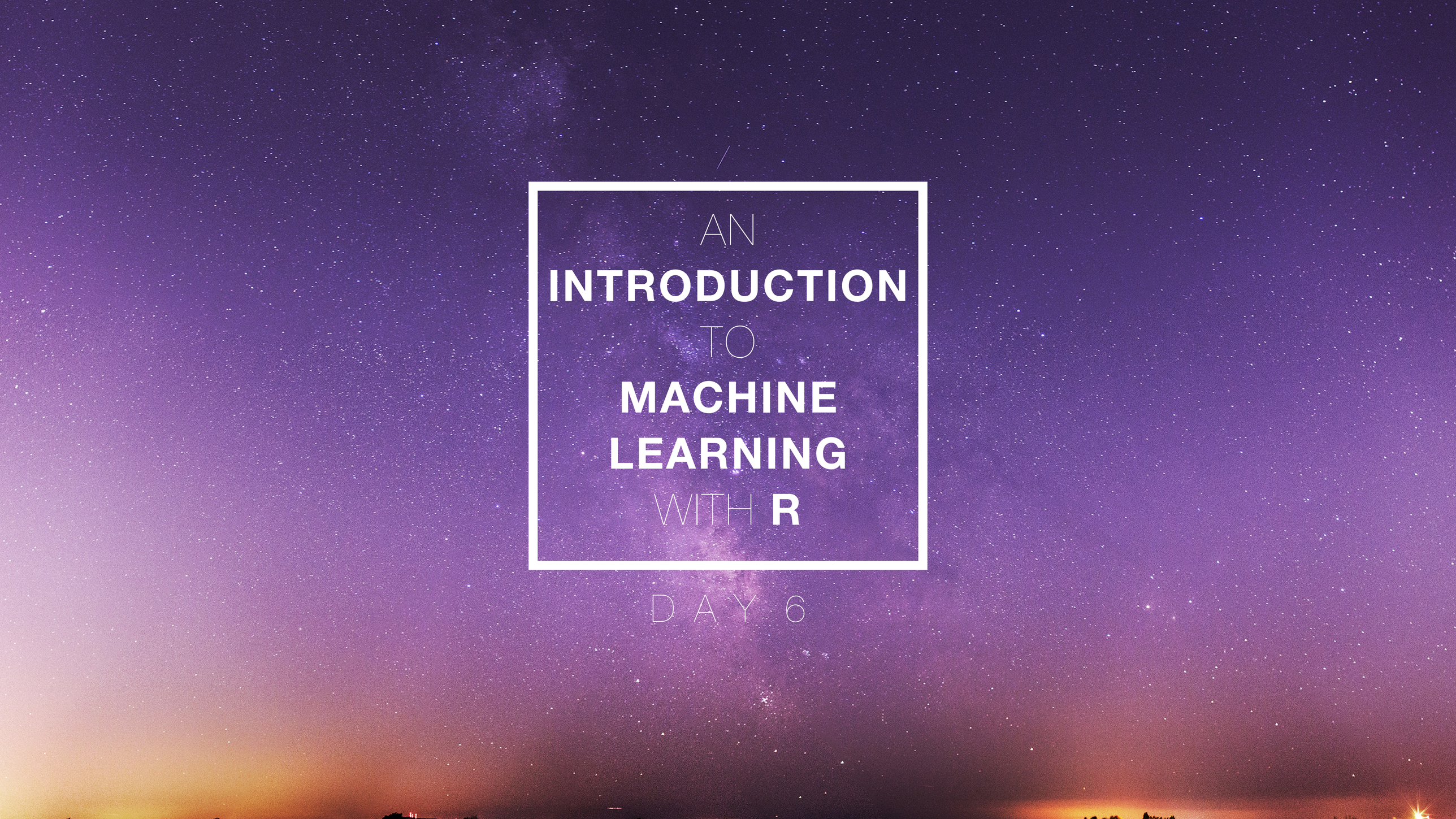

AN
INTRODUCTION
TO
MACHINE
LEARNING
WITH R
DAY 6

DAY 6

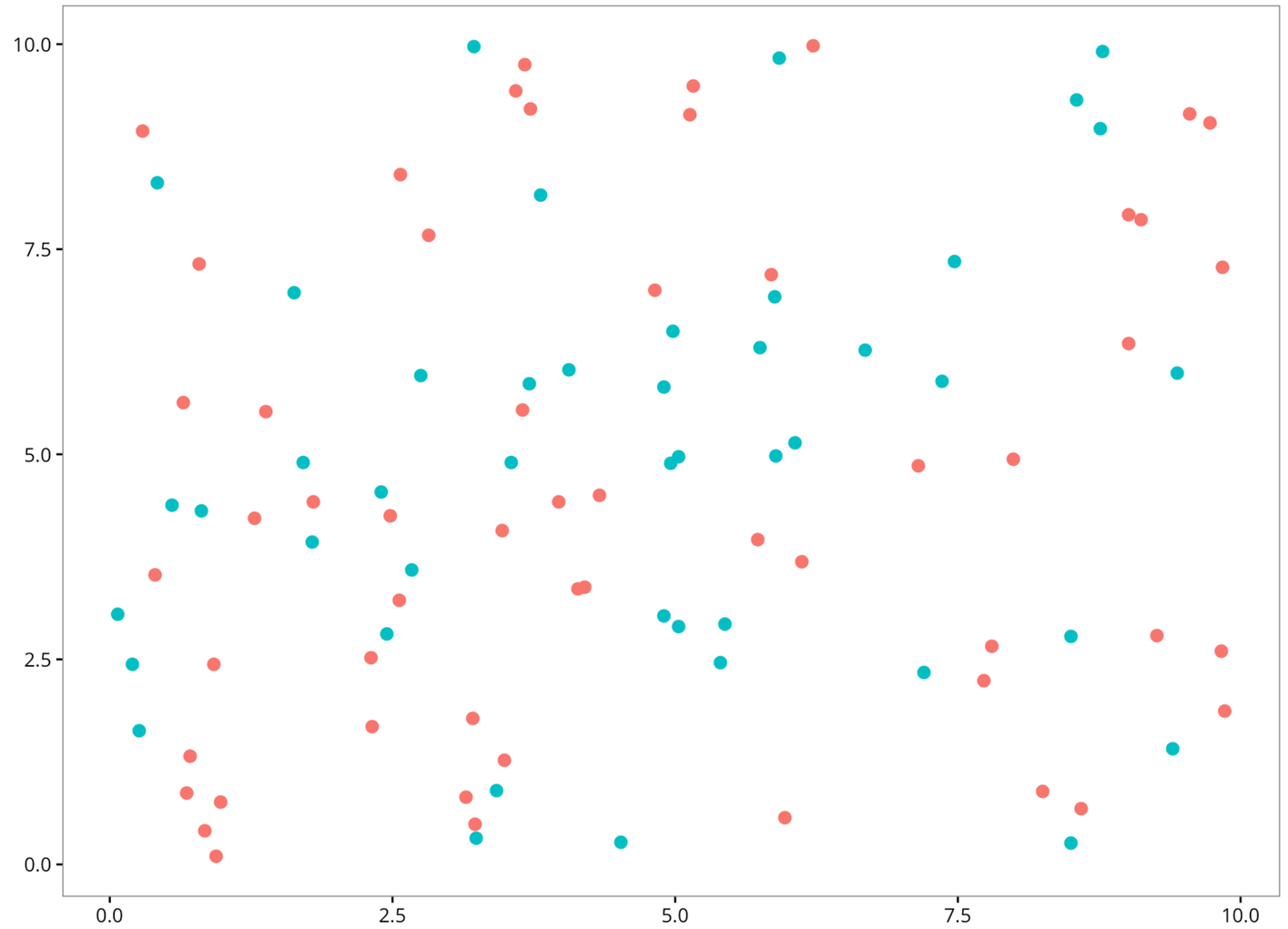
k Nearest Neighbor

k-Nearest Neighbor

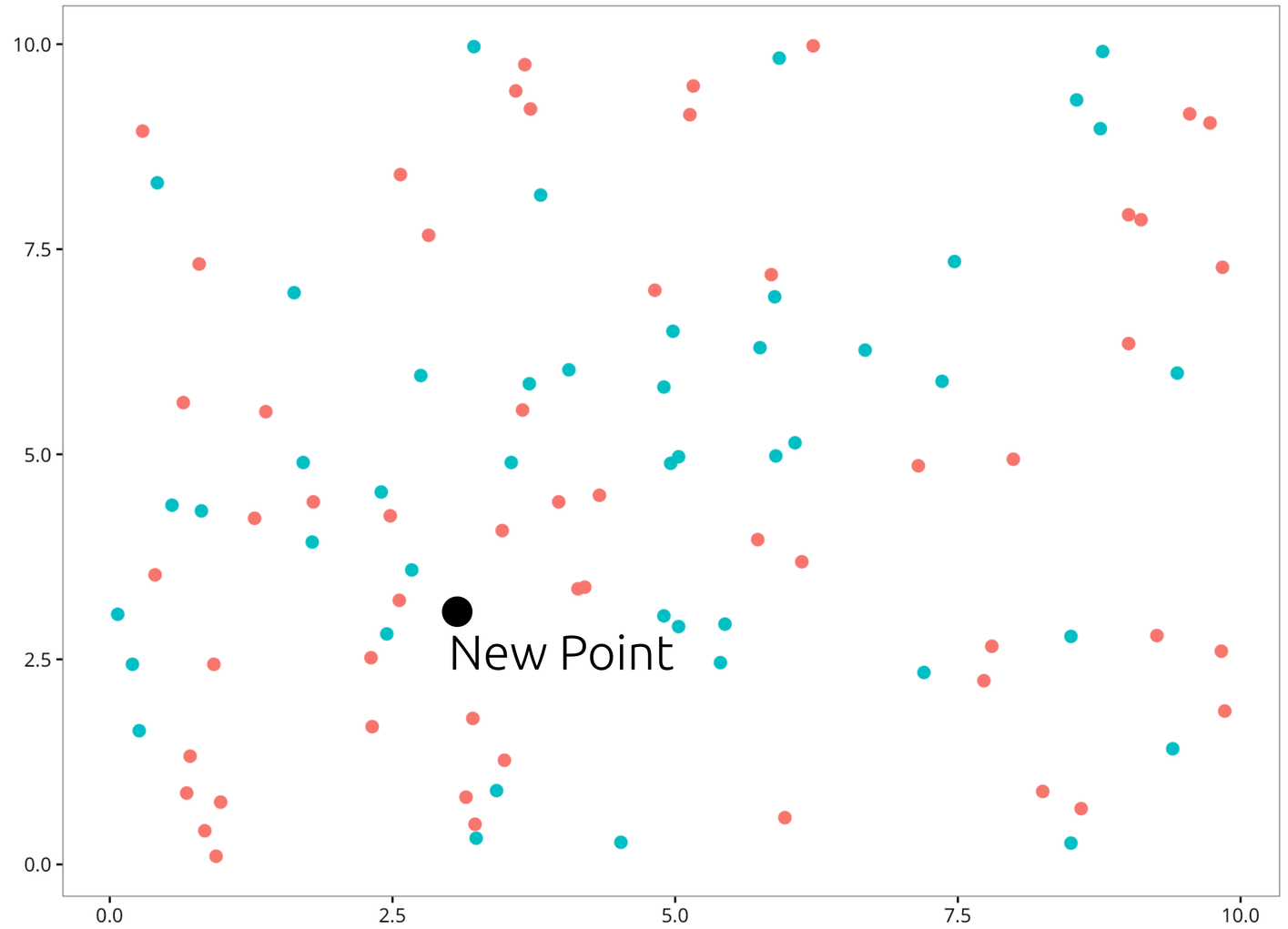
Nearest Neighbor?

가장 가까운 이웃?

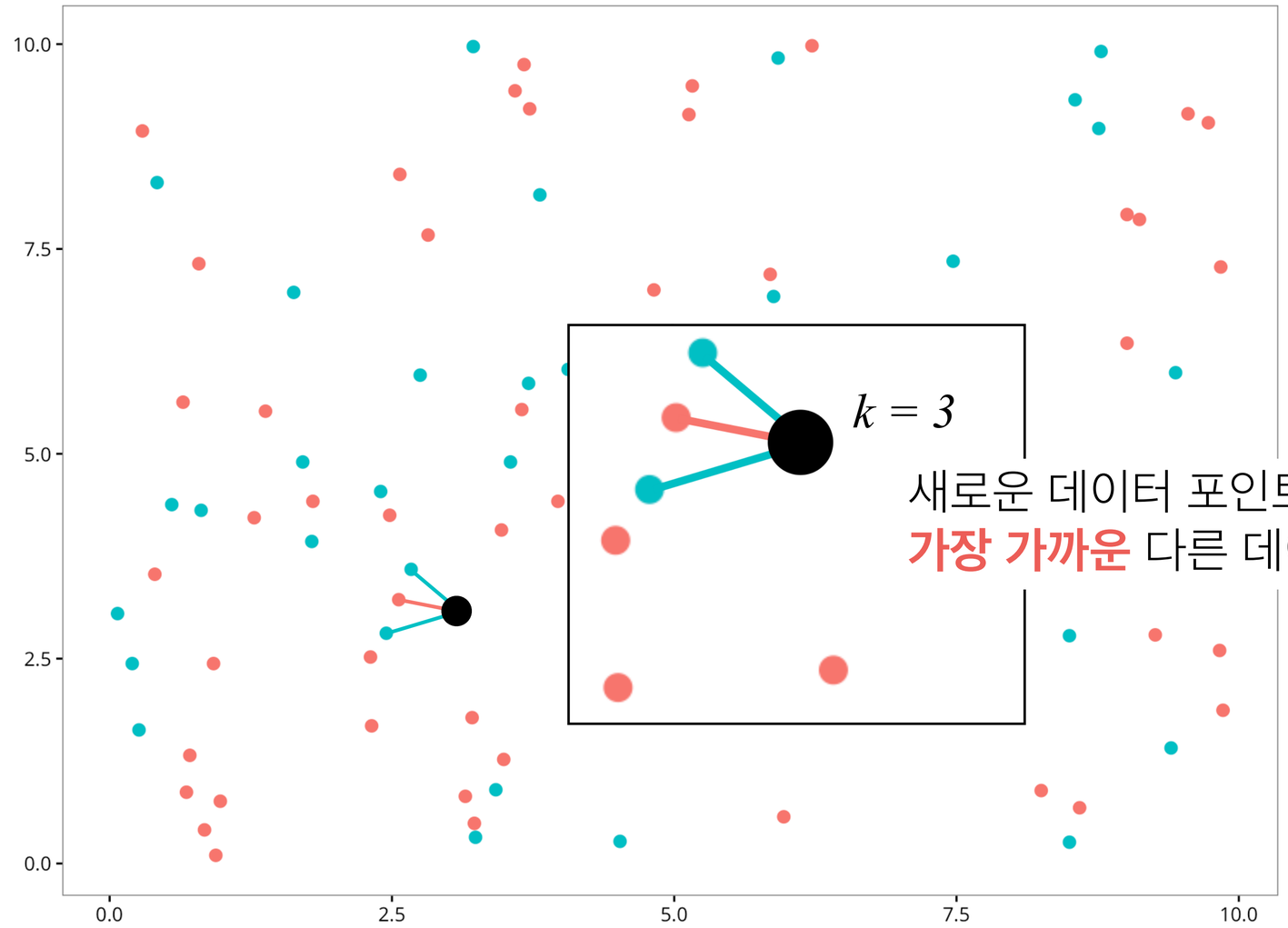
k-Nearest Neighbor



k-Nearest Neighbor

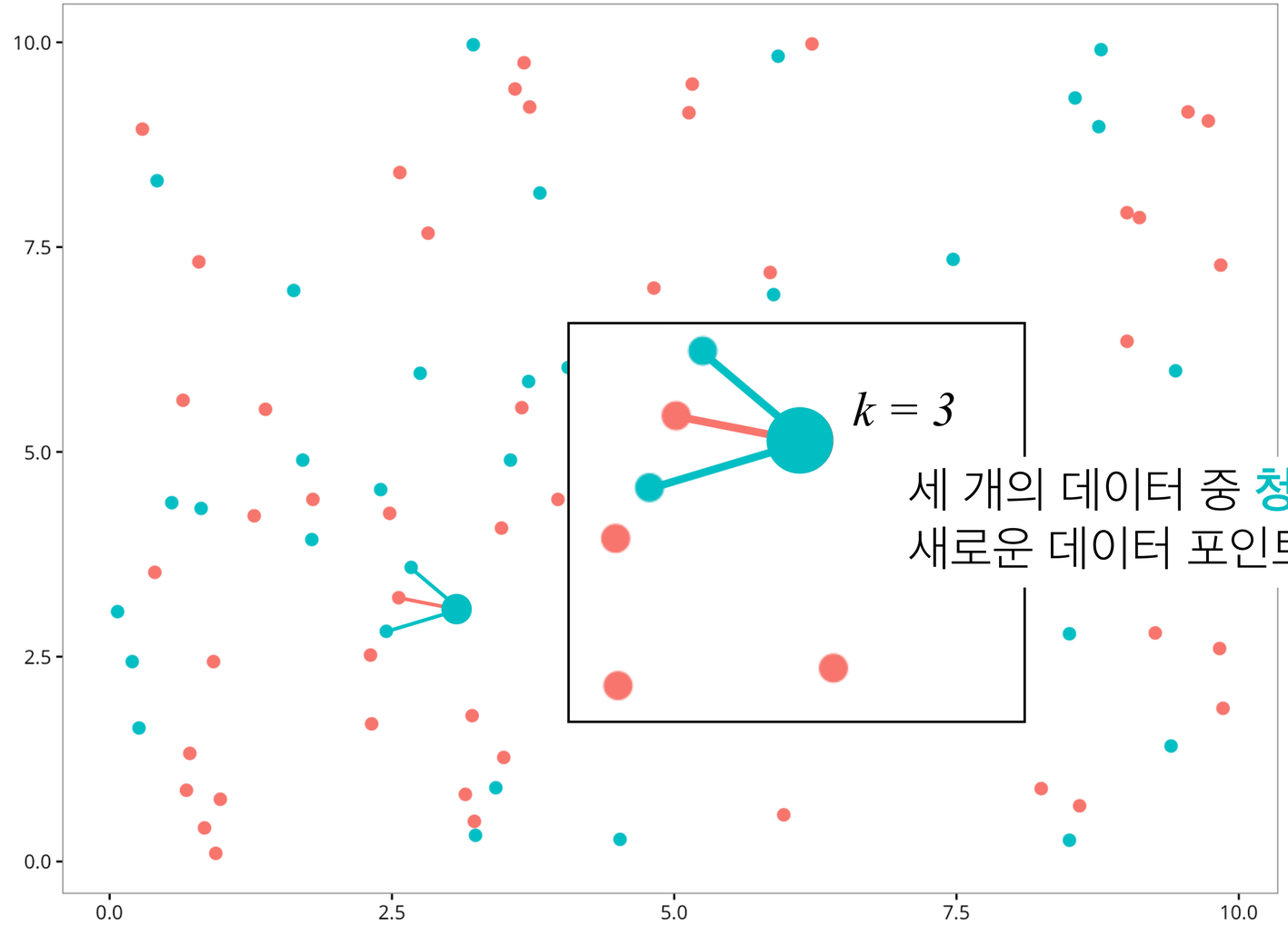


k-Nearest Neighbor



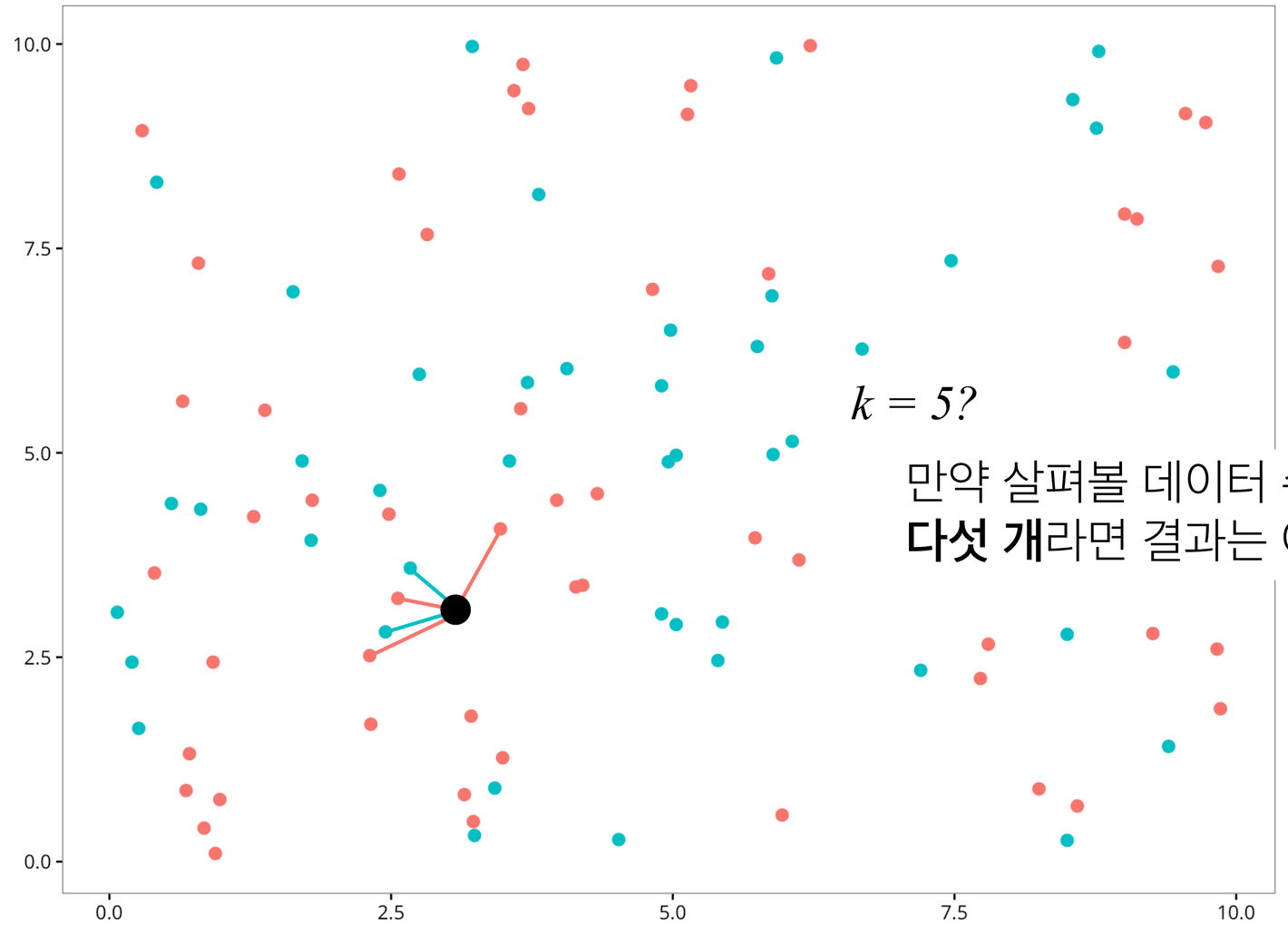
새로운 데이터 포인트에서
가장 가까운 다른 데이터 포인트들을 찾는다.

k-Nearest Neighbor



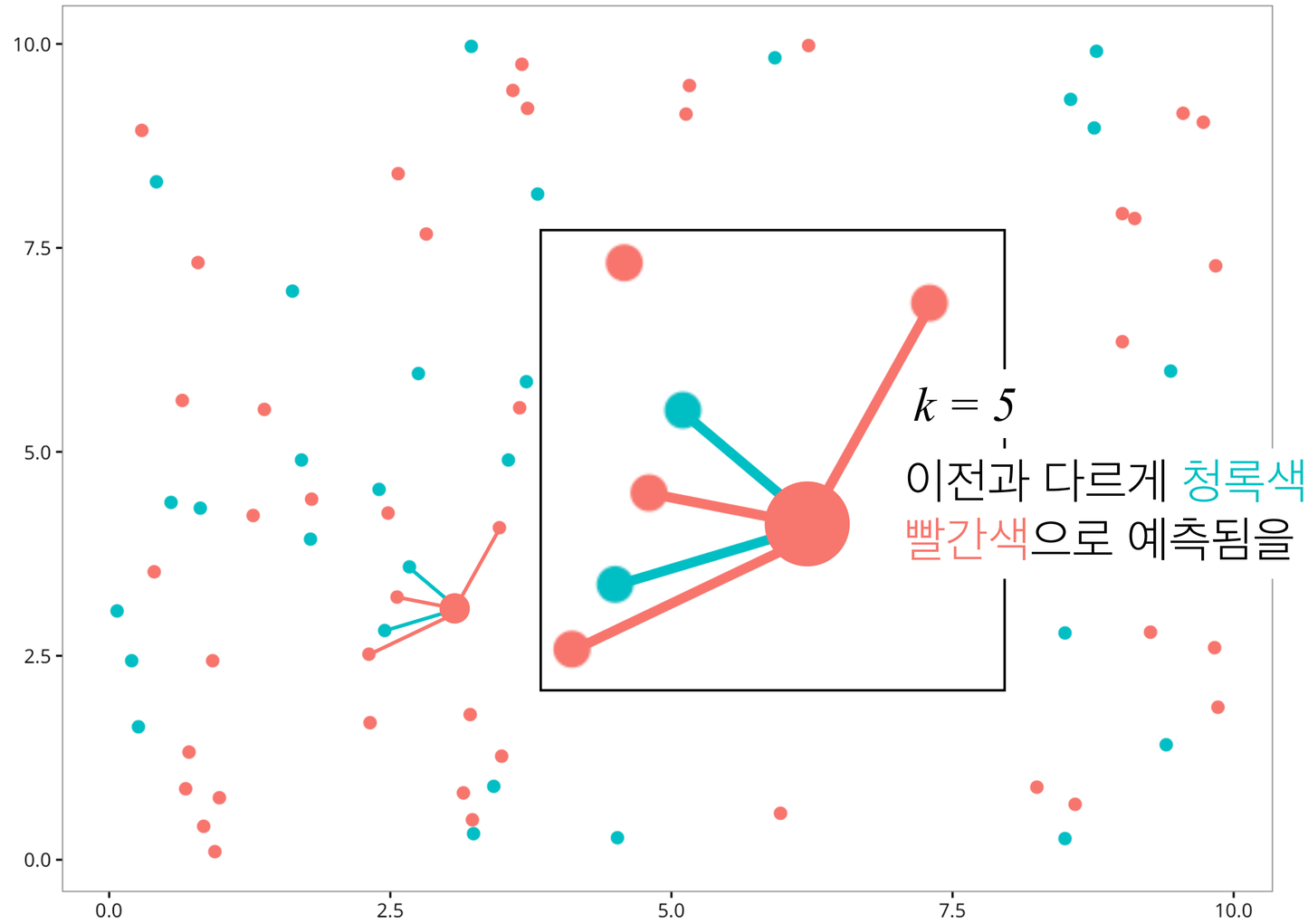
세 개의 데이터 중 **청록색이 더 많으므로**
새로운 데이터 포인트는 청록색으로 예측된다.

k-Nearest Neighbor



만약 살펴볼 데이터 수가 **세 개**가 아니라
다섯 개라면 결과는 어떻게 될까?

k-Nearest Neighbor



이전과 다르게 청록색이 아닌
빨간색으로 예측됨을 알 수 있다.

k-Nearest Neighbor

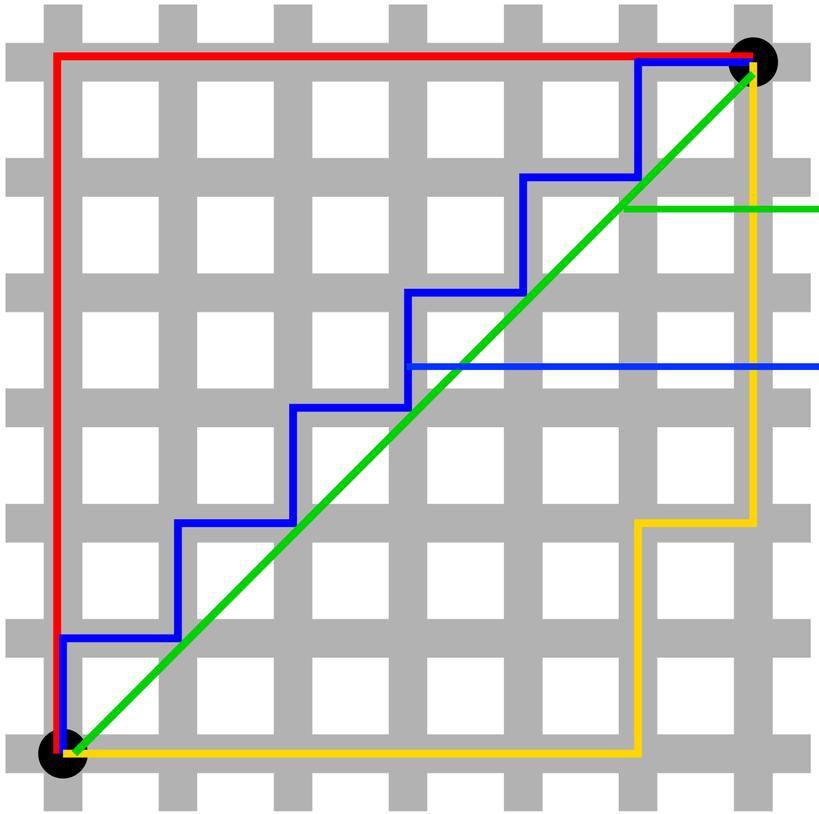
WE NEED TO DETERMINE

Distance

k

k-Nearest Neighbor

Distance



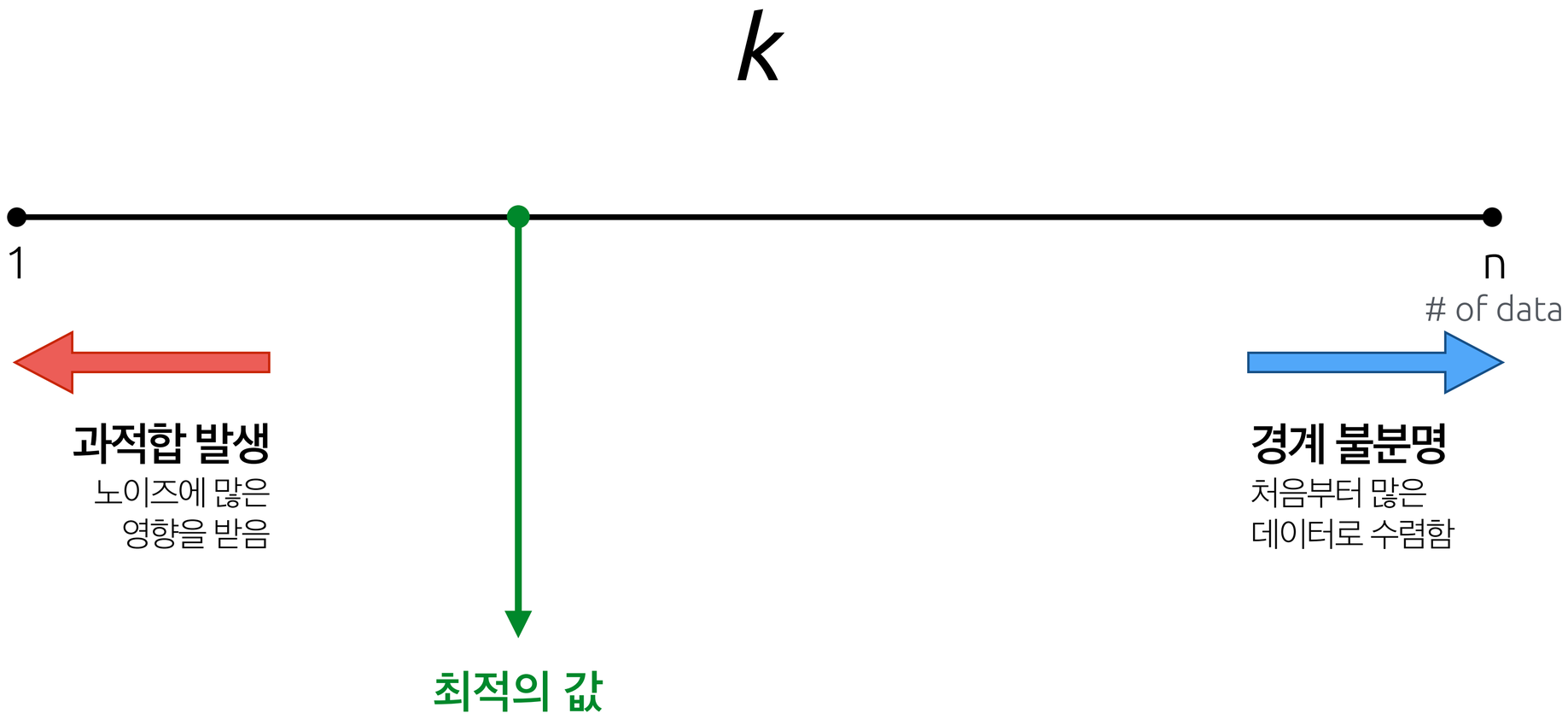
Euclidean Distance

Manhattan Distance

⋮

Cosine Similarity
Pearson Correlation

k-Nearest Neighbor



k-Nearest Neighbor

LAZY LEARNING

실습

k-최근접 이웃

BREAST CANCER DIAGNOSTIC

Feature Scaling

(Data Normalization)

Feature scaling is a method used to standardize the range of independent variables or features of data.

— Wikipedia

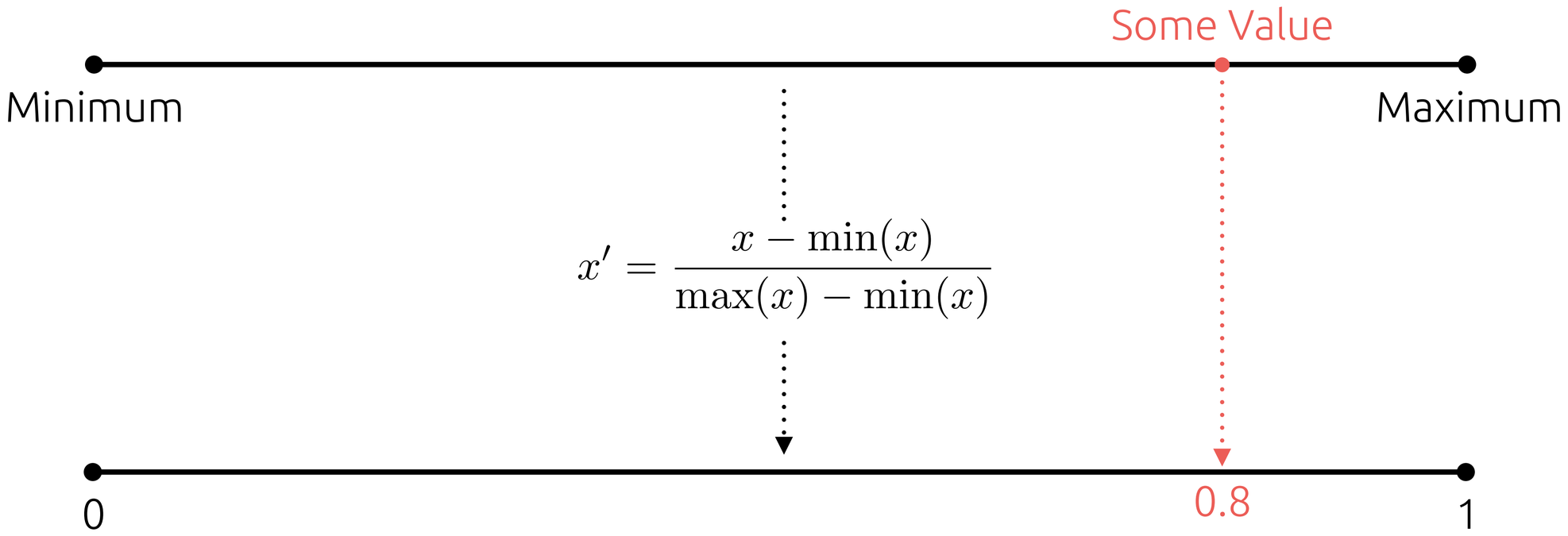
거리를 계산할 때,

어떤 Feature의 범위가 **눈에 띄게** 크다면

해당 Feature는 전체 변수에 대해서 **불필요하게 큰 영향력**을 가질 수 있다.

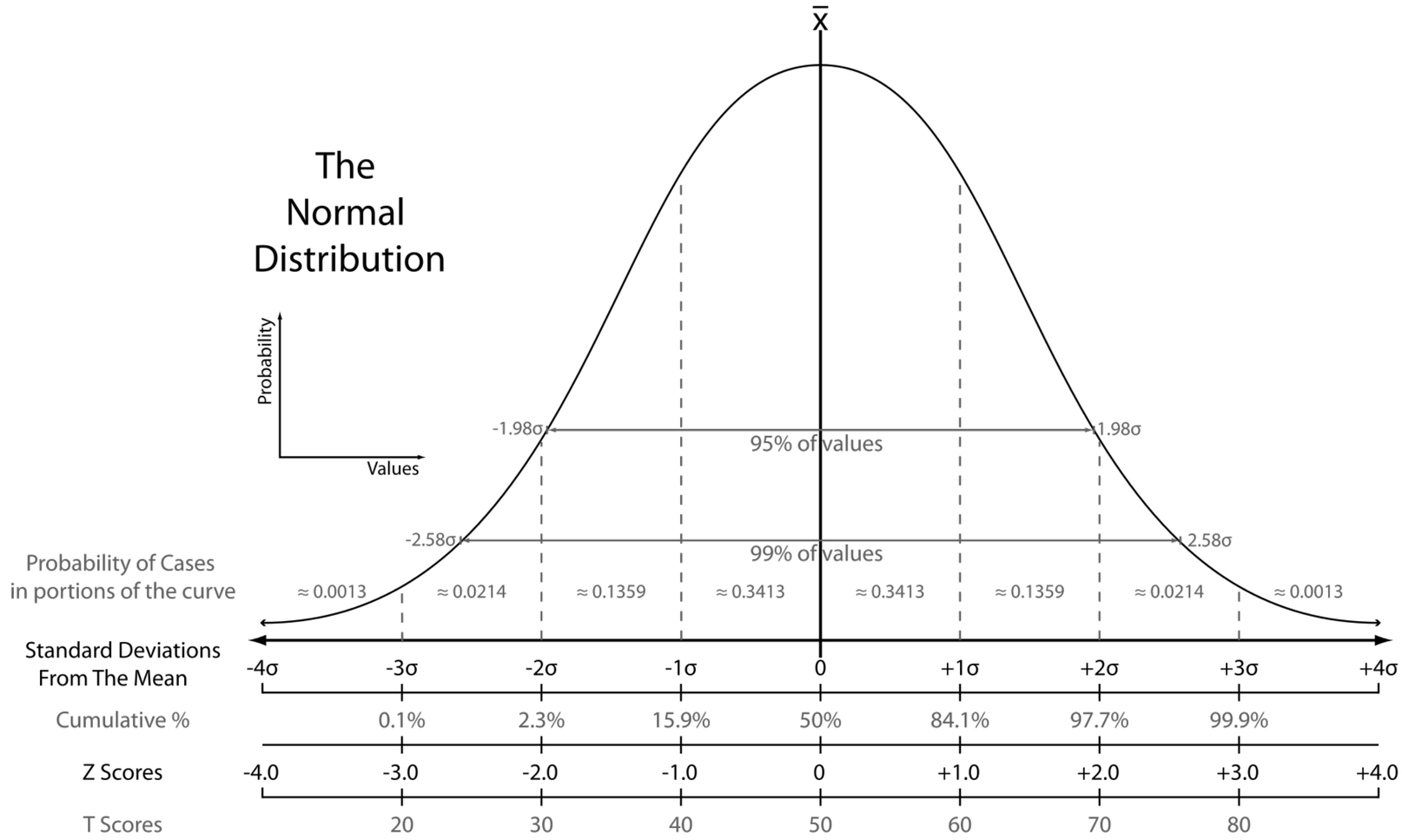
Min-Max Normalization

(Rescaling)



Standardization

(z Score)



		Actual Condition	
		Actual **Positive**	Actual **Negative**
Predicted Condition	Predicted **Positive**	**TP**	**FP**
	Predicted **Negative**	**FN**	**TN**

$$Accuracy = (TP + TN) / (P + N) = (TP + TN) / (TP + FP + FN + TN)$$

$$Sensitivity = TP / P = TP / (TP + FN)$$

$$Specificity = TN / N = TN / (FP + TN)$$

$$Positive\ Prediction\ Value = TP / (TP + FP)$$

$$Negative\ Prediction\ Value = TN / (TN + FN)$$


DAY 6
Clustering

Clustering

주어진 데이터를 구분할 수 있는 클래스(class)에 대한 **지식이 없는 상태**에서
각 데이터들의 **유사도에 근거**하여 데이터를 구분하는 방법

Unsupervised Learning

Clustering

WHY

WE USE THIS ALGORITHM

k-Means

k-Medoids

DBSCAN

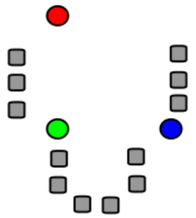
k-Means Algorithm

주어진 데이터를 k 개의 클러스터로 묶는 알고리즘

평균값을 중심으로

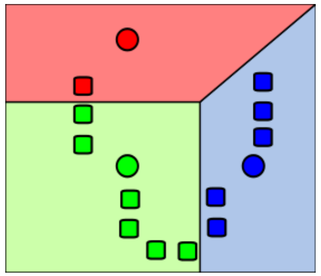
k-Means Algorithm

Step 1



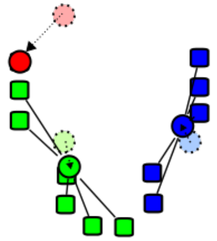
k 값을 설정하고
랜덤으로 뽑힌 데이터에서
평균값(중심)을 내서
표시한다.

Step 2



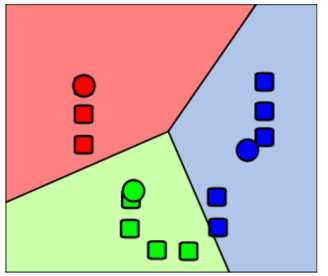
데이터 오브젝트들은
가장 가까운 평균값을
중심으로 묶인다.

Step 3



k 개의 클러스터의
중심점을 기준으로
평균값이 재조정된다.

Step 4



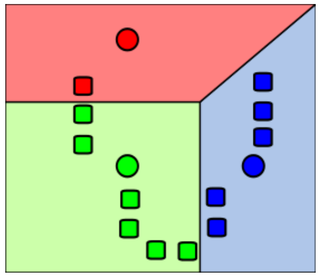
데이터들이 소속된
클러스터가 바뀌지
않을 때 까지 Step 2와
Step 3를 반복한다.

k-Means Algorithm

Step 1

k 값을 설정하고
랜덤으로 뽑힌 데이터에서
평균값(중심)을 내서
표시한다.

Step 2



데이터 오브젝트들은
가장 가까운 평균값을
중심으로 묶인다.

Step 3

k 개의 클러스터의
중심점을 기준으로
평균값이 재조정된다.

Step 4

데이터들이 소속된
클러스터가 바뀌지
않을 때 까지 Step 2와
Step 3를 반복한다.

데이터가 묶일 때,
아래의 값이 최소가 되도록 묶인다.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \mathrm{dist}(\mathbf{p}, \mathbf{c_i})^2$$

The sum of squared error

k-Means Algorithm

Scalable

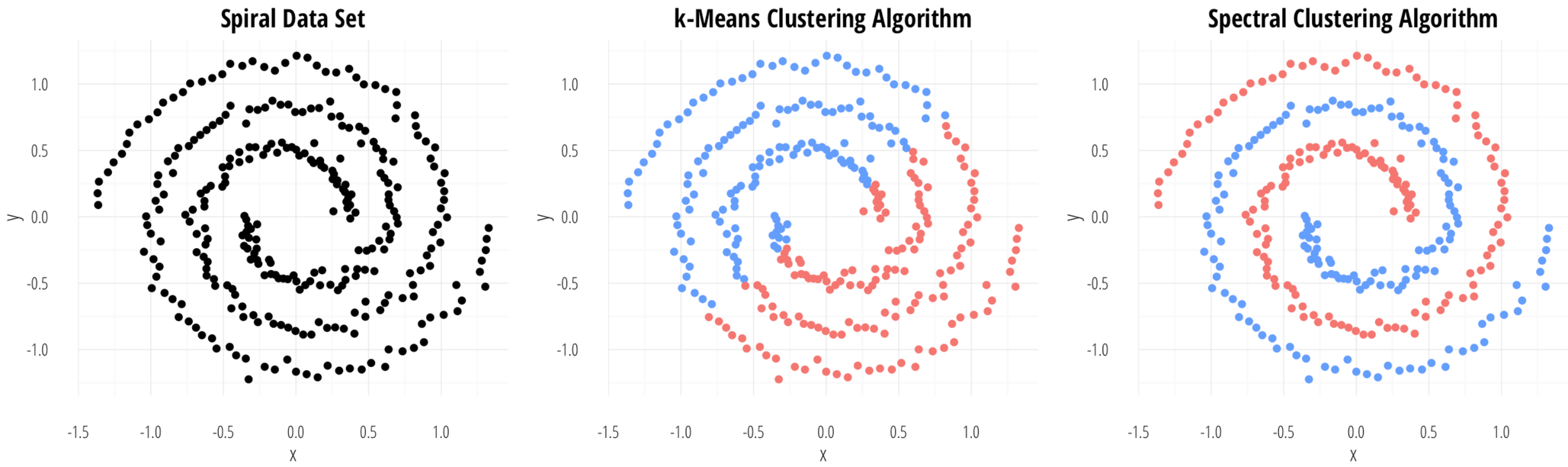
Efficient in Large Data Set

Select the appropriate value of *k* is challenging

Sensitive to noise and outlier

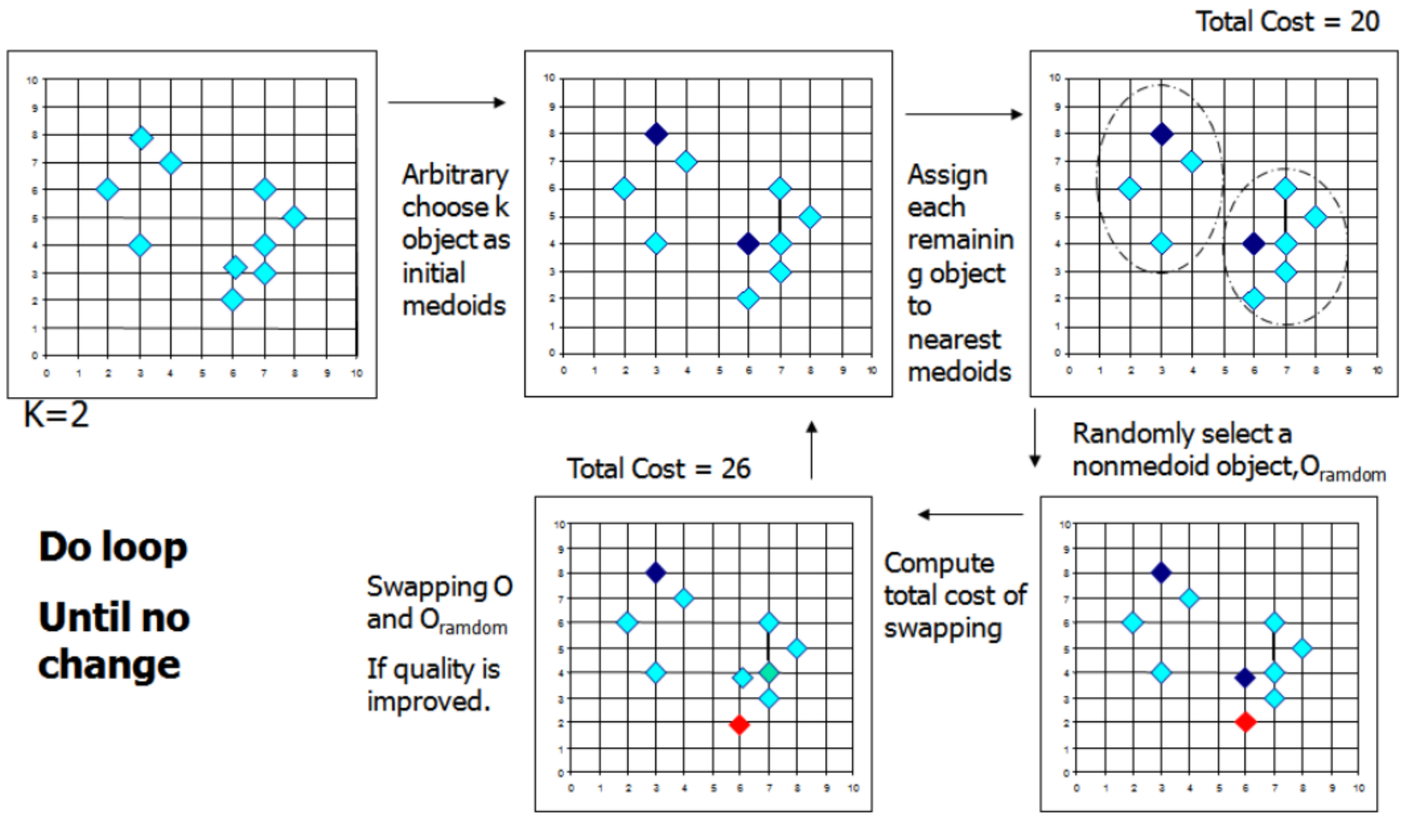
Only for Convex-shaped clusters

k-Means Algorithm



모든 데이터에 대해서 사용할 수는 없다

k-Medoids Algorithm

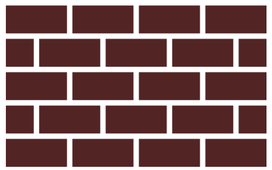


데이터가 묶일 때,
아래의 값이 최소가 되도록 묶인다.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \mathrm{dist}(\mathbf{p}, \mathbf{o_i})^2$$

Absolute-error criterion

k-Medoids Algorithm



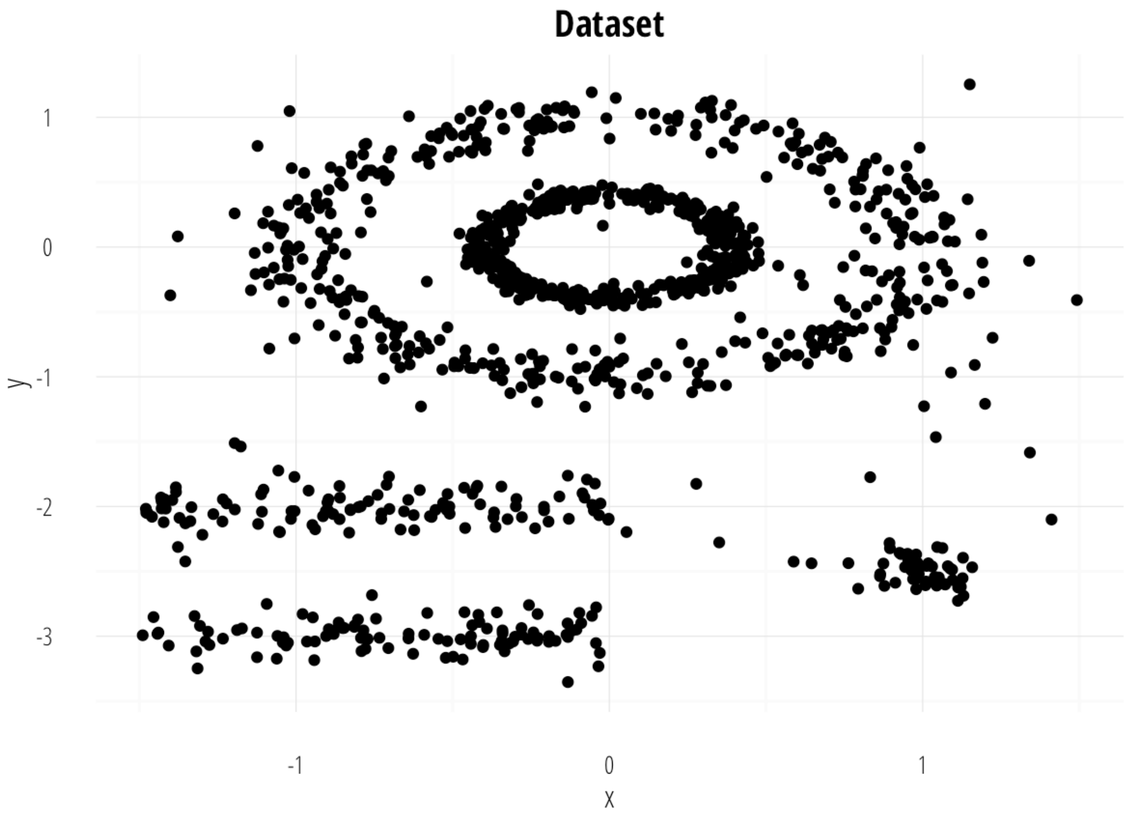
Robust to noise and outlier

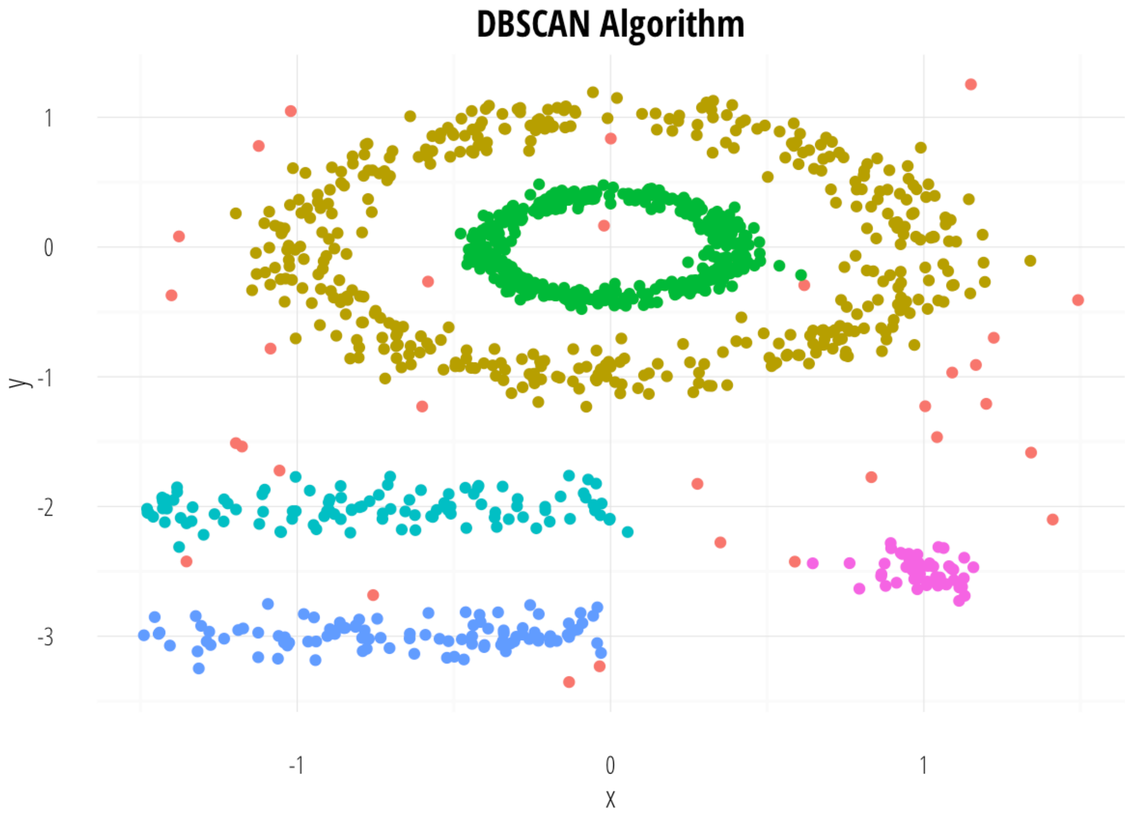
Select the appropriate value of *k* is challenging

Computationally Expensive

Only for Convex-shaped clusters

DBSCAN





과제

당뇨병 진단

k 최근접 이웃을 사용하여
당뇨병 여부를 예측하자.

트레이닝 데이터 : `diabetes_train`
테스트 데이터 : `diabetes_test`

과제

당뇨병 진단

k 최근접 이웃을 사용하여
당뇨병 여부를 예측하자.

1. 데이터 전처리가 필요할 경우 시도할 것
   (Max-Min Normalize, Z-scaling)
2. 최적의 k값을 찾을 것
3. 해당 모델을 **`confusionMatrix()`**를 이용하여
   모델의 퍼포먼스를 확인할 것

과제

당뇨병 진단

k 최근접 이웃을 사용하여
당뇨병 여부를 예측하자.

트레이닝 데이터 : diabetes_train
테스트 데이터 : diabetes_test

npreg : 임신 횟수
glu : 글루코스 부하 검사 결과
bp : 확장기 혈압
skin : 삼두근 피부 주름 두께
vmi : BMI 수치
bed : 가족력을 기반으로 한 당뇨병 발생 확률
age : 나이
type : 당뇨병 여부


THX :)