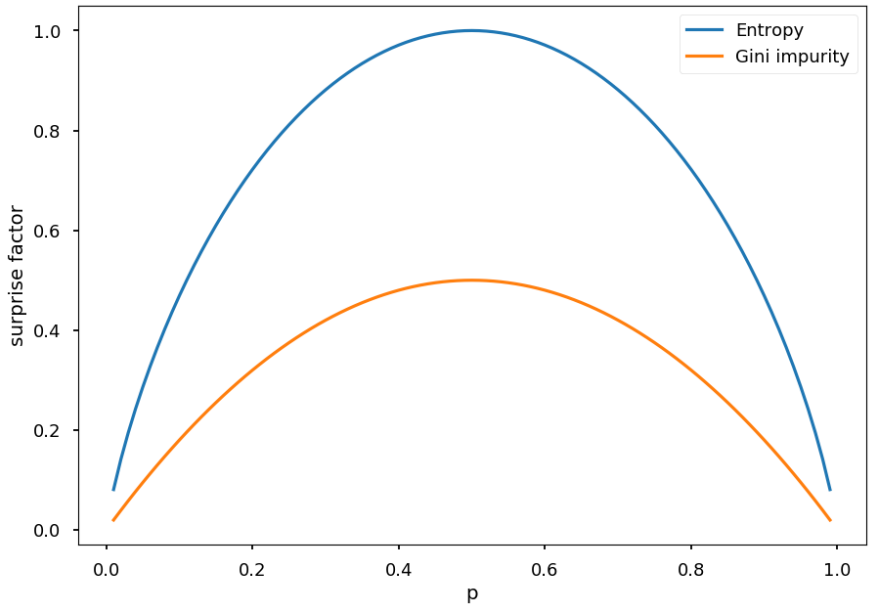
Classification tree

불순도를 수치화한 척도(2)

지니 지수(Gini index)

$$G(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

p_k: 한 영역 안에 존재하는 데이터 가운데 범주 k에 속하는 데이터의 비율



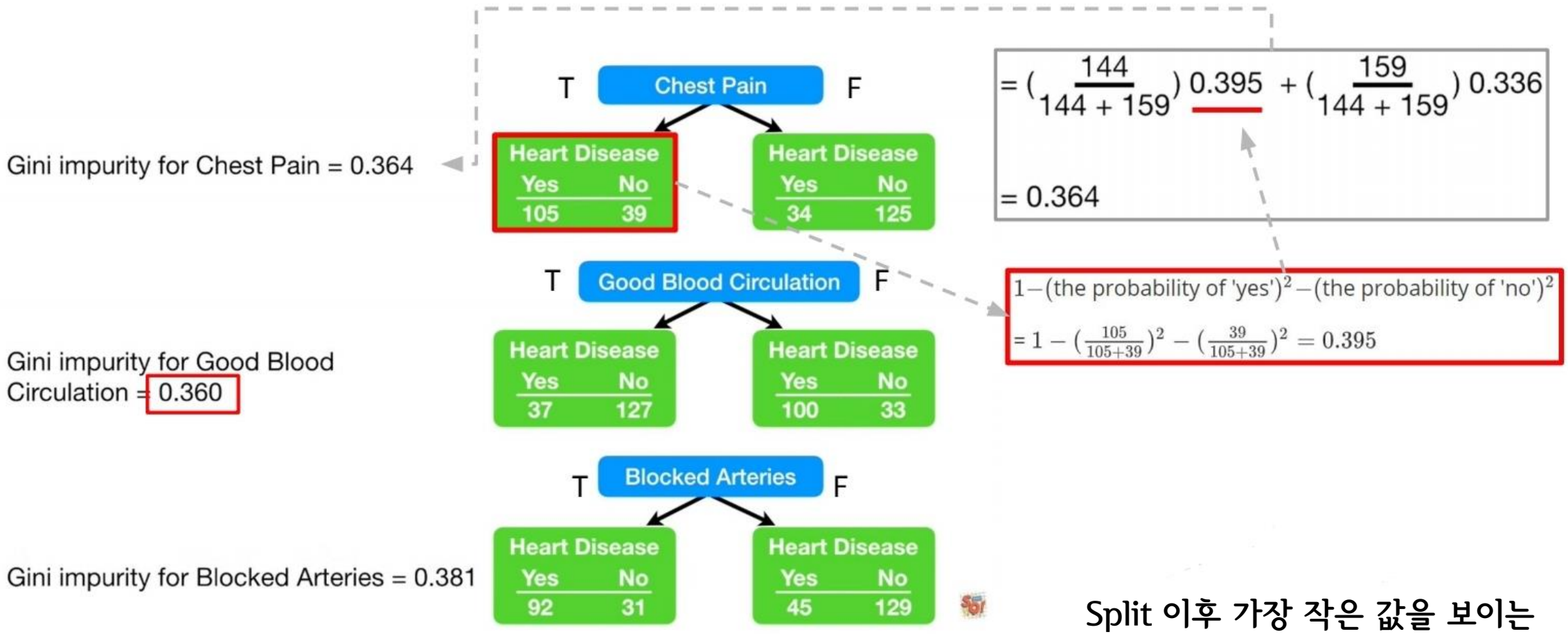
$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

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

$$\approx 0.47$$

지니 지수(Gini index) 예시

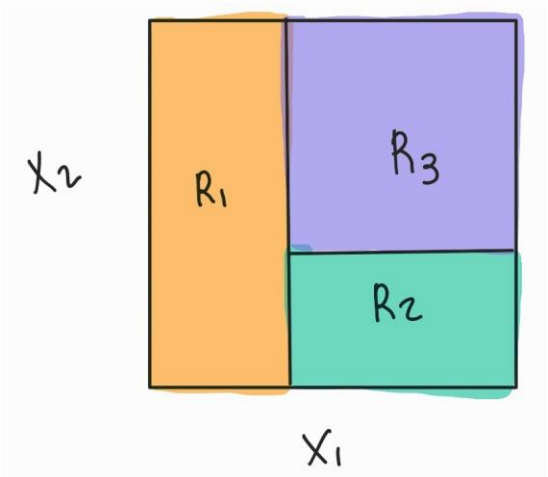
출처: statQuest

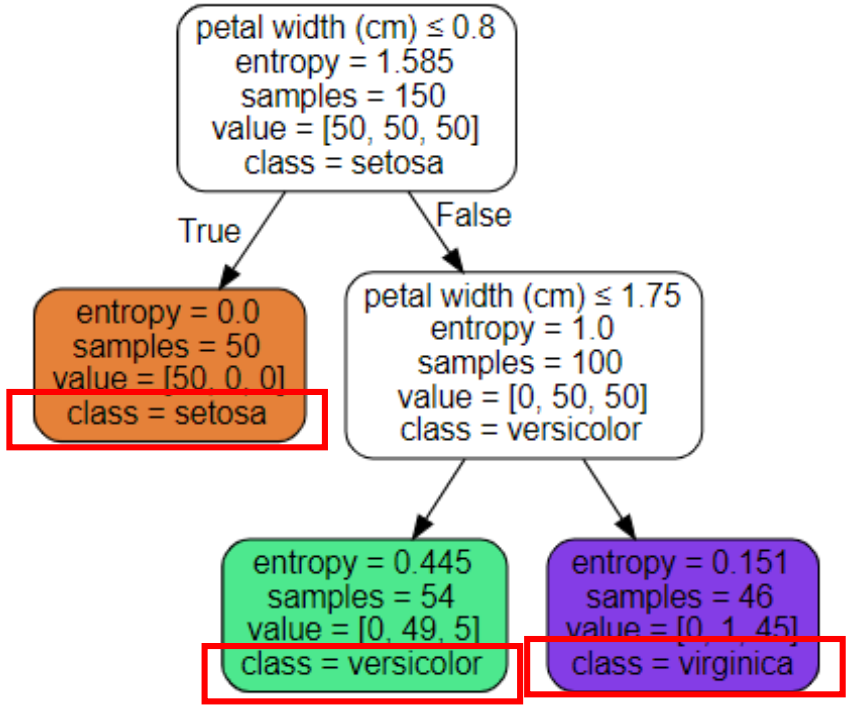


Split 이후 가장 작은 값을 보이는 변수를 선택

각 영역에서 예측된 Y의 범주

- 결정된 R_m에 대하여,
  - $\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$
  - 추정된 Y의 범주: $k(m) = \underset{k=1,\ldots,K}{\operatorname{argmax}}\, \hat{p}_{mk}$

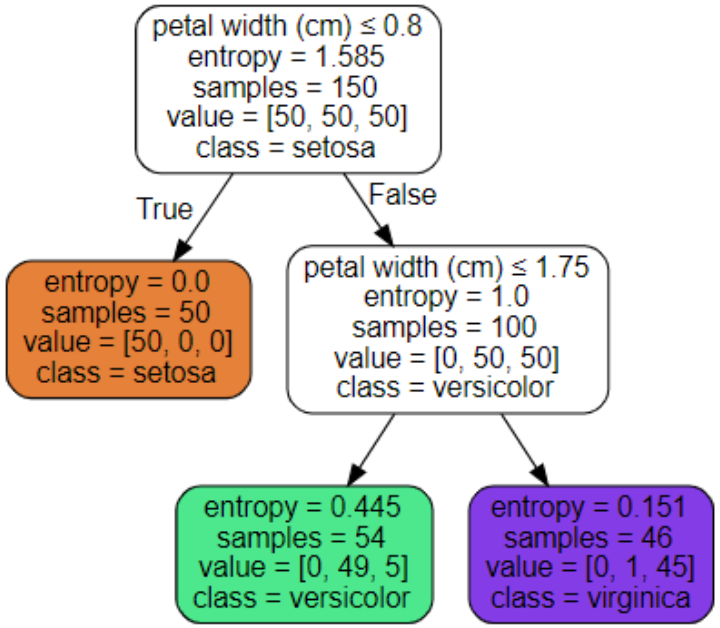




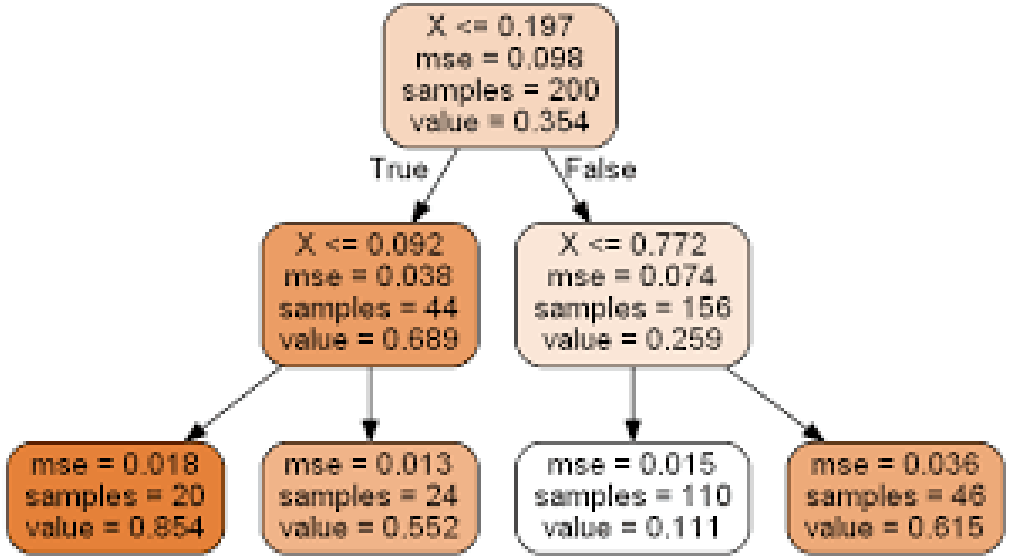
regression tree

Regression tree

영역을 나누는 것은 classification tree와 같으나



Class 예측하기



value 예측하기

Tree 조건에 따라 독립변수의 공간(X가 가질 수 있는 영역)을 block으로 나누는 개념
-> 나누어진 영역 안에 속하는 샘플의 특성을 통하여 Y를 예측

Regression tree

Regression tree의 영역을 나누기 위한 tree의 분기 조건 정하기

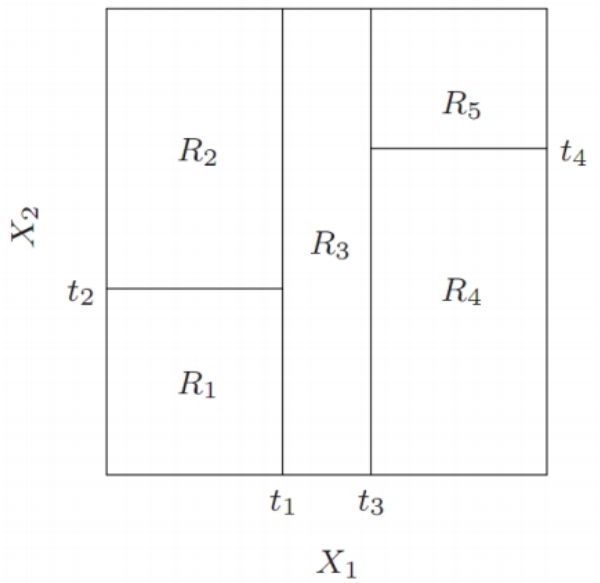
- Classification tree에서는 불순도(엔트로피, 지니 지수)를 낮추는 방향으로 분기 조건 (독립변수와 기준)을 결정했었음.
- X에는 범주형 변수와 연속형 변수가 모두 올 수 있음.
- Regression tree에서는 다음 measure을 가장 좋은 값으로 만드는 변수와 기준을 선택함.

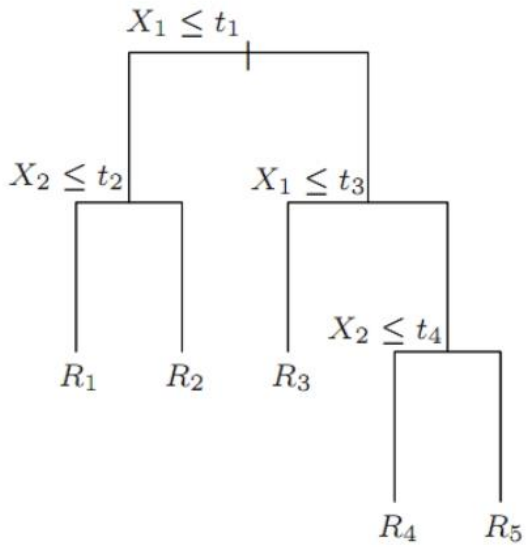
$$\min_{j,\,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

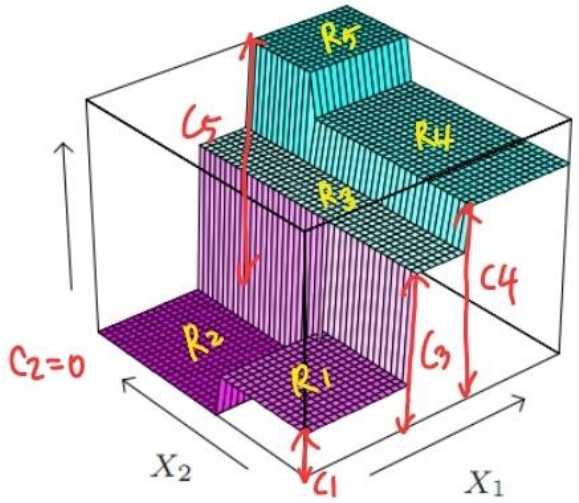
$$\hat{c}_1=\text{ave}(y_i|x_i\in R_1(j,s))\quad\text{and}\quad\hat{c}_2=\text{ave}(y_i|x_i\in R_2(j,s))$$

- 한 영역에 데이터 2개가 남을 때까지 node의 변수와 기준 선택 반복
  (한 영역에 n개 이하의 데이터가 있다면 tree를 grow하는 것을 멈추는 것도 하나의 파라미터: min_samples_split)

Regression tree







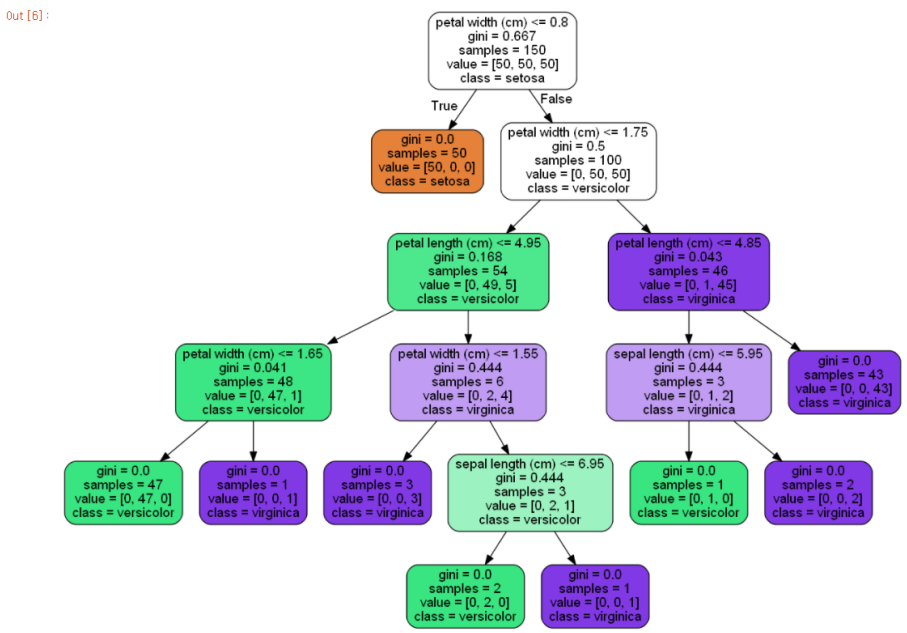
- Y의 예측값
  - 결정된 R_m에 대하여,
    - $\hat{c}_m = avg(y_i | x_i \in R_m)$
    - $\hat{y} = \hat{f}(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m)$

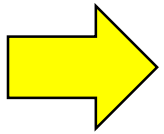
leaf node의 y의 평균을 예측값으로 반환
→ 예측값의 종류는 leaf node의 개수와 일치

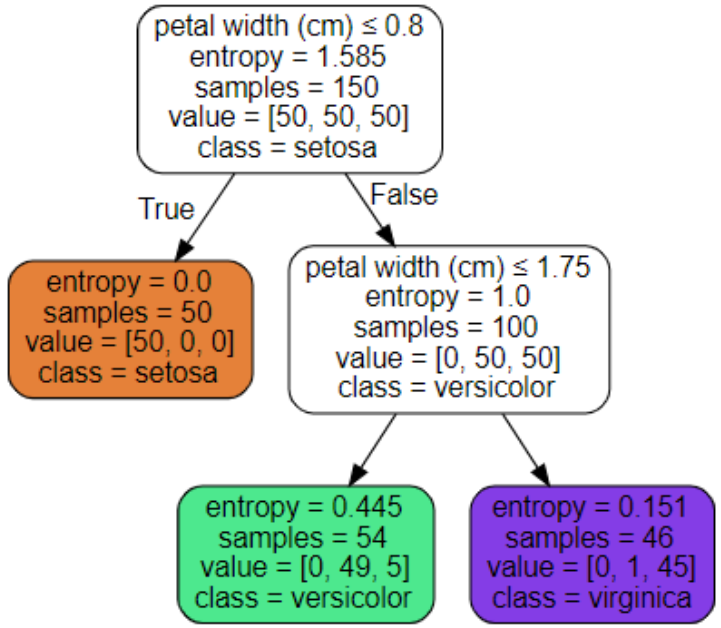
Pruning

Pruning(가지치기)

- Full tree는 영역이 너무 자세하게 구분되어 오버피팅 문제가 생길 수 있음.
- Full tree로 형성된 결정 트리의 특정 노드 밑의 하부 트리를 제거
- 마치 나뭇가지를 잘라내는 것과 같아서 이런 이름이 붙음.
- 일반화 성능 좋아짐.







사전 가지치기

트리의 최대 depth, 각 노드에 있어야 할 최소 관측값 수 등을 미리 지정하여 트리를 만드는 도중에 알고리즘을 멈추는 것.

```
max_depth: 트리의 최대 depth를 결정
min_samples_split: split하기 위해 노드가 가지고 있어야 하는 최소 샘플 개수
min_samples_leaf: leaf node가 가져야 하는 최소 샘플 개수
max_leaf_nodes: leaf node 최대 개수
```

사후 가지치기(Cost complexity pruning)

- 트리를 full tree로 만든 후 terminal node를 결합

Cost function: $R_\alpha(T) = R(T) + \alpha|T|$

Decision tree는 cost func을 최소로하는 분기를 찾아내도록 학습됨.

- $R_\alpha(T)$: decision tree의 비용 복잡도(cost-complexity measure)
- $R(T)$: 오분류율(inpurity)
- α : complexity parameter / 사용자에 의해 부여되는 가중치
- $|T|$: leaf node의 수 (구조의 복잡도)

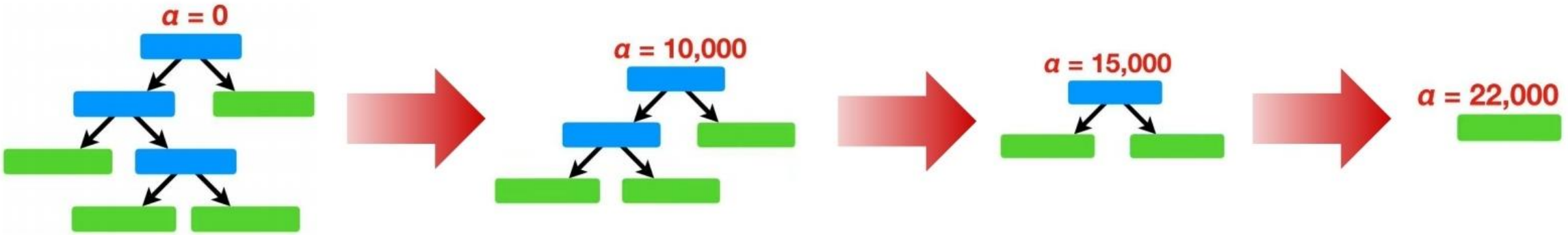
Rregression tree일 때

$$R(T) = \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \bar{y}_{R_m})^2$$

$|T|$ 가 커지면 $R(T)$ 무조건 작아짐. -> 뒤에 penalty term을 붙임.
α 크기가 커지면 $R(T)$ 줄어드는 속도보다 뒤의 term 커지는 속도가 더 커짐.
=> α가 클수록 트리 사이즈 작아짐.

사후 가지치기(Cost complexity pruning)

- α가 클수록 트리 사이즈 작아짐.
α를 0부터 시작해서 늘려간다고 생각하면, 어떤 α의 임계치(effective alpha)를 지나면 가지치기됨.



Effective alpha의 후보군들을 모음.
-> effective alpha 값에 대응하는 subtree 구함.
-> subtree 들에 대해 test accuracy 구하기
-> 가장 좋은 성능 보이는 effective alpha 선택.

Alpha가 커지면서 node가 잘림. Effective alpha가 node와도 대응한다고 생각할 수 있음.

Effective alpha가 작은 node부터 잘려나감.

Decision tree 장단점

Decision tree 장단점

장점

1. 해석력이 높음.
2. 범용성
3. 독립변수와 종속변수를 선형적인 관계로 설명하기 힘든 경우 사용 가능

단점

1. 과적합 위험성
2. 약간의 차이에 따라 트리의 모양이 많이 달라질 수 있음. -> randomforest