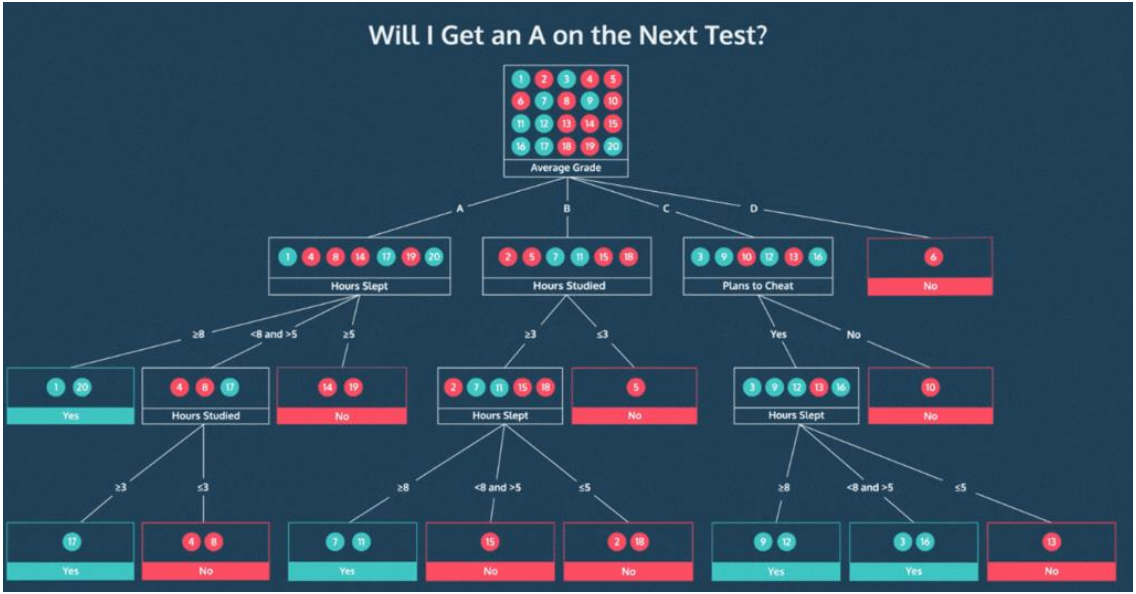
BAE Lab Bigdata Analytics & Engineering 부산대학교 PUSAN NATIONAL UNIVERSITY

Decision Tree (의사 결정 나무)

Prof. Hyerim Bae
Department of Industrial Engineering, Pusan National University
hrbae@pusan.ac.kr

수업 성적



출처: https://hleecaster.com/ml-decision-tree-concept/

Contents

산업현장에서 수집된 데이터를 분석하는데 필요한 기초 소양을 강의합니다.

01 Tree for Classification

02 Measure for purity

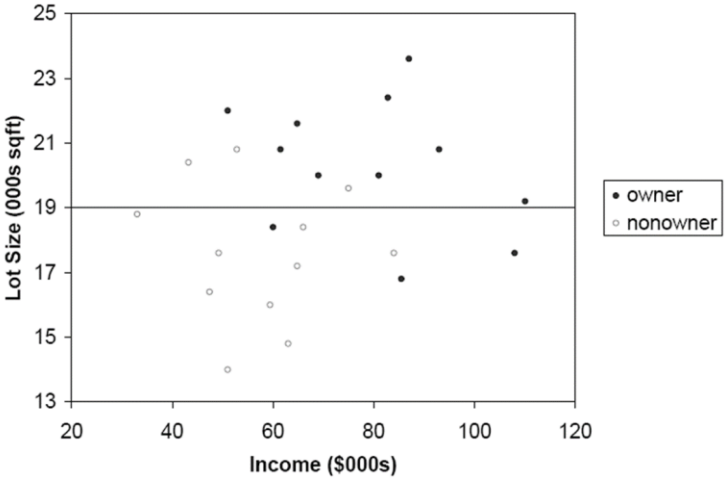
03 Decision Tree

의사결정나무

- Decision Tree

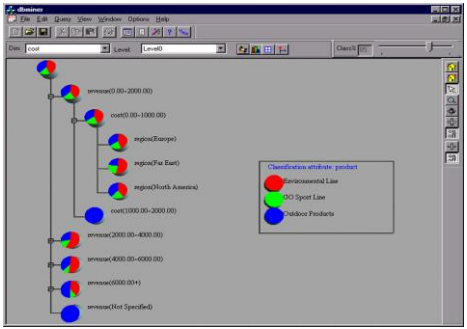
분류 문제란 무엇인가?

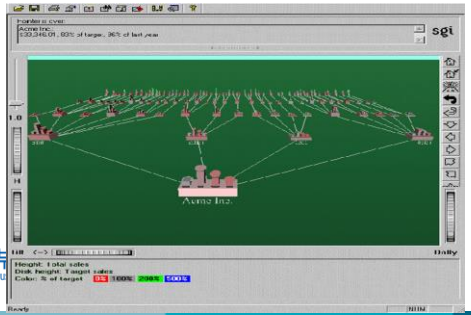
- 목표: 예측 변수 집합을 기반으로 결과를 분류
- 출력: 규칙들의 집합
- 예시)
  - 목표: 레코드를 '신용 카드 제안을 수락할 것' 또는 '수락하지 않을 것'으로 분류
  - 규칙: 'IF (Income 〉 92.5) AND (Education 〈 1.5) AND (Family 〈= 2.5) THEN Class = 0(수락하지 않음)



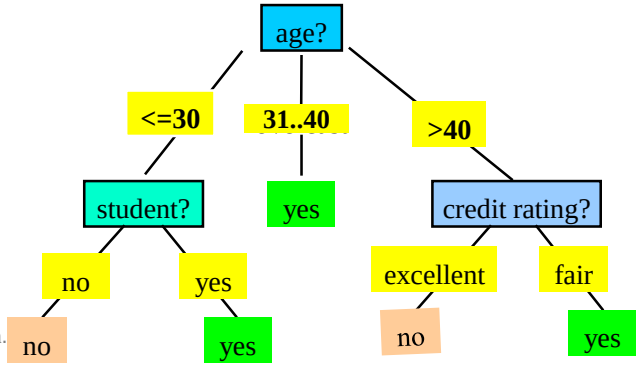
예시

- 계층적 분류(Hierarchical classification)
  - 재귀적 분기(Recursive partitioning)
  - 가지치기(Pruning the tree)





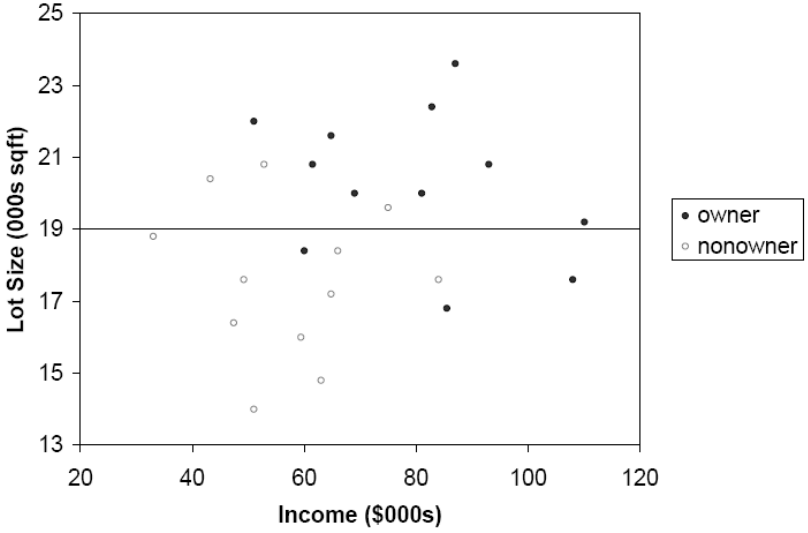
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

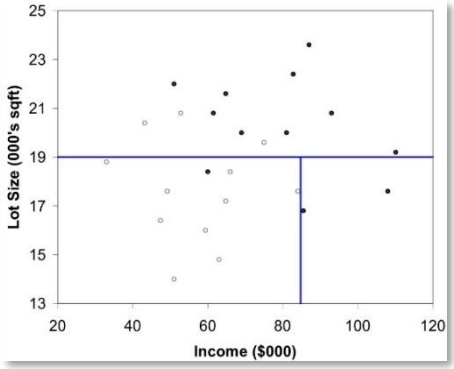


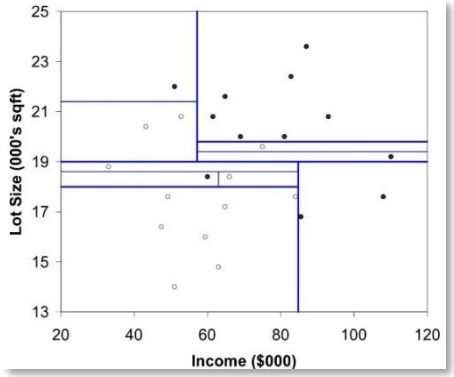
재귀적 분기(Recursive partitioning)

- 예측 변수 중 하나인 x_i 값을 선정
- 학습용 데이터를 두 부분으로 나누는 x_i값 선택
- 각 결과 부분이 순수한 정도('pure' or homogeneous) 측정
  - pure: 대부분의 데이터가 하나의 클래스를 포함하는 정도
- 알고리즘은 초기 분할에서 순도를 최대화 하기 위해 각자 다른 x_i을 시도함
- 최대 순도를 분할 한 뒤, 두번째 분할에 대해 프로세스 반복 진행
- **분할 알고리즘의 중지 조건**
  - 노드의 모든 샘플이 동일한 클래스에 속하는 경우
  - 추가로 분할 할 변수가 없는 경우
  - 남는 샘플이 없는 경우

첫번째 분할


Lot Size (000s sqft)
Income ($000s)
owner
nonowner

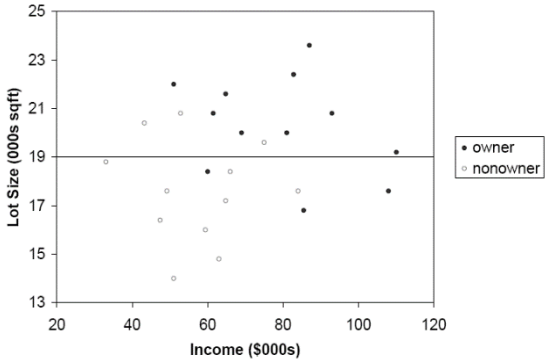

Lot Size (000's sqft)
Income ($000)


Lot Size (000's sqft)
Income ($000)

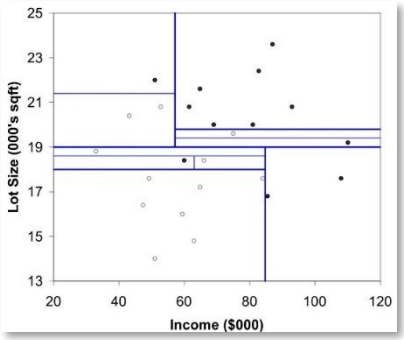
불순물 측정

- 데이터가 얼마나 순수한가?

Before split



After split



지니 지수

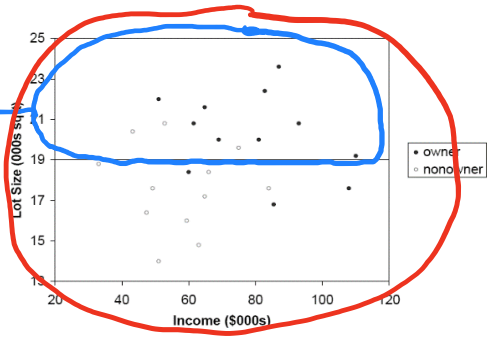
- p = 직사각형 A에서 클래스 k에 속하는 사례의 비율

m개 종류의 클래스를 포함하는 직사각형 A의 지니 지수

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

  - 모든 Case가 동일한 클래스에 속할 때의 지니 지수 I(A) = 0
  - 모든 클래스들이 동일한 경우, 최대 값을 가짐
    - 이진(Binary) 문제인 경우 = 0.5

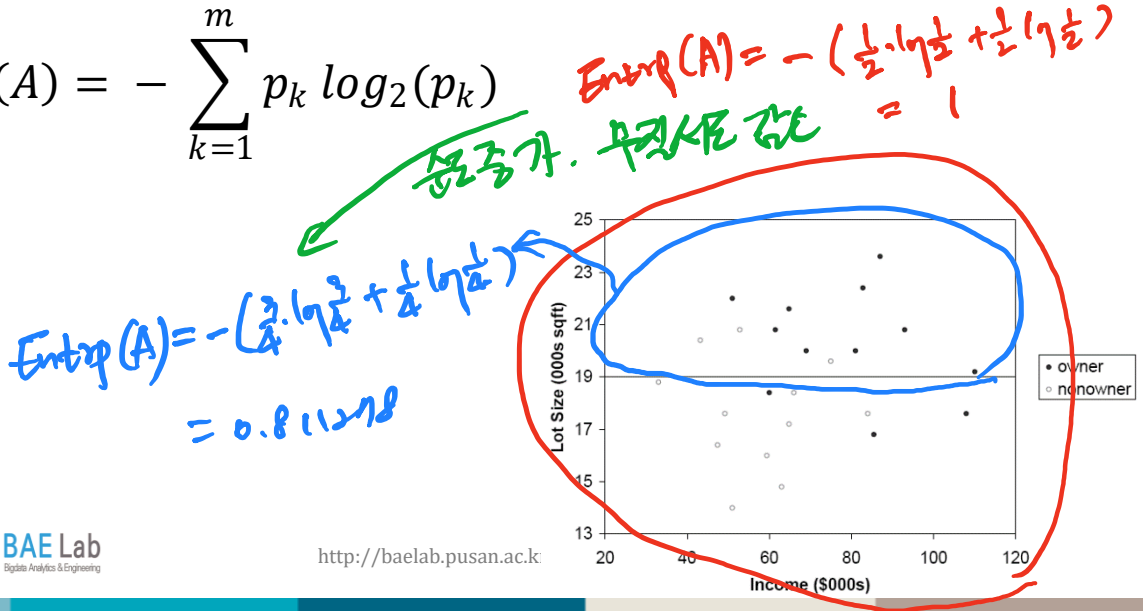
- 참고: XLMiner는 '델타 변형 규칙'이라는 변형 사용



엔트로피

- p = 직사각형 A에서 클래스 k에 속하는 사례의 비율(클래스 m 중에서)
- 엔트로피는 0에서 $log_2 m$ 사이의 범위를 가짐

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



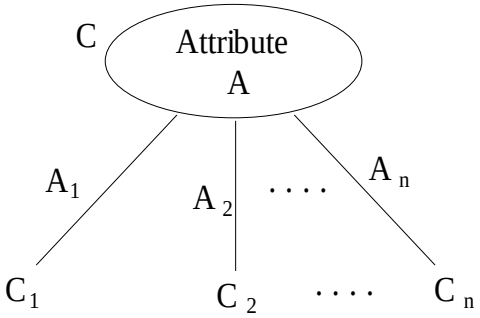
데이터로 트리 생성

	Height	Hair	Eyes	Class
1	short	blond	blue	+
2	tall	blond	brown	-
3	tall	red	blue	+
4	short	dark	blue	-
5	tall	dark	blue	-
6	tall	blond	blue	+
7	tall	dark	brown	-
8	short	blond	brown	-

- 문제
- Given:
  - 높이: {short, tall}, 머리: {blond, dark, red}, 눈: {blue, brown}, Class: {+, −}
- Find:
  - Given 문제를 올바르게 분류하는 최소 크기의 의사 결정 트리
  - (Attributes로 레이블이 지정된 노드)
  - (Values로 레이블이 지정된 아크(Arc))
  - (단일 클래스로 레이블이 지정된 잎(Leaf))

- 다음을 정의함

$B(C, A) = [P\{value(A) = A_i\} \times M(C_c)]$



- B(C, A)란 C를 A로 분류한 후에도 여전히 필요한 정보를 의미함
  - 즉, 새로운 정보에 대한 기댓값을 의미함

- 가장 많은 정보를 얻는 속성 계산

$MAXIMIZE\ M(C) - B(C, A)$

- C에는 3가지 + 클래스와 5가지 – 클래스가 존재함
  - 즉, $M(C) = -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} = .954$
- 첫 번째 변수인, 'Height'에 대해서 시도해본다면:

i = tall

tall blond brown -	short blond blue +
tall red blue +	short dark blue -
tall dark blue -	short blond brown -
tall blond blue +	
tall dark brown -	

j = short

$$M(c_i) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = .971$$

$$M(c_j) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = .918$$

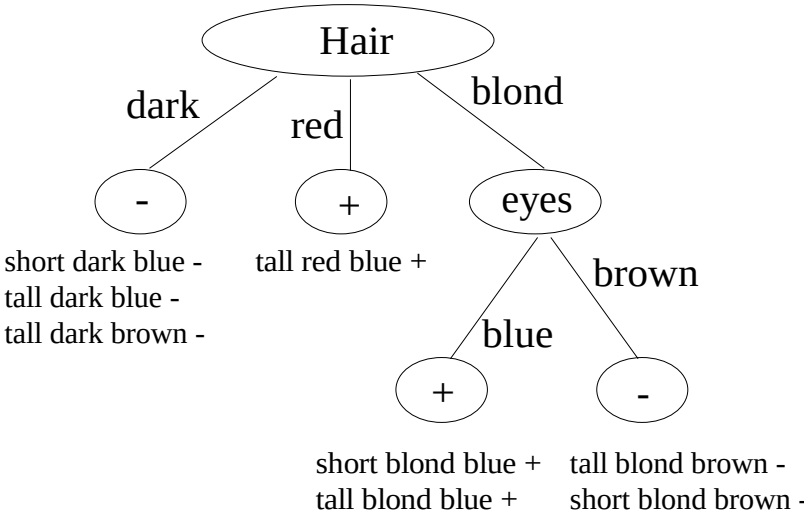
$$B(C, \text{'height'}) = \frac{5}{8} \times .971 + \frac{3}{8} \times .918 = .951$$

$$M(C) - B(C, \text{'height'}) = .954 - .951 = .003$$

- 다음으로, 'hair'에 대해 테스트:
  - $B(C, \text{'hair'}) = \left(\frac{3}{8}\right) \times 0 + \left(\frac{1}{8}\right) \times 0 + \left(\frac{4}{8}\right) \times 1 = .5$
  - $M(C) - B(C, \text{'hair'}) = .954 - .5 = .454$
- 마지막으로, 'eyes'에 대해 테스트:
  - $M(C) - B(C, \text{'eyes'}) = .347$

- 따라서 가장 먼저 테스트 할 속성은 'hair'임.
  - 'hair'의 3가지 중 2가지 클래스는 더이상의 정보가 필요하지 않음(dark, red의 경우).
  - 'hair'의 blond 경우, 더 분할해야 하므로, 해당 노드에서 'height'와 'eyes'를 테스트함.

- 최소 결정 트리



이 트리는 가장 작으며, 테스트 노드가 두 개 밖에 존재하지 않음

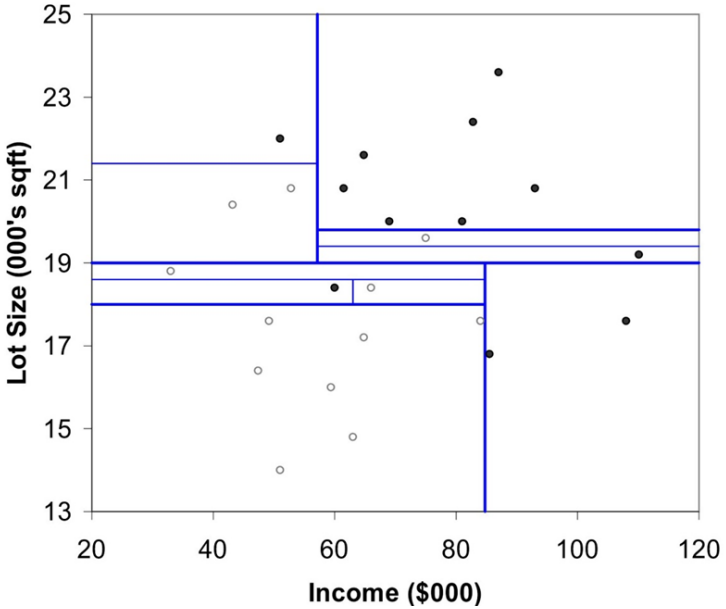
또한, '높이'의 속성은 테스트 되지 않음

리프(Leaf) 노드 레이블 결정

- 리프 노드의 라벨은 Voting(투표) 혹은 Cutoff 값으로 결정됨
- 각 리프 노드 데이터는 학습용 데이터에서 얻은 것임
- 대다수의 리프 노드 라벨은 기본 값으로 $cutoff = 0.5$를 선택함
  - 기본 값 $cutoff = 0.5$의 의미는 리프 노드의 레이블이 더 많은 클래스를 가지는 레이블로 정한다는 것
- $cutoff = 0.75$
  - $cutoff = 0.75$의 의미는 75% 이상의 '1'이라는 레이블이 있어야 리프 노드의 레이블을 '1'로 지정할 수 있다는 것

트리 과적합

- 다음 모델은 좋은 분류 모델인가?
  - '순도(purity)' 관점
  - 학습용 데이터의 경우
  - 새로운 데이터에 대해서는 어떤가?



문제 해결

- 과적합: 유도된 트리가 학습 데이터에 과적합 할 수 있음
  - 의사결정나무의 나뭇가지가 너무 많으면 노이즈 혹은 이상치(이상 징후)를 반영하는 경우도 존재함
  - Unseen 데이터에 대한 정확도가 낮음

- 과적합을 방지하기 위한 두가지 접근 방식
  - Prepruning: 트리 구성을 조기에 중지하는 방법. Goodness 측정값이 임계값보다 낮을 경우, 노드를 분할하지 않는 방법
    - 적절한 임계값을 찾기 힘듬
  - Postpruning: 완성된 트리에서 가지를 제거하는 방법. 순차적으로 가지치기 된 트리를 얻을 수 있음
    - 학습 데이터와 다른 데이터 집합을 사용하여 '가장 잘 분할된 트리'를 결정함

CHAID

- 조기 종료
- CART보다 오래된 CHAID는 카이-제곱 통계 검정을 사용하여 트리 성장을 제한함
- '순도(purity)'의 개선이 통계적으로 유의하지 않은 경우 트리의 분할을 중단함

각 노드에서 반응변수와 가장 큰 연관성을 보이는 예측변수로 분할하는데, 이 연관성의 강도를 chi-square검증으로 찾아내고, 가장 연관성이 큰 예측변수가 통계적으로 유의한 개선결과를 보이지 못하면 중단

가지치기

- CART는 나무를 완전히 성장시킨 후, 가지치기를 진행
  - 아이디어: 검증 오차가 상승하기 시작하는 지점을 찾는 것
  - Leaf들을 가지치기하여 작은 나무를 여러 개 생성함
  - 각 가지치기 단계에서 여러 종류의 트리가 구성될 수 있음

- Cost complexity를 사용하여 가장 좋은 나무를 선택

  CC(T) = 트리의 cost complexity

  Err(T) = 잘못 분류된 데이터의 비율

  L(T) = Leaf 노드의 수

  a = 트리의 크기에 추가되는 패널티 요소(분석가가 설정)

  $$CC(T) = Err(T) + a\,L(T)$$

  - 주어진 크기의 나무 중, CC가 가장 낮은 나무를 선택
  - 트리 각각의 크기에 대해 수행

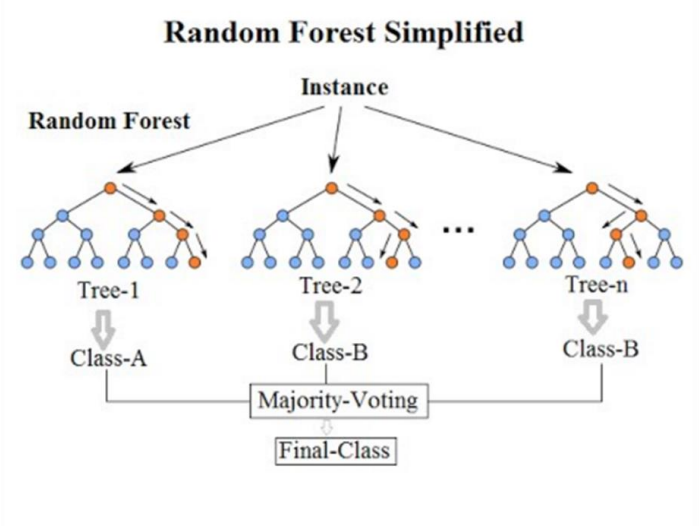
- 가지치기 프로세스는 서로 다른 크기 및 오류율을 가진 트리의 집합을 생성함
  - 최소 오류 트리
    - 검증 데이터에서 오류율이 가장 낮은 트리
  - 최상의 가지치기
    - 최소 오차를 가지며 그 표준 편차가 가장 작은 트리
    - 이 과정에서는 simplicity 및 parsimony를 위한 보너스가 추가됨

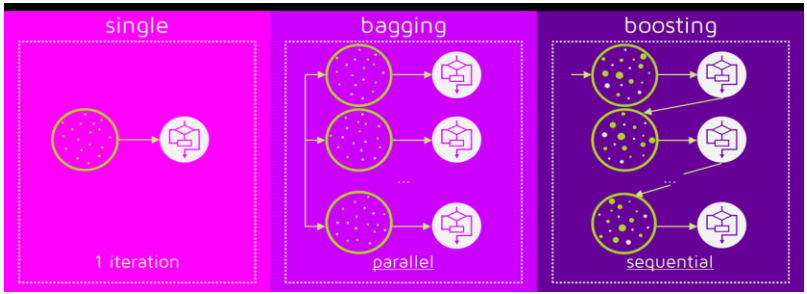
예측을 위한 회귀 트리

- 결과 변수가 연속형일때 사용
- 분류 트리와 유사한 절차를 가짐
- 많은 분할을 시도하고, '불순도(inpurity)'가 최소화 되는 것을 선택해야 함
- 예측은 직사각형에 존재하는 목표 변수의 평균값으로 계산됨(CT에서는 다수결로 계산)
- 불순도(Inpurity)는 Leaf 노드의 평균값들에 대한 편차의 제곱합으로 측정
- RMSE에 의해 측정된 성능(Root Mean Squared Error)

랜덤 포레스트

- 앙상블
  - 배깅
  - 부스팅





Source: swallow.github.io

트리의 장점

- 사용 및 이해하기 쉬움
- 의사 결정 규칙들의 해석 및 구현이 쉬움
- 변수 선택 및 축소는 자동으로 진행됨
- 통계 모델의 가정이 필요하지 않음
- 결측 데이터를 광범위하게 처리하지 않아도 작동함

단점

- 데이터에 수평 혹은 수직으로 분할되지 않는 구조가 있을 때, 성능이 좋지 않을 수 있음

- 한 번에 하나의 변수를 다루기 때문에 변수 간의 상호 작용을 캡처할 수 있는 방법이 존재하지 않음

정리

- 분류 및 회귀 트리는 새로운 데이터를 예측하거나 분류하기 위한 방법으로, 쉽게 사용할 수 있으며 또한 쉽게 이해할 수 있는 방법임

- 트리는 규칙을 그래피컬(Graphical) 하게 표현한 것임

- 훈련용 데이터의 과적합을 방지하기 위해 가지치기가 필요함

- 트리는 데이터 구조에 대해 어떠한 가정도 하지 않으므로, 일반적으로 큰 규모의 샘플 데이터가 필요함