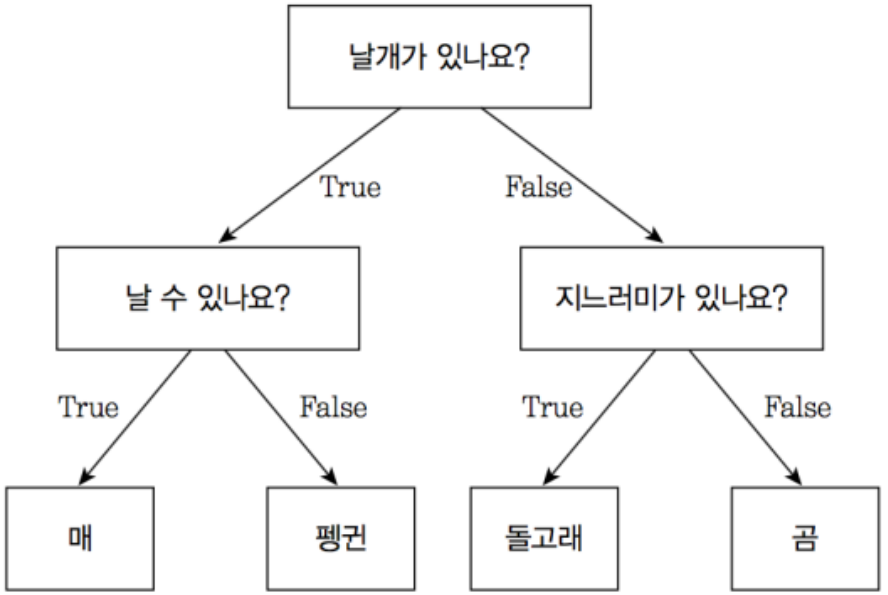
python™

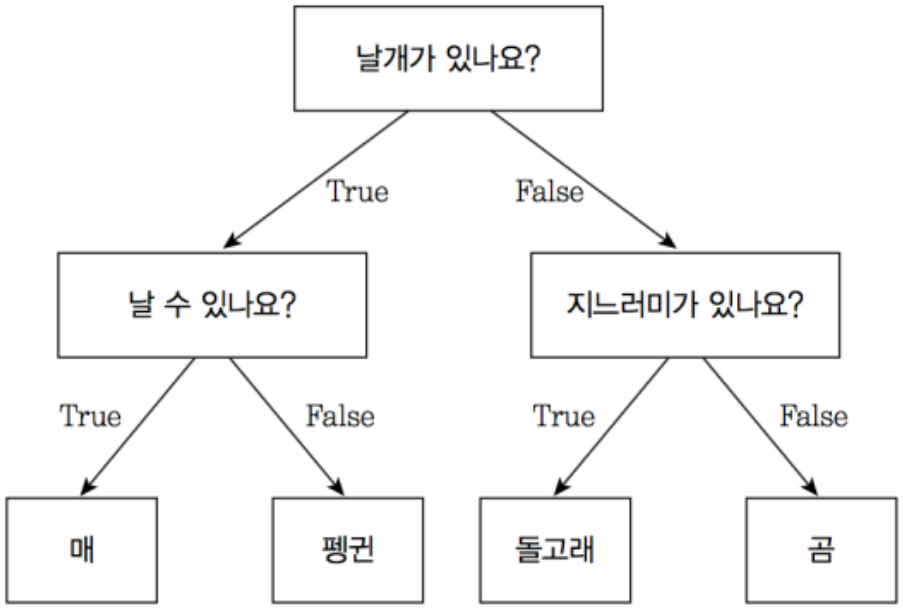
Decision Tree



- 스무고개를 하 듯 예/아니오의 질문을 이어나가면서 학습하는 지도학습 알고리즘
- regression과 classification 모두 가능하며 의외로 regression에서도 좋은 성능을 보임
- 이 과정에서 가장 중요한 것은 '어떤 질문을 해야하냐' 이다.

python™

Decision Tree



- 옆의 결정나무의 각각의 box를 'node'라고 부르며,
  맨 위의 node를 root node,
  중간에 질문하는 node를 decision node,
  마지막에 있는 node를 leaf node라고 한다.
  또한, 바로 직전의 node를 parent node,
  바로 직후의 node를 children node라고 부른다.

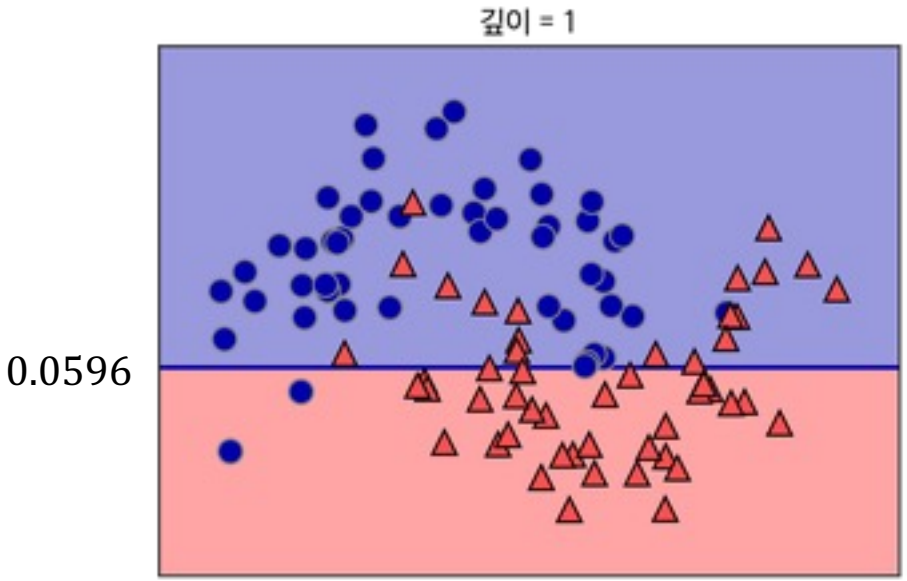
- node를 잇는 각각의 화살표는 branches라고 부르며,
  root node와 leaf node의 깊이를 depth라고 부른다.

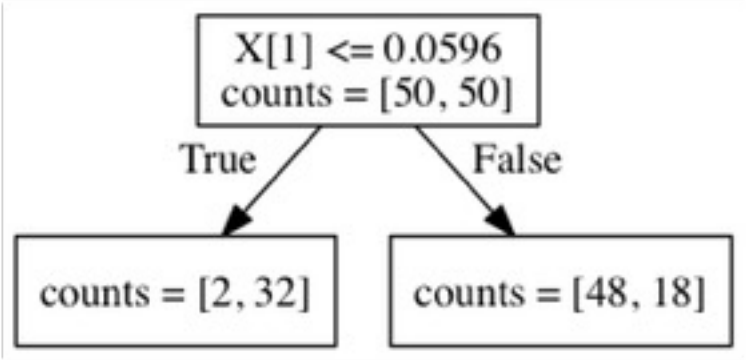
- depth가 너무 깊어지면 overfitting의 문제가 발생,
  적절한 depth를 찾는 것이 중요하다.

python™

Decision Tree

- 예시) depth=1

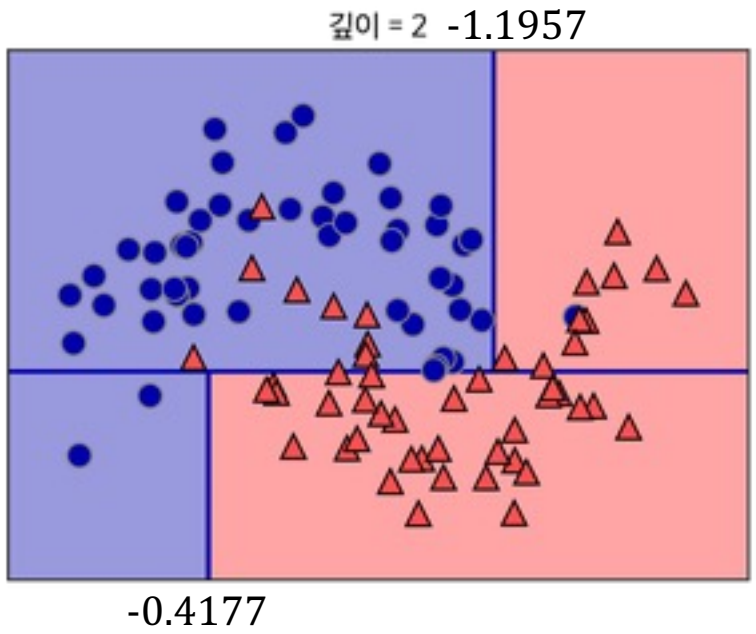


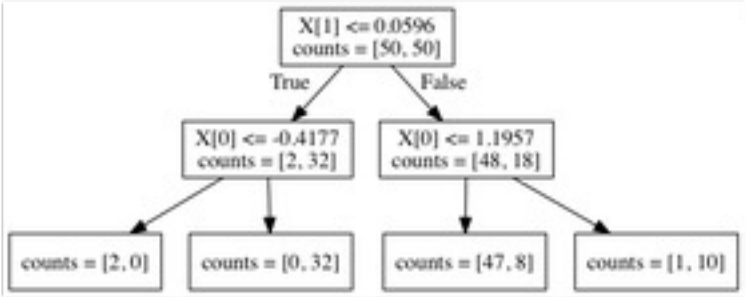


python™

Decision Tree

- 예시) depth=2

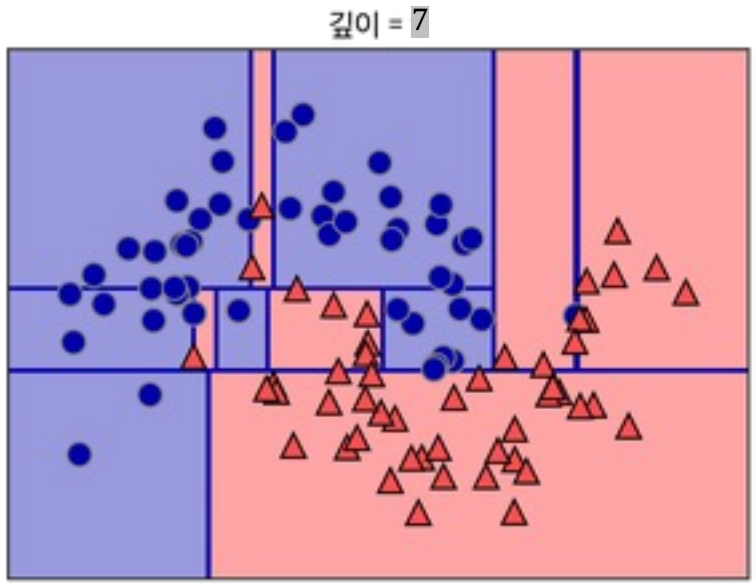


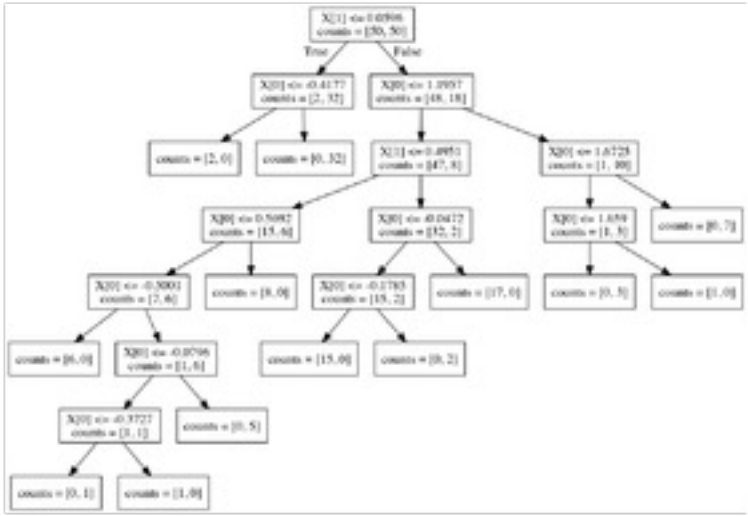


python™

Decision Tree

- 예시) depth=7





가지치기 (Pruning)

- Overfitting을 방지하기 위한 방법 중 하나
  말 그대로 **가지를 치는 작업**으로, 트리의 최대 depth나 node의 최대 개수,
  혹은 한 node가 분할하는 최소 데이터 수를 제한하는 것

- ex)
  min_sample_split = 10
  한 노드에 데이터가 10개가 된다면, 그 노드는 더이상 분할하지 않음

  max_depth = 4
  depth가 4가 되면 노드를 더이상 분할하지 않음

불순도 (Impurity)

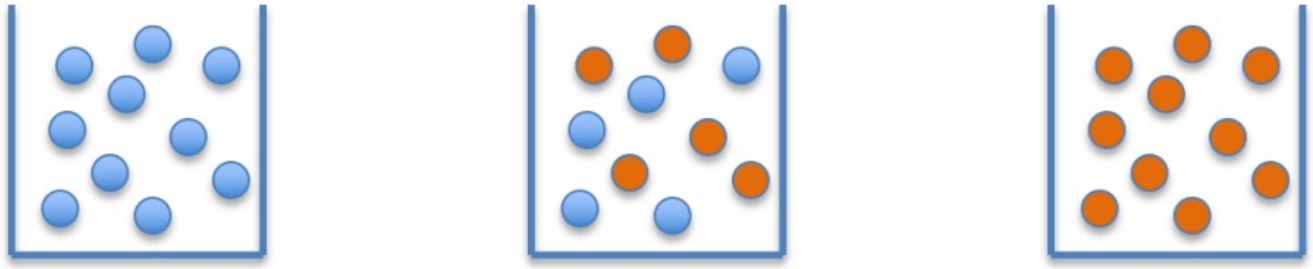
- 어떤 질문을 하는 것이 가장 좋을까?
  leaf node에서 같은 속성의 data들 끼리 분류가 되도록 하는 질문!
  즉, '추가적인 질문을 가장 적게 하는 질문'을 하는 것이 가장 좋다
  이 때, 새롭게 impurity(불순도)라는 개념이 나타난다.

항아리 1. 항아리 2. 항아리 3.

- 다음 그림에서 가장 불순도가 높은/낮은 그림은 어떤 그림인가?

python™

불순도 (Impurity)



- Impurity : 해당 범주 안에 서로 다른 데이터가 얼마나 나타나는지를 표현한 지표, 분류하기 어려운 정도

우리는 leaf node에서 불순도를 가장 낮추는 방향으로 질문을 해야 한다.

해당 지표로는 Entropy, Gini Index, Classification Error 등이 있다.

python™

ID3 알고리즘

- impurity를 엔트로피(entropy)로 계산한 알고리즘
- **엔트로피(entropy)**란? 불순도를 측정하는 지표, 정보량의 기대값

$$Entropy(S) = -\sum_{i=1}^{c} p_i \times \log_2 p_i$$

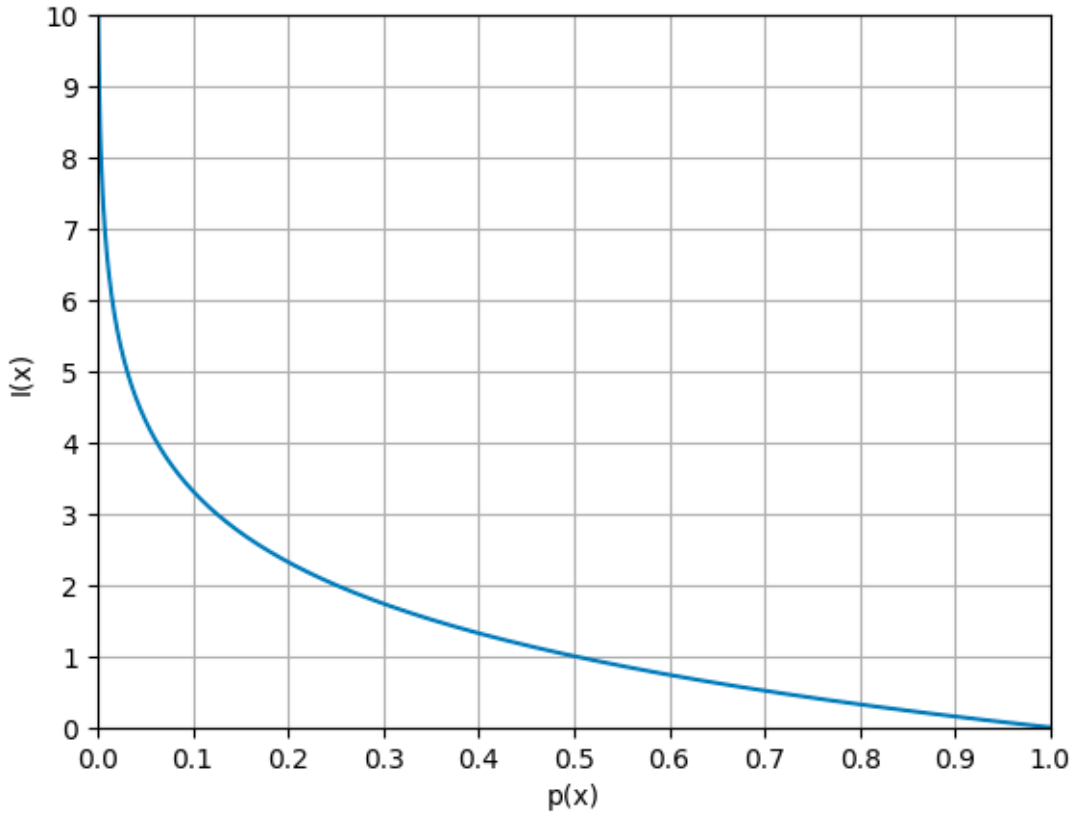
ID3 알고리즘

- 확률과 정보량(놀람의 정도)
  확률 : 사건 X_i가 발생할 확률 $\rightarrow \Pr(X_i)$

  정보량 : 사건 X_i가 가지고 있는 정보량

  $\rightarrow I(x)$

  쉽게 말하면 '놀람의 정도'
  드물게 발생할 사건일 수록 더 많이 놀란다!



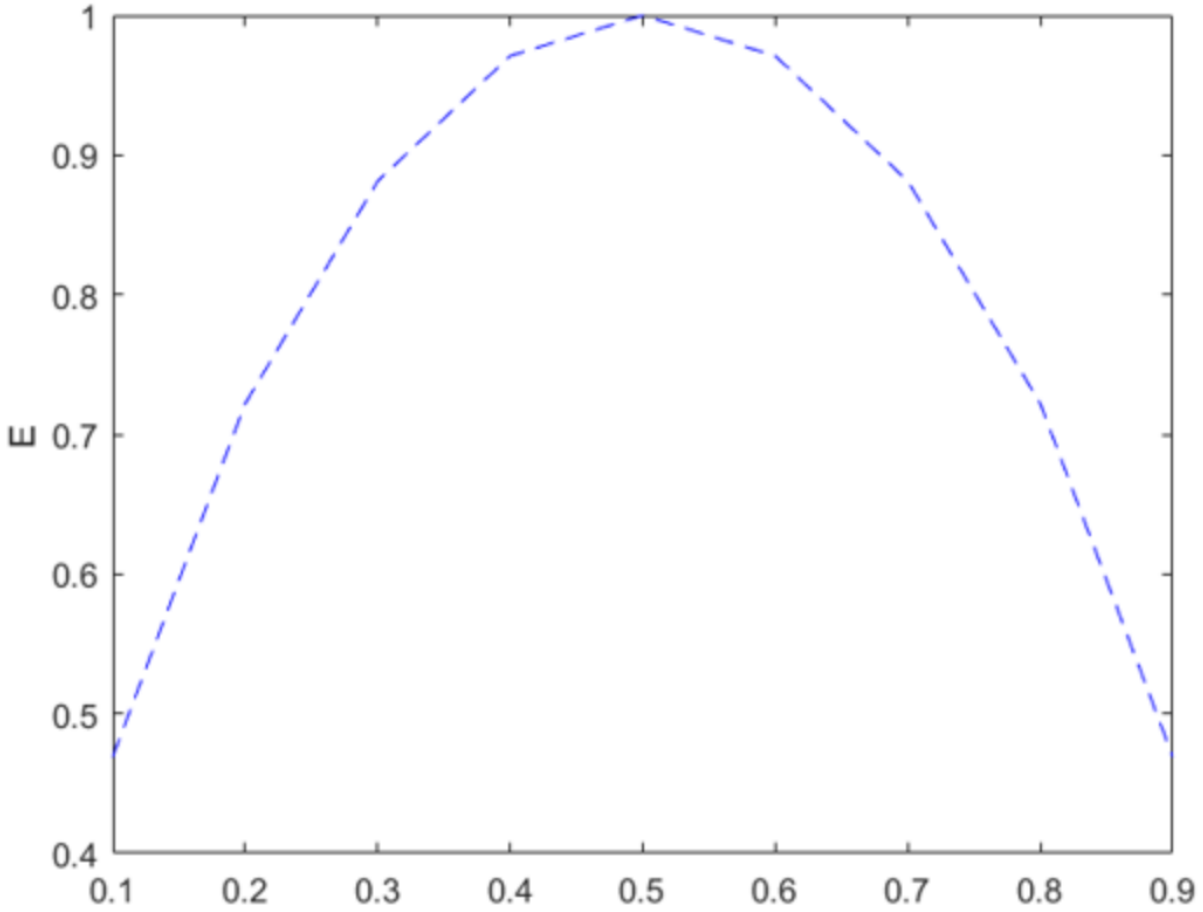
python™

ID3 알고리즘

- 엔트로피 (Entropy)

$$Entropy(S) = \sum_{i=1}^{c} p_i \times \log_2 \frac{1}{p_i}$$

$$= -\sum_{i=1}^{c} p_i \times \log_2 p_i$$



python™

ID3 알고리즘

- Information Gain (정보이득) : 새롭게 얻어낸 정보의 양
  = 이전 정보의 양 - 현재 정보의 양
  = 분할 전 엔트로피 - 분할 후 엔트로피
  = Entropy before split - weighted Entropy after split

$$IG(S,A) = E(S) - E(S|A)$$

A: 속성(Feature) E: 엔트로피

- 정보이득이 크다는 것은, 어떤 속성으로 분할했을 때, 불순도가 크게 줄어든다는 것!
  → 가지고 있는 모든 속성에 대해 분할한 후, 정보이득을 계산.
  → 이후 가장 큰 정보이득이 나오는 것을 질문으로 삼는다!

ID3 알고리즘

날짜	날씨	온도	습도	바람	참가여부
D1	맑음	더움	높음	약함	X
D2	맑음	더움	높음	강함	X
D3	흐림	더움	높음	약함	O
D4	비	포근	높음	약함	O
D5	비	서늘	정상	약함	O
D6	비	서늘	정상	강함	X
D7	흐림	서늘	정상	강함	O
D8	맑음	포근	높음	약함	X
D9	맑음	서늘	정상	약함	O
D10	비	포근	정상	약함	O
D11	맑음	포근	정상	강함	O
D12	흐림	포근	높음	강함	O
D13	흐림	더움	정상	약함	O
D14	비	포근	높음	강함	X

- 분할 전, 참가 여부에 대한 엔트로피 계산

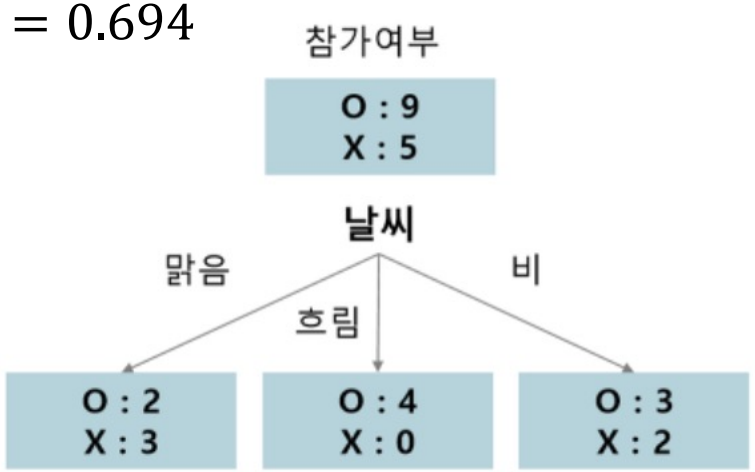
$$E(\text{경기}) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right)$$
$$= 0.940$$

ID3 알고리즘

날짜	날씨	온도	습도	바람	참가여부
D1	맑음	더움	높음	약함	X
D2	맑음	더움	높음	강함	X
D3	흐림	더움	높음	약함	O
D4	비	포근	높음	약함	O
D5	비	서늘	정상	약함	O
D6	비	서늘	정상	강함	X
D7	흐림	서늘	정상	강함	O
D8	맑음	포근	높음	약함	X
D9	맑음	서늘	정상	약함	O
D10	비	포근	정상	약함	O
D11	맑음	포근	정상	강함	O
D12	흐림	포근	높음	강함	O
D13	흐림	더움	정상	약함	O
D14	비	포근	높음	강함	X

- 각 속성에 대해 분할 후 엔트로피 계산
  1. $E(\text{경기}|\text{날씨})$

$$= \frac{5}{14} \times \left\{ -\left(\frac{2}{5}\log_2 \frac{2}{5} + \frac{3}{5}\log_2 \frac{3}{5} \right) \right\}$$
$$+ \frac{4}{14} \times \left\{ -\left(\frac{4}{4}\log_2 \frac{4}{4} + \frac{0}{4}\log_2 \frac{0}{4} \right) \right\}$$
$$+ \frac{5}{14} \times \left\{ -\left(\frac{2}{5}\log_2 \frac{2}{5} + \frac{3}{5}\log_2 \frac{3}{5} \right) \right\}$$
$$= 0.694$$



python™

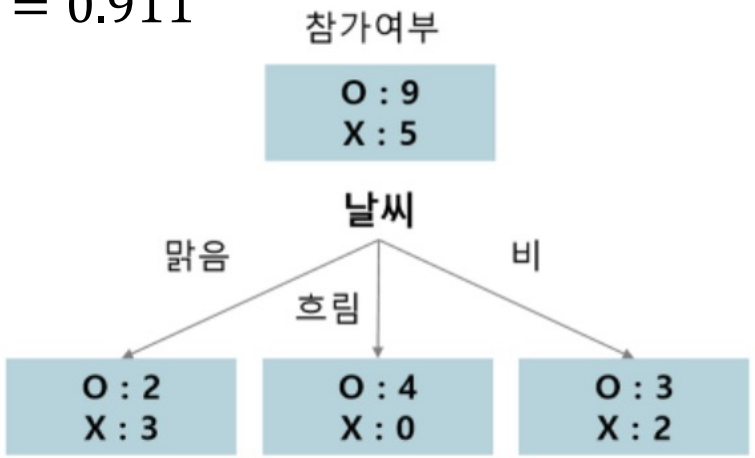
ID3 알고리즘

날짜	날씨	온도	습도	바람	참가여부
D1	맑음	더움	높음	약함	X
D2	맑음	더움	높음	강함	X
D3	흐림	더움	높음	약함	O
D4	비	포근	높음	약함	O
D5	비	서늘	정상	약함	O
D6	비	서늘	정상	강함	X
D7	흐림	서늘	정상	강함	O
D8	맑음	포근	높음	약함	X
D9	맑음	서늘	정상	약함	O
D10	비	포근	정상	약함	O
D11	맑음	포근	정상	강함	O
D12	흐림	포근	높음	강함	O
D13	흐림	더움	정상	약함	O
D14	비	포근	높음	강함	X

- 각 속성에 대해 분할 후 엔트로피 계산

2. $E(\text{경기}|\text{온도})$

$$= \frac{4}{14} \times \left\{ -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4} \right) \right\}$$
$$+ \frac{6}{14} \times \left\{ -\left(\frac{2}{6}\log_2\frac{2}{6} + \frac{4}{6}\log_2\frac{4}{6} \right) \right\}$$
$$+ \frac{4}{14} \times \left\{ -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4} \right) \right\}$$
$$= 0.911$$



python™

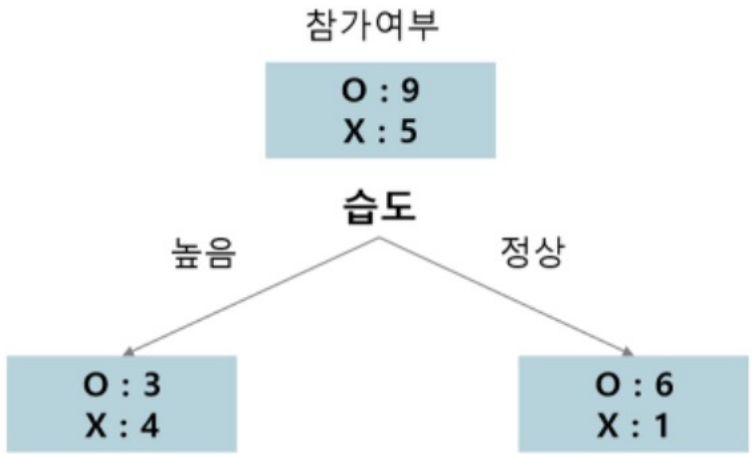
ID3 알고리즘

날짜	날씨	온도	습도	바람	참가여부
D1	맑음	더움	높음	약함	X
D2	맑음	더움	높음	강함	X
D3	흐림	더움	높음	약함	O
D4	비	포근	높음	약함	O
D5	비	서늘	정상	약함	O
D6	비	서늘	정상	강함	X
D7	흐림	서늘	정상	강함	O
D8	맑음	포근	높음	약함	X
D9	맑음	서늘	정상	약함	O
D10	비	포근	정상	약함	O
D11	맑음	포근	정상	강함	O
D12	흐림	포근	높음	강함	O
D13	흐림	더움	정상	약함	O
D14	비	포근	높음	강함	X

- 각 속성에 대해 분할 후 엔트로피 계산

3. $E(\text{경기}|\text{습도})$

$$= \frac{7}{14} - \left(\frac{3}{7}\log_2\frac{3}{7} + \frac{4}{7}\log_2\frac{4}{7}\right) + \frac{7}{14} - \left(\frac{6}{7}\log_2\frac{6}{7} + \frac{1}{7}\log_2\frac{1}{7}\right) = 0.789$$



python™

ID3 알고리즘

날짜	날씨	온도	습도	바람	참가여부
D1	맑음	더움	높음	약함	X
D2	맑음	더움	높음	강함	X
D3	흐림	더움	높음	약함	O
D4	비	포근	높음	약함	O
D5	비	서늘	정상	약함	O
D6	비	서늘	정상	강함	X
D7	흐림	서늘	정상	강함	O
D8	맑음	포근	높음	약함	X
D9	맑음	서늘	정상	약함	O
D10	비	포근	정상	약함	O
D11	맑음	포근	정상	강함	O
D12	흐림	포근	높음	강함	O
D13	흐림	더움	정상	약함	O
D14	비	포근	높음	강함	X

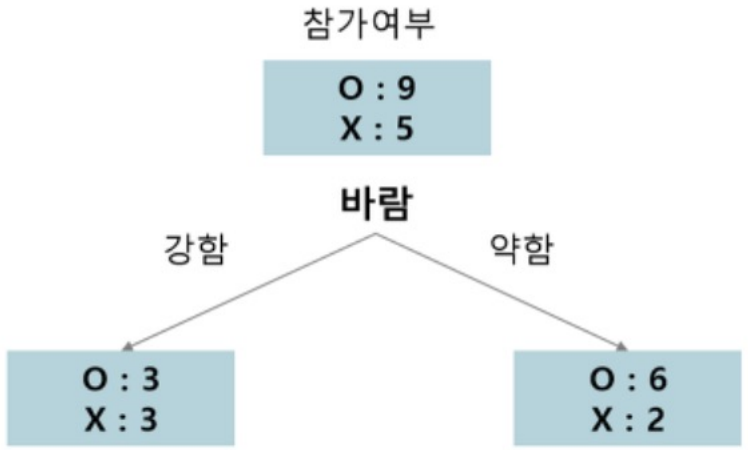
- 각 속성에 대해 분할 후 엔트로피 계산

4. $E(\text{경기}|\text{바람})$

$$= \frac{6}{14} \times \left\{ -\left(\frac{3}{6}\log_2 \frac{3}{6} + \frac{3}{6}\log_2 \frac{3}{6} \right) \right\}$$

$$+ \frac{8}{14} \times \left\{ -\left(\frac{6}{8}\log_2 \frac{6}{8} + \frac{2}{8}\log_2 \frac{2}{8} \right) \right\}$$

$$= 0.892$$



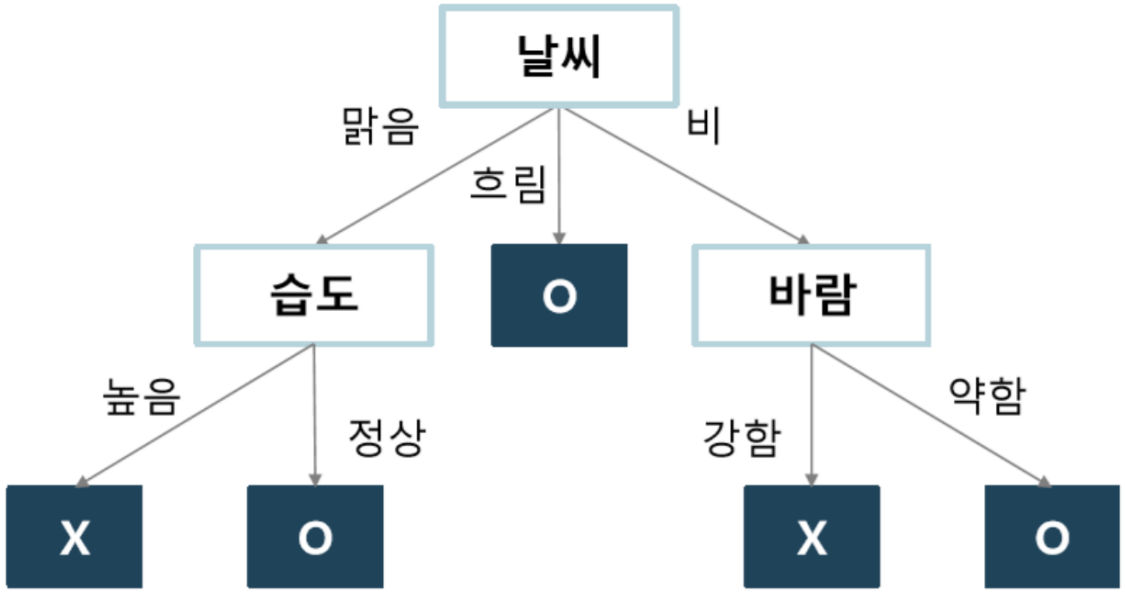
python™

ID3 알고리즘

- 각 속성에 대한 Information Gain 계산

  날씨: 0.940-0.694=0.246 → 날씨로 가장 먼저 분류
  온도: 0.940-0.911=0.029
  습도: 0.940-0.789=0.151
  바람: 0.940-0.892=0.048

  이후 다시 반복반복반복반복…



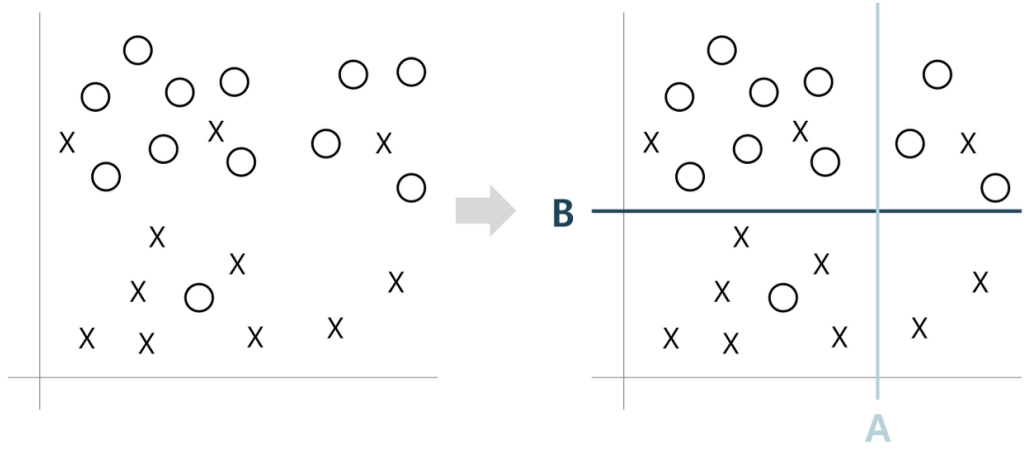
python™

CART 알고리즘

- CART (Classification and Regression Tree)
  impurity를 지니계수(Gini Index)로 계산한 알고리즘

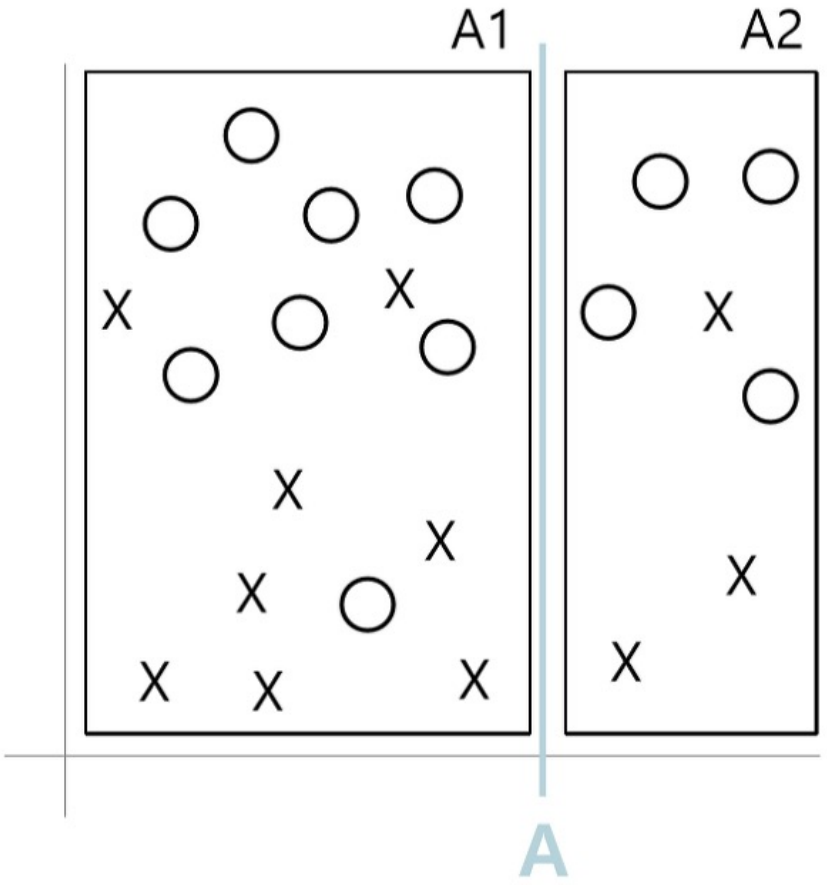
$$G(S) = 1 - \sum_{i=1}^{c} p_i^2$$

- A와 B 중 어떤 경우가 더 잘나눴다고 할 수 있을까?



python™

CART 알고리즘



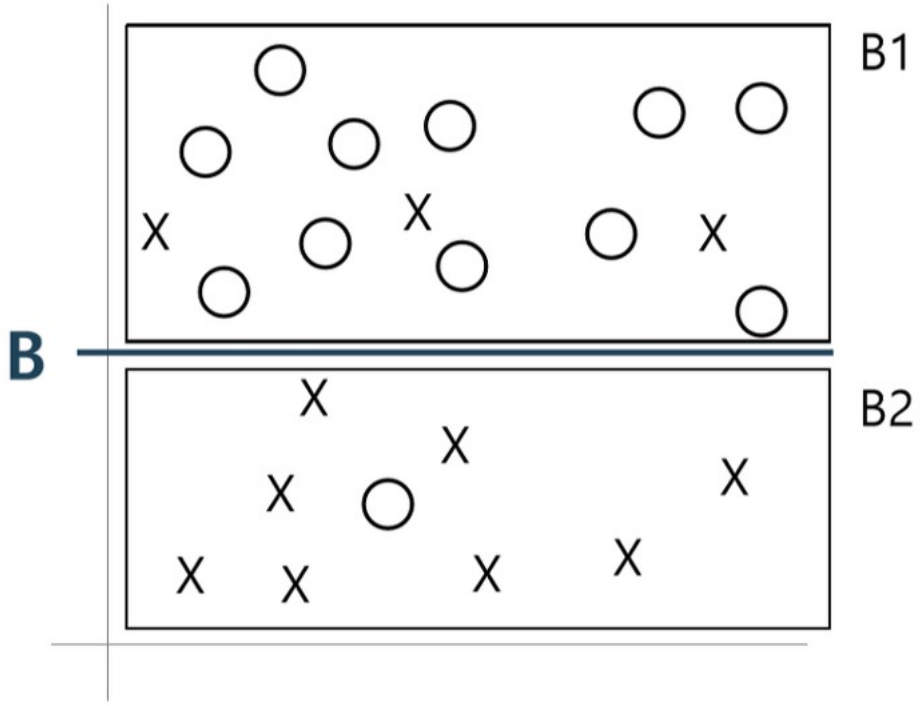
$$G_{A1} = 1 - \left\{ \left(\frac{8}{16}\right)^2 + \left(\frac{8}{16}\right)^2 \right\} = 0.5$$

$$G_{A2} = 1 - \left\{ \left(\frac{4}{7}\right)^2 + \left(\frac{3}{7}\right)^2 \right\} = 0.49$$

$$G_A = \left(\frac{16}{23}\right) \times 0.5 + \left(\frac{7}{23}\right) \times 0.4 = 0.497$$

python™

CART 알고리즘



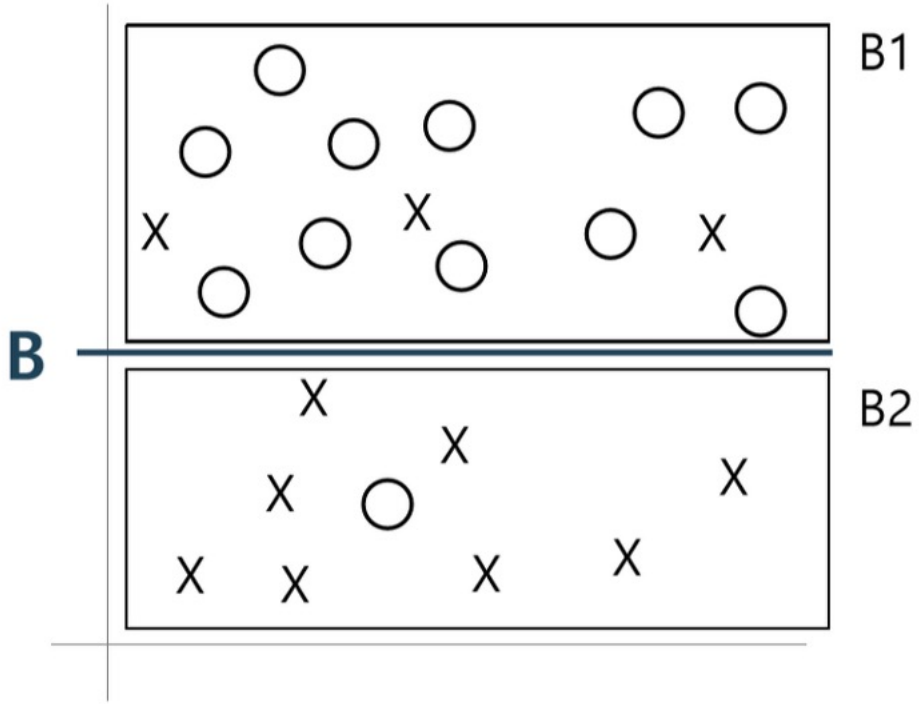
$$G_{B1} = 1 - \left\{ \left(\frac{11}{14}\right)^2 + \left(\frac{3}{14}\right)^2 \right\} = 0.34$$

$$G_{B2} = 1 - \left\{ \left(\frac{1}{9}\right)^2 + \left(\frac{8}{9}\right)^2 \right\} = 0.2$$

$$G_B = \left(\frac{14}{23}\right) \times 0.34 + \left(\frac{9}{23}\right) \times 0.2 = 0.28$$

python™

CART 알고리즘



- 지니계수는 불순도를 의미하기에, 불순도가 더 적은 B로 분할하는 것이 옳다

CART 알고리즘

- Regression은 어떻게??

  Regression Tree에서는 각 leaf에 속한 (관찰값 − 평균)^{2}을 기준으로 질문을 한다

$$\mathrm{MSE} = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - avg\left(y_{R_j} \right) \right)^2$$

  Parent node의 MSE와 Children node의 MSE의 가중합을 비교하여 큰 차이가 없을 때, 결정 나무는 node를 생성하는 일을 그만한다.

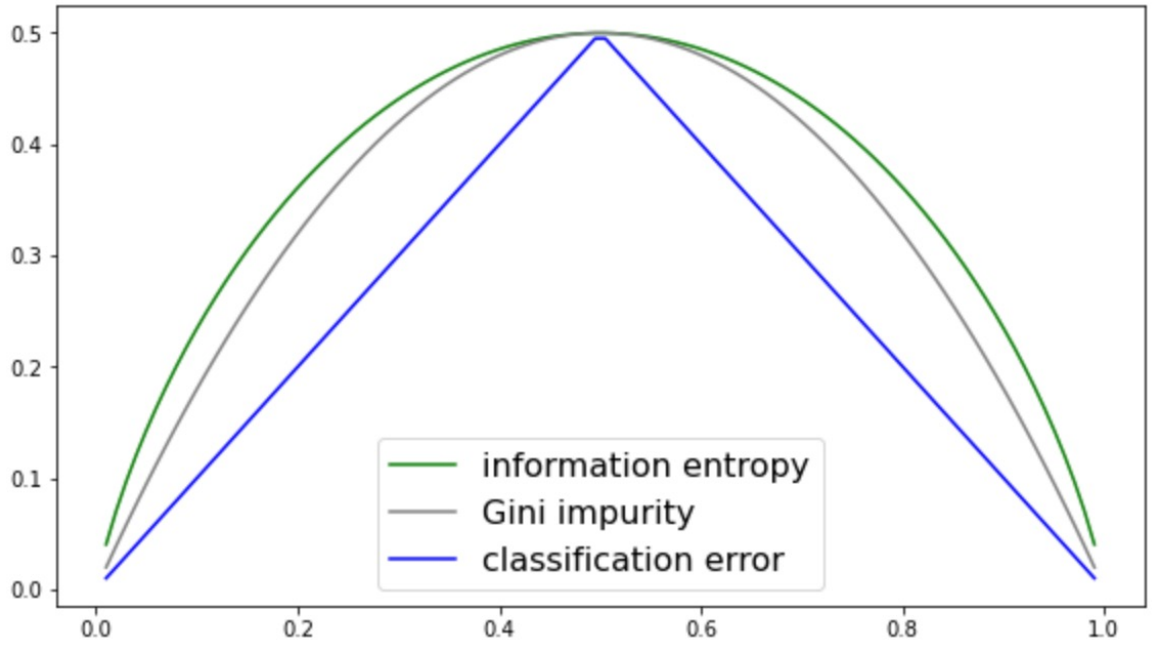
Classification Error

- Most common occurring class에 속하지 못한 class의 비율

$$\text{Classification Error} = 1 - \max_{c} p_c$$

- 가장 단순한 기준이지만 class가 2개일 때는 다른 알고리즘과 차이점 x
- 데이터가 불균형하게 있을 경우 예측 성능이 매우 떨어짐
- 거의 쓰이지 않는다

Impurity 알고리즘 비교



- Entropy : $-(p\log p + (1-p)\log(1-p))$
- Gini Index : $2p(1-p) = 1-(p^2+(1-p)^2)$
- Classification Error : $1-\max(p, 1-p)$

python™

ID3 vs CART

- ID3 알고리즘
  - impurity를 entropy로 계산
  - 직관적이고 쉬운 결과 해석
  - multi-class분류도 가능
  - regression은 지원하지 않음

- CART 알고리즘
  - impurity를 Gini index로 계산
  - 오직 binary 분류만 가능
  - regression을 지원

Decision Tree의 장단점

- DT의 장점
  - 시각화하기 편하고, 직관적으로 이해하기 쉽다
  - 단위와 계량 데이터에 대해서 따로 scaling을 해줄 필요가 없다

- DT의 단점
  - overfitting이 발생할 여지가 매우 크다.
    (파라미터 튜닝을 하지 않으면, 끝없이 나무는 커진다.
    그렇다고 파라미터 튜닝을 하자니, 데이터에 유연하게 대처하기가 어렵다.)

  → 앙상블(Ensemble) 기법을 통해 단점을 극복!

python™