천재교육

머신러닝 (Machine Learning) 이해 및 실습

K - 디지털 아카데미

머신러닝 교육과정 강의 일정

1일차 > 머신러닝

- > 머신러닝의 개요
- > [실습] 데이터 분석의 시작 Numpy, Pandas 활용하기

2일차 > Classification (분류)

- > 분류를 평가하는 지표 알아보기
- > 분류 알고리즘 (결정트리, 앙상블, 랜덤포레스트 등) 익히기
- > [실습] 분류를 통한 밀크T 만료및탈퇴회원 예측(이탈 회원 예측)

3일차 > Regression (회귀)

- > 회귀와 경사 하강법
- > 로지스틱 회귀와 소프트맥스 회귀
- > [실습] 로지스틱 회귀를 통한 문항별 정오답 예측

4일차 > 차원 축소와 Clustering(군집화)

- > PCA, LDA
- > K-means, DBSCAN 등 다양한 클러스터링 기법 알아보기
- > [실습] 밀크T중학 회원수준 군집화(GMM)

5일차 > 추천시스템과 최종 프로젝트

- > 추천시스템
- > 최종 프로젝트

분류(Classification)

01 분류란?

02 분류를 평가하는 지표

03 이진분류와 다중 분류

머신
러닝

Classification

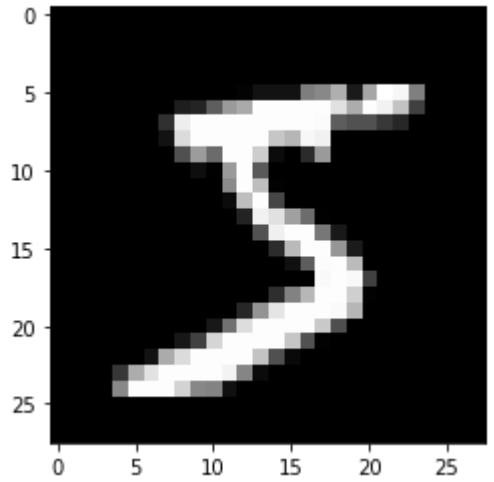
분류는

학습데이터로 주어진 데이터의 피쳐와 레이블값을 머신러닝 알고리즘으로 학습해 모델을 생성하고,

이렇게 생성된 모델에 새로운 데이터 값이 주어졌을 때 미지의 레이블 값을 예측하는 것

이진 분류
and
다중분류(다중 클래스 분류)

Classification



Classification

이진 분류기

: 입력된 데이터를 두 그룹(참 혹은 거짓)으로 분류하는 것

참 : 숫자 5를 가리키는 이미지
거짓 : 숫자 5 이외의 수를 가리키는 이미지

이진분류가 잘 되었는지 확인하는 방법?

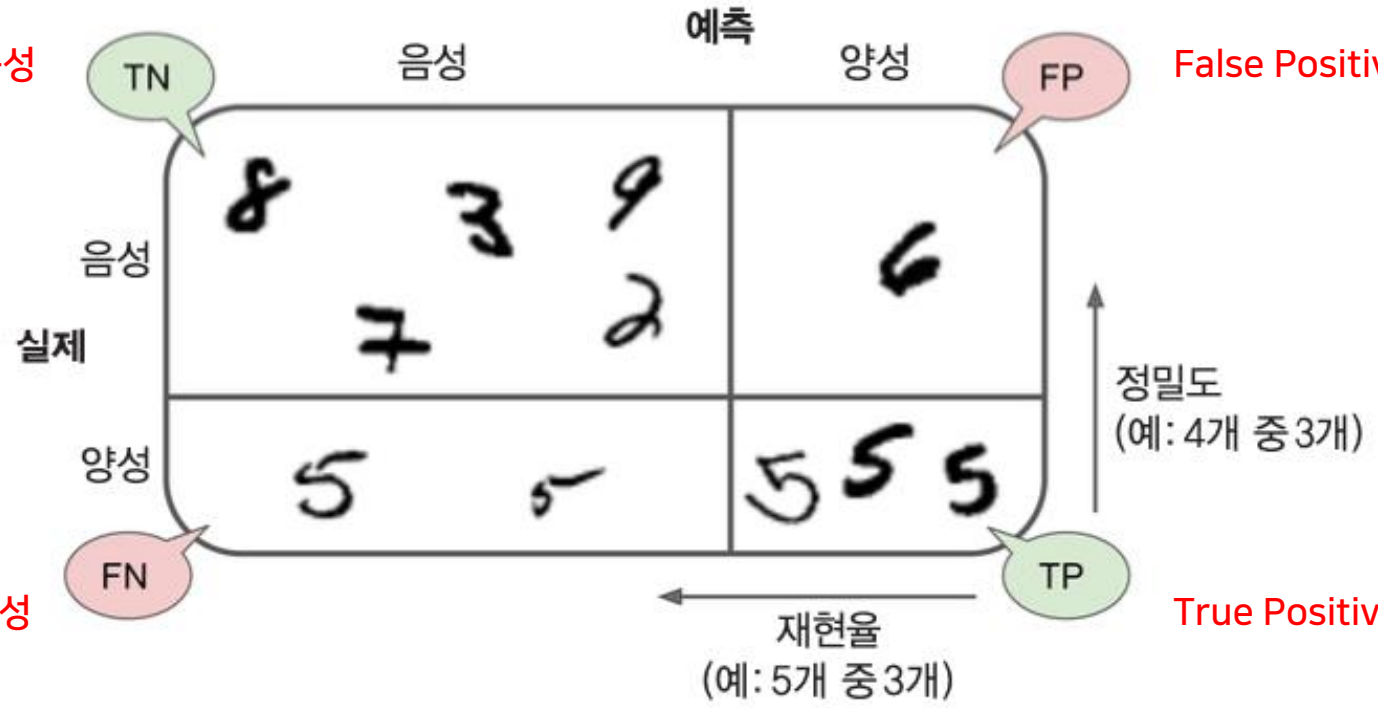
Confusion matrix (오차행렬)

레이블별 예측 결과를 정리한 행렬

참 : 숫자 5를 가리키는 이미지
거짓 : 숫자 5 이외의 수를 가리키는 이미지

True Negative : 진짜 음성

False Positive : 가짜 양성



False Negative : 가짜 음성

True Positive : 진짜 양성

```
array([[53892,   687],
       [ 1891,  3530]])
```

Precision / Recall

Precision

$$\text{정밀도} = \frac{TP}{TP + FP}$$

진짜 양성

진짜 양성 + 가짜 양성

= 양성이라 예측한 것 중 진짜 양성

Recall

$$\text{재현율} = \frac{TP}{TP + FN}$$

진짜 양성

진짜 양성 + 가짜 음성

= 진짜 양성 중 양성이라 잘 예측한 비율

Precision vs Recall

모델 사용 목적에 따라 Precision과 recall의 중요도가 다를 수 있음.

Recall > Precision

암 진단 기준
- 실제 암인 사람에게 암이라고 진단한 비율

Precision > Recall

아동용 동영상 선택기준
- 안전한 동영상 중 실제로 안전한 동영상의 비율

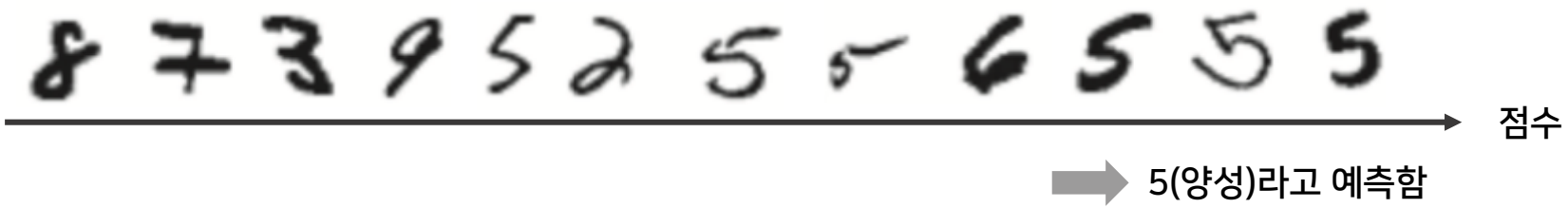
F1 점수

정밀도(Precision)와 재현율(Recall)의 조화 평균 F1 점수를 이용하여 분류기의 성능을 평가하기도 함.

$$\mathrm{F}_1 = \frac{2}{\frac{1}{\text{정밀도}} + \frac{1}{\text{재현율}}}$$

Precision과 Recall의 Trade Off

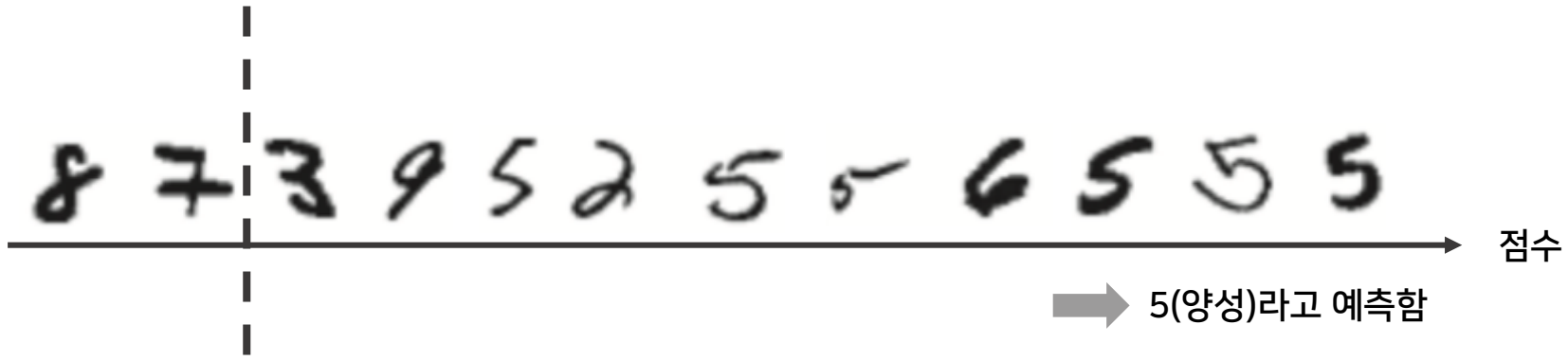
Precision과 Recall의 상호 반비례 관계



Precision과 Recall의 Trade Off

Precision과 Recall의 상호 반비례 관계

결정 임계값(decision threshold)

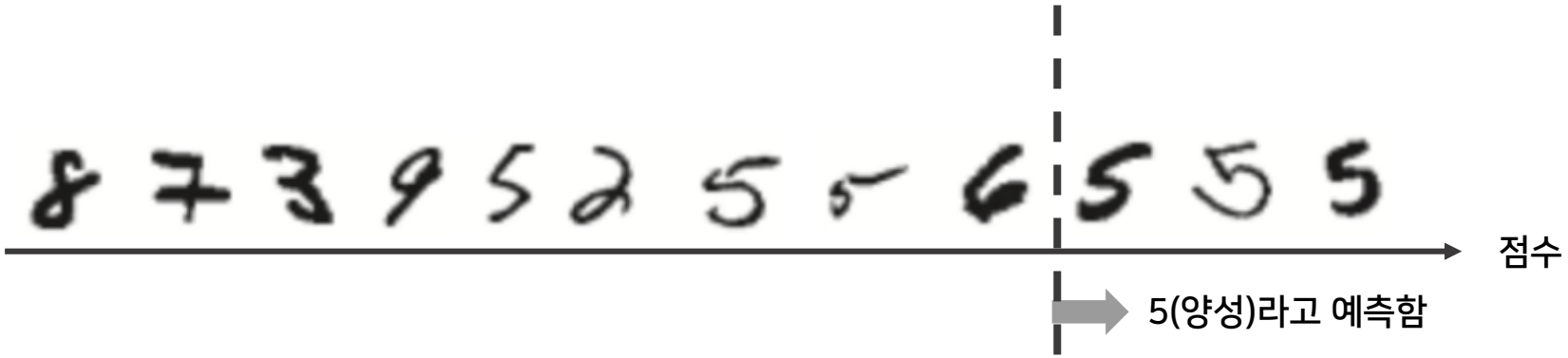


Precision : $\frac{6}{10} = 60\%$

Recall : $\frac{6}{6} = 100\%$

Precision과 Recall의 Trade Off

Precision과 Recall의 상호 반비례 관계

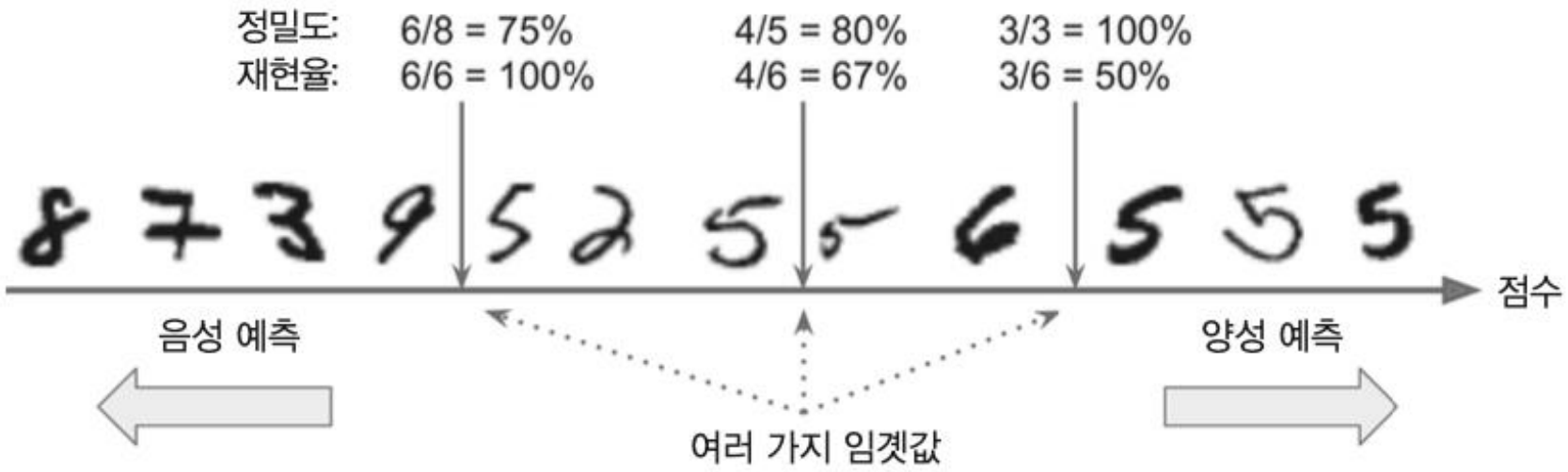


Precision : $\frac{3}{3} = 100\%$

Recall : $\frac{3}{6} = 50\%$

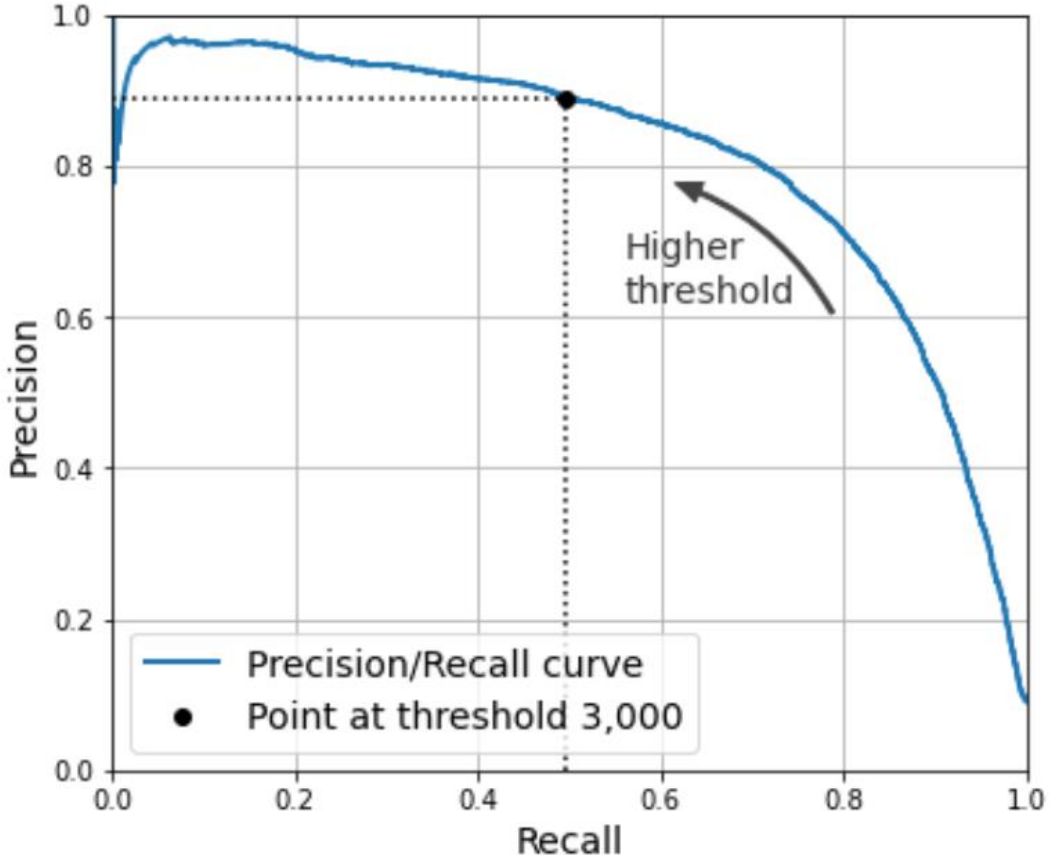
Precision과 Recall의 Trade Off

Precision과 Recall의 상호 반비례 관계



결정임계값(decision threshold)를 높이 설정할수록 Precision은 올라가지만 Recall 은 떨어진다.

Precision과 Recall의 Trade Off

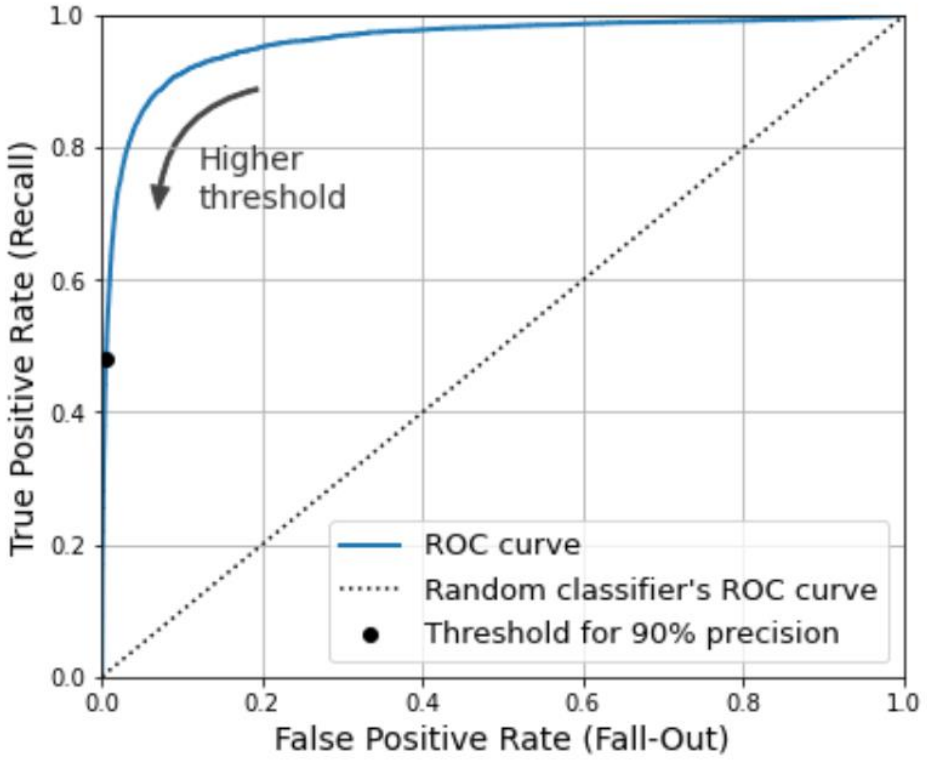


ROC(Receiver Operating Characteristic) 곡선

ROC 곡선

- ROC 곡선을 이용하여 이진 분류기의 성능을 측정할 수 있음.
- 거짓 양성 비율 (FP Rate) 에 대한 참 양성 비율 (TP Rate, Recall)의 관계를 나타내는 곡선임.

(참에 대해 참이라고 잘 예측한 비율)



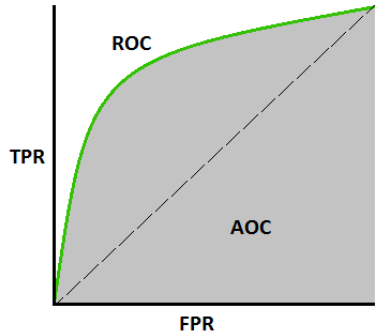
(거짓에 대해 참이라고 잘못 예측한 비율)

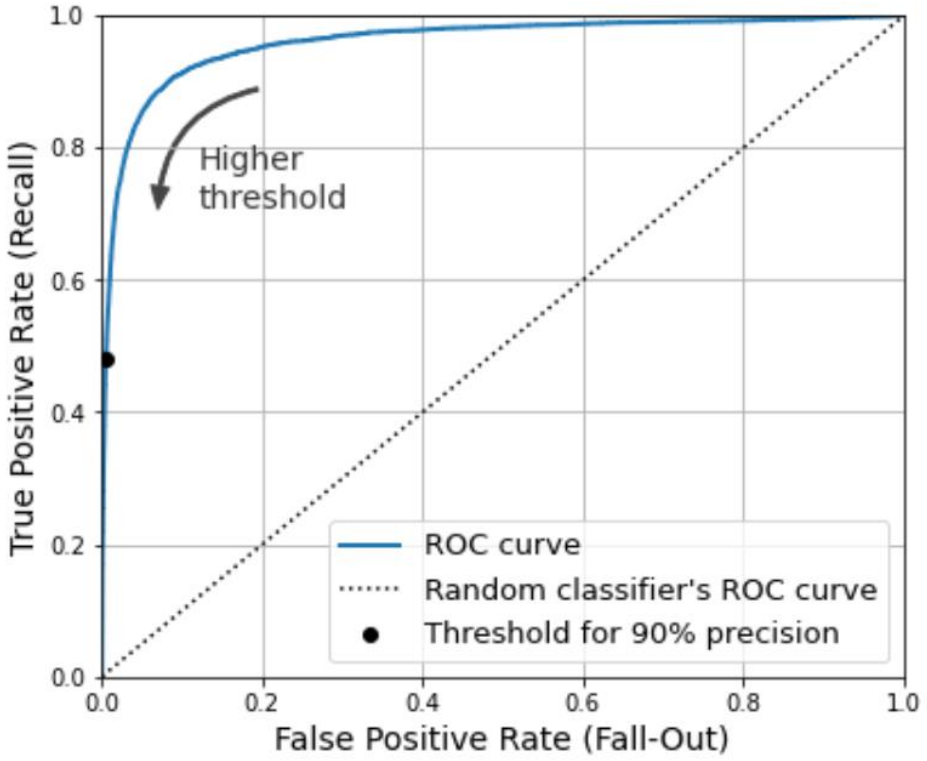
$$\mathrm{FPR} = \frac{FP}{FP + TN}$$

AUC(Area Under the Curve)

AUC

- Recall은 높게, 거짓양성비율은 낮게 유지할수록 좋은 분류기.
- ROC 커브가 y축에 최대한 근접해야 하고, ROC 커브 아래 면적, 즉 AUC가 1에 가까울 수록 좋은 성능임을 나타냄.





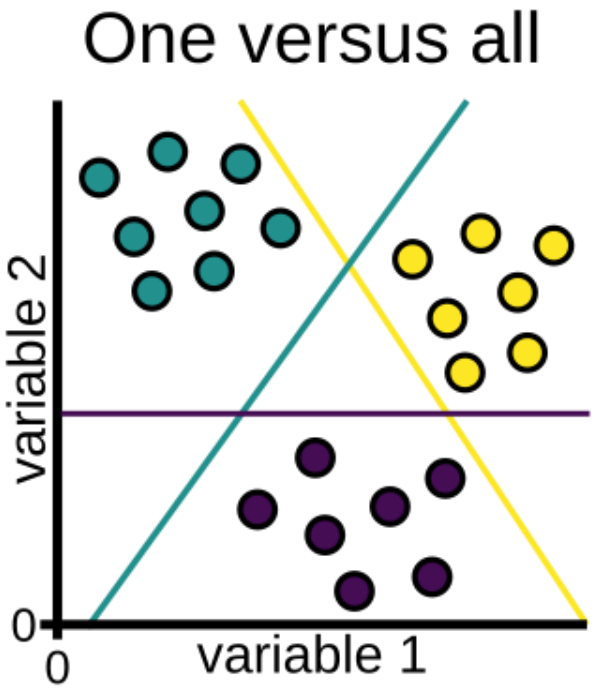
$$\mathrm{FPR} = \frac{FP}{FP + TN}$$

= 실제 거짓인 것 중
참이라고 잘못 예측한 비율

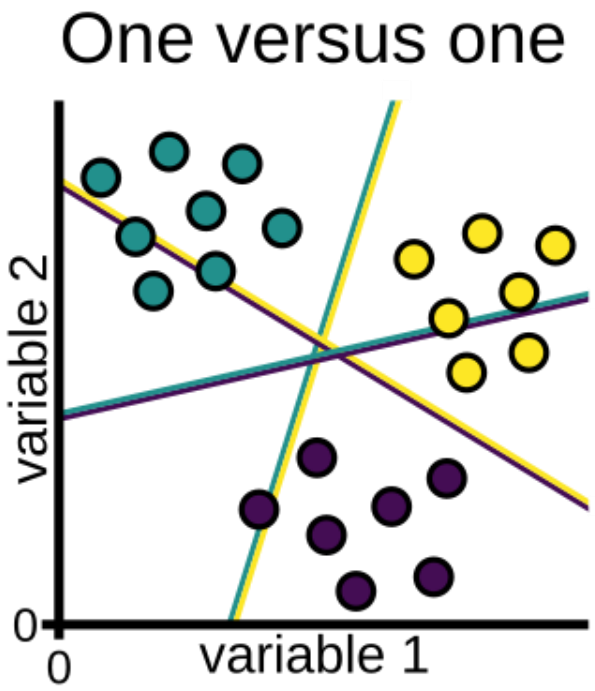
다중 클래스 분류

다중 클래스 분류(Multiclass Classification)

- 세개 이상의 클래스로 데이터를 분류하며, 다항 분류(multinomial Classification)라고도 불린다.



OvR, OvA



OvO

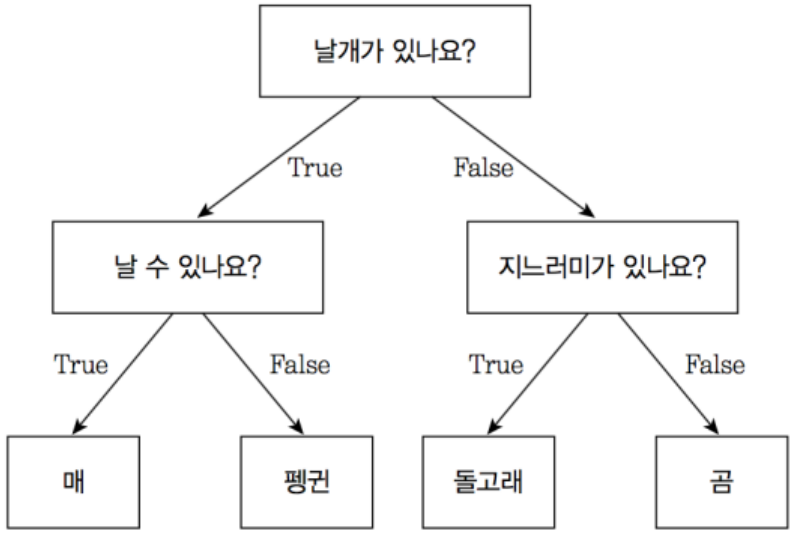
분류 알고리즘

01 결정트리

머신
러닝

결정트리

결정트리는 매우 쉽고 유연하게 적용 될 수 있는 알고리즘.

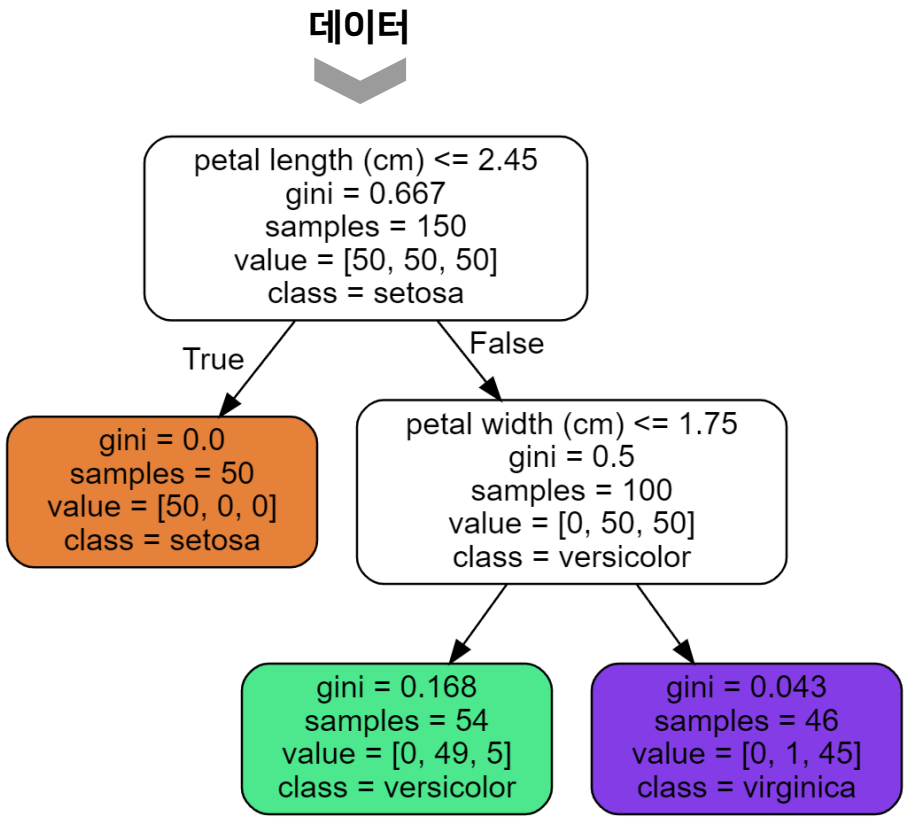


결정 트리 알고리즘은 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리(Tree) 기반의 분류 규칙을 만듦. (If-Else 기반 규칙)

결정트리

```
>>> from sklearn.tree import DecisionTreeClassifier

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

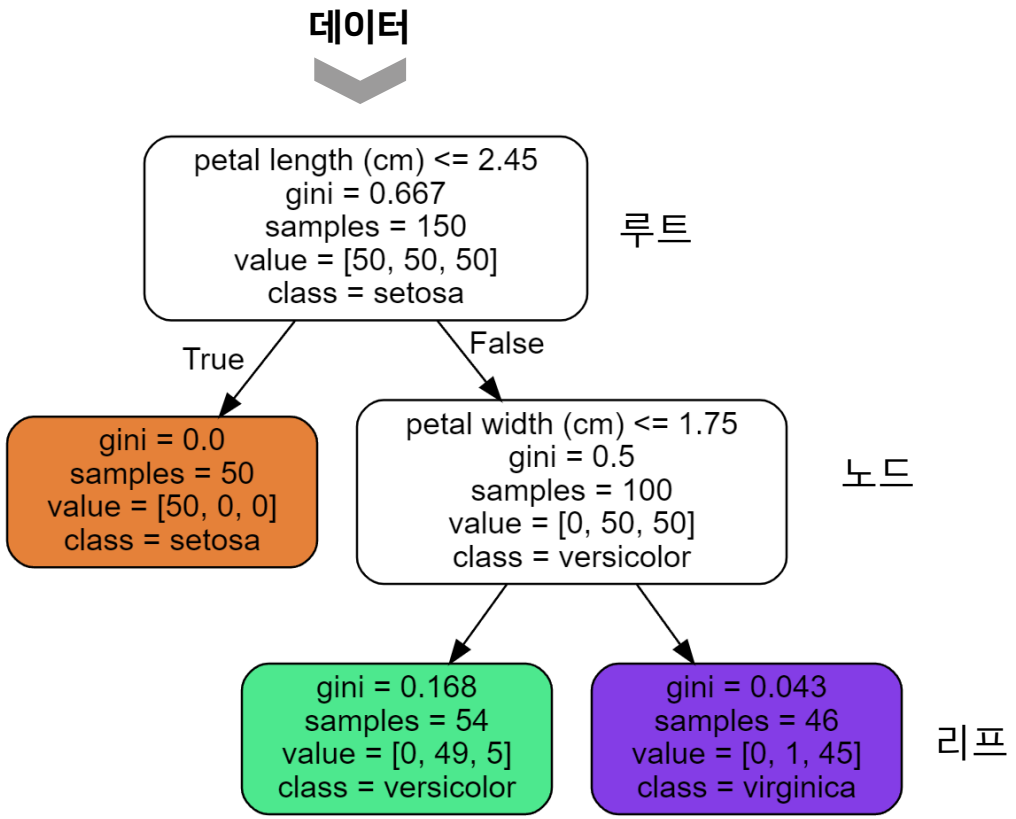


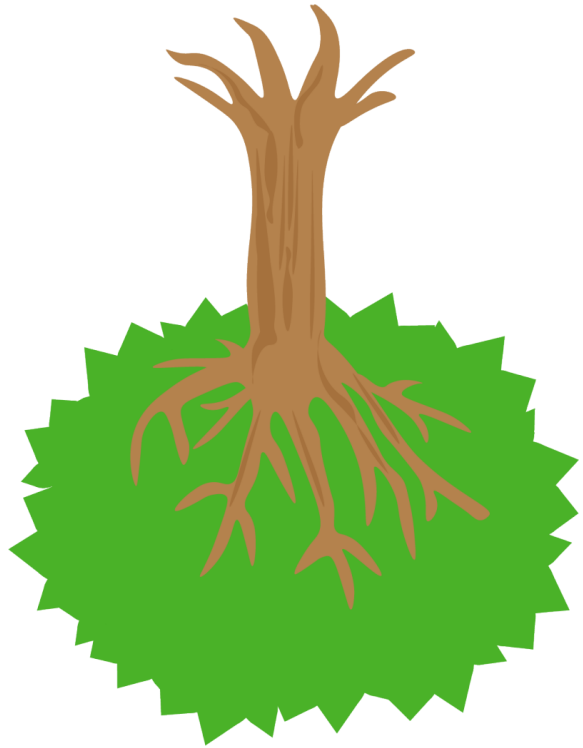
gini : 해당 노드의 지니 불순도 측정값.
samples : 해당 노드에 속하는 샘플 수
value : 해당 노드에 속하는 샘플들의 실제 클래스별 개수
class : 각 클래스별 비율을 계산하여 가장 높은 비율에 해당하는 클래스 선정

결정트리

```
>>> from sklearn.tree import DecisionTreeClassifier

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





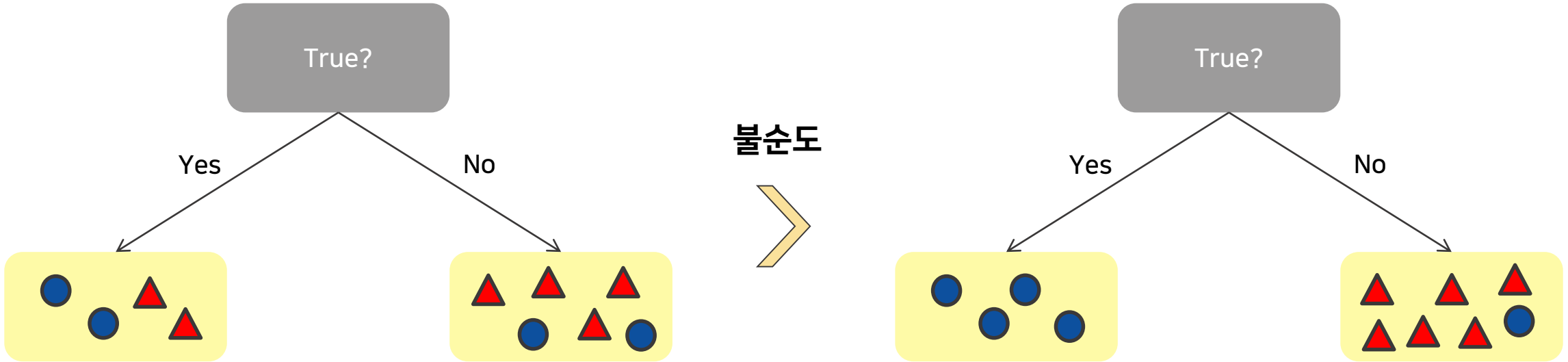
gini : 해당 노드의 지니 불순도 측정값.
samples : 해당 노드에 속하는 샘플 수
value : 해당 노드에 속하는 샘플들의 실제 클래스별 개수
class : 각 클래스별 비율을 계산하여 가장 높은 비율에 해당하는 클래스 선정

gini란?

불순도를 측정하는 gini

- 결정트리의 분리가 잘 된 것을 평가하기 위한 지표
- 0에 가까울 수록 해당 클래스에 섞인 다른 클래스의 양이 적음을 의미함.

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$



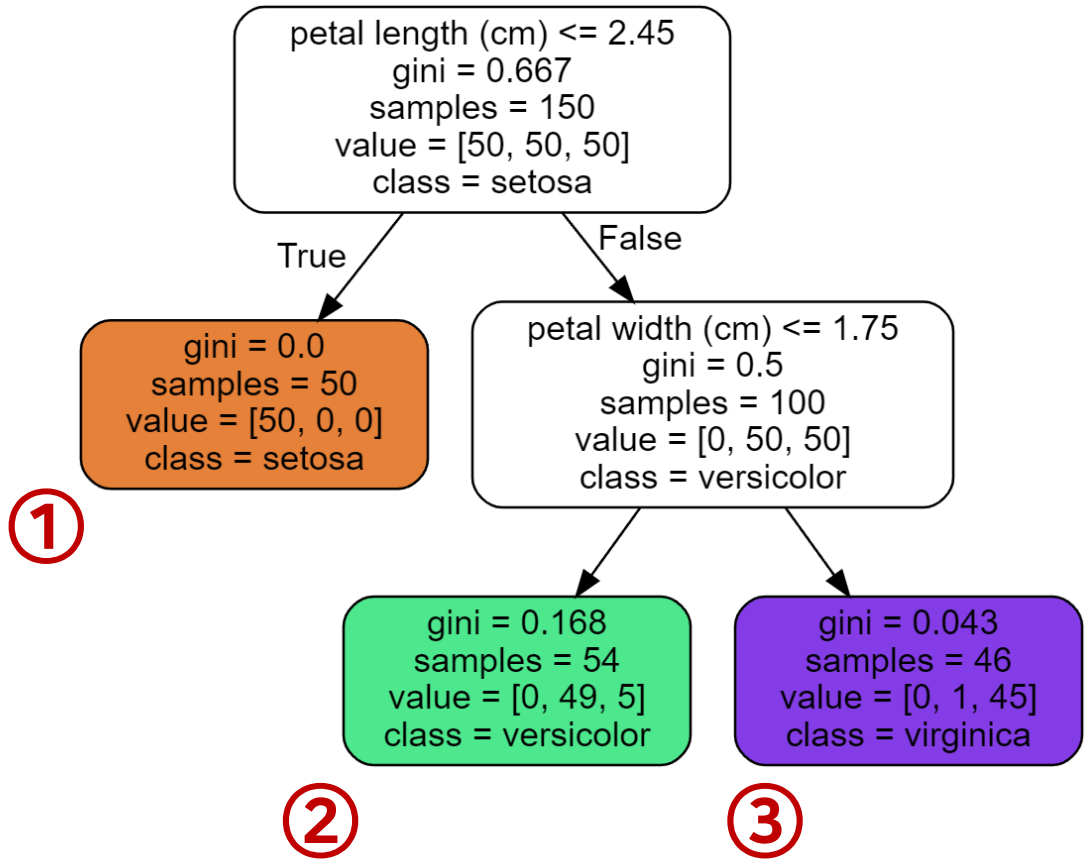
$1 - \{(\frac{2}{4})^2 + (\frac{2}{4})^2\} = 0.5$

$1 - \{(\frac{2}{6})^2 + (\frac{4}{6})^2\} = 0.44$

$1 - \{(\frac{0}{4})^2 + (\frac{4}{4})^2\} = 0.0$

$1 - \{(\frac{1}{7})^2 + (\frac{6}{7})^2\} = 0.244$

결정트리

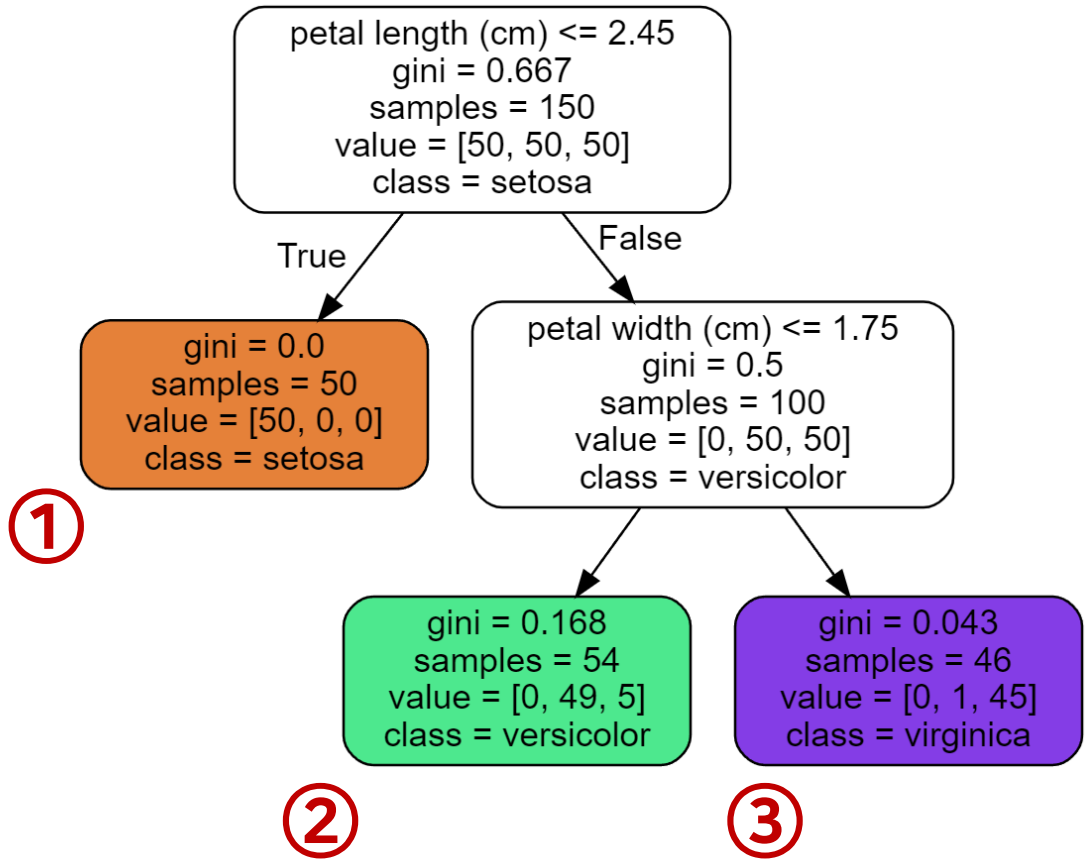


Q.

Sepal length	5.0 cm
Sepal width	2.0 cm
Petal length	3.0 cm
Petal width	1.5 cm

A. ???

결정트리

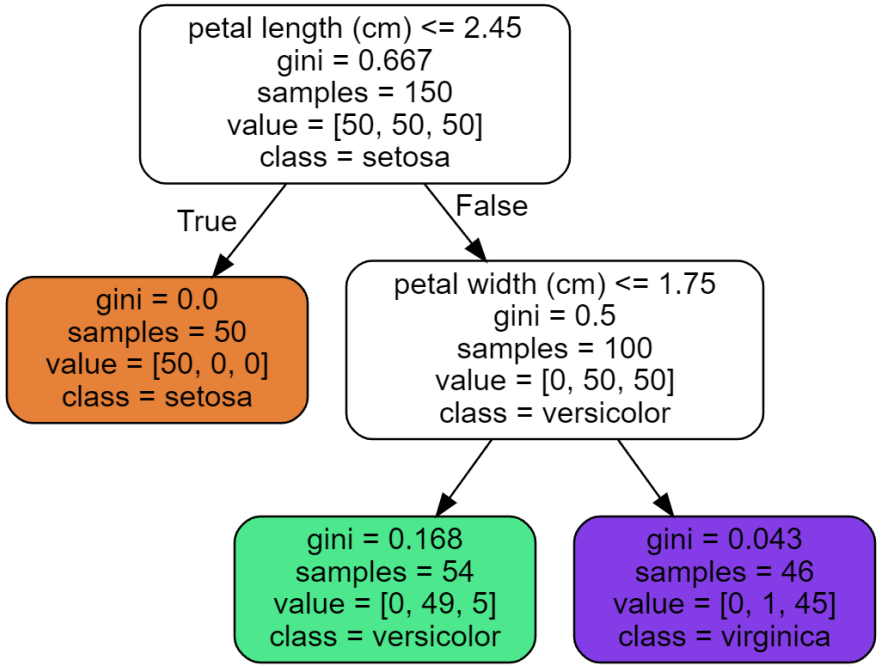


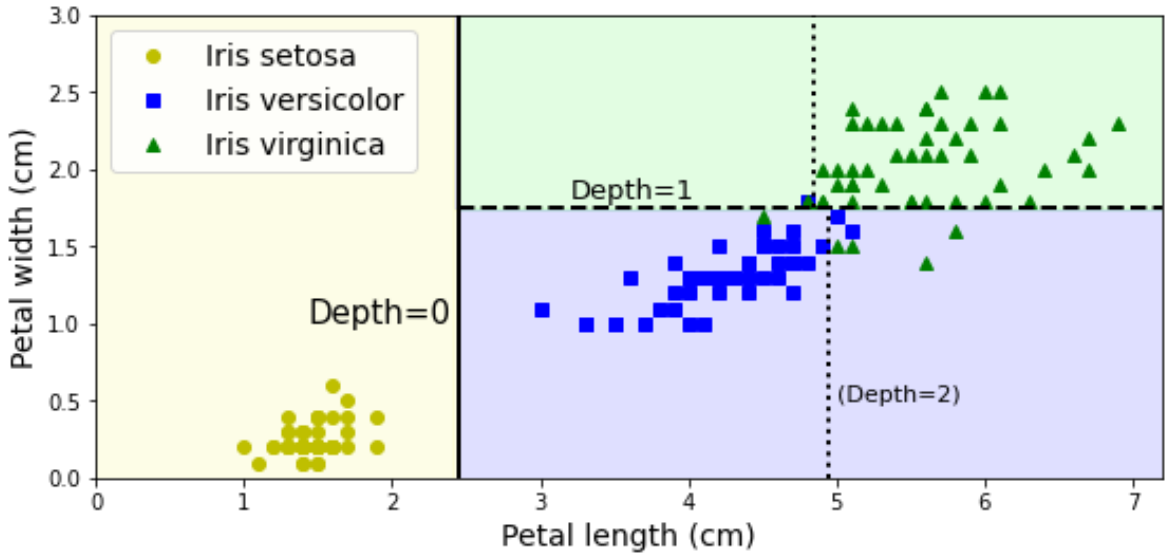
Q.

Sepal length	4.0 cm
Sepal width	1.0 cm
Petal length	2.45 cm
Petal width	2.0 cm

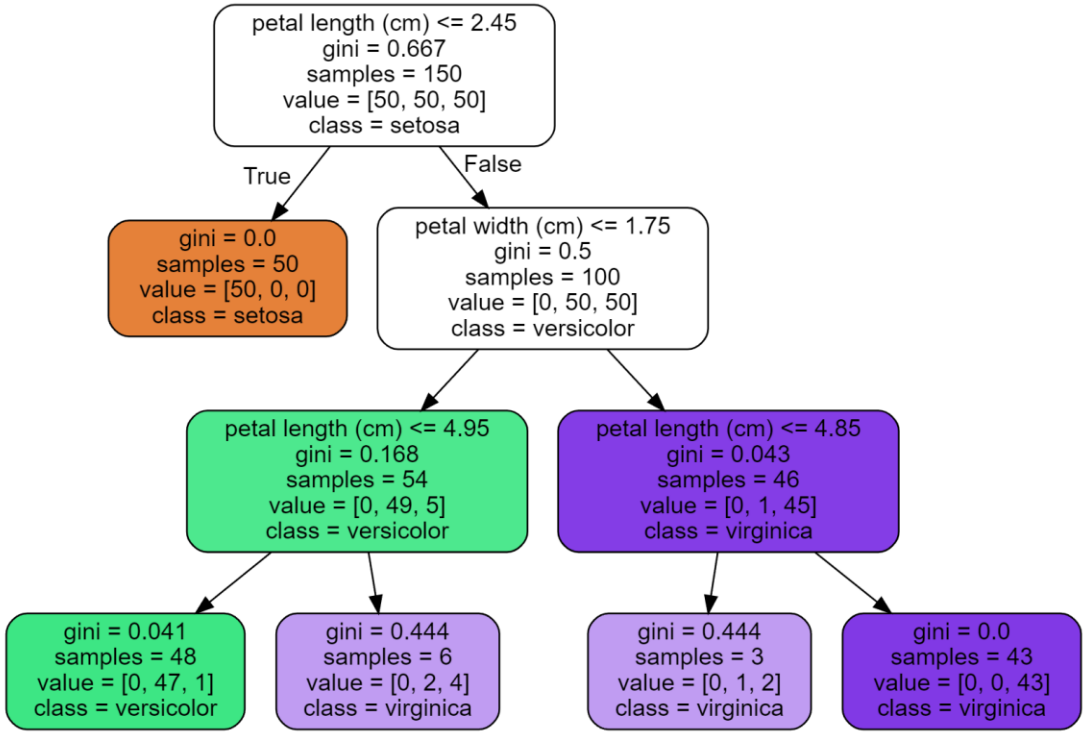
A. ???

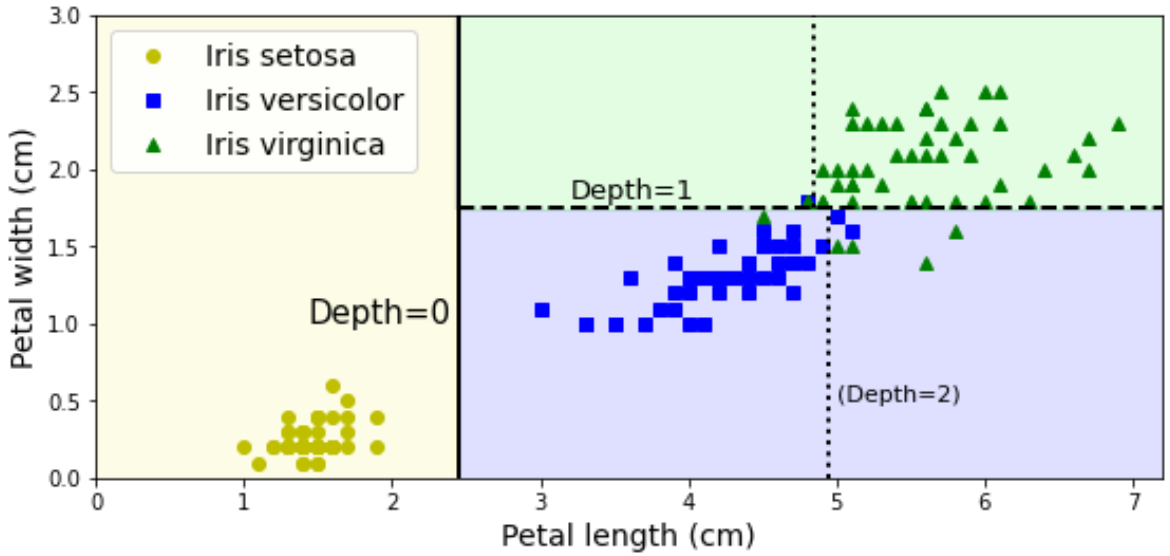
결정트리


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


Iris setosa
Iris versicolor
Iris virginica
Depth=1
Depth=0
(Depth=2)
Petal width (cm)
Petal length (cm)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7

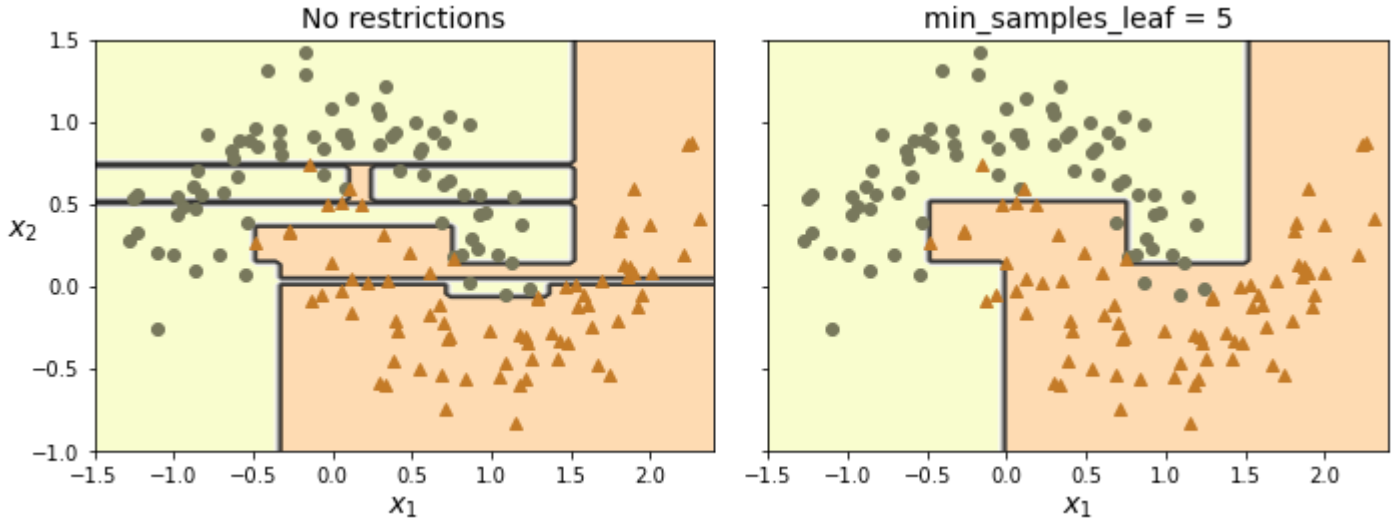
결정트리


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
petal length (cm) <= 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
petal length (cm) <= 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
gini = 0.041
samples = 48
value = [0, 47, 1]
class = versicolor
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
gini = 0.0
samples = 43
value = [0, 0, 43]
class = virginica


Iris setosa
Iris versicolor
Iris virginica
Depth=1
Depth=0
(Depth=2)
Petal width (cm)
Petal length (cm)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
6
7

규제 하이퍼파라미터

하이퍼파라미터	기능
max_depth	결정트리의 높이 제한
min_samples_split	노드 분할해 필요한 최소 샘플 개수
min_samples_leaf	리프에 포함된 최소 샘플 개수
min_weight_fraction_leaf	샘플 가중치 합의 최솟값
max_leaf_nodes	최대 리프 개수
max_features	분할에 사용되는 특성 개수



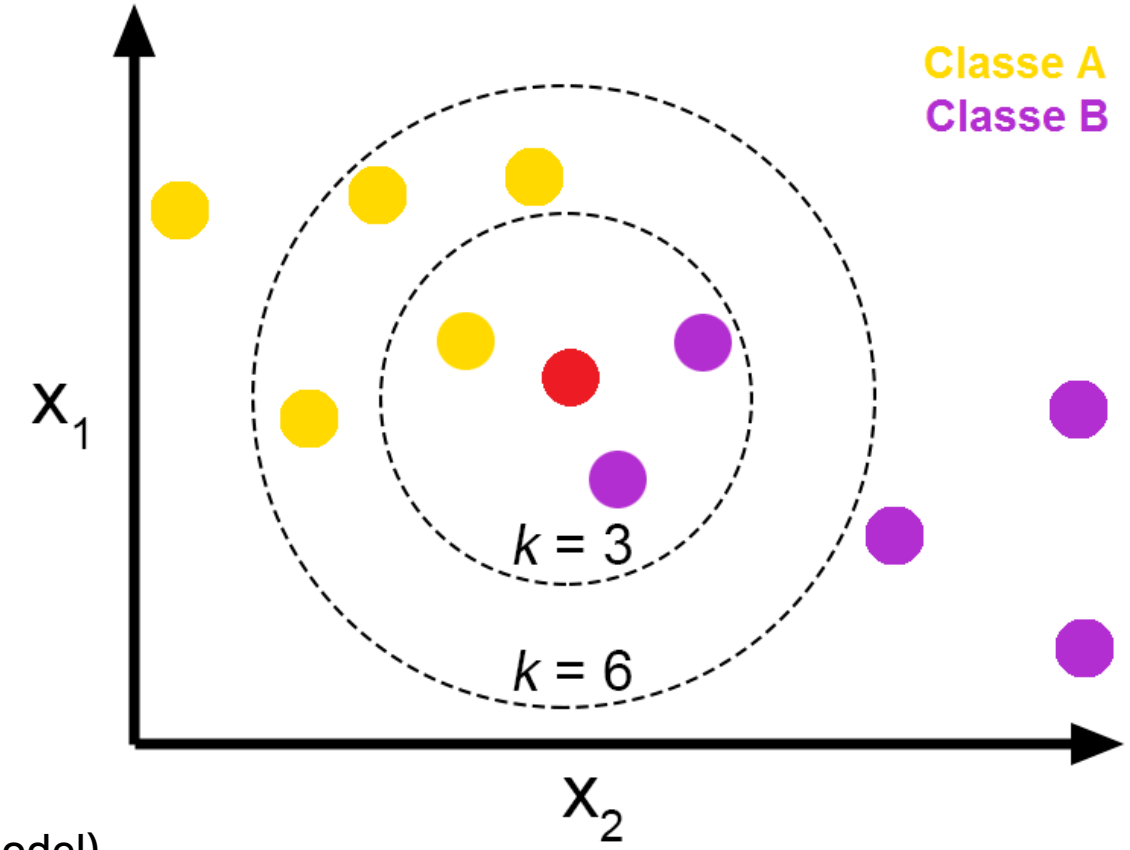
결정트리의 장단점

장점	쉽다. 직관적이다. 데이터 가공의 영향도가 크지 않다.
단점	과적합(Overfitting)으로 성능이 떨어질 수 있다. (규제 필요)

KNN(K-Nearest Neighbor) 알고리즘

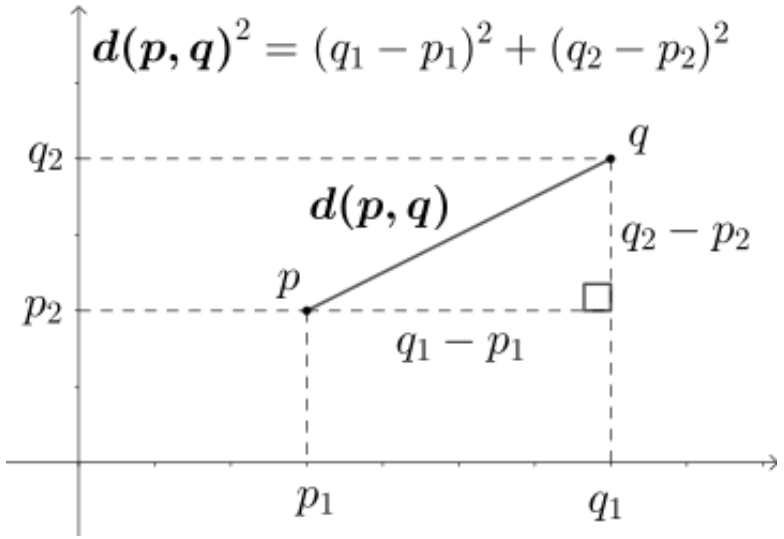
KNN은 주변 데이터(neighbor)를 살펴본 뒤 더 많은 데이터가 포함되어 있는 클래스로 분류하는 방식

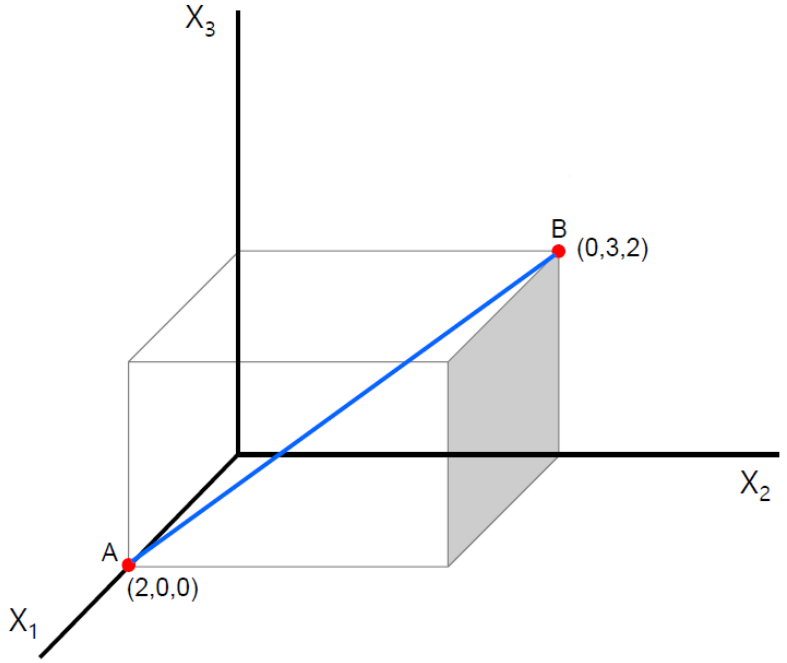
K의 개수만큼 주변의 데이터를 살펴본다는 뜻.



학습이 필요없다! (Lazy model)

유클리드 거리(Euclidean Distance)





$$d_{(A,B)} = \sqrt{(0-2)^2 + (3-0)^2 + (2-0)^2} = \sqrt{17}$$

[실습] iris 붓꽃 데이터를 활용한 결정트리, KNN 알고리즘 적용

Id	꽃받침길이	꽃받침너비	꽃잎길이	꽃잎너비	품종
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...

분류 알고리즘

01 앙상블

02 랜덤 포레스트

03 부스팅과 스태킹

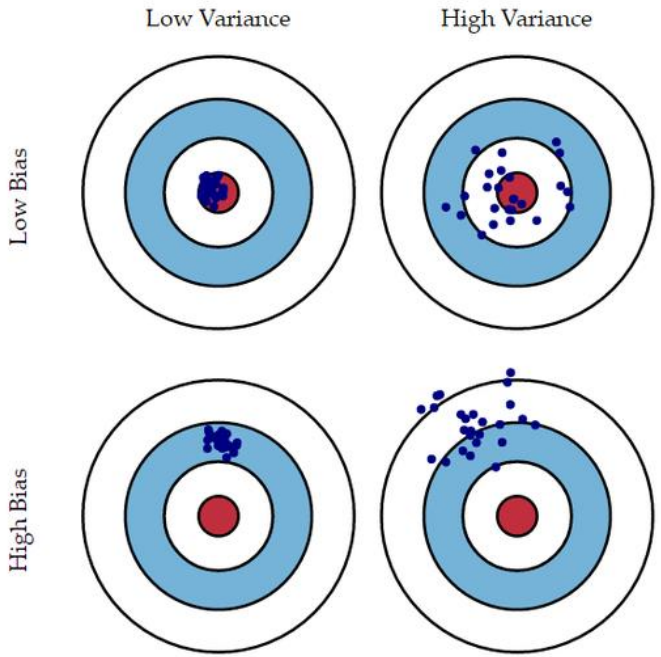
머신
러닝.

편향과 분산

앙상블 학습의 핵심은 편향과 분산을 줄인 모델을 구현하는 것

- 편향(Bias) : 예측값과 정답이 떨어져 있는 정도
- 정답에 대한 잘못된 가정으로부터 유발되며 편향이 크면 과소적합이 발생한다.

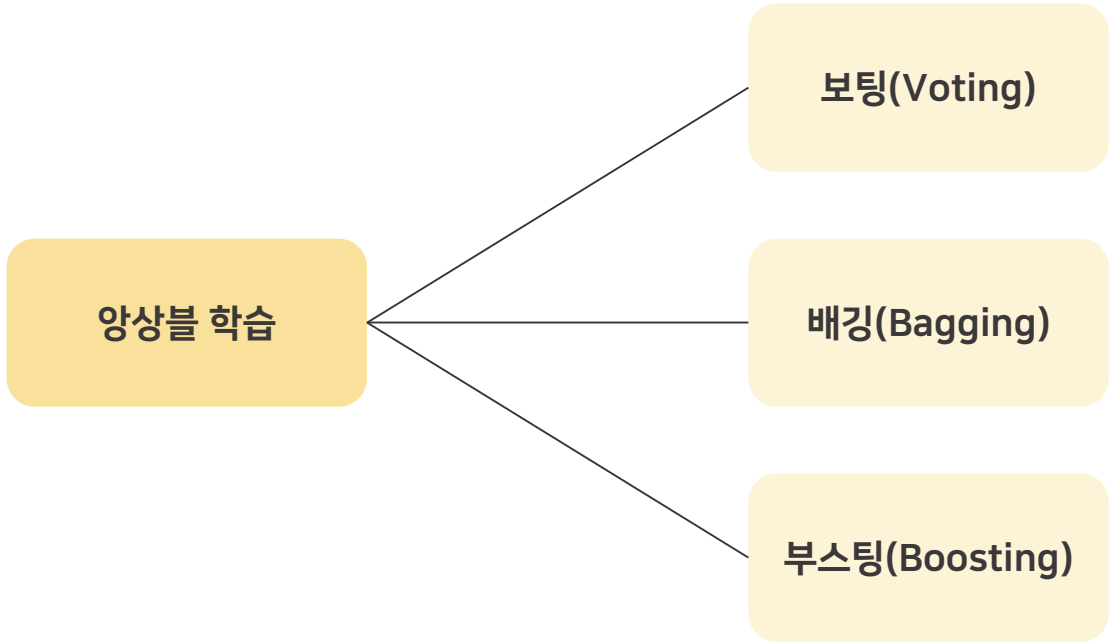
- 분산(Variance) : 입력 샘플의 작은 변동에 반응하는 정도
- 정답에 대한 너무 복잡한 모델을 설정하는 경우 분산이 커지며, 분산이 크면 과대적합이 발생한다.



앙상블 학습과 랜덤포레스트

앙상블 학습이란?

- 여러 개의 분류기를 생성하고, 그 예측을 결합함으로써 보다 정확한 예측을 도출하는 기법
- 강력한 하나의 모델을 사용하는 대신 약한 모델 여러 개를 조합하여 더 정확한 예측에 도움을 주는 방식



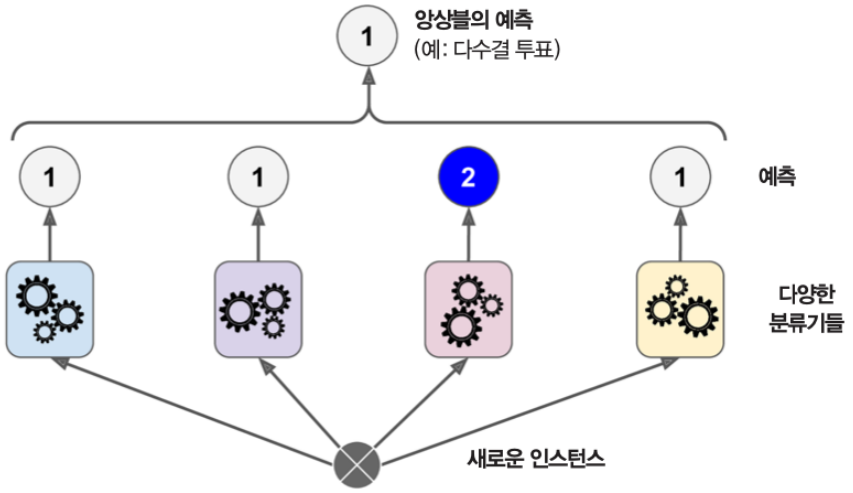
http://www.dinnopartners.com/__trashed-4/

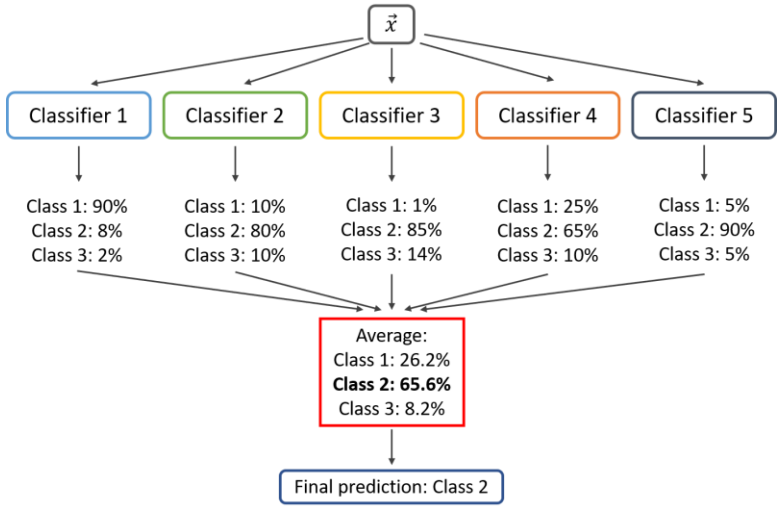
보팅(Voting)

보팅(Voting)

- 여러 개의 분류기가 투표를 통해 최종 예측결과를 결정하는 방식
- 서로 다른 알고리즘을 여러 개 결합하여 사용

하드 보팅(Hard Voting)	다수의 분류기가 예측한 결과값을 최종 결과로 선정
소프트 보팅(Soft Voting)	모든 분류기가 예측한 레이블 값의 결정 확률 평균을 구한 뒤 가장 확률이 높은 레이블 값을 최종 결과로 선정

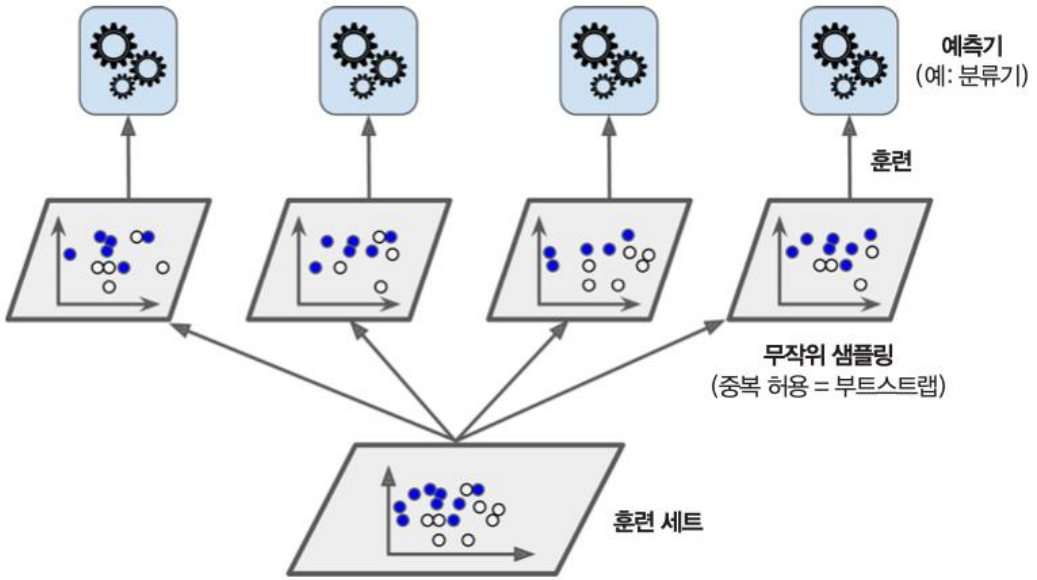




배깅(Bagging)

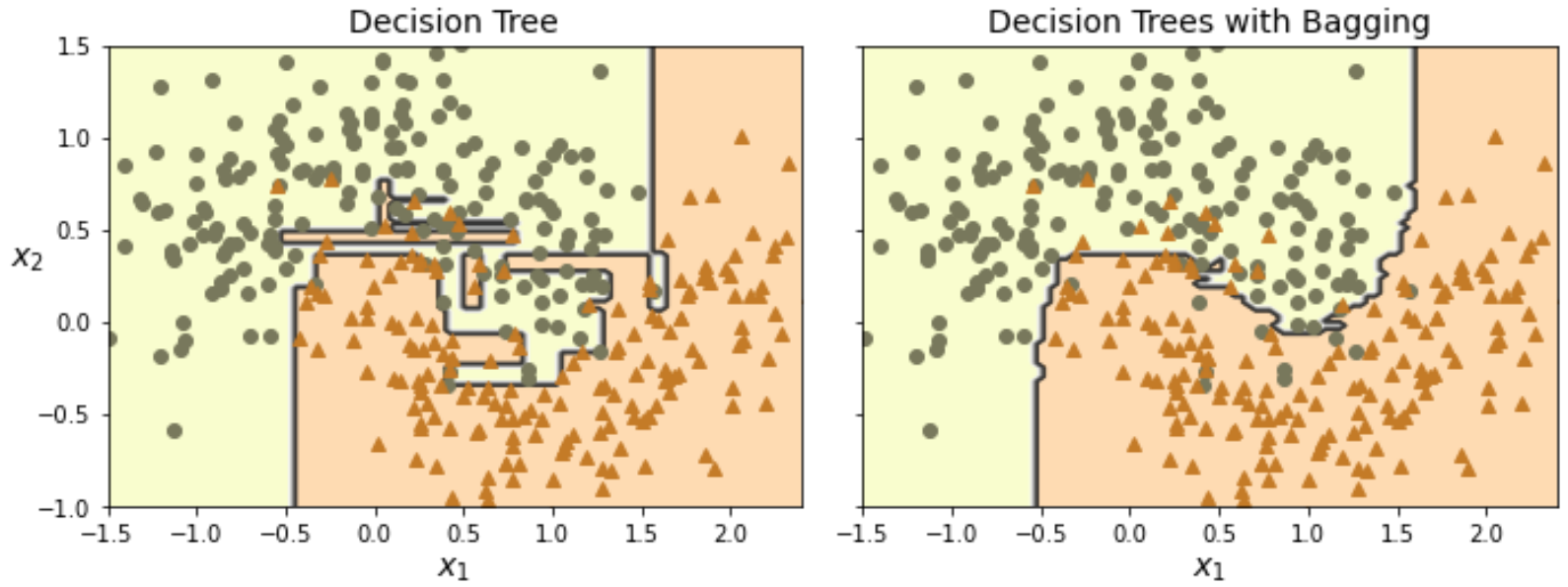
배깅(Bagging) – Bootstrap Aggregation의 줄임말 (Bootstrap : 중복허용 리샘플링)

- 데이터 샘플링을 통해 모델을 학습시키고 결과를 집계하는 방법
- 모두 같은 유형의 알고리즘 기반의 분류기를 사용
- 예측값의 최빈값을 최종 예측값으로 선택
- 대표 배깅 방식 : 랜덤 포레스트 알고리즘



배깅(Bagging)

데이터를 샘플링 하게 되면 표본의 다양성을 많이 추가하게 되기 때문에, 분산이 줄어드는 효과를 가져옴.

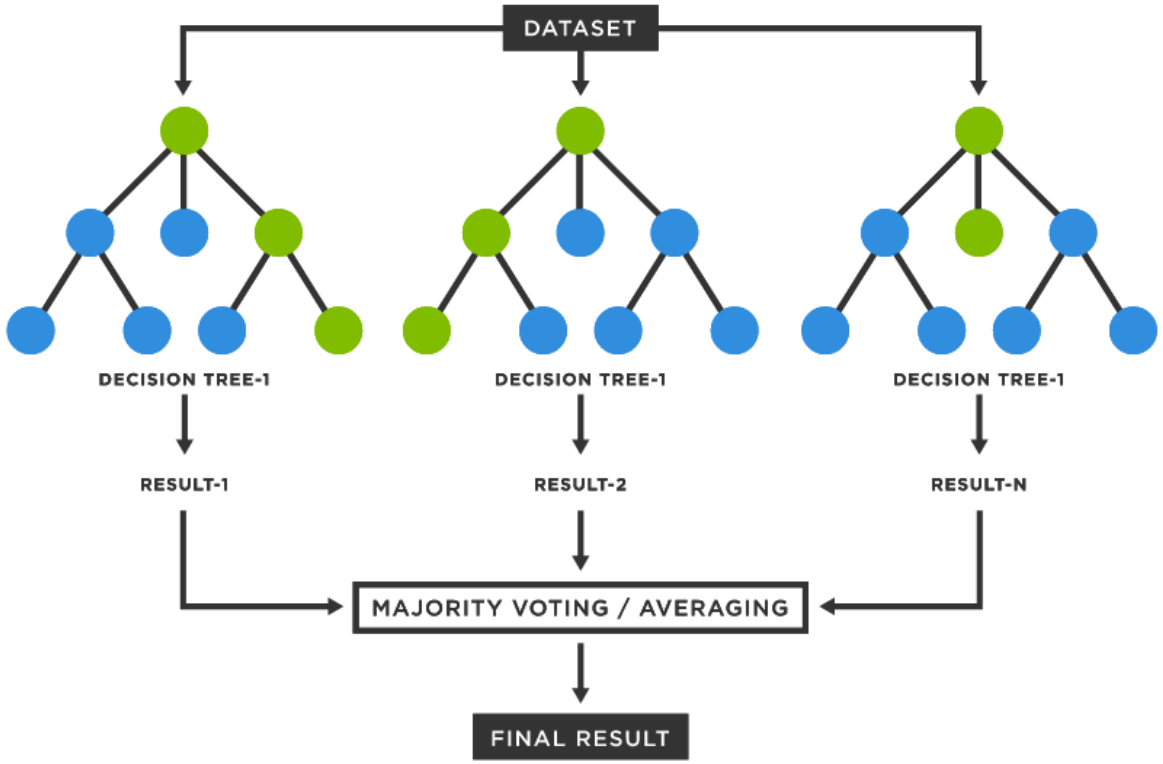


➔ 결정경계의 불규칙성이 줄어듦.
➔ 각 결정트리의 독립성이 커지고, 모델의 일반화 성능이 좋아짐.

랜덤 포레스트

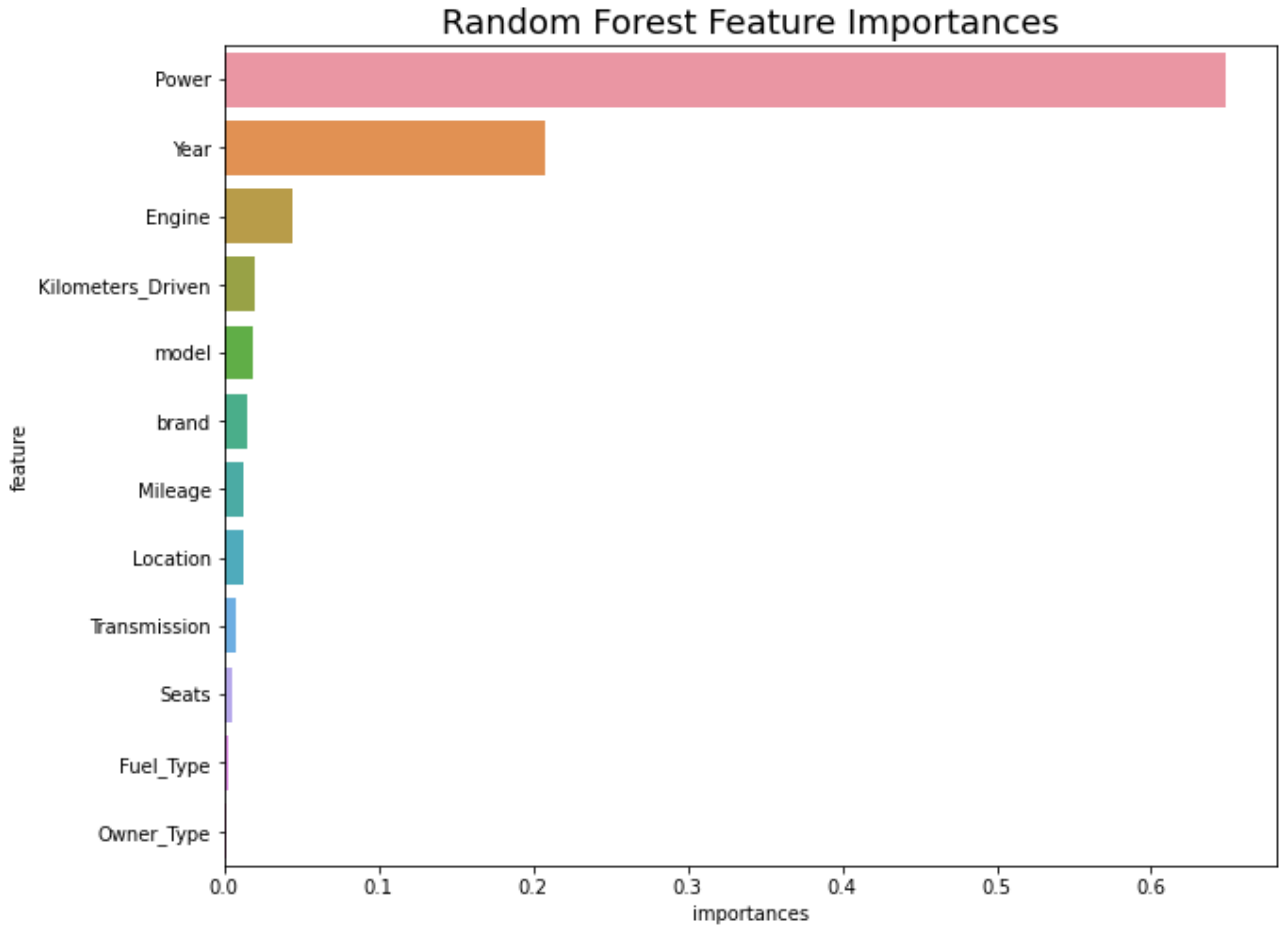
랜덤 포레스트(random forest)

- 배깅 기법을 결정트리의 앙상블에 특화시킨 모델



랜덤 포레스트

Feature Importances



부스팅(Boosting)

부스팅(Boosting)

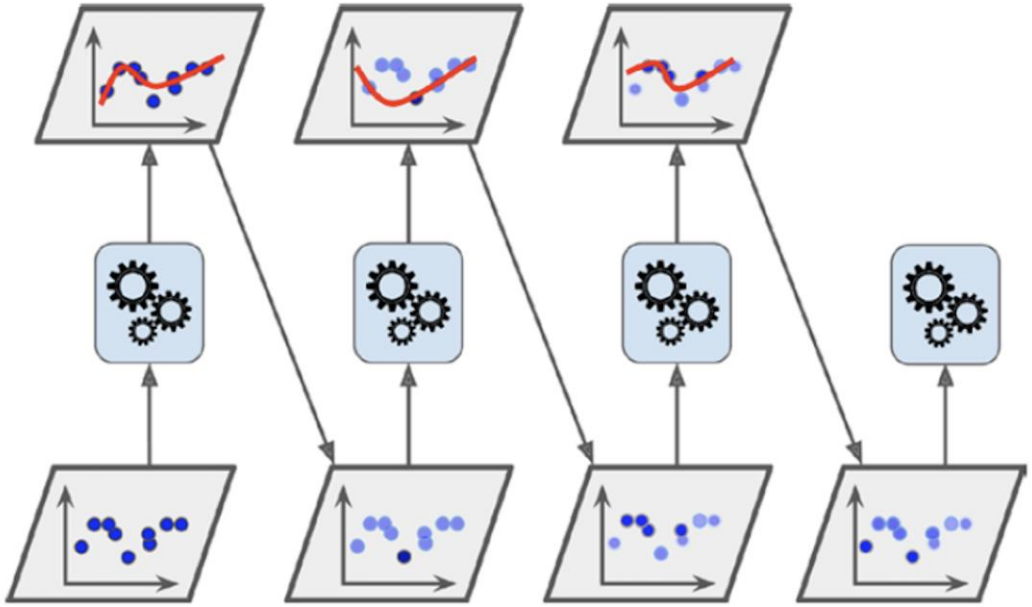
- 여러 개의 분류기가 순차적으로 학습을 진행
- 이전 분류기가 예측이 틀린 데이터에 대해 올바르게 예측할 수 있도록 **다음 분류기에게 가중치를 부여**하면서 학습과 예측을 진행
- 계속하여 분류기에게 가중치를 부스팅하며 학습을 진행하기에 부스팅 방식이라고 불림
- 순차적으로 학습하기에 배깅 방식과는 달리 훈련을 동시에 진행할 수 없으므로 훈련시간이 오래걸림

- **에이다 부스트**
- **그래디언트 부스팅**

에이다 부스트(AdaBoost)

에이다부스트(AdaBoost)

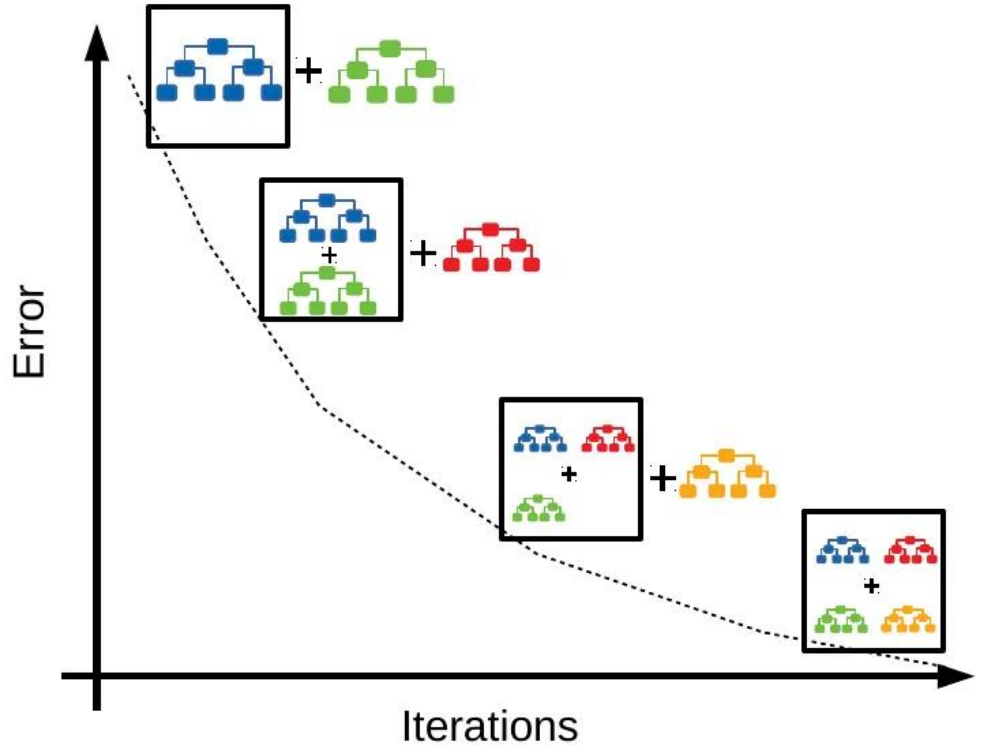
- 하나의 모델을 훈련시킨 후 잘 못 예측된 샘플을 보다 강조하면서 해당모델을 다시 훈련시킨다.



그래디언트 부스팅(Gradient Boosting)

그래디언트 부스팅(Gradient Boosting)

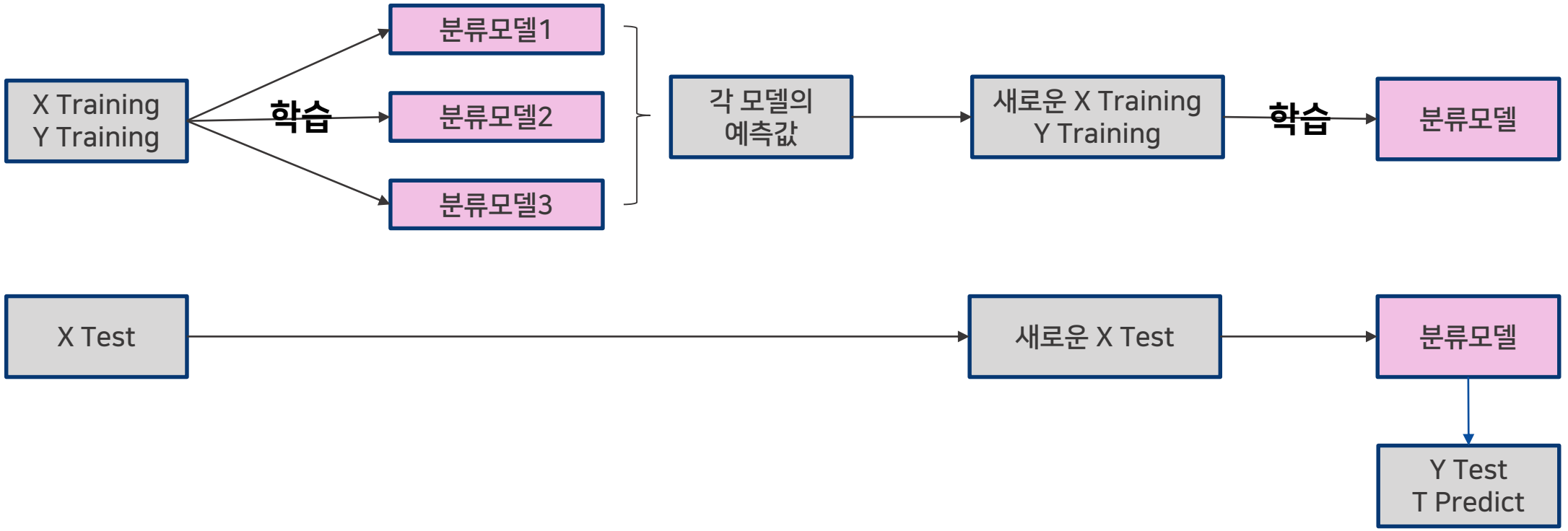
- 이전 모델의 예측에 오차가 있다면 그 오차를 보정하는 새로운 예측기를 새롭게 훈련시킨다.
- 에이다부스트 기법은 샘플의 가중치를 조정하는 반면, 그래디언트 부스팅기법은 이전 예측기에 의해 생성된 잔차(residual error)에 대해 새로운 예측기를 학습시킨다.



스태킹(Stacking)

스태킹(Stacking)

- 배깅방식의 응용
- 다수결을 이용하는 대신 여러 예측값을 학습데이터로 활용하는 예측기를 학습시킨다.



[실습] 앙상블, 랜덤포레스트, GBM

실습!

[실습] 밀크T 만료 및 탈퇴 회원 예측 분류

	userid	gender	membertype_codename	grade_codename	memberstatus	memberstatus_codename	memberstatus_change	status_null_count	statusgroup_10_count
0	0001809c-1725-4ccd-86b0-d02ed0937a83	M	초등	초2	11.0	학습생(정)	-,11,-,11,-,11,-,11,-,11,-,11,-,11,-,11	13	0
1	00028ac1-a0ab-486f-bfdd-de2b0bf70980	F	초등	초4	11.0	학습생(정)	-,11,-,11,-,11,-,11,-,11,-,11,-,11,-,11,-,11	15	0
2	00181cb5-7afd-4cb9-ac7c-37aa66796167	F	초등	초2	11.0	학습생(정)	-,11,-,11,-,11,-,11,-,11	10	0
3	001ad7ff-3db5-4705-a036-2d9b6260957d	M	초등	초3	11.0	학습생(정)	-,11,-,11	2	0
4	002a7014-ee46-4a0e-85e6-389214ca3421	M	초등	초3	11.0	학습생(정)	11,-,11,-,11,-,11,-,11	8	0

감사합니다