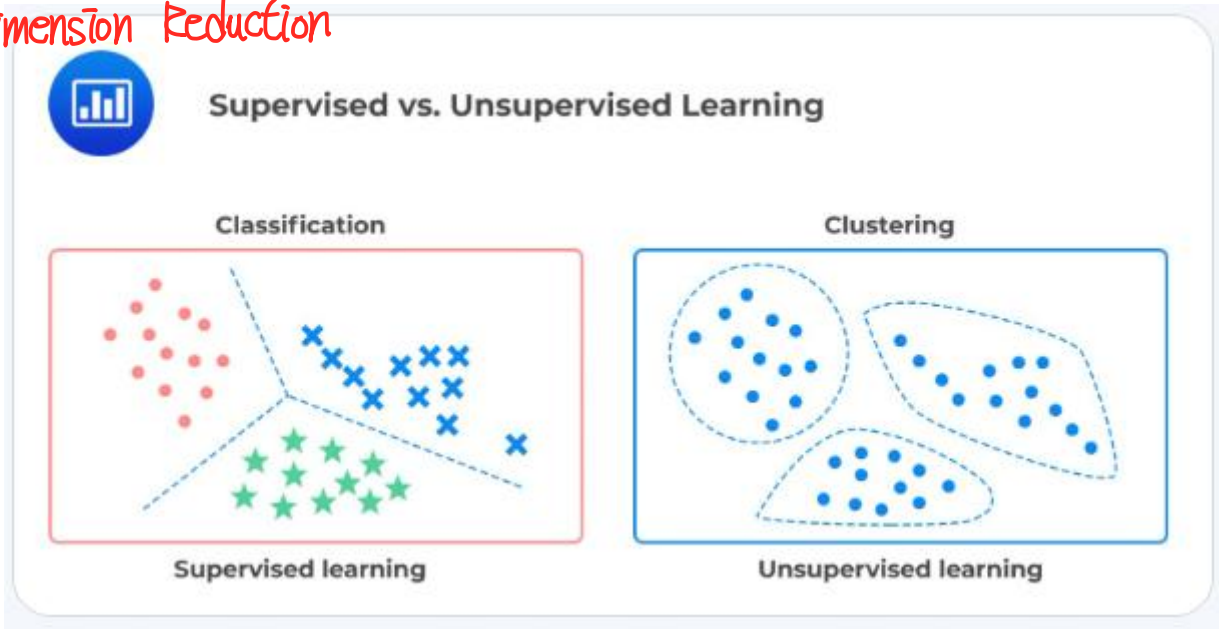
6. 군집화 알고리즘

목 차

서 론

군집화 개념

- 지도학습 **vs.** 비지도 학습
  - 분류
  - 회귀
  - clustering
  - Dimension Reduction



출처 : https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/

군집화의 적용 분야

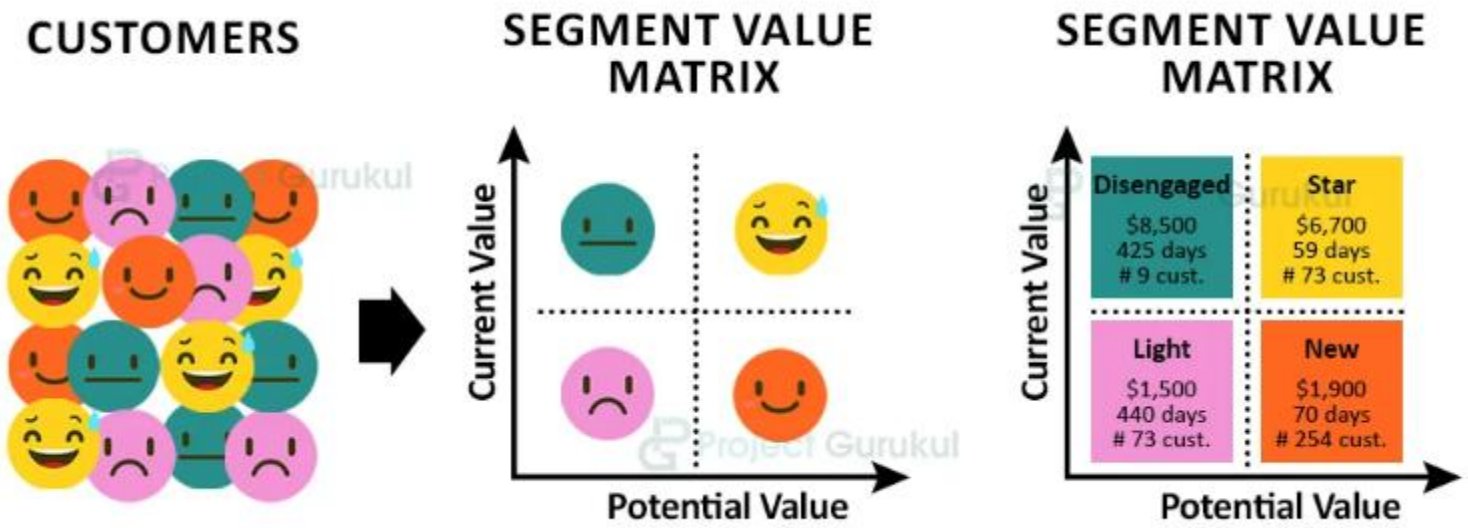
- Image segmentation with clustering

Example of clustering based image segmentation **a** Original image **b** Segmented image

출처 : https://link.springer.com/article/10.1007/s11042-021-10594-9

군집화의 적용 분야

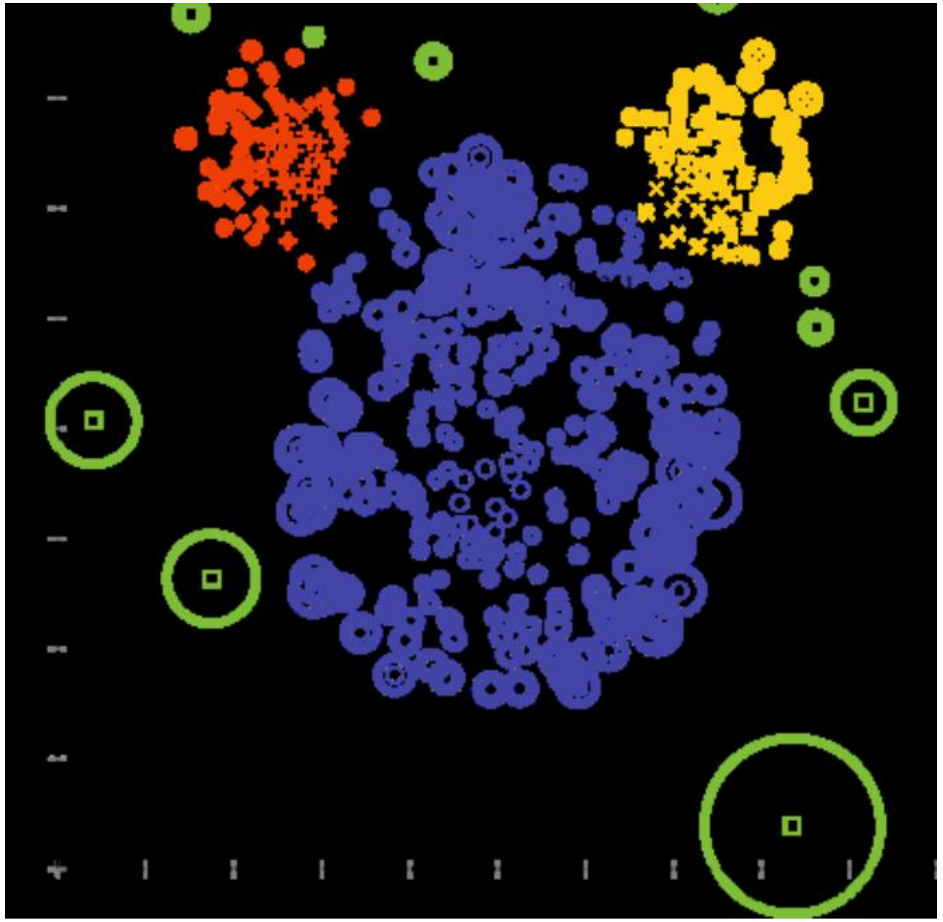
- Customer Segmentation



출처 : https://projectgurukul.org/customer-segmentation-project-machine-learning/

군집화의 적용 분야

- Fraud Detection
  사기 탐지



출처 : https://www.researchgate.net/figure/Fraud-Clustering-Classification-Example_fig2_344402267

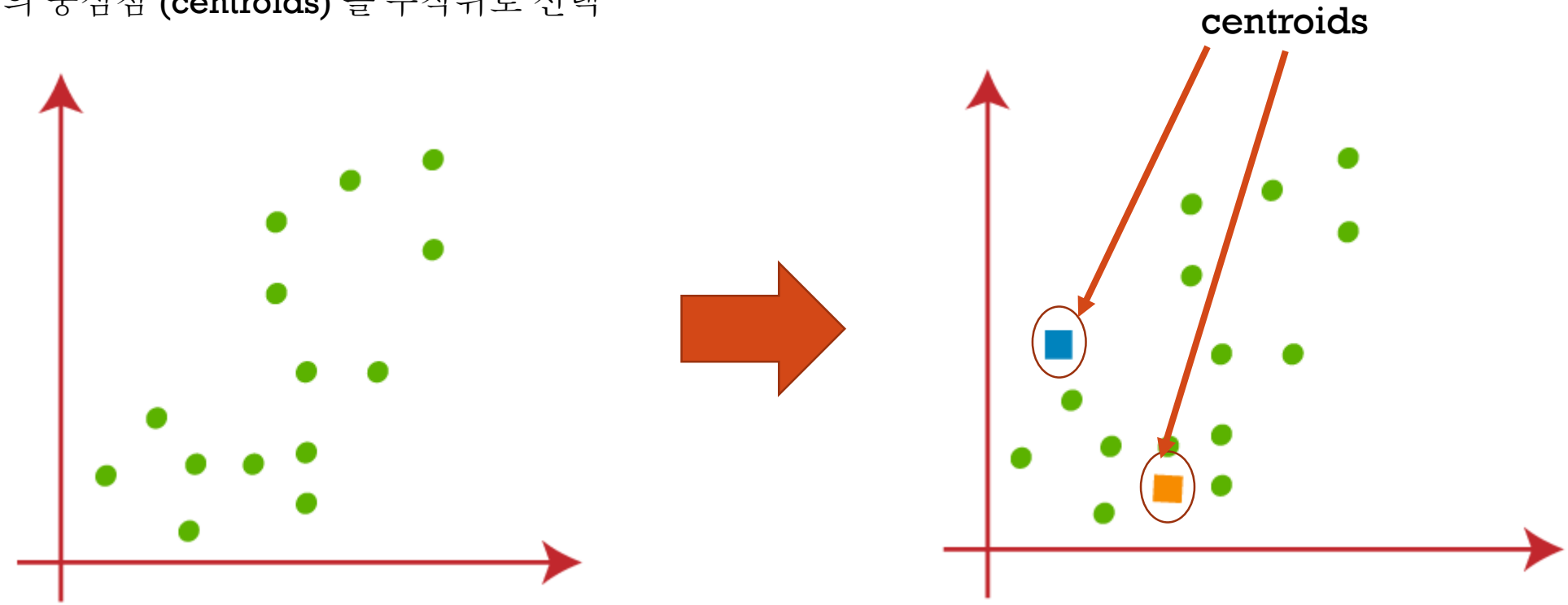
군집화 알고리즘 소개

K-MEANS 알고리즘

- **K-means** 알고리즘
  - 비지도 학습의 대표적인 군집화 알고리즘
  - 데이터 포인트를 **K**개의 그룹으로 나눔

K-MEANS 알고리즘

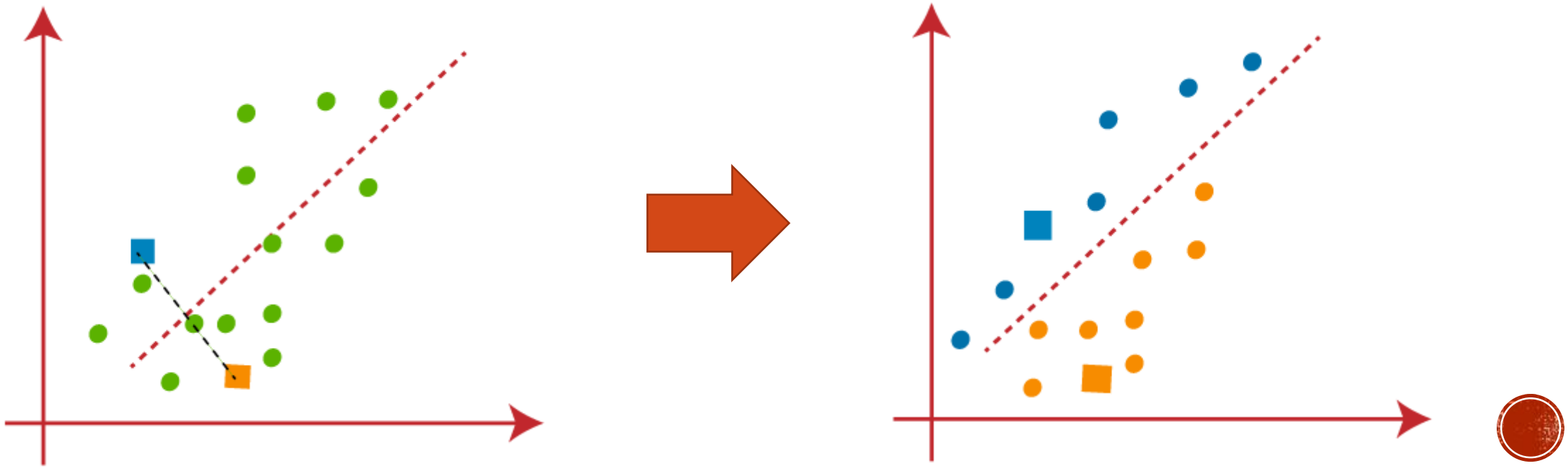
- 알고리즘의 주요 절차
  - 초기화 (Initialization)
    - K개의 중심점 (centroids) 을 무작위로 선택



출처 : https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

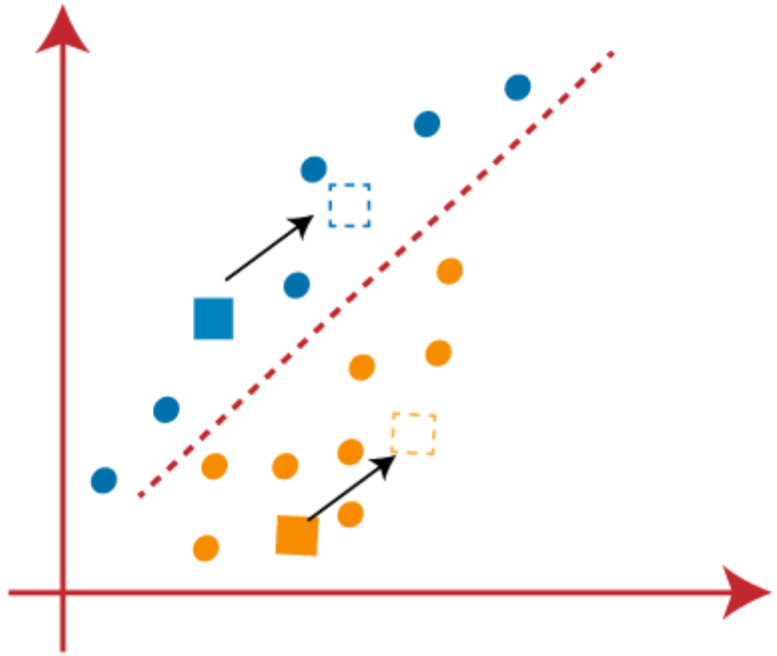
K-MEANS 알고리즘

- 할당 단계 **(Assignment Step)**
  - 각 데이터 포인트는 가장 가까운 센트로이드에 할당됨
  - 유클리드 거리로 계산
  - 자신과 과장 가까운 중심점을 기준으로 클러스터에 배정



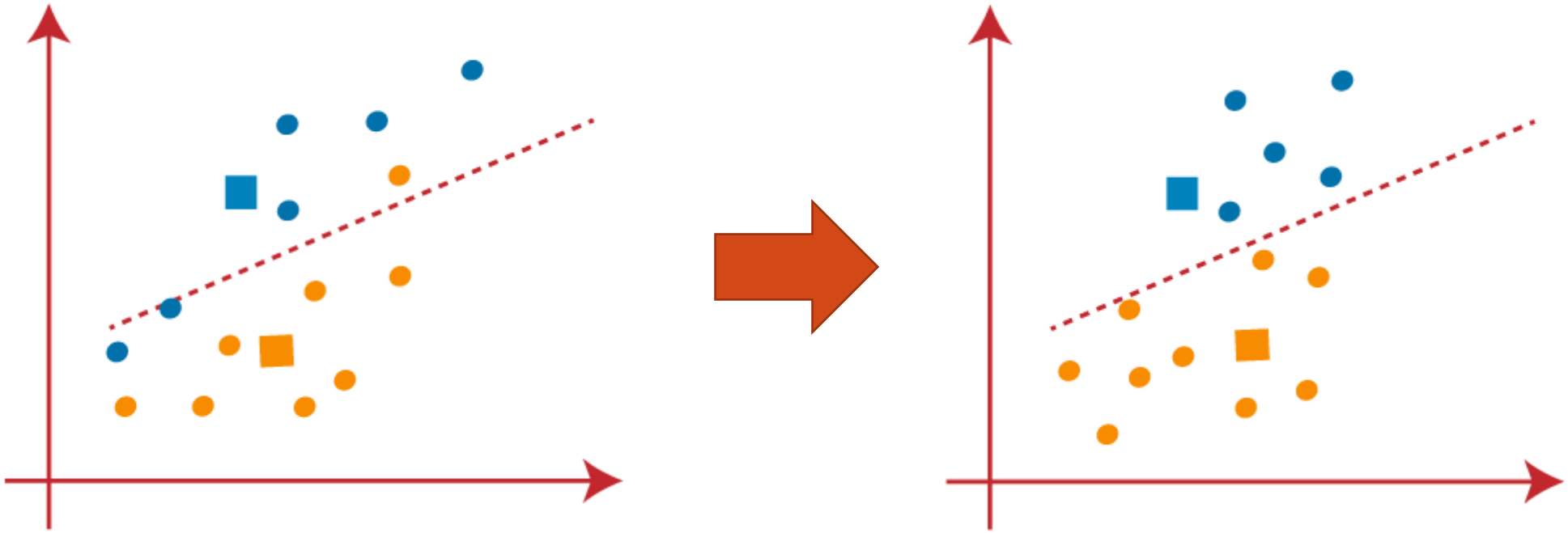
K-MEANS 알고리즘

- 중심점 업데이트 **(Update Step)**
  - 각 클러스터별로 평균을 계산하여 새로운 중심점 발굴

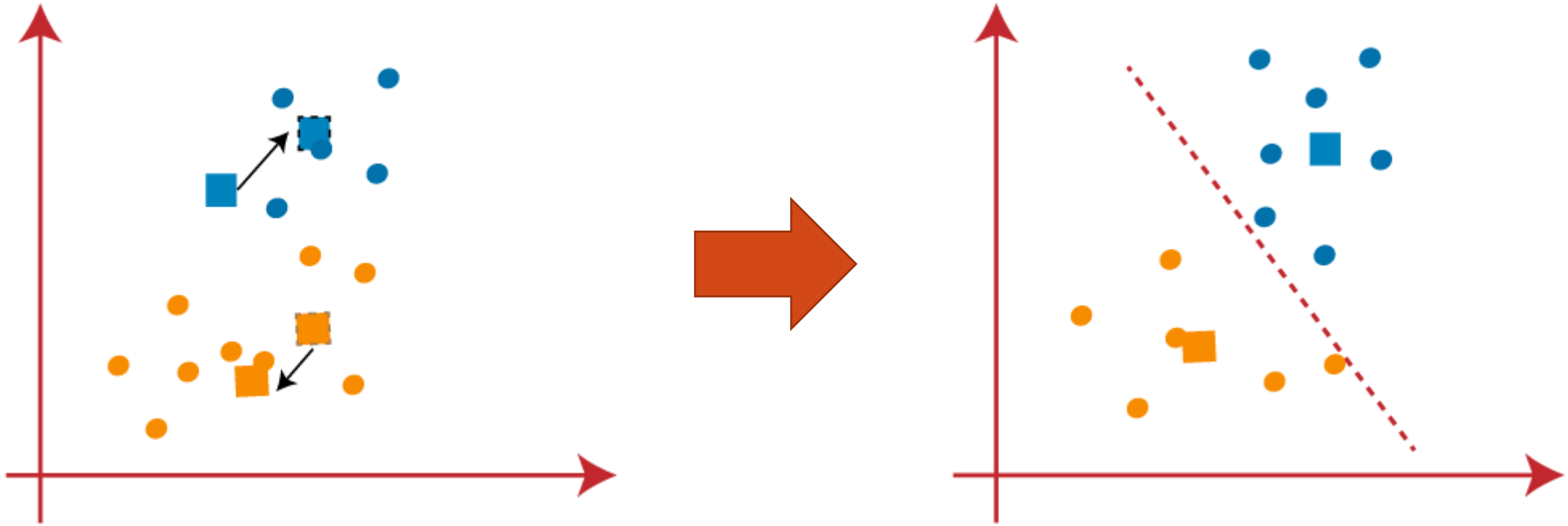


K-MEANS 알고리즘

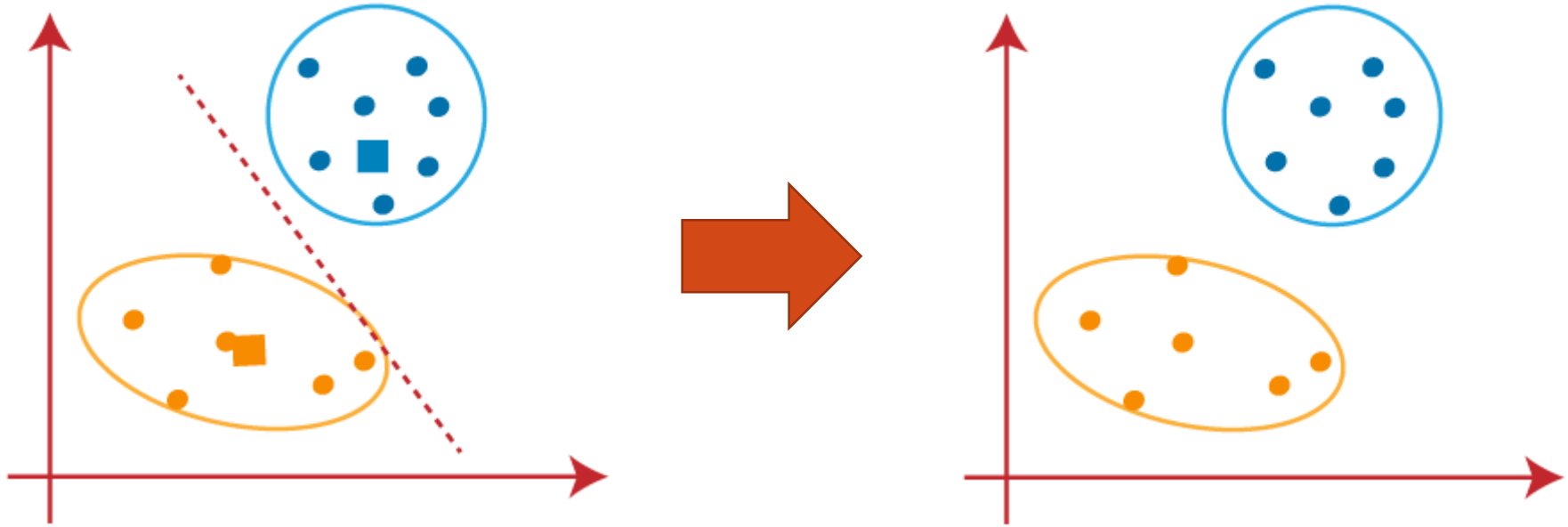
- 반목 (Iterate)
  - 할당 단계와 업데이트 단계를 반복
  - 데이터 포인트가 더 이상 다른 클러스터로 이동하지 않거나 중심점의 변화가 매우 작아질 때까지



K-MEANS 알고리즘

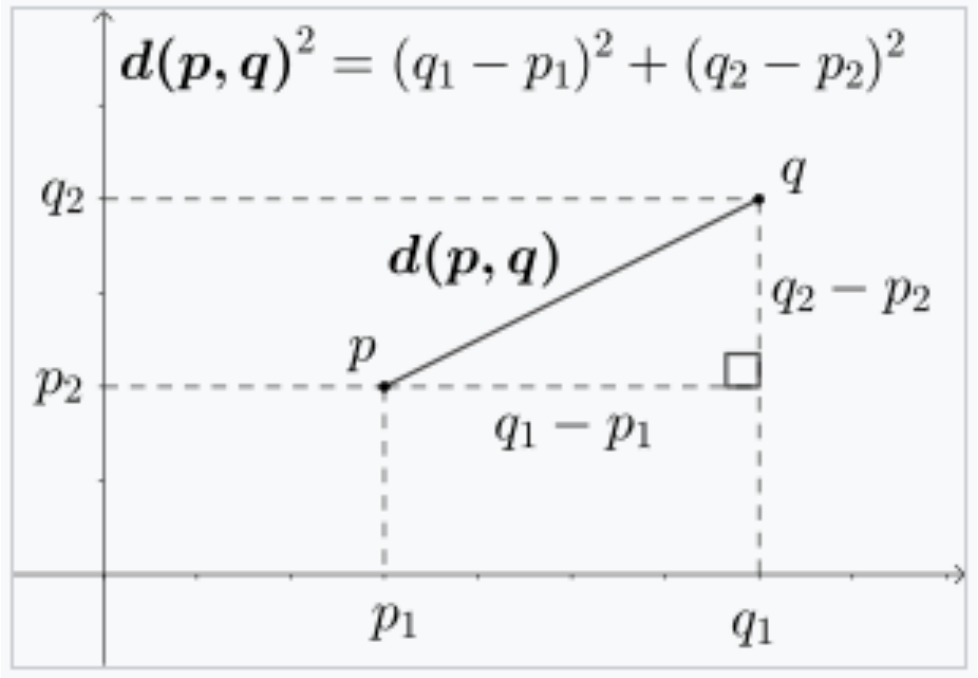


K-MEANS 알고리즘



K-MEANS 알고리즘

- Euclidean Distance



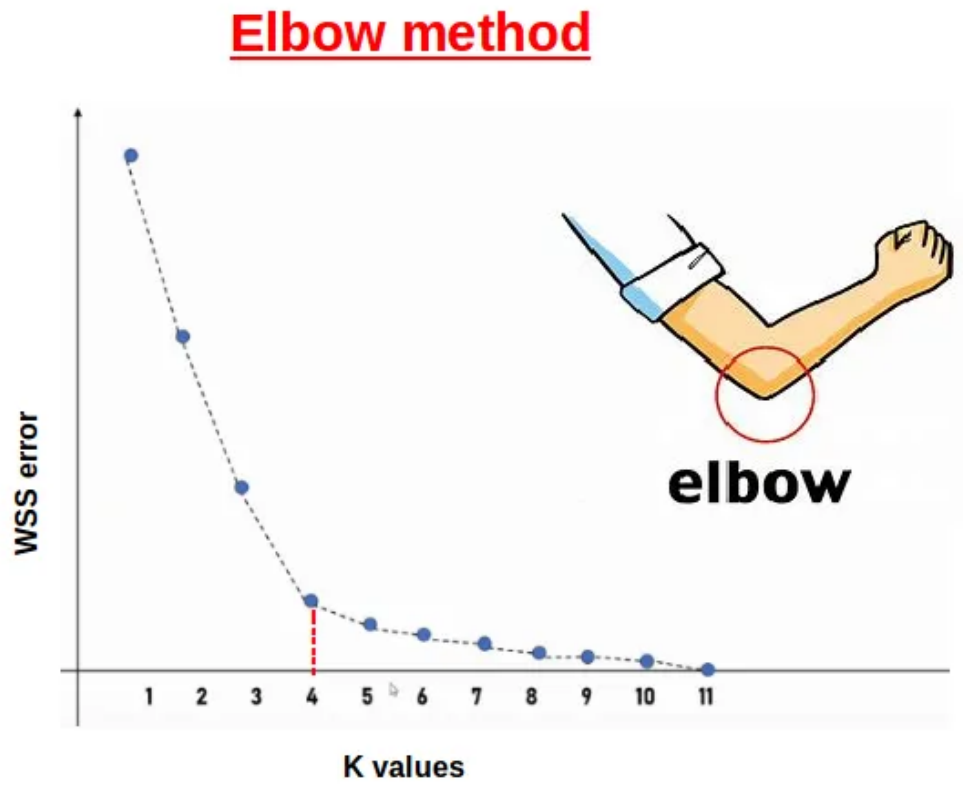
출처 : https://en.wikipedia.org/wiki/Euclidean_distance

K-MEANS 알고리즘

- 초기화의 민감성
  - 지역 최적화 (**Local Optima**)
    - 잘못된 초기 중심정을 설정하면 최종적으로 비효율적인 클러스터가 형성
    - 초기 중심점이 데이터 분포에서 한쪽으로 치우쳐져 있으면 전체 데이터 구조를 반영하지 못함
  - 재시작
    - 여러 번 초기화 한 후 가장 낮은 비용을 가지는 결과를 선택
      - 클러스터 내 데이터와 중심점 간 거리의 합이 가장 낮은 경우를 선택 (**K-means++** 알고리즘)

K-MEANS 알고리즘

- 군집 수(K)의 설정 중요성
  - 과적합과 과소적합 문제
    - K 값이 너무 크면 지나치게 많은 군집으로 분할되어 과적합 발생
    - K 값이 너무 작으면 서로 다른 특성의 데이터가 같은 클러스터로 할당되는 과소적합 발생
  - 엘보우 기법
    - K 값을 증가시키면서 각 군집 내 거리의 합이 급격히 줄어드는 구간을 K 로 선택



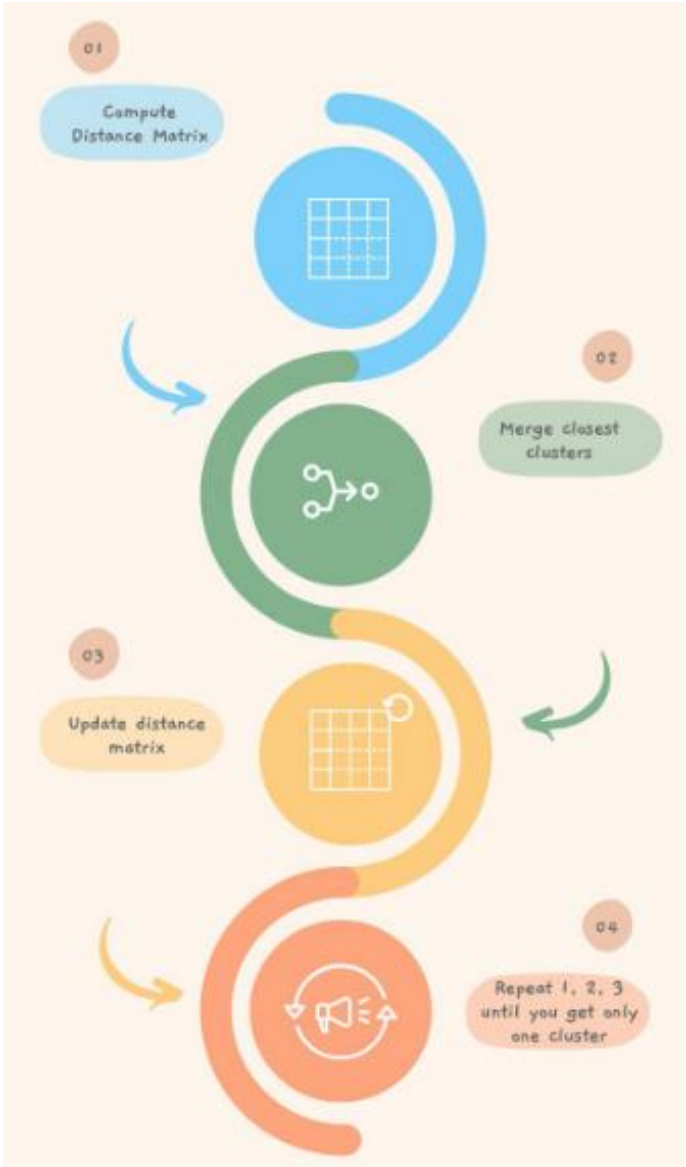
출처 : https://medium.com/@zalarushirajsinh07/the-elbow-method-finding-the-optimal-number-of-clusters-d297f5aeb189

계층적 군집화

- 계층적 군집화 **(Hierarchical Clustering)**
  - 데이터 포인트들을 계층적으로 군집화
  - 데이터셋을 트리 구조로 표현
  - 각 데이터 포인트들이 서로 어떻게 군집화되는지를 시각적으로 이해
  - 병합적 방법 **(agglomerative)** 와 분할적 방법 **(divisive)** 이 있음

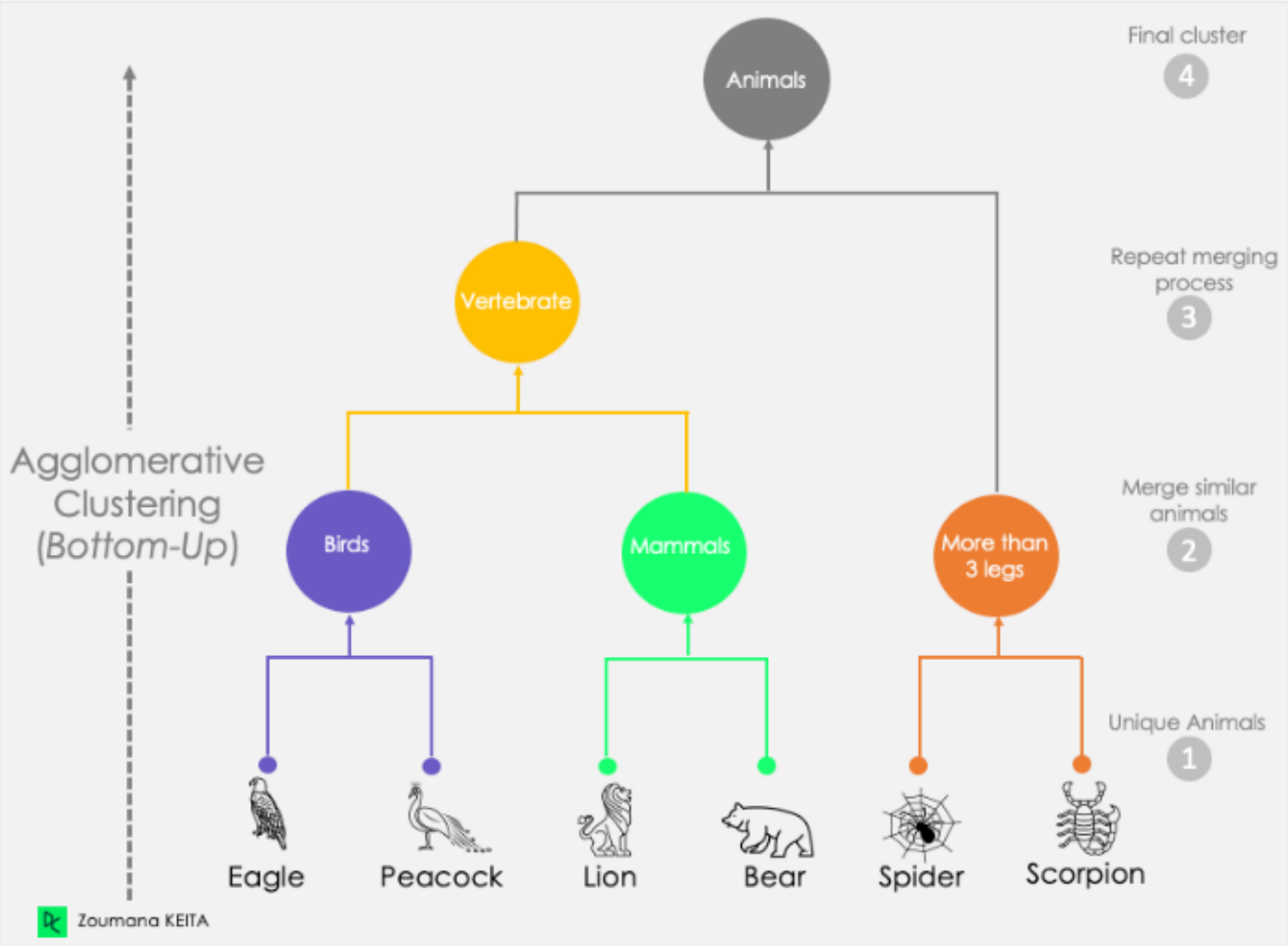
계층적 군집화

- 병합적 방법 (**Agglomerative Clustering**)
  - 상향식 (**Bottom-Up**) 방법
  - 각 데이터 포인트가 개별 클러스터로 지정
  - 가장 가까운 두 클러스터를 반복적으로 병합
  - 전체 데이터를 하나의 클러스터로 묶음
  - 덴드로그램 (**Dendrogram**) 이라는 트리 구조로 시각화



출처 : **https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python**

계층적 군집화



Dendrogram of Agglomerative Clustering Approach

출처 : https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python

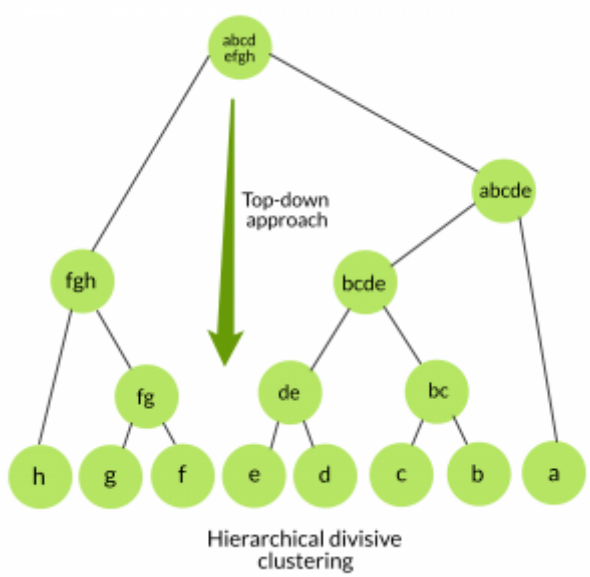
계층적 군집화

- 분할적 방법 (**Divisive Clustering**)
  - 모든 데이터 포인트를 하나의 클러스터에 소속시킴
  - 클러스터를 반복적으로 분할
  - 데이터의 전체 구조를 먼저 고려한 후 세분화

cluster

1. cluster 나뉨

2. cluster 나뉨



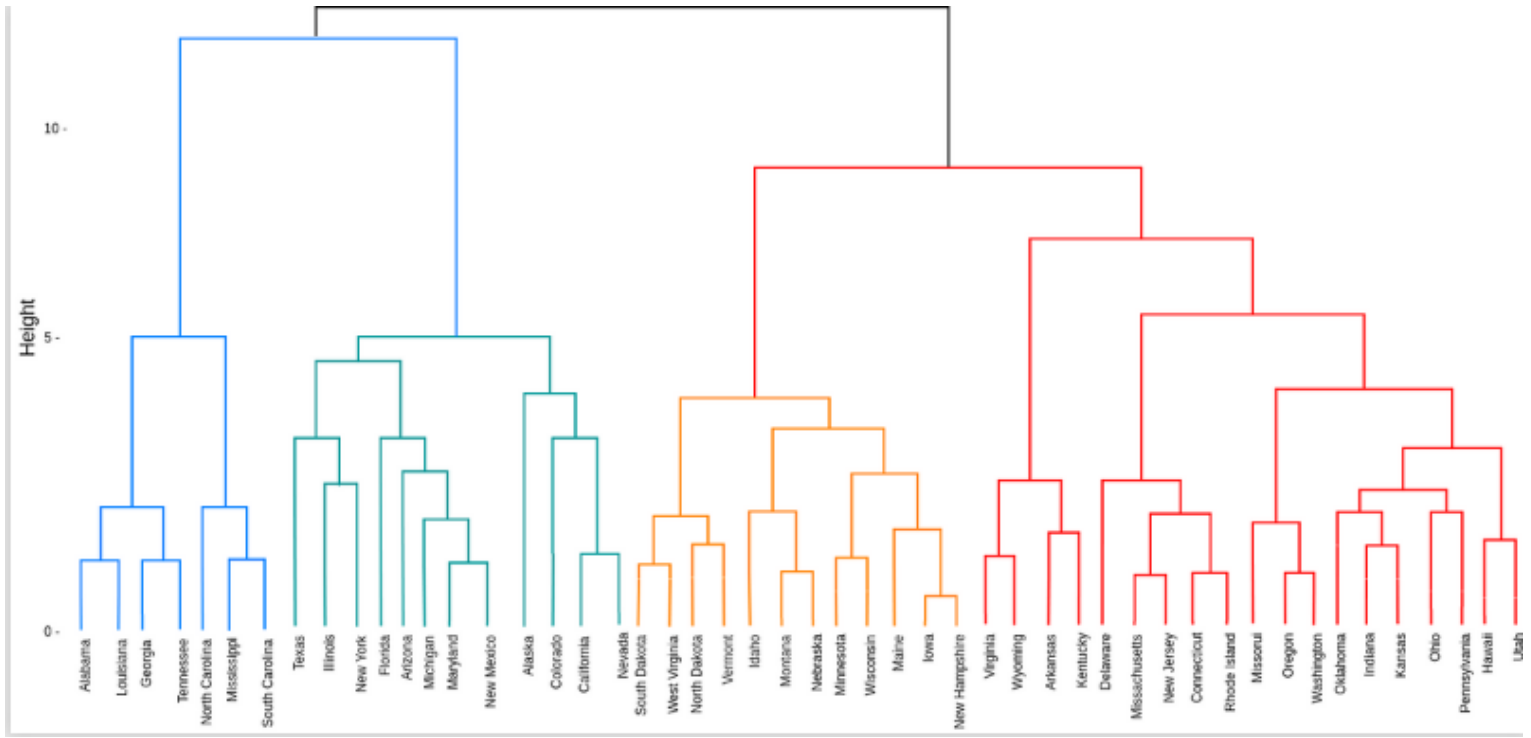
출처 : https://www.quantace.in/hierarchical-clustering-financial-market-analysis-india/

계층적 군집화

- 거리 계산 방법
  - 최단 거리 **(Single Linkage)**
    - 두 클러스터에서 가장 가까운 두 데이터 포인트 사이의 거리가 기준
  - 최장 거리 **(Complete Linkage)**
    - 두 클러스터에서 가장 먼 두 데이터 포인트 사이의 거리가 기준
  - 평균 거리 **(Average Linkage)**
    - 두 클러스터 간의 모든 데이터 포인트 간 평균 거리가 기준

계층적 군집화

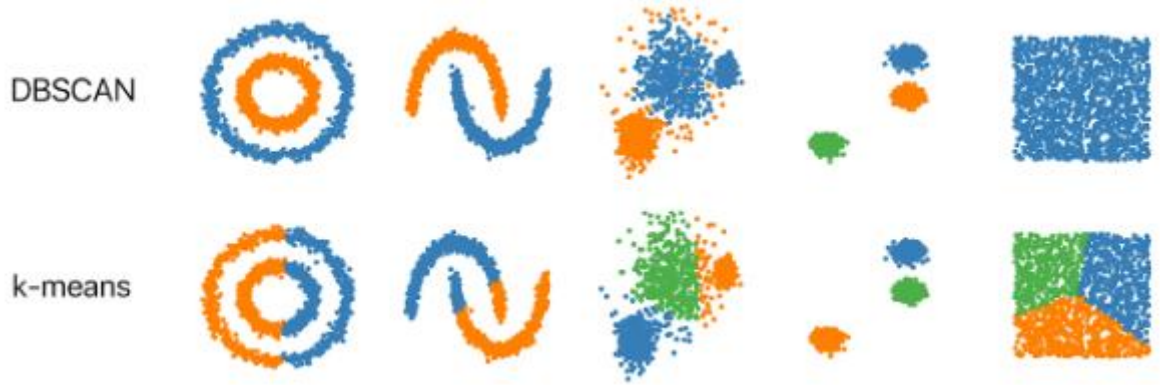
- **Dendrogram**
  - 계층적 군집화의 결과를 시각화하는 도구



출처 : https://online.visual-paradigm.com/ko/diagrams/templates/dendrogram/cluster-dendrogram/

DBSCAN

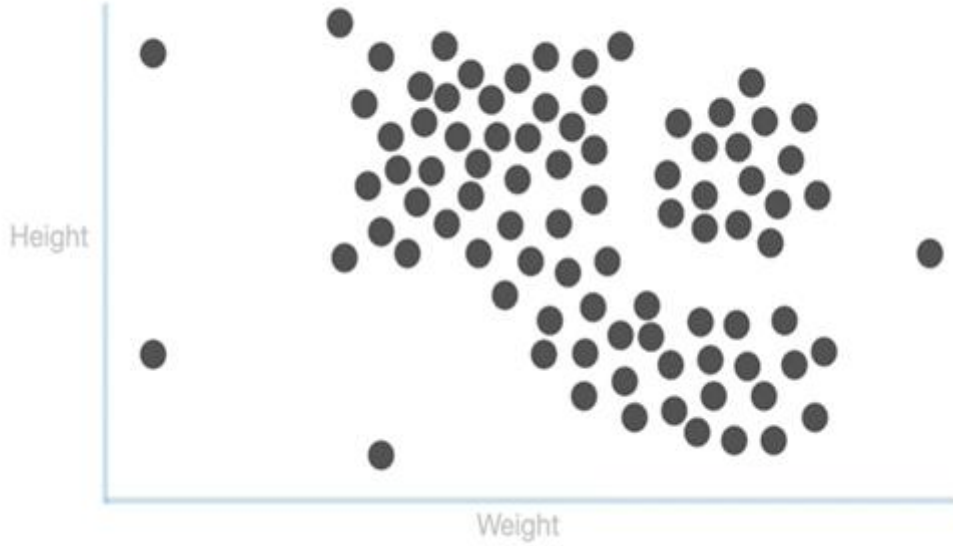
- **DBSCAN (Density-based Spatial Clustering of Application with Noise,** 밀도 기반 클러스터링)
  - 군집의 모양과 데이터의 밀도에 기반하여 클러스터를 형성
  - 미리 지정된 밀도 기준을 만족하는 데이터 포인트들을 하나의 군집으로 묶음
  - 이상치 **(outlier)** 를 효율적으로 처리



출처 : https://minutemaid.tistory.com/82

DBSCAN

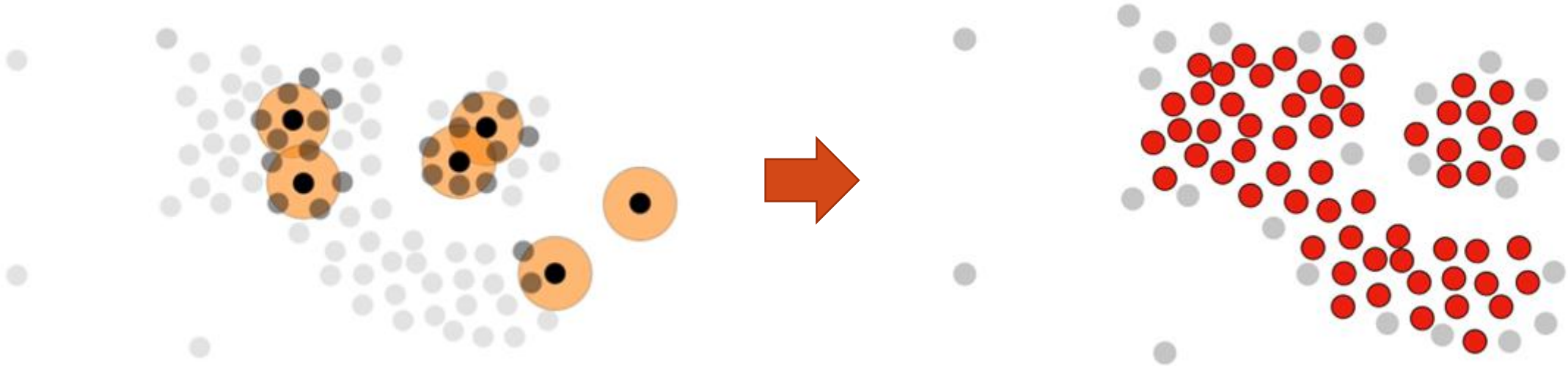
- 밀도 기반 클러스터링
  - 데이터가 세밀하게 몰려 있어서 밀도가 높은 부분을 클러스터링
  - 어느 점을 기준으로 반경 **x** 내에 점이 **n**개 이상 있으면 하나의 군집으로 인식



출처 : https://blog.naver.com/march03190/222792748678

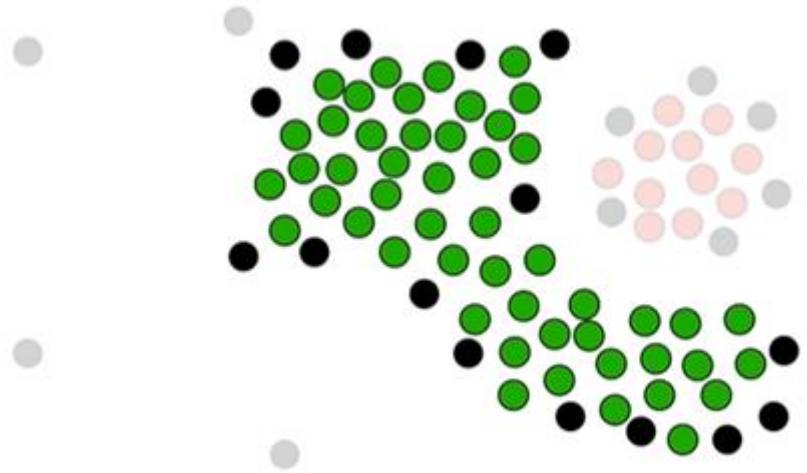
DBSCAN

- **Eps** (반지름), **Minpts** (최소 점 개수) 활용
- **Core Points:** 해당 원 내에 **minpts** 이상의 점들을 포함하고 있는 점들



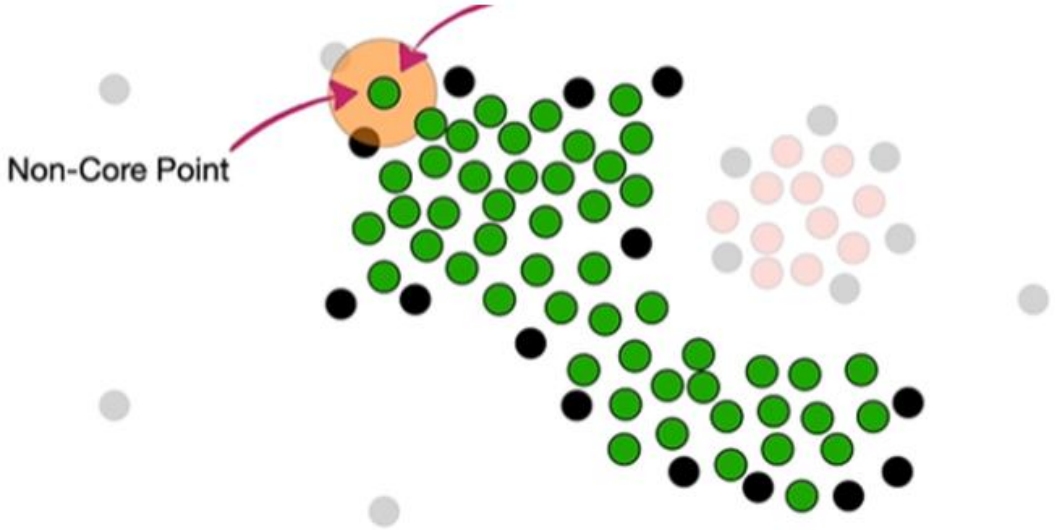
DBSCAN

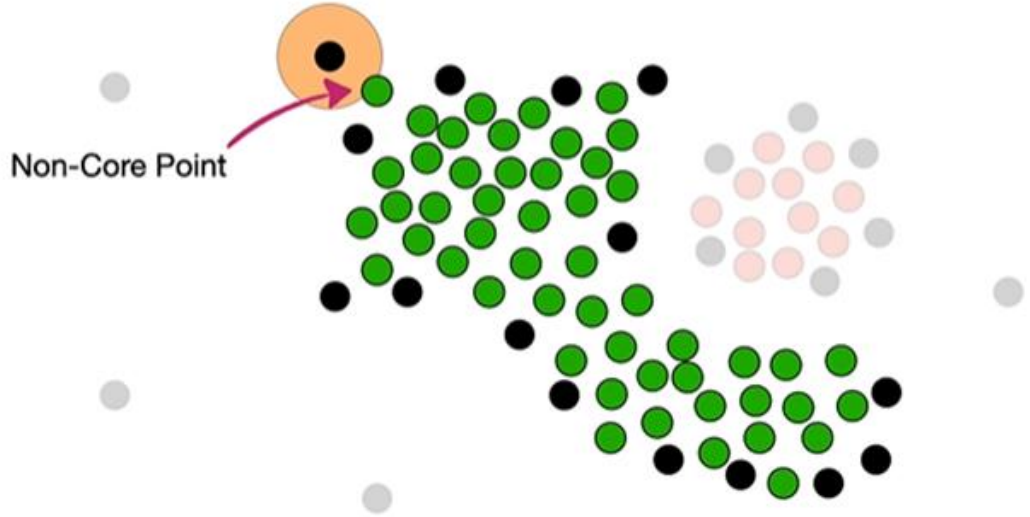
- **Core-points** 의 그루핑
  - 임의의 한점의 원 내에 속하게 되면 하나의 그룹으로 분류
  - **Core-points** 내에서만 분류



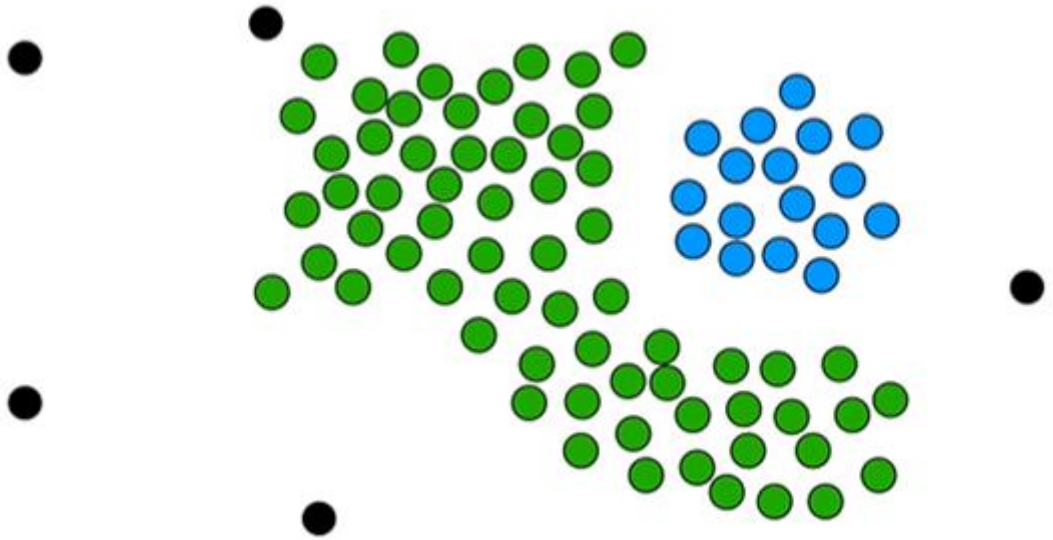
DBSCAN

- Non-Core points 처리
  - 원을 그렸을 때 **core-point** 를 포함하면 해당 그룹으로 분류
  - 원을 그렸을 때 **core-point** 로 분류된 **non-core point**를 포함하면 미분류





DBSCAN



알고리즘 비교 분석

- **K-Means** 거리만 측정 · 간단함.
  - 장점
    - 빠른 계산 속도, 간단한 구현
    - 대규모 데이터에서도 용이하게 작동
    - 구체적인 클러스터 개수를 지정하여 예측 가능한 클러스터의 개수를 얻을 수 있다
  - 단점
    - 적절한 **K** 를 찾기 어려움
    - 비정형적(비구형) 데이터에 비적합
    - 노이즈나 이상치에 민감
  - 활용 사례
    - 고객 세분화
      - 마케팅에서 비슷한 성향을 가진 고객 그룹을 식별
      - 소득 수준과 소비 패턴을 기준으로 분류

알고리즘 비교 분석

- 계층적 군집화 (**Hierarchical Clustering**)
  - 장점
    - 클러스터의 개수를 사전에 지정할 필요가 없음
    - 데이터 간의 계층적 관계 시각화 가능 (덴드로그램)
    - 다양한 거리 측정 방법 적용 가능
  - 단점
    - 데이터가 커질수록 계산복잡도가 증간
    - 한번 클러스터가 형성되면 병합이나 분할이 불가
    - 노이즈와 이상치에 민감
  - 활용 사례
    - 문서 분류
    - 논문이나 기사 같은 텍스트 데이터를 주제별로 그룹화

알고리즘 비교 분석

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
  - 장점
    - 클러스터의 모양에 상관없이 비구형 클러스터를 잘 찾음
    - 노이즈와 이상치에 강함
    - 클러스터의 개수를 미리 지정할 필요가 없음
  - 단점
    - 밀도 파라미터 **(eps, minPts)** 설정이 어려움
    - 밀도가 균일하지 않은 경우 성능이 떨어짐
    - 고차원 데이터에서 성능 저하
  - 활용 사례
    - 지리적 데이터 분석
    - 도시 내에서 교통사고 다발 지역 클러스터링 등

결 론

요약 표

알고리즘	장점	단점	활용 사례
K-means	빠르고 간단함, 대규모 데이터에 적합	클러스터 개수 지정 필요, 비구형 데이터에 부적합	고객 세분화
계층적 군집화	클러스터 개수 사전 지정 불필요, 계층적 구조 제공	큰 데이터에서 느림, 노이즈에 민감	문서 분류
DBSCAN	비구형 클러스터와 노이즈 처리에 강함, 클러스터 개수 자동 결정	밀도 설정이 어려움, 고차원 데이터에 성능 저하	지리적 데이터 분석