11주차:
분류를 위한 모델

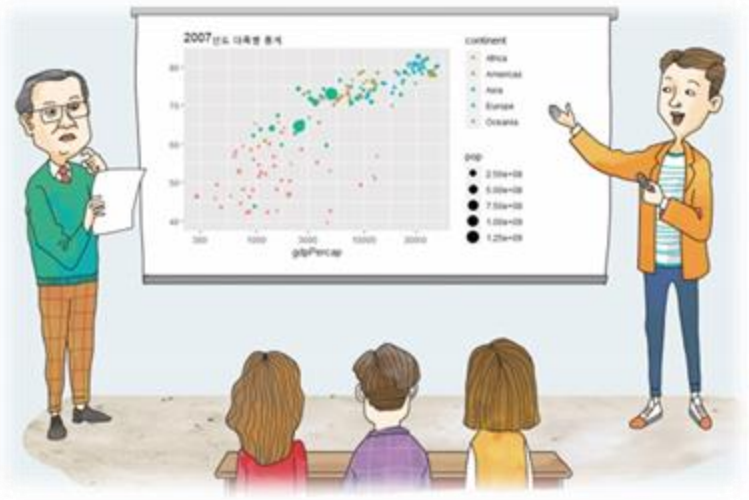
ChulSoo Park

School of Computer Engineering & Information Technology

Korea National University of Transportation

09 CHAPTER

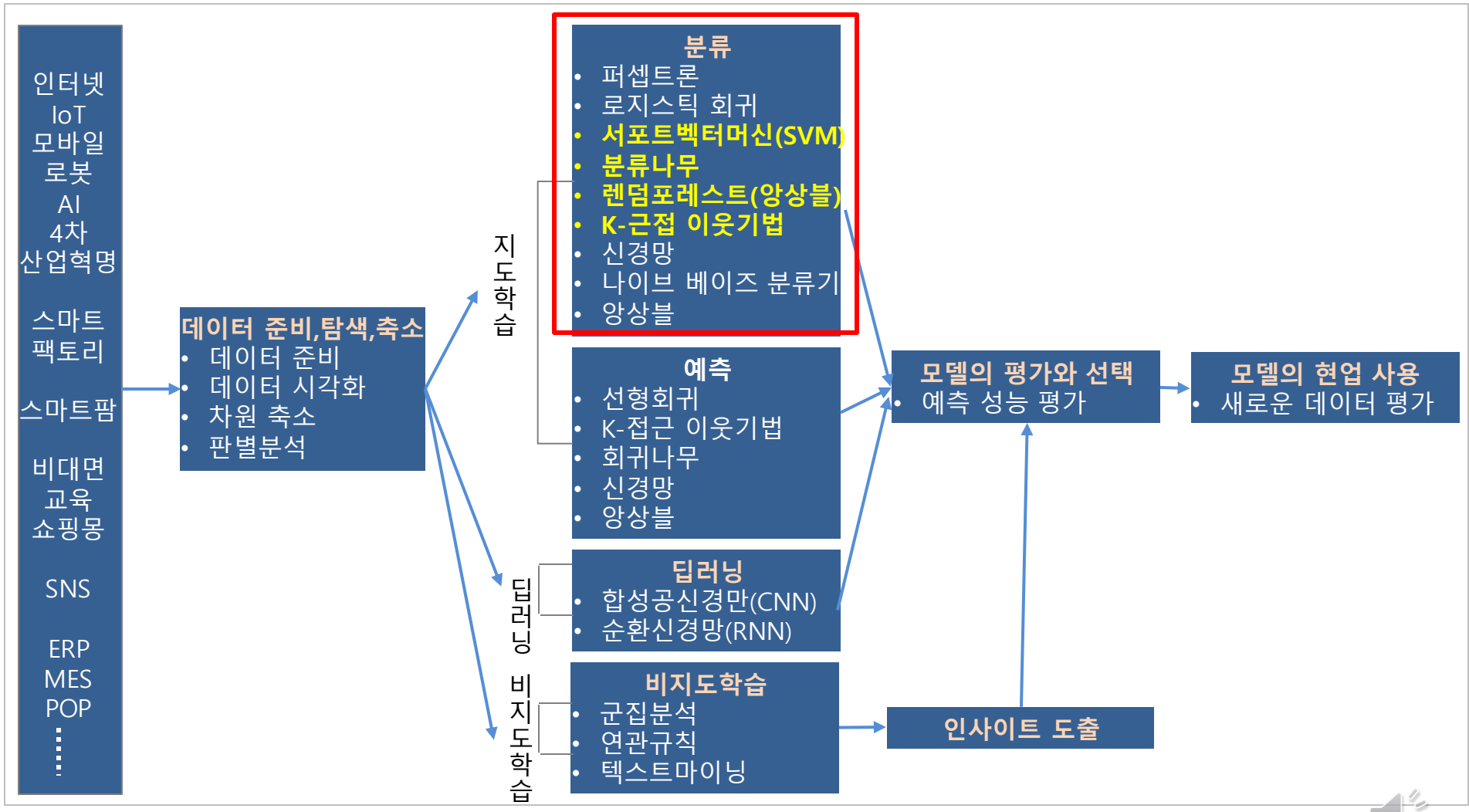
분류를 위한 모델



CONTENTS

9.1 회귀와 분류
9.2 결정 트리의 원리
9.3 결정 트리 함수의 사용
9.4 결정 트리의 해석
9.5 랜덤 포리스트
9.6 SVM과 k-NN
9.7 분류 모델의 다양한 적용
요약

9.3 결정 트리 함수의 사용

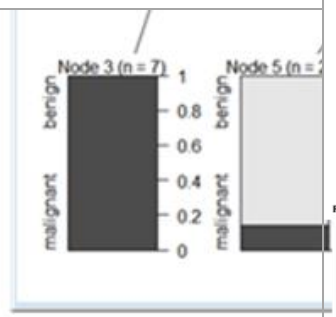


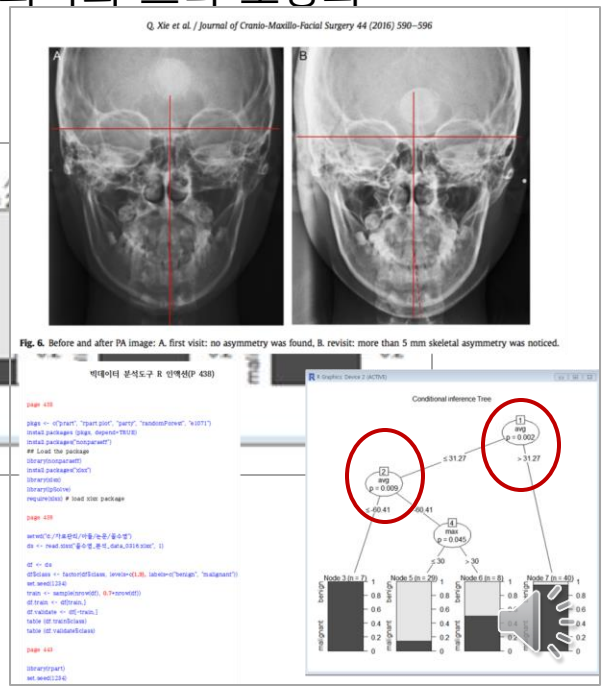
9.3 결정 트리 함수의 사용

결정 트리의 학습

- 우리에게 주어지는 것은 결정 트리가 아니라 **훈련 집합(data)← 문제**
- 훈련 집합을 가지고 결정 트리를 어떻게 학습하나? → 기계학습 연구자들이 학습 알고리즘을 개발해 놓았고 R은 충실히 구현하여 rpart 함수를 제공
- rpart는 훈련 집합을 최소 오류로 분류하는 결정 트리, 즉 최적의 트리 모양과 노드의 질문을 만들어 줌

```
df <- ds
df$class <- factor(df$class, levels=c(1,9), labels=c("benign", "malignant"))
set.seed(1234)
train <- sample(nrow(df), 0.7*nrow(df))
df.train <- df[train,]
df.validate <- df[-train,]
table (df.train$class)
table (df.validate$class)

page 443

library(rpart)
set.seed(1234)
```





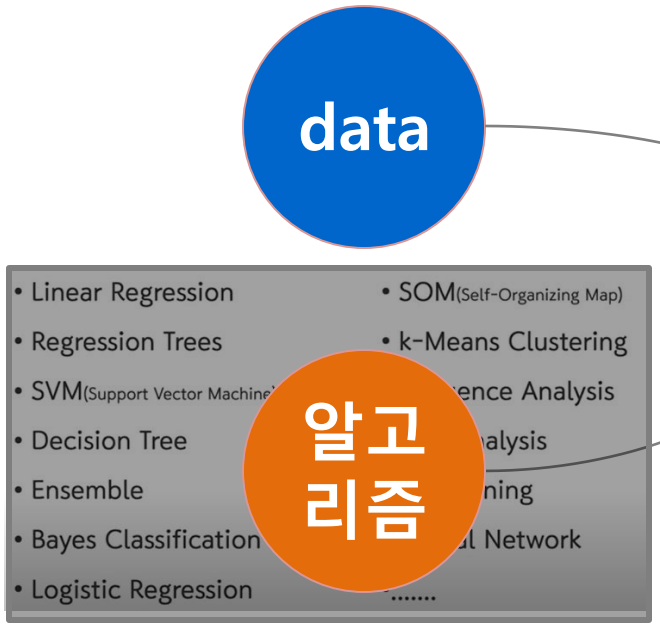
또다른 결정 트리 함수 : tree, party

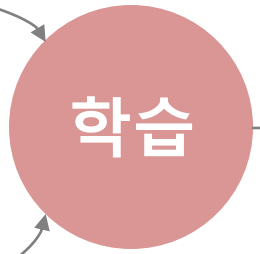
9.3 결정 트리 함수의 사용

알고리즘(Algorithm)과 모델

알고리즘(Algorithm)은 수학과 컴퓨터 과학, 언어학 또는 관련 분야에서 어떠한 문제를 해결하기 위해 정해진 일련의 절차나 방법을 공식화한 형태로 표현한 것. 출처 : 위키백과

어떤 문제를 해결하기 위한 절차, 방법, 명령어들의 집합. 출처 : 네이버 지식





y = ax + b

y = 2x + 5

+

9.3 결정 트리 함수의 사용

- iris 데이터에 rpart 적용하기
  - iris data 확인

```
Console  C:/RSources/
> head(iris,120)
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           5.1         3.5          1.4         0.2     setosa
2           4.9         3.0          1.4         0.2     setosa
3           4.7         3.2          1.3         0.2     setosa
4           4.6         3.1          1.5         0.2     setosa
5           5.0         3.6          1.4         0.2     setosa
  생략……
51          7.0         3.2          4.7         1.4 versicolor
52          6.4         3.2          4.5         1.5 versicolor
53          6.9         3.1          4.9         1.5 versicolor
54          5.5         2.3          4.0         1.3 versicolor
55          6.5         2.8          4.6         1.5 versicolor
  생략……
101         6.3         3.3          6.0         2.5  virginica
102         5.8         2.7          5.1         1.9  virginica
103         7.1         3.0          5.9         2.1  virginica
104         6.3         2.9          5.6         1.8  virginica
105         6.5         3.0          5.8         2.2  virginica
106         7.6         3.0          6.6         2.1  virginica
107         4.9         2.5          4.5         1.7  virginica
108         7.3         2.9          6.3         1.8  virginica
109         6.7         2.5          5.8         1.8  virginica
110         7.2         3.6          6.1         2.5  virginica
```

```
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

- 분류에서 반응변수는 **범주형** 이어야함.
- 범주형이 아닌 경우 **rpart(……. method='class')** 옵션을 설정
- 그렇지 않으면 분류가 아닌 회귀로 작동됨

9.3 결정 트리 함수의 사용

rpart 함수 활용

rpart(formula, data, weights, subset, na.action = na.rpart, **method**,
model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)

`method` one of `"anova"`, `"poisson"`, `"class"` or `"exp"`. If `method` is missing then the routine tries to make an intelligent guess. If `y` is a survival object, then `method = "exp"` is assumed, if `y` has 2 columns then `method = "poisson"` is assumed, if `y` is a factor then `method = "class"` is assumed, otherwise `method = "anova"` is

반응변수와 설명변수

```
Console C:/RSources/
> library(rpart)
> rpart_iris = rpart(Species~., data = iris)
> printcp(rpart_iris)

Classification tree:
rpart(formula = Species ~ ., data = iris)

Variables actually used in tree construction:
[1] Petal.Length Petal.Width

Root node error: 100/150 = 0.66667

n= 150

    CP nsplit rel error xerror     xstd
1 0.50      0      1.00   1.20 0.048990
2 0.44      1      0.50   0.66 0.060795
3 0.01      2      0.06   0.08 0.027520
```

사용 data set

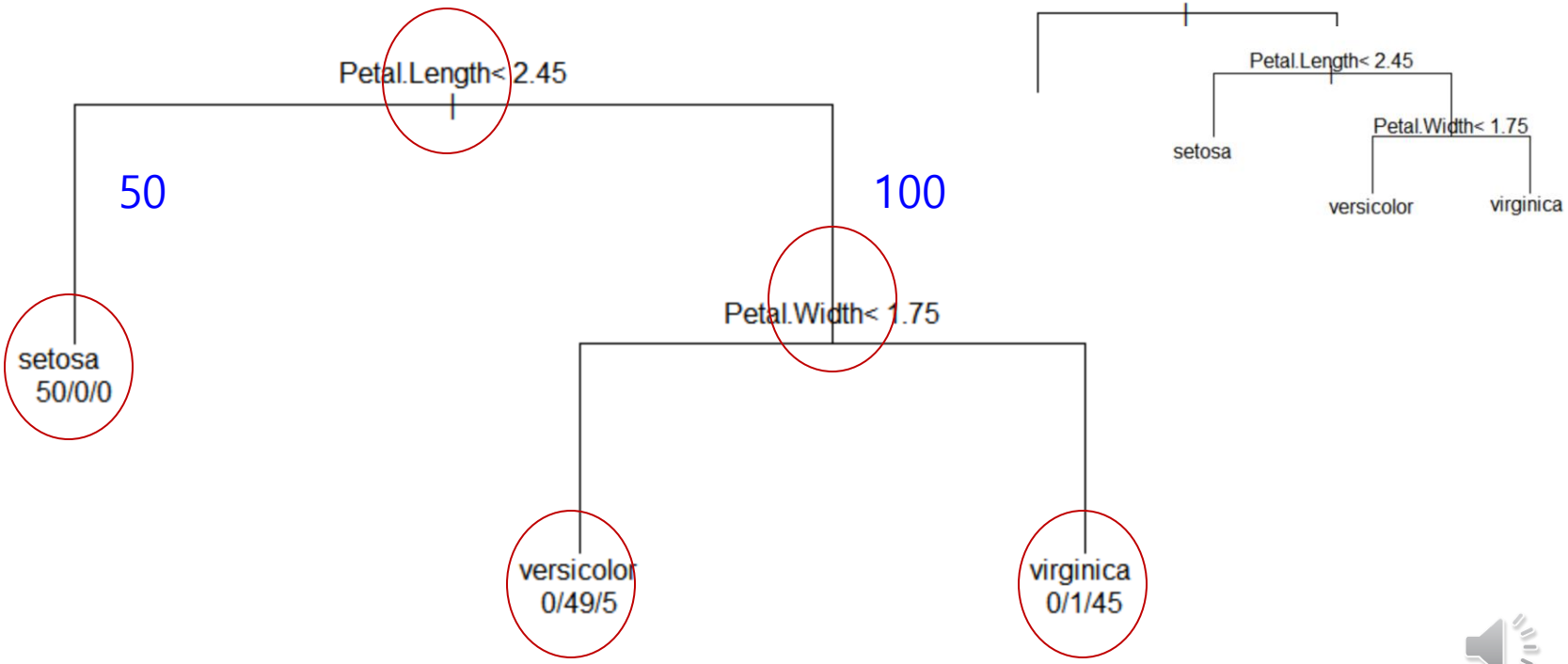
printcp : 결정 트리의 구체적인 내용을 알려주는 함수

9.3 결정 트리 함수의 사용

결정 Tree의 시각화

```
> par(mfrow = c(1, 1), xpd = NA)
> plot(rpart_iris)
> text(rpart_iris, use.n = TRUE)
```

xpd : FALSE=plot region 내부에 그림 출력. TRUE=figure region 내부에 그림 출력.
NA=(그림이 클 경우) 그래픽 장치 내부에 그림 출력.



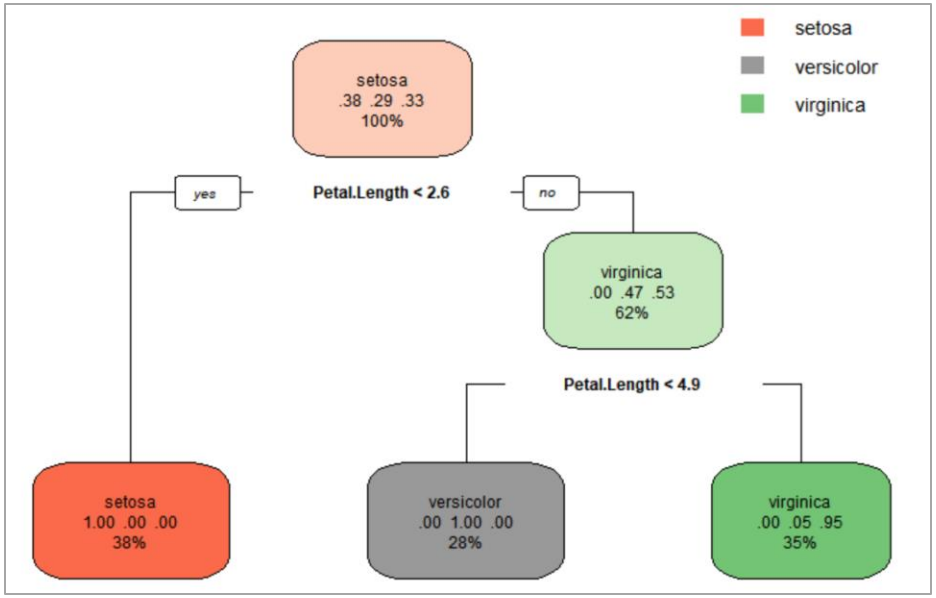
9.3 결정 트리 함수의 사용

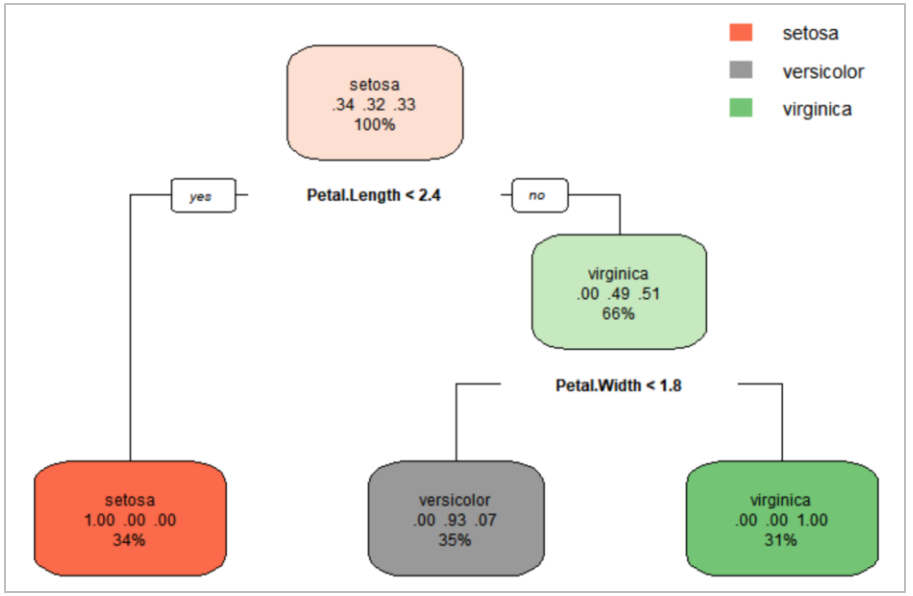
rpart 패키지의 rpart() 함수의 입력인자

formula	함수식
data	데이터
weights	가중치
subset	그룹변수
na.action	결측값 처리
method	분석방법 : ("anova", "poisson", "class", "exp", "dist", "mrt", "user")
model	모형
x	독립변수
y	종속변수
parms	매개변수
control	통제조건
cost	비음조건

9.3 결정 트리 함수의 사용

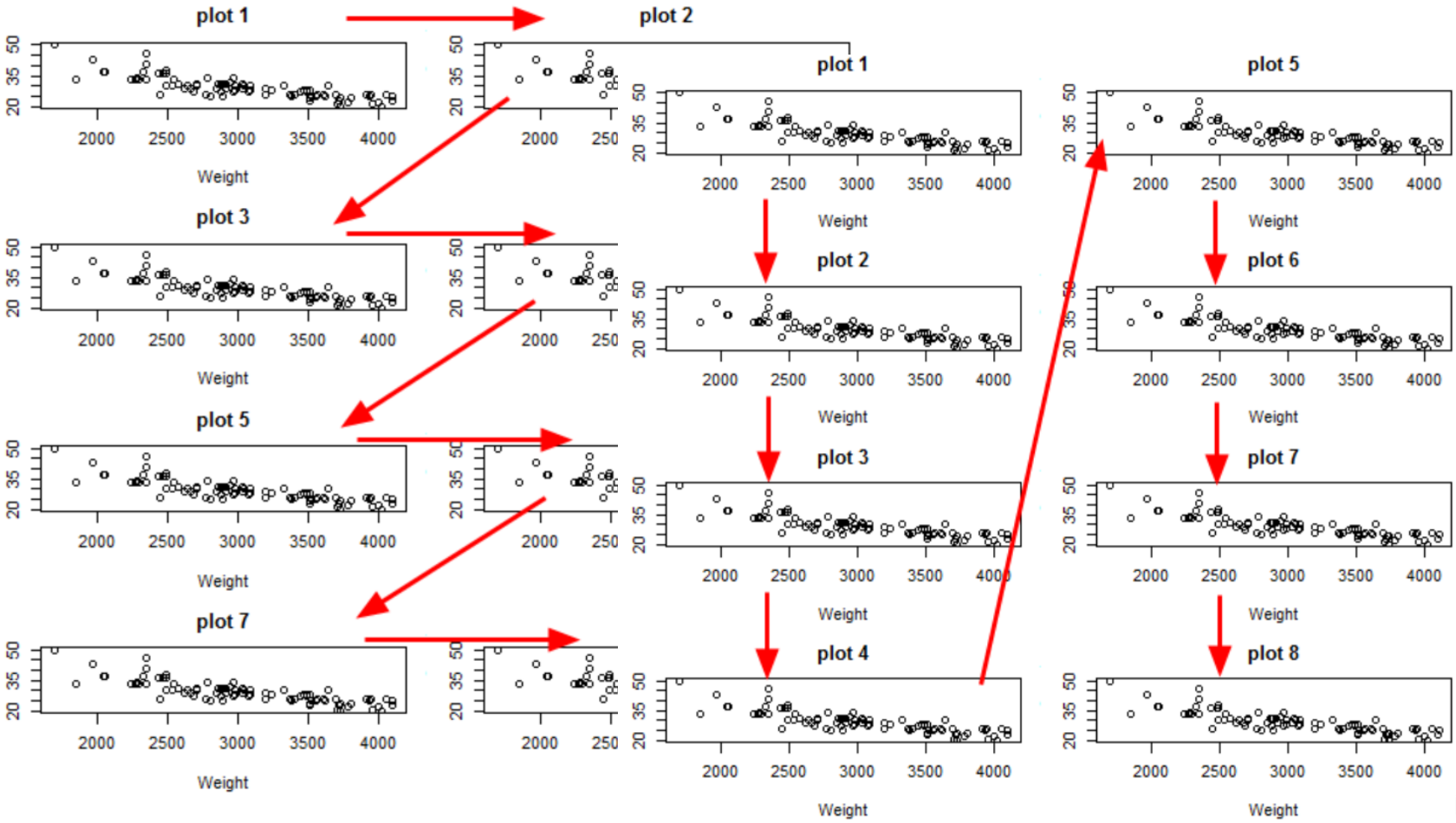
```
> iris_train <- sample(1:150, 120) #무작위로 120개 추출 (학습데이터)
> tree <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length +
+                         Petal.Width, data=iris, subset =iris_train, method = "class")
> rpart.plot(tree)
```





9.3 결정 트리 함수의 사용

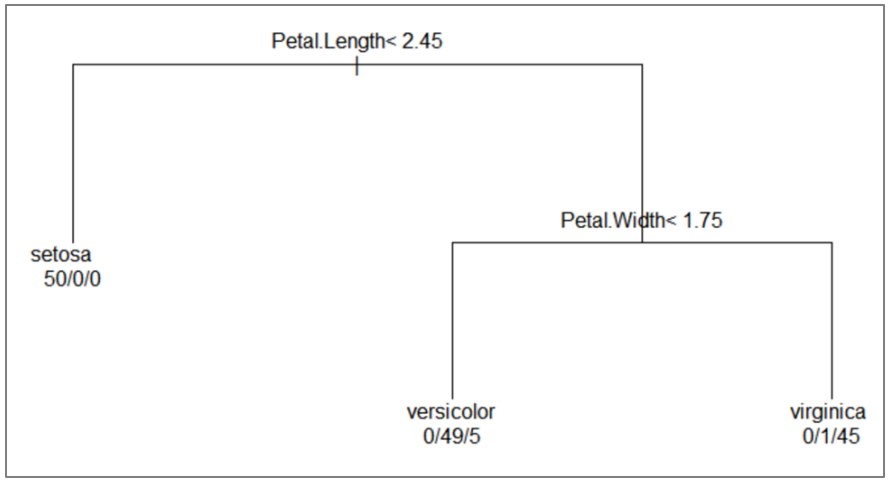
```
par(mfrow=c(4, 2), # make window by 4 row, 2 columns
par(mfcol=c(4, 2)
```



9.3 결정 트리 함수의 사용

결정 트리의 해석

- 루트 노드의 질문 [Petal.Length<2.45?]
  - ✓ 150개 샘플 중에서 (예)인 50개는 왼쪽, 아니오의 100개는 오른쪽으로 이동
  - ✓ 왼쪽 50개는 모두 setosa에 속하므로 멈춤 → 리프를 설정하고 setosa(50/0/0) 기록



- 루트의 오른쪽 자식 노드의 질문 [Petal.Width<1.75?]
  - ✓ 100개 샘플 중에서 (예)인 54개는 왼쪽, (아니오)의 46개는 오른쪽으로 이동
  - ✓ 왼쪽 54개에는 49개의 versicolor, 5개의 virginica → 과잉적합을 방지하기 위해 versicolor(0/49/5)로 지정하고 멈춤
  - ✓ 오른쪽 46개에는 1개의 versicolor, 45개의 virginica → 과잉적합을 방지하기 위해 virginica(0/1/45)로 지정하고 멈춤

9.3 결정 트리 함수의 사용

훈련 집합에 대한 예측

- predict 함수를 사용하여 예측
- type='class' 옵션은 부류를 출력함 (이 옵션이 없으면, type='prob'가 기본 값이므로 부류에 속할 확률을 출력함)

```
Console  C:/RSources/
> p_iris = predict(rpart_iris, iris, type = 'class')
> table(p_iris, iris$Species)

p_iris        setosa versicolor virginica
  setosa          50          0         0
  versicolor       0         49         5
  virginica        0          1        45
```

그라운드 트루스 (정답)

예측 값

9.3 결정 트리 함수의 사용

- 혼동 행렬confusion matrix
  - 부류별로 옳은 분류와 잘못된 분류의 상세 내용을 보여줌
  - 예를 들어, 50개의 versicolor를 49개는 옳게, 1개는 virginica로 잘못 분류함
  - 혼동 행렬은 10장 6절에서 자세히 다룸

```
Console  C:/RSources/
> p_iris = predict(prart_iris, iris, type = 'class')
> table(p_iris, iris$Species)

p_iris        setosa versicolor virginica
  setosa          50          0         0
  versicolor       0         49         5
  virginica        0          1        45
```

그라운드 트루스(정답)

예측

혼돈 행렬

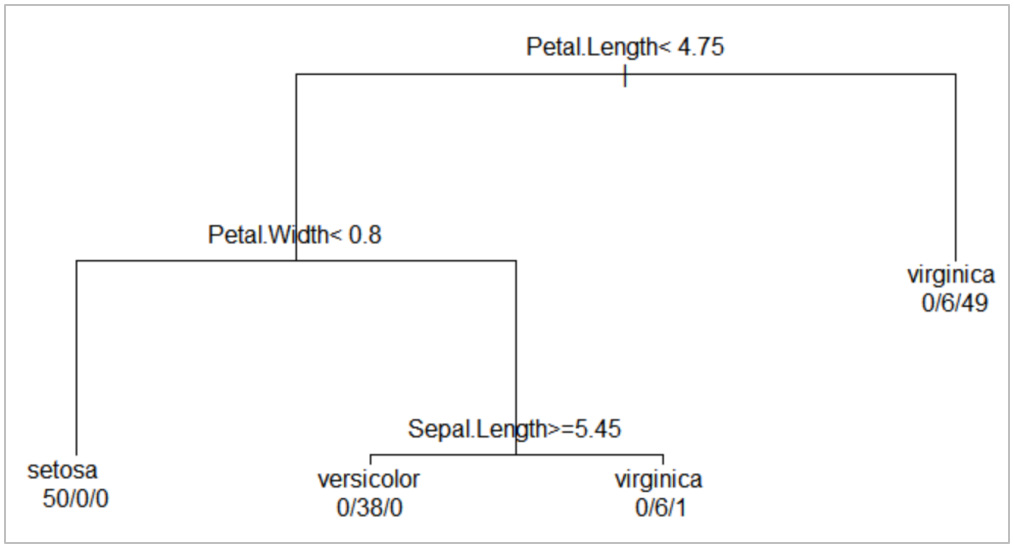
9.3 결정 트리 함수의 사용

- 사전 확률을 지정하는 옵션 예시
  - 사전 확률(prior probability)
    - 부류별 발생 확률
    - 예) 들판을 조사하여 setosa:versicolor:virginica의 발생 확률이 10:10:80이라는 사실을 알아냈다고 가정
  - 무턱대고 virginica라고 분류해도 80%의 정확률 확보 ← 기본 정확률이 80%
    - 사전 확률이 없다면 33.333%가 기본 정확률
    - 높은 성능을 얻는데 사전 확률은 중요한 정보임

9.3 결정 트리 함수의 사용

- 사전 확률을 지정한 rpart 사용 예
  - 7개 노드가 생성됨 (4개의 리프 노드)
  - 0/6/1을 가진 리프 노드는 versicolor가 더 많음에도 불구하고 virginica로 지정함 ← virginica의 사전 확률이 높기 때문

```
> r_prior = rpart(Species~., data = iris, parms = list(prior = c(0.1, 0.1, 0.8)))
> plot(r_prior)
> text(r_prior, use.n = TRUE)
```



9.3 결정 트리 함수의 사용

예측해 보기

- 훈련 집합에 있는 샘플을 약간 변형하여 새로운 샘플로 활용
- 확률을 출력함
  - ✓ 예를 들어, 두 번째 샘플은 versicolor일 확률 0.907, virginica일 확률 0.093

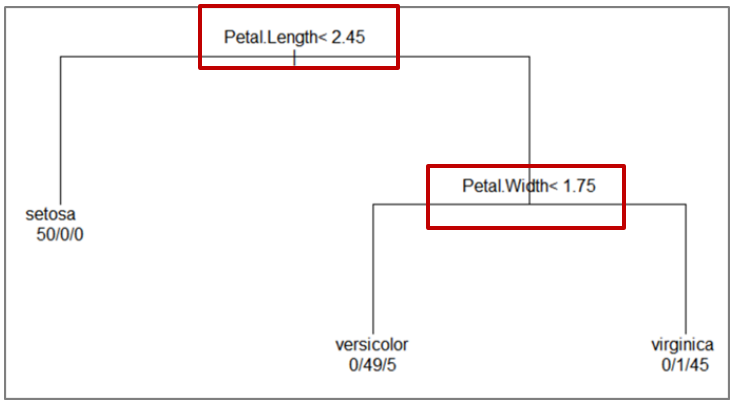
```
Console  C:/RSources/
> newd = data.frame(Sepal.Length = c(5.11, 7.01, 6.32), Sepal.Width = c(3.51, 3.2, 3.3
1), Petal.Length = c(1.4, 4.71, 6.02), Petal.Width = c(0.19, 1.4, 2.49))
> print(newd)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1         5.11        3.51         1.40        0.19
2         7.01        3.20         4.71        1.40
3         6.32        3.31         6.02        2.49
> predict(rpart_iris, newdata = newd)
  setosa versicolor  virginica
1      1 0.00000000 0.00000000
2      0 0.90740741 0.09259259
3      0 0.02173913 0.97826087
```

9.4 결정 트리의 해석

summary 함수로 결정 트리 해석하기

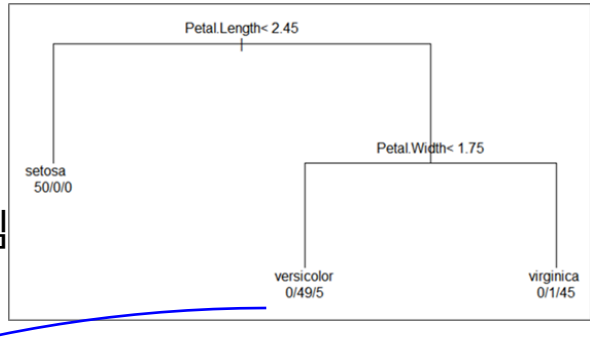
- Variable importance는 설명 변수의 중요성을 순서대로 보여줌
- 학습을 할 때 특징 선택을 동시에 수행한 셈
- [아래 그림]의 결정 트리는 4개 설명 변수 중에서 중요도가 높은 Petal.Length와 Petal.Width만 사용함

```
Console  C:/RSources/
> summary(rpart_iris)
Call:
rpart(formula = Species ~ ., data = iris)
  n= 150

    CP nsplit rel error xerror       xstd
1 0.50      0      1.00   1.20 0.04898979
2 0.44      1      0.50   0.66 0.06079474
3 0.01      2      0.06   0.08 0.02751969

Variable importance
 Petal.Width Petal.Length Sepal.Length  Sepal.Width
          34           31           21           14
```



9.4 결정 트리의 해석

summary 함수로 결정 트리 해석하기



- 노드 1, 2, 3, 6, 7에 대한 상세한 내용([그림 9-3]의 결정 트리)
  - 루트를 0, 그 아래를 1, 2, 그 아래를 3, 4, 5, 6으로 번호를 매김
- 예) 노드 6
  - 54개 샘플이 도달하고, versicolor로 분류하고, 분류 확률은 (0.0,0.907,0.093)이라는 정보

생략……

```
Node number 6: 54 observations
  predicted class=versicolor  expected loss=0.09259259  P(node) =0.36
    class counts:     0    49     5
   probabilities: 0.000 0.907 0.093

Node number 7: 46 observations
  predicted class=virginica   expected loss=0.02173913  P(node) =0.3066667
    class counts:     0     1    45
   probabilities: 0.000 0.022 0.978
```

=54/150

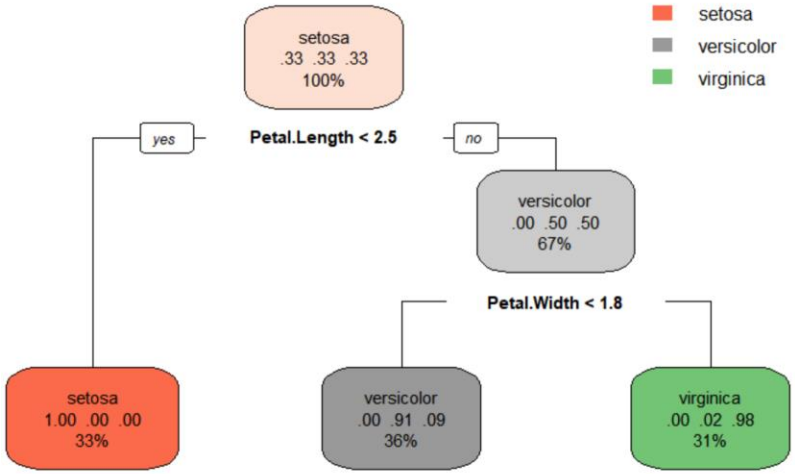
=5/54

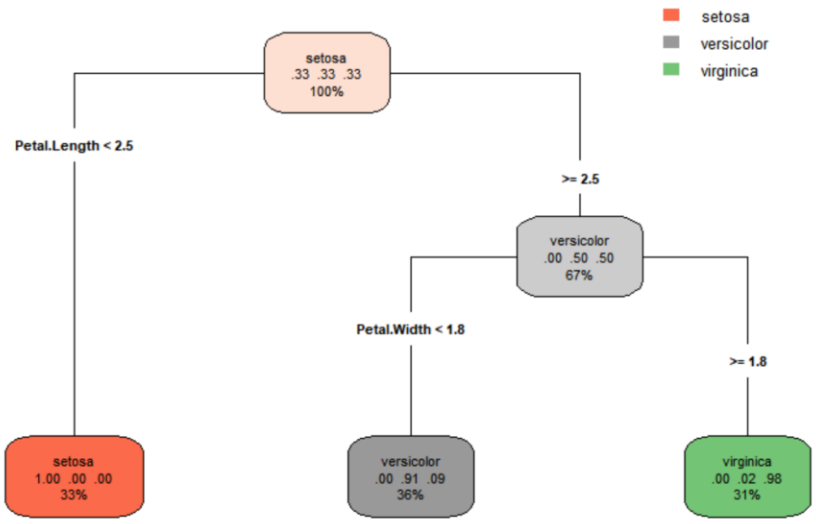
9.4 결정 트리의 해석

결정 트리 시각화하기

- 효과적인 시각화는 결정 트리를 해석하는데 크게 도움이 됨
- rpart.plot 라이브러리

```
32  install.packages("rpart.plot")
33  library(rpart.plot)
34  rpart.plot(rpart_iris)
35  rpart.plot(rpart_iris, type = 4)
36
```





9.4 결정 트리의 해석

- 결정 트리의 장점과 단점
  - 단점: 성능이 낮음
  - 장점
    - 해석 가능성 (예를 들어, "꽃받침 길이가 2.54보다 크고, 꽃받침 너비가 1.75보다 작아 versicolor로 분류"했다는 해석을 내놓을 수 있음)
    - 예측이 빠름 (몇 번의 비교 연산으로 분류)
    - 앙상블 기법을 사용하면 높은 성능 (랜덤 포리스트)
    - 결측값 처리 가능 (노드마다 대리 질문을 만들어 두고, NA인 경우 대리 질문 사용)
    - 범주형 변수를 그대로 사용할 수 있음 (예를 들어, [gender=여자?])
- 높은 성능이 필요한 경우에는 여러 결정 트리의 의사 결정을 결합하는 랜덤 포리스트를 사용

Thank you


Africa
Americas
Asia
Europe
Oceania
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09
gdpPercap