Day, Date

ASAC 데이터 분석가 과정
12.08.2023

PRESENTER

이혜준

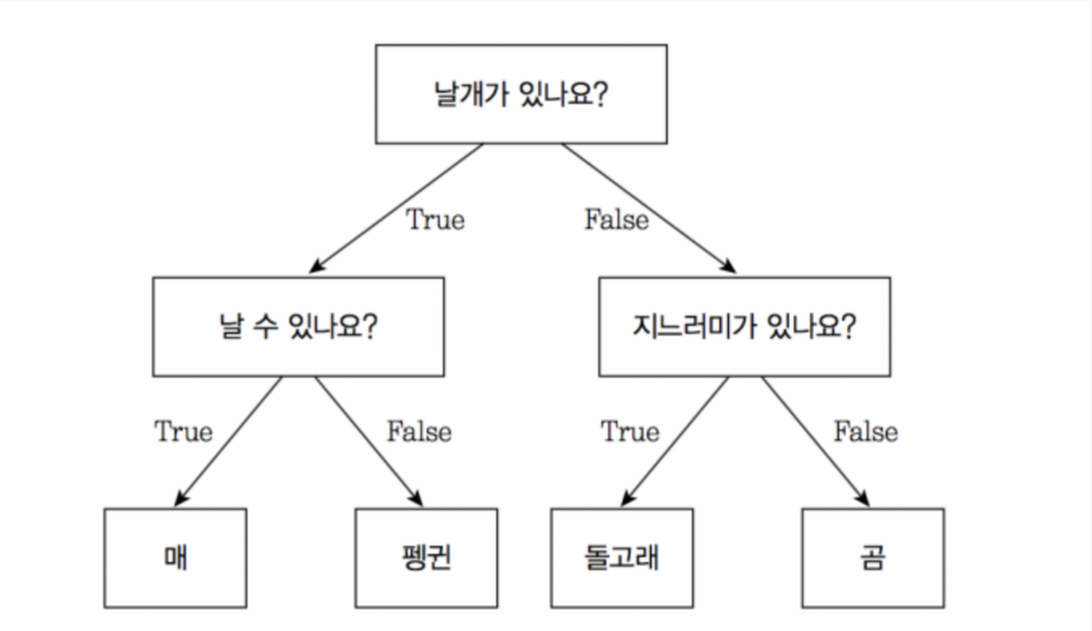
Decision Tree

Table of Contents

기본 프로세스

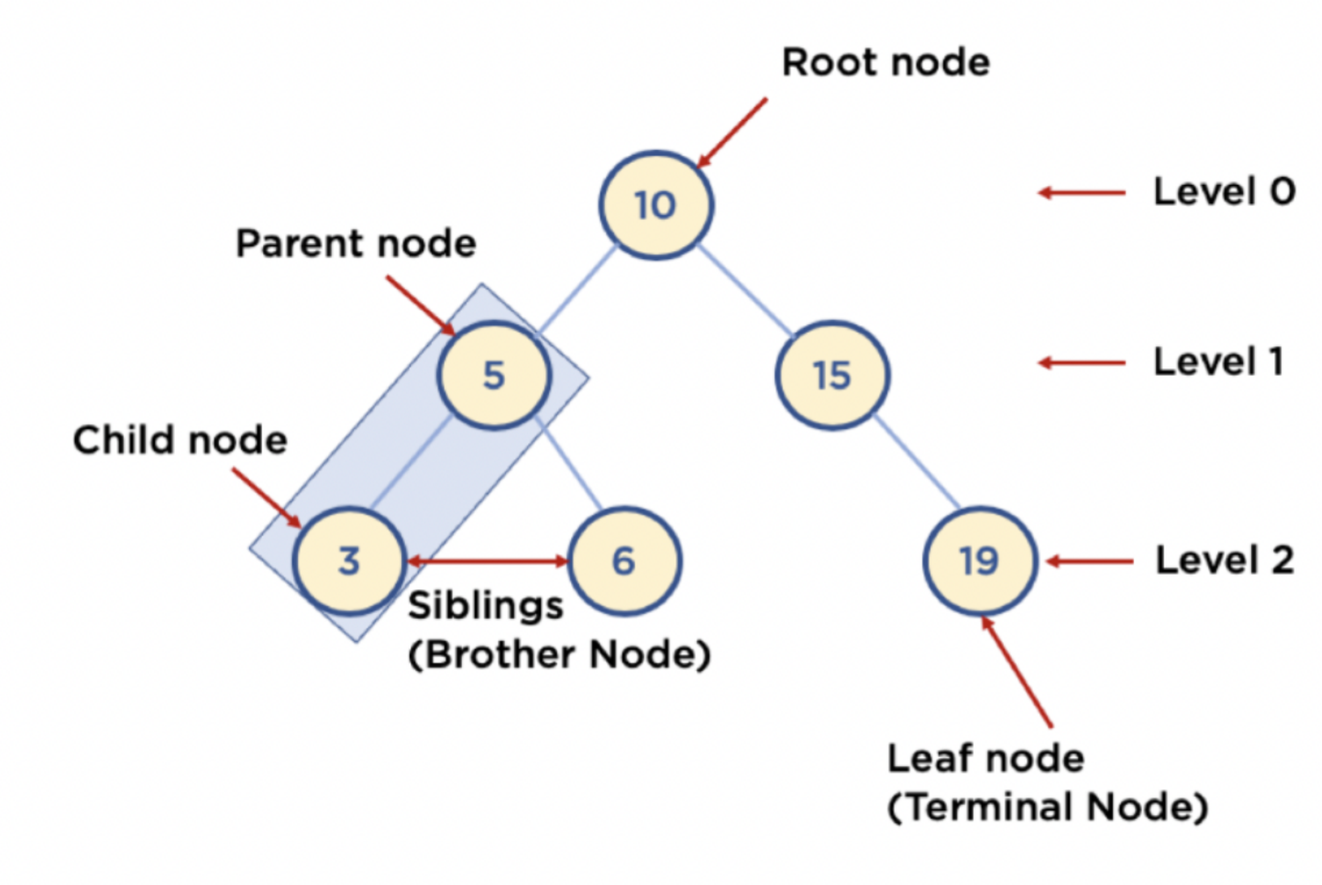
- 일련의 분류 규칙을 통해서 데이터를 분류, 회귀하는 지도 학습 모델
- 스무고개와 비슷
- 나무 구조에 기반하여 결정 진행
- 테스트: 결정 과정에서 던진 각 판정질문
- 결과:
  - 규칙의 집합
  - 최종결론 or 또다른 판정질문

목표: 일반화 성능이 뛰어난 트리 얻기 => 새로운 데이터를 잘 처리 할 수 있는 능력을 가진 의사결정 트리 얻기



기본 프로세스

- 루트노드
  - 한개뿐
  - 시작되는 마디 전체 데이터
- 내부노드
- 리프노드
  - 나무 구조 끝에 위치
- 부모 노드
- 자식 노드



기본 프로세스

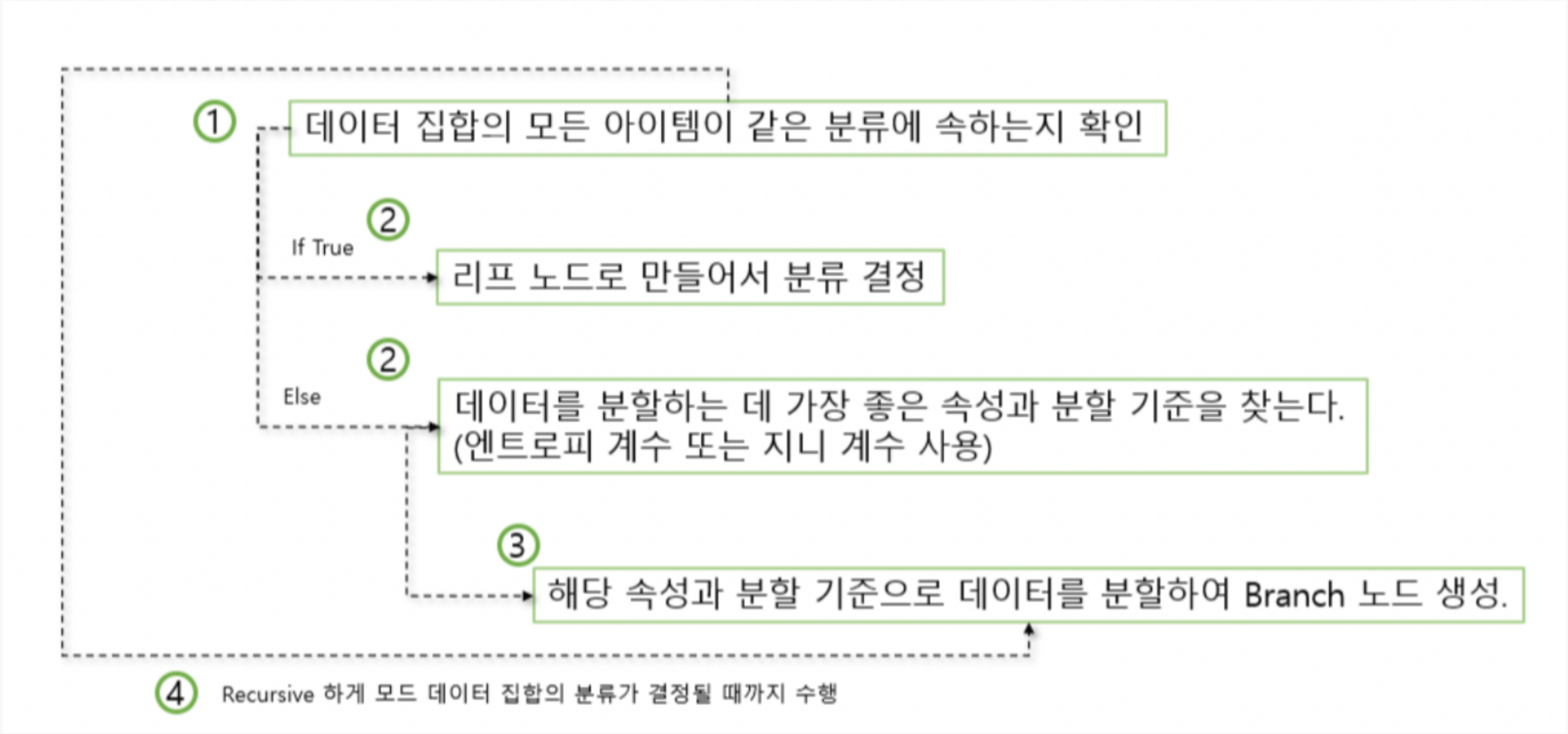
- 전략
  - 재귀적으로 간단
  - 직관적인 분할
- 재귀 과정이 일어나는 상황
  - 노드에 포함된 샘플들이 모두 같은 클래스에 속할때
  - 속성집합이 0 또는 같은 값일때
  - 샘플의 집합이 0

-> 분할을 진행 하지 않는다

```
# Pseudo Code for Decision Tree
input : training set D, Feature A
def TreeGenerate(D,A)
1: node 생성
2: if D의 샘플이 같은 클래스 C에 속하면:
3:       해당 node를 레이블이 C인 Terminal Node로 정함
4: if A = 0 or D의 샘플이 A 속성에 같은 값 취할 경우:
5:       해당 node를 Terminal Node로 정하고, 해당 클래스는 D 샘플 중 가장 많은 샘플의 수가 속한 속성으로 정함
6: A에서 최적의 분할 속성 a*를 선택
7: for a*의 각 값 a*[v]에 대해 다음:
        node에서 하나의 가지를 생성.
        D_v는 a*[v] 속성값을 가지는 샘플의 하위 집합으로 표기
        if D_v = 0:
            해당 가지 node를 Terminal Node로 정하고 해당 클래스는 D 샘플 중 가장 많은 클래스로 정함
        else:
            TreeGenerate(D_v, A\{a*})을 가지 노드로 정함

output : node를 Root Node로 하는 Decision Tree
```

기본 프로세스


①
데이터 집합의 모든 아이템이 같은 분류에 속하는지 확인
②
If True
리프 노드로 만들어서 분류 결정
②
Else
데이터를 분할하는 데 가장 좋은 속성과 분할 기준을 찾는다.
(엔트로피 계수 또는 지니 계수 사용)
③
해당 속성과 분할 기준으로 데이터를 분할하여 Branch 노드 생성.
④
Recursive 하게 모드 데이터 집합의 분류가 결정될 때까지 수행

2 분할 선택

- 최적의 분할 속성을 선택하는것
- 정보 엔트로피
  - 샘플 집합의 순도를 측정하는 지표

$$Ent(D) = -\sum_{k=1}^{|Y|} p_k \, log_2 p_k$$

    - D : 샘플집합
    - Pk: k번째 클래스 샘플이 차지하는 비율
  - Ent(D)의 비율이 작을수록 D 의 순도는 높아짐
    - 더 적은 질문

```
4: if A = 0 or D의 샘플이 A 속성에 같은 값 취할 경우:
5:         해당 node를 Terminal Node로 정하고, 해당 클래스는 D 샘플 중 가
6: A에서 최적의 분할 속성 a*를 선택
7: for a*의 각 값 a*[v]에 대해 다음:
           node에서 하나의 가지를 생성.
           D_v는 a*[v] 속성값을 가지는 샘플의 하위 집합으로 표기
           if D_v = 0:
```

2 분할 선택

- 정보이득
  - 샘플 집합 D에 대해서 속성 a가 분할을 통해 얻음
  - 분기 이전의 불순도와 분기 이후의 불순도 차이
  - 정복이득이 크면 속성 a를 사용해서 분할할때 얻을 수 있는 순도 상승
    - 정보 이득 기반으로 의사 결정 나무의 분할 속성 선택 가능
  - 취할 수 있는 값의 수가 비교적 많은 속성에 유리

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

정보 이득의 최대화 -> 불순도의 감소
-> 엔트로피의 감소

2 분할 선택

- 정보이득률
  - 취할 수 있는 값의 수가 비교적 적은 속성에 편향
  - 의사결정트리 방법 중 C4.5는 정보 이득률 사용
    - 휴리스틱한 방법을 사용하여 분할 속성 후보 중 정보 이득 높은 것 선택

$$Gain_Ratio(D,a) = \frac{Gain(D,a)}{IV(a)}$$

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}$$

2 분할 선택

- 지니계수
  - CART 의사결정나무에서 사용
  - 데이터 세트 D 에서 임의로 두 개의 샘플을 고르고 고른 두개가 다른 클래스일 확률
  - 최대값은 0.5
  - 작을수록 순도는 높음

$$Gini(D) = \sum_{k=1}^{|Y|} \sum_{k' \neq -k} p_k p_{k'}$$

$$= 1 - \sum_{k=1}^{|Y|} p_k^2$$

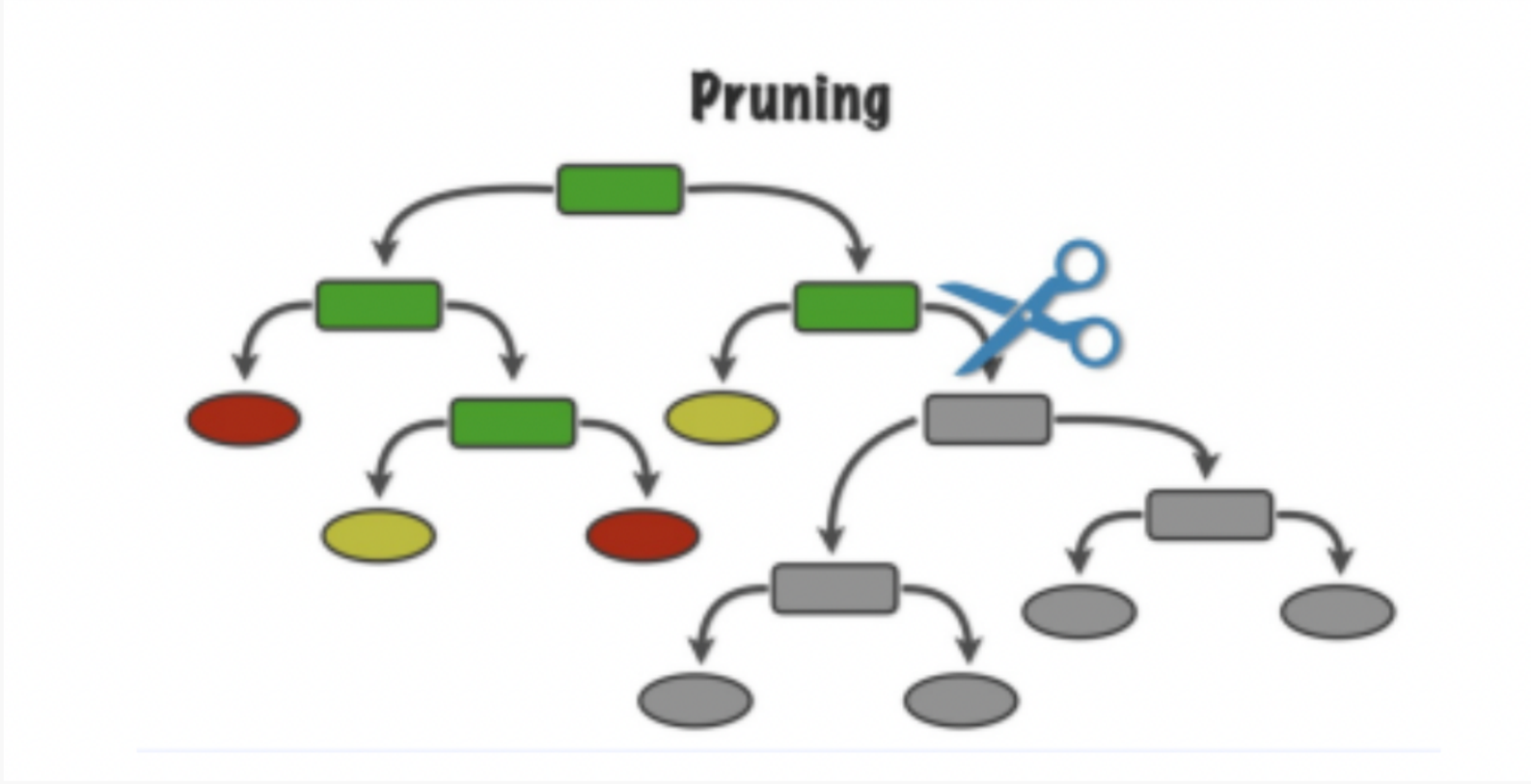
$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

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

$$\approx 0.47$$

3 가지 치기

- 과적합에 대응하기위한 주요 수단
  - 정확한 분류를 위해 노드의 분할 과정이 지속적으로 반복
    => 많은 가지 생성
- 적절한 수준에서 terminal node 결합



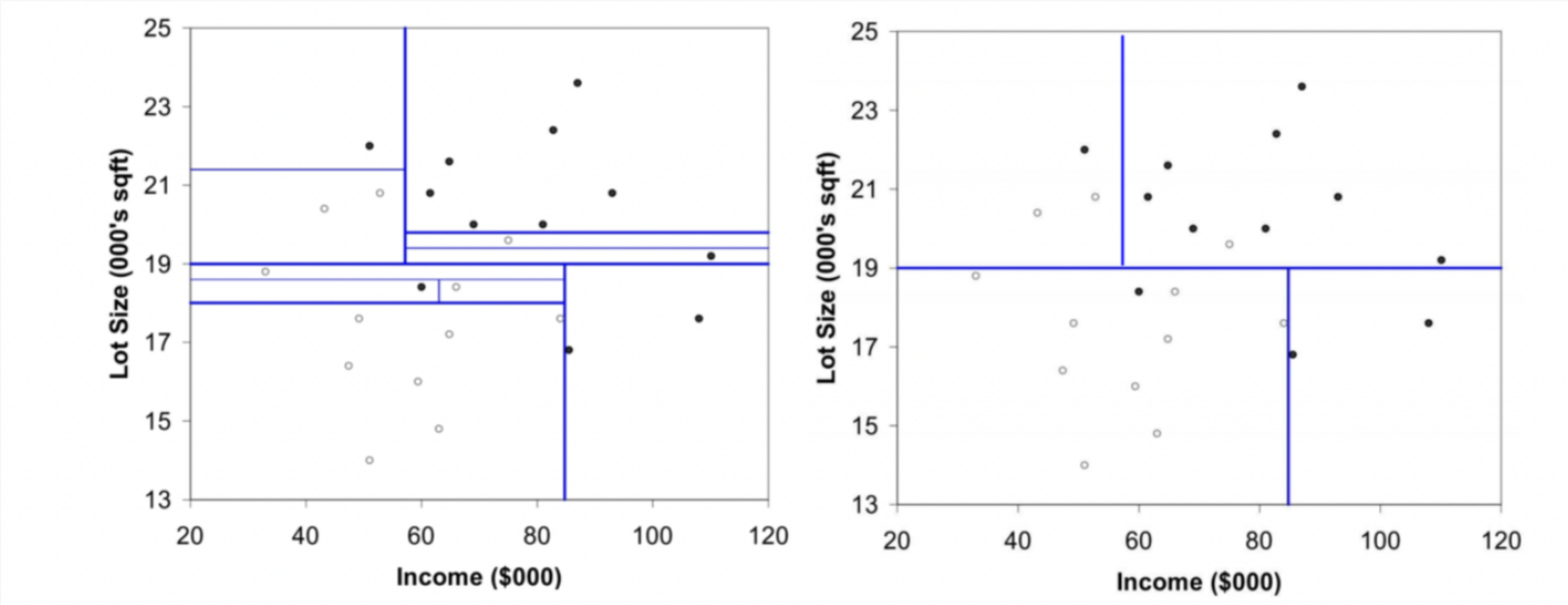
1. 사전 가지치기
2. 사후 가지치기

3 가지 치기

- 사전 가지치기
  - 의사결정 나무가 다 자라기 전에 알고리즘을 멈추는 방법
    - 연구자가 임의로 정한 숫자보다 인스턴스가 적어지면 멈춤
    - 임의로 불순도를 설정해 해당 지니계수 / 엔트로피에 도달하면 멈춤
  - 어떤 분할이 일반화 성능을 향상시키지 못하거나 일반화 성능을 낮추더라도 계속되는 분할을 통해 일반화 성능을 향상 시킬 가능성 차단
  - 과소 적합 위험을 높임

3 가지 치기

- 사후 가지치기
  - 풀트리 생성후 적절한 수준에서 터미널 노드를 결합하는것
  - 과소적합 위험이 낮음
  - 일반화 성능도 사전 가지치기에 비해 높음
  - 상향식으로 검토가 진행되어 훈련시간이 김



4 연속값과 결측값

- 연속값 처리
  - 지금까진 이산 속성에 기반한 의사결정 트리
  - 연속속성을 사용하는 방법
    - 이산화 작업 필요
      - 이분법 - 연속 속성 처리
        - 구간의 중암점을 후보 분할점으로 잡은뒤 이를 이산 속성값 처럼 여기고 샘플집합의 분할 진행

          -> 분할점으로 이분화된 정보 이득이 최대화는 분할점을 찾는것
  - 노드의 분할 속성이 연속 속성이라면 이는 이후 노드의 분할 속성이 될 수 있음

4 연속값과 결측값

- 결측값 처리
  - 해결해야 할 문제
    - 속성값이 결실된 상황에서 어떻게 분할 속성을 선택할 것인가?
    - 분할 속성을 정한 경우 샘플의 해당 속성값이 결측값이라면 샘플을 어떻게 분할 할것인가?
      - 하위 노드에 귀속
        - 분할속성에서 취한 값을 알고있을때
        - 분할속성에서 취한 값을 모를때

5 다변량 의사결정 트리

- 의사결정 트리가 만든 분류경계는 축에 평행하는 선으로 구성
  - 속성값에 상응하기때문에 높은 해석력
- 비터미널 노드가 어떠한 속성에 대한 테스트를 진행하는 것이 아닌 속성의 선형조합에 대해 테스트
  - 다변량 의사결정 트리는 단변량 의사결정 트리와 다르게 각 비터미널 노드를 위해 최적의 분할 속성을 찾는것이 아닌 적당한 선형 분류기를 만드는 것이 목적

표 4.5 ＼ 수박 데이터 세트 3.0α

번호	밀도	당도	잘 익은 수박
1	0.697	0.460	예
2	0.774	0.376	예
3	0.634	0.264	예
4	0.608	0.318	예
5	0.556	0.215	예
6	0.403	0.237	예
7	0.481	0.149	예
8	0.437	0.211	예
9	0.666	0.091	아니오
10	0.243	0.267	아니오
11	0.245	0.057	아니오
12	0.343	0.099	아니오
13	0.639	0.161	아니오
14	0.657	0.198	아니오
15	0.360	0.370	아니오
16	0.593	0.042	아니오
17	0.719	0.103	아니오

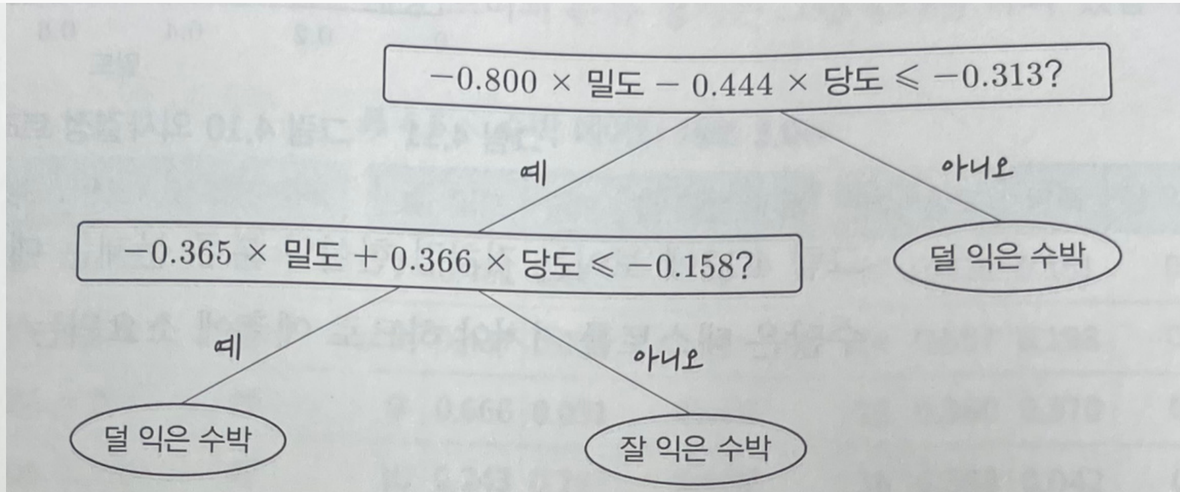


그림 4.13 ＼ 수박 데이터 세트 3.0α에서 생성된 다변량 의사결정 트리

5 다변량 의사결정 트리

- 의사결정 트리가 만든 분류경계는 축에 평행하는 선으로 구성
  - 속성값에 상응하기때문에 높은 해석력
- 비터미널 노드가 어떠한 속성에 대한 테스트를 진행하는 것이 아닌 속성의 선형조합에 대해 테스트
  - 다변량 의사결정 트리는 단변량 의사결정 트리와 다르게 각 비터미널 노드를 위해 최적의 분할 속성을 찾는것이 아닌 적당한 선형 분류기를 만드는 것이 목적

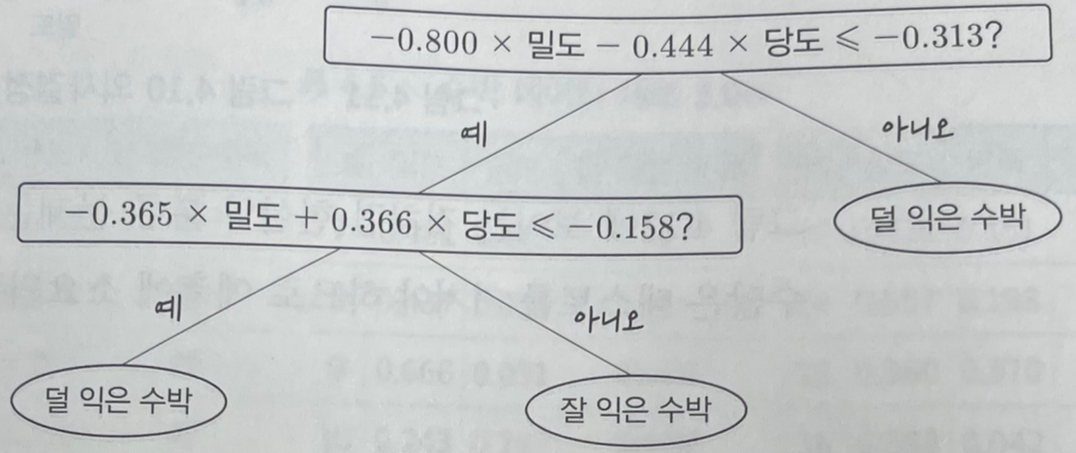


그림 4.13 ＼ 수박 데이터 세트 3.0α에서 생성된 다변량 의사결정 트리

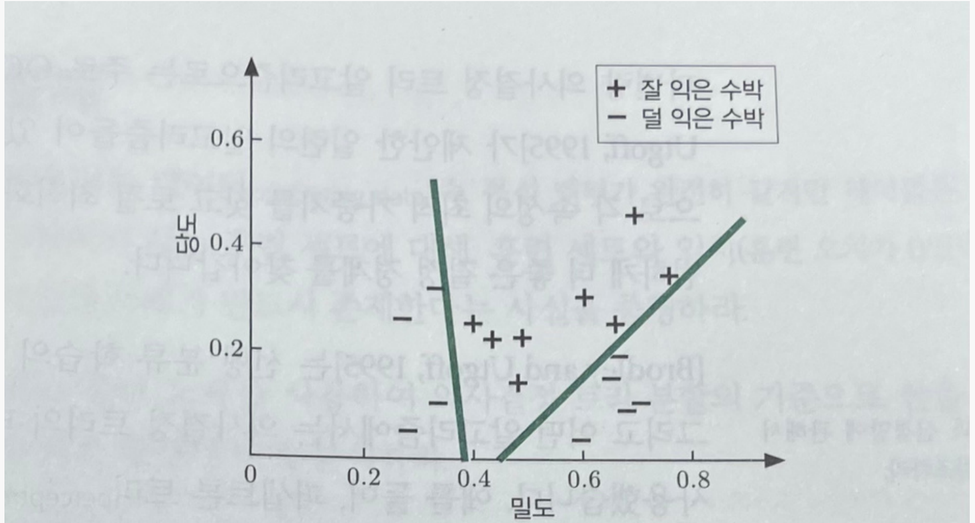


그림 4.14 ＼ 그림 4.13 다변량 의사결정 트리에서의 분류 경계