

# 데이터마이닝팀

4팀

김수빈  
조건우  
김보현  
이지원  
조성우

# CONTENTS

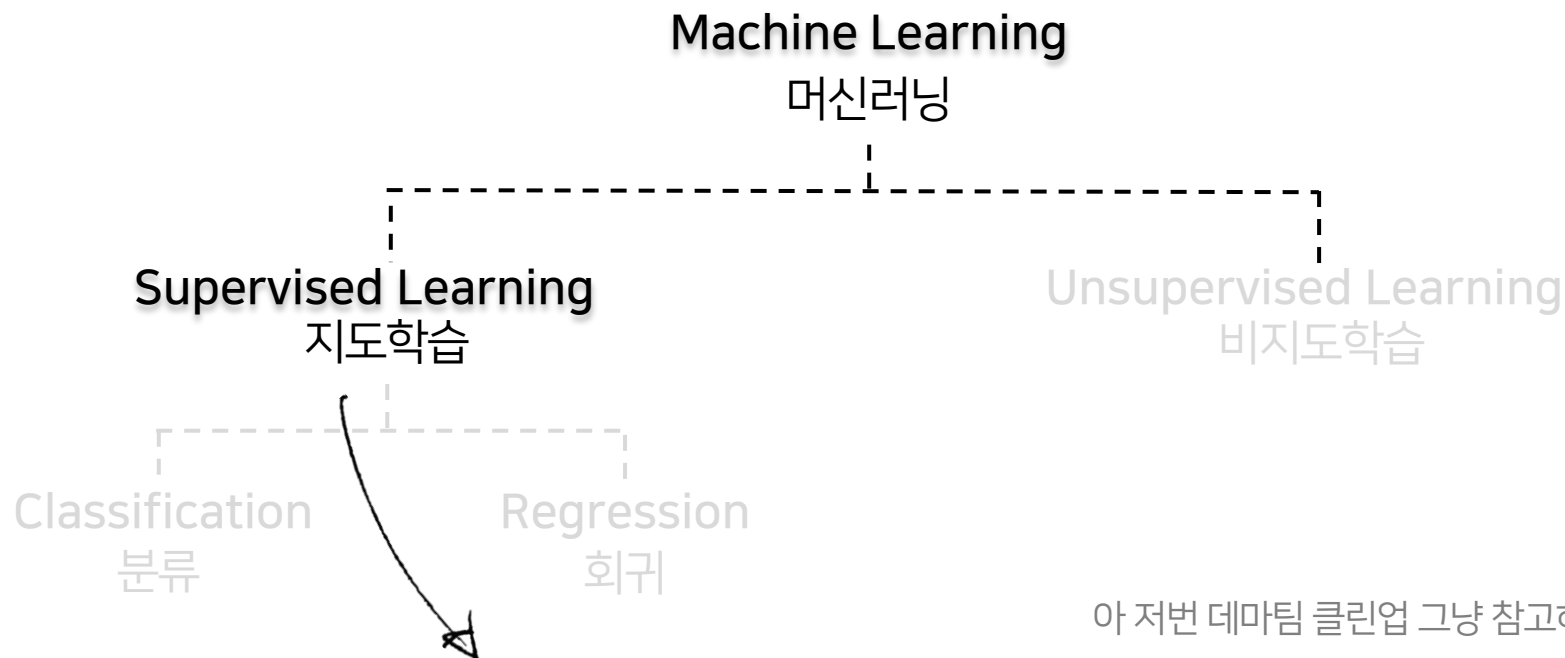
1. 클러스터링

2. 추천 시스템

1

클러스터링

## 클러스터링



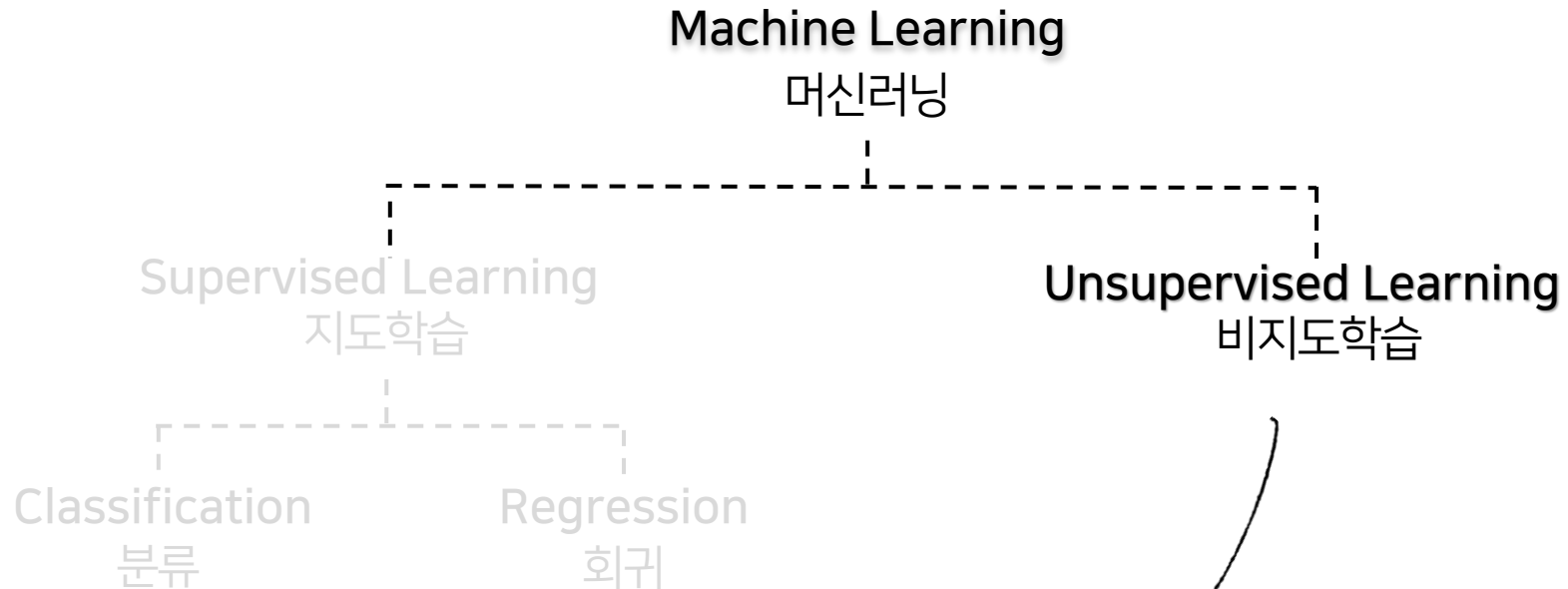
아 저번 데마팀 클린업 그냥 참고하라고~

**Y값 존재**

문제 수행 및 답 확인 가능



## 클러스터링

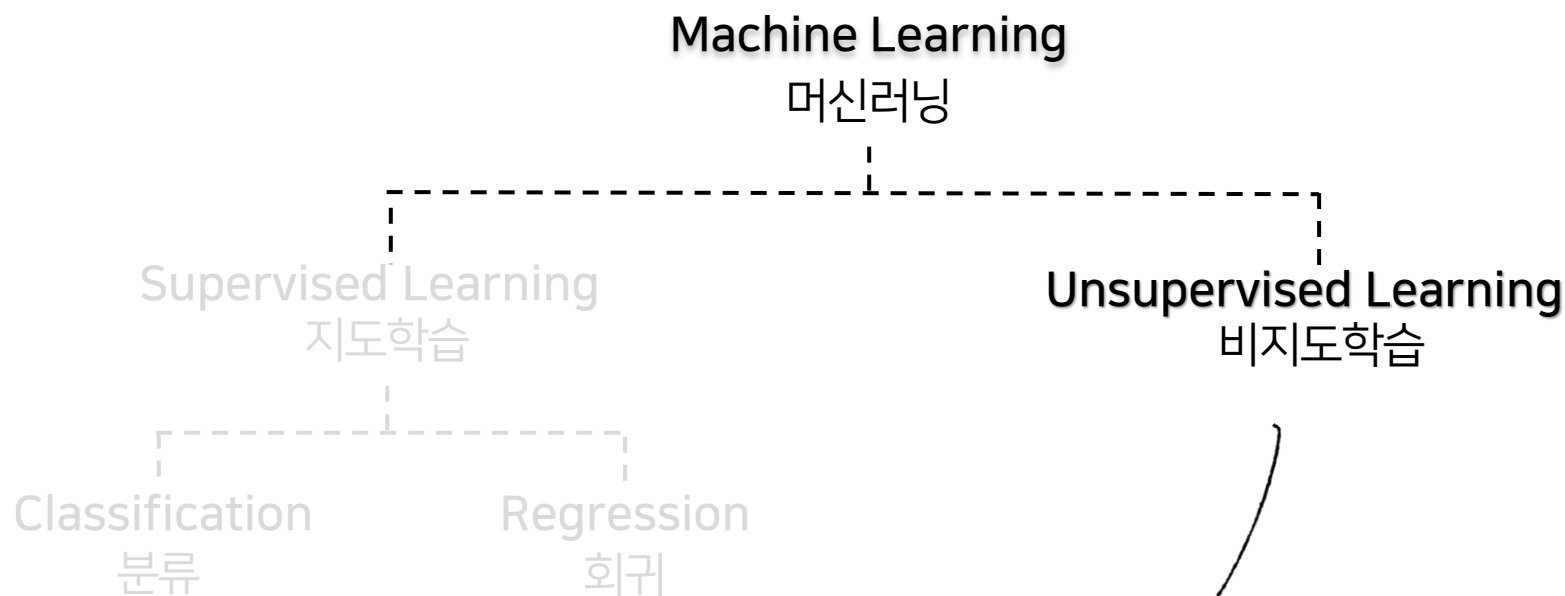


**Y값 존재 X**

데이터의 구조를 묘사하고 관계를 해석하는 데 초점

Ex) 클러스터링, 주성분 분석(PCA)

## 클러스터링



두 귀가 쫑긋

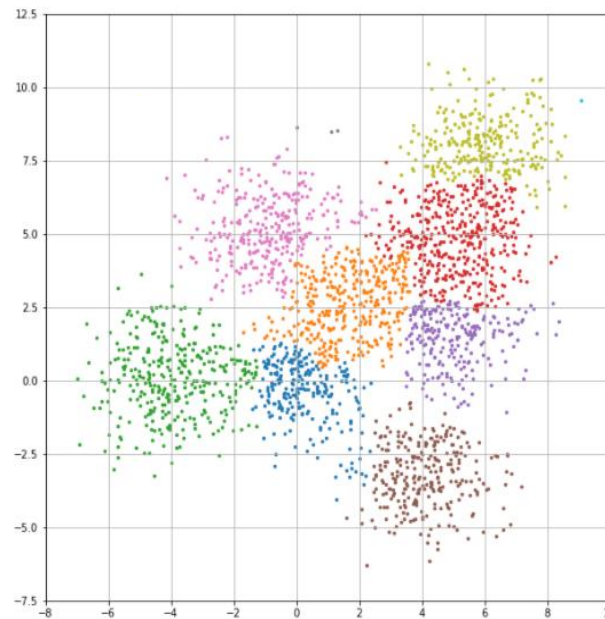
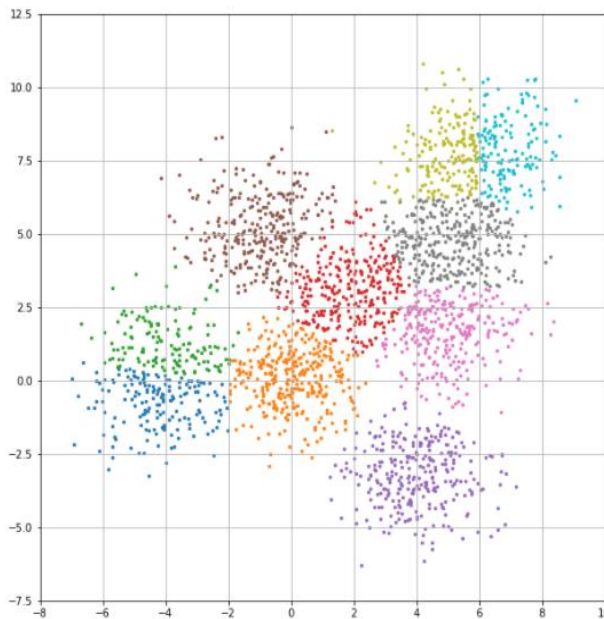
Y값 존재 X

데이터의 구조를 요약하고 관계를 해석하는 데 초점

(Ex) 클러스터링, 주성분 분석(PCA)

## 클러스터링

## 클러스터링



데이터 내에 숨어있는  
새로운 그룹(군집)을 찾아내는 것 !





## 클러스터링

클러스터링

# 군집이 잘 분리된 기준은 무엇일까?

✓ 같은 그룹 내의 객체들은 서로 **비슷**

✓ 다른 그룹들의 객체들끼리는 서로 **달라**

데이터 내에 숨어있는  
새로운 그룹(군집)을 찾아내는 것 !

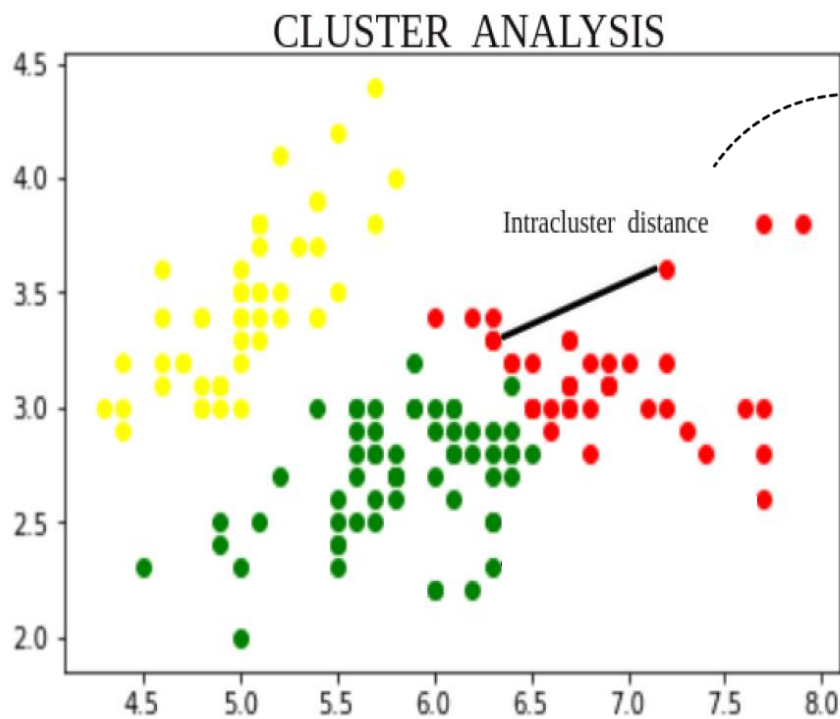
4달라~





## 클러스터링

클러스터링

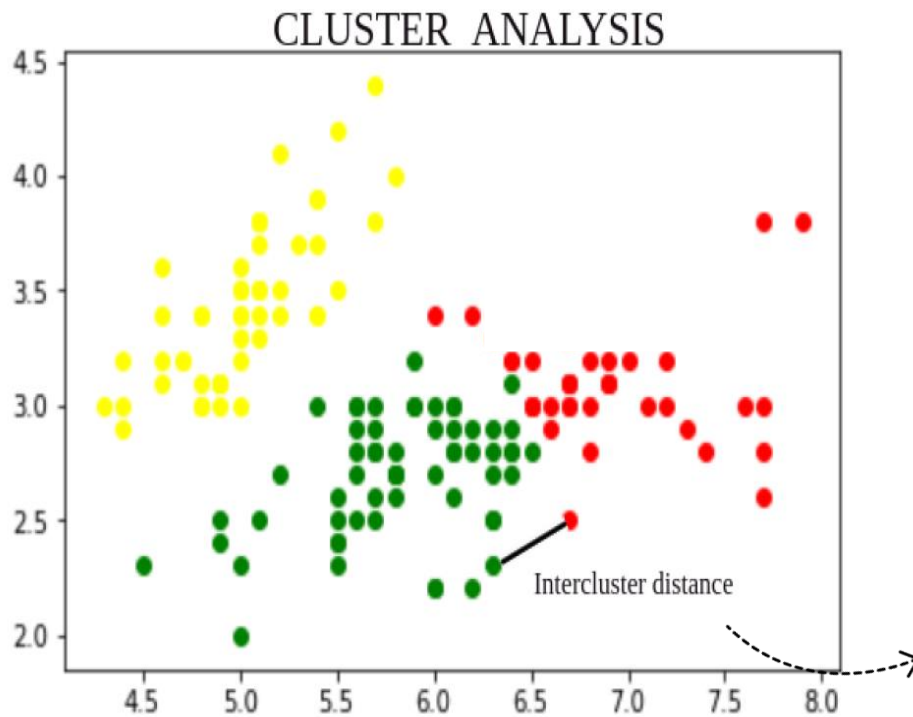


- Intra-Cluster Distance -

같은 군집 내의 거리

## 클러스터링

클러스터링

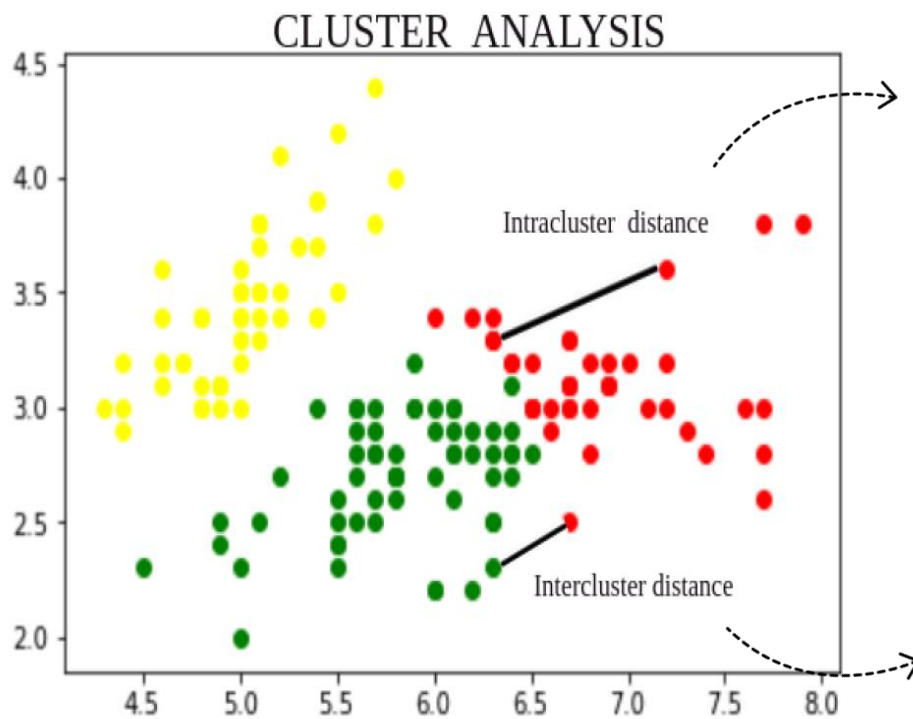


- Inter-Cluster Distance -

다른 군집 간의 거리

## 클러스터링

클러스터링



- Intra-Cluster Distance -

같은 군집 내의 거리

- Inter-Cluster Distance -

다른 군집 간의 거리



군집 내 거리는 최소화, 군집 간 거리는 최대화하는 것이 목표

## 클러스터링

클러스터링



클러스터가 적절히 생성되었나?  
클러스터 개수는 몇 개가 적절하지?

그게 할 소리야?



Dnn Family  
Index

Silhouette  
Method

Elbow Point  
Method

SD Validity  
Index

DB Index

## 클러스터링

클러스터링



클러스터가 적절히 생성되었나?  
클러스터 개수는 몇 개가 적절하지?

그게 할 소리야?



Silhouette  
Method



Elbow Point  
Method

## Silhouette Method

### Silhouette 방법

각각의 객체(데이터)별로 실루엣 계수를 확인하는 방법

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

$a(i)$

군집 내 거리 Intra - cluster variance

객체 i와 같은 군집 안에 속하는 나머지 객체들 간의 거리의 평균

$b(i)$

군집 간 거리 Inter-cluster variance

객체 i와 다른 군집에 속하는 나머지 객체들 간 거리의 평균의 최솟값

## Silhouette Method

### Silhouette 방법

각각의 객체(데이터)별로 실루엣 계수를 확인하는 방법

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

군집 내 거리인  $a(i)$  는 작을수록  
군집 간 거리인  $b(i)$  는 클수록 좋음 !

최고오오오



## Silhouette Method

Silhouette 계수

모든 데이터 포인트에 대해서 실루엣 계수 계산  
최종값으로 실루엣 계수들의 **평균값** 사용

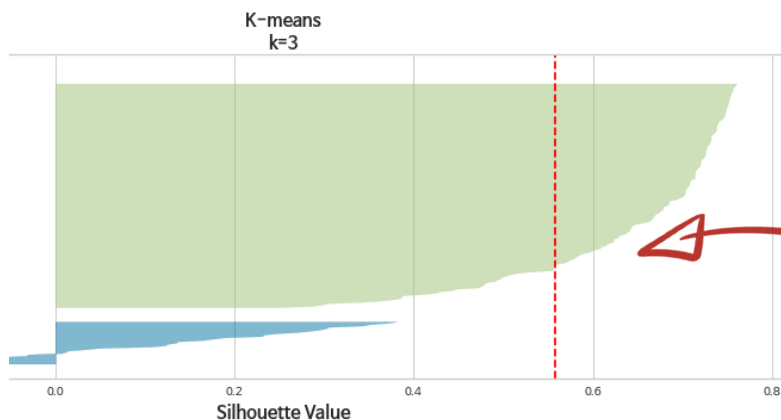


경험적으로 0.5가 넘으면 잘 묶인 클러스터링  
0.7이 넘으면 정말 잘 묶인 클러스터링이라고 판단



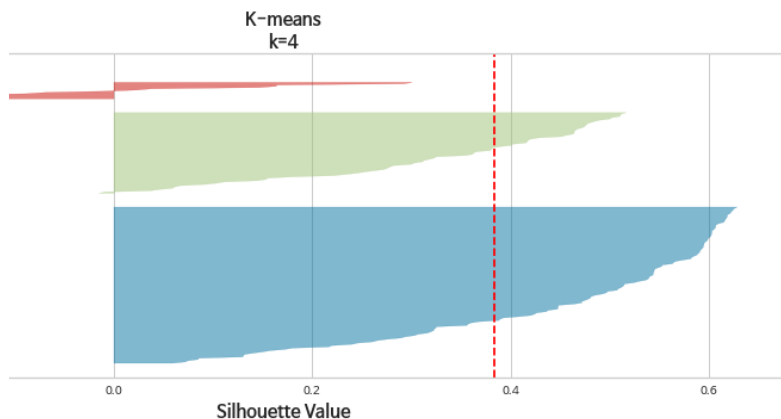
## Silhouette Method

### Silhouette 계수



클러스터 개수에 따른 실루엣 계수 시각화

아래로 내려올수록 클러스터 개수 증가  
빨간 점선으로 표시된 평균 실루엣 계수



3개로 나뉘었을 때가 군집 내 분산은 작고,  
군집 간 분산은 크구나!

## Elbow Point Method

Elbow Point 방법

클러스터 내 RSS가 **최소**가 되도록 클러스터의 중심을 결정해 나가는 방법

즉, 클러스터 내 분산이 최소가 되게끔 하는,  
클러스터 내 중심점과 객체들 간의 거리가 최소가 되게끔 하는 중심점 선택

## Elbow Point Method

### Elbow Point 방법

클러스터 내 RSS가 **최소**가 되도록 클러스터의 중심을 결정해 나가는 방법

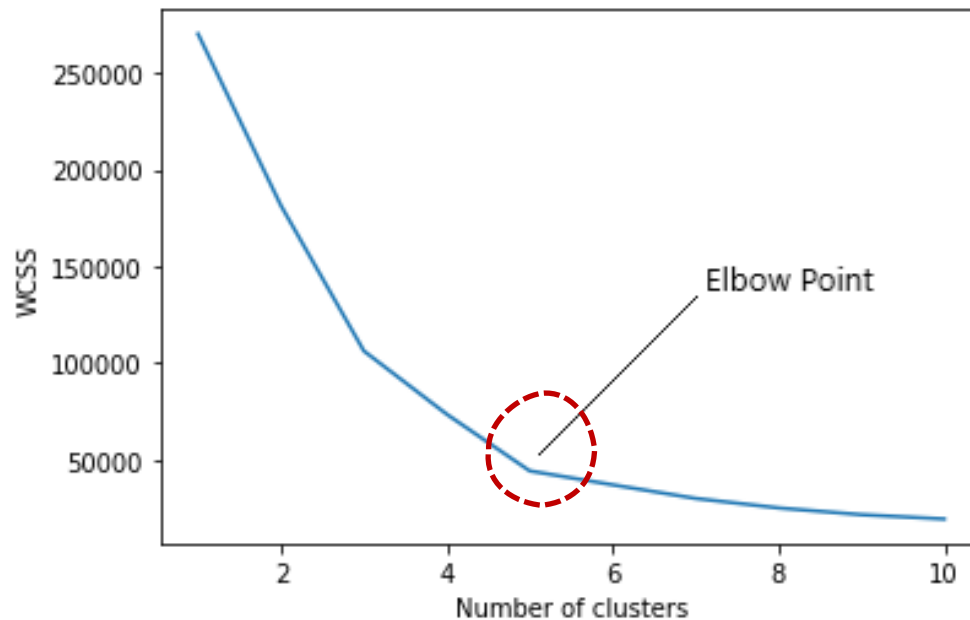
즉, 클러스터 내 분산이 최소가 되게끔 하는,  
클러스터 내 중심점과 객체들 간의 거리가 최소가 되게끔 하는 중심점 선택



클러스터 개수 증가할수록 RSS가 감소하는 문제 발생!

## Elbow Point Method

Elbow Point 방법



RSS가 급격하게 감소하는 지점

해당 지점에서의 클러스터 개수를 최종 클러스터 개수로 선택



## 클러스터링 방법

비계층적 클러스터링 VS 계층적 클러스터링

비계층적 클러스터링

K-means  
K-medoids  
DBSCAN

계층적 클러스터링

Hierarchical Clustering

## Non-Hierarchical clustering

### K-means Clustering

데이터들을 묶어 군집을 생성하는 것을 목표로 함

각 클러스터는 한 개의 중심점을 가짐



클러스터 내에서 중심점과의 거리의 분산은 최소화,  
클러스터 간의 거리의 분산은 최대화하는 방향으로 동작



이때 클러스터의 개수인  $k$ 를 사전에 정의 내려야 함

Hyperparameter

## Non-Hierarchical clustering?

### K-means Clustering

왜 이름이 K-Means일까?  
데이터들을 몇 개 군집을 형성하든 것은 우리가 정해 줌

↓  
각 클러스터는 하나의 중심점을 가짐

클러스터의 위치를 결정하기 위해,  
클러스터 간의 거리의 분산을 최소화하는 방향으로 동작  
데이터의 **평균값**을 이용!

↓  
이때 클러스터의 개수인  $k$ 를 사전에 정의 내려야 함

Hyperparameter

## Non-Hierarchical clustering

### K-means Clustering

$$WCSS = \sum_{k=1}^K n_k \sum_{C(i)=k} \|X_i - \bar{X}_k\|^2$$



클러스터 내 분산

$n_k = \sum_{i=1}^N I(C(i) = k)$  : k번째 클러스터의 point 개수

$\bar{X}_{jk} = \frac{1}{n_k} \sum_{C(i)=k} X_{ij}$  : k번째 클러스터의 j번째 속성의 평균

$$\bar{X}_k = (X_{1k}, X_{2k}, \dots, X_{pk})$$

**WCSS**

데이터에서 클러스터의 중심점과의 거리의 합



## Non-Hierarchical clustering

### K-means Clustering

$$WCSS = \sum_{k=1}^K n_k \sum_{C(i)=k} \|X_i - \bar{X}_k\|^2$$



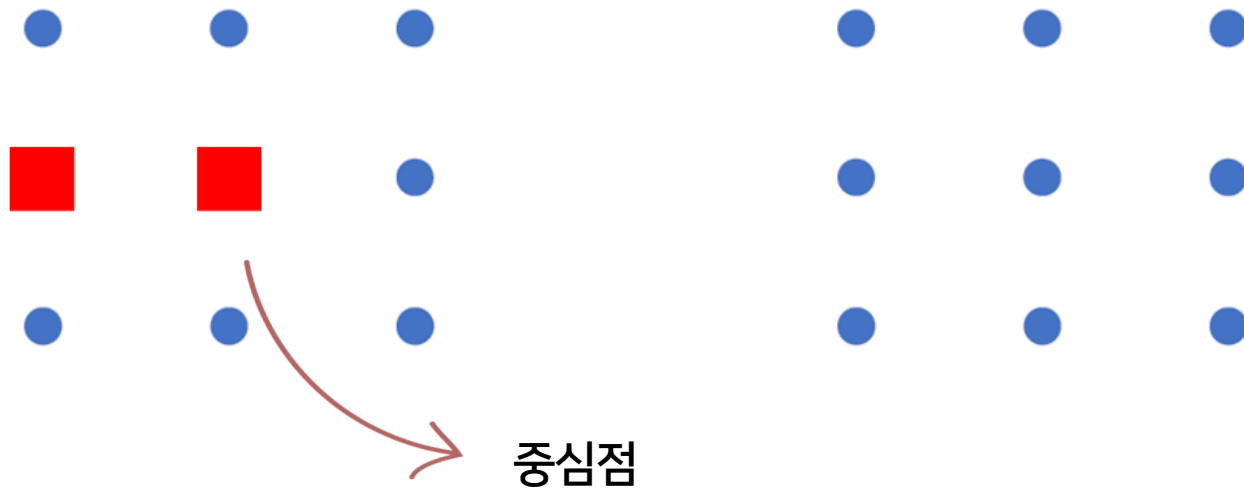
클러스터 내 분산

클러스터 내 분산을 최소화하는 것이 목표



## Non-Hierarchical clustering

K-Means Clustering의 학습과정

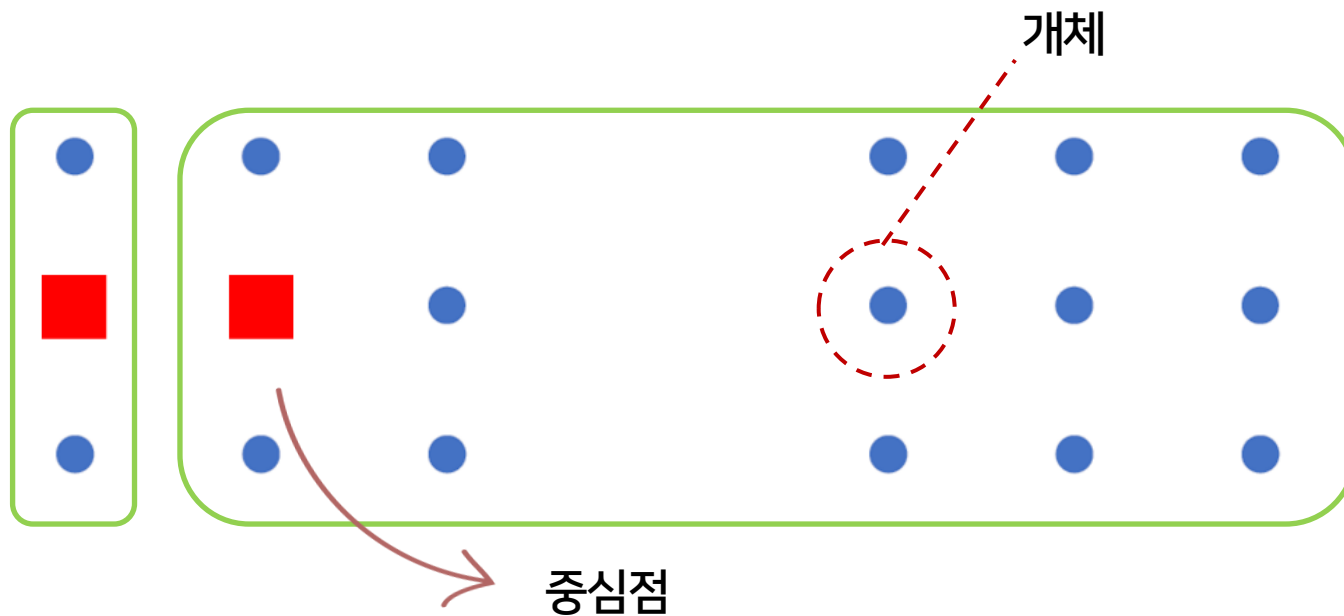


군집 수를 2로 정하고, 군집의 중심을 랜덤 초기화



## Non-Hierarchical clustering

K-Means Clustering의 학습과정

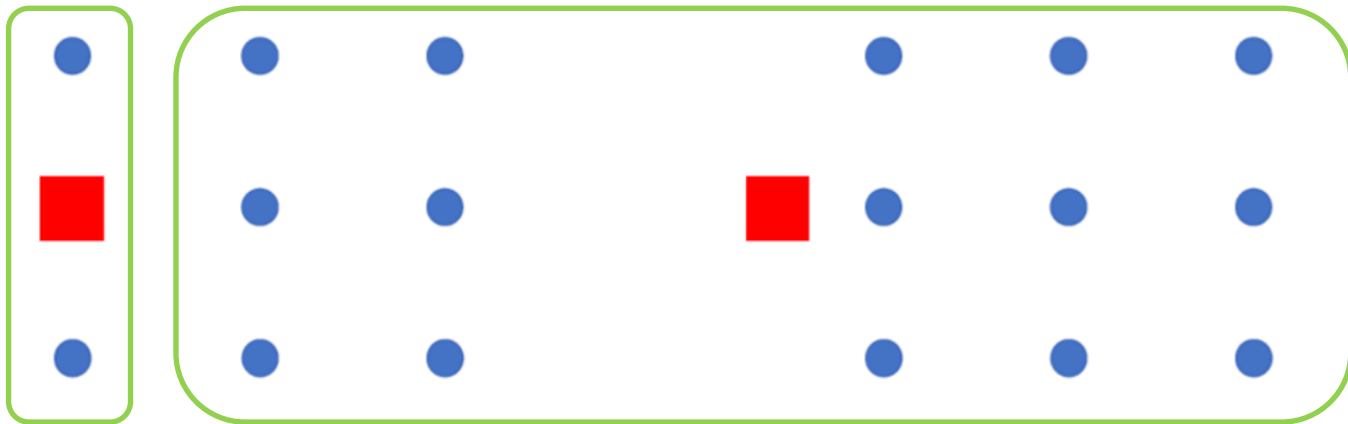


모든 개체들을 가장 가까운 중심에 군집(녹색 박스)으로 할당

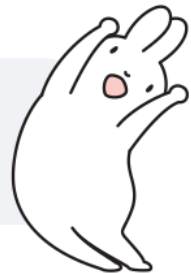


## Non-Hierarchical clustering

K-Means Clustering의 학습과정

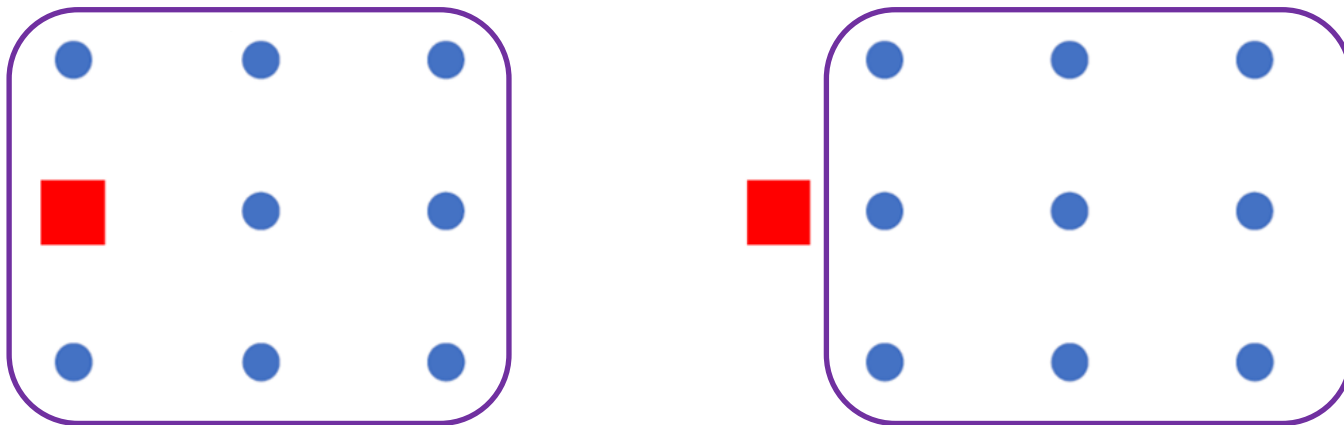


중심을 군집 경계에 맞게 업데이트



## Non-Hierarchical clustering

K-Means Clustering의 학습과정

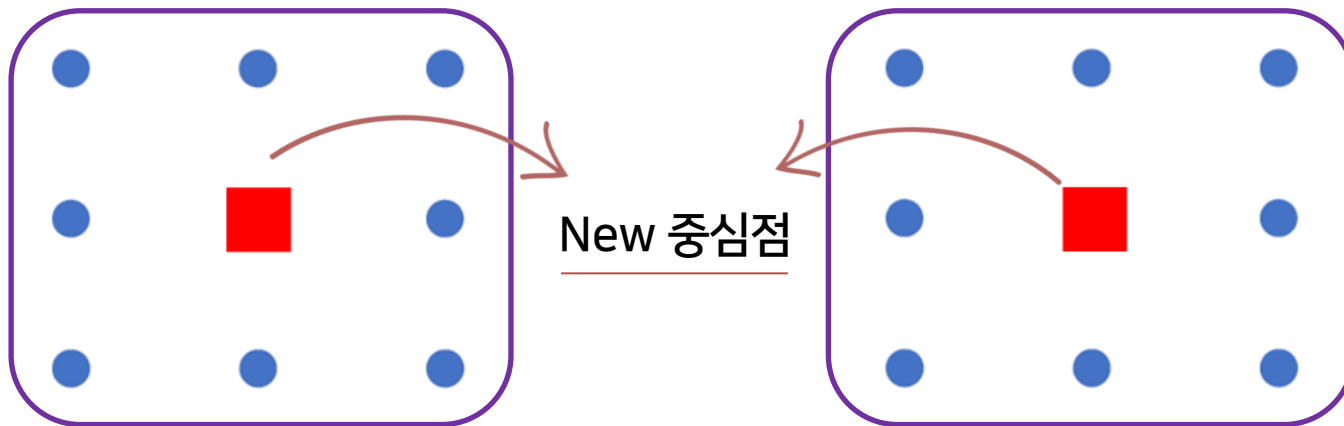


다시 모든 개체들을 가장 가까운 중심에 군집으로 할당



## Non-Hierarchical clustering

K-Means Clustering의 학습과정

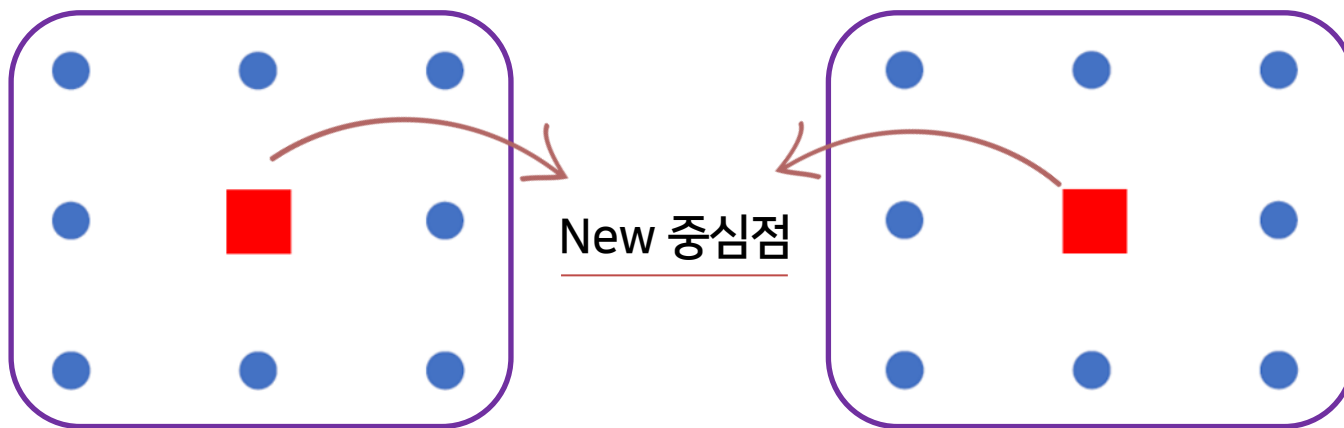


다시 중심을 군집 경계에 맞게 업데이트



## Non-Hierarchical clustering

K-Means Clustering의 학습과정



스텝을 반복 적용해도 결과가 바뀌지 않거나  
사용자가 정한 반복 수를 채우게 되면 종료



## Non-Hierarchical clustering

### K-Means Clustering의 학습과정

수치형 변수에만 적용 가능

데이터 간의 유클리드 거리 계산 필요 → 범주형 변수 적용 불가

Global Optimum이 아닌 Local Optima에 빠질 가능성 존재

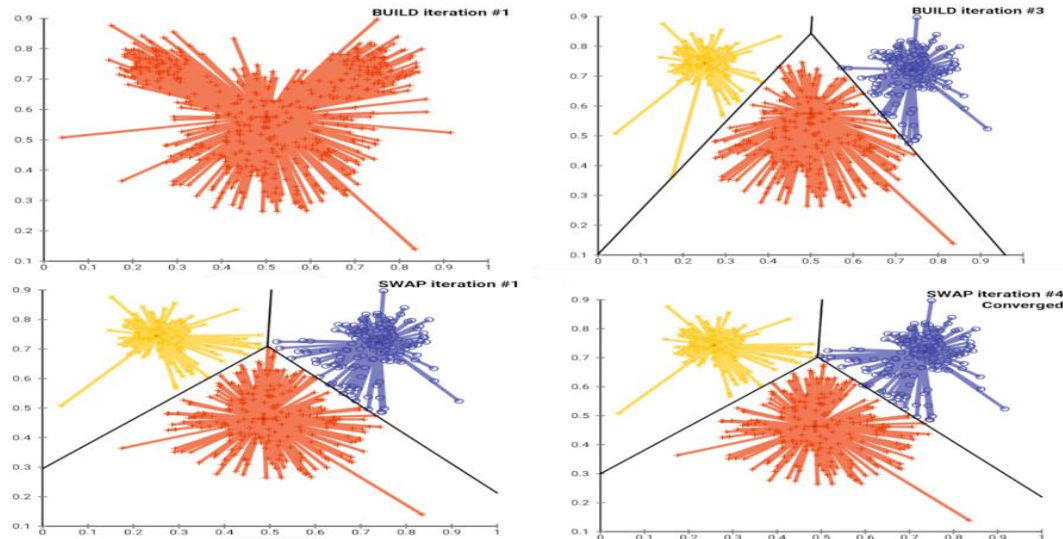




## Non-Hierarchical clustering

K-Medoids Clustering: PAM(Partitioning Around Medoids)

데이터의 중앙값으로 중심점을 이동



중앙값은 평균보다 이상치로부터 강건하며 계산이 빠르다는 장점 존재

## Density-Based Clustering: DBSCAN

중심점(Centroid) 기반

중심점과의 거리를  
바탕으로 클러스터링 진행

K-Means

K-Medoids

V/S

밀도(Density) 기반

데이터가 얼마나  
뭉쳐 있는가(밀도)를  
고려하여 클러스터링 진행

DBSCAN

## Density-Based Clustering: DBSCAN

중심점(Centroid) 기반

중심점과의 거리를  
바탕으로 클러스터링 진행

V/S

밀도(Density) 기반

데이터가 얼마나  
뭉쳐 있는가(밀도)를  
고려하여 클러스터링 진행

K-Means

K-Medoids

DBSCAN

## Density-Based Clustering: DBSCAN

중심점(Centroid) 기반

중심점과의 거리를  
바탕으로 클러스터링 진행

K-Means  
K-Medoids

V/S

밀도(Density) 기반

데이터가 얼마나  
**뭉쳐 있는가(밀도)**를  
고려하여 클러스터링 진행

DBSCAN

## Density-Based Clustering: DBSCAN



중심점(Centroid)

예시를 통해서 비교해보자!

밀도(Density) 기반

중심점과의 거리를  
기준으로 클러스터링 진행

- K-Means
- K-Medoids

V/S

데이터가 얼마나  
뭉쳐 있는가(밀도)를  
고려하여 클러스터링 진행

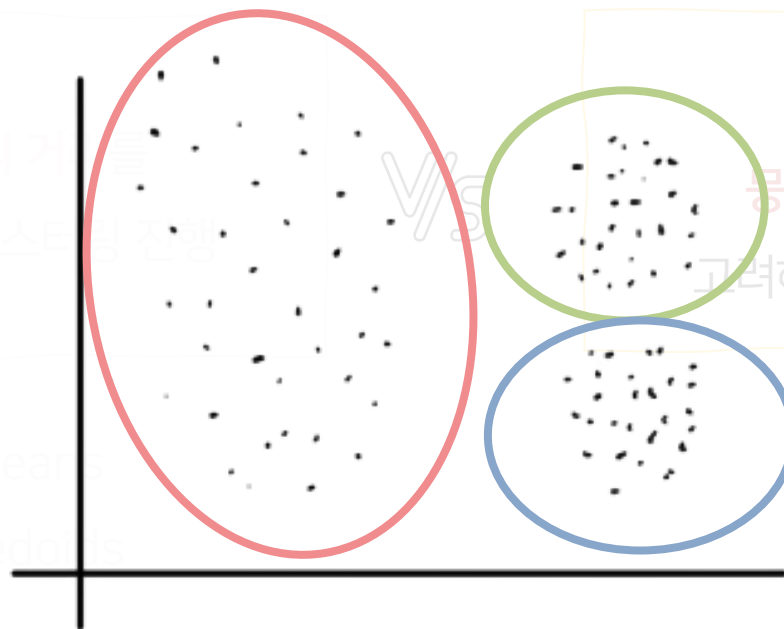
- DBSCAN

## Density-Based Clustering: DBSCAN



예시를 통해서 비교해보자!

밀도(Density) 기반



데이터가 얼마나

밀집 있는가(밀도)를

고려하여 클러스터링 진행

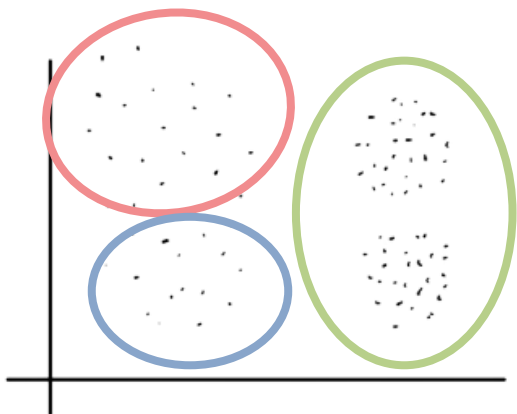
일반적으로 위와 같이 분류될 것으로 예상함



예상이 됩니다 예상이

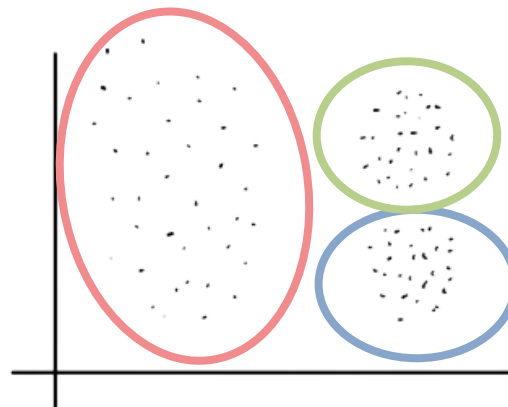
## Density-Based Clustering: DBSCAN


중심점(Centroid) 기반



중심점과의 거리를 바탕으로  
클러스터링 진행  
→ 가까이 있는 데이터끼리 묶음

밀도(Density) 기반







데이터의 **뭉쳐 있는가(밀도)**를  
고려하여 클러스터링 진행  
→ DBSCAN이 더 효과적 

## Density-Based Clustering: DBSCAN

### DBSCAN의 장점

장점



-  구형이 아닌 임의의 모양의 클러스터링을 찾아낼 수 있음
-  이상치나 일반적인 패턴에서 벗어나는 데이터들은 **noise**로 구분되므로 군집에 할당하지 않아도 되어 **이상치 탐지**에 사용 가능
-  군집의 랜덤성이 상당히 작음
-  군집의 개수  $k$ 를 정해주지 않아도 됨



## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



$\epsilon$ -Neighborhood of a point

점  $p$ 의  $\epsilon$ -neighborhood는  
 $p$ 와의 거리가  $\epsilon$  보다 작거나 같은 점  $q$ 들의 집합으로 정의

$$N_{\epsilon}(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$$

“  $\epsilon$  (epsilon) ”

점  $p$ 로부터 떨어진 거리

## Density-Based Clustering: DBSCAN

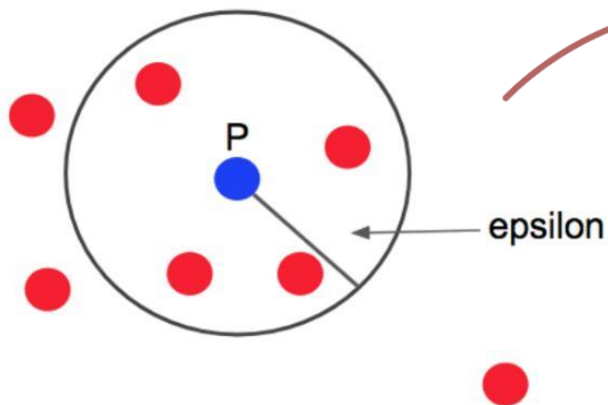
DBSCAN 용어 정리



$\epsilon$ -Neighborhood of a point

점  $p$ 의  $\epsilon$ -neighborhood는  
 $p$ 와의 거리가  $\epsilon$  보다 작거나 같은 점  $q$ 들의 집합으로 정의

$$N_{\epsilon}(p) = \{q \in D \mid dist(p, q) \leq \epsilon\}$$



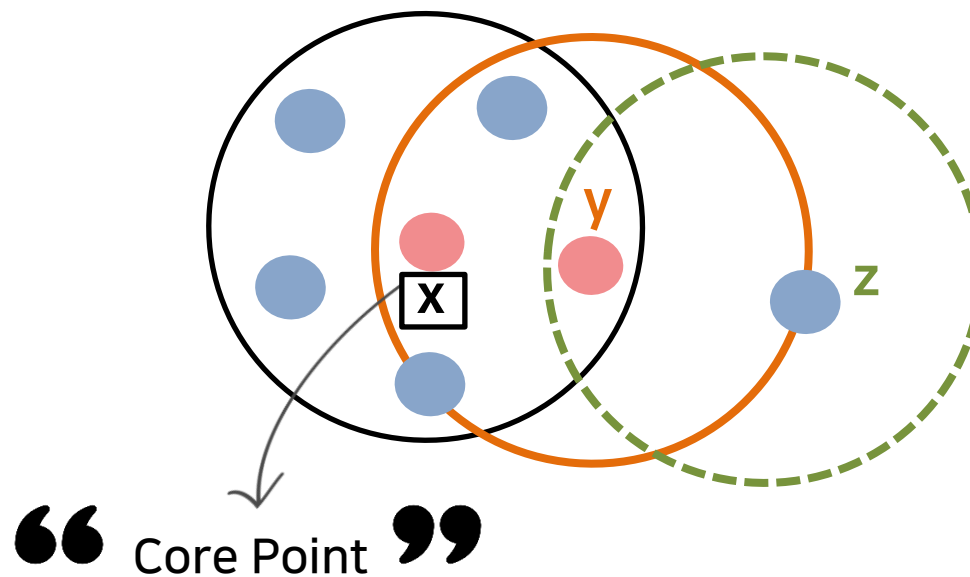
“minPts”

점  $p$ 가 하나의 군집을 이루게 되려면  
 $p$ 중심으로부터  $\epsilon$ -neighborhood  $q$ 들이

**최소 minPts 이상**은 있어야 함

## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



\* MinPts=6

한 점의  $\epsilon$  반경 내에

**minPts 이상**의 개체가 포함된 점

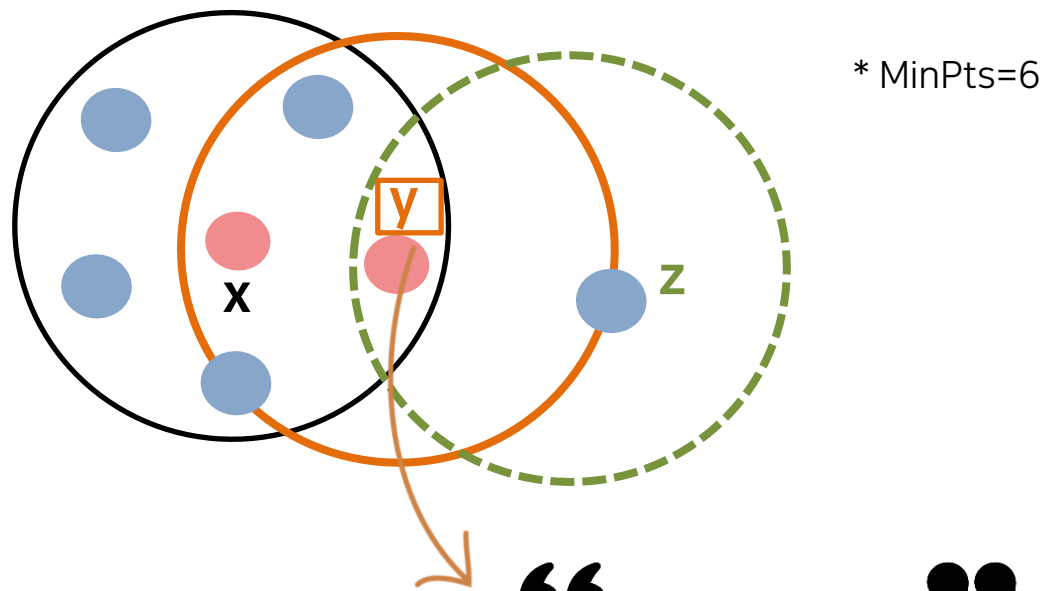
→ 해당 점을 중심으로 군집 형성 가능



데마팀의 Core 김수빈 팀장

## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



Core 결의 Border Points

“ Border Point ”

한 점의  $\epsilon$  반경 내에

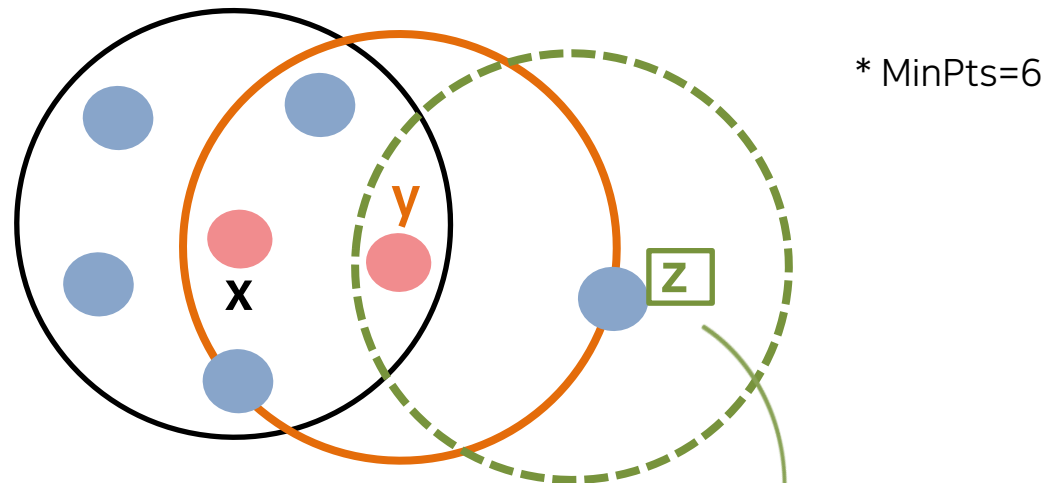
**minPts보다 적은 수**의 개체를 포함하지만

그 중 적어도 하나가 Core point인 경우



# Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



“ Noise Point ”

Core도 Border도 아닌 point

→ minPts 이하의 개체, 주변 점에 Core point 없음

PSAT  
파이팅!

멀리서 지켜보는 Noise Point

## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



Directly Density-Reachable

점  $p$ 가 점  $q$ 로부터 밀도 관점에서 직접적으로 연결 가능



2가지 조건 만족해야 함!

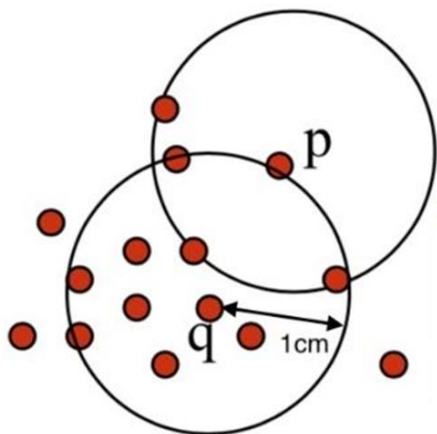
## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



Directly Density-Reachable

점  $p$ 가 점  $q$ 로부터 밀도 관점에서 직접적으로 연결 가능



MinPts = 5

Eps = 1 cm



Reachability  $p \in N_\epsilon(q)$

$p$ 는  $q$ 의 epsilon neighborhood에 속해야 함



Core Point Condition  $N_\epsilon(q) \geq \text{MinPts}$

$q$ 의 neighborhood의 개수  $\geq$  사용자가 정한 MinPts

( $q$ 는 core point)

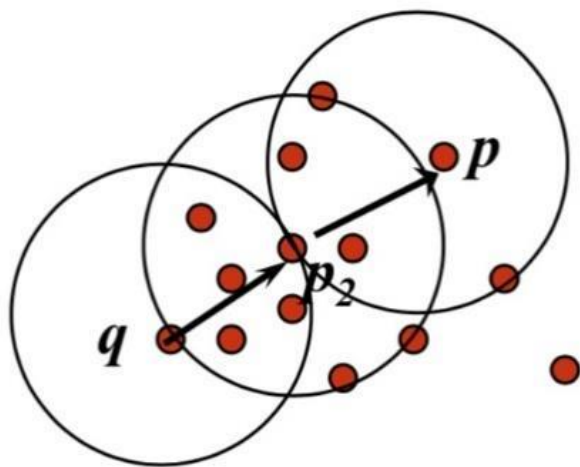
## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



Density-Reachable

점  $p$ 가 점  $q$ 로부터 밀도 기반 도달 가능한 관계



점  $p$ 가 점  $q$ 의  $\epsilon$  반경 안에 위치하지 못하더라도  
점  $p$ 와  $q$  사이에 점  $p_1, p_2, \dots, p_n$  존재하고,  
모든 점  $p_{i+1}$  이  $p_i$ 로부터  
Directly Density-Reachable





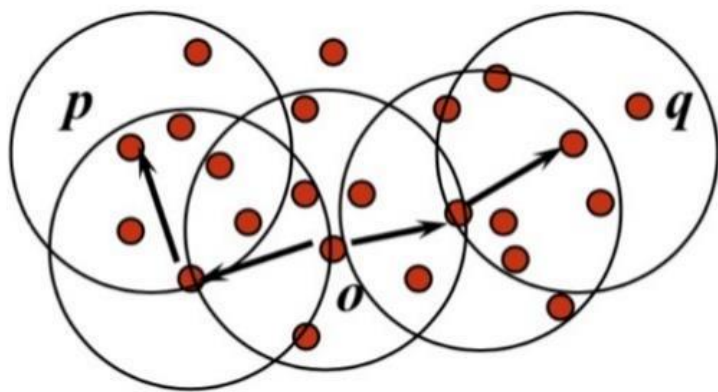
## Density-Based Clustering: DBSCAN

DBSCAN 용어 정리



Density-Connected

점  $p$ 가 점  $q$ 와 연결된 관계에 있다



두 점  $p, q$ 가 모두  
어떤 점  $o$ 로부터 반경 내  
MinPts 조건 하에  
Density-Reachable한 경우



## Density-Based Clustering: DBSCAN

DBSCAN 학습과정



임의의 데이터 포인트 선택



$\epsilon$  반경 내 minPts 개수 이상의 데이터가 있다면  
core point, 없다면 border point로 할당



$\epsilon$  반경 안에 있는 core point들을 서로 연결하여 군집 형성  
border point들을 어느 하나의 군집에 할당

## Density-Based Clustering: DBSCAN

DBSCAN 학습과정



임의의 데이터 포인트 선택



∈ 반경 내 minPts 개수 이상의 데이터가 있다면  
core point, 없다면 border point로 할당



∈ 반경 안에 있는 core point들을 서로 연결하여 군집 형성  
border point들을 어느 하나의 군집에 할당



학습이 끝난 후에도 군집에 속하지 않은 포인트 → Noise



## Density-Based Clustering: DBSCAN

DBSCAN 학습



### DBSCAN의 한계점



임의의 데이터 포인트 선택  
적절한  $\epsilon$ 과 minPts의 값을 알 수 없어



**Heuristic**하게 결정해야 하며,  
데이터셋이 바뀔때마다  $\epsilon$ 과 minPts 값 **달라질 수 있음**  
core point, 없다면 border point로 할당



$\epsilon$  반경 안에 있는 코어점들을 서로 연결하여 군집 형성  
border point들을 어느 하나의 군집에 할당



학습이 끝난 후에도 군집에 속하지 않은 포인트 → Noise

## Density-Based Clustering: DBSCAN

DBSCAN 학습



### DBSCAN의 한계점



임의의 데이터 포인트 선택  
적절한  $\epsilon$ 과 minPts의 값을 알 수 없어



**Heuristic**하게 결정해야 하며,  
데이터셋이 바뀔때 따라  $\epsilon$ 과 minPts 값 **달라질 수 있음**  
core point, 없다면 border point로 할당



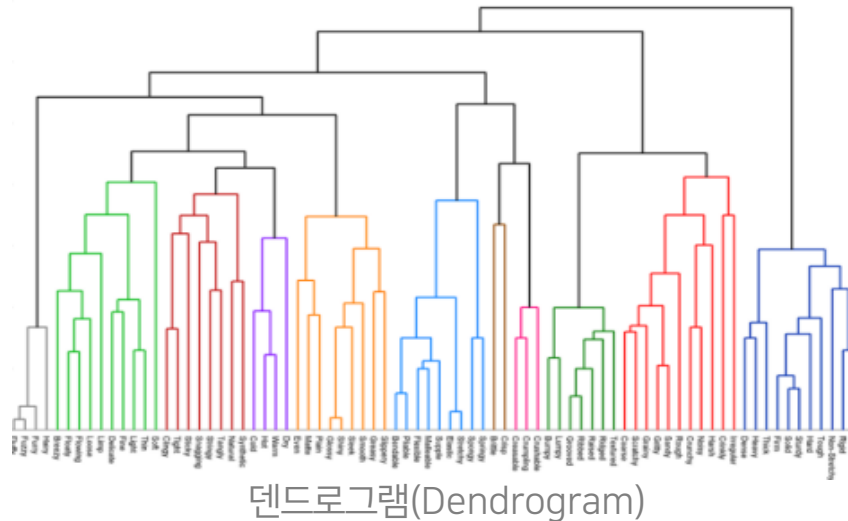
$\epsilon$  반경 안에 있는 코어점들을 서로 연결하여 군집 형성  
border point들을 어느 하나의 군집에 할당



학습이 끝난 후에도 군집에 속하지 않은 포인트 → Noise  
**Hierarchical DBSCAN (HDBSCAN)**

# Hierarchical Clustering

## 계층적 클러스터링이란?



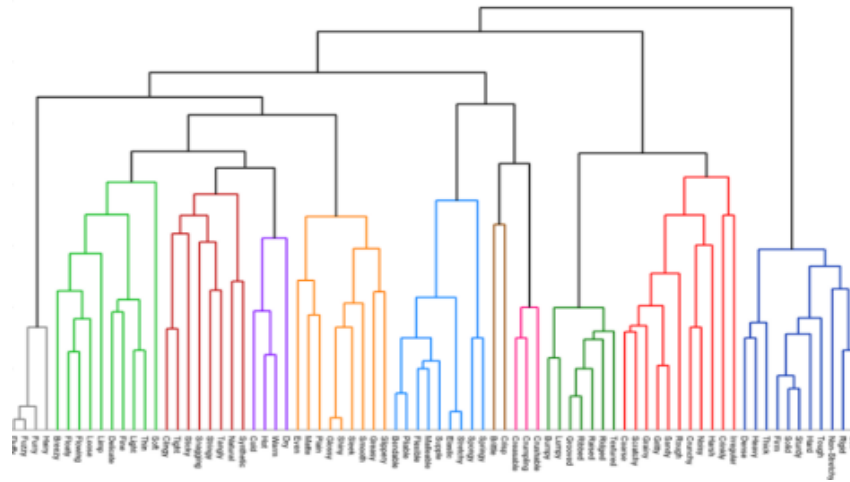
## ? 계층적 클러스터링

트리 모형을 이용해서 개별 개체들을 **순차적이고 계층적으로** 유사한 개체 혹은 그룹과 함께 클러스터를 만들어주는 알고리즘

클러스터의 개수를 사전에 정하지 않고도 학습이 수행 가능하다는 장점!

# Hierarchical Clustering

## 계층적 클러스터링이란?

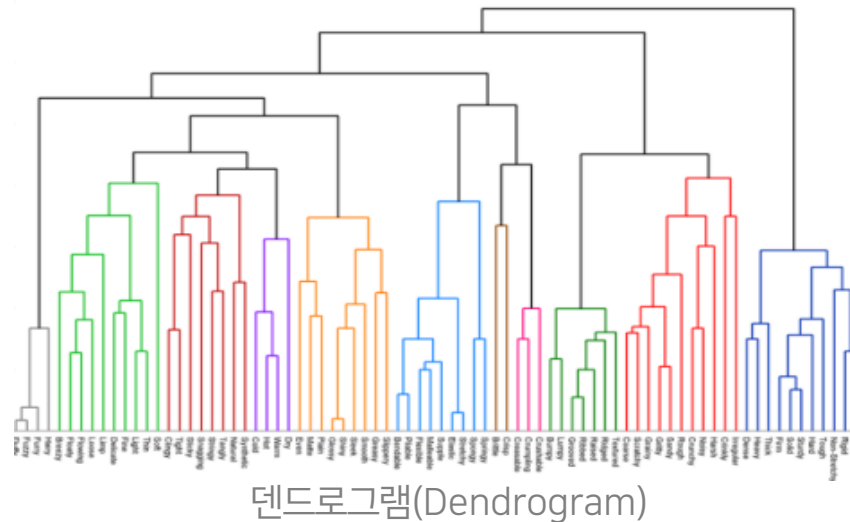


## 덴드로그램(Dendrogram)

트리 모형을 이용해서 개별 개체들을 **순차적이고 계층적으로**  
 계층적 클러스터링에서는  
 유사한 개체 혹은 그룹과 함께 클러스터를 만들어주는 알고리즘  
 트리 형태의 구조인 **덴드로그램**을 사용

# Hierarchical Clustering

## 계층적 클러스터링이란?



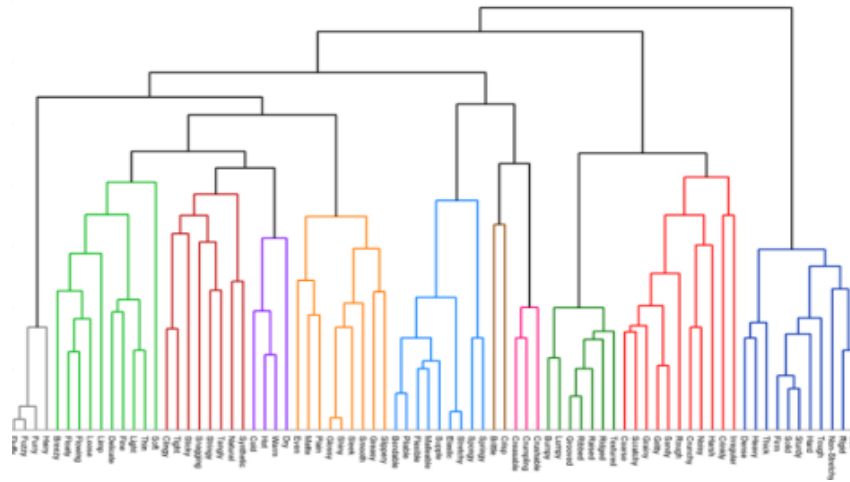
트리 모형을 이용해서 개별 개체들을 순차적이고 계층적으로  
유사한 개체 혹은 그룹과 함께 클러스터를 만들어주는 알고리즘

**덴드로그램을 그리는 방법??**



# Hierarchical Clustering

## 계층적 클러스터링이란?



## 덴드로그램(Dendrogram)

트리 모형을 이용해서 개별 개체들을 순차적이고 계층적으로  
유사한 개체 혹은 그룹과 함께 클러스터화시켜주는 알고리즘

덴드로그램을 그리는 방법??

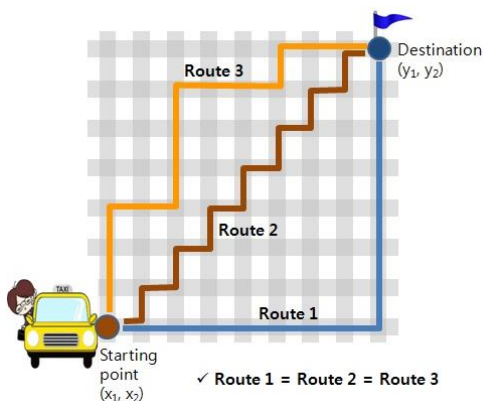
모든 개체들 간의 거리나 유사도가 이미 계산되어 있어야 함

## Hierarchical Clustering

거리 계산하는 다양한 방법

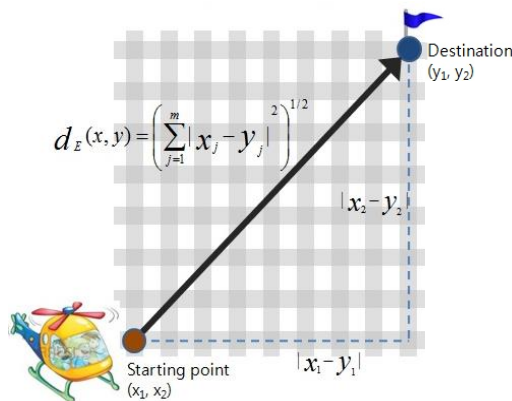
### 맨하탄 거리(L1)

좌표 축 방향으로  
이동할 때 계산되는 거리



### 유클리드 거리 (L2)

두 관측치 사이의  
직선 최단거리



### 마할라노비스 거리

변수 내 분산, 공분산을  
모두 반영해 계산한 거리

$$d_{Mahalanobis}(X, Y)$$

$$= \sqrt{(\bar{X} - \bar{Y})^T \Sigma^{-1} (\bar{X} - \bar{Y})}$$

where  $\Sigma^{-1}$  is the inverse  
of covariance matrix

제가 마할라노비스입니다ㅎㅎ



## Hierarchical Clustering

거리 계산하는 다양한 방법

맨하탄 거리(L1)

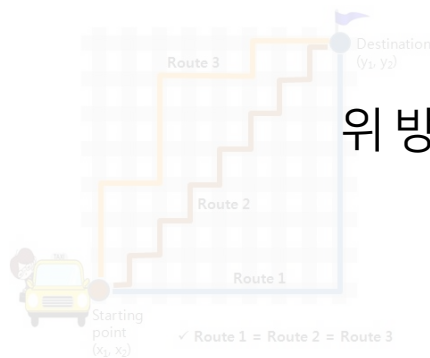
좌표 축 방향으로  
이동할 때 계산되는 거리

유클리드 거리 (L2)

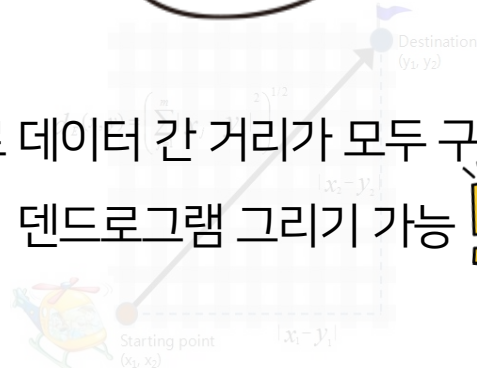


마할라노비스 거리

변수 내 분산, 공분산을  
모두 반영해 계산한 거리



위 방법으로 데이터 간 거리가 모두 구해졌으므로  
덴드로그램 그리기 가능!



$d_{Mahalanobis}(X,Y)$

$$= \sqrt{(\vec{X} - \vec{Y})^T \Sigma^{-1} (\vec{X} - \vec{Y})}$$

where  $\Sigma^{-1}$  is the inverse  
of covariance matrix

## Hierarchical Clustering

덴드로그램

	A	B	C	D
A		20	7	2
B	20		10	25
C	7	10		3
D	2	25	3	



거리나 유사도 값을 바탕으로 거리행렬식 생성

## Hierarchical Clustering

덴드로그램

	A	B	C	D
A		20	7	2
B	20		10	25
C	7	10		3
D	2	25	3	

A와 D로 묶기!



서로 가장 가까운 관측치 찾아 묶기

## Hierarchical Clustering

덴드로그램

A와 D로 묶음!



	AD	B	C
AD		?	?
B	?		?
C	?	?	



한번 묶은 후, 각 군집 간의 거리를 결정해야 함



## Hierarchical Clustering

AD와 B, C사이의 거리는 어떻게 결정해야 할까요?

최소기준

묶이기 전 각각 개체와  
나머지 개체의 거리 중 가장  
짧은 거리로 대체



A와 B 사이거리 = 20  
D와 B 사이거리 = 25

최대기준

묶이기 전 각각 개체와  
나머지 개체의 거리 중  
가장 긴 거리로 대체



A와 B 사이거리 = 20  
D와 B 사이거리 = 25

평균기준

최소 기준으로 계산된 거리와  
최대 기준으로 계산된 거리의  
평균으로 대체



$20(\text{A와 B거리}) +$   
 $25(\text{D와 B거리}) / 2 = 22.5$

한번 묶은 후, 각 군집 간의 거리를 결정해야 함

## Hierarchical Clustering

덴드로그램

	AD	B	C
AD		20	3
B	20		10
C	3	10	

최소 기준 적용!



최소 기준, 최대 기준, 평균 기준 중 하나를 적용하여 거리 행렬 채움



## Hierarchical Clustering

덴드로그램

	A	B	C
A		ADC	B
B	ADC		10
C	B	10	

최소 기준 적용!



과정을 반복하여 거리 행렬 완성!

최소 기준, 최대 기준, 평균 기준 중 하나를 적용하여 거리 행





## Hierarchical Clustering

덴드로그램

### 계층적 클러스터링의 단점

- ✓ 계산의 복잡성
- ✓ 대용량 데이터의 경우,  
많은 연산 시간과 컴퓨팅 파워 소모

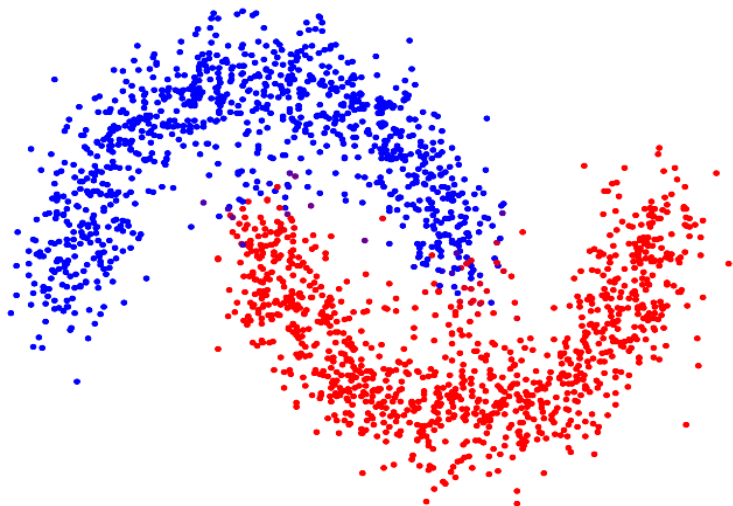


최소 기준, 최대 기준, 평균 기준 중 하나를 적용하여 거리 행렬 완성



## Other Clustering

### Spectral Clustering



#### 장점

이중 나선 형태에서 효과적

#### 단점

긴 연산 시간  
최적의 클러스터 개수를  
결정하기 어려움



## Other Clustering

Spectral Clustering

## 정리해봅시다

데이터 분포에 따라 이종 나선 형태에서 효과적

이상치 존재

치우친 분포

패턴이 발견됨

K-medoids

단점

DBSCAN / Spectral

최적의 클러스터 개수를

결정하기 어려움



## Other Clustering

Spectral Clustering

## 정리해봅시다

데이터 분포에 따라 이종 나선 형태에서 효과적

이상치 존재

치우친 분포

패턴이 발견됨

K-medoids

단점

DBSCAN / Spectral

최적의 클러스터 개수를

결정하기 어려움

각 클러스터링 기법의 특징을 이해하고 상황에 맞게 적용하는 것이 매우 중요



# 2

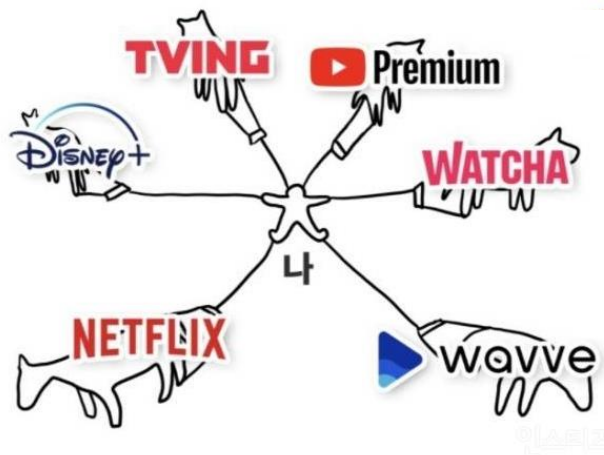
추천 시스템

## 추천 시스템



추천 시스템

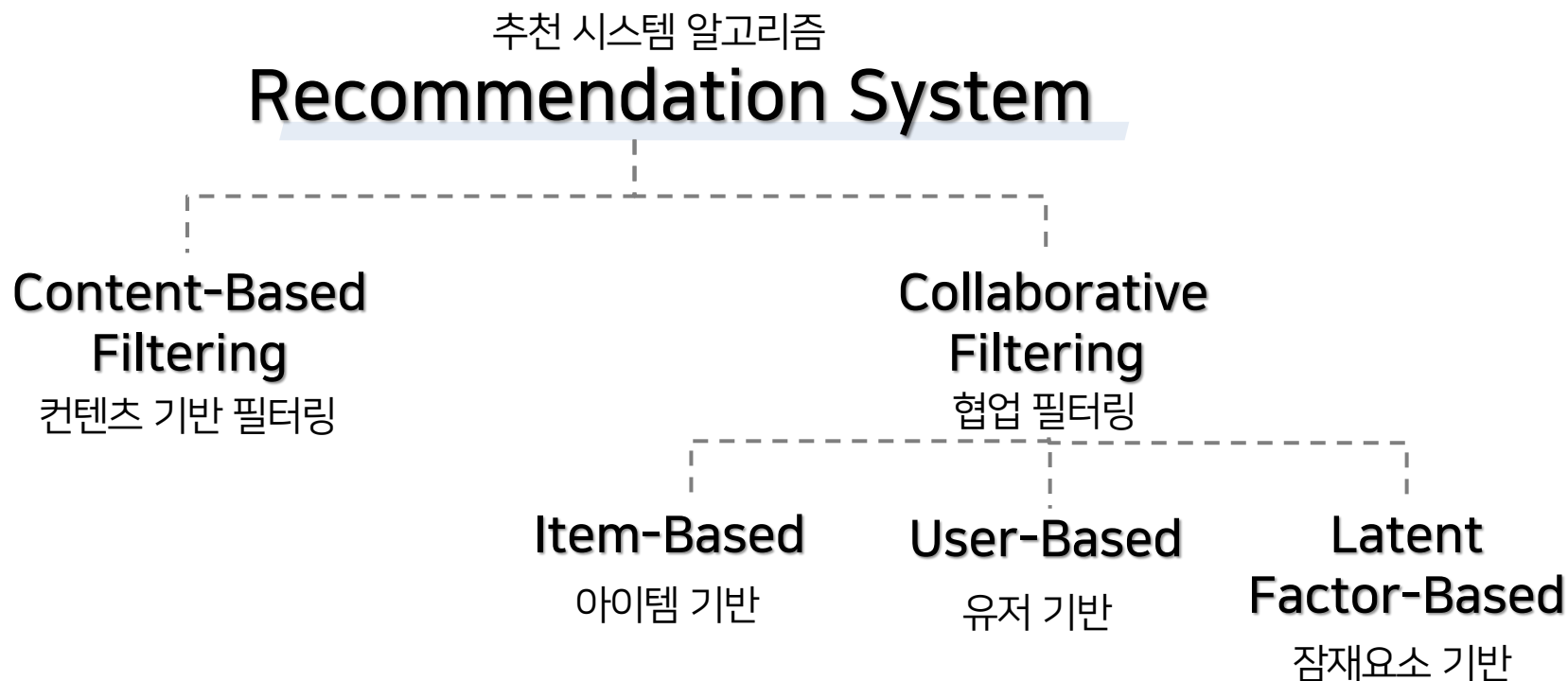
넷플릭스나 유튜브 등 OTT 플랫폼들이  
개인 취향에 맞는 콘텐츠를 추천해주는 시스템



정보량과 관련 플랫폼이 증가하며 개인의 취향에 맞는 콘텐츠를 추천할 필요성 ↗

## 추천 시스템

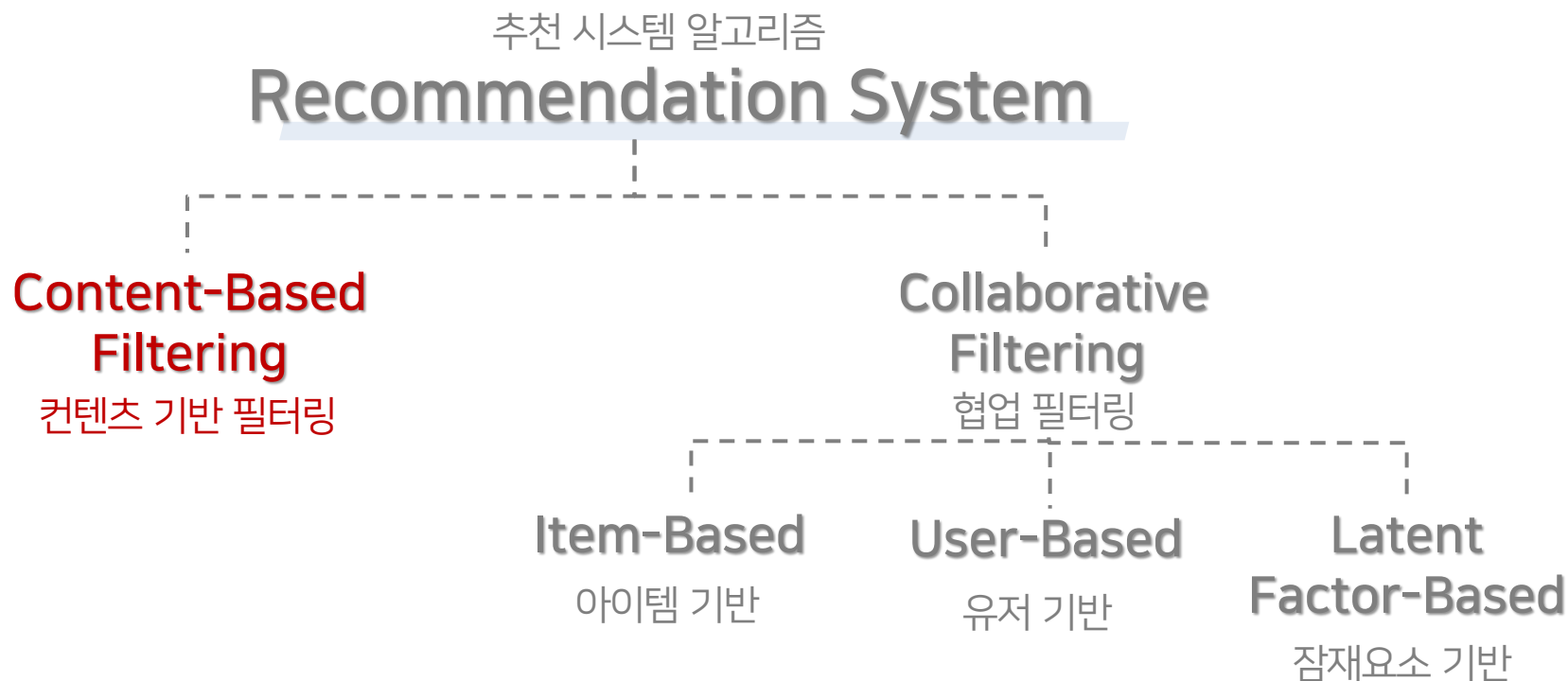
추천 시스템에 사용되는 알고리즘





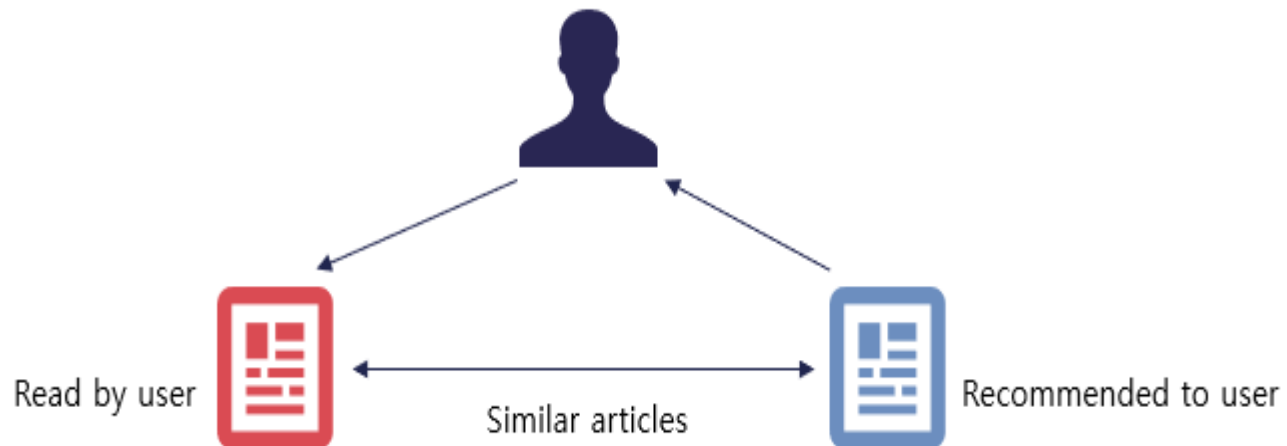
## 추천 시스템

추천 시스템에 사용되는 알고리즘



## 컨텐츠 기반 추천

컨텐츠 기반 추천이란?



해당 콘텐츠에 대한 정보만을 이용해 추천 실시  
과거에 소비했던 콘텐츠 특성을 분석하고 유사한 특성을 지닌 콘텐츠를 추천

## 컨텐츠 기반 추천

컨텐츠 기반 추천이란?

작가 기반 추천



<더 글로리>



<도깨비>



<미스터선샤인>



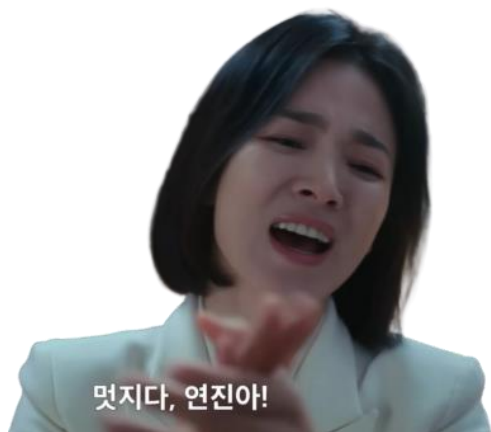
사용자의 드라마 감상 데이터를 바탕으로 다른 드라마를 추천한다 가정  
기존의 사용자가 들었던 드라마의 **메타데이터**를 사용

Ex) 장르, 배우, 줄거리, 감독, 작가...

## 컨텐츠 기반 추천

컨텐츠 기반 추천이란?

장르 기반 추천



<더 글로리>



<모범택시>



<부부의 세계>



사용자의 드라마 감상 데이터를 바탕으로 다른 드라마를 추천한다 가정  
기존의 사용자가 들었던 드라마의 **메타데이터**를 사용

Ex) 장르, 배우, 줄거리, 감독, 작가...

## 컨텐츠 기반 추천

컨텐츠 기반 추천이란?



장르 기반 추천

사용자가 소비한 콘텐츠와 추천하려는 콘텐츠가 얼마나 유사한지가 중요  
드라마의 예시에서 내용이 비슷한 드라마를 추천한다고 가정



드라마의 줄거리는 **비정형**인 텍스트 데이터이므로 유사도 측정에 어려움 있음

<더글로리>

<모범택시>

<브브의 세계>



사용자의 영화 감상 데이터를 바탕으로 다른 영화를 추천한다 가정  
기존의 사용자가 들었던 음악의 메타데이터를 사용

(Ex) 아티스트, 프로듀서, 장르,...



## 컨텐츠 기반 추천

컨텐츠 기반 추천이란?



장르 기반 추천

사용자가 소비한 콘텐츠와 추천하려는 콘텐츠가 얼마나 유사한지가 중요  
드라마의 예시에서 내용이 비슷한 드라마를 추천한다고 가정



드라마의 줄거리는 **비정형**인 텍스트 데이터이므로 유사도 측정에 어려움 있음

<더글로리>

<모범택시>

<브브의 세계>

TF-IDF



(Term Frequency-Inverse Document Frequency)

자연어 데이터에서 유사도 추출하는 객관적 지표



## 컨텐츠 기반 추천

### TF-IDF

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

$n_D$  : 전체 문서 수

$n_t$  : 단어 t가 나온 문서 수

TF (Term Frequency) : 단어 t가 하나의 문서에서 나온 빈도수

IDF (Inverse Document Frequency) : 전체 문서 중 단어 t가 나온 문서 수의 역수

전체 문서 중 단어t가 나온 문서 수의 역수에 Log를 취한 값을  
곱해줌으로 전체적으로 많이 등장하는 단어에 **패널티** 부여

## 컨텐츠 기반 추천

### TF-IDF

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

$n_D$  : 전체 문서 수

$n_t$  : 단어 t가 나온 문서 수

TF (Term Frequency) : 단어 t가 하나의 문서에서 나온 빈도수

IDF (Inverse Document Frequency) : 전체 문서 중 단어 t가 나온 문서 수의 역수

Log를 취해 전체 문서수가 많을 때 TF-IDF값이 너무 커지는 것 방지



## 컨텐츠 기반 추천

### TF-IDF

$$TF - IDF = TF \times \log \frac{n_D}{1 + n_t}$$

$n_D$  : 전체 문서 수

$n_t$  : 단어 t가 나온 문서 수

TF (Term Frequency) : 단어 t가 하나의 문서에서 나온 빈도수

IDF (Inverse Document Frequency) : 전체 문서 중 단어 t가 나온 문서 수의 역수

분모는  $1 + n_t$  을 취해 어떤 단어가 모든 문서에 들어가서  
 $\log IDF = \log 1 = 0$ 으로 계산되어 TF-IDF가 0이 되는 것을 방지

## 컨텐츠 기반 추천

TF-IDF

?

단어가 드라마를 대표하는 단어가 되려면?

한 문서에서 자주 등장 - High TF

다른 문서에서는 적게 등장 - Low IDF

높은 TF-IDF값을 가진 단어가 해당 문서의 키워드

## 컨텐츠 기반 추천

TF-IDF

?

단어가 드라마를 대표하는 단어가 되려면?



환영해, 연진아

더 글로리의 경우,  
"복수"라는 단어가  
드라마의 키워드가 될 것!

VENGEANCE!

## 컨텐츠 기반 추천



TD-IDF

?

단어가 문서(가사)를 대표하는 단어가 되려면?

이런 방식으로 문서에서 **중요한 단어(Feature)**를  
추출하여 이를 바탕으로 다른 콘텐츠와 추천 진행

한 문서(가사)에서 자주 등장 - High TF

다른 문서에서는 적게 등장 - Low IDF

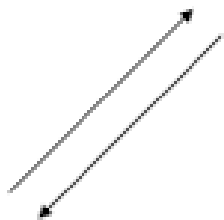
높은 TF-IDF값을 가진 단어가 해당 문서(노래)의 키워드



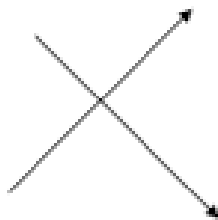
## 컨텐츠 기반 추천

코사인 유사도

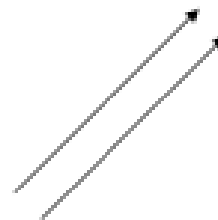
$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$



코사인 유사도 : -1



코사인 유사도 : 0



코사인 유사도 : 1

대표 단어들을 Word2Vec 등의 방법을 통해 수치형 벡터로 변환(임베딩)



벡터간 코사인 유사도를 통해 계산 가능

## 컨텐츠 기반 추천

코사인 유사도

$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

코사인 유사도 : -1

코사인 유사도 : 0

코사인 유사도 : 1

Word2Vec 등 더 많은 자연어 처리 기술은  
딥러닝 클린업 많은 관심 부탁드립니다

대표 단어들을 Word2Vec 등의 방법을 통해 수치형 벡터로 변환(임베딩)

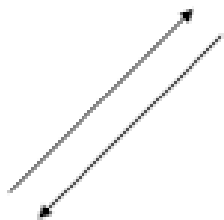


벡터간 코사인 유사도를 통해 계산 가능

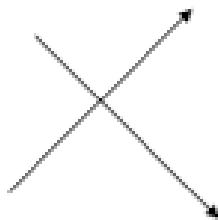
## 컨텐츠 기반 추천

코사인 유사도

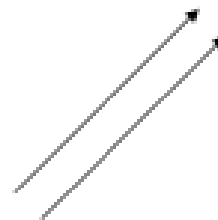
$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$



코사인 유사도 : -1



코사인 유사도 : 0



코사인 유사도 : 1

단어들 간 유사도를 구하여 유사한 단어들 많이 등장하는 컨텐츠를 추천

→ 이 과정에서 분석자와 개발자의 주관 개입

## 컨텐츠 기반 추천

### 장점

**Cold-Start** 현상에 크게 구애 받지 않음

서비스 초반 누적 데이터의 부족으로 제대로 된 추천이 어려운 문제

### 한계

새로운 사용자에게는 추천이 불가능 → 고질적인 문제

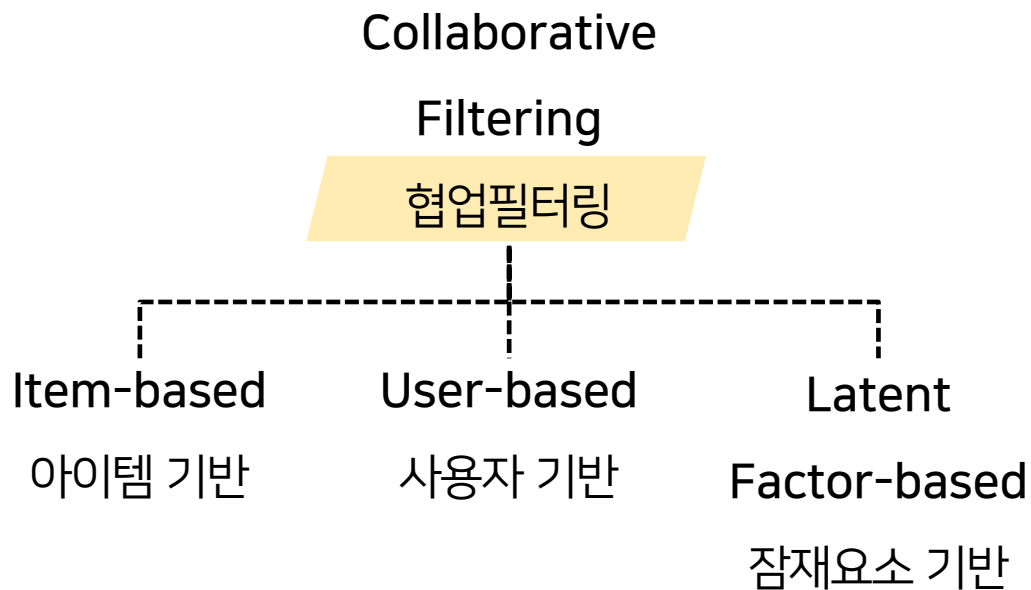
(Netflix 등 신규사용자에게 최초에 한해서 취향 등에 대한 질문함)

메타데이터로부터 주요 Feature를 추출하기 어려움



## 협업 필터링

협업 필터링이란?

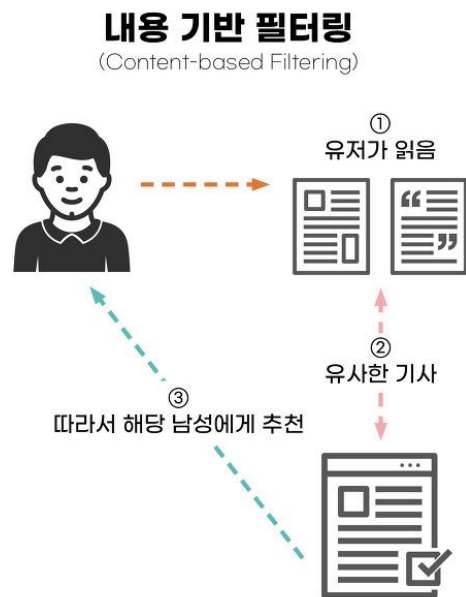


사용자 혹은 아이템들 간의 협업이 이루어짐



## 협업 필터링

### 아이템 기반 협업 필터링

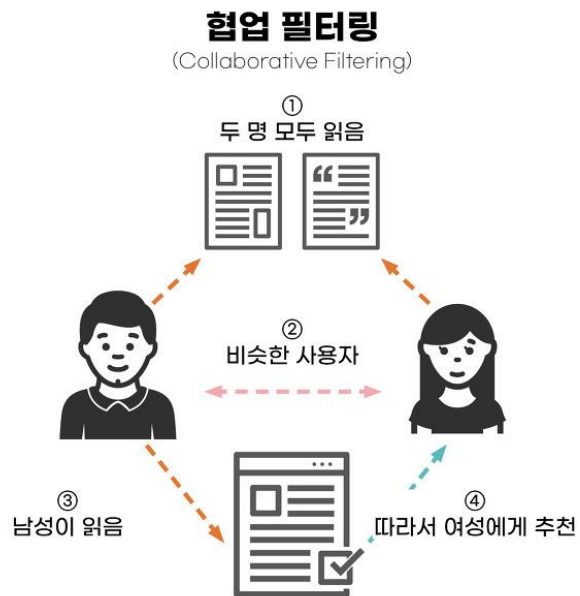


## 컨텐츠 기반

사용자의 히스토리와 콘텐츠 내부의 메타데이터를 활용하여 유사도 측정

## 협업 필터링

### 아이템 기반 협업 필터링



### 아이템 기반

아이템에 대한 구입내역, 선호도, 만족도를 기반으로  
사용자 혹은 제품 간의 **협업(상호작용)**을 통해 사용자가 선호하는 제품을 추천

## 협업 필터링

사용자 기반 협업 필터링



	스즈메의 문단속	아바타: 물의 길	앤트맨과 와스프	상견니	더 퍼스트 슬램덩크
건우	5	4	4	3	1
보현	1	0	1	3	4
지원	4	4	2	5	3
성우	4	2	3	2	2
수빈	5	3	1	2	?

협업 필터링을 위해서는 사용자와 제품간 상호작용 데이터 필요

데이터는 숫자로 표현되기 때문에 행렬로 표현 가능

## 협업 필터링

사용자 기반 협업 필터링



	스즈메의 문단속	아바타: 물의 길	앤트맨과 와스프	상견니	더 퍼스트 슬램덩크
건우	5	4	4	3	1
보현	1	0	1	3	4
지원	4	4	2	5	3
성우	4	2	3	2	2
수빈	5	3	1	2	?

사용자가 매긴 제품에 대한 평점을 바탕으로 시스템 설계

협업 필터링을 위해서 행렬의 형태로 표현 상호작용 데이터 필요

데이터는 숫자로 표현 가능 → 평점행렬에 행렬로 표현 가능



## 협업 필터링

사용자 기반 협업 필터링



	스즈메의 문단속	아바타: 물의 길	앤트맨과 와스프	상견니	더 퍼스트 슬램덩크
건우	5	4	4	3	1
보현	1	0	1	3	4
지원	4	4	2	5	3
성우	4	2	3	2	2
수빈	5	3	1	2	?

건우, 보현, 지원, 성우가 매긴 평점을 바탕으로  
수빈이가 아직 안 본 '더 퍼스트 슬램덩크'에 대한 평점 예측

## 협업 필터링

사용자 기반 협업 필터링

피어슨 상관 계수를 통해 유사도 계산

$$\text{Similarity}(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

	건우	보현	지원	성우
수빈	0.7171372	-0.2714484	0.4265617	0.5606119

## 협업 필터링

사용자 기반 협업 필터링



너굴맨이 또 처리했으니 안심하라고~

	건우	보현	지원	성우
수빈	0.7171372	-0.2714484	0.4265617	0.5606119



계산한 유사도를 가중치로 하여 슬램덩크의 평점을 가중합하여 예측평점으로 사용



## 협업 필터링

사용자 기반 협업 필터링

피어슨 상관 계수를 통해 유사도 계산

Rating Prediction

$$\frac{(0.717 * 1) + (-0.271 * 4) + (0.427 * 3) + (0.561 * 2)}{(0.717 - 0.271 + 0.427 + 0.561)} = 1.42$$

“예측 평점이 몇 점 이상일 때 사용자에게 추천할까?”

## 협업 필터링

사용자 기반 협업 필터링

피어슨 상관 계수를 통해 유사도 계산

Rating Prediction

$$\frac{(0.717 * 1) + (-0.271 * 4) + (0.427 * 3) + (0.561 * 2)}{(0.717 - 0.271 + 0.427 + 0.561)} = 1.42$$

“예측 평점이 몇 점 이상일 때 사용자에게 추천할까?”



분석자와 개발자의 주관에 따라 추천 진행



## 협업 필터링

사용자 기반 협업 필터링

Netflix has 230.7 Million Subscribers as of the fourth quarter  
2022. 2023. 2. 27.



한계

수많은 사용자와 콘텐츠가 있는 경우 평점행렬이 너무 커짐  
수 많은 컴퓨팅 파워와 시간 소모하여 **비현실적**



잠재 요인 기반 협업 필터링

## 협업 필터링

잠재 요소 협업 필터링

잠재 요소 협업 필터링

사용자와 아이템(컨텐츠)간에 상호작용(평점)이  
나타나는데 **잠재 요소**가 있다 가정

## 협업 필터링

잠재 요소 협업 필터링

잠재 요소 협업 필터링

사용자와 아이템(컨텐츠)간에 상호작용(평점)이  
나타나는데 **잠재 요소**가 있다 가정



잠재요소란?

사용자가 평점을 내리는 기준들

## 협업 필터링

잠재 요소 협업 필터링

잠재 요소 협업 필터링

사용자와 아이템(컨텐츠)간에 상호작용(평점)이  
나타나는데 **잠재 요소**가 있다 가정



잠재요소란?

사용자가 평점을 내리는 기준들



사용자-잠재요소, 잠재요소-아이템 관계를 행렬로 표현할 수 있도록  
적절히 사용자-아이템 관계 표현하는 **평점행렬 분해** !

## 협업 필터링

잠재 요소 협업 필터링



	어벤져스	포레스트 검프	매트릭스	엑시트	분노의 질주
건우	6	12	0	12	3
보현	12	10	6	8	10
지원	14	7	9	4	13
성우	16	4	12	0	16

장르가 평점에 중요한 요소라 가정  
 사용자×장르 , 장르×영화 이렇게 두 행렬로 분해

## 협업 필터링

잠재 요소 협업 필터링

사용자-장르

	Comedy	Action
건우	3	0
보현	2	2
지원	1	3
성우	0	4

장르-영화

	어벤져스	포레스트 검프	매트릭스	엑시트	분노의 질주
Comedy	2	4	0	4	1
Action	4	1	3	0	4

두가지 행렬로 분해해서 생각  
사용자×장르, 장르×영화 2개의 행렬이 생성



## 협업 필터링

잠재 요소 협업 필터링

특이값 분해(Singular Value Decomposition)

특이값을 원소로 가지는 대각행렬  $\Sigma$ 에서  
제일 작은 특이값부터 제외하는 방식으로 차원축소

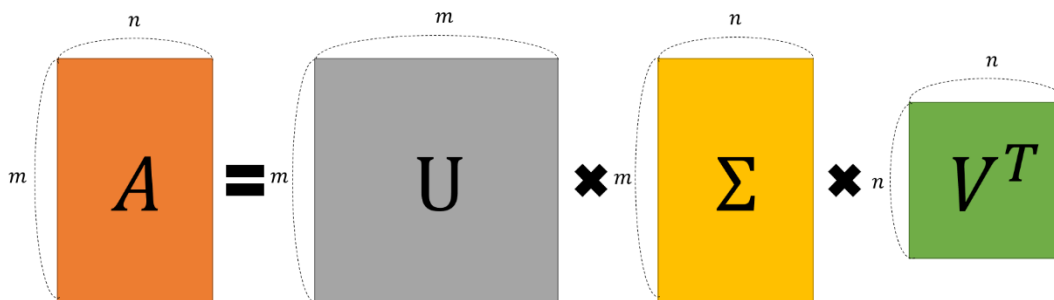
$$\begin{matrix} n \\ \text{---} \\ \boxed{A} \\ m \end{matrix} = \begin{matrix} m \\ \text{---} \\ \boxed{U} \\ m \end{matrix} \times \begin{matrix} n \\ \text{---} \\ \boxed{\Sigma} \\ m \end{matrix} \times \begin{matrix} n \\ \text{---} \\ \boxed{V^T} \\ n \end{matrix}$$

## 협업 필터링

잠재 요소 협업 필터링

특이값 분해(Singular Value Decomposition)

특이값을 원소로 가지는 대각행렬  $\Sigma$ 에서  
제일 작은 특이값부터 제외하는 방식으로 차원축소



A diagram illustrating the Singular Value Decomposition (SVD) process. It shows a matrix  $A$  (orange rectangle) with dimensions  $m$  (height) and  $n$  (width). This is equal to the product of three matrices:  $U$  (gray rectangle, dimensions  $m \times m$ ),  $\Sigma$  (yellow rectangle, dimensions  $m \times n$ ), and  $V^T$  (green rectangle, dimensions  $n \times n$ ). The dimensions are indicated by dashed lines and labels around each matrix. The equation is  $A = U \Sigma V^T$ .

저장공간을 절약함과 동시에 주요한 잠재요인만을 고려하여  
좀 더 정교한 추천이 가능



## 협업 필터링

잠재 요소 협업 필터링

특이값 분해(Singular Value Decomposition)

특이값을 원소로 가지는 대각행렬  $\Sigma$ 에서  
제일 작은 특이값부터 제외하는 방식으로 차원축소

SVD에 대한 보다 자세한 내용은 선대팀 2주차 클린업 참고

$$\begin{matrix} n \\ \boxed{A} \\ m \end{matrix} = \begin{matrix} m \\ \boxed{U} \\ m \end{matrix} \times \begin{matrix} m \\ \boxed{\Sigma} \\ m \end{matrix} \times \begin{matrix} n \\ \boxed{V^T} \\ n \end{matrix}$$

감  
사

저장공간을 절약함과 동시에 잠재요인만을 고려하여

좀 더 추천이 가능



## 협업 필터링

잠재 요소 협업 필터링

### 장점

연산이나 평점을 예측하는 방식이 합리적

### 한계

Cold-Start문제나 평점행렬의 특성으로 인한  
협업 필터링 기반 추천시스템의 근본적인 문제점 해결 못함

## 협업 필터링

잠재 요소 협업 필터링

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

현실에서 많은 소비자들이 모든 콘텐츠에 대해 평점을 내리지 않아  
대부분의 원소가 비어 있는 **희소 행렬(Sparse Matrix)** 형태



기본적인 행렬, 벡터 연산 & SVD 불가

## 협업 필터링

잠재 요소 협업 필터링



결측치에 대한 예측과 예측값이 원래 행렬의 값과 비슷하도록 **최적화** 하는 과정이 필요

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	4			2	
User 2		5		3	
User 3			5	4	4
User 4	5	2	1	2	



**SGD(Stochastic Gradient Descent)**

현실에서 **ALS(Alternating Least Squares)** 지 않아

대부분의 원소가 비어 있는 **희소 행렬(Sparse Matrix)** 형태



기본적인 행렬, 벡터 연산 & SVD 불가

# THANK YOU

지금까지 **야망!데마 클린업**이었습니다~  
클린업 3주 동안 수고 많으셨습니다!

