천재교육

# 머신러닝 (Machine Learning) 이해 및 실습

K - 디지털 아카데미

# 머신러닝 교육과정 강의 일정

**1일차 > 머신러닝**

- > 머신러닝의 개요
- > [실습] 데이터 분석의 시작 Numpy, Pandas 활용하기

**2일차 > Classification (분류)**

- > 분류를 평가하는 지표 알아보기
- > 분류 알고리즘 (결정트리, 앙상블, 랜덤포레스트 등) 익히기
- > [실습] 분류를 통한 밀크T 만료및탈퇴회원 예측(이탈 회원 예측)

**3일차 > Regression (회귀)**

- > 회귀와 경사 하강법
- > 로지스틱 회귀와 소프트맥스 회귀
- > [실습] 로지스틱 회귀를 통한 문항별 정오답 예측

**4일차 > 차원 축소와 Clustering(군집화)**

- > PCA, LDA
- > K-means, DBSCAN 등 다양한 클러스터링 기법 알아보기
- > [실습] 밀크T중학 회원수준 군집화(GMM)

**5일차 > 추천시스템과 최종 프로젝트**

- > 추천 시스템
- > 최종 프로젝트

# 차원축소

01 차원의 저주

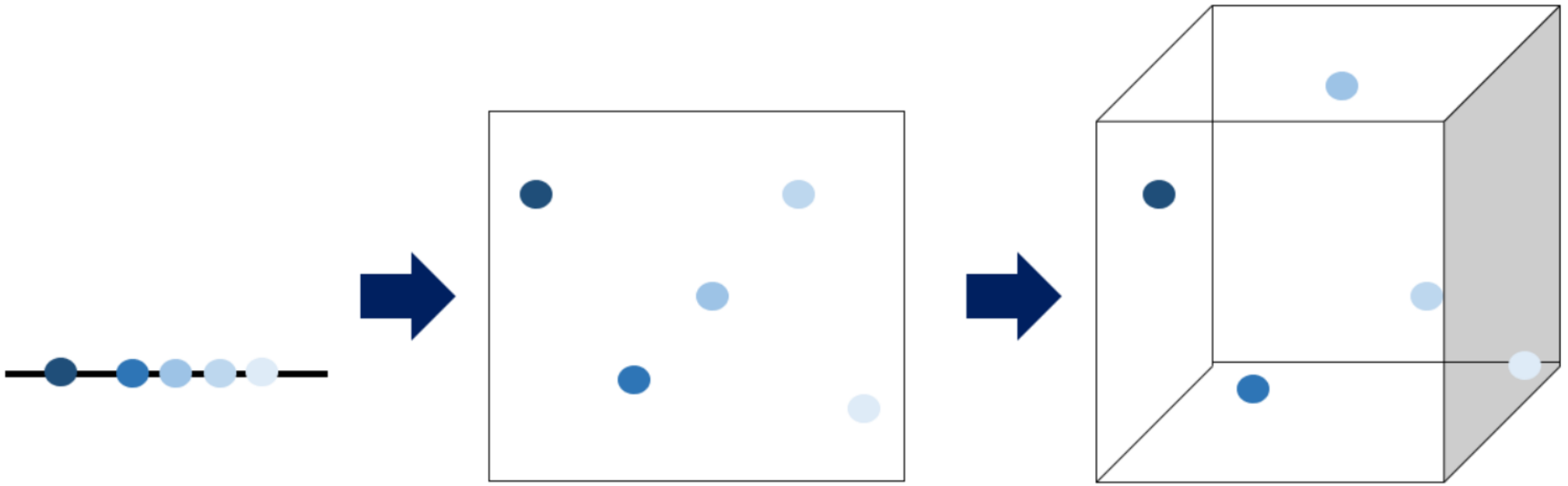
02 PCA

03 LDA

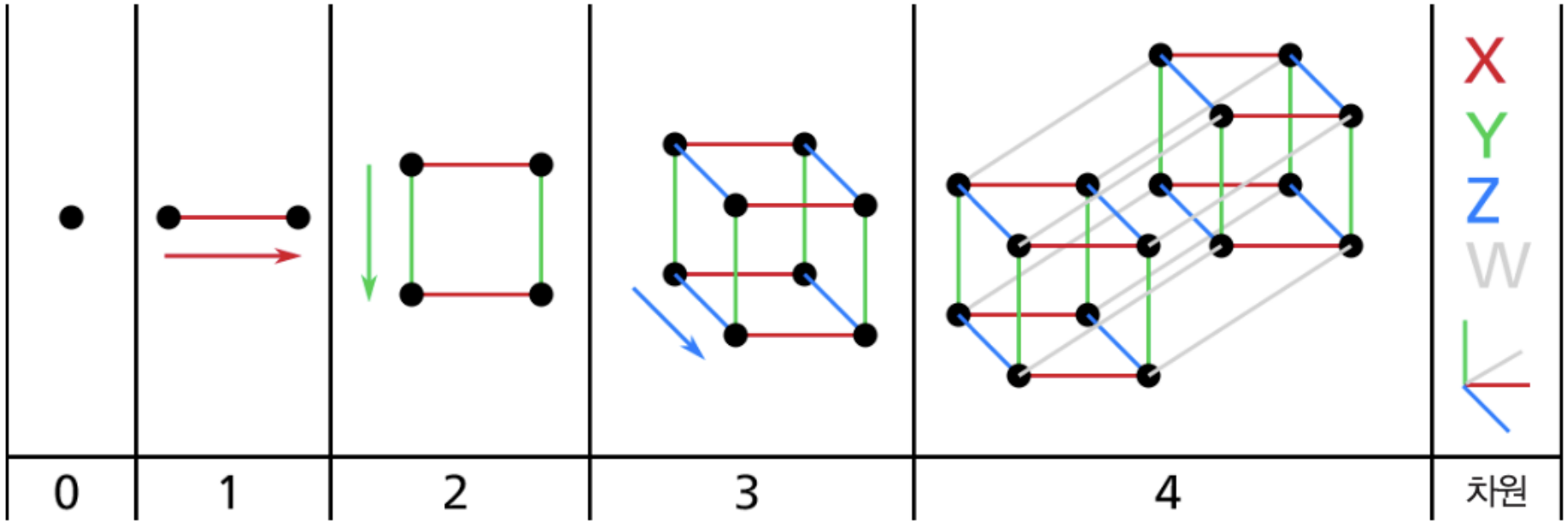
머신
러닝.

# 차원의 저주(The curse of dimensionality)

차원이 증가할 수록 데이터 간 빈 공간이 생기게 됨으로써 생기는 문제들을 차원의 저주라 칭한다.



# 차원의 저주(The curse of dimensionality)




- 차원이 커질수록 임의의 두 지점 사이의 평균 거리가 매우 멀어진다.
- 과대적합(Overfitting)의 위험도가 커진다.

# 차원 축소

**차원 축소**란,

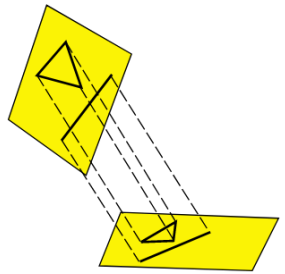
**차원의 저주 문제를 해결하기 위해 특성(Feature) 수를 줄여서**

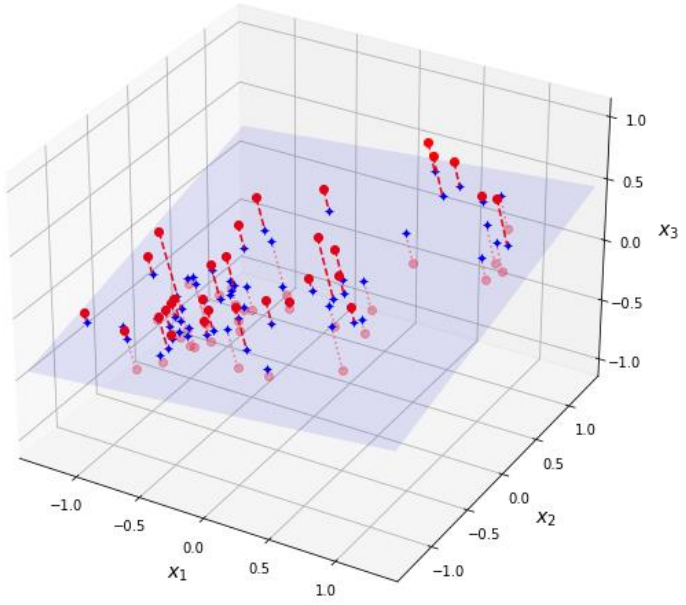
**학습 불가능한 문제를 학습 가능한 문제로 만드는 기법.**

**즉 정보의 손실이 크지 않은 방향으로 고차원의 데이터를 저차원의 데이터셋으로 변환시키는 것이다.**

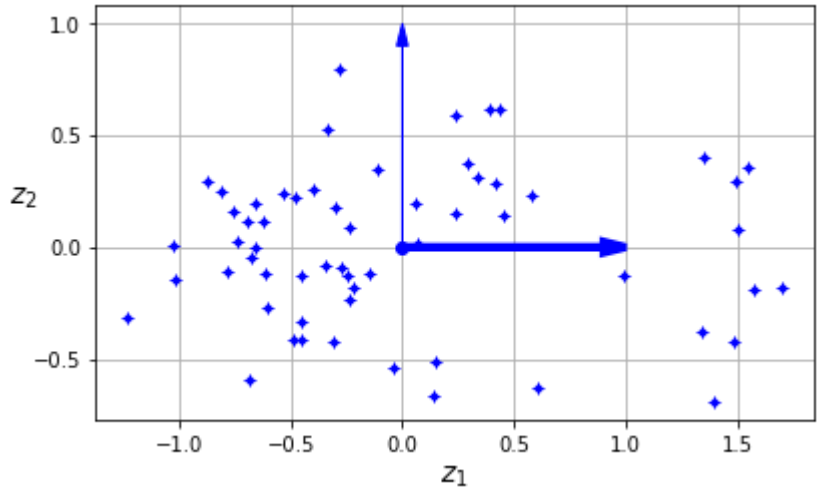
# 사영기법

n차원의 데이터셋을 차원이 낮은 d차원 데이터 셋으로 사영(Projection)하는 기법






* 사영을 위해 필요한 적절한 축을 찾는 것이 사영기법의 핵심!

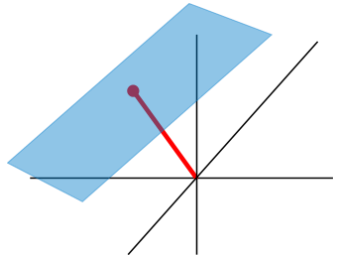


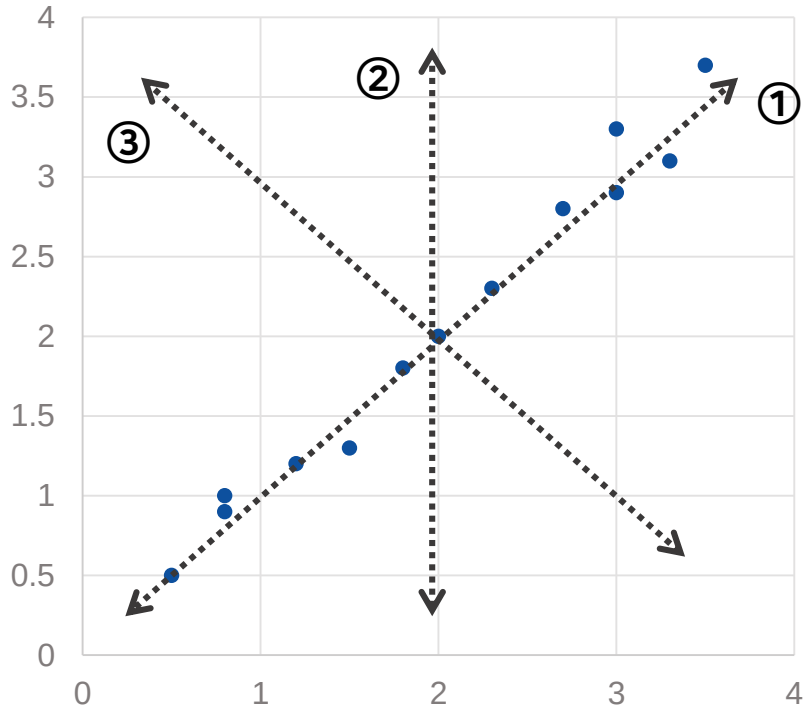
# PCA

## 주성분 분석(Principal Component Analysis)

- 학습 데이터 셋을 특정 초평면(Hyperplane)에 사영하는 기법

  ↳ 3차원 이상의 고차원에 존재하는 저차원 공간



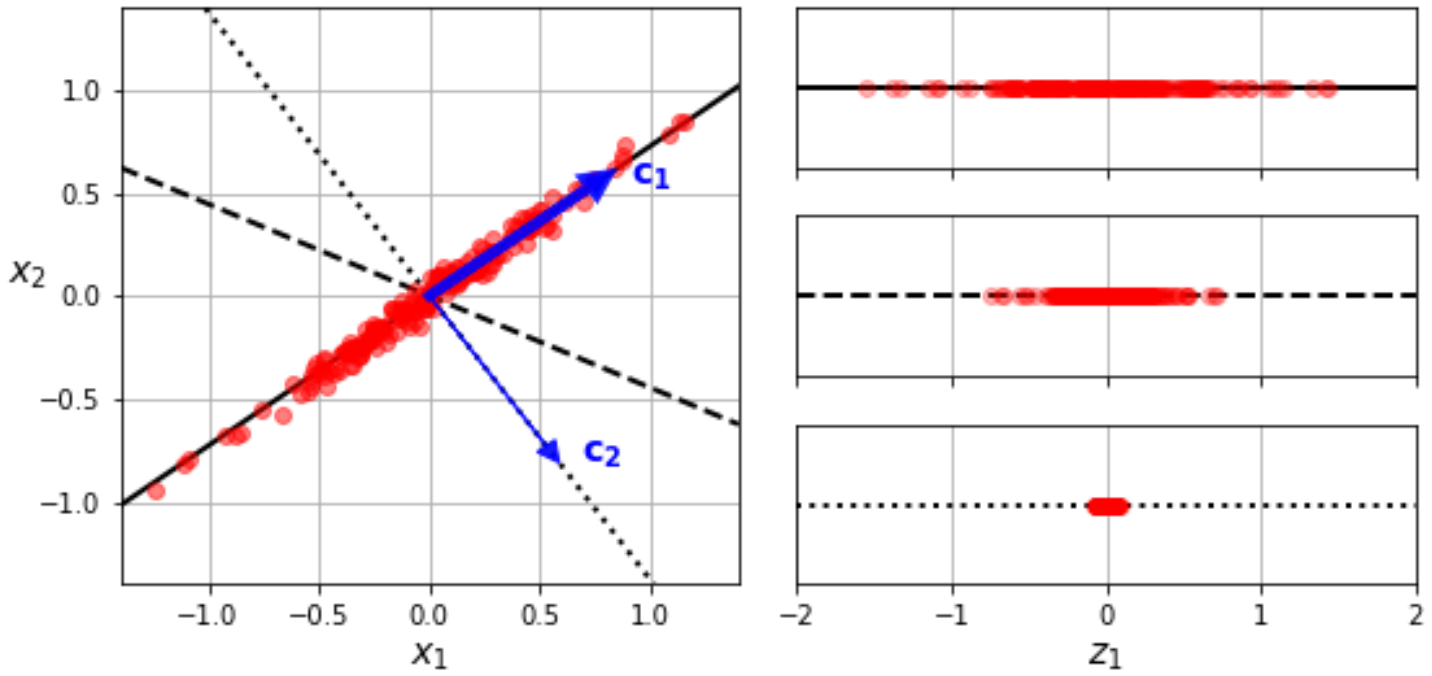



Q. 위 데이터를 차원을 줄여 표현하려면 어떤 축으로 사영해야 할까?

# PCA

## 주성분 분석(Principal Component Analysis)

- 학습 데이터 셋을 특정 초평면(Hyperplane)에 사영하는 기법



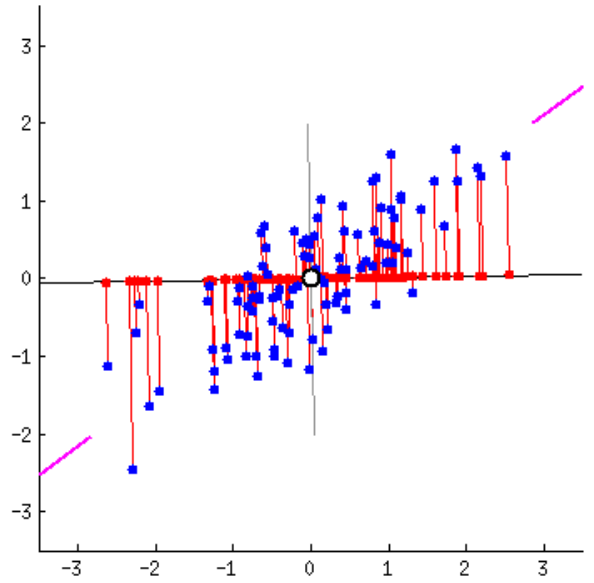

**분산 보존**

: 저차원으로 사영할 때 훈련셋의 분산이 최대한 유지되도록 축을 지정해야 한다.

# PCA

## 주성분 분석(Principal Component Analysis)

- 학습 데이터 셋을 특정 초평면(Hyperplane)에 사영하는 기법

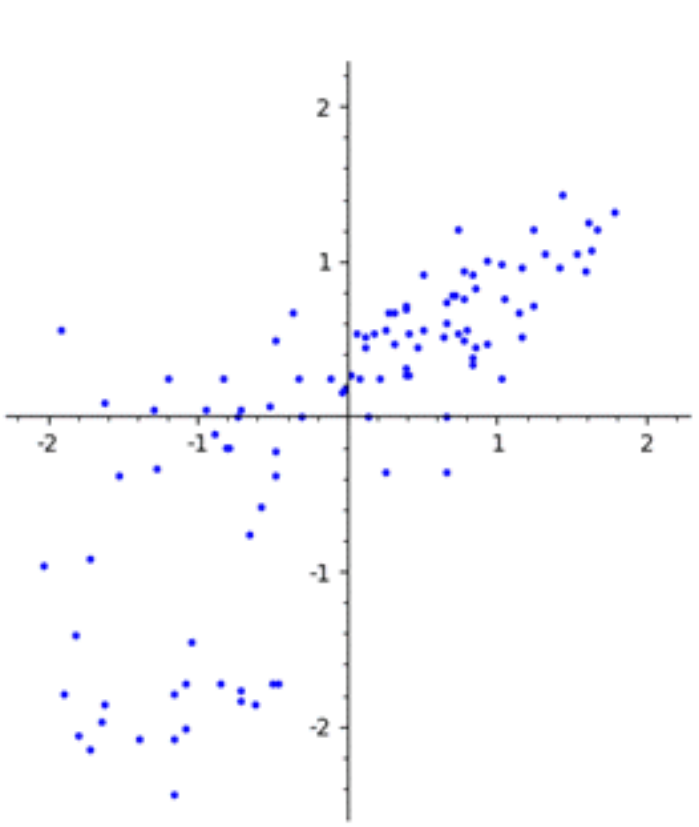


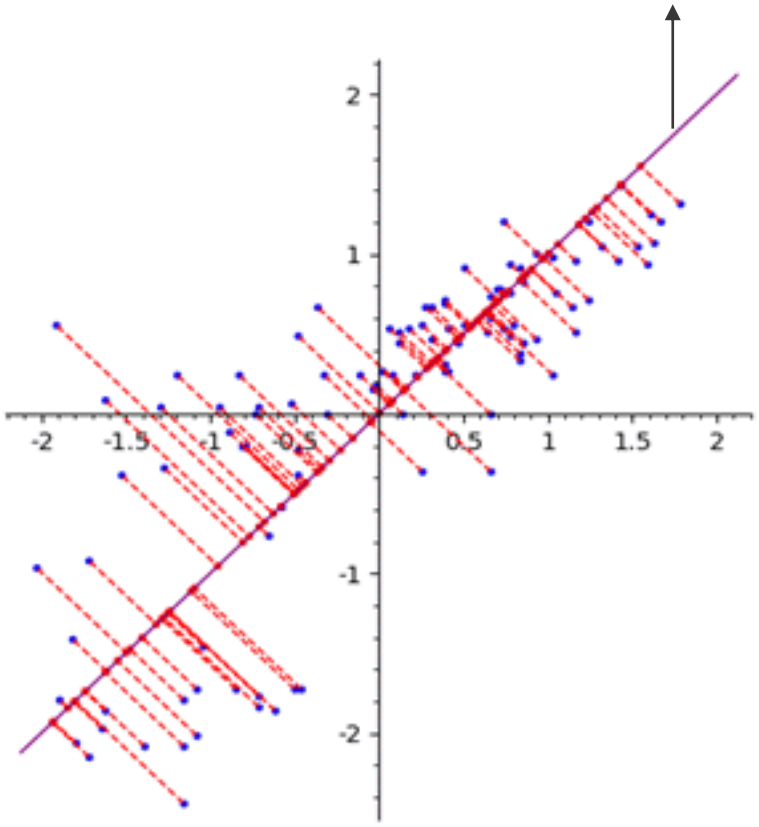

**분산 보존**

: 저차원으로 사영할 때 훈련셋의 분산이 최대한 유지되도록 축을 지정해야 한다.

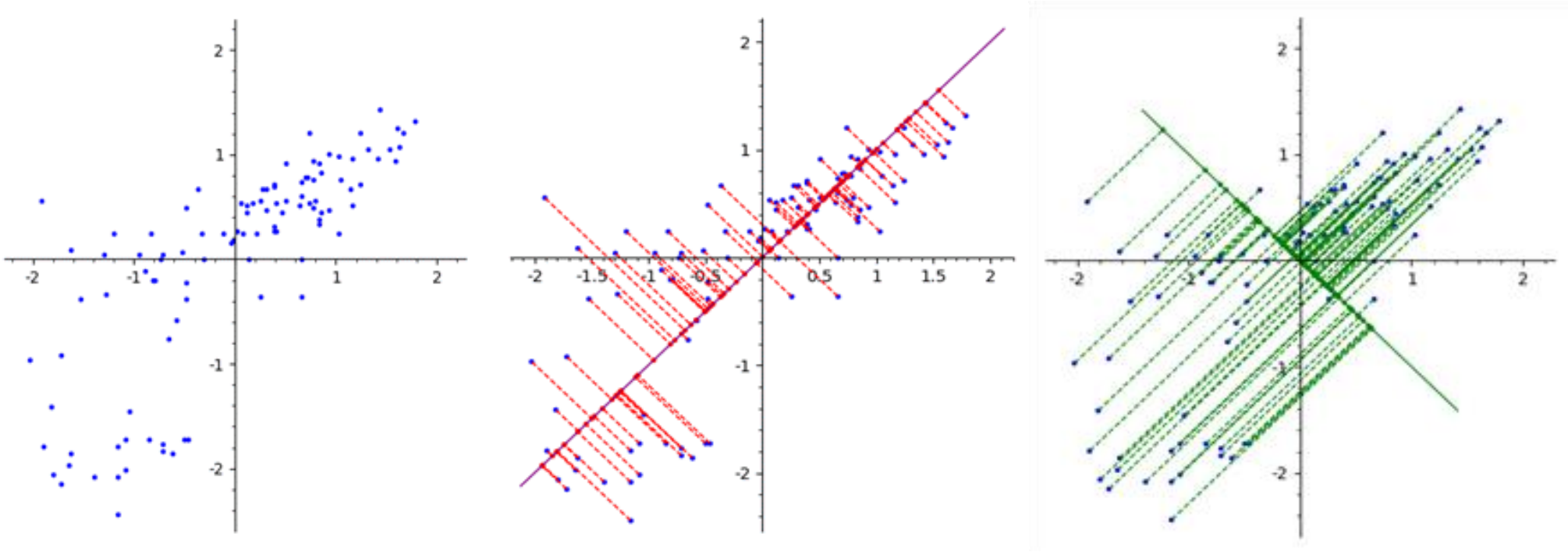
# PCA

주성분 : 전체 데이터들의 분산을 가장 잘 표현하는 성분
(Principal component, PC)



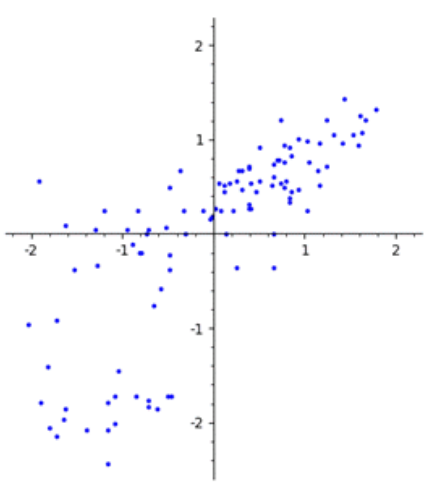



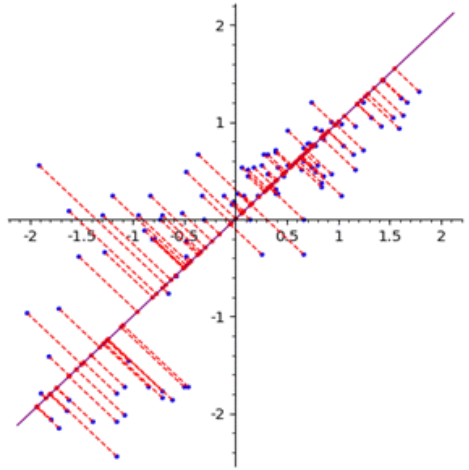
# PCA

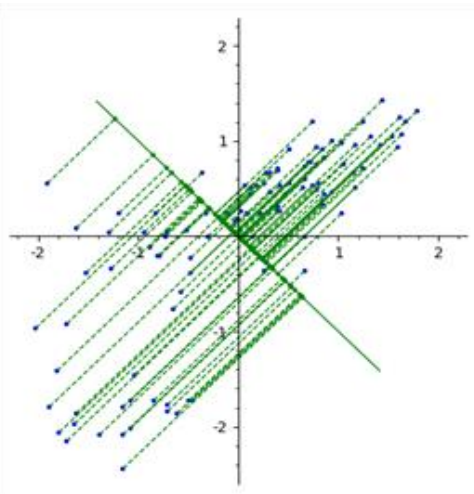


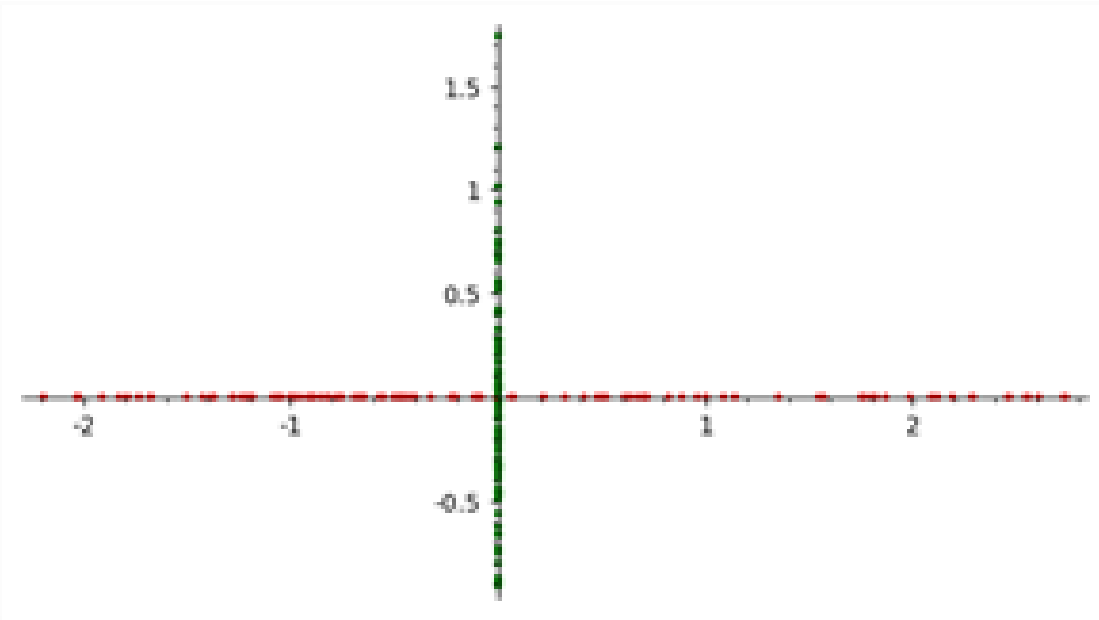

**주성분끼리는 서로 수직을 이루도록 한다.**

# PCA

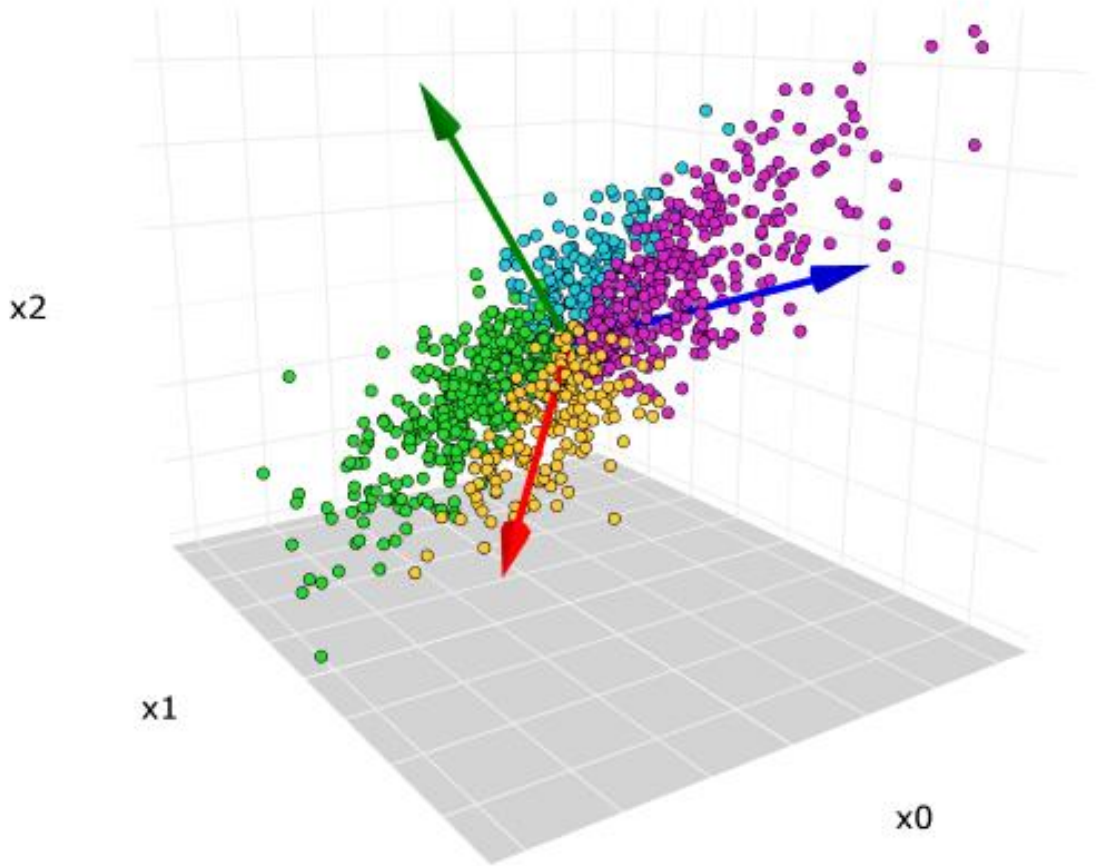


# PCA




**주성분 끼리는 관계가 없도록 하기 위해서 서로 수직을 이루는 축이어야 한다.**
**실제로 매번 분산이 큰 방향의 축을 찾으면 수직을 이루는 것이 당연하다.**

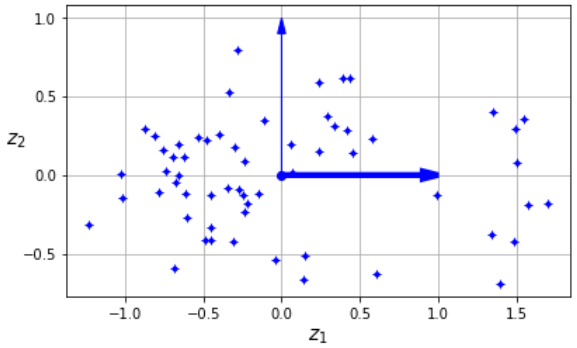
# Sklearn의 PCA 모델

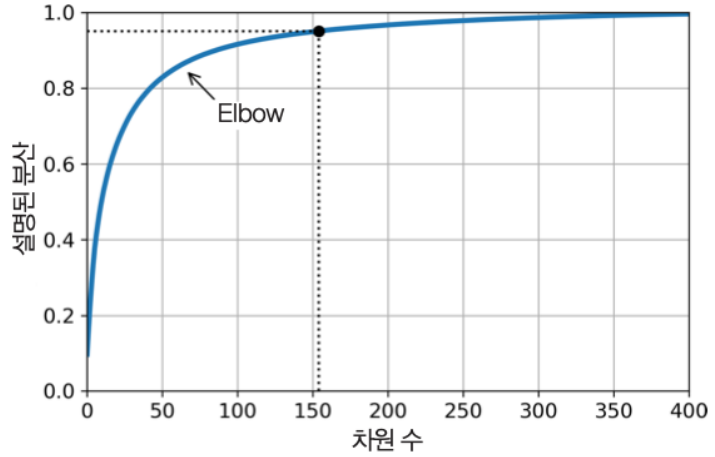
```
pca = PCA(n_components=2)
X2D = pca.fit_transform(X)
```

```
>>> pca.explained_variance_ratio_
array([0.7578477 , 0.15186921]) #output
```

: 각 주성분에 대한 원 데이터셋의 분산 비율

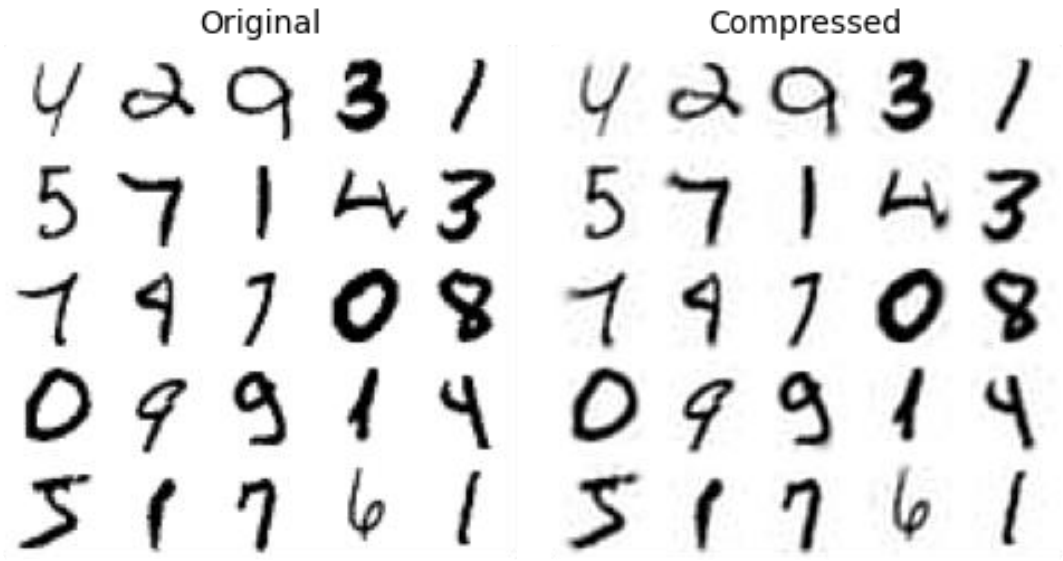
- $z_1$ 축: 75.8%
- $z_2$ 축: 15.2%




```
pca = PCA(n_components = 0.95)
X_reduced = pca.fit_transform(X_train)
```

# PCA의 활용

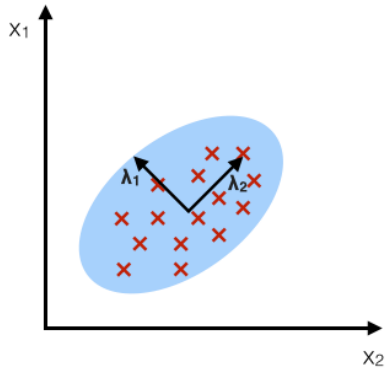



# LDA

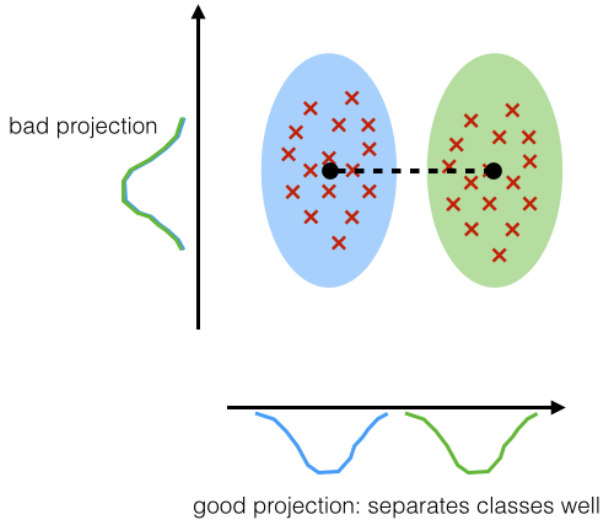
**LDA(선형 판별 분석, Linear Discriminant Analysis)**

- **PCA는 입력 데이터의 변동성이 가장 큰 축을 찾음 (분산분포)**
- **LDA는 입력 데이터의 결정값 클래스를 최대한 분리할 수 있는 축을 찾음**

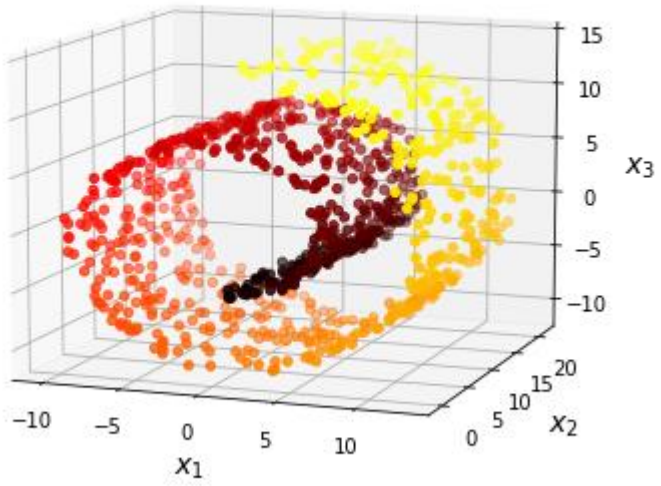
**PCA:**
component axes that
maximize the variance

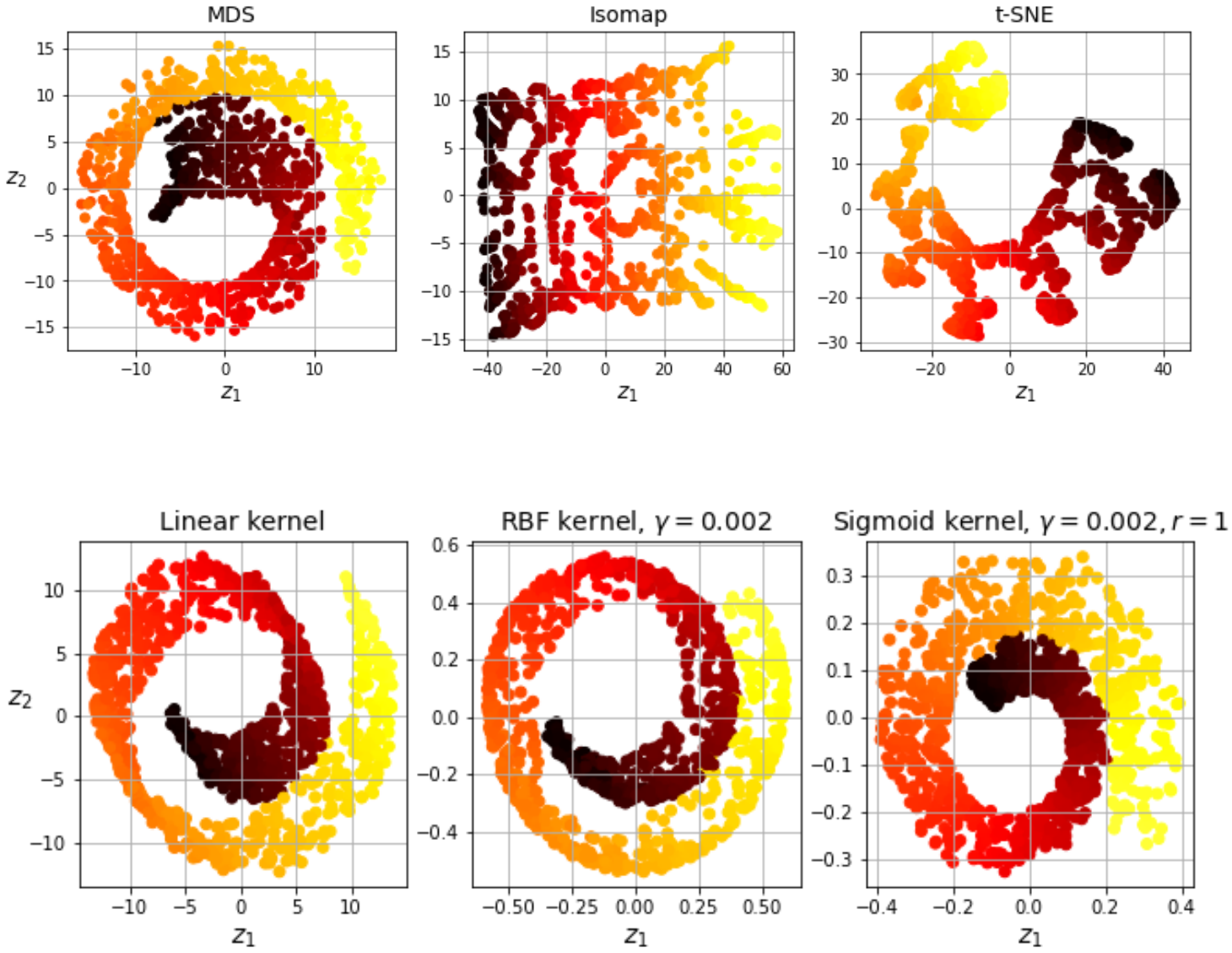



**LDA:**
maximizing the component
axes for class-separation



# 다양한 차원축소 모델


15
10
5
0
−5
−10
$x_3$
−10
−5
0
5
10
$x_1$
0
5
10
15
20
$x_2$


MDS
Isomap
t-SNE
Linear kernel
RBF kernel, $\gamma = 0.002$
Sigmoid kernel, $\gamma = 0.002, r = 1$
$z_1$
$z_2$

# [실습]

실습!

# 클러스터링(Clustering)

01 K-means

02 DBSCAN

03 GMM

머신
러닝.

# 비지도 학습

**비지도 학습**
**- 레이블이 없는 데이터를 학습하는 기법**

**군집화: 비슷한 샘플끼리의 군집을 형성하는 것이며, 아래 용도에 활용된다.**
- 데이터 분석
- 고객분류
- 추천 시스템
- 검색 엔진
- 이미지 분할
- 차원 축소
- 준지도 학습

**이상치 탐지: 정상 데이터와 이상치를 구분하는 데에 활용된다.**
- 생산라인에서 결함제품 탐지
- 새로운 트렌드 찾기

**데이터 밀도 추정: 데이터셋의 확률밀도를 추정한다.**
- 이상치 분류: 밀도가 낮은 지역에 위치한 샘플
- 데이터 분석
- 데이터 시각화

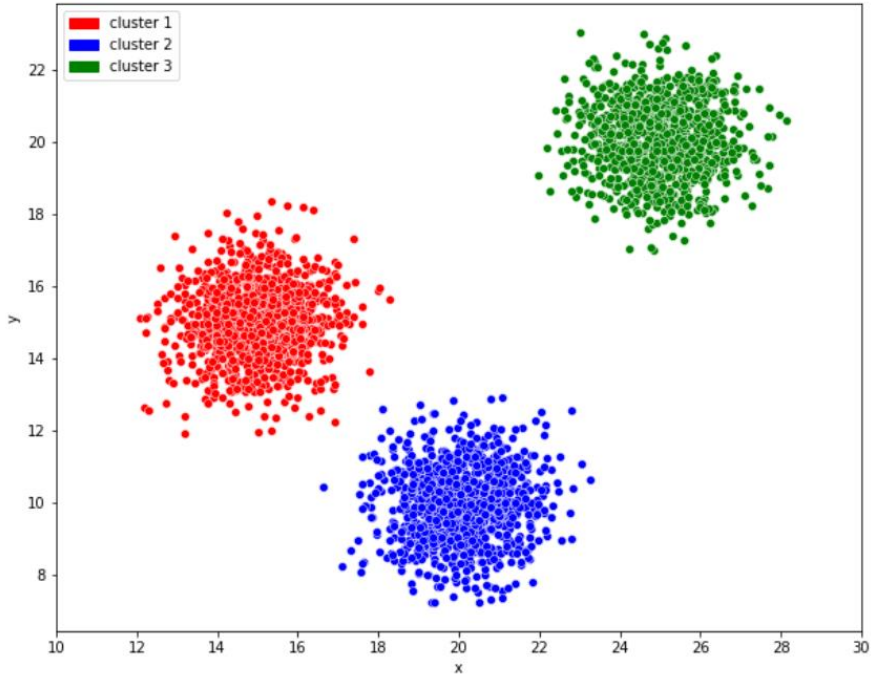
# 군집화(Clustering)

Cluster : 유사한 데이터들의 모음

Clustering : 데이터 포인트들을 별개의 군집으로 그룹화하는 것

유사성이 높은 데이터들을 동일한 그룹으로 분류하고

서로 다른 군집들이 상이성을 가지도록 그룹화 한다.



# 군집화 활용 분야

고객, 마켓, 브랜드, 사회 경제활동 분류

이상검출(Anomaly detection)

...

## 군집화(Clustering) 알고리즘

K-means

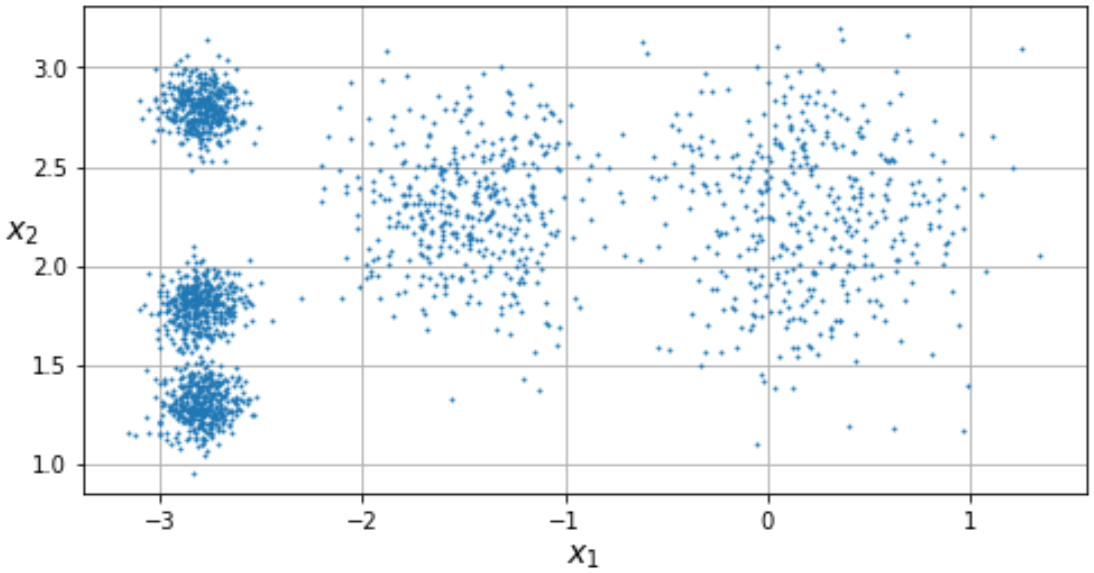
Hierarchical Clustering

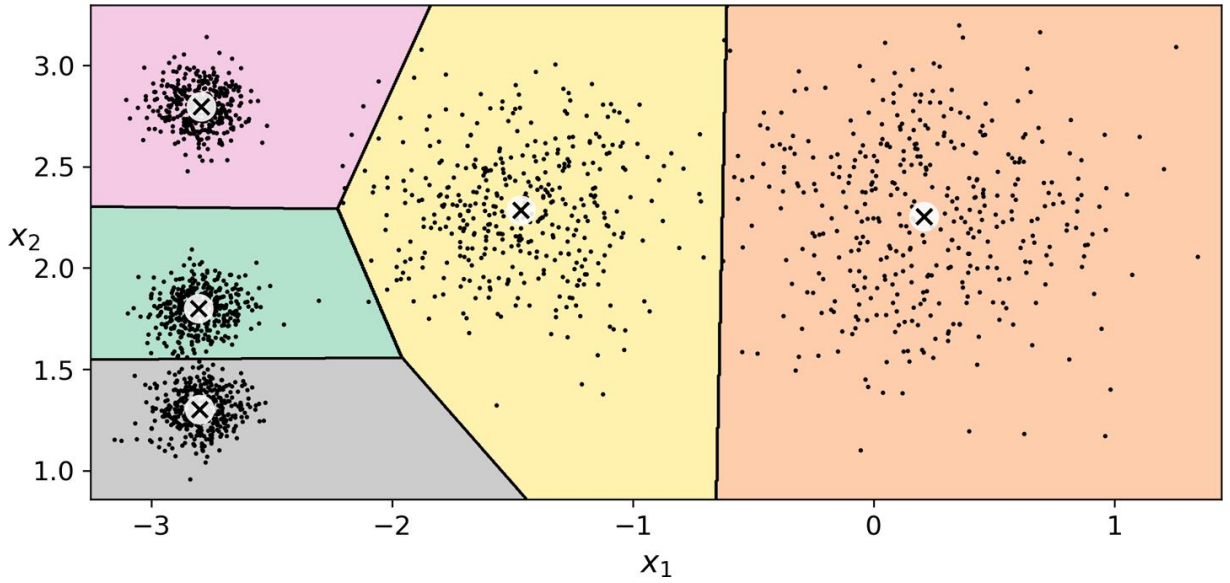
DBSCAN

Gaussian Mixture Model

# K-means

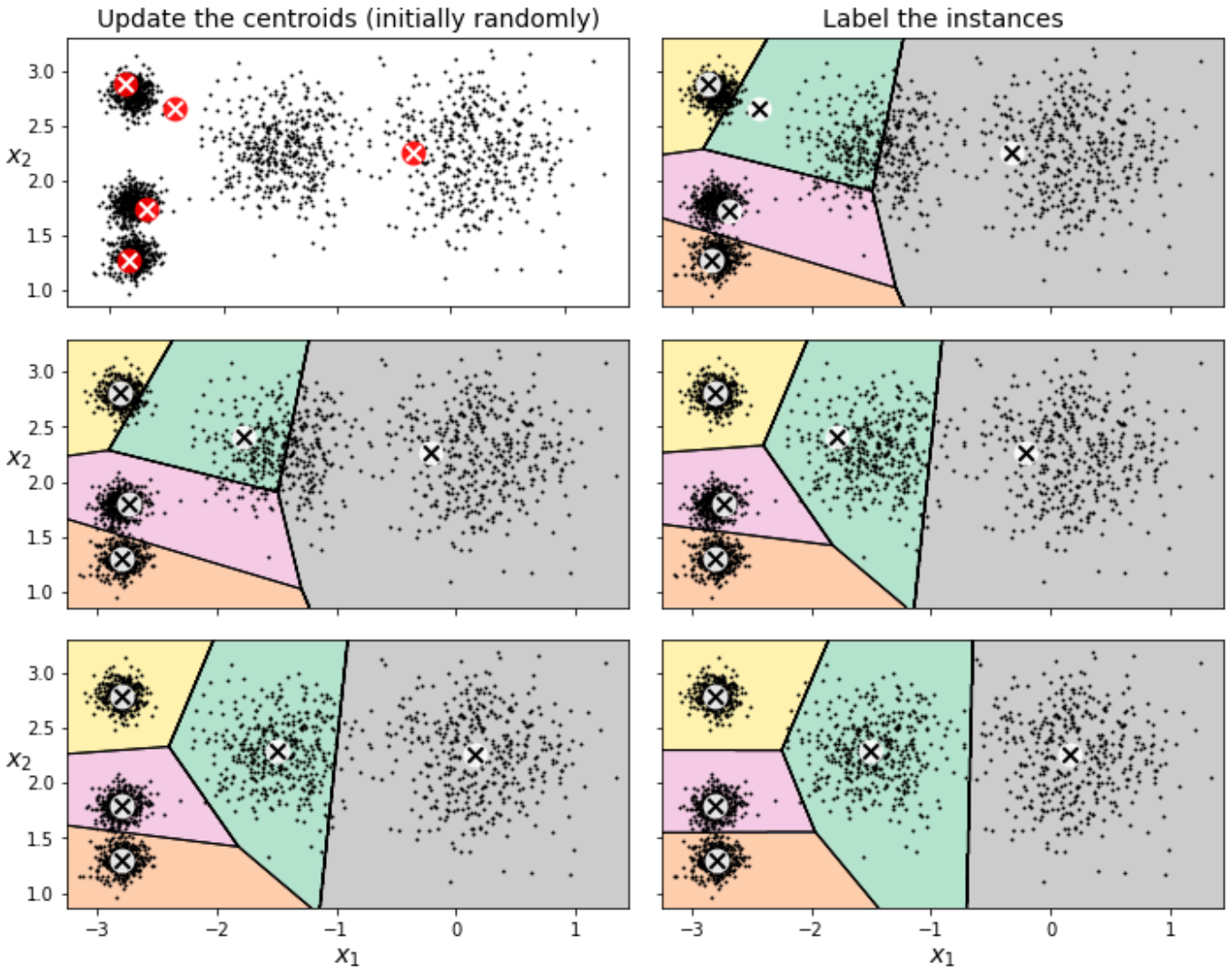
```
k = 5
kmeans = KMeans(n_clusters=k, random_state=42)
y_pred = kmeans.fit_predict(X)
```



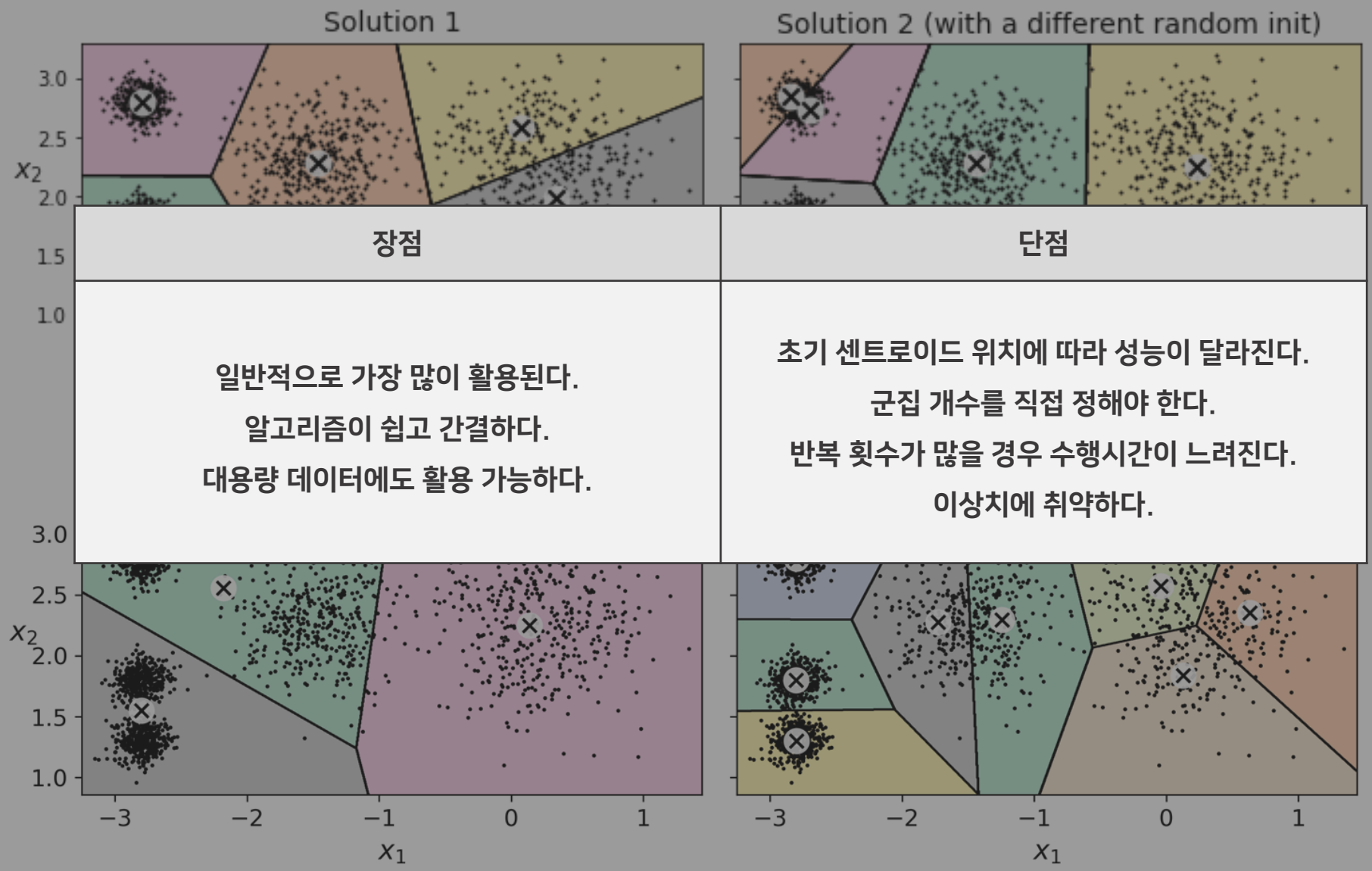




K = 5

# K-means

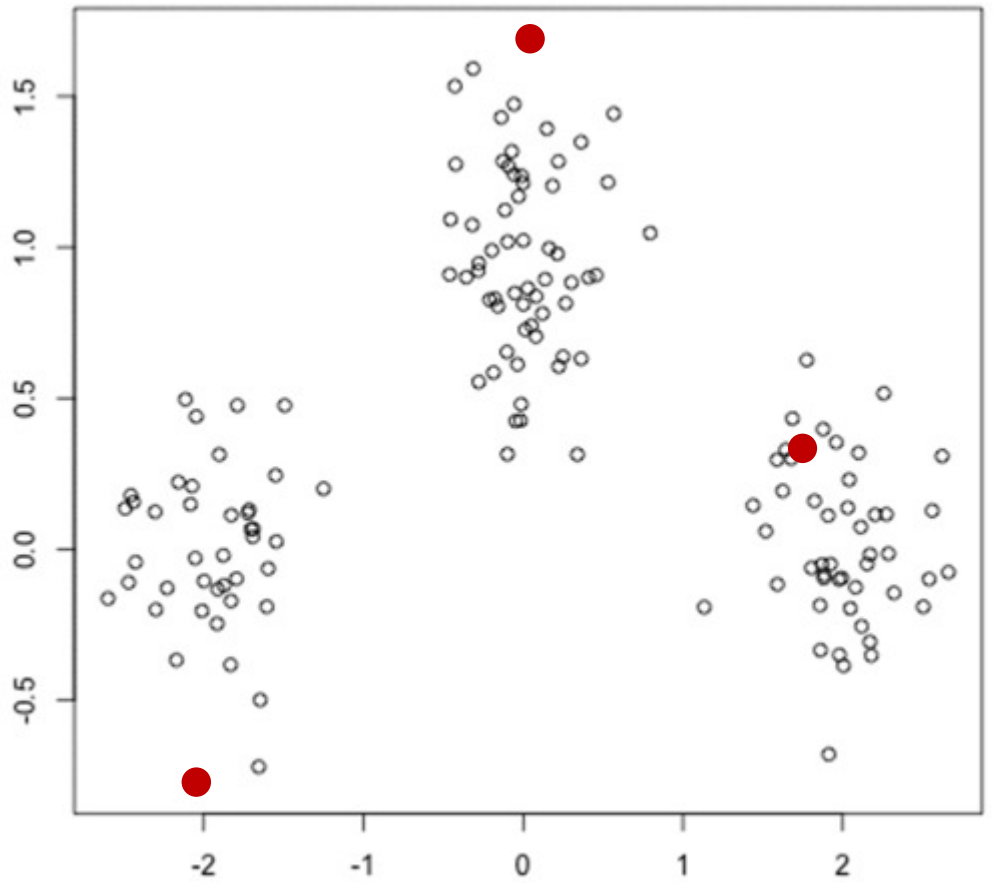

Update the centroids (initially randomly)
Label the instances
$x_2$
$x_1$
3.0
2.5
2.0
1.5
1.0
−3
−2
−1
0
1

# K-means의 단점




| 장점 | 단점 |
| --- | --- |
| 일반적으로 가장 많이 활용된다.<br>알고리즘이 쉽고 간결하다.<br>대용량 데이터에도 활용 가능하다. | 초기 센트로이드 위치에 따라 성능이 달라진다.<br>군집 개수를 직접 정해야 한다.<br>반복 횟수가 많을 경우 수행시간이 느려진다.<br>이상치에 취약하다. |

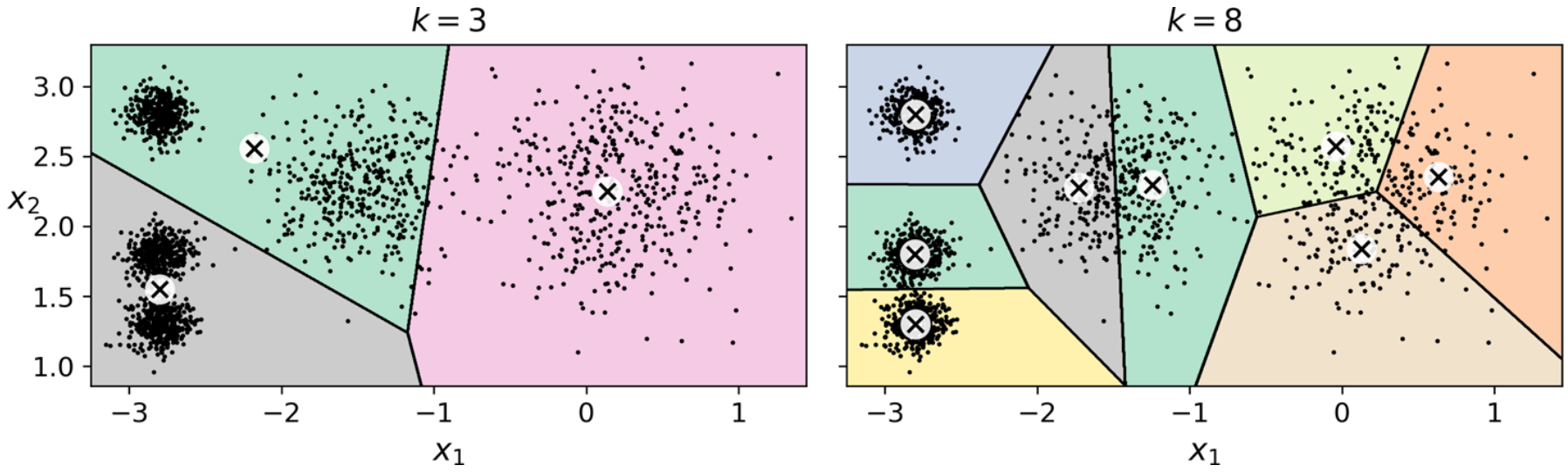
# K-means++



# sklearn의 K-means 모델

```
kmeans = KMeans(n_clusters=k,
                init='k-means++',
                n_init=10,
                max_iter=300,
                random_state=42)
```

# 최적의 군집수




**군집수가 적절하지 않으면 좋지 않은 모델로 수렴할 수 있다.**

**실루엣 계수**

# 실루엣 계수

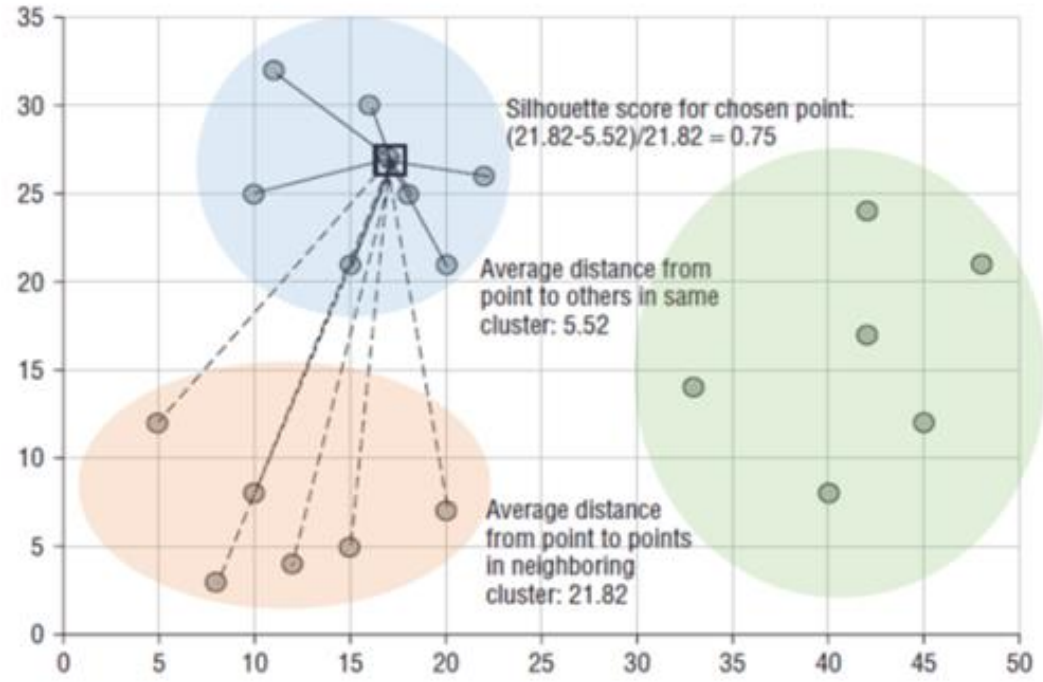
## 실루엣 계수 (Sihouette Coefficient)

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

군집 내에서는 밀도가 높을 수록 Good!

a : 동일 군집 내의 다른 데이터와의 거리의 평균값
b : 가장 가까운 타 군집에 속하는 데이터들과의 거리 평균값

다른 군집과는 거리가 멀수록 Good!




Cluster : 유사한 데이터들의 모음

Clustering : 데이터 포인트들을 별개의 군집으로 그룹화하는 것

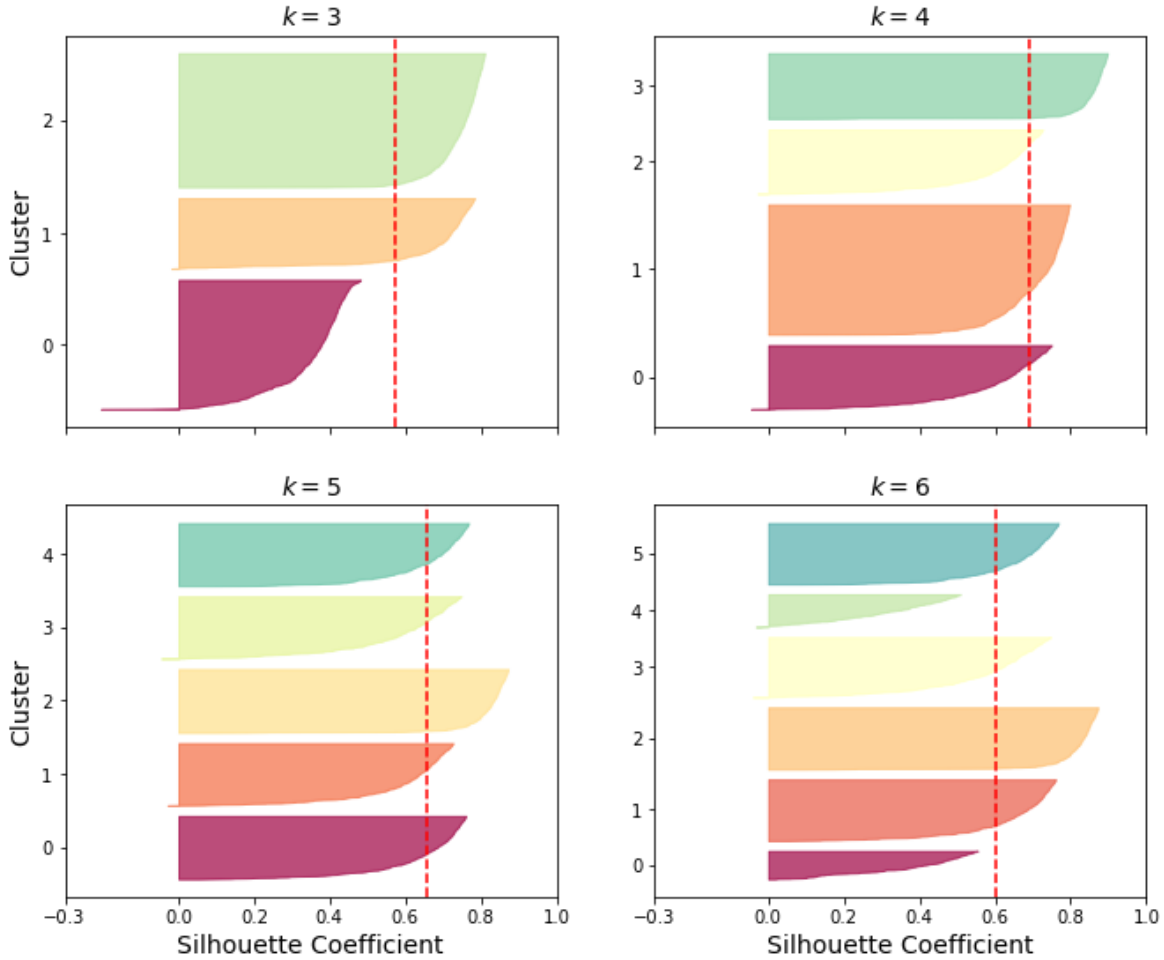
유사성이 높은 데이터들을 동일한 그룹으로 분류하고

서로 다른 군집들이 상이성을 가지도록 그룹화 한다.

# 실루엣 계수

**각 데이터의 실루엣 계수들을 모아놓은 그래프, 실루엣 다이어그램**

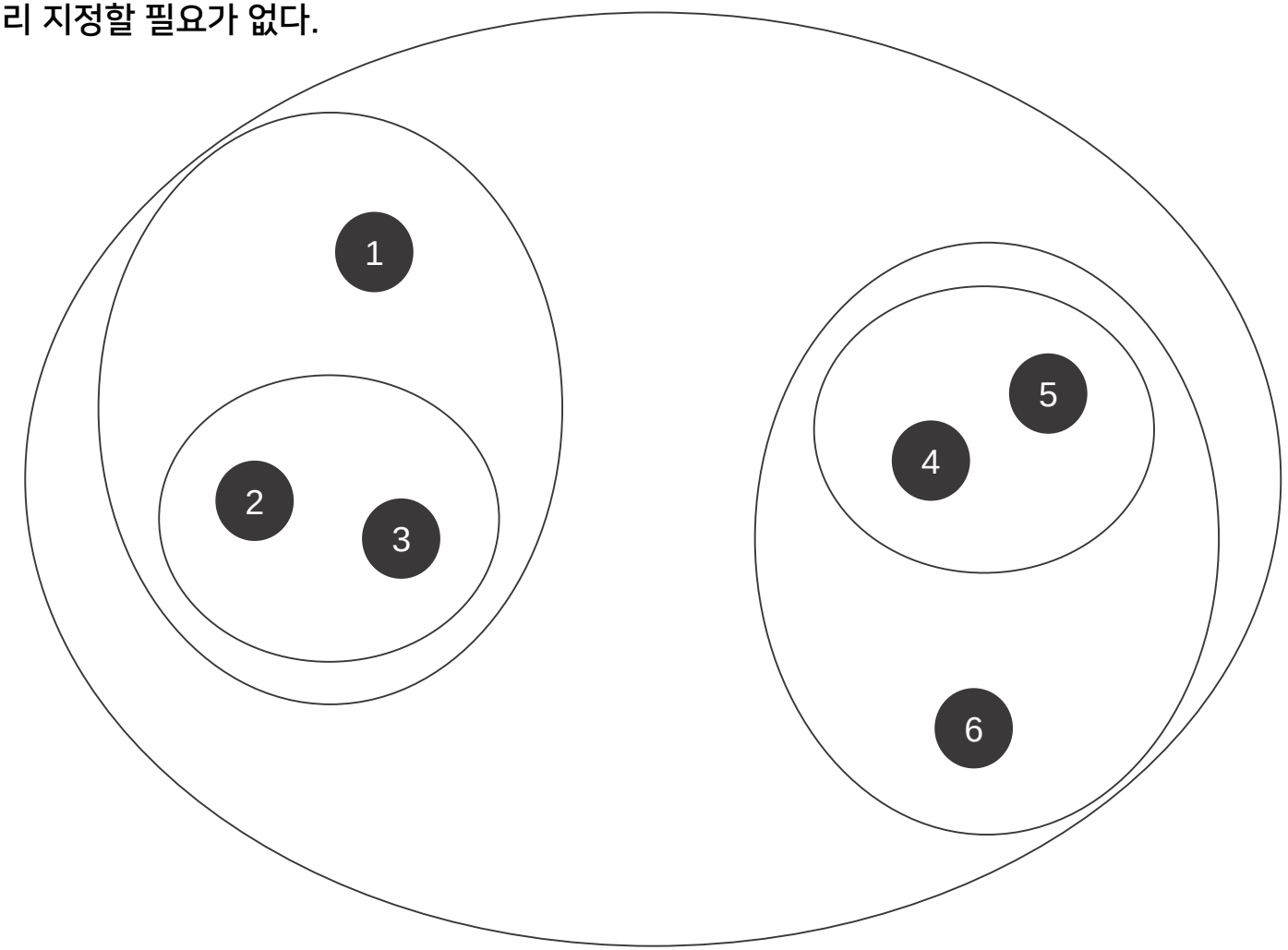
- 군집 개수별로 데이터들의 실루엣 계수들을 시각화
- 평균만 높다고 좋은 군집이 아니라, 군집별 크기가 비슷해야 좋은 모델



# Hierarchical Clustering

**Hierarchical Clustering(계층적 군집분석) : 데이터를 하나하나 계층에 따라 순차적으로 클러스터링 하는 기법**

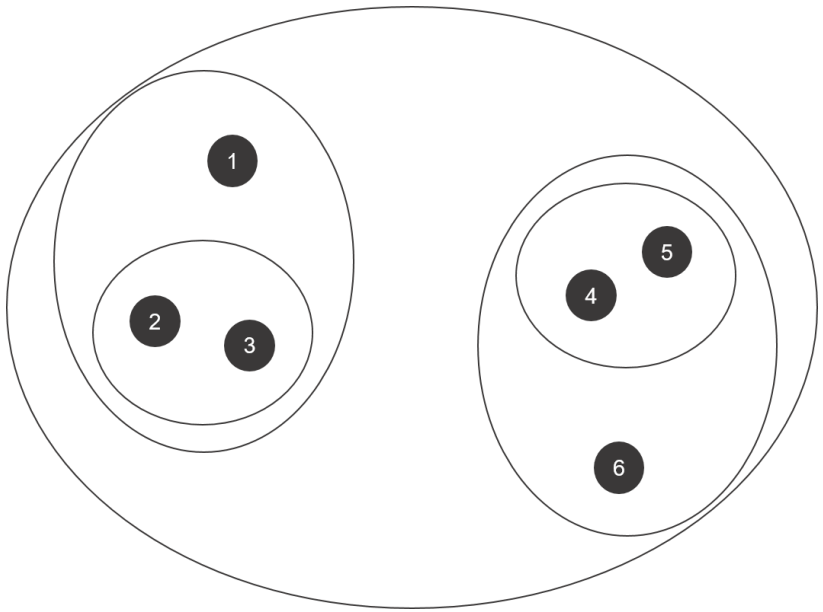
- 클러스터의 개수를 미리 지정할 필요가 없다.

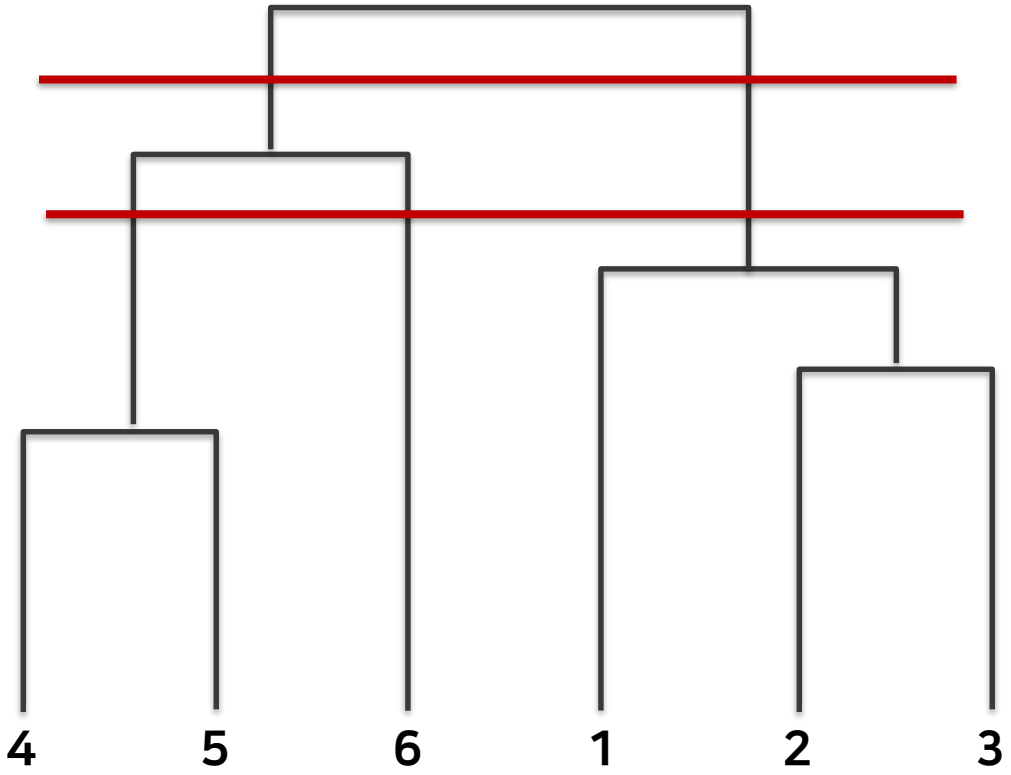


# Hierarchical Clustering

**Hierarchical Clustering(계층적 군집분석) : 데이터를 하나하나 계층에 따라 순차적으로 클러스터링 하는 기법**

- 클러스터의 개수를 미리 지정할 필요가 없다.

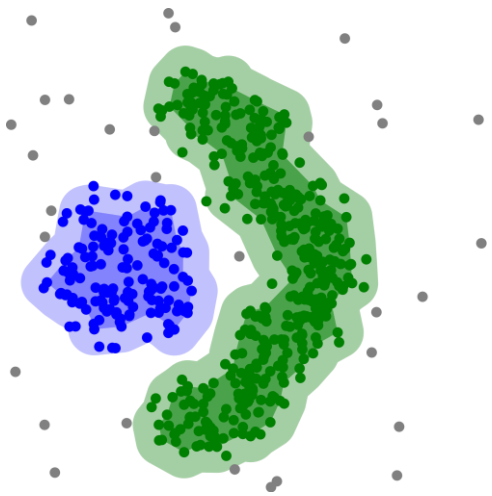


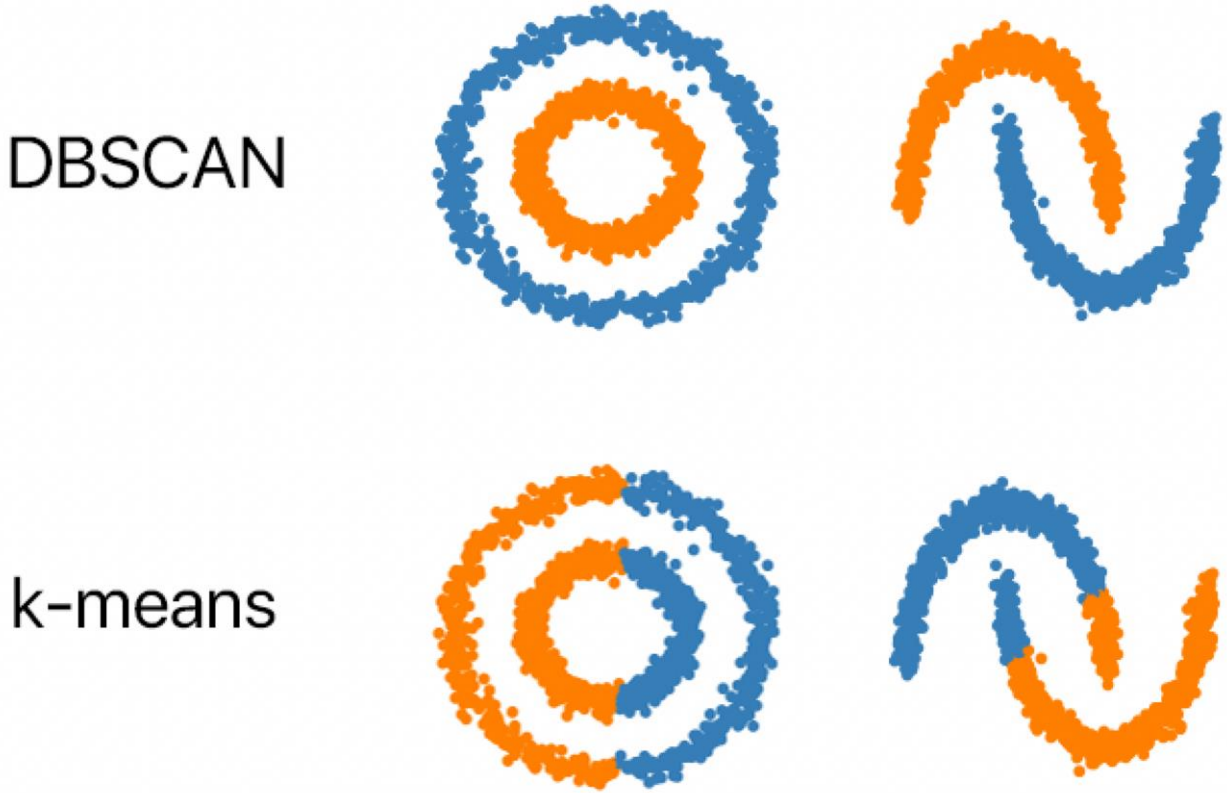



# DBSCAN

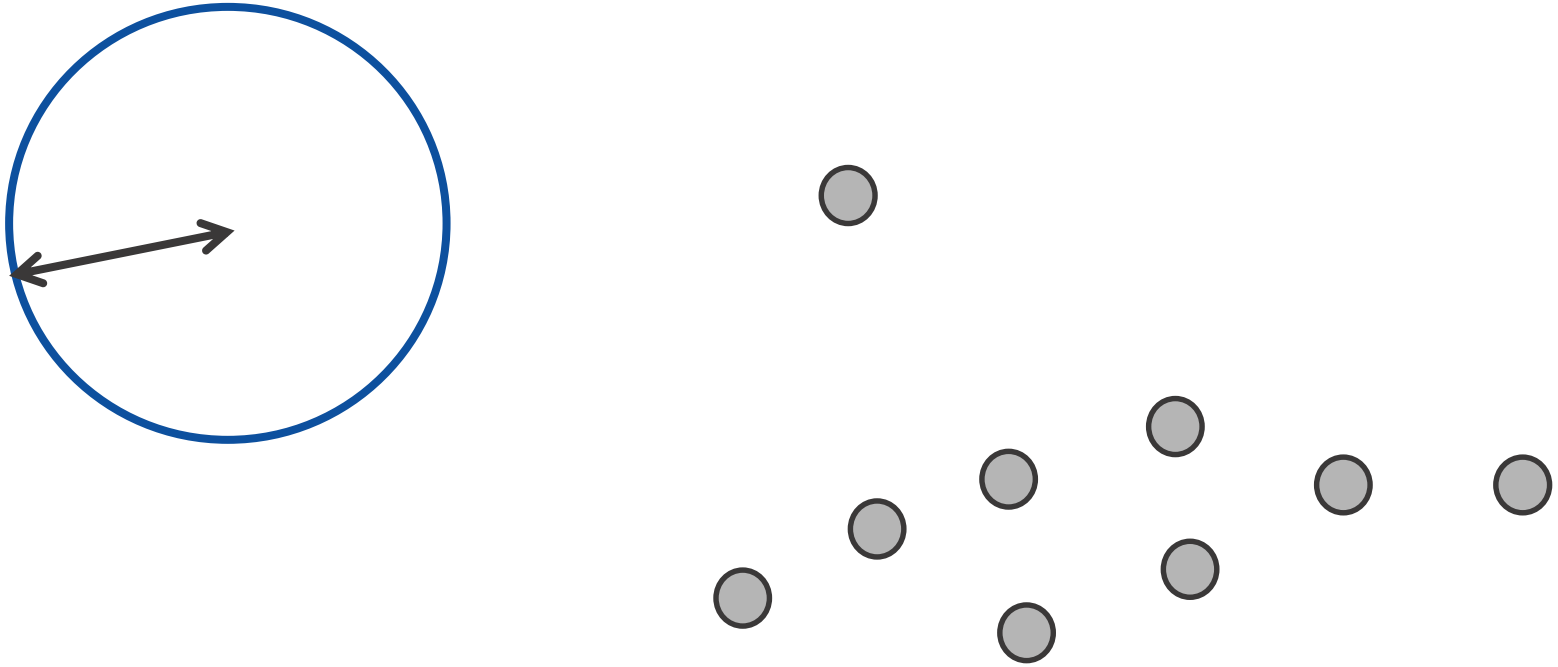
## DBSCAN : Density-Based Spatial Clustering of applications with Noise

- 클러스터의 개수를 미리 지정할 필요가 없다.
- 이상치(Outlier)를 효과적으로 제외할 수 있다.



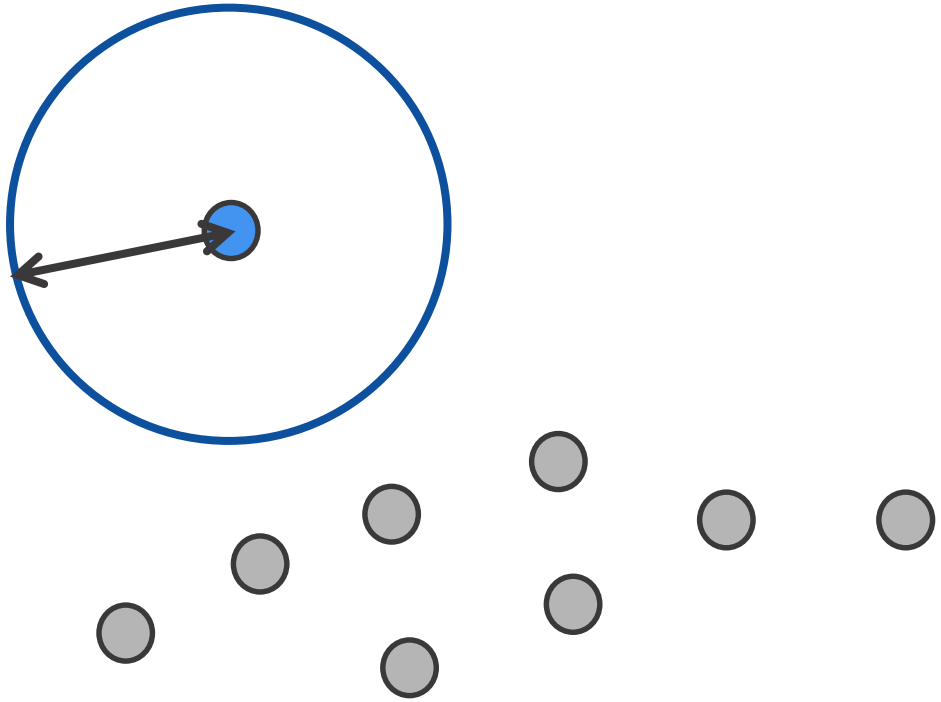


# DBSCAN



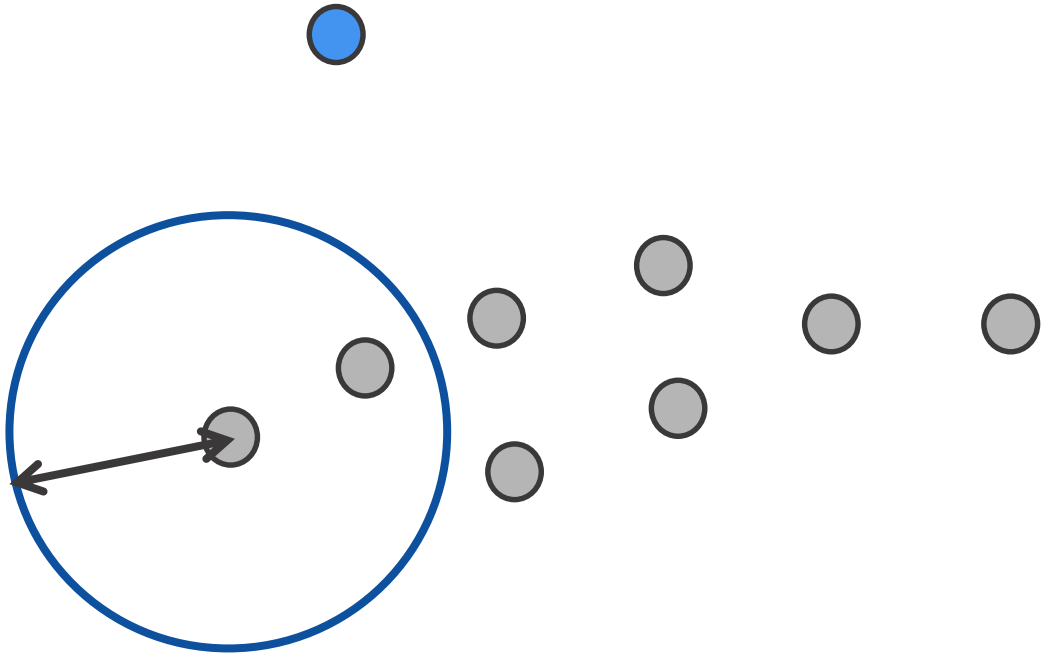
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



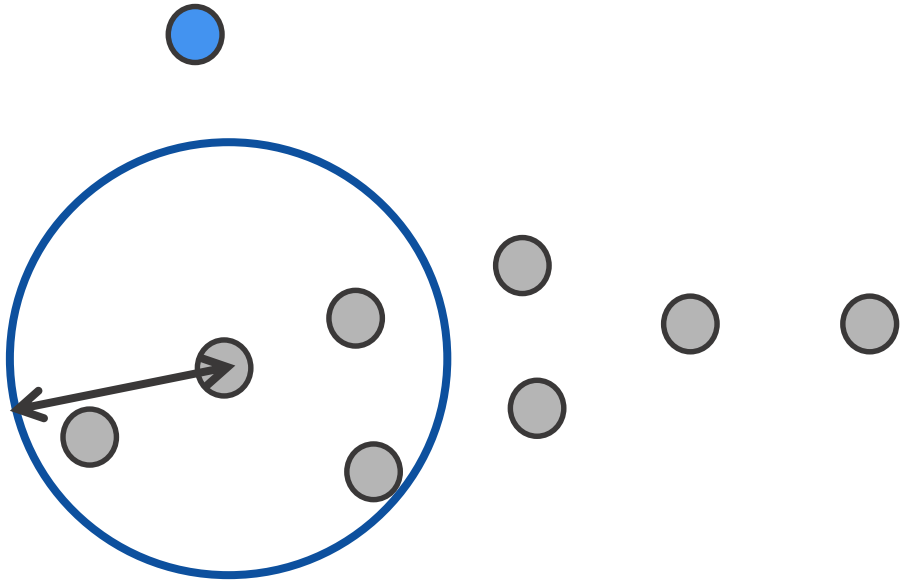
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



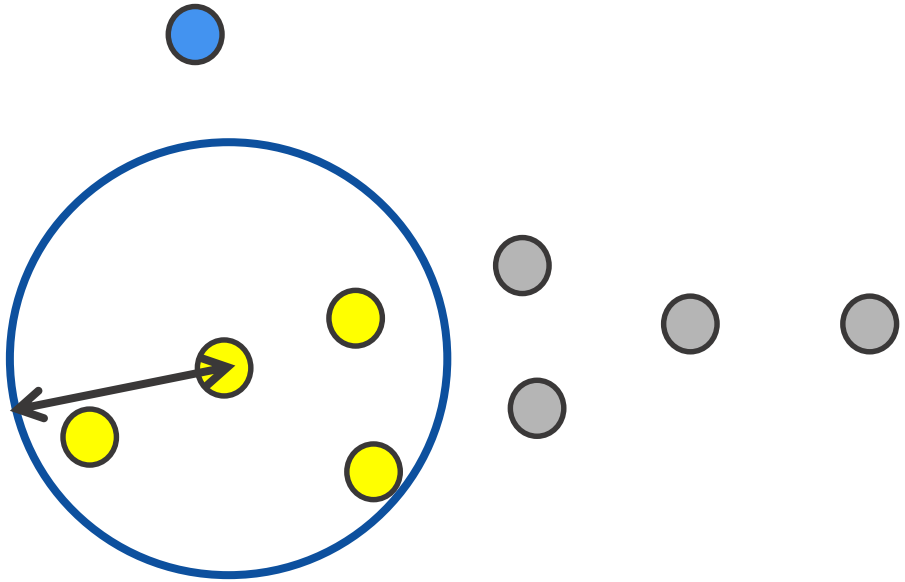
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



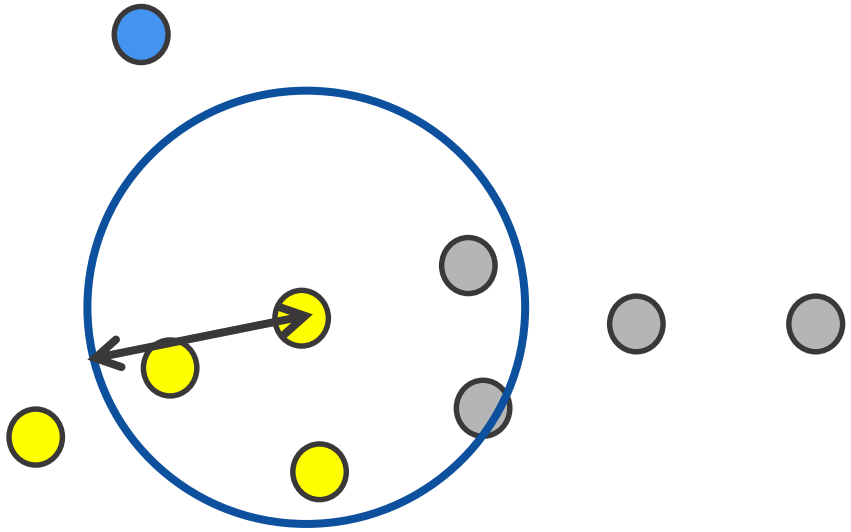
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



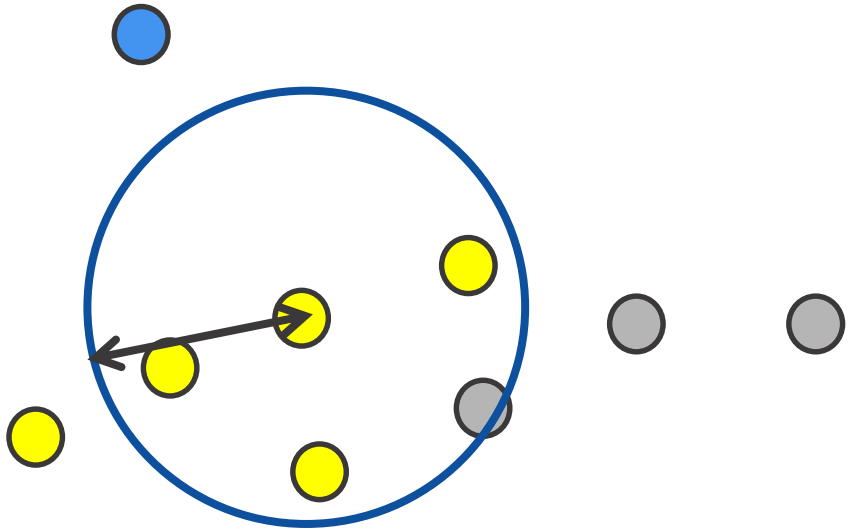
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



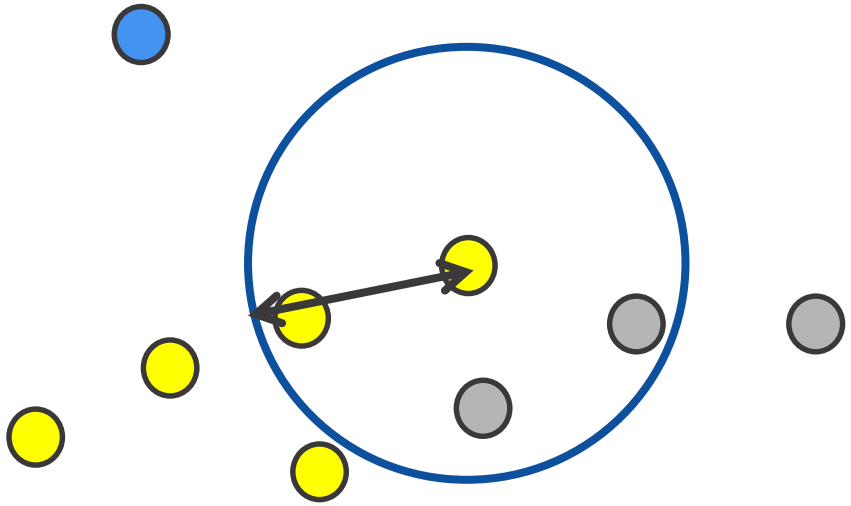
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



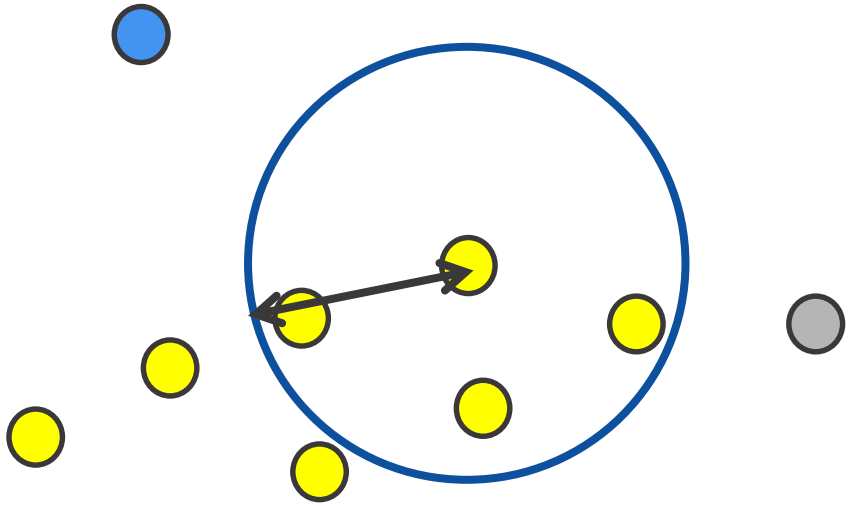
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



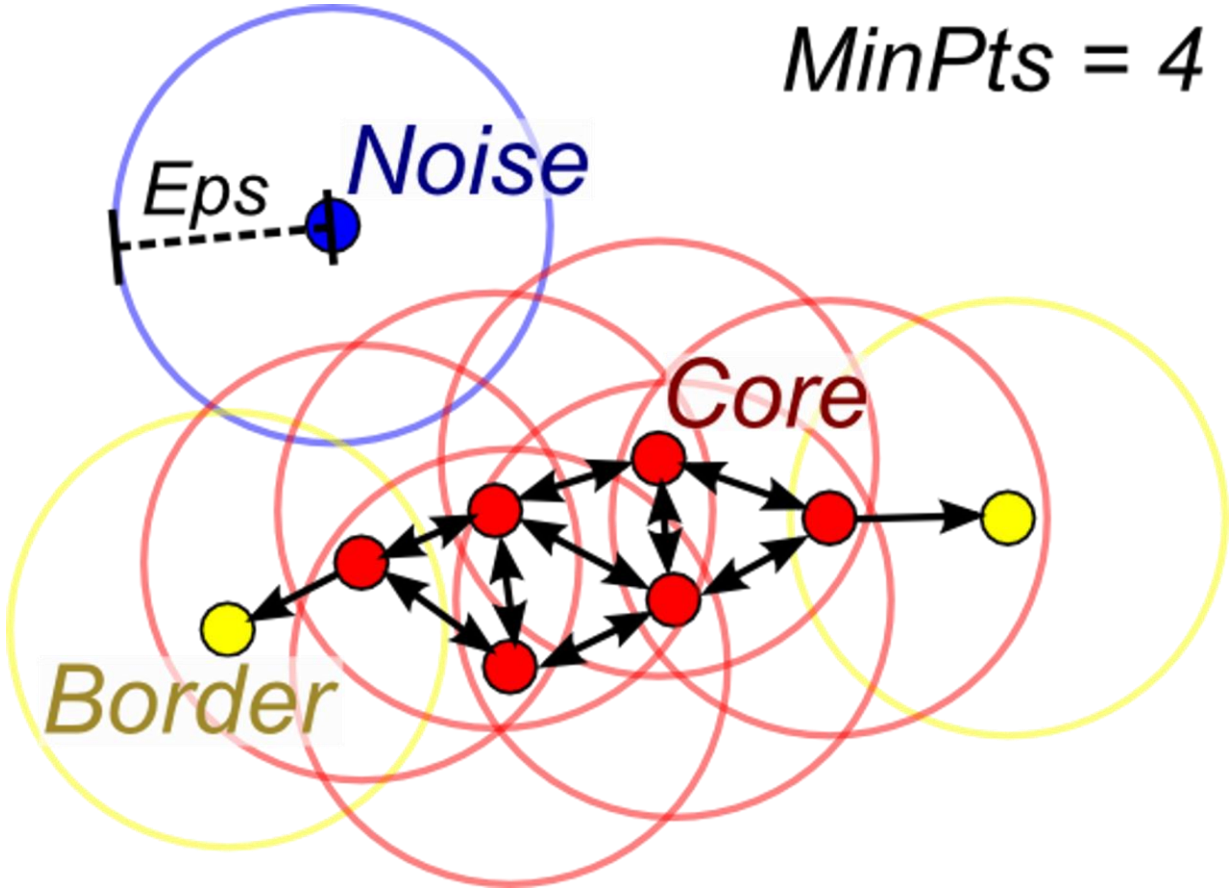
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN



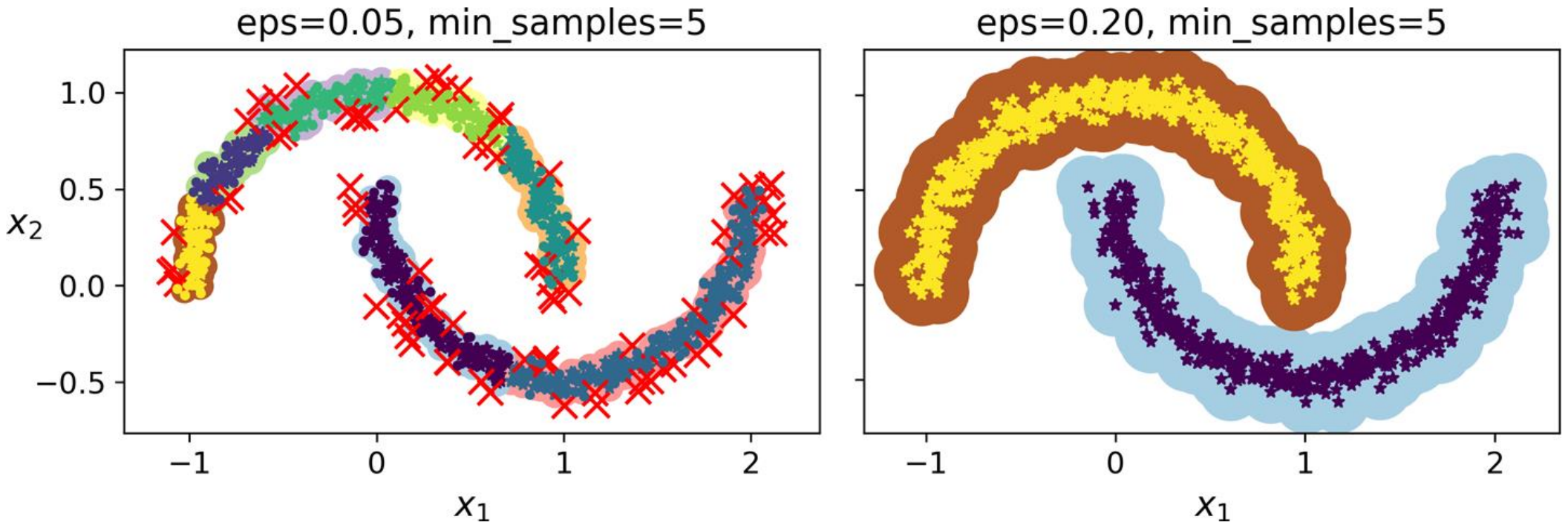
Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN

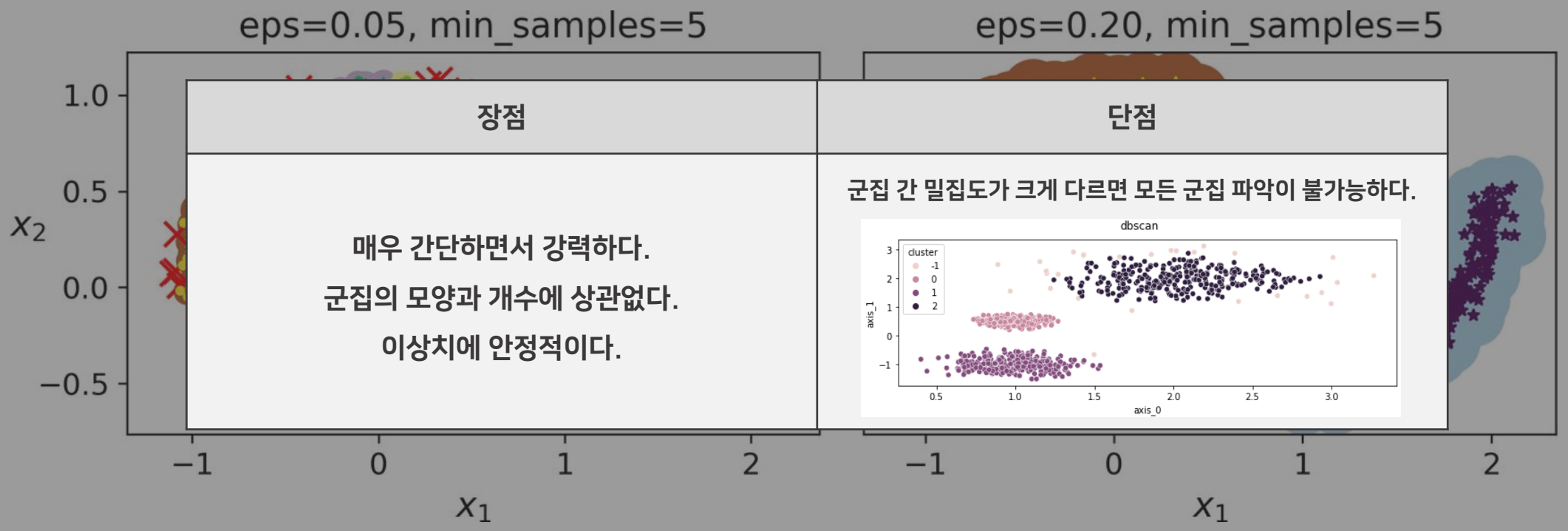



Eps : 반경의 크기
min samples : 최소 군집의 크기

# DBSCAN


eps=0.05, min_samples=5
eps=0.20, min_samples=5
1.0
0.5
0.0
−0.5
$x_2$
−1
0
1
2
$x_1$
−1
0
1
2
$x_1$

# DBSCAN

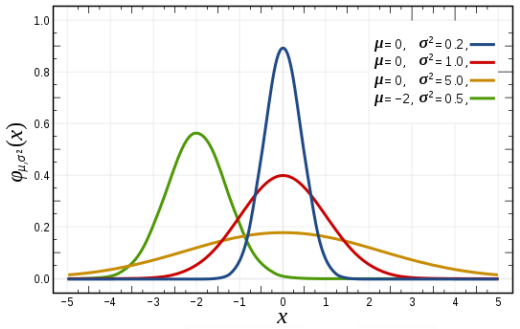


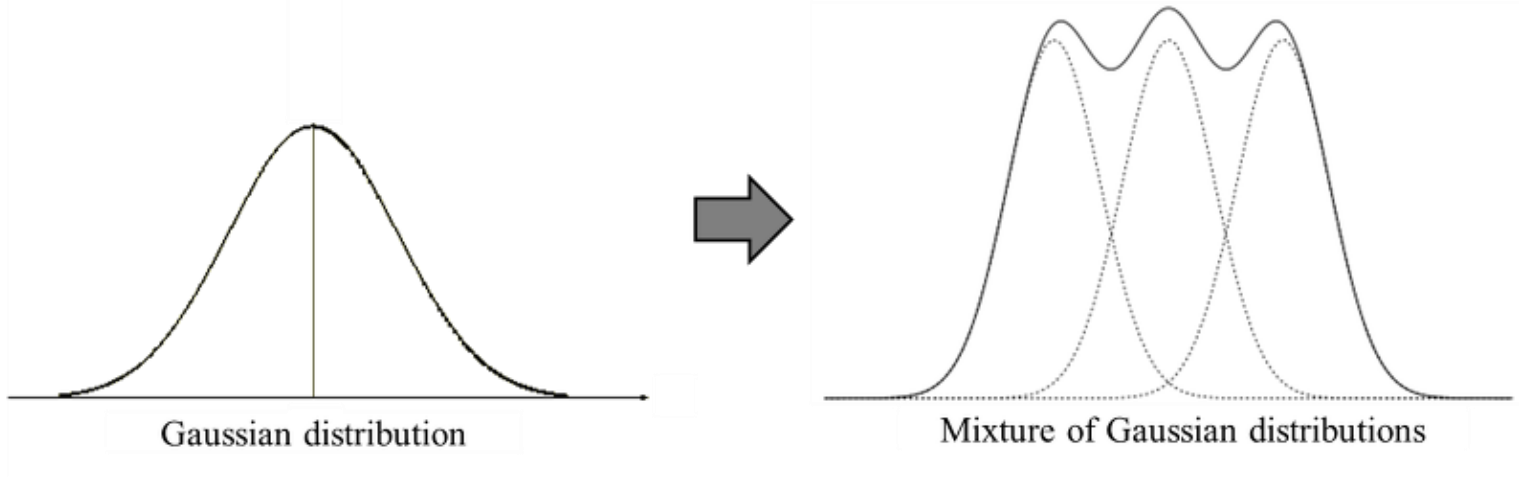

| 장점 | 단점 |
| --- | --- |
| 매우 간단하면서 강력하다.<br>군집의 모양과 개수에 상관없다.<br>이상치에 안정적이다. | 군집 간 밀집도가 크게 다르면 모든 군집 파악이 불가능하다. |

# Gaussian Mixture Model (GMM)

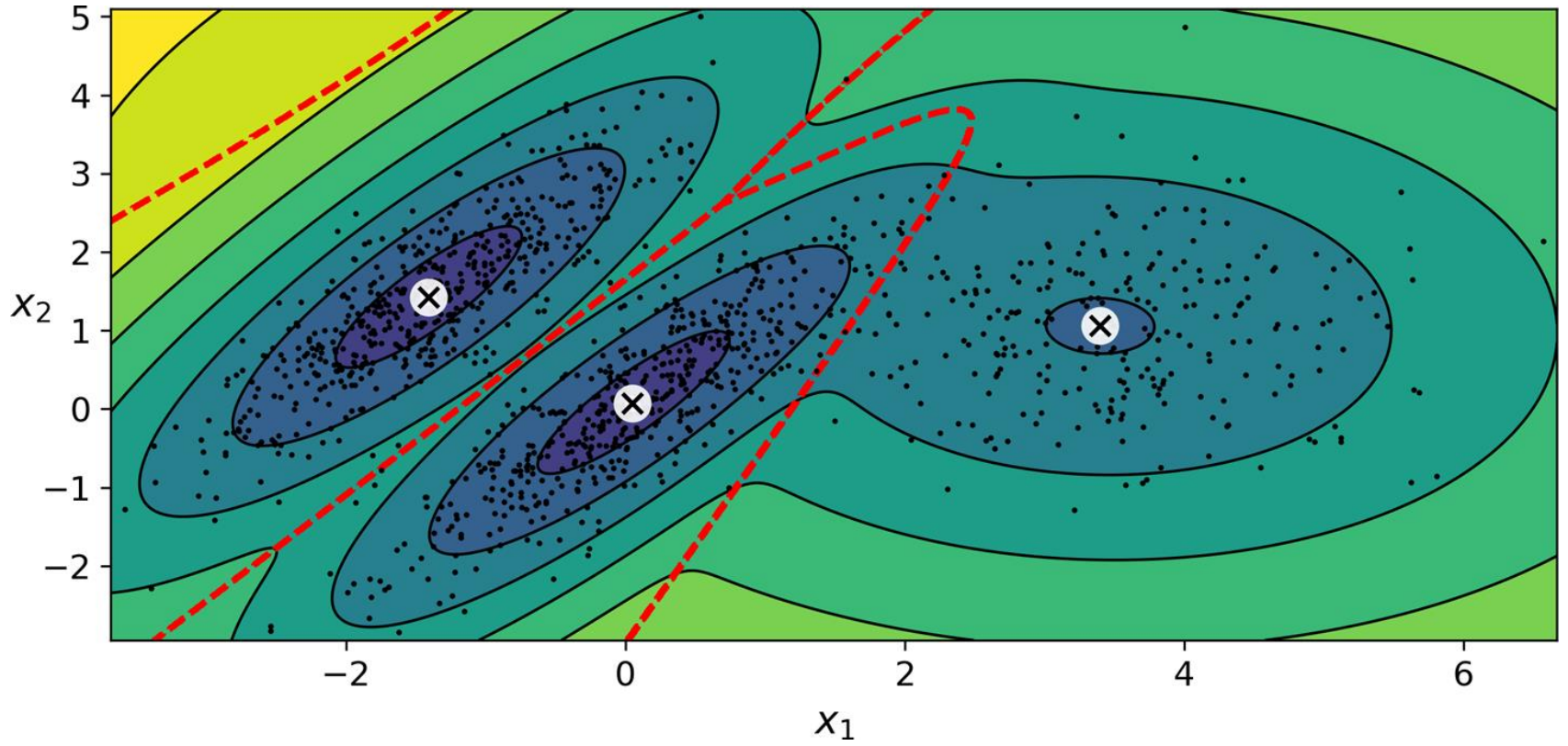
## 가우시안 혼합 모델

- 데이터셋이 여러 개의 혼합된 가우시안 분포를 따르는 샘플들로 구성되었다고 가정



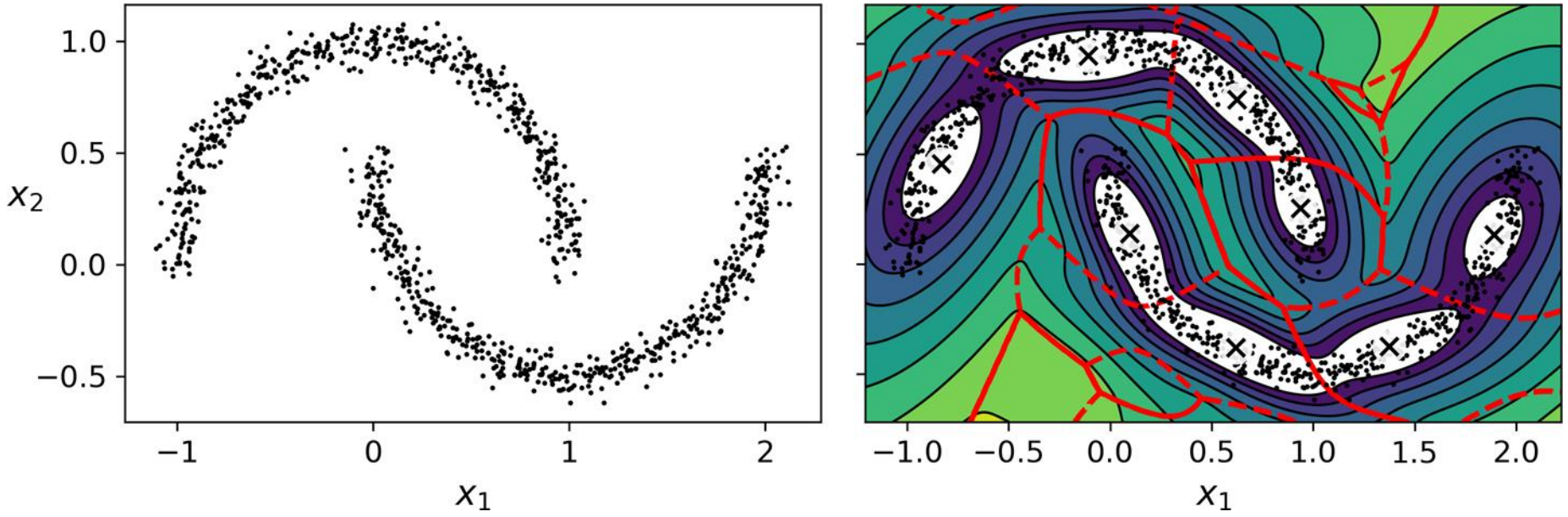



# Gaussian Mixture Model (GMM)




**군집의 모양, 크기, 밀집도, 방향이 다른 데이터 셋에 대해**
**가우시안 분포를 이루는 각 군집을 찾아냄.(타원형 군집 생성)**

# Gaussian Mixture Model (GMM)



# [실습]

K-means, Hierarchical, DBSCAN, GMM

# 감사합니다