

# 생성형AI

## Day 11

### 머신러닝 III



## 목차

---

1. 비지도 학습
2. K-평균 클러스터링 (K-Means Clustering)
3. 계층적 군집 분석 (Hierarchical Clustering)
4. DBSCAN
5. 주성분 분석 (PCA)
6. 아이소맵 (Isomap)
7. t-SNE
8. 연관 규칙 학습 (Association Rule Learning)
9. 실습과제



# 비지도학습

## 비지도 학습이란?

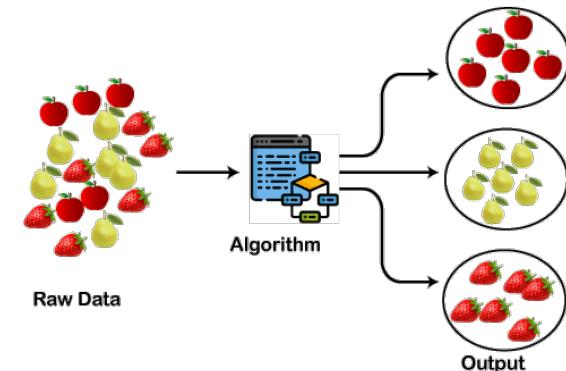
- 레이블이 없는 데이터를 사용하여 패턴이나 구조를 발견하는 머신러닝의 한 종류
- 입력 데이터를 기반으로 데이터 간의 유사성을 찾고, 데이터를 군집화하거나, 데이터를 압축하는 등의 작업을 수행

## 주요 특징

- 레이블 없음: 비지도학습은 레이블(목표 변수) 없이 입력 데이터만을 사용
- 패턴 발견: 데이터 내의 숨겨진 패턴이나 구조 탐색
- 탐색적 분석: 데이터의 특성을 이해하고, 잠재적인 그룹을 식별

## 주요 유형

- 클러스터링: 비슷한 특성을 가진 데이터 포인트들을 같은 그룹으로 묶는 과정
- 차원축소: 고차원 데이터를 더 적은 수의 차원으로 변환하여 데이터를 요약하는 과정
- 연관 규칙 학습: 데이터 내의 항목 간의 흥미로운 관계를 찾는 과정



## 비지도학습

---

### 비지도 학습 장점

- 레이블링 비용 절감: 레이블이 필요 없으므로 데이터 레이블링 비용이 절감됩니다.
- 데이터 탐색: 데이터를 탐색하고 이해하는 데 유용합니다.
- 새로운 패턴 발견: 사전에 정의되지 않은 새로운 패턴이나 구조를 발견할 수 있습니다

### 비지도 학습 단점

- 결과 해석의 어려움: 결과를 해석하는 것이 어려울 수 있습니다.
- 성능 평가의 어려움: 정량적으로 모델의 성능을 평가하기 어렵습니다.
- 정확성 보장 어려움: 지도학습에 비해 예측의 정확성을 보장하기 어렵습니다.

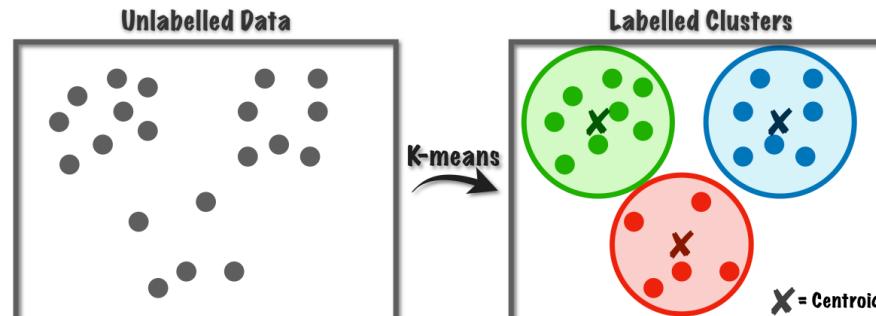
# K-평균 클러스터링 (K-Means Clustering)

## K-평균 클러스터링?

- 데이터를 K개의 군집으로 나누는 군집화 알고리즘
- 데이터 포인트를 유사한 특성을 가진 그룹으로 묶어 데이터의 구조를 이해
- 데이터 분석 및 시각화를 용이하게 합니다

## K-평균 클러스터링 특징

- 군집 수 K: 사용자가 군집 수 K를 사전에 정의, 모델의 성능과 직결되는 중요한 하이퍼파라미터
- 거리 기반 알고리즘: 유clidean 거리를 사용하여 데이터 포인트 간의 유사성을 측정
- 평균 중심: 각 군집의 중심은 해당 군집에 속한 데이터 포인트들의 평균 값으로 정의
- 반복적 과정: 군집 할당과 중심 업데이트를 반복하여 최적의 군집을 찾음
- 계산 효율성: 비교적 계산 효율성이 높아 대규모 데이터셋에서도 빠르게 동작



# K-평균 클러스터링 (K-Means Clustering)

---

## K-평균 클러스터링 기본원리

### 1. 초기화

- K개의 군집 중심(centroid)을 초기화, 초기 중심은 임의로 선택

### 2. 할당 단계

- 각 데이터 포인트를 가장 가까운 군집 중심에 할당
- 유clidean 거리(Euclidean Distance)를 사용하여 각 데이터 포인트와 군집 중심 간의 거리를 계산

### 3. 업데이트 단계

- 각 군집의 중심을 해당 군집에 속한 데이터 포인트들의 평균으로 업데이트

$$c_j = \frac{1}{C_j} \sum_{x_i \in C_j} x_i$$

- $c_j$ 는 군집  $j$ 의 새로운 중심,  $C_j$ 는 군집  $j$ 에 속한 데이터 포인트들의 집합

### 4. 반복

- 할당 단계와 업데이트 단계를 반복하여 군집 중심이 더 이상 변하지 않거나 지정된 반복 횟수에 도달할 때까지 수행

# K-평균 클러스터링 (K-Means Clustering)

---

## K-평균 클러스터링의 수식

- 유클리드 거리

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^n (x_{ik} - c_{jk})^2}$$

- 군집 중심 업데이트

$$c_j = \frac{1}{C_j} \sum_{x_i \in C_j} x_i$$

- 목적함수

- K-평균 클러스터링은 목적함수인 총 거리의 제곱합을 최소화하는 방향으로 작동

$$J = \sum_{j=1}^K \sum_{x_i \in C_j} d(x_i, c_j)^2$$

- $J$ 는 목적함수 값,  $K$ 는 군집의 수,  $C_j$ 는 군집  $j$ 에 속한 데이터 포인트들의 집합

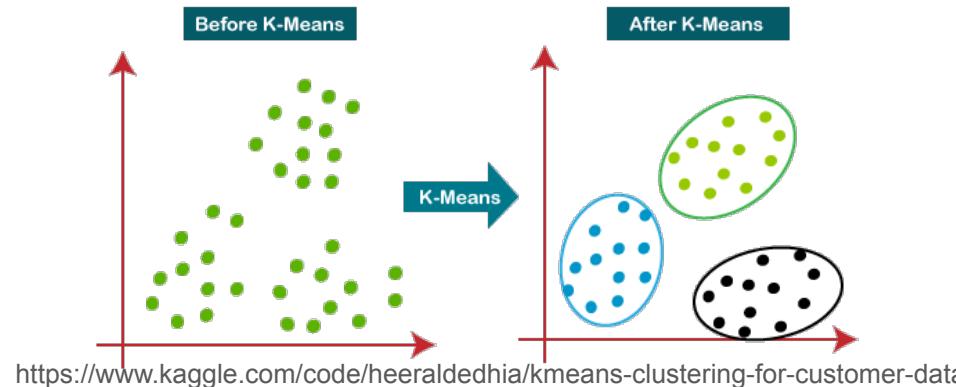
# K-평균 클러스터링 (K-Means Clustering)

## K- 평균 클러스터링 장점:

- 간단하고 빠름: 구현이 간단하고, 대규모 데이터셋에서도 빠르게 작동
- 해석 용이: 결과를 시각적으로 이해하고 해석 용이
- 확장성: 대규모 데이터에 대해 효율적으로 작동

## K- 평균 클러스터링 단점:

- K의 결정 어려움: 적절한 군집의 수 K를 사전에 결정하기 어려움
- 초기 중심에 민감: 초기 군집 중심의 선택에 따라 결과가 달라질 가능성 큼
- 구형 군집만 찾음: 군집의 모양이 구형(spherical)에 가깝다고 가정하기 때문에, 복잡한 형태의 군집을 찾기 어렵음
- 노이즈와 이상치에 민감: 노이즈와 이상치에 민감하게 반응할 가능성 높음



# K-평균 클러스터링 (K-Means Clustering)

## K-평균 클러스터링의 개선 방법

### - K-Means++ 초기화:

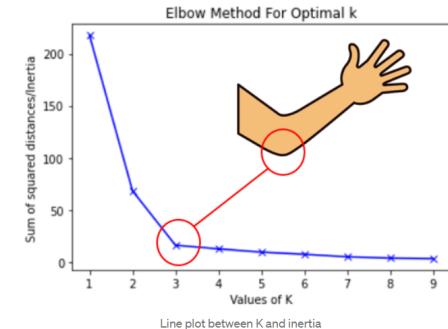
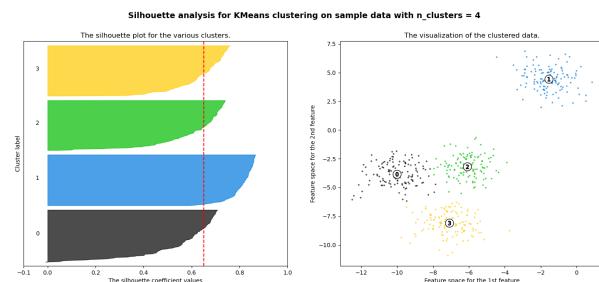
- 초기 군집 중심을 더 똑똑하게 선택하여, 수렴 속도를 높이고 성능을 향상
- 첫 번째 중심을 무작위로 선택한 후, 나머지 중심은 데이터 포인트 간의 거리를 고려하여 선택

### - 엘보우 방법 (Elbow Method):

- 최적의 군집 수 K를 선택하기 위한 방법
- 목적 함수 J의 감소율이 급격히 줄어드는 지점(엘보우 포인트)을 찾아 최적의 K를 결정

### - 실루엣 분석 (Silhouette Analysis):

- 각 데이터 포인트가 얼마나 잘 군집화되었는지를 측정
- 실루엣 값은 -1에서 1 사이의 값을 가지며, 값이 클수록 더 잘 군집화된 것



# 계층적 군집 분석 (Hierarchical Clustering)

## 계층적 군집 분석?

- 데이터 포인트들 간의 유사도를 바탕으로 계층적인 군집 구조를 형성하는 군집화 방법
- 데이터를 트리 구조로 표현하여, 군집화를 단계별로 진행
- 데이터를 계층적으로 분류하여, 데이터 간의 관계와 구조를 이해하는 데 도움

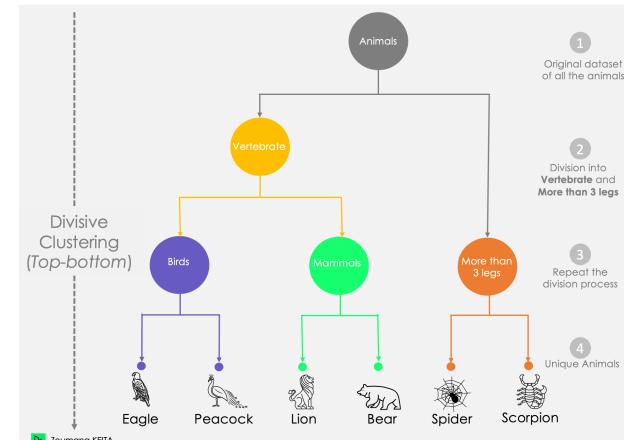
## 계층적 군집 분석의 유형

### 병합적 군집화 (Agglomerative Clustering):

- 데이터를 각각 하나의 군집으로 시작하여, 가장 가까운 군집들을 반복적으로 병합
- 최종적으로 하나의 군집이 형성될 때까지 병합 과정을 반복

### 분할적 군집화 (Divisive Clustering):

- 모든 데이터를 하나의 군집으로 시작하여, 가장 큰 군집을 반복적으로 분할
- 최종적으로 각 데이터가 하나의 군집을 형성할 때까지 분할 과정을 반복



# 계층적 군집 분석 (Hierarchical Clustering)

## 거리 측정 (Distance Measurement)

- 데이터 포인트들 간의 유사도를 측정하기 위해 거리 측정을 사용
- 일반적으로 유clidean 거리를 사용(맨해튼 거리, 코사인 유사도 등도 사용가능)

## 군집 간의 거리 측정 (Linkage Criteria)

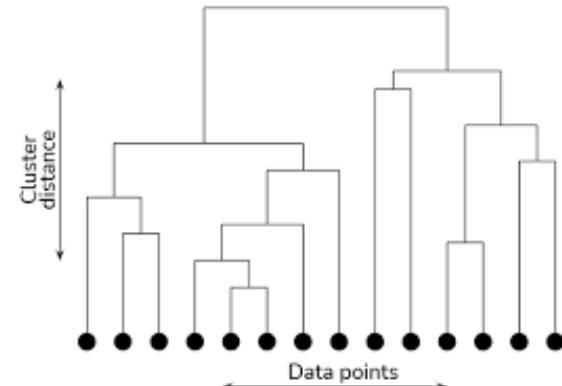
- 군집 간의 유사도를 측정하기 위한 기준을 설정

주요 연결 기준:

- 단일 연결 (Single Linkage): 두 군집 간의 최소 거리
- 완전 연결 (Complete Linkage): 두 군집 간의 최대 거리
- 평균 연결 (Average Linkage): 두 군집 간의 평균 거리
- 중심 연결 (Centroid Linkage): 두 군집의 중심 간 거리
- 워드 연결 (Ward's Linkage): 군집 내 분산의 증가량을 최소화하는 기준

## 덴드로그램 (Dendrogram)

- 계층적 군집화 과정을 시각적으로 표현한 트리 구조
- 각 노드는 군집을 나타내며, 노드 간의 높이는 군집 간의 거리 또는 유사도를 나타냄
- 덴드로그램을 통해 데이터의 군집화 과정을 시각적으로 분석 가능



# 계층적 군집 분석 (Hierarchical Clustering)

## 계층적 군집 분석 기본원리

### - 초기화 (Initialization)

- 각 데이터를 하나의 군집으로 시작
- 초기에는 n개의 군집이 존재 (n은 데이터 포인트의 수)

### - 거리 계산 (Distance Calculation)

- 모든 데이터 포인트 간의 거리를 계산

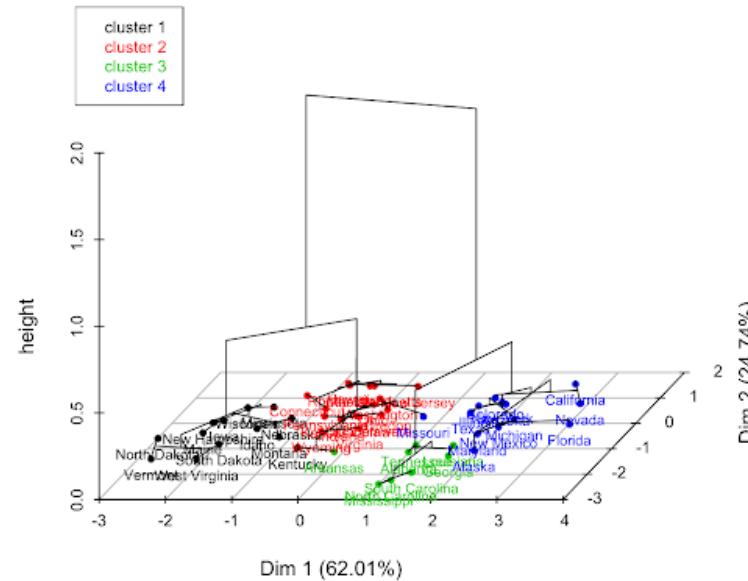
### - 군집 병합 (Cluster Merging)

- 가장 가까운 두 군집을 병합
- 병합 후 새로운 군집 간의 거리를 재계산

### - 반복 (Iteration)

- 군집의 수가 하나가 될 때까지 군집 병합 과정을 반복
- 덴드로그램 생성
- 군집 병합 과정을 덴드로그램으로 시각화

Hierarchical clustering on the factor map



# 계층적 군집 분석 (Hierarchical Clustering)

---

## 계층적 군집 분석 장점:

- 계층적 구조 시각화: 덴드로그램을 통해 데이터의 군집화 과정을 시각적으로 표현 가능
- 군집 수 결정 불필요: 사전에 군집 수를 결정할 필요가 없고 덴드로그램을 통해 적절한 군집 수를 선택 가능
- 유연성: 다양한 연결 기준을 사용하여 군집화를 수행할 수 있음

## 계층적 군집 분석 단점:

- 계산 비용: 데이터 포인트 간의 모든 거리를 계산하고 저장해야 하므로, 대규모 데이터셋에서는 계산 비용이 높음
- 병합 후 수정 불가: 병합된 군집은 다시 분할할 수 없으므로, 초기 병합 단계에서의 오류가 최종 결과에 영향을 미칠 수 있음
- 노이즈 민감성: 이상치나 노이즈 데이터에 민감할 수 있음

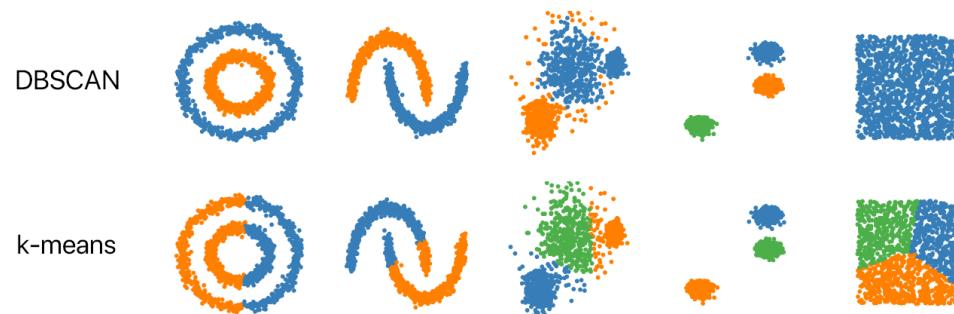
# DBSCAN

## DBSCAN이란?

- Density-Based Spatial Clustering of Applications with Noise
- 밀도 기반의 군집화 알고리즘
- 밀집된 영역은 군집으로 식별하고, 밀도가 낮은 데이터 포인트는 노이즈로 간주
- 데이터의 밀집 영역을 찾아내고, 군집의 크기나 형태에 구애받지 않는 유연한 군집화를 수행하는 것이 목표

## 특징

- 밀도 기반 군집화: 데이터 포인트의 밀도를 기준으로 군집을 형성
- 노이즈 처리: 밀도가 낮은 데이터 포인트를 노이즈로 간주하고, 군집에서 제외
- 유연한 군집 형성: 군집의 크기나 형태에 구애받지 않고 유연하게 군집을 형성
- 비교적 간단한 구현: 알고리즘이 비교적 간단하고, 구현이 용이



# DBSCAN

---

## 기본원리

- 밀도 기준 (Density Criteria):
  - DBSCAN은 두 개의 주요 하이퍼파라미터를 사용
  - $\epsilon$  (Eps): 데이터 포인트 간의 최대 거리, 최소 포인트 수 (MinPts): 군집을 형성하기 위해 필요한 최소 데이터 포인트 수
- 핵심 포인트 (Core Point):
  - 반경  $\epsilon$ 이내에 최소 MinPts 이상의 데이터 포인트가 있는 데이터 포인트를 핵심 포인트로 정의
$$|\{y \in D | d(x, y) \leq \epsilon\}| \geq \text{MinPts}$$
- 경계 포인트 (Border Point):
  - 반경  $\epsilon$ 이내에 최소 MinPts 이상의 데이터 포인트는 없지만, 핵심 포인트의 반경  $\epsilon$ 내에 있는 데이터 포인트를 경계 포인트로 정의
- 노이즈 포인트 (Noise Point):
  - 핵심 포인트와 경계 포인트 모두에 해당하지 않는 데이터 포인트를 노이즈 포인트로 정의

# DBSCAN

---

## 군집형성과정

### 1. 초기화

1. 모든 데이터 포인트를 방문하지 않은 상태로 초기화

### 2. 군집 확장

1. 방문하지 않은 데이터 포인트를 선택하여, 해당 포인트가 핵심 포인트인지 확인
2. 핵심 포인트인 경우, 반경  $\epsilon$  내의 모든 데이터 포인트를 포함하여 새로운 군집을 형성
3. 경계 포인트는 기존 군집에 포함되며, 노이즈 포인트는 군집에 포함되지 않음

### 3. 반복

1. 모든 데이터 포인트가 방문될 때까지 군집 확장 과정을 반복

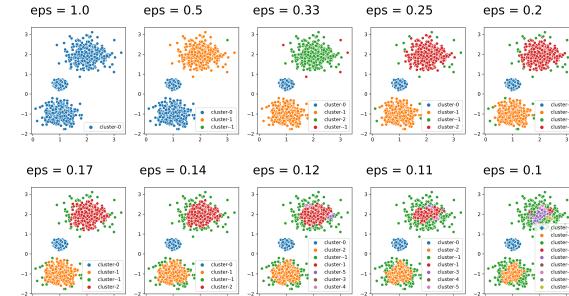
# DBSCAN

## DBSCAN 장점:

- 군집 수 미리 지정 불필요: 군집의 수를 사전에 지정할 필요가 없음
- 노이즈 데이터 처리: 노이즈 데이터를 효과적으로 처리하여 군집에서 제외할 수 있음
- 다양한 군집 형태: 데이터의 군집 형태가 구형(spherical)이 아니어도 유연하게 군집화 가능
- 확장성: 군집의 크기나 밀도에 관계없이 유연하게 적용 가능

## DBSCAN 단점:

- 하이퍼파라미터 설정 어려움:  $\epsilon$ 과 MinPts의 값을 설정하는 것이 어려움
- 밀도 변화에 민감: 데이터의 밀도가 균일하지 않을 경우, 군집화 성능이 저하 가능성
- 고차원 데이터에서의 성능 저하: 고차원 데이터에서는 거리 계산이 어려워 성능이 저하될 수 있음



# 주성분 분석 (PCA)

## 주성분 분석이란?

고차원 데이터를 저차원으로 변환하여 데이터의 주요 변동성을 보존하는 차원 축소 기법

데이터의 분산을 최대화하는 직교 축을 찾아 데이터를 새로운 좌표계로 변환하여 노이즈를 줄이고 시각화 및 해석 용이  
데이터 시각화, 노이즈 제거, 데이터 압축 등의 목적을 위해 사용

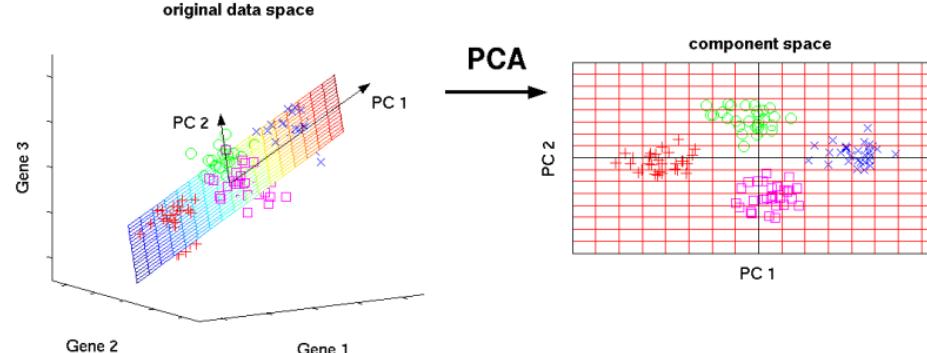
## 주성분 분석 특징

분산 최대화: PCA는 데이터의 분산을 최대화하는 방향으로 새로운 축 탐색

직교 축: 주성분은 서로 직교(orthogonal)하는 축으로 구성

순서 중요: 첫 번째 주성분은 가장 큰 분산을 가지며, 두 번째 주성분은 그 다음으로 큰 분산을 가짐

데이터 변환: 고유벡터를 사용하여 데이터를 새로운 좌표계로 변환



# 주성분 분석 (PCA)

## PCA 기본원리

### 데이터 중심화 (Centering the Data)

- 각 변수의 평균을 0으로 설정, 중심화된 데이터 행렬  $X$ 를 만듭니다

### 공분산 행렬 계산 (Calculating the Covariance Matrix):

- 중심화된 데이터의 공분산 행렬  $\Sigma$ 를 계산

$$\Sigma = \frac{1}{n-1} X^T X$$

### 고유값 분해 (Eigenvalue Decomposition):

- 공분산 행렬  $\Sigma$ 의 고유값과 고유벡터를 계산, 고유값  $\lambda_i$  와 고유벡터  $v_i$  를 탐색

$$\sum v_i = \lambda_i v_i$$

### 주성분 선택 (Selecting Principal Components):

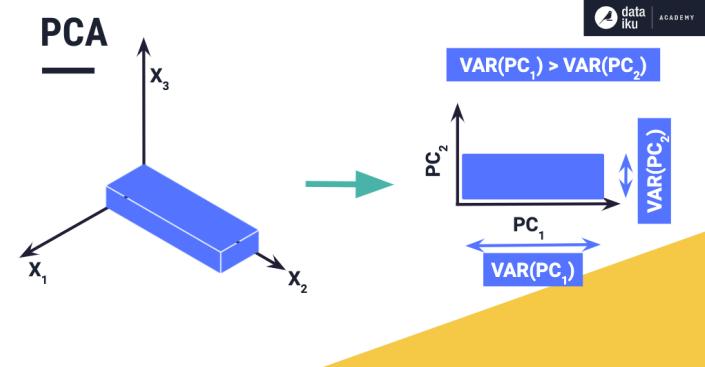
- 고유값의 크기 순으로 고유벡터를 정렬하여 주성분을 선택함
- 가장 큰 고유값에 해당하는 고유벡터가 첫 번째 주성분을 나타냄

### 데이터 변환 (Transforming the Data):

- 선택된 주성분을 사용하여 데이터를 새로운 좌표계로 변환

$$Z = X V$$

- 여기서  $Z$ 는 변환된 데이터 행렬,  $V$ 는 선택된 고유벡터 행렬



# 주성분 분석 (PCA)

---

## 주성분 분석 장점:

- 차원 축소: 데이터의 차원을 축소하여 계산 비용을 줄이고, 시각화가 쉬워짐
- 노이즈 제거: 주요 성분을 제외한 나머지 성분을 제거하여 데이터의 노이즈 감소
- 데이터 압축: 중요한 정보만을 보존하면서 데이터를 압축

## 주성분 분석 단점:

- 해석의 어려움: 변환된 주성분이 원래 변수와 어떤 관계가 있는지 해석하기 어려울 수 있음
- 선형성 가정: PCA는 선형 변환만을 사용하므로, 비선형 구조를 가진 데이터에는 적합하지 않음
- 정보 손실: 차원을 축소하는 과정에서 일부 정보가 손실될 가능성 높음

## 개선방법

- 비선형 차원 축소 기법:
  - PCA는 선형 변환만을 사용하므로, 비선형 구조를 가진 데이터에는 다른 차원 축소 기법을 사용
- 고차원 데이터 처리:
  - 고차원 데이터를 처리할 때는, 먼저 중요하지 않은 변수를 제거하거나 변수 선택(feature selection)을 수행하여 차원 감소
- PCA의 변형:
  - 커널 PCA (Kernel PCA): 비선형 변환을 통해 데이터의 비선형 구조를 반영
  - Sparse PCA: 희소성을 적용하여 주성분 탐색

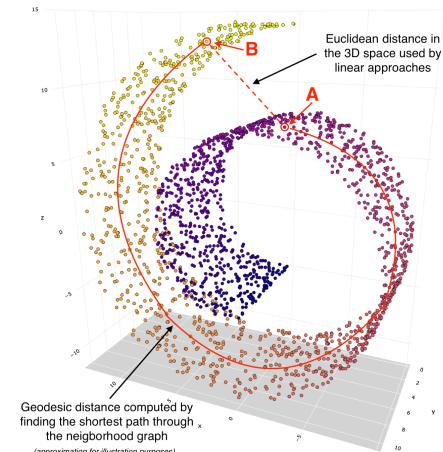
# 아이소맵 (Isomap)

## 아이소맵(Isomap)?

- 비선형 차원 축소 기법
- 고차원 데이터의 기하학적 구조를 보존하면서 저차원으로 변환하는 방법
- 지오데식 거리(Geodesic Distance)를 사용하여 데이터간의 거리를 측정하고, 저차원 공간에서 데이터의 구조를 시각화

## 아이소맵 특징

- 비선형 차원 축소: 비선형 데이터 구조를 보존하면서 차원을 축소
- 지오데식 거리 기반: 데이터 포인트 간의 지오데식 거리를 사용하여 데이터의 구조적 관계를 반영
- 그래프 기반 접근: 최근접 이웃 그래프를 구성하여 데이터 간의 관계를 모델링
- MDS 사용: 저차원 임베딩을 생성하기 위해 다차원 척도법(MDS)을 사용



# 아이소맵 (Isomap)

## Isomap 기본원리

최근점 이웃 그래프 구성 (Constructing Nearest Neighbor Graph):

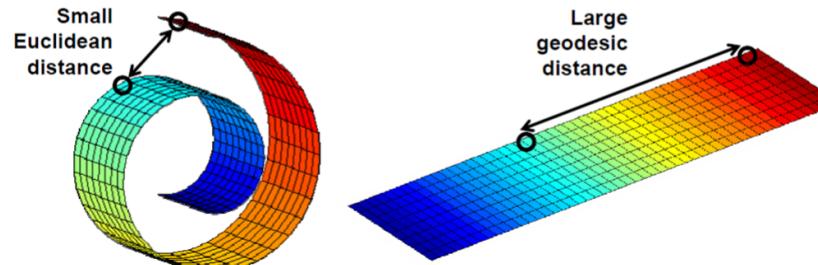
- 각 데이터 포인트에 대해, k개의 최근점 이웃 탐색
- 최근점 이웃들 간의 연결을 통해 그래프를 구성

지오데식 거리 계산 (Calculating Geodesic Distances):

- 그래프 내의 모든 데이터 포인트 쌍 간의 최단 경로를 계산하여 지오데식 거리를 측정
- 최단 경로 알고리즘 (예: 다익스트라 알고리즘)을 사용하여 각 포인트 간의 지오데식 거리를 계산

다차원 척도법 (Multidimensional Scaling, MDS):

- 지오데식 거리 행렬을 입력으로 받아, 저차원 임베딩을 생성
- MDS는 입력 거리 행렬을 보존하는 방식으로 저차원 공간에서 데이터 포인트를 배치



## 아이소맵 (Isomap)

---

지오데식 거리 계산(Calculating Geodesic Distances):

각 데이터 포인트  $i$ 와  $j$  사이의 지오데식 거리  $d_G(i, j)$ 는 그래프 내에서 최단 경로를 통해 계산

$$d_G(i, j) = \min_{p \in P(i, j)} \sum_{(u, v) \in p} d(u, v)$$

$P(i, j)$ 은 데이터 포인트  $i$ 와  $j$  사이의 모든 경로의 집합,  $d(u, v)$ 는 그래프에서 연결된 포인트  $u$ 와  $v$ 간의 거리

다차원 척도법 (MDS):

지오데식 거리 행렬  $D$ 를 사용하여 저차원 임베딩  $Y$ 를 생성

MDS는 거리 행렬  $D$ 를 저차원 공간에서 유사하게 재현할 수 있는 점들의 배치 탐색

## 아이소맵 (Isomap)

---

### Isomap 장점:

비선형 구조 보존: 데이터의 비선형 구조를 보존하면서 차원을 축소

지오데식 거리 사용: 실제 데이터의 구조적 관계를 반영하는 지오데식 거리를 사용

고차원 데이터 시각화: 고차원 데이터를 저차원 공간에서 시각화하여 이해하기 쉬움

### Isomap 단점:

계산 복잡성: 최근접 이웃 그래프 구성 및 지오데식 거리 계산의 계산 비용이 높음

노이즈에 민감: 데이터에 노이즈가 많은 경우, 그래프 구조가 왜곡될 수 있음

k값 결정: 적절한 k값을 선택하는 것이 어려움

### Isomap 개선 방법

#### - 최적의 k값 선택:

- 다양한 k값을 시도하여 최적의 k값을 선택, 그래프의 연결성과 데이터의 구조적 보존에 영향을 미침

#### - 노이즈 제거:

- 데이터 전처리 단계에서 노이즈를 제거하여 그래프 구조의 왜곡 축소

#### - 고차원 데이터 처리:

- 고차원 데이터를 처리할 때, 차원 축소 기법(PCA 등)을 먼저 적용하여 차원을 줄인 후 Isomap을 적용할 수 있음

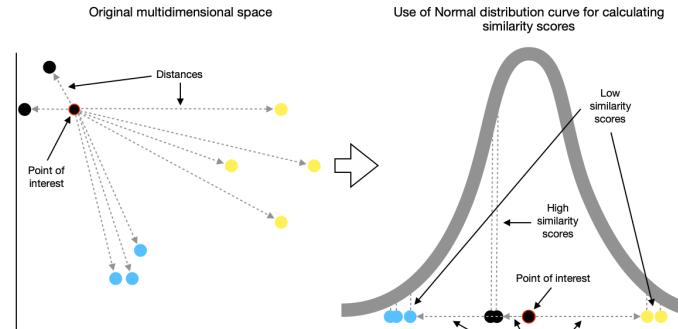
# t-SNE

## t-SNE란?

- t-Distributed Stochastic Neighbor Embedding
- 고차원 데이터를 저차원 공간에 효과적으로 시각화하기 위해 개발된 비선형 차원 축소 기법
- 고차원 데이터의 클러스터링 구조를 시각적으로 표현하는 데 사용
- 데이터 포인트 간의 지역적 유사성을 보존하는 데 강력한 성능

## t-SNE 특징:

- 비선형 차원 축소: 고차원 데이터의 비선형 구조를 저차원에서 보존
- 지역적 유사성 보존: 데이터 포인트 간의 지역적 유사성을 유지
- 확률 기반 접근: 데이터 포인트 간의 거리 정보를 확률로 변환하여 차원을 축소합니다.
- KL 발산 최소화: 고차원과 저차원 간의 확률 분포 차이를 최소화하여 최적의 저차원 임베딩 탐색



# t-SNE

## t-SNE 기본원리

고차원 공간에서 거리 계산 및 확률 변환

- 데이터 포인트 간의 거리를 계산하고 이를 확률로 변환하여 가까운 이웃 간의 유사성을 표현

저차원 공간에서 거리 계산 및 확률 변환:

- 초기화된 저차원 공간에서의 거리도 확률로 변환하여 고차원 공간과 비교

확률 분포 간의 차이 최소화:

- 고차원 공간과 저차원 공간의 확률 분포 간의 차이를 KL 발산을 통해 최소화하여 최적의 저차원 배치 탐색



[https://gaussian37.github.io/ml-concept-t\\_sne/](https://gaussian37.github.io/ml-concept-t_sne/)

# t-SNE

---

## t-SNE 장점:

- 고차원 데이터 시각화: 고차원 데이터의 복잡한 구조를 2차원 또는 3차원으로 효과적으로 시각화 가능
- 지역적 유사성 보존: 데이터 포인트 간의 지역적 유사성을 잘 유지하여, 클러스터 구조 명확히 구성
- 다양한 데이터 유형 처리: 이미지, 텍스트, 유전자 데이터 등 다양한 데이터 유형에 적용 가능

## t-SNE 단점:

- 시간 복잡도: 큰 데이터셋에서는 계산 비용이 높아 실행 시간이 오래 소요될 수 있음
- 매개변수 민감성: 매개변수(예: perplexity)에 민감하여, 적절한 값을 찾기 위해 실험이 필요
- 해석의 어려움: 결과가 저차원 공간에서 어떻게 해석되어야 하는지 명확하지 않을 수 있음

## t-SNE의 개선 방법

### 매개변수 최적화:

- perplexity, 학습률 등의 매개변수를 최적화하여 성능 향상
- 일반적으로 perplexity는 데이터 포인트 수의 1/10에서 1/5 사이의 값을 사용

### 대규모 데이터셋 처리:

- Barnes-Hut t-SNE와 같은 변형을 사용하여 대규모 데이터셋에서도 효율적으로 작동

### 차원 축소 전 전처리:

- 노이즈를 제거하고, 사전에 차원을 축소한 후 t-SNE를 적용하면 성능 향상 가능성 높음

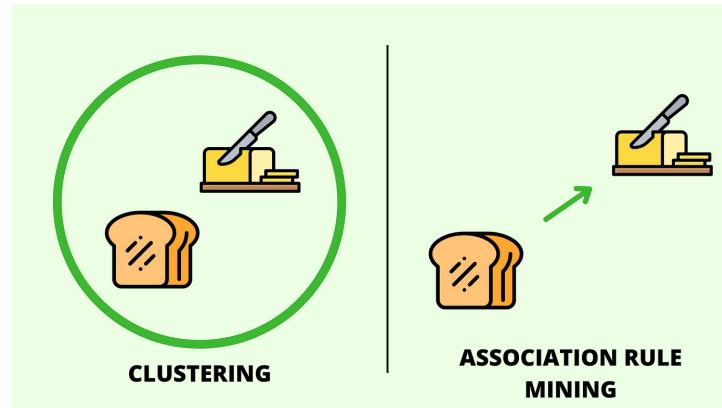
# 연관 규칙 학습(Association Rule Learning)

## 연관 규칙 학습?

- 항목 간의 흥미로운 관계를 찾는 비지도 학습의 한 방법
- 특정 항목이 나타날 때 다른 항목이 함께 나타날 확률을 계산하여 유용한 패턴을 도출
- 주로 Apriori 알고리즘과 FP-Growth 알고리즘

## 연관 규칙 학습 특징:

- 빈발 항목 집합 기반: 빈발 항목 집합을 기반으로 유의미한 연관 규칙을 도출
- 계산 효율성: Apriori와 FP-Growth 알고리즘을 통해 계산 효율성 향상
- 응용 가능성: 다양한 분야에서 유용하게 활용



# 연관 규칙 학습(Association Rule Learning)

---

## 주요용어

### 항목 (Item)

- 데이터베이스 내의 단일 항목, ex) 슈퍼마켓의 특정 상품을 의미

### 항목 집합 (Itemset)

- 여러 항목으로 구성된 집합, ex) 장바구니에 담긴 여러 상품들이 항목 집합 구성

### 거래 (Transaction)

- 항목 집합을 포함하는 단일 데이터 행, ex) 한 번의 구매 내역이 하나의 거래

### 빈발 항목 집합 (Frequent Itemset)

- 특정 지지도 기준을 충족하는 항목 집합, ex) 자주 함께 구매되는 상품들의 집합

### 지지도 (Support)

- 특정 항목 집합이 데이터베이스에서 나타나는 빈도, 항목 집합의 인기를 나타냄

$$Support(A) = \frac{\text{Number of transactions containing } A}{\text{Total number of transactions}}$$

### 신뢰도(Confidence)

- 항목 집합 A가 주어졌을 때 항목 집합 B가 발생할 확률, 연관 규칙의 강도를 나타냄

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

### 향상도(Lift)

- 항목 집합 A와 B가 독립적일 때에 비해 실제로 얼마나 더 자주 함께 나타나는지를 나타냄

$$Lift(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A) \times Support(B)}$$

# 연관 규칙 학습(Association Rule Learning)

---

## 연관 규칙 기본원리

### 1. 빈발 항목 집합 탐색:

#### - 최소 지지도 설정:

- 분석의 기준이 되는 최소 지지도를 설정, 이는 얼마나 자주 등장해야 의미 있는 패턴으로 간주할지를 결정

#### - 단일 항목 집합 탐색:

- 각 단일 항목의 지지도를 계산하고, 최소 지지도를 충족하는 항목을 선택

#### - 조합 항목 집합 탐색:

- 두 개 이상의 항목을 조합하여 새로운 항목 집합을 생성하고, 지지도를 계산
- 최소 지지도를 충족하는 항목 집합만을 남기고 삭제

#### - 반복:

- 더 이상 최소 지지도를 충족하는 항목 집합이 없을 때까지 단계를 반복

### 2. 연관 규칙 생성:

#### - 빈발 항목 집합 기반 규칙 생성:

- 빈발 항목 집합에서 가능한 모든 연관 규칙을 생성, ex) {A,B}에서  $A \rightarrow B$ 와  $B \rightarrow A$ 를 생성

#### - 신뢰도와 향상도 계산:

- 생성된 규칙에 대해 신뢰도와 향상도를 계산하여 규칙의 유의미성을 평가

#### - 최소 신뢰도 및 향상도 기준 설정:

- 최소 신뢰도와 향상도를 설정하여 의미 있는 규칙만을 선택

# 연관 규칙 학습(Association Rule Learning)

---

## 연관 규칙 학습 장점:

- 유용한 패턴 발견: 데이터에서 유용한 패턴과 규칙을 발견하여 비즈니스 전략 수립에 역할
- 다양한 응용 분야: 마케팅, 상품 배치, 추천 시스템 등 다양한 분야에 적용
- 단순성과 이해 용이성: 규칙이 명확하고 쉽게 해석될 수 있어 비전문가도 쉽게 이해 가능

## 연관 규칙 학습 단점:

- 대규모 데이터셋에서의 계산 복잡성: 대규모 데이터셋에서 빈발 항목 집합을 탐색하는 데 많은 계산이 필요
- 희소성 문제: 데이터가 희소한 경우 유의미한 규칙을 찾기 어려울 수 있음
- 과적합: 너무 많은 규칙이 생성될 경우, 과적합 문제가 발생할 수 있음

## 연관 규칙 학습 개선 방법

- 알고리즘 최적화:
  - Apriori 알고리즘의 경우, 후보 항목 집합 생성을 최소화하여 효율성 향상
  - FP-Growth 알고리즘의 경우, FP-Tree 구조를 최적화하여 탐색 속도를 향상
- 데이터 전처리:
  - 데이터 정규화, 중복 제거 등의 전처리 과정을 통해 데이터의 품질을 향상
- 평가 지표 다양화:
  - 신뢰도와 향상도 외에도 다양한 평가 지표를 사용하여 규칙의 유의미성을 평가
  - 예: 흥미도(Interestingness), 신뢰도 제곱(Confidence Squared)

## 예제 코드

---

colab : <https://colab.research.google.com/drive/1hyG8bXP4RluL3xGQ8MAEYYfyTcJLReJL?usp=sharing>

## 실습 과제

---

### 데이터셋을 활용한 지도학습 모델 적용

#### 1. 인터넷에서 자유롭게 데이터셋 확보

#### 2. 데이터 탐색 및 전처리

- 데이터셋을 로드하고, 결측치 처리, 인코딩, 표준화 등 전처리 과정을 진행
- 데이터의 특성과 분포를 시각화하여 이해

#### 3. 모델 선택 및 학습

##### - 배운 비지도 학습을 적용하여 모델 생성

- K-평균 클러스터링 (K-Means Clustering)
- 계층적 군집 분석 (K-Means Clustering)
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- 주성분 분석 (PCA, Principal Component Analysis)
- 아이소맵 (Isomap)
- t-SNE (t-Distributed Stochastic Neighbor Embedding)
- 연관 규칙 학습 (Association Rule Learning)

#### 4. 모델 평가 및 시각화

##### - 시각화 등을 통해 모델 평가

# 실습 진행