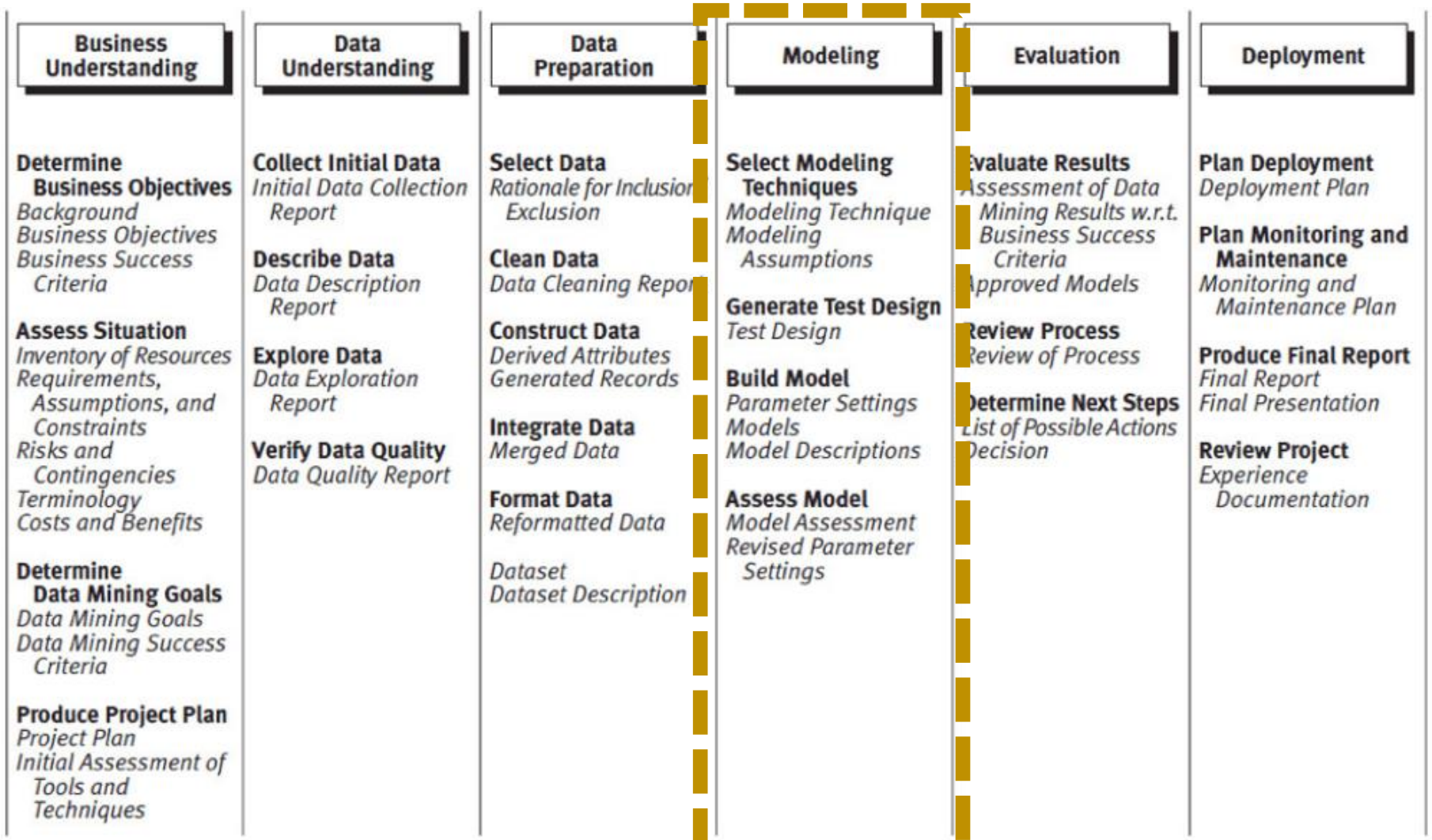
마이닝 알고리즘 1

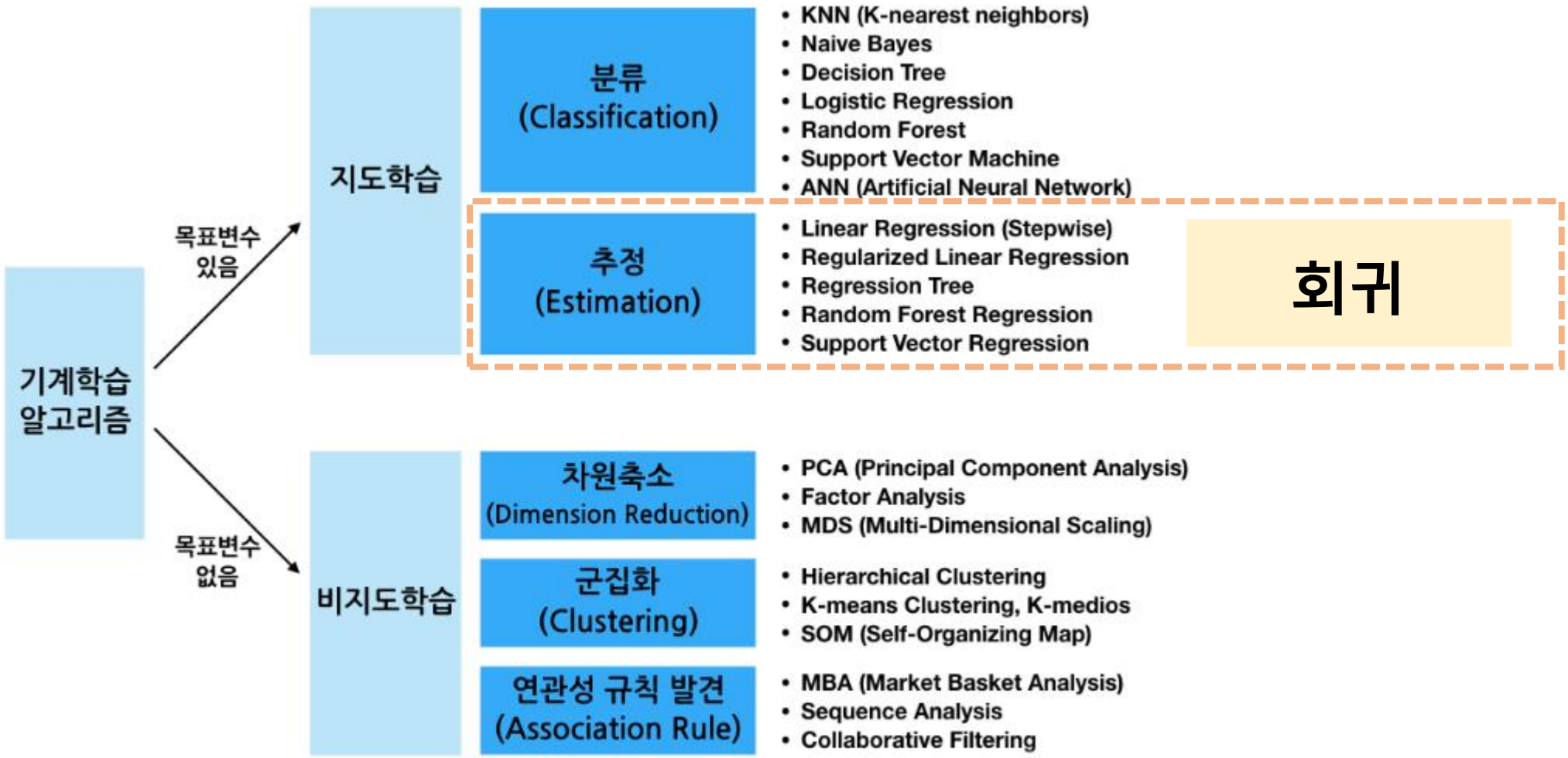
Knn, Decision Tree, Random Forest

마이닝 알고리즘 (머신러닝 모델)



CRISP-DM tasks in **bold**, and outcomes in *italic* (table from CRISP-DM Guide)

마이닝 알고리즘 (머신러닝 모델)



https://dbrang.tistory.com/1520

마이닝 알고리즘 (머신러닝 모델)

인공 신경망

- **하나의 독립된 모델**
- **머신러닝 및 인공지능 기술의 범주에 속하는 방법론과 기술**

유형	모델/기법	설명	학습 유형	모델 유형
신경망	인공 신경망 (Neural Networks)	입력 데이터로부터 복잡한 패턴을 학습하여 예측이나 분류를 수행할 수 있는 모델. 다양한 층(layer)과 노드(node)로 구성되며, 각 연결마다 가중치(weight)가 존재	지도 학습 /비지도 학습 /강화 학습	예측/묘사
	컨볼루션 신경망 (Convolutional Neural Networks, CNN)	이미지 인식, 비디오 분석, 이미지 분류 등 시각적 데이터를 처리하는데 특화된 신경망 구조. 로컬 패턴을 효과적으로 학습	지도 학습	예측
	순환 신경망(Recurrent Neural Networks, RNN)	시계열 데이터, 자연어 처리 등 순차적 데이터를 처리하는데 적합한 신경망 구조. 과거 정보를 현재의 결정에 반영		
	자기 조직화 맵(Self-Organizing Maps, SOM)	고차원 데이터를 저차원(보통 2차원)으로 매핑하여 시각화하고, 데이터의 패턴을 발견하는데 사용	비지도 학습	묘사
	생성적 적대 신경망 (Generative Adversarial Networks, GANs)	두 개의 신경망(생성자와 판별자)이 서로 경쟁하면서 학습하는 구조로, 새로운 데이터를 생성하는 데 사용		예측/묘사

마이닝 알고리즘 (머신러닝 모델)

인공 신경망

- **하나의 독립된 모델**
- **머신러닝 및 인공지능 기술의 범주에 속하는 방법론과 기술**

1.머신러닝 모델(Machine Learning Model):

데이터로부터 학습할 수 있는 능력을 가진 머신러닝 모델의 한 종류

지도 학습, 비지도 학습, 강화 학습 등 다양한 학습 방식 지원

2. 딥러닝 아키텍처(Deep Learning Architecture):

다층 퍼셉트론(Multilayer Perceptrons, MLP)이나 심층 신경망(Deep Neural Networks, DNN)과 같이 여러 층으로 구성된 신경망

3. 인공지능(AI) 기술:

인공지능 연구와 응용의 핵심 요소 중 하나로, 인공지능을 구현하는 기술

4. 계산 모델(Computational Model):

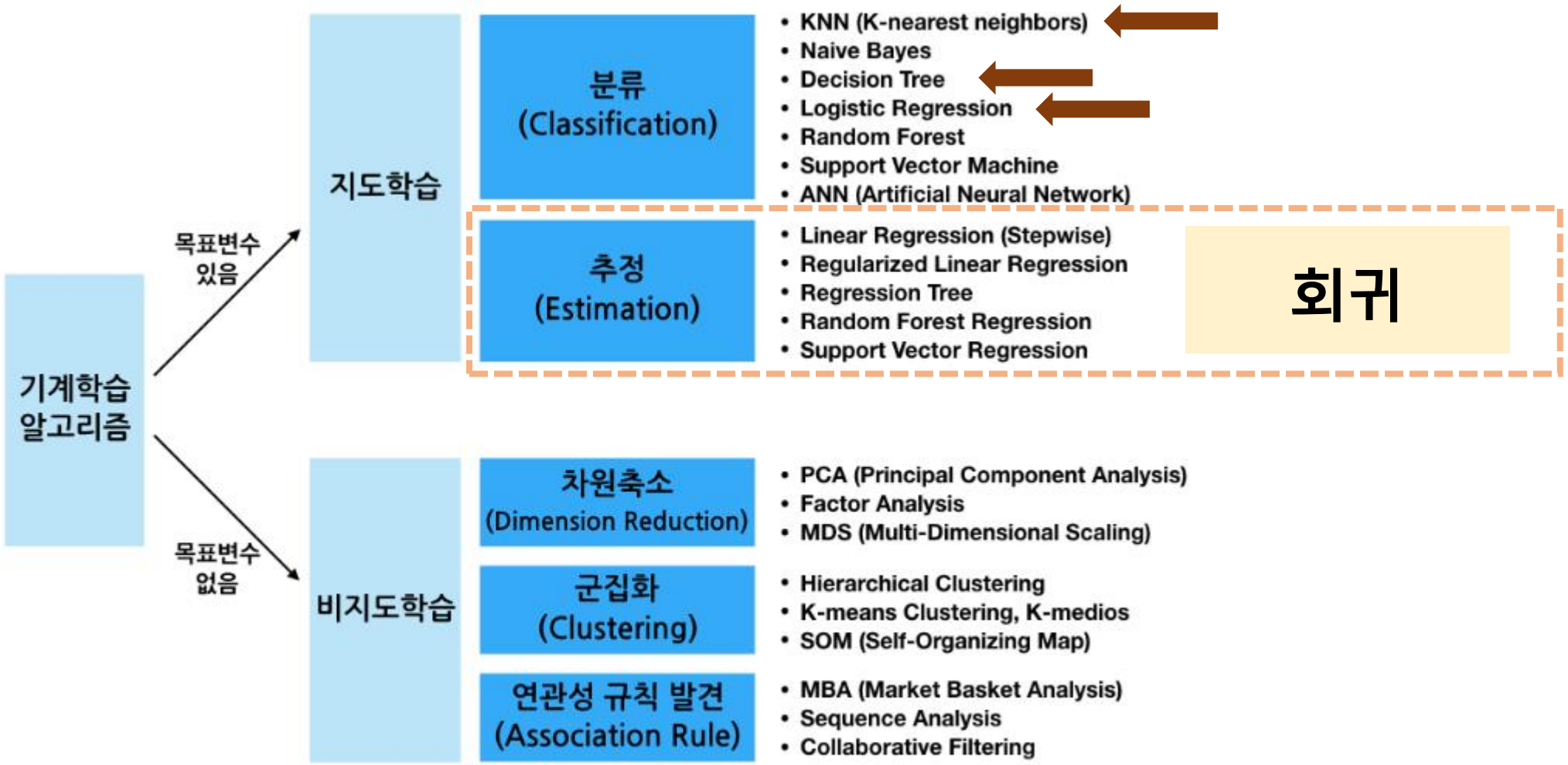
입력에서 출력으로의 매핑을 학습하는 계산 모델

데이터의 복잡한 관계의 모델링 및 예측에 사용

마이닝 알고리즘 (머신러닝 모델)

- 마이닝 알고리즘 1
  - Knn, Decision Tree, Random Forest
- 마이닝 알고리즘 2
  - Correlation, Regression, SVM
- 마이닝 알고리즘 3
  - Clustering, Association Rule, PCA
- Ensemble
- Artificial Neural Network

마이닝 알고리즘 (머신러닝 모델)



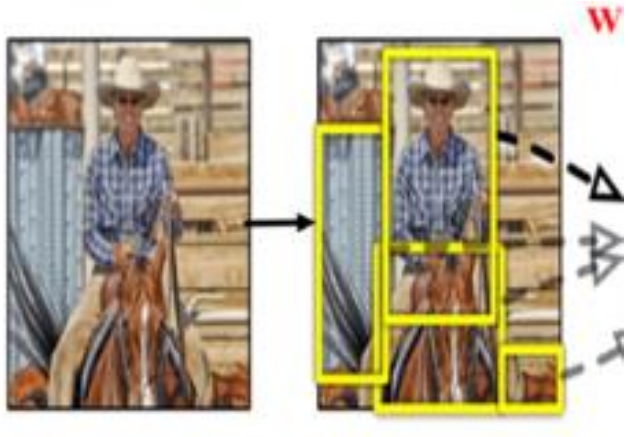
https://dbrang.tistory.com/1520

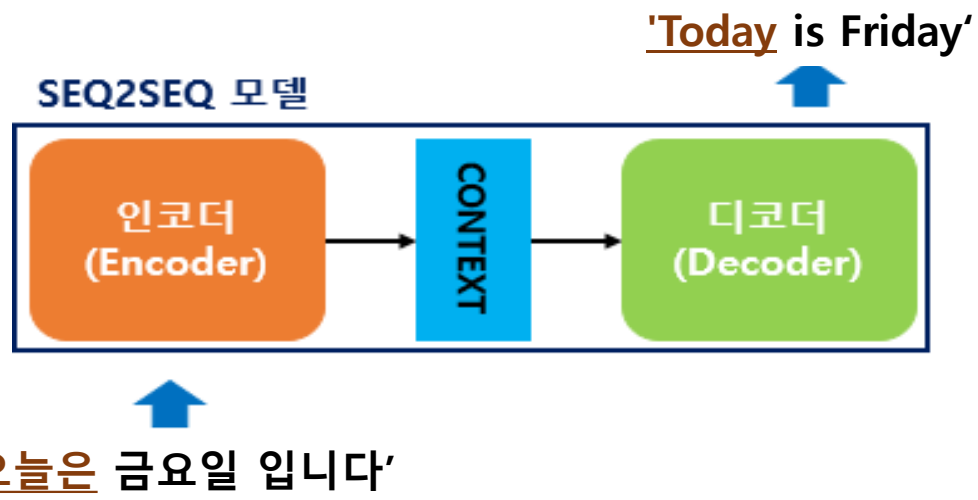
마이닝 알고리즘 (머신러닝 모델)

Supervised Learning

Artificial Neural Network Modeling

If we can train a model to **map X to Y** based on a labelled dataset then it can be used to predict ….





마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression

분류(Classification)	회귀(Regreeion)
미리 정해놓은 class label 중 하나를 예측	어떤 숫자 값을 예측
객관식 문제	주관식 문제

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression

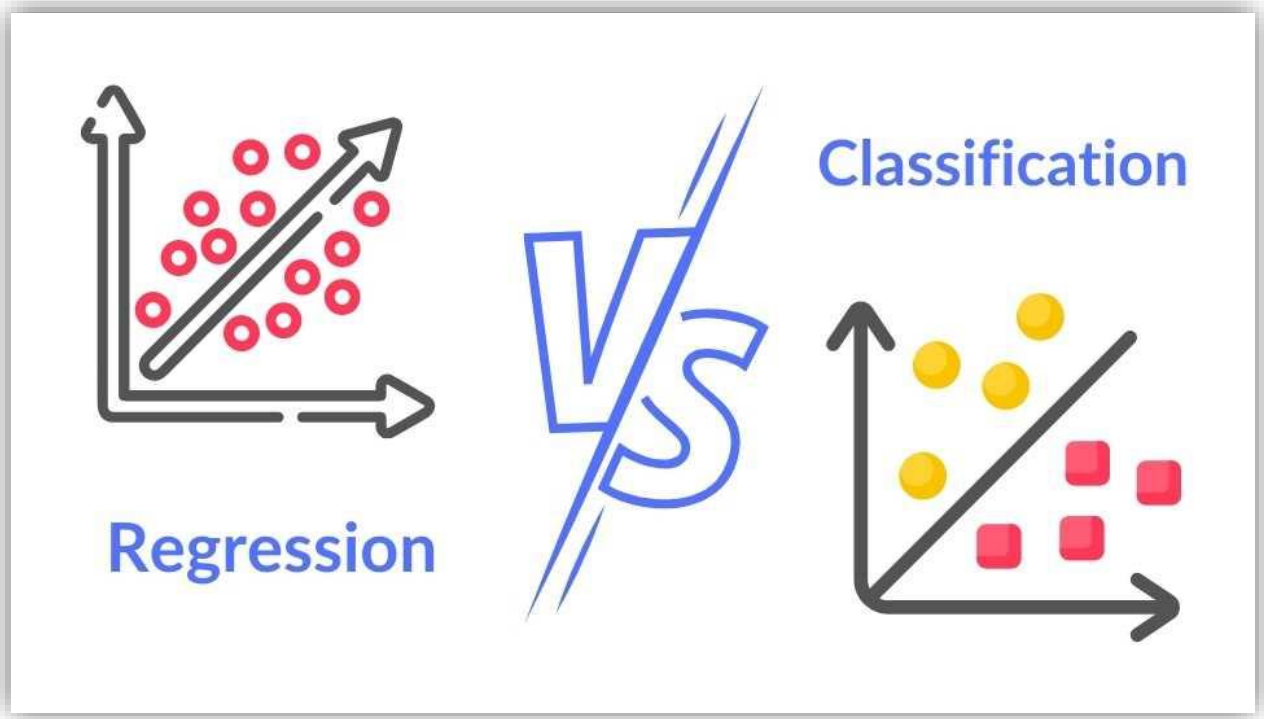
	분류(Classification)	회귀(Regression)
모델링	미리 정의된 클래스(범주) 중 하나로 분류	연속적인 값을 예측
출력 값	범주형 (예: 0, 1 또는 A, B)	연속형 (예: 실수 값)
알고리즘	로지스틱 회귀, 결정 트리, SVM	선형 회귀, 다중 회귀, 랜덤 포레스트 회귀
평가지표 (metric)	Cross-Entropy 정확도(Accuracy), 정밀도(Precision), 재현율(Recall)	평균 제곱 오차(MSE), 결정 계수(R^2)

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression

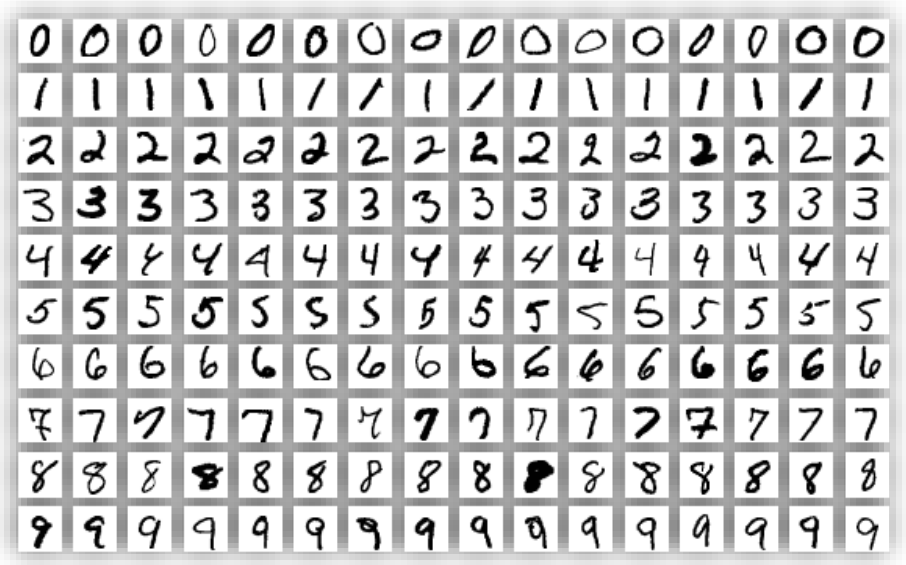
- 실습

  2.05.Classification.Regression.ipynb

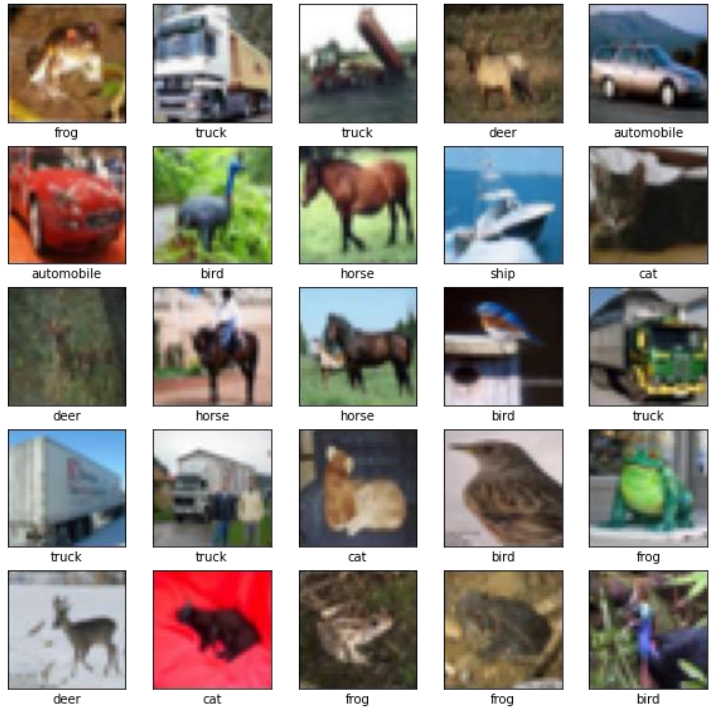


마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression



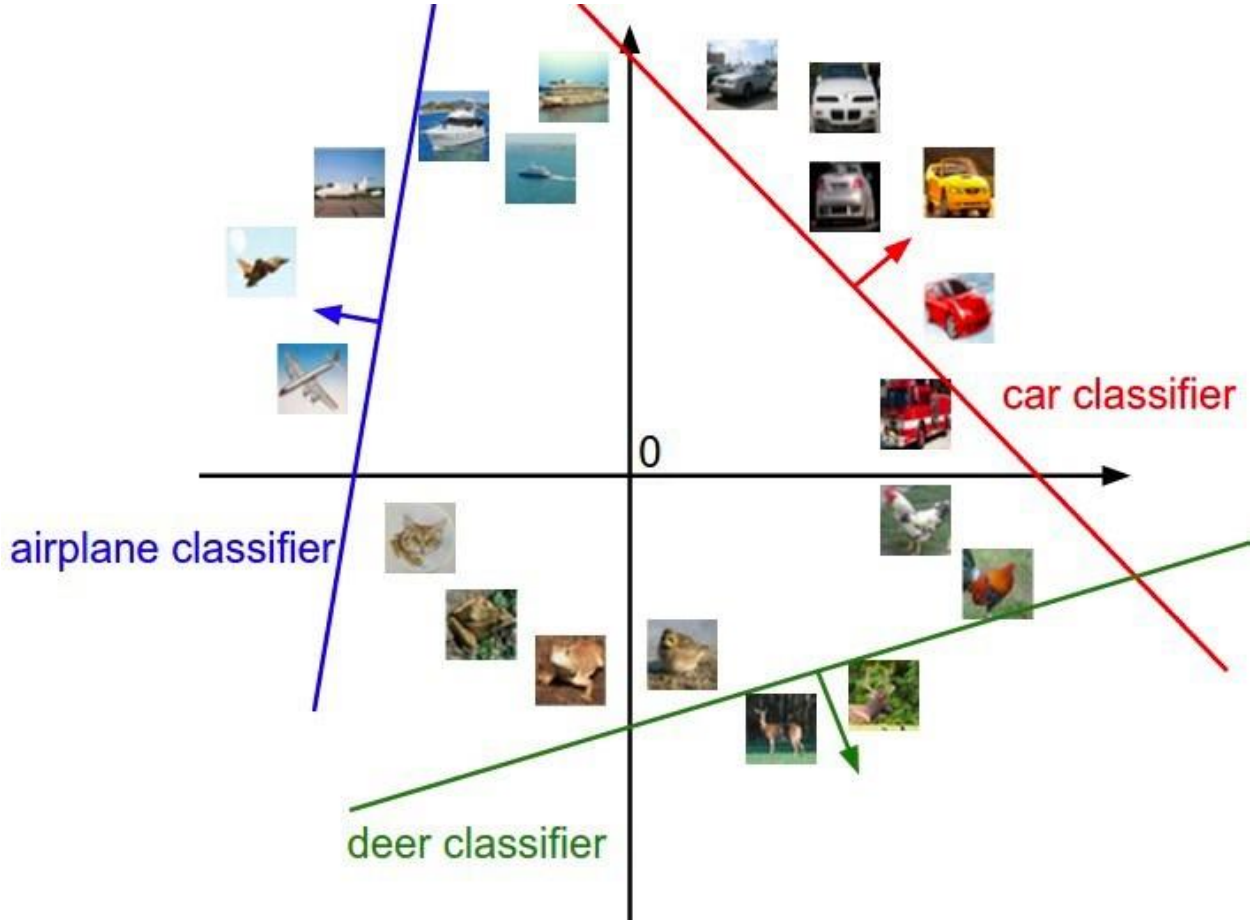
MNIST 손글씨 분류



CIFAR 사진 대상 분류

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression



마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression

텍스트나 표정의 감정 분석

https://monkeylearn.com/sentiment-analysis/

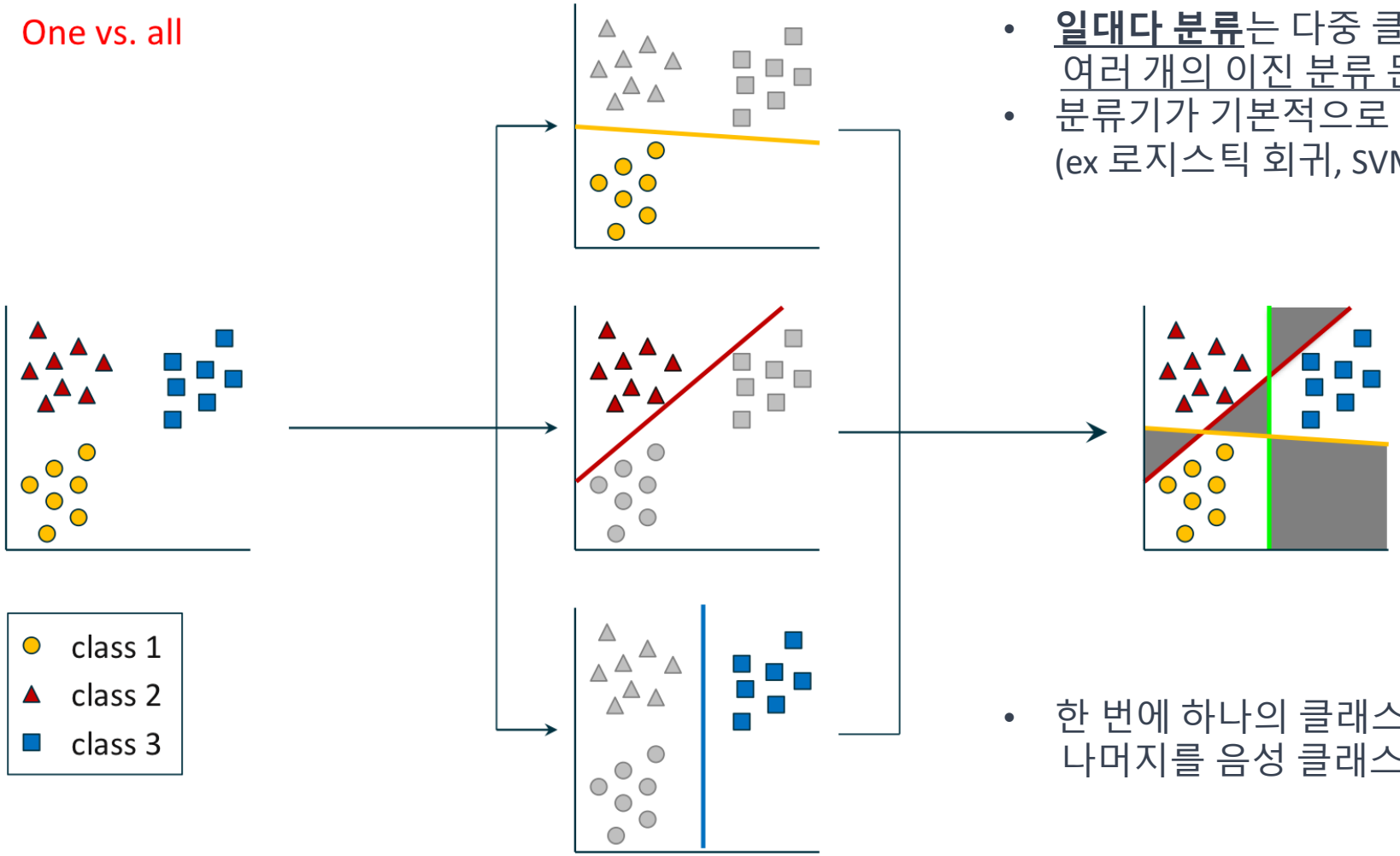
네이버 영화 리뷰 감정분석

	id	document	label
0	9976970	아 더빙 진짜 짜증나네요 목소리	0
1	3819312	흠포스터보고 초딩영화줄오버연기조차 가볍지 않구나	1
2	10265843	너무재밋었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 솔직히 재미는 없다평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화스파이더맨에서 늙어보이기만 했던 커스틴 던...	1

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* Regression

One vs. all

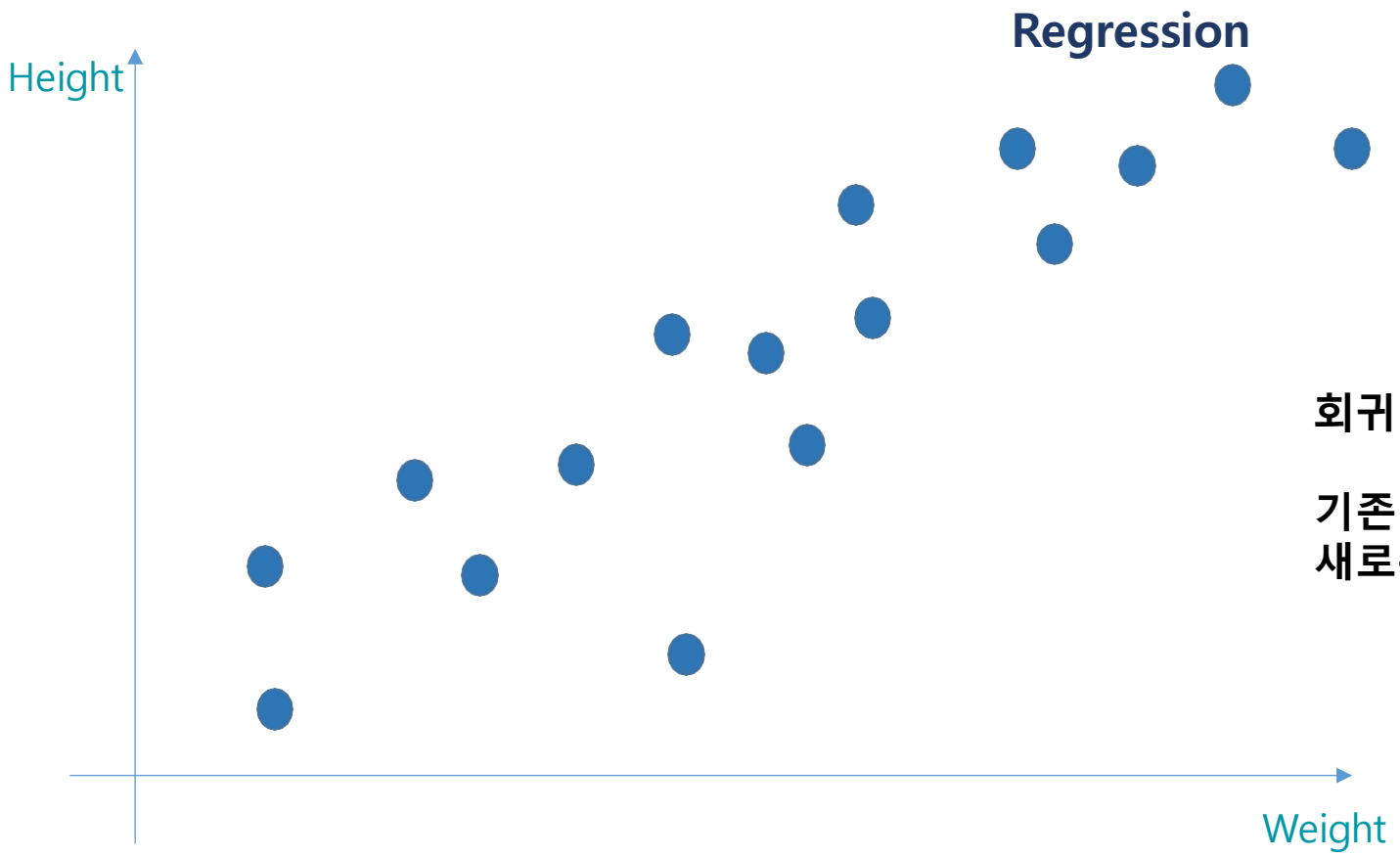


- **일대다 분류**는 다중 클래스 분류 문제를 여러 개의 이진 분류 문제로 변환하여 해결하는 기법.
- 분류기가 기본적으로 이진 분류만을 지원할 때 유용 (ex 로지스틱 회귀, SVM, 이진 트리)

- 한 번에 하나의 클래스를 양성 클래스로 간주하고 나머지를 음성 클래스로 취급하여 분류기를 훈련

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* **Regression**

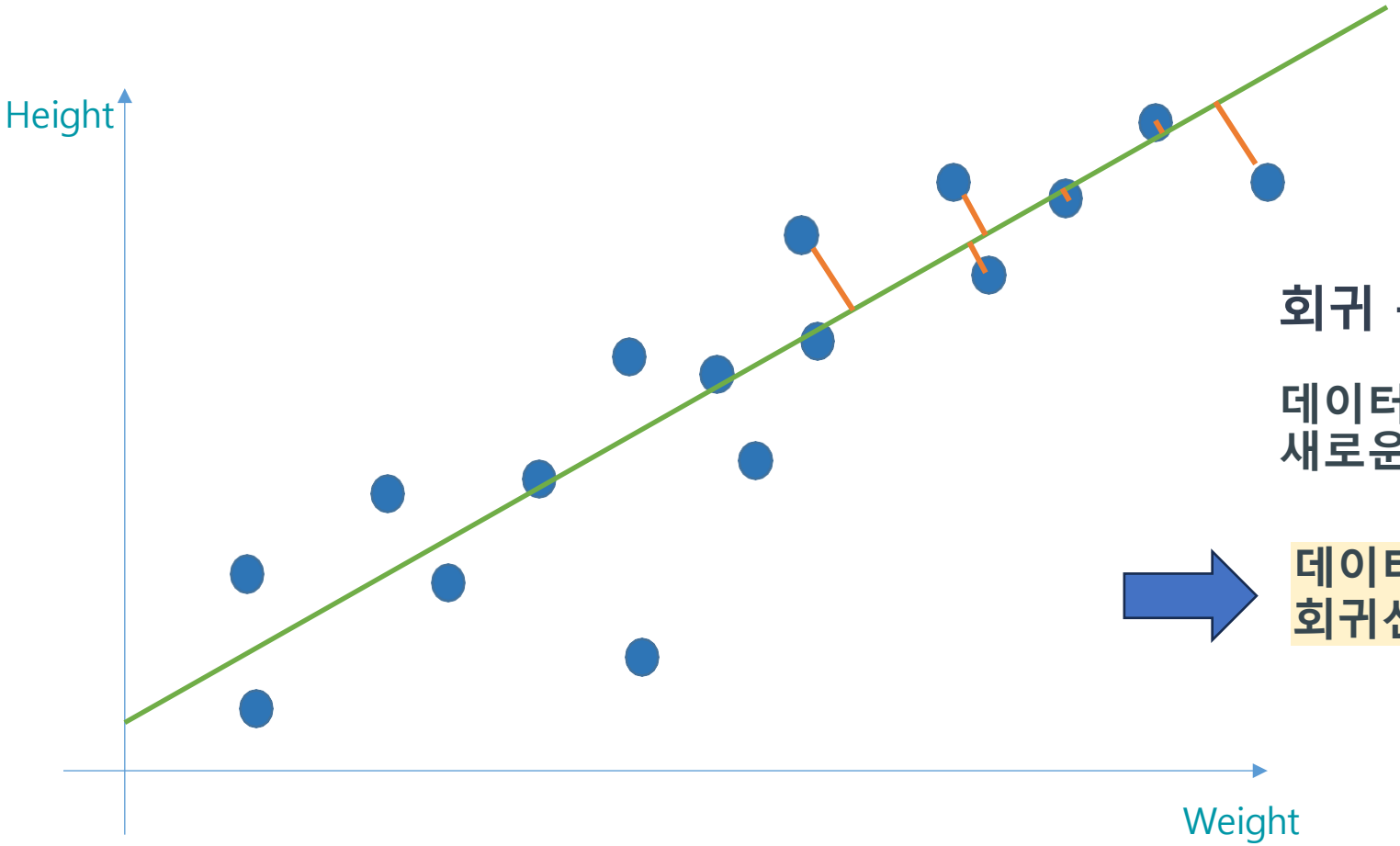


회귀 문제:

기존 데이터를 통한 학습 후,
새로운 데이터의 결과값 예측

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* **Regression**



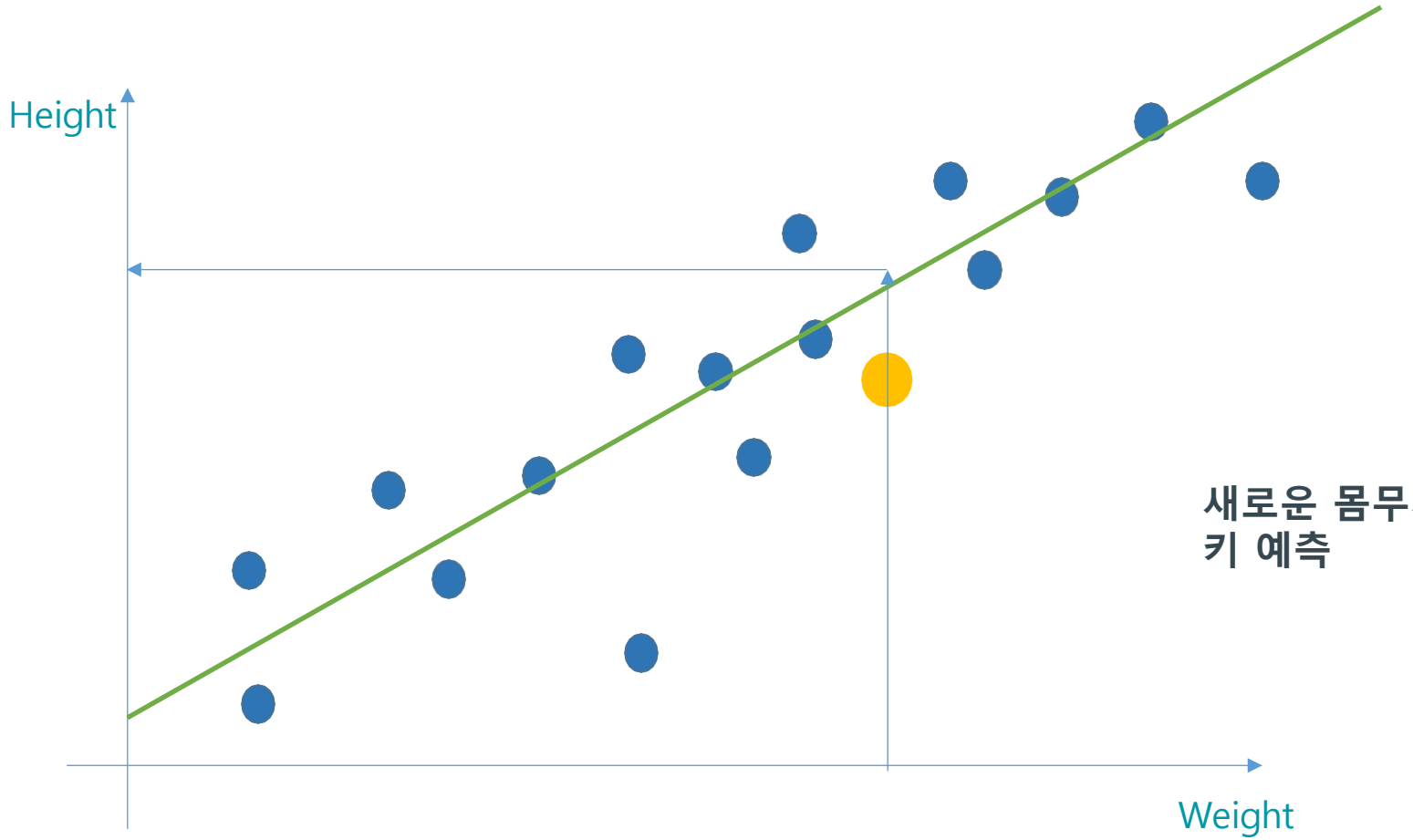
회귀 문제:

**데이터가 주어져 있을 때
새로운 데이터의 결과값 예측**

**데이터들로부터 가장 오차가 적은
회귀선(Regression Line) 계산!**

마이닝 알고리즘 (머신러닝 모델)

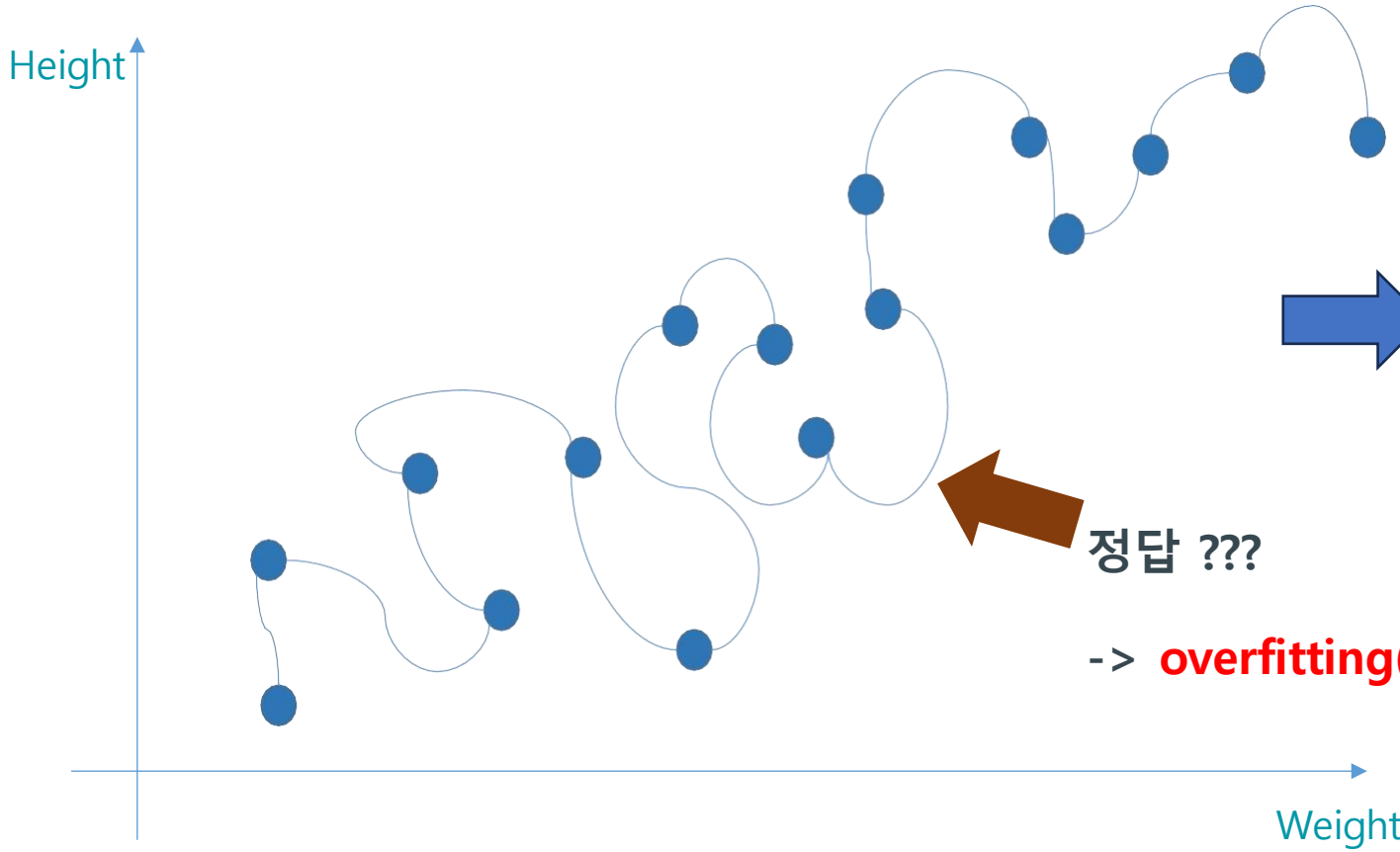
Classification *vs* **Regression**



**새로운 몸무게 데이터가 들어왔을 때,
키 예측**

마이닝 알고리즘 (머신러닝 모델)

Classification *vs* **Regression**



회귀 문제:

데이터가 주어져 있을 때
새로운 데이터의 결과값 예측?

데이터들로부터 가장 오차가 적은
회귀선(Regression Line) 계산!

정답 ???

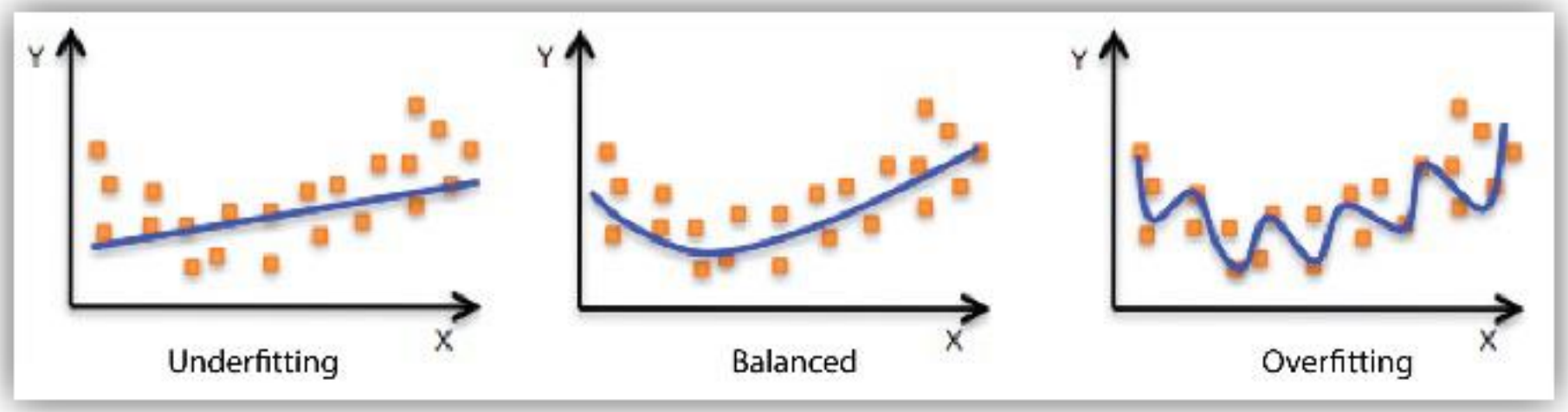
-> overfitting(오버피팅)

마이닝 알고리즘 (머신러닝 모델)

Generalization

- **과적합(Overfitting)**
  - 모델이 **훈련 데이터에 너무 지나치게 맞춰진(fitting) 상태**
  - 새로운 데이터에서는 성능이 크게 떨어지는 현상
  - 모델이 훈련 데이터의 **노이즈나 세부 패턴**까지 학습하여 일반화 능력이 떨어진 상황

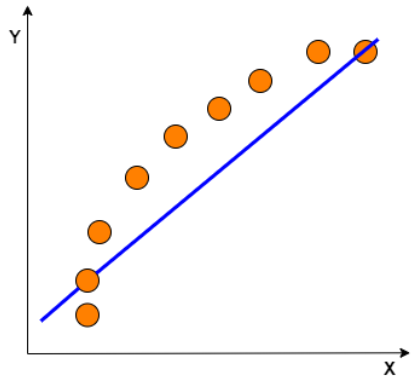
- **일반화(Generalization)**
  - 모델이 **새로운 데이터에 대해 잘 예측할 수 있는** 능력
  - 훈련 데이터 뿐만 아니라 경험하지 않은 **새로운 데이터**에서도 높은 성능을 유지할 수 있는지를 측정하는 개념
  - 일반화가 잘 된 모델은 새로운 데이터에서도 일관된 성능을 나타냄



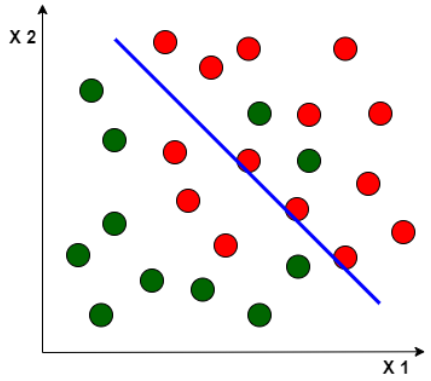
https://wp.wwu.edu/machinelearning/2017/01/22/generalization-and-overfitting/

마이닝 알고리즘 (머신러닝 모델)

Generalization

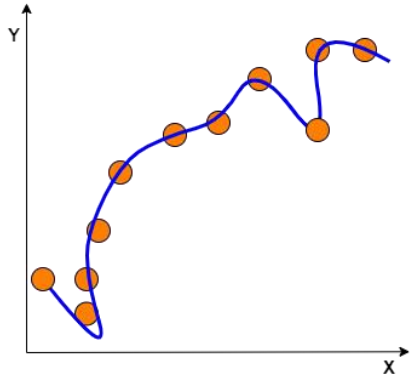


Linear Regression

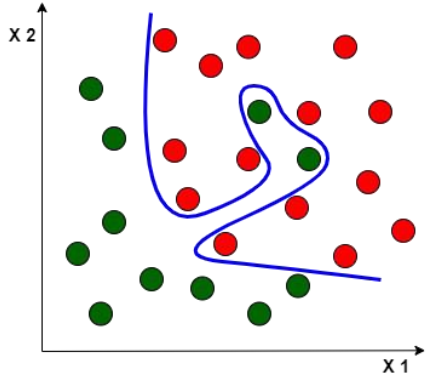


Logistic Regression

Underfitting(언더피팅)
: 학습 데이터의 특성을 제대로 잡지 못하고
피팅이 잘 되지 않은 상태



Linear Regression



Logistic Regression

Overfitting(오버피팅)
: 학습 데이터에 너무 많이 피팅되어
일반화할 수 없어 새로운 데이터에 대한 예측력이 낮아지는 상태

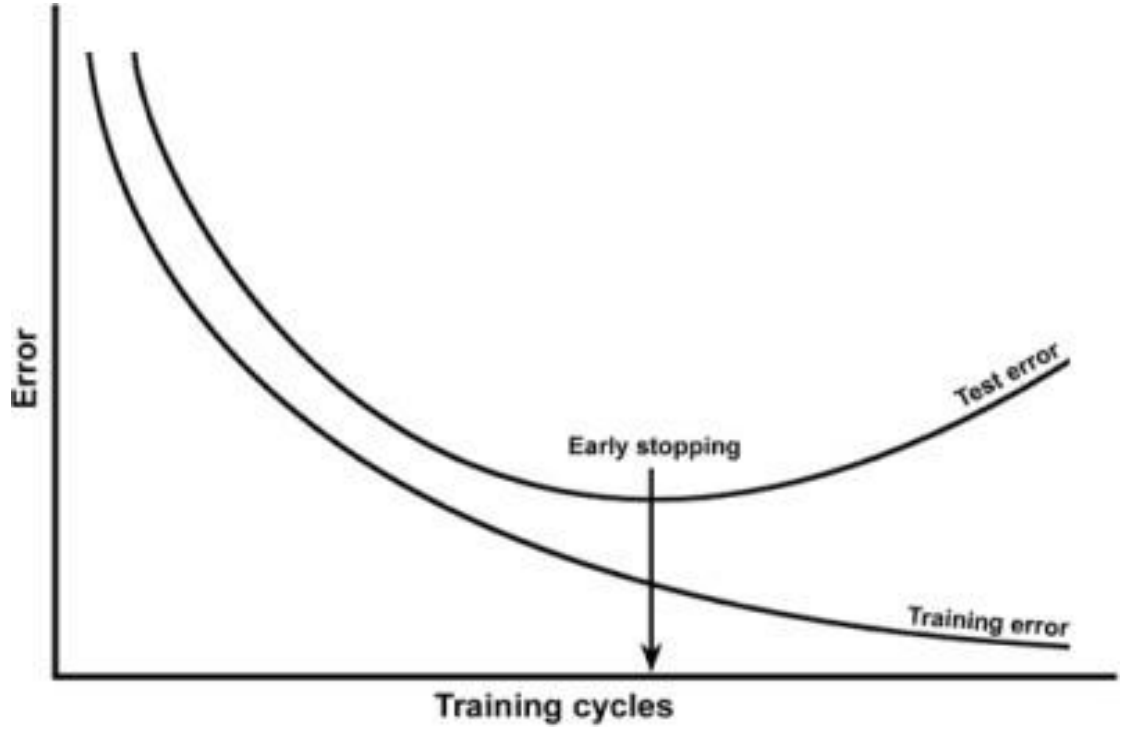
https://satishgunjal.com/underfitting_overfitting/

마이닝 알고리즘 (머신러닝 모델)

Generalization

- 언더피팅과 오버피팅을 피하며 적절히 학습시켜야

Trade off

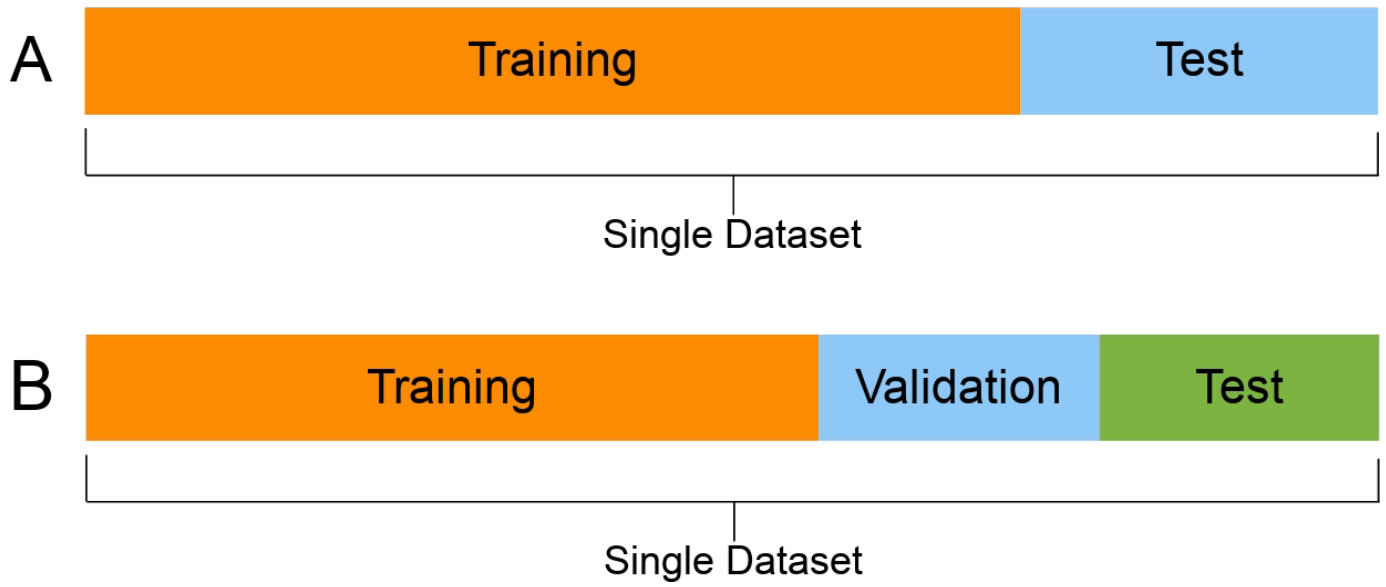


마이닝 알고리즘 (머신러닝 모델)

Generalization **How to train data**

Validation data

목표: 새로운(Unseen) 데이터가 들어왔을 때, 예측력을 높임



https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

Training
: 학습에 사용하는 데이터

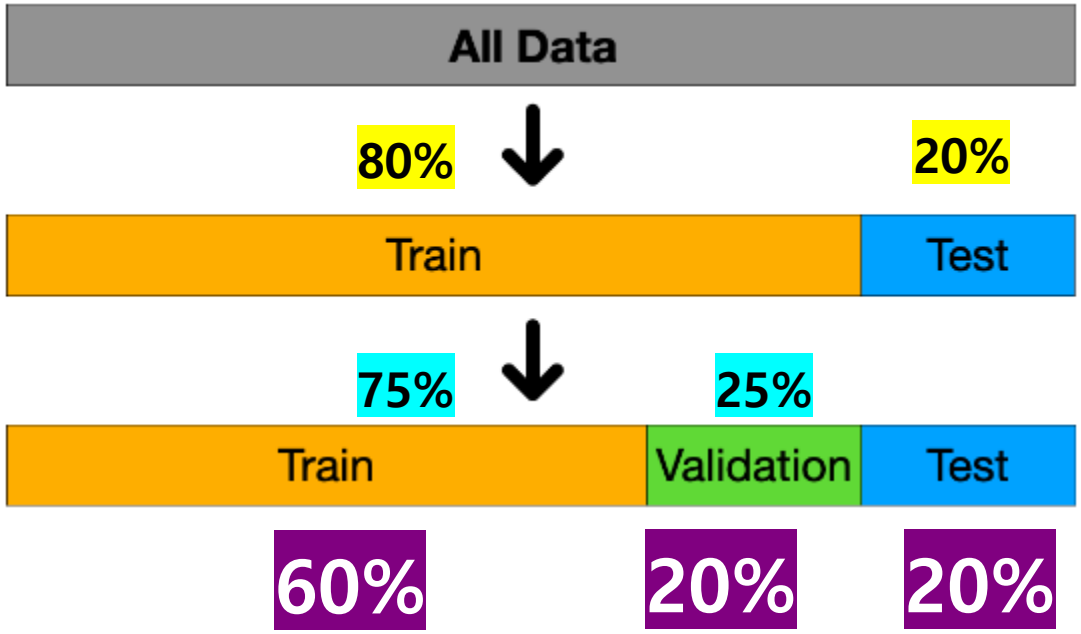
Test
: 학습에 사용하지 않고 성능을 확인 하는 데이터 Unseen data 역할

Validation
: 학습과정에서 모델 검증
accuracy 테스트 -> 오버피팅 방지!

마이닝 알고리즘 (머신러닝 모델)

Generalization **How to train data**

- train
- validation
- test



60%는 훈련 데이터,
20%는 검증 데이터,
20%는 테스트 데이터로 나눔

```
#Further split the training set into training and validation sets(e.g.,75% train, 25%validation)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25, random_state=42)

# 60% training, 20% validation, and 20% test sets
```

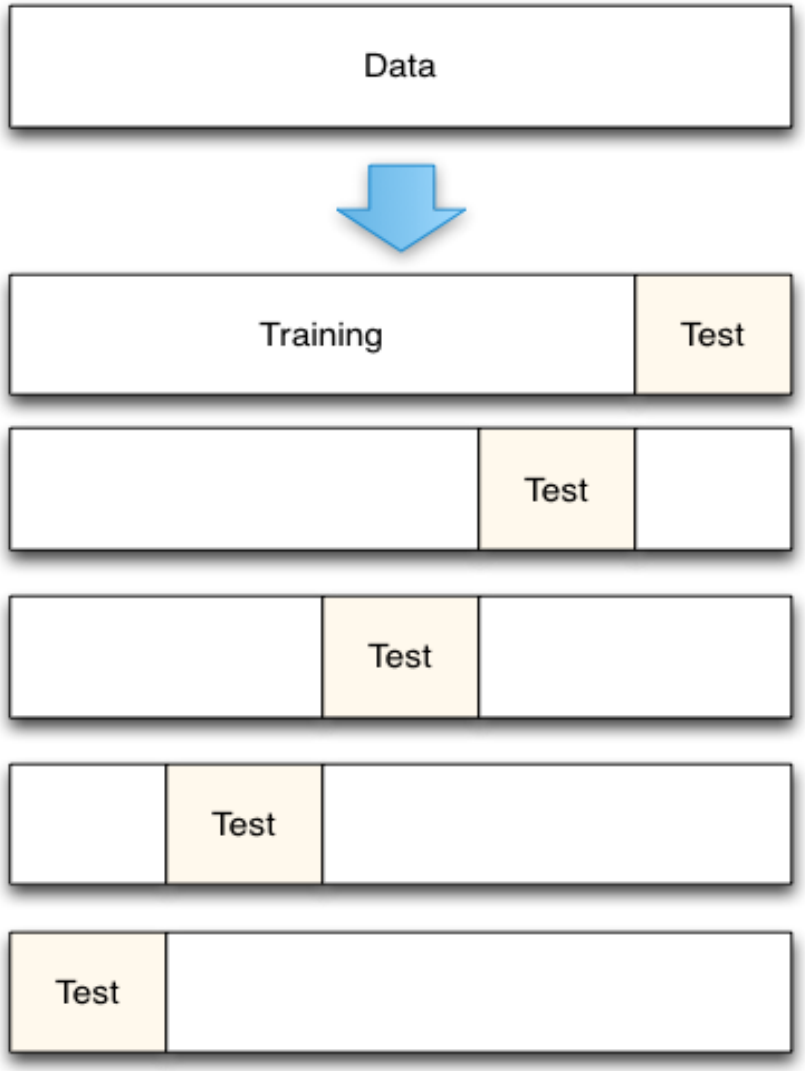
마이닝 알고리즘 (머신러닝 모델)

Generalization **How to train data**

- **K-fold crss validation(K겹 교차검증)**
  : K개의 fold를 만들어서 데이터를 교차로 분할해 검증하는 방법

1. Training data를 K개로 분할.
2. K-1개는 Training data로 1개는 validation data로 지정.
3. 모델 학습 -> predict -> loss 측정
4. 다음 fold에서 validation set를 변경
5. K번 반복.

if K = 5,
데이터를 5개의 폴드로 나뉘고 각 폴드는 한 번씩 검증 세트의 역할을 수행.
작은 데이터 세트에 대한 모델의 성능을 평가할 때 유용

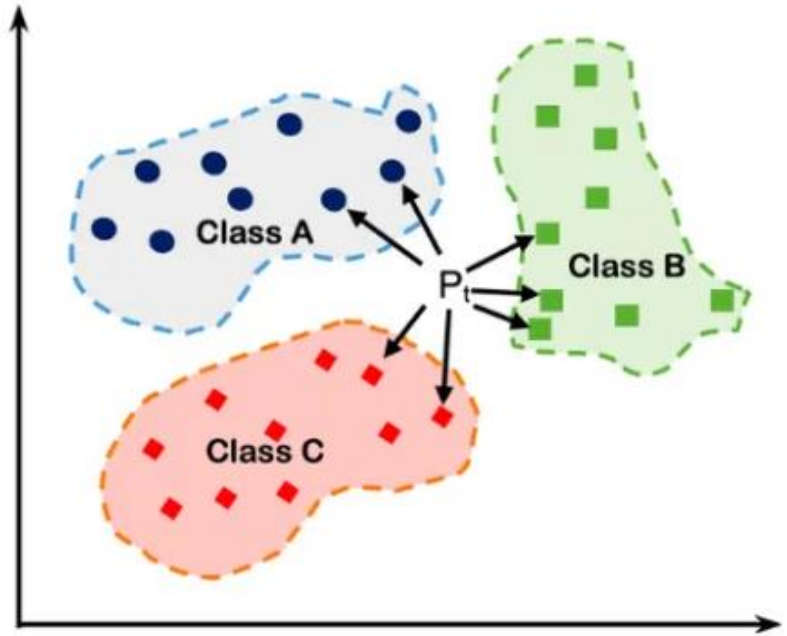


- **실습**

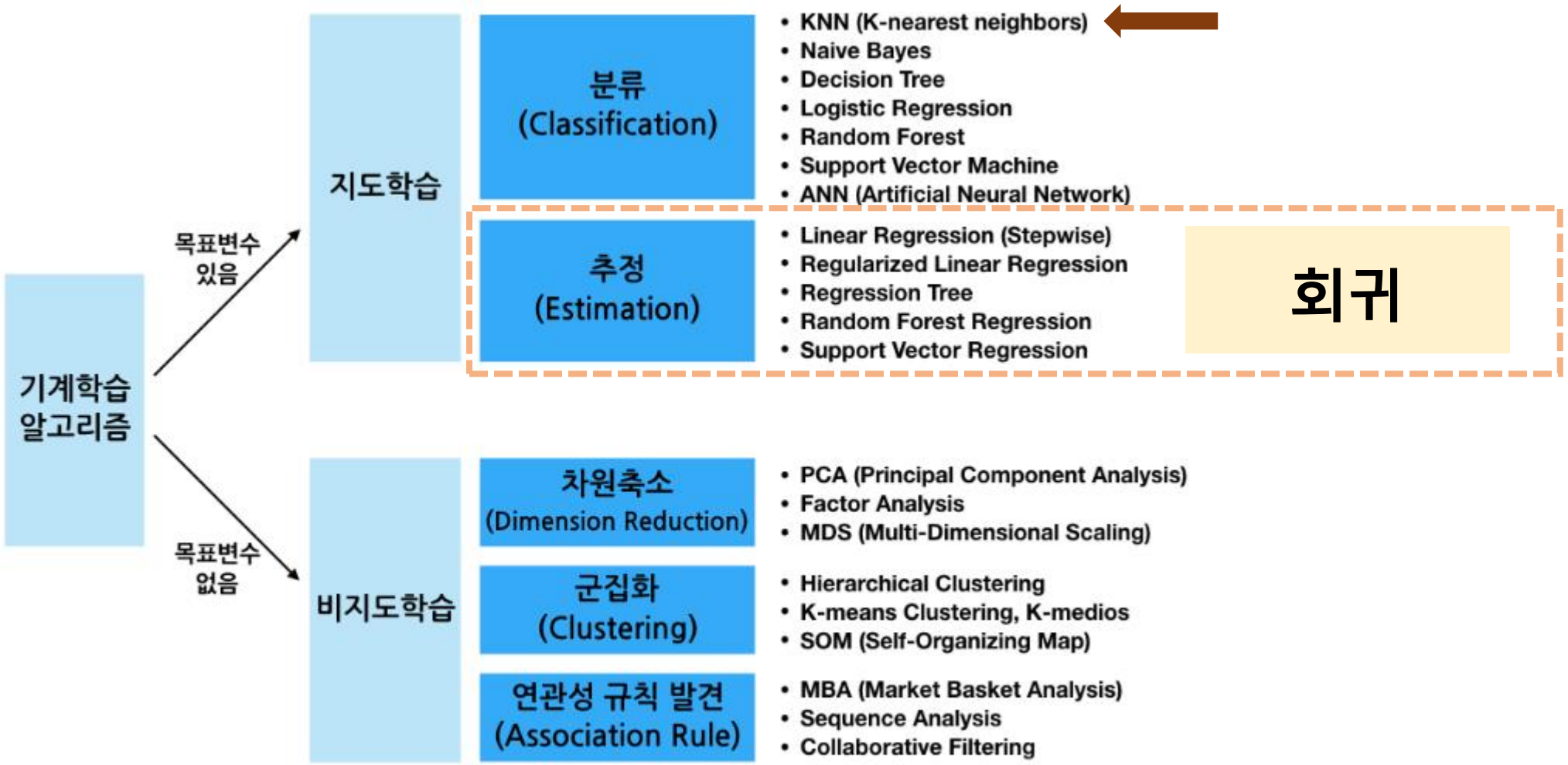
  2.03.PreProcessing.Basic.ipynb

  - Cross-Validation

ML Model : K-Nearest Neighbor



마이닝 알고리즘 (머신러닝 모델)

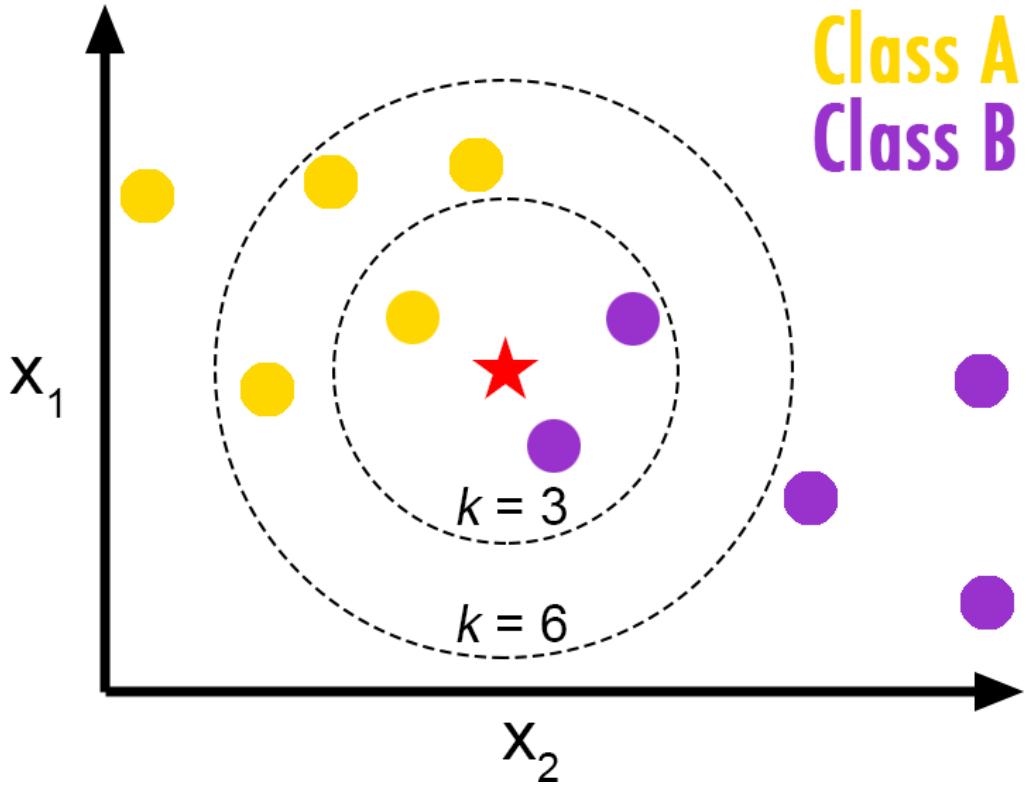


https://dbrang.tistory.com/1520

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

비슷한 속성을 가진 군집?



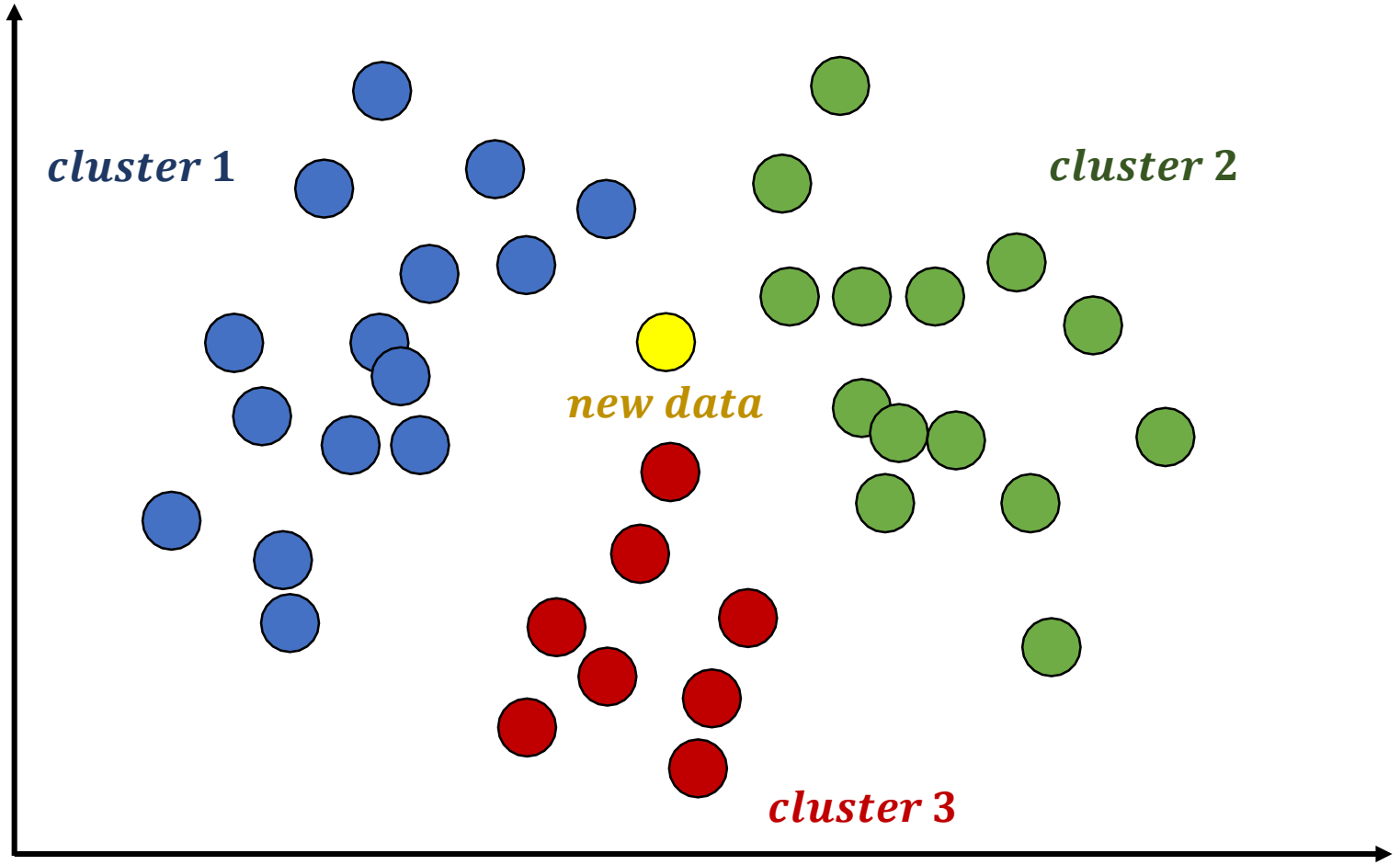
https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e

K-Nearest Neighbor Classifier

주변(이웃)에서
더 많은 데이터가 포함되어 있는 범주로 분류하는 방식

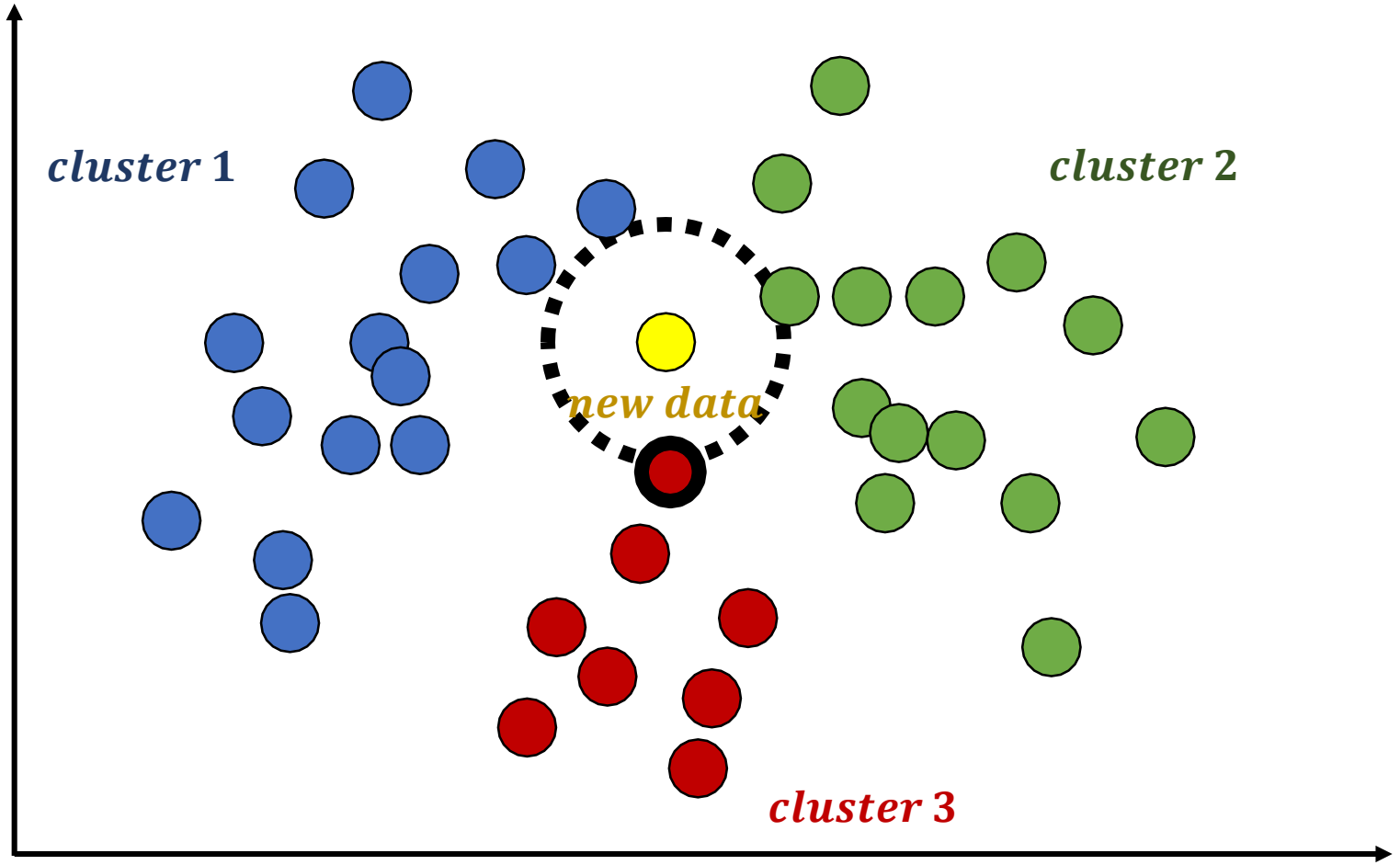
마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor



마이닝 알고리즘 (머신러닝 모델)

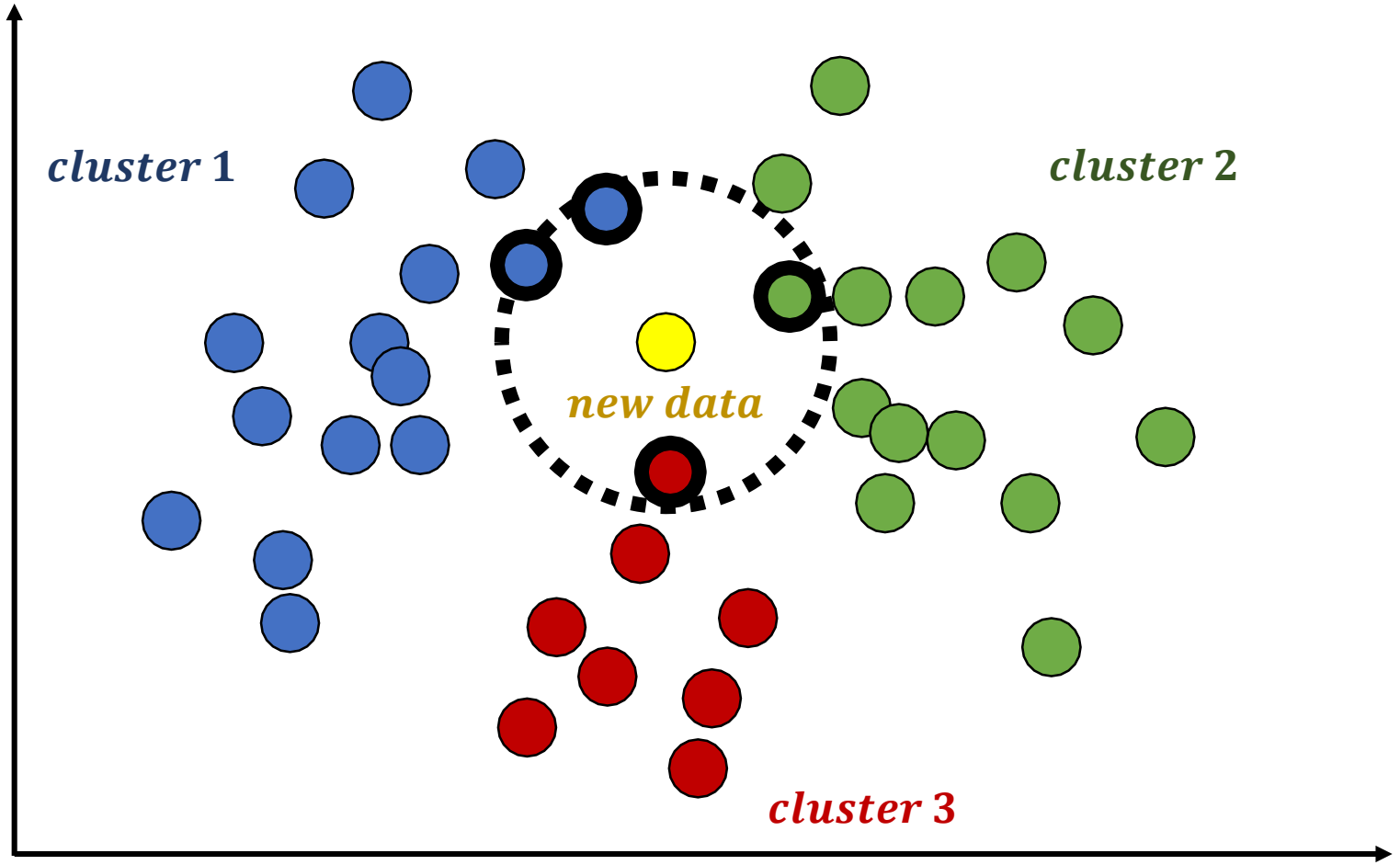
K-Nearest Neighbor



if $K = 1$, *then new data belongs to cluster 3*

마이닝 알고리즘 (머신러닝 모델)

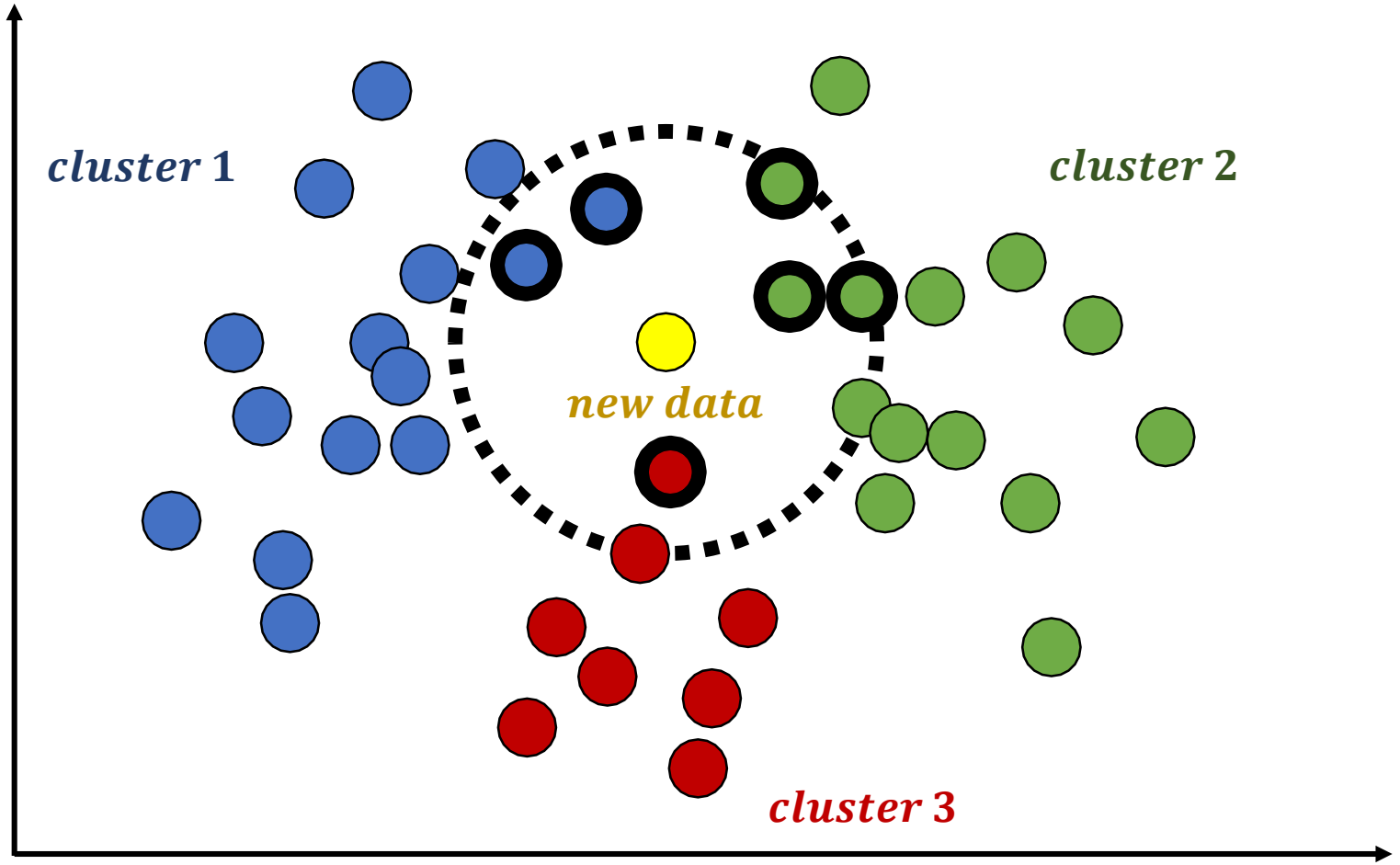
K-Nearest Neighbor



if $K = 4$, *then new data belongs to cluster* 1

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

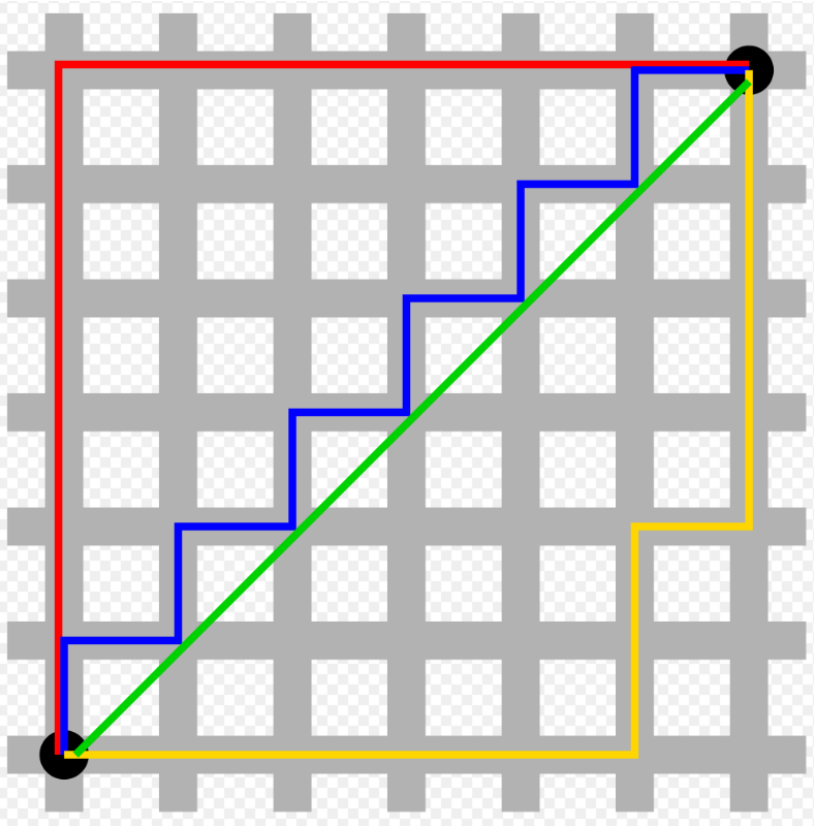


if K = 6, then new data belongs to cluster 2

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

Distance : 거리 계산 방법



맨해튼 거리 :

유클리드 거리 :

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

K : 클래스(cluster)를 판단하기 위해 활용되는 거리가 가까운 데이터 개수

Distance : 거리 계산 방법

① 유클리디안 거리(Euclidean Distance)

$$d(P,Q) = d(Q,P) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + \cdots + (q_i - p_i)^2}$$

Where $P = [p_1, p_2, p_3, \cdots, p_i]$, $Q = [q_1, q_2, q_3, \cdots, q_i]$

② 맨해튼 거리(Manhattan Distance)

$$d(P,Q) = d(Q,P) = |q_1 - p_1| + |q_2 - p_2| + |q_3 - p_3| + \cdots + |q_i - p_i|$$

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

Distance : 거리 계산 방법

- Point P has coordinates $P = (p_1, p_2)$
- Point Q has coordinates $Q = (q_1, q_2)$

$p1$ 과 $p2$: 점 P의 좌표,
$p1$: 첫 번째 차원(ex, x축)의 값
$p2$: 두 번째 차원(ex, y축)의 값

$q1$ 과 $q2$: 점 Q의 같은 차원들의 좌표

① 유클리디안 거리(Euclidean Distance)

$$d(P,Q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

P와 Q를 직접 연결하는 대각선 길이

② 맨해튼 거리(Manhattan Distance)

$$d(P,Q) = |q_1 - p_1| + |q_2 - p_2|$$

P에서 Q로 이동하기 위해 필요한 총 수평 및 수직 이동 거리의 합산
그리드와 같은 도시에서 길을 따라 이동할 때의 거리를 나타 냄

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

Distance : 거리 계산 방법

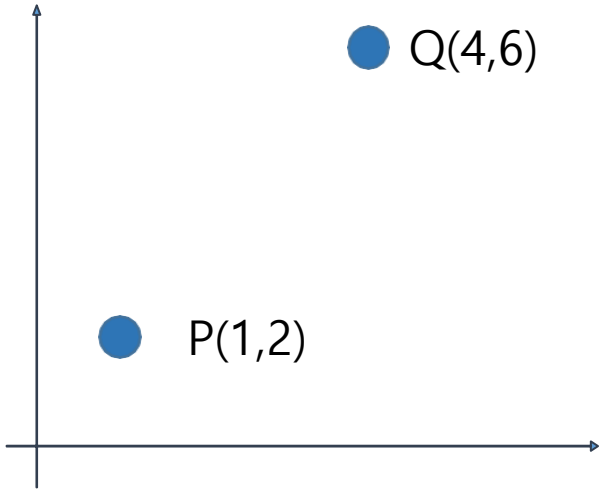
- $P = (1, 2)$
- $Q = (4, 6)$

Euclidean Distance Calculation:

$d(P, Q) =$ [redacted]

Manhattan Distance Calculation:

$d(P, Q) =$ [redacted]



마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

KNN - 학습

새로운 데이터 포인트가 주어졌을 때, **가장 가까운 n개의 이웃**을 찾아 이웃 중
가장 많은 클래스를 새로운 데이터의 클래스로 결정

- **Process**
  1. **거리 계산**: 새로운 데이터와 모든 훈련 데이터 포인트 간의 거리 계산
  2. **이웃 선택**: 계산된 거리 중 가장 가까운 K개의 이웃 선택
  3. **예측**: 선택된 K개의 이웃의 레이블을 기반으로 다수결 투표(분류)나 평균 계산(회귀)을 통해 새로운 데이터의 레이블이나 값 예측

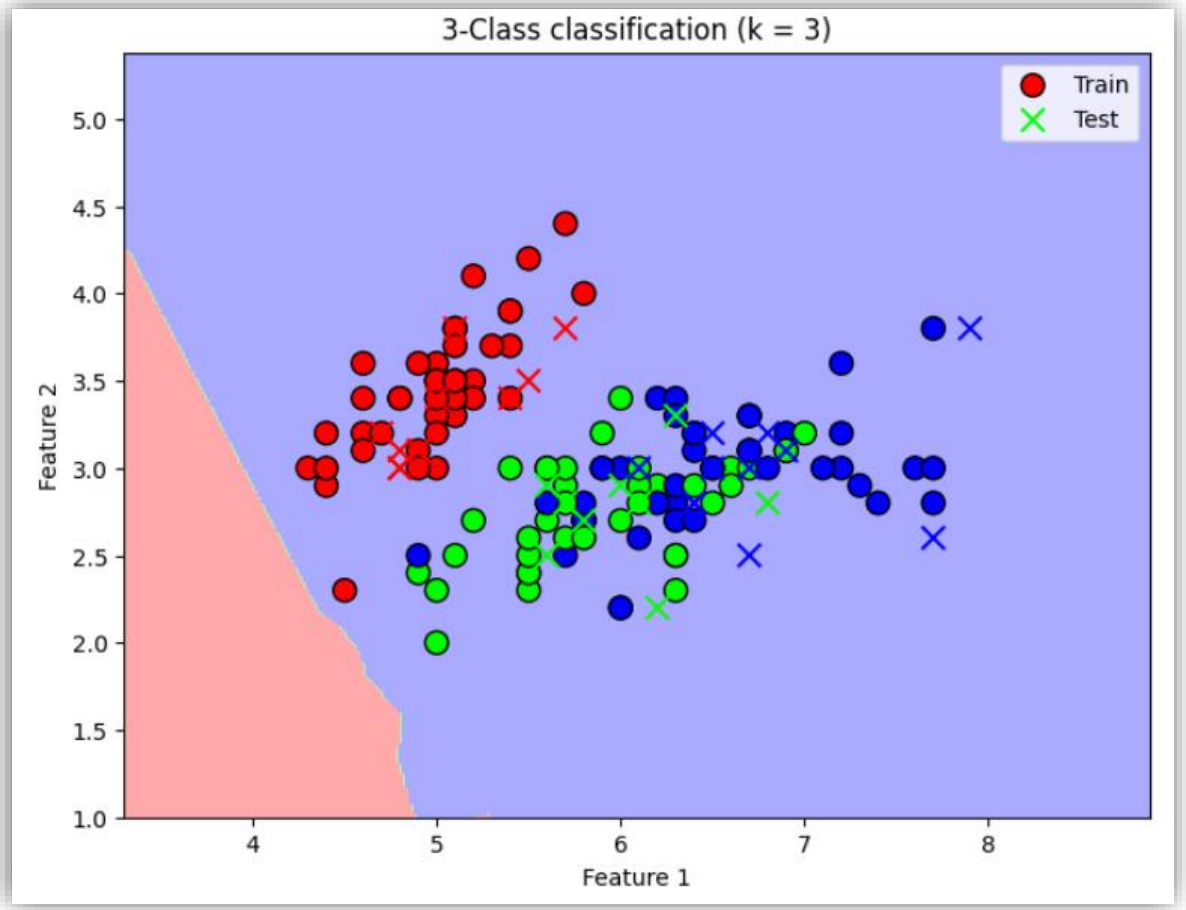
- **lazy learning(게으른 학습)**
  가장 많은 라벨 탐색? 학습?
  테스트 데이터가 들어오면 그때 계산을 하기 때문에 '**게으른 학습(lazy learning)**'이라고 표현

마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

- 실습
  2.05.KNN.ipynb

- TASK
  K값 조절(if 5, 7) 하여 성능 확인



마이닝 알고리즘 (머신러닝 모델)

K-Nearest Neighbor

- **실습**

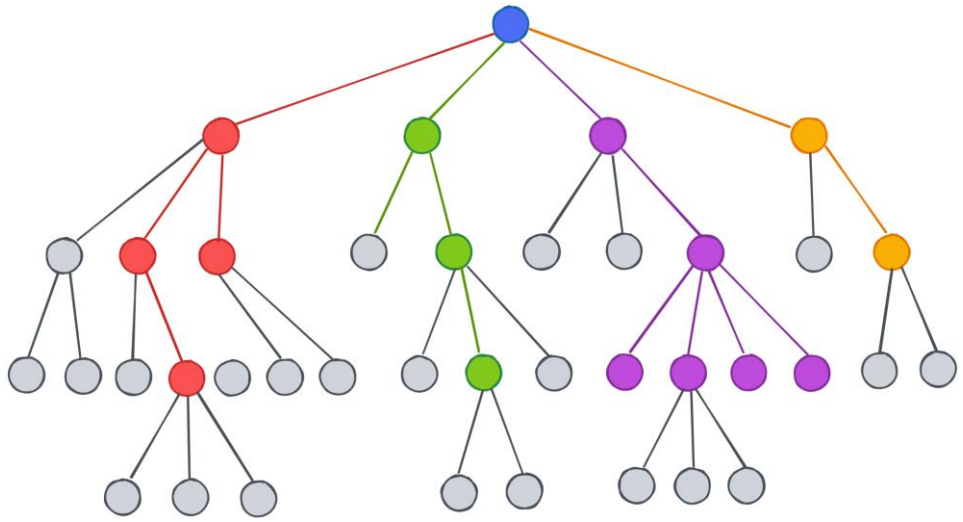
  2.05.KNN.ipynb – 교차 검증

  - **K 값의 선택**
    - K 값이 작을수록 모델이 노이즈에 민감
    - K 값이 클수록 모델이 과도하게 일반화되어 성능

  => 교차 검증을 통해 최적의 K 값 선택

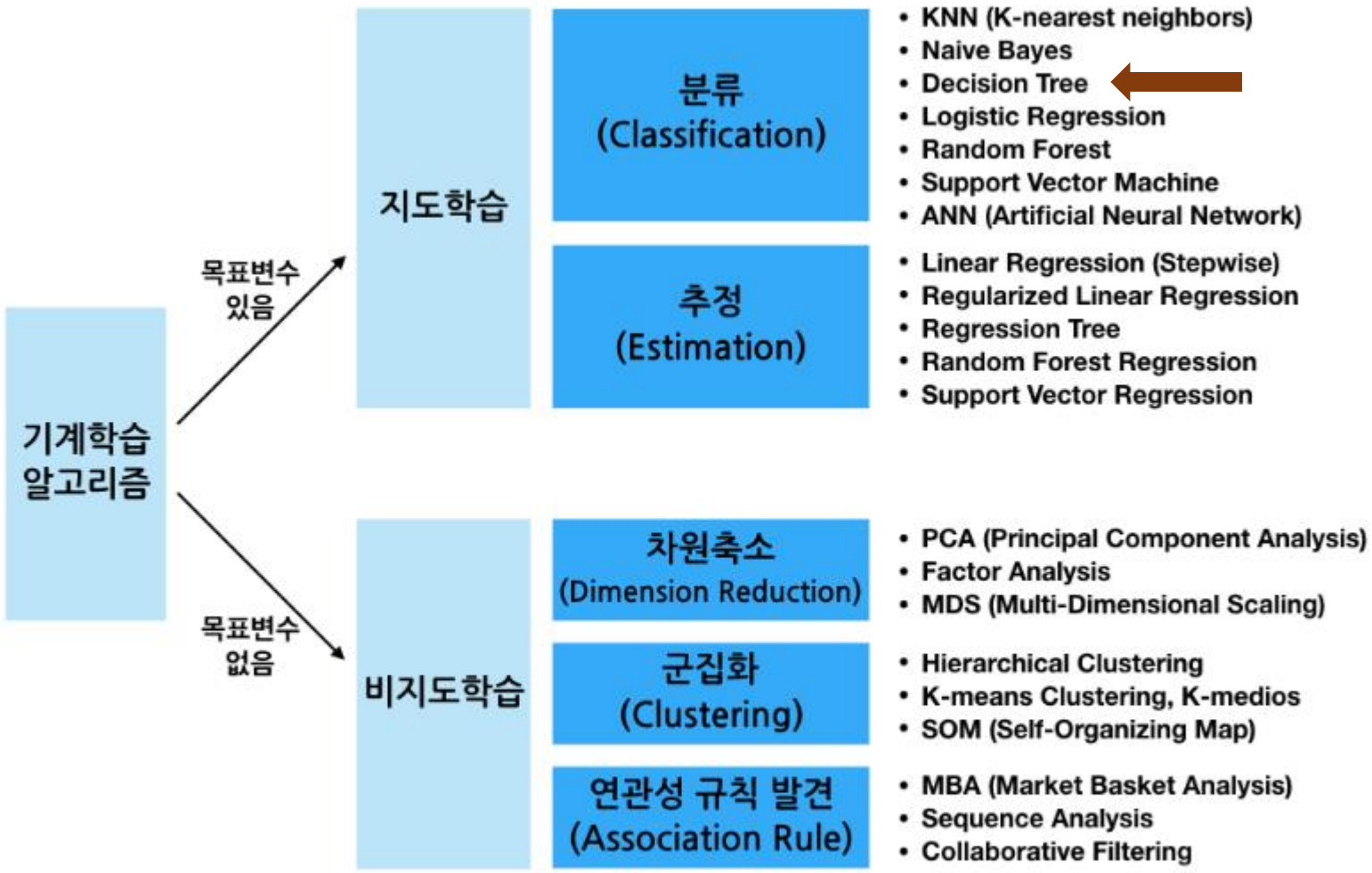
```
from sklearn.model_selection import cross_val_score
import numpy as np

# 다양한 K 값에 대해 교차 검증 점수 계산
k_range = range(1, 31)
k_scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X_train, y_train, cv=5, scoring='accuracy')
    k_scores.append(scores.mean())

# 최적의 K 값 선택
optimal_k = k_range[np.argmax(k_scores)]
```

ML Model : Decision Tree



마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)

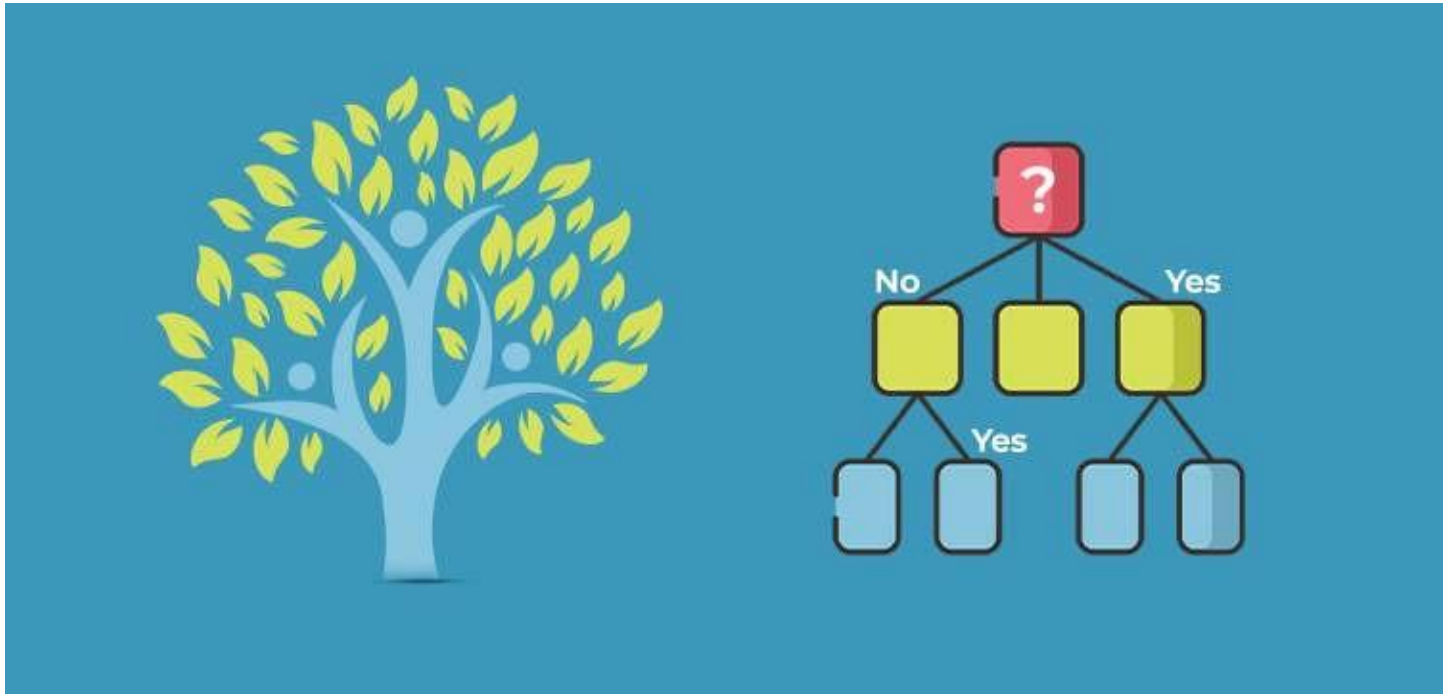


https://dbrang.tistory.com/1520

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)

- 특정 기준(질문)에 따라 데이터를 구분하는 모델
- 한번의 분기 때마다 변수 영역을 두 개로 구분

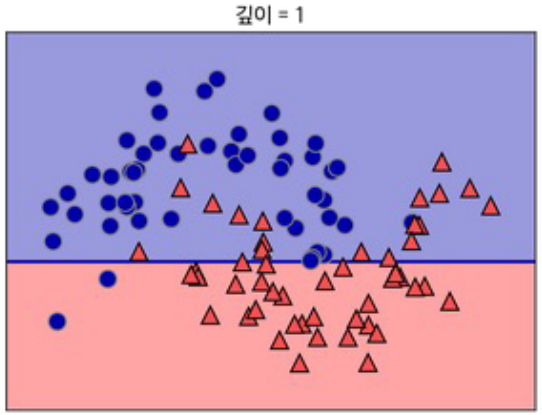


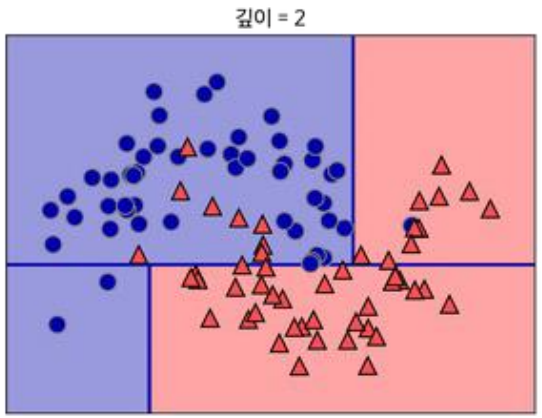
https://www.analyticssteps.com/blogs/classification-and-regression-tree-cart-algorithm

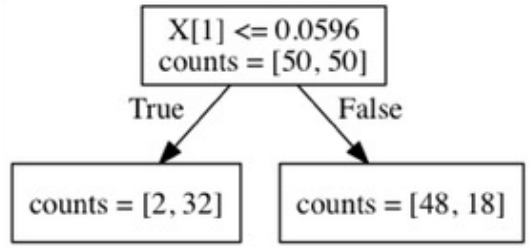
마이닝 알고리즘 (머신러닝 모델)

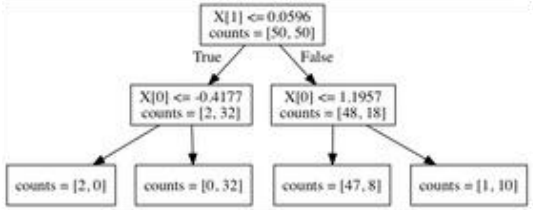
Decision Tree(의사결정나무)

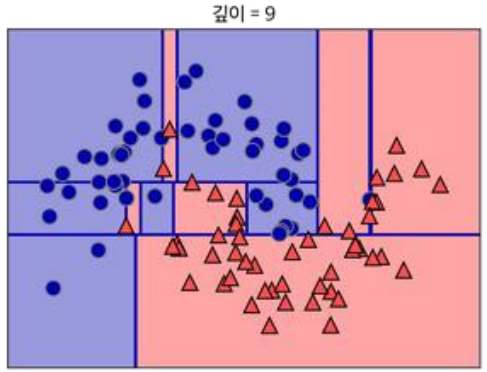
- 특정 기준(질문)에 따라 데이터를 구분하는 모델
- 한번의 분기 때마다 변수 영역을 두 개로 구분

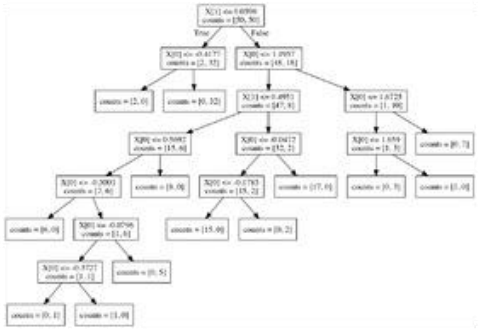












결정 트리의 오버피팅

https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-4-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%ACDecision-Tree?category=1057680

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) – Node 분할

- 엔트로피 : 데이터 세트의 무질서도를 측정
- **정보 이득** : 특성이 결과에 대한 엔트로피를 얼마나 감소시키는지를 나타냄.
- **정보 이득이 큰 방향으로 노드 분할**

- **엔트로피(Entropy)**
  - 데이터의 **불확실성**을 측정하는 지표
  - 엔트로피 값이 클수록 데이터가 혼합, 엔트로피가 낮을수록 데이터가 더 순수
  - 엔트로피 공식: $$\text{Entropy}(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$ Pi: 클래스 i의 확률

ex, 14개의 샘플
9개는 클래스 1, 5개는 클래스 0
클래스 1의 확률 p1=9/14, 클래스 0의 확률 p0=5/14

엔트로피 계산: $$\text{Entropy}(S) = -\left(\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right)$$

Entropy(S)≈0.940

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) – Node 분할

- 엔트로피 : 데이터 세트의 무질서도를 측정
- **정보 이득** : 특성이 결과에 대한 엔트로피를 얼마나 감소시키는지를 나타냄.
- **정보 이득이 큰 방향으로 노드 분할**

- **정보 이득(Information Gain)**
  - 노드를 분할 후 데이터의 엔트로피가 얼마나 줄어드는지를 나타내는 지수
  - 정보 이득이 클수록 분할 후 데이터가 더 순수해진다는 의미
  - 정보 이득 공식

$$\text{Information Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

S: 전체 데이터셋,
Sv: 속성 A값 v를 가진 데이터셋

ex, 엔트로피가 0.940인 데이터셋을 속성 A를 기준으로 분할
A=0: 엔트로피 = 0.811, 데이터셋 크기 = 6
A=1: 엔트로피 = 0.918, 데이터셋 크기 = 8

분할 후 엔트로피: $\frac{6}{14} \times 0.811 + \frac{8}{14} \times 0.918 = 0.864$

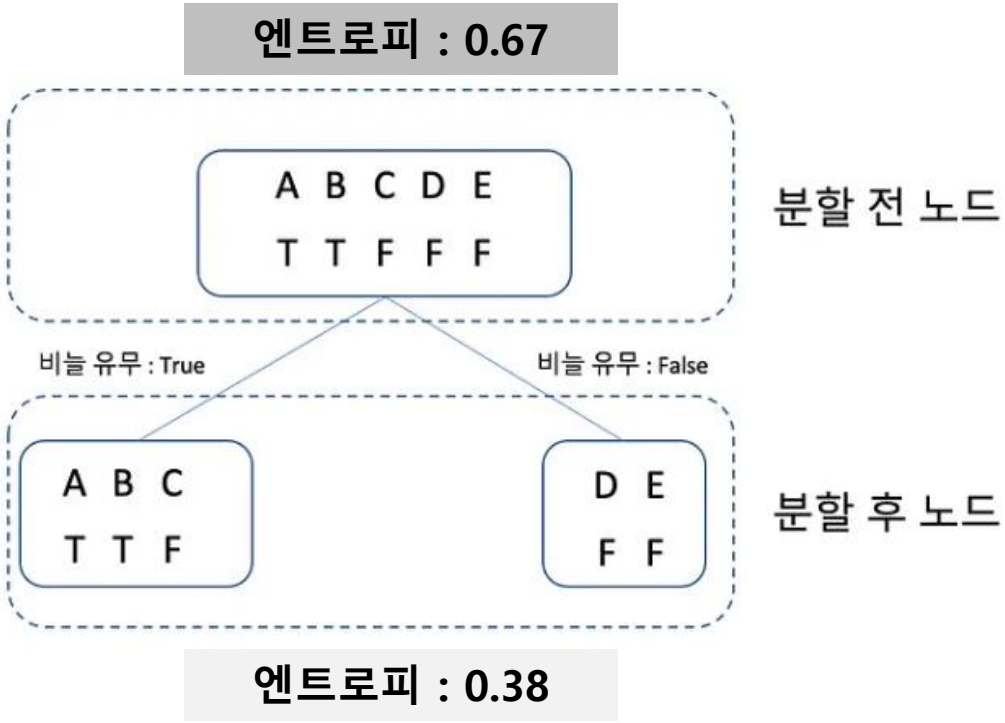
정보 이득: 0.940–0.864 = 0.076

분할 후 불확실성이 다소 감소

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) – Node 분할

- 엔트로피 : 데이터 세트의 무질서도를 측정
- **정보 이득** : 특성이 결과에 대한 엔트로피를 얼마나 감소시키는지를 나타냄.
- **정보 이득이 큰 특성**은 더 유용한 정보를 제공한다고 판단하여 선택



분할 전 엔트로피 = $-\left(\frac{2}{5}\log_2\left(\frac{2}{5}\right)+\frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)$

분할 후 엔트로피 = $-\left(\frac{3}{5}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right)+\frac{2}{3}\log_2\left(\frac{2}{3}\right)\right)+\frac{2}{5}\left(\frac{0}{2}\log_2\left(\frac{0}{2}\right)+\frac{2}{2}\log_2\left(\frac{2}{2}\right)\right)\right)$

정보 획득 = 분할 전 엔트로피 - 분할 후 엔트로피 = 0.67 - 0.38 = 0.29

0.29만큼 불확실성 감소

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)

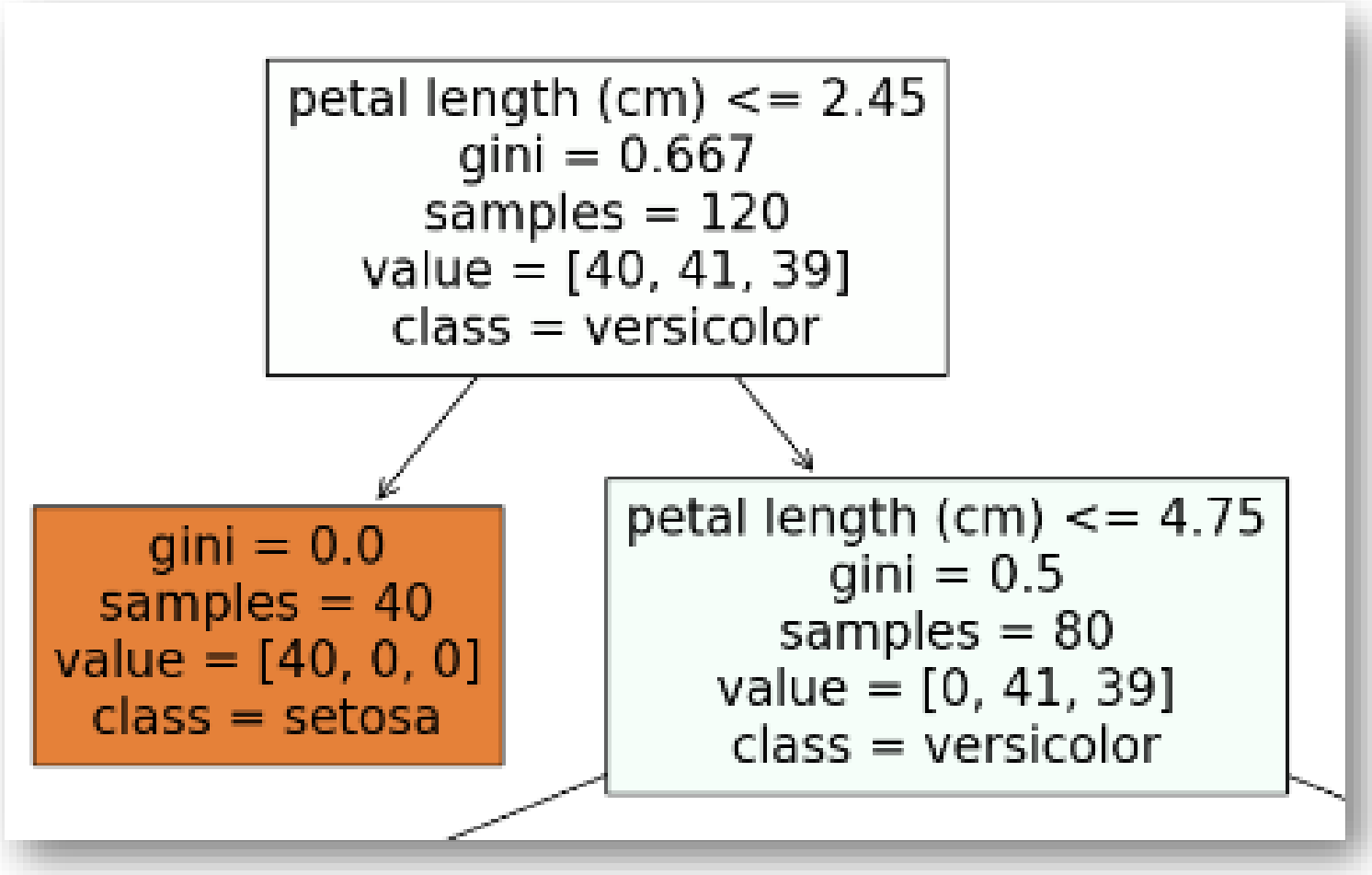
- 엔트로피 : 데이터 세트의 무질서도를 측정
- **정보 이득** : 특성이 결과에 대한 엔트로피를 얼마나 감소시키는지를 나타냄.
- **정보 이득이 큰 특성**은 더 유용한 정보를 제공한다고 판단하여 선택

- **참고**

  2.01.FeatureSelection.Entropy.ipynb

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)

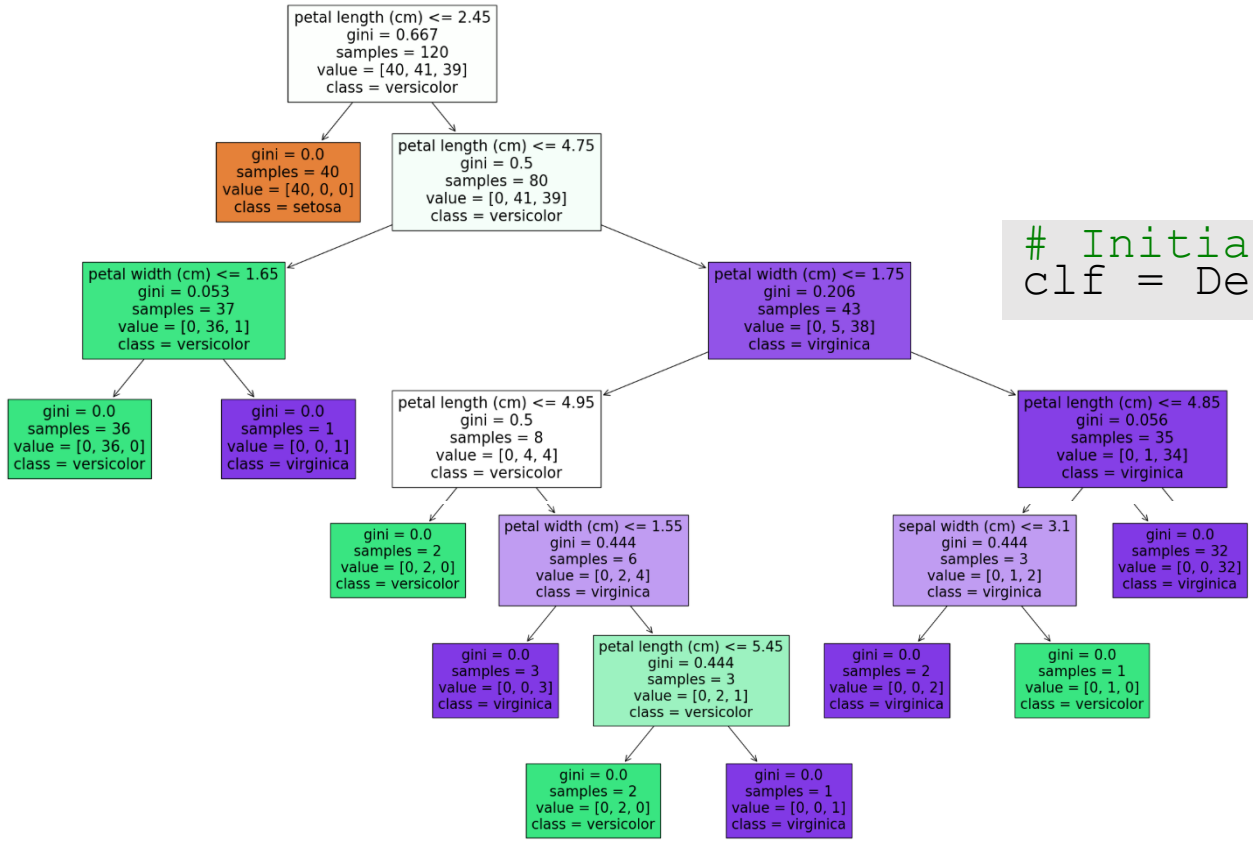


- **실습**

  2_05_DT.ipynb

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)



- **실습**

  2_05_DT.ipynb

```
# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)
```

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) - 지니 계수(Gini Coefficient)

- 불순도(Impurity)를 측정하는 지표
- 노드를 분할할 때 각 노드가 얼마나 순수한지(즉, 하나의 클래스에 속한 데이터가 얼마나 많은지) 평가
- 값이 0일 때 가장 순수한 상태를 의미, 값이 클수록 데이터가 더 혼합되어 있음을 나타냄

- **지니 계수(Gini Coefficient) 식:**

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2$$

•Pi 클래스i의 비율, C는 클래스 수

  - 값이 0에 가까울수록 해당 노드는 **순수**, 하나의 클래스만 포함하고 있음을 의미
  - 값이 0.5에 가까울수록 두 클래스가 **동일한 비율**로 섞여 있음을 의미
  - 값이 1에 가까울수록 노드의 불순도가 매우 높음

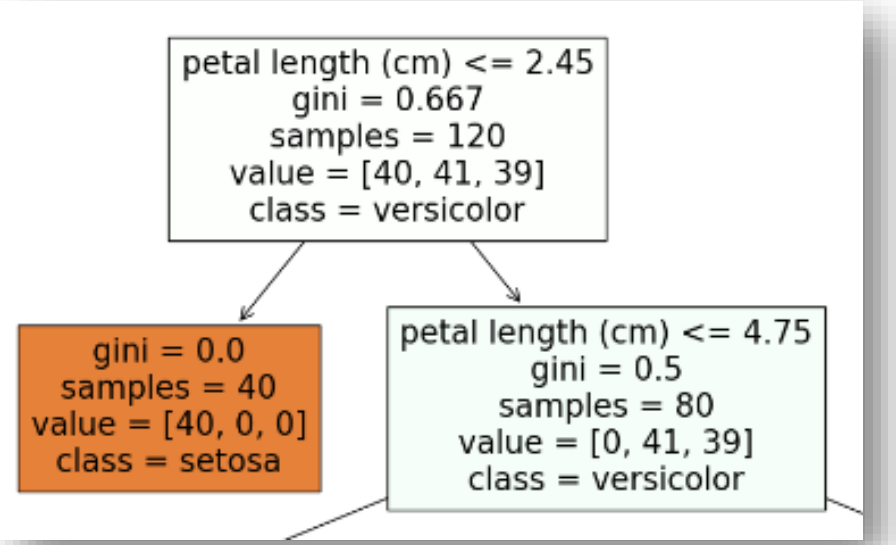
- **cf, 정보 이득(Information Gain):**
  - 노드를 분할 시 엔트로피(Entropy)가 얼마나 감소했는지 나타내는 지수
  - 분할 전과 분할 후 엔트로피의 차이로 계산

$$Entropy(S) = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

$$InformationGain = Entropy(S_{before}) - \sum_{v} \frac{|S_v|}{|S|} Entropy(S_v)$$

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)



- **실습**

  2_05_DT.ipynb

```
# 기본값 지니 계수
clf = DecisionTreeClassifier(random_state=42) # criterion='gini '  기본값
```

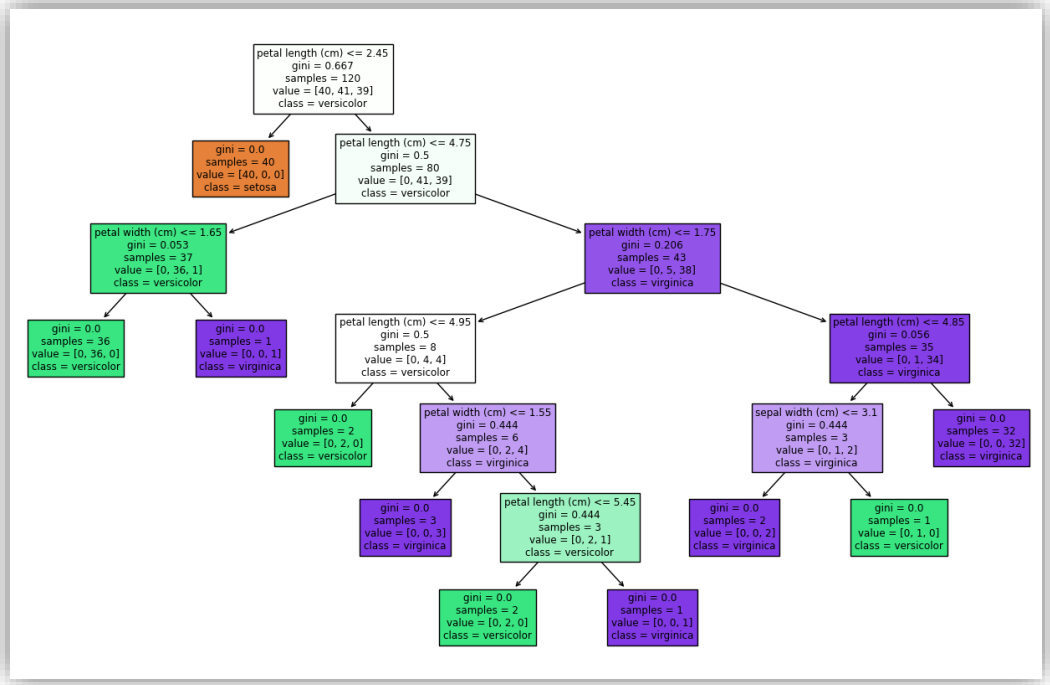
- **TASK**

  정보이득 사용, 성능 비교

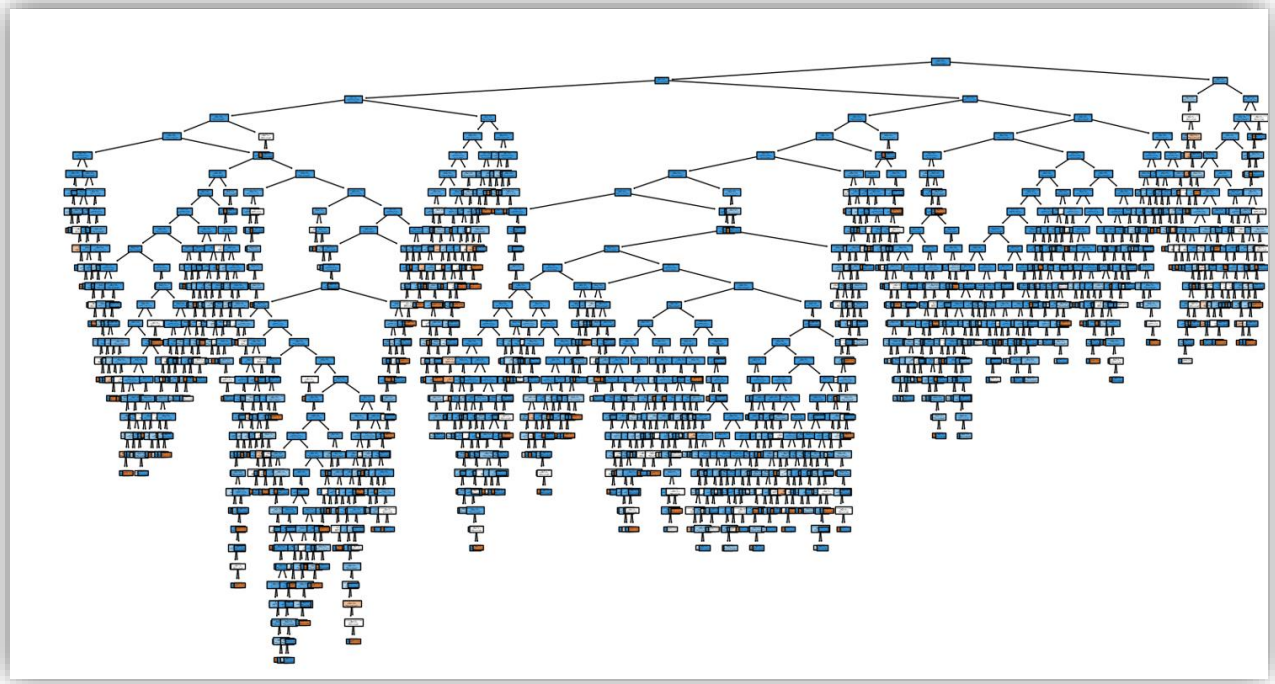
```
# 정보 이득(엔트로피) 사용
clf = DecisionTreeClassifier(criterion='entropy', random_state=42)
```

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)



과적합



마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무)

```
from sklearn.tree import DecisionTreeClassifier, export_text
# 결정 트리 규칙을 텍스트로 출력
tree_rules = export_text(clf, feature_names=iris.feature_names)
print(tree_rules)
```

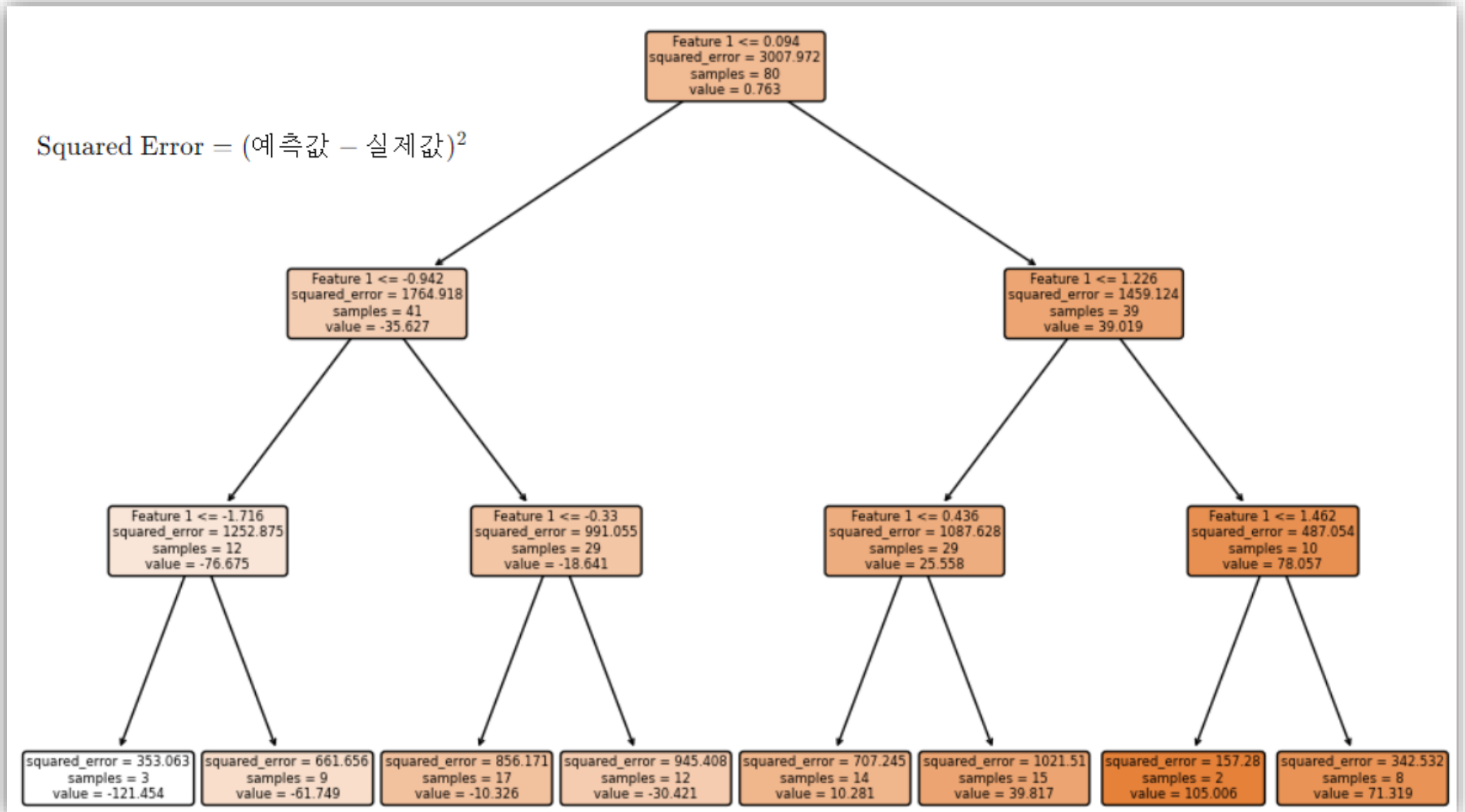
```
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal length (cm) <= 4.75
|   |   |--- petal width (cm) <= 1.65
|   |   |   |--- class: 1
|   |   |--- petal width (cm) >  1.65
|   |   |   |--- class: 2
|   |--- petal length (cm) >  4.75
|   |   |--- petal width (cm) <= 1.75
|   |   |   |--- petal length (cm) <= 4.95
|   |   |   |   |--- class: 1
|   |   |   |--- petal length (cm) >  4.95
|   |   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- petal width (cm) >  1.55
|   |   |   |   |   |--- petal length (cm) <= 5.45
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- petal length (cm) >  5.45
|   |   |   |   |   |   |--- class: 2
|   |   |--- petal width (cm) >  1.75
|   |   |   |--- petal length (cm) <= 4.85
|   |   |   |   |--- sepal width (cm) <= 3.10
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- sepal width (cm) >  3.10
|   |   |   |   |   |--- class: 1
|   |   |   |--- petal length (cm) >  4.85
|   |   |   |   |--- class: 2
```

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) - 회귀

- 실습

2.03.dt_regression.ipynb



마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) - 회귀

분류		회귀
DecisionTreeClassifier	*vs*	DecisionTreeRegressor

공통점

- **트리 구조**: 각 노드가 속성에 대한 테스트(분할 기준)를 나타내고 각 분기가 테스트 결과를 나타냄
  예측을 나타내는 리프 노드로 이어지는 트리형 결정 모델을 사용
- **Recursive Splitting** (재귀적 분할): 재귀적 이진 분할을 구현하여 데이터를 대상 변수와 관련하여
  최대한 동일한 하위 집합으로 분할
- Stopping Criteria : max_length, min_samples_split 및 min_samples_leaf와 같은 트리 성장을
  제어하는 매개변수 구성 -> 과적합 방지

마이닝 알고리즘 (머신러닝 모델)

Decision Tree(의사결정나무) - 회귀

분류		회귀
DecisionTreeClassifier	*vs*	DecisionTreeRegressor

차이점

Split 기준

- 하위 노드 내 클래스의 impurity(불순도)를 줄이는 것 (Gini impurity and entropy)
- 목표 : 클래스 레이블 內 동질적인 하위 그룹 달성

- 하위 노드 내의 분산 or MSE 를 줄이는 것
- 목표 : 응답이 가능한 한 서로 가까운 leaf을 갖는 것 = 각 노드에서 대상 변수의 분산 또는 MSE 최소화

Leaf Node Prediction

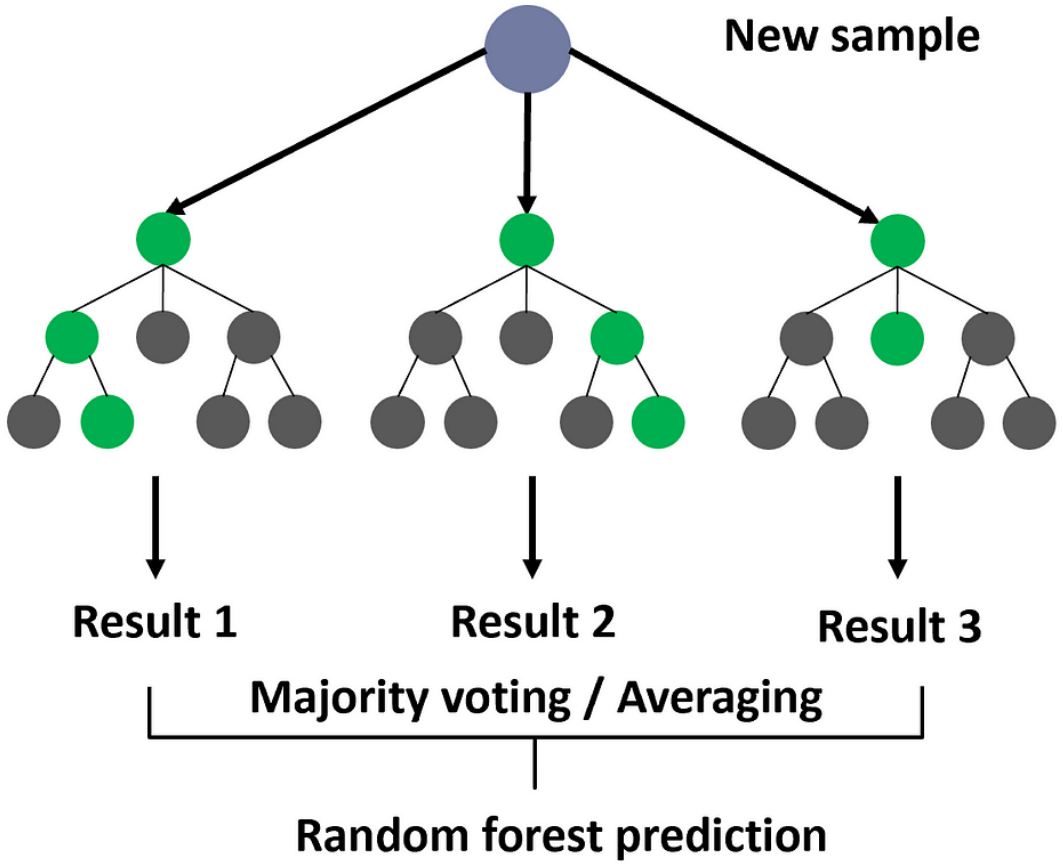
- 해당 리프 내의 훈련 인스턴스 중 대다수 클래스

- 해당 리프 내 인스턴스의 대상 값의 평균 (or 중앙값)

Example

- 이메일 스팸 구분

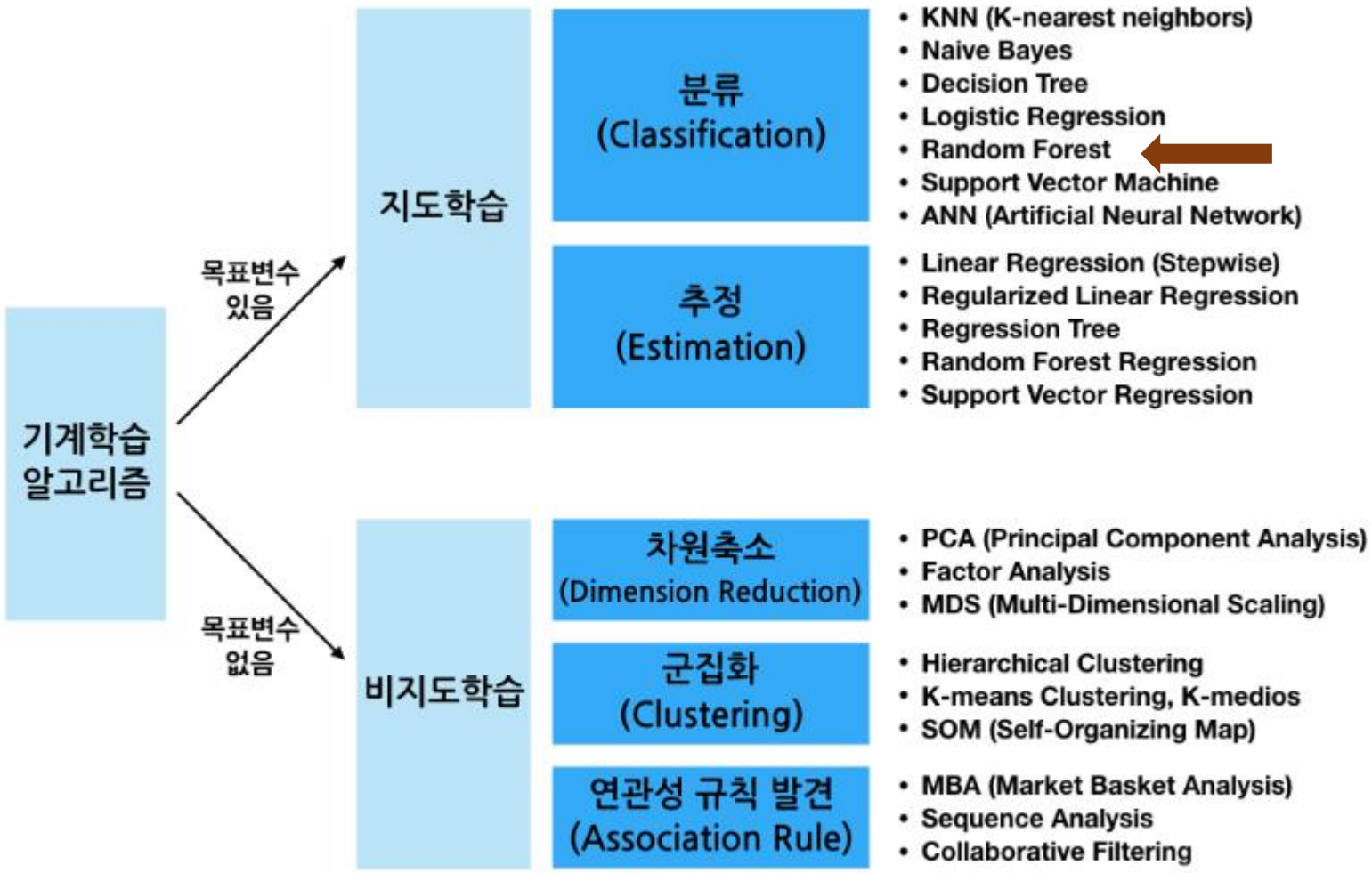
- 크기, 위치 등의 특징 기반으로 주택 가격 예측



Random Forest

마이닝 알고리즘 (머신러닝 모델)

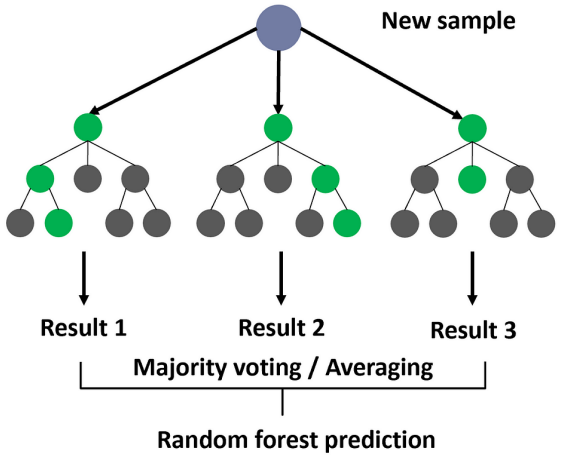
Random Forest



https://dbrang.tistory.com/1520

마이닝 알고리즘 (머신러닝 모델)

Random Forest

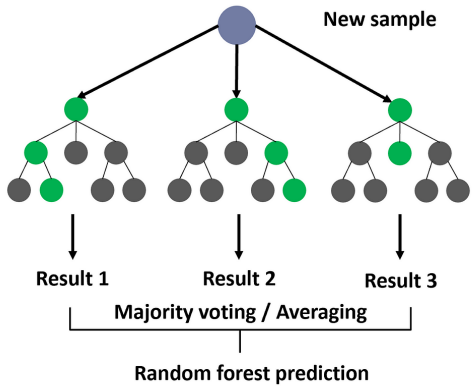


- 포레스트 : 숲(Forest)
- 결정 트리는 트리 : 나무(Tree)
- 나무가 모여 숲을 이룸 = 결정 트리(Decision Tree)가 모여 랜덤 포레스트(Random Forest)를 구성

- 결정 트리 단점 : 훈련 데이터에 **오버피팅**이 되는 경향
  여러 개의 결정 트리를 통해(랜덤 포레스트) 오버피팅 극복

마이닝 알고리즘 (머신러닝 모델)

Random Forest



If **<u>30개의 Feature</u>**를 기반으로 하나의 결정 트리를 만든다면

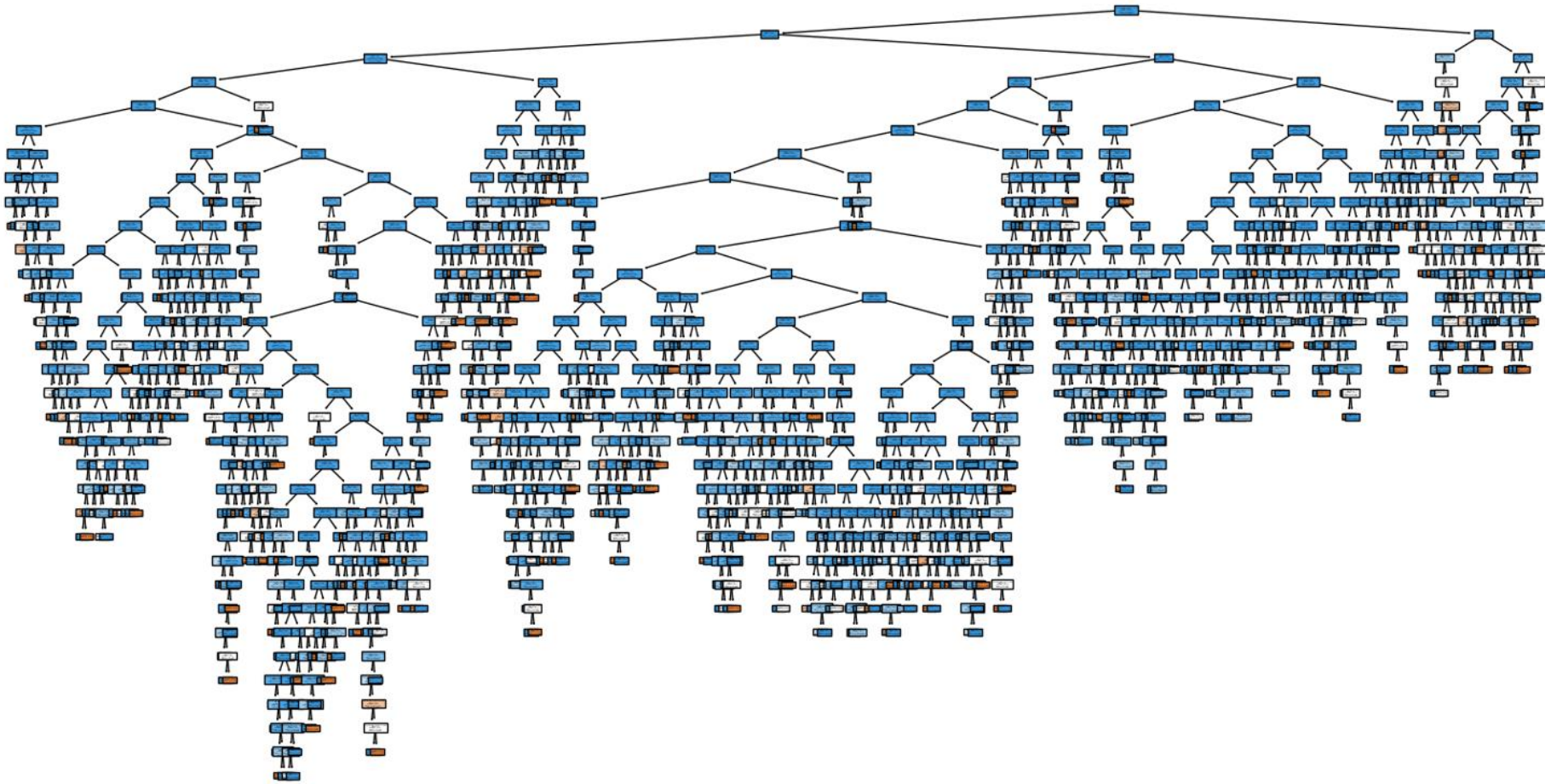
- 많은 가지 -> 오버피팅
- 30개의 Feature 중 랜덤으로 5개의 Feature만 선택, 하나의 결정 트리 구성
- 또 30개 중 랜덤으로 5개의 Feature를 선택해서 또 다른 결정 트리를 구성
- 반복... -> 여러 개의 결정 트리 구성.
- 결정 트리 하나마다 예측 값 산출
- 여러 결정 트리들이 내린 예측 값들 중 가장 많이 나온 값을 최종 예측값 선정
- 다수결의 원칙 -> **앙상블(Ensemble) (의견을 통합하거나 여러 가지 결과를 합치는 방식)**

요약 :

- 하나의 거대한 결정 트리를 만드는 것이 아니라 여러 개의 작은 결정 트리를 만듦
- 여러 개의 작은 결정 트리가 예측한 값들 중 가장 많은 값(분류일 경우)
  혹은 평균값(회귀일 경우)을 최종 예측 값으로 결정

마이닝 알고리즘 (머신러닝 모델)

Random Forest



마이닝 알고리즘 (머신러닝 모델)

Random Forest

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns; sns.set() # For a nicer plot style

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Random Forest Classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42) # 100 trees in the forest

# Fit the model with the training data
rf_clf.fit(X_train, y_train)

# Predict the test dataset
predictions = rf_clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)
print(f"Accuracy: {accuracy*100:.2f}%")

# Feature Importance
feature_importances = rf_clf.feature_importances_
features = iris.feature_names
plt.figure(figsize=(10, 6))
plt.barh(range(len(features)), feature_importances, align='center')
plt.yticks(range(len(features)), features)
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.title('Feature Importance in Random Forest Classifier')
plt.show()
exit()
```

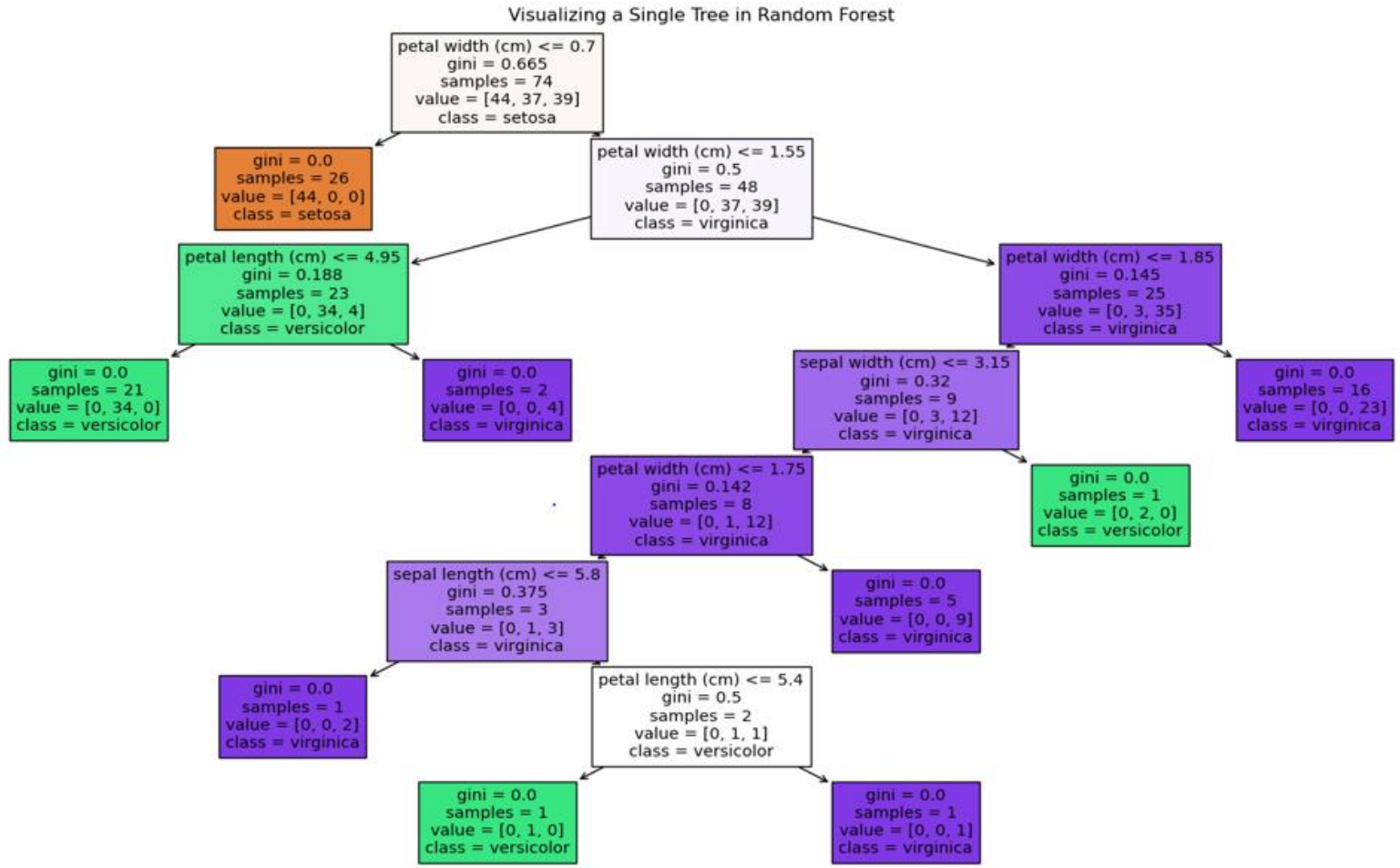
- 실습

  2.05_RF.ipynb

n_estimators: 랜덤 포레스트 안의 결정 트리 개수
max_features: 무작위로 선택할 Feature의 개수

마이닝 알고리즘 (머신러닝 모델)

Random Forest



THANK YOU