Chapter 5. ML models

태스크의 구분

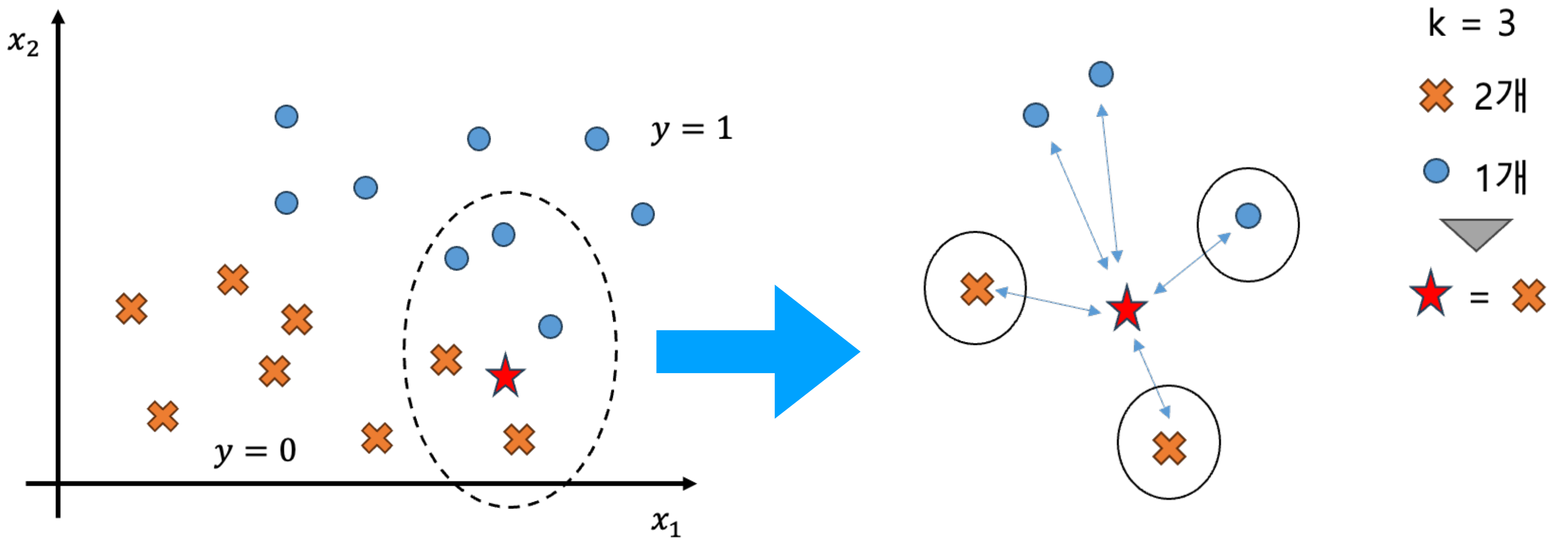
- 회귀 : 실수값, 연속적인 값 예측
- 분류 : 어떤 클래스에 해당하는지 예측
- 클러스터링 : 데이터를 여러개의 그룹으로 나누기
- 차원축소 : 데이터를 잘 설명하는 저차원 데이터로의 변환

다양한 머신러닝 모델들

- 선형 회귀 (Linear Regression)
- 로지스틱 회귀 (Logistic Regression)
- 의사결정 트리 (Decision Tree)
- 랜덤 포레스트 (Random Forest)
- **그래디언트 부스팅 트리 (Gradient Boosting Tree)**
  - XGBoost
  - LightGBM
  - CatBoost
- 서포트 벡터 머신 (Support Vector Machine, SVM)
- 나이브 베이즈 (Naive Bayes)
- K-최근접 이웃 (K-Nearest Neighbors, KNN)
- **신경망 (Neural Networks)**
  - 다층 퍼셉트론 (Multi-Layer Perceptron, MLP)
  - 컨볼루션 신경망 (Convolutional Neural Network, CNN)
  - 순환 신경망 (Recurrent Neural Network, RNN)
  - 트랜스포머 (Transformer)

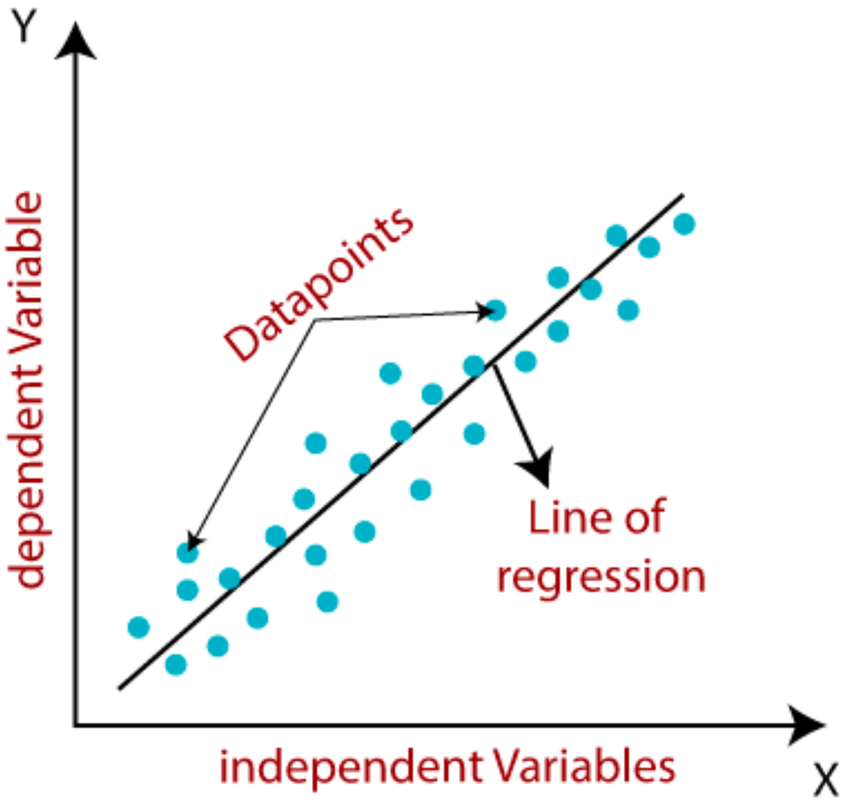
K-최근접 이웃 (KNN)

- 유유상종 : 비슷한 데이터는 가까이에 모여있을 것
- 최근접 이웃의 수, 거리계산 방법에 따라 결과가 달라짐
- **데이터에 대한 별도의 사전 학습 단계가 없음** → 추론시에 거리 계산 및 분류 수행, Lazy learning



선형 회귀

- 종속 변수 Y와 독립 변수 X 사이의 관계
- $Y = a + bx_1 + cx_2 + \ldots + \epsilon$
- 회귀 계수(a, b, c)와 선형 관계
- 선형 회귀에서의 학습은?
- 오차를 최소화 하는 회귀 계수를 찾는 과정
- 정규방정식과 경사하강법

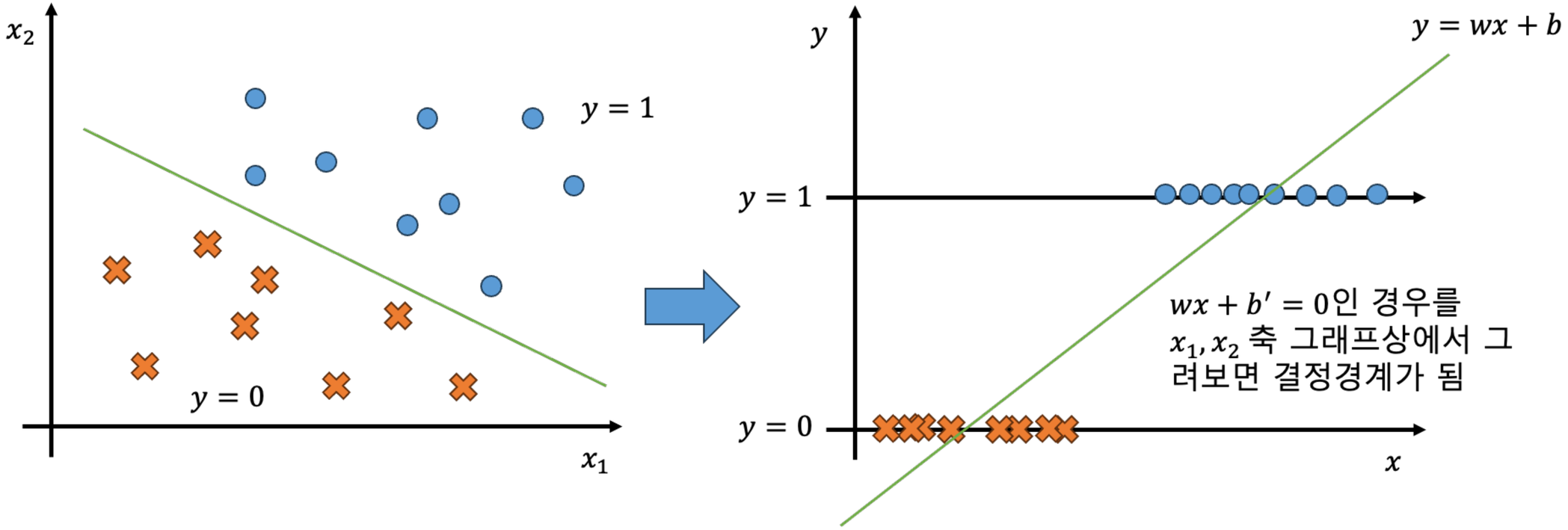


선형 회귀

특성	**Lasso regression** ***L*1 페널티**	**Ridge regression** ***L*2 페널티**
목적	회귀 계수를 0으로 만들어 모델을 단순화가능 특성 선택 가능	계수의 크기를 제한하여 모든 계수가 작게 유지되도록 함
수학적 표현	$\lambda \sum_{i=0}^{d} \lvert w_i \rvert$	$\lambda \sum_{i=0}^{d} (w_i)^2$
효과	변수의 수를 줄임으로써 sparse한 모델을 생성	계수를 축소하지만 모든 변수를 모델에 포함
최적화 난이도	경사하강법을 통한 최적화 수행	정규방정식을 통한 해석적 해가 존재
모델 해석성	높음 (중요 변수만 선택)	낮음 (모든 변수 포함, 계수 축소에 중점)
상황	변수의 수가 많고 중요 변수를 선택하고자 할 때 유용	변수 간의 상관관계가 높고, 변수 제거 없이 모델 복잡도를 제어하고자 할 때 유용

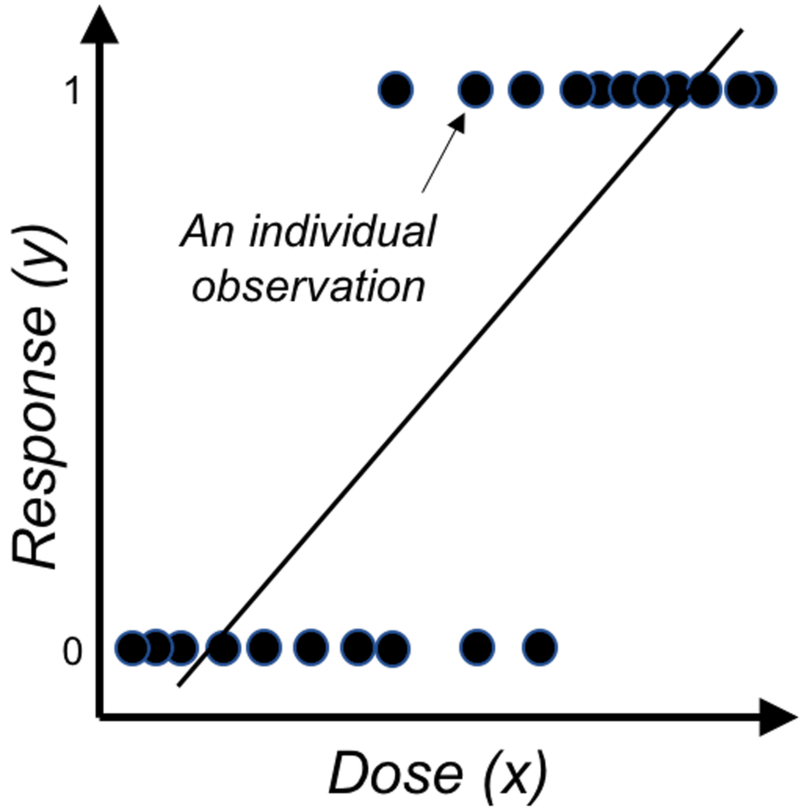
로지스틱 회귀

- 분류 문제를 위한 모델
- 선형 회귀 + 범주형 종속 변수 Y
  → 이진 분류 수행 p 또는 1-p의 확률로 클래스 구분
- 선형 모델 사용시의 문제점은?



로지스틱 회귀

- Log-odd을 독립 변수의 선형 조합으로 예측
- Recall : 선형회귀로 실수값을 예측
- Odds 란? 어떤 사건이 발생할 확률을 p라 할 때 발생확률과 발생하지 않을 확률의 비율
  - p/(1-p) → 클래스가 1일 확률과 1이 아닐 확률
  - p=0일 때 0, p가 1에 가까워지면 ∞
  - Log-odd란? odds를 로그변환 → - ∞ ~ + ∞ (실수범위)
- Logistic function을 통해 확률로 변환: $p = 1/(1 + \exp(-z))$



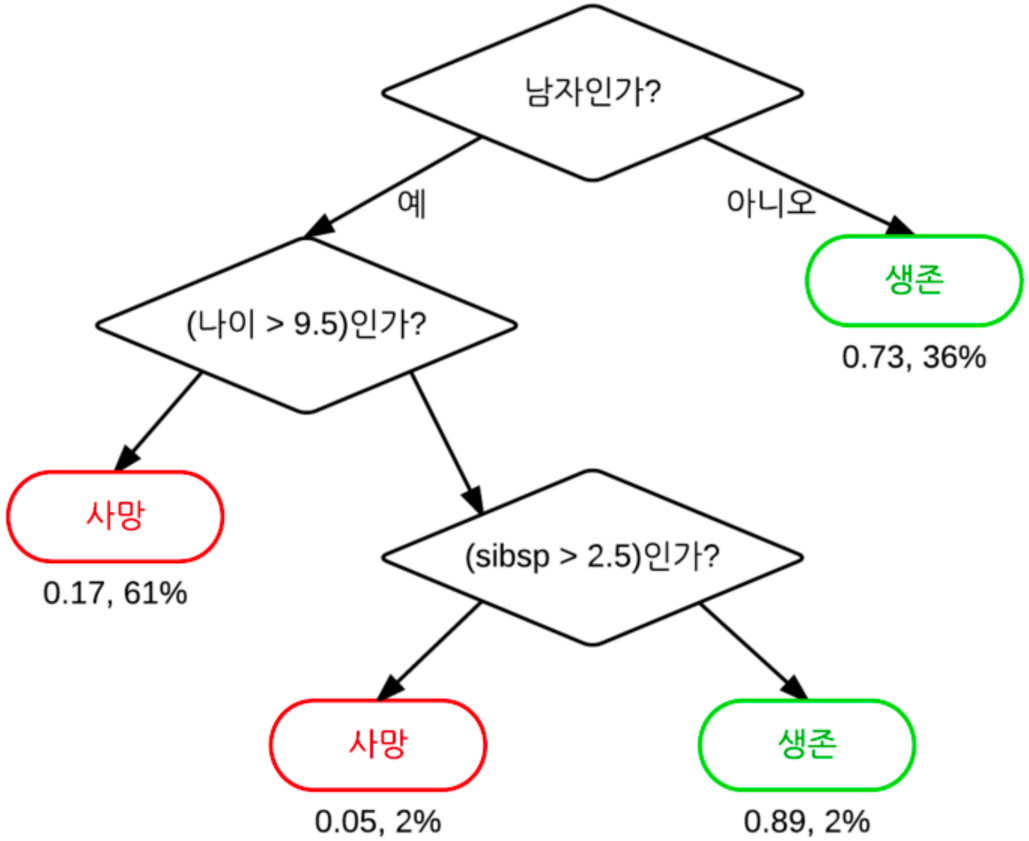
의사결정트리

- 0을 제외한 정수 i 가 양수인지 음수인지 판단하는 방법은?
- IF - then - else 구문을 활용
- If $i > 0$ then $i = positive$ else $i = negative$

의사결정트리

- 0을 제외한 정수 i 가 3과 10 사이에 있는지 아닌지를 판단하는 방법은?
- IF - then - else 구문을 두번 활용
- If $i < 3$ then False else (if $i > 10$ then True else False)
- 그렇다면 보다 복잡한 문제에서는?

의사 결정 트리

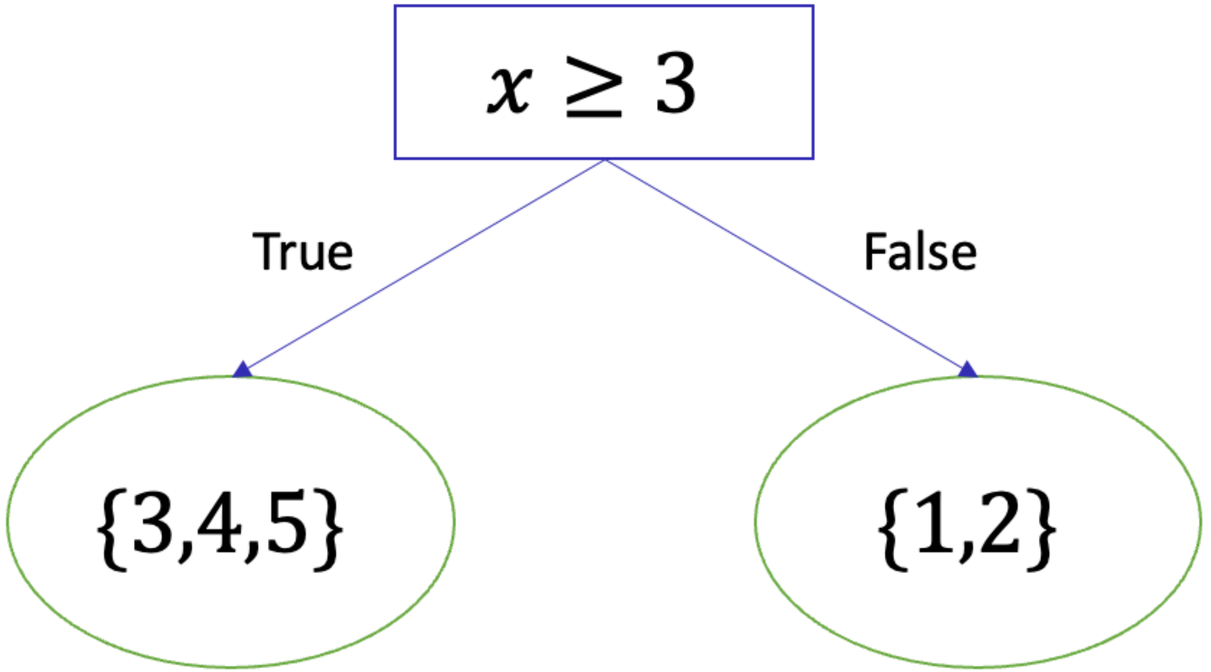


타이타닉 생존자 분류를 위한 의사 결정 트리

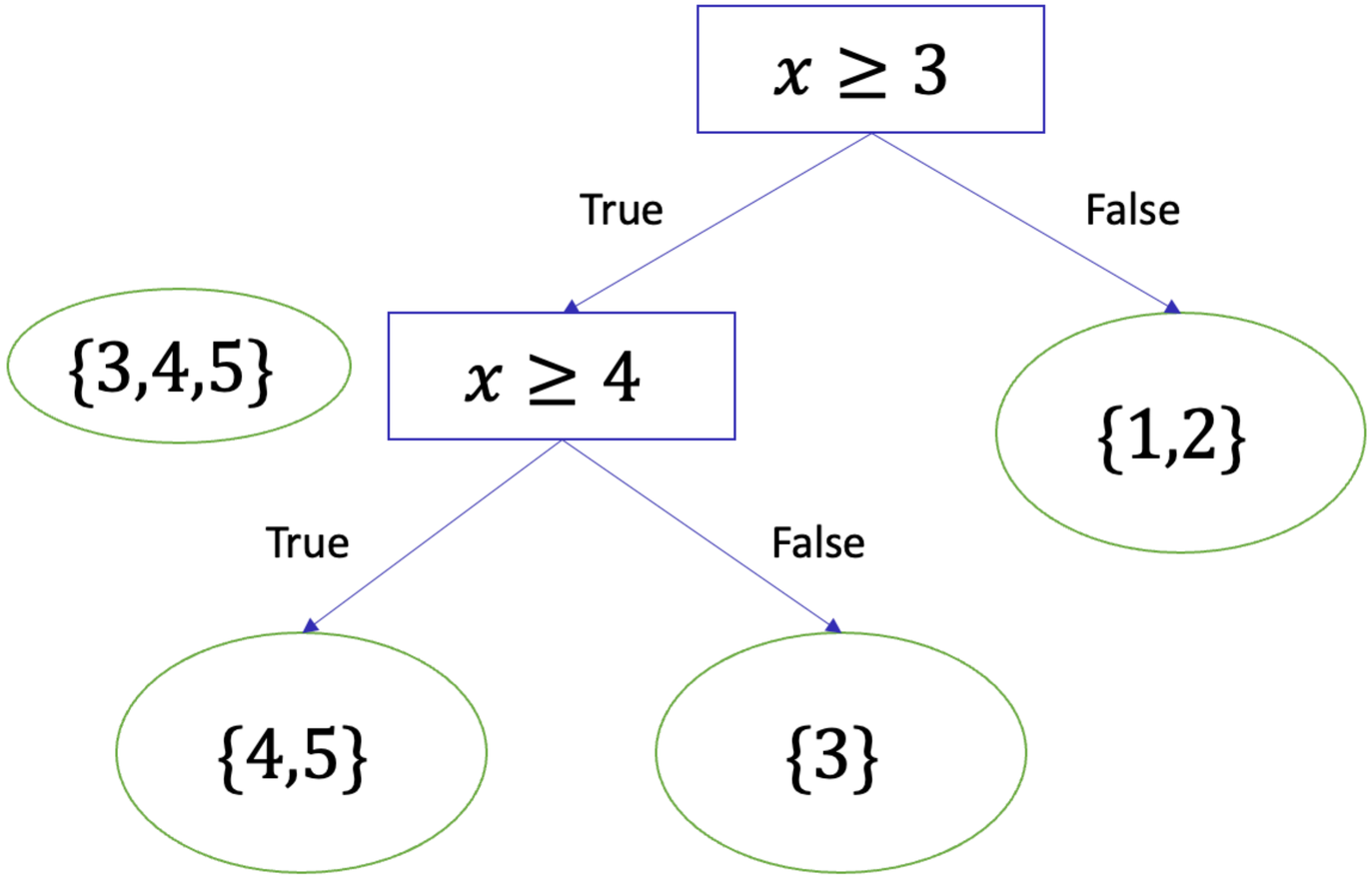
https://wooono.tistory.com/104

하나의 규칙은 데이터를 두개로 분할

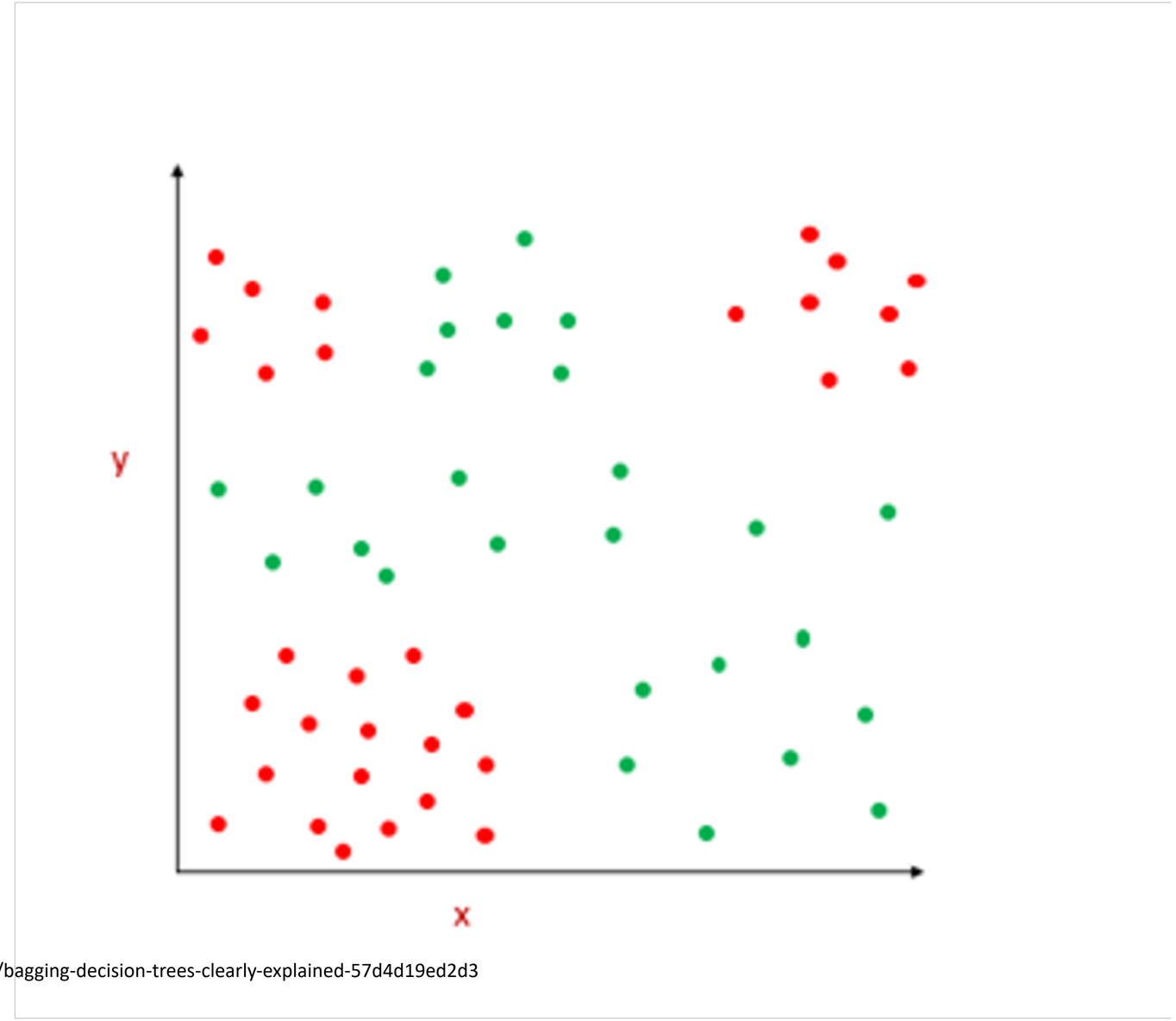
- Dataset $X = \{1,2,3,4,5\}$



하나의 규칙은 데이터를 두개로 분할

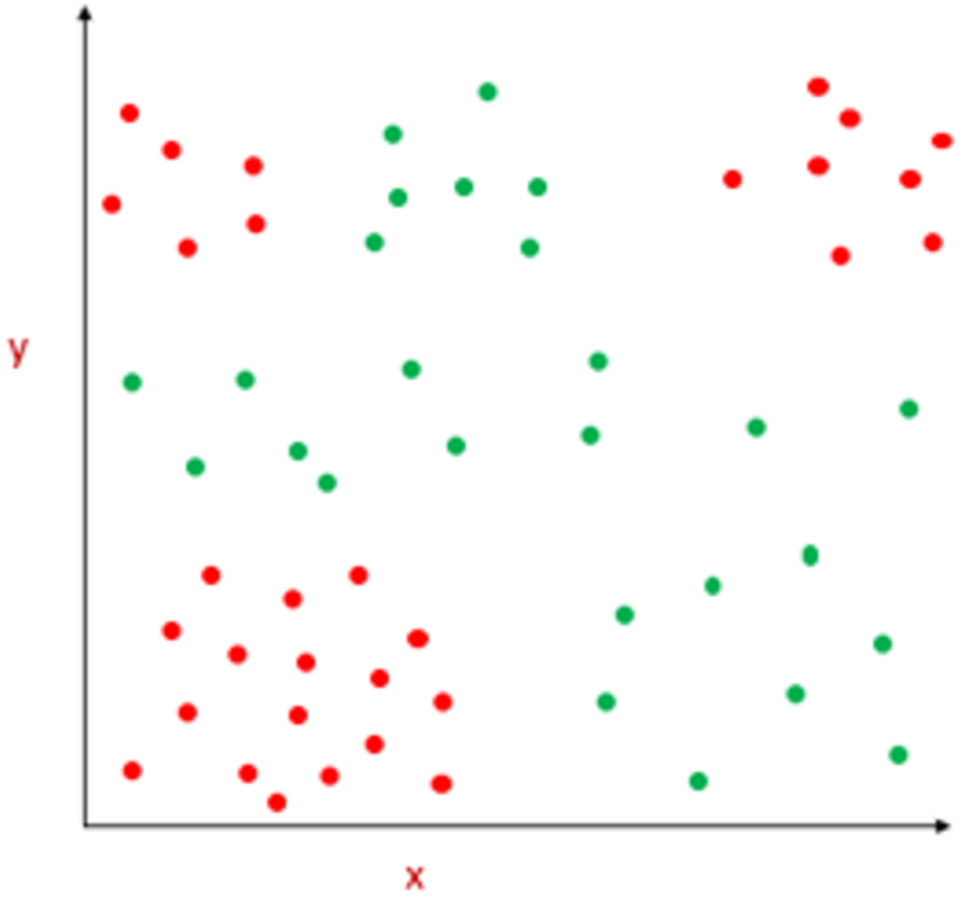


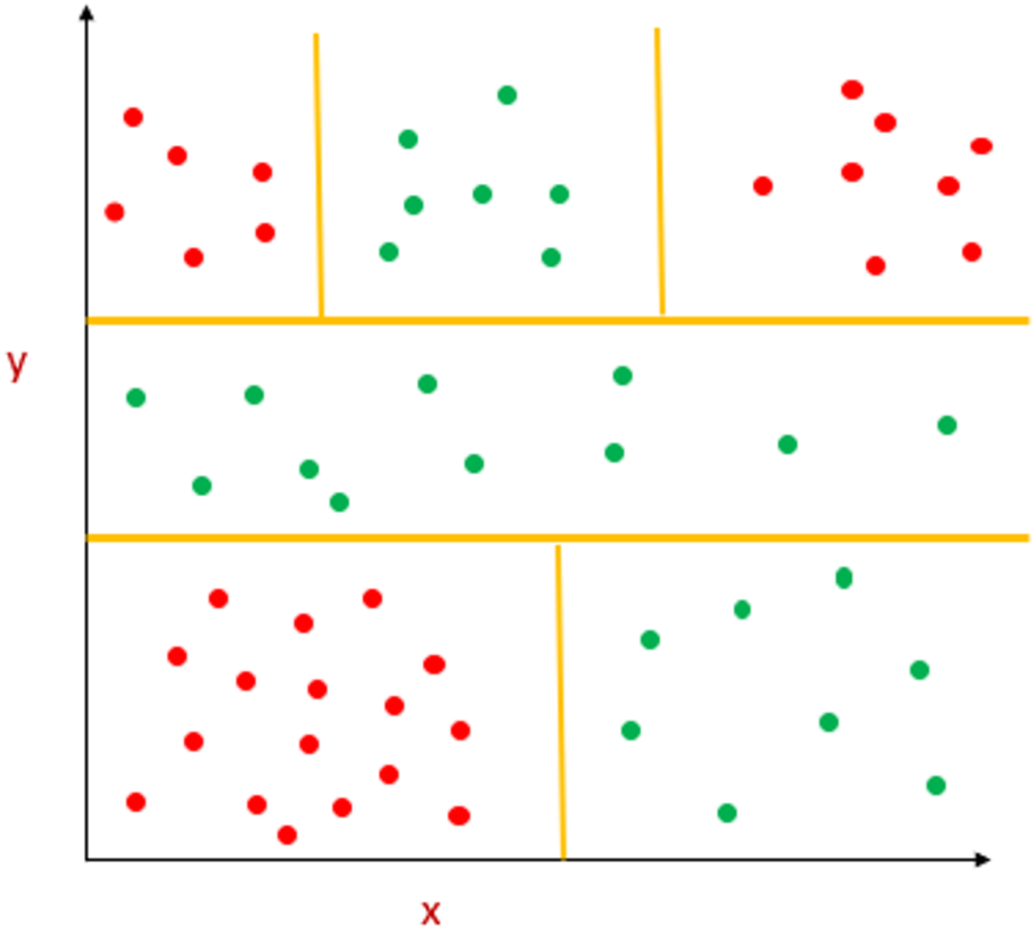
색깔에 따른 분류를 수행하는 방법



https://medium.com/towards-data-science/bagging-decision-trees-clearly-explained-57d4d19ed2d3

색깔에 따른 분류를 수행하는 방법





https://medium.com/towards-data-science/bagging-decision-trees-clearly-explained-57d4d19ed2d3

규칙을 결정하는 방법은?

- Expert system (전문가 시스템): 인간 전문가의 규칙을 차용
- Decision tree (의사결정트리): 데이터로 부터 규칙을 학습
- 이러한 규칙은 “Information gain”을 최대화 하는 규칙으로 설정

정보 엔트로피

- 정보 엔트로피(Information Entropy)는
  주어진 데이터셋의 예측 불확실성 또는 **무작위성(randomness)**을
  정량적으로 측정하는 지표이다.

- 엔트로피가 높다
  → 클래스가 균등하게 섞여 있음 → 예측하기 어려움 → 불확실성 큼

- 엔트로피가 낮다
  → 특정 클래스에 데이터가 몰려 있음 → 예측이 쉬움 → 불확실성 작음

정보 엔트로피

- 랜덤변수 $X \sim P$, 정보엔트로피 $H(X)$는 다음과 같이 정의됨.

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

- 여기서 x_i는 샘플이 아닌 클래스를 의미 (i.e., n개의 클래스)

엔트로피 계산 예시

- 엔트로피 계산 예시 (log base = 2)
- x_1 클래스일 확률과 x_2 클래스일 확률이 같을 때
  - $P(x_1) = P(x_2) = 0.5 \rightarrow -0.5log(0.5) - 0.5log(0.5) = 1$
- x_1 클래스일 확률이 1, x_2 클래스일 확률이 0일 때
  - $-log(1) = 0$

정보 이득

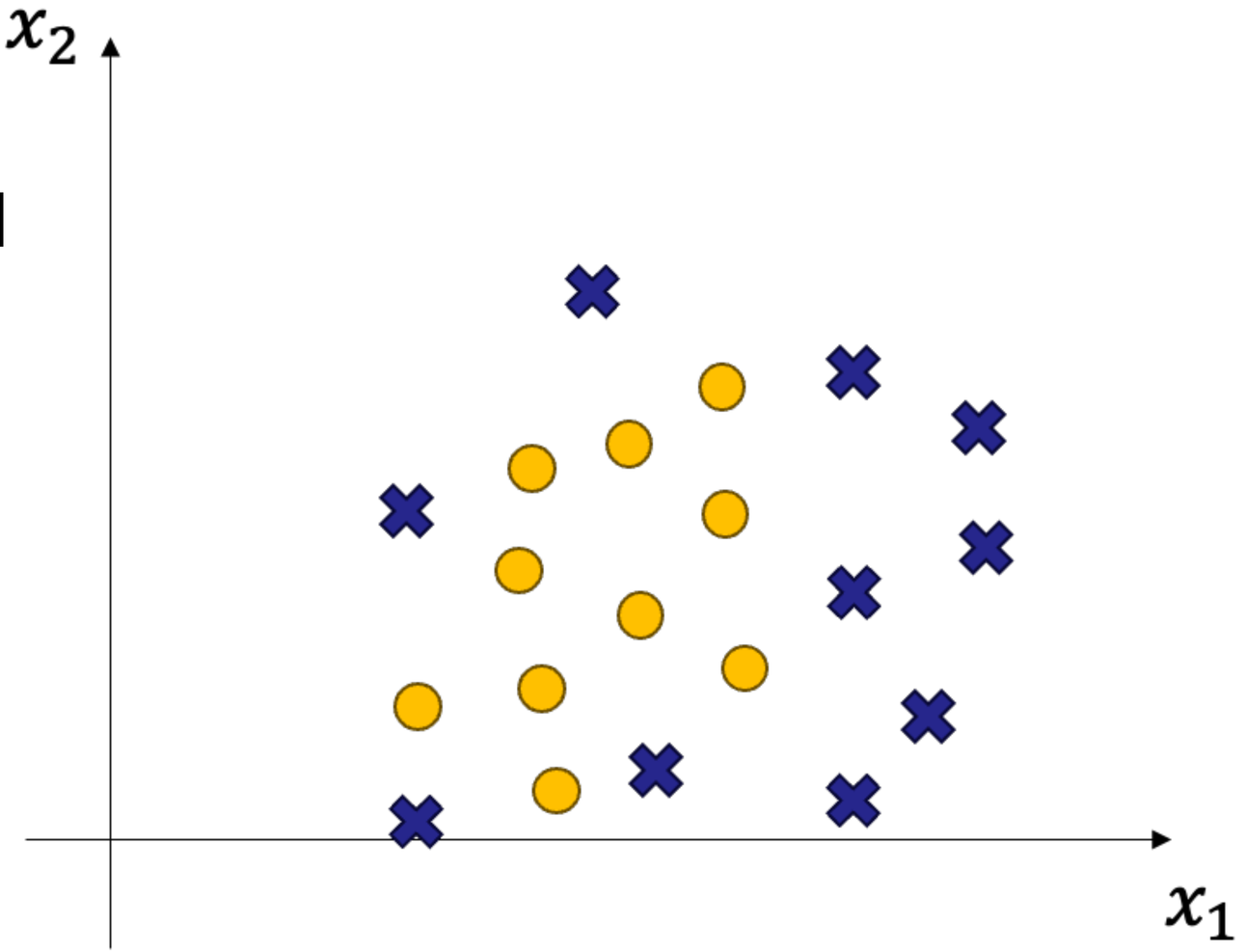
- 정보이득은 어떤 규칙으로 인해 분할되기 전과 후의 정보엔트로피의 차이를 의미

$$IG = H(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} H(S_i)$$

- $H(S)$: 부모 노드의 엔트로피 (규칙에 의한 분할 전)
- $H(S_i)$: i번째 자식 노드의 엔트로피 (규칙에 의한 분할 후 하나의 서브셋)
- $|S|, |S_i|$: 부모 노드 또는 i번째 자식 노드의 데이터 수

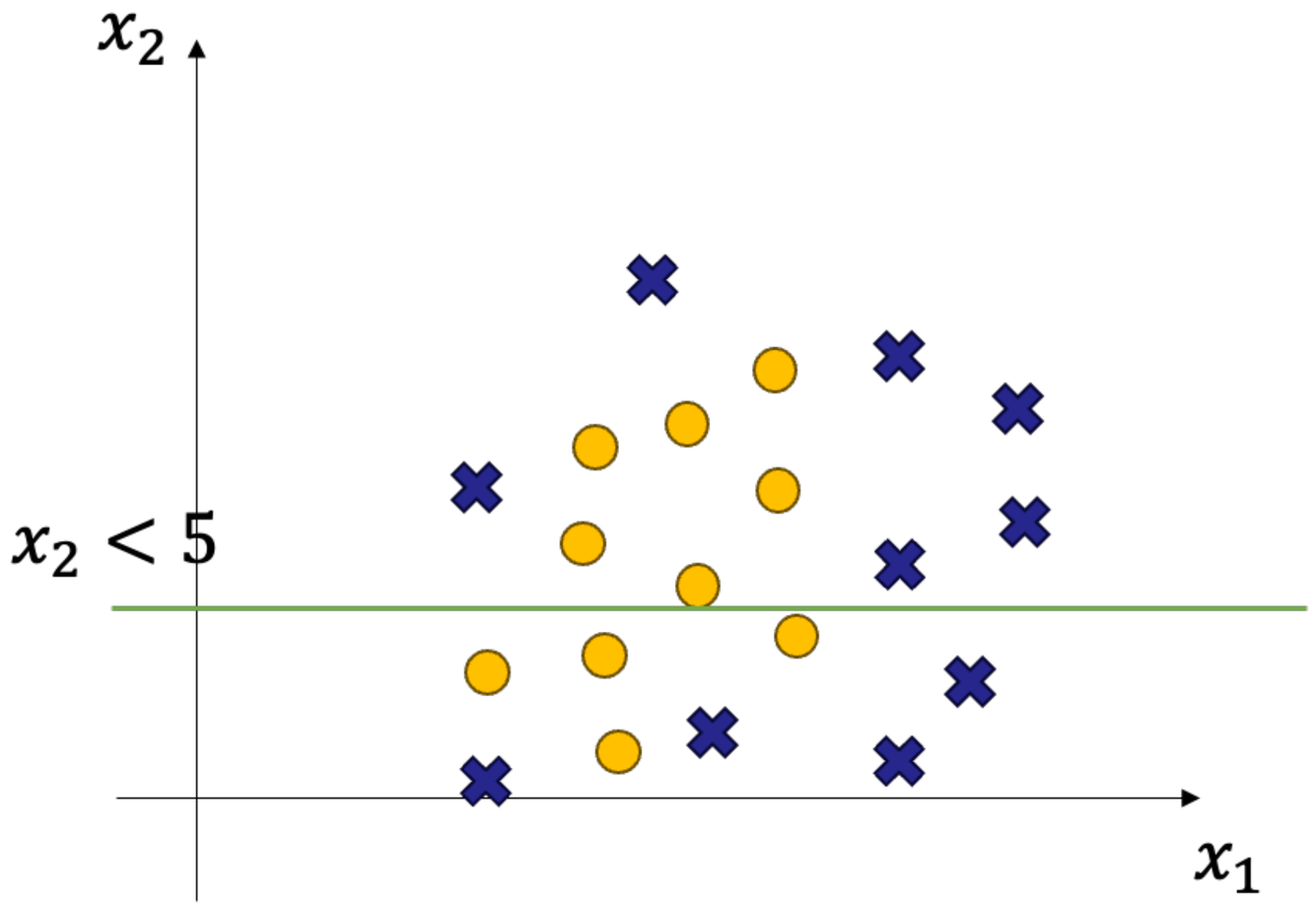
규칙에 따른 정보이득 계산

- 다음과 같이 데이터가 주어져있을때
- 분할전 엔트로피는?
- $P(X) = P(O) = 0.5$
- $H(S) = 1$



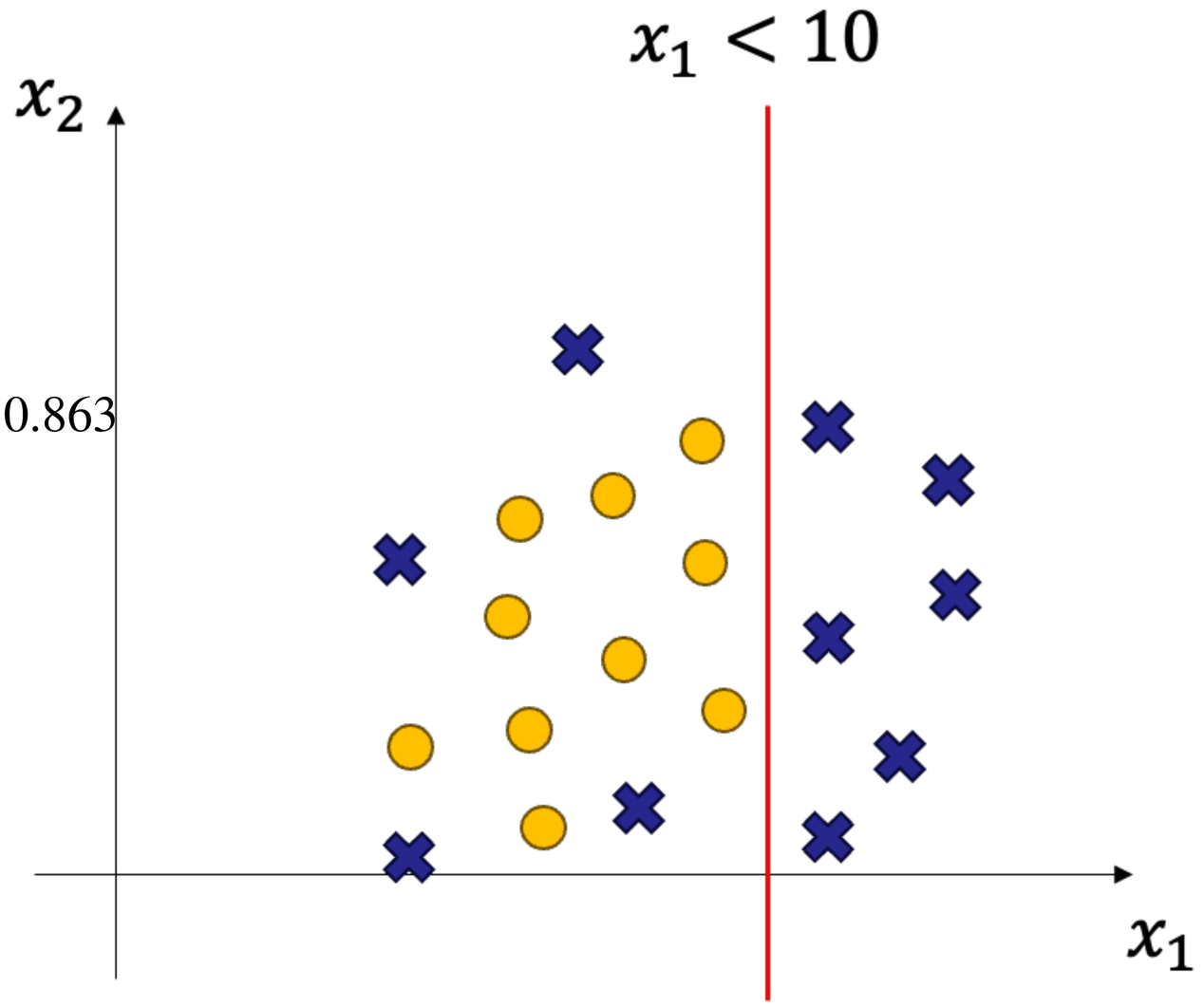
규칙에 따른 정보이득 계산

- $x_2 < 5$ 규칙에 따라 분할하는 경우
- S_1 : 분할 규칙 상단
  - $P(O) = 0.5, P(X) = 0.5$
  - $H(S_1) = 1$
- S_2 : 분할 규칙 하단
  - $P(O) = 0.5, P(X) = 0.5$
  - $H(S_2) = 1$
- $IG = 1 - 8/20 - 12/20 = 0$



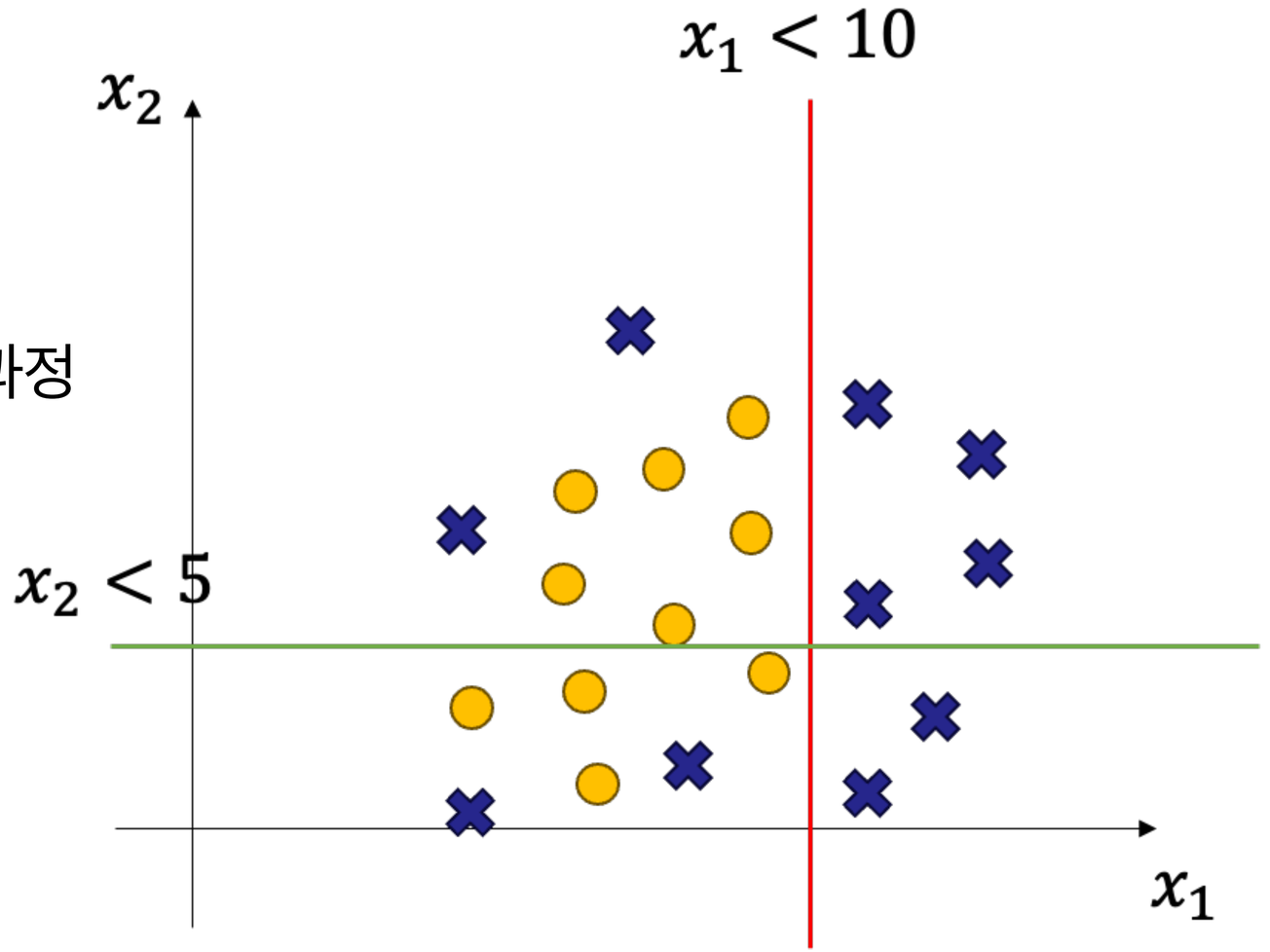
규칙에 따른 정보이득 계산

- $x_1 < 10$ 규칙에 따라 분할하는 경우
- S_1 : 분할 규칙 왼쪽
  - $P(O) = 10/14, P(X) = 4/14$
  - $H(S_1) = -P(O)log_2P(O) - P(X)log_2P(X) = 0.863$
- S_2 : 분할 규칙 오른쪽
  - $P(O) = 0, P(X) = 1$
  - $H(S_2) = 0$
- $IG = 1 - \frac{14}{20} \cdot 0.863 - \frac{6}{20} \cdot 0 = 0.396$



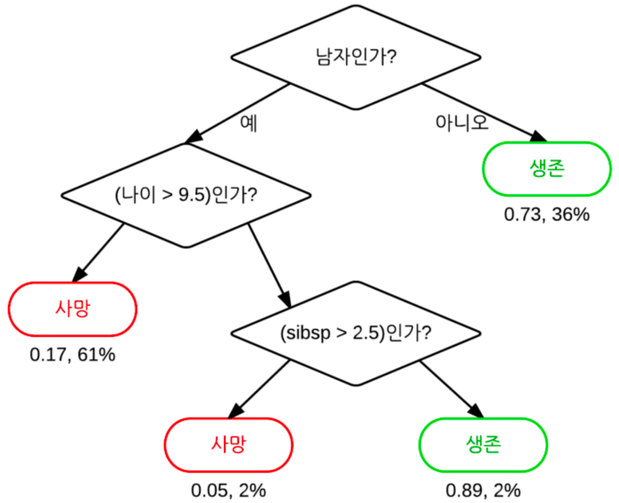
규칙에 따른 정보이득 계산

- $x_1 < 10, x_2 < 5$
- 정보이득이 높은 규칙으로 분할
- 결정트리의 학습은 규칙을 확장하는 과정



의사 결정 트리

- 일련의 분류 규칙을 통해 데이터를 구분
- 엔트로피(또는 불순도)가 낮아지는 방향으로 노드를 확장 → 규칙 생성
  - 높은 엔트로피 = 클래스가 섞여있다 = 데이터가 구분이 어려움
- 여러 개의 규칙 중 Information gain을 통해 규칙 결정.
  - Information gain이란? 부모 노드의 엔트로피 - 자녀 노드의 엔트로피
- 이후 overfitting 된 tree에 대한 pruning (가지치기)
- 따라서 의사 결정 트리에서의 학습이란 트리를 확장시켜 나가는 개념으로 볼 수 있음.



의사결정트리

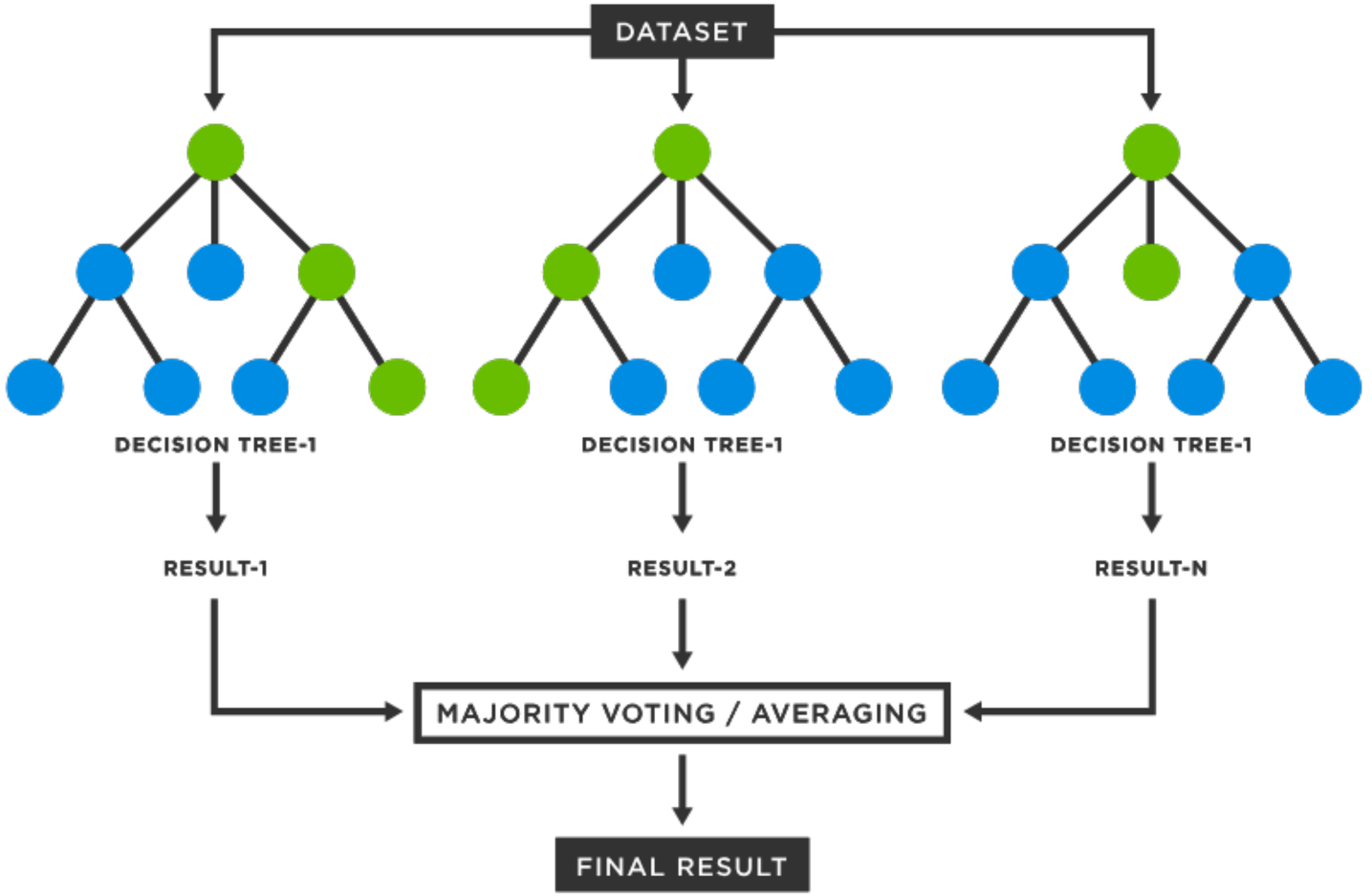
앙상블 모델

- 여러 모델을 결합하여 성능 향상을 이끌어 내는 방법
- 간단한 다수의 모델 >= 복잡한 단일 모델 → 집단 지성의 힘?
- 가장 단순하게는 여러 모델 학습 후 결과의 평균값을 활용 가능
- e.g., 사람의 키 예측하기

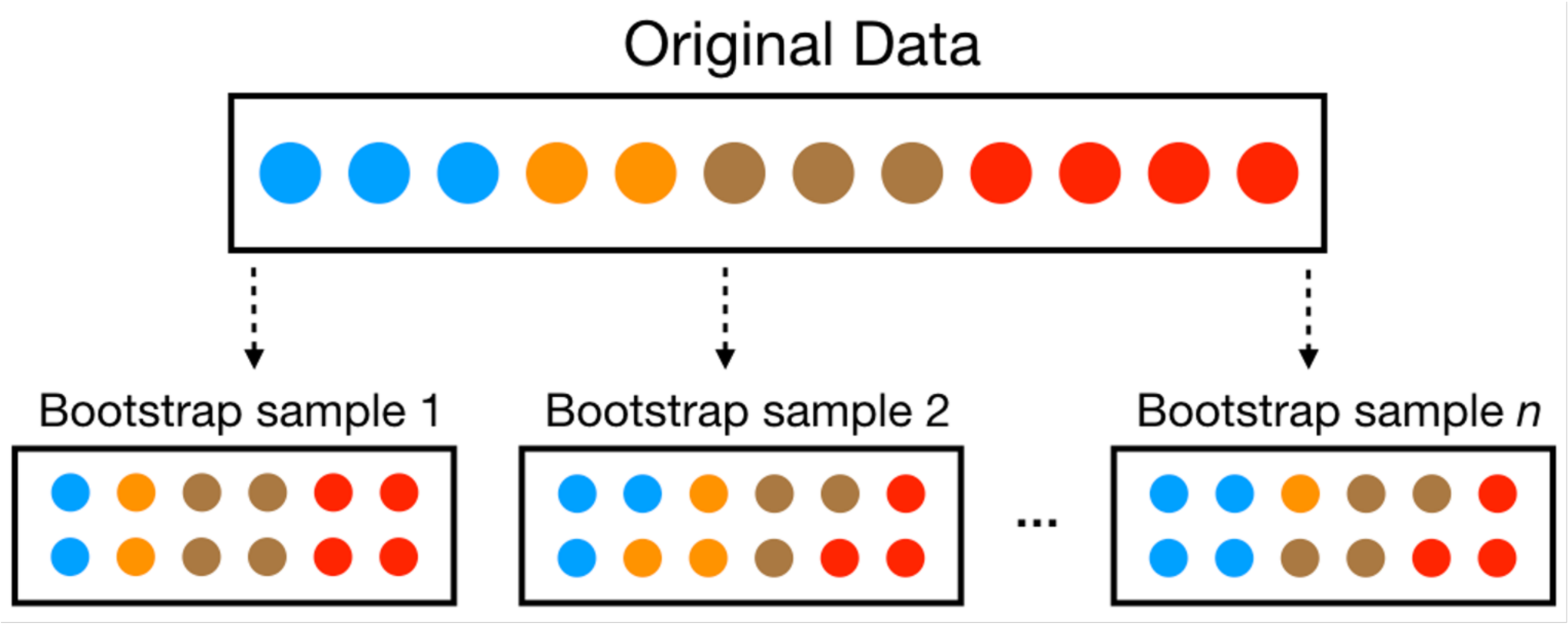
앙상블 모델

- Bagging(Bootstrap aggregating)
  - 랜덤 포레스트
- Boosting
  - Adaboost, XGBoost, LGBM, etc

랜덤 포레스트

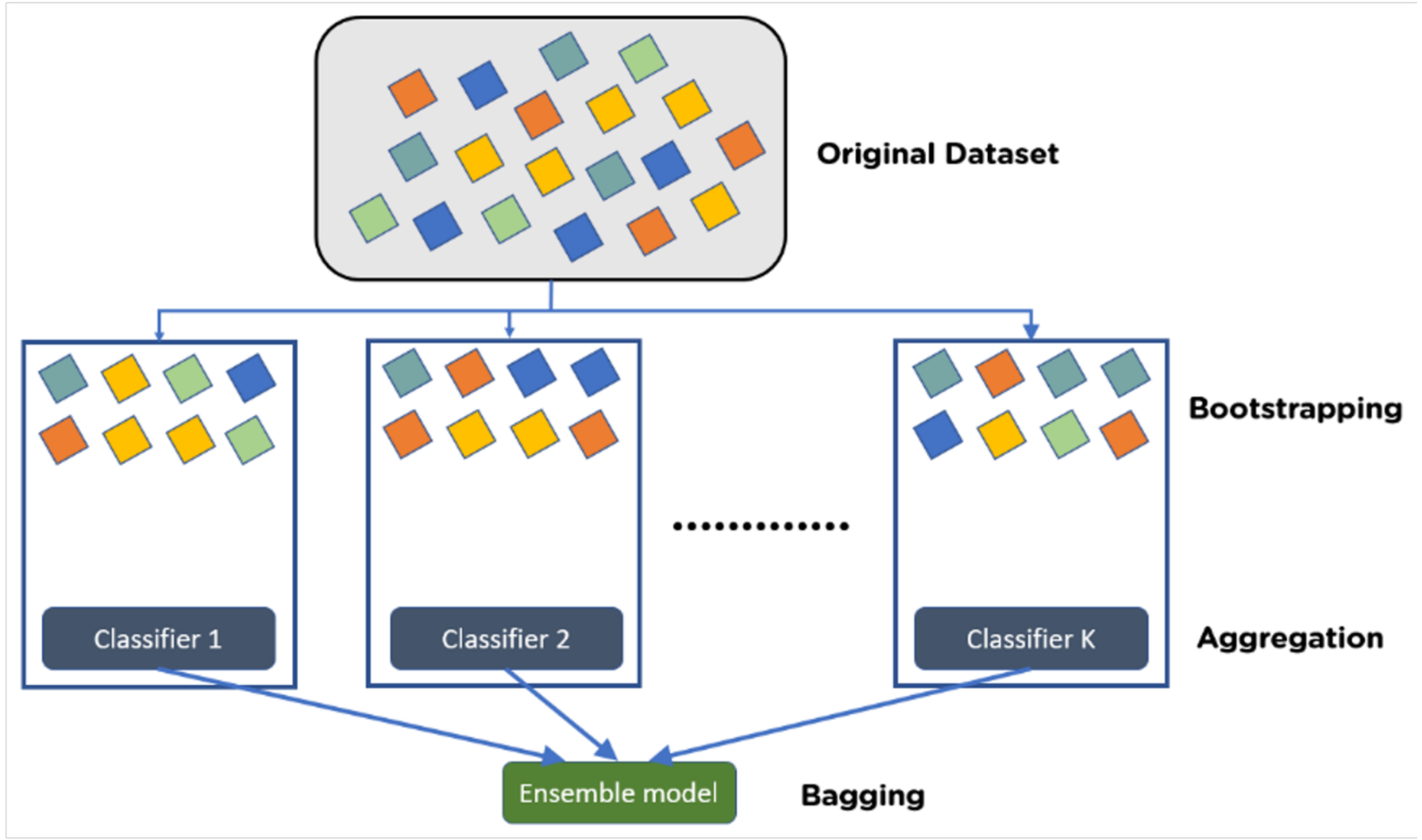

DATASET
DECISION TREE-1
DECISION TREE-1
DECISION TREE-1
RESULT-1
RESULT-2
RESULT-N
MAJORITY VOTING / AVERAGING
FINAL RESULT

Bootstrap sampling



https://datasciencedojo.com/blog/bootstrap-sampling/

Bagging

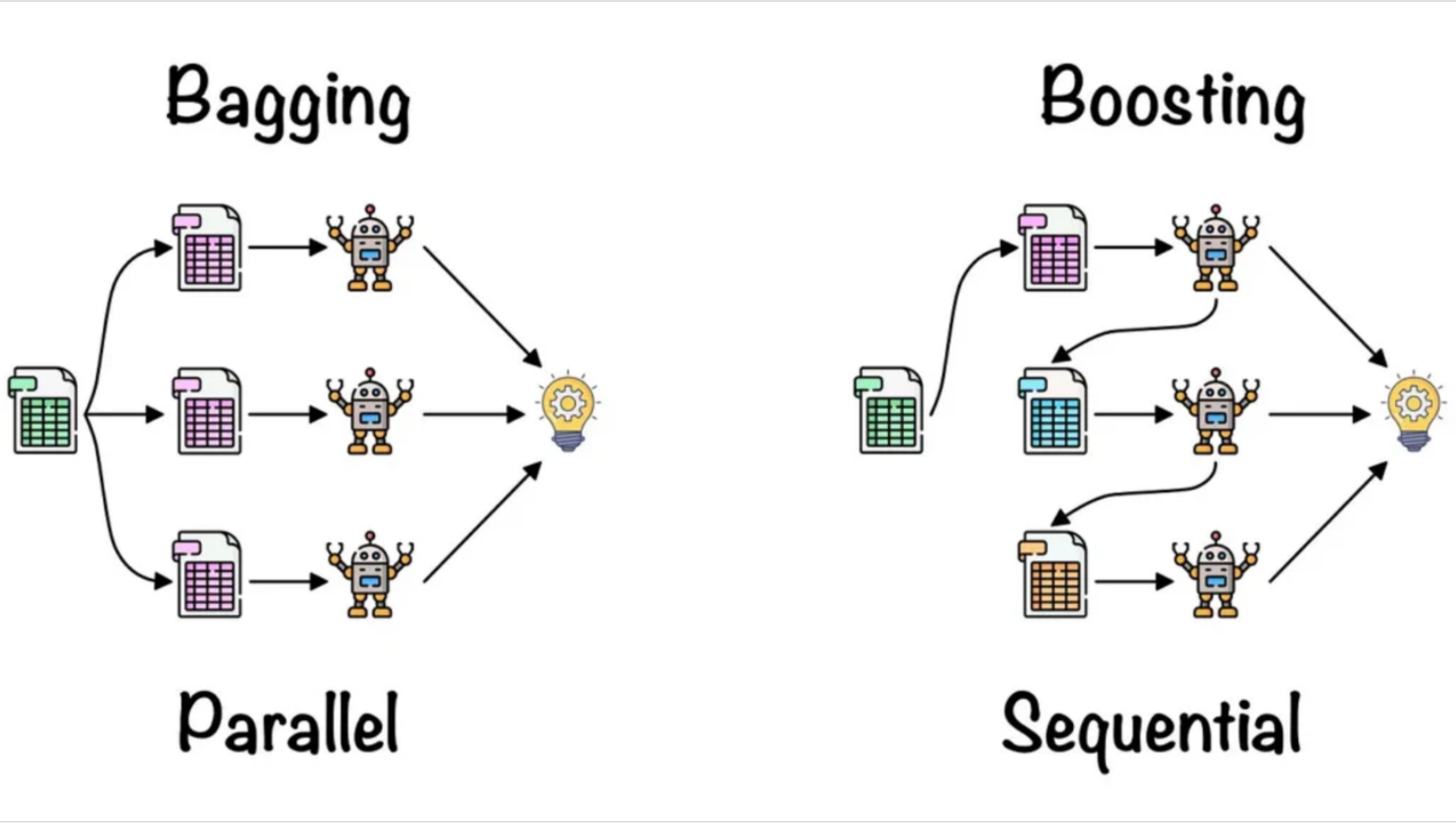


https://github.com/harjeet-blue/Ensemble-Learning-Bagging-and-Boosting?tab=readme-ov-file

랜덤 포레스트

- 의사 결정 트리 모델들의 앙상블
- Bagging (Bootstrap Aggregating) :
  부트스트랩 샘플링한 데이터로 여러 모델 학습
  - 부트스트랩 샘플링 : 데이터를 중복 샘플링하여 새로운 데이터셋 생성
- Random feature selection :
  노드 확장시 모든 변수들이 아닌 랜덤하게 선택된 변수들을 바탕으로 규칙 생성
- 각 모델들은 병렬적으로 학습이 가능함.

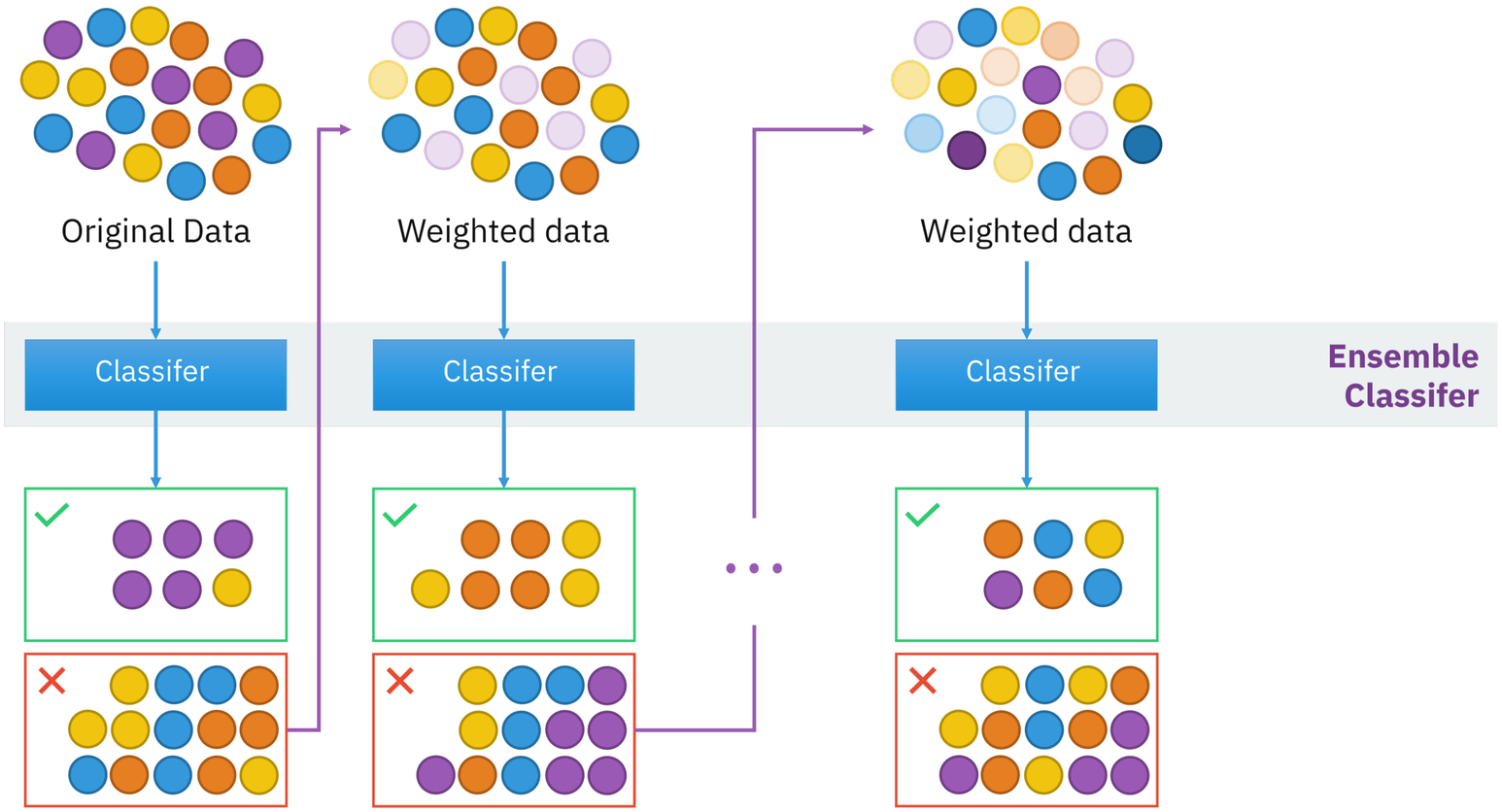
Bagging and Boosting



https://www.appliedaicourse.com/blog/bagging-and-boosting-in-machine-learning/

Boosting

- Boosting 기법
- 분류기들을 순차적으로 학습
- 이전에 잘못했던 부분을 더 잘 할 수 있도록 보완



AdaBoost

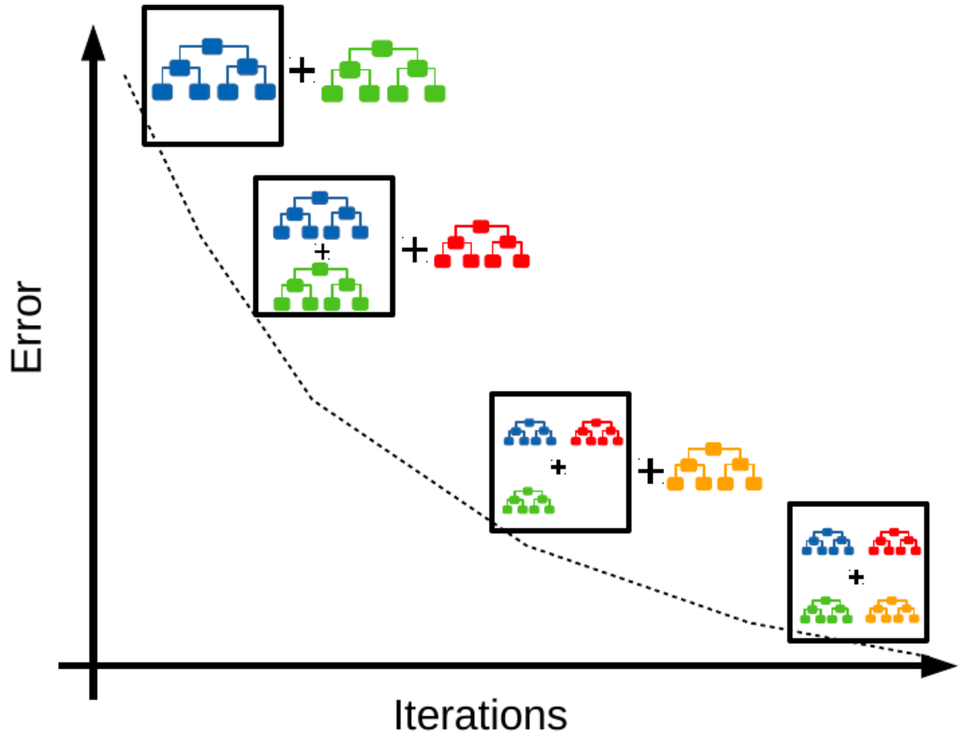
- $F_T(x) = F_{T-1} + \alpha_T f_T(x)$
- F: 부스팅 모델
- $f(x)$: 약한 분류기
- Exponential loss를 최소화하도록 학습
- Loss 계산시 이전 모델이 잘 예측하지 못한 샘플에 더 큰 가중치를 부여

Gradient boosting

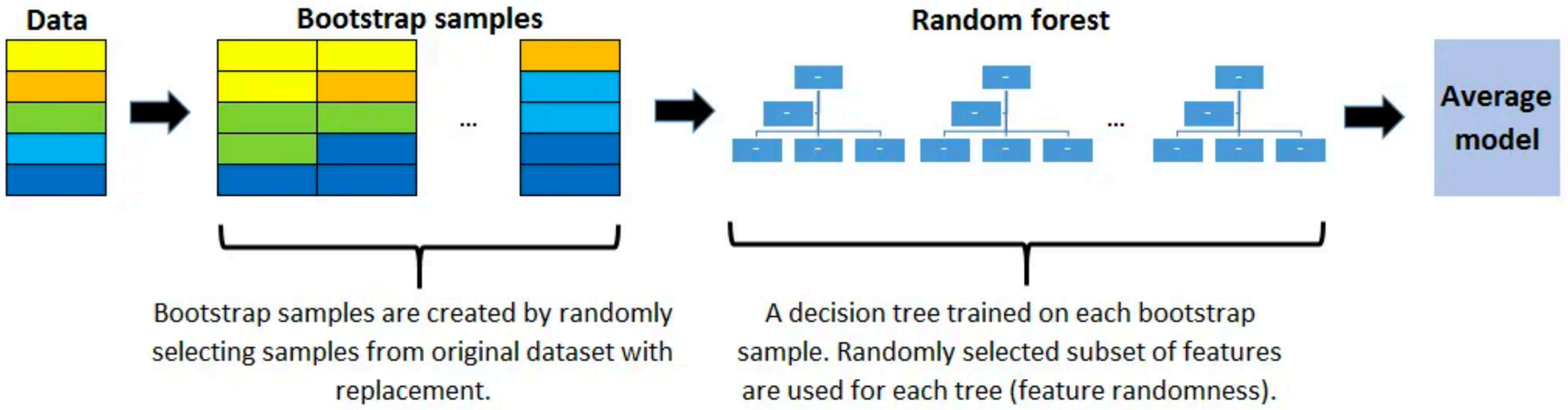
- Ada boost: $F_t(x) = F_{t-1} + \alpha_t f_t(x)$
- Gradient descent: $W_t = W_{t-1} - \alpha \nabla L(W)$
- Gradient boosting: $F_t(x) = F_{t-1}(x) - \gamma_t \nabla L(y, F_{t-1}(x))$

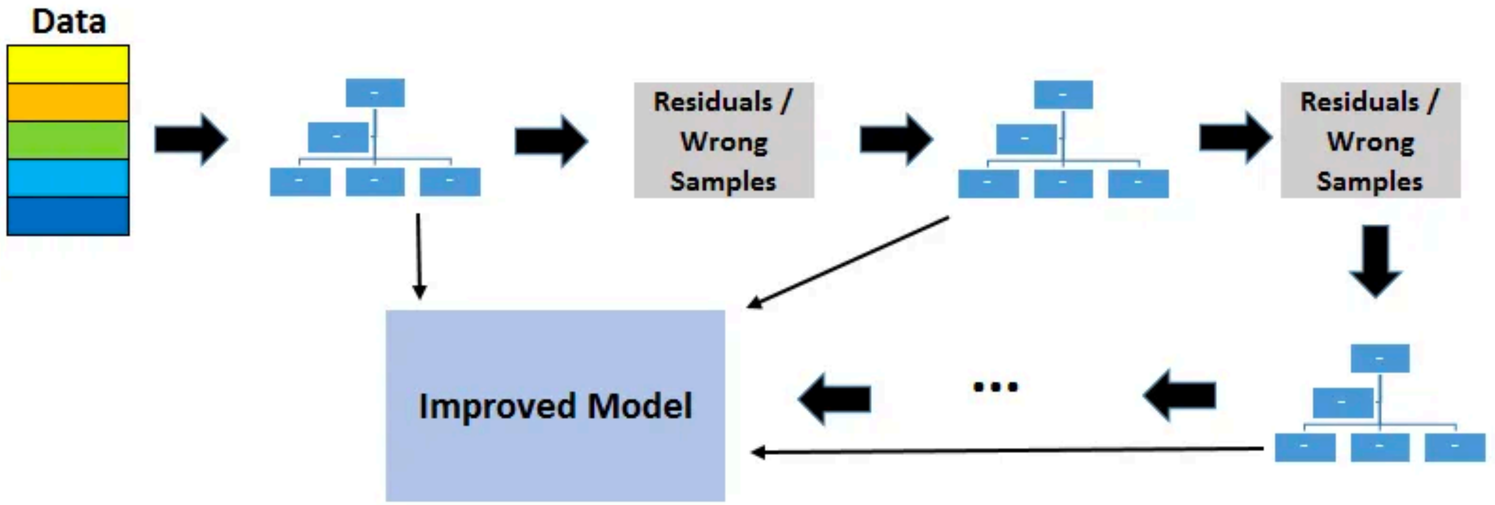
Gradient Boosted Decision Tree

- Gradient : Residual error를 줄이는 방향
- Boosting : 다수의 weak learner (model)
  → 하나의 strong learner (model)
- Decision Tree : 결정트리 모델을 활용
- XGBoost, LightGBM, CatBoost

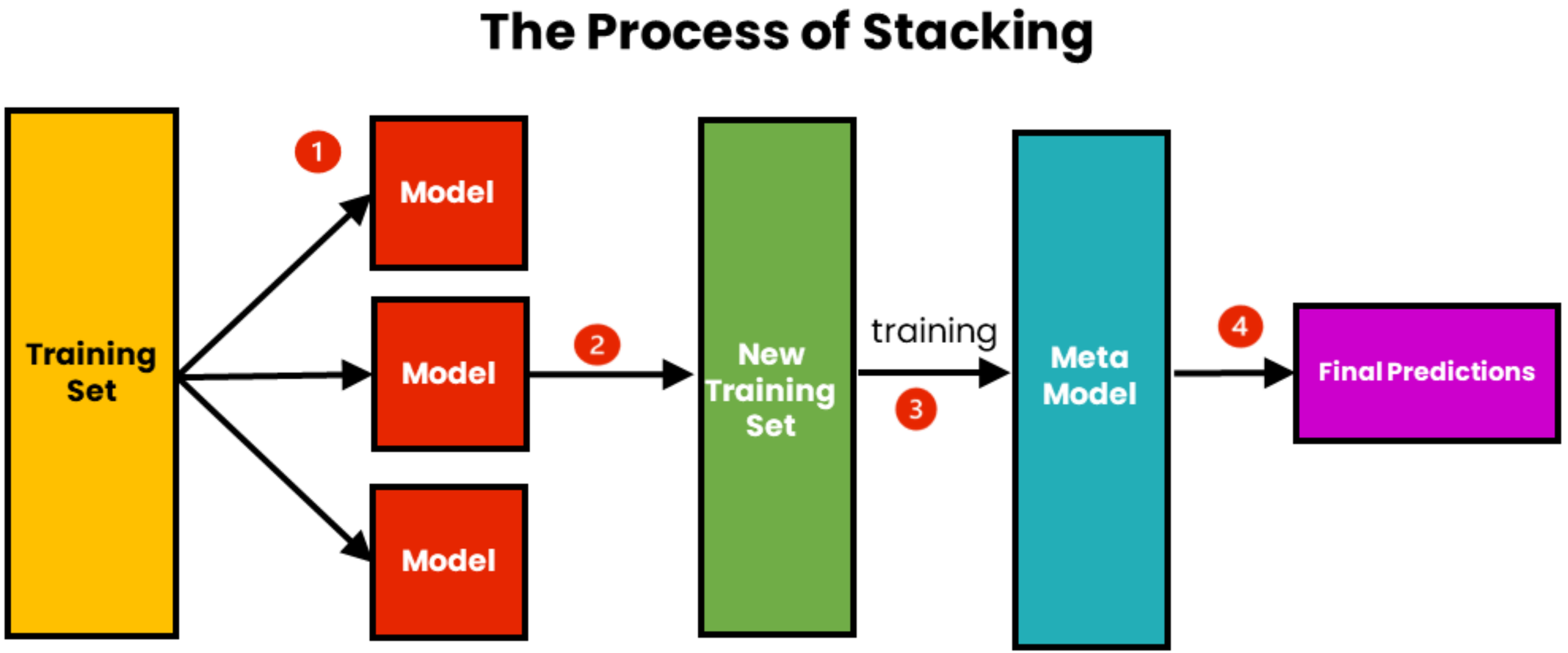


RF vs GBDT

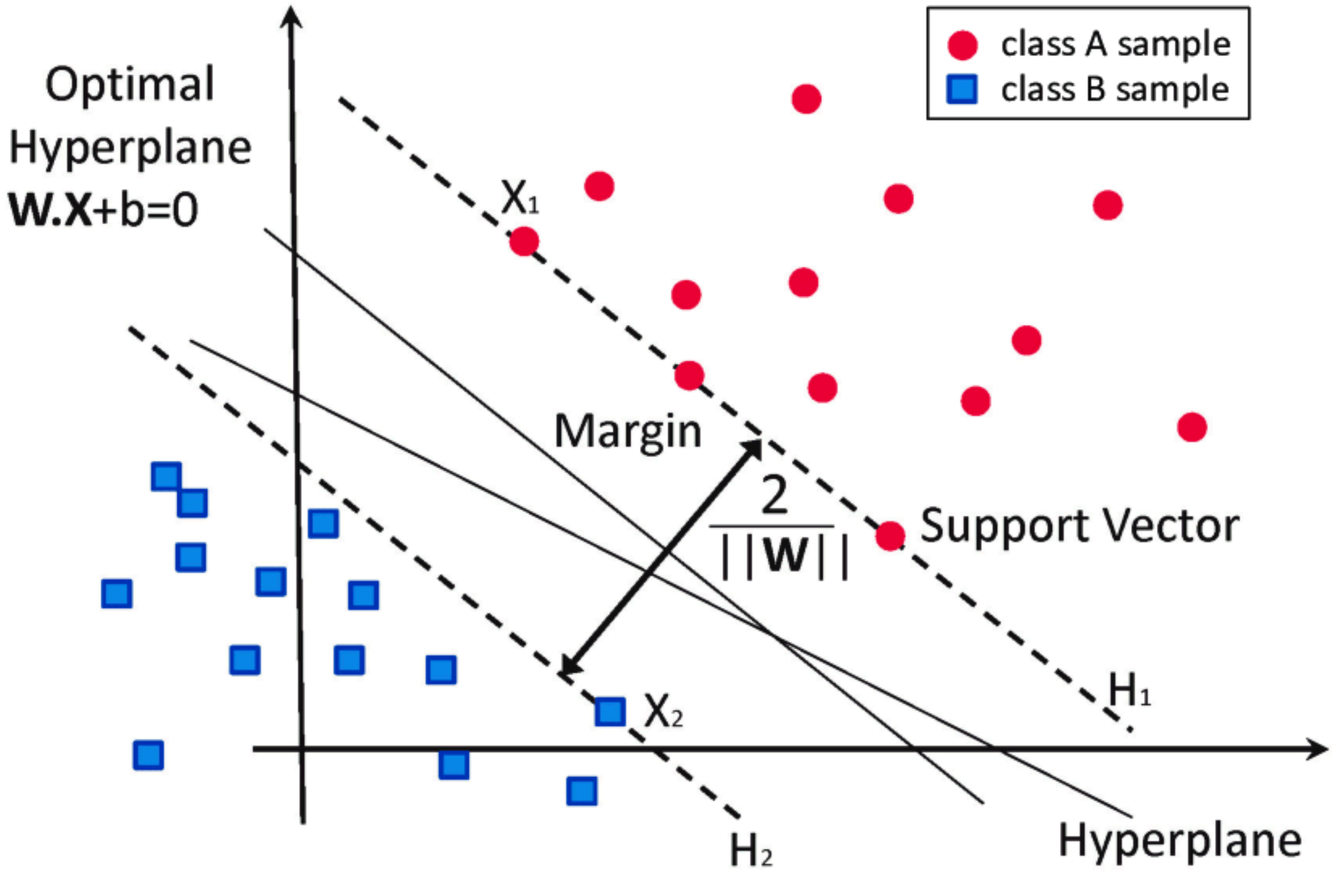

Data
Bootstrap samples
Random forest
Average model
Bootstrap samples are created by randomly selecting samples from original dataset with replacement.
A decision tree trained on each bootstrap sample. Randomly selected subset of features are used for each tree (feature randomness).


Data
Residuals / Wrong Samples
Residuals / Wrong Samples
Improved Model

Stacking


The Process of Stacking
Training Set
1
Model
Model
Model
2
New Training Set
training
3
Meta Model
4
Final Predictions

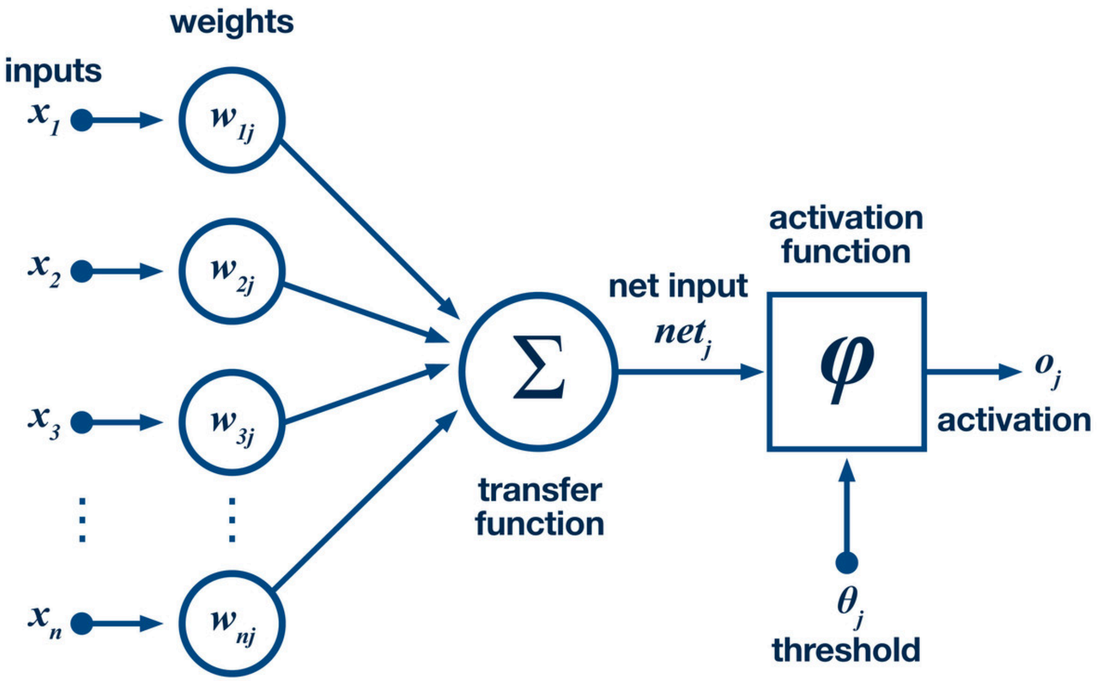
서포트 벡터 머신


class A sample
class B sample
Optimal
Hyperplane
W.X+b=0
X1
X2
Margin
2
||W||
Support Vector
H1
H2
Hyperplane

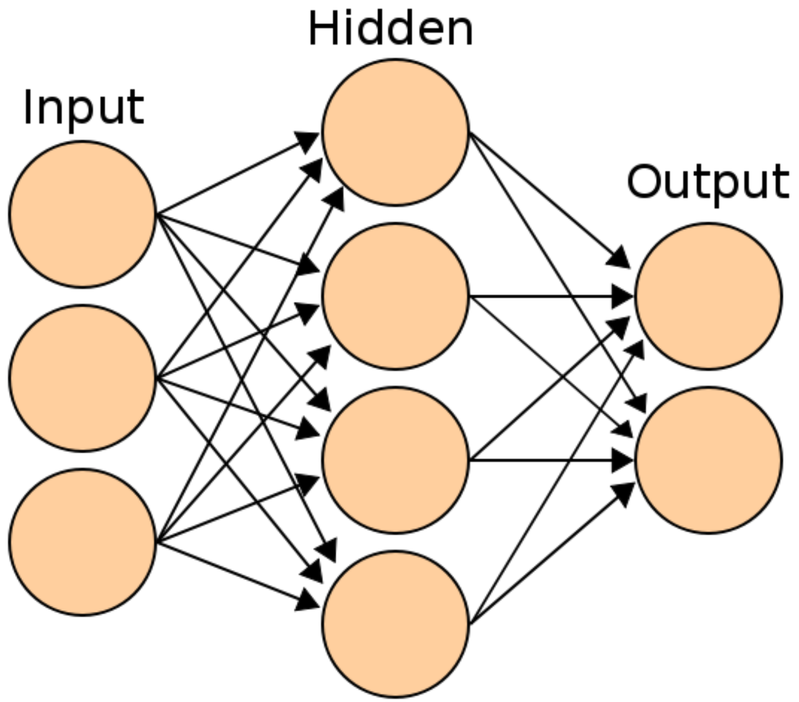
서포트 벡터 머신

- 두 범주를 최대 마진으로 분리하는 초평면 탐색
- 결정 경계와 가장 가까운 “서포트 벡터”와의 거리를 최대화
- 커널 트릭을 활용한 고차원 공간 상 선형 분리 → 저차원 공간 상 비선형 분리
  - 커널 트릭 : 고차원 공간에서의 내적을 저차원 공간 상에서 계산하는 기법

인공신경망



인공 신경



인공 신경망

인공신경망

- 인공 신경 : Vector-to-Scalar Nonlinear function
- 레이어 : Vector-to-Vector Nonlinear function (입력을 공유하는 인공신경의 집합)
- 인공신경망 : Vector-to-Vector Nonlinear function (레이어들의 집합)