

머신러닝 강의자료

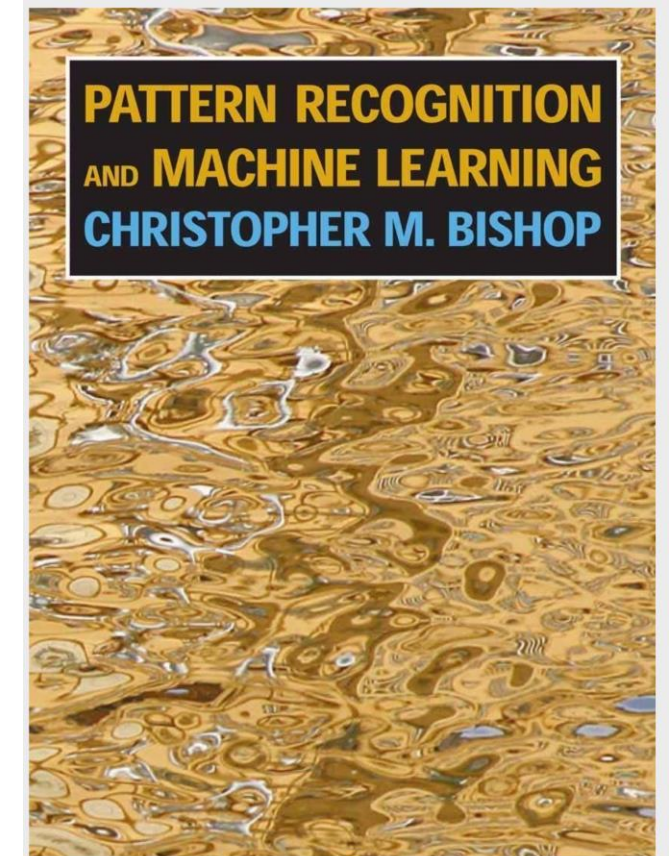
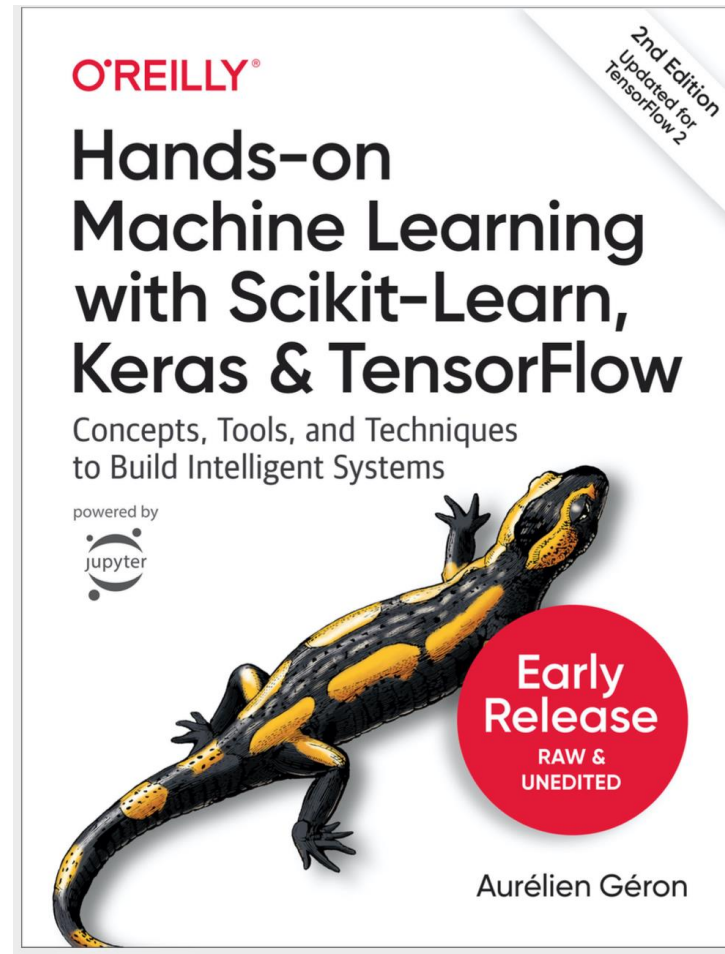
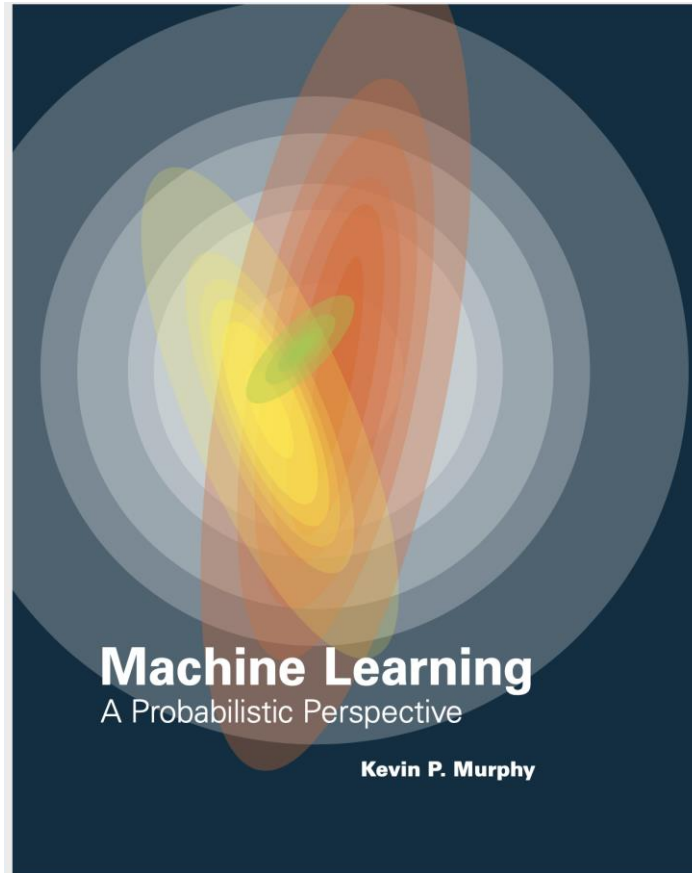
6강 - 결정 트리

닥터윌컨설팅 딥러닝 R&D 책임연구원
고려대학교 인공지능대학원 박사과정

류회성(Hoe Sung Ryu)



들어가기 앞서



들어가기 앞서

● GitHub

- <https://github.com/hoesungryu/blockchain-devML-course>

hoesungryu / blockchain-devML-course

Watch 1 Unstar 1 Fork 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags Go to file Add file Code

hoesungryu and hoesungryu 2일차 업데이트 ec864ee 2 days ago 9 commits

File/Folder	Commit Message	Time Ago
code	2일차 업데이트	2 days ago
data	1일차 자료 업로드	3 days ago
img	2일차 업데이트	2 days ago
lecture_note	2일차 업데이트	2 days ago
.DS_Store	2일차 업데이트	2 days ago
.gitignore	Initial commit	4 days ago
LICENSE	Initial commit	4 days ago
README.md	1일차 자료 업로드	3 days ago
requirements.txt	2일차 업데이트	2 days ago

README.md

블록체인 처음여게 공부기전

About
No description, website, or topics provided.

Readme
MIT License

Releases
No releases published
[Create a new release](#)

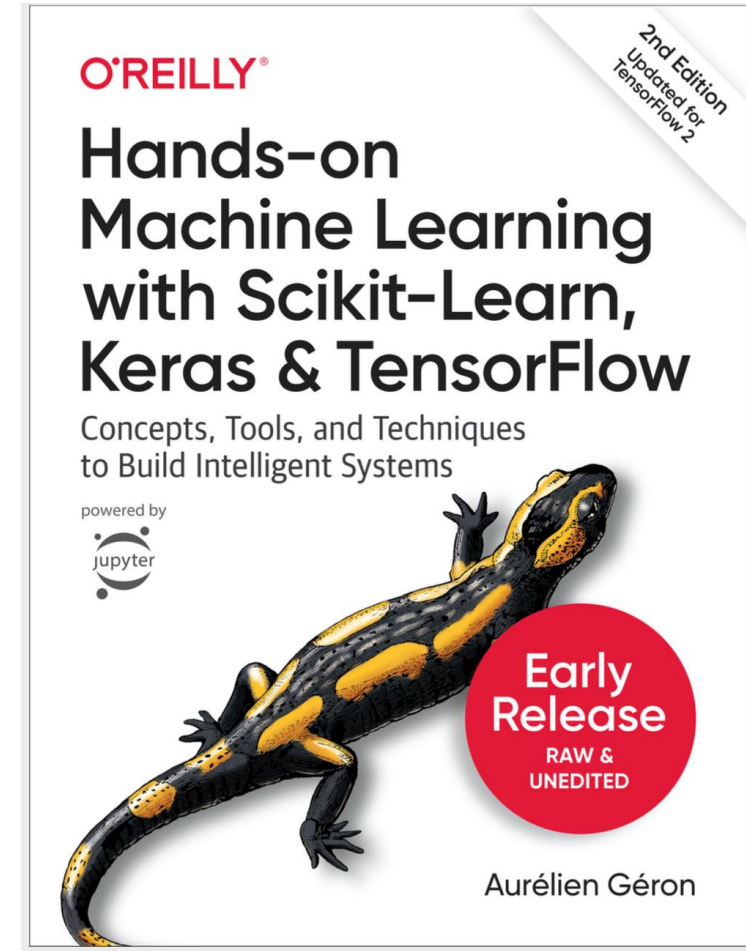
Packages
No packages published
[Publish your first package](#)

Contributors 2
mssung94 mssung94

강의내용

● 분류

- 선형 회귀
- 경사 하강법
- 다항 회귀
- 학습 곡선
- 규제가 있는 선형 모델
- 로지스틱 회귀



[데이터 전처리 개요]

데이터 전처리 개요

● 데이터 전처리가 필요한 이유

- 실무데이터에서는 분석 기법을 바로 적용하기 힘든 형태이다.
- Ex) 비어 있음(결측치), 잡음(noise), 적합하지 않은 데이터 구조

● 데이터 품질 저하 원인

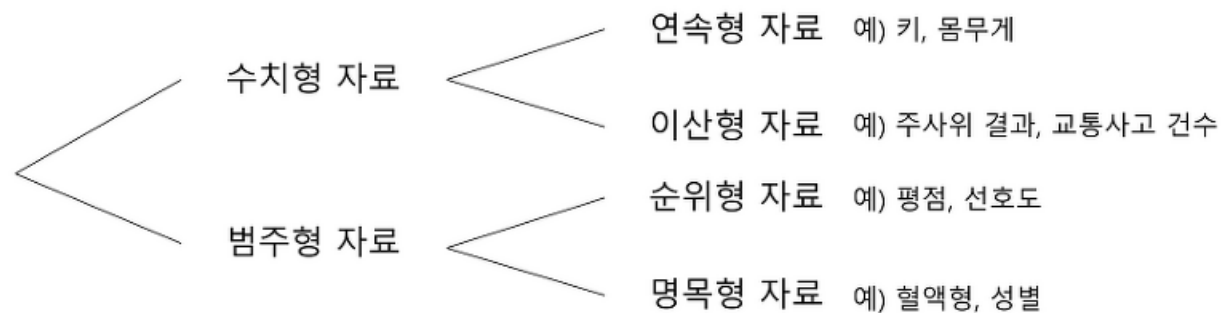
- 불완전(incomplete): 데이터가 비어있는 있는 경우 DB 테이블의 속성값이 NULL 인 경우
- 잡음(noisy): 데이터 오류(error)가 포함된 경우
- 모순(inconsistent): 데이터 간의 정합성이 없는 경우

● 고품질 데이터라 하더라도 전처리

- 실무에서 존재하는 데이터의 구조 형태가 분석목적에 적합한 경우가 드물다.

데이터 전처리 개요

● 데이터 타입



데이터 전처리 개요

● 데이터 인코딩 방법

- LabelEncoder
- pd.get_dummies
- map

1.3 LabelEncoder

```
1 from sklearn.preprocessing import LabelEncoder
```

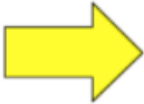
executed in 584ms, finished 20:39:05 2020-08-10

1.5 Map

```
1 df['map_name'] = df.pName.map({'TV':0, '공기청정기':1, '냉장고':2, '!'})
```

executed in 5ms, finished 21:33:23 2020-08-10

Color			
Red			
Red			
Yellow			
Green			
Yellow			



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

데이터 전처리 개요

● 데이터 스케일링

- 기본적으로 scikit-learn 에서는 다음과 같은 스케일링 클래스를 제공한다.

- StandardScaler: 평균이 0과 표준편차가 1이 되도록 변환.

```
from sklearn.preprocessing import StandardScaler
```

- MinMaxScaler: 최대값이 각각 1, 최소값이 -1 이 되도록 변환


```
from sklearn.preprocessing import MinMaxScaler
```

- RobustScaler(X): 중앙값(median)이 0, IQR(interquartile range)이 1이 되도록 변환.

```
from sklearn.preprocessing import RobustScaler
```

- MaxAbsScaler(X): 0을 기준으로 절대값이 가장 큰 수가 1또는 -1이 되도록 변환

```
from sklearn.preprocessing import MaxAbsScaler
```



[실습]

[결정 트리 학습과 시각화]

결정 트리



어떻게 데이터를
구분할 수 있을까?

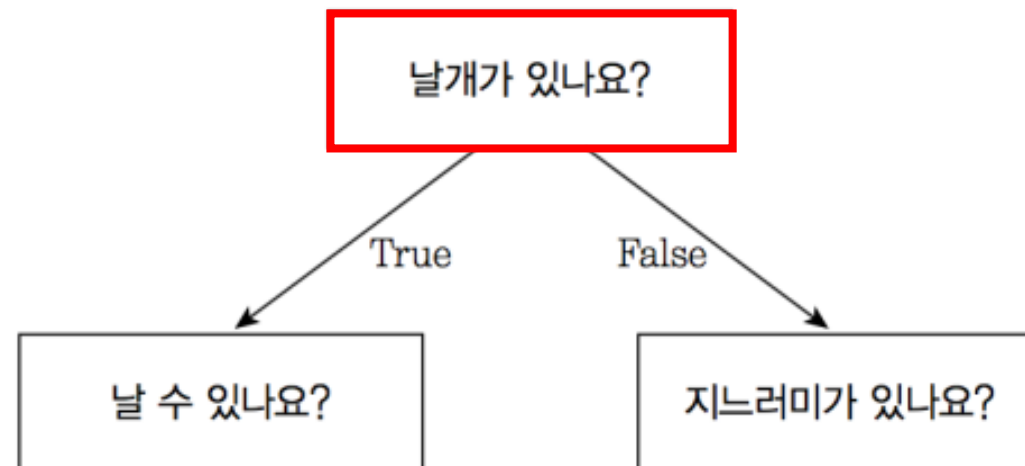
결정 트리



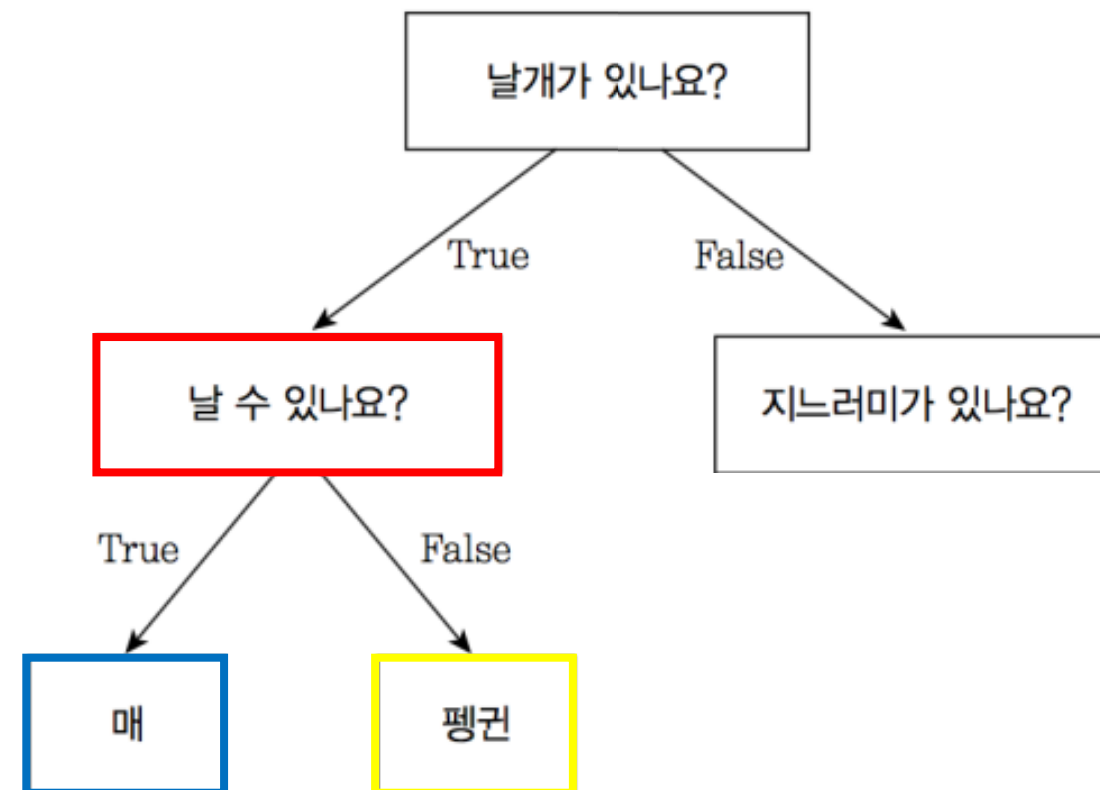
날개가 있나요?



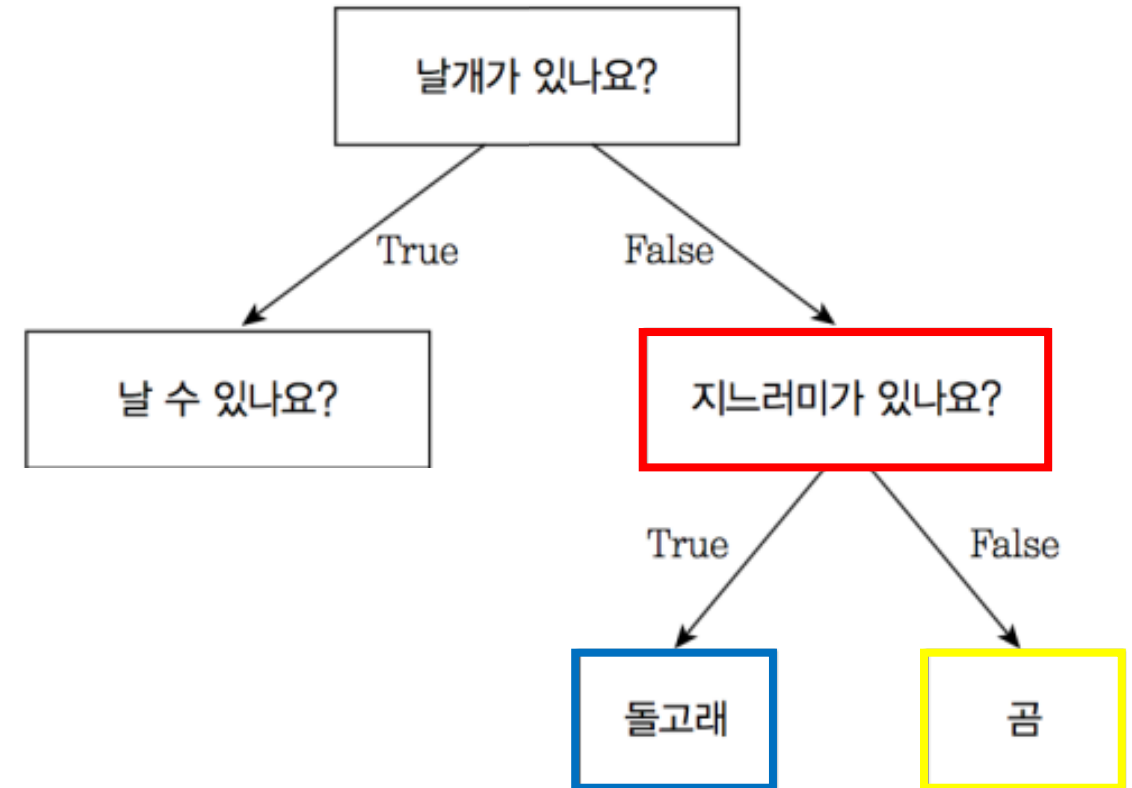
결정 트리



결정 트리



결정 트리



결정 트리

- 의사결정 트리의 분할 속성 선택

- 어떤 입력 변수를 이용하여 어떻게 분리하는 것이 종속 변수의 분포를 가장 잘 구별해 주는지를 파악하여 자식 마디가 형성되는데, 목표 변수의 분포를 구별하는 정도를 순수도(Purity)나 불순도(Impurity)에 의해서 측정

- 분할 속성의 선택

- 부모 마디의 순수도에 비해서 자식 마디들의 순수도가 증가하도록 자식 마디를 형성

- 불순도의 측정

- 지니 지수(Gini index)
- 엔트로피 지수(Entropy index)

결정 트리의 장/단점

● 장점

- 이해하고 해석하기 쉽다
- 데이터 전처리가 상대적으로 적게 필요
 - 정규화나 피처엔지니어링이 상대적으로 덜 필요
 - 그래도 Null 값 처리는 해야함
- 모델 평가가 간단함
 - 시간 복잡도: $O(\log(N))$

● 단점

- 과적합 발생 가능성이 높음
- 데이터에 민감함
- 데이터 분포가 편중 시, 성능 저하됨

[지니 지수 또는 엔트로피 지수]

지니 지수 또는 엔트로피 지수

● 지니 지수(Gini index)

- 정의

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

- 의미

- 지니 불순도가 높을수록 데이터가 분산되어 있음을 의미
- 이 값이 낮아질수록 깔끔하게 분류되어 있음을 의미

- 계산

- 초록색 부분의 지니 불순도 계산

$$1 - \left(\frac{0}{54}\right)^2 - \left(\frac{49}{54}\right)^2 - \left(\frac{5}{54}\right)^2 \approx 0.168$$

- 두 개의 범주가 50:50으로 구성될 때 최대의 불순도 값인 0.5

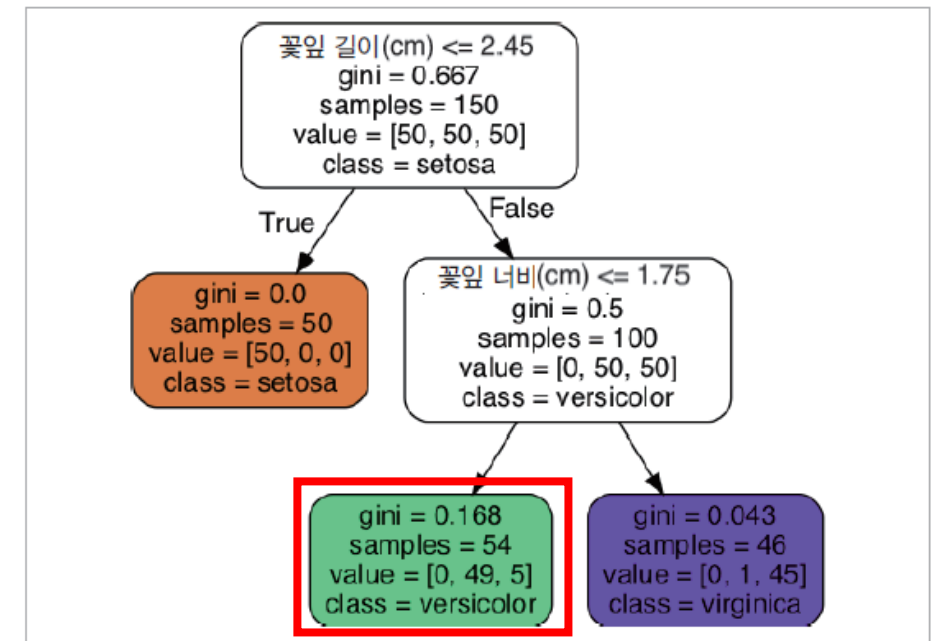


그림 6-1 붓꽃 결정 트리

지니 지수 또는 엔트로피 지수

● 엔트로피 지수(Entropy index)

- 정의

$$Entropy_i = - \sum_{k=1}^n p_{i,k} \log_2(p_{i,k})$$

- 의미

- 무질서함을 측정하는 지표

- 어떤 세트가 한 클래스의 샘플만 담고 있다면 엔트로피는 0

- 계산

- 초록색 부분의 지니 불순도 계산

$$1 - \left(\frac{0}{54}\right) \log_2 \left(\frac{0}{54}\right) - \left(\frac{49}{54}\right) \log_2 \left(\frac{49}{54}\right) - \left(\frac{5}{54}\right) \log_2 \left(\frac{5}{54}\right) \approx 0.445$$

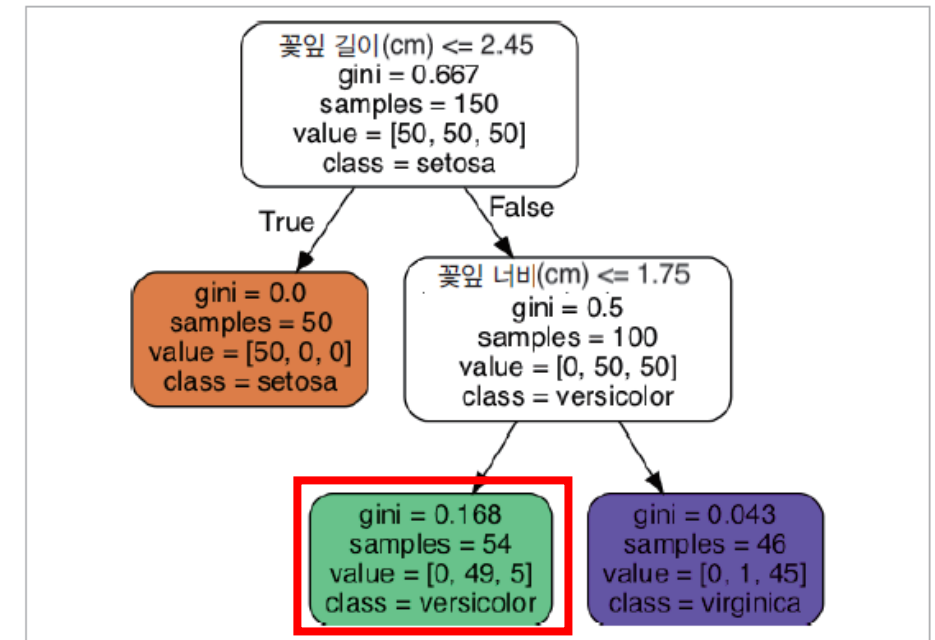


그림 6-1 붓꽃 결정 트리

지니 지수 또는 엔트로피 지수

- 지니 지수와 엔트로피 지수 중 무엇을 써야 할까?
 - 실제로는 큰 차이가 없음
 - 결과적으로 비슷한 트리를 생성
 - 지니 불순도가 조금 더 계산이 빠르기 때문에 기본값으로 좋음
 - 지니 지수가 가장 빈도 높은 클래스를 한쪽 가지로 고립시킴
 - 엔트로피는 조금 더 균형 잡힌 트리를 생성

[결정트리로 예측하기]

결정 트리로 예측하기

● 예제

- 결정 트리로 붓꽃 데이터 학습 후 예측 시각화

```
from sklearn.tree import export_graphviz

export_graphviz(
    tree_clf,
    out_file=image_path("iris_tree.dot"),
    feature_names=iris.feature_names[2:],
    class_names=iris.target_names,
    rounded=True,
    filled=True
)
```

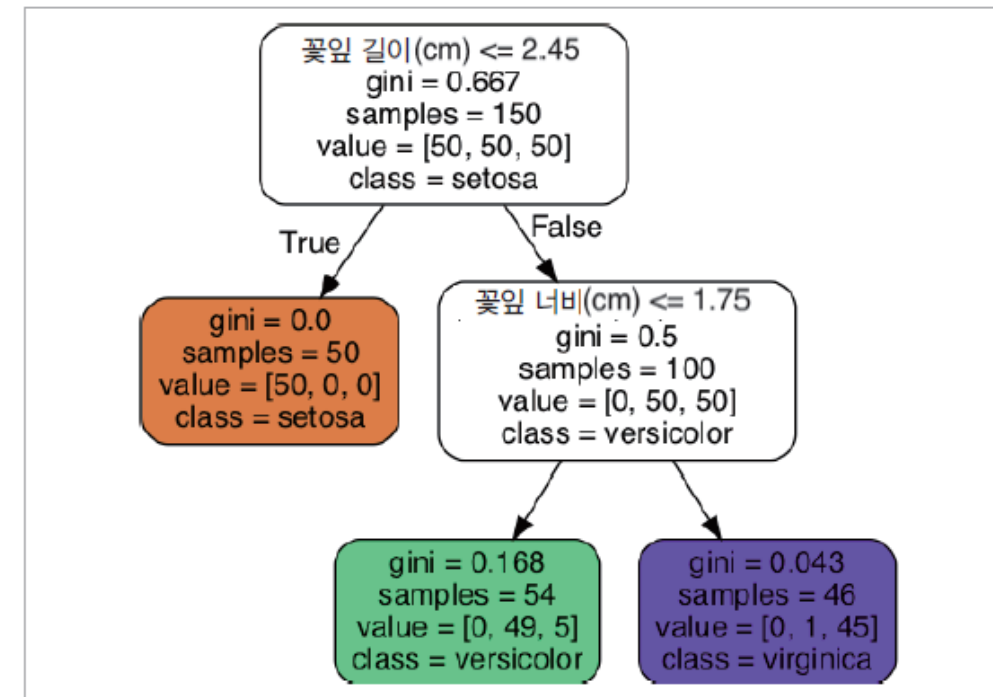


그림 6-1 붓꽃 결정 트리

결정 트리로 예측하기

● 트리가 예측하는 과정

- 먼저 루트 노드에서 시작
 - 꽃잎의 길이가 2.45cm보다 짧은 지 검사
- 첫번째 조건에서 참이라면 왼쪽으로
 - 이 때 gini 계수가 0이므로 추가적인 검사하지 않음
- 첫번째 조건에서 거짓이라면 오른쪽으로
 - 꽃잎의 너비가 1.75cm보다 작은 지 검사
 - 두번째 조건에서 참이라면 왼쪽으로
 - 두번째 조건에서 거짓이라면 오른쪽으로

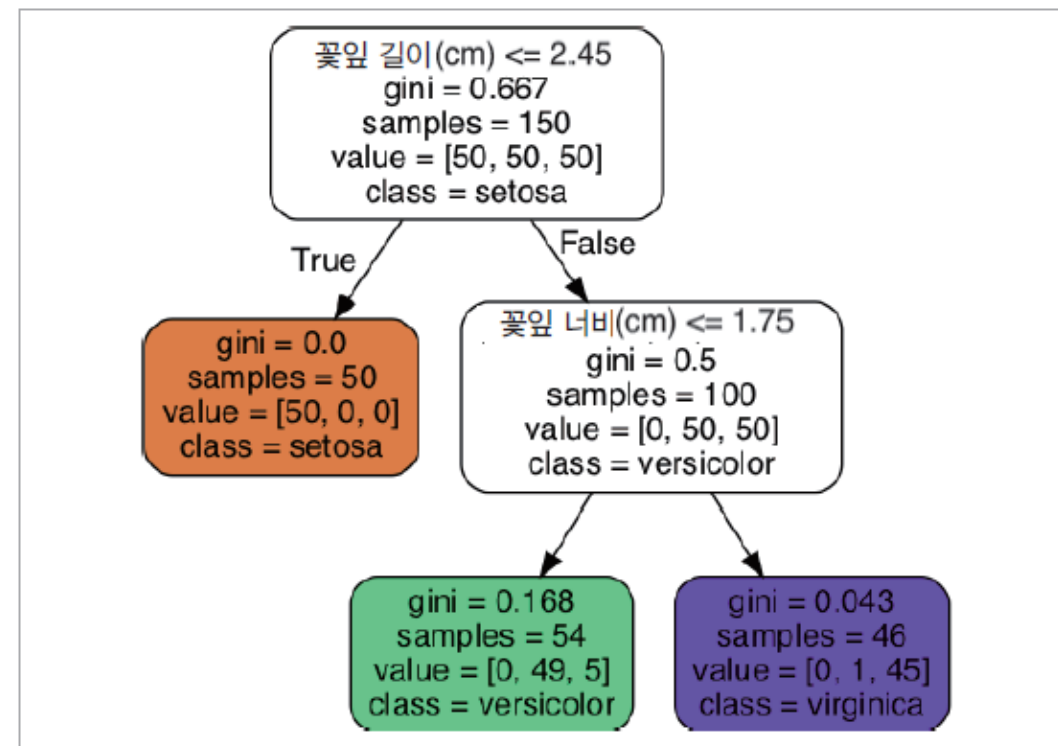


그림 6-1 붓꽃 결정 트리

[클래스 확률 추정하기]

클래스 확률 추정하기

- predict_proba 함수를 이용해서 확률도 추정 가능

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

Quiz1

- Q1. 고품질데이터는 전처리가 필요 없는가?
 - A1. 고품질이라 할지라도 필요하다.
- Q2. 머신러닝은 범주형 데이터를 읽을 수가 없는가?
 - A2. 인코딩(범주형 → 수치형) 필수
- Q3. 불순도 종류
 - A3. 지니, 엔트로피



(1/3)

[CART 훈련 알고리즘]

CART 훈련 알고리즘

● Classification And Regression Trees (CART)

Classification vs Regression Tree 비교

- Classification : 카테고리컬 변수에 사용
- Regression : 연속형 변수에 사용

- 가장 많이 있는 속성을 기준으로 분류
- 성적 중 C가 가장 많다면, C인지 아닌지?

CLASSIFICATION - USE MODE / CLASS

Mode = happens most often



REGRESSION - USE MEAN / AVERAGE

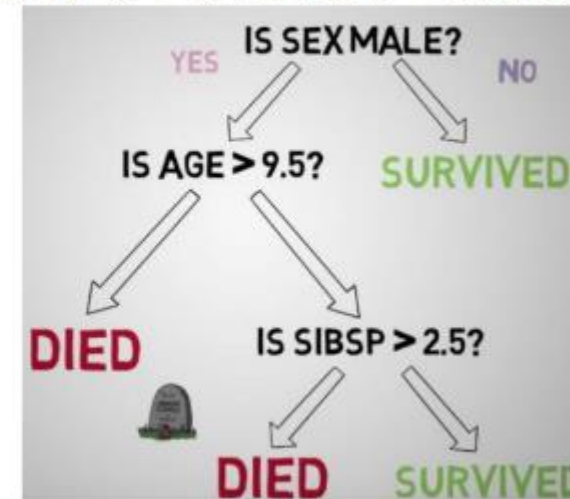
Mean Average



- 수치형 값의 평균으로 분류
- 선형으로 수치값을 분류할 때.

함수(알고리즘)을 어떻게 트리로 표현할까?

- 많이 사용하는 타이타닉 생존자를 분류하는 예시를 보자
- 수치형 값을 기준으로 3개의 알고리즘(함수)으로 데이터를 분류



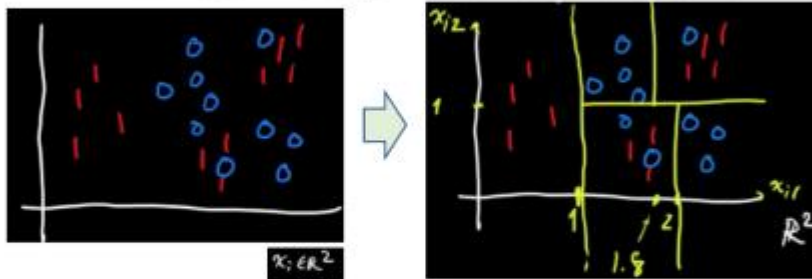
CART 훈련 알고리즘

● Classification And Regression Trees (CART)

- 각 Leaf Node의 불순도를 최소화하는 이진 트리(Binary Tree)

Classification Tree 예시

- 아래와 같이 임의의 데이터를 생성
- Decision Tree는 데이터를 2개로 분할(Binary split)한다.
 - 이때 분류된 노드의 에러(불순도)를 최소화하는 분류기준을 선택



- 데이터를 binary로 분류하면, 아래와 같은 트리가 생성됨

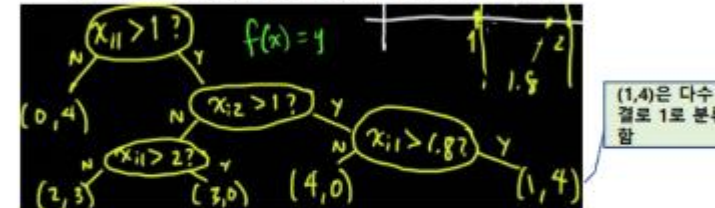
불순도 = 0
모두 1로 구성됨
더 이상 분류할 필요없음



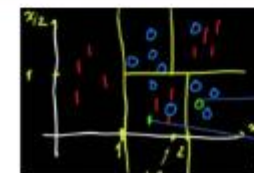
불순도 = 0: 10개, 1: 7개
불순도를 최소화하기 위해
데이터 분류

분류 알고리즘을 트리로 표현

- 그런데 특정 leaf node는 불순도가 높은 것이 보인다
- 이런 leaf에서는 어떻게 데이터를 분류할까?
 - → 다수결을 이용하여 분류
 - → 예를 들어 (2,3)은 0:2개, 1:3개이므로 1로 분류함



그럼 새로운 데이터가 입력되면 어떻게 분류할까?



(3, 0)은 0이 3개이므로 "0"으로 분류

(2, 3)은 0이 2개, 1이 3개 이므로 "1"로 분류

CART 훈련 알고리즘

● ID3 vs CART

	ID3	CART
분류기준	Shannon entropy Information gain	Gini Index
분류 방식	<ul style="list-style-type: none"> Field별 값을 기준으로 분할 Grade란는 field에 (A, B, C)라는 값이 있으면, 각 속성별로 데이터를 분류함 (총 3번) 	<ul style="list-style-type: none"> 이진 트리를 사용하여, 1개의 field의 값을 특정 기준값을 기준으로 크면 right, 작으면 left로 분류
지원 모델	Classification	Classification + Rregression
과적합 방지	X	O (Leaf 데이터 갯수, 최소 변화량)
트리	트리	이진트리
회기모델 지원 (수치 데이터)	X	O
장점	구현 용이	부등호 질의가능 사후 가지치기 가능 (Leaf Node 통합) 데이터 해석이 용이(설명력)
단점	수치형 속성 사용불가 카테고리 속성이 많은 경우, Tree가 깊어짐 (가지가 많아지기 때문)	학습데이터가 충분해야 함 (과적합 유발) → 배깅/부스팅 활용

[결정 트리 계산 복잡도]

결정 트리 계산 복잡도

- 예측을 하기 위해선 결정 트리를 루트 노드에서부터 리프 노드까지 탐색해야 함
 - 일반적으로 트리는 거의 균형을 이루고 있음
 - 결정 트리를 탐색하기 위해서 $O(\log_2 m)$ 개의 노드를 거쳐야 함
 - 각 노드는 하나의 특성값만 확인하기 때문에 예측에 필요한 전체 복잡도는 특성 수와 무관
 - 큰 훈련 세트를 다룰 때도 예측 속도가 매우 빠름

Quiz2

- Q1. 지니 계수의 정의는 무엇인가요?
 - A1.
- Q2. 엔트로피의 의미는 무엇인가요?
 - A2.
- Q3. 지니 계수와 엔트로피의 차이는 무엇인가요?
 - A3.



(2/3)

[규제 매개변수]

규제 매개변수

● 훈련 데이터에 대한 제약사항이 거의 없음

- 결정 트리는 모델 파라미터가 전혀 없는 것이 아니라
- 훈련 되기 전에 파라미터 수가 결정되지 않는 모델의 경우
 - 이러한 모델을 비파라미터 모델이라고 부름
- 선형 모델과 같이 모델 구조가 데이터에 맞춰져서 고정되지 않고 자유로울 경우
 - 파라미터 모델이라고 부름
 - 이러한 경우 자유도가 제한되고 과대적합될 위험이 줄어듬

● 결정 트리의 과대 적합을 피하기 위해

- 사이킷런에서 규제해야할 파라미터
 - max_depth
 - min_sample_split 등등

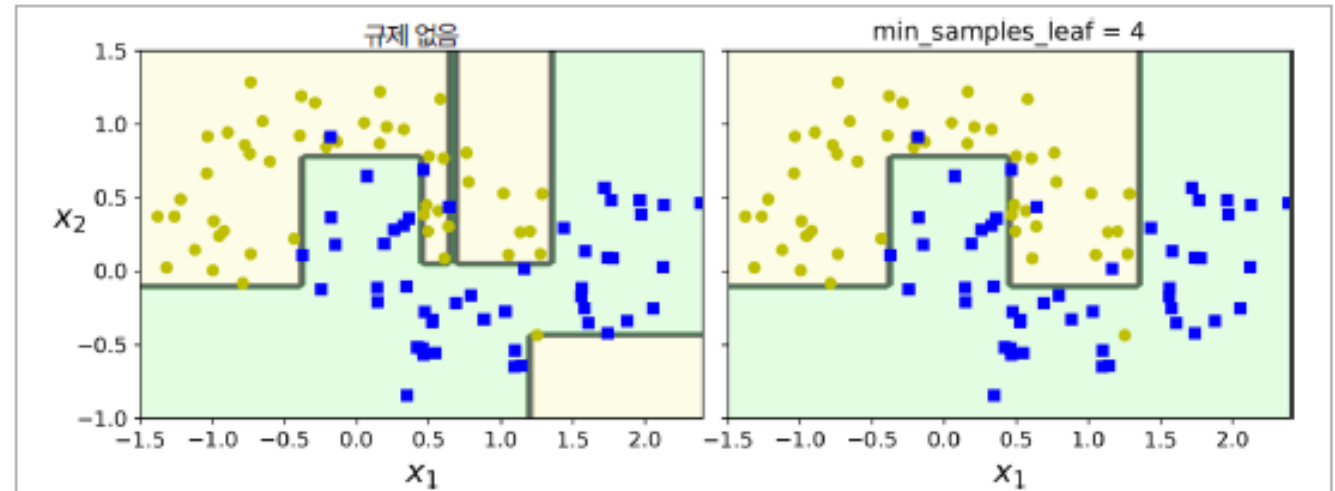


그림 6-3 min_samples_leaf 매개변수를 사용한 규제

규제 매개변수

● 사이킷런에서 결정트리

Parameters: **criterion : string, optional (default="gini")**

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.

트리를 구성할 때 사용하는 불순도
[gini, entropy]

splitter : string, optional (default="best")

The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

트리를 split할 때의 전략
[best, random]

max_depth : int or None, optional (default=None)

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

최종 생성되는 트리의 Depth의 최대값

min_samples_split : int, float, optional (default=2)

The minimum number of samples required to split an internal node:

- If int, then consider min_samples_split as the minimum number.
- If float, then min_samples_split is a fraction and $\text{ceil}(\text{min_samples_split} * n_{\text{samples}})$ are the minimum number of samples for each split.

노드를 split할 때 필요한 최소 샘플 수

Changed in version 0.18: Added float values for fractions.

min_samples_leaf : int, float, optional (default=1)

The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

- If int, then consider min_samples_leaf as the minimum number.
- If float, then min_samples_leaf is a fraction and $\text{ceil}(\text{min_samples_leaf} * n_{\text{samples}})$ are the minimum number of samples for each node.

Leaf node에 있는 최소 샘플 수

Changed in version 0.18: Added float values for fractions.

규제 매개변수

● 사이킷런에서 결정트리

Parameters: `max_features` : *int, float, string or None, optional (default=None)*

The number of features to consider when looking for the best split.

- If int, then consider `max_features` features at each split.
- If float, then `max_features` is a fraction and `int(max_features * n_features)` features are considered at each split.
- If "auto", then `max_features=sqrt(n_features)`.
- If "sqrt", then `max_features=sqrt(n_features)`.
- If "log2", then `max_features=log2(n_features)`.
- If None, then `max_features=n_features`.

split할 때 고려하는 feature의 개수

Note: the search for a split does not stop until at least one valid partition of the node samples is found, even if it requires to effectively inspect more than `max_features` features.

class_weight : *dict, list of dicts, "balanced" or None, default=None*

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of `y`.

Note that for multilabel (including multilabel) weights should be defined for each class of every column in its own dict. For example, for four-class multilabel classification weights should be `[[{0: 1, 1: 1}, {0: 1, 1: 5}, {0: 1, 1: 1}, {0: 1, 1: 1}]]` instead of `[[{1: 1}, {2: 5}, {3: 1}, {4: 1}]]`.

The "balanced" mode uses the values of `y` to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

For multi-output, the weights of each column of `y` will be multiplied.

Note that these weights will be multiplied with `sample_weight` (passed through the fit method) if `sample_weight` is specified.

Target label의 가중치 부여

- default = None: 모두 같은 가중치 (1)

- balanced: target label 분포에 따라 weight 설정

결정 트리의 사용시 TIP

- 차원이 너무 많거나, 샘플의 크기가 너무 작으면 오버피팅되는 경향이 있음
 - 차원이 많은 경우, PCA를 고려해보자
- Max_depth를 3으로 하고, 우선 트리를 생성해 보고, 점점 깊이를 늘려가면서 테스트해보자
- 트리 모델을 도식화해서 보라
- 오버피팅 발생 시, 아래 파라미터들을 조절해보자
 - Max_depth
 - Min_sample_split
 - Split되는 노드의 최소 샘플 개수이며 이 개수에 도달하는 노드가 만들어지면 더 이상 트리를 확장하지 않음
 - Min_sample_leaf
 - Leaf 노드의 최소 샘플 개수이며 이 개수에 도달하는 리프 노드가 만들어지면 더 이상 트리를 확장하지 않음
- 트리가 편향되지 않도록 데이터 분포를 구성하자

[결정트리 회귀]

결정트리 회귀

● 기본 아이디어

- 분류와 매우 비슷
- 각 노드에서 클래스를 예측하는 대신 어떤 값을 예측한다는 점
 - 리프 노드에 있는 훈련 샘플의 평균 타깃값의 MSE값의 평균이 예측값이 됨

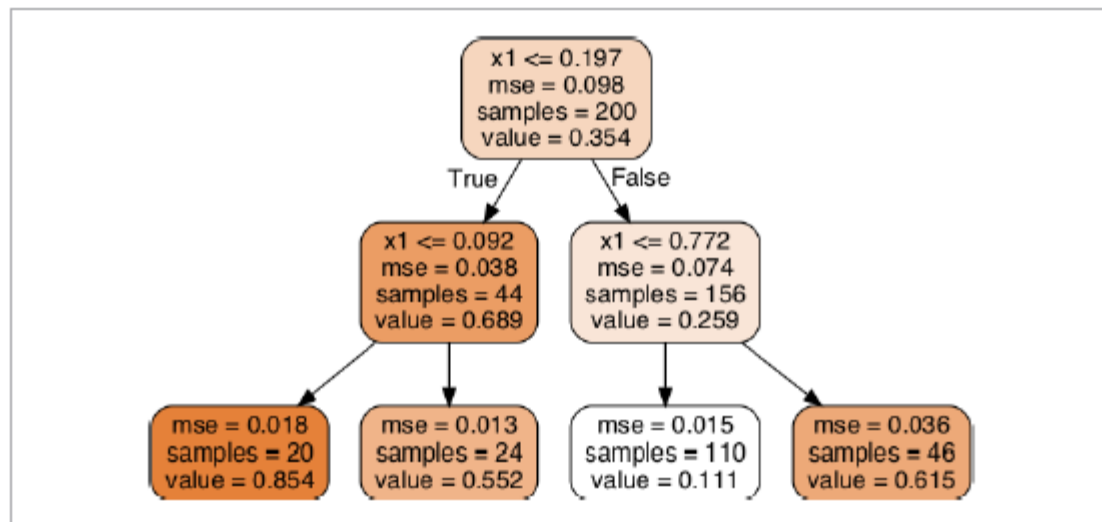


그림 6-4 회귀 결정 트리

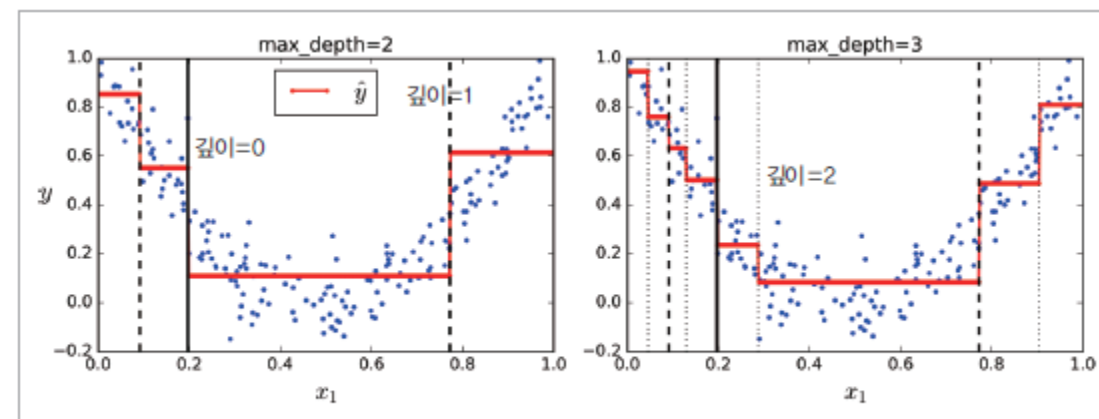


그림 6-5 두 개의 결정트리 회귀 모델의 예측

[결정트리 불안정성]

결정트리 불안정성

- 계단 모양의 결정 경계
 - 훈련 세트의 회전에 민감
- 훈련 데이터의 작은 변화에도 민감함
 - 이전에 만든 결정 트리와 달라질 수도 있음

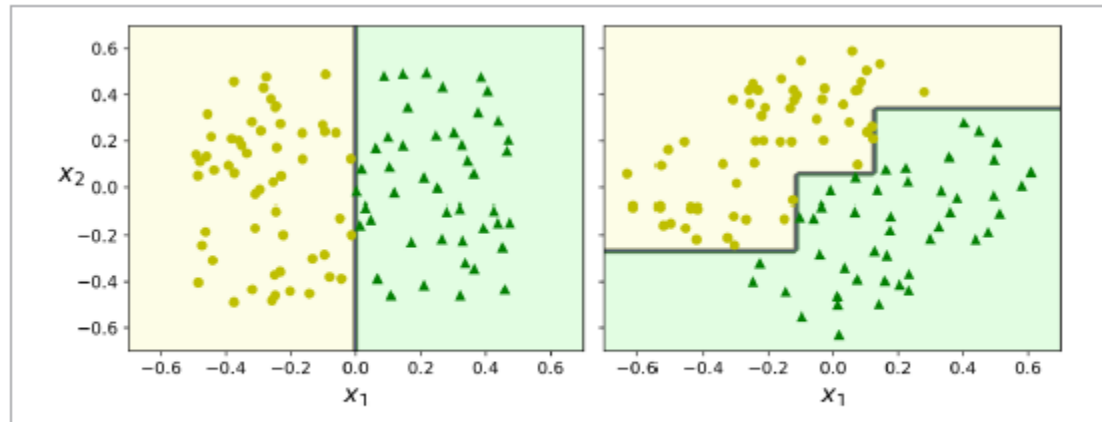


그림 6-7 훈련 세트의 회전에 민감한 결정 트리

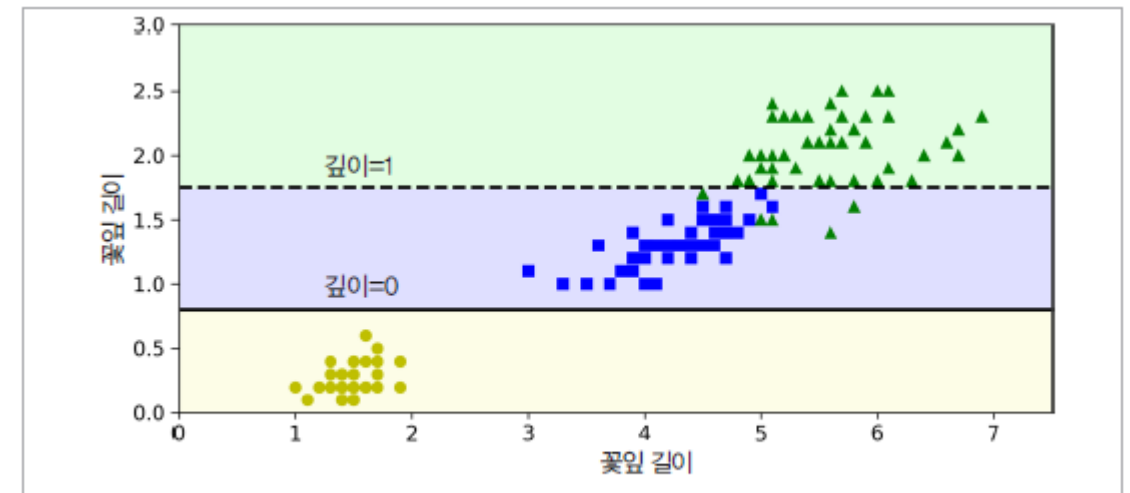


그림 6-8 훈련 세트의 세부사항에 민감한 결정 트리



(3/3)

[실습]

- 다음 시간에 -