Chapter
6

지도학습 – 의사결정나무

1. 의사결정나무란?
2. **반응 변수 적용 분리 기준**
3. 의사결정나무 학습 과정
4. 코드 실습

1

의사결정나무란?

의사결정나무(Decision Tree)란?

정의

지도 학습의 일종으로, 의사결정규칙을 나무구조로 도표화하여
분류와 예측을 수행하는 기법

목적

의사결정나무 분석을 이해하고 의사결정나무 분석을 실습해보자

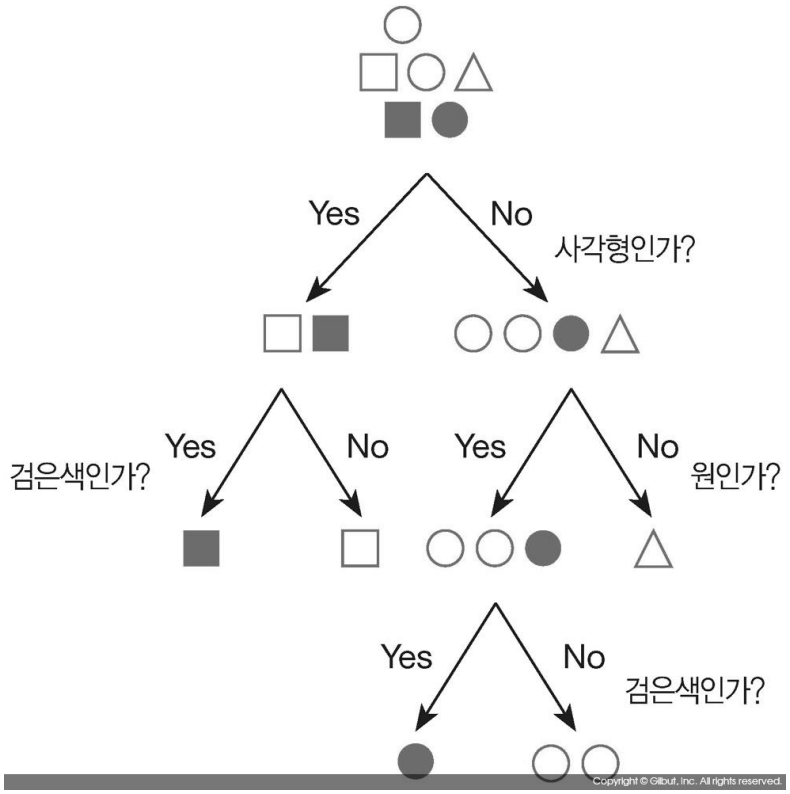
Goal!

1. 의사결정나무란 무엇인가?
2. 의사결정나무는 어떻게 활용될 수 있는가?
3. 의사결정나무에서 반응 변수에 적용되는 분리기준에는 무엇이 있는가?
4. 의사결정나무의 학습과정
5. 예제 R 코드를 통해 의사결정나무 분석을 실습해보자

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

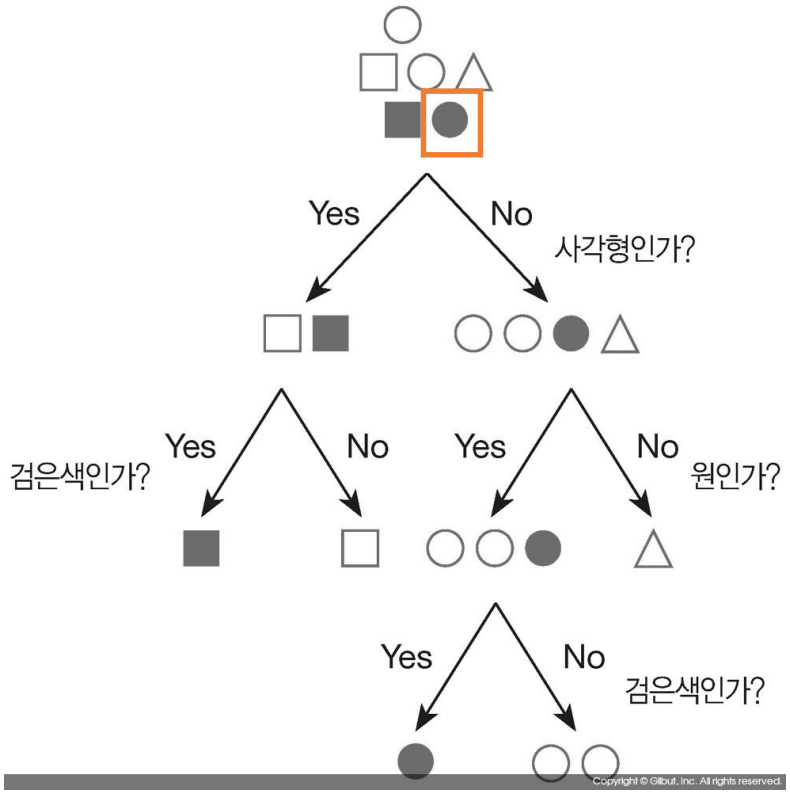
의사결정나무(Decision Tree)란?

예시: 다양한 도형을 분류하는 의사 결정 나무



Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

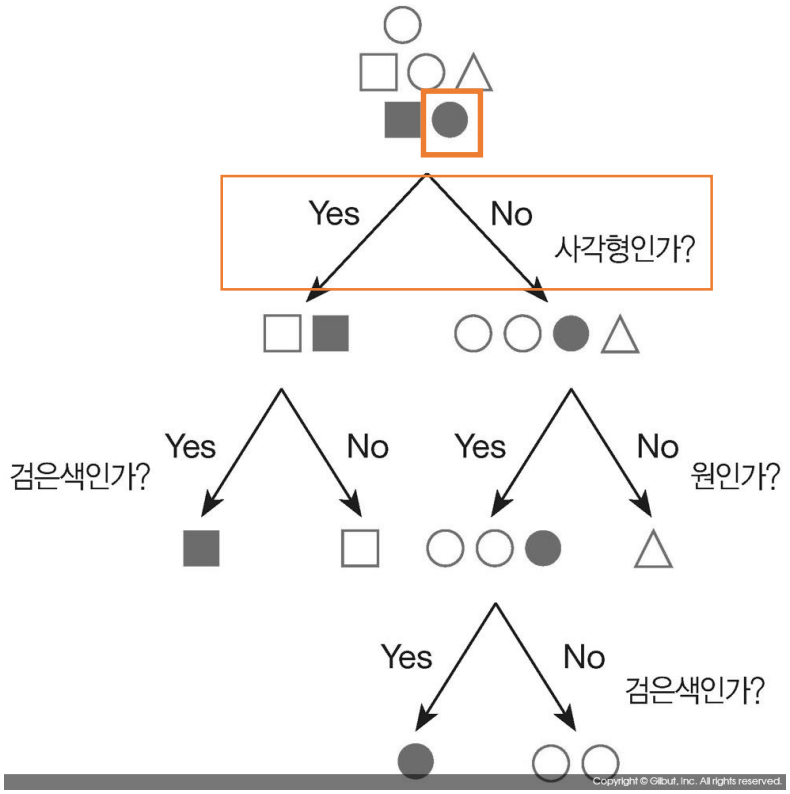


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

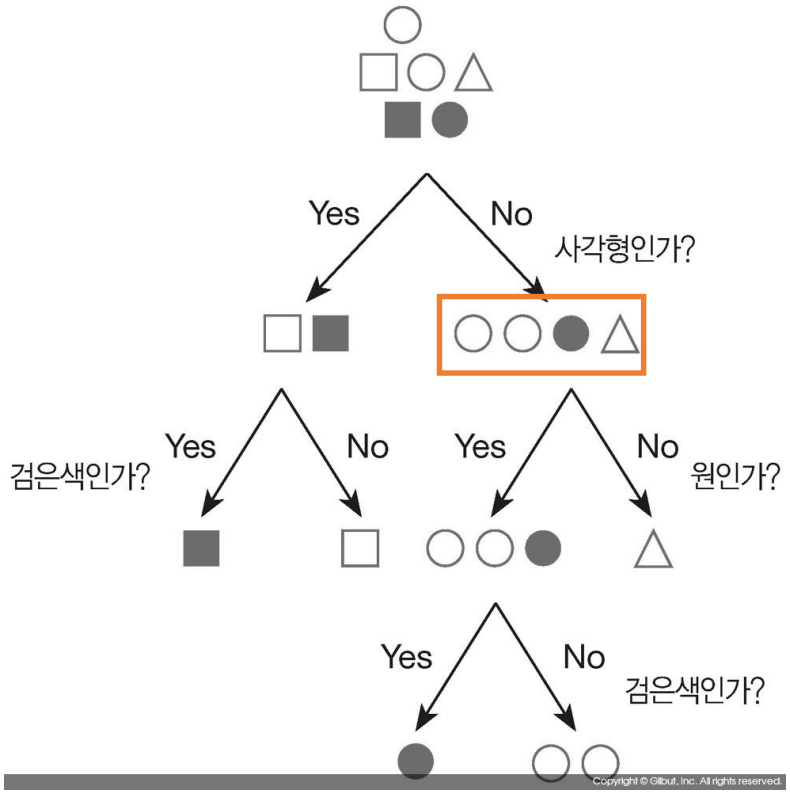


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가?

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

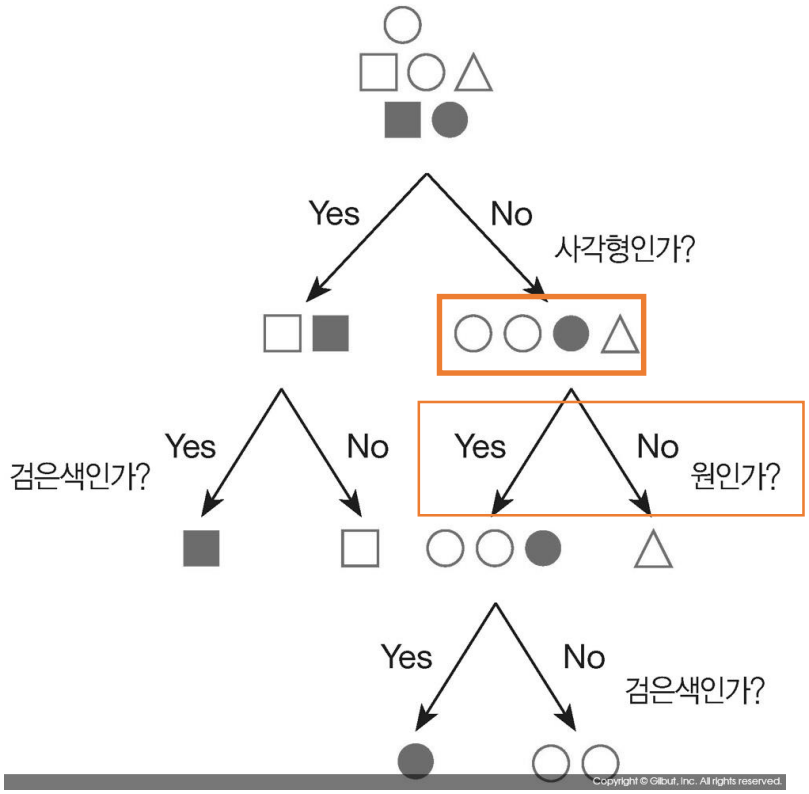


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

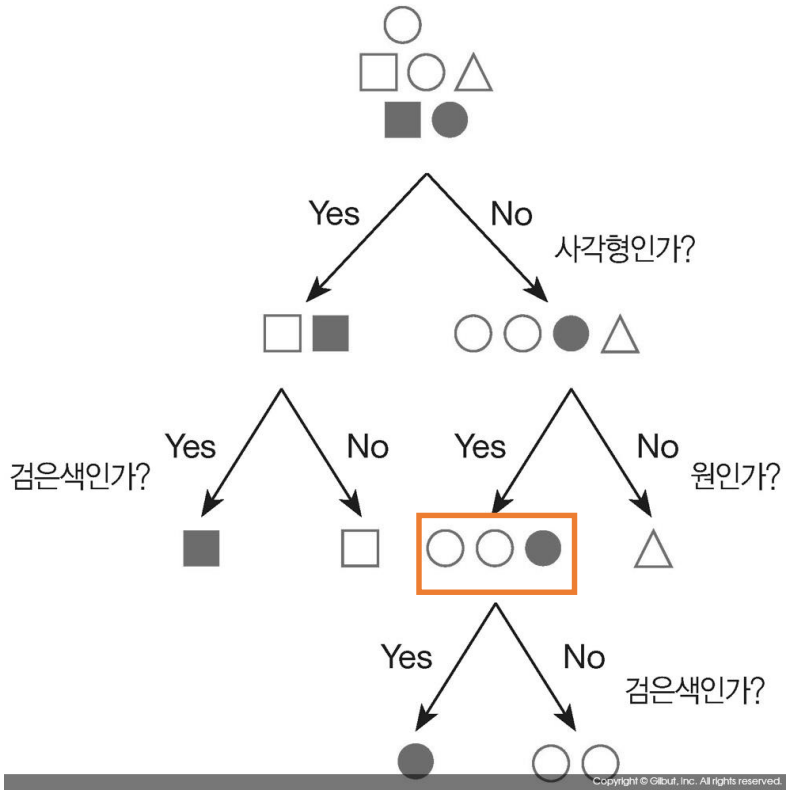


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가?

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

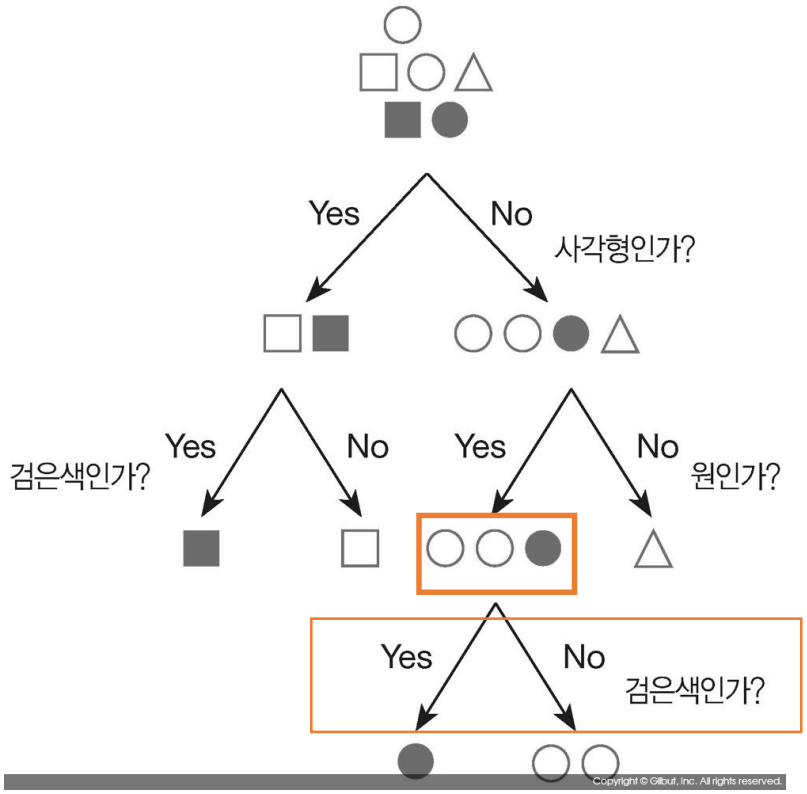


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

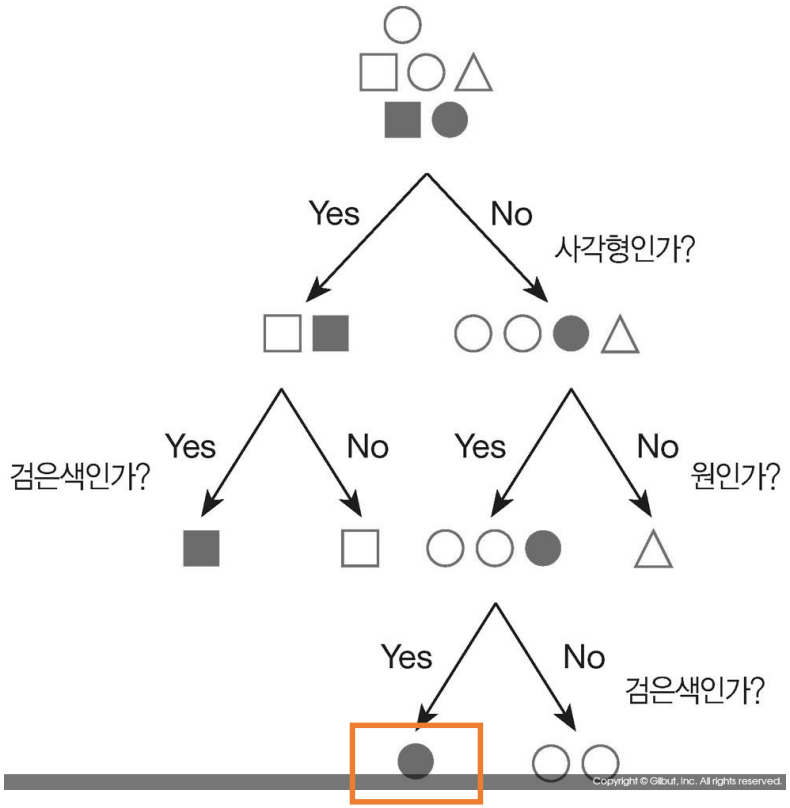


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가?

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

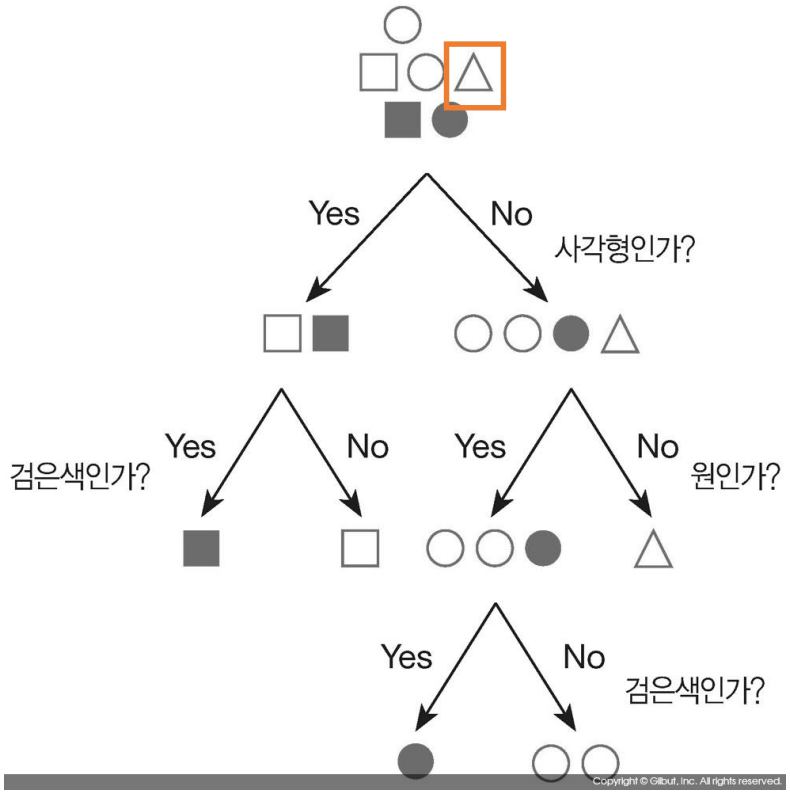


예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?



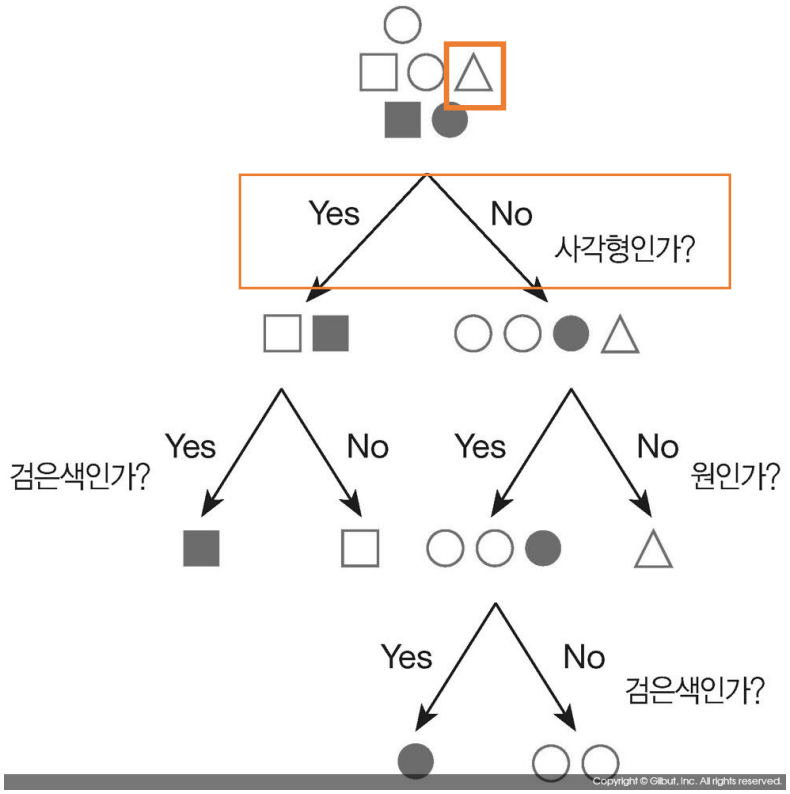
예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

흰색 삼각형이 분류되는 과정

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?



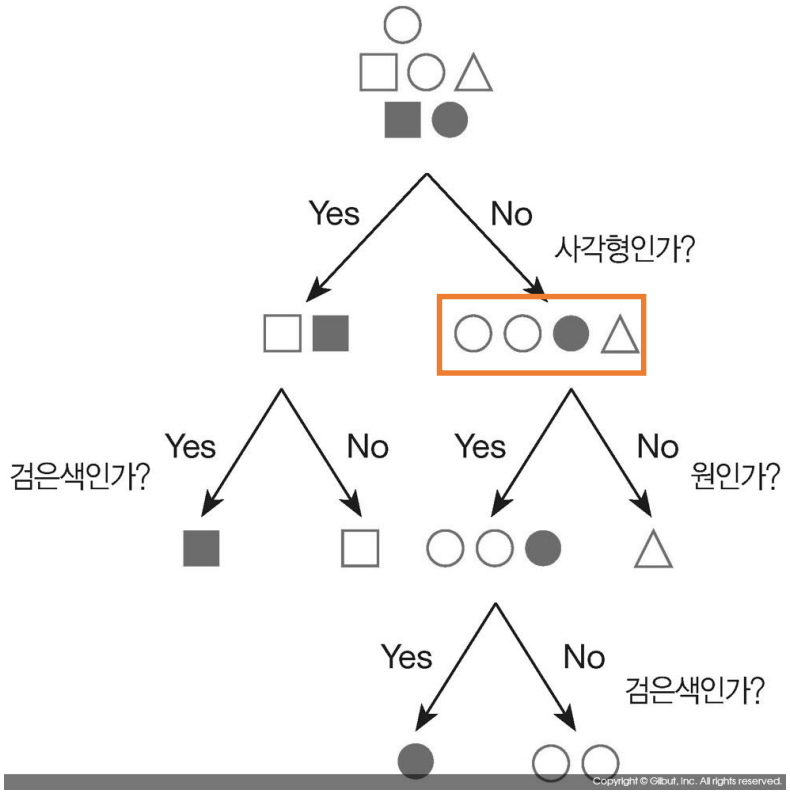
예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

흰색 삼각형이 분류되는 과정
1. 사각형인가?

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?



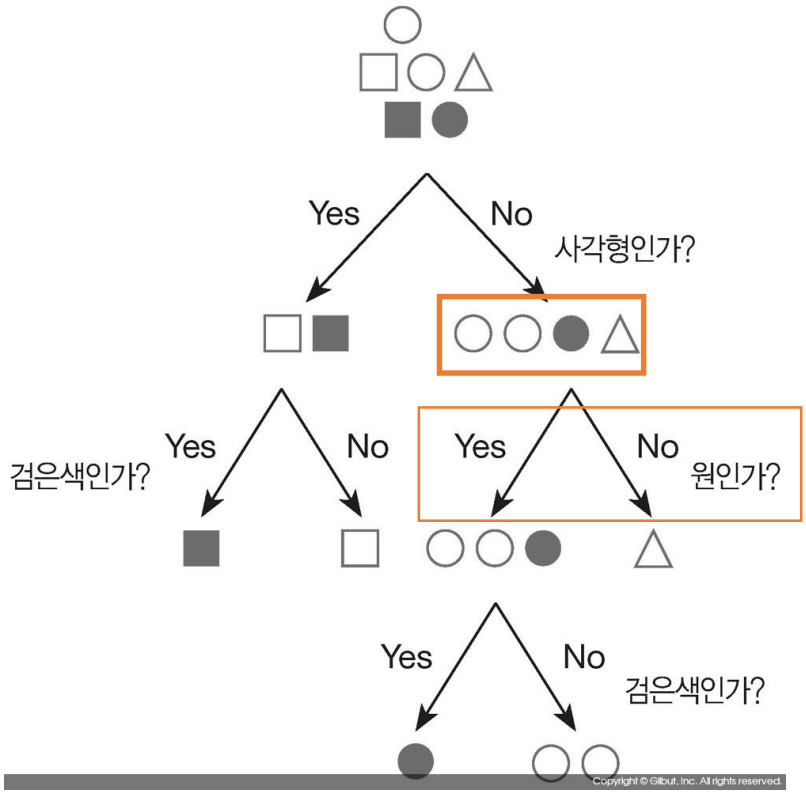
예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

흰색 삼각형이 분류되는 과정
1. 사각형인가? -> No

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?



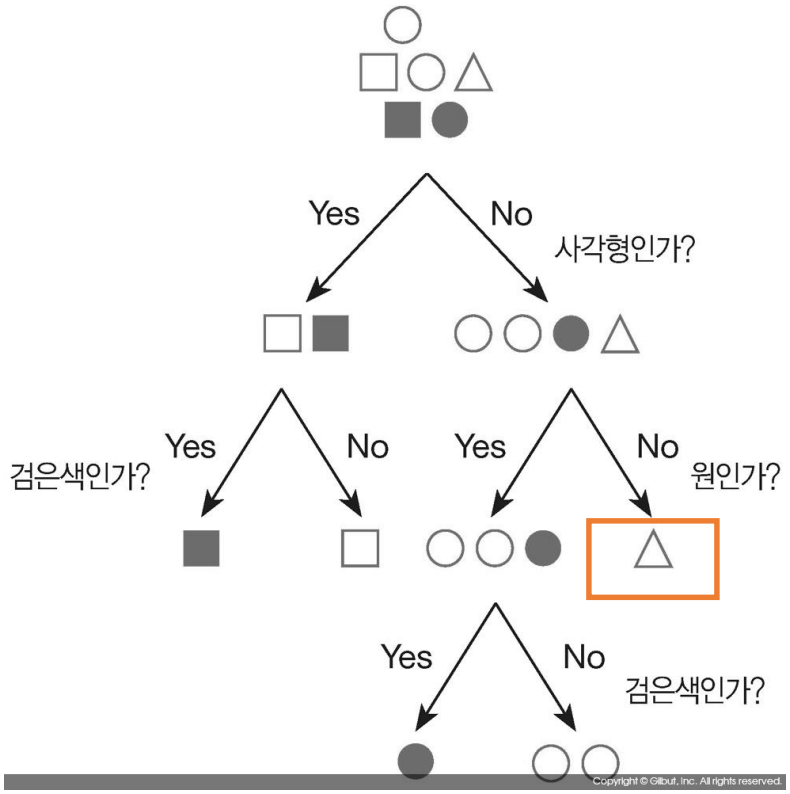
예시: 다양한 도형을 분류하는 의사 결정 나무

검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

흰색 삼각형이 분류되는 과정
1. 사각형인가? -> No
2. 원인가?

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?



예시: 다양한 도형을 분류하는 의사 결정 나무

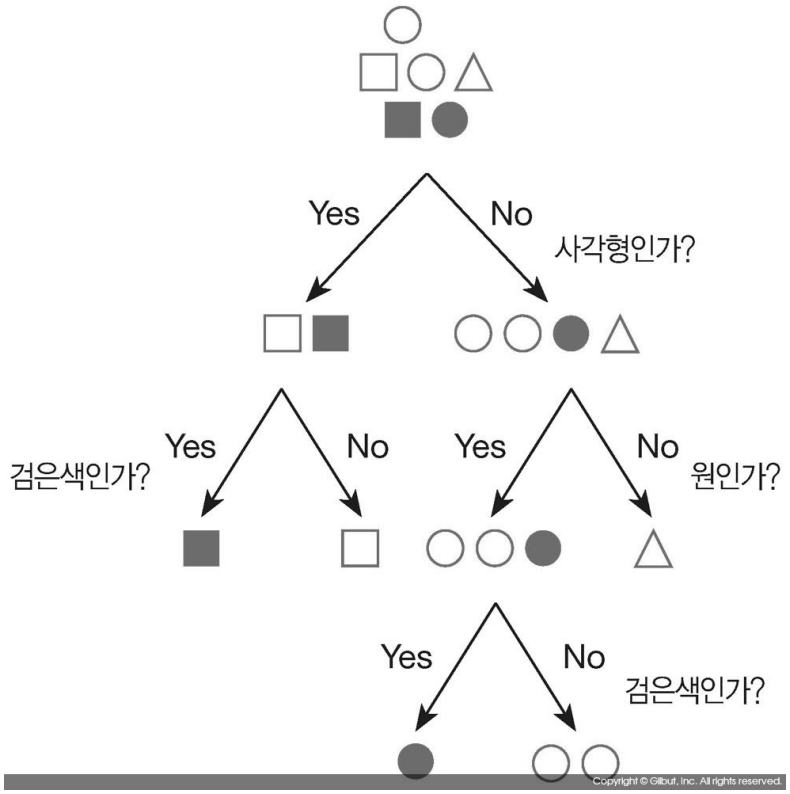
검은색 원이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> Yes
3. 검은색인가? -> Yes

흰색 삼각형이 분류되는 과정
1. 사각형인가? -> No
2. 원인가? -> No

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

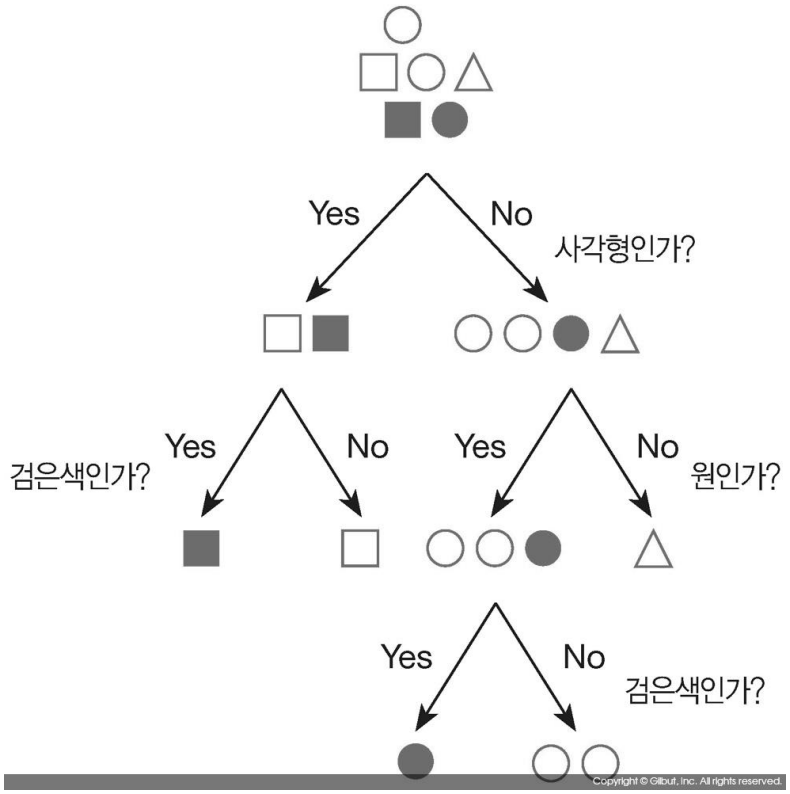
데이터의 특징에 대한 질문을 하며 응답에 따라 데이터를 분류해가는 알고리즘



Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

데이터의 특징에 대한 질문을 하며 응답에 따라 데이터를 분류해가는 알고리즘

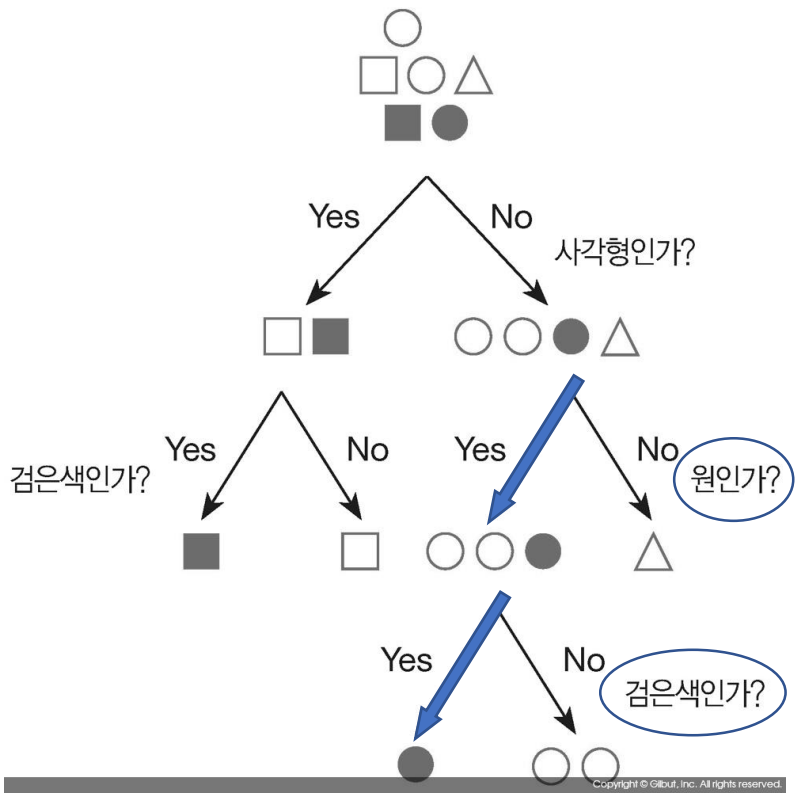


각 단계에서의 질문은 상위 단계의 질문과 연관성이 있다.

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

의사결정나무(Decision Tree)란?

데이터의 특징에 대한 질문을 하며 응답에 따라 데이터를 분류해가는 알고리즘



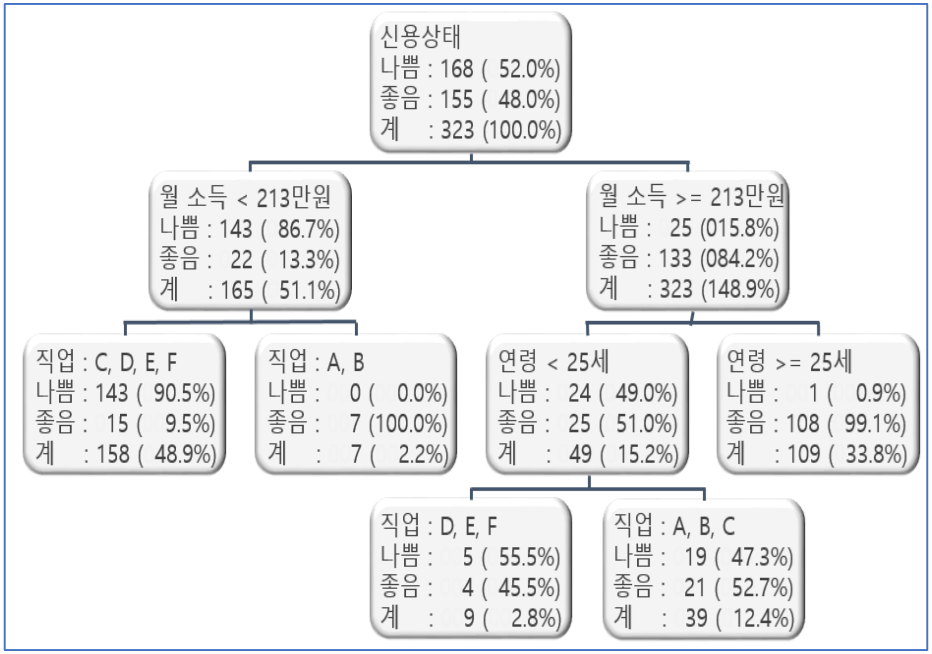
검은색 원을 분류해내는 마지막 질문인 "검은색인가?"라는 질문은 "원인가?"라는 질문에 대한 대답이 Yes인 경우에 이어지는 질문이다.

따라서 "검은색인가?"라는 질문은 모양에 대한 질문과 연관성이 있다.

이런 특징 때문에 나무 모형은 특징(feature)의 연관성을 잘 표현한다고 한다.

의사결정나무(Decision Tree)란?

의사결정규칙을 나무구조로 도표화하여 분류와 예측을 수행하는 통계 분석 기법이며,
통계적인 알고리즘을 활용하여 의사결정규칙을 시각적으로 나타냄



예) '나쁨/좋음 신용상태'를 여러 변수들에 따라 분류하는 의사결정나무

의사결정나무의 주요 목적

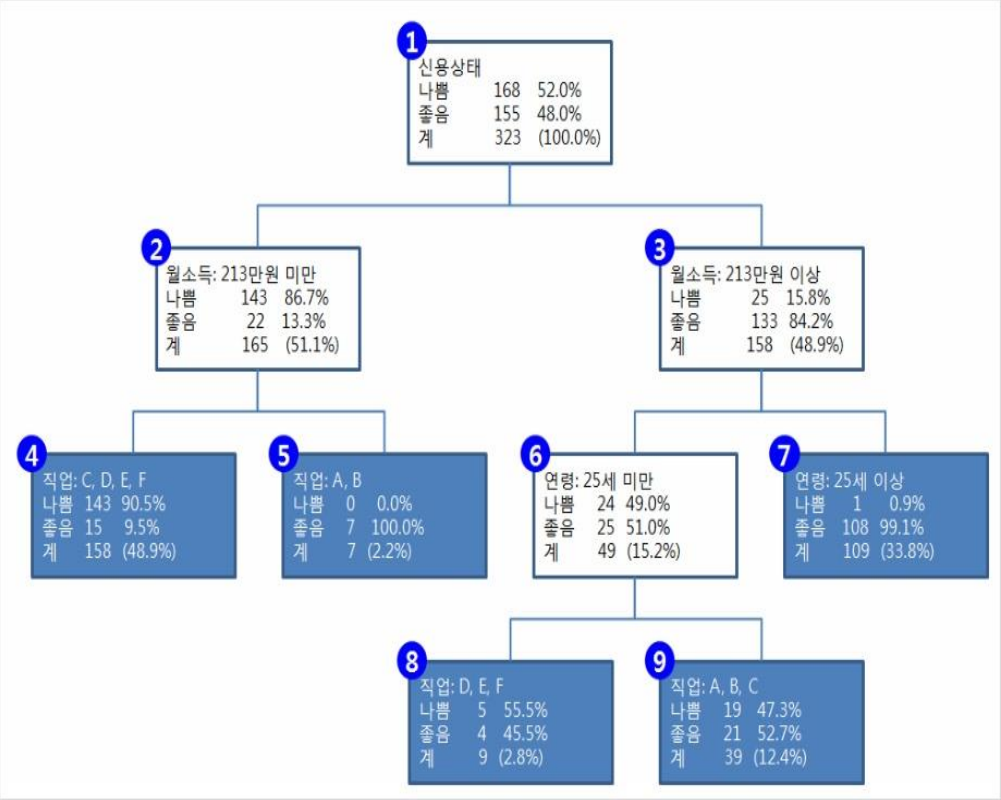
① 세분화: 각 고객이 어떤 집단에 속하는지 파악하고자 하는 경우

② 분류: 여러 변수에 근거해 반응 변수의 범주를 분류하고자 하는 경우

③ 예측: 데이터에서 규칙을 찾아내서 미래의 사건을 예측하고자 하는 경우

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

주요 특징



- 뿌리 마디(Root node)

  : 나무 구조가 시작되는 노드 ①

- 중간 마디(Intermediate node)

  : 끝 마디가 아닌 노드들 ②③⑥

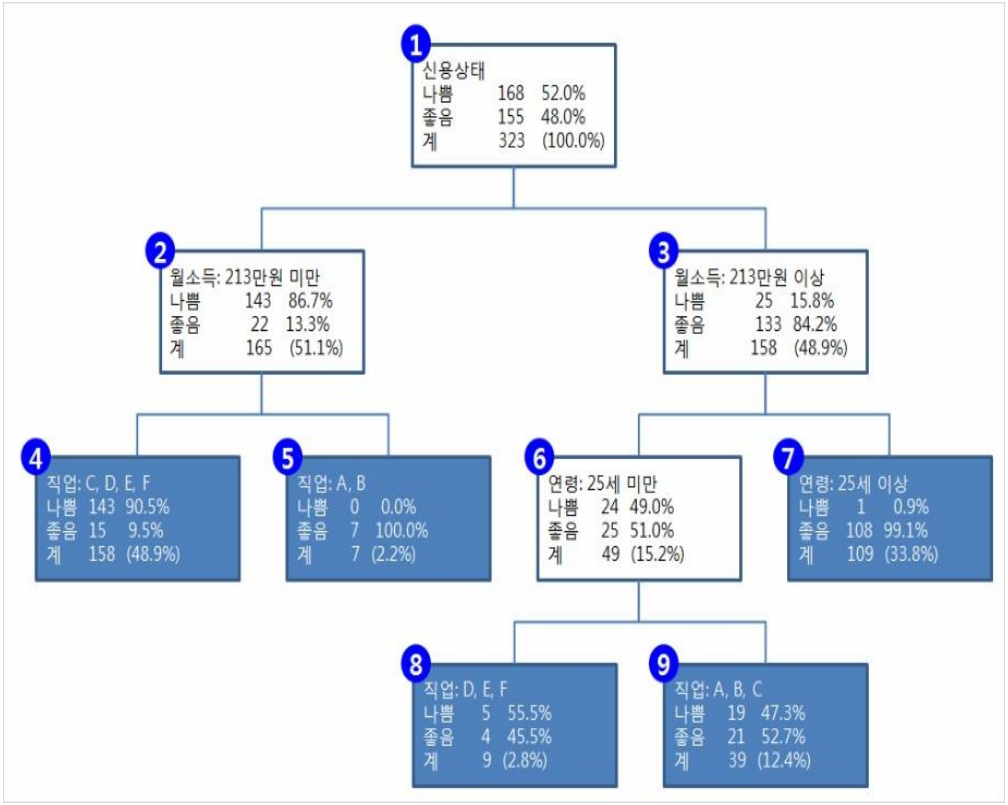
- 끝 마디(Terminal node)

  : 각 나무줄기의 끝으로 분류 규칙은 끝나는 노드 개수만큼 생성

  ④⑤⑦⑧⑨

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

주요 특징



- 뿌리마디의 질문이 왜 소득인가?

  -> 분할기준(splitting rule)의 선택

- 4번, 5번, 7번 마디들은 끝마디인 반면 6번 마디는 왜 중간마디인가?

  -> 정지규칙(stopping rule)

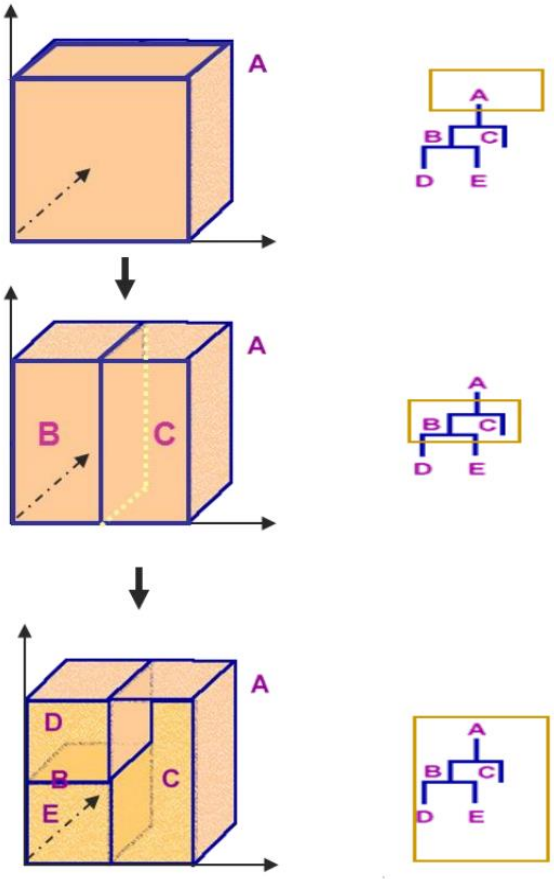
- 7번 마디에 속하는 자료는 신용상태를 어떻게 결정하여야 하는가?

  -> 끝마디에서의 예측값 할당법

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

분류(Classification)와 회귀(Regression)의 의미

Ex) 특징(feature)이 3개인 데이터에 의사결정나무를 적용



1. 전체 데이터 A : 아무런 분할이 일어나지 않은 상태

2. 전체 데이터 A가 두 개의 부분집합 B와 C로 분할된 상태

3. B가 D와 E로 분할되면서 전체 데이터 A가 세 개의 부분집합으로 분할 된 상태

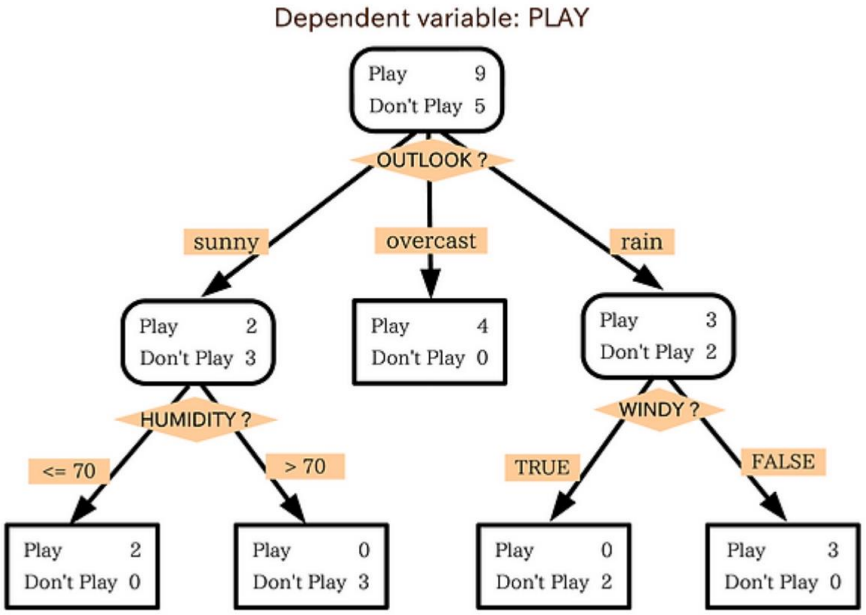
Ex) D특성을 가진 새로운 데이터가 주어졌을 때 D집합을 대표할 수 있는 값(분류=최빈값, 회귀=평균)을 반환하는 방식으로 예측

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

분류 나무(Classification Tree)

- 의사결정나무는 분류(Classification)와 회귀(Regression) 모두 가능



의사결정나무의 분류(Classification)

새로운 데이터가 특정 Terminal node에 속한다는 정보를 확인

해당 Terminal node에서 가장 빈도가 높은 범주에 새로운 데이터를 분류

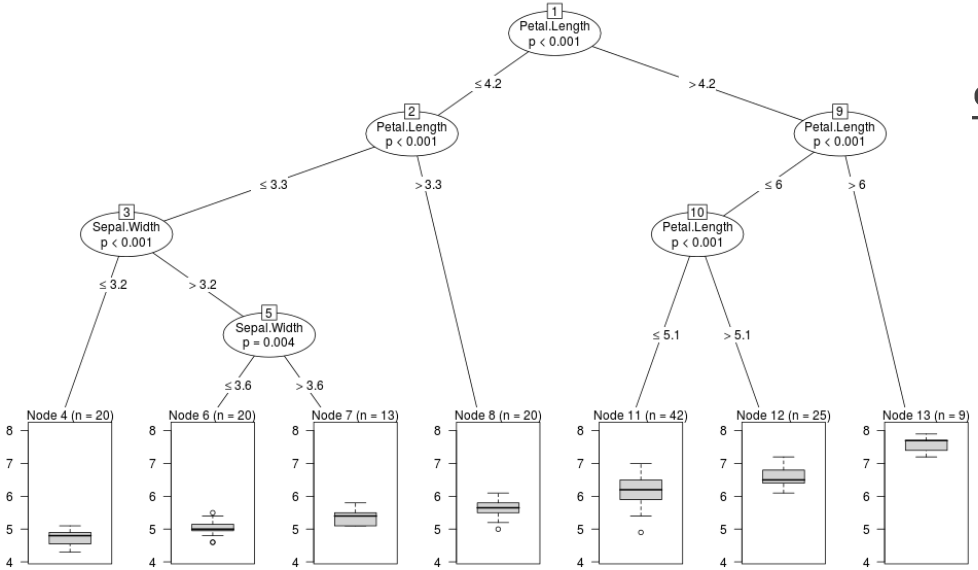
- ✓ Ex) 날씨는 맑은데(sunny) 습도가 70을 넘는 날(HUMIDITY ? > 70)은 경기가 열리지 않을 것이다.

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 1. 의사결정나무(Decision Tree)란 무엇인가?

회귀 나무(Regression Tree)

- 의사결정나무는 분류(Classification)와 회귀(Regression) 모두 가능



의사결정나무의 회귀(Regression)

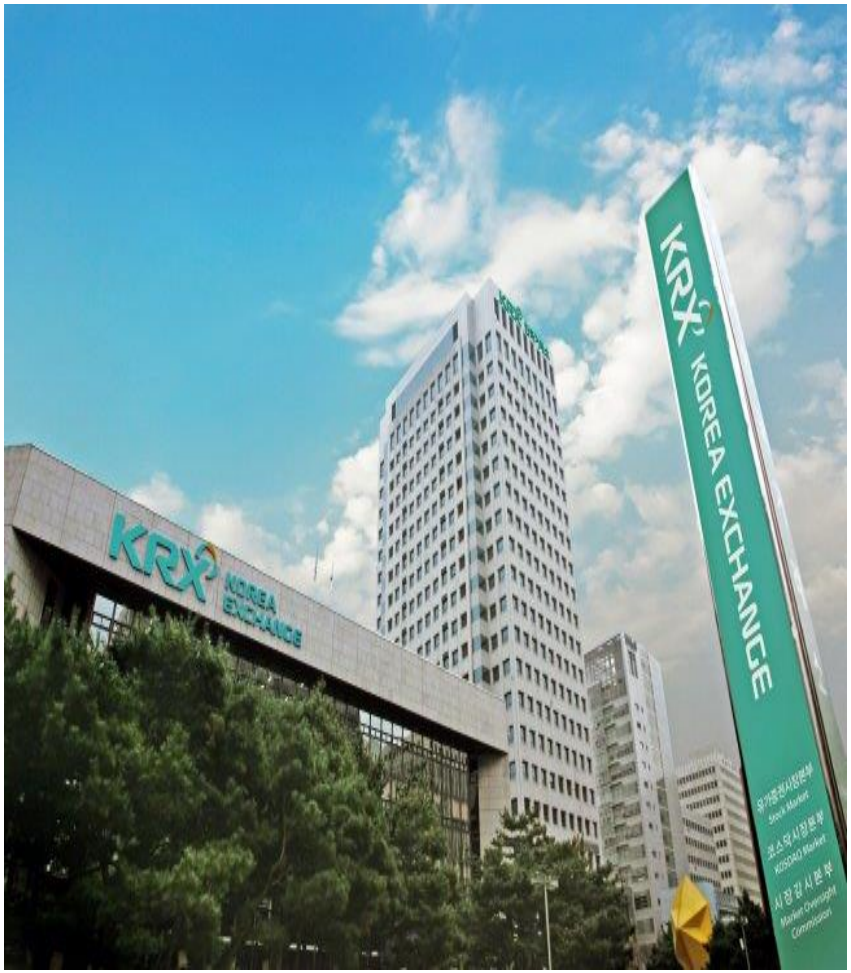
해당 Terminal node의 종속변수(y)의 평균을 예측값으로 반환

예측값의 종류는 Terminal node의 개수와 일치

source : https://stackoverflow.com/questions/29131254/how-to-generate-a-prediction-interval-from-a-regression-tree-rpart-object

Goal 2. 의사결정나무는 어떻게 활용될 수 있는가?

한국거래소 시장감시시스템 구축 사례



한국거래소 'AI 기반 차세대 시장감시시스템 EXIGHT 가동'
노컷뉴스 | 5일 전 | 네이버뉴스

[CBS (사진=**한국거래소** 제공)**한국거래소** 시장감시위원회는 3일 인공지능(AI) 기반 차세대... 또, 설명력 높은 다양한 변수를 적용한 인공지능모델(**XGboost**)을 통해 계좌의 불공정 혐의를 판단함으로써 기존에...

- **한국거래소**, '지방선거 대비' AI 시장… 국제신문 | 4일 전
- **한국거래소**, 작전세력 잡는 AI '엑사… 서울경제 | 5일 전 | 네이버뉴스
- **한국거래소**, AI로 작전세력 잡는다.… 베타뉴스 | 5일 전
- **한국거래소**, 불공정거래 잡는 인공… 프라임경제 | 4일 전

관련뉴스 25건 전체보기 >

거래소 AI 기반 차세대 시장감시시스템 본격 가동 아주경제 | 5일 전

최신형 기자 tlsgud80@ajunews.com **한국거래소**는 3일 서울사옥에서 인공지능(AI) 기반 차세대 시장감시시스템... **거래소** 관계자는 "새로운 시장감시시스템은 최신 AI 모델 엑스지부스트(**XGboost**)를 통해 이상 거래가...

AI·빅데이터가 자본시장 '경찰 역할' 한다 Korea IT Times | 4일 전

한국거래소 시장감시위원회는 지난 3일 총 18개월에 걸쳐 개발된 차세대 시장감시시스템 EXIGHT 가동을... EXIGHT가 적용하고 있는 모델은 의사결정 트리 기계학습 기법을 활용한 AI 모델 'XGboost'이다. 이 시스템은 계좌...

한국거래소, 세계 첫 AI '시장감시시스템' 구축 완료…내달 본격 가동
이투데이 | 2018.02.20.

이투데이=하유미 기자 | 불공정거래 혐의 계좌 검색 '5일→1시간' **한국거래소**가 인공지능(AI)과 빅데이터... 최신 AI 모델인 '엑스지부스트(XGBoost)'를 사용한 AI시장감시시스템은 기존 2~3개에서 54개로 변수를...

한국거래소 "AI로 불공정거래 의심계좌 1시간만에" 서울경제 | 2018.01.25. | 네이버뉴스

적용하면 **한국**이 첫 사례가 될 것"이라고 설명했다. AI 시장감시시스템은 최신 모델인 엑스지부스트(XGBoost)를 사용, 총 80억원을 들여 **거래소**에서 단독으로 개발했다. **거래소**는 이와 함께 잠재적 불공정거래군에 대한...

- AI·빅데이터로 신종 불공정거래 잡는다 내일신문 | 2018.01.25.

Source : https://blog.naver.com/yskinn/221269882569

Goal 2. 의사결정나무는 어떻게 활용될 수 있는가?

KB 국민카드 스마트 오퍼링 시스템 사례

신용카드의 '진화'...소비패턴 따라 맞춤 혜택!

KB국민카드도 자체 빅데이터를 활용해 카드사가 자동으로 각 상황에 맞는 최적의 혜택을 제공하는 '스마트 오퍼링 시스템(Smart Offering System)'을 출시했다.

KB국민카드의 스마트 오퍼링 시스템은 일별 800만건 이상 카드 승인 데이터를 '아프리오리(Apriori) 알고리즘' 및 '디시전트리(Decision Tree)' 기법으로 분석한 뒤, 고객의 행동 시점 니즈에 적합한 혜택을 실시간으로 제공하는 마케팅 시스템이다.

앞서 신한카드는 지난 4월 2200만 빅데이터를 기반으로 별도의 할인쿠폰 없이 고객의 라이프스타일과 소비패턴 등을 고려해 자동으로 할인해주는 개인별 맞춤형 서비스 '샐리(Sally)'를 출시한 바 있다.

정훈 KB금융지주경영연구소 연구위원은 "카드정보는 금융산업 측면에서 양적이나 질적으로 가장 좋은 정보를 담고 있다"며 "카드사들의 이 같은 흐름은 고객의 라이프스타일에 맞는 최적의 상품이나 부가서비스 제공을 가능하게 했다"라고 말했다.

이어 "빅데이터 활용은 기존 카드사들이 지급결제나 카드론 등의 업무에 치우치던 성향을 변화시켜 가맹점 컨설팅 등 신사업을 확대하는 계기가 될 가능성이 높다"라고 전망했다.

Source : http://www.seoulfn.com/news/articleView.html?idxno=231989

2 반응 변수 적용 분리 기준

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

✓ 반응 변수가 범주형인 경우

- **카이제곱 통계량의 p값** : p값이 가장 작은 예측 변수와 그 때의 최적 분리에 의해서 자식 마디를 형성
- **지니 지수** : 불순도를 측정하는 하나의 지수로서 지니 지수를 가장 감소시켜주는 예측 변수와 그 때의 최적 분리에 의해서 자식 마디를 선택
- **엔트로피 지수** : 이 지수가 가장 작은 예측 변수와 그 때의 최적 분리에 의해서 자식 마디 형성

✓ 반응 변수가 수치형인 경우

- **분산 분석에서의 F 통계량** : p값이 가장 작은 예측 변수와 그 때의 최적 분리에 의해서 자식 마디 형성
- **분산의 감소량** : 예측 오차를 최소화하는 것과 동일한 기준으로 분산의 감소량을 최대화하는 기준의 최적 분리에 의해서 자식 마디 형성

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

지니 지수(Gini Index)의 정의

- 집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다

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

✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 비율

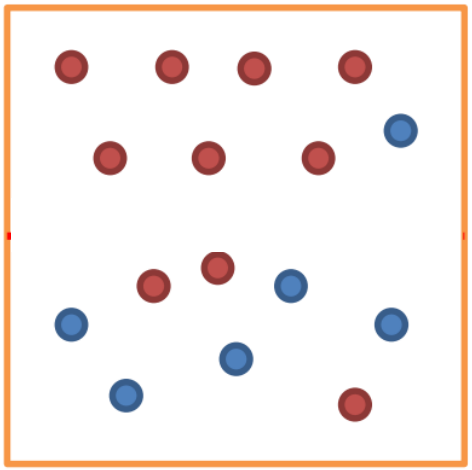
Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

지니 지수(Gini Index)의 정의

집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다

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

✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 비율



예시: 전체 16개 중 빨간색 10개, 파란색 6개

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

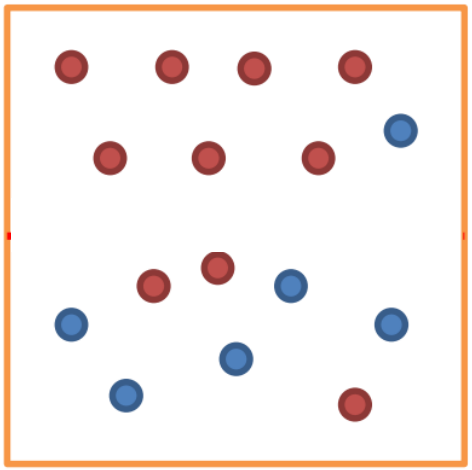
Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

지니 지수(Gini Index)의 정의

집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다

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

✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 비율



예시: 전체 16개 중 빨간색 10개, 파란색 6개

$$p_{\text{빨강}} = \frac{10}{16}$$

$$p_{\text{파랑}} = \frac{6}{16}$$

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

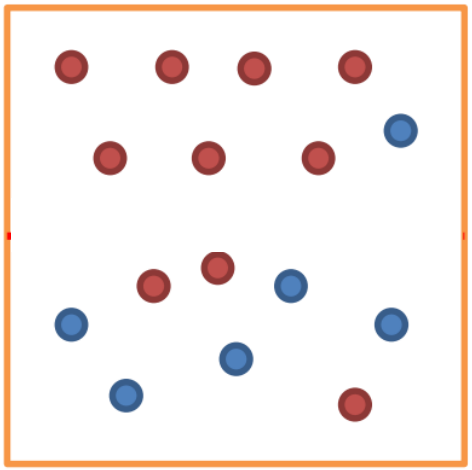
Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

지니 지수(Gini Index)의 정의

집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다

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

✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 비율



예시: 전체 16개 중 빨간색 10개, 파란색 6개

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

$$= 0.47$$

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

지니 지수(Gini Index)의 정의

- 집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다

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

✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 비율

✓ 모든 레코드가 동일한 범주에 속할 경우 $I(A) = 1 - \left(\frac{\text{전체 개수}}{\text{전체 개수}}\right)^2 = 1 - 1 = 0$

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

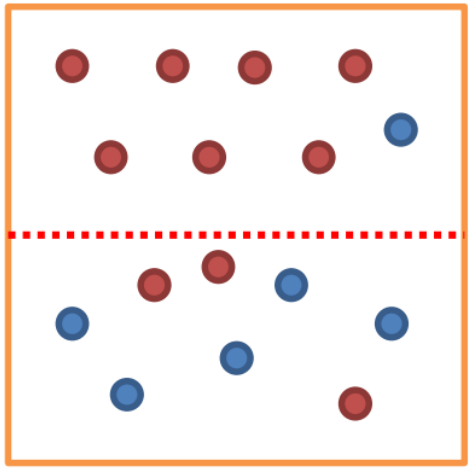
✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율



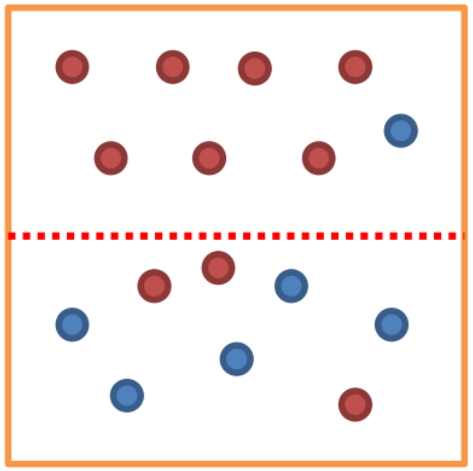
예시: 전체 16개를 8개/8개로 분할
위 : 빨간색 7개/파란색 1개 아래 : 빨간색 3개/파란색 5개

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율



예시: 전체 16개를 8개/8개로 분할

위 : 빨간색 7개/파란색 1개 아래 : 빨간색 3개/파란색 5개

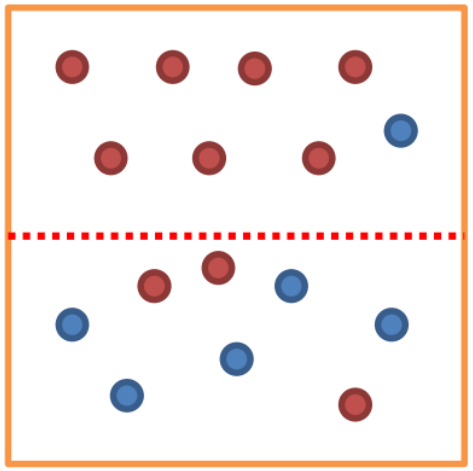
$$I(A) = (\ )\times\left(1-\left(\frac{7}{8}\right)^2-\left(\frac{1}{8}\right)^2\right) + (\ )\times\left(1-\left(\frac{3}{8}\right)^2-\left(\frac{5}{8}\right)^2\right)$$

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율



예시: 전체 16개를 8개/8개로 분할

위 : 빨간색 7개/파란색 1개 아래 : 빨간색 3개/파란색 5개

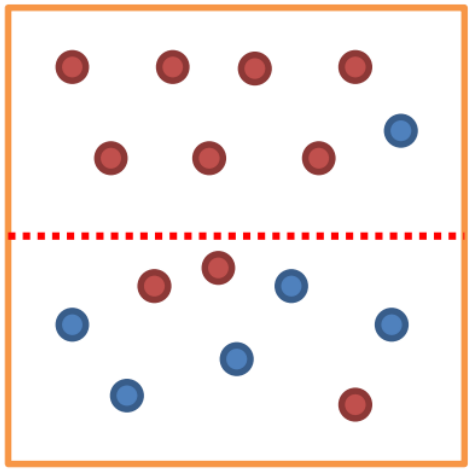
$$I(A) = \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2\right) + \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2\right)$$

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율



예시: 전체 16개를 8개/8개로 분할

위 : 빨간색 7개/파란색 1개 아래 : 빨간색 3개/파란색 5개

$$I(A) = R_1 \times I(A_1) + R_2 \times I(A_2)$$

$$= \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2\right) + \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2\right)$$

$$= 0.34$$

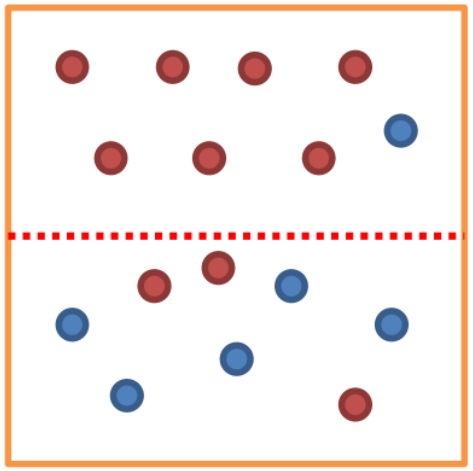
(각각에 대해서 지니 계수를 계산하고, 비율 R_i를 가중치로 고려한다고 생각해도 좋다.)

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 지니 지수(Gini Index)

$$I(A) = \sum_{i=1}^{d} (R_i (1 - \sum_{k=1}^{m} p_{ik}^2))$$

✓ R_i = 분할 전 레코드 중 분할 후 i 영역에 속하는 레코드의 비율



예시: 전체 16개를 8개/8개로 분할
위 : 빨간색 7개/파란색 1개 아래 : 빨간색 3개/파란색 5개

$$I(A) = \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2\right) + \left(\frac{8}{16}\right) \times \left(1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2\right) = 0.34$$

✓ 분할 후의 지니 지수 변화: 0.47 - 0.34 = 0.13 만큼 지니 지수 감소

(=불확실성 감소=순도 증가=정보 획득)

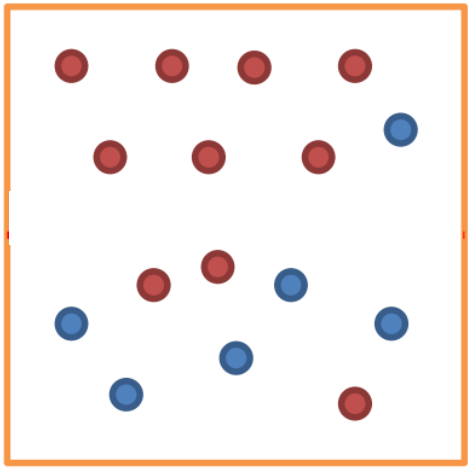
Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

엔트로피 지수(Entropy Index)의 정의

집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표로 불순도를 측정한다.

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

- ✓ m개의 레코드가 속하는 A영역에 대한 엔트로피
- ✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 수



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

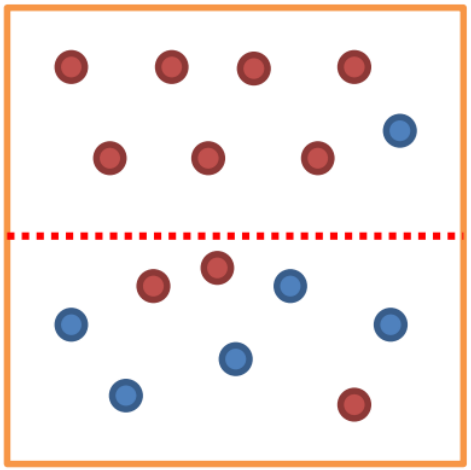
- ✓ 모든 레코드가 동일한 범주에 속할 경우 $Entropy(A) = 0$

Goal 3. 반응 변수에 적용되는 분리 기준에는 무엇이 있는가?

두 개 이상의 영역에 대한 엔트로피 지수(Entropy Index)

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

- ✓ m개의 레코드가 속하는 A영역에 대한 엔트로피
- ✓ p_k = A영역에 속한 레코드 중 k 범주에 속하는 레코드의 수
- ✓ R_i = 분할 전 레코드 가운데 분할 후 i 영역에 속하는 레코드의 비율



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

- ✓ 분할 후 엔트로피 변화: 0.95 - 0.75 = 0.2 만큼 엔트로피 감소

  (=불확실성 감소=순도 증가=정보 획득)

3

의사결정나무 학습과정

Goal 4. 의사결정나무의 학습 과정

의사결정나무의 학습 과정

- 재귀적 분기(Recursive Partitioning)
  - ✓ 구분하기 전보다 구분된 뒤에 각 영역의 순도(purity, homogeneity)가 증가하도록 입력 변수의 영역을 두 개로 구분한다.
- 가지치기(Pruning)
  - ✓ 과적합(Overfitting)을 방지하기 위하여 너무 자세하게 구분된 영역을 통합한다.

의사결정나무의 학습 진행 방향

- 구분 뒤 각 영역의 순도가 증가/불확실성(엔트로피, 지니 지수)이 최대한 감소하는 방향으로 학습을 진행

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법

- 특정 영역에 속하는 개체들을 하나의 기준 변수 값의 범위에 따라 분기한다.
- 새로 생성될 자식 노드의 동질성이 최대화되도록 분기점을 선택한다.
- 불순도(Impurity)를 측정하는 기준으로는 범주형 변수에 대해 지니계수, 수치형 변수에 대해 분산을 이용한다.
- 분기 횟수가 정해지지 않은 채로 사전에 설정한 기준을 만족할 때까지 분기를 반복하는 데서 재귀적 분기라는 이름이 붙었다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

- 24개 가정을 대상으로 소득(Income), 주택크기(Lot size) 잔디깎기 기계 구입 여부(Ownership)를 조사한 데이터
- Owner 12명, Non-owner 12명

Income	Lot size	Ownership	Income	Lot size	Ownership
60.0	18.4	Owner	75.0	19.6	Non-owner
85.5	16.8	Owner	52.8	20.8	Non-owner
64.8	21.6	Owner	64.8	17.2	Non-owner
61.5	20.8	Owner	43.2	20.4	Non-owner
87.0	23.6	Owner	84.0	17.6	Non-owner
110.1	19.2	Owner	49.2	17.6	Non-owner
108.0	17.6	Owner	59.4	16.0	Non-owner
82.8	22.4	Owner	66.0	18.4	Non-owner
69.0	20.0	Owner	47.4	16.4	Non-owner
93.0	20.8	Owner	33.0	18.8	Non-owner
51.0	22.0	Owner	51.0	14.0	Non-owner
81.0	20.0	Owner	63.0	14.8	Non-owner

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

- 소득과 주택크기를 설명변수(X), 기계 구입 여부를 종속변수(Y)로 하는 분류나무 모형

X		Y			
Income	Lot size	Ownership	Income	Lot size	Ownership
60.0	18.4	Owner	75.0	19.6	Non-owner
85.5	16.8	Owner	52.8	20.8	Non-owner
64.8	21.6	Owner	64.8	17.2	Non-owner
61.5	20.8	Owner	43.2	20.4	Non-owner
87.0	23.6	Owner	84.0	17.6	Non-owner
110.1	19.2	Owner	49.2	17.6	Non-owner
108.0	17.6	Owner	59.4	16.0	Non-owner
82.8	22.4	Owner	66.0	18.4	Non-owner
69.0	20.0	Owner	47.4	16.4	Non-owner
93.0	20.8	Owner	33.0	18.8	Non-owner
51.0	22.0	Owner	51.0	14.0	Non-owner
81.0	20.0	Owner	63.0	14.8	Non-owner

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 1. 한 변수 기준으로 정렬 ex) Lot size를 오름차순으로 정렬

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 1. 한 변수 기준으로 정렬 ex) Lot size를 오름차순으로 정렬
- 2. 가능한 모든 분기점에 대해 지니 지수를 구해
  분할 전과 비교해 정보획득을 조사한다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 1. 한 변수 기준으로 정렬 ex) Lot size를 오름차순으로 정렬
- 2. 가능한 모든 분기점에 대해 지니 지수를 구해
  분할 전과 비교해 정보획득을 조사한다.

✓ 2-1. 분기지점을 첫 레코드와 나머지 23개 레코드로 설정

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 1. 한 변수 기준으로 정렬 ex) Lot size를 오름차순으로 정렬
- 2. 가능한 모든 분기점에 대해 지니 지수를 구해 분할 전과 비교해 정보획득을 조사한다.
  - ✓ 2-1. 분기지점을 첫 레코드와 나머지 23개 레코드로 설정
  - ✓ 1번 레코드와 2~24번 레코드 간의 지니 지수를 구한 뒤 이를 분기 전 지니 지수와 비교해 정보획득을 조사한다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 1. 한 변수 기준으로 정렬 ex) Lot size를 오름차순으로 정렬
- 2. 가능한 모든 분기점에 대해 지니 지수를 구해 분할 전과 비교해 정보획득을 조사한다.

✓ 2-1. 분기지점을 첫 레코드와 나머지 23개 레코드로 설정

✓ 1번 레코드와 2~24번 레코드 간의 지니 지수를 구한 뒤 이를 분기 전 지니 지수와 비교해 정보획득을 조사한다.

✓ 분기 전 지니 지수 = $1-(\frac{12}{24})^2-(\frac{12}{24})^2 = 0.5$

분기 후 지니 지수 = $\frac{1}{24}(1-(\frac{0}{1})^2-(\frac{1}{1})^2) + \frac{23}{24}(1-(\frac{12}{23})^2-(\frac{11}{23})^2)$

$= 0.478$

정보획득 = 0.5 – 0.478 = 0.022

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 2-2. 이후 분기 지점을 두번째 레코드로 두고 처음 두 개 레코드와 나머지 22개 레코드 간의 지니 지수를 계산한 뒤 정보획득을 알아본다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 2-2. 이후 분기 지점을 두번째 레코드로 두고 처음 두 개 레코드와 나머지 22개 레코드 간의 지니 지수를 계산한 뒤 정보획득을 알아본다.

✓ 분기지점을 첫 두 개의 레코드와 나머지 22개 레코드로 설정

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

- 2-2. 이후 분기 지점을 두번째 레코드로 두고 처음 두 개 레코드와 나머지 22개 레코드 간의 지니 지수를 계산한 뒤 정보획득을 알아본다.

✓ 분기지점을 첫 두 개의 레코드와 나머지 22개 레코드로 설정

✓ 1~2번 레코드와 3~24번 레코드 간의 지니 지수를 구한 뒤 이를 분기 전 지니 지수와 비교해 정보획득을 조사한다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner
15	69.0	20.0	owner
16	81.0	20.0	owner
17	43.2	20.4	non-owner
18	61.5	20.8	owner
19	93.0	20.8	owner
20	52.8	20.8	non-owner
21	64.8	21.6	owner
22	51.0	22.0	owner
23	82.8	22.4	owner
24	87.0	23.6	owner

☑ 2-2. 이후 분기 지점을 두번째 레코드로 두고 처음 두 개 레코드와 나머지 22개 레코드 간의 지니 지수를 계산한 뒤 정보획득을 알아본다.

✓ 분기지점을 첫 두 개의 레코드와 나머지 22개 레코드로 설정

✓ 1~2번 레코드와 3~24번 레코드 간의 지니 지수를 구한 뒤 이를 분기 전 지니 지수와 비교해 정보획득을 조사한다.

✓ 분기 전 지니 지수 = $1-(\frac{12}{24})^2-(\frac{12}{24})^2 = 0.5$

분기 후 지니 지수 = $\frac{1}{24}(1-(\frac{0}{2})^2-(\frac{2}{2})^2) + \frac{22}{24}(1-(\frac{12}{22})^2-(\frac{10}{22})^2)$

$= 0.455$

정보획득 = 0.5 – 0.455 = 0.045

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership	
1	51.0	14.0	non-owner	0.4782609
2	63.0	14.8	non-owner	0.4545455
3	59.4	16.0	non-owner	0.4285714
4	47.4	16.4	non-owner	0.4000000
5	85.5	16.8	owner	0.4526316
6	64.8	17.2	non-owner	0.4259259
7	108.0	17.6	owner	0.4621849
8	84.0	17.6	non-owner	0.4375000
9	49.2	17.6	non-owner	0.4074074
10	60.0	18.4	owner	0.4428571
11	66.0	18.4	non-owner	0.4125874
12	33.0	18.8	non-owner	0.3750000
13	110.1	19.2	owner	0.4125874
14	75.0	19.6	non-owner	0.3714286
15	69.0	20.0	owner	0.4074074
16	81.0	20.0	owner	0.4375000
17	43.2	20.4	non-owner	0.3949580
18	61.5	20.8	owner	0.4259259
19	93.0	20.8	owner	0.4526316
20	52.8	20.8	non-owner	0.4000000
21	64.8	21.6	owner	0.4285714
22	51.0	22.0	owner	0.4545455
23	82.8	22.4	owner	0.4782609
24	87.0	23.6	owner	

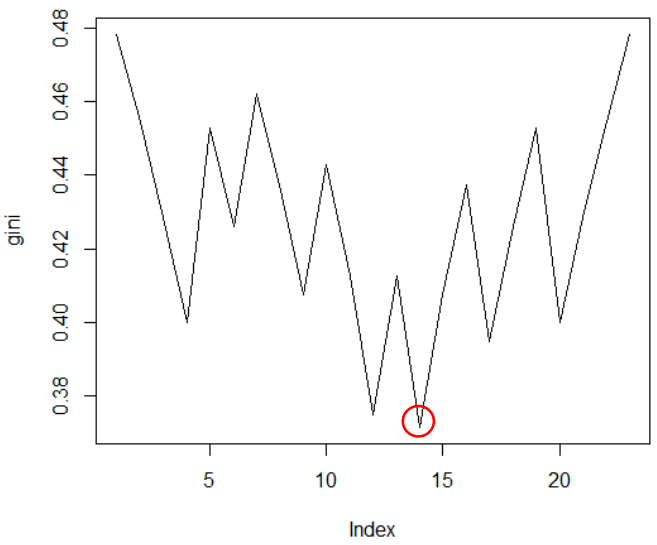
2-3. 이후 분기 지점을 세 번째, 네 번째, …, 23번째까지 두고 지니 지수를 계산해서 정보 획득이 최대가 되는 분기 지점을 찾는다.

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership	
1	51.0	14.0	non-owner	0.4782609
2	63.0	14.8	non-owner	0.4545455
3	59.4	16.0	non-owner	0.4285714
4	47.4	16.4	non-owner	0.4000000
5	85.5	16.8	owner	0.4526316
6	64.8	17.2	non-owner	0.4259259
7	108.0	17.6	owner	0.4621849
8	84.0	17.6	non-owner	0.4375000
9	49.2	17.6	non-owner	0.4074074
10	60.0	18.4	owner	0.4428571
11	66.0	18.4	non-owner	0.4125874
12	33.0	18.8	non-owner	0.3750000
13	110.1	19.2	owner	0.4125874
14	75.0	19.6	non-owner	0.3714286
15	69.0	20.0	owner	0.4074074
16	81.0	20.0	owner	0.4375000
17	43.2	20.4	non-owner	0.3949580
18	61.5	20.8	owner	0.4259259
19	93.0	20.8	owner	0.4526316
20	52.8	20.8	non-owner	0.4000000
21	64.8	21.6	owner	0.4285714
22	51.0	22.0	owner	0.4545455
23	82.8	22.4	owner	0.4782609
24	87.0	23.6	owner	

2-3. 이후 분기 지점을 세 번째, 네 번째, …, 23번째까지 두고 지니 지수를 계산해서 정보 획득이 최대가 되는 분기 지점을 찾는다.



14번째 분기 지점에서 지니 지수 최소

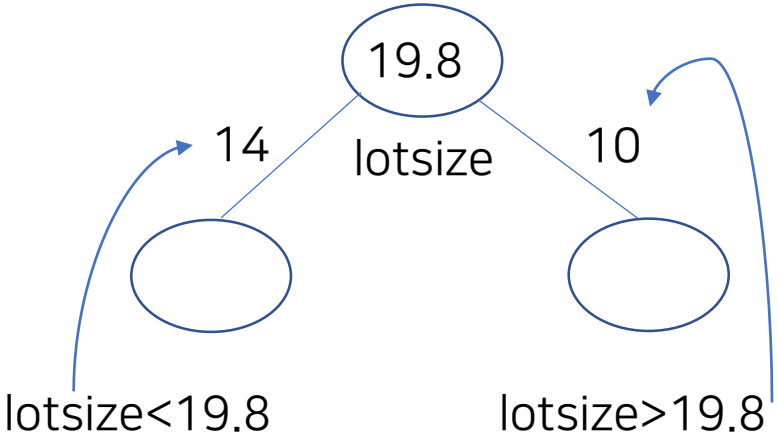
-> 정보 획득은 최대

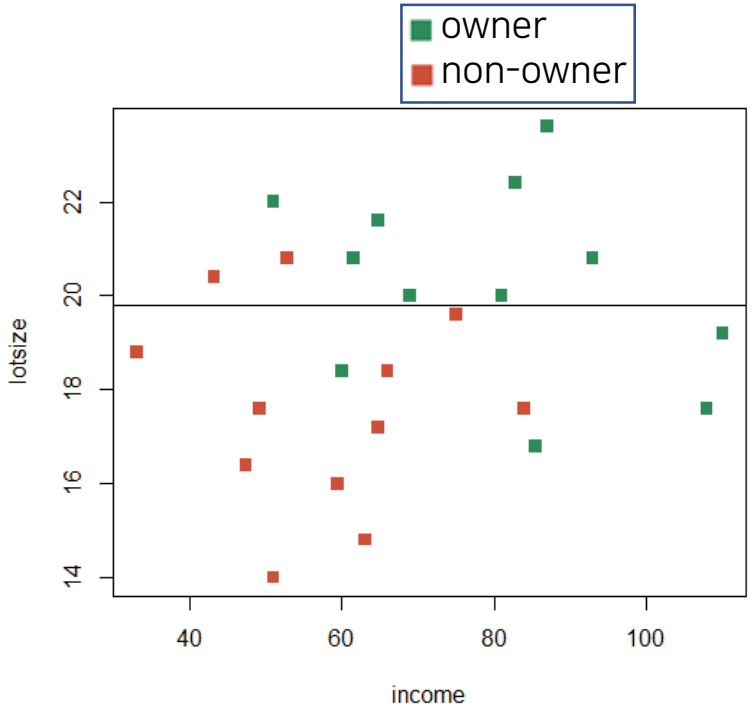
분기점은

(19.6+20.0)/2=19.8

재귀적 분기(Recursive Partitioning) 방법 예

2-3. 이후 분기 지점을 세 번째, 네 번째, …, 23번째까지 두고 지니 지수를 계산해서 정보 획득이 최대가 되는 분기 지점을 찾는다.





Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	51.0	14.0	non-owner
2	63.0	14.8	non-owner
3	59.4	16.0	non-owner
4	47.4	16.4	non-owner
5	85.5	16.8	owner
6	64.8	17.2	non-owner
7	108.0	17.6	owner
8	84.0	17.6	non-owner
9	49.2	17.6	non-owner
10	60.0	18.4	owner
11	66.0	18.4	non-owner
12	33.0	18.8	non-owner
13	110.1	19.2	owner
14	75.0	19.6	non-owner

- 3. 다른 변수인 소득을 기준으로 정렬하고 같은 작업을 반복

Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

	income	lotsize	ownership
1	33.0	18.8	non-owner
2	47.4	16.4	non-owner
3	49.2	17.6	non-owner
4	51.0	14.0	non-owner
5	59.4	16.0	non-owner
6	60.0	18.4	owner
7	63.0	14.8	non-owner
8	64.8	17.2	non-owner
9	66.0	18.4	non-owner
10	75.0	19.6	non-owner
11	84.0	17.6	non-owner
12	85.5	16.8	owner
13	108.0	17.6	owner
14	110.1	19.2	owner

- 3. 다른 변수인 소득을 기준으로 정렬하고 같은 작업을 반복

Goal 4. 의사결정나무의 학습 과정

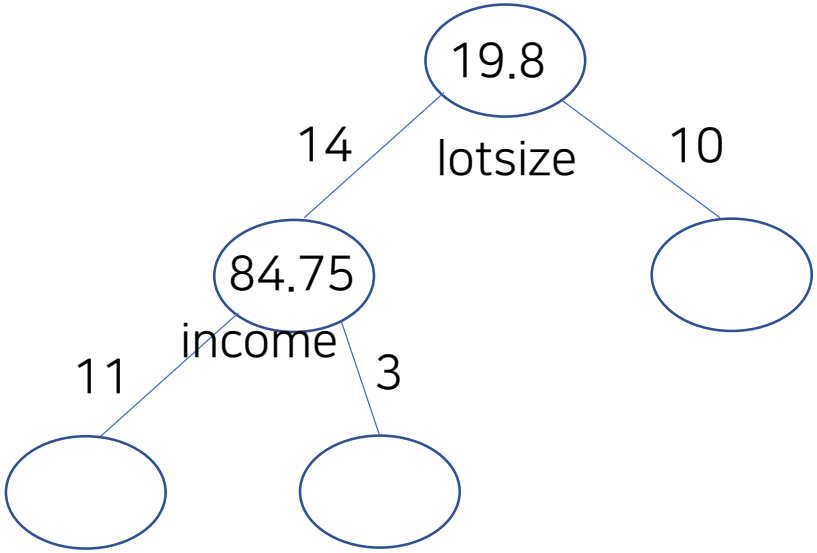
재귀적 분기(Recursive Partitioning) 방법 예

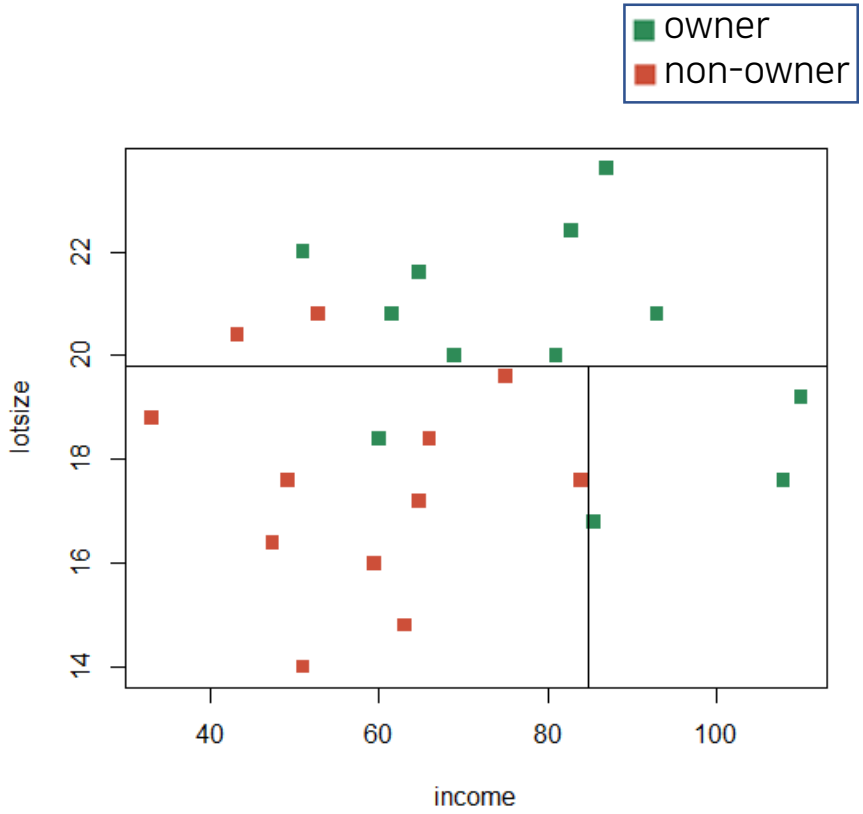
	income	lotsize	ownership
1	33.0	18.8	non-owner
2	47.4	16.4	non-owner
3	49.2	17.6	non-owner
4	51.0	14.0	non-owner
5	59.4	16.0	non-owner
6	60.0	18.4	owner
7	63.0	14.8	non-owner
8	64.8	17.2	non-owner
9	66.0	18.4	non-owner
10	75.0	19.6	non-owner
11	84.0	17.6	non-owner
12	85.5	16.8	owner
13	108.0	17.6	owner
14	110.1	19.2	owner

- 3. 다른 변수인 소득을 기준으로 정렬하고 같은 작업을 반복

재귀적 분기(Recursive Partitioning) 방법 예

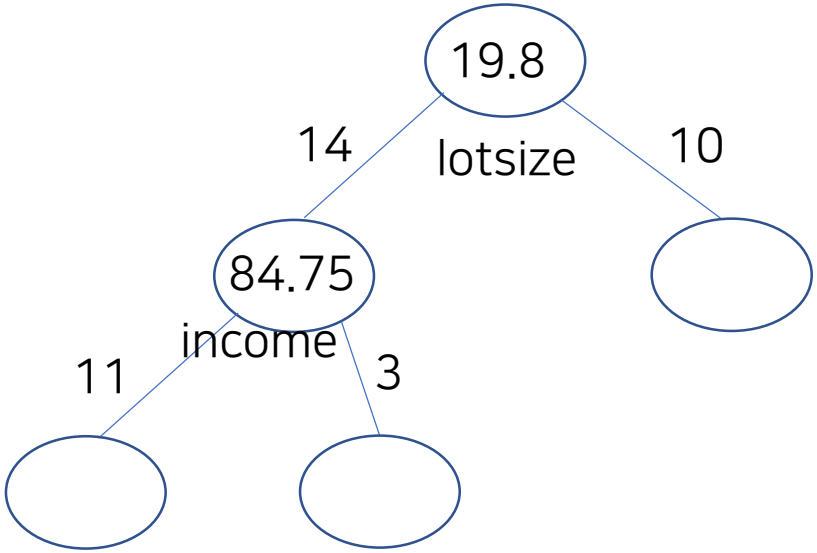
3. 다른 변수인 소득을 기준으로 정렬하고 다시 같은 작업을 반복

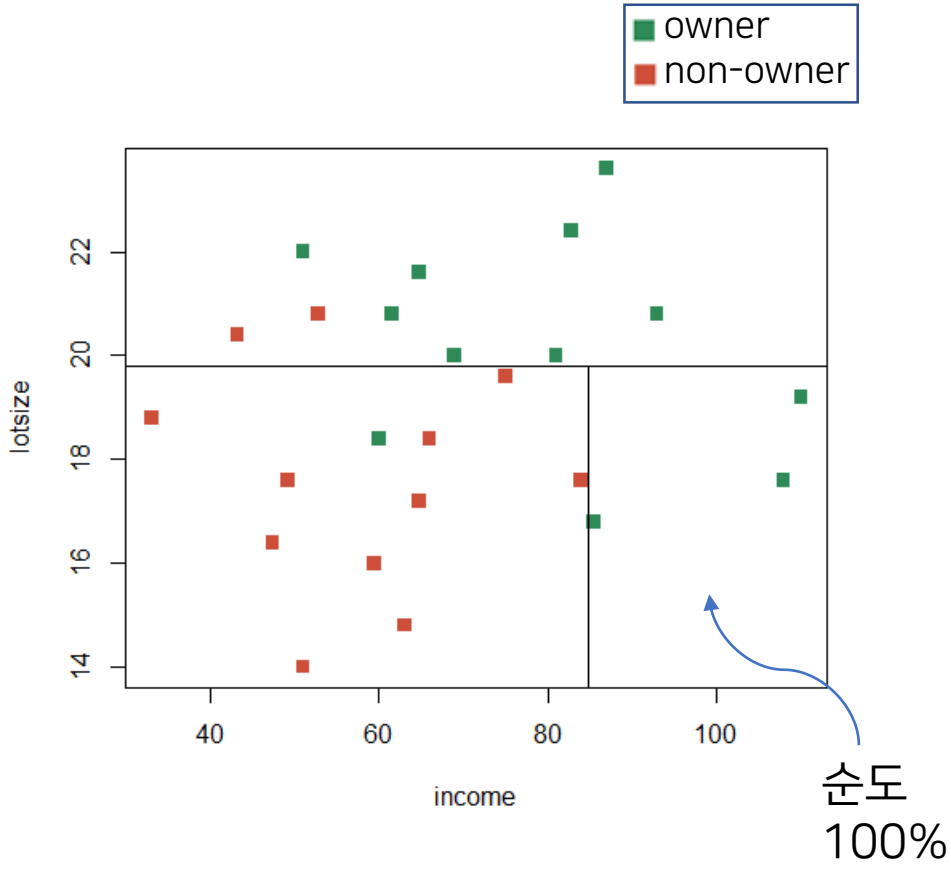




재귀적 분기(Recursive Partitioning) 방법 예

3. 다른 변수인 소득을 기준으로 정렬하고 다시 같은 작업을 반복
   (순도가 100%이면 분류 끝)

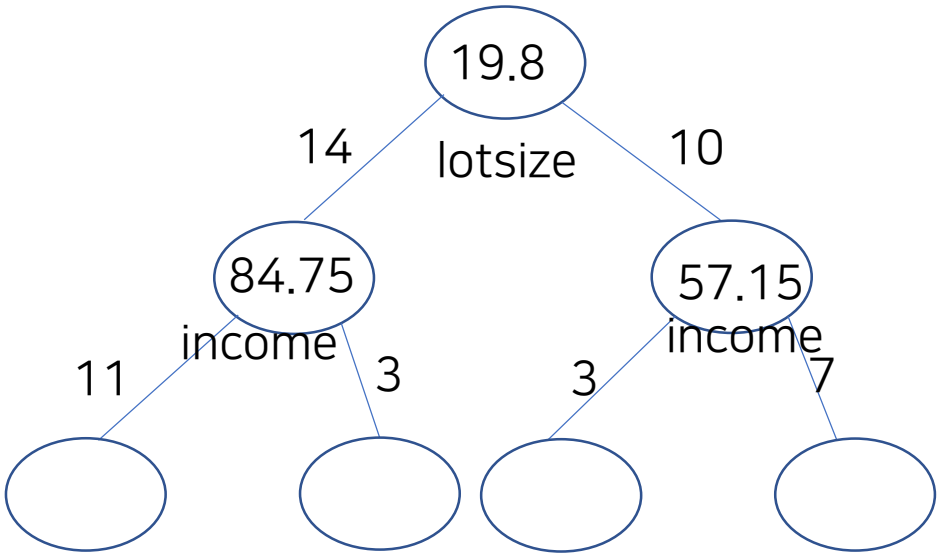


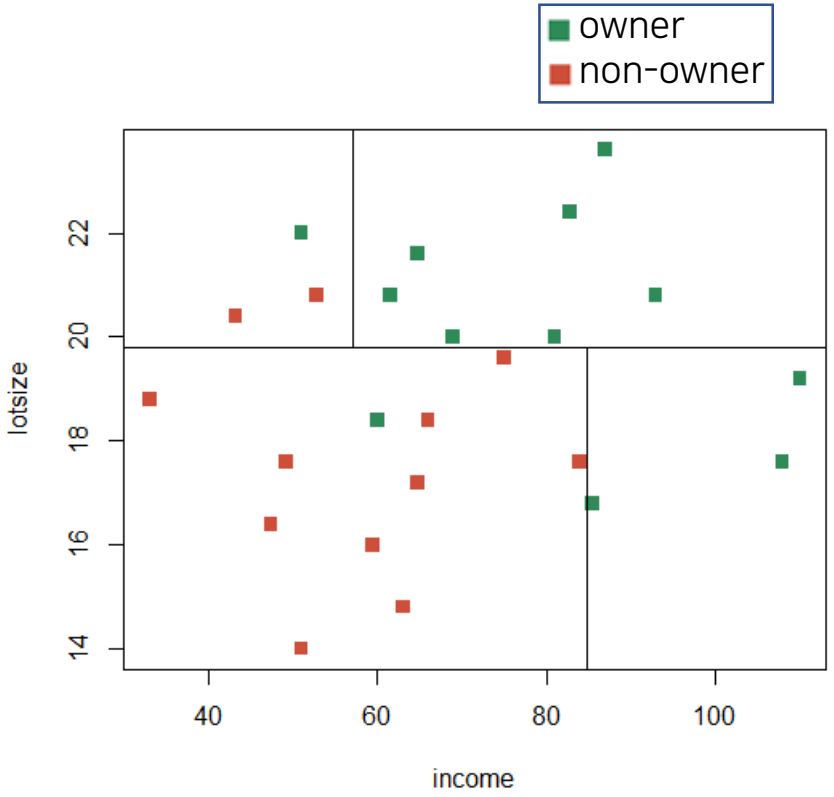


Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

3. 다른 변수인 소득을 기준으로 정렬하고 다시 같은 작업을 반복

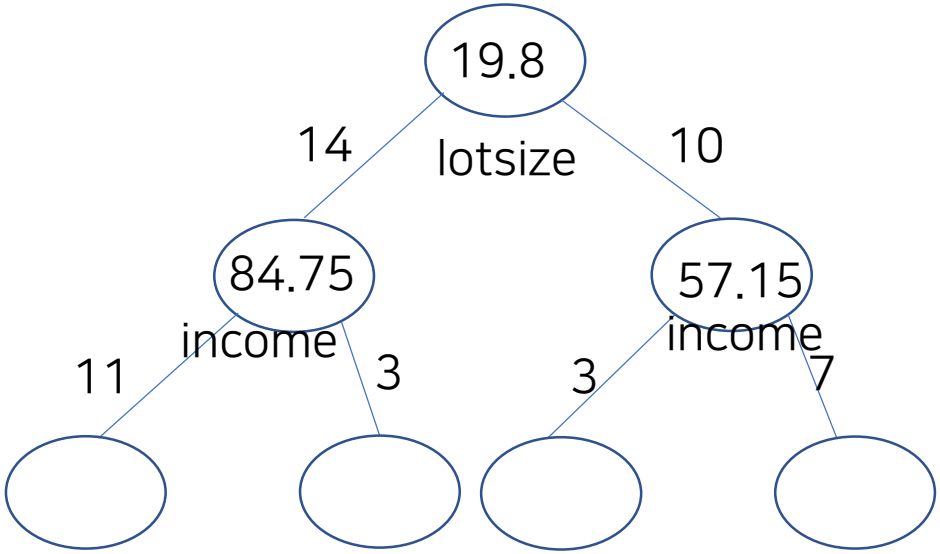


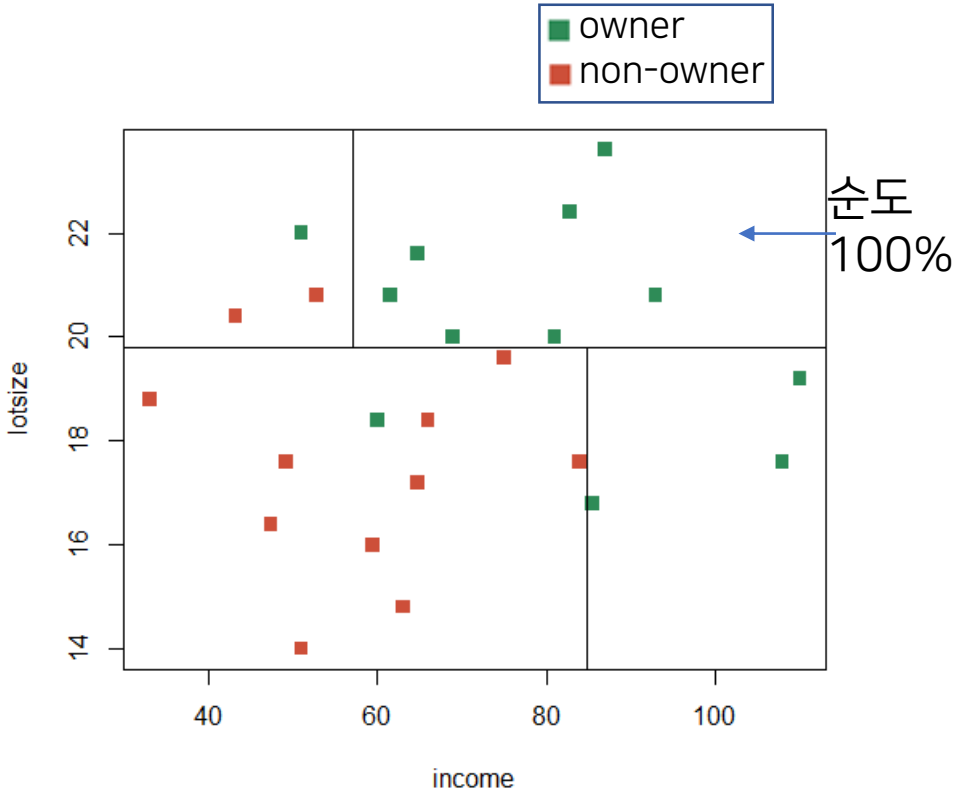


Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

3. 다른 변수인 소득을 기준으로 정렬하고 다시 같은 작업을 반복

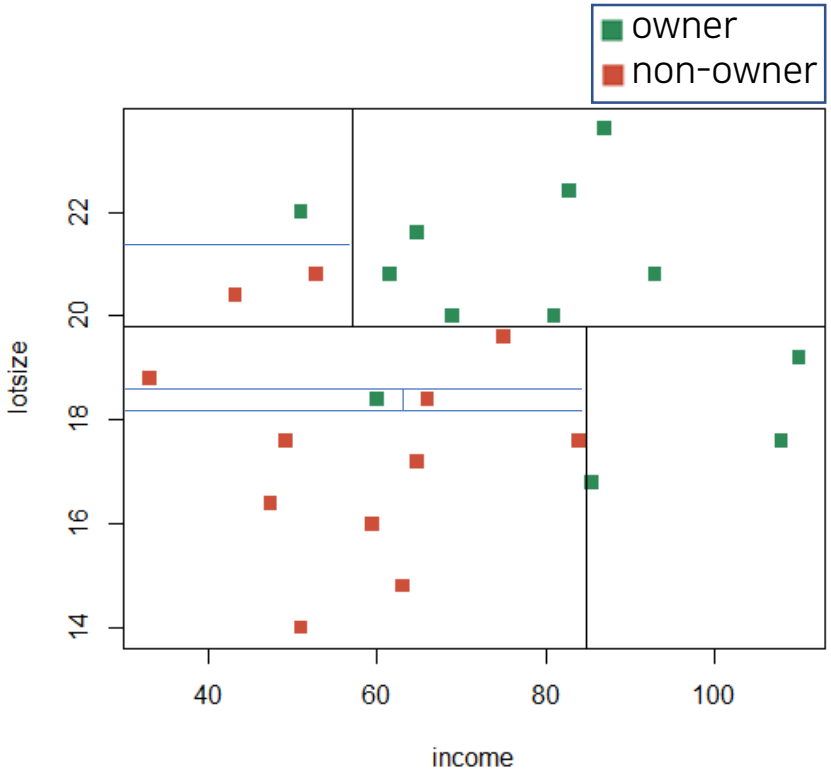




Goal 4. 의사결정나무의 학습 과정

재귀적 분기(Recursive Partitioning) 방법 예

4. 모든 (terminal) node의 순도가 100%가 될 때까지 반복 -> full tree



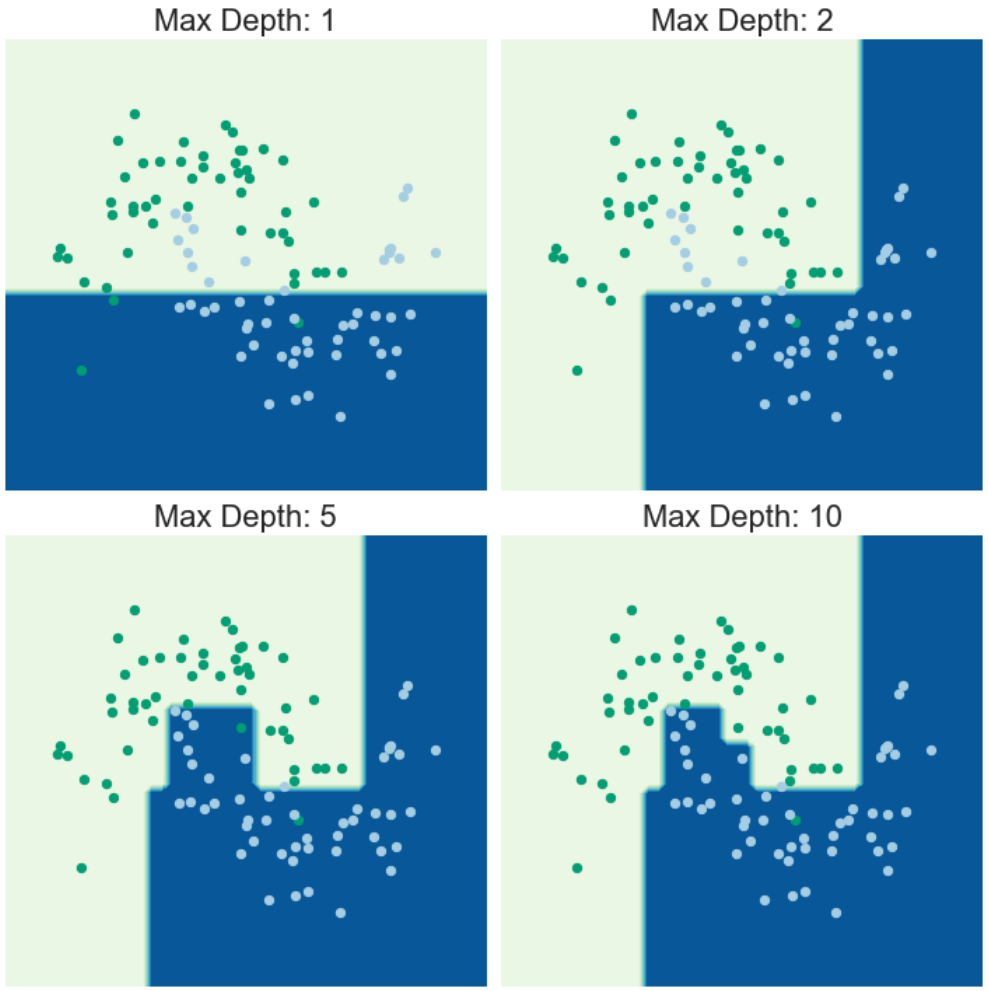
Goal 4. 의사결정나무의 학습 과정

가지치기(Pruning) 방법 (1/2)

- Full tree : 모든 Terminal node의 순도(homogeneity)가 100%인 상태
- Full tree를 생성한 뒤 적절한 수준에서 Terminal node를 결합한다.
- 의사결정나무의 분기 수가 증가할 때 처음에는 새로운 데이터에 대한 오분류율이 감소하나 일정 수준 이상이 되면 오분류율이 되레 증가한다.

- ✓ 검증데이터에 대한 오분류율이 증가하는 시점에서 적절히 가지치기를 수행해야 한다.
- ✓ 가지치기는 데이터를 버리는 개념이 아니라 분기를 합치는 개념으로 이해해야 한다.

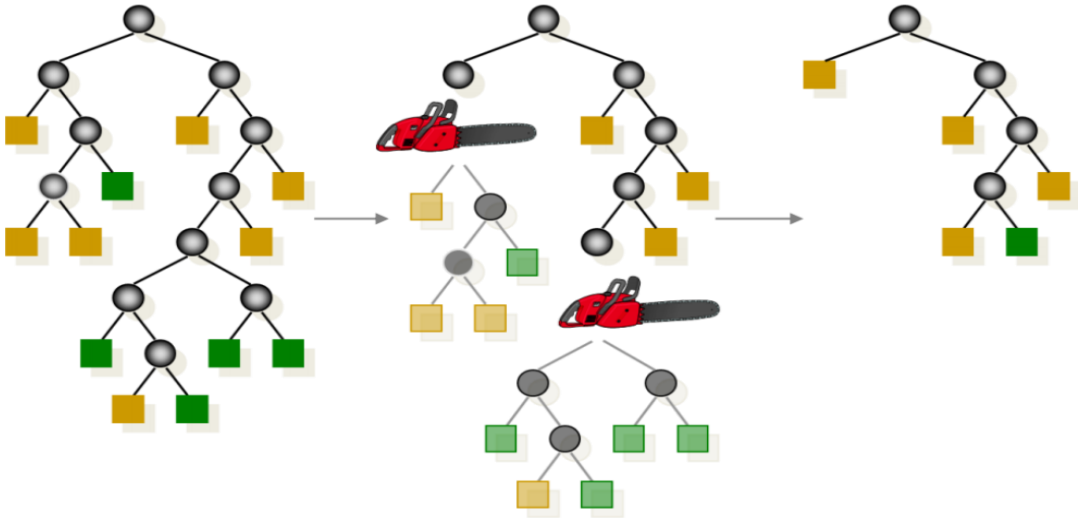
Goal 4. 의사결정나무의 학습 과정



source : http://engineering.pivotal.io/post/interpreting-decision-trees-and-random-forests/

가지치기(Pruning) 방법 (2/2)

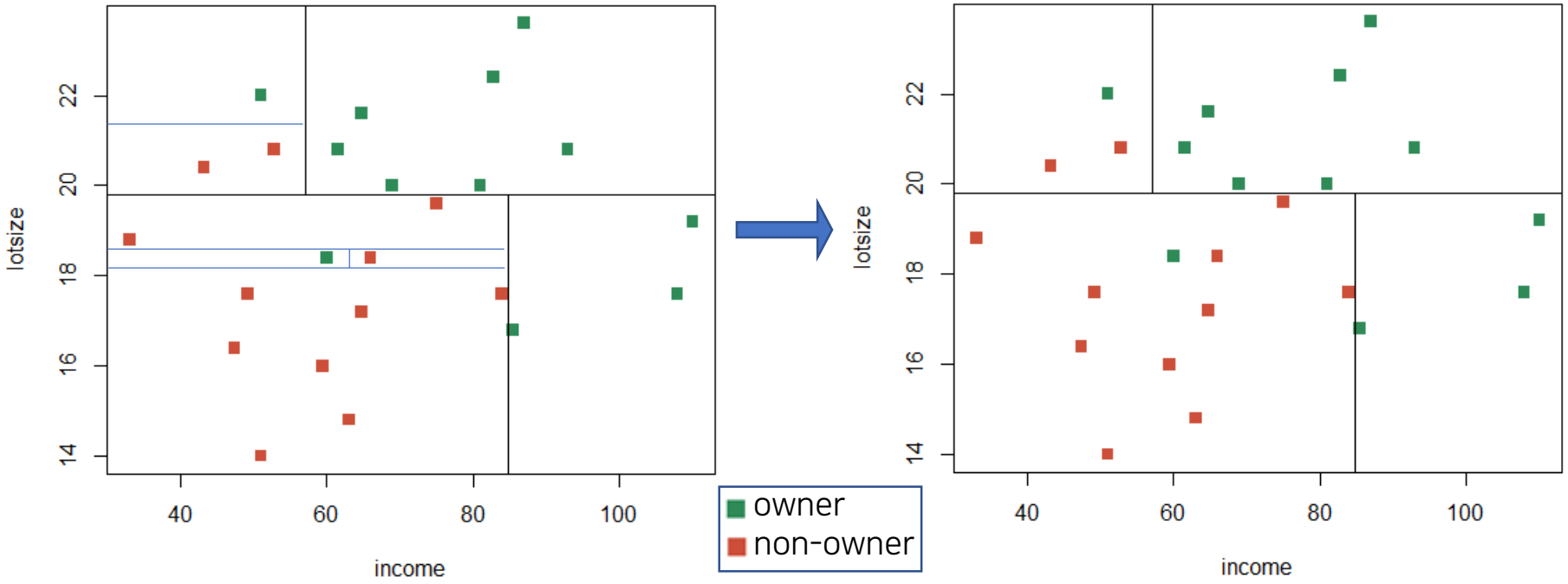
- 과적합(Overfitting)을 방지하기 위하여 하위 노드들을 상위 노드로 결합한다.
- Pre-pruning : Tree를 생성하는 과정에서 최소 분기 기준을 이용하는 사전적 가지치기
- Post-pruning : Full-tree 생성 후, 검증 데이터의 오분류율과 Tree의 복잡도 (끝 노드의 수)등을 고려하는 사후적 가지치기



< 가지치기의 예 >

가지치기(Pruning) 방법 (2/2)

- 과적합(Overfitting)을 방지하기 위하여 하위 노드들을 상위 노드로 결합한다.
- Pre-pruning : Tree를 생성하는 과정에서 최소 분기 기준을 이용하는 사전적 가지치기
- Post-pruning : Full-tree 생성 후, 검증 데이터의 오분류율과 Tree의 복잡도 (끝 노드의 수)등을 고려하는 사후적 가지치기



Goal 4. 의사결정나무의 학습 과정

(참고) 가지치기의 비용함수(cost function)

일반적으로 의사결정나무에서 가지치기를 할 때는 높은 정확도(낮은 오차)뿐만 아니라 모형의 복잡도(model complexity)까지 같이 고려하는 아래와 같은 형태의 비용함수(cost function)를 사용한다.

$$CC(T) = Err(T) + \alpha \times L(T)$$

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

(참고) 가지치기의 비용함수(cost function)

일반적으로 의사결정나무에서 가지치기를 할 때는 높은 정확도(낮은 오차)뿐만 아니라 모형의 복잡도(model complexity)까지 같이 고려하는 아래와 같은 형태의 비용함수(cost function)를 사용한다.

$$CC(T) = Err(T) + \alpha \times L(T)$$

$CC(T)$ = 의사결정나무의 비용 복잡도
(=오류가 적으면서 terminal node 수가 적은 단순한 모형일수록 작은 값)

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

(참고) 가지치기의 비용함수(cost function)

일반적으로 의사결정나무에서 가지치기를 할 때는 높은 정확도(낮은 오차)뿐만 아니라 모형의 복잡도(model complexity)까지 같이 고려하는 아래와 같은 형태의 비용함수(cost function)를 사용한다.

$$CC(T) = Err(T) + \alpha \times L(T)$$

$CC(T)$ = 의사결정나무의 비용 복잡도
(=오류가 적으면서 terminal node 수가 적은 단순한 모형일수록 작은 값)

$Err(T)$ = 검증데이터에 대한 오분류율

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

(참고) 가지치기의 비용함수(cost function)

일반적으로 의사결정나무에서 가지치기를 할 때는 높은 정확도(낮은 오차)뿐만 아니라 모형의 복잡도(model complexity)까지 같이 고려하는 아래와 같은 형태의 비용함수(cost function)를 사용한다.

$$CC(T) = Err(T) + \alpha \times L(T)$$

$CC(T)$ = 의사결정나무의 비용 복잡도
(=오류가 적으면서 terminal node 수가 적은 단순한 모형일수록 작은 값)

$Err(T)$ = 검증데이터에 대한 오분류율

$L(T)$ = terminal node의 수 -> 구조의 복잡도를 의미

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Goal 4. 의사결정나무의 학습 과정

(참고) 가지치기의 비용함수(cost function)

일반적으로 의사결정나무에서 가지치기를 할 때는 높은 정확도(낮은 오차)뿐만 아니라 모형의 복잡도(model complexity)까지 같이 고려하는 아래와 같은 형태의 비용함수(cost function)를 사용한다.

$$CC(T) = Err(T) + \alpha \times L(T)$$

$CC(T)$ = 의사결정나무의 비용 복잡도
(=오류가 적으면서 terminal node 수가 적은 단순한 모형일수록 작은 값)

$Err(T)$ = 검증데이터에 대한 오분류율

$L(T)$ = terminal node의 수 -> 구조의 복잡도를 의미

α = $Err(T)$와 $L(T)$를 결합하는 complexity parameter; hyperparameter
α보다 개선되지 않으면 가지치기 중단, 보통 0.01~0.1의 값을 씀

source : https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

의사결정나무(Decision Tree)의 장/단점

의사결정나무의 장점

- 화이트 박스(white box) 모형이며, 결과를 해석하고 이해하기 쉽다.
- 자료를 가공할 필요가 거의 없다.
- 수치 자료와 범주 자료 모두에 적용할 수 있다.
- 이상치(outlier) 자체를 하나의 경우로 분류하기 때문에 이상치에 안정적이다.
- 대규모의 데이터 셋에서도 잘 동작한다.

Source : https://ko.wikipedia.org/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95

의사결정나무(Decision Tree)의 장/단점

의사결정나무의 단점

- 최적 결정 트리를 알아낸다고 보장할 수는 없다.
- 과적합(overfitting): 훈련 데이터를 제대로 일반화하지 못할 경우 너무 복잡한 결정 트리를 만들 수 있다.
- 데이터의 특성이 특정 변수에 수직/수평적으로 구분되지 못할 때 분류율이 떨어지고, 트리가 복잡해지는 문제가 발생한다.
- 약간의 데이터 변화에 트리의 모양이 전혀 달라질 수 있다.
  즉 분산이 큰 불안정한 방법이다.

Source : https://ko.wikipedia.org/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95