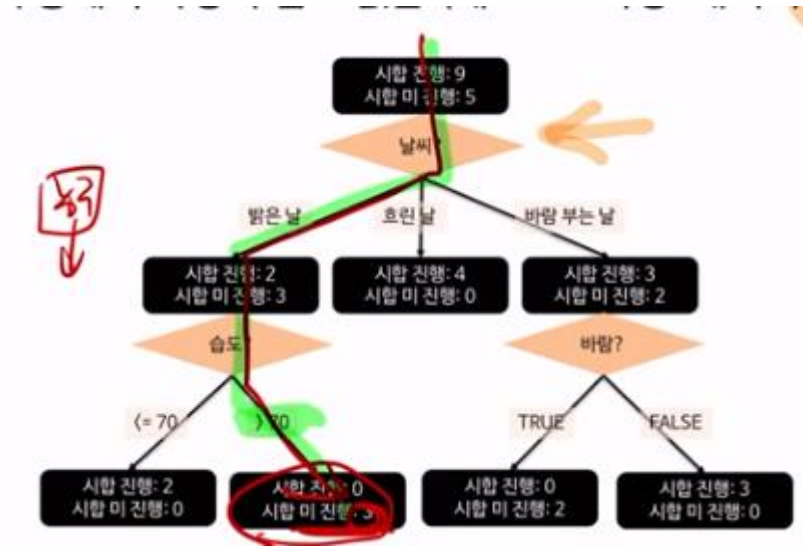
의사 결정 나무 (Decision Tree)

- 학습 데이터를 분석하여 데이터에 내재되어 있는 패턴을 통해 새롭게 관측된 데이터를 예측 및 분류하는 모델
- 개념적으로 질문을 던져서 대상(정답 후보)를 좁혀 나가는 스무고개 놀이와 비슷한 개념 (목적(Y)와 자료(X)에 따라 적절한 분리 기준과 정지 규칙을 지정하여 의사결정 나무를 생성)
- 의사결정방식 과정의 표현법이 '나무'와 같다고 해서 의사결정나무라고 불림 (Tree model)

의사결정나무의 장점

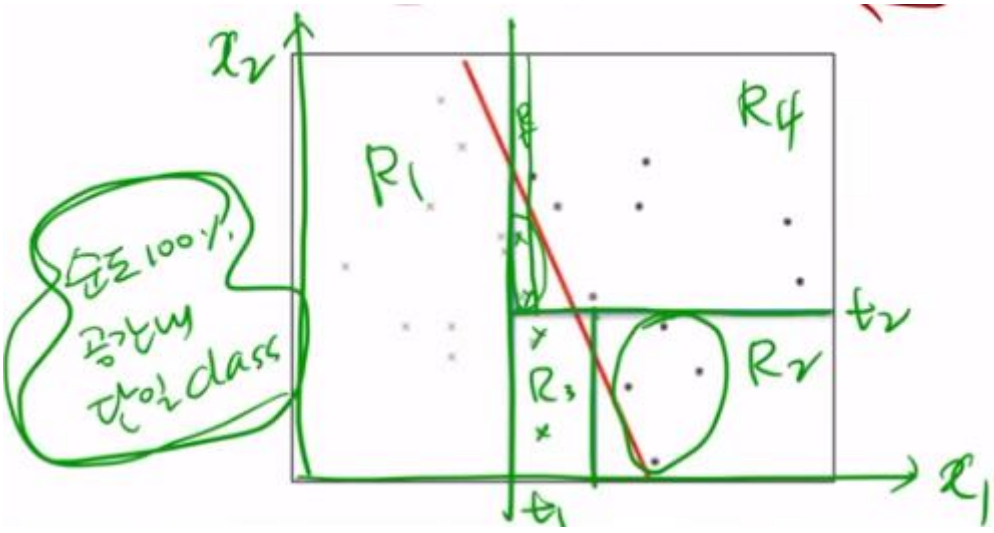
- 이해하기 쉽고 적용하기 쉬움 (나무 구조(if-then 규칙)에 의해 표현되기 때문에 모델을 쉽게 이해할 수 있음
- **의사결정과정에 대한 설명(해석) 가능** (오늘 야구 경기의 취소 사례의 이유 설명 가능 등)
- 중요한 변수 선택에 유용 (상단에서 사용된 설명 변수가 중요한 변수. ex 날씨)



- 데이터의 통계적 가정이 필요 없음 (ex LDA 가정 : 데이터 정규성)

의사결정나무의 단점

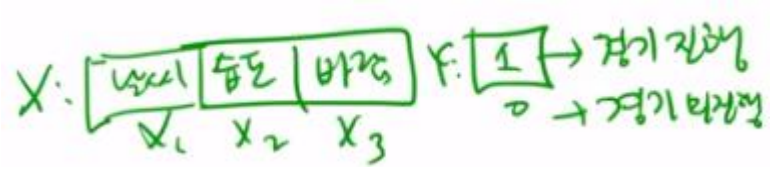
- 좋은 모형을 만들기 위해 많은 데이터가 팔요
- 모형을 만드는데 상대적으로 시간이 많이 소요 (Tree building)
- 데이터의 변화에 민감 (데이터에 따라 모델이 변화함)
  - ✧ 학습과 테스트 데이터의 도메인이 유사해야 함 (domain gap이 작아야 함)
- 선형 구조형 데이터 예측이 더 복합
  - ✧ 붉은 선: 선형회귀 결정 경계, 푸른 선: 의사결정나무 결정 경계



의사결정나무를 활용한 데이터 분석

순서 : 데이터 -> 모델 학습 -> 추론

- 데이터 : 다변량 변수 사용 (X가 많은)



- 모델 학습 (트리 구조 이용)
  - ✧ 한번에 설명 변수 하나씩 데이터를
  - ✧ 2개 혹은 그 이상의 부분집합으로
  - ✧ 데이터 순도가 균일해지도록 재귀적 분할. (종료 조건) :
    - ◆ 분류 : 끝 노드에 비슷한 범주 (클래스)를 갖고 있는 관측 데이터끼리 모아질 때
    - ◆ 예측 : 끝 노드에 비슷한 수치(연속된 값)을 가지고 있는 관측 데이터 끼리 모아질 때
- 추론 (판별)
  - ◆ 분류 : 끝 노드에서 가장 빈도가 높은 종속변수(y)를 새로운 데이터에 부여
  - ◆ 회귀 : 끝 노드의 종속변수(y)의 평균을 예측 값으로 반환

의사결정나무 카테고리

- 구분
  - ■ 분류 나무 (classification tree) : 목표 변수가 범주형 변수 (0, 1, 2) -> 분류
  - ■ 회귀 나무 (regression tree) : 목표 변수가 수치형 변수 -> 예측
- 재귀적 분할 알고리즘
  - ■ CART (Classification And Regression Tree)
  - ■ C4.5, C5.0
  - ■ CHAID (Chi-square Automatic Interaction Detection)

- 불순도 알고리즘 (재귀적 분할 알고리즘에서 쓰이는 분할 기준들)
  - ■ 지니 지수 (Gini index)
  - ■ 엔트로피 지수 (Entropy index), 정보 이익,률 (Information Gain ,Ratio)
  - ■ 카이제곱 통계량 (Chi-Square Statistic)

분류 나무 (Classification Tree)

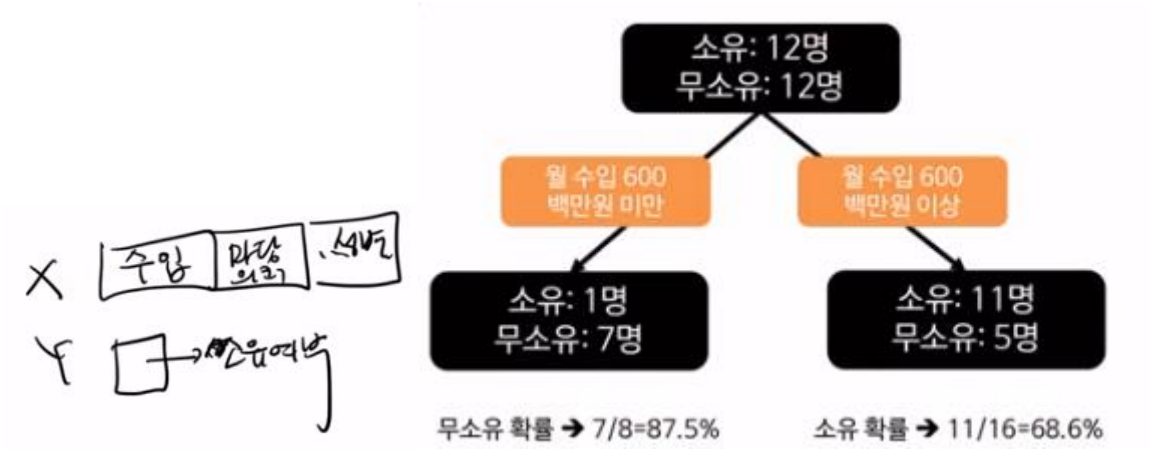
1. 목표 변수 : 범주형 변수 (분류)

✧ 분류 알고리즘과 불순도 지표 :

- ◆ CART : 지니 지수
- ◆ C4.5 : 엔트로피, 정보이익, 정보이익률
- ◆ CHAID : 카이 제곱 통계량

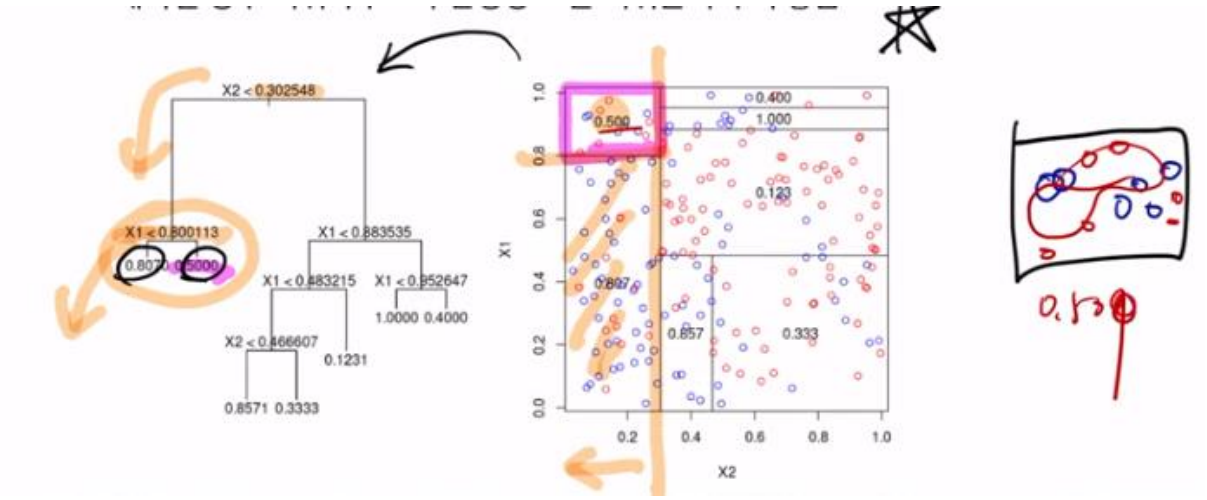
✧ 분류 결과 (판별, 추론)

- ◆ 소속 집단 판단, 경향성도 확률로 표현 가능



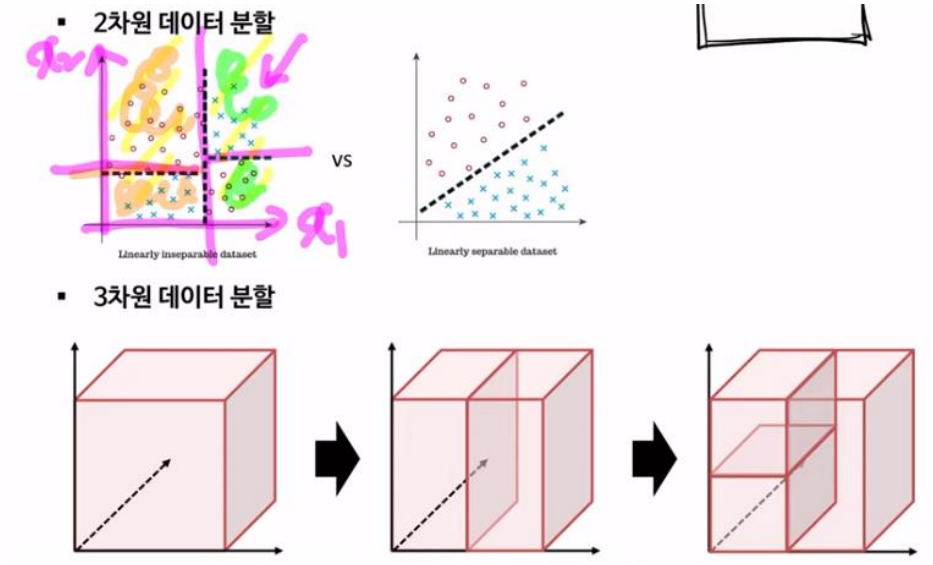
회귀 나무 (Regression Tree)

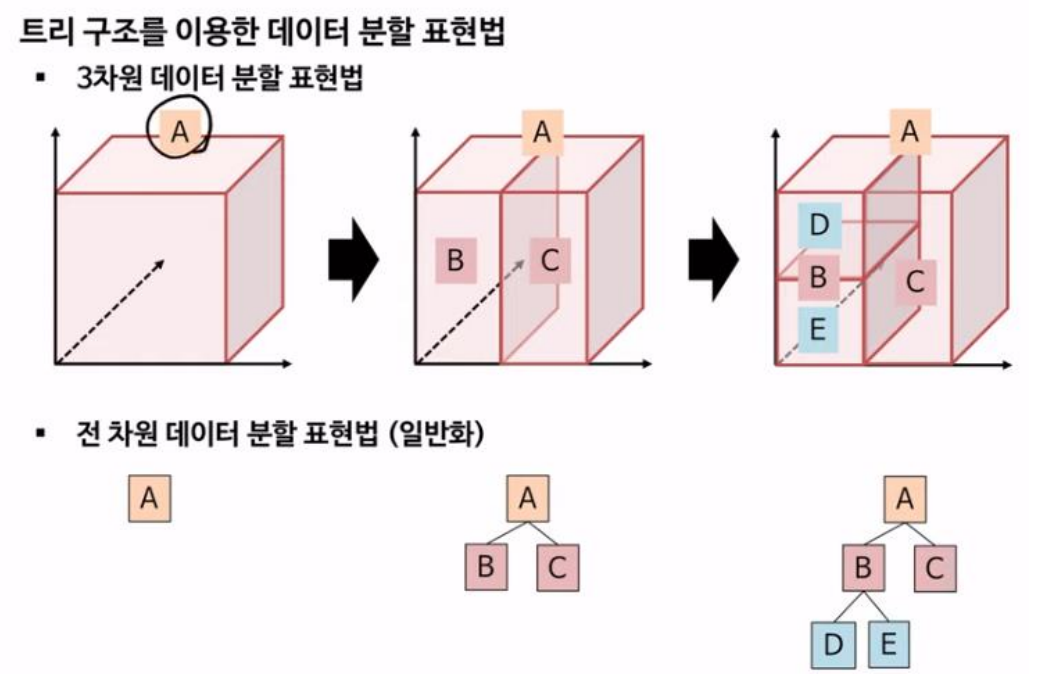
1. 목표 변수 : 수치형 변수 (예측, ex 키, 몸무게)
2. 회귀 알고리즘과 불순도 지표
   - ◆ CART : F 통계량과 분산 감소량 (실제 값과 예측 값의 평균 차이가 작도록!)
3. 회귀 결과
   - ◆ 끝 마디 집단의 평균
   - ◆ 예측일 경우 회귀 나무 보다 신경망 또는 회귀 분석이 더 좋다!!



분할

- 이진 분할 (binary split) : CART





- 다중 분할 (multi-way split) : CHAID, C4.5, C5.0 ...

재귀적 분할 알고리즘 정리

	CART	C4.5	CHAID
분류 나무(분류)	O	O	O
회귀 나무(예측)	O	O	X
예측변수(Y)	범주, 수치	범주, 수치	범주
불순도 알고리즘	Gini index	Entropy	Chi-square, 통계량
분리	Binary	Multi-way	Multi-way
나무성장	완전 모형 생성(full tree) 후 가지치기		최적 모형 개발 (즉, 완전모형생성 없음)
가지치기 (교차검증)	학습시 학습 데이터 검증시 검증 데이터	학습 데이터만 사용	X
개발연도	1984	1993	1980
	②	③	④

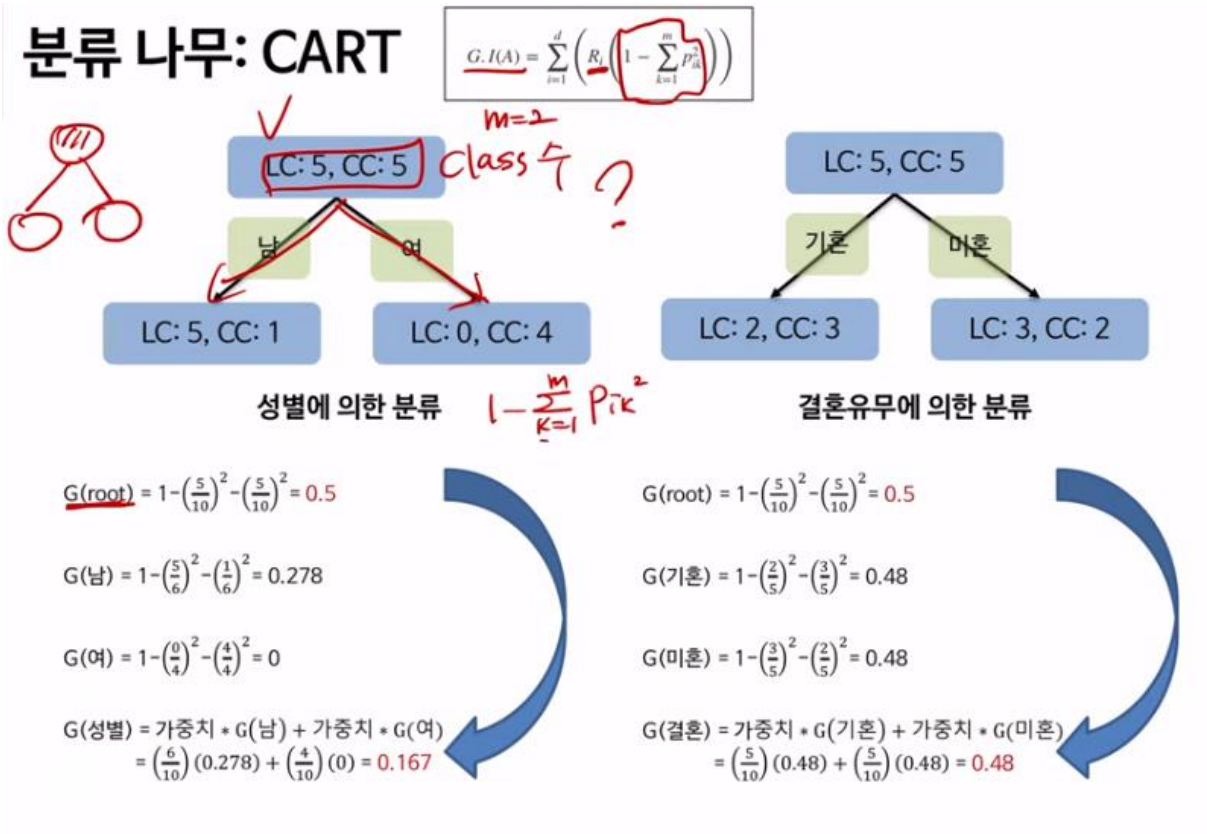
분류 나무

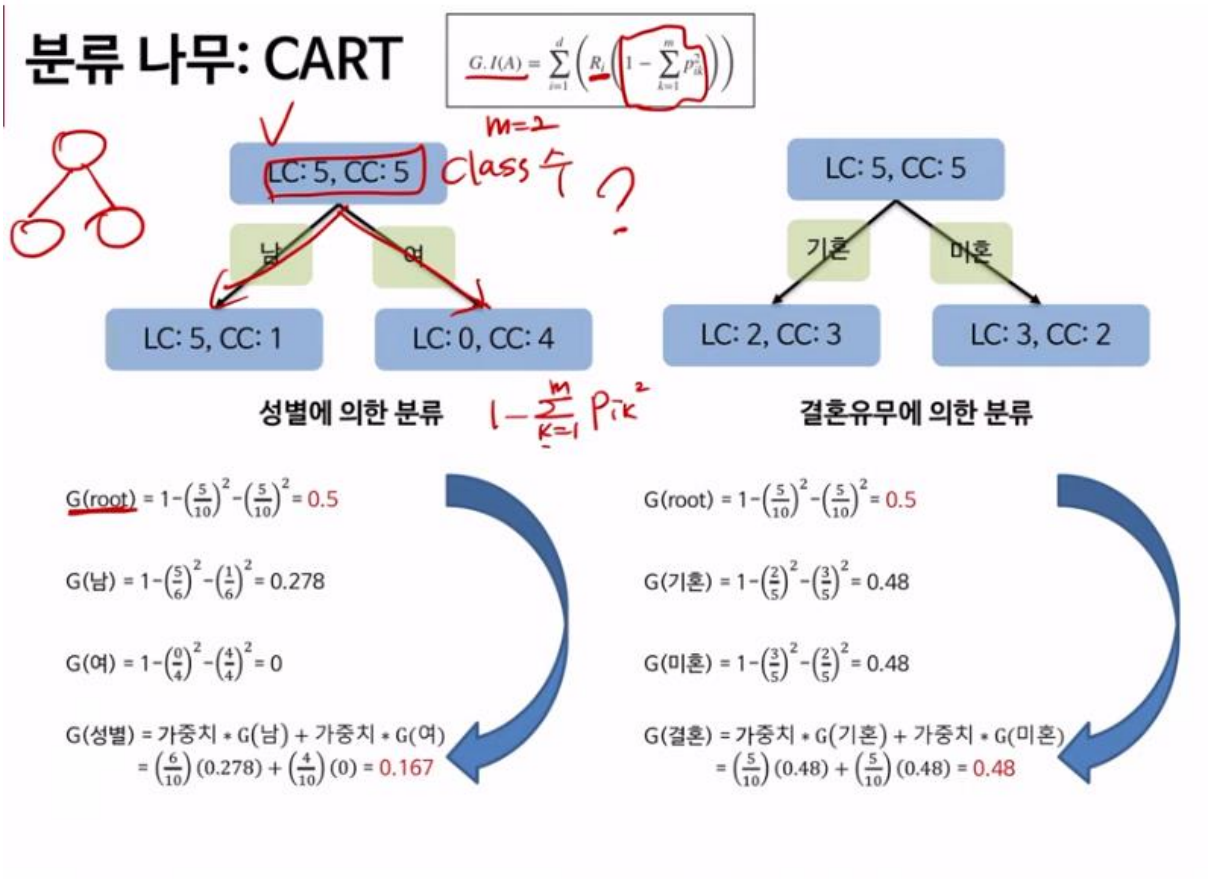
CART (Classification And Regression Tree)

- Breiman 등이 개발
- 종류 : 분류 나무, 회귀 나무
- 분리 : 이진 분할
- 가지치기 (교차 타당도) : 학습 데이터로 나무 생성, 검증용 데이터로 가지치기
- 불순도 알고리즘 : Gini index (불확실성)은 낮아 지는게 좋음

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

- 예제





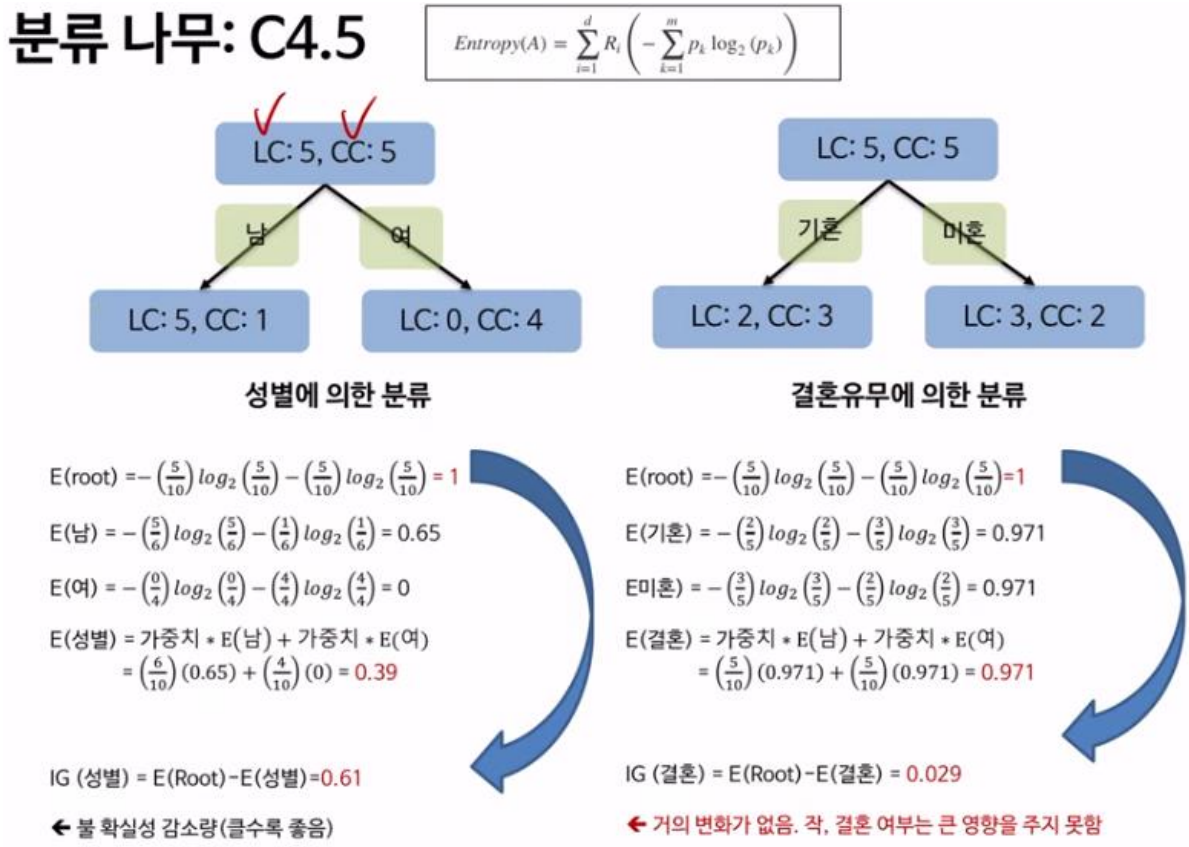
C4.5

- Quinlan 등이 개발
- 종류 : 분류 나무 , 회귀 나무
- 분리 : 다중 분할
- 불순도 알고리즘 : 엔트로피(불확실성), 정보이론, 정보이득율
- 정보 이론 -> 엔트로피

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

  - ✧ $\log_2$ 로 계산하는 이유 : bit 수로 정보 계산 ($\log_2(8) = 3bit$)
  - ✧ - $\log_2$로 계산하는 이유 : $\log_2(1/2) = -1$ 이기 때문에 +로 전환 필요 (확률 $p < 1$)
- 정보 이익 (IG : Information Gain) 정보의 가치 높아야 좋음

IG = E(before) – E(after)



정보이득율 (Information gain ratio)

- C4.5에서는 information gain -> information gain ratio 추가 도입
- 가지수가 많을수록 information gain이 높아지는 경향을 보인다.
- 단점 보안 위해 IV (Intrinsic Value) 도입하여 정보 이득율을 정규화 : 가지 많으면 감소

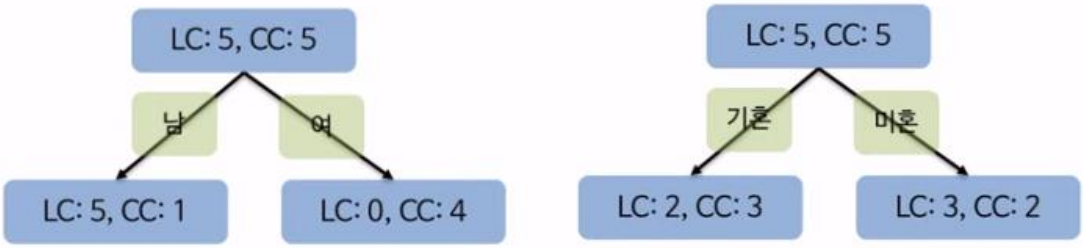
$$IV(A) = -\sum_{k=0}^{n} \frac{1}{n} log_2 \left(\frac{1}{n}\right)$$

- **이득율**

$$IGR(A) = \frac{IG(A)}{IV(A)}$$

분류 나무: C4.5

$$IV(A) = -\sum_{k=0}^{n} \frac{1}{n} log_2(\frac{1}{n})$$



성별에 의한 분류

결혼유무에 의한 분류

IG (성별) = E(Root)-E(성별)=0.61

$IV(성별) = -\left(\frac{6}{10}\right) log_2\left(\frac{6}{10}\right) - \left(\frac{4}{10}\right) log_2\left(\frac{4}{10}\right) = 0.97$

IGR(성별) = IG(성별)/IV(성별) = 0.61/0.97 = 0.63

IG (결혼) = E(Root)-E(결혼)= 0.029

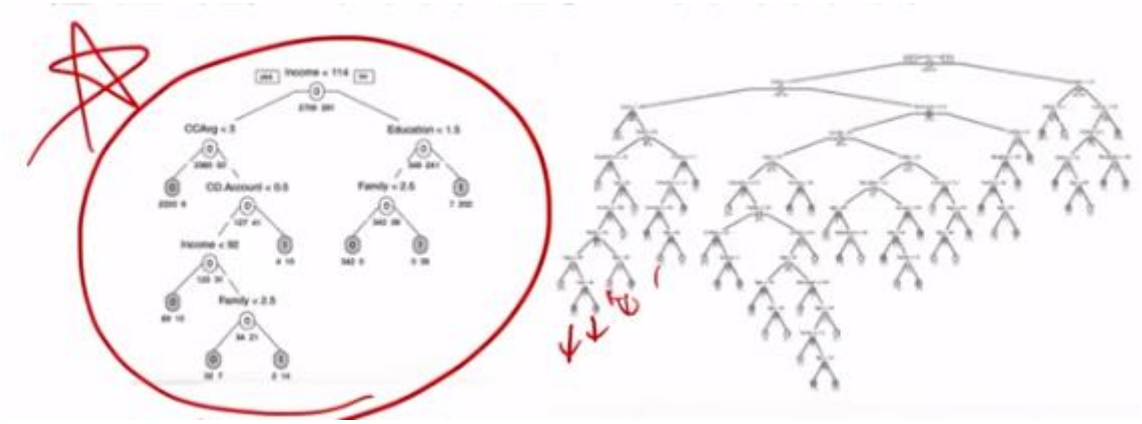
$IV(결혼) = -\left(\frac{5}{10}\right) log_2\left(\frac{5}{10}\right) - \left(\frac{5}{10}\right) log_2\left(\frac{5}{10}\right) = 1$

IGR(결혼) = IG(결혼)/IV(결혼) = 0.029/1 =0.029

**다중 분할 예시로 변경하면 - log2()이 계속 붙게 되고 IV가 1을 넘기도 함*

끝 없는 분할의 단점

- 과적합 (overfitting) : 학습용 데이터에 완전히 적합. 학습용 데이터에는 노이즈도 있기 때문에 테스트 데이터에서 오차는 일반적으로 증가.



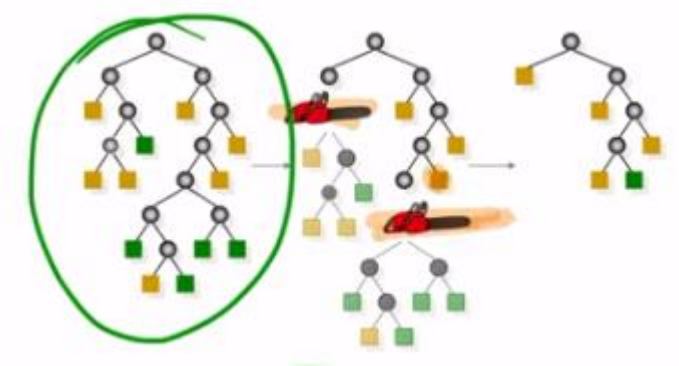
피하는 법 : 나무 성장 중단, 가지치기

모델 학습의 목적

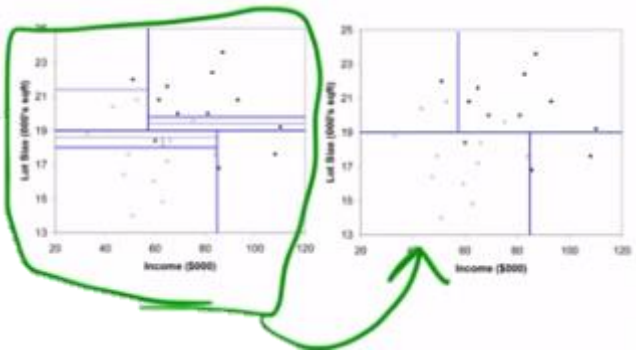
- 잘못된 학습 : 학습용 데이터에서는 높은 성과 -> 평가용 데이터에서는 낮은 성과
- 올바른 학습 : 현재 데이터의 설명 -> 미래데이터 예측

과적합 방지

- 성장 멈추기 (Strop condition)
  - ✧ 나무 모델의 깊이 파라미터로 설정 (depth 설정)
  - ✧ 나무 모델을 성장시키면서 특정 조건에서 성장을 중단
  - ✧ 노드 내의 최소 관측치의 수 설정
  - ✧ 불순도 최소 감소량 설정
    - ◆ CHAID에서 사용
    - ◆ 가지치기 사용하지 않고 종료

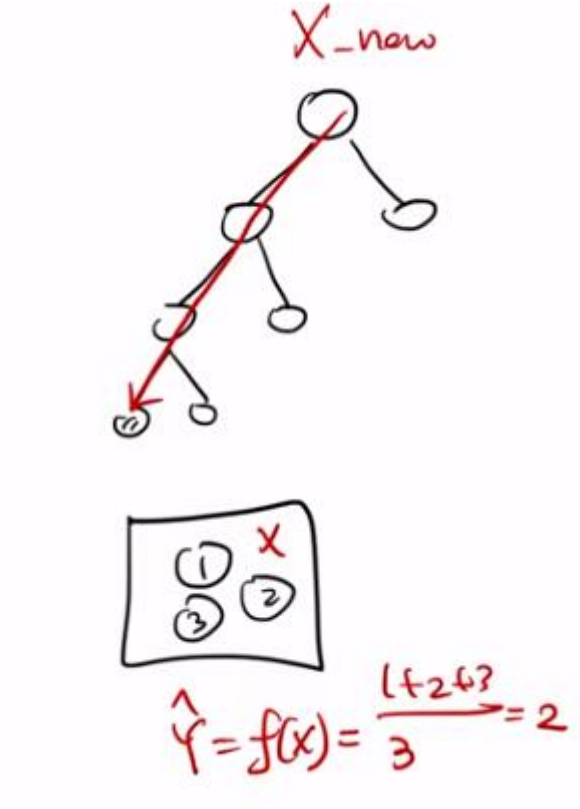


- 가지치기 (pruning) : 다 해보고 결정
  - ✧ 완전 모형 생성 후 가지치기
  - ✧ 데이터 버리는 개념이 아니라 합치는(merge) 개념
  - ✧ 나무 모델 생성 후 필요 없는 가지 제거
  - ✧ 성장 멈추기 보다 성능 우수
  - ✧ 가지치기 비용함수를 최소로 하는 분기를 찾음



회귀 나무

- 입력 데이터(변수 값)의 결과 예측 : 데이터가 도달한 끝 노드 데이터들의 평균으로 결정



- 불순도 측정 방법 :
  - ✧ 제곱 오차 합 (the sum of the squared errors)
  - ✧ 오차 = 실제 값 – 예측 값
  - ✧ $$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
- 성능 평가 방법 : RMSE (root mean squared error)

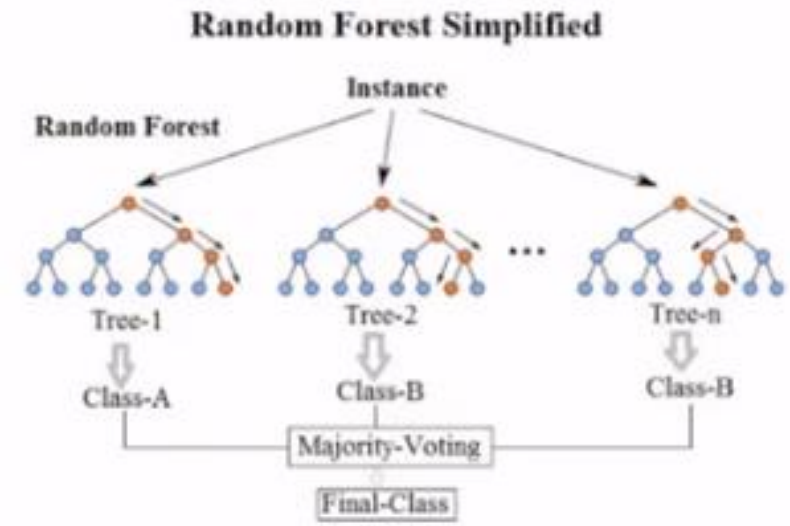
앙상블(Ensemble)

- 여러 모델을 함께 사용하자!
  - ✧ 의사결정나무, KNN, LDA, 로지스틱 등
  - ✧ 퀴즈쇼에서 사용되는 힌트 : 100인의 정답

✧ 설명보다는 예측이 중요할 경우에 사용

✧ 예측 알고리즘을 조합하여 예측 성능을 향상

✧ 랜덤 숲(Random Forest), Boosted Trees : 좋은 의사결정나무를 모아서 숲을 만들자!



- Random Forest

  ✧ Bootstrap 사용 : 데이터에서 여러 개의 샘플을 뽑아서 각각의 나무를 만들고 거기서 몇 개 뽑고 다시 샘플링 후 나무들을 만드는 방식

  ✧ Forest 생성 : 무작위로 예측 변수(날씨, 습도, 바람 등)를 선택하여 모델 구축 (의사결정 나무는 예측 변수 선택 시 기준 지표(지니, 엔트로피, 정보 이득 등)를 사용하였으나 무작위 숲에서는 무작위로 선택함) -> 과적합을 막으면서 예측을 잘 하는 성능이 나오면서 많이 쓰인다. 그러나 Tree에서 Forest가 되면서 해석 가능하다는 장점은 사라졌음. 그래도 결과 분석을 통해 설명 변수 중 중요한 변수를 판별 할 수 있다.

요약

- 주요 방법

  ✧ Trees and Rules 구조

    ◆ 규칙은 나무 모델로 표현

    ◆ 결과는 규칙으로 표현

  ✧ 재귀적 분할 (Recursive Partitioning)

    ◆ 의사결정나무 생성 과정

- ◆ 그룹이 최대한 동질 하도록 반복적으로 하위 그룹으로 분리

✧ 가지치기 (Pruning the trees)

- ◆ 생성된 나무를 자르는 과정 (정교화)
- ◆ 과적합을 피하기 위해 필요 없는 가지를 정리하는 과정이다.(Merge)

✧ 구분

- ◆ 분류 나무 : 목표 변수가 범주형 변수
- ◆ 회귀 나무 : 목표 변수가 수치형 변수

- 재귀적 분할 알고리즘
  - ✧ CART
  - ✧ C4.5
  - ✧ CHAID
- 불순도 알고리즘
  - ✧ 지니 지수
  - ✧ 엔트로피 지수
  - ✧ 카이 제곱 통계량
- 의사 결정 나무 과정
  - ✧ 나무 모델 생성 -> 과적합 문제 해결 -> 검증 -> 해석 및 예측
    - ◆ 생성 : CART(Gini), C4.5(Entropy), CHAID(Chi-Square)
    - ◆ 과적합문제 : 완전나무모형생성 후 가지치기 (CART, C4.5)
    - ◆ 검증 : 교차 타당성을 이용하여 의사결정나무 평가
    - ◆ 해석 및 예측 : 의사결정나무를 해석하고 예측 모형 설정

      분류 : 끝 노드에 가장 많은 클래스

      회귀 : 끝 노드에 있는 데이터들의 평균값
- 앙상블 : Random Forest

실습