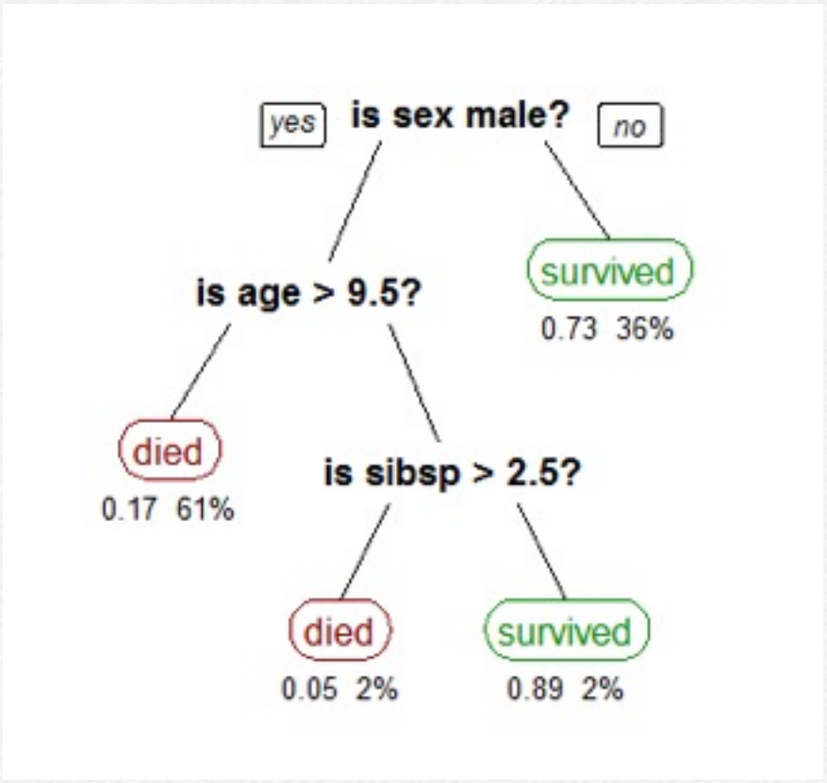
Decision Tree

의사결정나무 decision tree 개념

- 목표변수 범주를 가장 잘 분류하는 예측변수들로 분류해 가는 방법
- 목표변수는 일반적으로 이진형, (넓게는 범주형)
- 예측변수는 혼합형 가능(범주형+측정형)



타이타닉, 생존자 판별 - 성별, 나이(age), 탑승자 중 가족, 친구 관계인 승객 수

- 노드 : 각 노드는 분할(split)c기준에 의해 결정
- 계층적 노드 순서는 목표변수를 가장 잘 분류하는 변수부터 시작
- 1번 노드 성별 - 여자 생존 확률은 36%, 남자는 제2 노드로 감
- 2번 노드 나이 -9.5세 이상이면 사망으로 분류 - 즉 남자&9.5이상인 집단은 61%사망
- 3번 노드 가족친지 수 - 남자, 9.5세 이하, 가족 수 2.5이상이면 사망 확률 2%, 남자, 9.5세 이하, 가족 수 2.5이하이면 2% 생존

노드 변수 결정

- 범주형이면 교차분석 F-통계량 Classification tree
- 측정형이면 범위의 값을 이분화 하여 교차분석을 반복하면서 F통계량을 가장 크게 하는 설명변수 값을 찾는다. - Regression tree

노드 수 결정

- 분리기준 split criterion, 정지 규칙 stopping rule
- 가지치기 : 오분류 가능성이 높아지는 경우 노드를 정지하게 됨
- 그러면 언제까지 나무모형을 성장시킬 것인가 ? 너무 큰 나무모형은 자료를 과대적합 하고 반대로 너무 작은 나무모형은 자료를 과소적합할 위험
- 일반적으로 사용되는 방법은 마디에 속하는 자료 가 일정 수(가령 5) 이하일 때 분할을 정지 (thumb rule)

유명한 데이터 의사결정나무

1) Kyphosis 사례 - RPART 패키지 이용

목표변수 : 소아 수술 후 Kyphosis(기형) 존재여부

예측변인 : 나이(개월) age, 수술 필요한 척추골 수(number), 수술한 첫 척주골 개수 (start)

```
library(rpart)
table(kyphosis$Kyphosis) #target var. table
fit = rpart(Kyphosis ~ Age+ Number + Start, data = kyphosis,
            method="anova")
plot(fit, uniform=TRUE,
     main="Classification Tree for Kyphosis")
text(fit, use.n = TRUE)
summary(fit)  #summary
fit #node classification
```

- table() - 목표변수의 범주 빈도분석 absent=64, present=17 (총 81)

```
absent present
    64      17
```

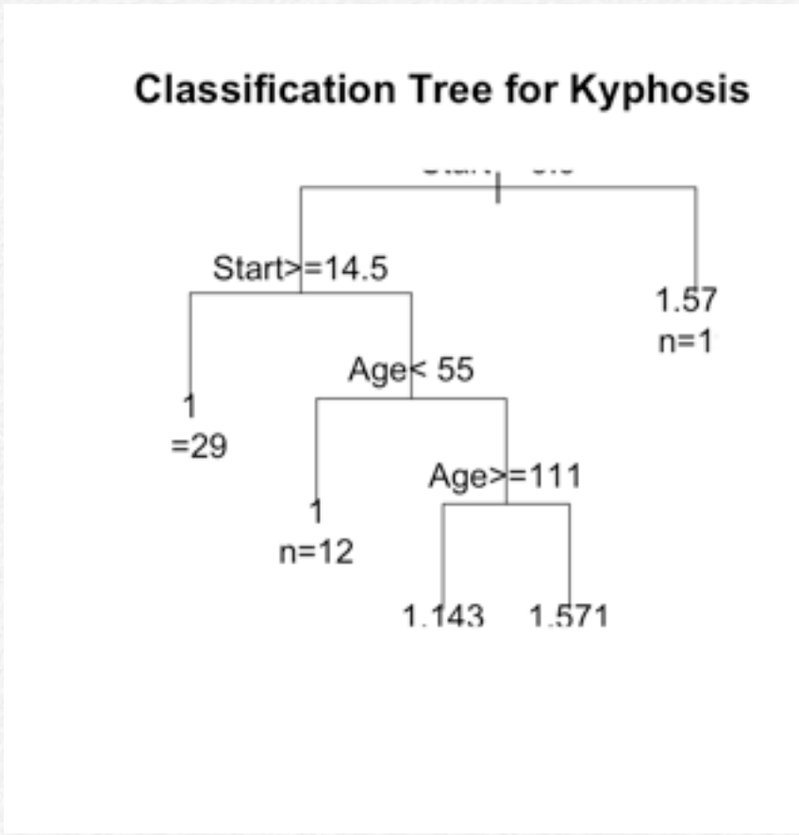
- method - “anova” 회귀분석나무 “class” 분류나무 방법
- summary() : 분석결과
- fit에는 노드 요약 결과 출력 (오른쪽 그림 참고)

(결과 해석) - 그림과 노드요약 이용 (“anova 방법”)

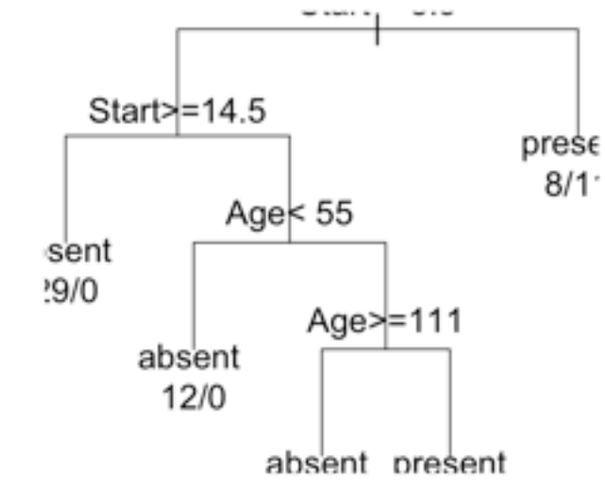
- 노드 1 : ST>=8.5 - absent로 분류(62), 그러므로 <8.5인 개체(n=19 소아)는 present로 분류
- 노드 2 : ST>=14.5 - absent로 분류(29명 소아)
- 노드 3 : 8.5<ST<14.5 & (나이 <55개월) - absent 분류(12명)

(결과 해석) - 그림과 노드요약 이용 (“class 방법”)

- 노드 1 : ST>=8.5 - absent로 분류(62), 그러므로 <8.5인 개체(n=19 소아)는 present로 분류 (그중 8개 absent-오분류, 11개 present)
- 노드 2 : ST>=14.5 - absent로 분류(29명 소아)



(anova)



(class)

```
> fit #node classification
n= 81

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

 1) root 81 13.432100 1.209877
   2) Start>=8.5 62  5.419355 1.096774
     4) Start>=14.5 29  0.000000 1.000000 *
     5) Start< 14.5 33  4.909091 1.181818
      10) Age< 55 12  0.000000 1.000000 *
      11) Age>=55 21  4.285714 1.285714
        22) Age>=111 14  1.714286 1.142857 *
        23) Age< 111 7  1.714286 1.571429 *
   3) Start< 8.5 19  4.631579 1.578947 *
```

(anova)

(class)

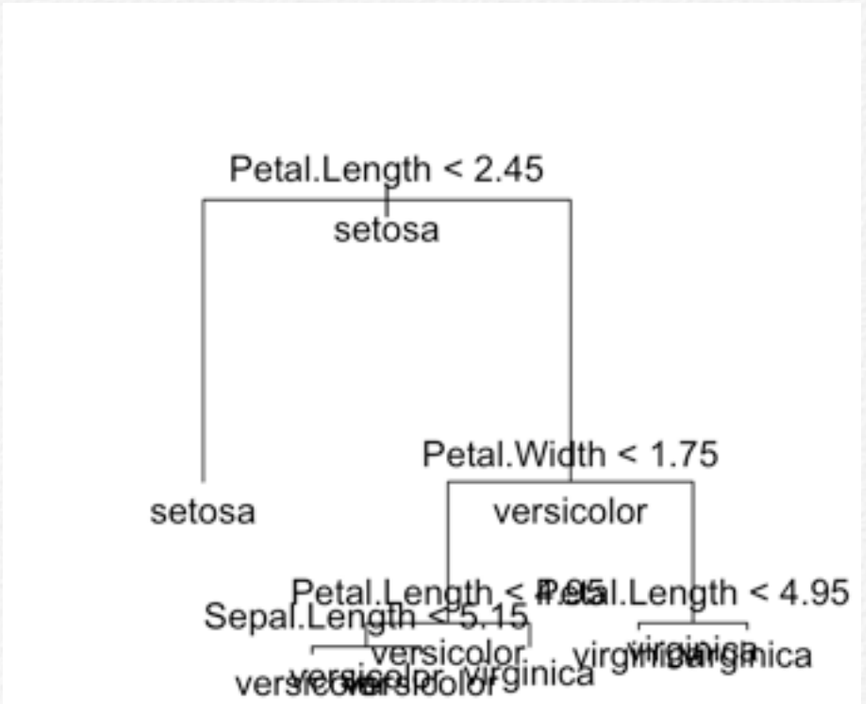
```
1) root 81 17 absent (0.79012346 0.20987654)
  2) Start>=8.5 62  6 absent (0.90322581 0.09677419)
    4) Start>=14.5 29  0 absent (1.00000000 0.00000000) *
    5) Start< 14.5 33  6 absent (0.81818182 0.18181818)
     10) Age< 55 12  0 absent (1.00000000 0.00000000) *
     11) Age>=55 21  6 absent (0.71428571 0.28571429)
       22) Age>=111 14  2 absent (0.85714286 0.14285714) *
       23) Age< 111 7  3 present (0.42857143 0.57142857) *
  3) Start< 8.5 19  8 present (0.42105263 0.57894737) *
```

IRIS 분꽃 데이터 : TREE 패키지 이용

```
library(MASS)
library(tree)
data(iris)
table(iris$Species)
ir.fit=tree(Species ~., iris)
summary(ir.fit)
ir.fit
ir.fit0 = snip.tree(ir.fit, nodes = c(12, 7)) #pruning
plot(ir.fit)
text(ir.fit, all = T)
plot(ir.fit0)
text(ir.fit0, all = T)
ir.fit2 = prune.misclass(ir.fit,best=4)
```

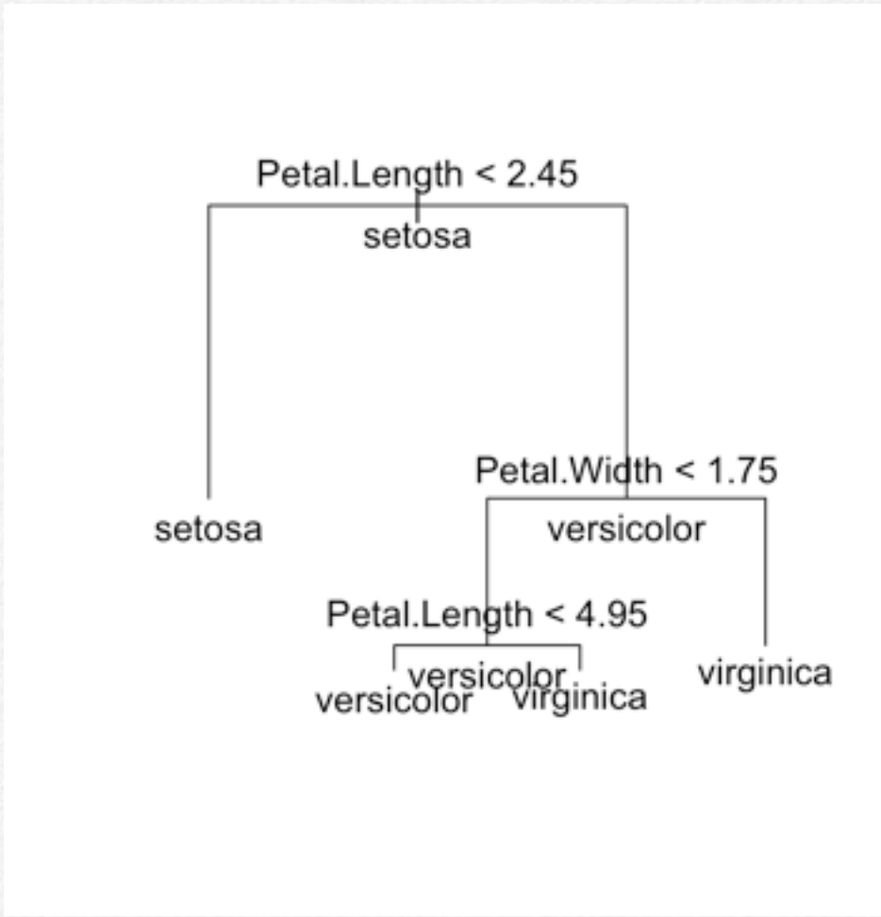
plot(ir.fit2)

text(ir.fit2, all = T)




```
> summary(ir.fit)

Classification tree:
tree(formula = Species ~ ., data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"  "Sepal.Length"
Number of terminal nodes:  6
Residual mean deviance:  0.1253 = 18.05 / 144
Misclassification error rate: 0.02667 = 4 / 150
```



(가지치기 후

```
> summary(ir.fit0)

Classification tree:
snip.tree(tree = ir.fit, nodes = c(12L, 7L))
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"
Number of terminal nodes:  4
Residual mean deviance:  0.1849 = 26.99 / 146
Misclassification error rate: 0.02667 = 4 / 150
```