Tree

Tree-based Methods

- 회귀와 분류 모두 가능 (Regression and Classification)
- 계층화(stratifying), 세분화(segmenting)
- 기준에 따라 if-then-else 로 표현되는 규칙은 나무 형태로 요약 가능하기 때문에 의사결정나무(Decision Tree)방법이라고 함

Tree-based Methods

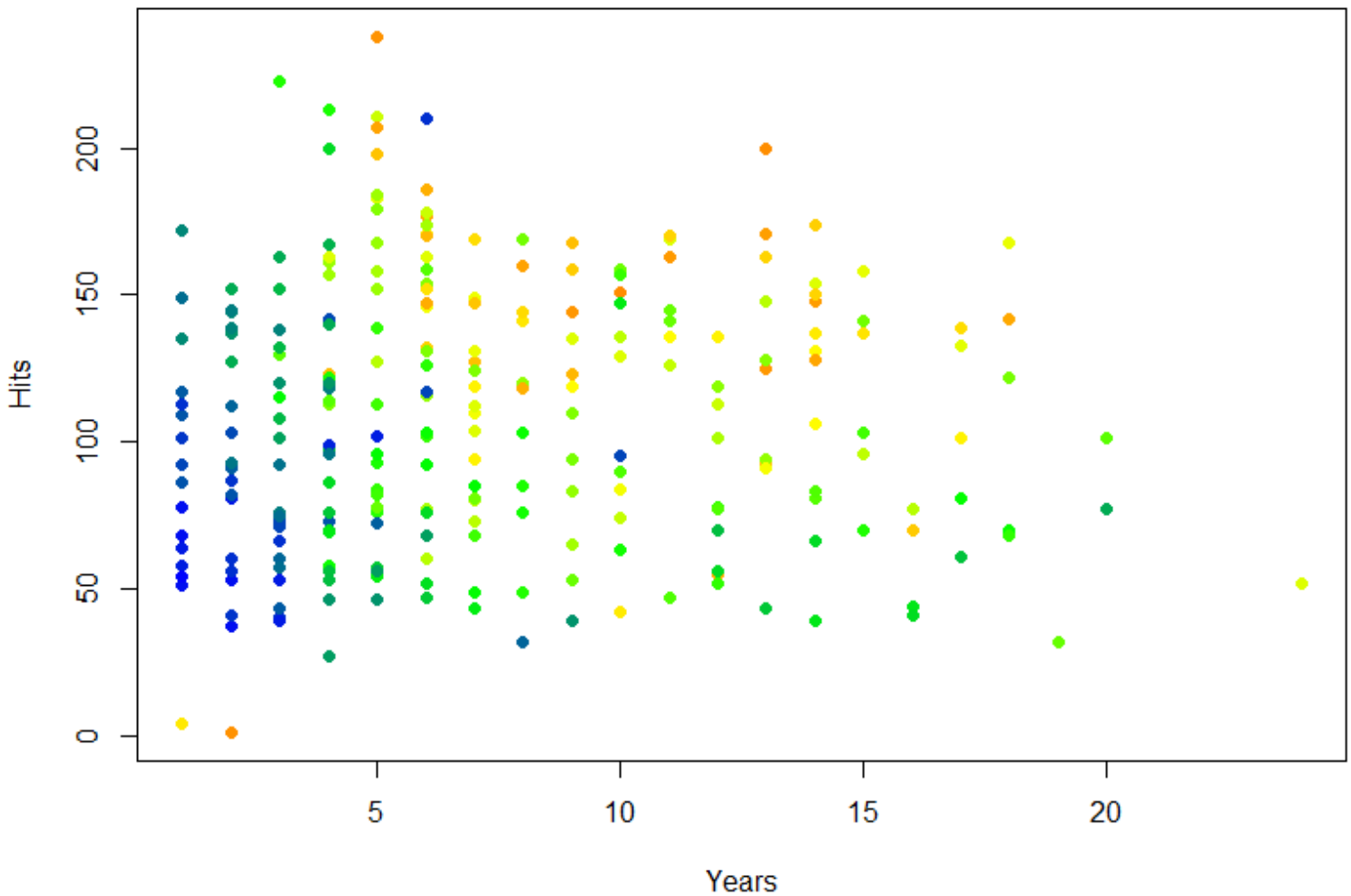
- 간단하고 설명력이 좋음
- 반면 다른 최적의 지도학습 모형에 비해 정확도가 떨어질 수 있음
- 앙상블 모형(Bagging, Boosting, Random Forest)을 이용하여 모형의 정확도를 향상할 수 있지만, 설명력은 떨어짐

Example : Baseball Salary Data

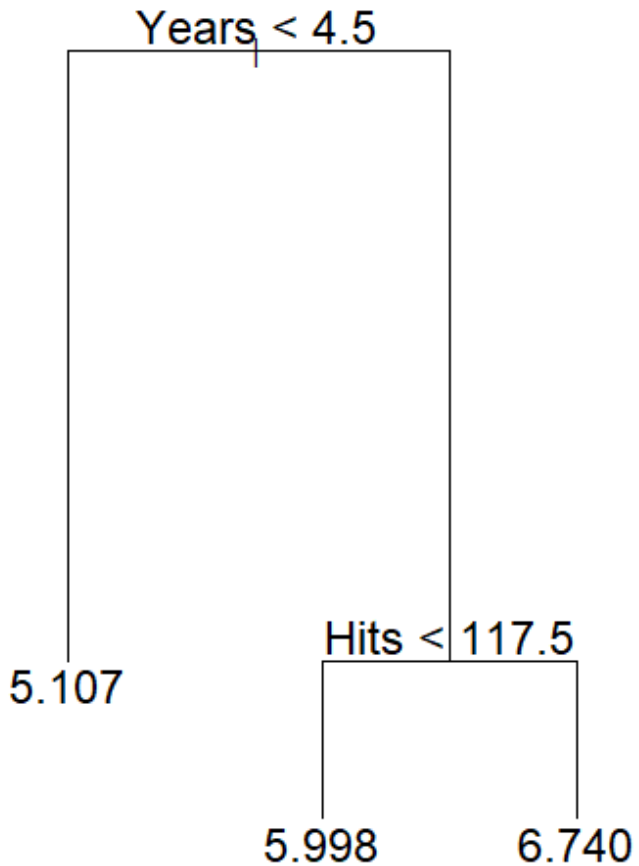
Table: Baseball Salary Data

	Hits	Years	Salary
1	66	1	NA
2	81	14	475.00
3	130	3	480.00
4	141	11	500.00
5	87	2	91.50
6	169	11	750.00
⋮	⋮	⋮	⋮

Example : Baseball Salary Data

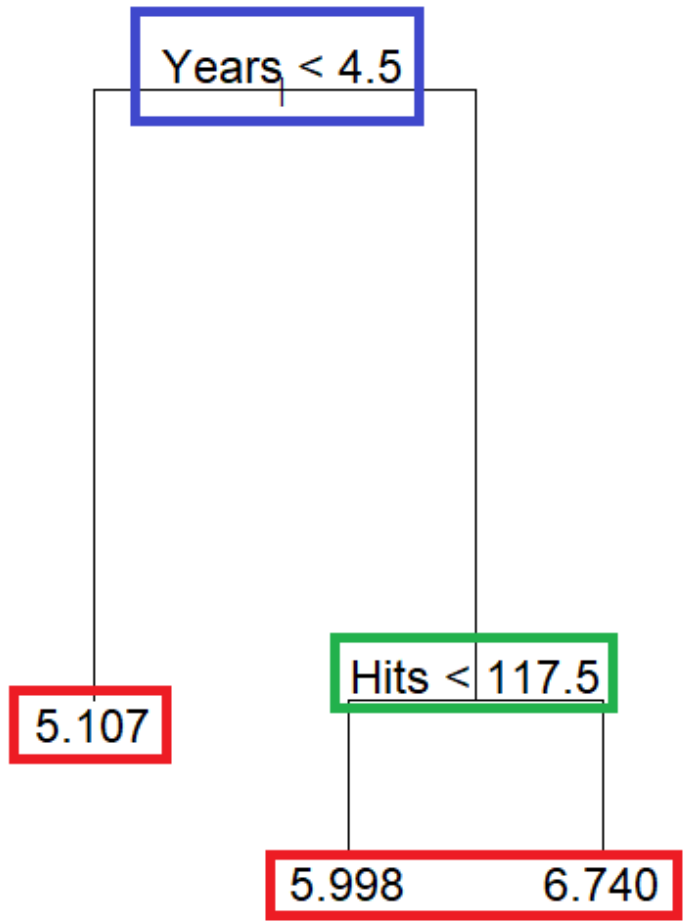


Example : Baseball Salary Data - Decision Tree

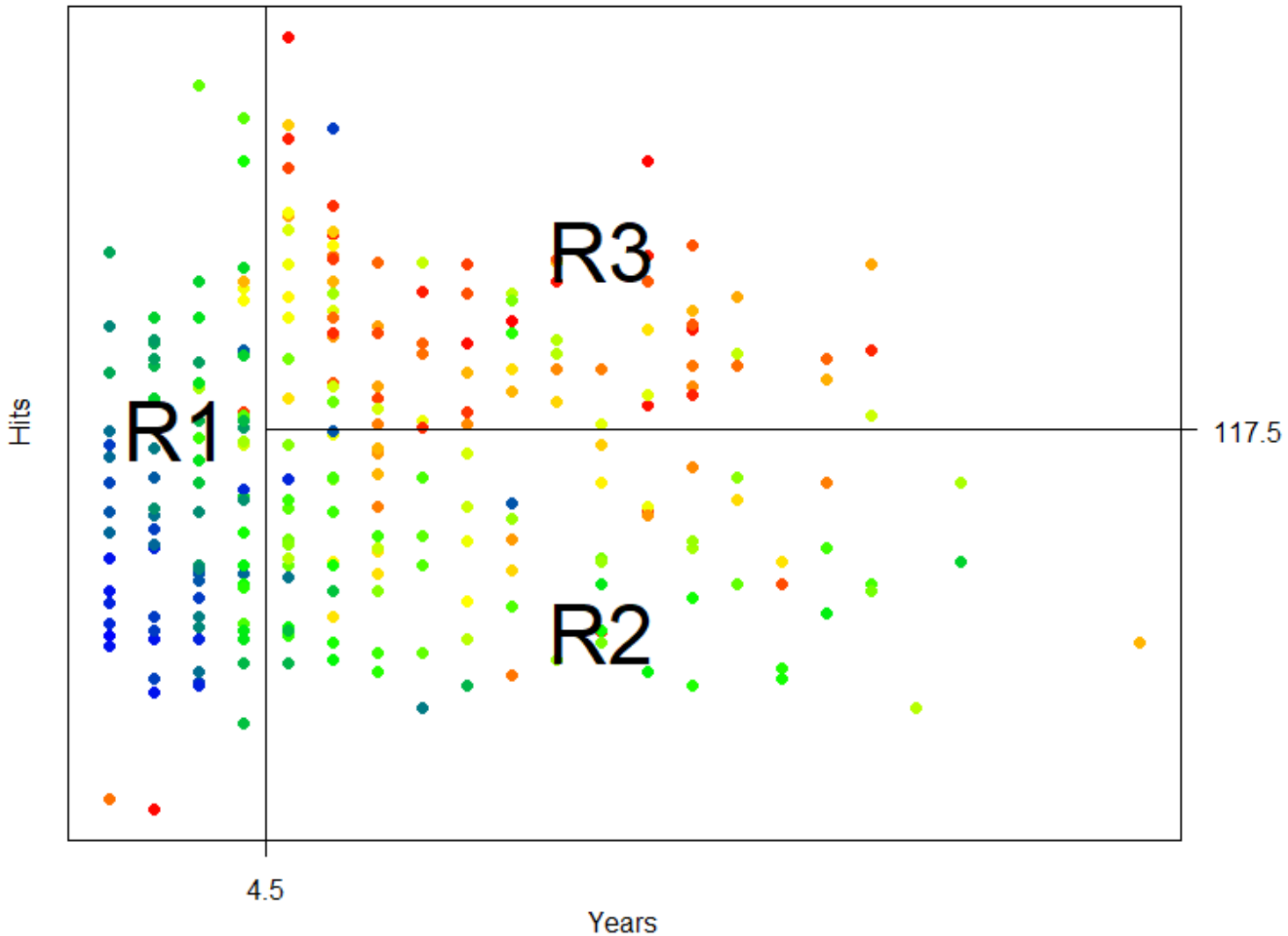


구성요소

- 뿌리마디(root node)
- 중간마디 (internal node)
- 끝마디 (terminal node)/ 잎(leaf)
- 깊이 (depth)



Example : Result



분석단계

- 의사결정나무의 형성 (Growing)
  - 각 마디에서 최적의 분리규칙 (split rule)을 찾아서 나무를 성장시키는 단계
  - 분석의 목적과 자료 구조에 따라 적절한 분리기준과 정지규칙 (stopping rule) 을 지정
  - 정지규칙을 만나면 중단
- 가지치기 (Pruning)
  - test error를 크게 할 위험이 높거나 부적절한 규칙을 가지고 있는 가지를 제거

형성단계

1. 설명변수들$(X_1, X_2, \ldots, X_p)$의 가능한 조합을 이용하여 예측 공간을 J개의 겹치지 않는 (non-overlapping) 구역으로 분할
2. 각 관측값은 R_j 구역에 포함되며, R_j 구역에 포함된 training data의 반응변수(y)의 평균을 이용하여 예측

$$\hat{y}_{R_j} = \frac{1}{n_j} \sum_{k \in R_j} y_k$$

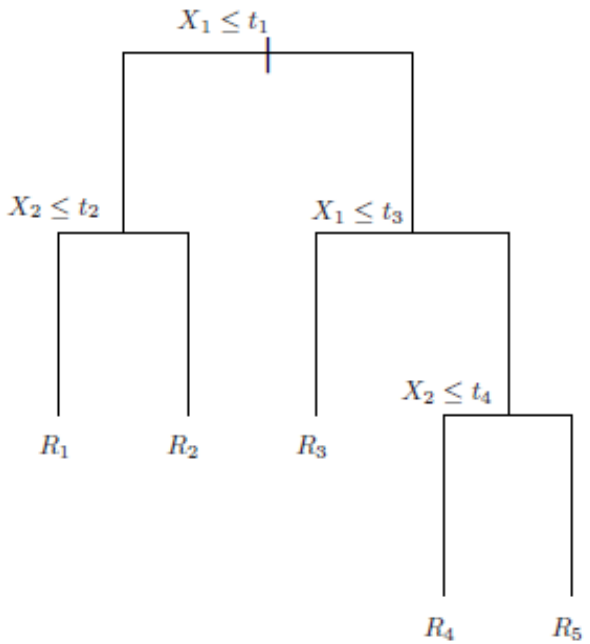
형성단계

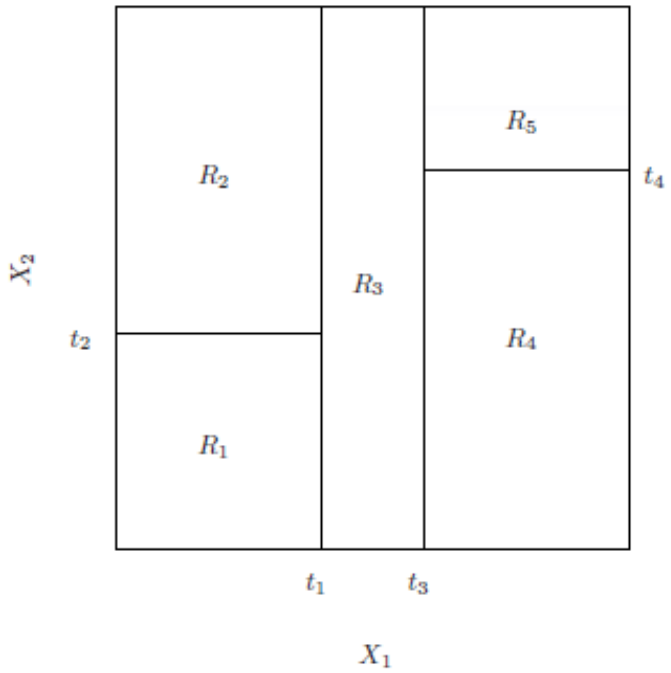
- 목표 : 다음의 RSS를 최소화 하는 구역 $R_1, \ldots, R_J$을 찾는 것

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

- 모든 조합을 확인하는 것은 불가능
- 재귀적인 이진 분리법(recursive binary splitting) 사용
- top-down approach
- 주어진 단계의 노드에서의 최적 분리 고려

분할


$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


R_5
R_2
t_4
X_2
R_3
t_2
R_4
R_1
t_1
t_3
X_1

분리규칙

- 분리변수가 연속형인 경우
  - 분리기준 : 관측값을 순서대로 정렬했을 때의 관측값 사이의 중간값
  - 기준보다 작으면 왼쪽 가지로, 크면 오른쪽 가지로 분리
- 분리변수가 범주형인 경우
  - 분리기준은 전체 범주를 두 개의 부분 집합으로 나누는 것
  - 범주가 $\{a, b, c\}$일 때, 분리 기준이 $\{a, b\}, \{c\}$ 라면 분리변수가 범주 $\{a, b\}$에 속하면 왼쪽 가지로 범주 $\{c\}$ 에 속하면 오른쪽 가지로 분리

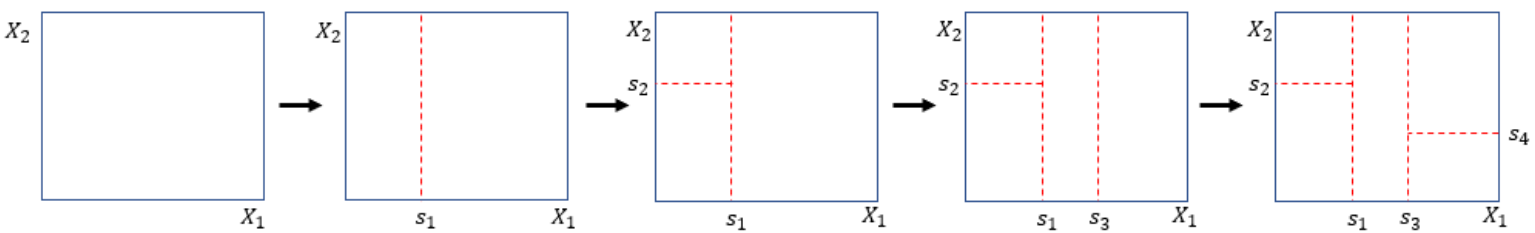
분할 알고리즘

1. 각 설명변수 X_j에 대하여 $(j = 1, \ldots, p)$
   (1) 설명변수 X_j에 해당하는 각 분리기준 s_i 대하여 $(i = 1, \ldots, n)$
      - 현재 노드의 모든 데이터를 $X_j < s_i$인 부분과 $X_j \geq s_i$인 부분으로 분할
      - 현재 노드의 각 하위 분할 영역의 RSS 측정

   (2) 현재 노드와 하위 노드의 RSS의 차이를 가장 크게 해주는 s_i 선택
2. 노드의 RSS를 가장 작게 해주는 변수 X_j와 s_i선택

재귀분할 알고리즘

1. $A =$ 전체 데이터
2. A를 A_1, A_2로 나누기 위한 분할 알고리즘 적용
3. A_1, A_2각각에 2번의 과정 반복
4. 분할을 해도 더는 하위 영역의 MSE가 개선되지 않을 정도로 충분히 분할이 진행되었으면 분할 종료



형성단계 - 정지규칙

- 정지규칙(Stopping Rule)
  - 현재의 노드가 더이상 분리가 일어나지 못하게 하는 규칙
  - 종류
    - 모든 자료가 한 범주에 속할 때
    - 노드에 속하는 자료가 일정 수 이하일 때
    - MSE의 감소량이 아주 작을 때
    - 뿌리마디로부터의 깊이가 일정 수 이상일 때 등

과적합

- 분리가 많이 이루어져 많은 노드를 가질수록 training data 에서의 예측력은 증가
- 하지만 test data에서의 오차는 커지는 과적합(overfitting) 발생
- 분리의 수를 줄여 간단한 나무를 사용하면 편차(bias)가 조금은 커질 수 있지만 분산(variance)를 줄이면서 설명력을 높일 수 있음

가지치기

- 가장 복잡한 의사결정나무(T_0)를 생성한 후 가지치기를 통해 하위나무(subtree) 추출
- Cost complexity pruning

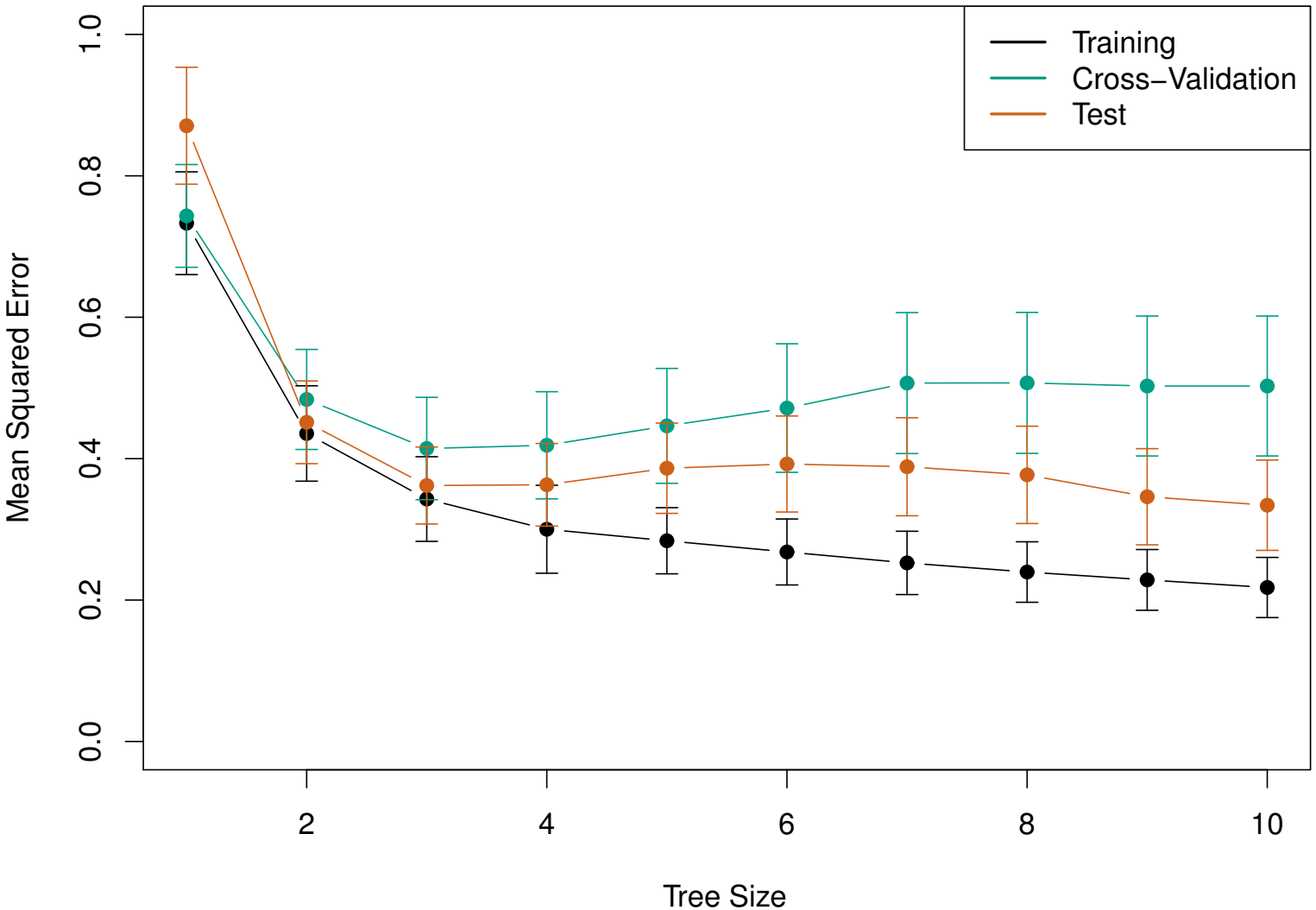
$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

  - $T \subset T_0$: 하위나무
  - $|T|$: 하위나무 T의 terminal node의 갯수
  - $\alpha > 0$: tuning parameter

가지치기

- α : 하위나무의 모형 복잡도와 training data의 오차 사이의 trade-off 를 조정하는 역할을 함
- 교차타당성(Cross-validation)을 통해 최적의 $\hat{\alpha}$ 선택
- 위에서 구한 $\hat{\alpha}$을 training data를 이용하여 만들어진 tree에 적용하여 가지치기 시행

Example


Training
Cross–Validation
Test
Mean Squared Error
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Tree Size

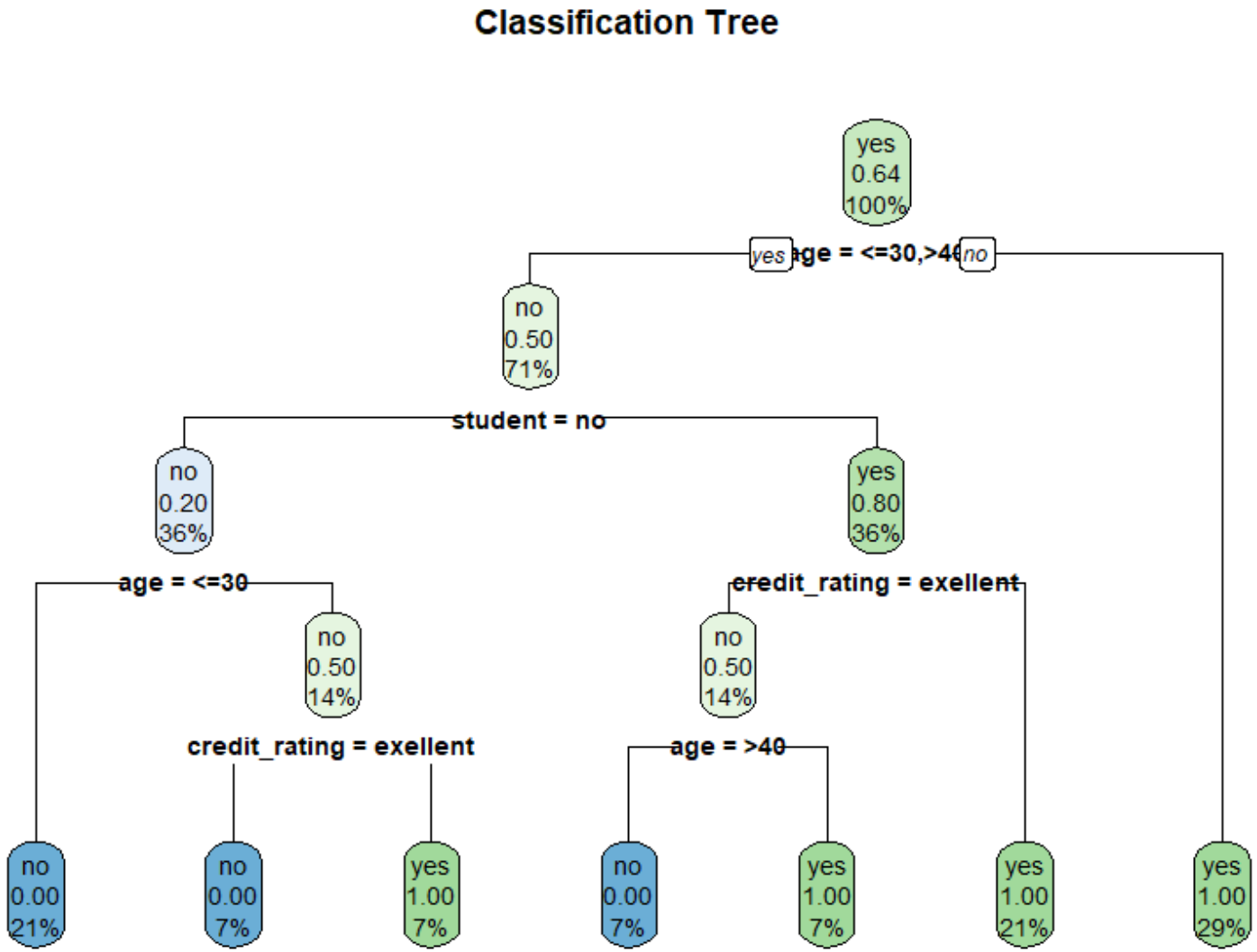
Classification Tree

- regression tree 와 같은 방법으로 분석
- voting으로 예측값 결정
- terminal node 안에 있는 범주들 중 가장 많은 관측값을 갖는 범주를 예측값으로 사용

Example - Buy Computer

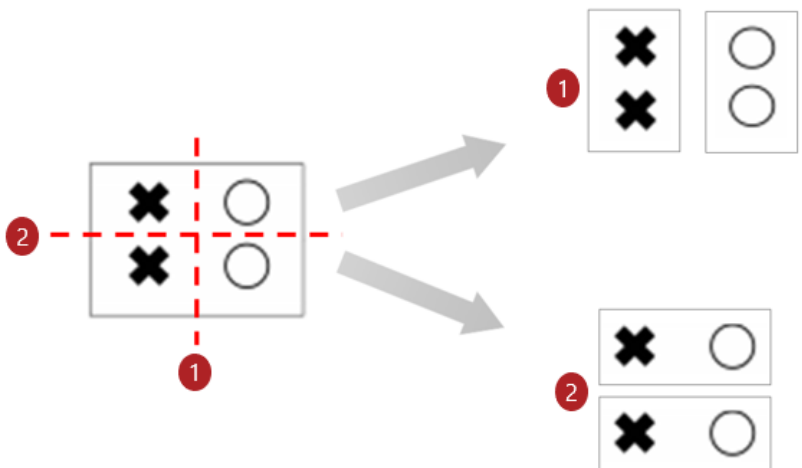
	age	income	student	credit_rating	buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	exellent	no
3	31...40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	exellent	no
7	31...40	low	yes	exellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	exellent	yes
12	31...40	medium	no	exellent	yes
13	31...40	high	yes	fair	yes
14	>40	medium	no	exellent	no

Example - Classification Tree



순수도(동질성)와 불순도(Impurity)

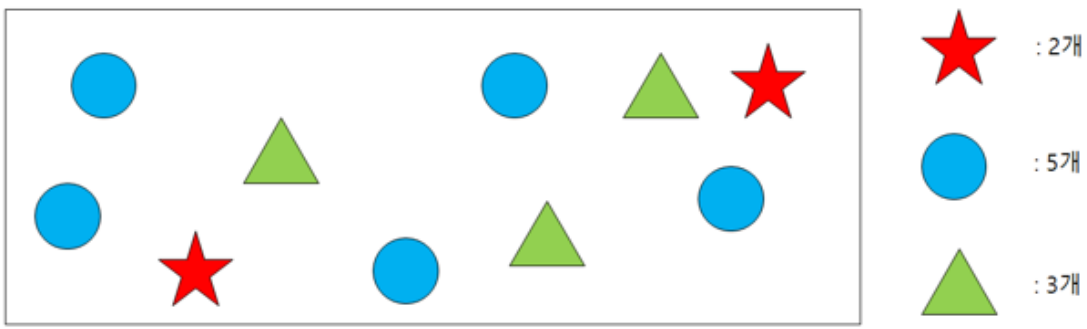
- regression tree와는 달리 RSS 계산 불가능
- 노드의 순수도를 높이거나 불순도를 낮추는 분할 선택



불순도 측도 - Gini Index

- 지니 지수 (Gini Index)

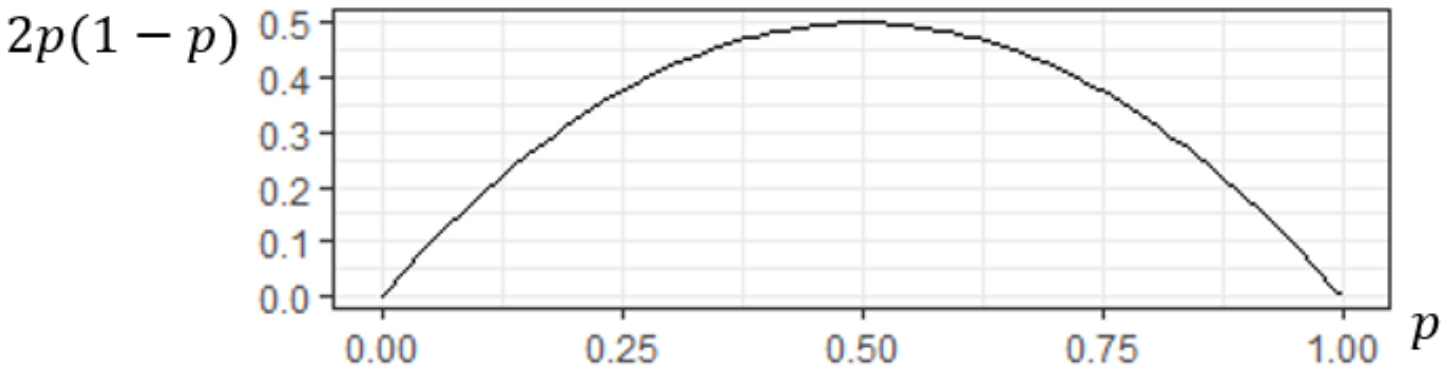
  - $Gini(D) = 1 - \sum_{k=1}^{K} p_k^2 = \sum_{k=1}^{K} p_k(1 - p_k)$
  - p_k : data set D에서 k번째 범주에 속하는 관측 비율
  - K : 범주의 갯수



  - $Gini(D) = 1 - \left\{ \left(\frac{2}{10}\right)^2 + \left(\frac{5}{10}\right)^2 \left(\frac{3}{10}\right)^2 \right\} = 0.62$

불순도 측도 - Gini Index

- 지니 지수 (Gini Index)



- 지니 지수는 $0 \sim (K-1)/K$ 사이의 값을 갖음
- $K = 2$인 경우 지니 지수 :

  $Gini(D) = 1 - (p_1^2 + p_2^2) = 2p_1(1-p_1)$ (최대값 $= 0.5$)

불순도 측도 - Gini Index

- 지니 지수 (Gini Index)

  - 분리규칙 A에 의해서 data set D가 D_1, D_2로 분리된다면

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

  - 불순도 감소 (reduction in impurity)

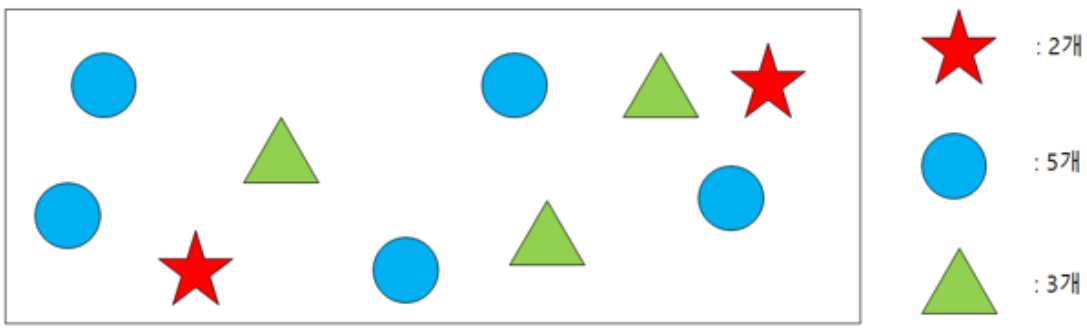
$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

  - $Gini_A(D)$를 가장 작게 하거나 $\Delta Gini(A)$를 가장 크게 하는 분리규칙 선택

불순도 측도 - Entropy

- 엔트로피 (Entropy)

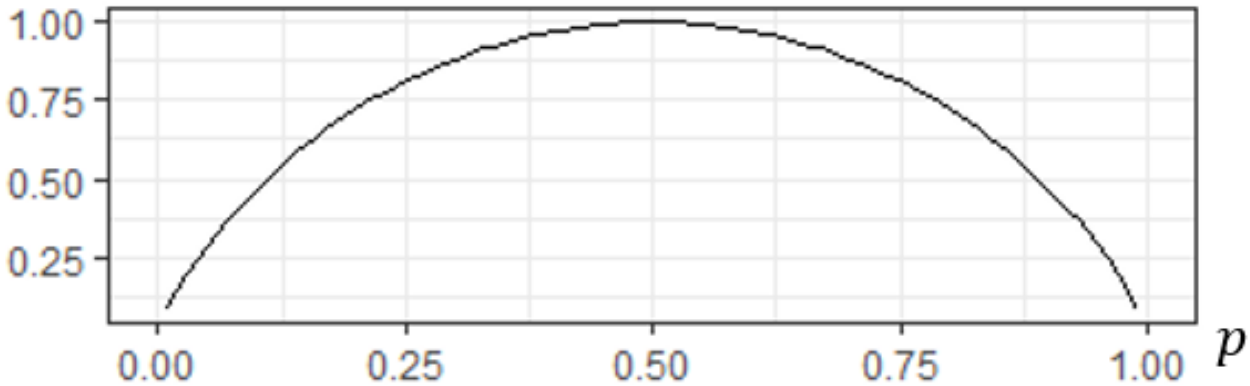
  - $E(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$
  - p_k : data set D에서 k번째 범주에 속하는 관측 비율
  - K : 범주의 갯수



  - $E(D) = -\{0.2 \times \log_2 0.2 + 0.5 \times \log_2 0.5 + 0.3 \times \log_2 0.3\} = 1.4854$

불순도 측도 - Entropy

- 엔트로피 (Entropy)



- $K = 2$인 경우 : $E(D) = -p\log_2(p) - (1-p)\log_2(1-p)$
  (최대값 $= 1$)
- 활률변수의 불확실성에 관한 측도
- Higher(Lower) entropy $\Rightarrow$ Higher(Lower) uncertainty

불순도 측도 - Entropy

- Information Gain

  - data set D의 expected information(=entropy)

$$Info(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

  - 분리규칙 A에 의해서 data set D가 D_1, D_2로 분리된다면

$$Info_A(D) = \frac{|D_1|}{|D|} Info(D_1) + \frac{|D_2|}{|D|} Info(D_2)$$

  - 정보이익 (Information Gain)

$$\Delta Gain(A) = Info(D) - Info_A(D)$$

  - $Info_A(D)$를 가장 작게 하거나 $\Delta Gain(A)$를 가장 크게 하는 분리규칙 선택

Example

Example

정리

- 의사결정나무의 장점
  - 이해하기 쉬움. 분류 작업이 용이
  - 교호작용 파악 가능
  - 시각화
  - 설명변수로 연속형/범주형 사용 가능
- 의사결정나무의 단점
  - 일반적으로 다른 모형에 비해 예측력이 낮은 경향이 있음