

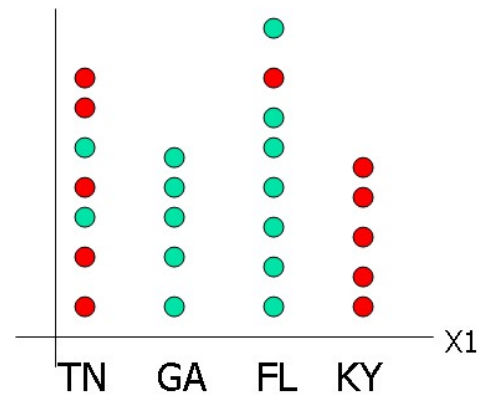
14강. 나무모형(2)

- 나무모형 분할 방법
- 나무모형 선택
- R 나무모형 분석
- 파이썬 나무모형 분석

2. 나무모형 분할방법

1) CART 방법

(2) 범주형 변수의 예 : 1개 변수이며 2개 집단에 속한 범주형 자료



총 25개의 관찰치중 적색 그룹은 11개, 녹색 그룹은 14개

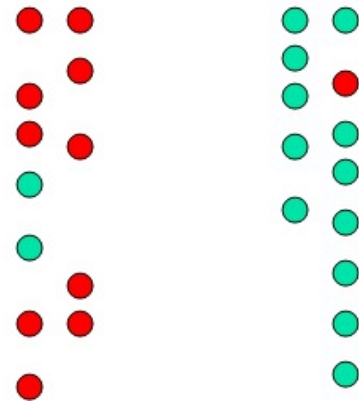
$$p(1|t) = \frac{11}{25}, \text{ and } p(2|t) = \frac{14}{25}.$$

$$\text{지니지수} = 1 - \left(\frac{11}{25}\right)^2 - \left(\frac{14}{25}\right)^2 = 0.492$$

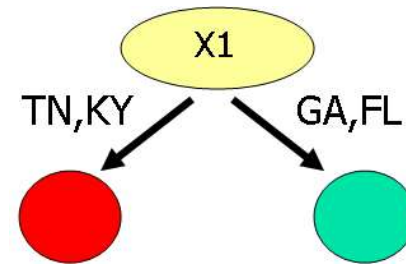
2. 나무모형 분할방법

1) CART 방법

● 분할점 {TN,KY} vs. {GA,FL}



$$\text{지니지수} = 1 - \left(\frac{2}{12}\right)^2 - \left(\frac{10}{12}\right)^2 = 0.278$$



$$\text{지니지수} = 1 - \left(\frac{12}{13}\right)^2 - \left(\frac{1}{13}\right)^2 = 0.142$$

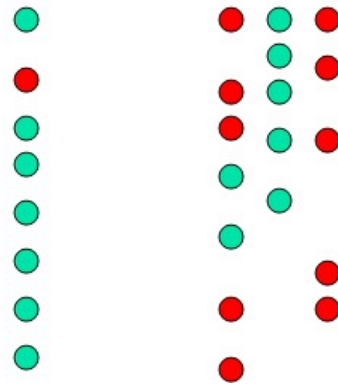
$$\text{가중평균} = 0.278 \times \frac{12}{25} + 0.142 \times \frac{13}{25} = 0.207$$

2. 나무모형 분할방법

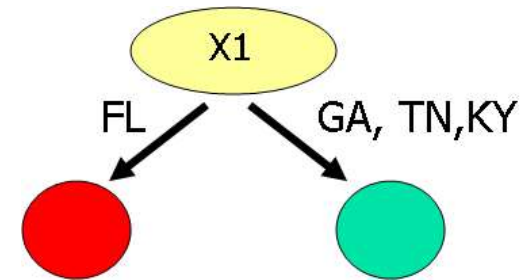
1) CART 방법

● 분할점

{FL} vs. {TN,GA,KY}



$$\text{지니지수} = 1 - \left(\frac{1}{8}\right)^2 - \left(\frac{7}{8}\right)^2 = 0.219$$



$$\text{지니지수} = 1 - \left(\frac{10}{17}\right)^2 - \left(\frac{7}{17}\right)^2 = 0.484$$

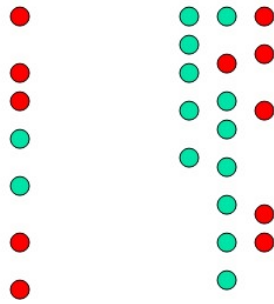
$$\text{가중평균} = 0.219 \times \frac{8}{25} + 0.484 \times \frac{17}{25} = 0.399$$

2. 나무모형 분할방법

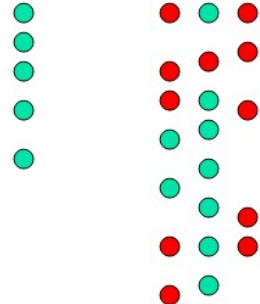
1) CART 방법

- 기타 분할점

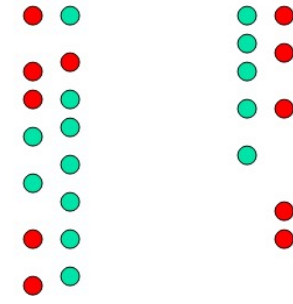
{TN} vs. {GA,FL, KY}



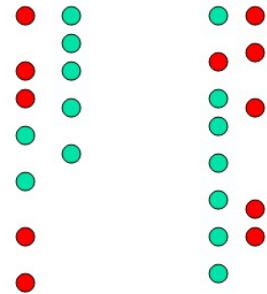
{GA} vs. {TN,FL,KY}



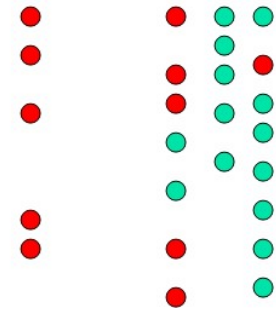
{TN,FL} vs. {GA,KY}



{TN,GA} vs. {FL,KY}



{KY} vs. {TN,GA,FL}



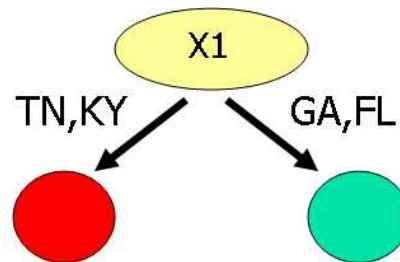
2. 나무모형 분할방법

1) CART 방법

- 모든 분할점 비교

좌측분할	우측분할	지니지수
TN	GA, FL, KY	0.434
GA	TN, FL, KY	0.396
FL	TN, GA, KY	0.399
KY	TN, GA, FL	0.300
TN, GA	FL, KY	0.492
TN, FL	GA, KY	0.488
TN, KY	GA, FL	0.207

- 최적 분할점



2. 나무모형 분할방법

1) CART 방법

(3) 연속형과 범주형 변수가 혼재되어 있는 경우

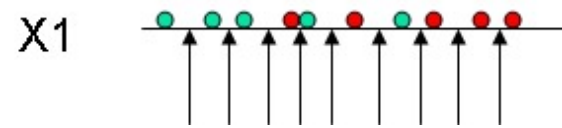
Class	X1	X2	X3
●	1.9	1	A
●	1.7	1	A
●	1.6	1	B
●	1.3	0	B
●	1.2	0	B
●	1.5	1	A
●	1.25	0	B
●	1	0	C
●	0.9	0	C
●	0.7	0	C

총 10개의 관찰치중 적색 그룹은 5개, 녹색 그룹은 5개

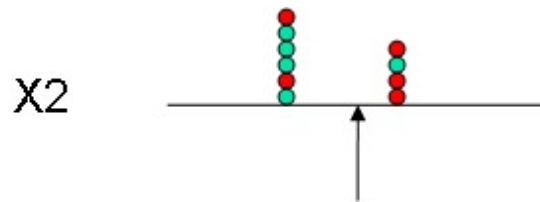
2. 나무모형 분할방법

1) CART 방법

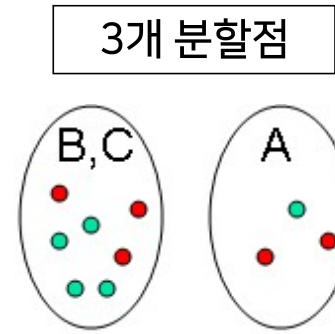
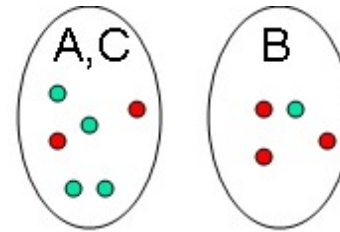
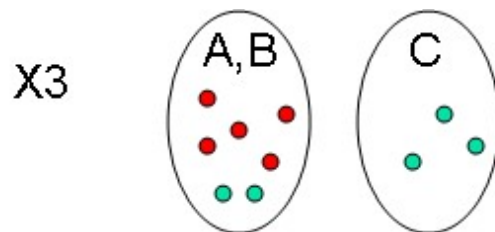
(3) 연속형과 범주형 변수가 혼재되어 있는 경우



9개 분할점



1개 분할점

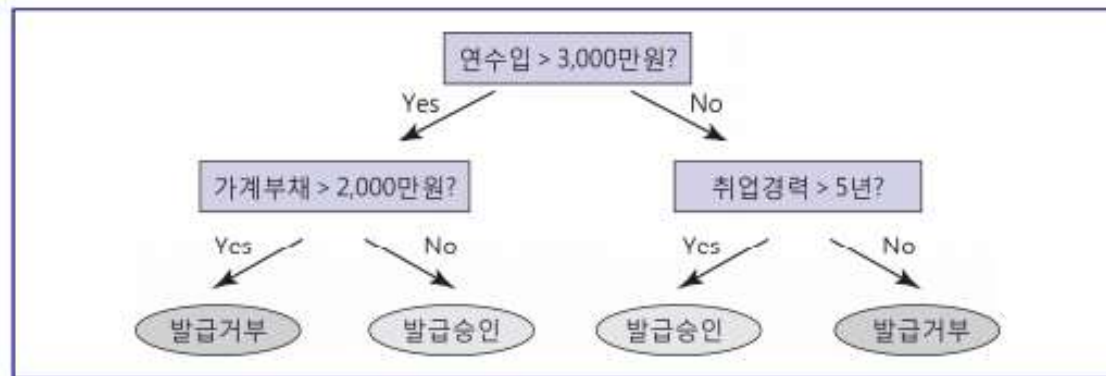


불순도가 가장 많이 감소되는 최적분리점 선택

2. 나무모형 분할방법

1) CART 방법

(4) 반복적 분할

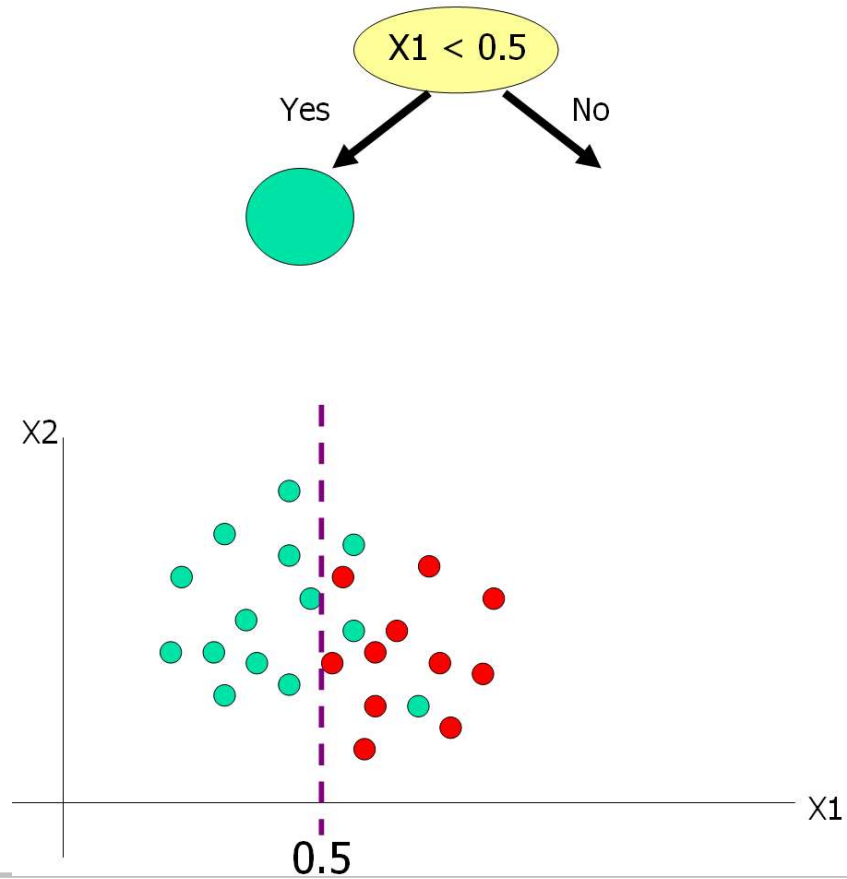


- ❖ (1) - (3) 절차를 매단계의 노드마다 반복적으로 적용함
- ❖ 나무모형은 단계 단계마다 불순도가 적은 분할규칙을 결정
- ❖ 같은 그룹의 자료를 최종노드에 포함하도록 하는 방법

2. 나무모형 분할방법

1) CART 방법

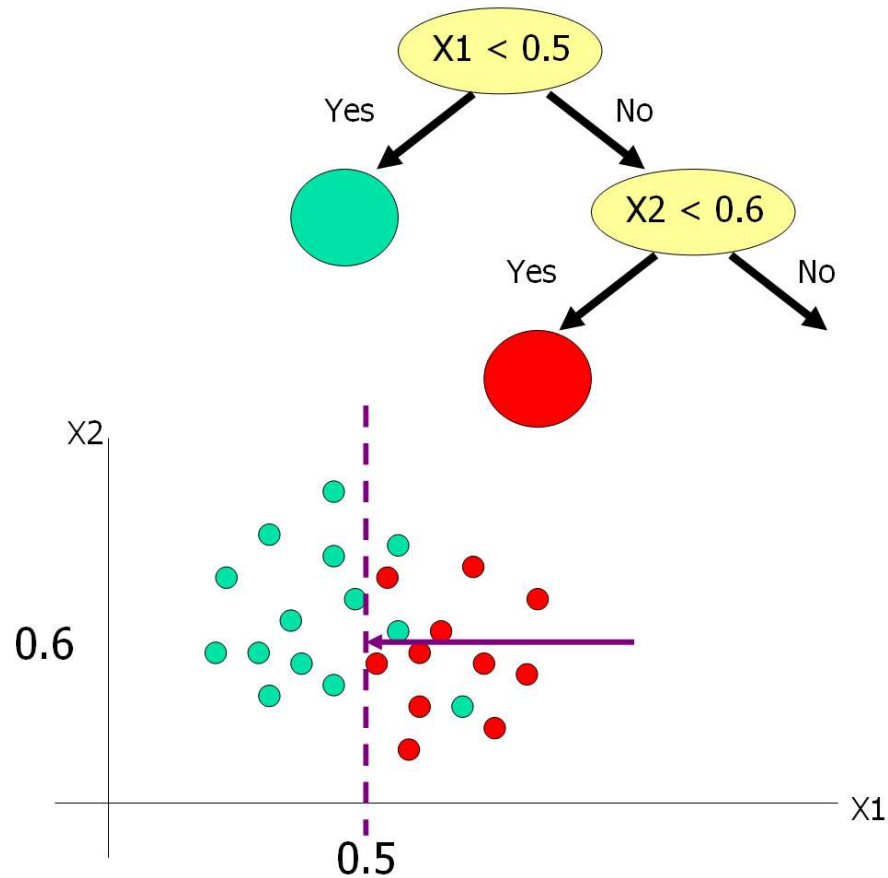
(4) 반복적 분할



2. 나무모형 분할방법

1) CART 방법

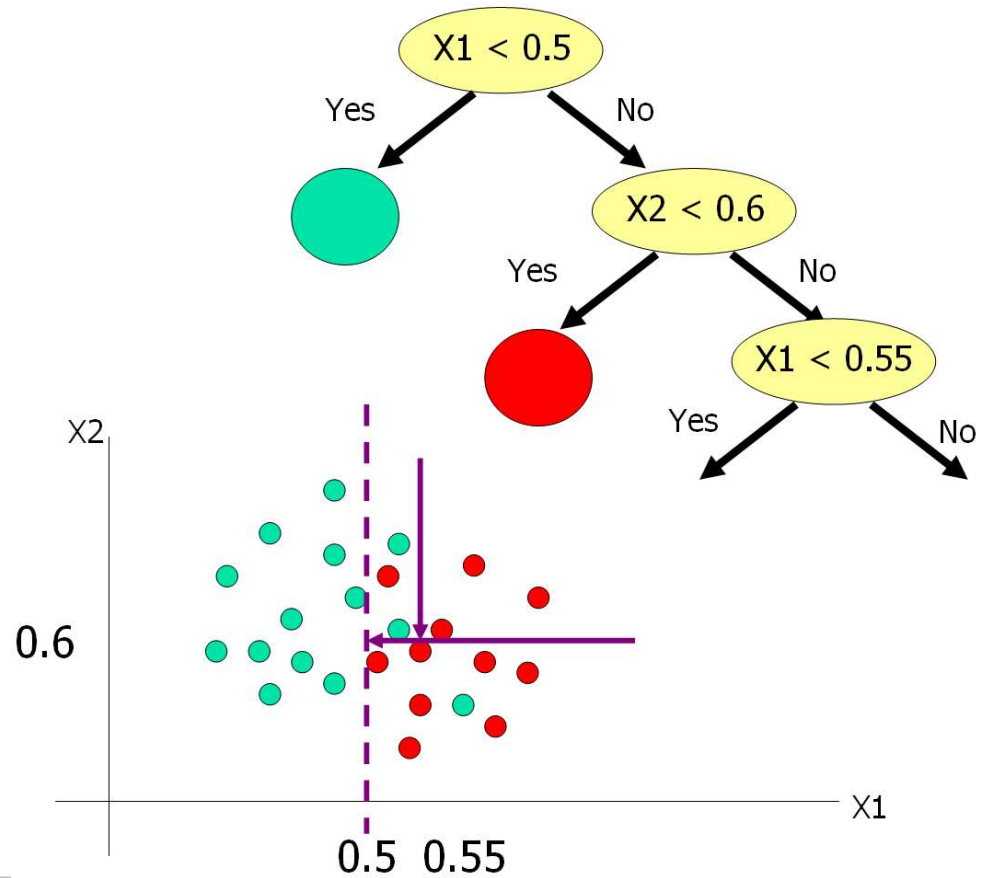
(4) 반복적 분할



2. 나무모형 분할방법

1) CART 방법

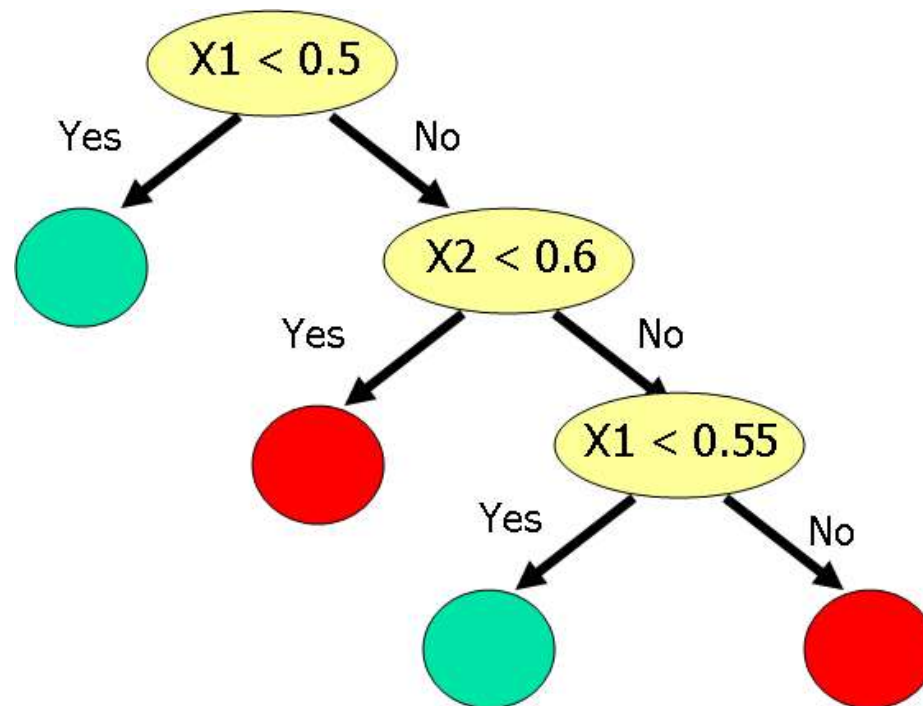
(4) 반복적 분할



2. 나무모형 분할방법

1) CART 방법

(4) 반복적 분할



2. 나무모형 분할방법

2) CHAID 방법

CHAID (CHi - squared Automatic Interaction Detection) 방법

: 분할표 검정에서 사용되는 카이제곱 검정방법을 사용하여
분할점 혹은 분할집합을 결정하는 방법.

2. 나무모형 분할방법

3) QUEST 방법

QUEST (Quick Unbiased Efficient Statistical Tree) 방법

: CART의 변수선택에 대한 편향 현상을 수정하기 위해 개발된 방법임.
CART의 편향 현상에 대한 이유가 변수선택과 분할점 선택이
동시에 이루어지기 때문이므로 이를 분리하여 먼저 변수를 선택하고,
그리고 선택된 변수에 대해서 분할점을 선택하는 방법임.

3. 나무모형 크기 결정

- 지나치게 많은 노드와 가지를 가진 나무모형은 해석이 복잡함
새로운 자료에 적용시킬 때 예측오차가 매우 커지는 문제도 발생
- 적절한 크기의 나무모형을 선택하면
해석력뿐만 아니라 예측정확도를 향상 시킴

3. 나무모형 크기 결정

1) 유의성 방법

- CHAID(Chi-squared Automatic Detection) 에서 사용
- 나무구조를 만들어 갈 때, 단계마다 분할이 꼭 필요한 것인지 통계적 유의성을 이용하여 평가
 - 만약 분할이 꼭 필요하다면 계속 분할함
- 만약 분할이 유의하게 필요한 것이 아니면 분할을 정지하고 그때까지 구해진 나무모형을 최종모형으로 채택
- 빠른 시간에 나무모형을 완성할 수 있다는 장점
- 예측 정확도가 떨어지는 단점

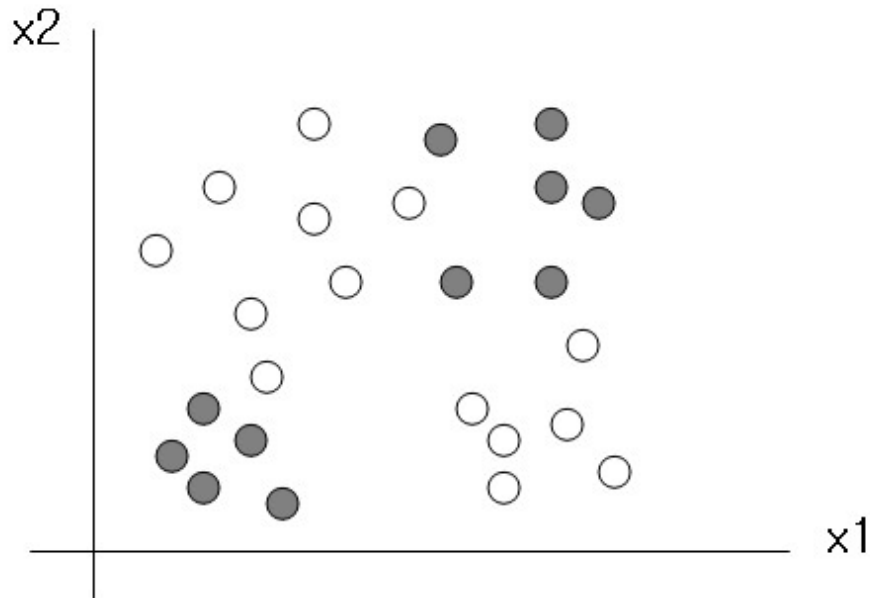
3. 나무모형 크기 결정

2) 정지규칙 방법

- 단계마다 분할의 유의성을 평가하지 않고 계속적으로 분할
- 궁극적으로 하위 노드의 어떤 단계에서는 분할을 멈춤
(하위노드에 속한 관찰치의 수가 적기 때문)
- 잠정적으로 구해진 나무구조는 매우 큰 규모임
→ 불필요한 가지를 제거하여 적당한 크기의 나무구조로 축소시킴
- 유의성 방법보다 더 우월한 나무구조를 찾아낼 수 있다는 장점
- 연산이 오래 걸린다는 단점
- CART 방법과 QUEST(Quick Unbiased Efficient Statistical Tree) 방법이 채택

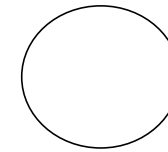
3. 나무모형 크기 결정

2) 정지규칙 방법



나무모형

-뿌리노드만 존재

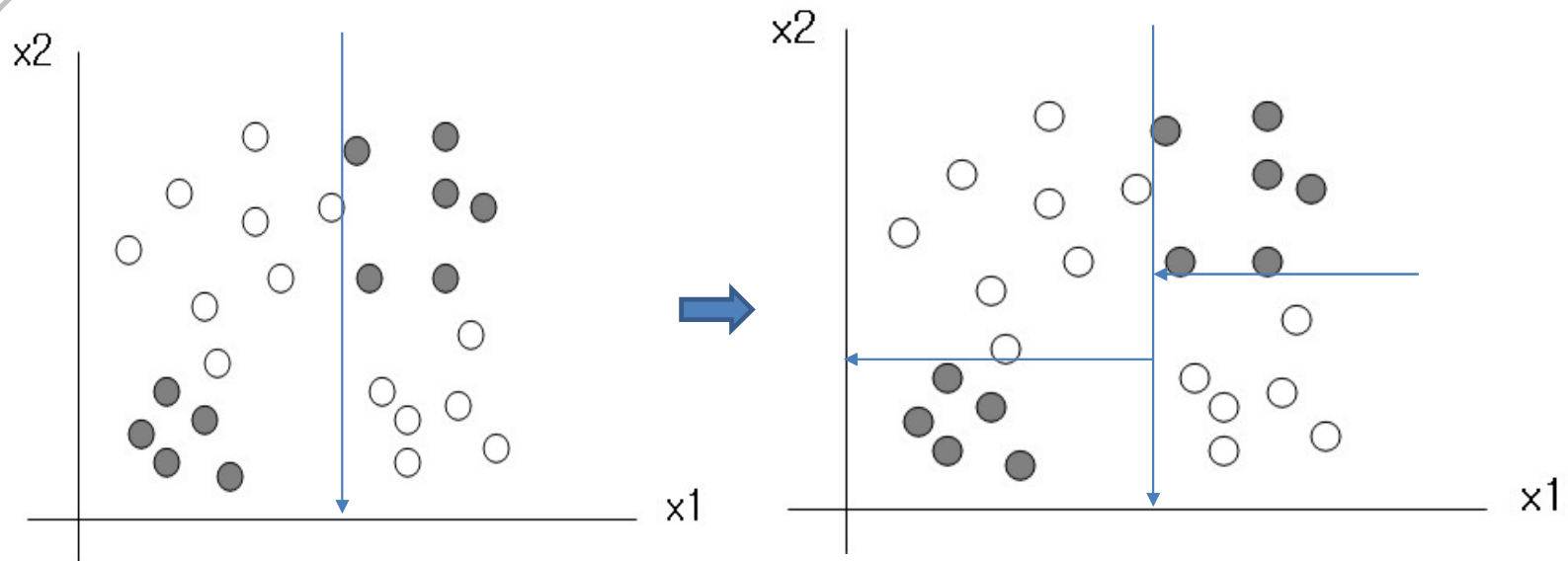


분할정지 방법을 사용하면
분할의 유의성이 없어서
분할하지 않는 노드

정지규칙: 관찰치의 개수가 10개보다 작아지면 정지하기로 함.

3. 나무모형 크기 결정

2) 정지규칙 방법



노드 t 를 분할의 유의성이 없음에도 불구하고 분할을 강행

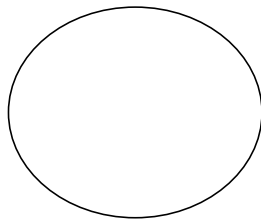
t_1 과 t_2 는 분할의 유의성이 있는 노드가 됨

노드 t_1 과 t_2 의 관찰치의 개수가 10개 이상
이므로 분할 계속함.

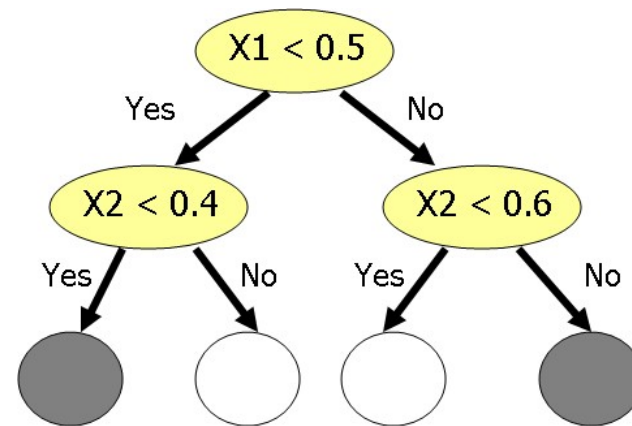
3. 나무모형 크기 결정

2) 정지규칙 방법

분류나무



유의성 방법에 의한 분류



정지규칙 방법에 의한 분류

3. 나무모형 크기 결정

3) 정지규칙 종류

- 최대깊이(maxdepth)

: 나무모형이 뿌리노드에서 출발하여 아래로 분할해 갈 수 있는 최대 허용 깊이를 의미.
데이터의 크기를 고려하여 최대깊이를 설정하는 것이 좋으나,
일반적으로 3~5 정도로 설정해야 나무모형을 해석하는데 어려움이 없음.

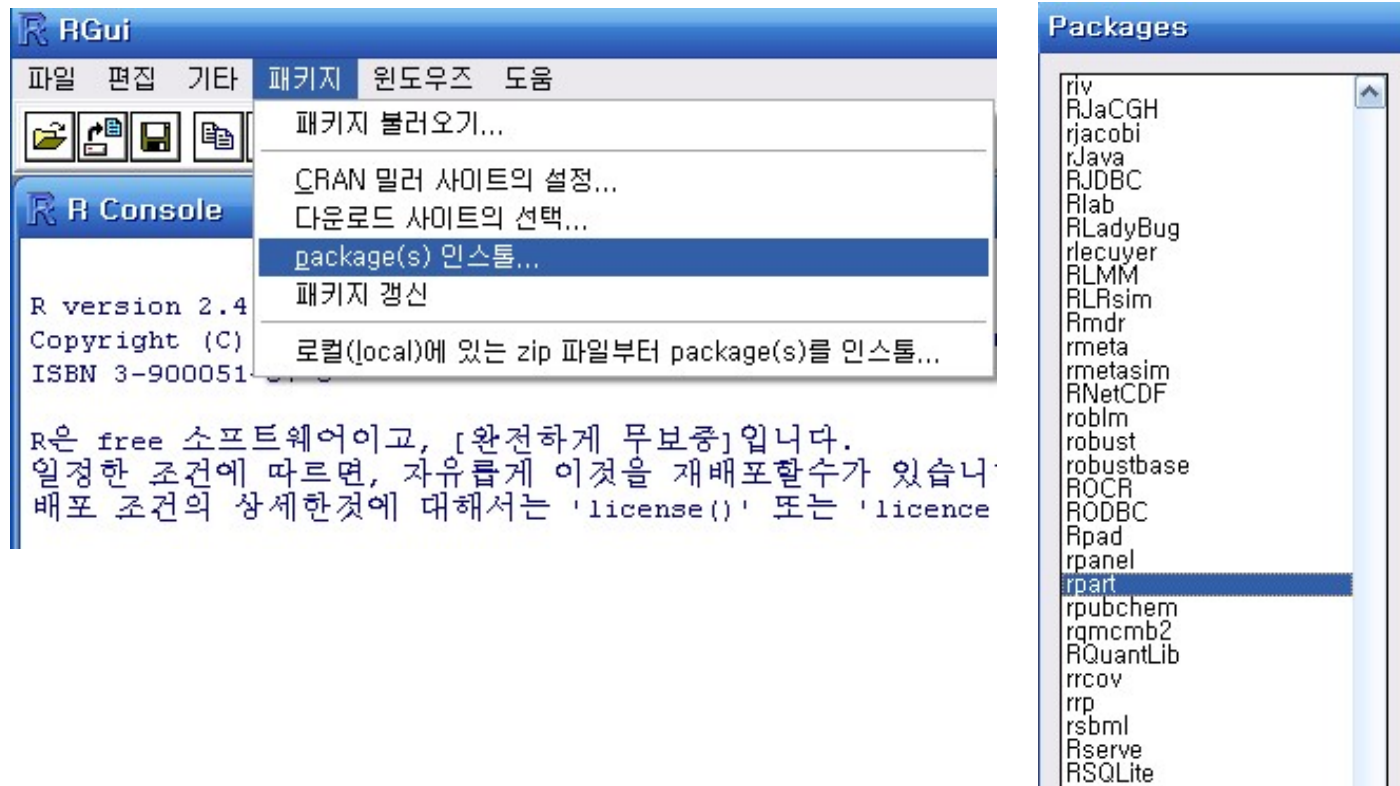
- 최소 데이터 수(minsplit)

: 나무모형의 한 노드를 분할하기 위해 필요한 노드 내의 데이터 수.

사전에 분할에 필요한 최소 데이터 수를 설정해 놓고,
이에 못미치는 노드는 더 이상 분할하지 않고, 최종노드로 결정하는 방법.
일반적으로 최소 데이터수의 값은 20을 많이 사용하지만,
분석 데이터의 크기를 고려하여 더 크게 혹은 더 작게 설정할 수도 있음.

4. R 나무모형 분석

나무모형 패키지 rpart 설치



The screenshot displays the RGui application window. The 'package(s) install...' option is selected in the 'package' menu. The 'R Console' shows the R version 2.4.0 and copyright information. The 'Packages' list on the right includes various installed packages, with 'rpart' highlighted.

R version 2.4
Copyright (C)
ISBN 3-900051-10-9

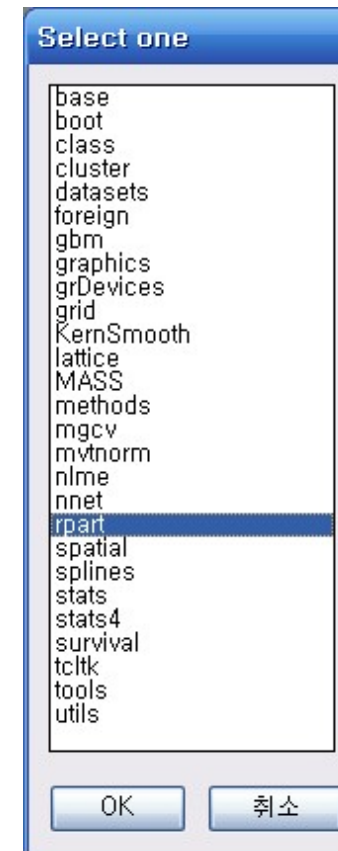
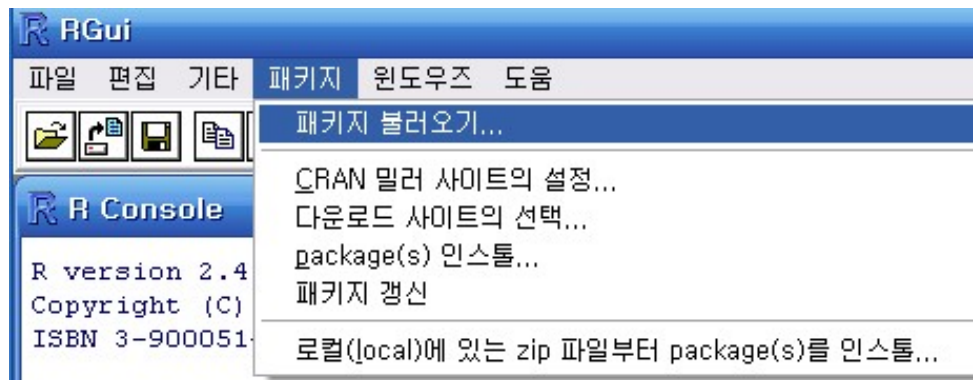
R은 free 소프트웨어이고, [완전하게 무보종]입니다.
일정한 조건에 따르면, 자유롭게 이것을 재배포할수가 있습니다.
배포 조건의 상세한것에 대해서는 'license()' 또는 'licence'

Packages

- riv
- RJaCGH
- rjacob
- rJava
- RJDBC
- Rlab
- RLadyBug
- rlecuyer
- RLMM
- RLRsim
- Rmdr
- rmeta
- rmetasim
- RNetCDF
- robim
- robust
- robustbase
- ROCR
- RODBC
- Rpad
- rpanel
- rpart**
- rpubchem
- rgmcm2
- RQuantLib
- rrcov
- rrp
- rsbml
- Rserve
- RSQLite

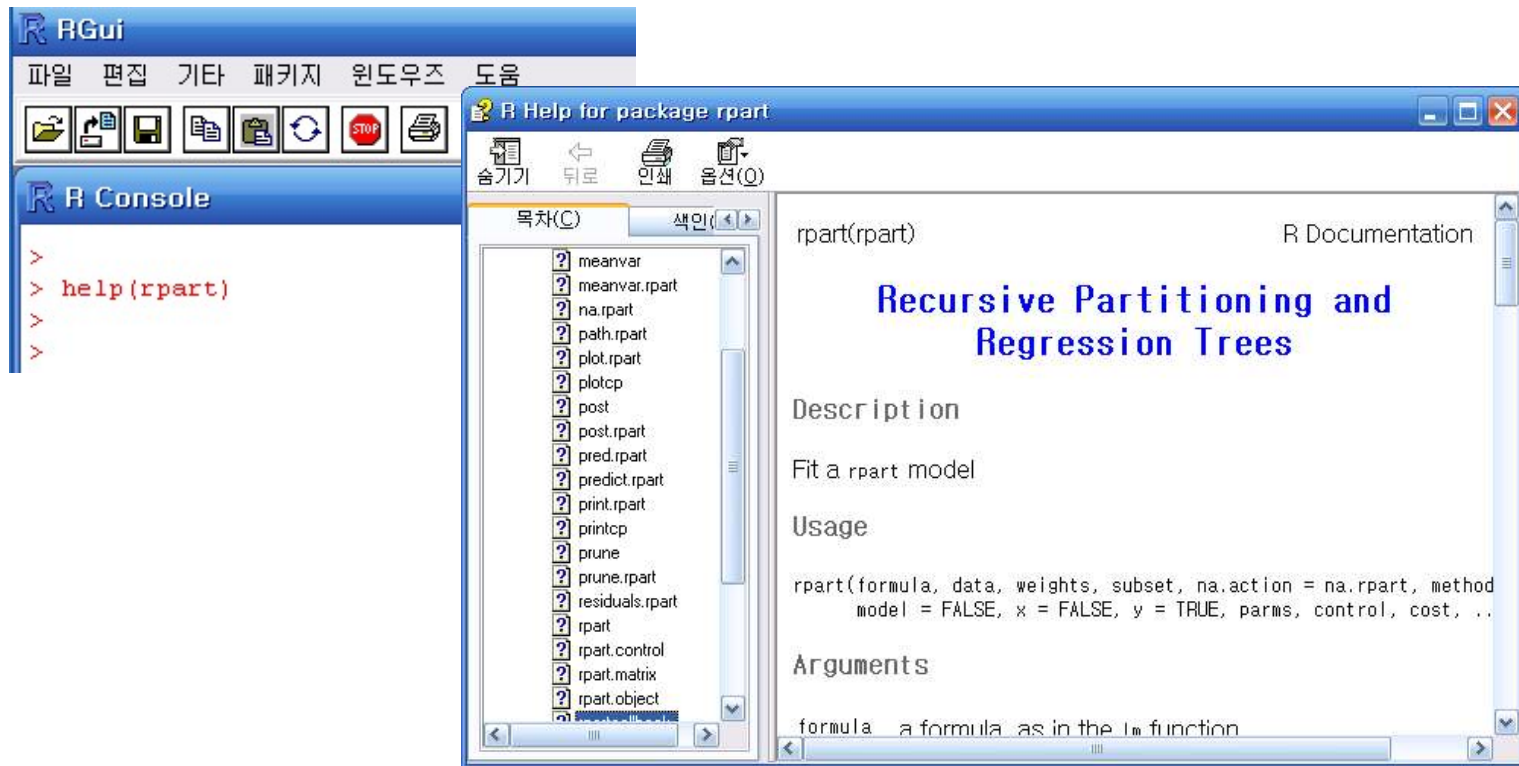
4. R 나무모형 분석

rpart 패키지 불러오기



4. R 나무모형 분석

Help 불러보기



4. R 나무모형 분석 : 타이타닉

	좌석등급				
	선원	1등석	2등석	3등석	총계
생존	212	203	118	178	711
사망	673	122	167	528	1490
총계	885	325	285	706	2201
생존률	0.24	0.62	0.41	0.25	0.32

	성별		
	남성	여성	총계
생존	367	344	711
사망	1364	126	1490
총계	1731	470	2201
생존률	0.21	0.73	0.32

	연령별		
	성인	어린이	총계
생존	654	57	711
사망	1438	52	1490
총계	2092	109	2201
생존률	0.31	0.52	0.32

4. R 나무모형 분석 : 타이타닉

"titanic.csv"

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Class, Age, Sex, Survived  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes  
First, Adult, Male, Yes
```

4. R 나무모형 분석 : 타이타닉

데이터 읽기

```
> titanic = read.csv("c:/data/mva/titanic.csv", header=T)
```

```
> head(titanic, 3)
```

	Class	Age	Sex	Survived
1	First	Adult	Male	Yes
2	First	Adult	Male	Yes
3	First	Adult	Male	Yes

```
> summary(titanic)
```

Class	Age	Sex
Length:2201	Length:2201	Length:2201
Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character

```
> table(titanic$Survived, titanic$Class)
      Crew First Second Third
No    673   122    167   528
Yes   212   203    118   178
```

```
> table(titanic$Survived, titanic$Age)
      Adult Child
No    1438    52
Yes    654    57
```

```
> table(titanic$Survived, titanic$Sex)
      Female Male
No      126 1364
Yes     344 367
```

4. R 나무모형 분석 : 타이타닉

참고 : 그룹별 count 가 주어진 경우

```
> titanic_w = read.csv("c:/data/mva/titanic_count.csv", header=T)
```

```
> head(titanic_w, 3)
```

	Class	Age	Sex	Survived	Count
1	Crew Adult	Male	Yes	192	
2	Crew Adult	Male	No	670	
3	Crew Adult	Female	Yes	20	

```
> titanic = titanic_w[rep(1:nrow(titanic_w), titanic_w$Count), -5]
```

```
> head(titanic, 3)
```

	Class	Age	Sex	Survived
1	Crew Adult	Male	Yes	
1.1	Crew Adult	Male	Yes	
1.2	Crew Adult	Male	Yes	

4. R 나무모형 분석 : 타이타닉

```
> library(rpart)
> # Default tree : maxdepth=30, minsplit=20
> cart_def = rpart(Survived ~ Class + Age + Sex, data=titanic)
> print(cart_def)
n= 2201

node), split, n, loss, yval, (yprob)
      * denotes terminal node

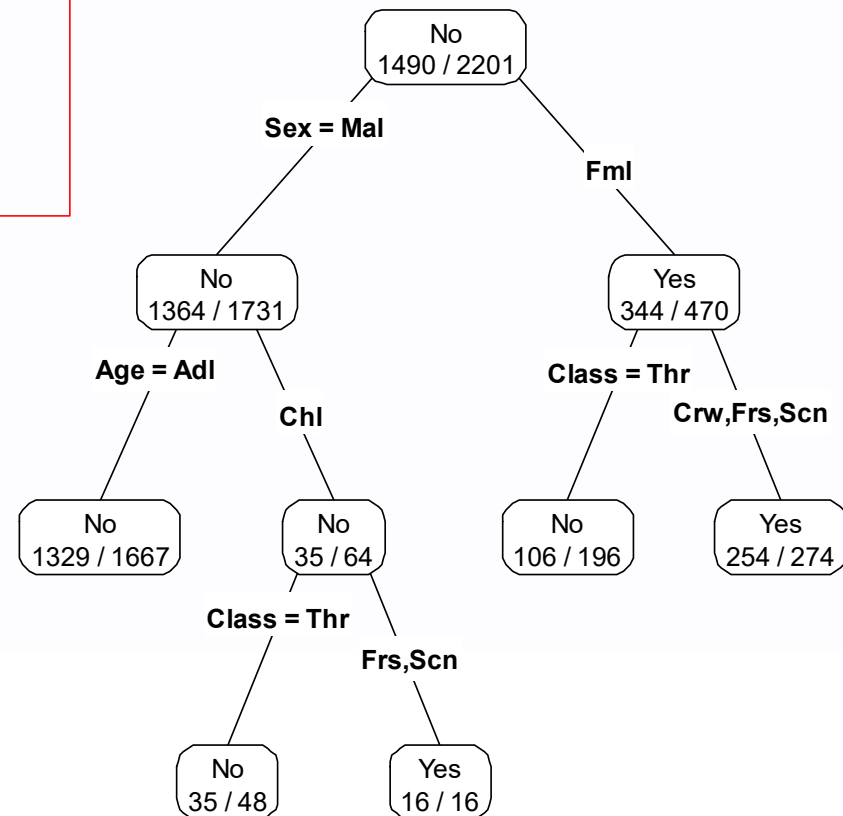
1) root 2201 711 No (0.6769650 0.3230350)
  2) Sex=Male 1731 367 No (0.7879838 0.2120162)
    4) Age=Adult 1667 338 No (0.7972406 0.2027594) *
    5) Age=Child 64 29 No (0.5468750 0.4531250)
      10) Class=Third 48 13 No (0.7291667 0.2708333) *
      11) Class=First,Second 16 0 Yes (0.0000000 1.0000000) *
  3) Sex=Female 470 126 Yes (0.2680851 0.7319149)
    6) Class=Third 196 90 No (0.5408163 0.4591837) *
    7) Class=Crew,First,Second 274 20 Yes (0.0729927 0.9270073) *
```

>

4. R 나무모형 분석 : 타이타닉

나무모형 그리기

```
> # Tree 그리기  
> library(rpart.plot)  
> prp(cart_def, type=4, extra=2, digits=3)  
>
```



4. R 나무모형 분석 : 타이타닉

정지규칙 설정하여 나무모형 실행하기 : maxdepth=1

```
> # Stopping rule 1
> my.control = rpart.control(maxdepth=1)
> cart_fit1 = rpart(Survived ~ Class + Age + Sex, control=my.control, data=titanic)
> print(cart_fit1)
n= 2201

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 2201 711 No (0.6769650 0.3230350)
  2) Sex=Male 1731 367 No (0.7879838 0.2120162) *
  3) Sex=Female 470 126 Yes (0.2680851 0.7319149) *
```

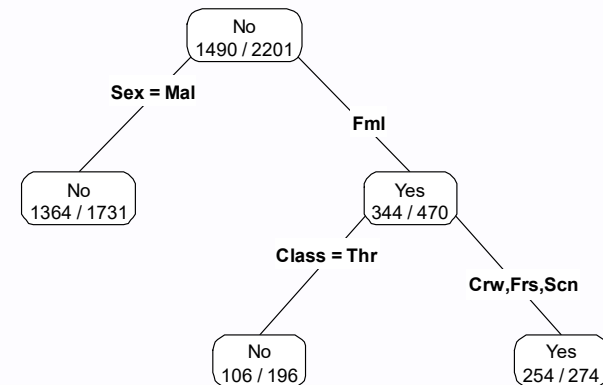

4. R 나무모형 분석 : 타이타닉

정지규칙 설정하여 나무모형 실행하기 : maxdepth=5, minsplit=50

```
> # Stopping rule 2
> my.control = rpart.control(maxdepth=5, minsplit=50)
> cart_fit2 = rpart(Survived ~ Class + Age + Sex, control=my.control, data=titanic)
> prp(cart_fit2, type=4, extra=2, digits=3)
> print(cart_fit2)
n= 2201
```

node), split, n, loss, yval, (yprob)
* denotes terminal node

```
1) root 2201 711 No (0.6769650 0.3230350)
  2) Sex=Male 1731 367 No (0.7879838 0.2120162) *
  3) Sex=Female 470 126 Yes (0.2680851 0.7319149)
    6) Class=Third 196 90 No (0.5408163 0.4591837) *
    7) Class=Crew,First,Second 274 20 Yes (0.0729927 0.9270073) *
```



4. R 나무모형 분석 : 타이타닉

상세한 결과

```
> summary(cart_fit2)
Call:
rpart(formula = Survived ~ Class + Age + Sex, data = titanic,
      control = my.control)
n= 2201
```

	CP	nsplit	rel error	xerror	xstd
1	0.30661041	0	1.0000000	1.0000000	0.03085662
2	0.02250352	1	0.6933896	0.6933896	0.02750982
3	0.01000000	2	0.6708861	0.7130802	0.02778312

Variable importance

Sex	Class	Age
79	20	2

Node number 1: 2201 observations, complexity param=0.3066104
predicted class=No expected loss=0.323035 P(node) =1
class counts: 1490 711
probabilities: 0.677 0.323
left son=2 (1731 obs) right son=3 (470 obs)
Primary splits:
Sex splits as RL, improve=199.821600, (0 missing)
Class splits as LRRL, improve= 69.684100, (0 missing)
Age splits as LR, improve= 9.165241, (0 missing)

4. R 나무모형 분석 : 타이타닉

상세한 결과

Node number 2: 1731 observations
predicted class=No expected loss=0.2120162 P(node) =0.7864607
class counts: 1364 367
probabilities: 0.788 0.212

Node number 3: 470 observations, complexity param=0.02250352
predicted class=Yes expected loss=0.2680851 P(node) =0.2135393
class counts: 126 344
probabilities: 0.268 0.732
left son=6 (196 obs) right son=7 (274 obs)
Primary splits:
Class splits as RRRL, improve=50.015320, (0 missing)
Age splits as RL, improve= 1.197586, (0 missing)
Surrogate splits:
Age splits as RL, agree=0.619, adj=0.087, (0 split)

Node number 6: 196 observations
predicted class=No expected loss=0.4591837 P(node) =0.08905043
class counts: 106 90
probabilities: 0.541 0.459

Node number 7: 274 observations
predicted class=Yes expected loss=0.0729927 P(node) =0.1244889
class counts: 20 254
probabilities: 0.073 0.927

5. R 회귀나무모형 분석: cu.summary 자료

데이터 읽기

```
> library(rpart)
> data(cu.summary)
> head(cu.summary)
```

	Price	Country	Reliability	Mileage	Type
Acura Integra 4	11950	Japan	Much better	NA	Small
Dodge Colt 4	6851	Japan	<NA>	NA	Small
Dodge Omni 4	6995	USA	Much worse	NA	Small
Eagle Summit 4	8895	USA	better	33	Small
Ford Escort 4	7402	USA	worse	33	Small
Ford Festiva 4	6319	Korea	better	37	Small

```
> summary(cu.summary)
```

Price	Country	Reliability	Mileage	Type
Min. : 5866	USA :49	Much worse :18	Min. :18.00	Compact:22
1st Qu.:10125	Japan :31	worse :12	1st Qu.:21.00	Large : 7
Median :13150	Germany :11	average :26	Median :23.00	Medium :30
Mean :15743	Japan/USA: 9	better : 8	Mean :24.58	Small :22
3rd Qu.:18900	Korea : 5	Much better:21	3rd Qu.:27.00	Sporty :26
Max. :41990	Sweden : 5	NA's :32	Max. :37.00	Van :10
	(Other) : 7		NA's :57	

```
> is.factor(cu.summary$Type)
[1] TRUE
```

5. R 회귀나무모형 분석: cu.summary 자료

디폴트 정지규칙을 이용한 CART 회귀나무모형

```
> # Default tree
> cu_fit = rpart(Price ~ Country + Reliability + Mileage + Type, data=cu.summary)
> print(cu_fit)
n= 117

node), split, n, deviance, yval
  * denotes terminal node

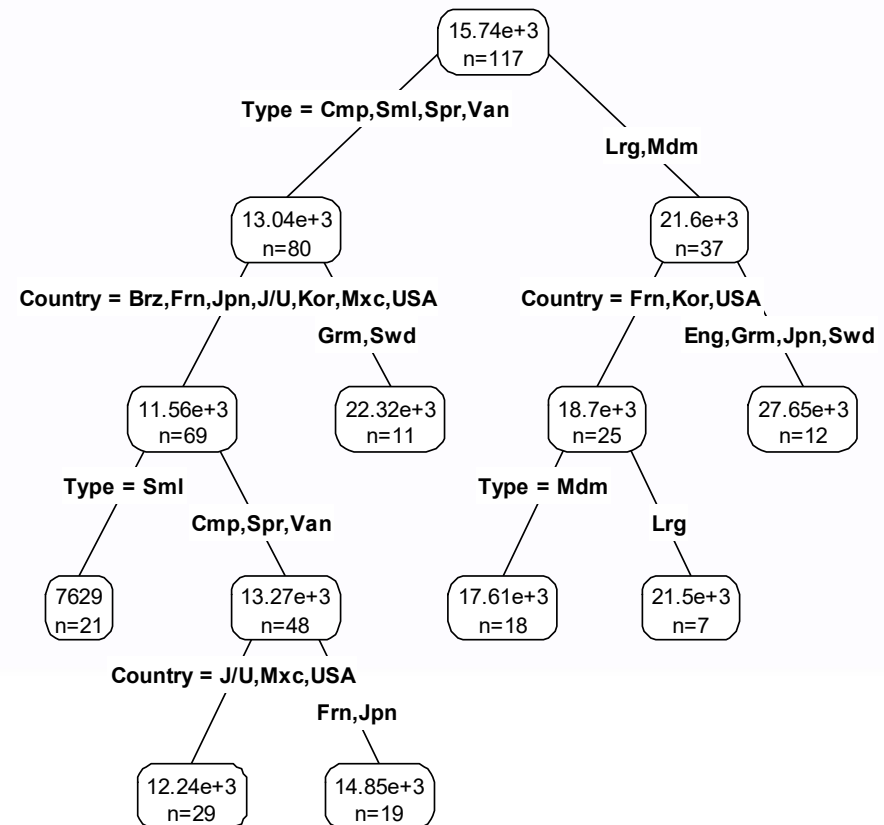
1) root 117 7407473000 15743.460
  2) Type=Compact,Small,Sporty,Van 80 3322389000 13035.010
    4) Country=Brazil,France,Japan,Japan/USA,Korea,Mexico,USA 69 1426421000 11555.160
      8) Type=Small 21 50309830 7629.048 *
      9) Type=Compact,Sporty,Van 48 910790000 13272.830
        18) Country=Japan/USA,Mexico,USA 29 482343500 12241.550 *
        19) Country=France,Japan 19 350528000 14846.890 *
    5) Country=Germany,Sweden 11 797004200 22317.730 *
  3) Type=Large,Medium 37 2229351000 21599.570
    6) Country=France,Korea,USA 25 1021102000 18697.280
      12) Type=Medium 18 741101600 17607.440 *
      13) Type=Large 7 203645100 21499.710 *
    7) Country=England,Germany,Japan,Sweden 12 558955000 27646.000 *
```

>

5. R 회귀나무모형 분석: cu.summary 자료

CART 나무모형을 도형화하는 R 명령문

```
> # Tree 그리기  
> library(rpart.plot)  
> prp(cu_fit, type=4, extra=1, digits=4)  
>
```



5. R 회귀나무모형 분석: cu.summary 자료

정지규칙을 설정한 CART 회귀나무모형

```
> # Stopping rule
> my.control = rpart.control(maxdepth=3, minsplit=30)
> cu_fit2 = rpart(Price ~ Country + Reliability + Mileage + Type,
                  control=my.control, data=cu.summary)
> print(cu_fit2)
n= 117
node), split, n, deviance, yval
  * denotes terminal node

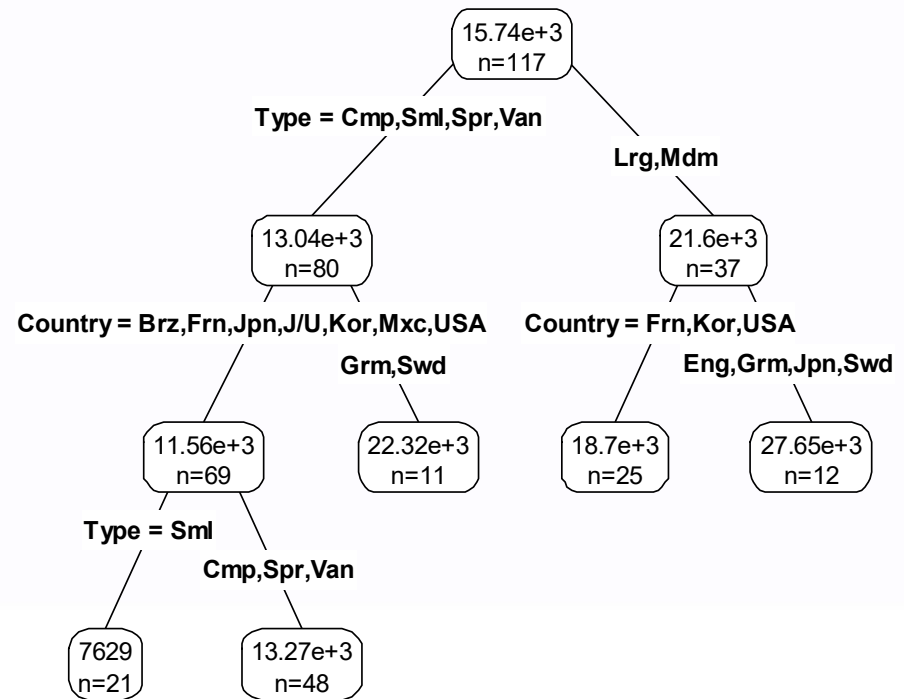
1) root 117 7407473000 15743.460
  2) Type=Compact,Small,Sporty,Van 80 3322389000 13035.010
    4) Country=Brazil,France,Japan,Japan/USA,Korea,Mexico,USA 69 1426421000
      11555.160
        8) Type=Small 21 50309830 7629.048 *
        9) Type=Compact,Sporty,Van 48 910790000 13272.830 *
    5) Country=Germany,Sweden 11 797004200 22317.730 *
  3) Type=Large,Medium 37 2229351000 21599.570
    6) Country=France,Korea,USA 25 1021102000 18697.280 *
    7) Country=England,Germany,Japan,Sweden 12 558955000 27646.000 *
```

>

5. R 회귀나무모형 분석: cu.summary 자료

CART 나무모형을 도형화하는 R 명령문

```
> # Tree 그리기  
> library(rpart.plot)  
> prp(cu_fit2, type=4, extra=1, digits=4)  
>
```



6. 파이썬 나무모형 분석: 타이타닉

데이터 읽기

```
import numpy as np
import pandas as pd
# 데이터 읽기
titanic = pd.read_csv("c:/data/mva/titanic.csv")
titanic.head(3)
```

Out[1]:

	Class	Age	Sex	Survived
0	First	Adult	Male	Yes
1	First	Adult	Male	Yes
2	First	Adult	Male	Yes

자료 (행의 수, 열의 수)

```
titanic.shape
```

Out[3]: (2201, 4)

기술 통계량 구하기

```
titanic.describe()
```

Out[2]:

	Class	Age	Sex	Survived
count	2201	2201	2201	2201
unique	4	2	2	2
top	Crew	Adult	Male	No
freq	885	2092	1731	1490

6. 파이썬 나무모형 분석: 타이타닉

빈도표 구하기

```
pd.crosstab(titanic['Survived'], titanic['Sex'], margins=True)
```

Out[4]:

Sex	Female	Male	All
Survived			
No	126	1364	1490
Yes	344	367	711
All	470	1731	2201

```
pd.crosstab(titanic['Survived'], titanic['Class'], margins=True)
```

Out[5]:

Class	Crew	First	Second	Third	All
Survived					
No	673	122	167	528	1490
Yes	212	203	118	178	711
All	885	325	285	706	2201

6. 파이썬 나무모형 분석: 타이타닉

나무모형을 적합하기 위한 데이터 변환

```
# 문자형을 이산형으로 변환
titanic['Age'] = titanic['Age'].replace({'Child':0, 'Adult':1})
titanic['Sex'] = titanic['Sex'].replace({'Male':0, 'Female':1})
titanic['Class'] = titanic['Class'].replace({'First':1, 'Second':2,
                                             'Third':3, 'Crew':4})

titanic.head(3)
Out[8]:
   Class  Age  Sex  Survived
0      1    1    0        Yes
1      1    1    0        Yes
2      1    1    0        Yes

# X 데이터와 y 데이터
X = titanic[["Class", "Age", "Sex"]]
y = titanic["Survived"]
```

6. 파이썬 나무모형 분석: 타이타닉

CART 나무모형 실행

```
# 나무모형 생성
from sklearn.tree import DecisionTreeClassifier
titanic_tree = DecisionTreeClassifier(max_depth=3, min_samples_split=50)
titanic_tree.fit(X, y)
Out[10]: DecisionTreeClassifier(max_depth=3, min_samples_split=50)
```

6. 파이썬 나무모형 분석: 타이타닉

CART 나무모형을 이용한 분류

```
# 적합된 나무모형을 이용한 분류
y_pred = titanic_tree.predict(X)

# Confusion Matrix
from sklearn.metrics import classification_report, confusion_matrix
cm = confusion_matrix(y, y_pred)
print(cm)
[[1470  20]
 [ 441 270]]

# 기타 분류 성능 지표
cm_report = classification_report(y, y_pred)
print(cm_report)
```

	precision	recall	f1-score	support
No	0.77	0.99	0.86	1490
Yes	0.93	0.38	0.54	711
accuracy			0.79	2201
macro avg	0.85	0.68	0.70	2201
weighted avg	0.82	0.79	0.76	2201

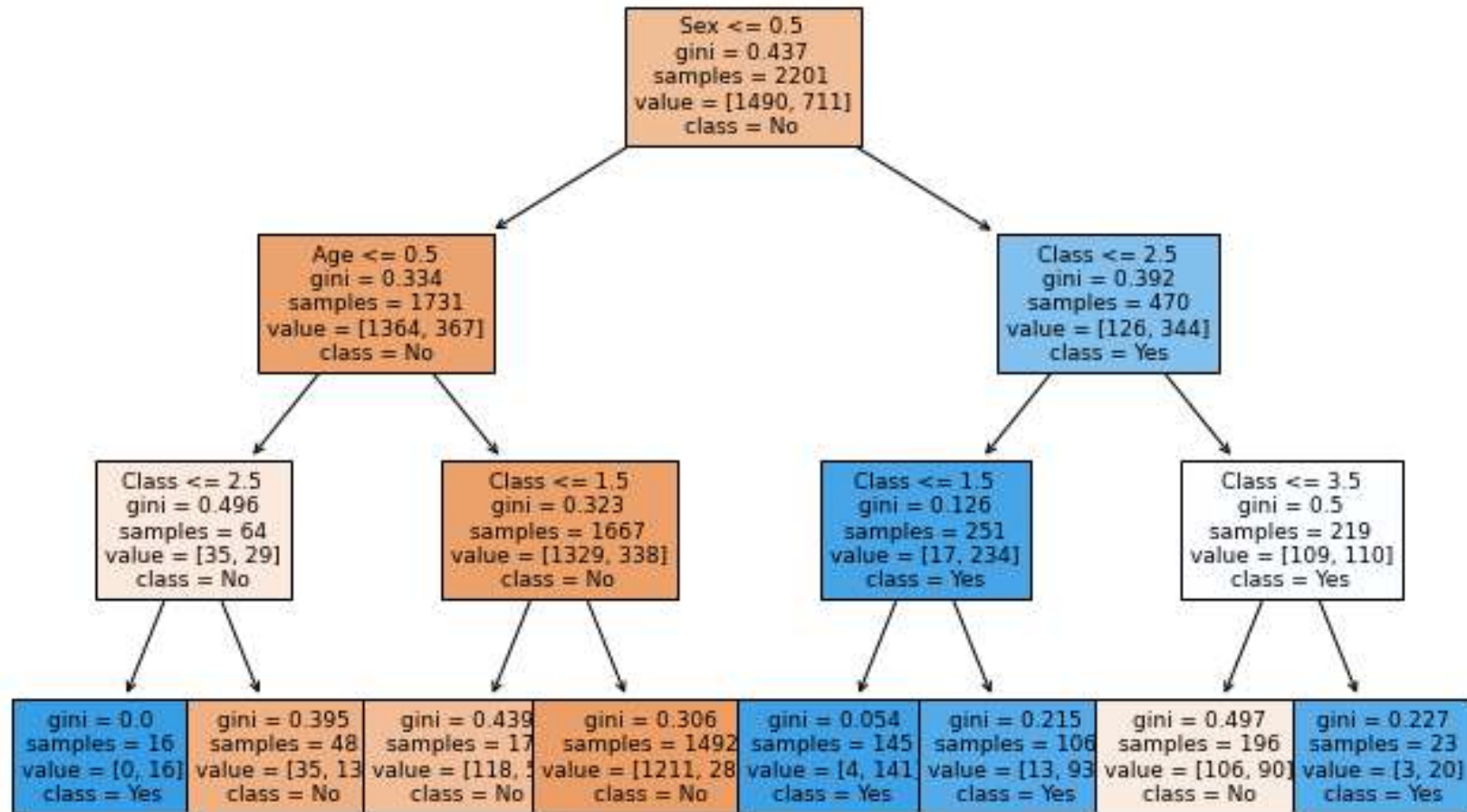
6. 파이썬 나무모형 분석: 타이타닉

나무모형 그리기

```
import matplotlib.pyplot as plt

# Tree 그리기
from sklearn.tree import plot_tree
plt.figure(figsize=(11,7))
plot_tree(titanic_tree, feature_names=X.columns,
          class_names=['No','Yes'], filled=True, fontsize=9)
plt.show()
```

6. 파이썬 나무모형 분석: 타이타닉



7. 파이썬 회귀나무: cusummary

데이터 읽기

```
import numpy as np
import pandas as pd
# 데이터 읽기
cu = pd.read_csv("c:/data/mva/cusummary.csv", index_col='Model')
```

```
# 결측값 케이스 없애기
```

```
cu = cu.dropna()
```

```
cu.shape
```

```
Out[18]: (49, 5)
```

```
cu.head(3)
```

```
Out[19]:
```

		Price	Country	Reliability	Mileage	Type
Model						
Eagle Summit	4	8895	USA	better	33.0	Small
Ford Escort	4	7402	USA	worse	33.0	Small
Ford Festiva	4	6319	Korea	better	37.0	Small

7. 파이썬 회귀나무: cusummary

범주형 변수의 가변수 만들기

```
# X 데이터와 y 데이터
X = cu.drop('Price', axis=1)
y = cu['Price']
# 이산변수 혹은 가변수(dummy variable) 만들기
X['Reliability'] = X['Reliability'].replace({'Much worse':1, 'worse':2,
'average':3, 'better':4, 'Much better':5})
dX = pd.get_dummies(data=X, drop_first=True)
dX.head()
Out[20]:
```

	Reliability	Mileage	...	Type_Sporty	Type_Van
Model			...		
Eagle Summit 4	4	33.0	...	0	0
Ford Escort 4	2	33.0	...	0	0
Ford Festiva 4	4	37.0	...	0	0
Honda Civic 4	5	32.0	...	0	0
Mazda Protege 4	5	32.0	...	0	0

```
[5 rows x 13 columns]

# 변수 이름 보기
dX.columns
Out[21]:
Index(['Reliability', 'Mileage', 'Country_Japan', 'Country_Japan/USA',
'Country_Korea', 'Country_Mexico', 'Country_Sweden', 'Country_USA',
'Type_Large', 'Type_Medium', 'Type_Small', 'Type_Sporty', 'Type_Van'],
dtype='object')
```

7. 파이썬 회귀나무: cusummary

회귀나무 실행

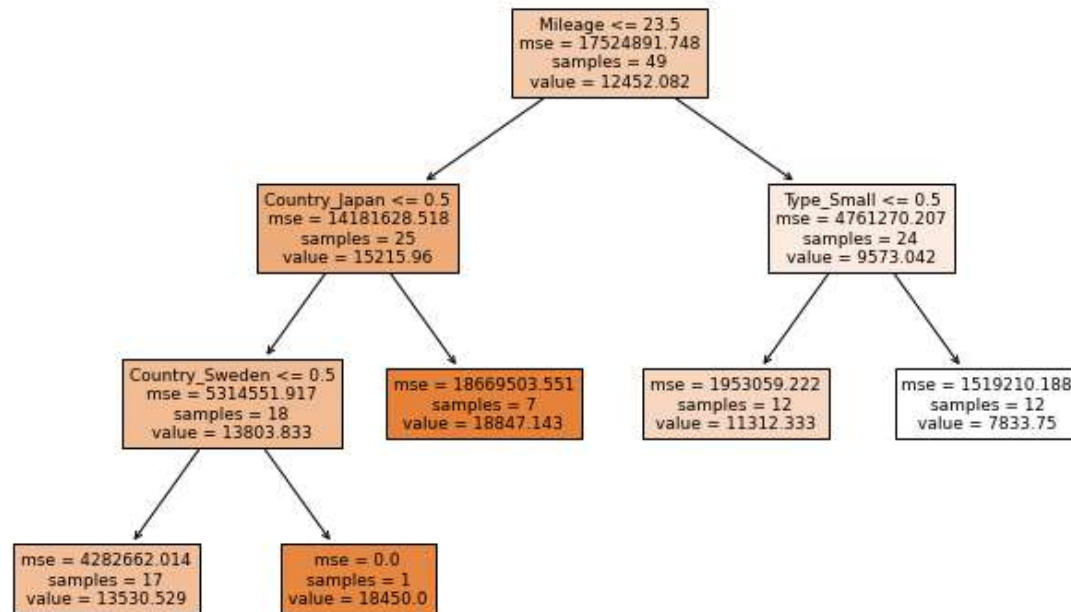
```
# 회귀나무모형 생성
from sklearn.tree import DecisionTreeRegressor
cu_tree = DecisionTreeRegressor(max_depth=3, min_samples_split=15)
cu_tree.fit(dX, y)

# 추정값 구하기
y_pred = cu_tree.predict(dX)
y_pred[0:5]
Out[22]: array([7833.75, 7833.75, 7833.75, 7833.75, 7833.75])
```

7. 파이썬 회귀나무: cusummary

Tree 그리기

```
# Tree 그리기
from sklearn.tree import plot_tree
plt.figure(figsize=(11,7))
plot_tree(cu_tree, feature_names=dX.columns, filled=True, fontsize=9)
plt.show()
```



다음시간에는

15강 다변량분석 총정리

 수고했습니다.