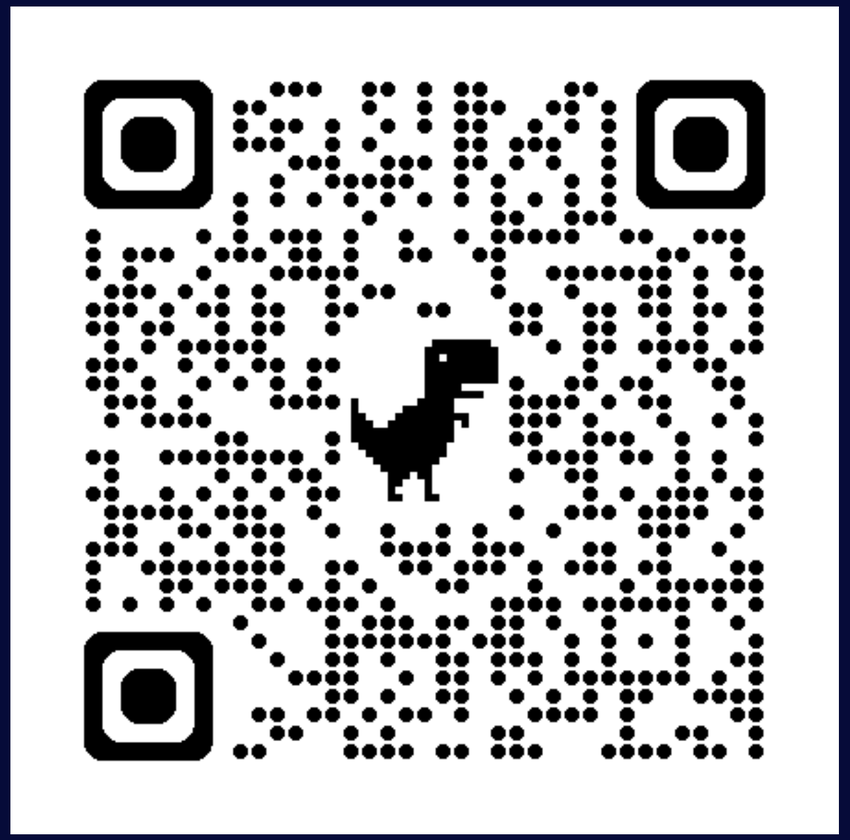
Data Preprocessing

ML Session 4차시

빅데이터 분석 학회 D&A

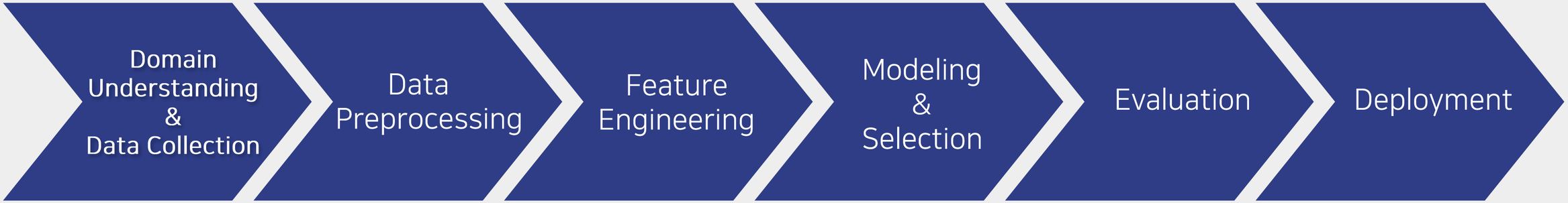
출석체크



CONTENTS.

Overview

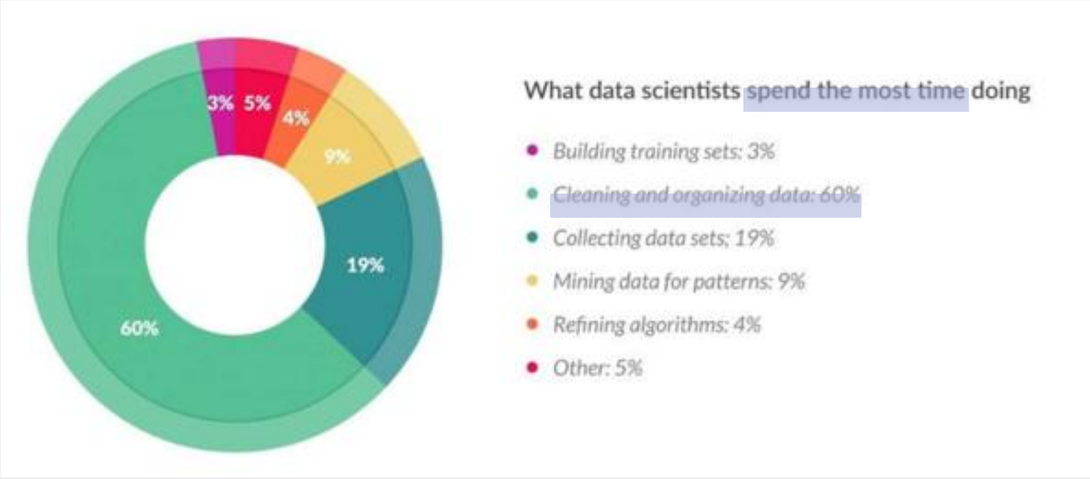
ML Process

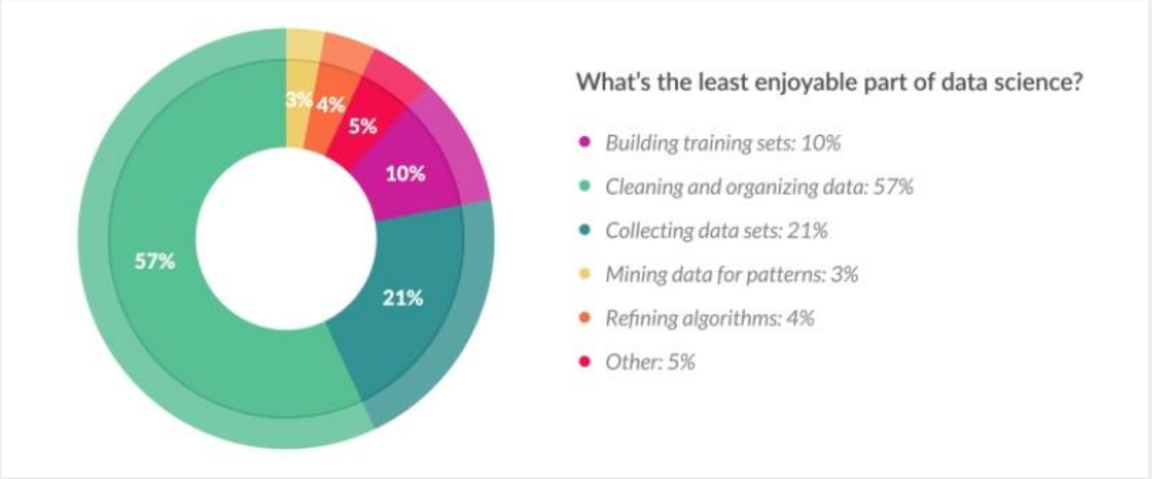


Data Preprocessing이란?

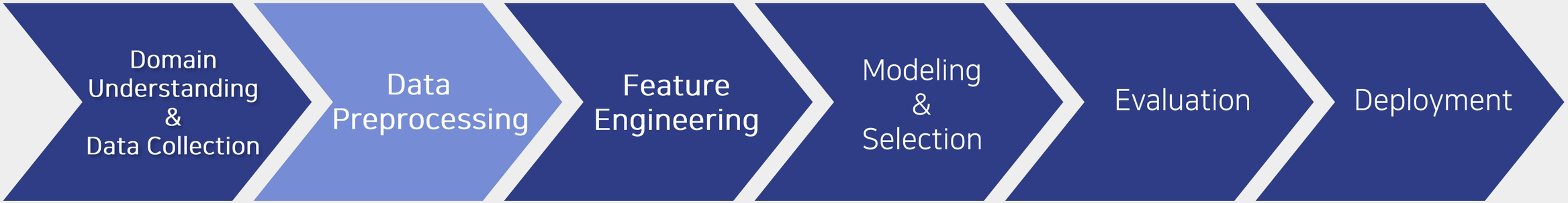
데이터 전처리

- 데이터를 분석하기 좋은 형태로 가공하고 처리하는 과정을 총칭하는 개념
- 머신러닝에서 가장 많은 시간과 노력을 투자해야 하는 단계
- 결측치 처리, 이상치 제거, 데이터 정규화, 데이터 인코딩, 샘플링 및 스케일링 등이 해당
- 데이터를 일관되고 구조화된 형식으로 변환하여 모델이 정확하게 학습하고 예측할 수 있도록 하는 것이 목표

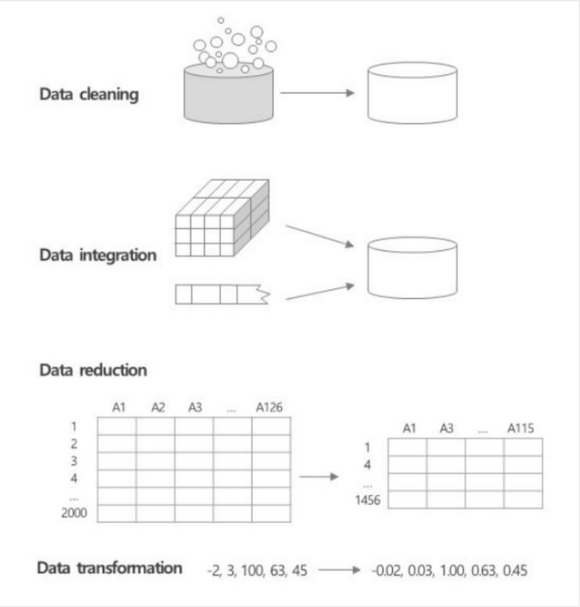




Data Preprocessing 세부 과정



- Data Cleansing: 결측치, 이상치를 수정/제거하여 분석의 신뢰성을 높이는 과정
- Data Integration: 다양한 출처에서 수집한 데이터를 일관된 형식으로 결합
- Data Reduction: 데이터 크기나 차원을 줄여 중요 정보는 유지, 불필요한 데이터는 제거



Data Preprocessing 세부 과정

Data Integration (데이터 통합)

- 다양한 데이터 소스에서 수집된 데이터를 하나의 일관된 형식으로 합치는 작업
- 통합된 데이터를 통해 전체적인 분석을 가능하게 함 → 비즈니스 인사이트를 도출하는 데 기여
- Ex) 여러 지역에서 수집된 판매 데이터와 고객 데이터를 통합하여 하나의 데이터 세트로 통합

Data Reduction (데이터 축소)

- 데이터의 크기나 차원을 줄이는 것을 의미
- 분석에 필요한 중요한 정보만을 유지하면서도, 데이터 전체의 정보 손실을 최소화하기 위한 과정
- 주요 방법
  - 데이터 샘플링 (Data Sampling) : 데이터의 양이나 차원을 줄이는 모든 방법, 전체 데이터의 일부를 사용
  - 차원 축소 (Dimensionality Reduction), 특정 속성 선택 (Feature Selection)

    → 데이터는 피처로 이루어져 있음. 따라서 Feature Engineering 단계로 볼 수도 있음.

Data Cleansing의 목적

Data Cleansing정의 및 목적

- 부정확하거나 불완전한 데이터를 식별하고 수정하는 과정
- 정확하고 신뢰할 수 있는 데이터를 확보하기 위한 과정

  → 일관성 있는 분석과 모델링 품질 향상에 기여

Data Cleansing의 종류

1. 결측치 처리
2. 이상치 탐지 및 처리
3. 중복 데이터 제거

Data Cleansing

결측치 처리

결측치 처리 정의 및 목적

- 결측치가 있는 상태로 모델 생성할 경우, 변수 간 관계에 왜곡 생길 수 있음
- 분석과 모델링의 정확성을 높이고 데이터 손실을 최소화하기 위해 사용
- 결측치가 발생하는 유형 혹은 해당 Feature의 특성에 따라 결측치를 올바르게 처리해야 함
- train 데이터에서 결측치를 처리하는 방법과 동일한 방법으로 test 데이터의 결측치 처리

결측치 처리의 종류

1. 삭제 2. 대체

Data Cleansing

결측치 처리- 삭제

결측치 삭제

- 주로 결측치가 완전 무작위로 발생한 경우에 사용
- ex) 결측치가 변수의 성격과 전혀 무관하게 발생한 경우
- 반면, 무작위 결측치가 아닐 경우, 결측치 삭제는 모델을 왜곡할 수 있으므로 신중해야 함

삭제 방법

- 전체 삭제: 결측치가 발생한 모든 관측치(행) 삭제

  → 간편한 반면, 관측치가 줄어들어 모델의 유효성 낮아짐

  ex) 환자 데이터에서 결측치가 있는 A, B, C, E를 모두 삭제하고 D 환자만 남게 됨.
- 부분 삭제: 분석할/모델에 포함시킬 변수에 결측치가 발생한 경우만 관측치(행) 삭제

  → 분석 주제/모델에 따라 변수가 제각각 다르기 때문에 관리 Cost가 늘어남

  ex) 키와 몸무게 분석 시: B, D만 사용 / 혈압과 몸무게 분석 시: C, D만

환자	키 (cm)	몸무게 (kg)	혈압 (mmHg)
A	175	NaN	120
B	168	75	NaN
C	NaN	80	130
D	182	85	140
E	165	NaN	NaN

Data Cleansing

결측치 처리- 대체(1)

단순 대체(Single Imputation)와 다중 대체(Multiple Imputation)가 있음

단순 대체(Single Imputation)

: 한 가지 값으로 대체하는 경우

(1) 평균값 대체: Column 내 값들의 평균으로 대체

→ 연속형 변수만 사용 가능

(2) 중앙값 대체: Column 내 값들의 중앙값으로 대체

→ 연속형 변수만 사용 가능

(3) 최빈값 대체: Column 내 값들의 가장 많이 나온 값으로 대체

→ 연속형, 범주형 변수 모두 사용 가능

→ Imputer을 사용하기도 함 (단순 대체에서는 SimpleImputer)

Data Cleansing

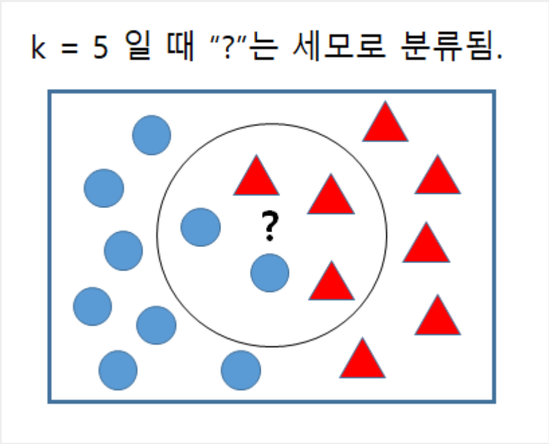
결측치 처리- 대체(2)

여러 값으로 대체하는 경우

결측치가 아닌 데이터들을 train으로 사용해서, Imputation 알고리즘으로 **결측치의 값을 예측**

(1) KNN Imputation (K-Nearest Neighbors)

- K-최근접 이웃 알고리즘(K-Nearest Neighbors, KNN)을 사용하여 결측치를 대체하는 방법
- 문자열(범주형) 데이터는 직접 처리할 수 없음 (수치적 거리를 계산할 수 없기 때문에)
  1. 결측치가 있는 데이터와 가장 가까운 K개의 이웃(n_neighbors)을 찾음
  2. 그 이웃들의 값(평균, 중위수 등)을 이용해 결측치를 대체



(2) MICE Imputation (Multivariate Imputation by Chained Equations)

- 연쇄 방정식(Chained Equations)을 사용하여 대체하는 방법
- 결측치가 있는 각 컬럼의 데이터를 다른 컬럼의 데이터를 활용해 예측
  1. 결측치가 있는 컬럼에 임시로 값을 채움
  2. 결측치가 있는 컬럼에 나머지 컬럼들의 데이터를 기반으로 해당 컬럼의 결측치를 예측
  3. 이 과정을 여러 번 반복하며, 각 컬럼의 결측치를 점차적으로 대체

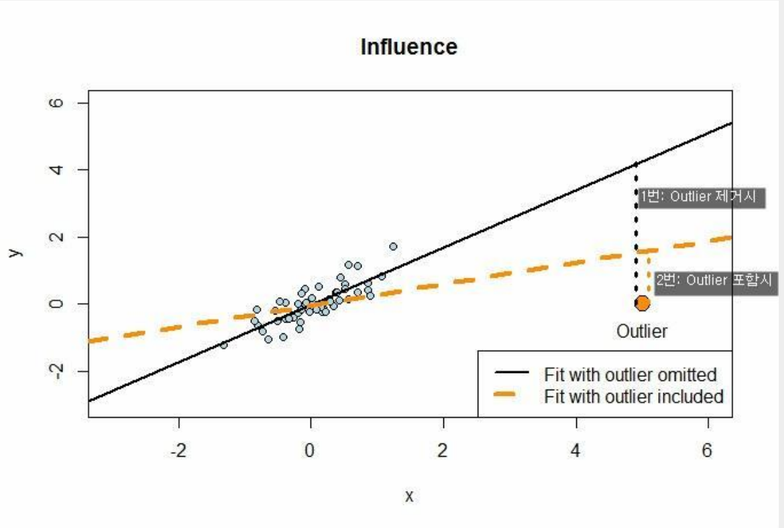
Data Cleansing

이상치 처리

이상치 정의 및 목적

- 값의 범위가 일반적인 범위를 벗어나는 특별한 값
- 회귀 모형의 경우 이상치에 민감하게 반응*

  ex) 특정 지점이 이상치의 영향으로 왜곡을 일으킬 수 있음.
- 데이터의 왜곡을 줄이고 모델의 신뢰성과 예측 정확성을 향상 시키기 위해 사용



이상치 탐지

(1) 시각화: Boxplot, Histogram, Scatter plot

(2) 통계적 기법: Z-score, IQR

이상치 기준 및 처리 방식

(1) : Z-score(표준점수)로 변환　(2) IQR 방식　(3)도메인 지식 이용/ Binning(구간화) 처리 방식

Data Cleansing

이상치 처리

Z-score로 변환

- 표준정규분포로 변환 후, 주로(-3 이하 3 이상)의 값들을 이상치로 판단 후 제거하거나 대체하는 방식

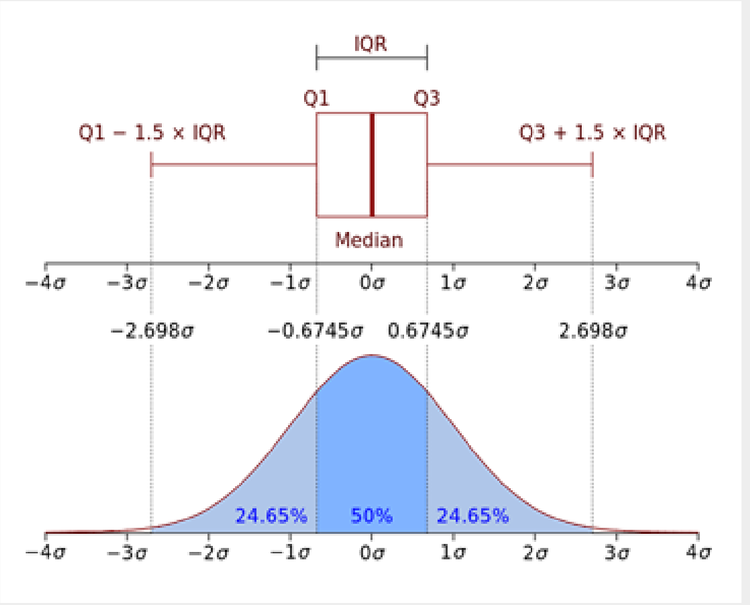
  ⇒ 이상치 = 평균에서 3 표준편차 위의 값들, 평균에서 3 표준편차 아래의 값들 (전체 데이터의 0.3% 미만)
  - 68%의 데이터는 평균에서 ±1 표준편차 안에 존재
  - 95%의 데이터는 평균에서 ±2 표준편차 안에 존재
  - 99.7%의 데이터는 평균에서 ±3 표준편차 안에 존재
- 이는 평균에서 매우 멀리 떨어진 극단적인 값이라는 의미

IQR 방식

1사분위수보다 낮은 IQR의 1.5배를 벗어나는 포인트 or

3사분위수보다 높은 IQR의 1.5배를 벗어나는 포인트의 경우,

이상치로 처리함 (제거 or 대체)

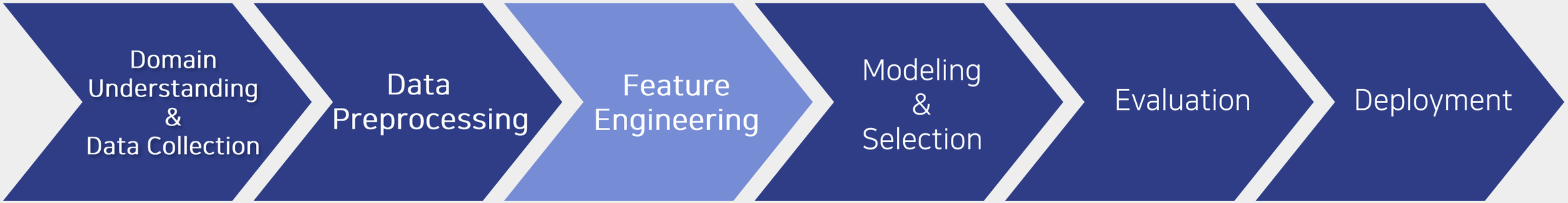


Feature Engineering이란?

Feature Engineering

- 유의미한 변수(피처)를 생성, 변환, 선택하는 과정
- raw data를 모델에 맞도록 변환하여 모델의 성능을 극대화하는 것을 목표로 함
  - 데이터의 품질 향상: 모델이 더 좋은 결과를 내기 위해 데이터를 재구성
  - 예측 정확도 개선: 유의미한 피처를 찾아내어 모델의 예측 성능을 향상시킴
  - 모델 학습 시간 단축: 불필요한 피처를 제거하거나 차원을 축소하여 학습 효율을 높임

Feature Engineering 세부 과정



- Feature Selection: 모델 성능에 영향을 미치지 않는 피처를 제거하여 중요한 피처만 선택
- Feature Transformation: 모델 학습 성능을 향상시키는 형식으로 피처를 변환하는 과정
- Feature Extraction/Generation: 기존 피처를 변형하거나 조합하여 새로운 피처를 생성

Feature Engineering 세부 과정

Feature Selection

- 중요하지 않거나 성능에 악영향을 미치는 피처를 제거하는 과정

→ 5차시 세션에서!

Feature Selection/Extraction

- Feature Selection
  - 기존 Feature들로부터 부분 집합으로 일부 중요한 Feature들만 선택적으로 사용
  - ex) Shap, Lime
- Feature Extraction
  - 기존 Feature에 기반하여 새로운 Feature 생성
  - ex) PCA

Feature Engineering

Feature Transformation의 목적

Feature Transformation 정의 및 목적

- 기존 피처를 변환하여 모델의 성능을 최적화하는 과정
- 피처의 크기나 분포를 균일하게 만들고, 학습 과정에서 특정 피처가 과도하게 영향을 미치지 않도록 함

Feature Transformation의 종류

1. Scaling (스케일링) : 데이터를 일정한 범위로 변환
2. Encoding (인코딩) : 범주형 데이터를 수치형 데이터로 변환
3. 함수 변환 : 지수, 로그, 제곱근 등을 적용하여 데이터 분포를 변형하는 과정
4. Binning (구간화) : 연속형 데이터를 범주형 데이터로 변환하는 과정 ex) 15세, 23세 → 10대, 20대

Feature Transformation - Scaling

Scaling의 정의 및 목적

- 데이터의 범위와 단위를 동일하게 맞춰 모든 변수의 크기를 일정하게 조정하는 과정
- 변수의 분포가 편향되어 있거나 크기 및 단위가 모델 성능에 과도하게 영향을 미치는 것을 방지

Scaling 방법

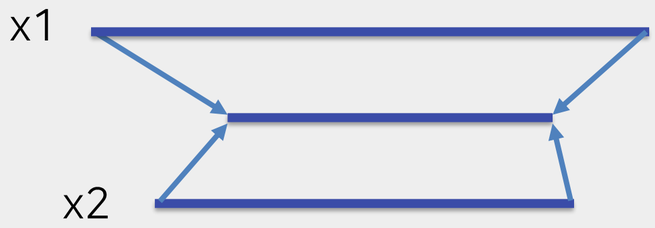
1. MinMax Scaler (Normalization)
2. Standard Scaler (Standardization)
3. Robust Scaler

Feature Transformation - Scaling

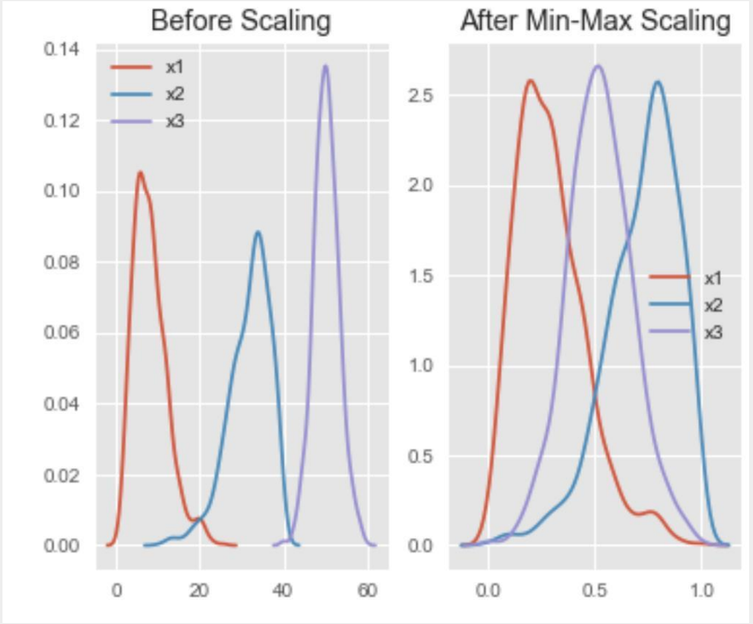
MinMax Scaler(Normalization)

- 개별 Feature의 크기를 모두 똑같은 단위(주로 0-1사이)로 변경하는 것 ⇒ 정규화(Normalization)
- 분산 데이터의 정보는 변형시키지 않는다는 장점이 존재
- 반면, 이상치에 영향을 많이 받는다는 단점 존재

ex) 데이터셋의 범위가 0 ~ 10 사이인 경우에 하나의 이상치가 100의 값을 가진다면,
대부분은 0 ~ 0.01로 변환되지만, 이상치는 1로 변환됨



$$Y = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$



Feature Transformation - Scaling

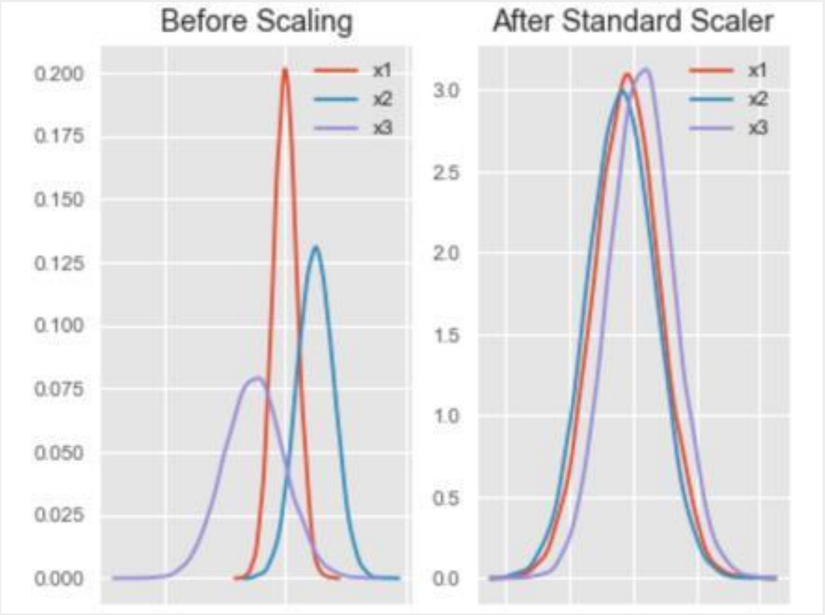
Standard Scaler (Standardization)

- 개별 Feature에서 평균값을 빼고 분산으로 나누어 평균은 0, 분산은 1 (정규분포)로 변환 ⇒ 표준화 (Standardization)
- 정규 분포를 갖도록 변환하는 것은 몇몇 알고리즘*에서 매우 중요

  ex) SVM, Linear Regression, Logistic Regression, Deep Learning
- 각 Feature 사이의 상대적 크기를 왜곡시킬 수 있다는 단점 존재
- 표준화는 각 Feature의 값이 정규 분포를 따를 경우에만 정상적으로 기능을 발휘
- 평균과 표준편차를 기준으로 데이터를 변환하기 때문에 이상치에 취약함

  → 이상치 탐지에서도 활용 가능 (z-score 변환에 해당)

$$Y = \frac{(X - X_{mean})}{\sigma_Y}$$



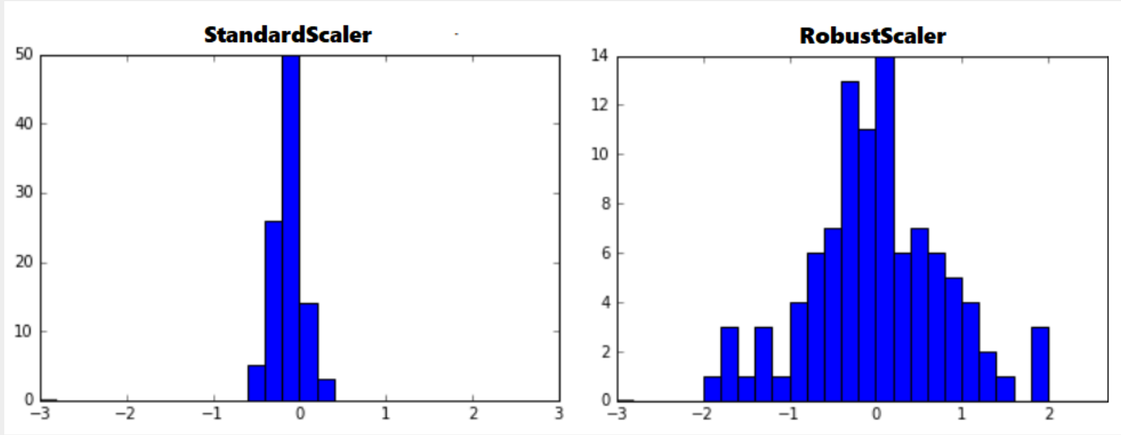
Feature Transformation - Scaling

Robust Scaler

- 개별 Feature 값에서 median을 빼고 IQR 범위로 나누는 것
- 각 Feature의 범위는 Min-Max 보다 작은 편
- 상대적으로 이상치에 덜 민감하게 처리함

  median이 (혹은 mean) IQR의 편차(75%-25%=50%)는 이상치의 영향을 상대적으로 덜 받기 때문

$$Y = \frac{(X - X_{median})}{(X_{IQR,75\%} - X_{IQR,25\%})}$$



Feature Transformation - Encoding

Encoding 정의 및 목적

- 데이터를 컴퓨터가 이해하고 처리할 수 있는 형태로 변환하는 과정
- Encoding = 코드화 = 암호화
- 데이터를 약속된 규칙에 따라 컴퓨터가 이해할 수 있는 숫자로 변환
- 주로 범주형 변수를 수치형 변수로 변환

Encoding 종류

1. Label Encoding
2. One-hot Encoding
3. Target Encoding(=Mean Encoding)

Feature Transformation - Encoding

Label Encoding

- 카테고리 Feature를 숫자로 변환 (ex. Apple → 1, Chicken → 2 ...)
- 모델은 숫자를 기반으로 연산 → 서열 변수가 아닌 경우 치명적임

1. **서열 변수일 경우** 더 단순한 데이터로 인식
   ex) (맛)맵다(1) > 안맵다(0)
2. **서열 존재 가능성**으로 잘못 인식
   ex) 0<1<2<3 → 딸기<바나나<사과<포도

	과일
0	바나나
1	사과
2	사과
3	포도
4	딸기

```
sorted(set(df['과일']))
```

```
['딸기', '바나나', '사과', '포도']
```

0 1 2 3

```
encoder = LabelEncoder()
labels = encoder.fit_transform(fruit)
df['label']=labels
df
```

	과일	label
0	바나나	1
1	사과	2
2	사과	2
3	포도	3
4	딸기	0
5	포도	3
6	바나나	1

Feature Transformation - Encoding

One-hot Encoding

- Feature의 고유값에 해당하는 Column에만 1, 나머지는 0으로 표현
- 차원의 저주에 걸릴 수 있음 (sparse하기 때문)

  Ex) 어떤 Column 내의 값이 100가지라고 가정,

  one-hot encoding을 진행했을 때 100가지의 column이 생성

  → 데이터프레임의 0으로 채워지는 부분이 많아짐(1이 상대적 희소해짐을 의미 = 데이터를 포함하는 부분이 적어짐)

Numerical value	Animal
1.5	cat
3.6	cat
42	dog
7.1	crocodile

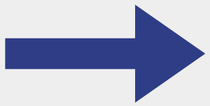
Numerical value	Cat	Dog	Tiger	Crocodile
1.5	1	0	0	0
3.6	1	0	0	0
42	0	1	0	0
7.1	0	0	0	1

Feature Transformation - Encoding

Target Encoding (= Mean Encoding)

- Label Encoding과 유사하지만, Target값과 Encoding 값이 연관이 있다는 점에서 차이가 존재
- 각 카테고리의 값을 학습 데이터의 Target값의 평균값으로 설정하는 방법

```
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64
```



	Sex	Sex_mean
0	male	0.188908
1	female	0.742038
2	female	0.742038
3	female	0.742038
4	male	0.188908

Feature Transformation - Encoding

Target Encoding의 장단점

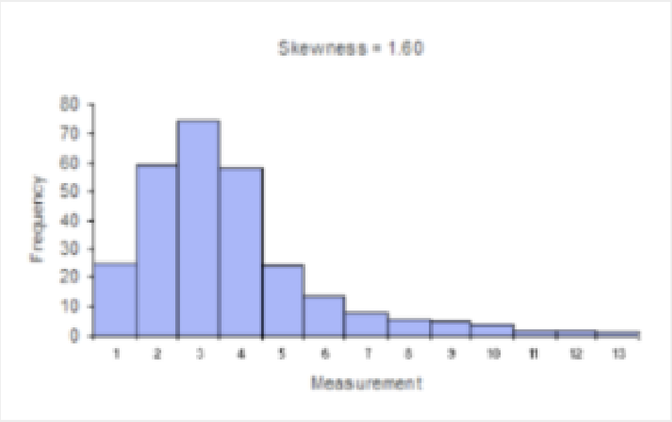
- 장점
  - 카테고리의 개수가 많을수록, Label Encoding은 Label 수가 계속 증가
  - Target Encoding은 보다 적은 split이 생기고 학습이 더욱 빠르게 이루어짐
  - Encoding된 Label값이 Target과 관련된 의미를 가짐 → less bias
- 단점
  - Overfitting 가능성 높음
  - 구현과 검증이 까다로움(*Data leakage)

Feature Transformation – 함수 변환

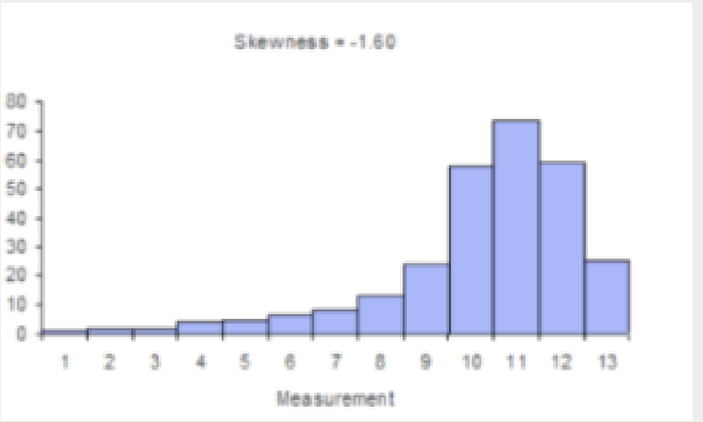
Skewness (왜도)

- 분포의 정규분포에 비해서 얼마나 비대칭인지 나타내는 척도
- 왜도 값이 양의 값 - 정규 분포보다 오른쪽에 위치
- 왜도 값이 음의 값 - 정규 분포보다 왼쪽에 위치
- Positive Skew(왼쪽으로 치우쳐짐)를 정규분포로 변환

Positive Skew



Negative Skew



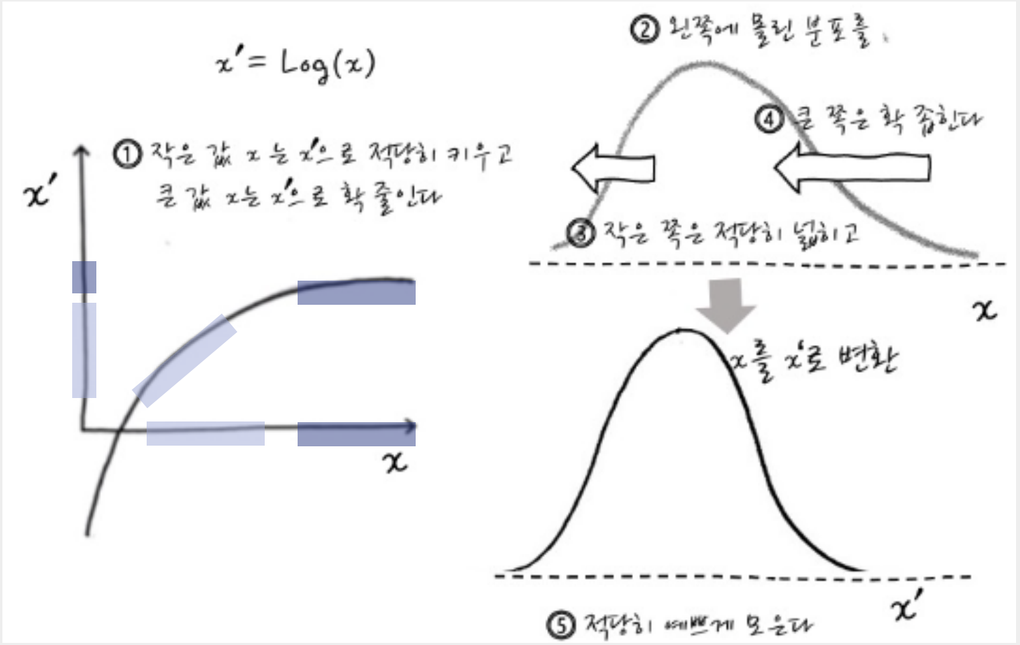
변환하는 이유?

- 꼬리에 있는 값을 모델에 제대로 학습시키기 위함
- 꼬리 부분이 상대적으로 데이터의 양이 적기 때문에 모델 학습에 반영이 적게 됨

Feature Transformation – 함수 변환

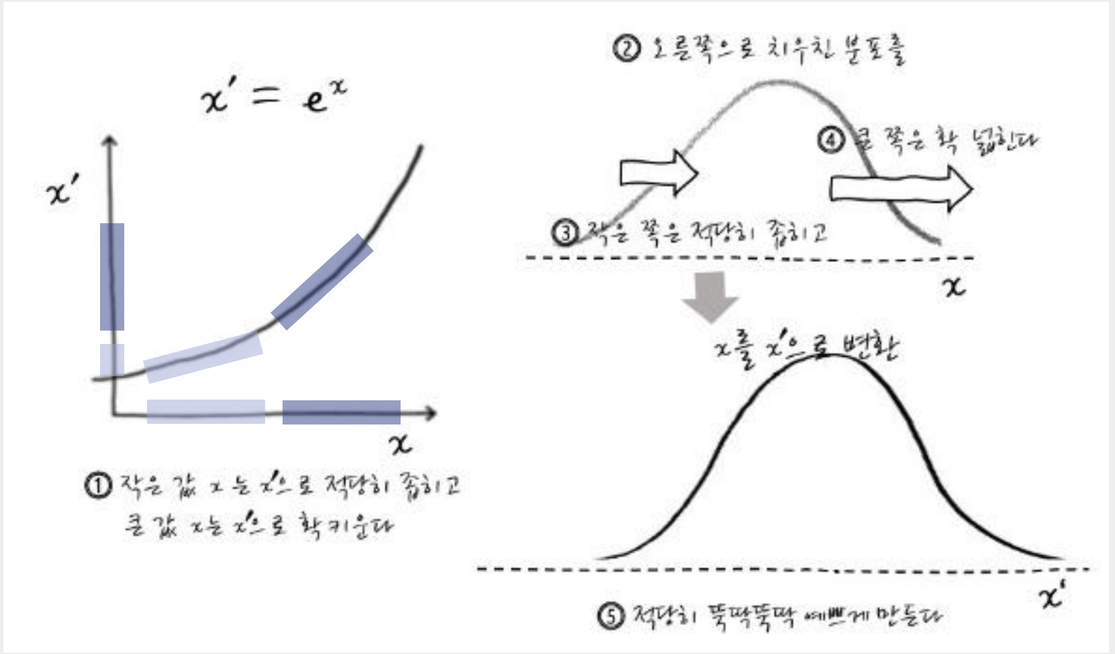
로그 (Log) 변환

- 로그 변환, 루트 변환, 역수 변환: 큰 숫자를 줄여줌.
- Positive Skew(왼쪽으로 치우쳐짐)를 정규분포로 변환



지수 (Exp) 변환

- 지수 변환(로그 함수의 역함수): 작은 숫자를 늘려줌.
- Negative Skew(오른쪽으로 치우쳐짐)를 정규분포로 변환



Feature Extraction의 목적

Feature Extraction 정의

- 많은 feature를 만들고 유의미하다고 판단되는 feature를 feature selection을 통해 골라서 사용
- 모델이 값을 잘 예측할 수 있는 유의미한 feature를 제공해야 성능이 좋은 모델을 만들 수 있음

Feature Extraction과 Selection의 사용 이유

- 데이터의 특성 중, 모델에 중요한 정보를 제공하지 않거나 중복되는 특성들이 존재
- 따라서 중요한 Feature를 선택 or 기존 Feature의 특징 추출 등 차원을 축소하여 사용
- 차원의 저주
- 모델의 속도 증진, 과적합 위험 감소 (차원 감소 → 모델의 복잡도 감소), 모델의 간결함 등의 이점을 얻기 위함

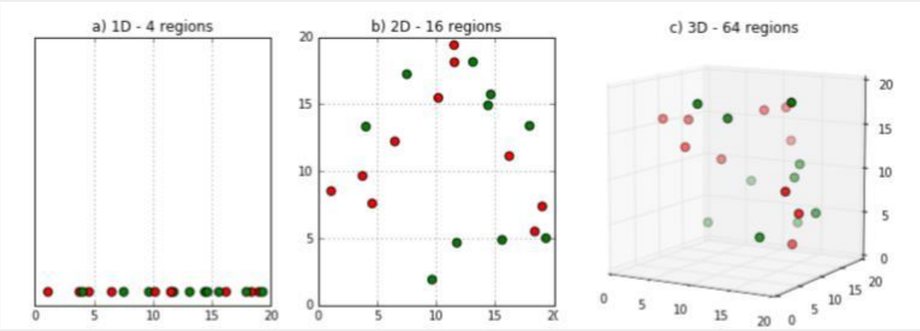
Feature Extraction

차원의 저주

차원의 저주 정의

- 데이터의 차원이 증가함수록 필요한 데이터 양이 기하급수적으로 늘어나는 현상
- 차원이 증가하면서 학습 데이터 수가 부족해져 적어도 성능이 저하됨
- 관측치보다 변수가 많아지는 경우, 차원의 저주 문제가 발생

  (변수가 증가한다고 반드시 차원의 저주가 발생하는 것은 아님)



차원의 저주 특징

1. 차원의 저주 특징데이터의 양(행)은 동일한데, 데이터의 차원(열)이 커지면 데이터의 밀도가 떨어짐
   - 차원이 커질 수록 데이터 간 거리가 멀어짐
   - 빈 공간이 많이 생기게 되는데, 이는 정보가 없는 공간이라고 할 수 있음
   - 따라서, 빈 공간이 많은 데이터에 대해 학습을 하게 되면 모델 성능이 저하될 수밖에 없음
2. 원하는 정보를 찾는 데 Computing Cost가 많이 소요

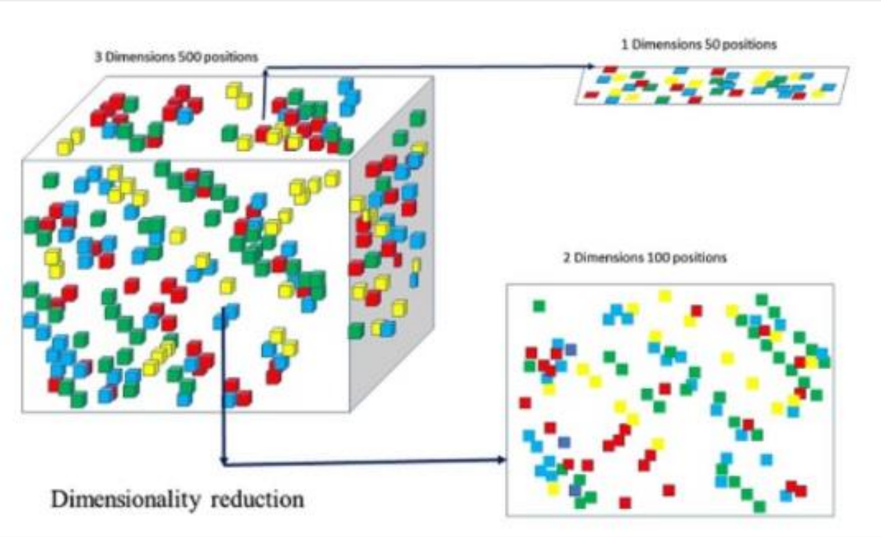
   → 데이터의 차원을 낮춰서 학습을 진행 (데이터 차원을 낮추는 방법: Extraction, Selection)

Feature Extraction

PCA (주성분 분석)

PCA의 정의

- 주성분 분석이라고도 함
- 고차원의 데이터를 저차원의 데이터로 축소시키는 것이 목표!
- 분산이 최대한 보존되는 축을 선택하는 것이 정보가 가장 적게 손실되므로 중요함
- One-hot-Encoding 같이 실질적인 데이터는 적은데, 차원은 많은 Sparse data 등을 줄이기 위한 방법으로도 사용된다!



Feature Extraction

PCA (주성분 분석)

PCA의 장점

- 시각화 3차원이 넘어간 시각화는 우리 눈으로 볼 수 없기 때문에 PCA를 통해 차원을 축소하여 시각화 → 데이터 패턴을 쉽게 인지 가능
- 노이즈 제거
- 설명할 수 없는 Feature를 제거함으로써 노이즈 제거 가능
- 메모리 절약 및 퍼포먼스 향상
- 불필요한 Feature를 제거해 모델 성능 향상에 기여

과제

과제 안내

D&A_2024_ML_4주차_과제.ipynb에 있는 문제를 풀고,
10월 14일(월) 23:59까지 홈페이지에 제출해주세요.

THANK YOU

ML Session 4차시

빅데이터 분석 학회 D&A