[6주차] 5/10
결정 트리 (Decision Tree)

1기 권지수
1기 문소연
1기 최주희

EURON

목차

6.1 결정 트리 학습과 시각화

결정 트리(decision tree)

- 학습을 통해 데이터의 규칙을 자동으로 찾아내 트리(Tree)기반의 분류 규칙을 만드는 것
- If/else 기반으로 표현
- 분류, 회귀, 다중출력 작업이 가능한 알고리즘
- 랜덤 포레스트(7장)의 기본 구성 요소

6.1 결정 트리 학습과 시각화

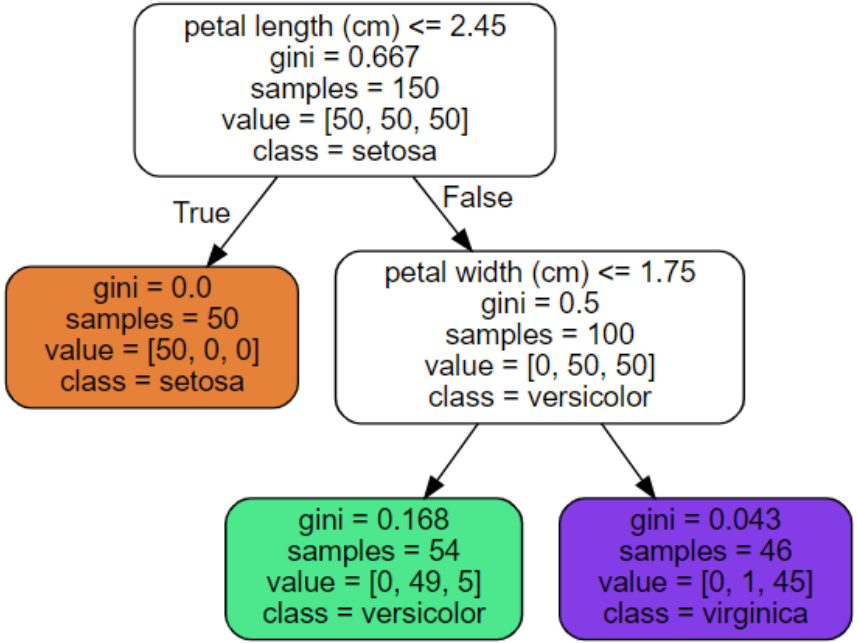
결정 트리 학습시키기

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] #꽃잎의 길이와 너비
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(X, y)
```

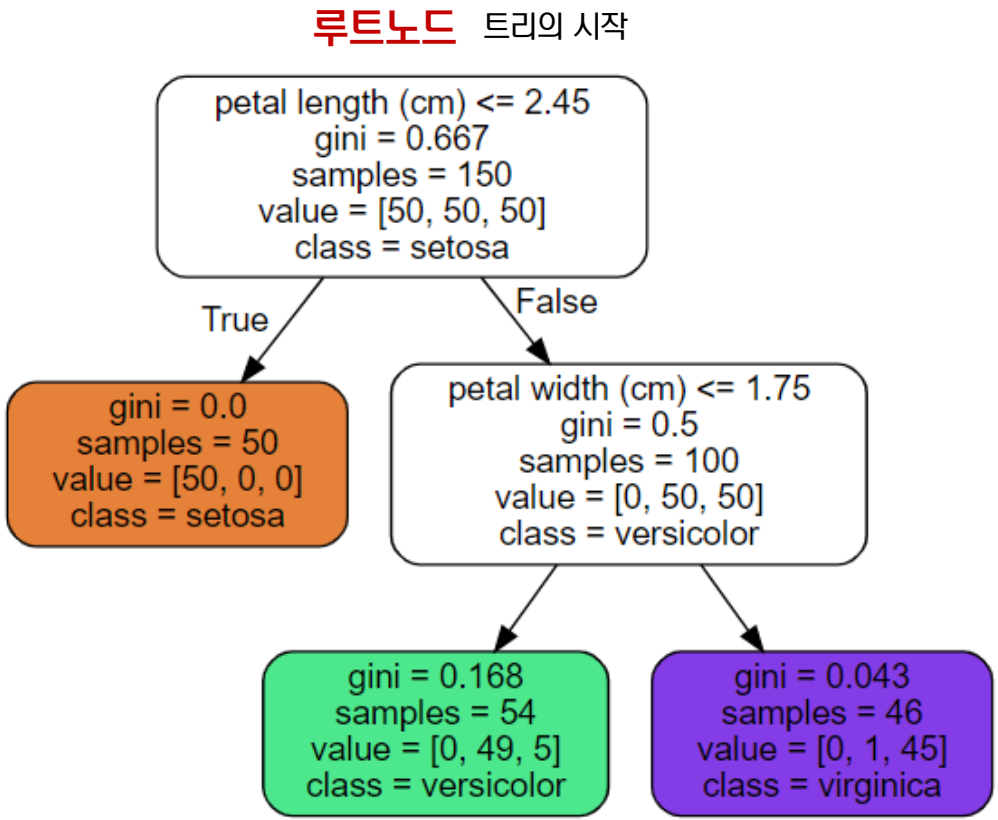
6.2 예측하기

결정 트리 시각화

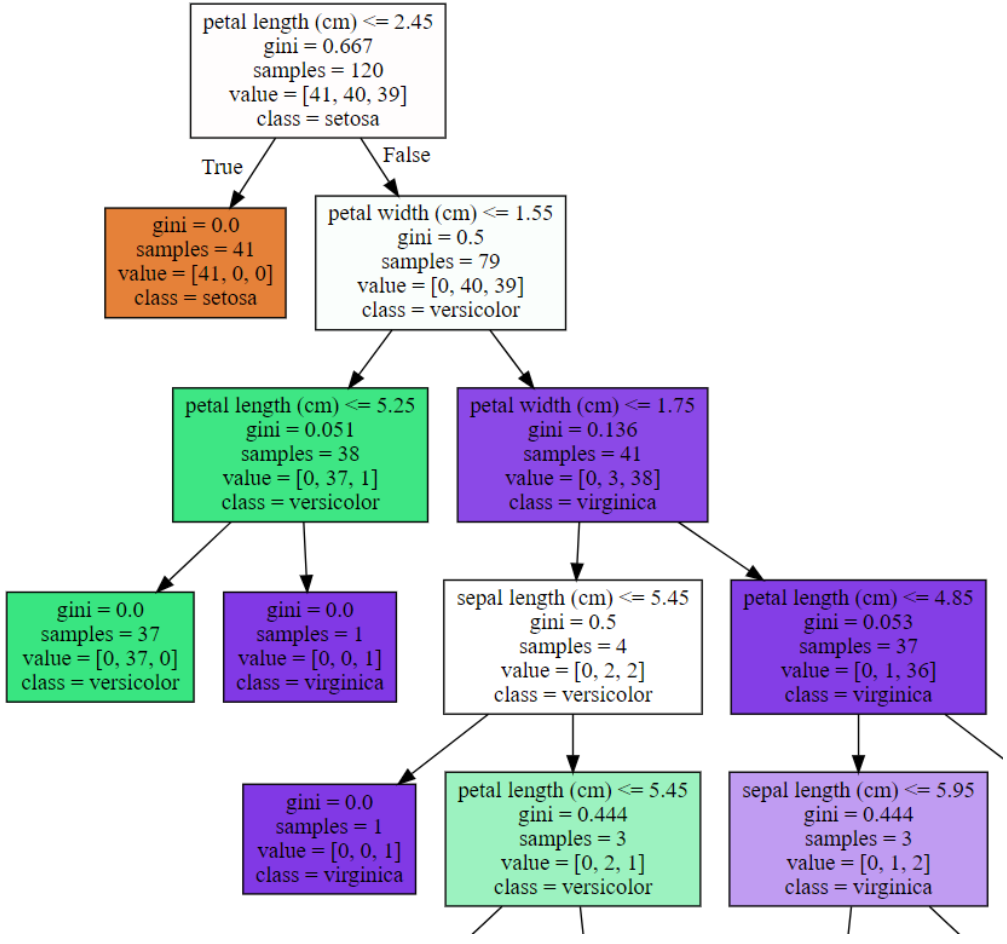
Export_graphiviz() 함수를 사용하여 결정 트리 시각화



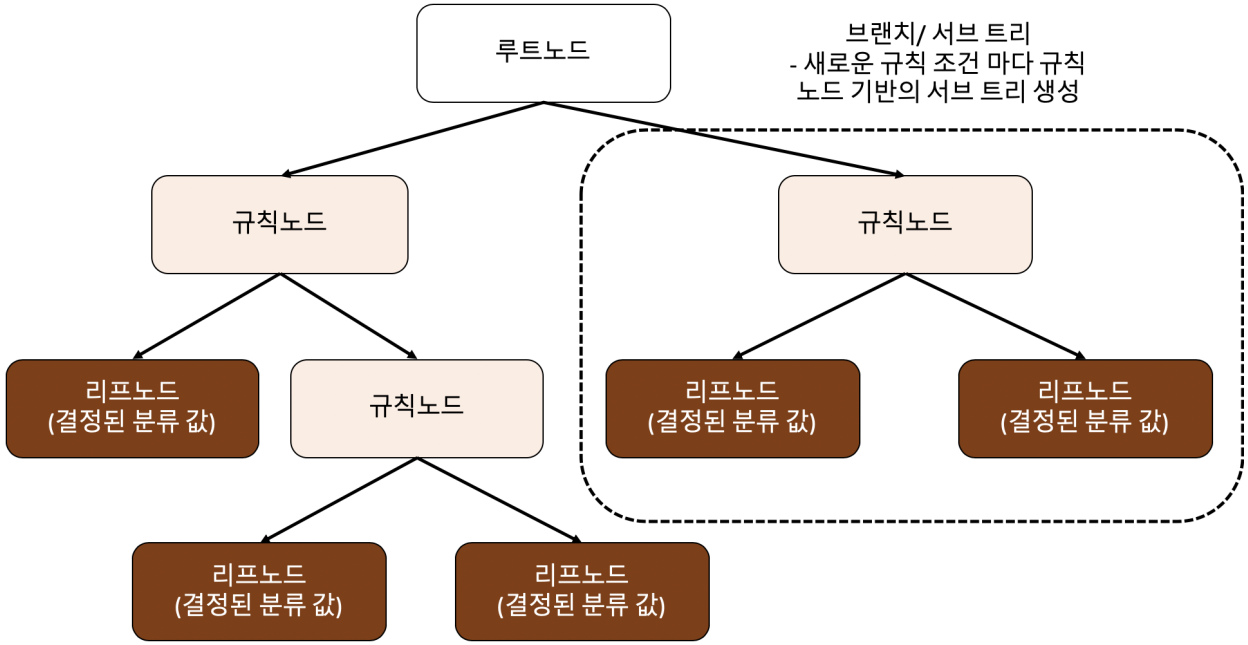
6.2 예측하기


루트노드
트리의 시작
petal length (cm) <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
리프노드
자식 노드를
가지지 않음
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
Depth 1
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
Depth 2

6.2 예측하기

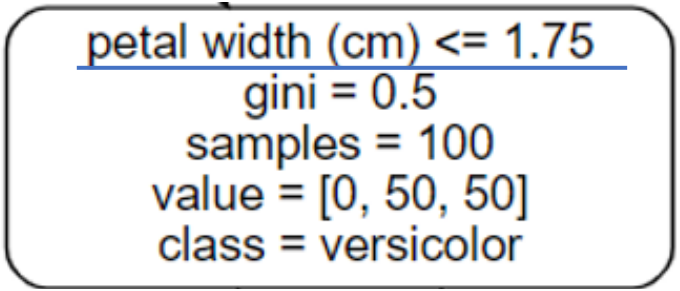

petal length (cm) <= 2.45
gini = 0.667
samples = 120
value = [41, 40, 39]
class = setosa
True
False
gini = 0.0
samples = 41
value = [41, 0, 0]
class = setosa
petal width (cm) <= 1.55
gini = 0.5
samples = 79
value = [0, 40, 39]
class = versicolor
petal length (cm) <= 5.25
gini = 0.051
samples = 38
value = [0, 37, 1]
class = versicolor
petal width (cm) <= 1.75
gini = 0.136
samples = 41
value = [0, 3, 38]
class = virginica
gini = 0.0
samples = 37
value = [0, 37, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
sepal length (cm) <= 5.45
gini = 0.5
samples = 4
value = [0, 2, 2]
class = versicolor
petal length (cm) <= 4.85
gini = 0.053
samples = 37
value = [0, 1, 36]
class = virginica
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
petal length (cm) <= 5.45
gini = 0.444
samples = 3
value = [0, 2, 1]
class = versicolor
sepal length (cm) <= 5.95
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica

6.2 예측하기


루트노드
브랜치/ 서브 트리
- 새로운 규칙 조건 마다 규칙
노드 기반의 서브 트리 생성
규칙노드
규칙노드
리프노드
(결정된 분류 값)
규칙노드
리프노드
(결정된 분류 값)
리프노드
(결정된 분류 값)
리프노드
(결정된 분류 값)
리프노드
(결정된 분류 값)

6.2 예측하기

규칙



식 6-1: 지니 불순도

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

1-(50/100)^2-(50/100)^2 = 0.5

samples

현 규칙에 해당하는 데이터 수

gini

불순도 측정. 모두 같은 클래스로 분류된다면 순수(gini=0)하다고 함
순수 노드라면 하위 노드로 나눌 수 없음

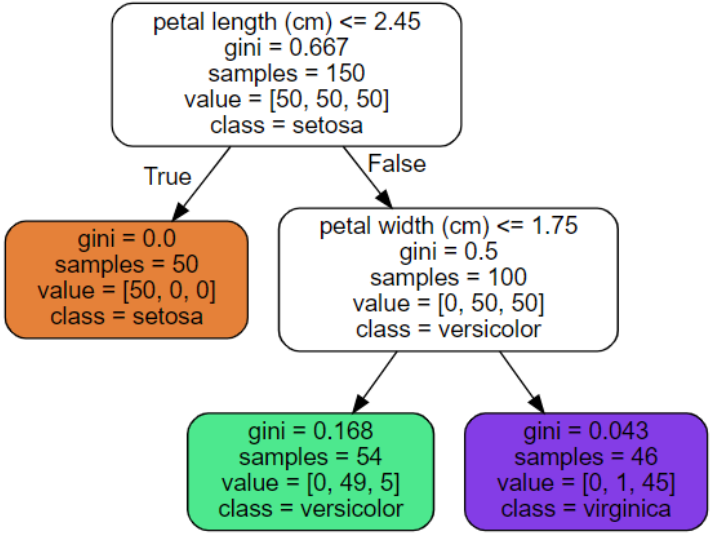
value

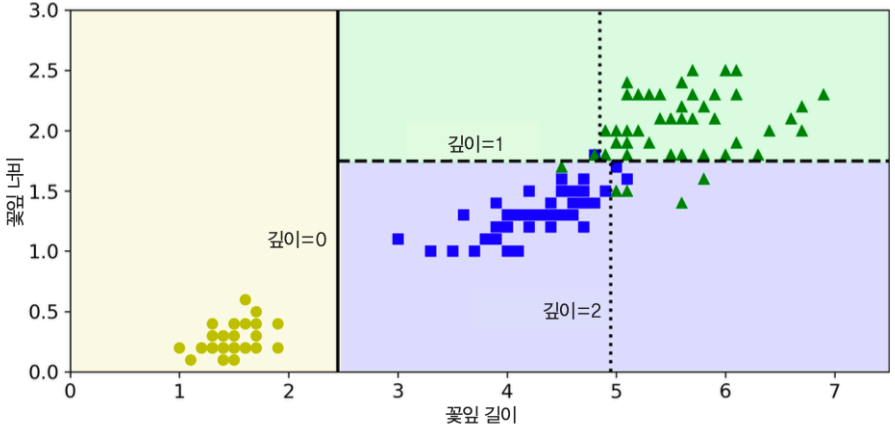
클래스 값 기반의 데이터 건수

class

가장 많은 건수의 클래스

6.2 예측하기


petal length (cm) <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica

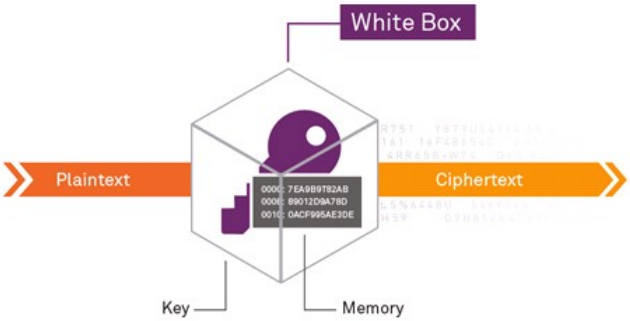

깊이=1
깊이=0
깊이=2
꽃잎 너비
꽃잎 길이
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7

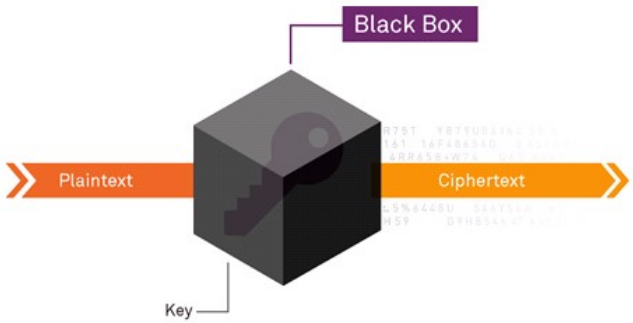
6.2 예측하기

결정 트리의 장/단점

- 쉽고 직관적
- 데이터의 균일도만 신경쓰기 때문에 데이터 전처리(피처 스케일링, 정규화 등)가 거의 필요하지 않음
- 트리의 깊이가 깊어질수록 과적합으로 예측 성능이 저하될 가능성이 높아짐
  > 크기를 사전에 제한하는 튜닝 필요!

6.2 예측하기





화이트박스/블랙박스 알고리즘

- 해석가능한 기계학습(Interpretable ML)은 시스템의 행동과 예측을 인간이 이해할 수 있게 하는 방법과 모델을 말함. 결정 트리, 규칙 목록(rule, lists), 회귀 알고리즘이 이에 해당됨.
- 블랙박스 알고리즘은 왜 그런 예측이 나왔는지 명확한 설명을 제공하지 않아 프로세스와 결과를 이해하기 어려움. DNN, 랜덤 포레스트, 그래디언트 부스팅 등이 해당됨.
- 최근 블랙박스 모델의 투명성을 높이고자 다양한 연구들이 진행되고 있으며, 학습 과정 후의 시각화도 투명성을 향상시키는 방법.

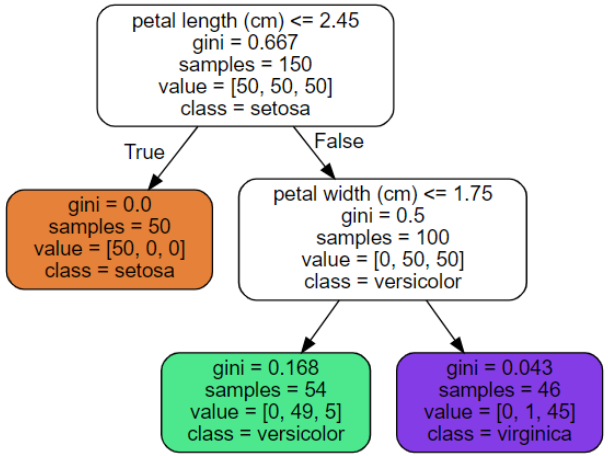
https://eair.tistory.com/3
https://www.sas.com/ko_kr/solutions/ai-mic/blog/interpretation-power.html

6.3 클래스 확률 추정

하나의 샘플이 특정 클래스 k에 속할 확률을 추정할 수 있음.

① 샘플의 리프 노드를 찾기 위해 트리를 탐색 ② 해당 노드의 클래스 k의 훈련 샘플 비율 반환

```
[ ]  tree_clf.predict_proba([[5, 1.5]])

     array([[0.        , 0.90740741, 0.09259259]])

[ ]  tree_clf.predict([[5, 1.5]])

     array([1])
```

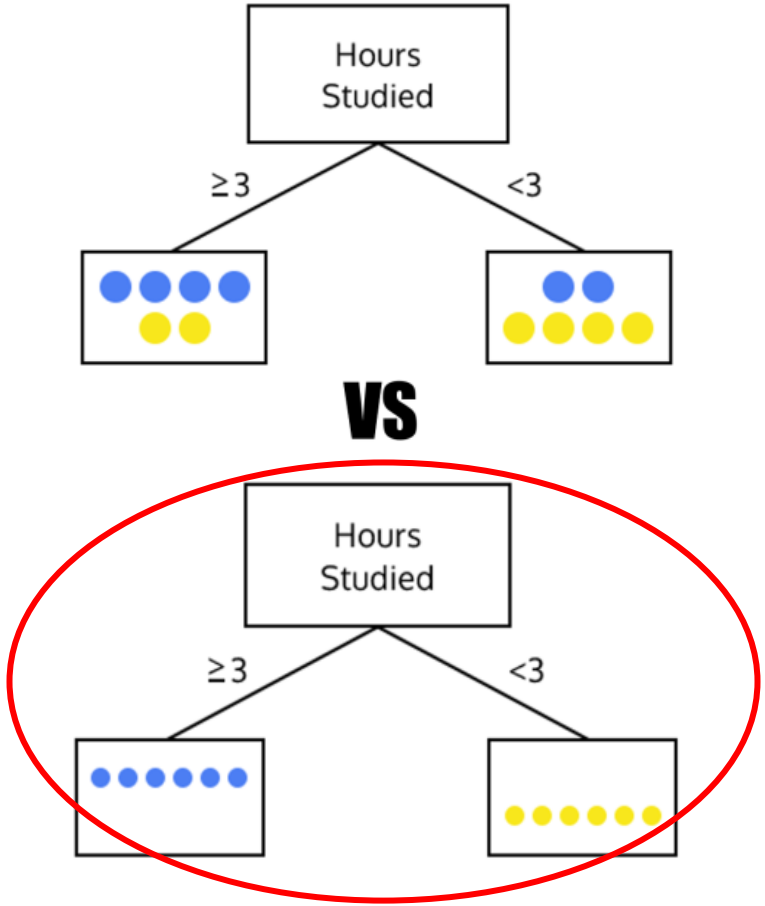


6.4+6.6 불순도(균일도)

교재 p.234~ 참고

Q. 불순도(균일도; impurity)란?

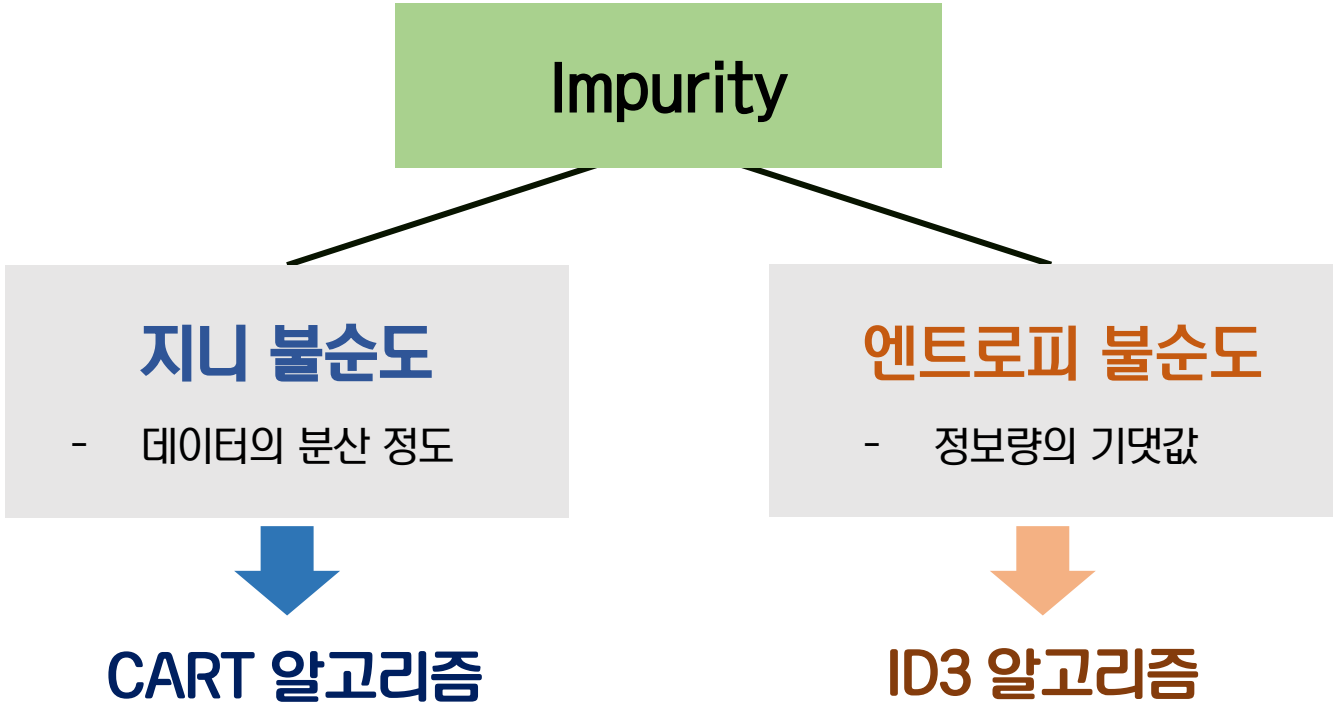
각 노드가 나타내는 클래스에
얼마나 다른 종류의 샘플들이 포함되어 있는지를
나타내는 정도



이미지 출처: http://hleecaster.com/ml-decision-tree-concept/

6.4+6.6 불순도(균일도)

** 불순도의 종류



6.4+6.6 결정 트리 알고리즘

Part 1. CART 알고리즘 (with 지니 불순도)

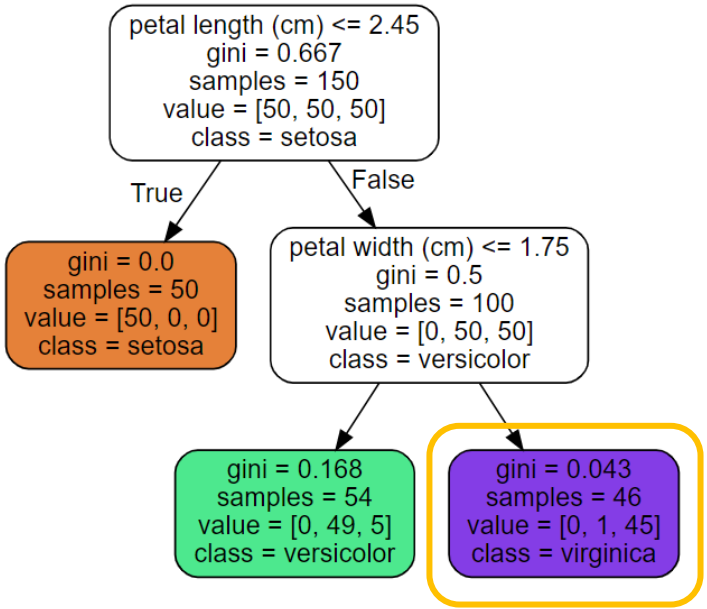
* 지니 불순도 (Gini index)

- 경제학 용어인 '지니 계수'의 의미 차용

참고) 불평등 지수를 나타내는 지니 계수;
0에 가까울수록 평등함, 1에 가까울수록 불평등함을 의미.

➔ 0에 가까울 수록 균일,
1에 가까울수록 불균일함

(데이터 간의 공통성, 흩어진 정도)

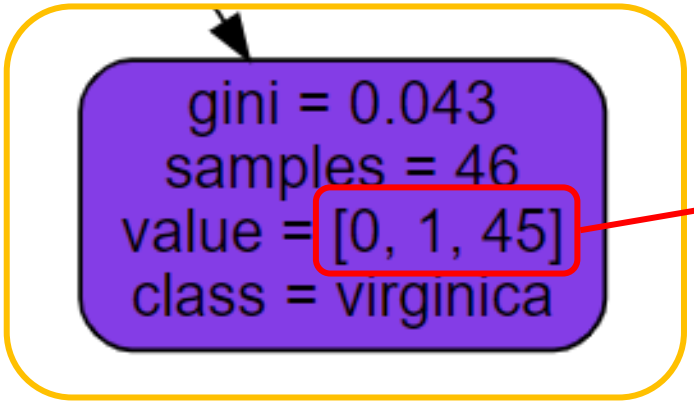


→ 사이킷런의 DecisionTreeclassifier에서
Criterion= 'gini' (default)

6.4+6.6 결정 트리 알고리즘

Part 1. CART 알고리즘 (with 지니 불순도)

* 지니 불순도 (Gini index)



식 6-1: 지니 불순도

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

$$G_2 = 1 - (\frac{0}{46})^2 - (\frac{1}{46})^2 - (\frac{45}{46})^2 = 0.0425$$

6.4+6.6 결정 트리 알고리즘

이미지 출처:
https://leedakyeong.tistory.com/entry/%EC%9D%98%EC%82%AC%EA%B2%B0%EC%A0%95%EB%82%98%EB%AC%B4Decision-Tree-CART-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EC%A7%80%EB%8B%88%EA%B3%84%EC%88%98Gini-Index%EB%9E%80

Part 1. CART 알고리즘 (with 지니 불순도)

CART 알고리즘

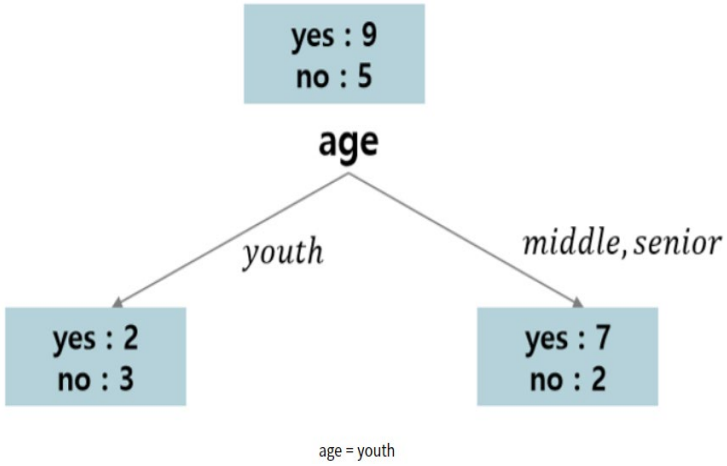
지니 계수를 이용해
한 가지 특성에 대한 임곗값을 정해야 함.
** Binary split !

식 6-2: 분류에 대한 CART 비용 함수

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

여기에서 $\begin{cases} G_{\text{left/right}} \text{는 왼쪽/오른쪽 서브셋의 불순도} \\ m_{\text{left/right}} \text{는 왼쪽/오른쪽 서브셋의 샘플 수} \end{cases}$

지니 불순도



6.4+6.6 결정 트리 알고리즘

Part 1. CART 알고리즘 (with 지니 불순도)

CART 알고리즘

지니 계수를 이용해
한 가지 특성에 대한 임곗값을 정해야 함.
** Binary split !

→ 이러한 과정을 각 노드마다 반복.
"max_depth(깊이의 최댓값) 옵션에 의해" or "분할할 만한 특성이 없으면" 중단

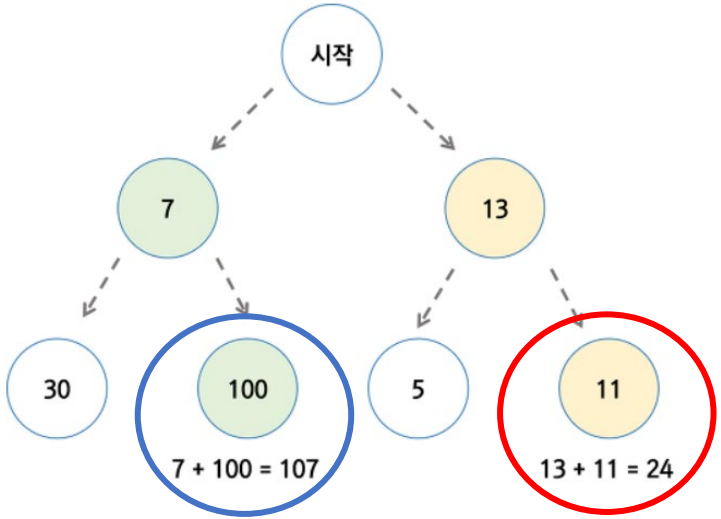
★ 참고) 탐욕적 알고리즘

각 단계에서 가장 최적의 선택을 하는 알고리즘. CART 알고리즘도 탐욕적 알고리즘에 해당됨!
그러나 최종적인 결과가 가장 좋은 선택이 아닐 수 있음.

이미지 출처: https://gomguard.tistory.com/119

6.4+6.6 결정 트리 알고리즘

Part 1. CART 알고리즘 (with 지니 불순도)



탐욕적 알고리즘은
각 단계에서 가장 최적의 선택을 하는 알고리즘.

➔ 따라서 탐욕적 알고리즘에 의하면 최종 답은 11.

But 전체적인 구조에서 최적의 해는 100.

∴ 탐욕적 알고리즘은 가장 좋은 해를 산출한다(x), '납득할 만한 괜찮은 해'를 산출한다(o)

6.4+6.6 결정 트리 알고리즘

Part 2. ID3 알고리즘 (with 엔트로피 불순도)

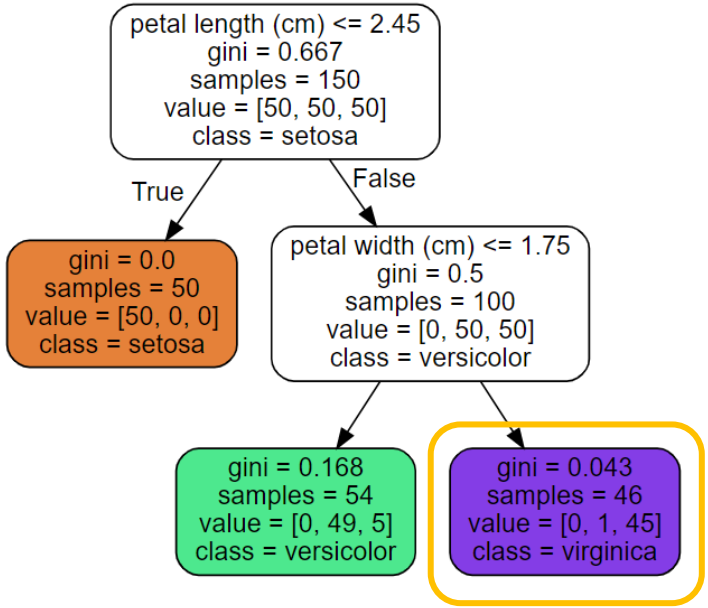
* 엔트로피 불순도 (entropy impurity)

- 열 역학 용어인 '엔트로피' 의 의미 차용

참고) 엔트로피는 분자의 무질서함을 나타내는 정도;
0에 가까울수록 분자가 안정되고 질서정연하며,
1에 가까울수록 분자가 불안정하고 무질서함

➔ 0에 가까울 수록 균일,
1에 가까울수록 불균일함

(정보량의 기댓값)



→ 사이킷런의 DecisionTreeclassifier에서
Criterion= 'entropy'

6.4+6.6 결정 트리 알고리즘

Part 2. ID3 알고리즘 (with 엔트로피 불순도)

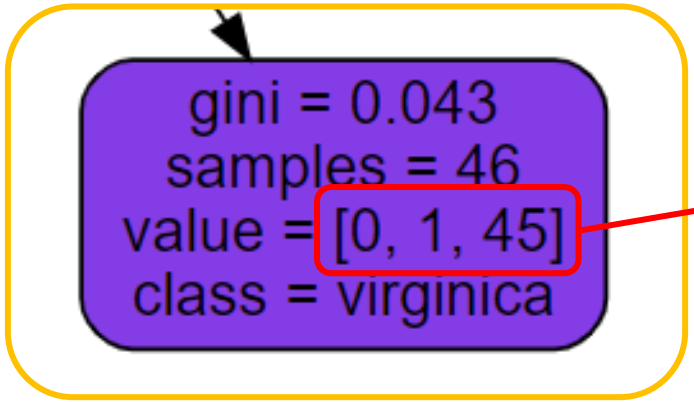
* 엔트로피 불순도 (entropy impurity)

(정보량의 기댓값)

$$Entropy(S) = \sum_{i=1}^{c} p_i * I(x_i)$$

(정보량)

$$I(x) = log_2 \frac{1}{p(x)}$$



$$H_2 = -\frac{1}{46} log_2(\frac{1}{46}) - \frac{45}{46} log_2(\frac{45}{46}) = 0.1511$$

6.4+6.6 결정 트리 알고리즘

Part 2. ID3 알고리즘 (with 엔트로피 불순도)

ID3 알고리즘

노드를 분할하는 것을 통해 엔트로피를 작게 하는 방향으로!

★ 정보 이득 (Information Gain)
: 분할 전의 엔트로피와 분할 후의 엔트로피의 차이

$$IG(S, A) = E(S) - E(S|A)$$

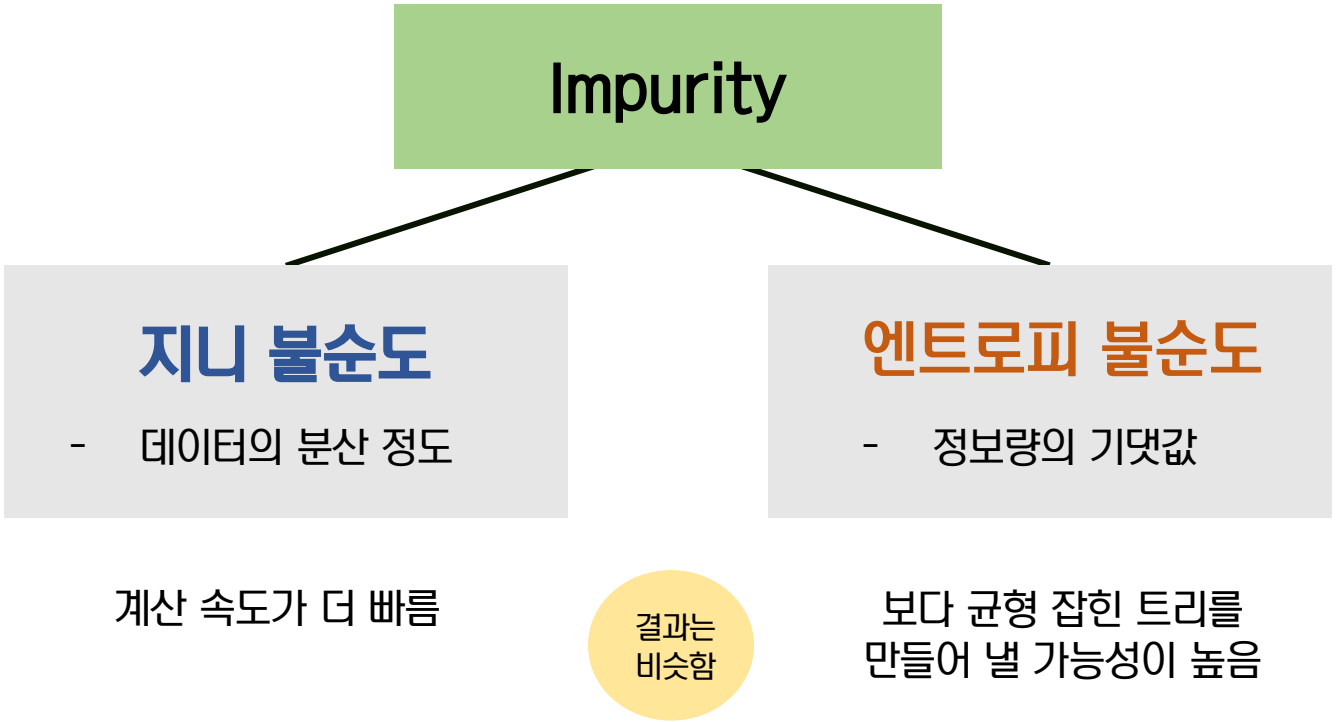
‘ 정보 이득이 크다 ’ = ‘ 분할 후 엔트로피가 많이 줄어들었다 ’

각 속성에 대해서 엔트로피를 계산해보거나, 속성별로 분할한 후 정보 이득을 계산해서
엔트로피가 가장 많이 줄어드는 방향을 선택할 수 있음.

* 참고: https://leedakyeong.tistory.com/entry/Decision-Tree%EB%9E%80-ID3-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98

6.4+6.6 불순도(균일도)

** 불순도의 종류



6.5 결정 트리의 계산 복잡도

교재 p.235~ 참고

< 결정 트리의 계산 복잡도 >

깊이	리프 노드(Leaf node)의 개수
d →	→ 2^d
$log_2(m)$ ←	← m

➔ $O(log_2(m))$ 개의 노드 확인 = 계산 복잡도

n개의 모든 특성에 대해서는 $O(n * log_2(m))$

6.7 규제 매개변수

* **비파라미터 모델** : 파라미터를 사용하지 않거나, 훈련되기 전에 파라미터 수가 결정되지 않는 모델
  - 모델의 구조가 데이터에 맞춰져 고정되어 있지 않고, 자유로움
  - **결정 트리**는 모델 파라미터가 있지만, 훈련되기 전에는 파라미터의 수가 미정인 비파라미터 모델

* **파라미터 모델** : 미리 정의된 모델 파라미터를 사용하는 모델
  - 자유도가 제한되고 과대적합될 위험이 낮음
  - 대표적으로 **선형 회귀 모델** 등이 있음

결정트리는 훈련데이터에 대한 제한이 거의 없음 → 과대적합될 위험이 큼

➔ 따라서, **규제**를 통해 자유도를 제한하여 과대적합을 피해야 함

6.7 규제 매개변수

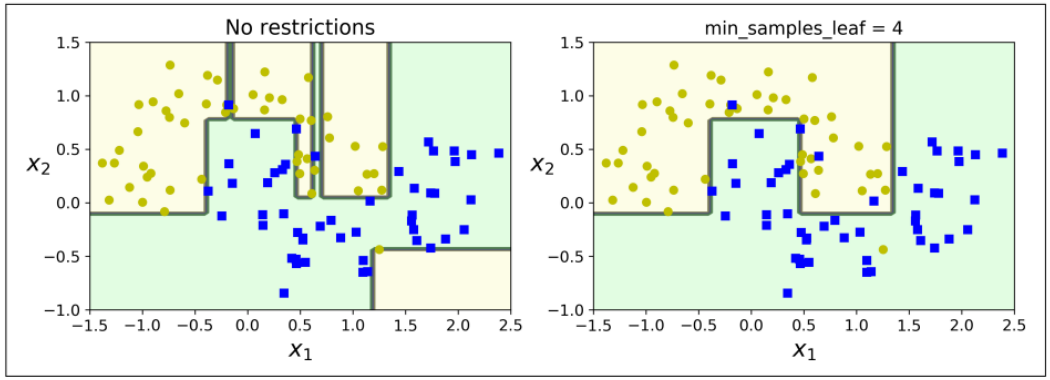


Figure 6-3. Regularization using min_samples_leaf

규제 방법; 규제 매개변수 추가!

- max_depth : 결정 트리의 깊이 제어
- min_samples_split : 분할되기 위해 노드가 가져야 하는 최소 샘플 수
- min_samples_leaf : 리프 노드가 가지고 있어야 할 최소 샘플 수
- min_weight_fraction_lead : 가중치가 부여된 전체 샘플 수에서의 비율
- max_leaf_nodes : 리프 노드의 최대 수
- max_features : 각 노드에서 분할에 사용할 특성의 최대 수

max_로 시작하는 매개변수를 감소시키거나 min_으로 시작하는 매개변수를 증가시키면,

➔ **결정 트리 모델에 규제 이루어짐!**

6.8 회귀

```
: from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(max_depth=2, random_state=42)
tree_reg.fit(X, y)
```

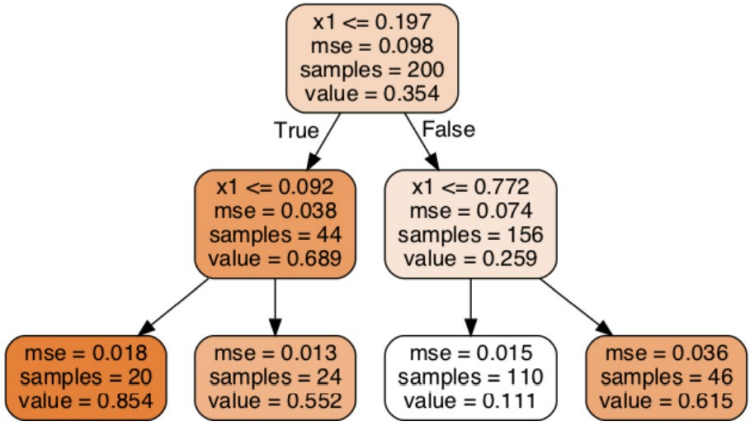
잡음이 섞인 2차 함수 형태의 데이터셋에서 max_depth=2 규제하여
DecisionTreeRegressor 로 회귀 트리 출력!

* 앞서 만든 *분류 트리와의 차이점*

: 각 노드에서 클래스를 예측하는 대신에 *어떤 값을* 예측한다는 점

** 예측값($\hat{y}$): 리프 노드에 있는 모든 훈련샘플의 타깃값의 **평균**

: 분류 트리는 훈련 세트의 불순도를 최소화 시키는 방향으로 분할, **회귀 트리**는 **평균 제곱 오차(MSE)를 최소화 시키는 방향**으로 분할

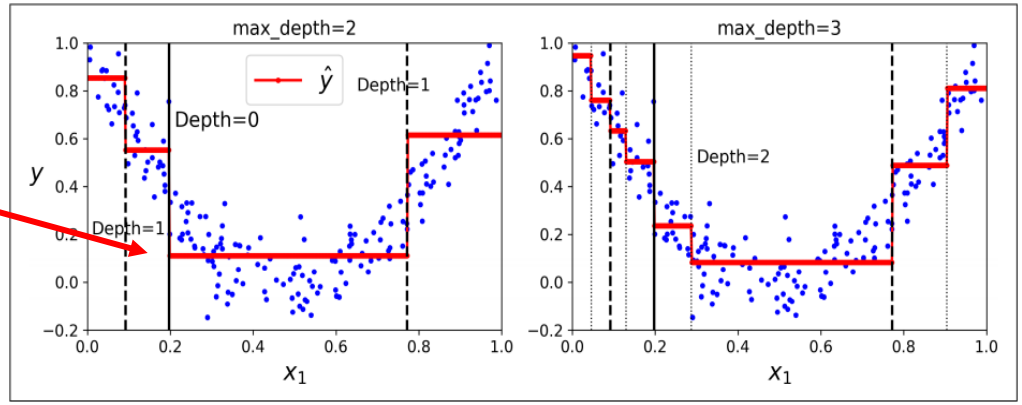


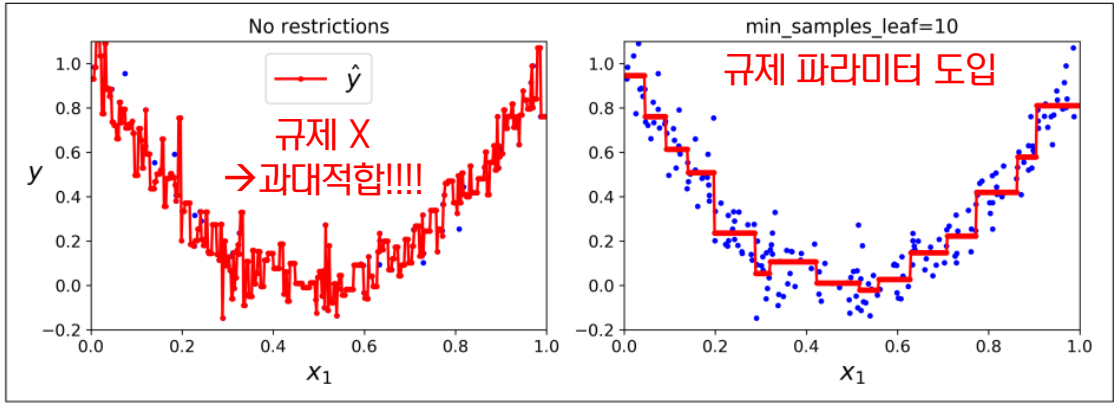
6.8 회귀

DecisionTreeRegressor의 알고리즘은

예측값($\hat{y}$)에 가능한 한 많은 샘플이 가까이 있도록 영역을 분할함

해당 node의 평균값

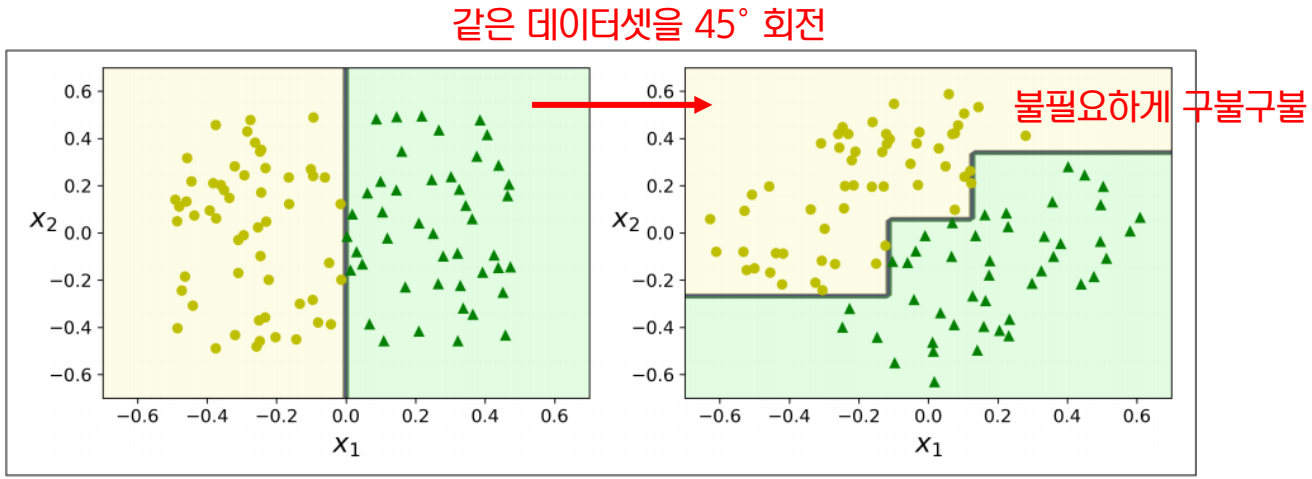




6.9 불안정성

결정 트리의 두가지 문제점

1. 결정 트리는 항상 계단 모양의 결정 경계 생성 (모든 결정 경계는 축에 수직)
   ➔ 훈련 세트의 회전에 매우 민감



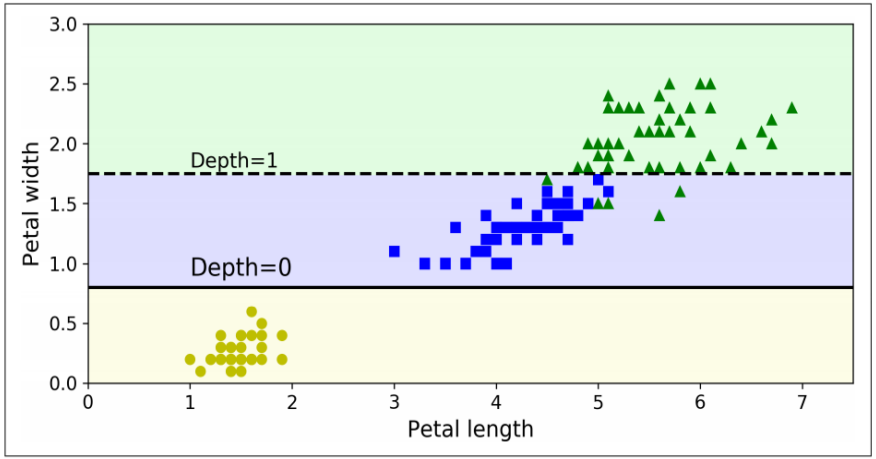
* 해결 방법 : 훈련 세트를 더 잘 학습시키는 **최적의 방향으로 회전시키는 PCA 기법**을 사용! (8장 참고)

6.9 불안정성

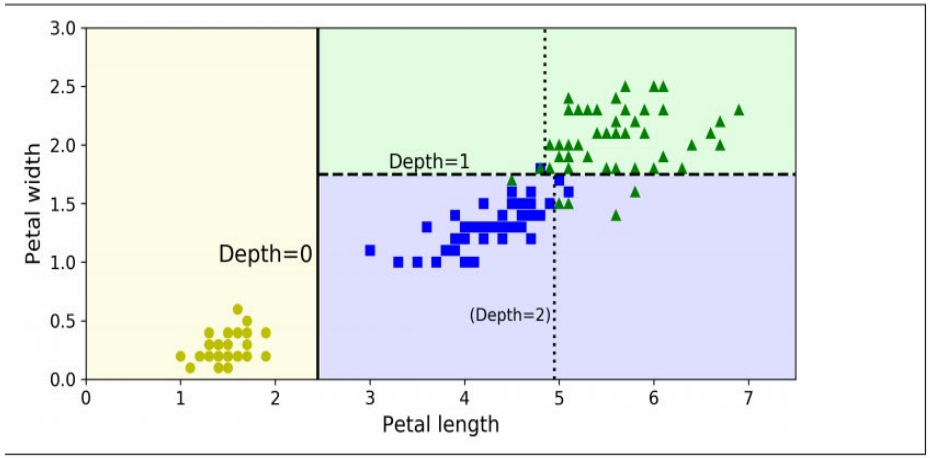
결정 트리의 두가지 문제점

2. 훈련 데이터의 작은 변화에도 매우 민감!

Iris-Versicolor를 제거하고 결정트리 훈련



앞서 만들었던 원래 결정트리



➔ 훈련 세트의 세부사항에 민감하게 반응하여 완전히 다른 결과를 출력함

이러한 불안정성을 “랜덤 포레스트 " 기법을 통해 극복 가능! (7장 참고)

감사합니다

Q&A