

20기 정규세션

ToBig's 19기 최다희

KNN & Clustering

20기 정규세션

ToBig's 19기 최다희

KNN

Contents

Unit 01 | KNN

Unit 02 | KNN Hyperparameter

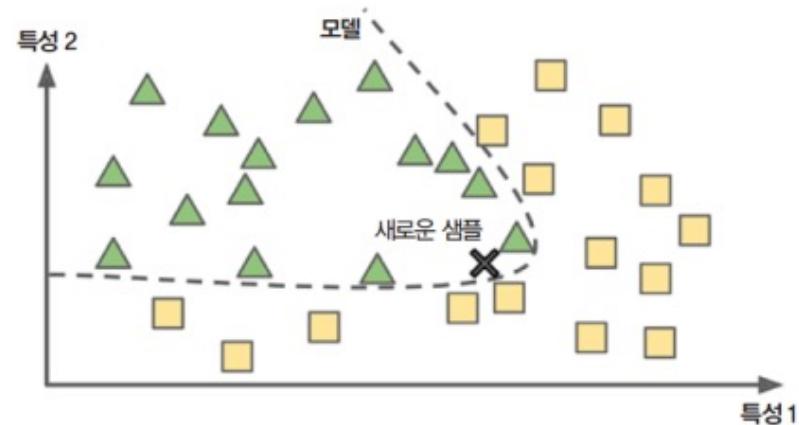
Unit 03 | KNN 고려사항

Unit 04 | 장단점

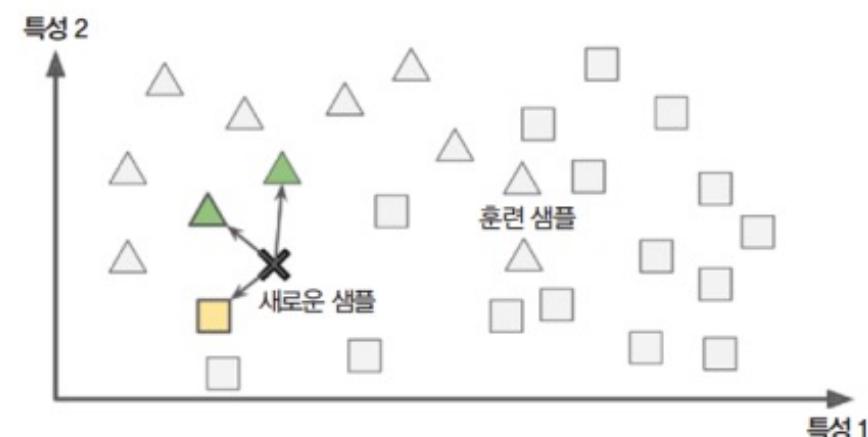
Unit 01 | KNN

사례 기반 학습과 모델 기반 학습**모델 기반 학습**

- 데이터로부터 모델을 생성하여 분류/예측 진행

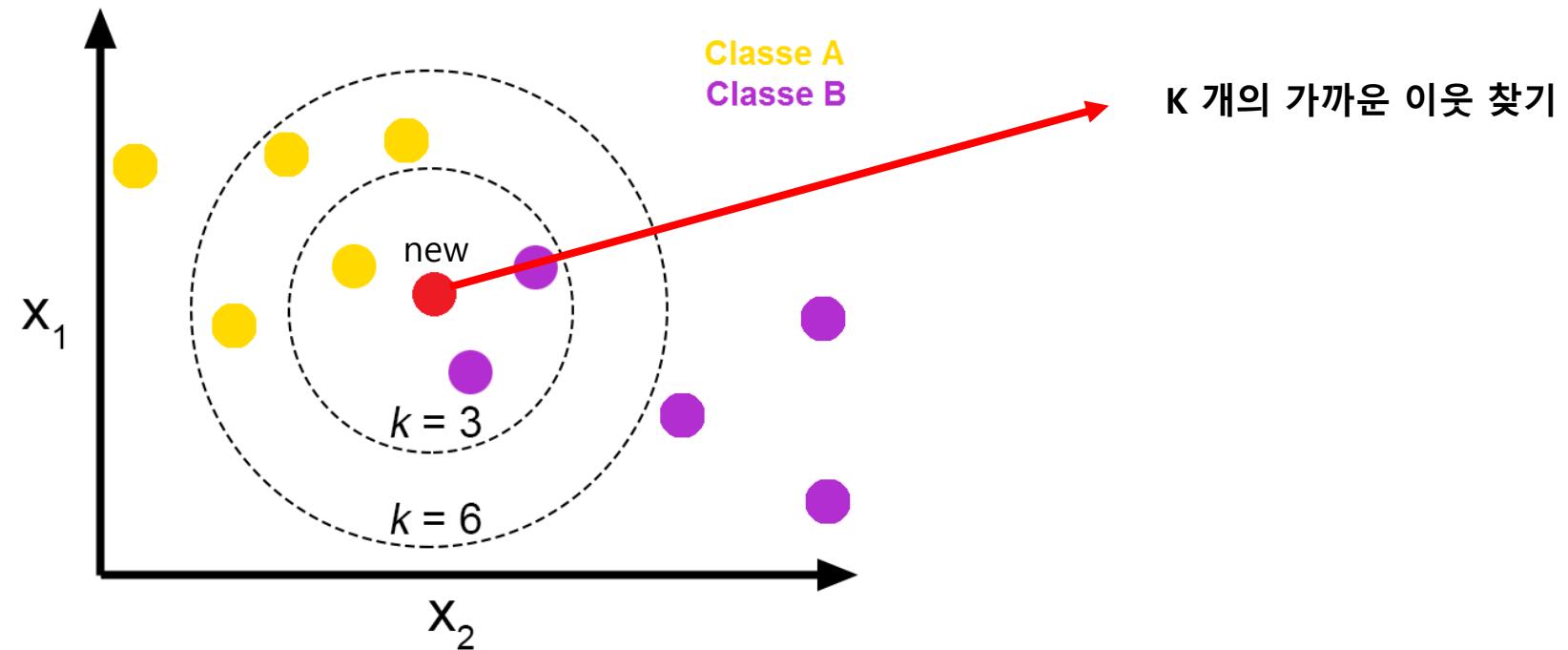
**사례 기반 학습**

- 모델생성 없이 인접 데이터를 분류/예측에 사용
- 새로운 데이터가 들어오면 계산 시작



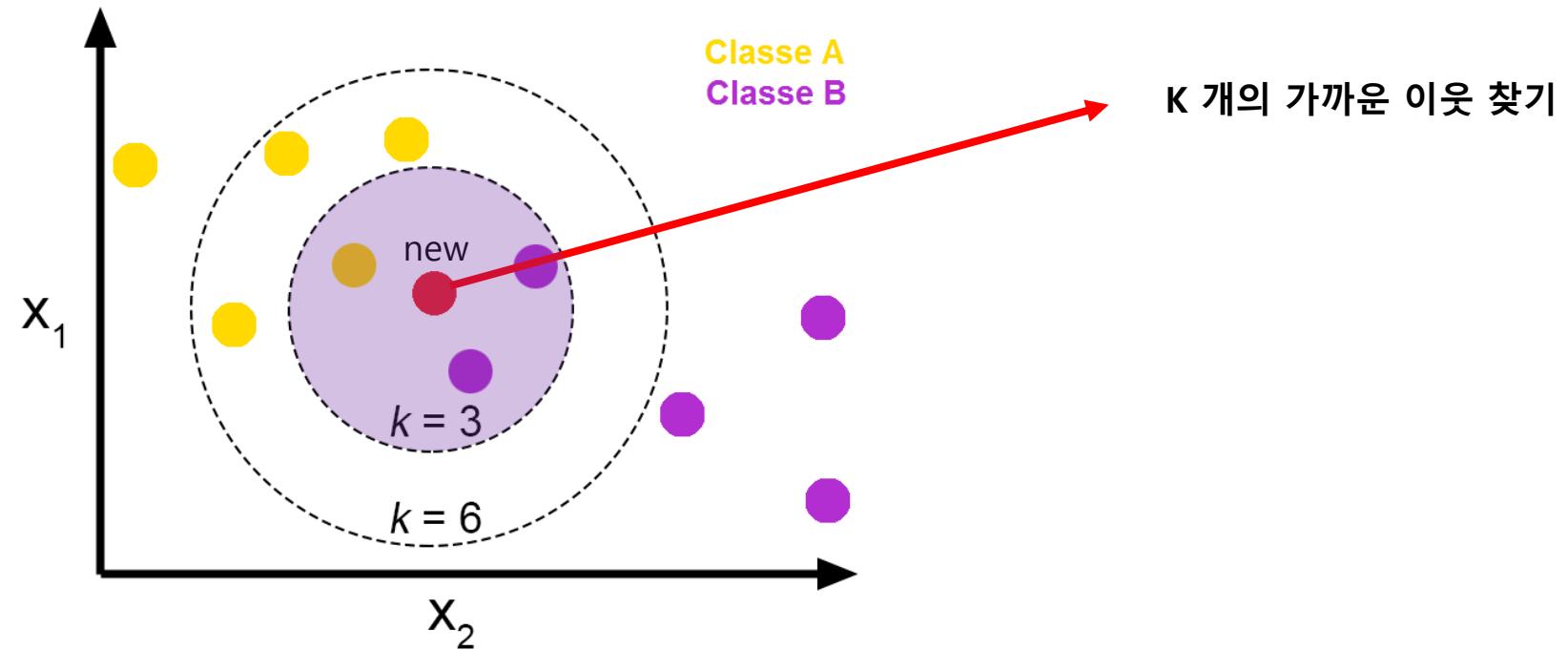
Unit 01 | KNN

최근접 이웃 Nearest Neighbors



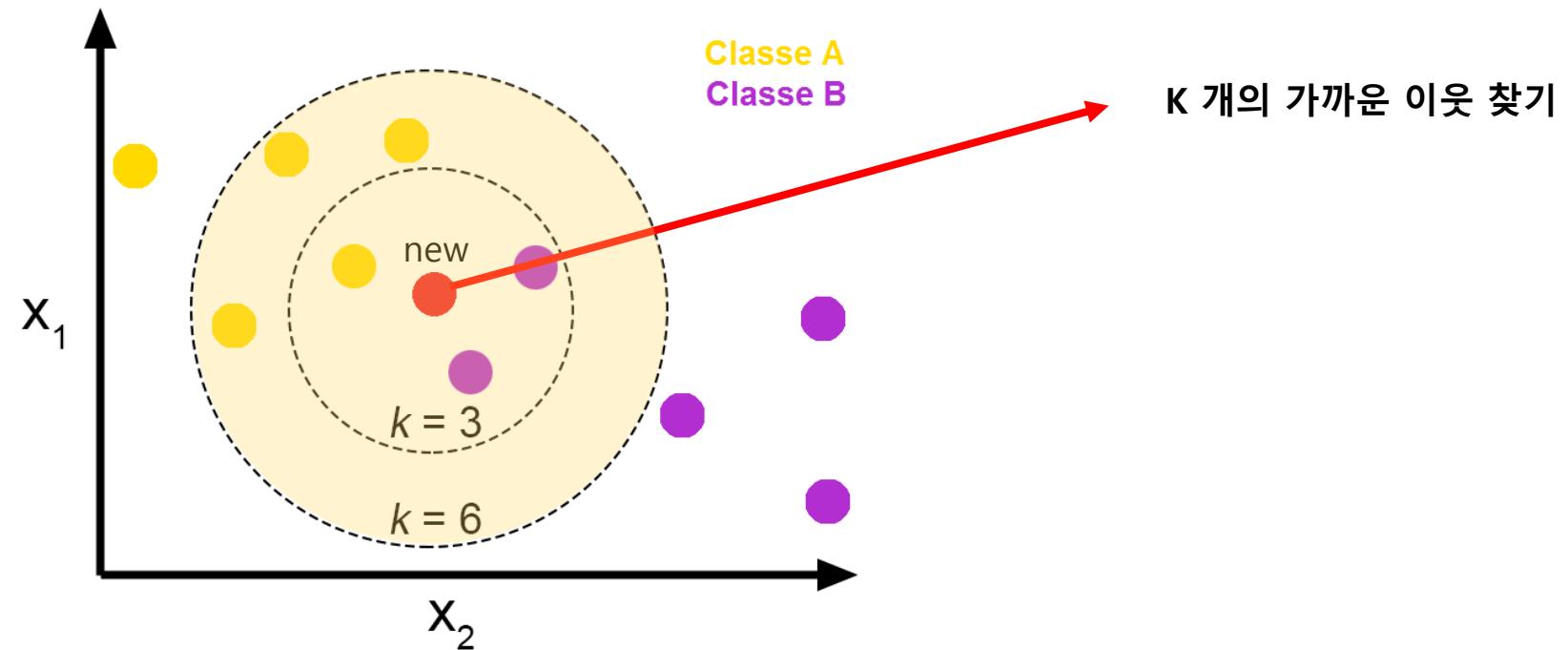
Unit 01 | KNN

최근접 이웃 Nearest Neighbors



Unit 01 | KNN

최근접 이웃 Nearest Neighbors



Unit 01 | KNN

KNN 알고리즘

K-Nearest Neighbors Algorithm

1. Instance-based Learning

: 각각의 관측치 (instance) 만을 이용하여 새로운 데이터에 대한 예측을 진행

2. Memory-based Learning

: 모든 학습 데이터를 메모리에 저장한 후, 이를 바탕으로 예측 시도

3. Lazy Learning

: 모델을 별도로 학습하지 않고, 테스팅 데이터가 들어와야 비로소 작동하는 게으른 알고리즘

Unit 01 | KNN

KNN 분류문제

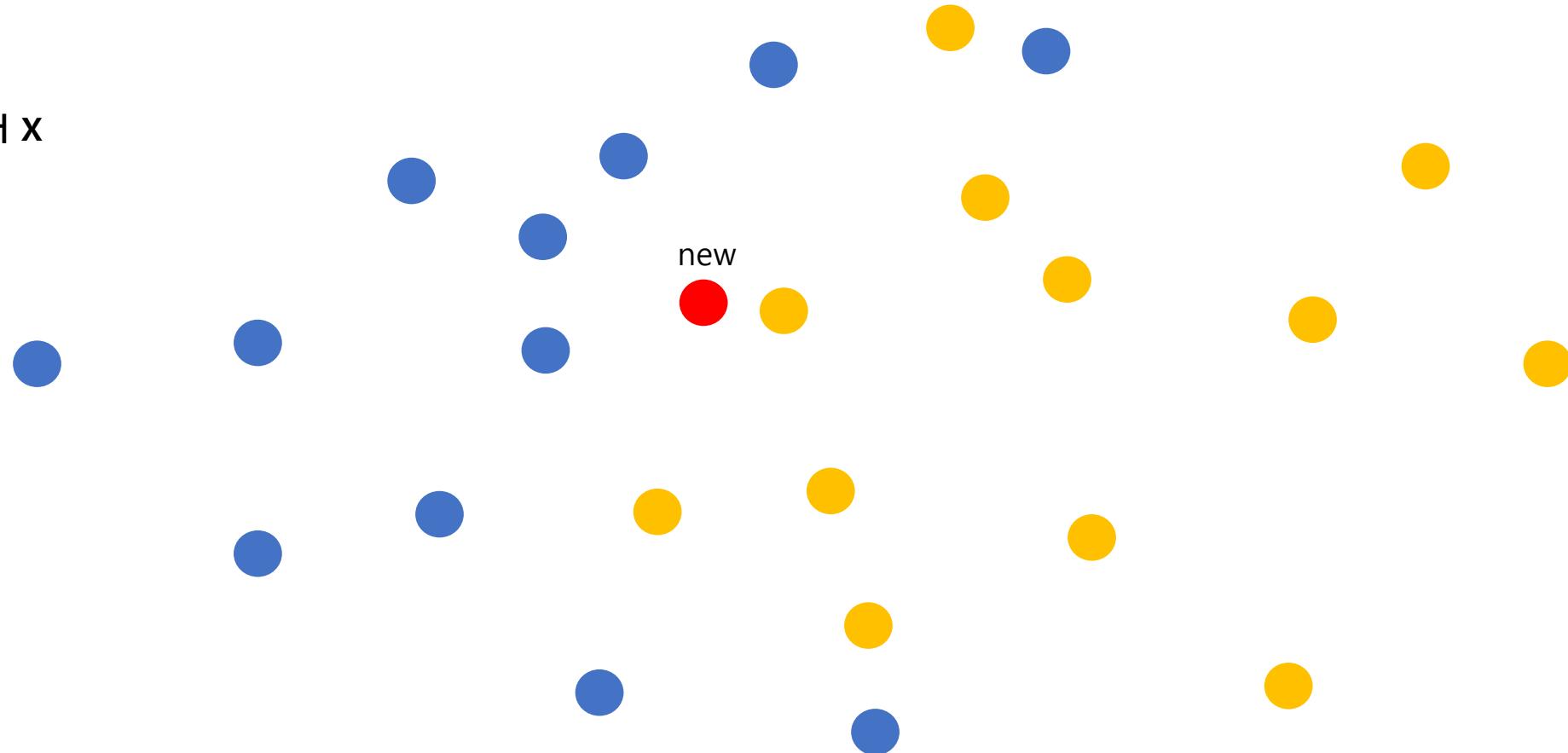
Classification

1. 새로운 데이터 X
2. X 로 부터 인접한 K 개의 학습 데이터 선택
3. 선택된 K 개 학습 데이터의 majority class 선택

Unit 01 | KNN

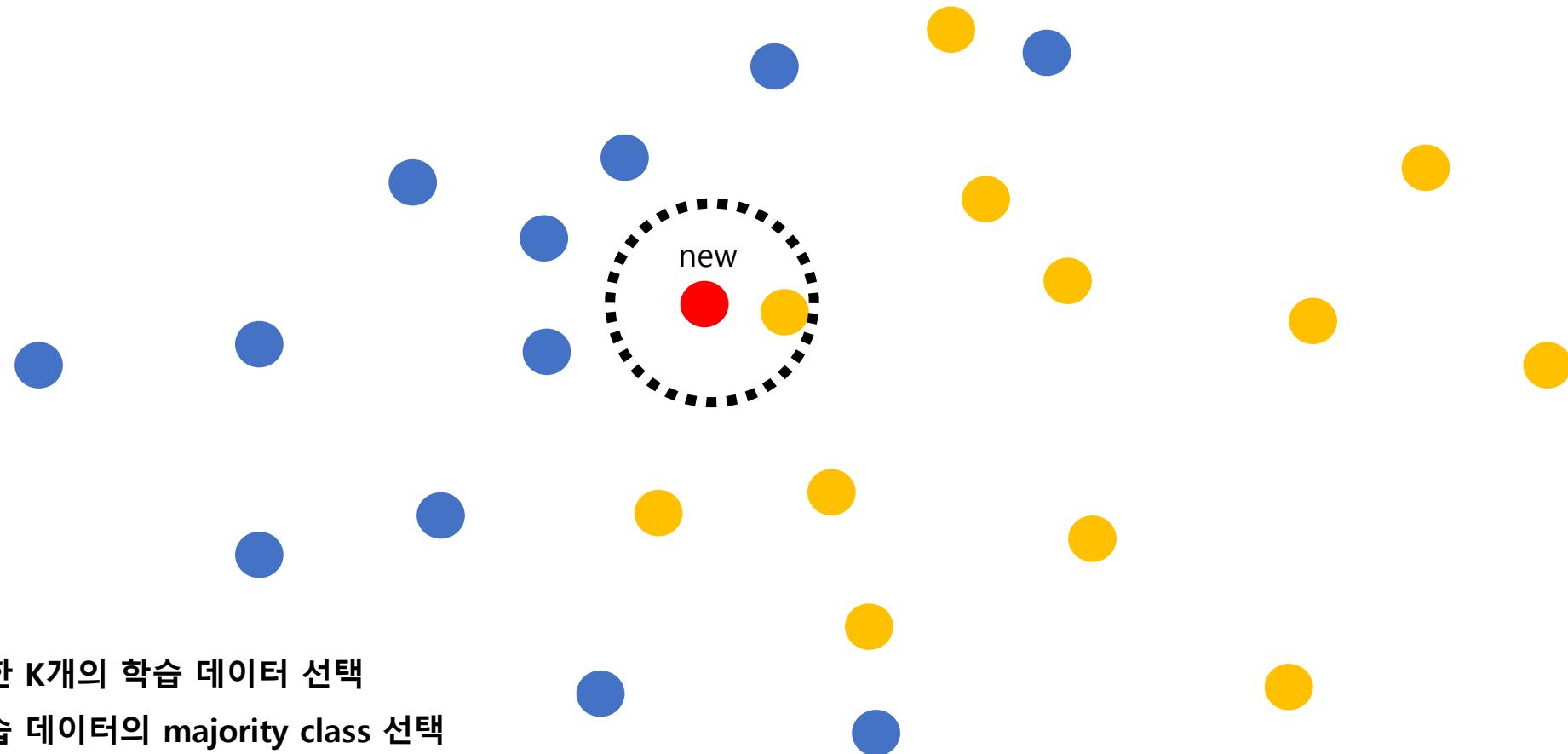
KNN 분류문제 Classification

1. 새로운 데이터 X



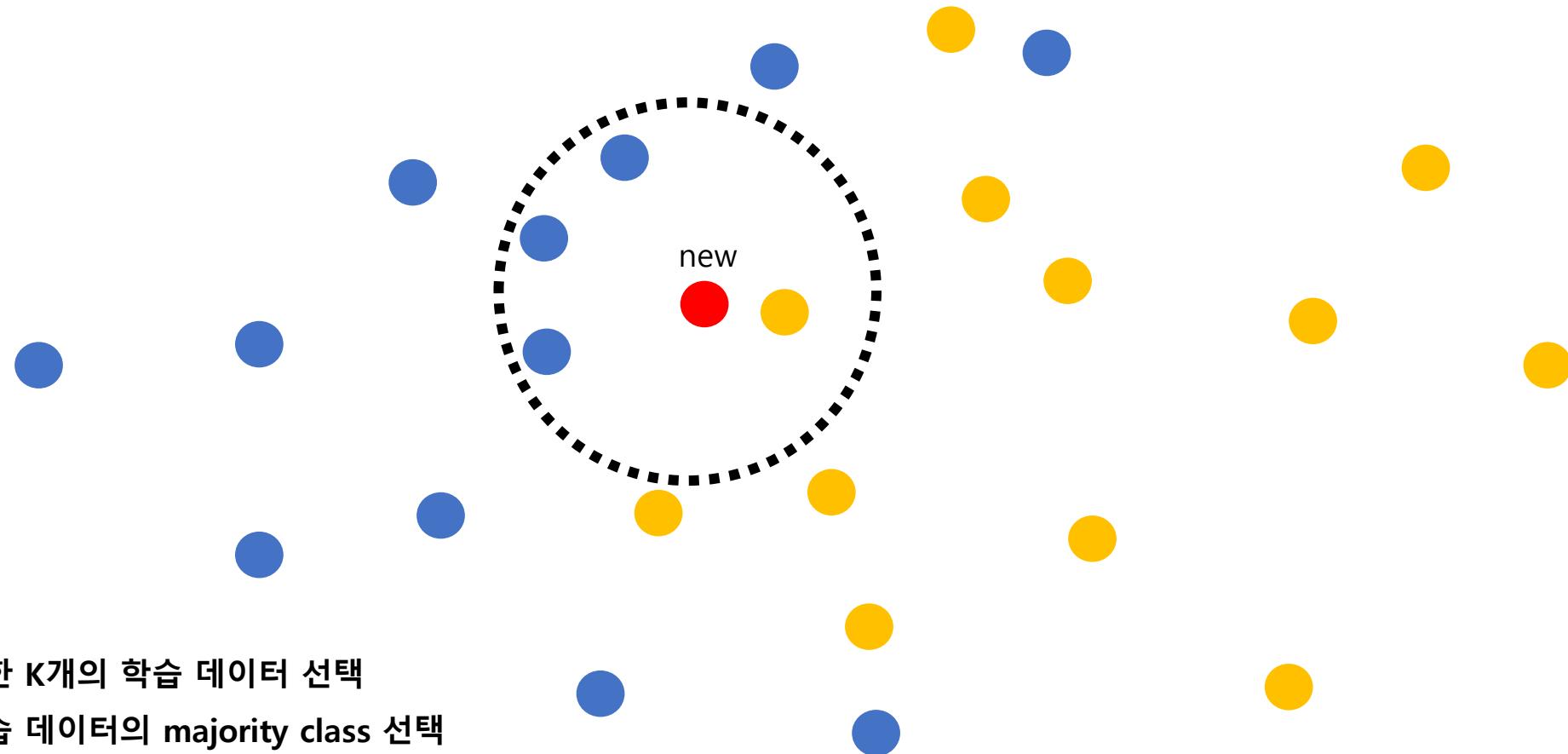
Unit 01 | KNN

KNN 분류문제 Classification



Unit 01 | KNN

KNN 분류문제 Classification



Unit 01 | KNN

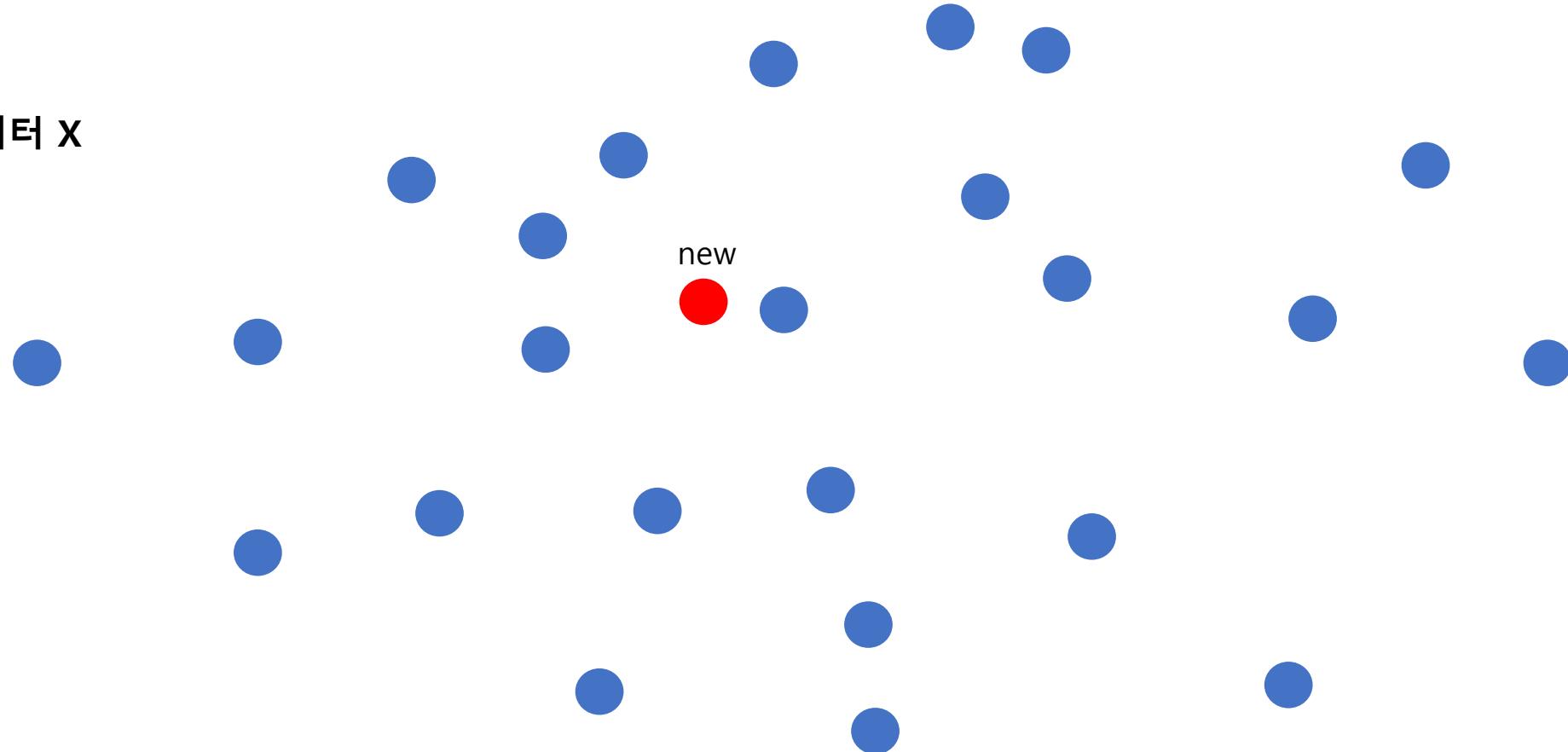
KNN 예측문제 Regression

1. 새로운 데이터 X
2. X 로 부터 인접한 K 개의 학습 데이터 선택
3. 선택된 K 개 학습 데이터의 평균값 계산

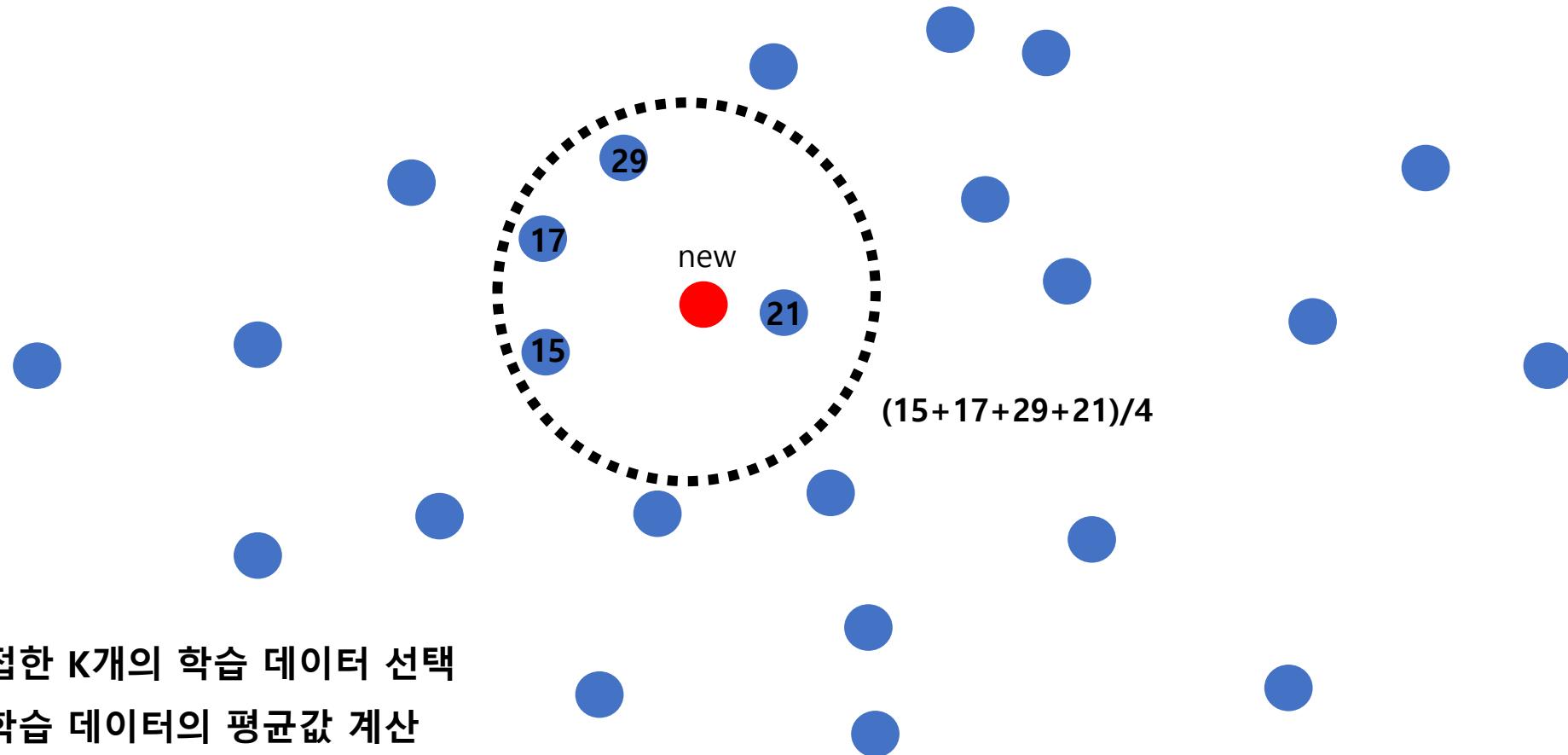
Unit 01 | KNN

KNN 예측문제 Regression

1. 새로운 데이터 X



Unit 01 | KNN

KNN 예측문제
Regression

Unit 02 | KNN Hyperparameter

KNN Hyperparameter

Hyperparameter ?

일반적으로 머신러닝에서 어떠한 임의의 모델을 학습시킬 때, 사람이 직접 튜닝 해주어야 하는 변수

1. K

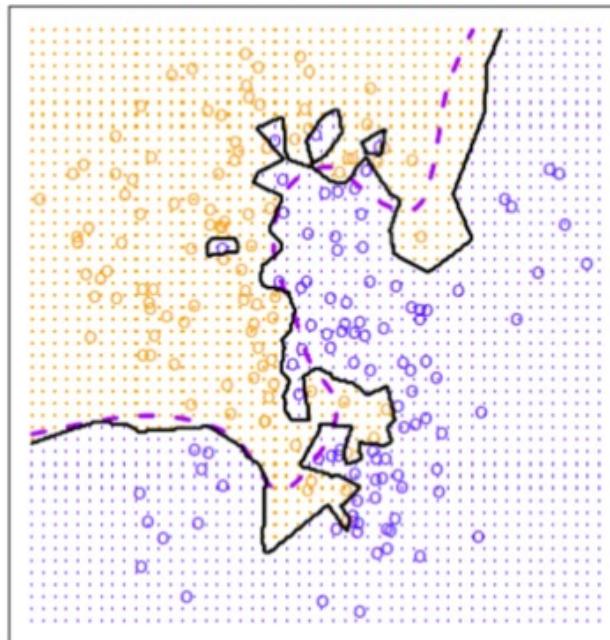
2. Distance Measure

- 1) Euclidean Distance 2) Manhattan Distance 3) Mahalanobis Distance

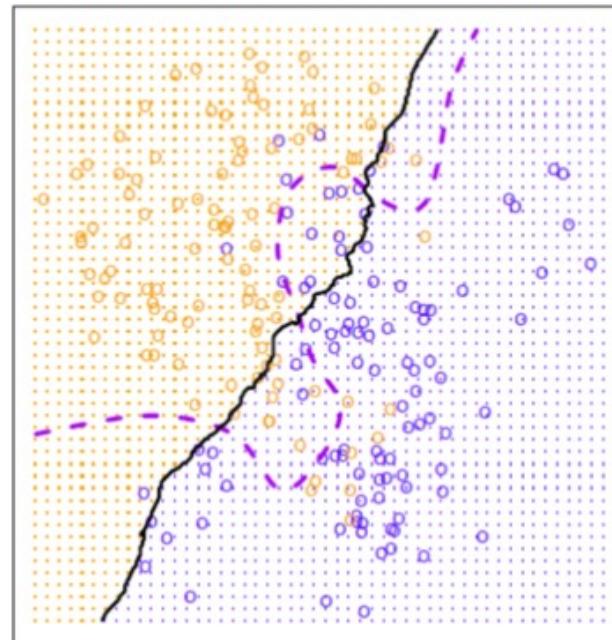
Unit 02 | KNN Hyperparameter

KNN Hyperparameter K

KNN: K=1



KNN: K=100



K가 작은 경우

- 데이터의 지역적 특성을 지나치게 반영
- 분류 경계면이 noise에 민감하게 반응
- Overfitting

K가 큰 경우

- 다른 범주의 개체를 너무 많이 포함하게 됨
- Underfitting

Unit 02 | KNN Hyperparameter

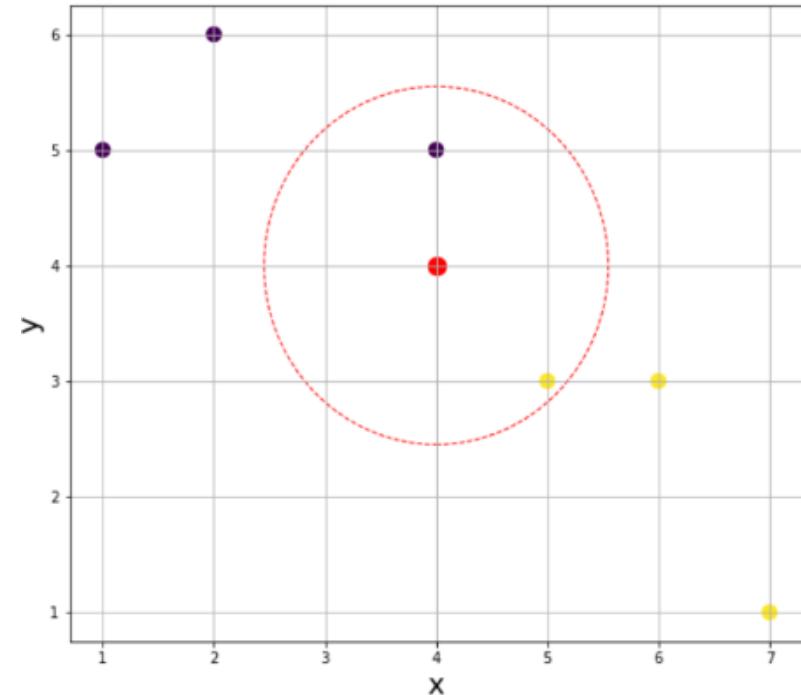
KNN Hyperparameter **K**

최적의 K값 선택 방법 ?

- 노이즈가 없고 잘 구조화된 데이터의 경우 K값이 작을수록 좋다
- 일반적인 규칙은 없음
- 보통 1~20 사이의 값으로 설정
- 보통 홀수를 사용

Unit 02 | KNN Hyperparameter

KNN Hyperparameter K



보편적으로 K값을 정할 때, 동률이 나오지 않도록 홀수로 지정

Unit 02 | KNN Hyperparameter

KNN Hyperparameter **Distance Measure**

Euclidean Distance
유클리드 거리

Manhattan Distance
맨해튼 거리

Mahalanobis Distance
마할라노비스 거리

데이터 간의 거리를 측정하는 방법은 매우 다양

보통 특성 간 값의 범위와 분산 등이 다르기 때문에 측정 단위의 영향력을 없애기 위해
거리 측정 이전에 정규화 혹은 표준화 등의 데이터 스케일링을 통해 범위와 분산을 맞추는 것이 중요!

Unit 02 | KNN Hyperparameter

KNN Hyperparameter Distance Measure

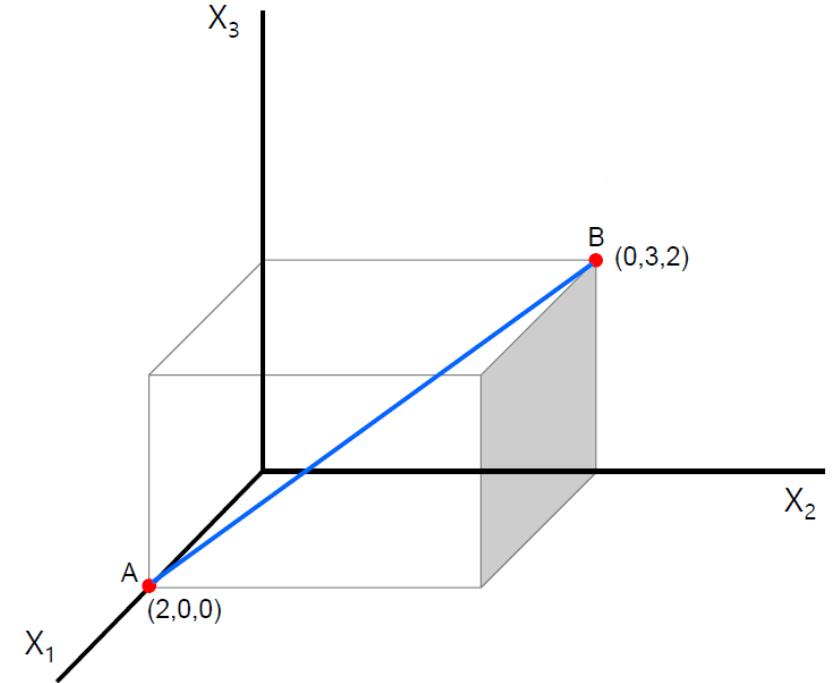
1) Euclidean Distance

Euclidean Distance : 두 점 간의 기하학적 거리를 측정

$$X = (x_1, x_2, \dots, x_n)$$

$$Y = (y_1, y_2, \dots, y_n)$$

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



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

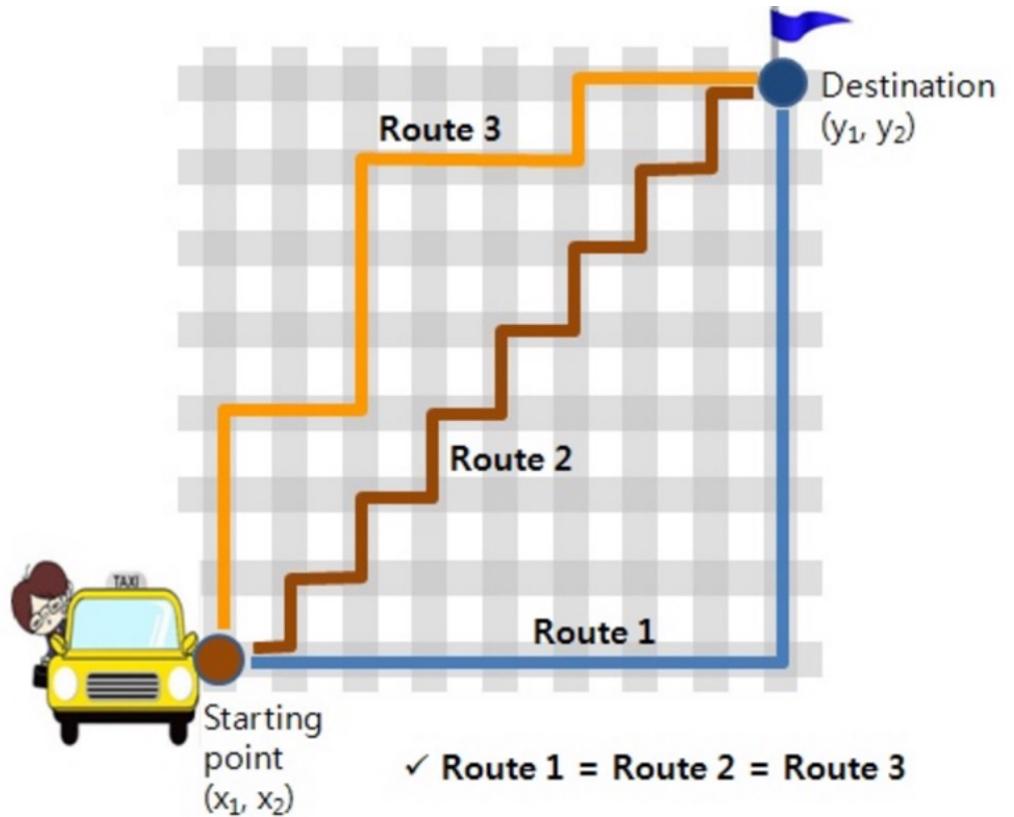
Unit 02 | KNN Hyperparameter

KNN Hyperparameter Distance Measure

2) Manhattan Distance

Manhattan Distance

- : 각 좌표축 방향으로만 이동했을 때의 거리
- : 항상 유클리드 거리보다 같거나 크다는 성질이 있음



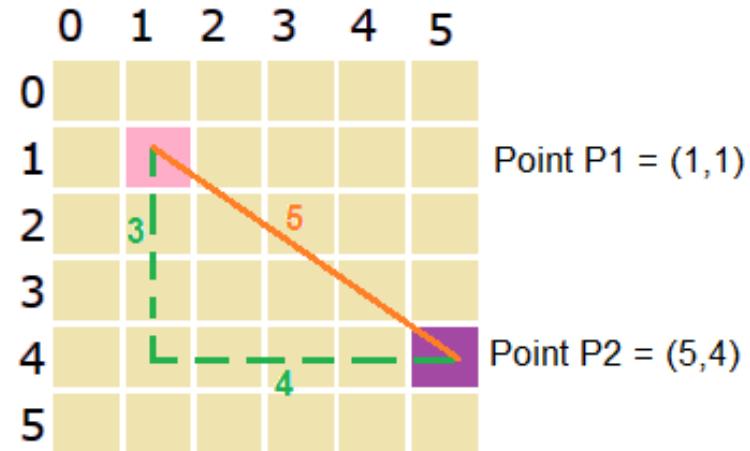
Unit 02 | KNN Hyperparameter

KNN Hyperparameter Distance Measure

2) Manhattan Distance

Manhattan Distance

- : 각 좌표축 방향으로만 이동했을 때의 거리
- : 항상 유clidean 거리보다 같거나 크다는 성질이 있음



$$\text{Euclidean distance} = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$\text{Manhattan distance} = |5-1| + |4-1| = 7$$

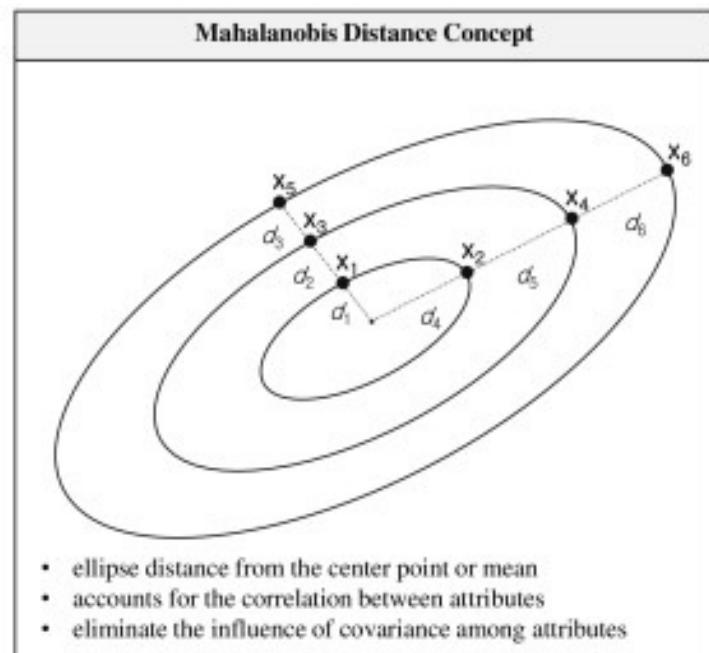
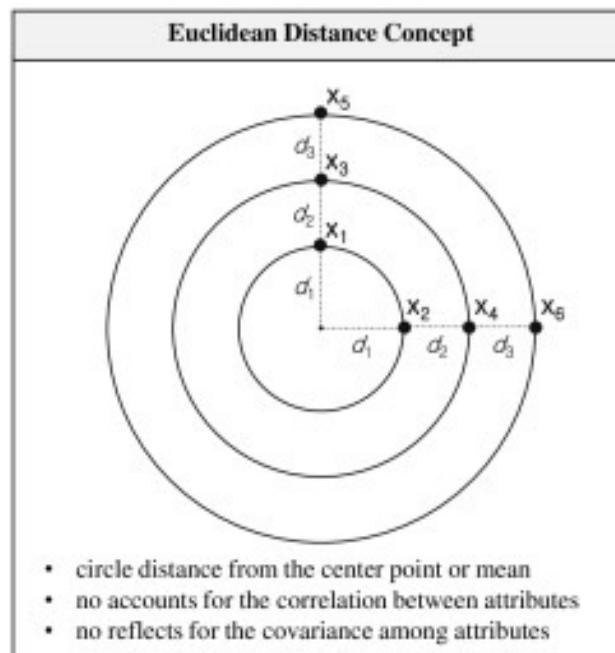
Unit 02 | KNN Hyperparameter

KNN Hyperparameter Distance Measure

3) Mahalanobis Distance

Mahalanobis Distance

- : 변수 내 분산, 변수 간 공분산을 모두 반영하여 거리를 계산하는 방식
- : 변수들 간에 상관관계가 존재하는 경우 유리



Unit 03 | KNN 고려사항

KNN 고려사항

1. Distance 기반 알고리즘

2. 더 좋은 방법은 ?

Unit 03 | KNN 고려사항

Distance 기반 알고리즘

1. Feature Scaling

	X1	X2 (\$)
A	1	5
B	2	6
C	4	4

$$\begin{aligned} \text{Distance}(A,C) &= \sqrt{(1-4)^2 + (5-4)^2} \\ &= 3.162278 \end{aligned}$$

$$\begin{aligned} \text{Distance}(B,C) &= \sqrt{(2-4)^2 + (6-4)^2} \\ &= 2.828427 \end{aligned}$$

	X1	X2 (₩)
A	1	5000
B	2	6000
C	4	4000

$$\begin{aligned} \text{Distance}(A,C) &= \sqrt{(1-4)^2 + (5000-4000)^2} \\ &= 1000.004 \end{aligned}$$

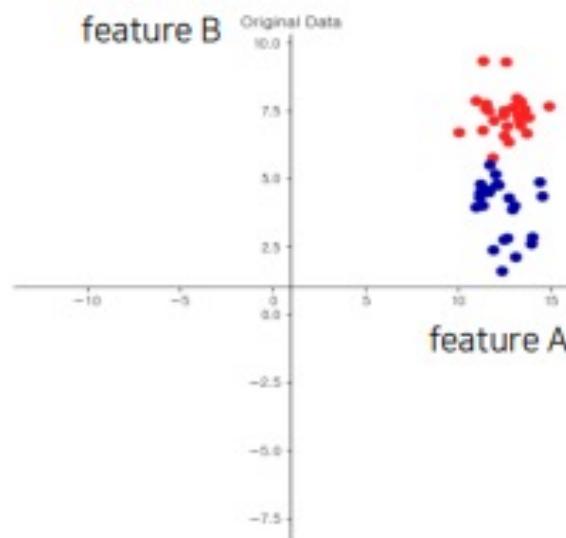
$$\begin{aligned} \text{Distance}(B,C) &= \sqrt{(2-4)^2 + (6000-4000)^2} \\ &= 2000.001 \end{aligned}$$

Unit 03 | KNN 고려사항

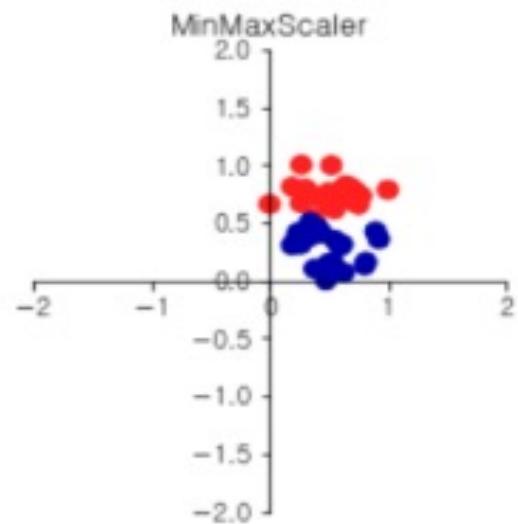
Distance 기반 알고리즘

1. Feature Scaling

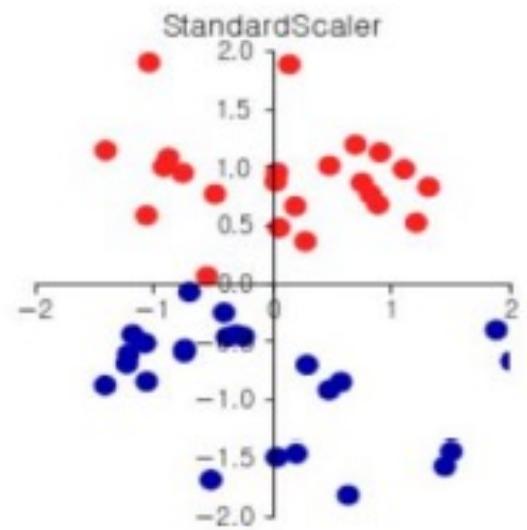
Original data



Min-Max Normalization



Standardization



Unit 03 | KNN 고려사항

Distance 기반 알고리즘**2. One-Hot encoding**

1개만 Hot(True) 이고 나머지는 Cold(False)

KNN은 거리기반이기 때문에 input에 numerical 와야 함.

The diagram illustrates the process of one-hot encoding. On the left, there is a vertical table with a single column labeled "color". It contains four rows: "Red" (in red), "Green" (in green), "Blue" (in blue), and another "Red" (in red). An arrow points from this table to a larger table on the right. The right table has three columns labeled "color_Red", "color_Green", and "color_Blue". It has four rows. The first row corresponds to the first "Red" entry in the left table, with "1" in the "color_Red" column and "0"s in the other two. The second row corresponds to "Green", with "0" in "color_Red" and "1" in "color_Green". The third row corresponds to "Blue", with "0" in "color_Red" and "1" in "color_Blue". The fourth row corresponds to the second "Red" entry in the left table, with "1" in "color_Red" and "0"s in the others.

color
Red
Green
Blue
Red

color_Red	color_Green	color_Blue
1	0	0
0	1	0
0	0	1
1	0	0

Unit 03 | KNN 고려사항

Weighted KNN

Majority Voting, average 보다 더 좋은 방법은?

거리가 가까운(유사도가 높은) 이웃의 정보에 좀 더 가중치를 주는 방법

-> 단순 평균, 다수결로 값을 정하지 않고 **거리에 따라서 영향력을 다르게 주고 싶을 때 사용**

```
neighbors.KNeighborsClassifier(n_neighbors, weights=weights)  
clf.fit(X,y)
```

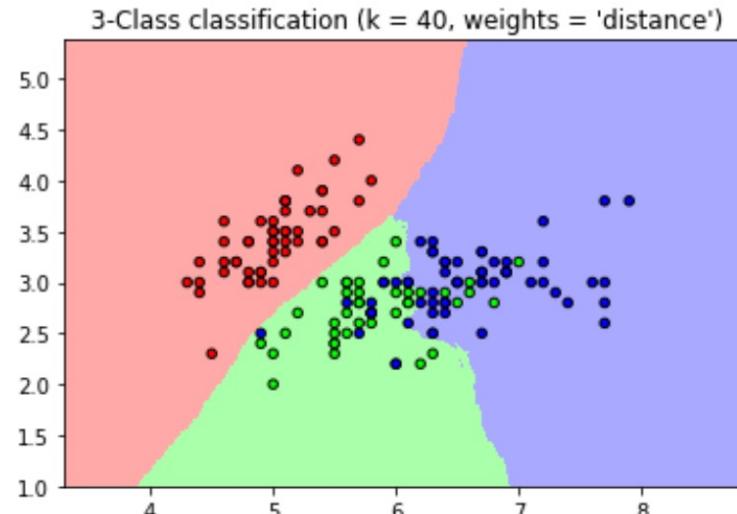
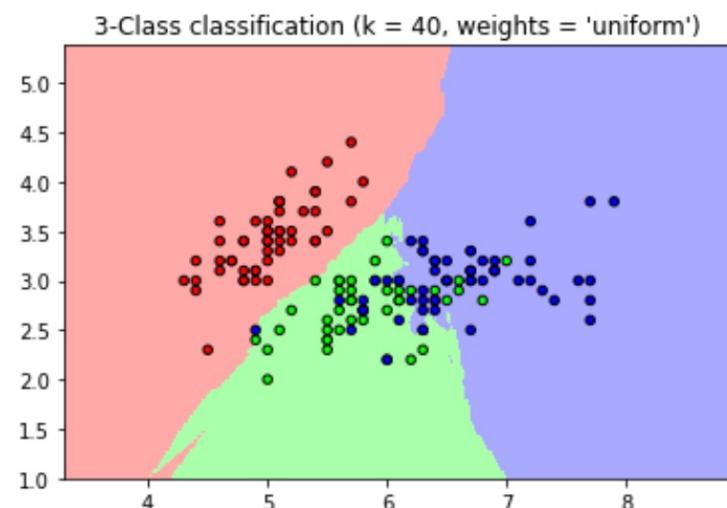
Unit 03 | KNN 고려사항

Weighted KNN

Majority Voting, average 보다 더 좋은 방법은?

거리가 가까운(유사도가 높은) 이웃의 정보에 좀 더 가중치를 주는 방법

-> 단순 평균, 다수결로 값을 정하지 않고 **거리에 따라서 영향력을 다르게 주고 싶을 때 사용**



Unit 04 | 장단점

KNN 장점

1. 이해하기 쉽다
2. 학습데이터의 수가 많을 경우 효과적이다
3. 데이터 내 노이즈 영향을 크게 받지 않으며, 특히 Manhalanobis distance와 같이 데이터의 분산을 고려할 경우 강건하다

Unit 04 | 장단점

KNN 단점

1. 하나의 데이터를 예측할 때마다 전체 데이터와의 거리를 계산하기 때문에, 연산속도가 다른 알고리즘에 비해 느리다
2. 어떤 거리 척도가 분석에 적합한지 불분명하기에 데이터의 특성에 맞는 거리측도를 임의로 선정해야 한다.

과제

주어진 데이터로 주석과 함께 자유롭게 분석을 진행해주세요 ☺

- **Preprocessing / EDA**
- **KNN & Hyperparameter tuning**
- **Evaluation**

참고자료

- 투빅스 18기 강효은님 강의자료
- 투빅스 17기 이지수님 강의자료
- 투빅스 16기 박한나님 강의자료
- 투빅스 13기 김민정님 강의자료

20기 정규세션

ToBig's 19기 최다희

Clustering

Contents

Unit 01 | Clustering

Unit 02 | Hierarchical clustering

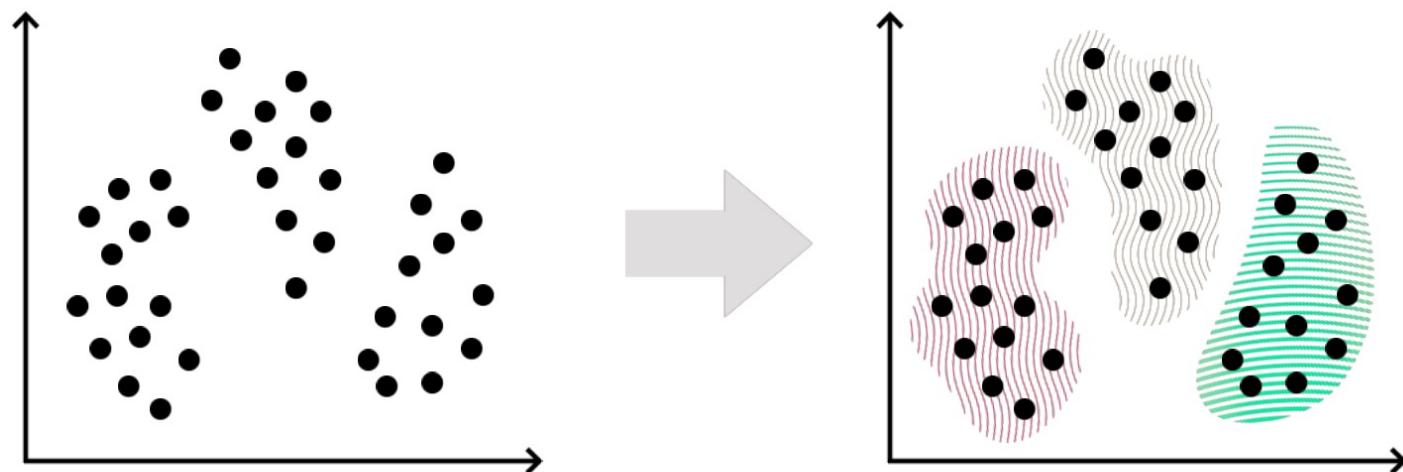
Unit 03 | K-Means & DBSCAN

Unit 04 | 모델평가

Unit 01 | Clustering

Clustering (군집화)

유사한 속성을 갖는 관측치들을 묶어 전체 데이터를 몇 개의 군집으로 나누는 것



Unit 01 | Clustering

Clustering vs Classification

1. Classification (Supervised)

- 소속 집단의 정보를 이미 알고 있는 상태에서 비슷한 집단으로 묶는 방법
- Label이 있는 데이터를 나누는 방법

2. Clustering (Unsupervised)

- 소속 집단의 정보가 없고, 모르는 상태에서 비슷한 집단으로 묶는 방법
- Label이 없는 데이터를 나누는 방법

Unit 01 | Clustering

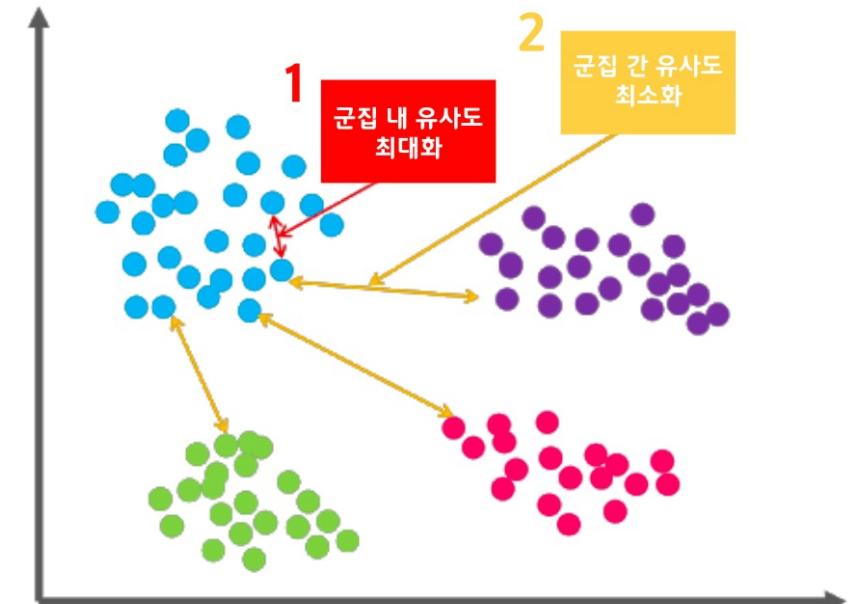
군집분석의 원리

1. 군집 내 응집도 최대화

- 같은 군집 내 응집도를 최대화 하는 것으로, 같은 군집 내 객체들의 거리를 최소화 하는 것

2. 군집 내 분리도 최대화

- 다른 군집 간 거리를 최대화 하는 것



Unit 01 | Clustering

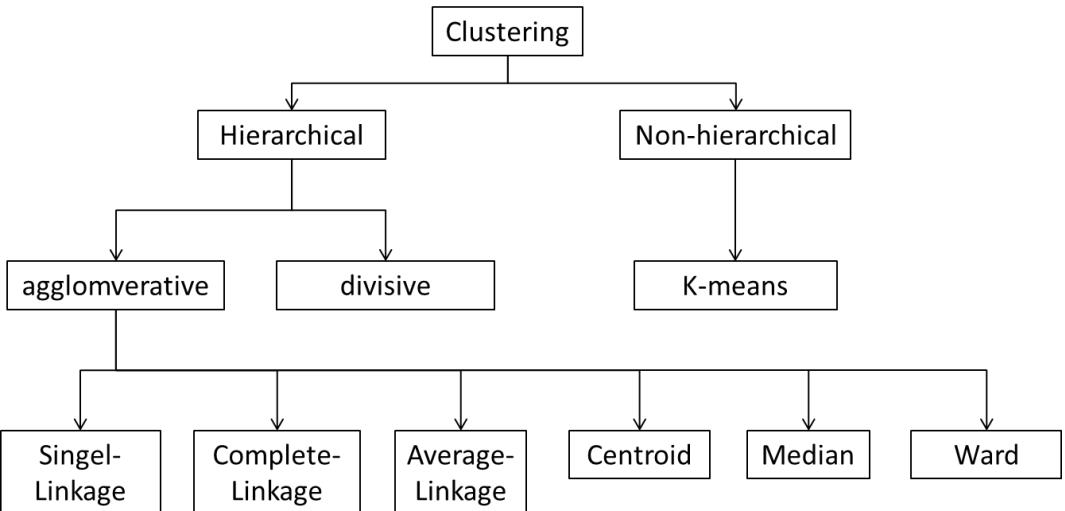
Clustering 방법

1. Hierarchical clustering (계층적 군집화)

- 개체들을 가까운 집단부터 차근차근 묶어가는 방식
- 군집화 결과 뿐만 아니라 유사한 개체들이 결합되는 절차까지
- Agglomerative / Divisive

2. Partitioning clustering (분리형 / 비계층적 군집화)

- 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- K-Means / DBSCAN



Unit 02 | Hierarchical clustering

Hierarchical Clustering

- 개체들을 가까운 집단부터 차근차근 묶어가는 방식
- 클러스터 수를 미리 정해주지 않아도 되는 장점
- Agglomerative / Divisive

Unit 02 | Hierarchical clustering

Hierarchical Clustering - Agglomerative (병합군집)

1. 계층적 트리 모형을 이용해 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합하여 군집화를 수행
2. 덴드로그램을 통해 시각화
3. 사전에 군집의 개수를 정하지 않아도 수행 가능



Unit 02 | Hierarchical clustering

Hierarchical Clustering - Agglomerative (병합군집)

1. 계층적 트리 모형을 이용해 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합하여 군집화를 수행
2. 덴드로그램을 통해 시각화
3. 사전에 군집의 개수를 정하지 않아도 수행 가능



Unit 02 | Hierarchical clustering

Hierarchical Clustering - Agglomerative (병합군집)

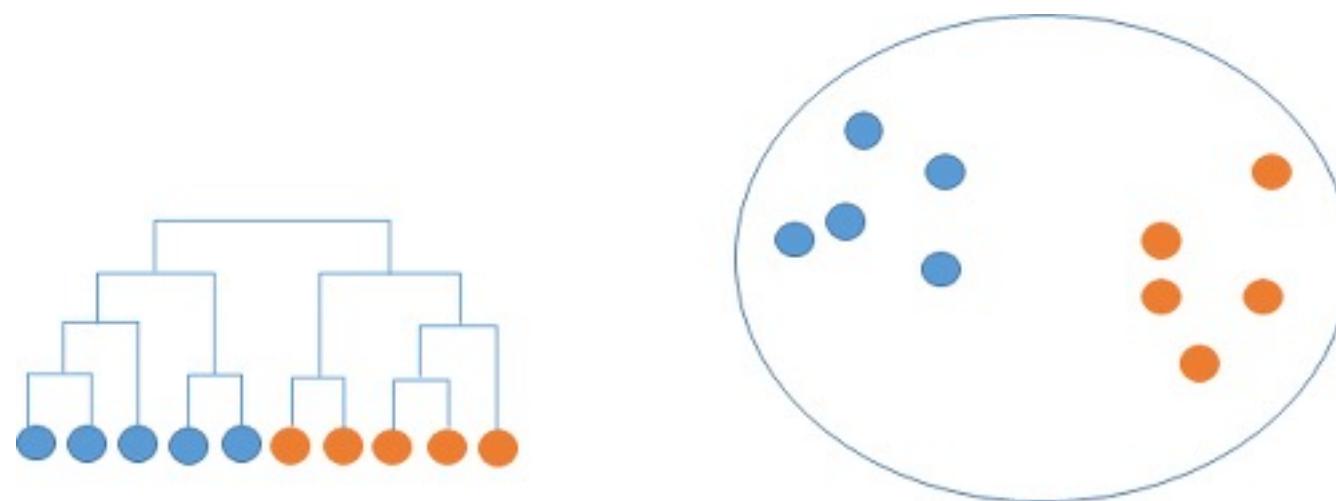
1. 계층적 트리 모형을 이용해 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합하여 군집화를 수행
2. 덴드로그램을 통해 시각화
3. 사전에 군집의 개수를 정하지 않아도 수행 가능



Unit 02 | Hierarchical clustering

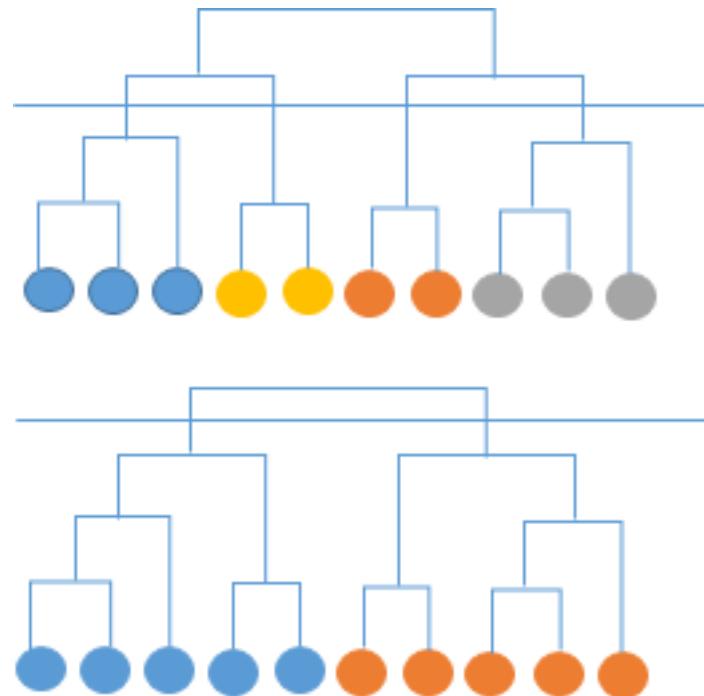
Hierarchical Clustering - Agglomerative (병합군집)

1. 계층적 트리 모형을 이용해 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합하여 군집화를 수행
2. 덴드로그램을 통해 시각화
3. 사전에 군집의 개수를 정하지 않아도 수행 가능



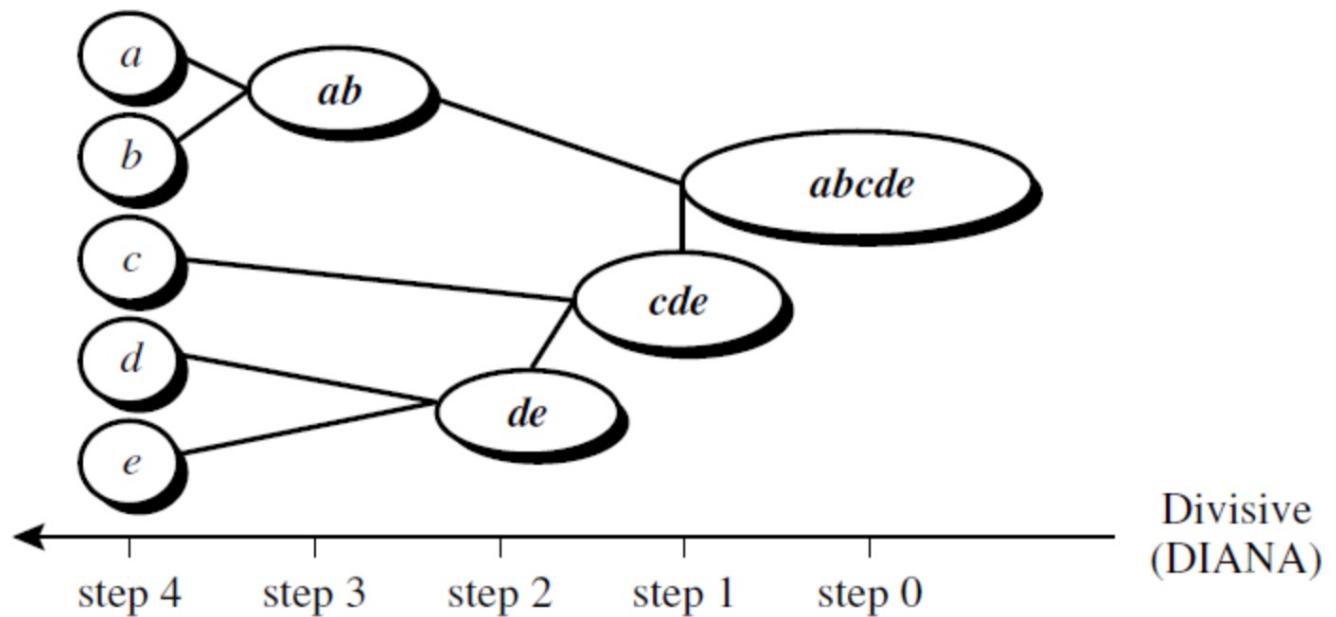
Unit 02 | Hierarchical clustering

Hierarchical Clustering - Agglomerative (병합군집)



Unit 02 | Hierarchical clustering

Hierarchical Clustering - Divisive (분할군집)



Unit 03 | K-Means & DBSCAN

Partitioning Clustering

- 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- K-Means / DBSCAN

Unit 03 | K-Means & DBSCAN

K-Means

- 대표적인 분리형 군집화 알고리즘
- 각 군집은 하나의 중심을 갖는다
- 각 객체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성
- 사전에 군집의 수 K가 정해져야 알고리즘을 실행할 수 있다

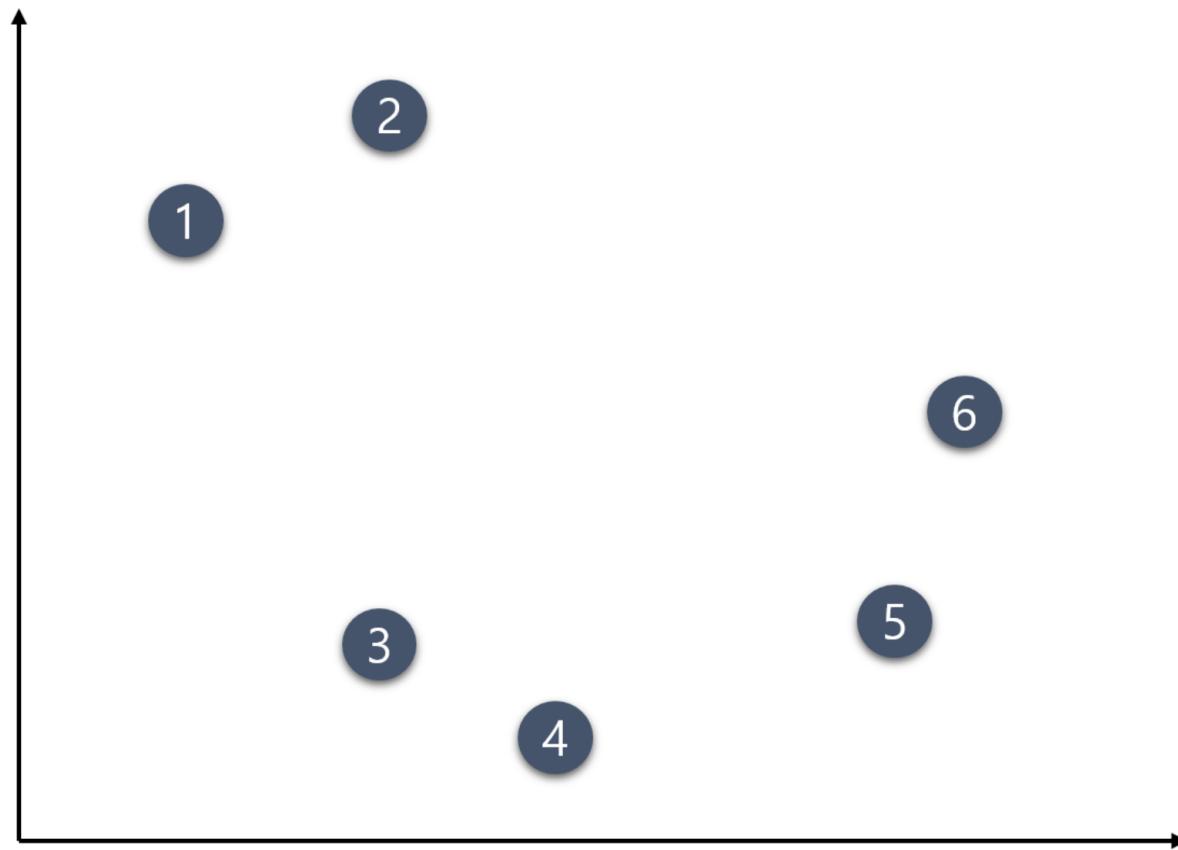
Unit 03 | K-Means & DBSCAN

K-Means

1. K 결정
2. 초기 Centroid 선택
 1. 랜덤
 2. 수동
 3. Kmean ++
3. 모든 데이터를 순회하며 각 데이터마다 가장 가까운 Centroid 가 속해 있는 클러스터로 assign
4. Centroid를 클러스터의 중심으로 이동
5. 3, 4 반복

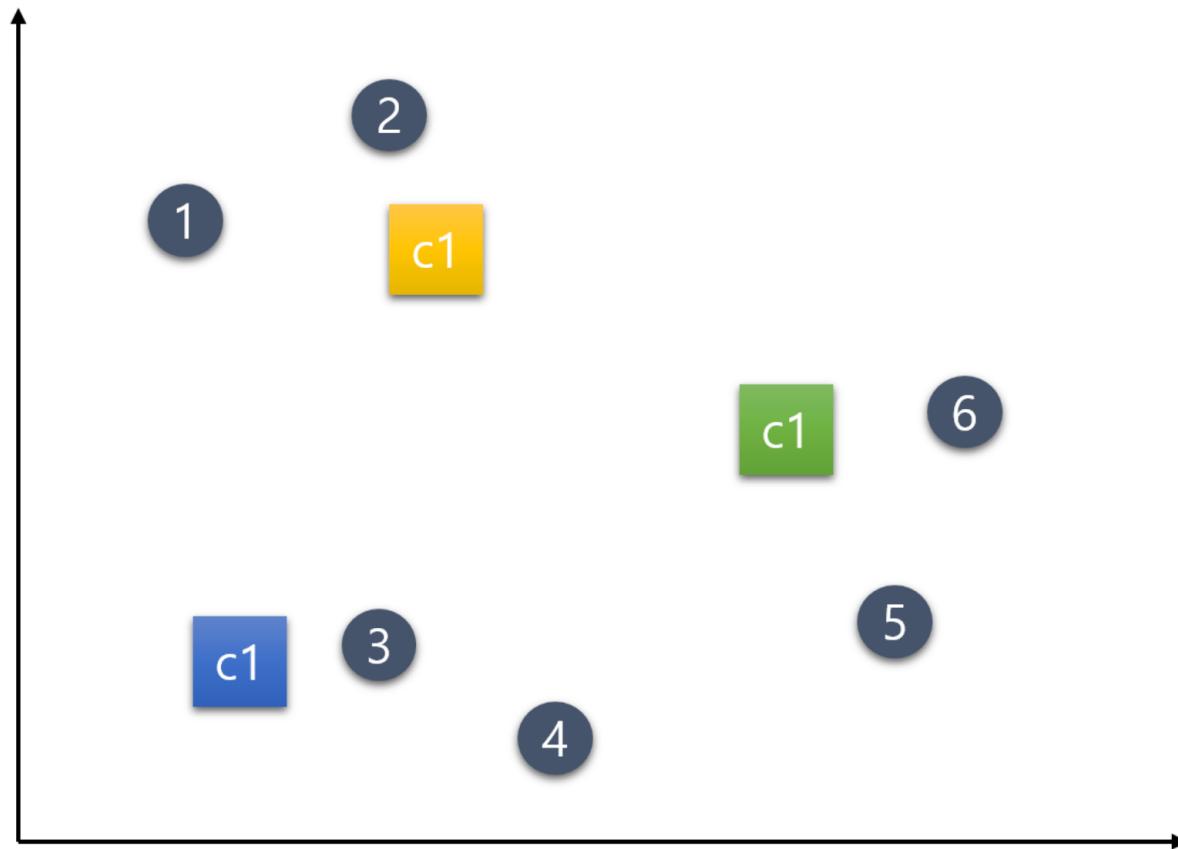
Unit 03 | K-Means & DBSCAN

K-Means



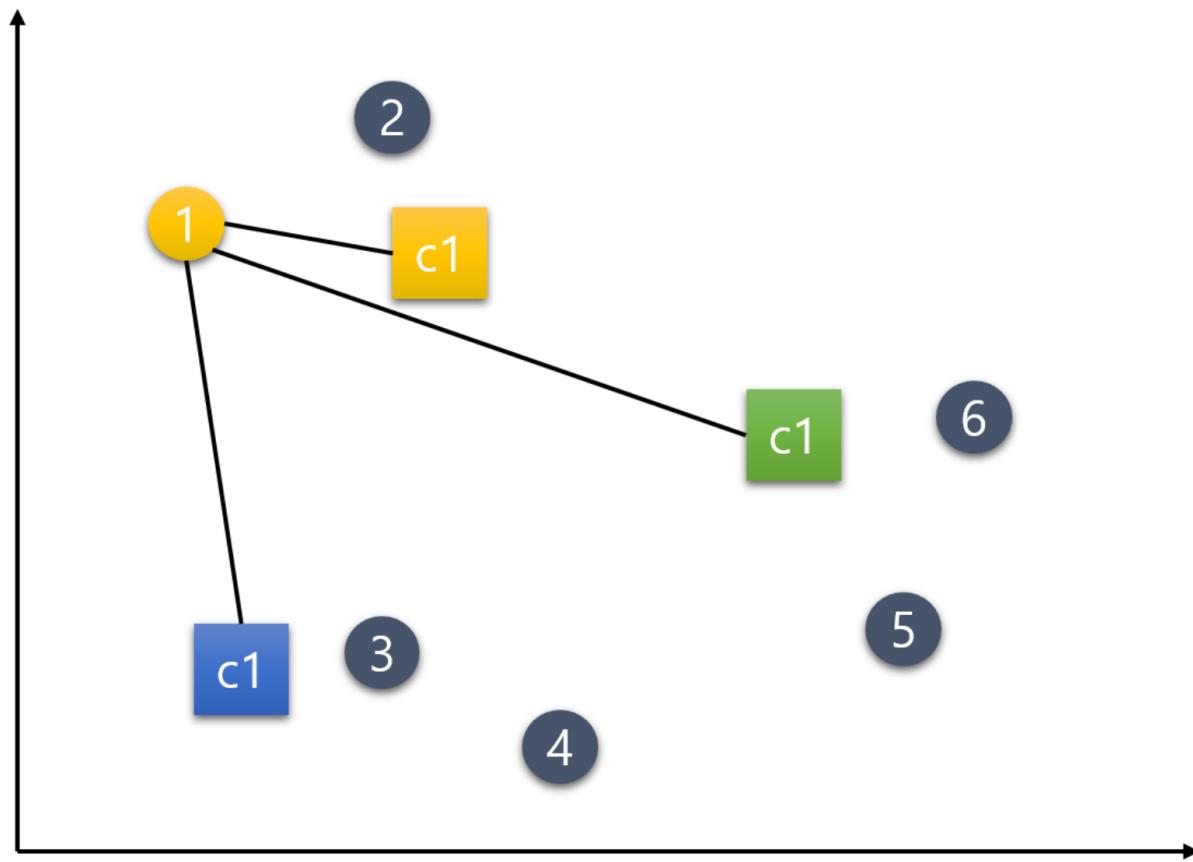
Unit 03 | K-Means & DBSCAN

K-Means



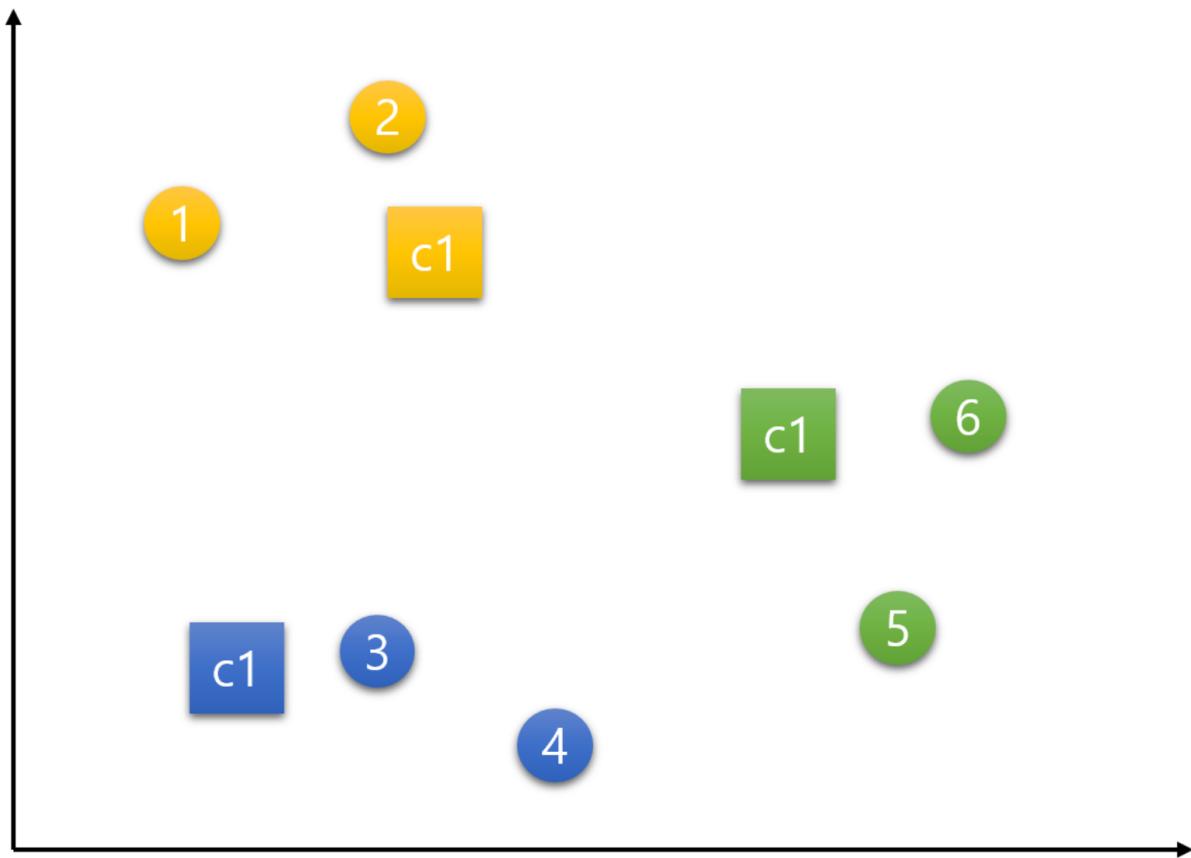
Unit 03 | K-Means & DBSCAN

K-Means



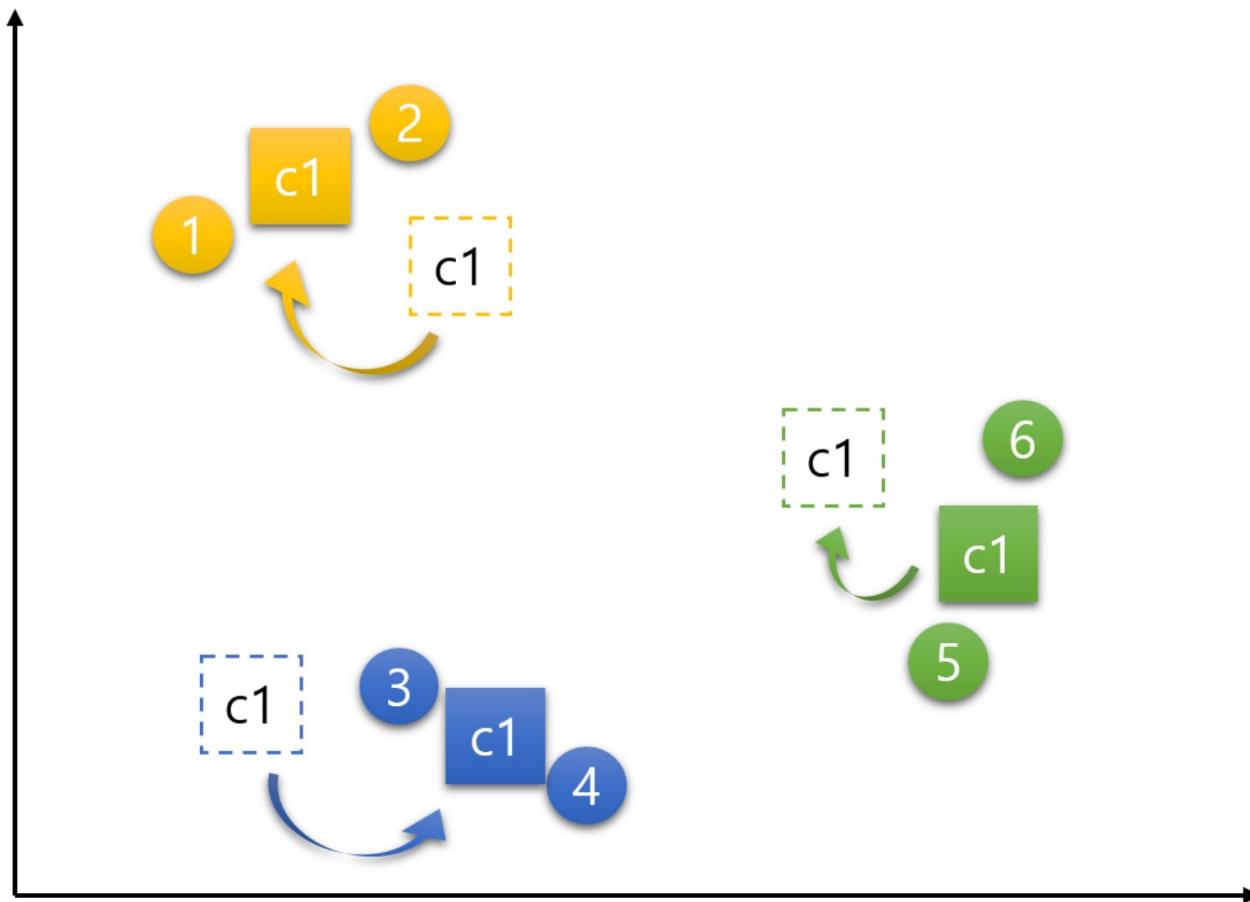
Unit 03 | K-Means & DBSCAN

K-Means



Unit 03 | K-Means & DBSCAN

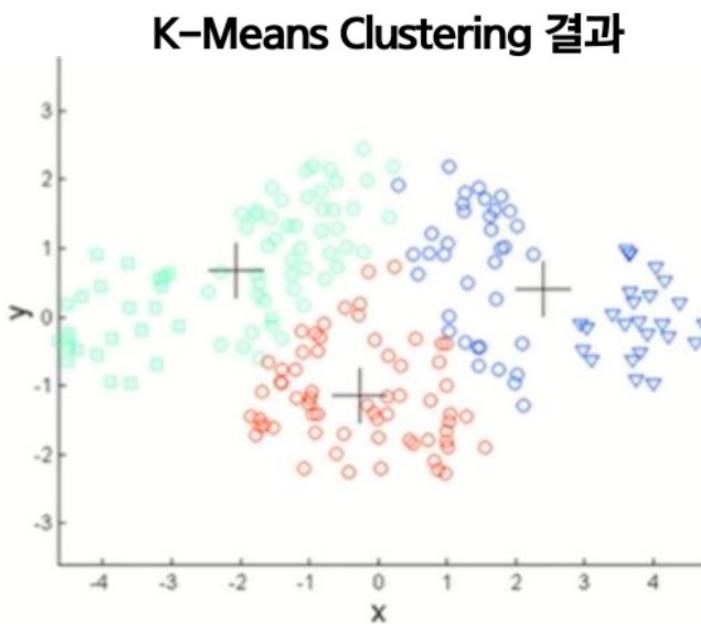
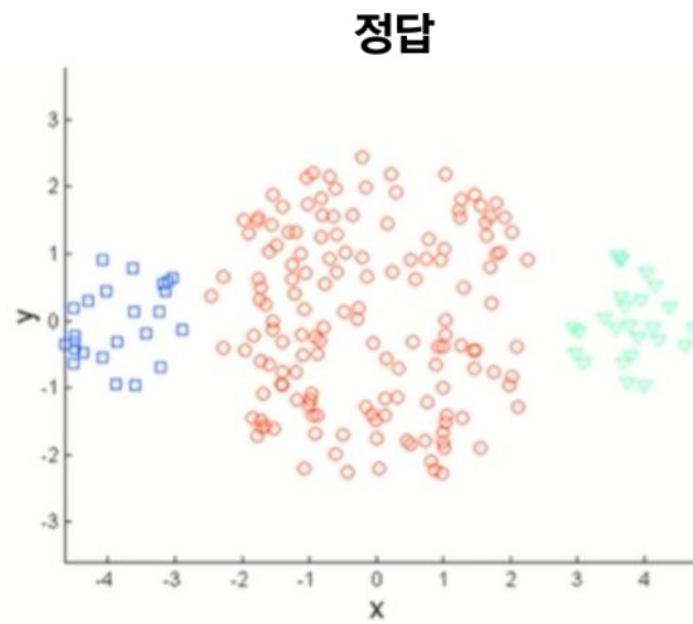
K-Means



Unit 03 | K-Means & DBSCAN

K-Means 문제점

- 서로 다른 크기의 군집을 잘 찾지 못함

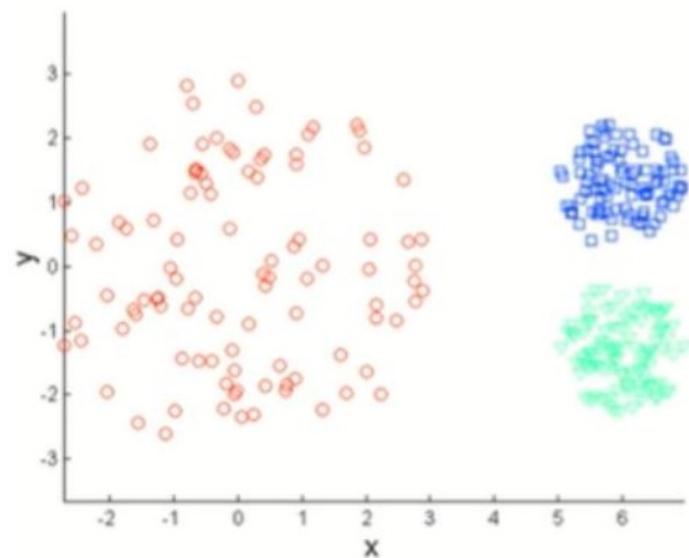


Unit 03 | K-Means & DBSCAN

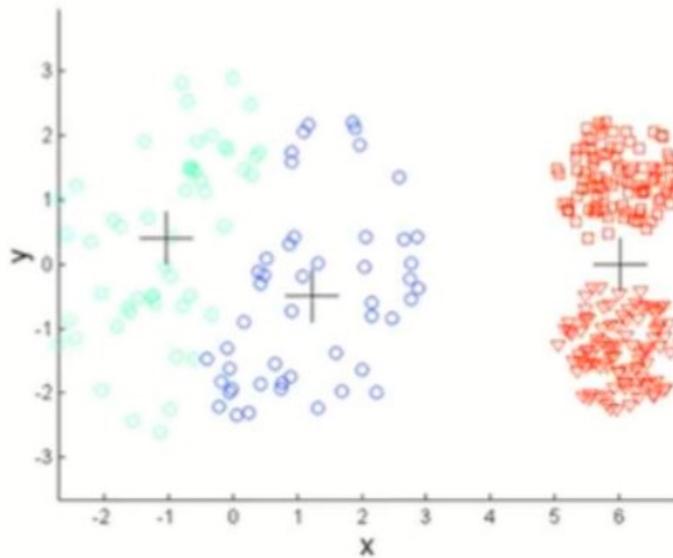
K-Means 문제점

- 서로 다른 밀도의 군집을 잘 찾지 못함

정답



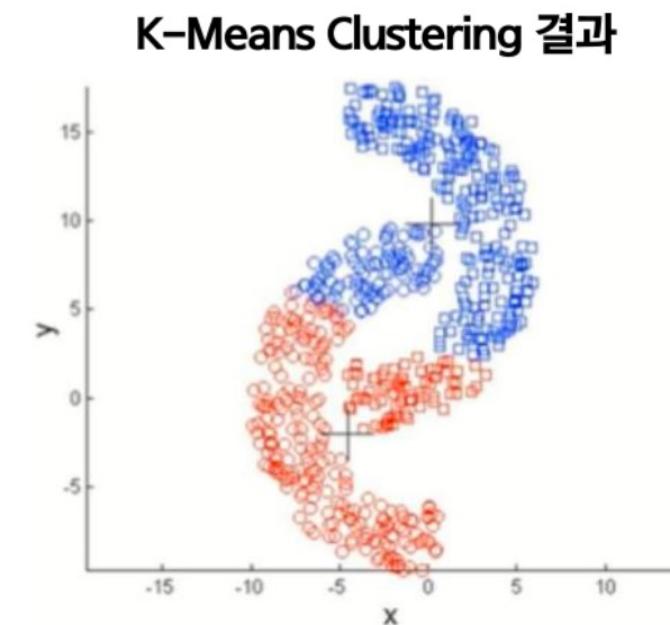
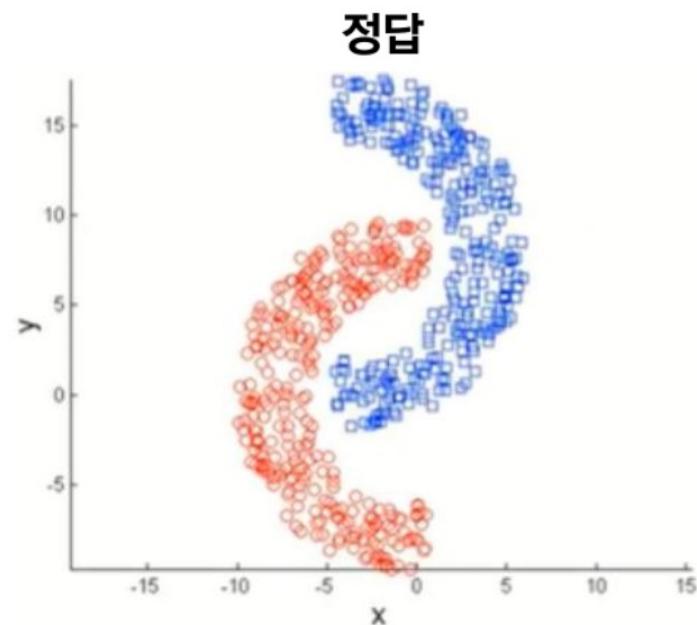
K-Means Clustering 결과



Unit 03 | K-Means & DBSCAN

K-Means 문제점

- 지역적 패턴이 있는 군집 판별에 어려움이 있음

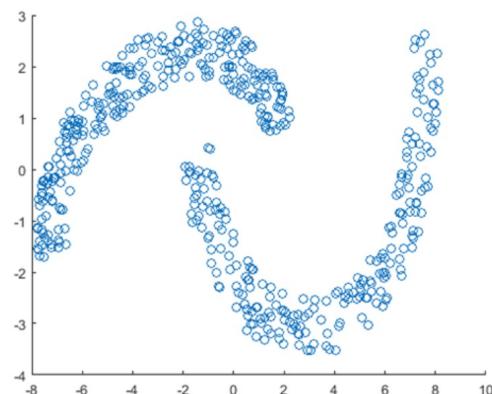


Unit 03 | K-Means & DBSCAN

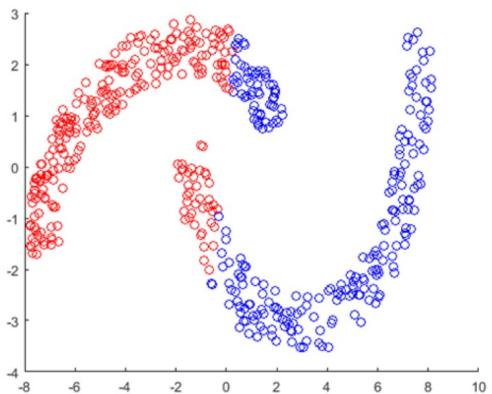
DBSCAN

Density-Based Spatial Clustering of Applications with Noise

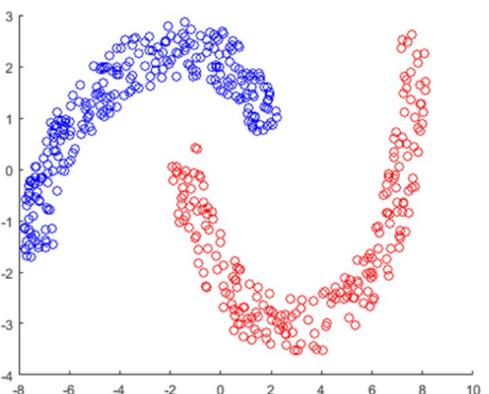
- 클러스터의 개수를 미리 지정할 필요 없음
- 복잡한 형상도 찾을 수 있으며, 어떤 클래스에도 속하지 않는 포인트를 구분할 수 있음
- 다소 느리지만 큰 데이터셋에도 적용 가능



(a) 원본 데이터



(b) k-means clustering의 결과



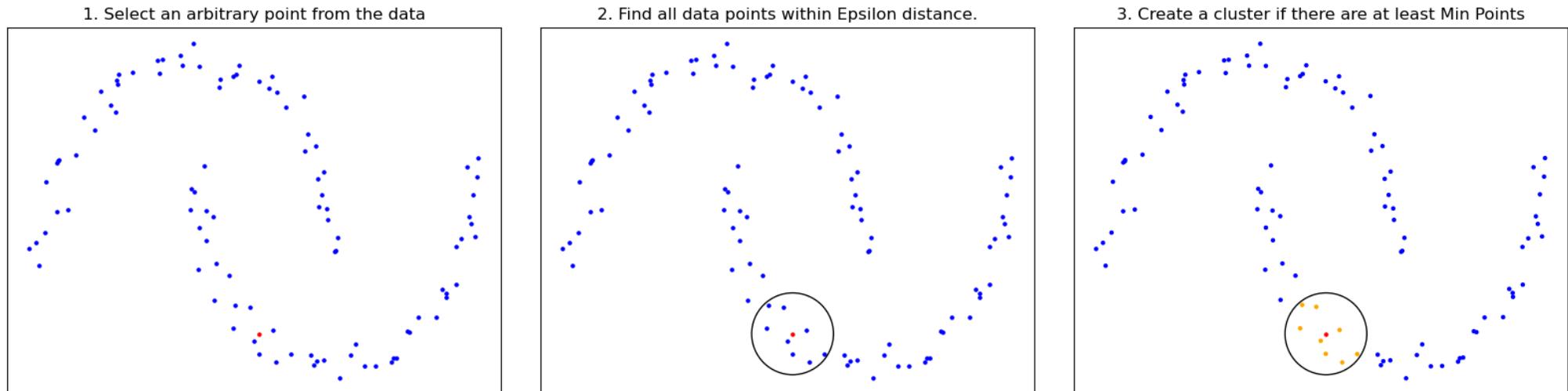
(c) DBSCAN의 결과

Unit 03 | K-Means & DBSCAN

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

- Epsilon : cluster 를 구성하는 최소의 거리
- Min points : cluster 구성 시, 필요한 최소 데이터 포인트 수

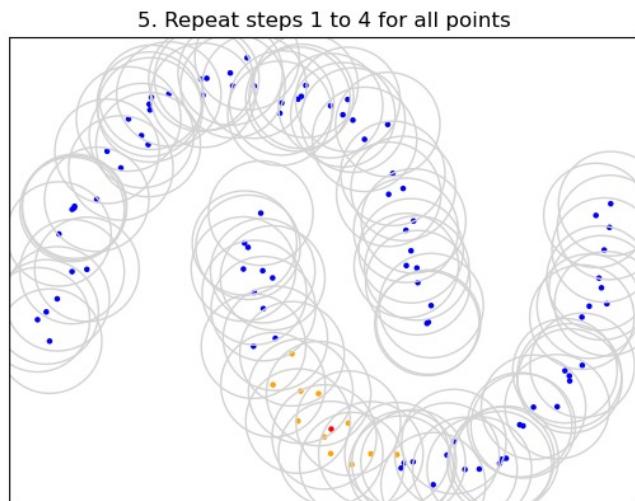
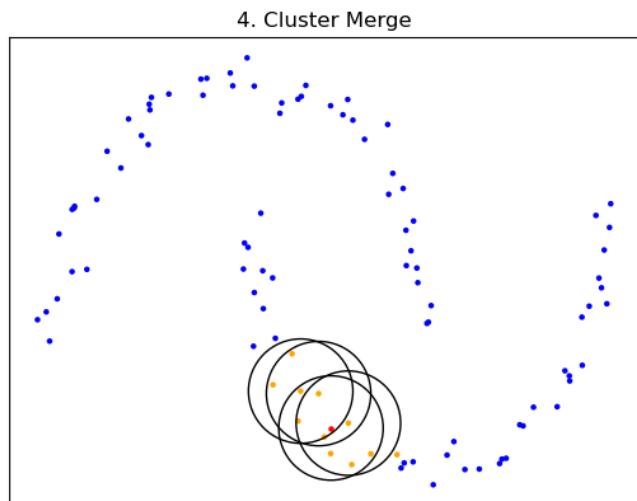


Unit 03 | K-Means & DBSCAN

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

- Epsilon : cluster 를 구성하는 최소의 거리
- Min points : cluster 구성 시, 필요한 최소 데이터 포인트 수



Unit 03 | K-Means & DBSCAN

DBSCAN 한계점

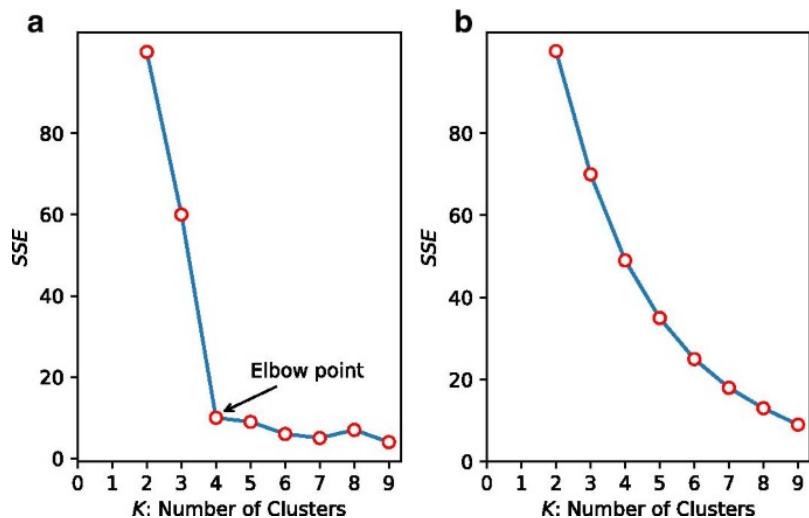
Density-Based Spatial Clustering of Applications with Noise

- 사전에 데이터에 대한 충분한 이해도를 갖고 있지 않다면 eps 와 min_samples 의 값을 정하기 어려움
- 연산량이 많아서 K-Means 에 비해 속도가 느림
- 차원의 저주 문제

Unit 04 | 모델평가

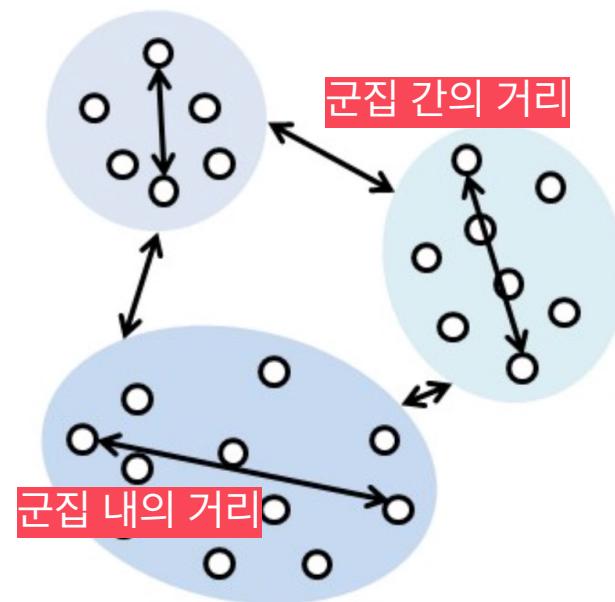
최적의 군집 수 결정

- 다양한 군집 수에 대해 성능평가 지표를 도시하여 최적의 군집 수를 선택한다.
- Elbow point에서 최적 군집 수가 결정되는 경우가 일반적이다.



Unit 04 | 모델평가

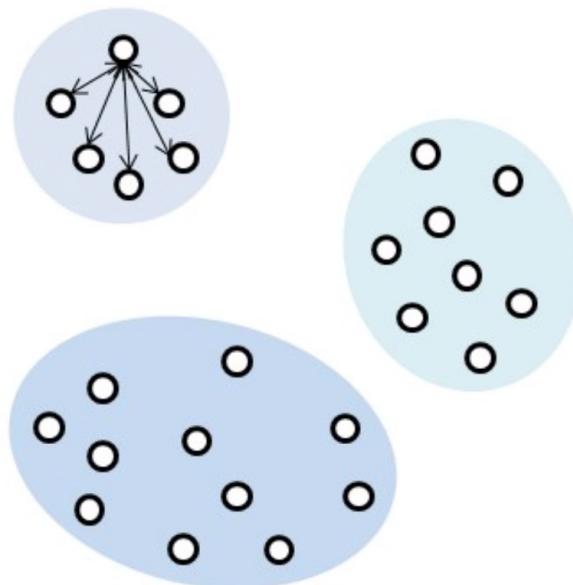
최적의 군집 수 결정 - Dunn Index



$$I(C) = \frac{\min_{i \neq j} \{d_c(C_i, C_j)\}}{\max_{1 \leq l \leq k} \{\Delta(C_l)\}}$$

- 분자 값이 크면 클수록 군집과 군집사이의 거리가 크고, 분모 값이 작으면 작을 수록 군집내의 데이터들이 모여 있으므로 Dunn index 가 클수록 군집이 잘 된 것으로 볼 수 있음

Unit 04 | 모델평가

최적의 군집 수 결정 - **Silhouette**

$a(i)$: 개체 i 로 부터 같은 군집 내에 있는 모든 개체들 사이의 평균 거리

$b(i)$: 개체 i 로 부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 가장 작은 값

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$
$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

Unit 04 | 모델평가

최적의 군집 수 결정 - **Silhouette**

$a(i)$: 개체 i 로 부터 같은 군집 내에 있는 모든 개체들 사이의 평균 거리

$b(i)$: 개체 i 로 부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 가장 작은 값

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

$$\bar{S} = \frac{1}{n} \sum_{i=1}^n S(i)$$

일반적으로 0.5 보다 크면 군집 결과가 타당,
-1에 가까우면 군집이 전혀 되지 않음

과제

주어진 데이터로 주석과 함께 자유롭게 분석을 진행해주세요 ☺

- **Preprocessing / EDA**
- **Clustering**
- **Evaluation**

참고자료

- 투빅스 18기 강효은님 강의자료
- 투빅스 17기 이지수님 강의자료
- 투빅스 16기 박한나님 강의자료
- 투빅스 13기 김민정님 강의자료

Q & A

들어주셔서 감사합니다.