# 비지도학습

정혜선



Growth Hackers

# CONTENTS





Growth Hackers

# 비지도학습?

- **지도 학습 (Supervised Learning)**
  - 이미 라벨이 존재하는 데이터를 모델을 통해 학습

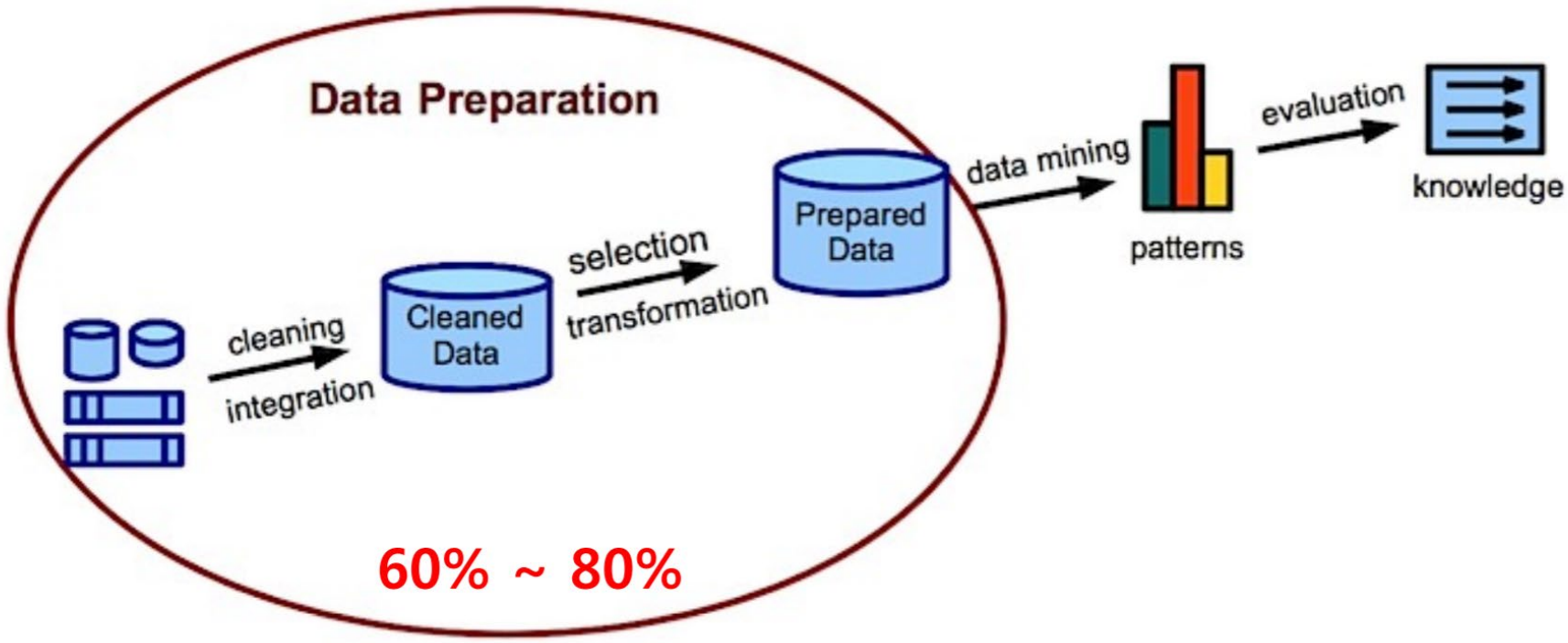
  →새로운 데이터의 라벨을 예측

- **비지도 학습(Unsupervised Learning)**
  - 데이터를 분류하는 라벨이 존재하지 않음
  - 데이터에 내재된 특성을 분석하여 유사한 데이터를 구별하거나 묶는 과정

| | | |
|---|---|---|
| 지도학습 | Classification | kNN |
| | | Naïve Bayes |
| | | Support Vector machine |
| | | Decision Tree |
| | Regression | Linear regression |
| | | Locally weighted linear regression |
| | | Ridge |
| | | Lasso |
| 비지도학습 | | Clustering |
| | | K means |
| | | Density estimation |
| | | Expectation maximization |
| | | Pazen window |
| | | DBSCAN |

# 데이터 전처리

- 분석 및 처리에 적합한 형식으로 데이터를 조작하는 것






Growth Hackers

# 데이터 전처리

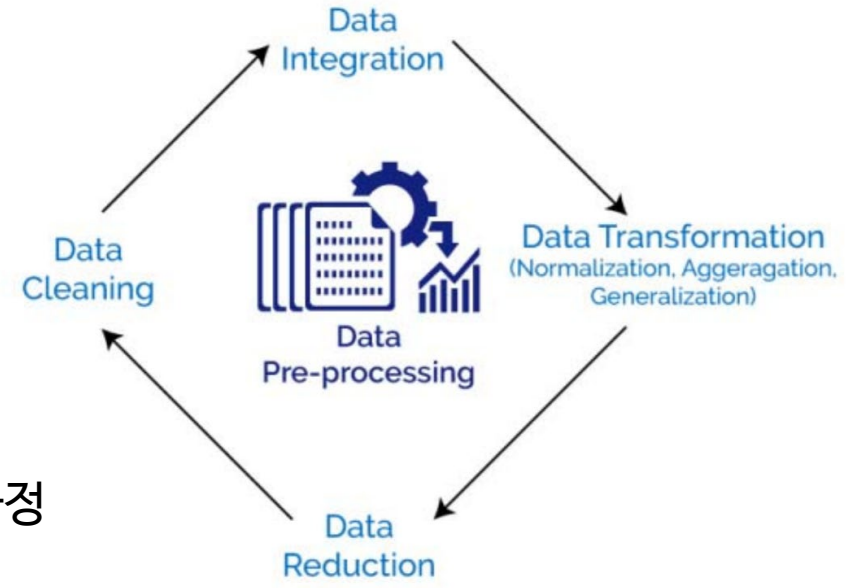
- 분석 및 처리에 적합한 형식으로 데이터를 조작하는 것

**데이터 정제**
- 이상치, 결측값 검색, 수정 및 제거
- 데이터의 신뢰도를 높이는 과정

**데이터 통합**
- 데이터, 스키마 통합
- 여러 소스의 데이터를 통합하는 과정

**데이터 변환**
- 데이터 요약, 집계 작업
- 노이즈 제거, 새로운 속성 추가 등
- 데이터 정규화
- 효과적인 분석을 위해 데이터를 변환 및 변형하는 과정

**데이터 정리**
- 데이터 크기 축소






Growth Hackers

# 데이터 전처리

- 데이터 정규화

Scaling

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- 데이터 군 내에서 특정 데이터가 차지하는 위치
- 데이터 포인트 간의 거리가 중요한 분석에서 주로 쓰임 (e.g. SVM, kNN)
- Outlier에 주의해야 함

Standardization

$$x' = \frac{x - x_{mean}}{\sigma}$$

- 특정 데이터 값과 평균까지의 상대적 거리
- z-score
- Linear regression, Logistic regression, Linear discriminate analysis.

# 차원 축소

- 차원 축소?
  - 고차원의 데이터 → 데이터 간의 관계를 설명할 수 있는 중요한 차원으로 변환

- 차원 축소의 필요성
  - 차원: 공간 내에 있는 점 등의 위치를 나타내기 위해 필요한 축의 개수
  - 변수의 수가 늘어난다 = 차원의 늘어난다 = 데이터 공간이 커진다
    **= 분석을 위해 필요한 최소한의 데이터 건수가 많아진다**
  - 만약 큰 공간을 충분히 표현할 만큼의 데이터 수집이 되지 않은 채 분석을 한다면? 과적합 (Overfitting)이 발생

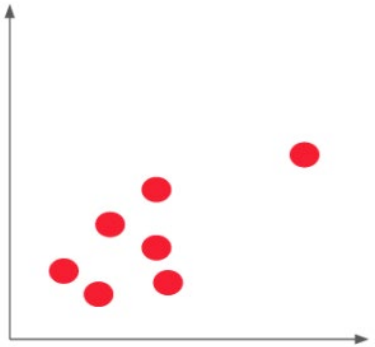
- 차원 축소의 효과
  - 1) 차원의 저주 탈피 (차원이 증가하면 데이터를 표현하기 위한 공간은 기하급수적으로 커지고 그로 인해 차원이 낮을 때 없었던 문제 (신뢰도 및 정확도 감소, 러닝 타임 증가)가 발생한다)
  - 2) 시각화의 용이성

# 차원 축소

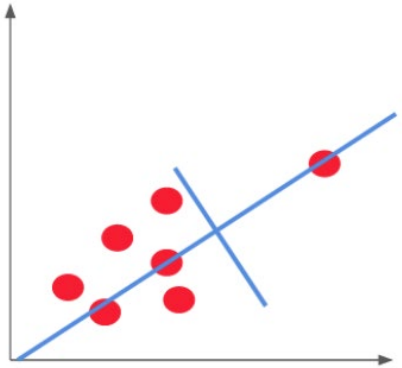
- 차원 축소 방법
  - 1) Feature Selection
    - 가지고 있는 여러 변수들 중 중요한 것을 고르기
    - 분석 주제: 변수 간에 중첩이 있는가? 어떤 변수가 중요한가? 어떤 변수가 타깃에 큰 영향을 주는가?
    - 분석 방법: 상관 분석 (Correlation) / VIF(분산팽창지수, Variance Inflation Factor) 분석 / Random Forest, XGBoost 등을 이용한 Variable importance 분석 / 등..
  - 2) Feature Extraction
    - 모든 변수를 조합하여 전체 데이터를 잘 표현할 수 있는 중요 성분을 가진 새로운 변수 추출
    - 분석 방법: 주성분분석 (Principle Component Analysis) / TSNE (T-Distributed Stochastic Neighbor Embedding) / 비음수 행렬 분해 (NMF) / 등..

# 차원 축소

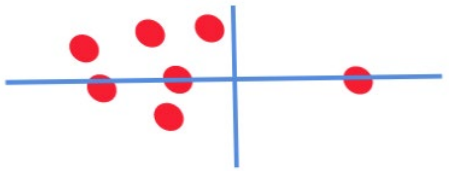
## 주성분분석 (Principle Component Analysis)



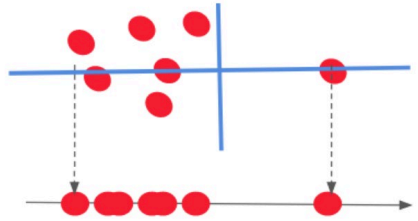
- 원본 데이터!



- 데이터의 변화의 폭이 가장 큰 축 & 그것과 직교하는 축 찾기



- 축의 방향과 위치를 전환시켜 데이터를 균등하게 분포시키기



- 1차원으로도 축소할 수 있음!

# 차원 축소

## 주성분분석 (Principle Component Analysis)

이 10송이의 표본은 꽃잎의 길와 폭이 제각각이지만 그 값에는 공통적인 특징이 있다. 꽃잎의 길이가 크면 꽃잎의 폭도 커지며 그 비율은 거의 일정하다. 그 이유는 (꽃잎의 길이, 꽃잎의 폭)이라는 2차원 측정 데이터는 사실 "꽃의 크기"라는 보다 근본적인 데이터가 두 개의 다른 형태로 표현된 것에 지나지 않기 때문이다. 바로 측정되지는 않지만 측정된 데이터의 기저에 숨어서 측정 데이터를 결정짓는 데이터를 **잠재변수(latent variable)**이라고 부른다.

PCA에서는 잠재변수와 측정 데이터가 선형적인 관계로 연결되어 있다고 가정한다. 즉, $i$번째 표본의 측정 데이터 벡터 $x_i$의 각 원소를 선형조합하면 그 뒤에 숨은 $i$번째 표본의 잠재변수 $u_i$의 값을 계산할 수 있다고 가정한다. 이를 수식으로 나타내면 다음과 같다.

$$u_i = w^T x_i$$

이 식에서 $w$는 측정 데이터 벡터의 각 원소를 조합할 가중치 벡터이다.

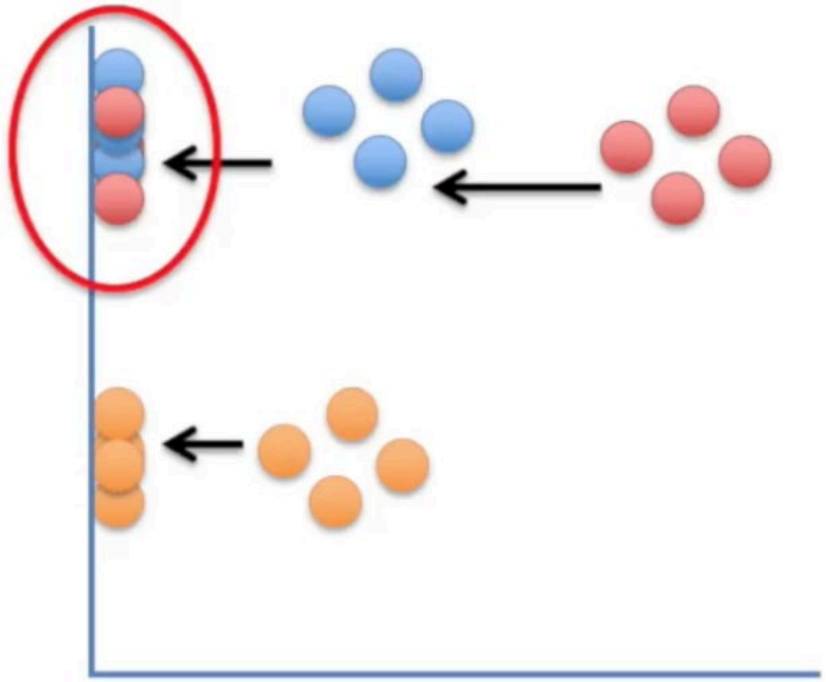
붓꽃의 예에서는 꽃잎의 길이와 꽃잎의 폭을 선형조합하여 꽃의 크기를 나타내는 어떤 값을 찾은 것이라고 생각할 수 있다.

$$u_i = w_1 x_{i,1} + w_2 x_{i,2}$$

# 차원 축소

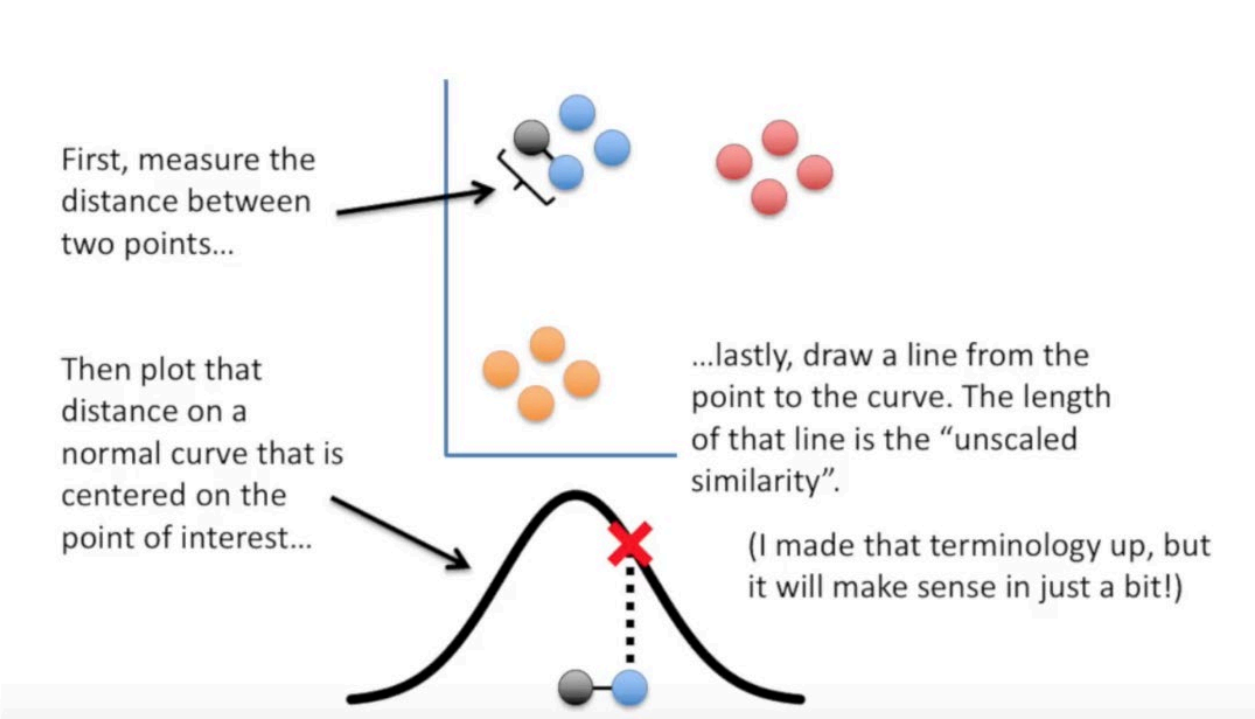
## 주성분분석 (Principle Component Analysis)

- 문제점



# 차원 축소

## TSNE (T-Distributed Stochastic Neighbor Embedding)




- 데이터 사이의 거리를 잘 보존하는 2차원 표현을 찾기
- 각 데이터를 2차원에 무작위로 표현 → 원본 특성 공간에서의 거리가 가까운 데이터는 가깝게, 먼 데이터는 멀게 표현
- 가까운 데이터 군집을 구별하여 표현하는 데 효과적
- 단, 계산할 때마다 축의 위치가 바뀌어 다른 모양이 나옴 → 데이터 분석에는 유용하지만, 학습 모델에서의 피처로는 적절하지 않음

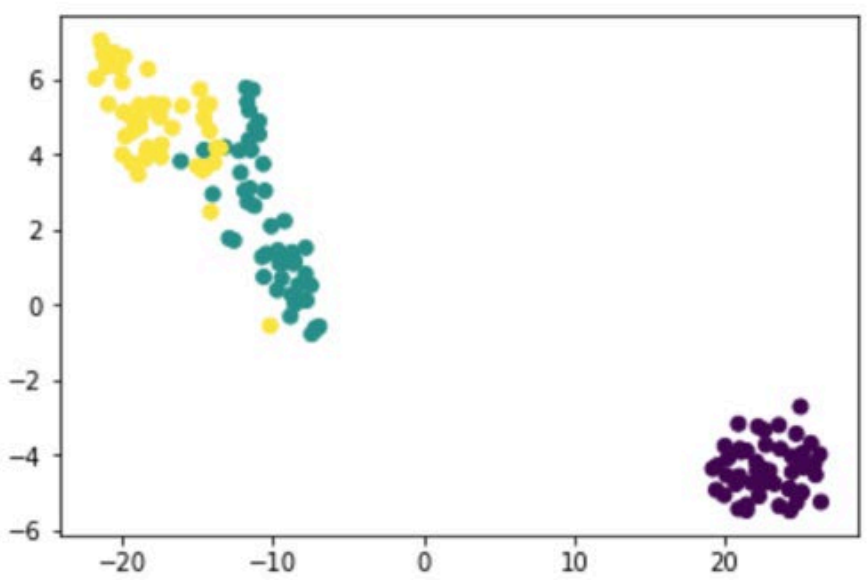


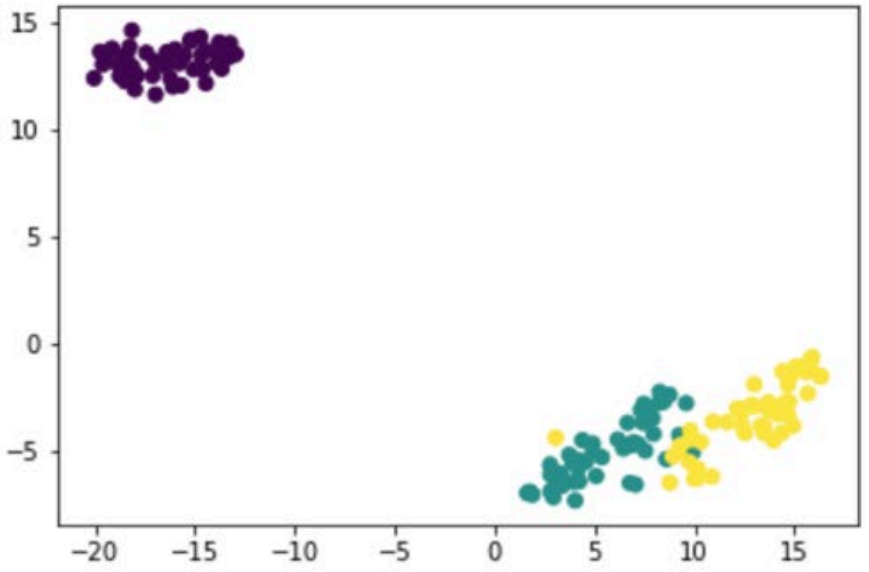
Growth Hackers

# 차원 축소

## TSNE (T-Distributed Stochastic Neighbor Embedding)

- 문제점



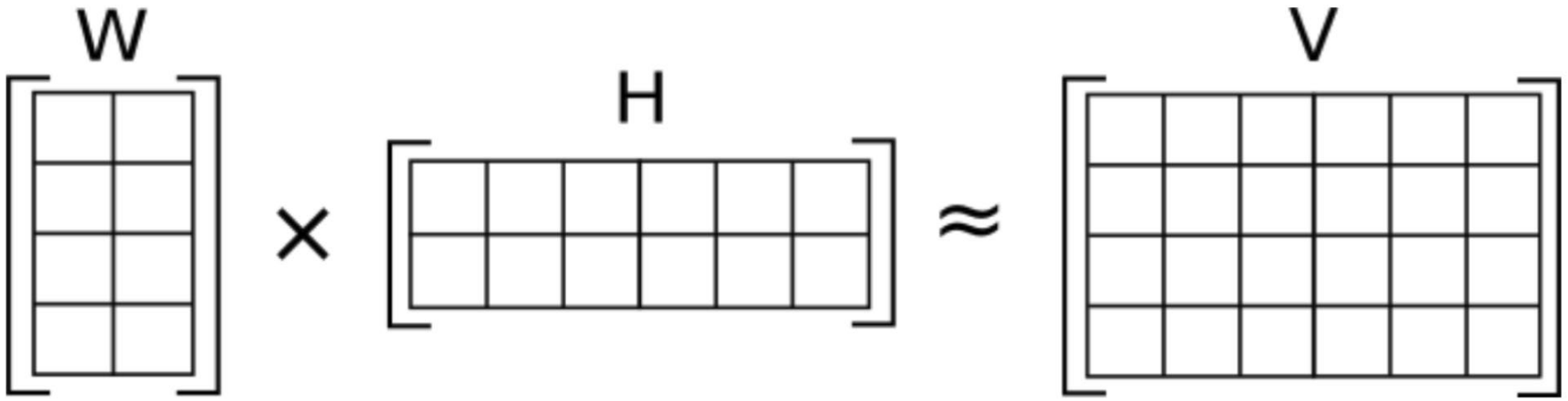



Growth Hackers

# 차원 축소

## 비음수 행렬 분해 (NMF)

- 행렬 인수분해 알고리즘 → 해석 가능한 특징을 추출
- 성분 간의 우열이 있는 PCA와 달리 양수이기만 하면 성분이 우열 없이 특징을 구분할 수 있음




음수 미포함 가중치 행렬
(Weight Matrix)

특성 행렬
(Feature Matrix)

원본 행렬



Growth Hackers

# 차원 축소

## 비음수 행렬 분해 (NMF)

| 책제목 | 협상 | 스타트업 | 투자 | 비지니스 | 데이타 | ... | ... |
|---|---|---|---|---|---|---|---|
| 협상의법칙 | 0.9 | 0 | 0.3 | 0.8 | 0 | | |
| 린스타트업 | 0 | 0.8 | 0.7 | 0.9 | 0.3 | | |
| 빅데이타 | 0 | 0 | 0.5 | 0 | 0.8 | | |

< 그림. 책 제목과, 그 책에 나온 단어의 TFIDF 값으로 이루어진 행렬 V >

# 차원 축소

## 비음수 행렬 분해 (NMF)

행렬 W는 다음과 같은 모양을 가지게 되고

| 책제목 | 특징1 | 특징2 | 특징3 | 특징4 |
|---|---|---|---|---|
| 협상의법칙 | 0.9 | 0 | 0.1 | 0.2 |
| 린스타트업 | 0 | 0.8 | 0 | 0 |
| 빅데이타 | 0.2 | 0.1 | 0.8 | 0.1 |

행렬 H는 다음과 같은 모양을 가지게 된다.

| | 협상 | 스타트업 | 투자 | 비지니스 | 데이타 | ... | ... |
|---|---|---|---|---|---|---|---|
| 특징1 | 0.92 | 0 | 0.1 | 0.2 | 0 | | |
| 특징2 | 0 | 0.85 | 0.5 | 0.3 | 0.3 | | |
| 특징3 | 0 | 0 | 0.3 | 0 | 0.8 | | |
| 특징4 | 0 | 0 | 0 | 0 | 0 | ... | |

# Classification & Clustering

Goal : 유사한 데이터를 같은 그룹으로 묶는 모델 생성,
새로운 instance의 그룹 예측

| 분류(classification) | 군집화(clustering) |
|---|---|
| 주어진 데이터 집합을 이미 정의된 몇 개의 클래스로 구분하는 문제 | 입력 데이터의 분포 특성(입력값의 유사성)을 분석하여 임의의 복수 개의 그룹으로 나누는 것 |
| 입력 데이터와 각 데이터의 클래스 라벨이 함께 제공 -> $\{x_i, y(x_i)\}$ | 클래스에 대한 정보 없이 단순히 입력값만 제공 ->$\{x_i\}$ |
| 숫자인식, 얼굴인식 등 | 영상분리, market segmentation |
| K-Nearest Neighbor<br>Support Vector Machine<br>Bayes Classifier | K-means clustering<br>Hierarchical clustering<br>Gaussian clustering |


분류의 예 군집화의 예



Growth Hackers

# Classification & Clustering

- **'유사성'**을 어떻게 측정할 것인가?

  → 그룹이 잘 나뉘었는지 평가하는 지수에 따라 다름

- 물리적 거리가 가까우면 좋은 군집이야! ⇒ Euclidean distance
- 같은 분포에 속하면 같은 군집이야! ⇒ Mahalanobis distance
- 한 군집 내에서는 밀도가 높을 거야! ⇒ Density

# Classification & Clustering

- 참고
  - Euclidean distance of $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$

$$d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

  - Mahalanobis distance

$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$ for S = x와 y가 따르는 분포의 분산

직관적 의미: 한 점이 어떤 분포에 포함되는지를 알고 싶을 때 분포의 분산에 따라 퍼져있는 정도가 다르므로 Euclidean distance를 분산으로 나눈 scaled distance

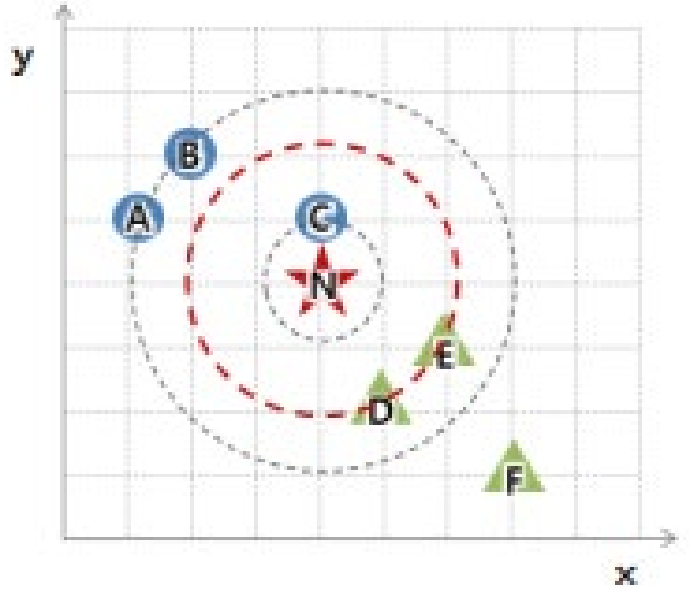
  - Cosine similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$ A=$(A_1, \ldots, A_n)$, B=$(B_1, \ldots, B_n)$

두 벡터가 이루는 각이 유사함의 기준

# Classification_ kNN

- 전제
  - 서로 가까운 점들은 유사하다
  - '가까움'은 정의하기 나름
  - from sklearn.neighbors import KNeighborsClassifier 에서 default  metric  은 minkowski (물리적 거리의 가까움)
- Goal
  - 새로운 instance의 레이블 예측
- How
  - 물리적으로 가장 가까운 k개 데이터의 레이블을 보고 다수결로 정하자.






Growth Hackers

# Clustering_ k-means

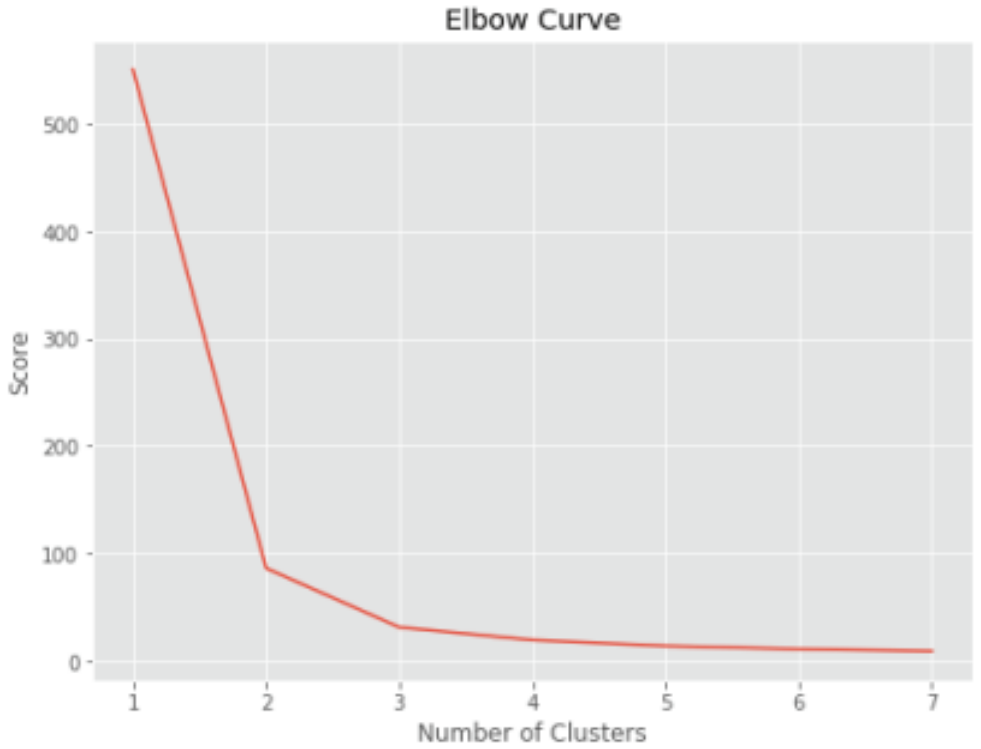
- Goal : k개의 군집으로 데이터를 나누기

1. 임의로 점 k개를 찍고, 각 군집의 중심점으로 잡는다.
2. 각 중심점과 데이터의 거리를 재서 가장 가까운 군집에 배정
3. 배정된 군집이 이전 배정과 같다면, 알고리즘 종료
4. 배정된 군집이 이전 배정과 하나라도 다르면, 배정된 군집 내의 평균을 계산해서 새로운 중심점으로 잡는다. → 2단계

# Clustering_ k-means

- K 선택: 몇 개의 군집으로 나눌 것인가

  ⇒ (중심점~군집 내 데이터)의 제곱합을 작게 하는 'Elbow' 지점의 k 선택

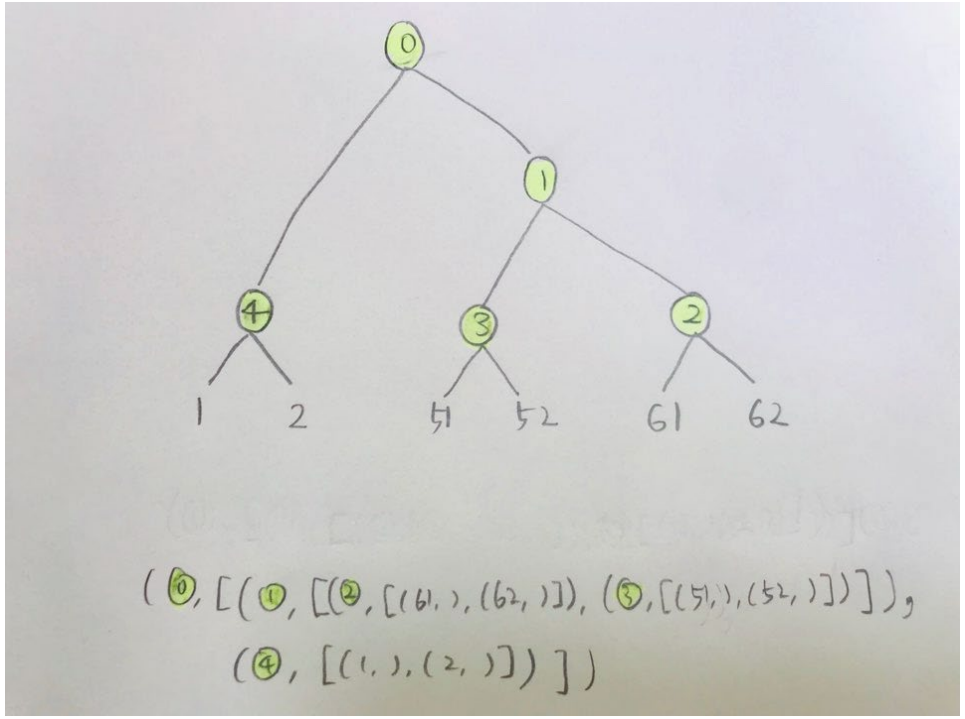





Growth Hackers

# Clustering_ Hierarchical clustering

- 1) 모든 데이터가 각각 개별 군집에 포함된다는 가정으로 시작
- 2) 군집이 두 개 이상이면, 가장 가까운 두 개의 군집을 하나로 묶는다.
- 1 & 2를 원하는 개수의 군집이 남을 때까지 반복






Growth Hackers

# Quest

- 1) 복습 ☺
- 2) 모든 ipynb 한번씩 다 실행한 후 궁금한 점 질문해주세요!

  (전부 이해하셨어도 메시지 보내주셔야 퀘스트 완료입니다)
- 3) 2017.csv
  - 간단한 EDA
  - 데이터 정규화
  - 원하는 두 변수를 고른 후, k-means clustering 실시
    - 적정 k 값 찾기
    - 결과 해석

# Reference


- Scaling VS Normalization (https://kharshit.github.io/blog/2018/03/23/scaling-vs-normalization)
- Feature Scaling with scikit-learn (http://benalexkeen.com/feature-scaling-with-scikit-learn/)
- Dimension / 차원 / 차원의 저주 / 차원축소 (https://kkokkilkon.tistory.com/127)
- 비지도_PCA, NMF, 매니폴드 학습(T-SNE) (https://data-newbie.tistory.com/24)
- PCA (https://datascienceschool.net/view-notebook/f10aad8a34a4489697933f77c5d58e3a/)
- NMF 알고리즘을 이용한 문서 검색과 구현 (https://bcho.tistory.com/1216?category=555440)
- kmeans clustering (https://www.kaggle.com/vjchoudhary7/kmeans-clustering-in-customer-segmentation/notebook?login=true)
- 조엘 그루스, 밑바닥부터 시작하는 데이터 과학, 인사이트
- 정용석, 이정윤(GH 2기), KNN 세션 자료
- 김소정(GH 2기), Clustering 세션 자료
- 배민영 (GH 3기), kNN + Clustering 세션 자료