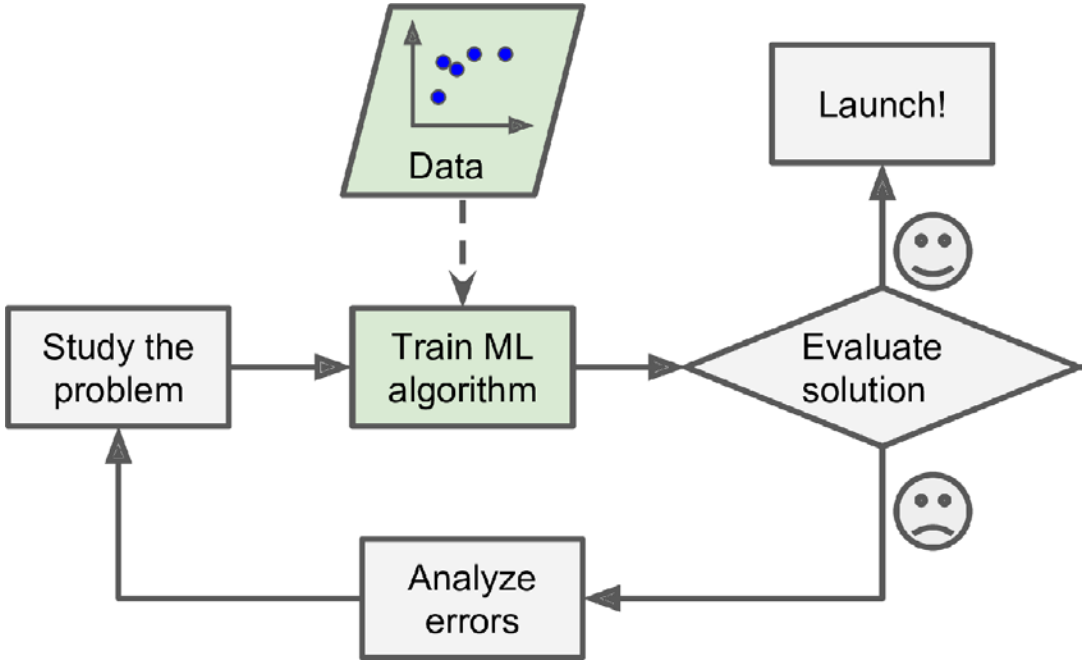
6장 데이터 분석기법(분류 모델)

- 머신 러닝
- 분류 모델
  - 의사결정트리
  - 렌덤 포리스트
  - SVM

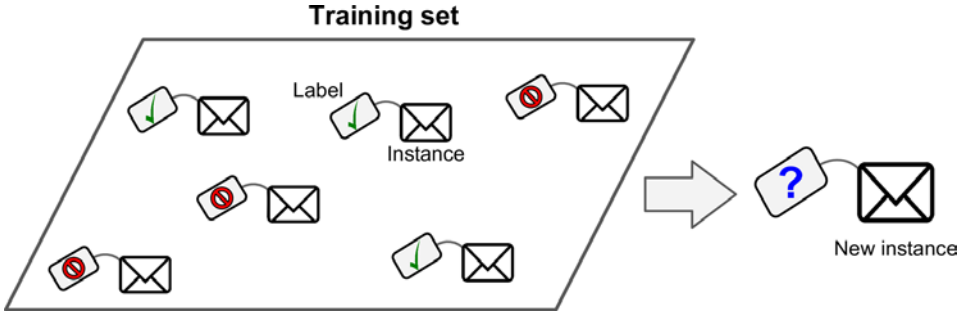
01 머신 러닝

- **머신 러닝(Machine Learning)**
  - 인간이 다양한 현상들을 경험하거나 목격을 통해 학습하는 것에 비유
  - 기계(컴퓨터)에게 다수의 데이터를 제공하여 일정한 법칙 등을 찾아내도록 하는 것
  - ML 알고리즘: 지도학습/비지도학습



01-1 지도 학습

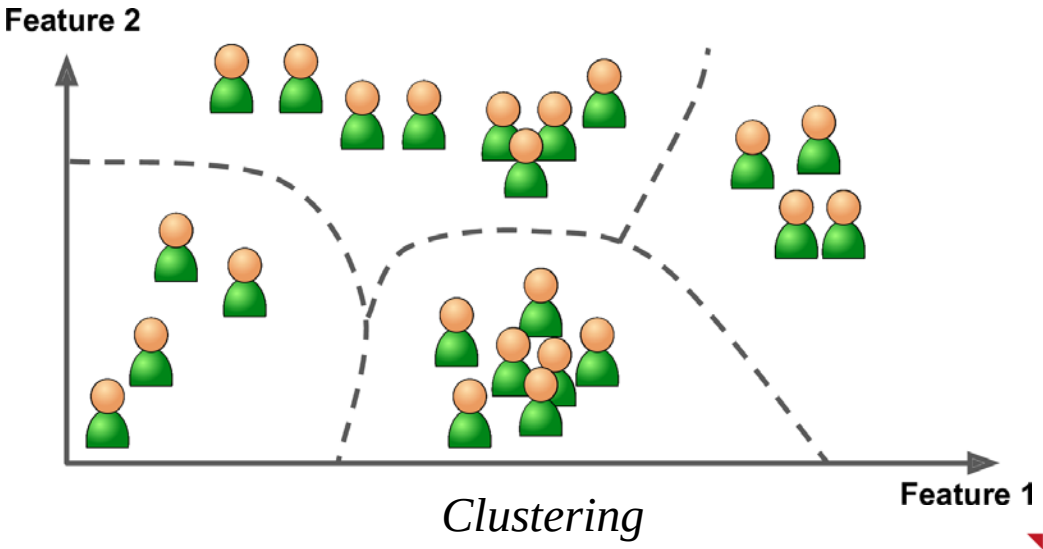
- **지도 학습 (supervised learning)**
  - 과거 데이터에 대한 정답/오답을 알고 있는 상황에서 기계에게 가능한 정답률을 올리도록 "학습"시키는 것
  - 분류(Classification) / 회귀(Regression): 회귀는 반응 변수가 연속값을 가지며, 분류는 반응 변수가 이산값을 가짐
  - 알고리즘
    - ✓ k-Nearest Neighbors
    - ✓ Linear Regression
    - ✓ Logistic Regression
    - ✓ Decision Trees and Random Forests
    - ✓ Support Vector Machines (SVMs)
    - ✓ Neural networks



A labeled training set for spam classification

01-2 비지도 학습

- **비지도 학습(unsupervised learning)**
  - 정답 혹은 오답 정보가 누락된 상황에서 특정한 규칙을 찾아내는 것
  - 알고리즘
    - ✓ Clustering
      - ✓ K-Means
      - ✓ DBSCAN
      - ✓ Hierarchical Cluster Analysis (HCA)
    - ✓ Association rule learning
      - ✓ Apriori
      - ✓ Eclat
-



Clustering

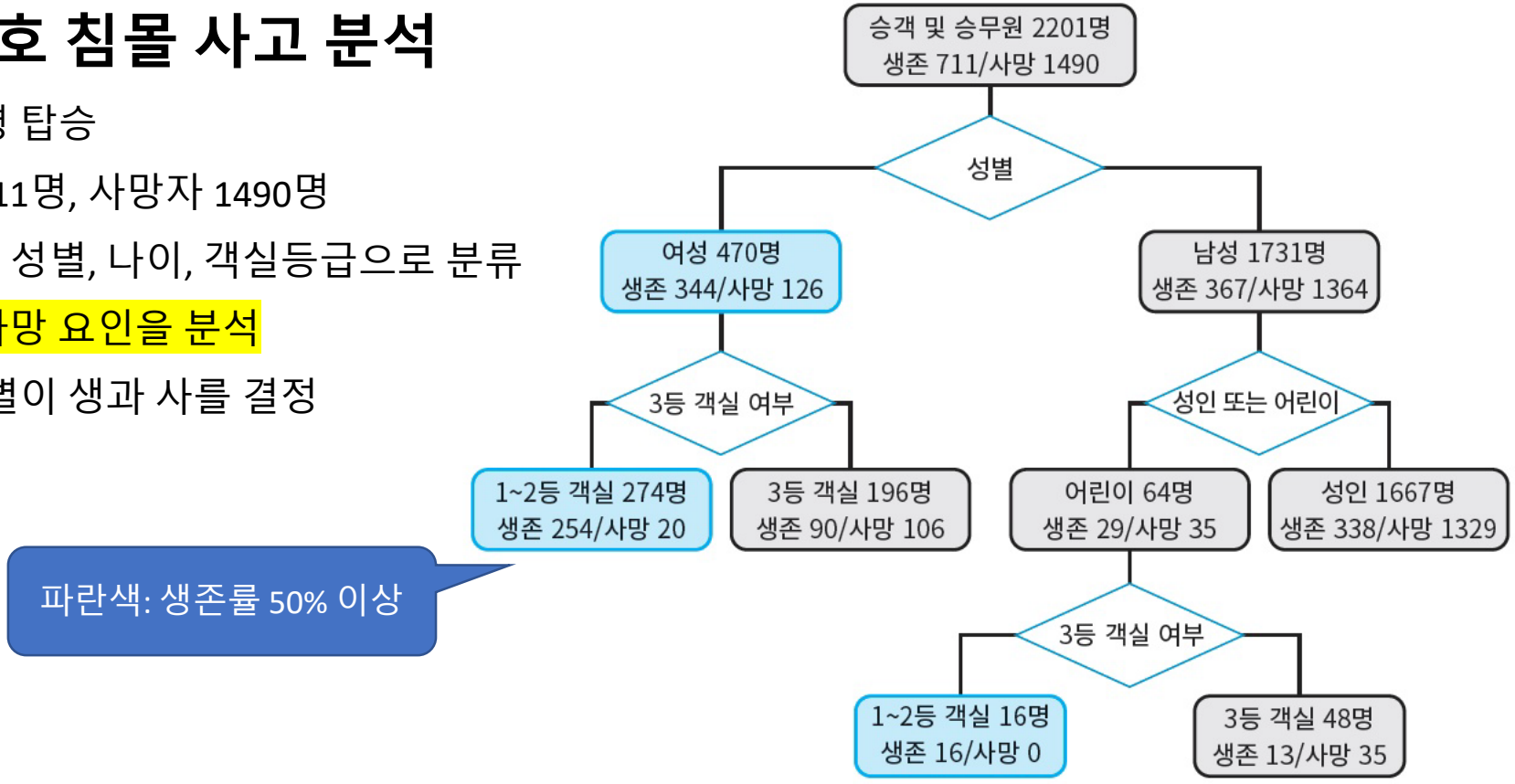
01-3 분류 알고리즘

- 의사 결정 트리(Decision Tree)
- 랜덤 포리스트 (Random Forest)
- SVM (Support Vector Machine)

02 의사 결정 트리 (decision Tree)

- **의사결정트리:** 의사결정을 위한 규칙을 트리 구조로 나타내어 전체 데이터를 몇 개의 소집단으로 분류하고, 새로운 데이터에 대한 예측을 수행하는 분석방법이다.

- **타이타닉호 침몰 사고 분석**
  - 총 2201명 탑승
  - 생존자 711명, 사망자 1490명
  - 탑승자를 성별, 나이, 객실등급으로 분류
  - 생존과 사망 요인을 분석
  - 결과: 성별이 생과 사를 결정

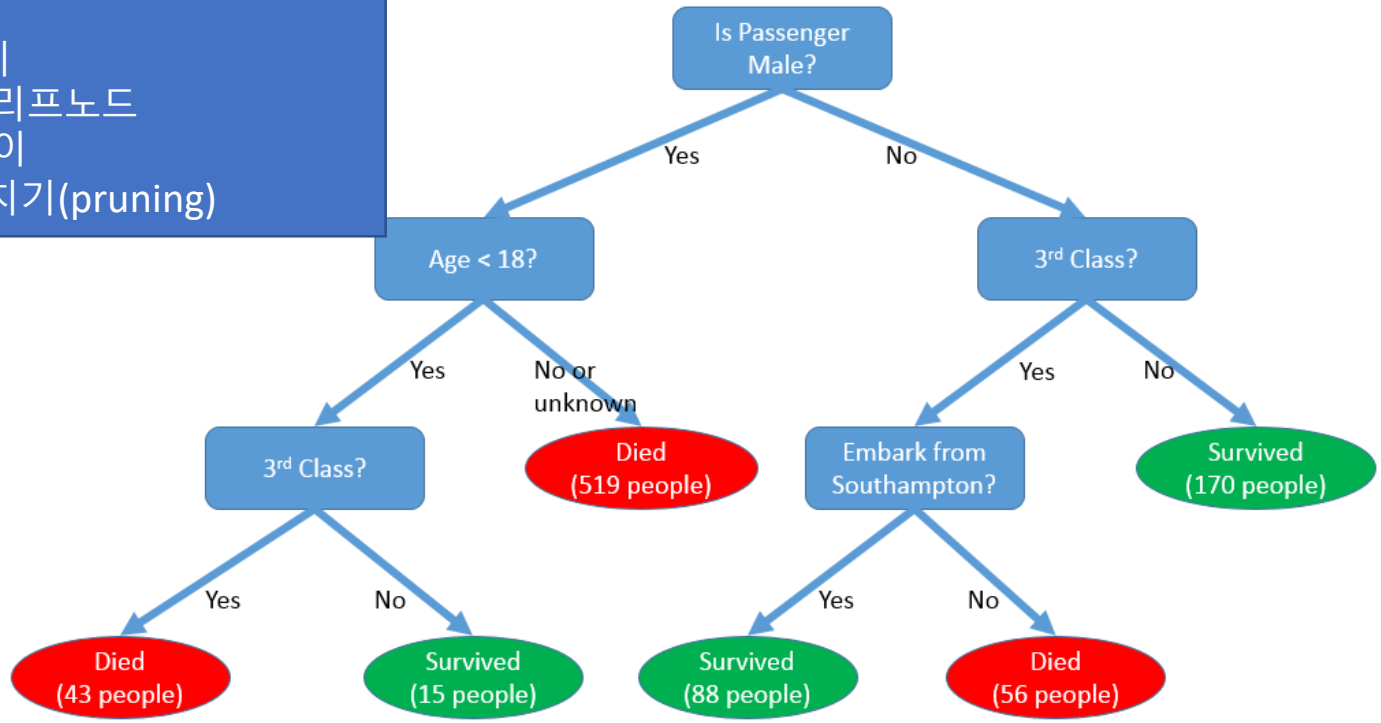


02 의사 결정 트리 (decision Tree)

- **Titanic**

	Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings,	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mi	male		0	0	330877	8.4583		Q

이진트리
노드/에지
부모/자식/루트/리프노드
트리의 깊이
가지분할(split)/가지치기(pruning)



- Survived: 생존 여부 => 0 = No, 1 = Yes
- pclass: 티켓 등급 => 1 = 1st, 2 = 2nd, 3 = 3rd
- Sex: 성별
- Age: 나이
- Sibsp: 함께 탑승한 형제자매, 배우자의 수
- Parch: 함께 탑승한 부모, 자식의 수
- Ticket: 티켓 번호
- Fare: 운임
- Cabin: 객실 번호
- Embarked: 탑승 항구 => C = Cherbourg, Q = Queenstown, S = Southampton

02-1 의사 결정 트리의 특징

- 분류, 회귀문제에 모두 적용 가능함.
- 복잡한 데이터셋도 학습할 수 있음.
- 데이터 전처리가 필요하지 않음 : 스케일링 등이 필요 없음
- 알고리즘
  - CART 알고리즘 : 이진트리만 생성(지니 지수 사용), 연속값 처리, 회귀가능
  - CHAID, ID3, C4.5, C5.0 등 다양한 알고리즘이 개발되어 있음
- 랜덤 포레스트와 그래디언트 부스팅 앙상블 학습의 기본학습기
- 화이트 박스(white box) 모델 (해석하기 쉽다) < = > 블랙 박스 모델 (랜덤 포레스트, 신경망)
- 비모수모델(nonparametric model) : 수학적 분포함수에 의존하지 않음
  (비모수모델은 데이터 분포에 제한이 없어 복잡도가 큼 ➔ 과대적합)
- 장단점:
  - 장점: 쉽고 직관적이며, 스케일링/정규화 같은 전처리 작업의 영향도가 크지 않다.
  - 단점: 규칙을 추가하여 서브트리를 만들어 갈수록 모델이 복잡해지고 과적합에 빠지기 쉽다. 따라서 트리 크기를 사전에 제한하는 튜닝이 필요하다.

02-2 의사 결정 트리의 분석과정

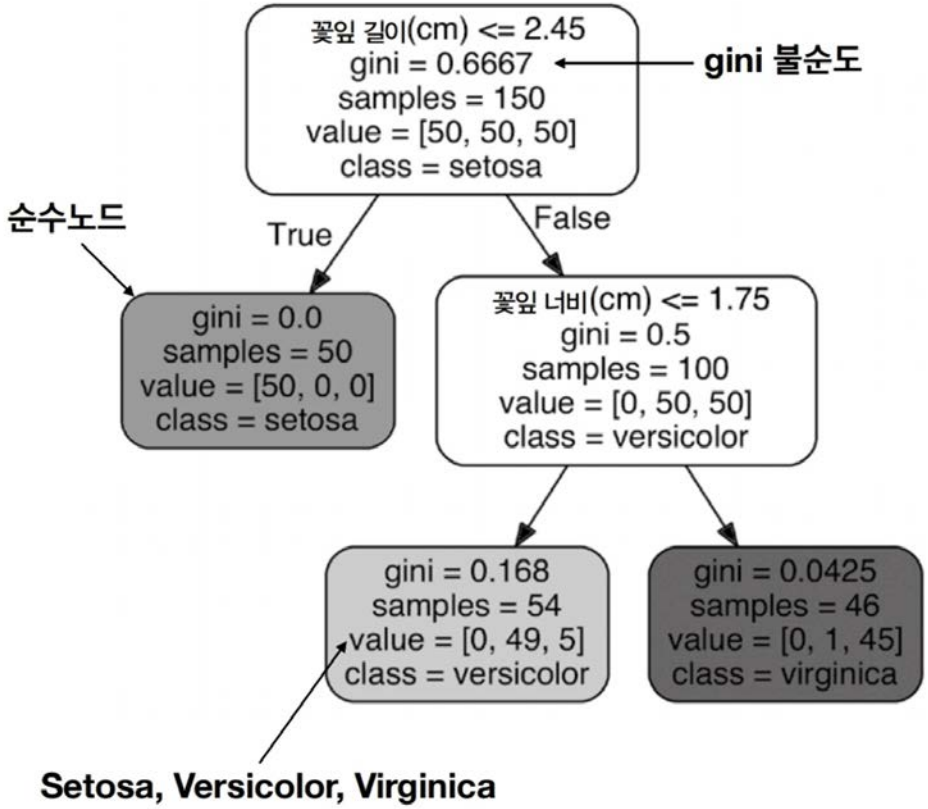
- 반응변수와 관계가 있는 설명변수들의 선택
- 적절한 분리규칙과 정지규칙을 정하여 트리 생성
- 부적절한 가지 치기
- 이익/위험/비용 등을 고려한 모델의 성능 평가
- 분류 및 예측 수행

02-3 Simple Example (iris 데이터)

꽃받침길이

꽃잎 길이

		Sepal.Leng	Sepal.Widt	Petal.Leng	Petal.Widt	Species
1						
2	1	5.1	3.5	1.4	0.2	setosa
3	2	4.9	3	1.4	0.2	setosa
4	3	4.7	3.2	1.3	0.2	setosa
52	51	7	3.2	4.7	1.4	versicolor
53	52	6.4	3.2	4.5	1.5	versicolor
54	53	6.9	3.1	4.9	1.5	versicolor
149	148	6.5	3	5.2	2	virginica
150	149	6.2	3.4	5.4	2.3	virginica
151	150	5.9	3	5.1	1.8	virginica



02-4 알고리즘

- **CART(Classification And Regression Tree) 알고리즘:** 지니 지수(Gini index) 또는 분산의 감소량을 사용하여 트리의 가지를 이진(binary) 분리한다. 범주형 변수에 대하여는 지니 지수를 사용하고, 연속형 변수에 대하여는 분산의 감소량을 사용한다.

노드분할 : 특징 k 에 임계값 t_k로 나눔. (아래 비용함수를 최소화하는 k, t_k 찾아서...)

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

여기서 $G_{left/right}$는 왼쪽/오른쪽 서브셋의 불순도

$m_{left/right}$는 왼쪽/오른쪽 서브셋의 샘플 수

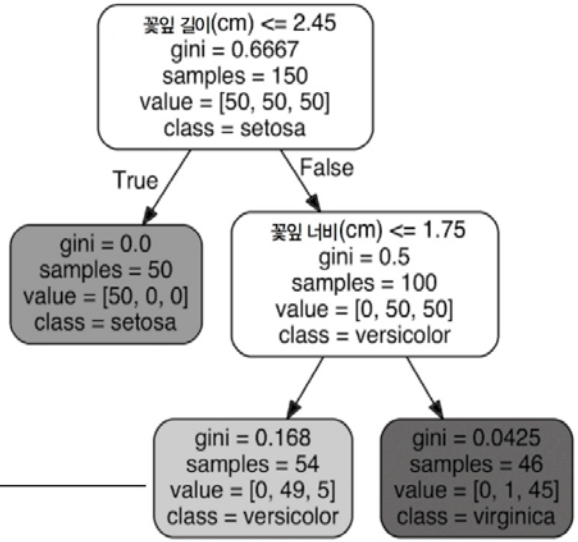
- 노드 분할을 이용해서 root 노드(전체 훈련세트)를 둘로 나눔, root 노드의 깊이는 0
- 자식 노드들을 각각 둘로 나눔 : 깊이 1씩 증가
  (Gini 불순도 값이 감소하지 않는 노드는 나누지 않음 ➔ 리프 노드가 됨)
- 이 과정을 계속 반복 : 깊이가 max_depth가 되면 중지

02-4 알고리즘-계속

지니 불순도 (Gini impurity) 지수

- 지니 불순도는 노드의 샘플 클래스가 얼마나 분산되어 있는지를 측정합니다.
- DecisionTreeClassifier(criterion='gini'), 기본값
- 최악 0.5 ~ 최상 0 (n=2의 경우)

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$



- $p_{i,k}$는 i 번째 노드에 있는 훈련 샘플 중 클래스 k에 속한 비율

$$1 - (0/54)^2 - (49/54)^2 - (5/54)^2 \approx 0.168$$

높은 이질성 ⇔ 낮은 순수도

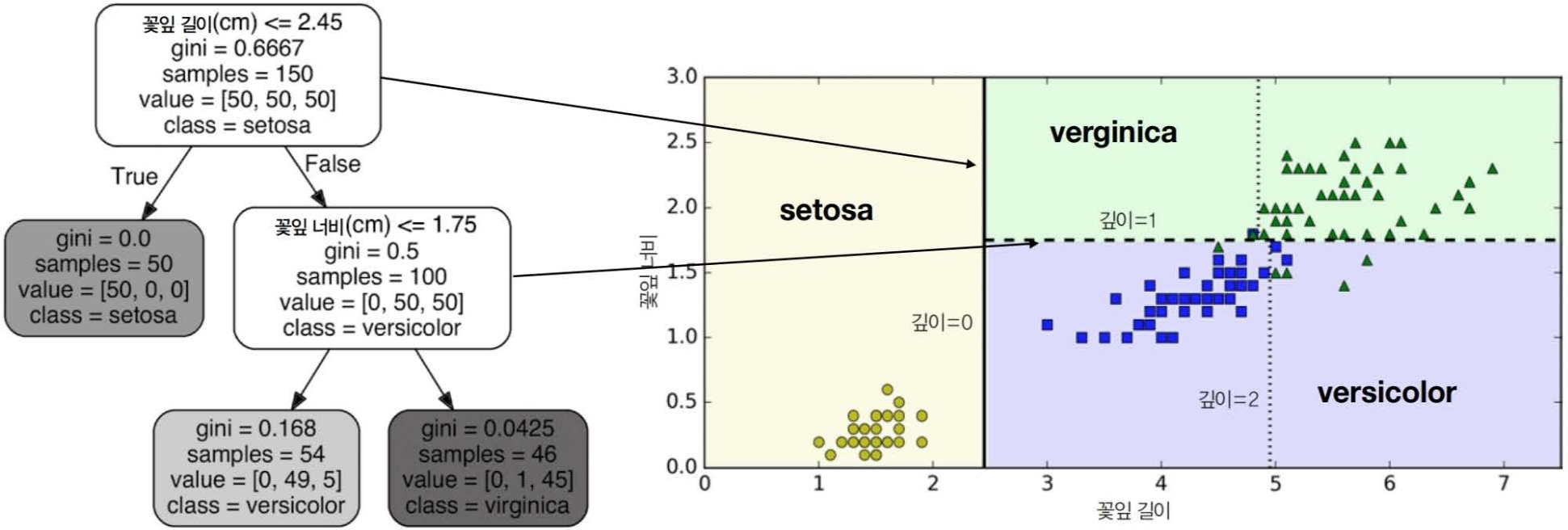
$G=1-(3/8)^2-(3/8)^2-(1/8)^2-(1/8)^2=0.69$

낮은 이질성 ⇔ 높은 순수도

$G=1-(7/8)^2-(1/8)^2=0.24$

02-4 알고리즘-계속

결정 경계 (decision boundary)



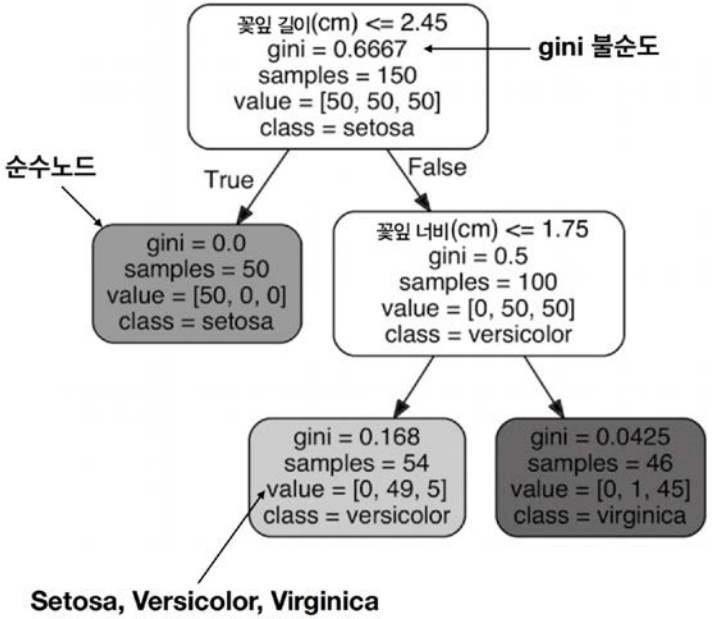
02-5-1 R 패키지의{rpart}의 rpart() 함수 이용

- rpart는 훈련 집합을 최소 오류로 분류하는 결정 트리를 생성한다.

  이 외에 tree(), ctree() 함수 등을 사용할 수 있다.

```
> library(rpart)
> r = rpart(Species~., data = iris)
```

반응변수가 범주형이어야 한다. 그렇지 않으면 회귀모델로 작동한다.



function description

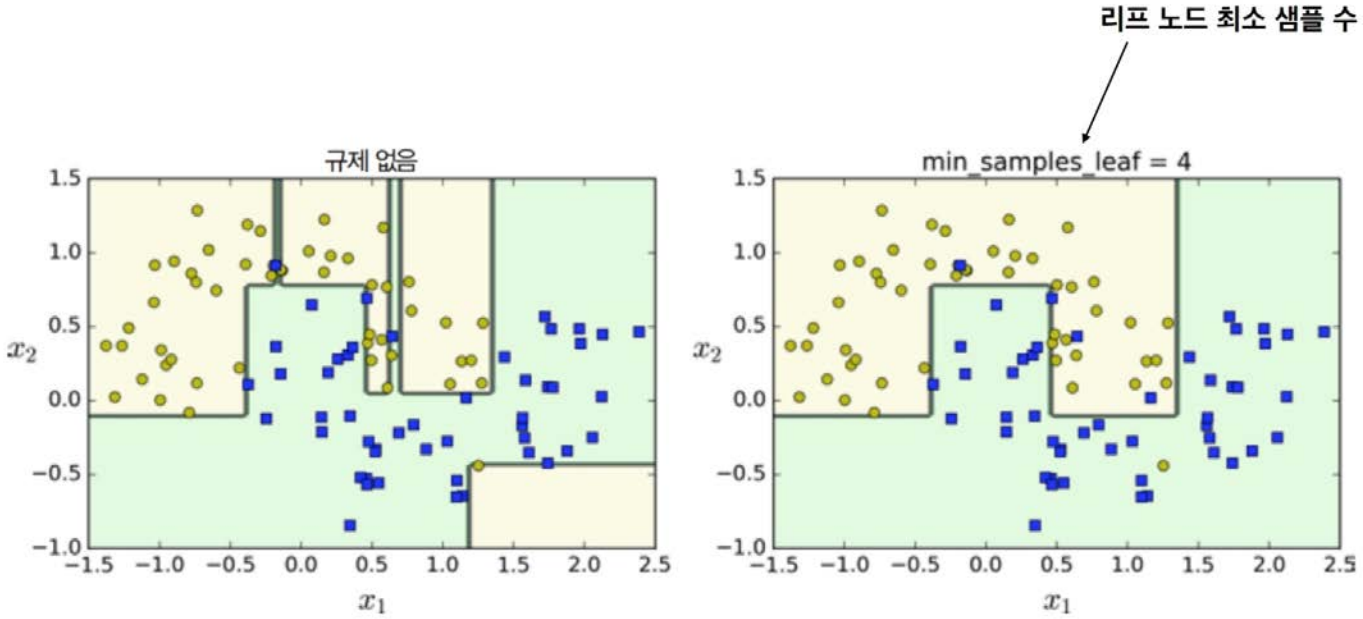
formula	다음 형식을 따라야 합니다
.	outcome ~ predictor1+predictor2+predictor3+ect.
data=	specifies the data frame
method=	"class" 범주형 변수 (classification tree)
.	"anova" 연속형 변수 (regression tree)
control=	Tree 크기를 제한하는 옵션입니다
.	For example, control=rpart.control(minsplit=30, cp=0.001) requires that the minimum number of observations in a node be 30 before attempting a split and that a split must decrease the overall lack of fit by a factor of 0.001 (cost complexity factor) before being attempted.

02-5-1 R 패키지의{rpart}의 rpart() 함수 이용

control= Tree 크기를 제한하는 옵션입니다

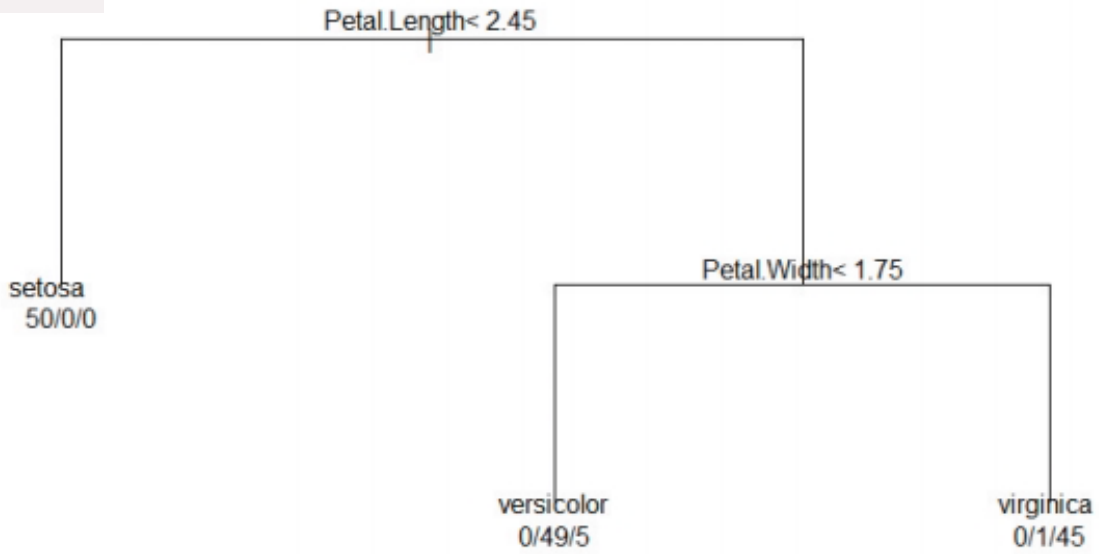
For example, control=rpart.control(minsplit=30, cp=0.001) requires that the minimum number of observations in a node be 30 before attempting a split and that a split must decrease the overall lack of fit by a factor of 0.001 (cost complexity factor) before being attempted.

- 과잉적합(overfitting)은 훈련 집합을 너무 완벽하게 처리하려다가 새로운 샘플에 대한 성능을 망치는 현상
- 모델링의 궁극적인 목표는 일반화(generalization) 능력, 즉 새로운 샘플에 대한 높은 성능 달성이므로 적당한 조건에서 멈추는 전략 사용



02-5-2 rpart()/plot() 함수 이용

```
> library(rpart)
> r = rpart(Species~., data = iris)
> par(mfrow = c(1, 1), xpd = NA)
> plot(r)
> text(r, use.n = TRUE)
```



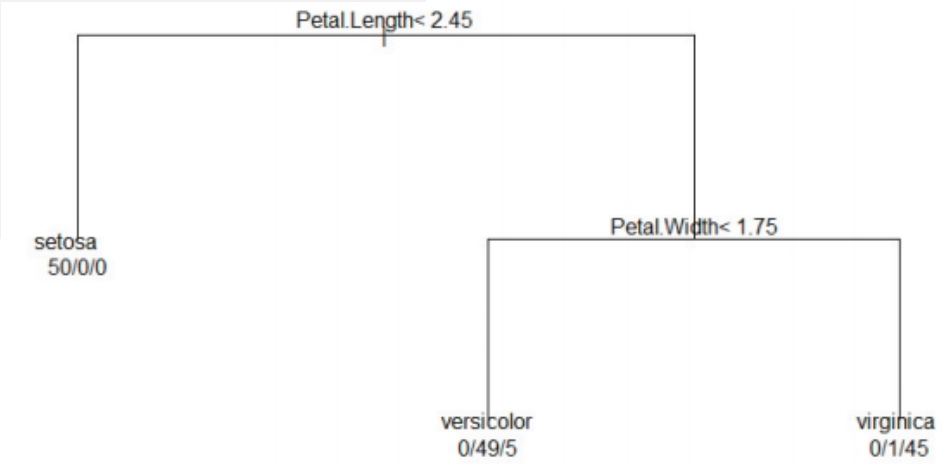
function	description
printcp(fit)	display cp table
plotcp(fit)	plot cross-validation results
rsq.rpart(fit)	plot approximate R-squared and relative error for different splits (2 plots). labels are only appropriate for the “anova” method.
print(fit)	print results
summary(fit)	detailed results including surrogate splits
plot(fit)	plot decision tree
text(fit)	label the decision tree plot

02-5-3 predict() 함수를 이용한 예측

```
> # 예측
> newd = data.frame(Sepal.Length = c(5.11, 7.01, 6.32), Sepal.Width = c(3.51,
3.2, 3.31), Petal.Length = c(1.4, 4.71, 6.02), Petal.Width = c(0.19, 1.4, 2.49))

> # 출력
> 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(r, newdata = newd)
  setosa versicolor virginica
1      1 0.00000000 0.00000000
2      0 0.90740741 0.09259259
3      0 0.02173913 0.97826087
```



예를 들어, 두 번째 샘플은 versicolor일 확률 0.907, virginica일 확률 0.093임을 나타냄

02-5-3 predict() 함수를 이용한 예측-계속

- 훈련 집합에 대한 예측
  - predict 함수를 사용하여 예측
  - type='class' 옵션은 부류를 출력함 (이 옵션이 없으면, type='prob'가 기본값이므로 부류에 속할 확률을 출력함)

```
> p = predict(r, iris, type = 'class')
> table(p, iris$Species)
```

예측값

p	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	49	5
virginica	0	1	45

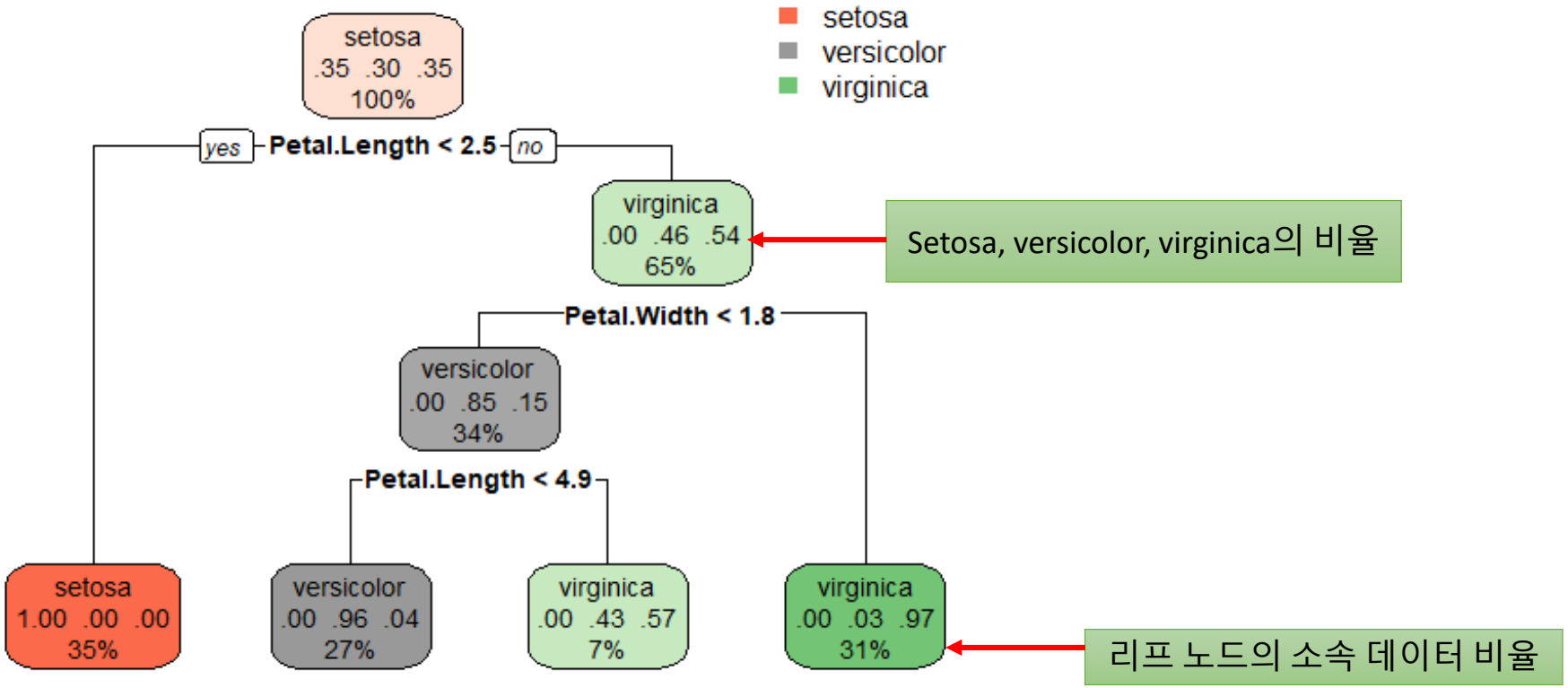
그라운드 트루스(정답)

혼동 행렬

- 혼동 행렬 (confusion matrix): 정확도 계산 가능
  - 부류별로 옳은 분류와 잘못된 분류의 상세 내용을 보여줌
  - 예를 들어, 50개의 versicolor를 49개는 옳게, 1개는 virginica로 잘못 분류함

02-5-4 rpart()/rpart.plot() 함수 이용

```
> library(rpart) #rpart()함수 포함 패키지
> library(rpart.plot) #rpart.plot()함수 포함패키지
> train <- sample(1:150, 100) #무작위로 100개 추출 (학습데이터)
> tree <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length +
                Petal.Width, data=iris, subset =train, method = "class")
> rpart.plot(tree)
```



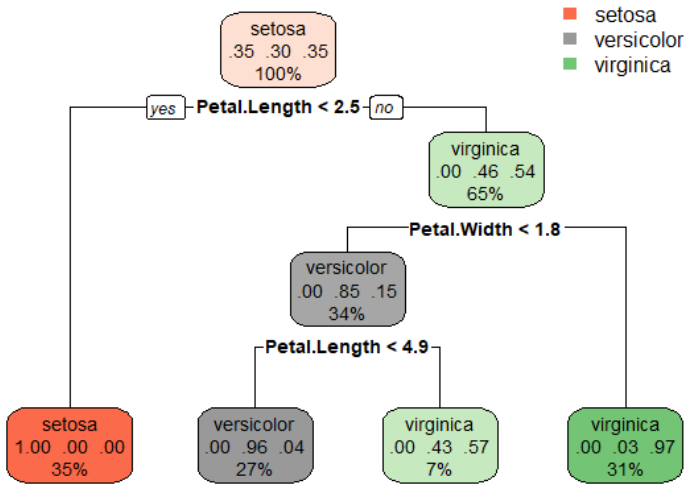
02-5-5 summary() 함수: 결정 트리 해석

```
> summary(tree)
Call:
rpart(formula = Species ~ Sepal.Length + Sepal.Width + Petal.Length +
    Petal.Width, data = iris, subset = train, method = "class")
  n= 100

          CP nsplit  rel error    xerror       xstd
1 0.53846154      0 1.00000000 1.1538462 0.06661734
2 0.36923077      1 0.46153846 0.4615385 0.07050116
3 0.01538462      2 0.09230769 0.1692308 0.04813689
4 0.01000000      3 0.07692308 0.1692308 0.04813689

Variable importance
 Petal.Width Petal.Length Sepal.Length  Sepal.Width
          33           31           21           14
```

1.0-0.4615=0.5384



- CP (Complexity Parameter): 복잡도, nsplit(분기회수)가 커지면 값이 줄어 든다. 최소값 0.01로 더 이상 분기하지 않음.
- rel_error (오류율), xerror (교차검증 오류율), xstd (교차검증오류의 표준편차) : pruning을 위한 최적의 lowest level 선택에 사용됨

prune(tree, cp, …): pruning 함수

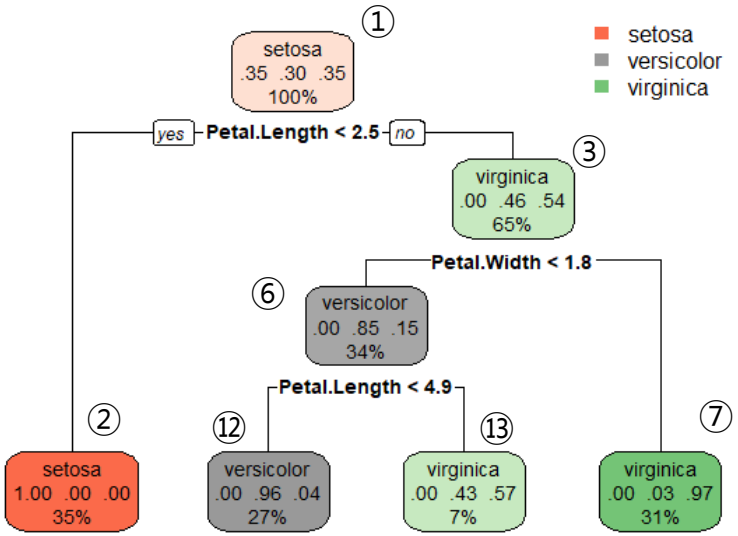
- Variable importance는 설명 변수의 중요성을 순서대로 보여줌. 4개 설명 변수 중에서 중요도가 높은 Petal.Width와 Petal.Length만 사용함

02-5-5 summary() 함수: 결정 트리 해석 - 계속

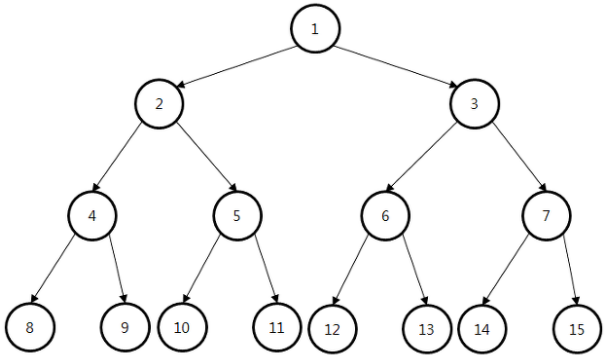
```
Node number 1: 100 observations,    complexity param=0.5384615
  predicted class=setosa      expected loss=0.65  P(node) =1
    class counts:    35    30    35
   probabilities: 0.350 0.300 0.350
  left son=2 (35 obs) right son=3 (65 obs)

Node number 2: 35 observations
  predicted class=setosa      expected loss=0  P(node) =0.35
    class counts:    35     0     0
   probabilities: 1.000 0.000 0.000

Node number 13: 7 observations
  predicted class=virginica   expected loss=0.4285714  P(node) =0.07
    class counts:     0     3     4
   probabilities: 0.000 0.429 0.571
```



노드 1, 2, 3, 6, 7, 12, 13에 대한 상세한 내용
예) 노드 13: 7개 샘플이 도달하고, virginica로 분류하고,
분류 확률은 (0.0,0.429,0.571)이라는 정보를 나타냄



요약

- **Decision Tree**
  - ✓CART 알고리즘
  - ✓rpart()를 이용한 모델 생성/예측
  - ✓결과 분석