빅데이터 처리 (화요일 (1:3교시))

5주차 강의

실습-의사결정나무

2022.04.05

Institutor: JS LEE

필요패키지

- R에는 의사결정나무 분석을 할 수 있는 대표적인 3개의 패키지는 tree, rpart, party 임.

- 각각의 패키지는 의사결정나무를 만들 때 가지치기를 하는 방법에 차이임
- 본 실습은 rpart, party패키지를 활용함

- rpart 는 CART 알고리즘을 기반으로 하며, CART 알고리즘 기반의 의사결정나무는 2가지의 문제가 존재함
  - ✓ 첫 번째는 통계적 유의성에 대한 판단 없이 노드를 분할하는데 대한 과적합(Overfitting)
  - ✓ 두 번째는 다양한 값으로 분할 가능한 변수가 다른 변수에 비하여 선호되는 현상

- 이 두 가지를 문제를 해결한 새로운 방법이 조건부 추론나무(Conditional Inference Tree) 이며, R에서는 party 패키지에 ctree 명령어로 수행

- 의사결정나무의 모델생성은 " 의사결정나무만들기 - 가지치기 - 예측 및 모델 평가"의 단계를 반복함

- ctree : Pruning할 필요가 없으나. 입력 변수의 레벨이 31개로 제한됨

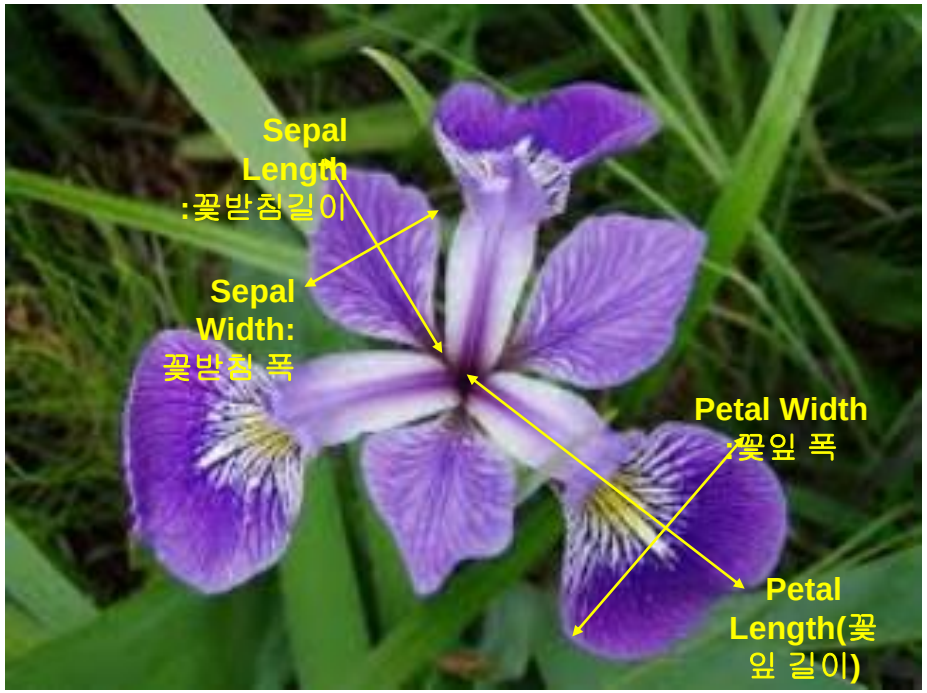
필요패키지

필요 패키지명	설명
rpart (Recursive Partitioning And Regression Trees)	• CART기반의 의사결정나무 생성 및 시각화를 위한 함수 포함
party	• 조건부 추론 기반의 의사결정나무 생성 및 시각화를 위한 함수 포함
caret (Classification And REgression Training)	• 복잡한 회귀와 분 류 문제에 대한 모형 훈련(training)과 조절(tuning) 과정을 간소화하는 함수를 포함. • 훈련 데이터의 전처리, 변수의 중요성 계산 및 모형 시각화를 위한 방법을 포함
rpart.plot	• 고급화된 rpart 시각화

데이터 셋 소개

아이리스(iris, 붓꽃)

- 꽃을 피우는 식물, 지구 곳곳에서 광범위하게 관찰된다.
- 아이리스의 속(屬)은 300개 이상의 서로 다른 종(種)
- 각각의 종은 꽃과 꽃잎의 모양, 크기와 같은 물리적 특성이 각기 다르다.



✓ petal: 꽃잎

✓ sepal: 꽃받침

데이터 셋 소개

아이리스 데이터셋

- Ronald Fisher가 “분류 문제에서 다중 척도의 사용”(Fisher, 1936)이라는 판별분석 세미나에서 소개
- 이해하기 간단하고 설명하기 쉽다.
- 많은 데이터 마이닝 기법들에서 공통적으로 사용된다.
- 기법들의 성능을 비교하는 데 유용

- 150개의 관찰치
- 3개의 종(Iris setosa, Iris virginica, Iris versicolor)에 대해 각각 50개씩
- 각 관찰치는 4개의 속성: 꽃받침 길이, 꽃받침 폭, 꽃잎 길이, 꽃잎 폭
- 별도의 다섯 번째 속성은 관찰된 종의 이름

데이터 셋 소개

아이리스 데이터셋 (iris.xls)

IrisID	**꽃받침길이**	꽃받침 폭	**꽃잎 길이**	**꽃잎폭**	아이리스 종류
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
⋮	⋮	⋮	⋮	⋮	⋮
146	6.7	3.0	5.2	2.3	Iris-virginica
147	6.3	2.5	5.0	1.9	Iris-virginica
148	6.5	3.0	5.2	2.0	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3.0	5.1	1.8	Iris-virginica

실습

1. 데이터 셋을 훈련용(Training)와 테스트용(Test) 구분하기

createDataPartition(y,p,list) : Data Splitting functions

입력항목	설명
y	추출할 팩터 데이터
p	항목별 추출할 비율 (50%이면, 0.5로 입력)
list	추출할 백터 내 위치를 리스트 타입으로 받고 싶은 경우 TRUE(기본값 TRUE)

- createDataPartition은 데이터 자체를 나누지는 않으며, 지정한 조건에 따라 데이터를 추출 후, 벡터 내 위치정보를 결과값으로 반환하므로, 각 위치정보의 데이터를 추출하여 데이터 프레임화 해야 함

실습

1. 필요패키지 설치 및 로딩

```
install.packages("rpart")
library(rpart)
install.packages("caret")
library(caret)
install.packages("rpart.plot")
library(rpart.plot)
```

2. 데이터 셋 확인

```
# Iris 데이터 셋은 R에서 제공하는 데이터임
# 데이터를 불러온 후, 데이터셋 구조를 살펴보자

data("iris")
View(iris)
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 ...
```

실습

 # 3. 데이터 셋 훈련용과 테스용으로 구분

타겟변수인 Species를 기준으로 각 종류별로 80%씩 추출하여 훈련용(train) 데이터 셋으로

```
iris_row_idx <- createDataPartition(iris$Species, p=0.8, list=FALSE) # list=FALSE => 추출한 정보를 factor로
iris_train <- iris[iris_row_idx,]
str(iris_train)
```

```
'data.frame':  120 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.9 5.4 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 3.1 3.7 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.5 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.1 0.2 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

iris-train 데이터 셋의 꽃 종류별 데이터 수 확인

```
table(iris_train$Species)
```

```
setosa versicolor  virginica
        40         40         40
```

test data set 생성

```
iris_test <- iris[-iris_row_idx,]   # iris_row_idx 벡터 내의 존재하는 인덱스는 제외한 행 추출, “-”기호 사용

table(iris_test$Species)
```

```
setosa versicolor  virginica
        10         10         10
```

실습

훈련용 데이터 셋과 검증용 데이터 셋의 특성은 유사해야 함. 그러므로 두 데이터 셋의 분포를 간단히 살펴보고자 함

=> 데이터 탐색을 위해 사용한 함수 summary()를 활용해서 전반적인 데이터 특성을 살펴본다

```
> summary(iris_train)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.200   Min.   :1.000   Min.   :0.100   setosa    :40
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.575   1st Qu.:0.300   versicolor:40
 Median :5.800   Median :3.000   Median :4.300   Median :1.300   virginica :40
 Mean   :5.854   Mean   :3.079   Mean   :3.756   Mean   :1.208
 3rd Qu.:6.400   3rd Qu.:3.325   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> summary(iris_test)
  Sepal.Length    Sepal.Width    Petal.Length    Petal.Width          Species
 Min.   :4.400   Min.   :2.00   Min.   :1.300   Min.   :0.200   setosa    :10
 1st Qu.:5.200   1st Qu.:2.70   1st Qu.:1.625   1st Qu.:0.325   versicolor:10
 Median :5.800   Median :3.00   Median :4.400   Median :1.350   virginica :10
 Mean   :5.800   Mean   :2.97   Mean   :3.767   Mean   :1.167
 3rd Qu.:6.375   3rd Qu.:3.20   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.300   Max.   :3.80   Max.   :6.300   Max.   :2.300
> summary(iris_test)
```

실습

4. 의사결정나무 생성하기

```
rpart(formular, train_data, control)
```

입력항목	설명
formular	표현식 : 종속변수 ~ 독립변수+독립변수/+… 모든 독립변수 사용 시, . 을 이용 표현식 : 종속변수 ~.
train_data	사용할 훈련데이터 data = 데이터명
control	rpart 함수를 실행하는데 필요한 세부 설정 등.. ?rpart.control 명령어를 통해 다양한 설정 방법 확인 가능 대표적인 rpart 설정은 minsplit: 분류 시 포함 될 최소데이터 개수 control= rpart.control(minsplit=2)

실습

4. 의사결정나무 생성하기

```
iris_result <- rpart(Species ~., data=iris_train, control=rpart.control(minsplit = 2))

iris_result
```

```
n= 120

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

 1) root 120 80 setosa (0.33333333 0.33333333 0.33333333)
   2) Petal.Length< 2.45 40  0 setosa (1.00000000 0.00000000 0.00000000) *
   3) Petal.Length>=2.45 80 40 versicolor (0.00000000 0.50000000 0.50000000)
     6) Petal.Width< 1.75 43  4 versicolor (0.00000000 0.90697674 0.09302326)
      12) Petal.Length< 4.95 39  1 versicolor (0.00000000 0.97435897 0.02564103)
        24) Petal.Width< 1.65 38  0 versicolor (0.00000000 1.00000000 0.00000000) *
        25) Petal.Width>=1.65 1  0 virginica (0.00000000 0.00000000 1.00000000) *
      13) Petal.Length>=4.95 4  1 virginica (0.00000000 0.25000000 0.75000000)
        26) Sepal.Length>=6.5 1  0 versicolor (0.00000000 1.00000000 0.00000000) *
        27) Sepal.Length< 6.5 3  0 virginica (0.00000000 0.00000000 1.00000000) *
     7) Petal.Width>=1.75 37  1 virginica (0.00000000 0.02702703 0.97297297) *
> iris_result
```

실습

- 강의안과 수치가 약간씩 상의할 수 있음
- 데이터 구분 시, 랜덤샘플링 결과가 상이해서임

4. 의사결정나무 생성하기 - 도식화

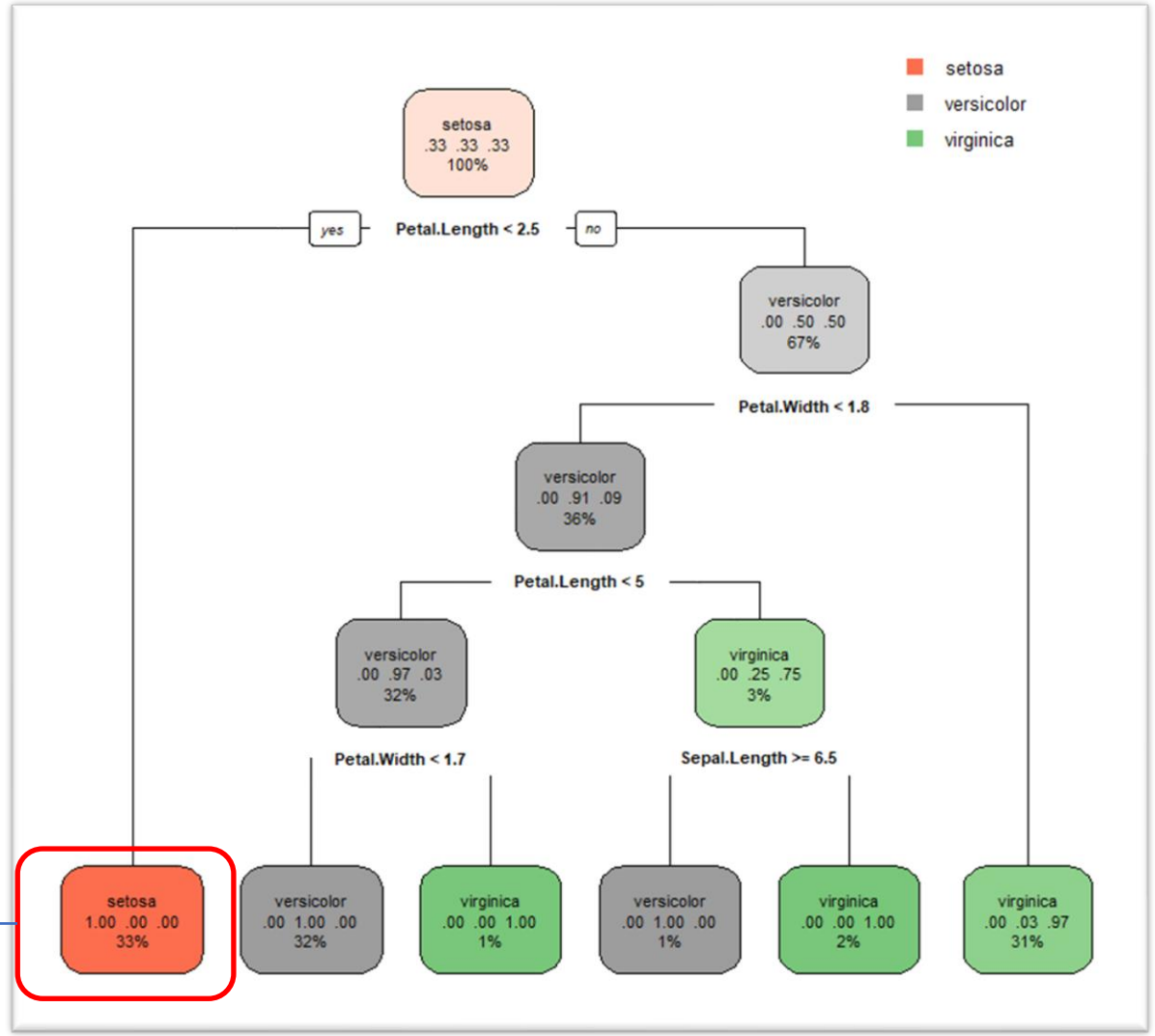
rpart.plot (의사결정나무 결과파일)

```
rpart.plot(iris_result)
```

범례: 색상으로 꽃의 종류 확인 가능하며
범례의 순서 또한 그래프 이해에 중요함

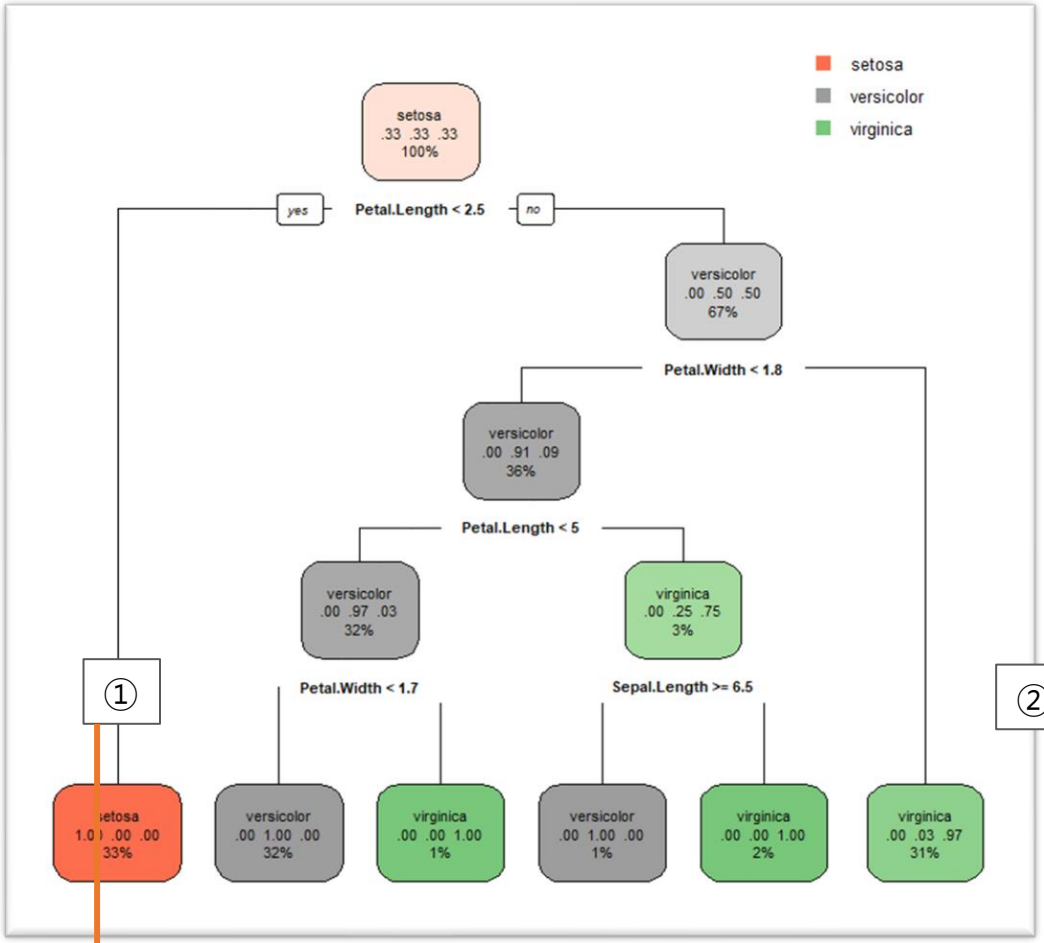
제일 상위의 노드: 뿌리 노드 (Root 노드)

- Sentosa로 분류된 노드이며,
- 1.0, 0.0,0.0
  :범례 순서 대로 이 노드에 포함 비율을 나타냄
  =>setosa, versicolor, viginica의 비율
- 33% : 전체 데이터 중 이 노드로 분류된 비율



실습

4. 의사결정나무 생성하기 -도식화



- If Petal.Length >= 2.5 and Petal. Width <1.8 then verginica
- 이 노드에는 setosa는 포함되어 있지 않으며, verisicolor : 3% 포함되어 있고, verginica는 97% 존재함
- 이 노드에 포함된 관측치수는 전체 데이터의 31%임

If Petal Length <2.5 then setosa

실습

5. 가지치기

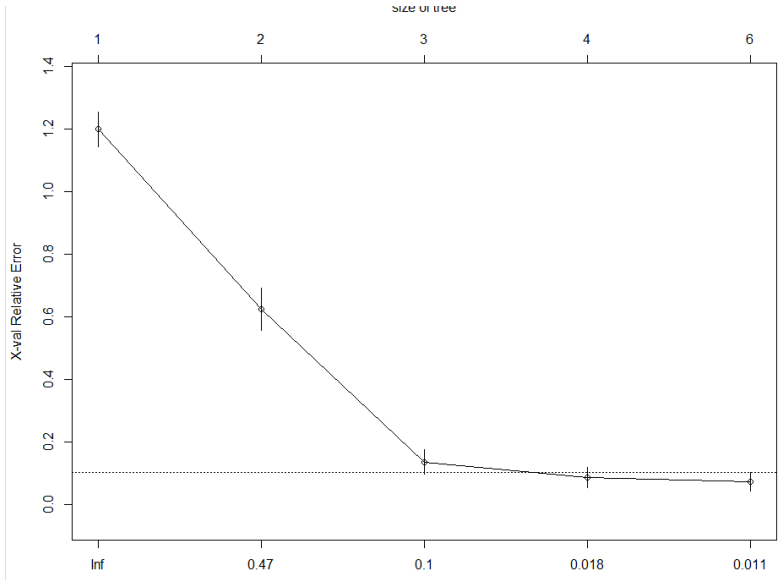
- 의사결정나무를 통해 직관적으로 모델을 이해 할 수 있음. 그러나 분류기준이 많아 질 수록 그래프의 복잡성이 높아지며 이해도가 떨어질 수 있음
- 가장 중요한 과적합이 일어날 수 있음
- 이를 위해 모델의 성능을 크게 훼손하지 않은 상태해서 모델의 단순화가 필요함
- 이를 위해 CP(Complexity Parameter)를 사용함 (CP : 가지를 생성될 때 소요되는 복잡도 의미함)
- rpart함수는 CP값에 따른 나뭇가지 수와 그에 따른 분류오류율을 cptable을 통해 확인 가능함
- plotcp는 cptable을 그래프로 표현

```
iris_result$cptable
```

	CP	nsplit	rel error	xerror	xstd
1	0.5000	0	1.0000	1.2000	0.05477226
2	0.4375	1	0.5000	0.6250	0.06750772
3	0.0250	2	0.0625	0.1375	0.03951200
4	0.0125	3	0.0375	0.0875	0.03209280
5	0.0100	5	0.0125	0.0750	0.02984334

*xerror : 교차타당성 에러 ,기본 10fold

```
plotcp(iris_result)
```

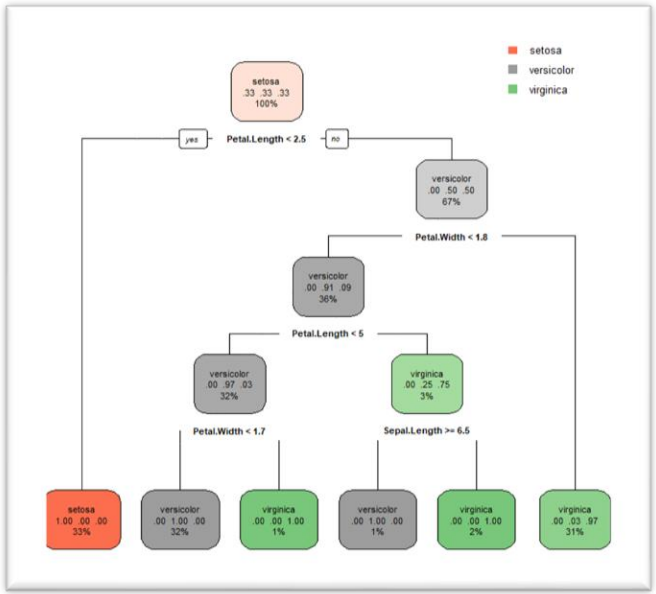


실습

가지치기는 prune함수를 이용함 prune(가지치기대상 의사결정나무, cp= r기준)

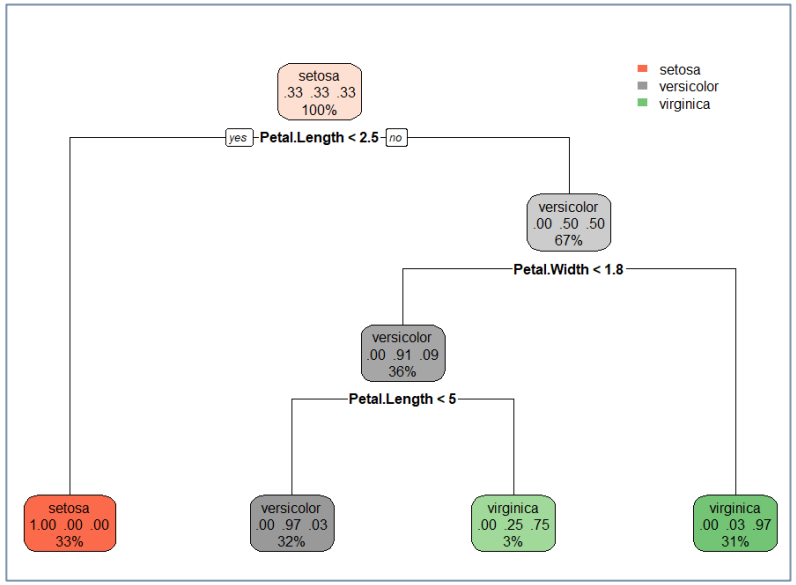
```
iris_prune <- prune(iris_result, cp=0.0018)
raprt.plot(iris_prune)
```

가지치기 전



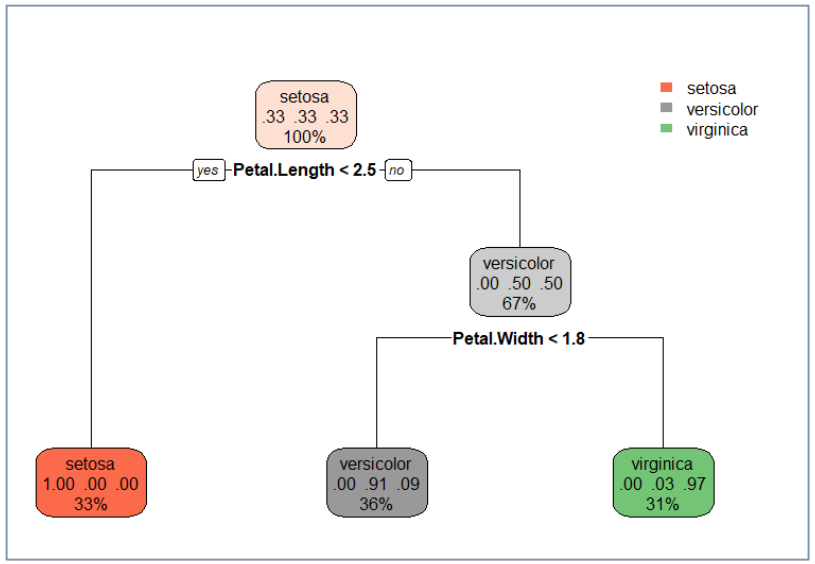
오류율: 3%

CP =0.0125



오류율: 3.75%

CP =0.025 ,



오류율: 6.25%

- 오류율이 다소 증가 했을지라도, 의사결정기준은 2개로 줄어들었으며, 종을 구분이 명쾌해짐
- 오류율이 다소 증가해도, 모델을 단순화하는 것이 좀 더 범용적인 케이스에 적용가능하며 더욱 직관적인 모델을 만들 수 있음

실습

2. 조건부 추론 기반의 의사결정나무 생성　ctree() 함수 이용

 # 1. 필요패키지 설치 및 로딩

```
install.packages("party")
library(party)
```

```
iris_ctree_result <- ctree(Species ~.,
data=iris_train,control=ctree_control(minsplit = 2))
```

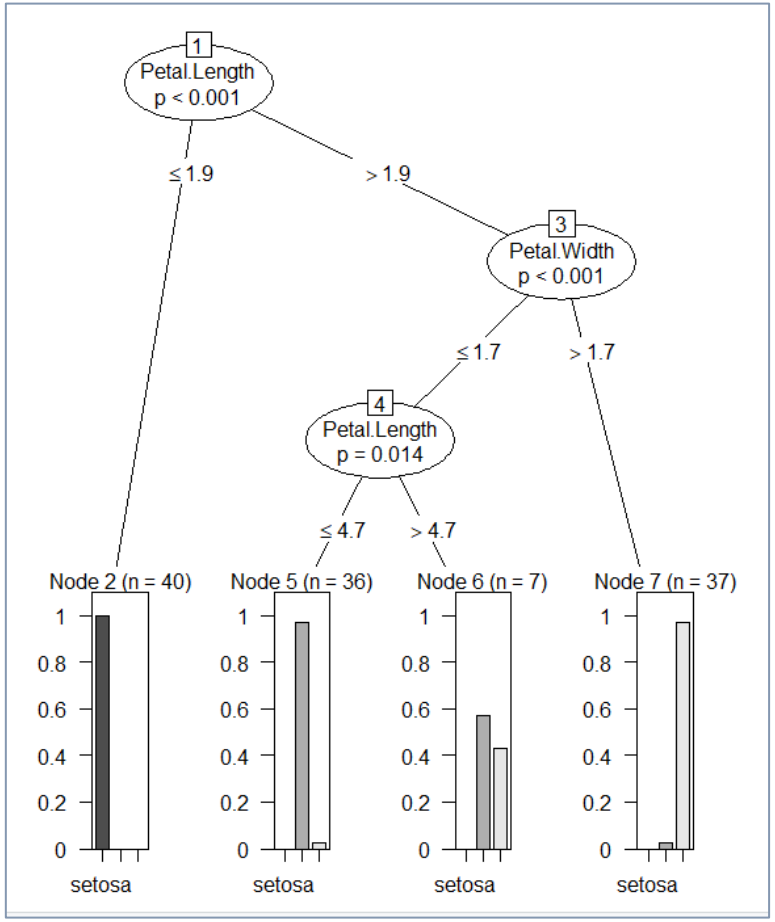
ctree(formular, train_data, control)

입력항목	설명
formular	표현식 : 종속변수 ~ 독립변수+독립변수/+… 모든 독립변수 사용 시, . 을 이용 표현식 : 종속변수 ~.
train_data	사용할 훈련데이터 data = 데이터명
control	ctree함수를 실행하는데 필요한 세부 설정 등.. ?ctree_control 명령어를 통해 다양한 설정 방법 확인 가능 control= c_tree.control(minsplit=2)

실습

```
iris_ctree_result <- ctree(Species ~.,data=iris_train, control=ctree_control(minsplit = 2))
iris_ctree_result
plot(iris_ctree_result)  # 도식화
```

도식화



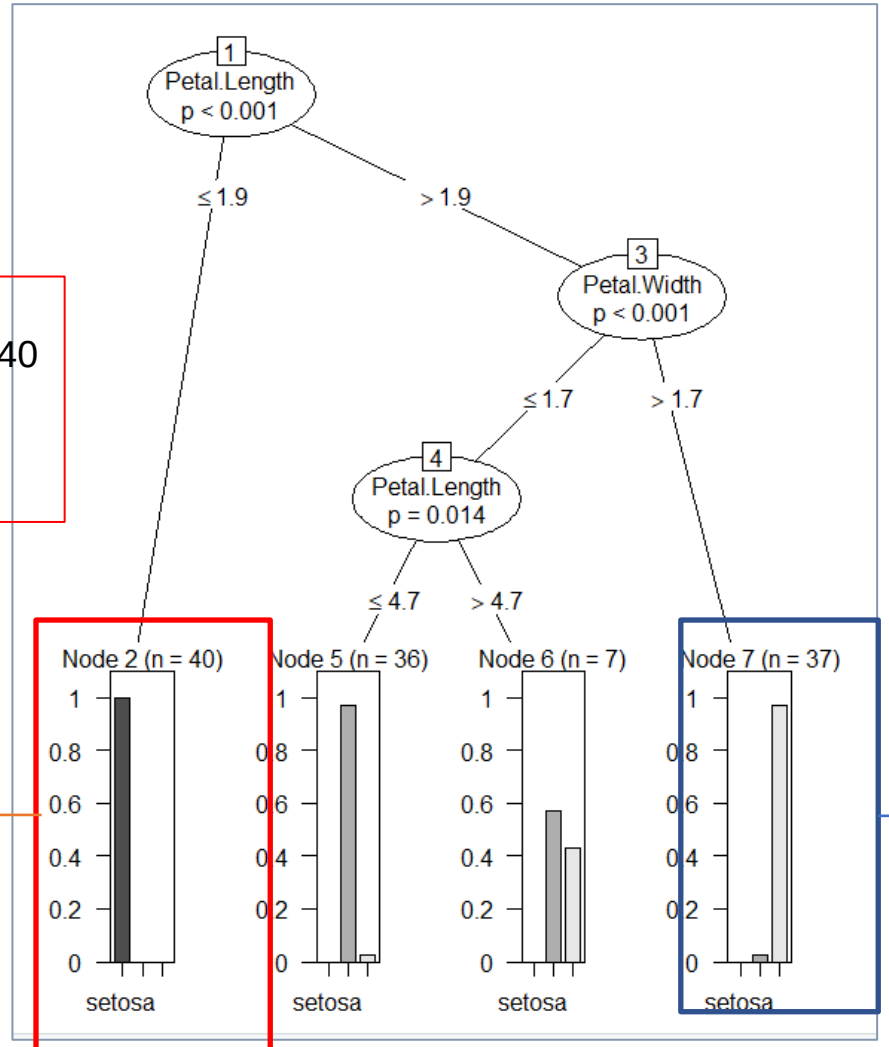
규칙

```
                Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  120

1) Petal.Length <= 1.9; criterion = 1, statistic = 111.779
  2)*  weights = 40
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 54.743
    4) Petal.Length <= 4.7; criterion = 0.986, statistic = 8.583
      5)*  weights = 36
    4) Petal.Length > 4.7
      6)*  weights = 7
  3) Petal.Width > 1.7
    7)*  weights = 37
```

실습



Node 2 (n=40) :
노드명 : Node2 , 노드에 포함된 관측 개수 : 40

PetalLength <= 1.9 then setosa
100% Setosa

Node 7 (n=37) :
노드명 : Node7 , 노드에 포함된 관측 개수 : 37

PetalLength > 1.9 and petal Width > 1.7 then then verginica

실습

모형평가

1) 예측하기 : 검증용 데이터 셋(전체 데이터 셋에서 모형 검증을 위해 분할 한 데이터 셋) 을 생성한 의사결정나무 모델에 적용함

    * 모델을 결정 후에는 모델을 적용할 새로운 데이터 셋을 의사결정나무 모델에 적용

2) 실제 타겟 값과 모델링을 통해 예측한 타겟 값을 비교함

3) 다양한 평가기준으로 모형 평가하기

실습

모형평가

1) 예측하기

predict()함수 이용

```
rpart(analysis_object, newdata, type)
```

입력항목	설명
analysis_object	적용할 분석 모델 객체
newdata	예측할 데이터 프레임 (독립변수의 같은 컬럼명이어야 함. 새로운 데이터 적용시에는 종속변수는 존재하지 않음)
type	class : 분류 결과를 라벨명으로 표현 prob : 분류 결과를 확률로 표현

3) 혼동행렬을 이용하여 모형평가 하기

confusionMatrix()함수이용

```
confusionMatrix(data, reference, mode
```

입력항목	설명
data	예측분류된 결과값이 있는 벡터
reference	실제값을 가지고 있는 벡터
mode	평가기준 everything (모든 기준)

실습

모형평가

```
# 모형 적용하여 예측 값 생성

expect <- predict(iris_prune, iris_test, type="class")

# 실제 종속변수
actual <- iris_test$Species

# 데이터셋 만들기
iris_performance <- data.frame(actual, expect)
```

```
            expect
actual       setosa versicolor virginica
  setosa         10          0         0
  versicolor      0          9         1
  virginica       0          1         9
```

```
# 3. 혼동행렬을 이용하여 모형평가 하기
confusionMatrix(expect,actual,mode="everything")
```

```
          Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0          9         1
  virginica       0          1         9

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 8.747e-12

                  Kappa : 0.9

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9000           0.9000
Specificity                 1.0000            0.9500           0.9500
Pos Pred Value              1.0000            0.9000           0.9000
Neg Pred Value              1.0000            0.9500           0.9500
Precision                   1.0000            0.9000           0.9000
Recall                      1.0000            0.9000           0.9000
F1                          1.0000            0.9000           0.9000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3000           0.3000
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.9250           0.9250
>
```

정확도

F1 score는 precision 과 recall의 조화평균 입니다
0에서 1사이의 값을 가지며, 클수록 좋음

실습

모형평가

ctree 로 적용하기

```
expect1 <- predict(iris_ctree_result, iris_test, type="response")
```

ctree인 경우 범주형으로 예측값을 출력하고자 할때 “response” 사용

```
iris_performance1 <- data.frame(actual,expect1)
table(iris_performance1)
```

```
            expect1
actual       setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         0
  virginica       0          1         9
```

```
confusionMatrix(expect1,actual,mode="everything")
```

```
            expect1
actual       setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         0
  virginica       0          1         9
> confusionMatrix(expect1,actual,mode="everything")
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         1
  virginica       0          0         9

Overall Statistics

               Accuracy : 0.9667
                 95% CI : (0.8278, 0.9992)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 2.963e-13

                  Kappa : 0.95

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9000
Specificity                 1.0000            0.9500           1.0000
Pos Pred Value              1.0000            0.9091           1.0000
Neg Pred Value              1.0000            1.0000           0.9524
Precision                   1.0000            0.9091           1.0000
Recall                      1.0000            1.0000           0.9000
F1                          1.0000            0.9524           0.9474
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3000
Detection Prevalence        0.3333            0.3667           0.3000
Balanced Accuracy           1.0000            0.9750           0.9500
```

실습

두 모형 비교

rpart

```
            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0          9         1
  virginica       0          1         9

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 8.747e-12

                  Kappa : 0.9

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9000           0.9000
Specificity                 1.0000            0.9500           0.9500
Pos Pred Value              1.0000            0.9000           0.9000
Neg Pred Value              1.0000            0.9500           0.9500
Precision                   1.0000            0.9000           0.9000
Recall                      1.0000            0.9000           0.9000
F1                          1.0000            0.9000           0.9000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3000           0.3000
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.9250           0.9250
>
```

ctree결과

```
            expect1
actual       setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         0
  virginica       0          1         9
> confusionMatrix(expect1,actual,mode="everything")
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         10         1
  virginica       0          0         9

Overall Statistics

               Accuracy : 0.9667
                 95% CI : (0.8278, 0.9992)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 2.963e-13

                  Kappa : 0.95

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9000
Specificity                 1.0000            0.9500           1.0000
Pos Pred Value              1.0000            0.9091           1.0000
Neg Pred Value              1.0000            1.0000           0.9524
Precision                   1.0000            0.9091           1.0000
Recall                      1.0000            1.0000           0.9000
F1                          1.0000            0.9524           0.9474
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3000
Detection Prevalence        0.3333            0.3667           0.3000
Balanced Accuracy           1.0000            0.9750           0.9500
```