3-2-3 의사결정나무

의사결정 나무(decision tree) 또는 나무(tree) 모형

- 의사결정 규칙을 **나무 구조로 나타내어 전체 자료를 몇 개의 소집단으로** 분류(classification)하거나 예측(prediction)을 수행하는 분석방법

특징

- 상위 노드로부터 하위노드로 나무 구조를 형성하는 데 단계마다 분류변수와 분류기준의 선택이 중요
- 상위 노드에서의 분류변수, 분류기준의 이 기준에 의해 분기되는 하위노드에서 **노드(집단) 내에서는 동질성, 노드(집단) 간에는 이질성이 가장 커지도록** 선택
- 나무 모형의 크기는 과대적합(또는 과소적합) 되지 않도록 합리적 기준에 의해 적당히 조절되도록
- 나무 구조는 연속적으로 발생하는 의사결정 문제를 **시각화한 의사결정이 이뤄지는 시점&성과를 한눈에** 파악
- 계산 결과가 의사결정 나무에 직접 나타나기 때문에 해석이 간편
- 의사결정나무는 **주어진 입력값에 대하여 출력 값을 예측하는** 모형으로 분류 나무와 회귀 나무 모형

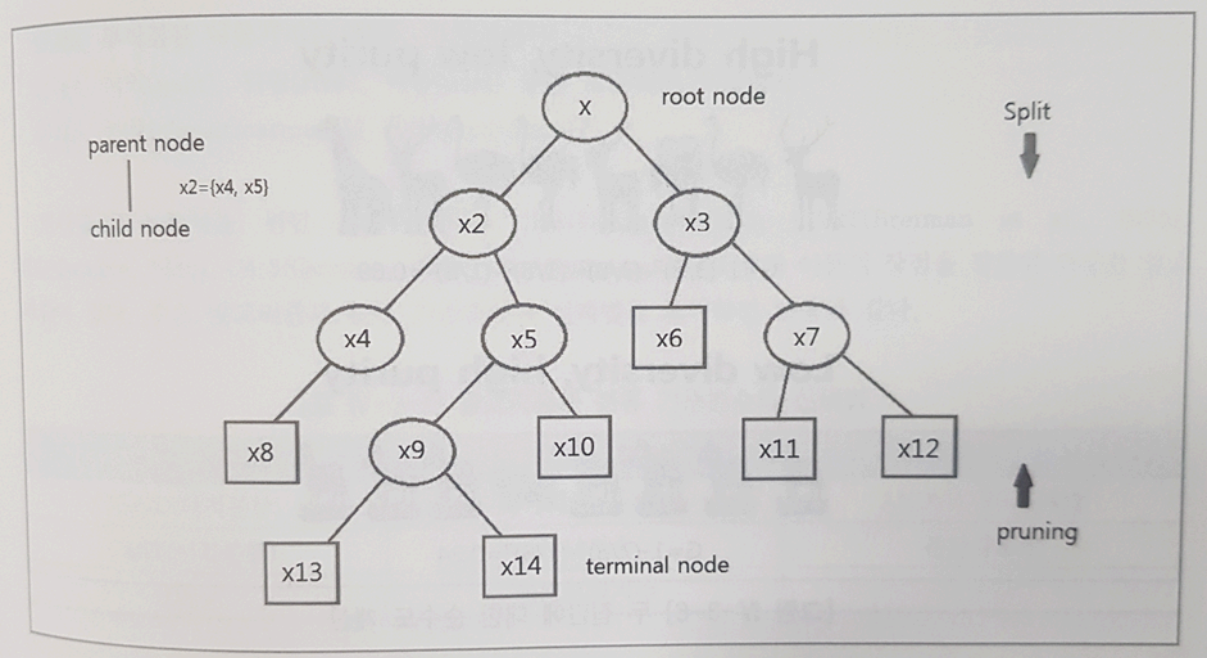
가지분할(split): 나무의 가지를 생성하는 과정

가지치기(pruning): 생성된 가지를 잘라내어 모형을 단순화하는 과정

의사결정 나무 구성요소

- **뿌리마디**(root node)**:** 시작되는 마디로 전체 자료 포함
- **자식마디**(child node)**:** 하나의 마디로부터 분리되어 나간 2개 이상의 마디들
- **부모마디**(parent node)**:** 주어진 마디의 상위 마디
- **끝마디,** 최종 마디(terminal node)**:** 자식이 없는 마디, 더 이상 분기되지 않는 마디

- **중간마디**(non-terminal node)**:** 부모와 자식마디가 모두 있는 마디
- **가지:** 뿌리 마디로부터 끝마디까지 연결된 마디들
- **깊이:** 뿌리 마디부터 끝마디까지 중간 마디의 수



[그림 Ⅳ-3-5] 의사결정나무의 구조

분류나무와 회귀나무

의사결정나무는 목표변수가 이산형인 경우의 분류나무(classification tree)와 목표변수가 연속형인 경우의 회귀나무(regression tree)로 구분

분할 기준

목표변수가 이산형인 분류나무의 경우 상위노드에서 가지분할을 수행할 때 분류(기준)변수와 분류기준값의 선택 방법으로

- 카이제곱 통계량(Chi-square statistic)의 p-값
- 지니 지수(Gini index)

$$G = 1 - \sum_i^c p_i^2, \quad 0 \le G \le \frac{1}{2}$$

- 엔트로피 지수(entropy index) 등 사용

$$E = -\sum_i^c p_i \log_2 p_i, \quad 0 \leq E \leq 1$$

- E: 주어진 셋 S의 엔트로피 값
- pi: 셋 S에서 i-번째 클래스의 확률
- n: 클래스의 수

지니 지수의 특징:

- **지니 지수가 0**일 경우, 노드는 **순수(pure)** 상태
  - 즉, 노드에 있는 데이터가 모두 동일한 클래스에 속함
- **지니 지수가 클수록** 노드는 더 **불순**한 상태
  - 클래스들이 고르게 분포할수록 지니 지수는 커짐

예시:

- 만약 노드에 **50%의 클래스 A**와 **50%의 클래스 B**가 있을 때, 지니 지수는 최대
- 만약 노드에 **100%의 클래스 A**만 있을 때, 지니 지수는 0

엔트로피 지수 vs 지니 지수

엔트로피의 특징:

- **엔트로피가 0**일 경우, 노드는 **순수(pure)** 상태
  - 즉, 노드에 있는 데이터가 모두 동일한 클래스에 속하는 상태
- **엔트로피가 클수록**, 해당 노드는 더 **불확실한 상태**
  - 모든 클래스가 고르게 분포하는 상태에서 엔트로피는 최대

예시:

- 만약 노드에 **50%의 클래스 A**와 **50%의 클래스 B**가 있을 때, 엔트로피는 최대가 됩니다(불확실성이 가장 크므로)
- 만약 노드에 **100%의 클래스 A**만 있을 때, 엔트로피는 0입니다.

엔트로피 지수 vs 지니 지수

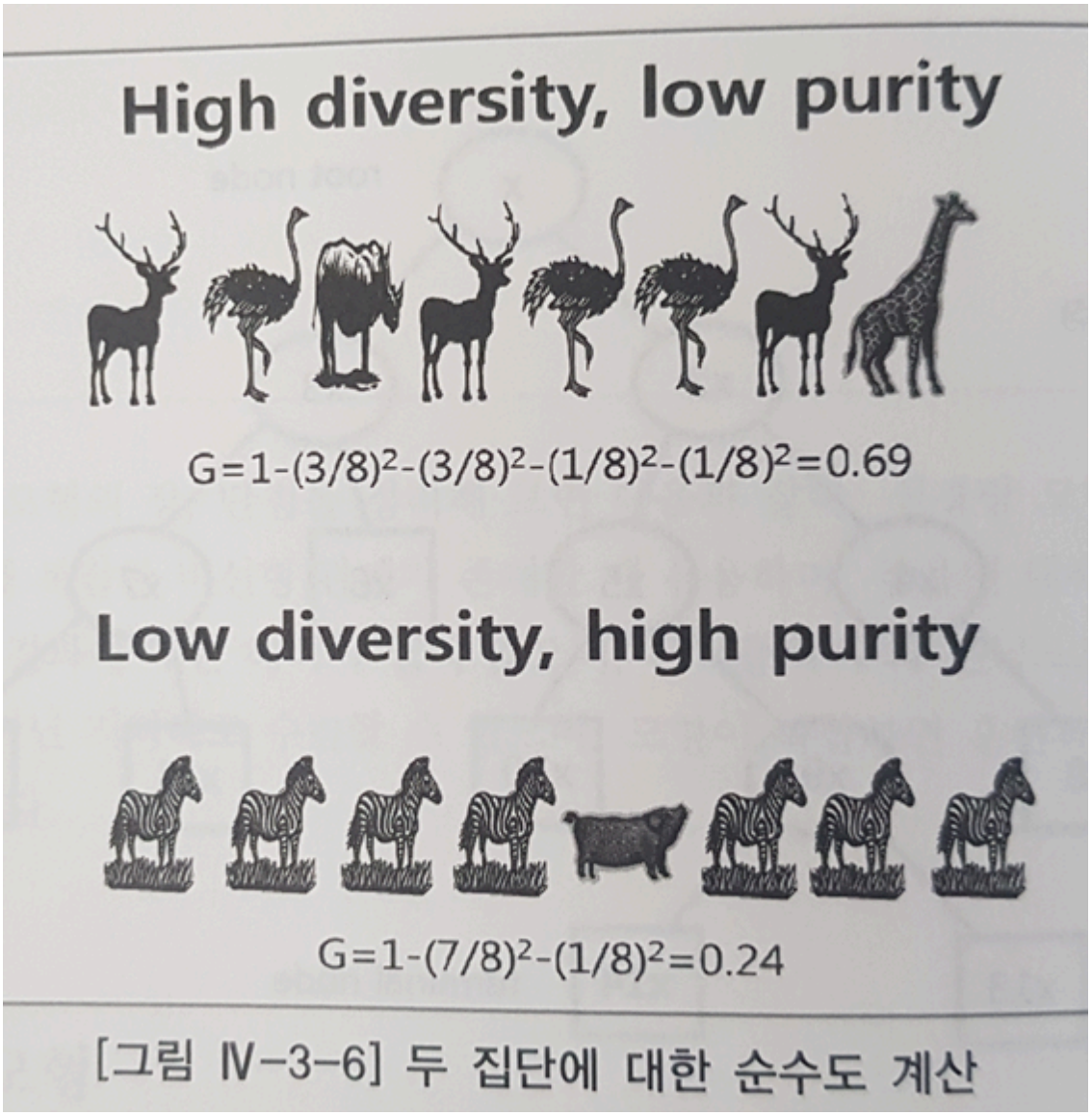
특징	엔트로피 지수 (Entropy)	지니 지수 (Gini Index)
목적	불확실성 또는 정보량 측정	불순도 측정
수식		
최소 값	0 (순수 노드)	0 (순수 노드)
최대 값	1 (모든 클래스가 동일한 확률로 분포)	0.5 (두 클래스가 동일한 확률로 분포)
효율성	계산이 다소 복잡 (로그 계산 필요)	계산이 간단하고 빠름
선호도	보통 더 복잡한 문제에서 사용	주로 빠른 계산을 요하는 문제에서 사용

선택된 기준에 의해 분할이 일어날 때,

- 카이제곱통계량의 p-값은
    - 그 값이 작을수록 자식 노드내의 불확실성(이질성)이 큼을 나타내며
- 자식노드에서의 지니 지수나 엔트로피 지수 역시
    - 그 값이 클수록 자식노드 내의 이질성이 큼을 의미
    - 따라서 이 값들이 가장 작아지는 방향으로 가지분할을 수행하게 된다.
        - 자식 노드 내부는 동질성이 높아지도록 해야 하므로

예를 들어, 아래의 그림에서 두 노드(집단)에 대한 지니 지수는 다음과 같이 계산된다.

- 지니 지수의 값이 클수록 이질적이며 순수도(purity)가 낮다고 할 수 있다.



[그림 Ⅳ-3-6] 두 집단에 대한 순수도 계산

불확실성 측정지표(uncertainty measure)인 지니 지수와 엔트로피 지수에 대한 정의는 다음과 같다.

두 지수의 값의 범위는 다르나, 해석은 값의 크기에 따라 유사하다.

- 지니 지수:

$$G = 1 - \sum_i^c p_i^2, \quad 0 \leq G \leq \frac{1}{2}$$

- 엔트로피 지수:

$$E = -\sum_i^c p_i \log_2 p_i, \quad 0 \leq E \leq 1$$

위 식에서 c는 목표변수의 범주의 수이다.

지니지수

범주 간의 불순도를 나타내는 방법

- 지니지수의 값이 클수록 이질적, 값이 작을수록 순도(동질성)가 높음

$$G = 1 - \sum p_i^2$$

엔트로피 지수

열역학에 쓰는 개념으로 무질서 정도에 대한 척도

- 엔트로피 지수의 값이 클수록 순도도가 낮음
- 엔트로피 지수가 가장 낮은 예측변수와 이때의 최적분리 규칙에 의해 자식마디 형성

$$E = -\sum_{i=1}^{c} p_i \log_2 p_i$$

- E: 엔트로피
- c: 클래스의 수
- pi: 클래스 i에 속할 확률 (즉, 비율)
- log2pi: 밑이 2인 로그

두 지수 모두 분류 문제에서 가지 분할의 기준으로 사용되며

- 분할된 하위 노드의 지니 또는 엔트로피 값이 **가장 작아지는 방향**으로 나무가 성장함

목표변수가 연속형인 회귀나무의 경우

- 분류(기준)변수와 분류기준값의 선택방법으로
  - F-통계량, p-값, 분산의 감소량 등이 사용

각 값의 해석

- F-통계량은 일원배치법에서의 검정통계량으로 그 값이 클수록 오차의 변동에 비해 처리(treatment)의 변동이 크다는 것을 의미하며,
  - 이는 자식 노드(처리들) 간의 이질성이 크다는 의미이므로
  - 이 값이 커지는(p-값은 작아지는) 방향으로 가지분할을 수행(자식 노드를 생성)하게 됨
- 분산의 감소량(variance reduction)도
  - 이 값이 최대화 되는 방향으로 가지분할을 수행하게 된다.

의사결정나무의 분석과정

- 단계1. 목표변수와 관계가 있는 설명변수들의 선택
- 단계2. 분석목적과 자료의 구조에 따라 적절한 분리기준과 정지규칙을 정하여 의사결정나무의 생성을 수행
- 단계3. 부적절한 나뭇가지는 제거: 가지치기
- 단계4. 이익(gain), 위험(risk), 비용(cost) 등을 고려하여 모형평가
- 단계5. 분류(classification) 및 예측(prediction)

의사결정나무분석을 위한 알고리즘

- CHAID(Kass, 1980),
- CART(Breiman et al., 1984),

- ID3(Quinlan, 1986),
- C4.5(Quinlan, 1993),
- C5.0(Quinlan, 1998) 등
- 이들의 장점을 결합한 다양한 알고리즘이 있음

주요 알고리즘과 분류 기준변수의 선택법을 요약

- **알고리즘과 분류 기준변수의 선택법**

알고리즘 이름	주요 분할 기준	적용 분야	특징 및 설명
ID3	**정보 이득 (Information Gain)**	분류 (Classification)	- 엔트로피 감소가 큰 속성을 선택 - 다중 분기(다지선다) 허용 - 연속형 변수는 처리 어려움
C4.5	**정보 이득 비율 (Gain Ratio)**	분류 (Classification)	- ID3 개선 버전 - 연속형 속성 처리 가능
CART (Classification And Regression Trees)	**지니 지수 (Gini Impurity)** or **MSE**	분류 & 회귀 (Classification & Regression)	- 이진 분기만 허용 - 회귀일 땐 MSE(평균제곱오차) 기준 사용
CHAID (Chi-squared Automatic Interaction Detector)	**카이제곱 통계량 (Chi-square)**	분류 (범주형 변수 위주)	- 다지선다 허용 - 카이제곱 검정 기반으로 통계적으로 유의한 분할만 사용
Regression Tree	**분산 감소 (Reduction in Variance)**	회귀 (Regression)	- 연속형 목표변수 예측 전용 - 각 노드에서 분산이 가장 많이 줄어드는 지점 기준

간단한 샘플 예제

지니 지수와 엔트로피 지수 계산 문제

📌 문제

어떤 노드에 데이터가 총 10개 있다고 하자. 그 중

- 클래스 A: 6개
- 클래스 B: 4개

이때,

1. 지니 지수를 구하시오.
2. 엔트로피 지수를 구하시오.
3. 어떤 지수가 더 순수(purity)가 높은지를 판단하시오.

✅ 계산 및 해설

1. 지니 지수 계산

$$G = 1 - \sum_{i=1}^{c} p_i^2$$

여기서 $p_A = \frac{6}{10} = 0.6$, $p_B = \frac{4}{10} = 0.4$

$$G = 1 - (0.6)^2 - (0.4)^2 = 1 - 0.36 - 0.16 = 0.48$$

2. 엔트로피 지수 계산

$$E = -\sum_{i=1}^{c} p_i \log_2 p_i$$

$$E = -(0.6 \log_2 0.6 + 0.4 \log_2 0.4)$$

$$\log_2 0.6 \approx -0.737, \quad \log_2 0.4 \approx -1.322$$

$$E \approx -(0.6 \times -0.737 + 0.4 \times -1.322) = -(-0.442 + -0.529) = 0.971$$

3. 해석 및 결론

- 지니 지수: 0.48
- 엔트로피 지수: 약 0.971
- 지니 지수는 **최대 0.5**, 엔트로피는 **최대 1**이므로,

  이 지니 값 0.48은 거의 혼합 상태, 엔트로피도 거의 최대에 가까움.

→ 즉, **순수도가 낮은 집단**이다.

→ 이 노드는 **더 분할할 필요가 있다**.

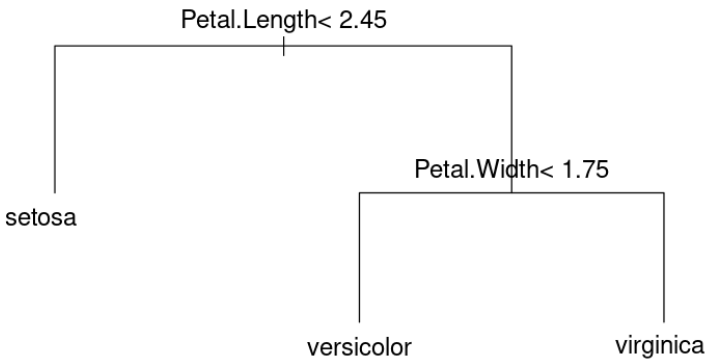
실습

rpart 패키지 개요

- `rpart`는 **Recursive Partitioning and Regression Trees**의 약자야.
- CART 알고리즘을 기반으로 함.
- **분류(classification)** 및 **회귀(regression)** 분석을 위한 트리 구조 모델을 만든다.
- **내장된 기본 함수**는 다음과 같아:
  - `rpart()` : 모델 생성
  - `printcp()` : 복잡도 파라미터 테이블 출력
  - `plotcp()` : 복잡도 시각화
  - `predict()` : 예측
  - `plot()` + `text()` : 트리 시각화

```
c ← rpart(Species~., data=iris)
c
```

```
plot(c, compress=T, margin=0.3)
text(c, cex=1.2)
```




```
> c ← rpart(Species~., data=iris)
> c
n= 150

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

1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
  2) Petal.Length< 2.45 50   0 setosa (1.00000000 0.00000000 0.0000000
0) *
  3) Petal.Length>=2.45 100  50 versicolor (0.00000000 0.50000000 0.500
00000)
    6) Petal.Width< 1.75 54   5 versicolor (0.00000000 0.90740741 0.092592
59) *
    7) Petal.Width>=1.75 46   1 virginica (0.00000000 0.02173913 0.9782608
7) *
```

요약:

- 이 코드는 R에서 **iris 데이터셋**을 이용해 `Species`를 예측하는 **분류 의사결정나무 모델** 생성
- 트리 결과는 **꽃잎 길이(Petal.Length)**와 **꽃잎 너비(Petal.Width)** 두 변수를 기준으로 세 갈래의 단말 노드로 분류한 구조야.
- 정확도도 아주 좋아. 특히 `Petal.Length < 2.45`일 때는 전부 setosa로 100% 정확!

🌳 코드 및 출력 해석: `rpart()` 모델 결과

📌 코드 요약

```
c ← rpart(Species~., data=iris)
```

- `Species`를 **종속변수**로 하고 나머지 모든 변수(`.`)를 **독립변수**로 사용함
- 데이터셋은 **내장 iris (꽃 데이터, 총 150개 샘플)**

🧾 결과 해석 (노드 번호 중심으로)

출력 포맷:

```
노드번호) 분할조건, 샘플 수, 오분류 수, 예측 클래스 (클래스별 확률)
```

🌱 1) root 노드

```
1) root 150 100 setosa (0.3333 0.3333 0.3333)
```

- 전체 샘플 수: 150
- 오분류 수 (loss): 100
  - 즉, `setosa`로 일단 예측했으니 versicolor와 virginica는 오분류됨
- 클래스 확률: 1/3씩 균등 (setosa, versicolor, virginica)

🌿 2) 노드 – Petal.Length < 2.45

```
2) Petal.Length < 2.45 50 0 setosa (1 0 0) *
```

- 샘플 수: 50
- 오분류 없음! 완벽하게 `setosa`로만 구성
- *은 단말 노드(더 이상 분할되지 않음)

🧠 해석: 꽃잎 길이가 2.45cm보다 짧으면 100% setosa

🌿 3) Petal.Length ≥ 2.45

```
3) Petal.Length >=2.45 100 50 versicolor (0 0.5 0.5)
```

- 샘플 수: 100
- 50개는 오분류 (versicolor로 예측했으나 절반은 virginica)
- 클래스 확률이 0.5 vs 0.5 → 불순도 높음 → 추가 분기 필요!

🌿 6) Petal.Width < 1.75

```
6) Petal.Width < 1.75 54 5 versicolor (0 0.9074 0.0926) *
```

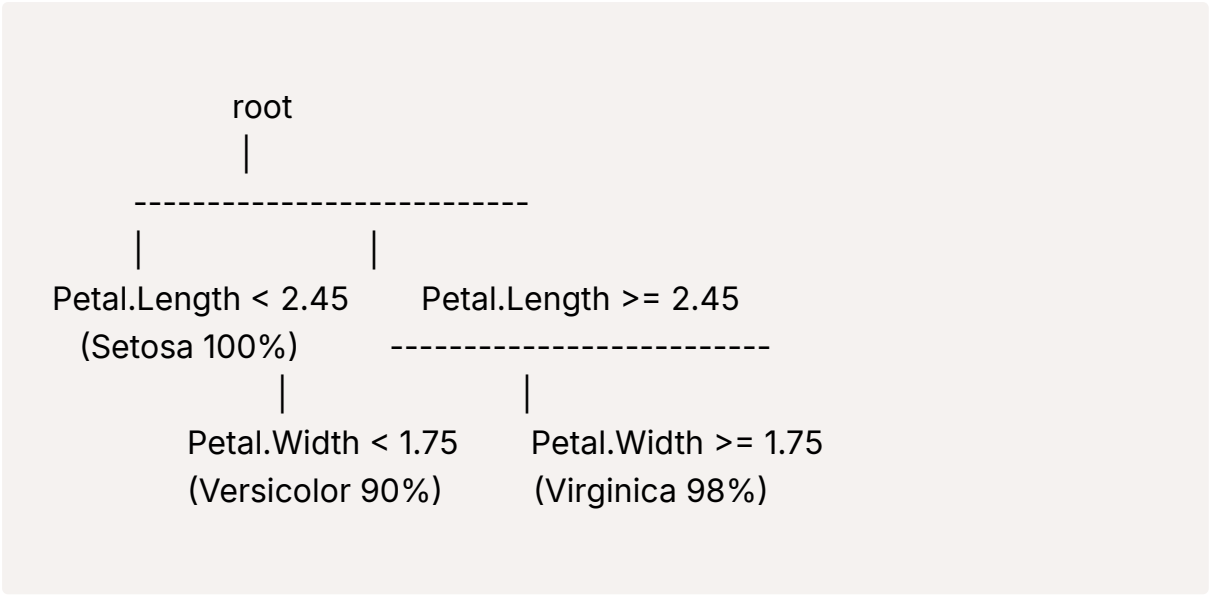
- 샘플 수: 54
- 오분류 수: 5
- 대부분이 `versicolor` → 거의 정확

🌿 7) Petal.Width ≥ 1.75

```
7) Petal.Width >=1.75 46 1 virginica (0 0.0217 0.9783) *
```

- 샘플 수: 46
- 오분류 수: 단 1개!
- 거의 전부가 `virginica`

📈 구조 요약 (트리 구조)



🧠 노드 1에 대한 정보 줄, 완전 분석해줄게:

```
1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
```

각 항목이 의미하는 바는 아래와 같아:

항목	의미
`1)`	노드 번호 (root는 항상 1번)
`root`	루트 노드라는 뜻 (전체 데이터 포함하는 가장 위 노드)
`150`	이 노드에 속한 총 샘플 수 = **iris 전체 데이터**
`100`	**오분류 샘플 수 (loss)** → "setosa"로 예측했는데 틀린 개수
`setosa`	이 노드에서 가장 많은 클래스 = **예측 클래스 (yval)** **사실 모두 50개 이므로** 팩터의 첫 값인 setasa를 선정
`(0.3333 0.3333 0.3333)`	클래스별 확률 (setosa, versicolor, virginica)

🔍 왜 loss = 100일까?

루트 노드는 **아직 분기 안 된 상태**야. 즉,

- 전체 데이터 150개 중
    - setosa 50개

- versicolor 50개
- virginica 50개

  → 각 클래스가 **1/3씩 균등**

rpart는 이 상태에서 가장 많은 클래스(여기선 아무거나 가능하지만 setosa로 선택)를 예측값으로 사용해.

하지만 setosa는 **전체의 50개뿐**이니까,

나머지 100개(versicolor + virginica)는 잘못 예측하게 돼.

➡️ 그래서 **loss = 100**

🎯 결론적으로

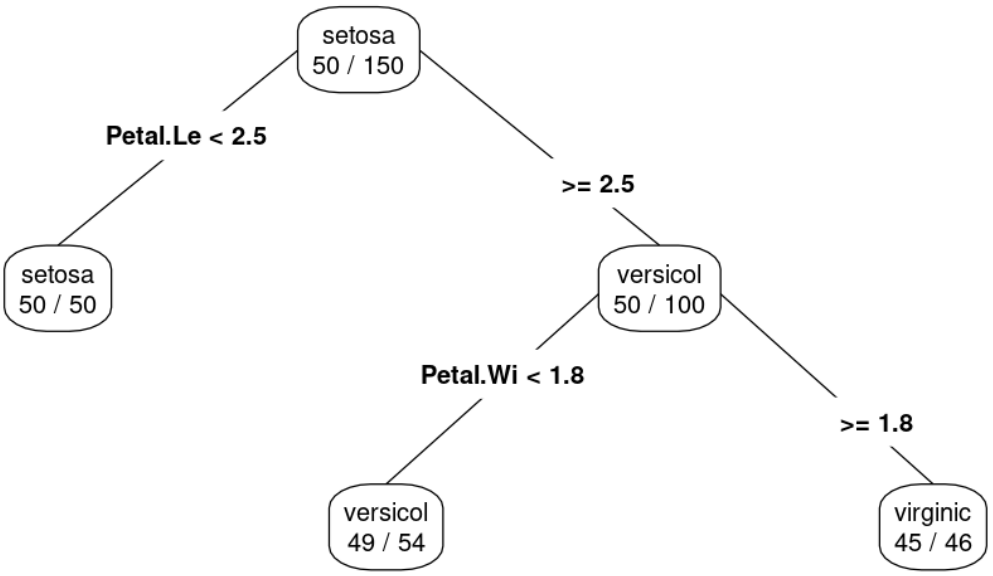
- 루트 노드에서는 전체 150개 중 **1/3씩 섞여 있음**
- `setosa`로 예측하면 100개는 틀림
- 그래서 rpart는 분기(split)를 통해 이 불순도(혼합상태)를 **낮추려고 노력**함

🤖 이 줄을 의사결정나무식으로 말하면:

> "현재 노드에 150개 데이터가 있고, 일단 setosa라고 예측하면 100개가 틀립니다. 세 클래스의 비율은 정확히 1/3씩이고요."

코드

```
install.packages("rpart.plot")
library(rpart.plot)
prp(c, type=4, extra=2)
```



🧾 1. 코드 해석

```
install.packages("rpart.plot")  # 시각화용 패키지 설치
library(rpart.plot)             # 패키지 불러오기

prp(c, type = 4, extra = 2)     # 트리 시각화 실행
```

- `prp()`는 `rpart.plot` 패키지의 함수로 트리를 예쁘게 그려줘
- `type = 4`: 예측된 클래스 이름을 노드 위에 표시
- `extra = 2`: 각 노드에 **예측 클래스 / 샘플 수** 표시

🌳 2. 시각화된 트리 해석

🔷 루트 노드 (맨 위)

```
setosa
50 / 150
```

- 전체 150개 중, 가장 많은 클래스가 `setosa` (동률이므로 factor 첫 번째로 선택됨)
- 자식 노드 분기 기준은 `Petal.Length < 2.5`

🔷 왼쪽 자식 노드

setosa
50 / 50

- 조건: `Petal.Length < 2.5`
- 이 노드에는 50개 모두 `setosa`
- 예측도 정확: 💯 완전 순수 노드

🔷 오른쪽 자식 노드

versicol
50 / 100

- 조건: `Petal.Length >= 2.5`
- 총 100개 샘플 중 `versicolor`가 절반이라 예측 클래스는 `versicol`

→ 여기서 다시 분기됨! 기준은 `Petal.Width`

▶ `Petal.Width < 1.8` → 왼쪽

versicol
49 / 54

- 거의 대부분이 `versicolor`
- 오차는 5개뿐 → **정확도 높음**

▶ `Petal.Width >= 1.8` → 오른쪽

virginic
45 / 46

- `virginica` 클래스가 거의 대부분
- 역시 매우 정확!

🧠 3. 트리 전체 구조 요약

```
plaintext
복사편집
        Petal.Length < 2.5
         /            \
   setosa(50/50)     Petal.Width < 1.8
                      /           \
              versicol(49/54)   virginic(45/46)
```

트리 깊이도 얕고, 분기 기준도 단순한데 정확도 엄청 높아!

→ 이게 바로 iris 데이터가 교과서적인 이유야 🌼

🎯 모델 성능 분석 (간단 정리)

노드	조건	예측	정확도
root	전체	setosa	33% (50/150)
1	Petal.Length < 2.5	setosa	💯 (50/50)
2	Petal.Length ≥ 2.5	versicol	50% (50/100)
2-1	Petal.Width < 1.8	versicol	👍 (49/54)
2-2	Petal.Width ≥ 1.8	virginic	👍 (45/46)

종료