R 교육 세미나

Tobig's 8기 황다솔

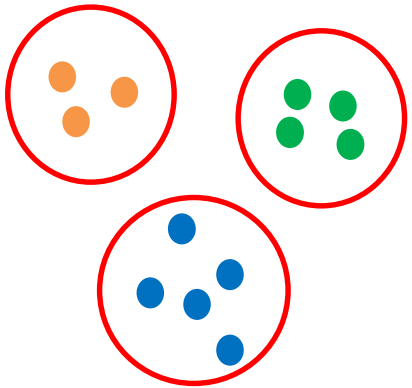
K-Nearest Neighbor

KNN 알고리즘

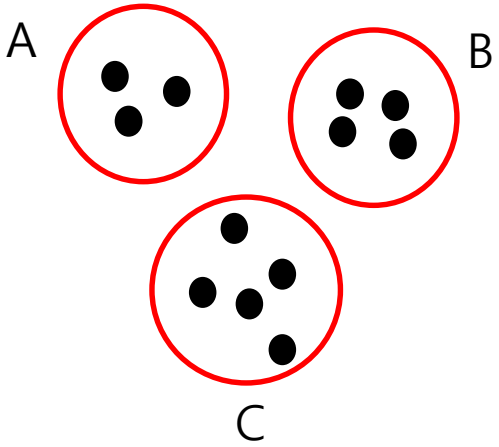
Contents

Classification vs Clustering

Classification



Clustering



분류(Classification) : 소속 집단을 **알고** 비슷한 집단으로 묶는 방법
즉, **Label 이 있는 data**를 분류 **Supervised Learning** → 지도 학습

군집(Clustering) : 소속 집단을 **모르는** 상태에서 비슷한 집단으로 묶는 방법
즉, **Label 이 없는 data**를 군집화 **Unsupervised Learning** → 비지도 학습

K Nearest Neighbor

K-NN 이란?

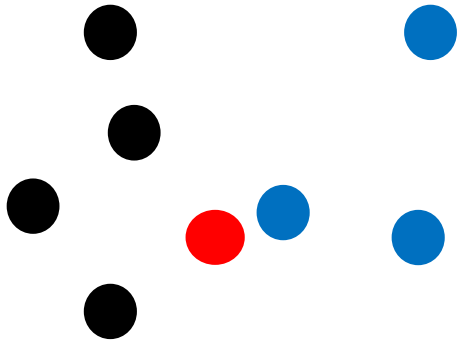
K-NN 이란?

K 개의

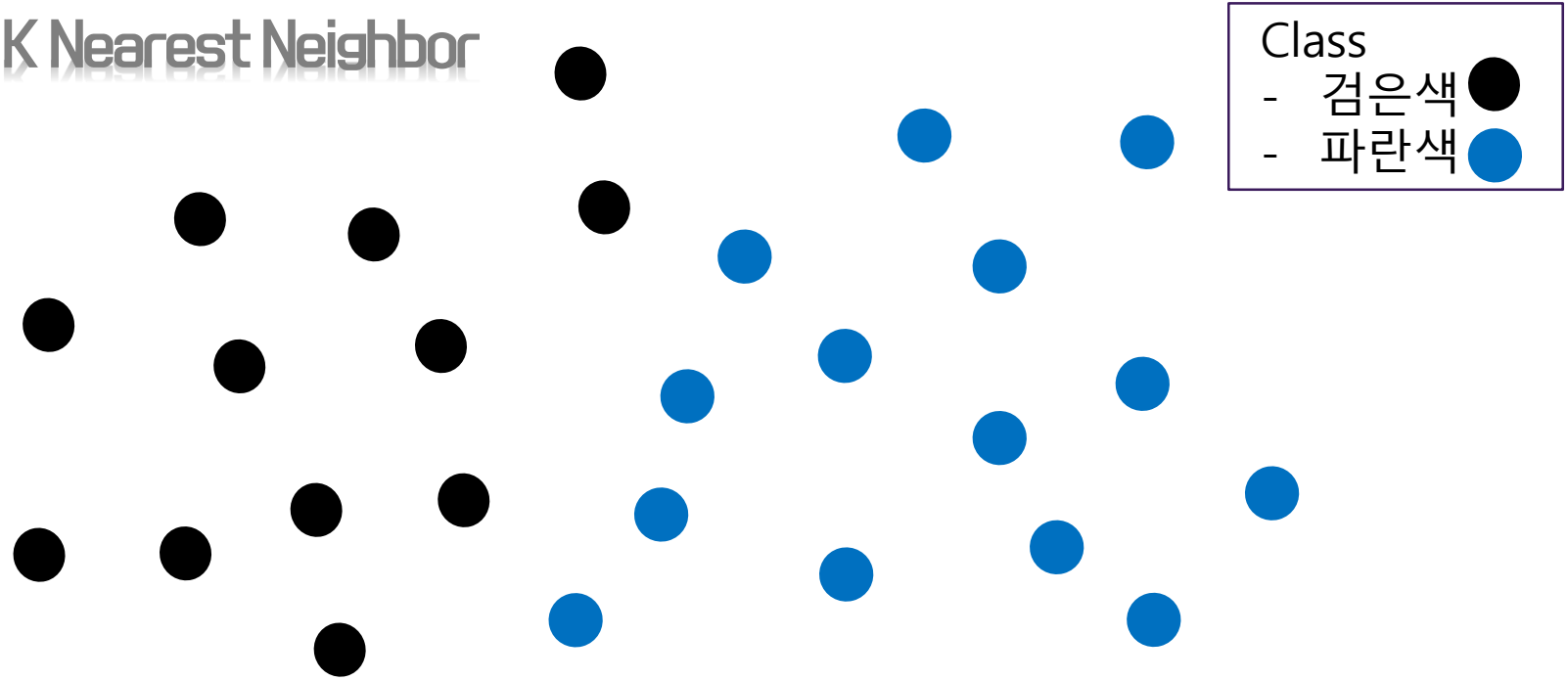
Nearest (가까운)

Neighbor (이웃)

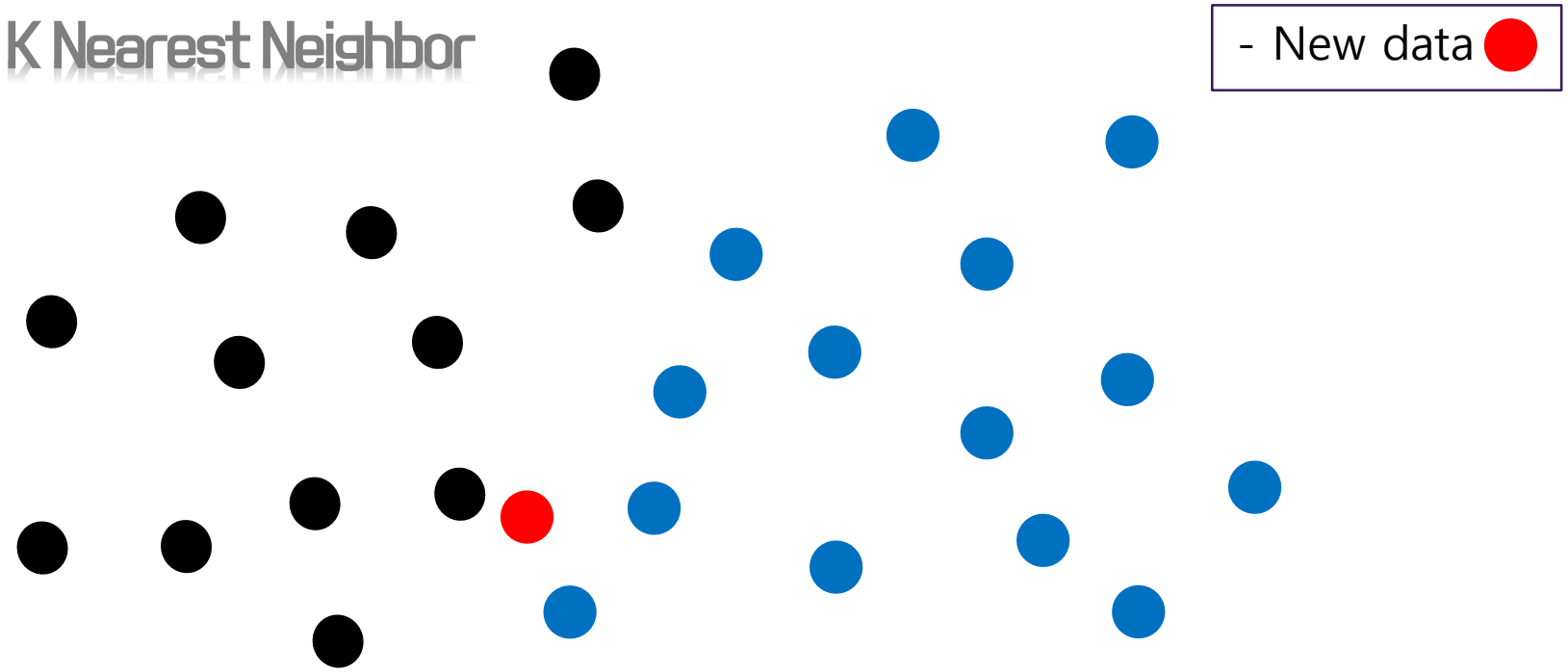
→



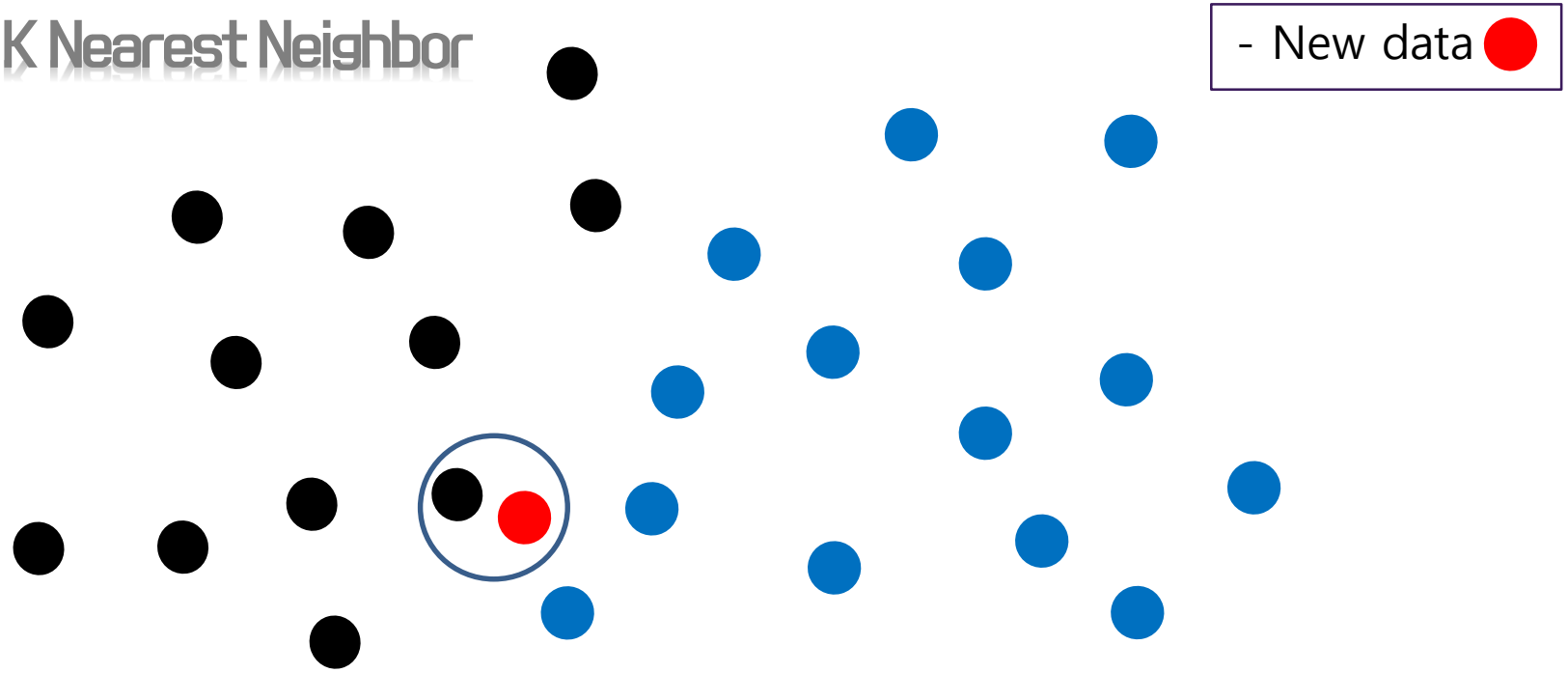
K Nearest Neighbor



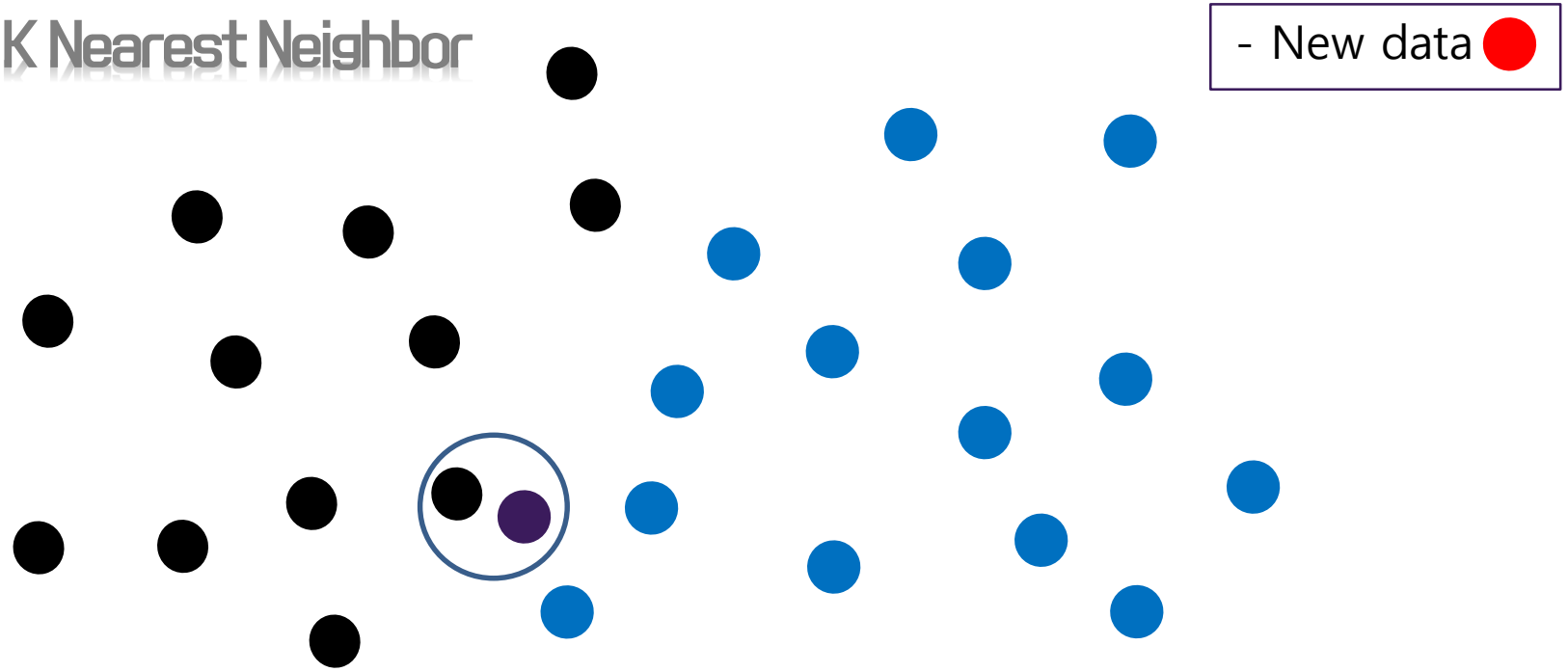
K Nearest Neighbor



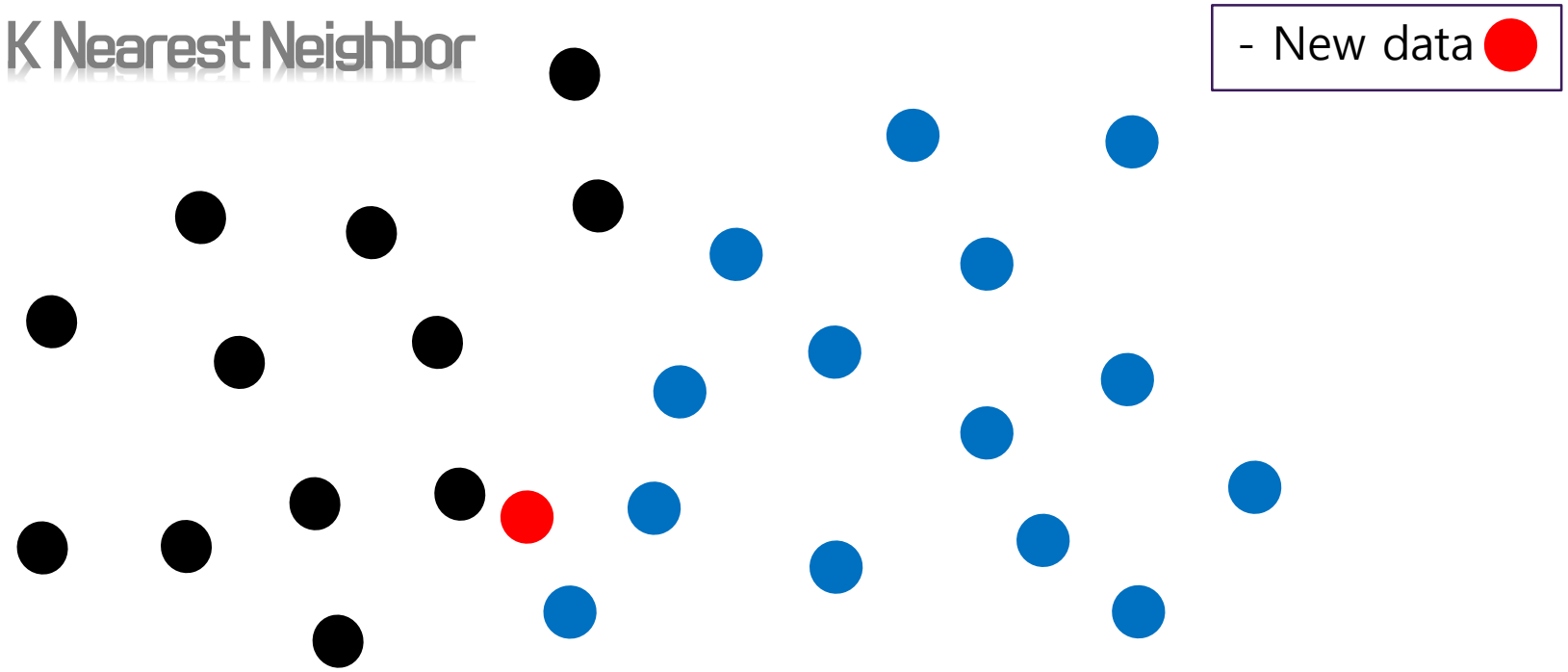
K Nearest Neighbor



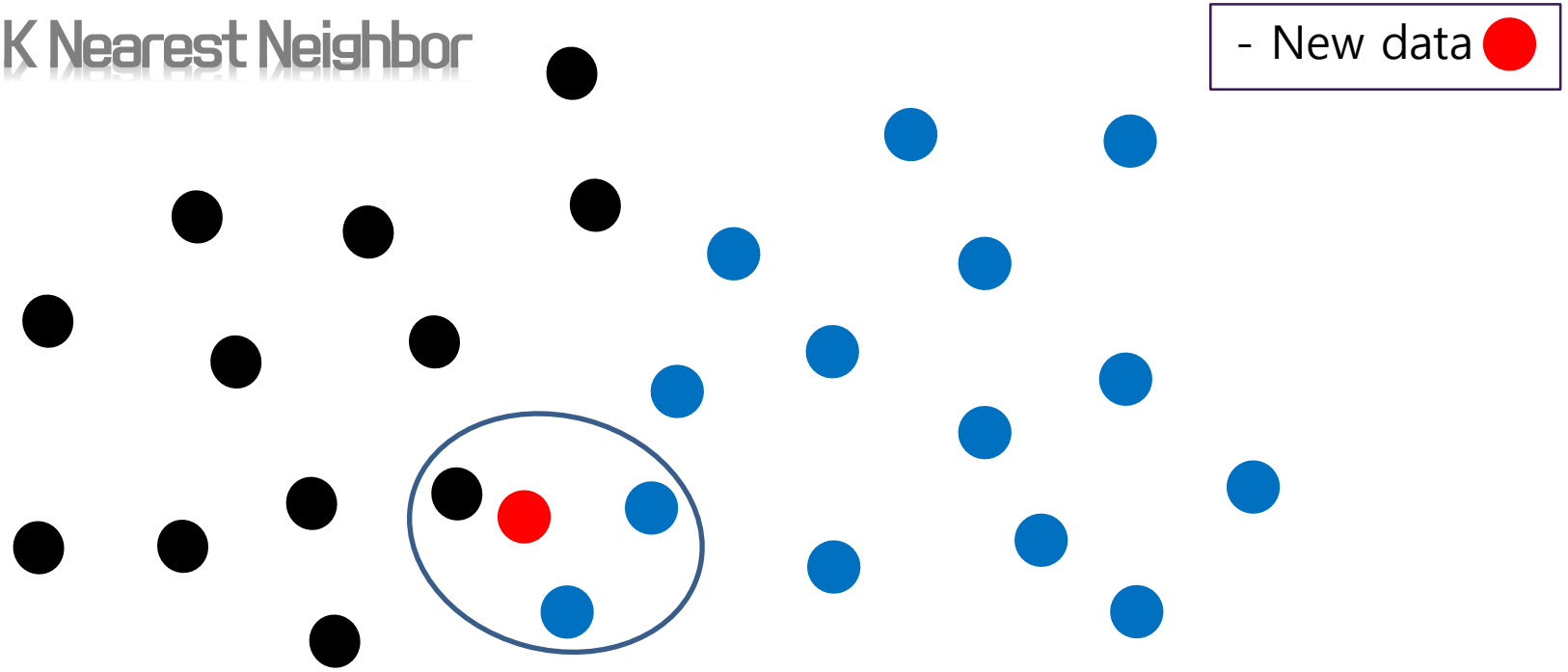
K Nearest Neighbor



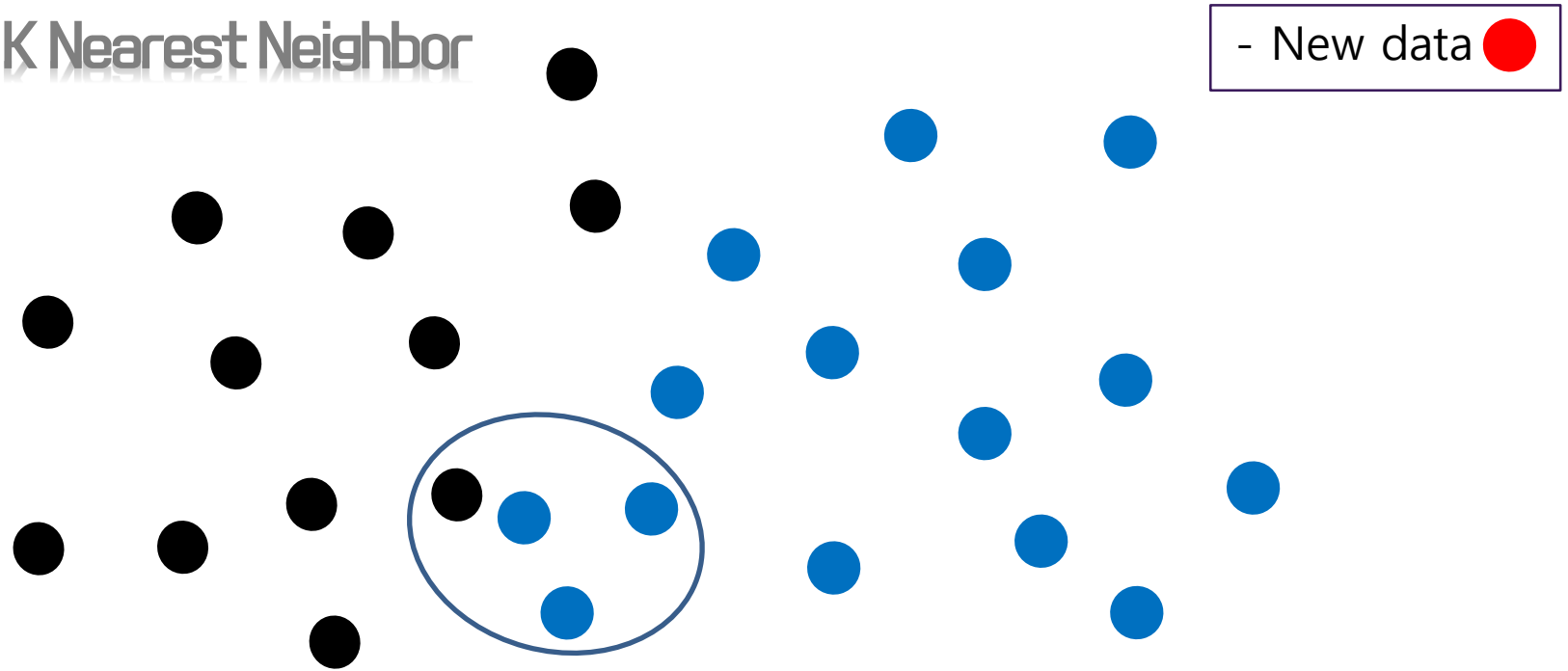
K Nearest Neighbor



K Nearest Neighbor



K Nearest Neighbor



k-NN : distance를 이용해 k 개의 가까운 이웃을 보자

> 기존 데이터 중 가장 유사한 k개의 이웃 데이터들을 이용해서 새로운 데이터를 예측하는 방법

[수치형 데이터] : k개 데이터의 평균

[명목형 데이터]
: k개의 데이터에서 majority voting
혹은 가중치를 추가해 선택

[Hyperparameter]
K 와 Distacne 구하는 방법

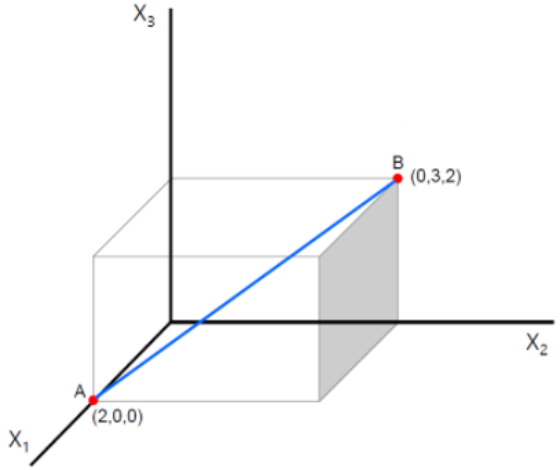
Hyperparameter

Distance(거리) & K

Distance 를 구하는 방법 ?

Euclidean Distance

$$X = (x_1, x_2, \ldots, x_n)$$
$$Y = (y_1, y_2, \ldots, y_n)$$
$$d_{(X,Y)} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

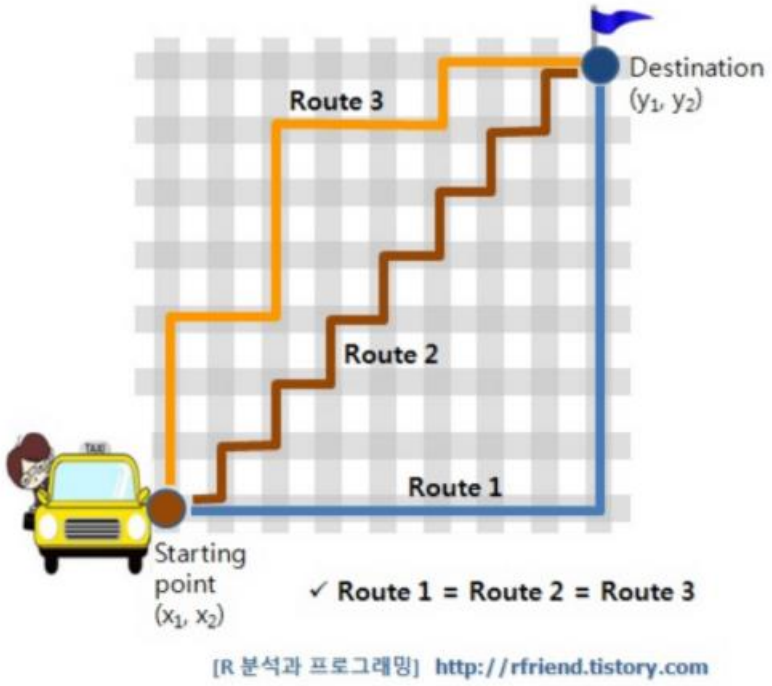


$$d_{(A,B)} = \sqrt{(0-2)^2 + (3-0)^2 + (2-0)^2} = \sqrt{17}$$

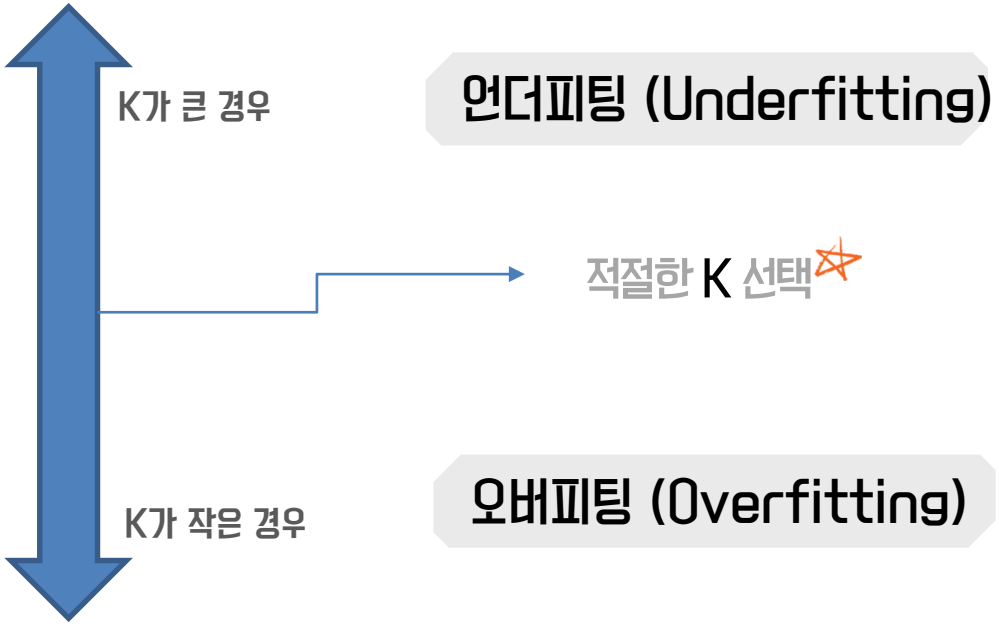
Distance 를 구하는 방법 ?

Manhattan Distance

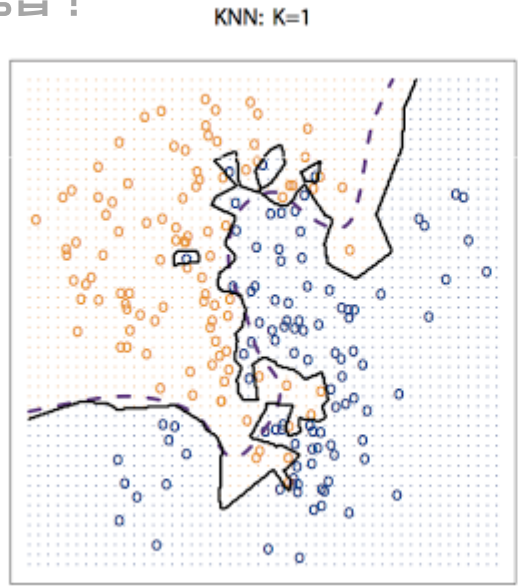
$$d_{Manhattan(X,Y)} = \sum_{i=1}^{n} |x_i - y_i|$$



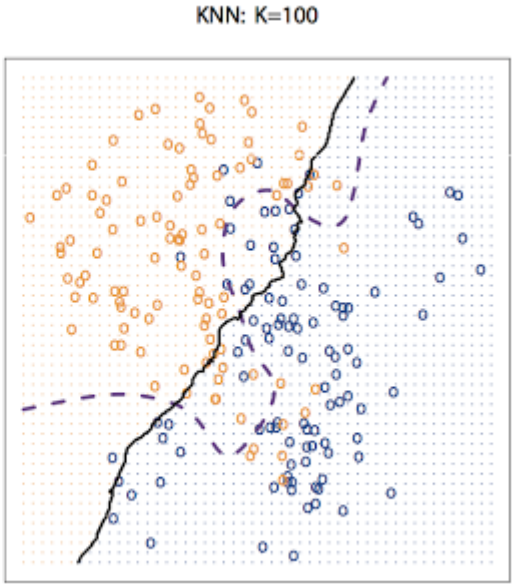
K 를 구하는 방법 ?



K 를 구하는 방법 ?

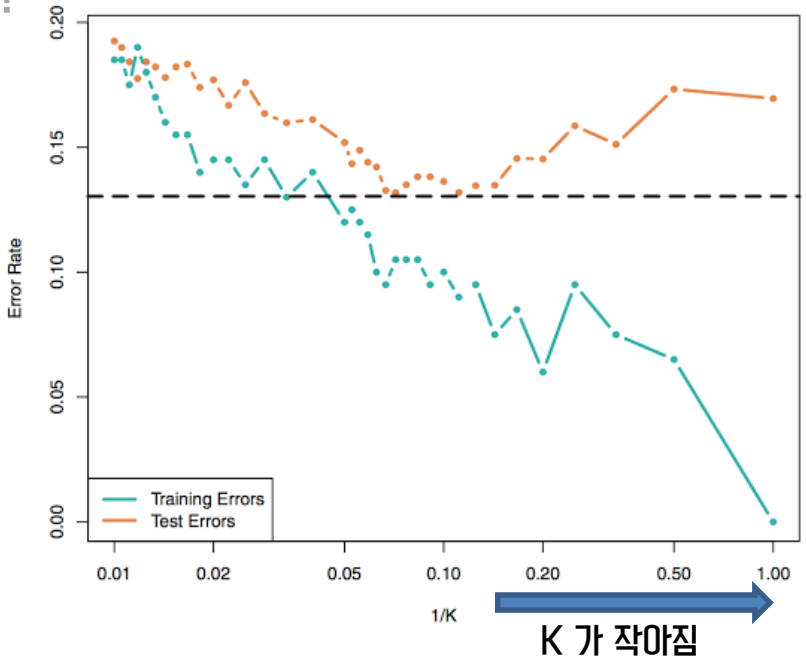


오버피팅 (Overfitting)



언더피팅 (Underfitting)

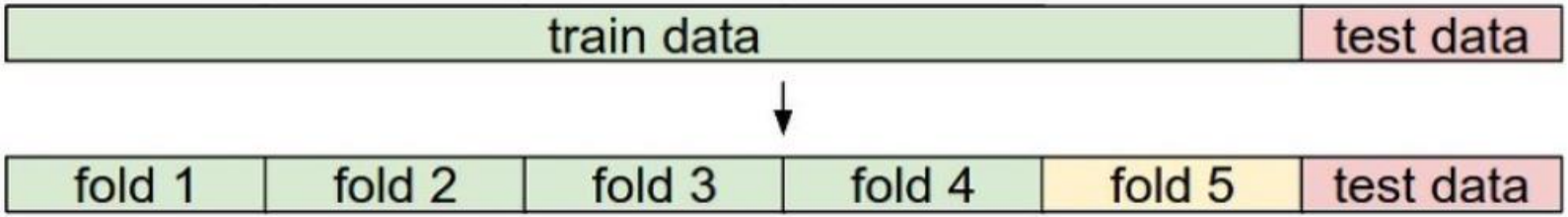
K 를 구하는 방법 ?



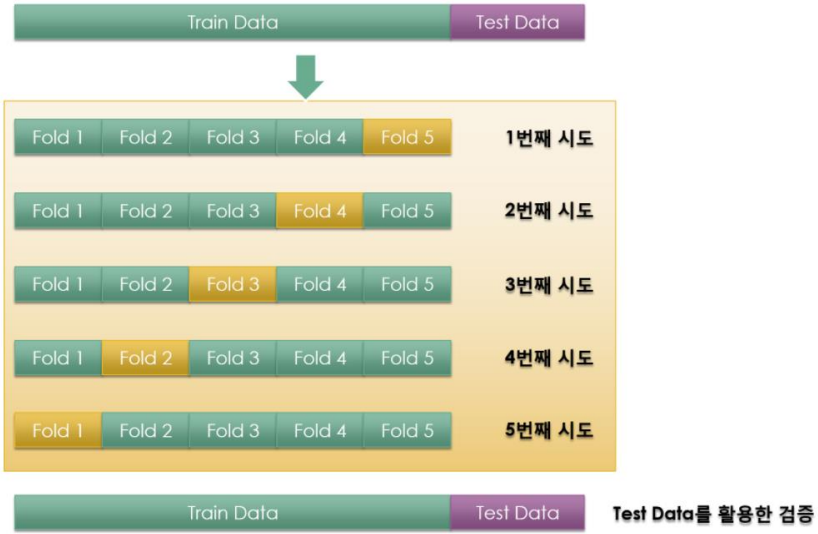
K 가 작아짐

교차 검증법 (Cross-Validation)

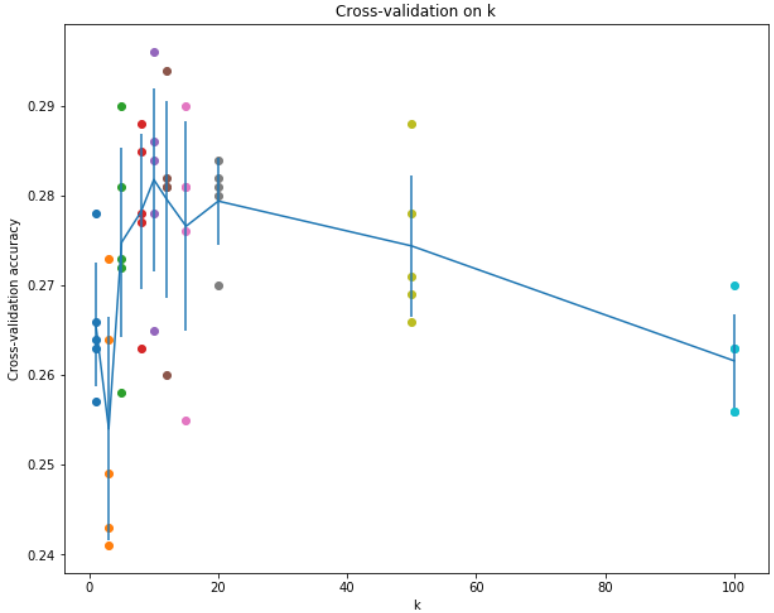
K-Fold Cross-Validation



교차 검증법 (Cross-Validation)



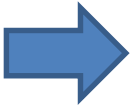
교차 검증법 (Cross-Validation)



더 나아가서

1. Feature scaling

이웃	특성1	특성2	특성3
N1	0.8	400	0.5
N2	12	134,000	0.9
N3	0	20,000	1.1
N4	67	32,000	0.1



정규화

이웃	특성1	특성2	특성3
N1	0.012	0	0.4
N2	0.179	1	0.8
N3	0	0.147	1
N4	1	0.237	0

$$\text{distance}(N3, N4) = \sqrt{(0-67)^2 + (20{,}000-32{,}000)^2 + (1.1-0.1)^2}$$

$$= \sqrt{(4489 + 144{,}000 + 1.0)}$$

$$\text{distance}(N3, N4) = \sqrt{(0-1)^2 + (0.147-0.237)^2 + (1-0)^2}$$

$$= \sqrt{(1 + 0.0081 + 1)}$$

1. Feature scaling

가장 많이 쓰이는 건 min-max normalization이다.

$$X_{new} = \frac{X - min(X)}{max(X) - min(X)}$$ 로 표현된다. 0~1 사이의 값을 가지게 된다.

z-score normalization 또한 많이 쓰이며

$$X_{new} = \frac{X - \mu}{\sigma}$$ 로 표현된다. 통계에서 자주 보는 방법이다.

2. Weight

새로운 데이터 x가 들어왔을 때 (k = 4)

클래스	이웃	특성1	특성2	특성3	거리	유사도	가중치
A	N1	0.012	0	0.4	1	1	0.44
B	N2	0.179	1	0.8	2	0.5	0.22
C	N3	0	0.147	1	3	0.33	0.15
B	N4	1	0.237	0	4	0.25	0.11

$$\text{유사도} = \frac{1}{\text{거리}}$$

$$\text{N1의 가중치} = \frac{N1\text{의 유사도}}{\text{모든 이웃의 유사도 합}}$$

정리

1. 데이터 scale 확인 -> 정규화,표준화
2. 거리 구하기
3. (가중치를 줄 경우에) 가중치 구하기
4. 적절한 k 선택 (CV)
5. 가까운 k 개의 이웃을 보고 다수결, 가중치로 최종 분류

KNN의 장점

- 이해하고 구현하기 쉬움
- 학습 시간이 빠르다

KNN의 단점

- 메모리가 많이 필요함
- 분류 속도가 느림
- 명목형 변수나 결측치 처리가 따로 필요함

들어주셔서 감사합니다.