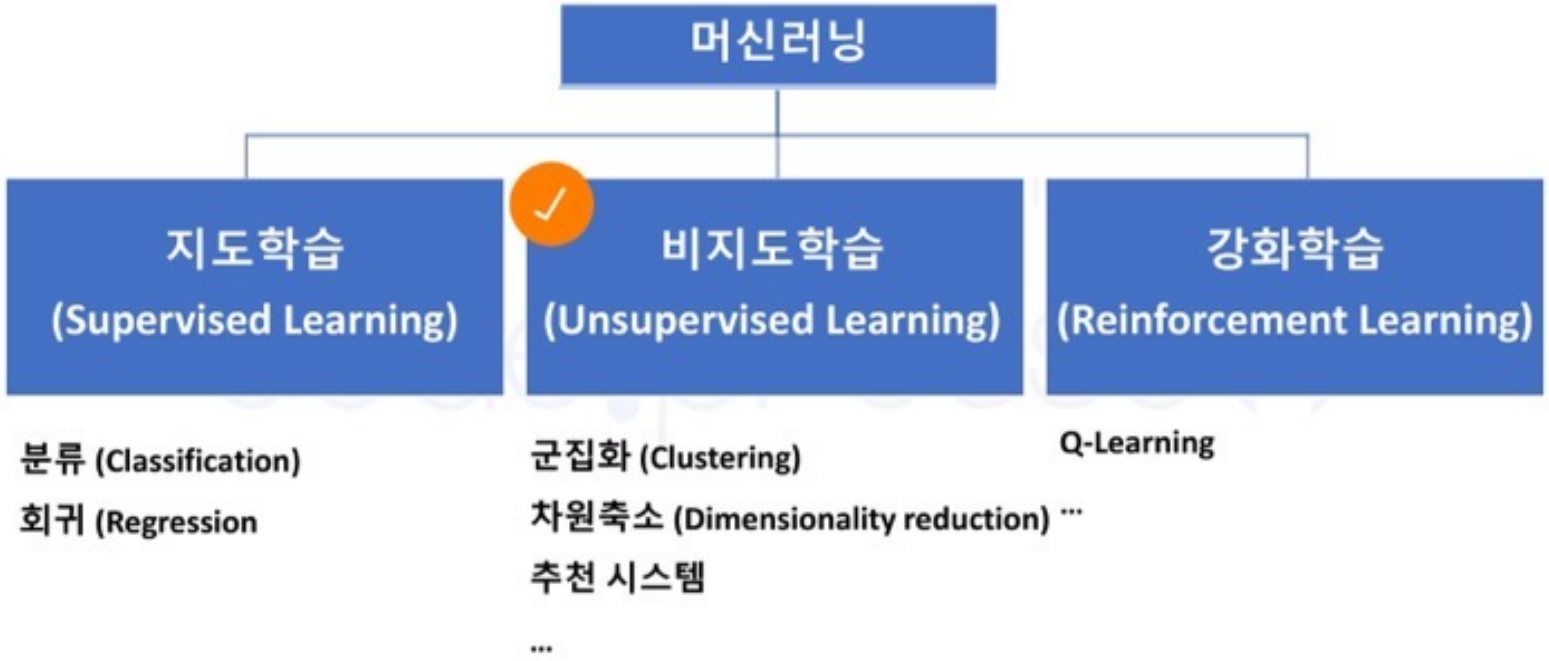
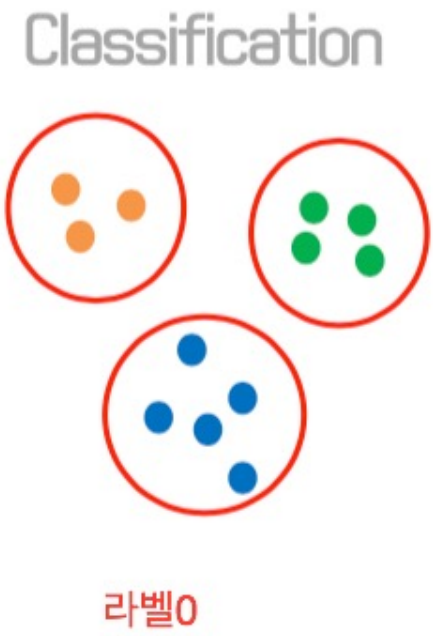

# 머신러닝 - 군집화

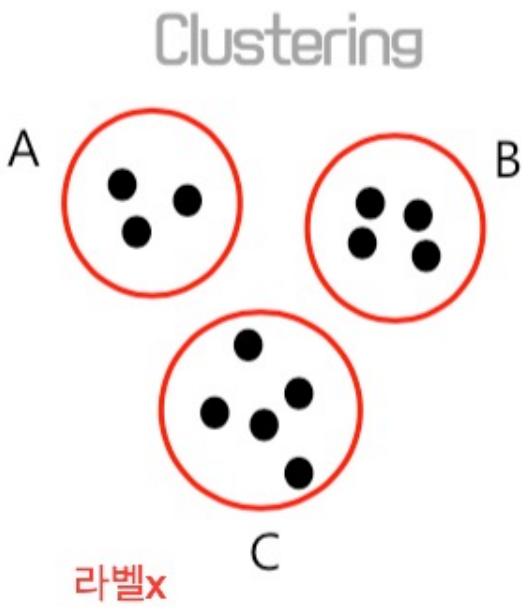
# 비지도 학습


머신러닝
지도학습
(Supervised Learning)
비지도학습
(Unsupervised Learning)
강화학습
(Reinforcement Learning)
분류 (Classification)
회귀 (Regression
군집화 (Clustering)
차원축소 (Dimensionality reduction) ···
추천 시스템
···
Q-Learning

# 군집화

## 군집화란?



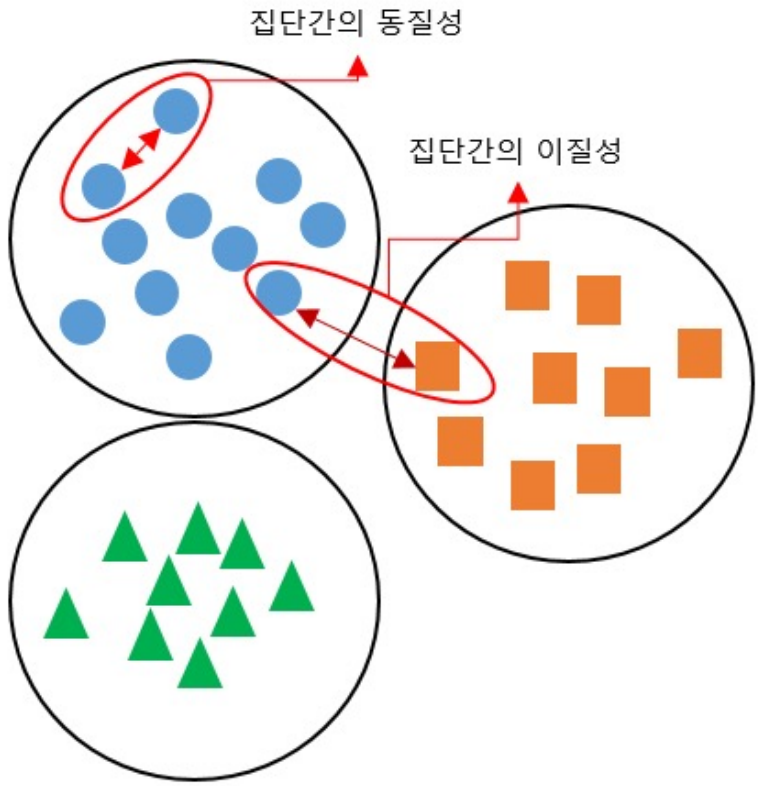



# 군집화

- 데이터 포인트들을 별개의 군집으로 그룹화하는 것을 의미

- 유사성이 높은 데이터들을 동일한 그룹으로 분류하고 서로 다른 군집들이 상이성을 가지도록 그룹화한다.

# 군집화



# 군집화

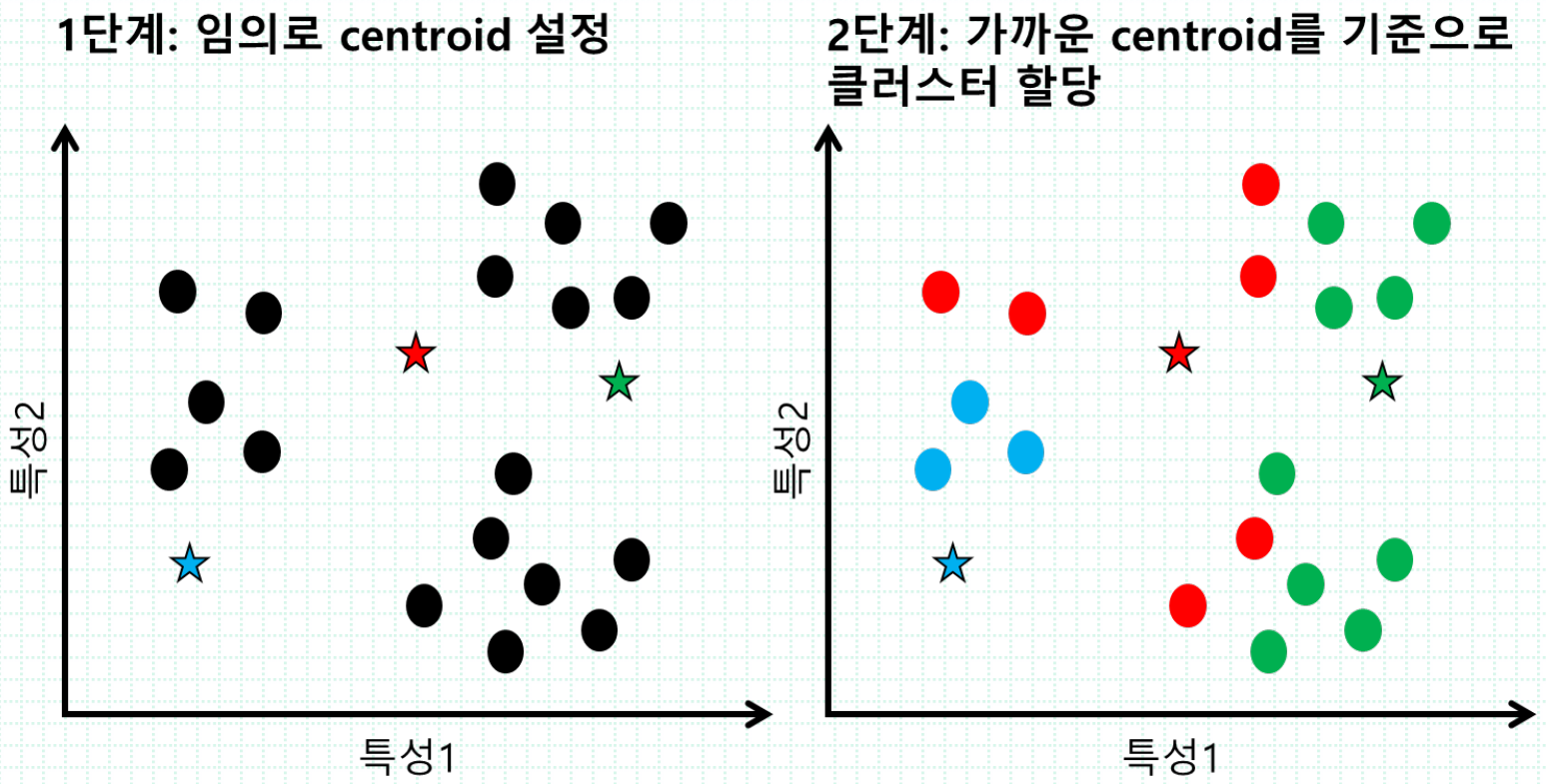
| 클러스터링에서 가장 많이 사용되는 알고리즘 → K – 평균(Means) Clustering

| K – 평균은 군집 중심점(centroid)이라는 특정한 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법

| 군집 중심점은 선택된 포인트의 평균 지점으로 이동하고 이동된 중심점에 다시 가까운 포인트를 선택 → 반복

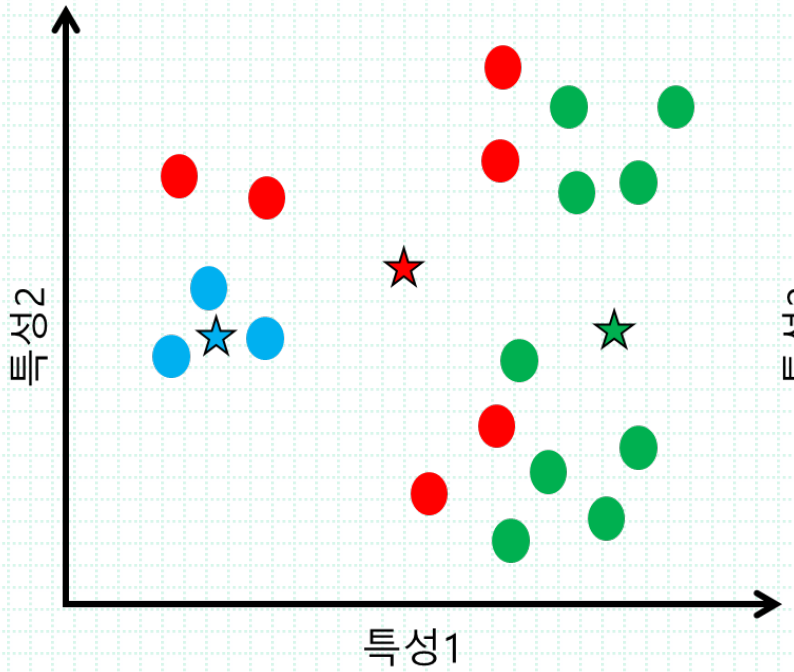
| 더이상 중심점의 이동이 없을 경우에 반복을 멈추고 해당 중심점에 속하는 데이터 포인트들을 군집화하는 기법

# 군집화

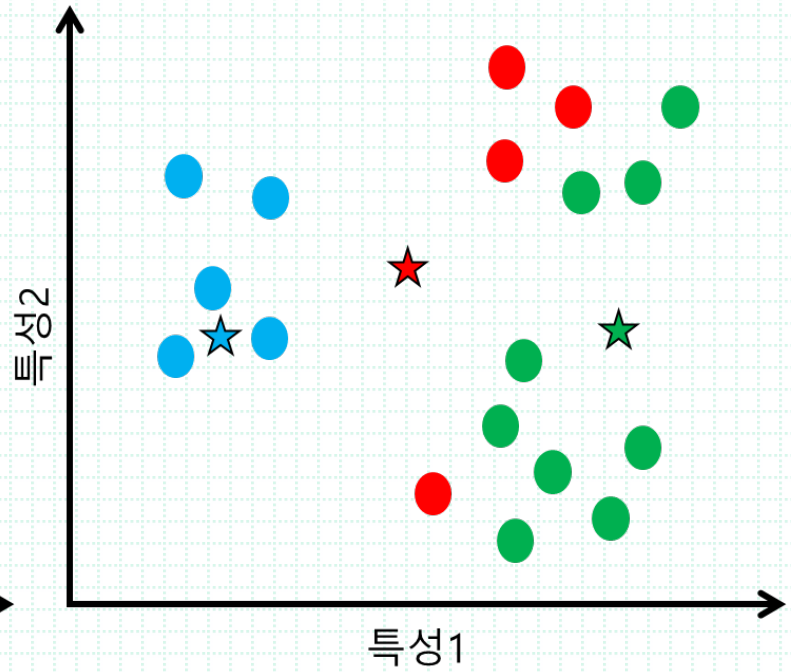

1단계: 임의로 centroid 설정
특성2
특성1
2단계: 가까운 centroid를 기준으로 클러스터 할당
특성2
특성1

# 군집화

3단계: 각 클러스터 centroid 갱신

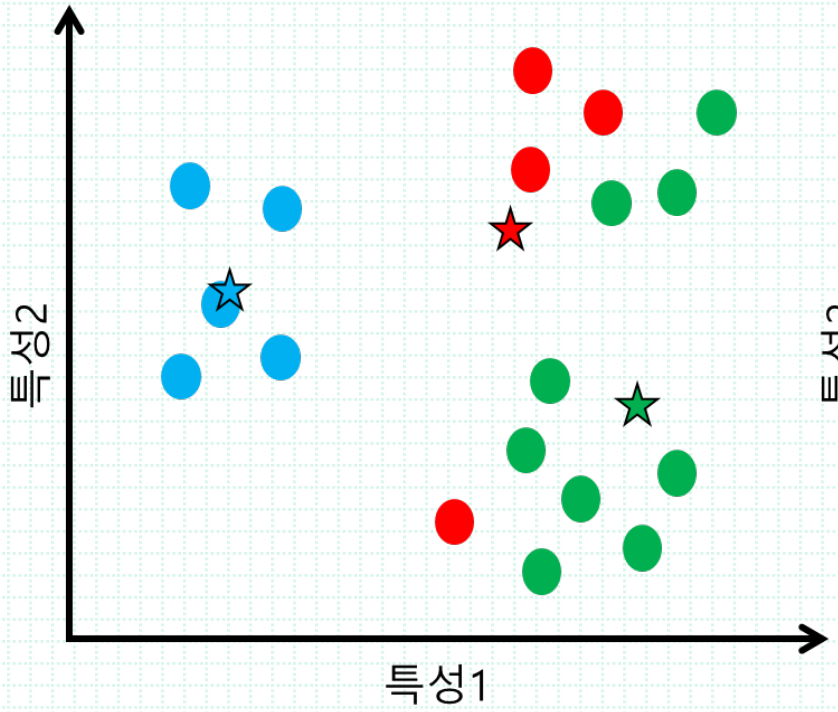



4-1단계: 가까운 centroid를 기준으로 클러스터 할당

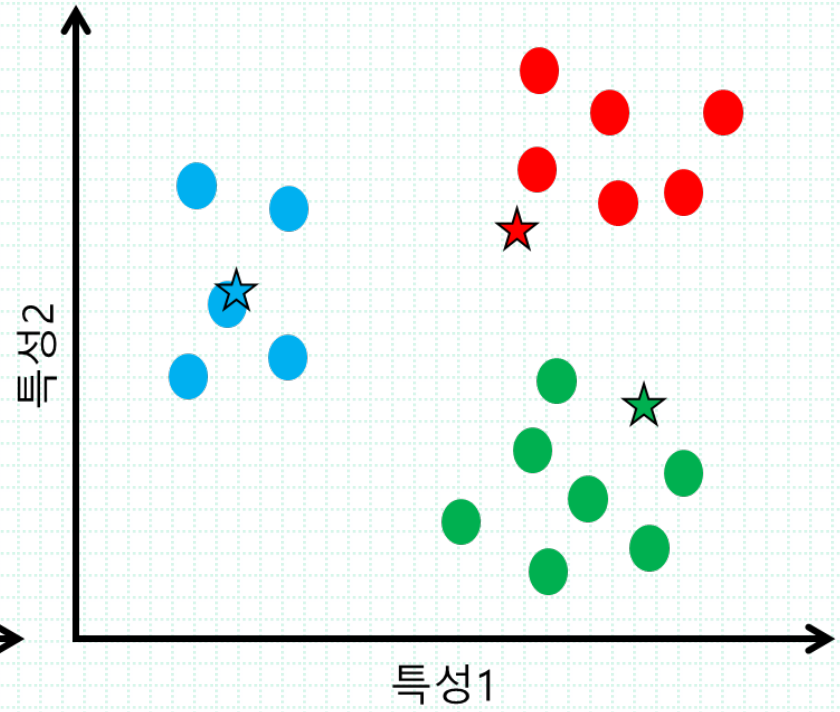


# 군집화

4-2단계: 각 클러스터 centroid 갱신

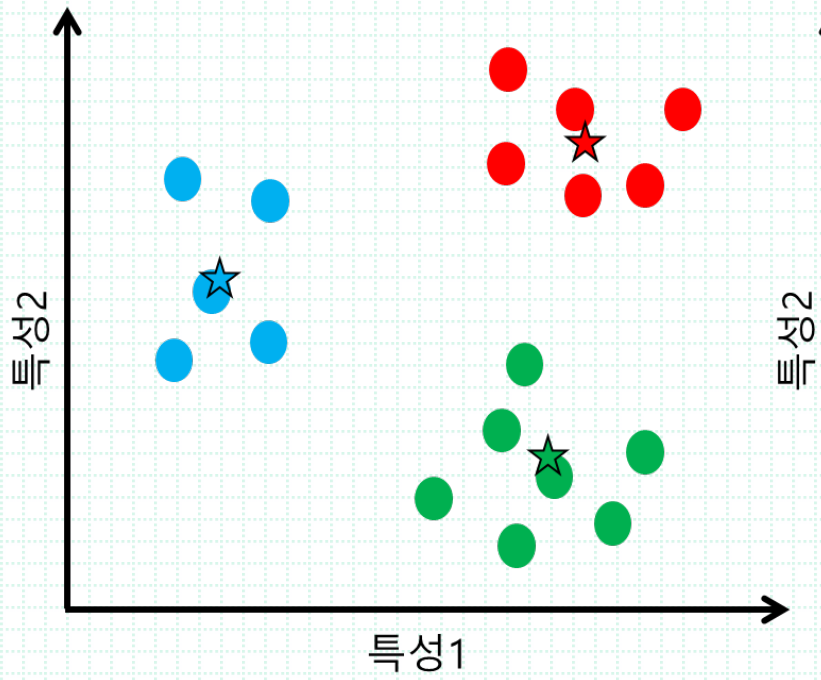



4-3단계: 가까운 centroid를 기준으로 클러스터 할당

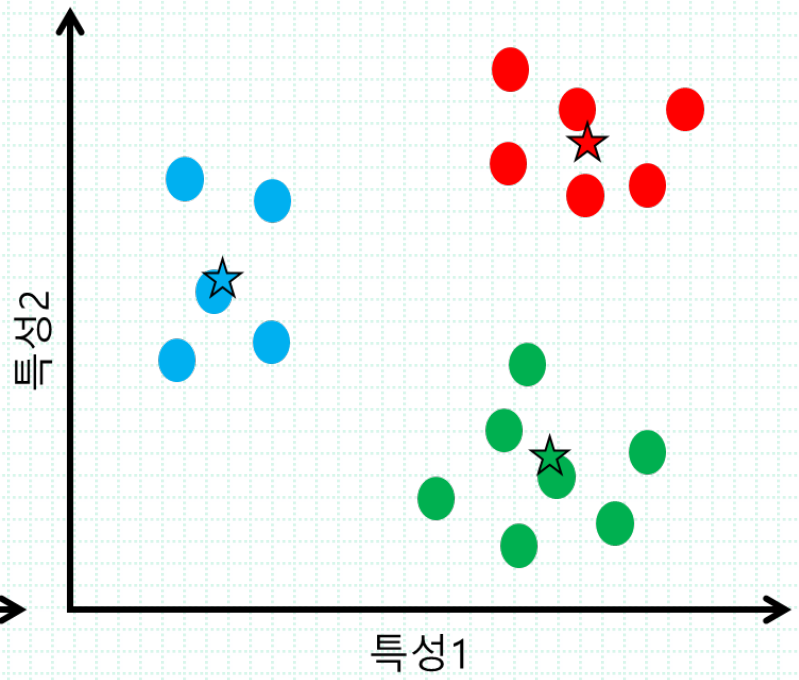


# 군집화

4-4단계: 각 클러스터 centroid 갱신




5단계: 클러스터 할당에 변동이 없으므로 알고리즘 종료



# 군집화

| 장점 – 알고리즘이 쉽고 간결, 일반적으로 많이 사용되는 알고리즘

| 단점 – 거리 기반 알고리즘으로 속성 개수가 많을 경우 군집화 정확도가 떨어진다.(차원의 저주)
→PCA를 해야 할 수도 있다

– 반복 횟수가 너무 많으면 수행시간이 느려짐

- 이상치 데이터에 취약해서 centroid가 이상해질 수 있다.

- 반복 시마다 K – Mean이 이상하게 설정될 수 있다.

# 군집화

# 군집화

## 간단 하이퍼 파라미터

- n_clusters : 군집화 할 개수, 여러 개를 시도해보며 가장 적절한 군집 중심점(elbow point) 개수를 찾아야 한다

- init : 초기에 군집 중심점의 좌표 설정할 방식

- max_iter : 최대 반복 횟수이며, 이 횟수 이전에 데이터의 중심점 이동이 없으면 종료

# 군집 평가

- 대부분의 군집화 데이터 세트는 비교할 만한 타깃 레이블을 가지고 있지 않다

-> 비지도 학습이니까 당연히

- 그럼 어떻게 평가를 해?

# 실루엣 분석

- 실루엣 분석은 군집화 평가 방법으로 각 군집 간의 거리가 얼마나 효율적으로 분리되어 있는지를 나타낸다.

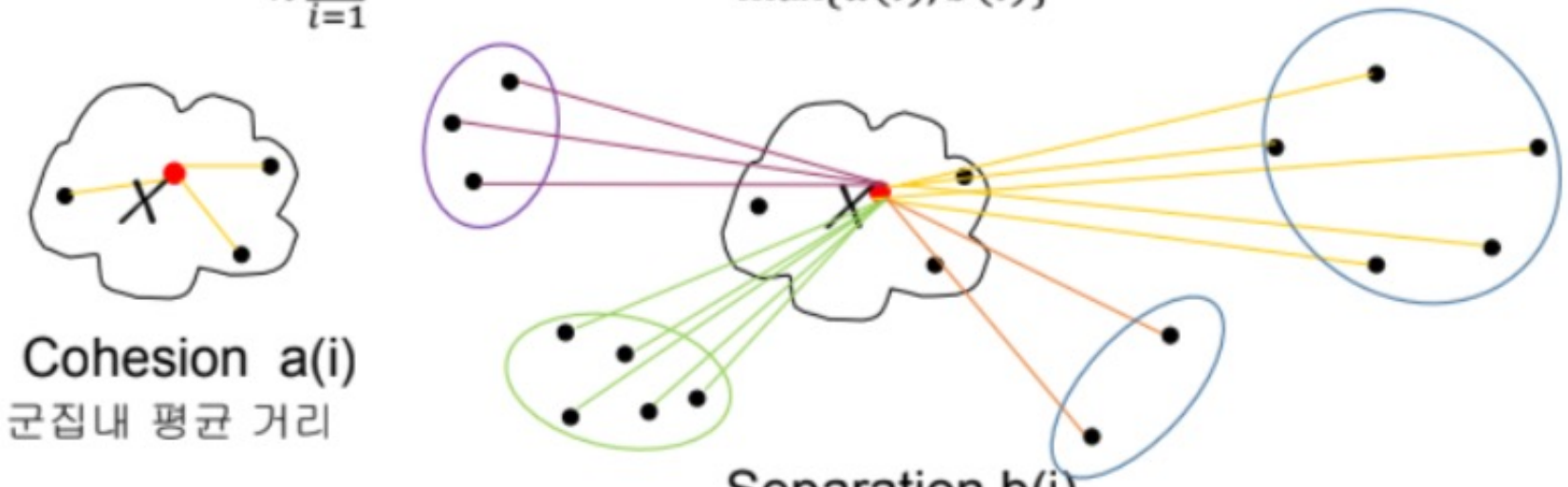
- 다른 군집과의 거리는 멀고 동일 군집끼리의 거리는 가까움

- 실루엣 분석은 실루엣 계수(silhouette coefficient)를 기반으로 한다.

# 실루엣 분석

## 실루엣 계수(silhouette coefficient)

$$SC = \frac{1}{n}\sum_{i=1}^{N} s(i), \qquad s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



# 실루엣 분석

- 실루엣 계수(silhouette coefficient)는 -1에서 1사이의 값을 가진다.
- 1로 가까워질 수록 근처의 군집과 더 멀리 떨어져 있다는 뜻이다.
- 0에 가까울 수록 근처의 군집과 가까워진다는 의미
- -값은 아예 다른 군집에 데이터 포인트가 할당 되었음을 의미한다

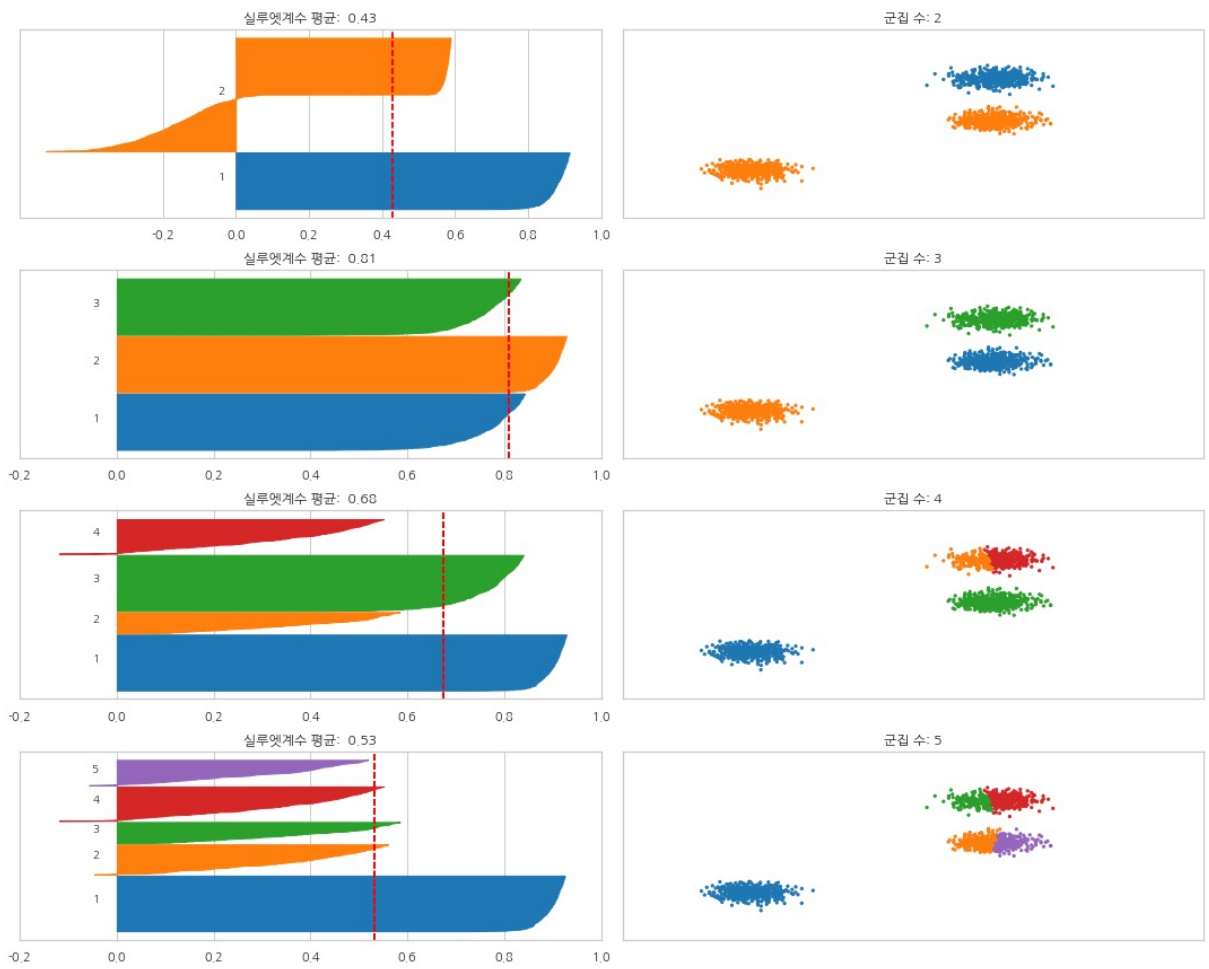
# 실루엣 분석

- 그러면 실루엣 계수가 높을 수록 군집화가 잘 된 것인가?

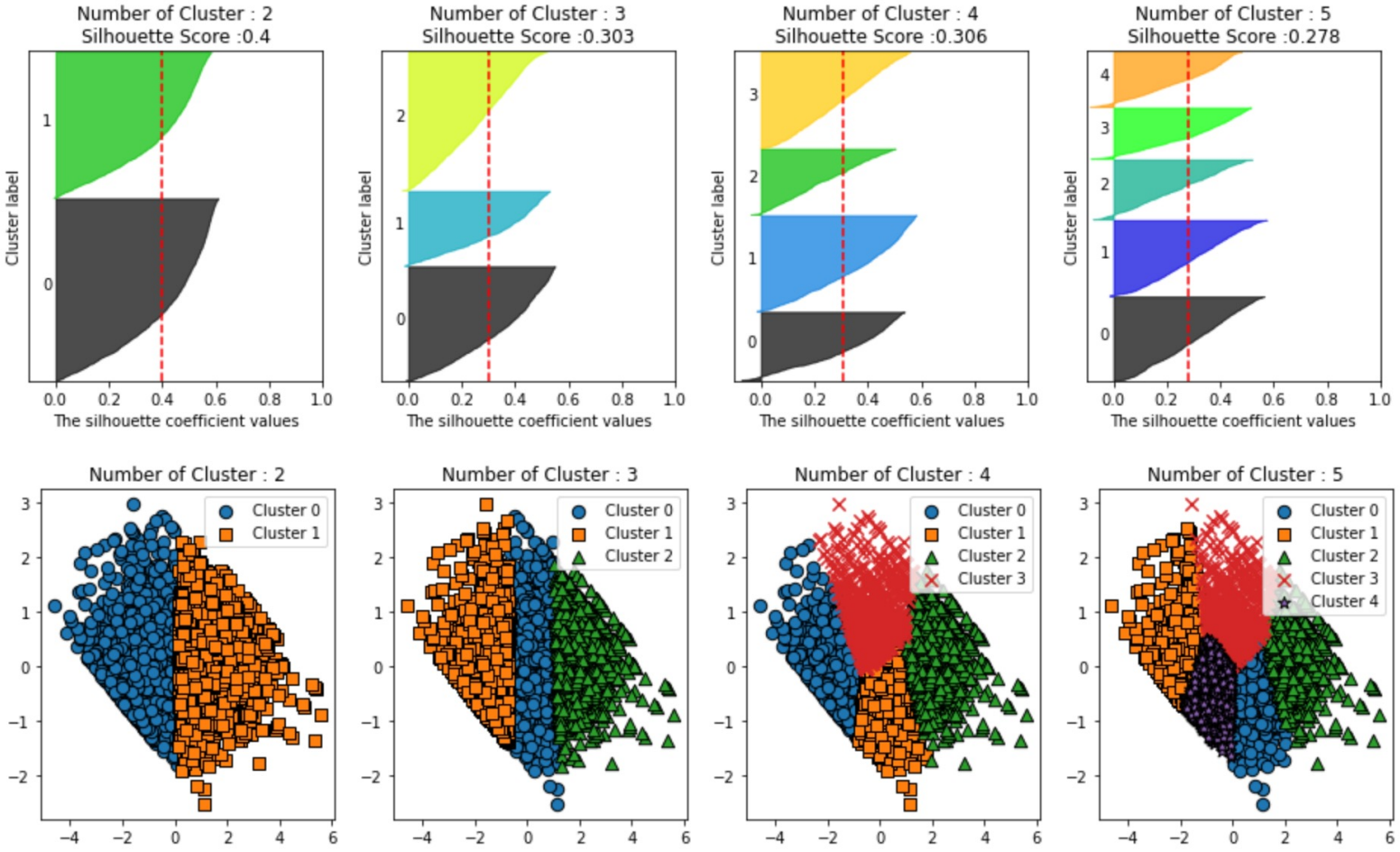
- 꼭 그런 것은 아니다...

- 특정 군집 내의 실루엣 계수가 너무 높고 다른 군집은 내부 데이터끼리의 거리가 너무 떨어져 있어서 평균적으로 높은 값을 가질 수 있다.

# 실루엣 분석


실루엣계수 평균: 0.43
군집 수: 2
실루엣계수 평균: 0.31
군집 수: 3
실루엣계수 평균: 0.68
군집 수: 4
실루엣계수 평균: 0.53
군집 수: 5
-0.2
0.0
0.2
0.4
0.6
0.8
1.0

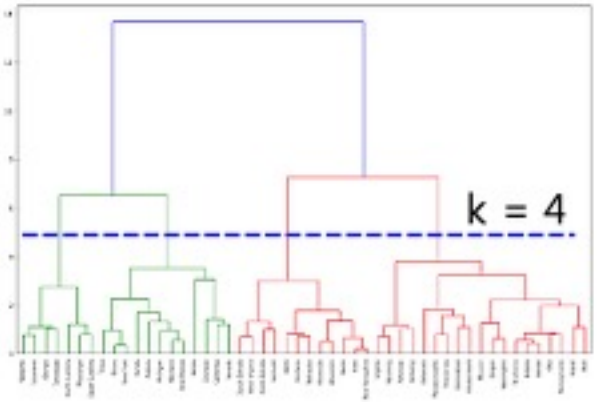
# 실루엣 분석


Number of Cluster : 2
Silhouette Score :0.4
Number of Cluster : 3
Silhouette Score :0.303
Number of Cluster : 4
Silhouette Score :0.306
Number of Cluster : 5
Silhouette Score :0.278
Cluster label
The silhouette coefficient values
Number of Cluster : 2
Number of Cluster : 3
Number of Cluster : 4
Number of Cluster : 5
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4

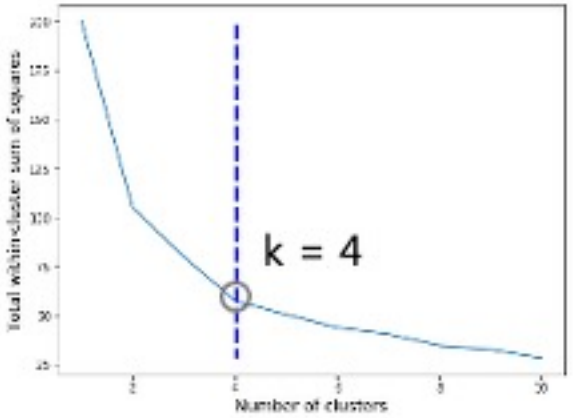
# 실루엣 분석

## 군집의 개수 결정 방법 (Determining the number of cluster k)

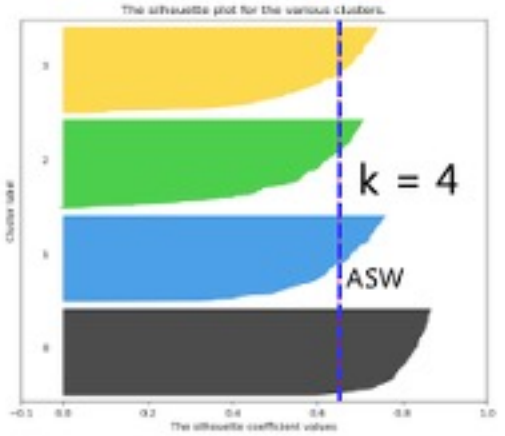
**1** **Hierarchical Clustering**




**2** **The Elbow Method**




**3** **The Silhouette Method**



# 평균 이동

| 평균이동(Mean Shift)은 K – 평균과 유사하게 중심을 군집의 중심으로 지속적으로 움직이면서 군집화를 수행

| 그러나 평균 이동 평균 거리 중심이 아닌 데이터가 모여 있는 밀도가 가장 높은 곳으로 이동

| 그렇다면 당연하게 확률 밀도 함수(probability density function)을 이용할 것

| 주변 데이터와의 거리 값을 KDE 함수 값으로 입력한 뒤 그 반환 값을 현재 위치에서 업데이트 하면서 이동하는 방식

# 평균 이동

| KDE..?

| KDE란 Kernel 함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 대표적인 방법

| 관측된 데이터 각각에 커널 함수를 적용한 값을 모두 더한 뒤 데이터 건수로 나누어 확률 밀도 함수를 추정

| 확률 밀도 함수는 확률 변수의 분포를 나타내는 함수, 정규 분포, 감마 분포, t – 분포 등이 있다.

# 평균 이동

| KDE

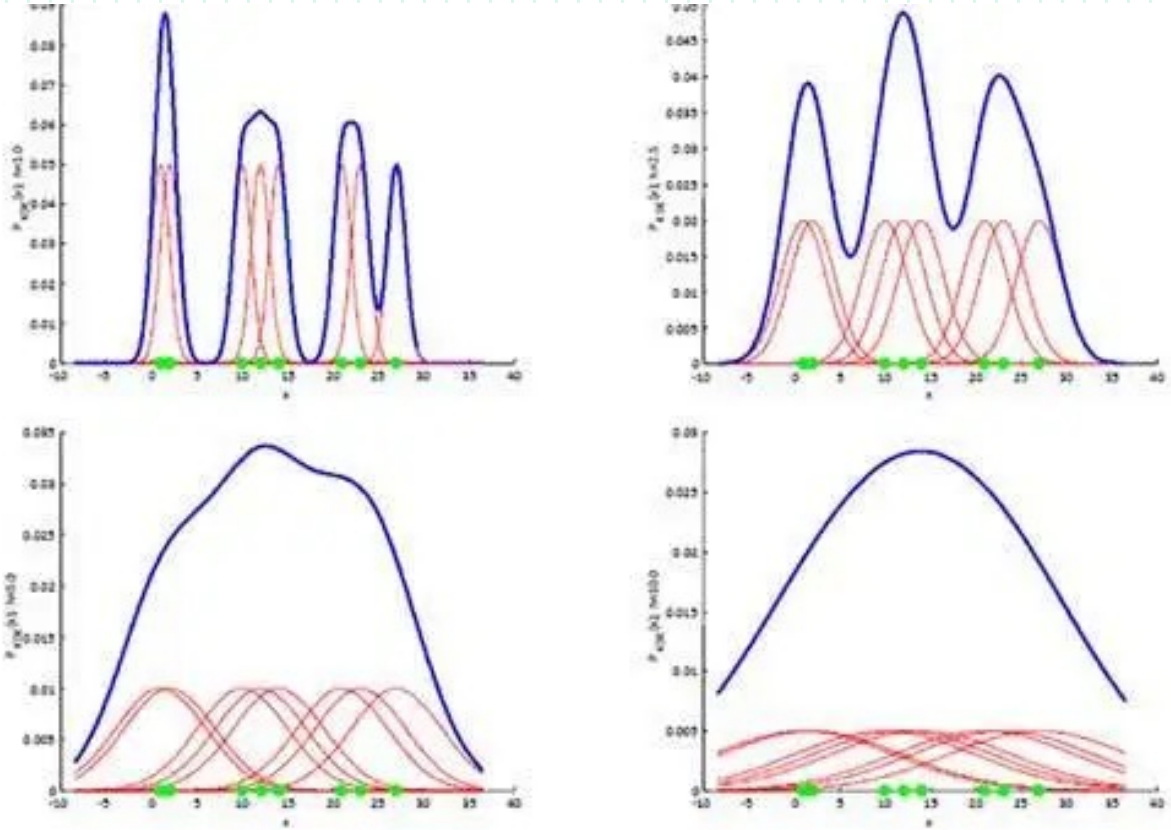
$$KDE = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h})$$

| K는 커널 함수, x는 확률 변숫값, xi는 관측 값, h는 대역폭

| 여기서 h의 대역폭에 따라 KDE의 형태가 변한다 그것 만이라도 기억하자

# 평균 이동

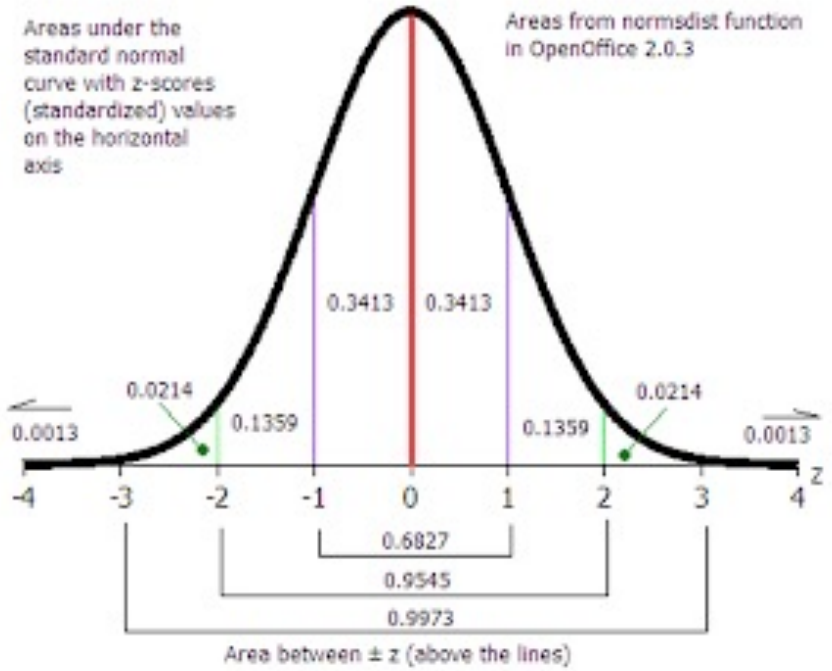
- h가 증가하면 평탄한 형태, h가 감소하면 뾰족한 형태
- 뾰족할 수록 과적합이 쉽다.
- 일반적으로 평균 이동 군집화는 대역폭이 클 수록 적은 군집 중심점
- Meanshift 클래스의 bandwith를 통해 h 값 조절 가능



# GMM

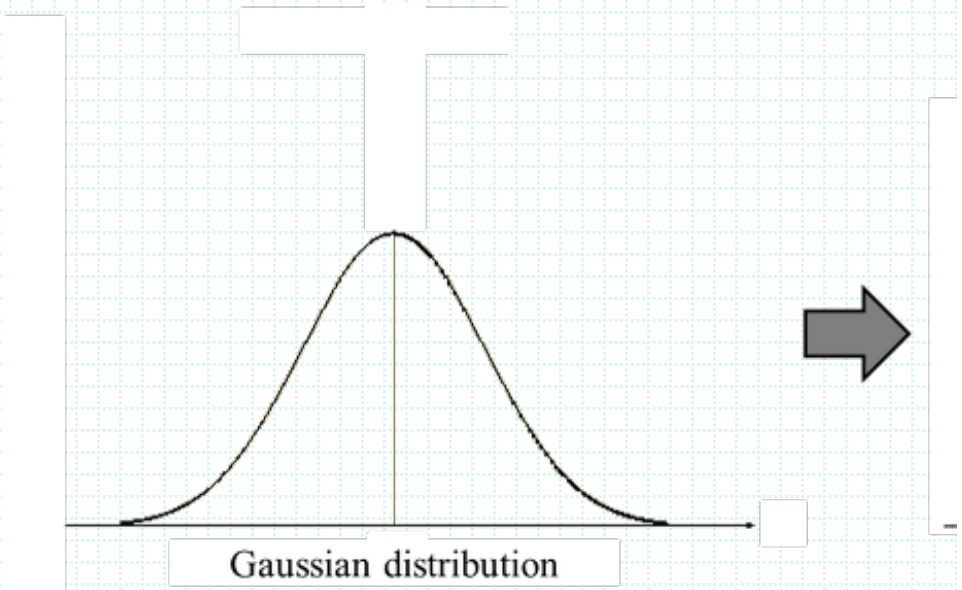
| GMM 군집화는 군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것이라는 가정하에 군집화를 수행

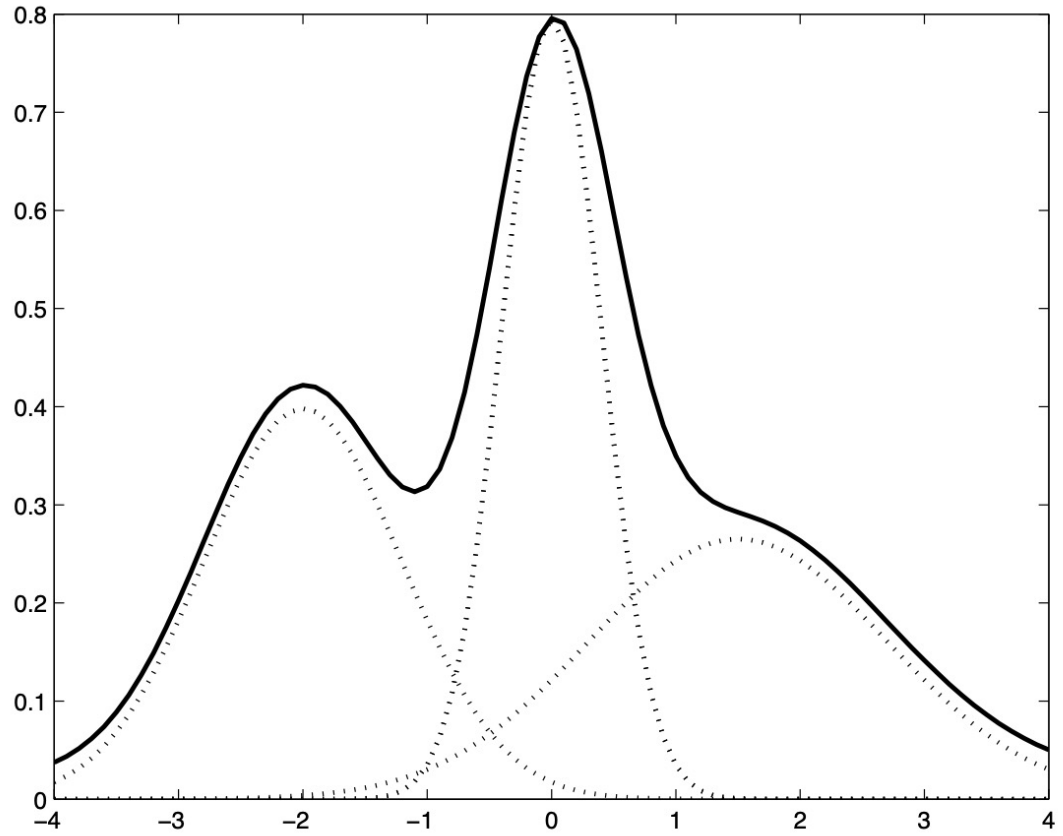
| 가우시안 분포 = 정규 분포



# GMM

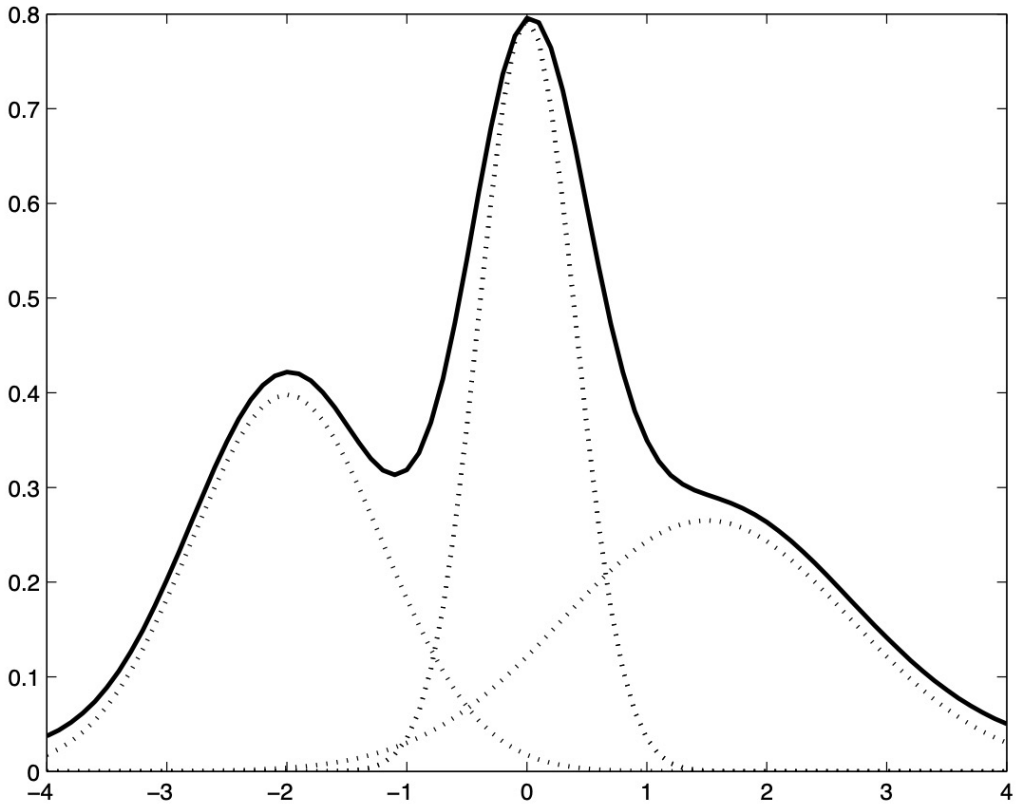
| GMM은 데이터를 여러 개의 가우시안 분포가 섞인 것으로 간주



# GMM

- GMM은 데이터를 여러 개의 가우시안 분포가 섞인 것으로 간주



# GMM

| 모수 추정이란 대표적으로 2 가지를 추정

- 개별 정규 분포의 평균과 분삭
- 각 데이터가 어떤 정규분포에 해당되는지 확률

| GMM은 모수 추정을 위해 EM(Expectation and Maximization) 방법을 적용

수학 공부 열심히 하자!

| 사이킷런에서 GausianMixture 클래스 지원

# GMM

| 모수 추정이란 대표적으로 2 가지를 추정

- 개별 정규 분포의 평균과 분삭
- 각 데이터가 어떤 정규분포에 해당되는지 확률

| GMM은 모수 추정을 위해 EM(Expectation and Maximization) 방법을 적용

| 사이킷런에서 GausianMixture 클래스 지원

# GMM

| Expectation = 개별 데이터 각각에 대해 특정 정규분포에 소속될 확률과 가장 높은 확률을 가진 정규 분포에 소속

| Maximization = 데이터들이 특정 정규분포로 소속되면 다시 해당 정규분포의 평균과 분산을 구함

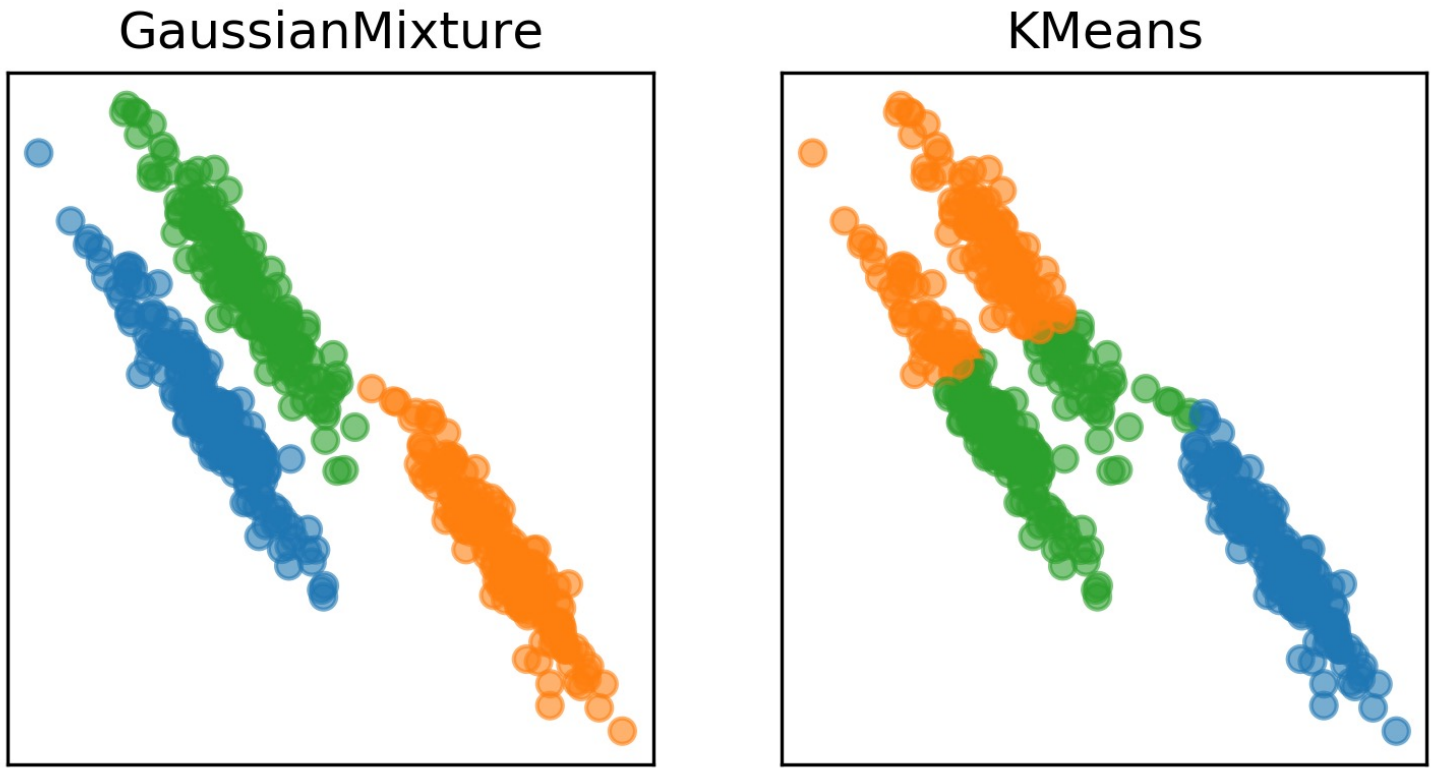
| 이것을 계속 반복해서 평균과 분산이 더이상 변경되지 않으면 확정

# GMM

- 근데 굳이 K – Means가 제일 많이 쓰이는데 이것을 써야 하나..?
- K – Means는 거리를 기반으로 하기 때문에 원 형태의 군집을 잘 분리
- 만약 원이 아니면..?

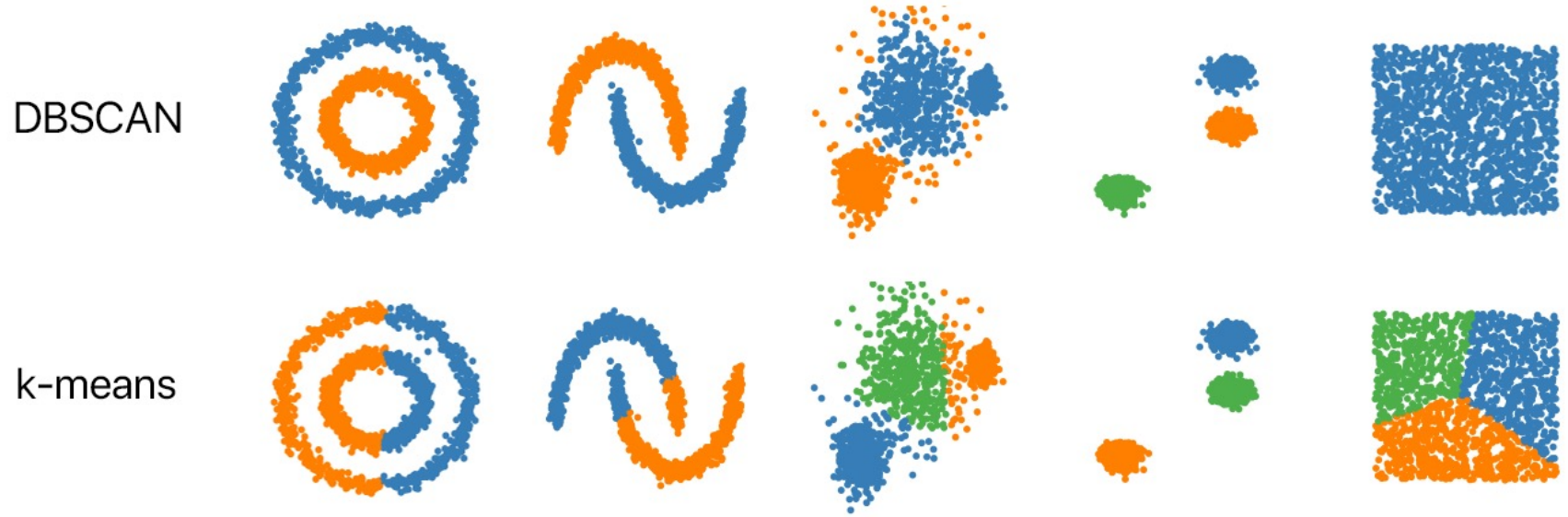
# GMM

## GMM vs K - Means



# DBSCAN

- DBSCAN은 밀도 군집화의 대표적인 알고리즘
- 기하학적으로 복잡한 데이터 세트에도 효과적인 군집화 가능

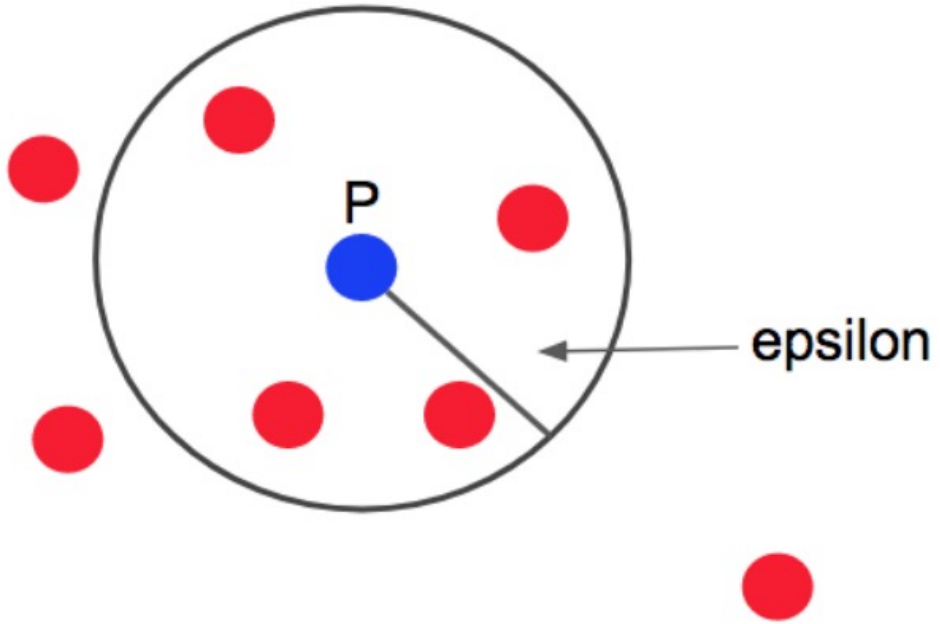


# DBSCAN

- 핵심 포인트 – 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우
- 이웃 포인트 – 주변 영역 내에 위치한 타 데이터
- 경계 포인트 – 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만 핵심 포인트를 이웃 포인트로 가지고 있는 데이터
- 잡음 포인트 – 나머지

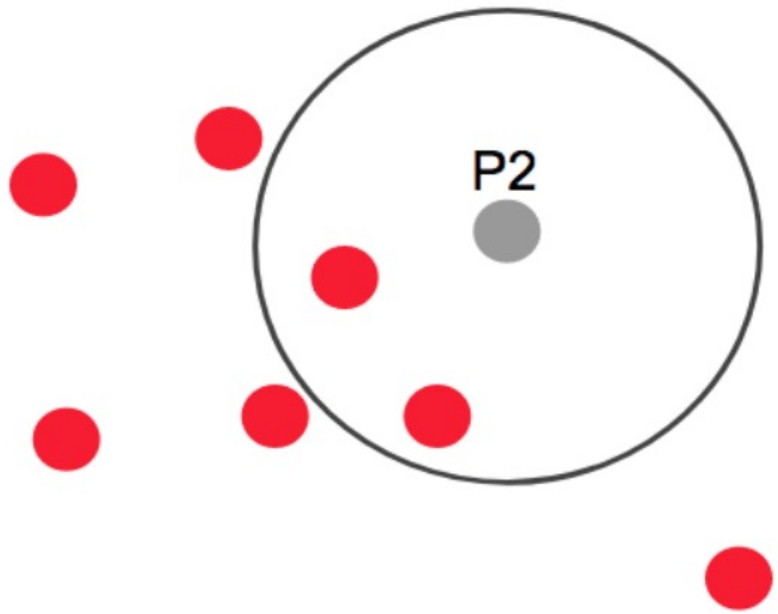
# DBSCAN

반경 내에 4개 이상은 핵심 포인트라고 하자



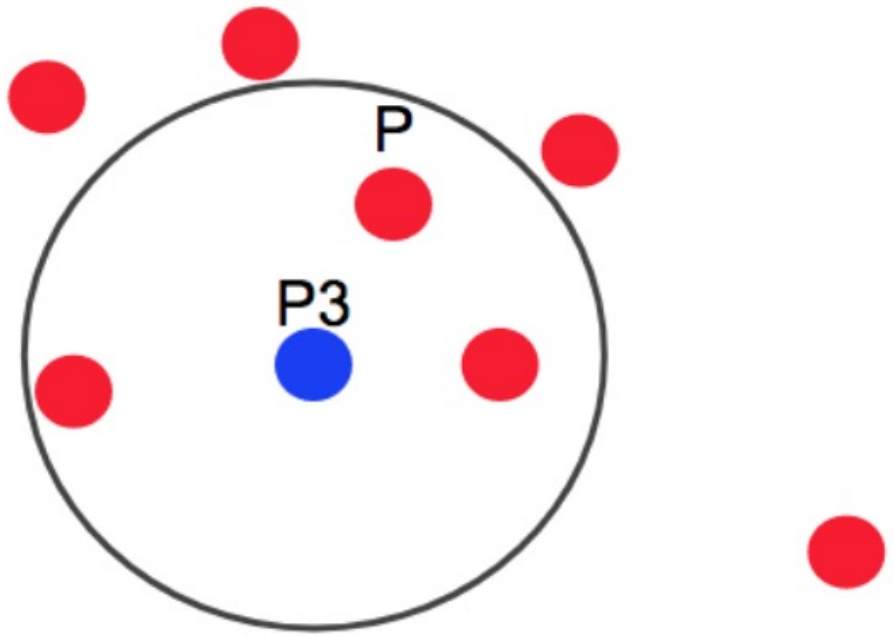
# DBSCAN

- 얘는 이웃 포인트



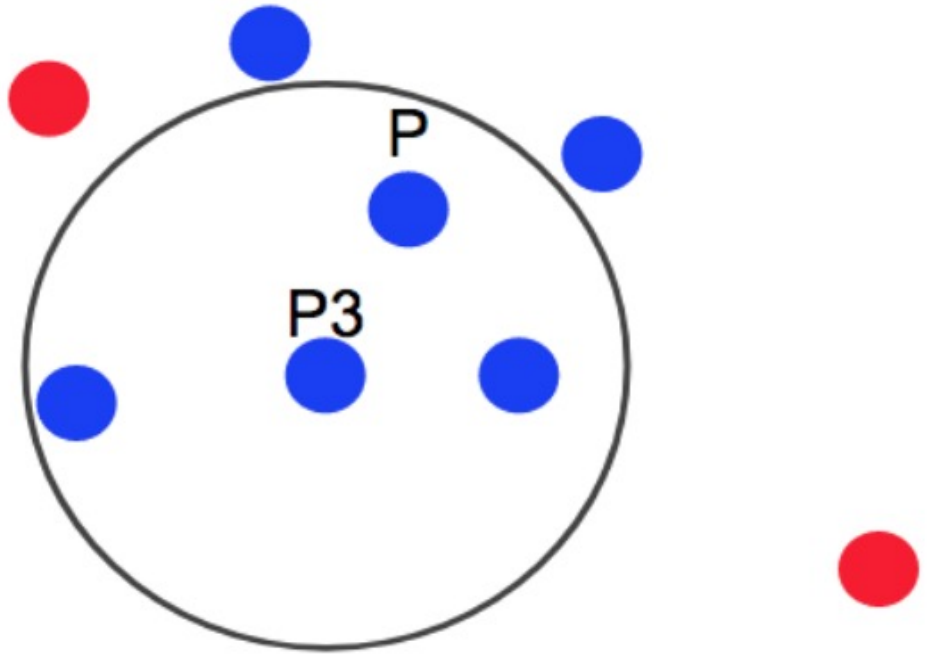
# DBSCAN

| 근데 P도 반경에 4개가 속하네?



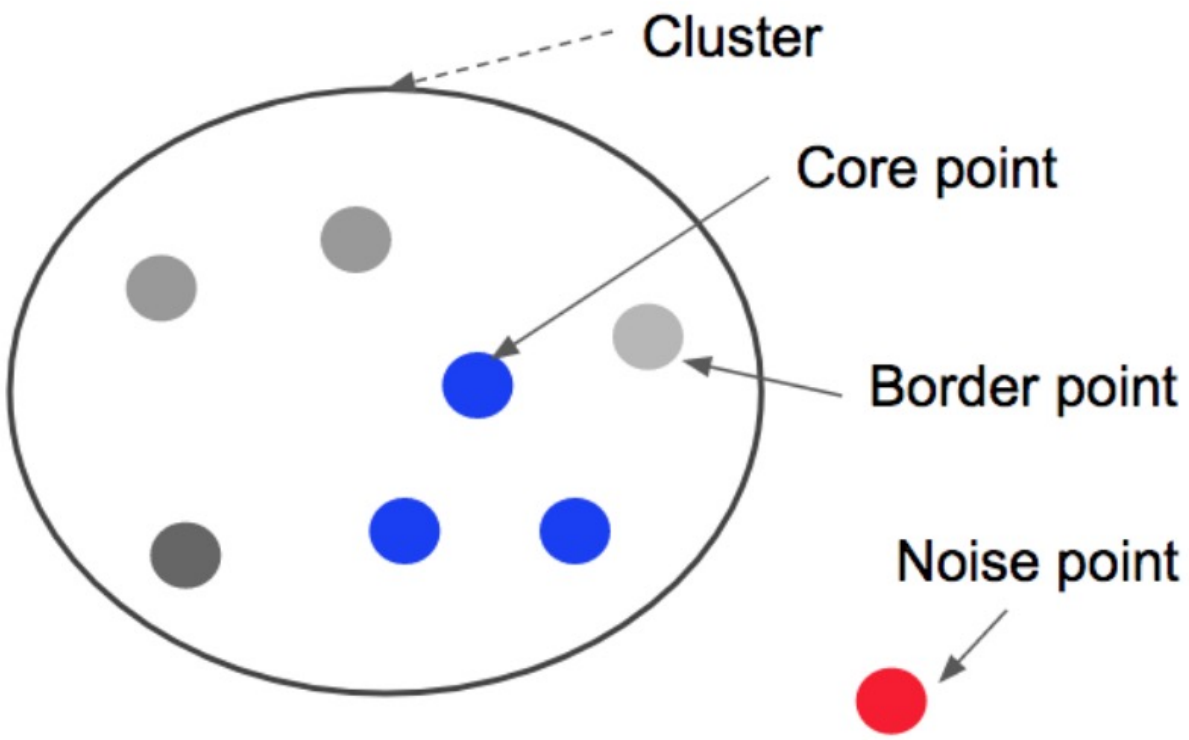
# DBSCAN

| 그럼 합치자! 그리고 이 과정을 계속 반복



# DBSCAN

## 결과



# DBSCAN

| DBSCAN클래스로 알고리즘을 지원한다

| eps : 입실론 주변 반경

| min_samples : 핵심 포인트가 되기 위해 입실론 주변 영역 내에 포함되어야 할 데이터의 최소 개수

# 마무리

수고 하셨습니당