핸즈온 머신러닝 CH6 결정트리

결정트리란,

분류와 회귀 작업 그리고 다중출력 작업도 가능한 머신러닝 알고리즘
매우 복잡한 데이터셋도 학습 가능한 강력한 알고리즘

→ 랜덤 포레스트의 기본 구성 요소

▼ 6.1 결정 트리 학습과 시각화

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

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

tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X, y)
```

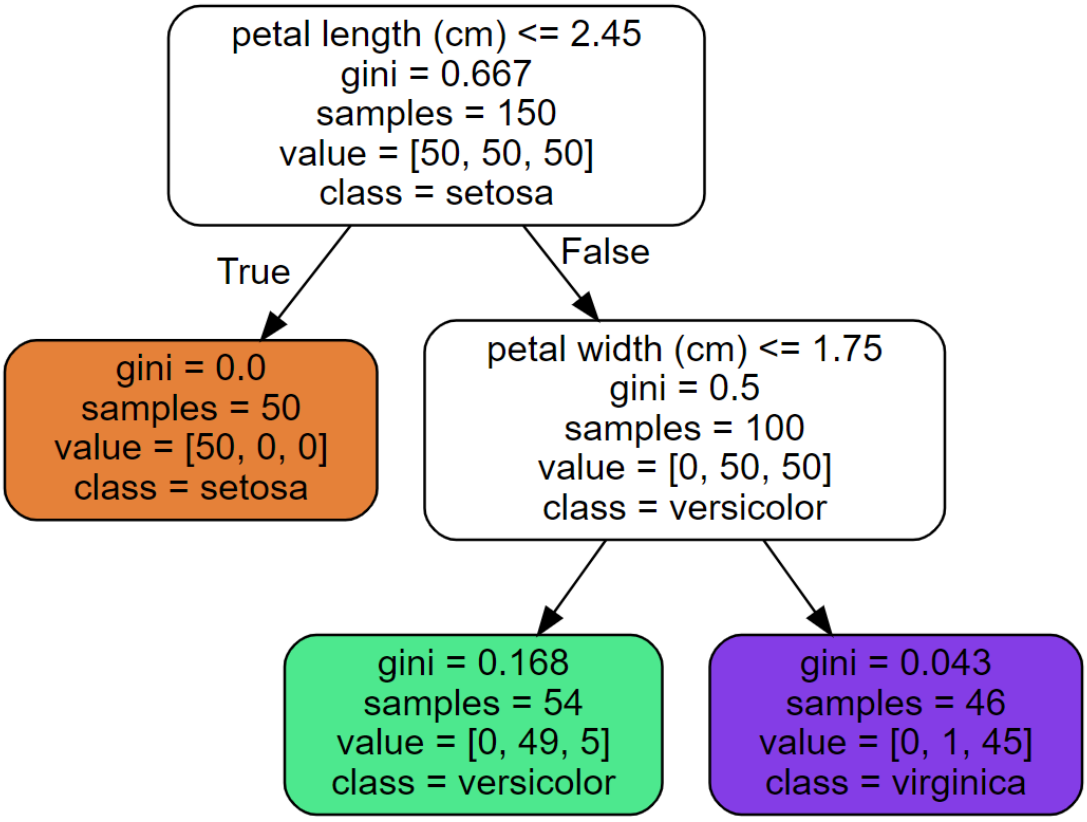


그림 6-1

▼ 6.2 예측하기

그림 6-1 해석:

Root node는 petal_length≤2.45 검사 → 만족하면 왼쪽, 아니면 오른쪽 자식노드
왼쪽 자식 노드의 경우 leaf node(자식 노드를 가지지 않는 노드)이므로 추가검사 X
→ setosa 로 분류
2.45보다 petal_length가 긴 경우 오른쪽 자식노드 → petal_width 검사

gini 불순도

한 노드의 모든 샘플이 같은 클래스에 속해있으면, 순수 gini=0

1 에서 '전체 데이터 개수 중 각 레이블이 차지하는 개수의 비율'을 제곱해서 빼

화이트박스와 블랙박스

화이트박스 : 직관적인 결정 방식 → 결정트리

블랙박스 : 성능이 뛰어나지만 예측 과정 설명 어려워랜덤 포레스트, 신경망

▼ 6.3 클래스 확률 추정

결정 트리는 한 샘플이 특정 클래스 k에 속할 확률 추정 가능

```
tree_clf.predict_proba([[5,1.5]]) #array([[0.        , 0.90740741, 0.09259259]])
tree_clf.predict([[5,1.5]]) #array([1])-> Versicolor로 추정
```

▼ 6.4 CART 훈련 알고리즘

사이킷런은 결정트리 훈련 위해 CART 알고리즘 사용
훈련세트를 하나의 특성 k의 임계값 tk를 사용해 두 개의 서브셋으로 나눠

이때 k 와 tk는 크기에 따른 가중치가 적용된 가장 순수한 서브셋으로 나눌 수 있는 짝
→ 이 과정 계속 반복 until max_depth or 불순도 줄이는 분할을 찾을 수 없을 때

▼ 6.5 계산 복잡도

예측에 필요한 전체 복잡도 = $O(log2(m))$개 (특성의 개수와 상관 X)

훈련 알고리즘은 각 노드에서 모든 훈련 샘플이 모든 특성 비교하므로
훈련 복잡도 = $O(n * mlog2(m))$개

▼ 6.6 지니 불순도 또는 엔트로피

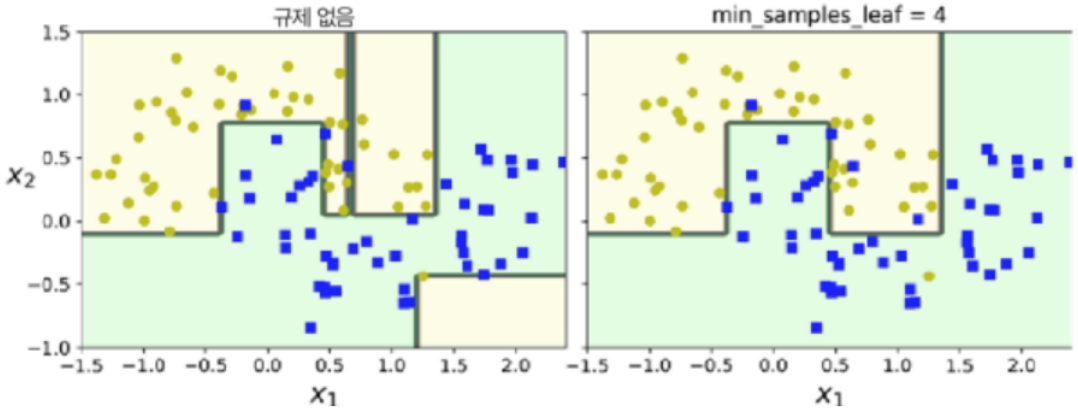
엔트로피 불순도 : `criterion`의 매개변수를 "entropy"로 지정

열역학 이론에서 분자가 안정된 상태일 때 엔트로피를 0으로 보는 것과 같이
정보이론에서도 **모든 메시지가 동일할 때 엔트로피를 0**으로 봄

→ 지니 불순도와 엔트로피 불순도의 결과 큰 차이 X
→ 둘 다 비슷한 트리를 만들어냄
→ 지니 불순도가 조금 더 계산이 빨라 기본값으로 더 타당
→ 그러나 엔트로피가 보다 균형 잡힌 트리를 만들어냄

▼ 6.7 규제 매개변수

결정 트리 모델은
훈련되기 전에 파라미터 수가 결정되지 X : **비파라미터 모델**
→ 모델 구조가 데이터에 맞춰져서 고정되지 않고 자유로워
→ 과대적합의 위험 존재
→ 따라서 결정 트리의 자유도를 제한할 필요가 있어→ **"규제"** by **`max_depth`**
→ `max_depth`를 줄이면 과대적합의 위험 감소



왼쪽 결정트리는 과대적합
오른쪽 결정트리가 min_samples_leaf 로 규제를 둔 것→ 일반화 성능이 더 좋을 것

cf) 파라미터 모델 : ex. 선형모델
미리 정의된 모델 파라미터 수를 가져 자유도가 제한 되고 과대적합 위험 감소

▼ 6.8 회귀

```
from sklearn import DecisionTreeRegressor

tree_reg=DecisionTreeRegressor(max_dpth=2)
tree_reg.fit(X,y)
```

앞선 모델에서는 클래스를 예측한 것과 달리
회귀 결정 트리에서는 **어떤 값**을 예측

회귀 결정트리에서는 CART 알고리즘이 불순도 최소화 방법이 아닌,
MSE 최소화를 하며 분할하는 방식으로 작동

마찬가지로 회귀결정트리도 규제가 없다면 과대적합 위험
→ 규제 이용해야 ex. min_samples_leaf=10

▼ 6.9 불안정성

결정 트리의 장점 : 이해 및 해석이 쉽고, 사용이 편리, 성능 좋아
단점: 계단 모양의 결정 경계 생성 (모든 분할이 축에 수직)
→ 훈련 세트의 회전에 민감
→ 해결 방법 : 훈련 데이터를 더 좋은 방향으로 회전시키는 PCA 기법 사용