ML SESSION

#4
Data Preprocessing

DATA&ANALYTICS

INDEX

0

우수과제팀

팀명	모델성능
교수님저희싫어하시조	1.399
과적합의노예조	1.416
도와조	1.426
에러났조	1.429

1 Data Preprocessing 이란?

STAGE 1	STAGE 2	STAGE 3	STAGE 4	STAGE 5	STAGE 6
Domain Understanding and Data Collection	Data Preprocessing	Modeling and Ensemble	Prediction	Evaluation	Deployment

1 Data Preprocessing 이란?

STAGE 1	STAGE 2	STAGE 3	STAGE 4	STAGE 5	STAGE 6
Domain Understanding and Data Collection	Data Preprocessing	Modeling and Ensemble	Prediction	Evaluation	Deployment

재료 준비 과정

1 Data Preprocessing 이란?

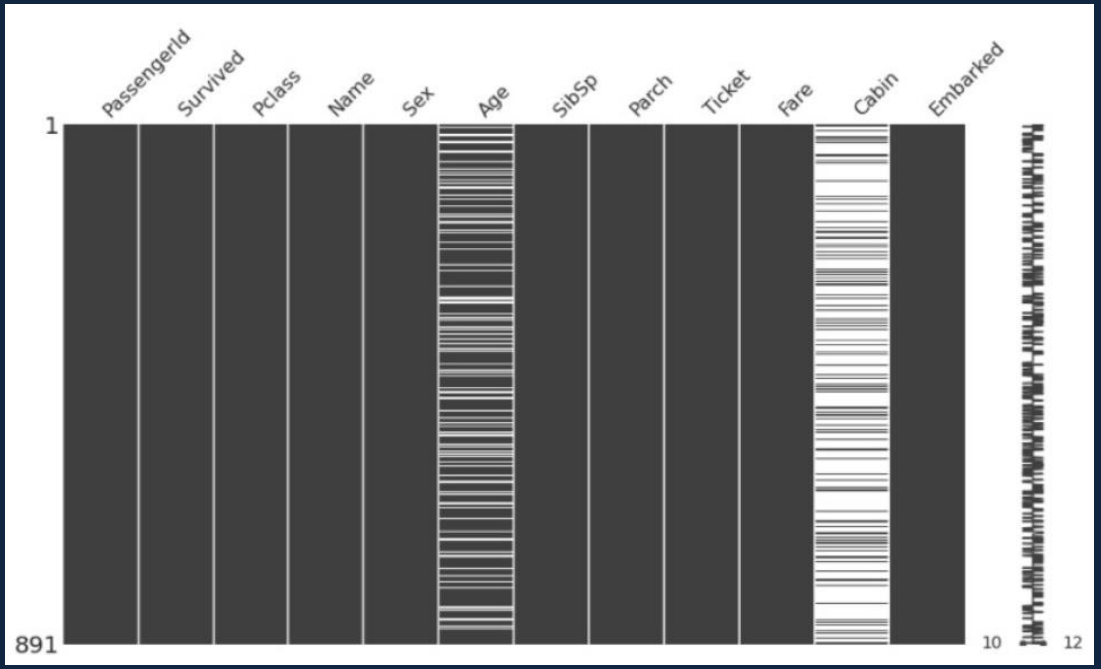
50% 이상

STAGE 1	STAGE 2	STAGE 3	STAGE 4	STAGE 5	STAGE 6
Domain Understanding and Data Collection	Data Preprocessing	Modeling and Ensemble	Prediction	Evaluation	Deployment

- **Data Cleaning**
  노이즈 제거
  이상치
  결측치 처리
- **Data Integration**
  다수의 데이터 소스들 병합
- **Feature Construction / Extraction**
  원본 데이터에서 원하는 feature 가져오기
- **Feature Selection**
  필요 없는 피쳐는 버리고 관련 있는 피쳐만 고르기
- **Scaling**
  데이터의 특성에 맞춰 scaling 진행
- **Sampling**
  모델을 훈련시키기 위한 train, valid, test set을 샘플링

2 결측치 처리

결측치 처리란?



<Titanic Data 에서 결측값의 개수>

- 사이킷런 패키지는 NaN값을 허용하지 않음
  -> ML알고리즘을 적용하기 전 모든 결측치를 처리해야 함

- 각 피쳐의 특성에 맞게 결측치를 처리해야함

  - Numeric?
  - Categoric?

2 결측치 처리:제거

결측치 확인

```
DF.isnull( ).sum( )          -> 결측치가 존재하는 열, 개수 확인
```

결측치 제거

```
DF.dropna(axis = 0)      -> 결측치가 들어간 행 제거
DF.dropna(axis = 1)      -> 결측치가 들어간 열 제거
```

inplace = True
-> 자동 대체

⚠ 주의

결측치 제거는 함부로 하지 말 것
특히나 해당 Feature에서 결측치 비율이 10%를 넘어간다면 하지 않는 것을 추천

2 결측치 처리:대체

Imputation – single

평균

Column 내 값들의 평균으로 결측치를 대체
연속형 변수만 사용가능

중앙값

Column 내 값들의 중앙값으로 결측치를 대체
연속형 변수만 사용가능

최빈값

Column 내 값들 중 가장 많이 나온 값으로 결측치를 대체
연속형, 범주형 모두에서 사용 가능

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean( )

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

⚠ 주의

single column 대체는 다른 변수와의 관계를 생각하지 않고, 하나의 값으로만 대체하기 때문에 결측값이 많을 때에는 다른 방법을 추천

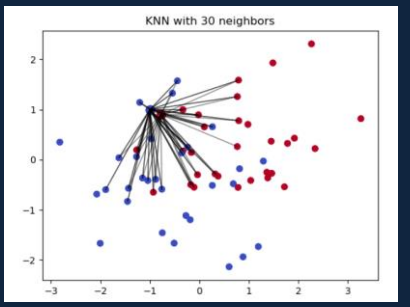
2 결측치 처리:대체

Imputation - multiple

결측치가 아닌 데이터들을 train으로 두고 model을 돌려 값을 예측

KNN Imputation

KDTree를 구성한 후 NN(최근접 이웃)을 계산해
k-NN을 찾은 후 가중 평균 취함



MICE

연쇄 방정식을 이용한 대치
누락된 데이터를 여러 번 채우는 방식으로 작동

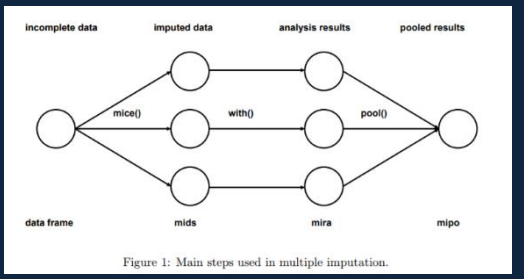


Figure 1: Main steps used in multiple imputation.

[ETC) Mice, Amelia , MissForest, Hmisc, Mi ...]

2 결측치 처리:가이드

적절성 확인

1. 하나라도 결측이 있는 변수를 제외한 dataset을 생성
2. imputation한 dataset 2개 정도를 생성 (방법 무관)
3. 1)과 2)의 dataset 3개에 대해서 결과값이 서로 일관성이 있음을 보여줌

   → 결측치 대치 방법이 sensitive하지 않다는 것을 제시

가이드 라인

- 10% 미만 : 삭제 or 대치
- 10 ~ 20% : Hot deck (매년자료->해당년자료 추정) or regression or model based imputation
- 20 ~ 50% 이상 : regression or model based imputation
- 50% 이상 : 해당 칼럼(변수)자체 제거

범주형 변수 처리

One Hot Encoding

피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표시하고 나머지 칼럼에는 0을 표시하는 방식

color
red
green
blue
red

one-hot encoding →

color_red	color_green	color_blue
1	0	0
0	1	0
0	0	1
1	0	0

3 이상치 처리

Outlier?

값의 범위가 일반적인 범위를 벗어나 특별한 값을 갖는 것

- why? 회귀모형의 경우 이상치 값에 민감하게 반응하기 때문

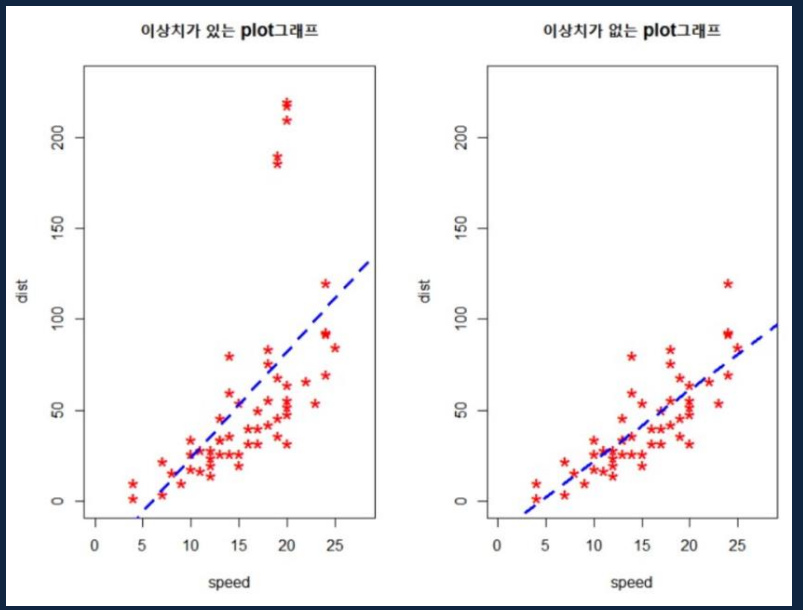
그렇다면 이상치의 범위를 어떻게 설정해야 할까?

- 분석가에 따라 다르지만 일반적인 형태 존재

1) 표준점수로 변환

2) IQR 방식

3) 도메인 지식 이용이나 Binning 처리 방식

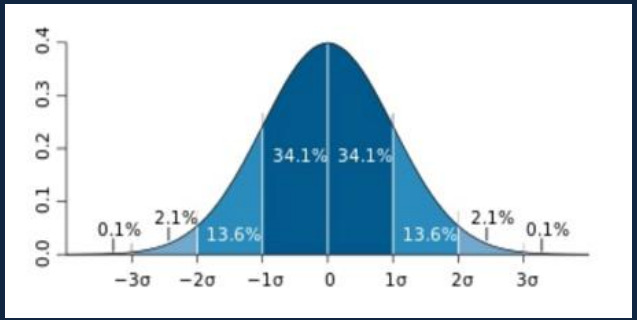


Notice

반드시 이상치 처리를 할 필요는 없다 모델에 따라, 스케일링에 따라
의도적으로 이상치를 처리하지 않을 수도 있음

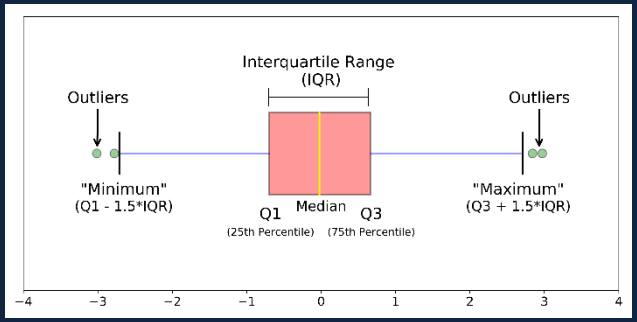
3 이상치 처리

표준점수 변환



표준정규분포로 변환 후 -3 이하 및 3 이상 값들을 이상치로 판단 후 제거 하거나 대체

IQR 방식



1사분위수보다 낮은 IQR의 1.5배를 벗어나는 포인트 혹은 3사분위수 보다 높은 IQR의 1.5배를 벗어나는 포인트를 이상치로 처리(제거 or 대체)

4 Scaling

Scaling이 왜 중요한가?

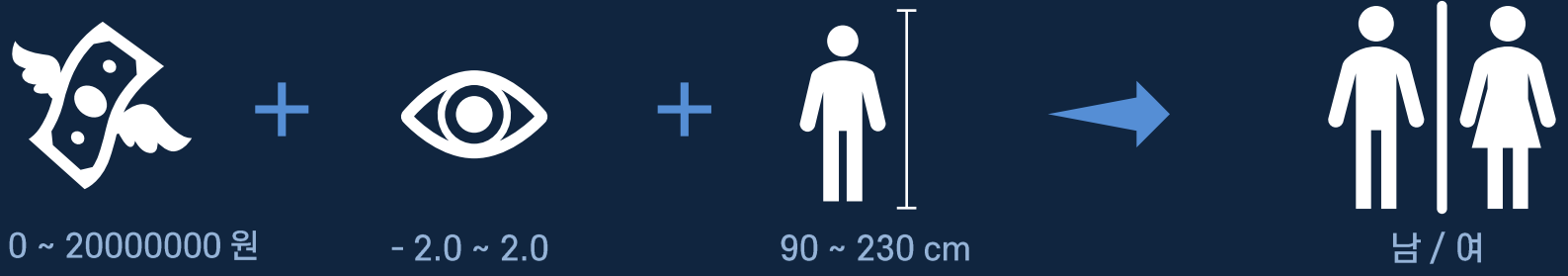
Scaling: 서로 다른 변수의 값 범위 혹은 분포를 일정한 수준으로 맞추어 주는 행위

why?

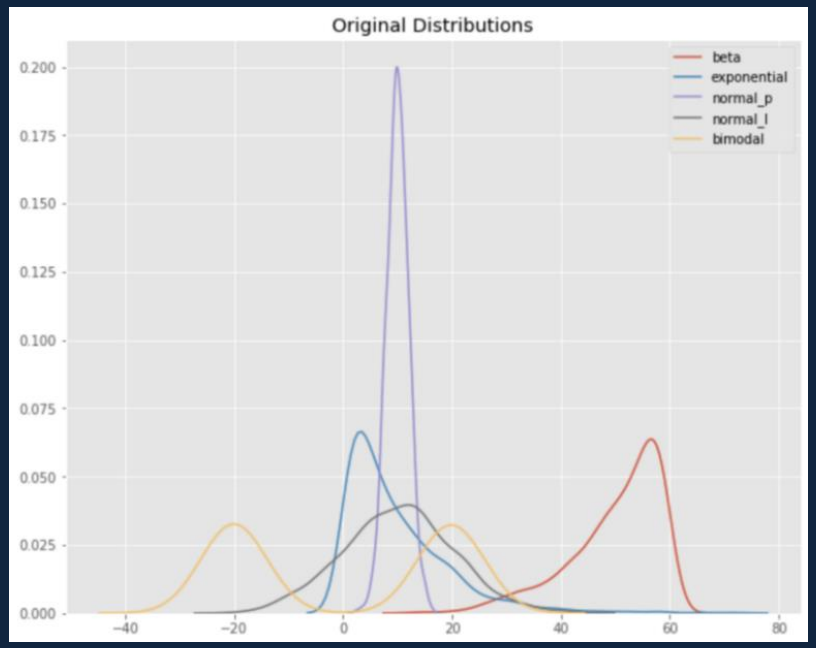
머신 러닝 알고리즘은 숫자를 동일하게 이해함

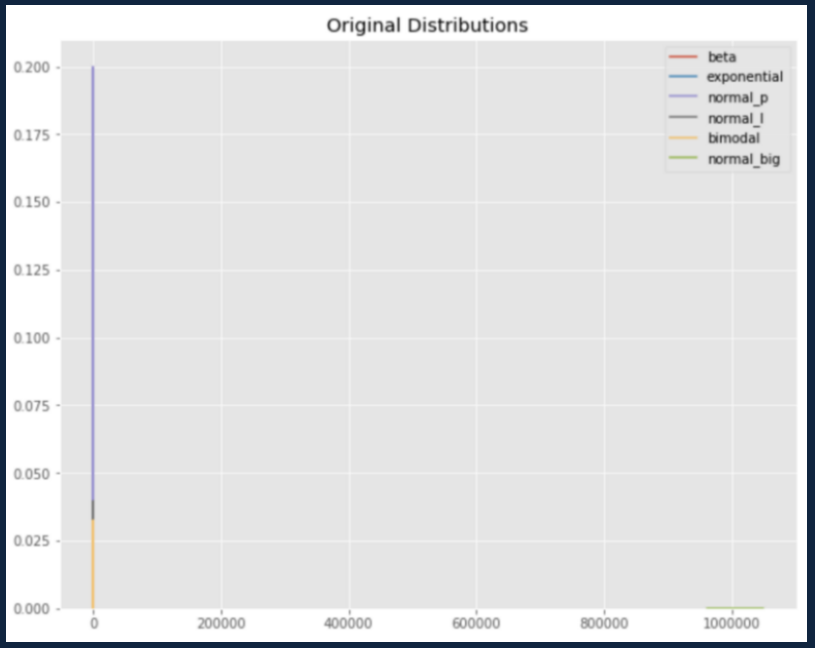
경사하강법을 더 빨리 진행 할 수 있음

EX)



4 Scaler의 종류





<Positive Skew> <Negative Skew>

	beta	exponential	normal_p	normal_l	bimodal	normal_big
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03
mean	50.024249	10.028387	9.994006	10.175745	-0.076622	1.000259e+06
std	8.474545	9.733928	2.013971	10.104004	20.165208	9.935564e+03
min	13.854022	0.007617	2.356844	-19.539980	-28.709856	9.692079e+05
25%	45.793283	2.951421	8.687478	3.566822	-19.995311	9.936191e+05
50%	52.337504	7.018565	9.983498	10.326331	0.237049	1.000241e+06
75%	56.722191	14.022485	11.306914	16.615057	19.891202	1.007335e+06
max	59.990640	71.344341	16.214364	42.072915	28.252151	1.040677e+06

Scale이 다름

4 Scaler의 종류

StandardScaler

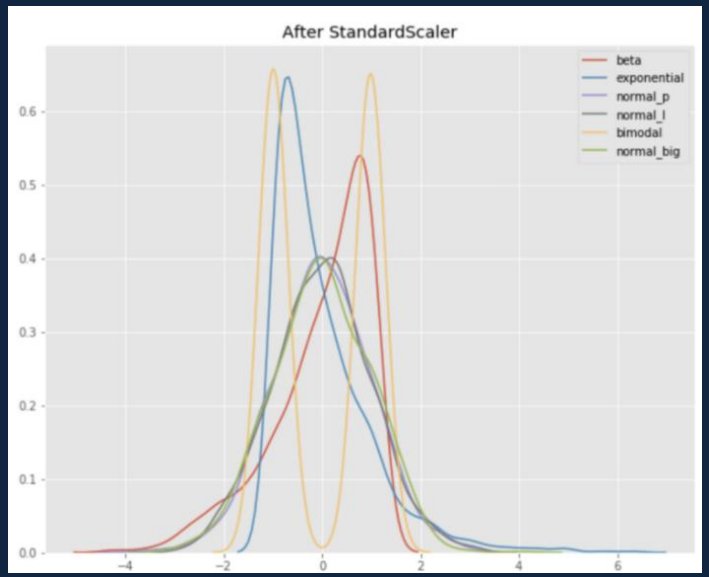
개별 feature에서 평균값을 빼고, 분산을 나누어 평균은 0, 분산은 1로 변환 – Standardization

가우시안 정규 분포를 갖도록 변환하는 것은 몇몇 알고리즘에서 매우 중요

ex) SVM, Linear Regression, Logistic Regression, Deep Learning

각 feature들 사이의 상대적 거리를 왜곡시킬 수 있다는 단점

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



4 Scaler의 종류

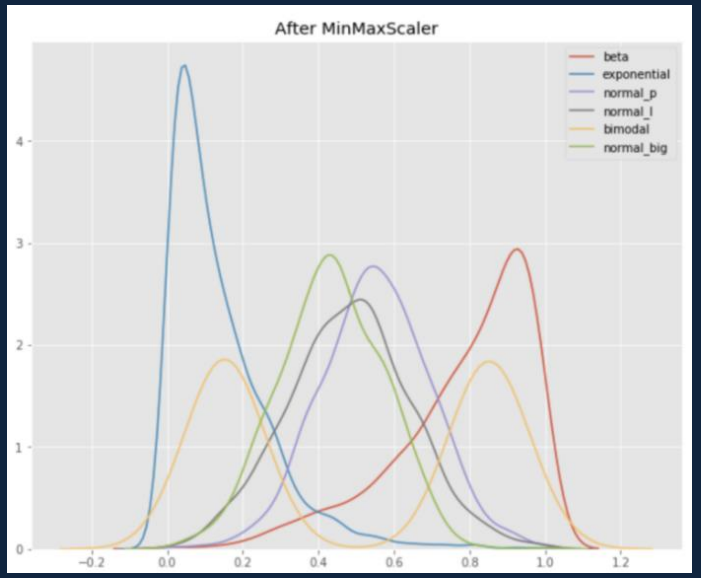
MinMaxScaler

개별 feature의 크기를 모두 똑같은 단위(0에서 1 사이)로 변경하는 것 – Normalization

본래 데이터의 정보를 변형시키지 않는다는 장점

이상치에 영향을 많이 받는다는 단점

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



4 Scaler의 종류

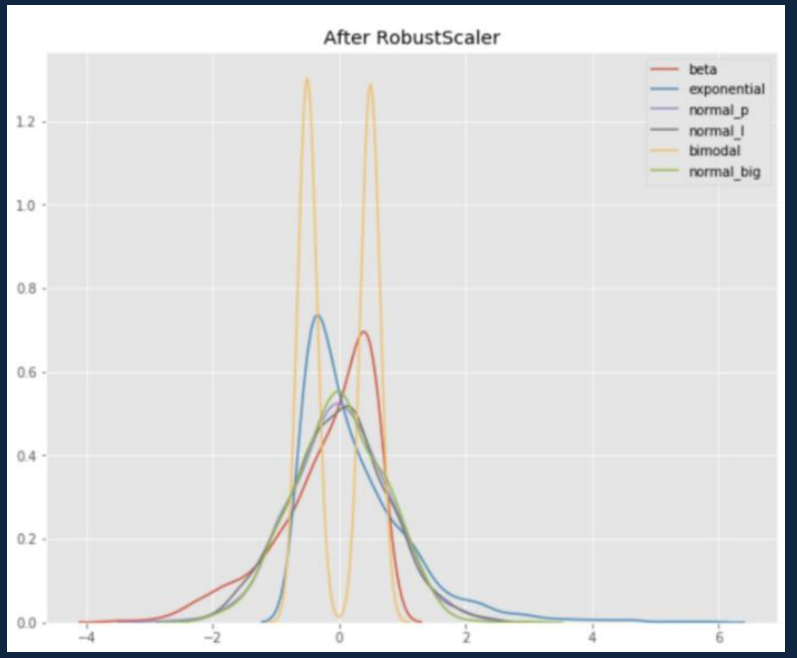
Robust Scaler

개별 feature값에서 median을 빼고 IQR 범위로 나눈 것

각 feature의 범위는 MinMax 보다는 큼

상대적으로 이상치의 효과를 줄이기에 적합

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



4 Scaler의 종류

Normalizer

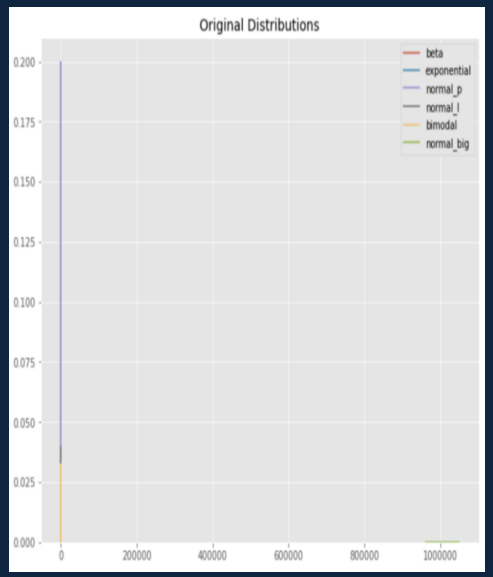
선형대수에서의 정규화 개념이 차용되어 일반적 정규화와는 약간의 차이 존재
각 feature의 열(column)값이 아닌 행(row)값에 적용되는 scaler
대부분의 경우 위에 이전에 언급된 것들이 효율적임

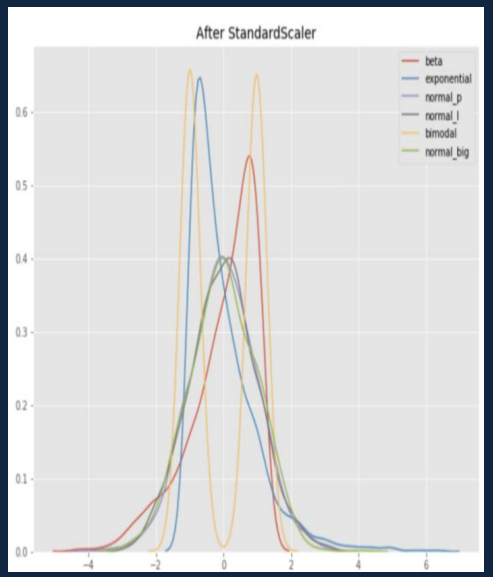
$$Y_i = \frac{\left(X_i\right)}{\sqrt{\left(\sum_{j=1}^{N} X_j^{\,2}\right)}}$$

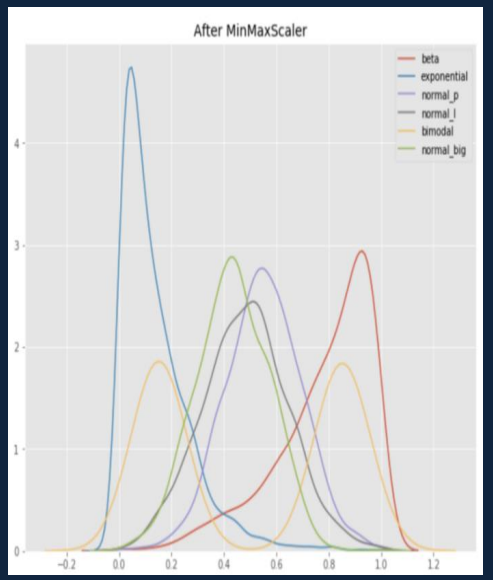
4 Scaling

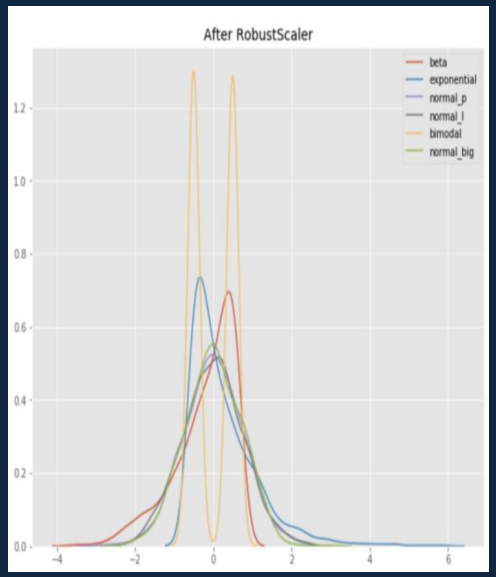
정리

- 데이터의 왜곡이 없이 순수하게 분포를 비교하고자 하면 MinMax
- 이상치가 존재하고 그 영향을 줄이고 싶다면 Robust
- 모든 데이터의 분포를 정규분포로 보고싶다면 Standard
- 다양한 Scaler를 섞어서 시행하는 것도 고려해볼 사항



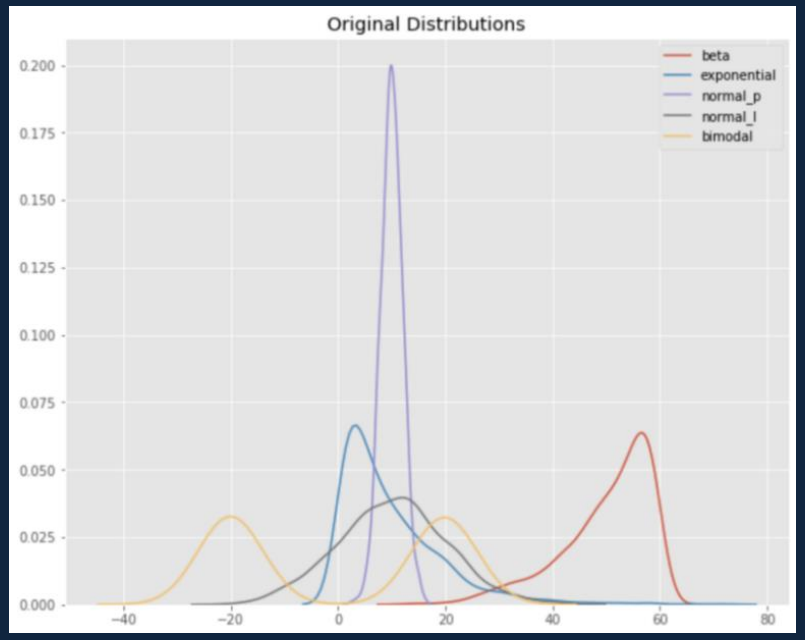




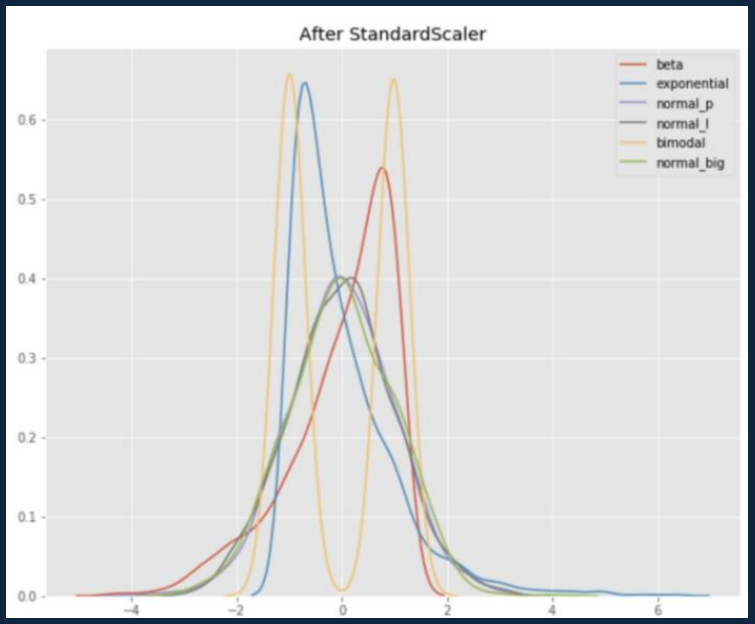


+ Transformation

skew

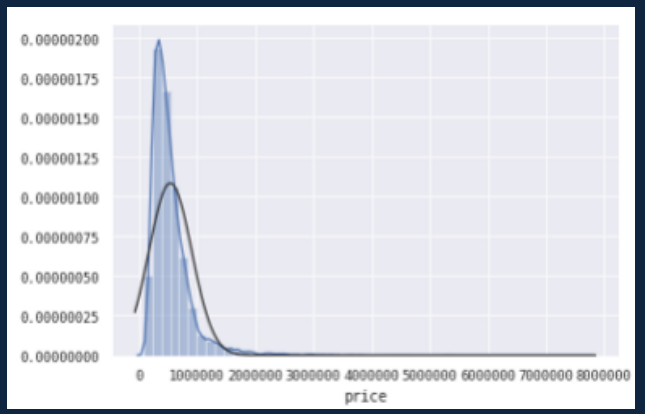


<Positive Skew> <Negative Skew>



Skewed data (왜도가 높은 데이터)에 대해 Scaling을 적용해도 여전히 치우친 분포라는 문제

Transformation



<원본 데이터 분포>

<log 변환 후 데이터 분포>

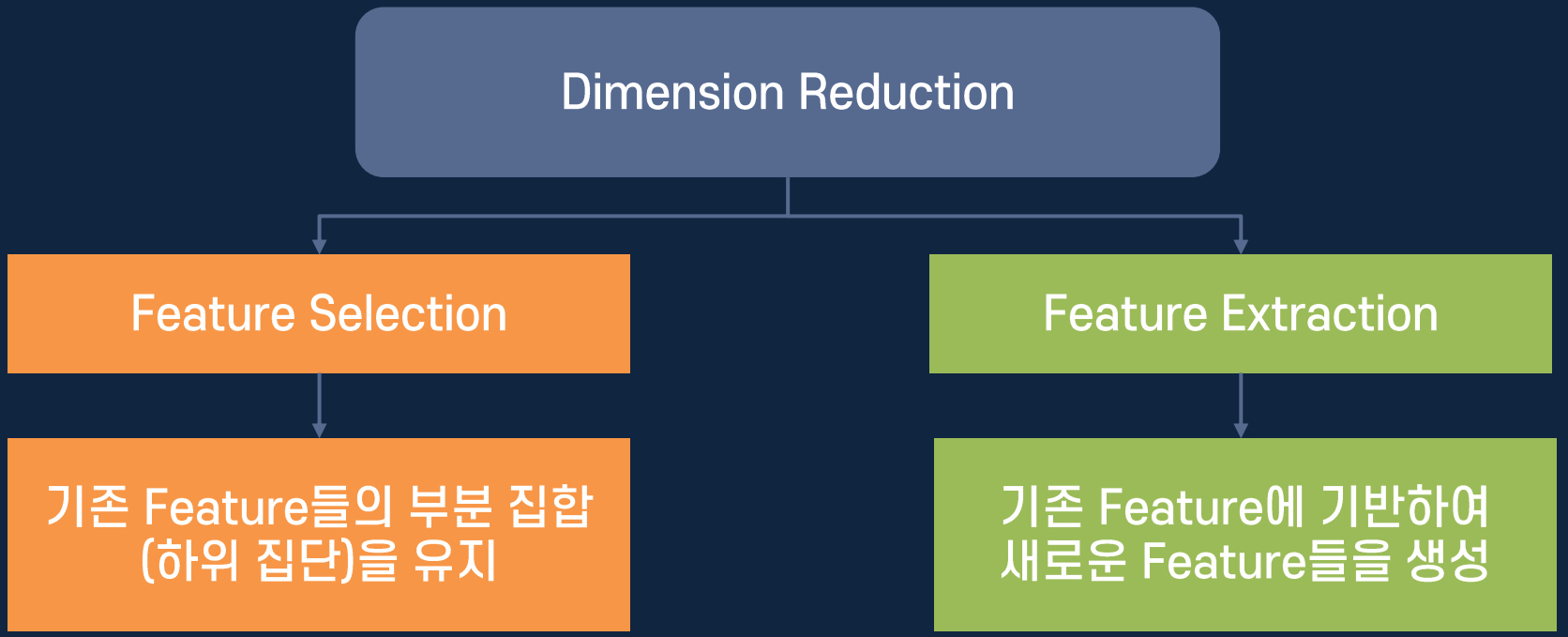
Transformation 종류

- np.log : 로그변환
- np.exp : 지수변환
- np.sqrt : 루트변환

Transformation과 Scaling을 적절히 섞어 사용하며 데이터를 정규분포화 시키자!

5 Feature Selection

Feature Selection VS Feature Extraction

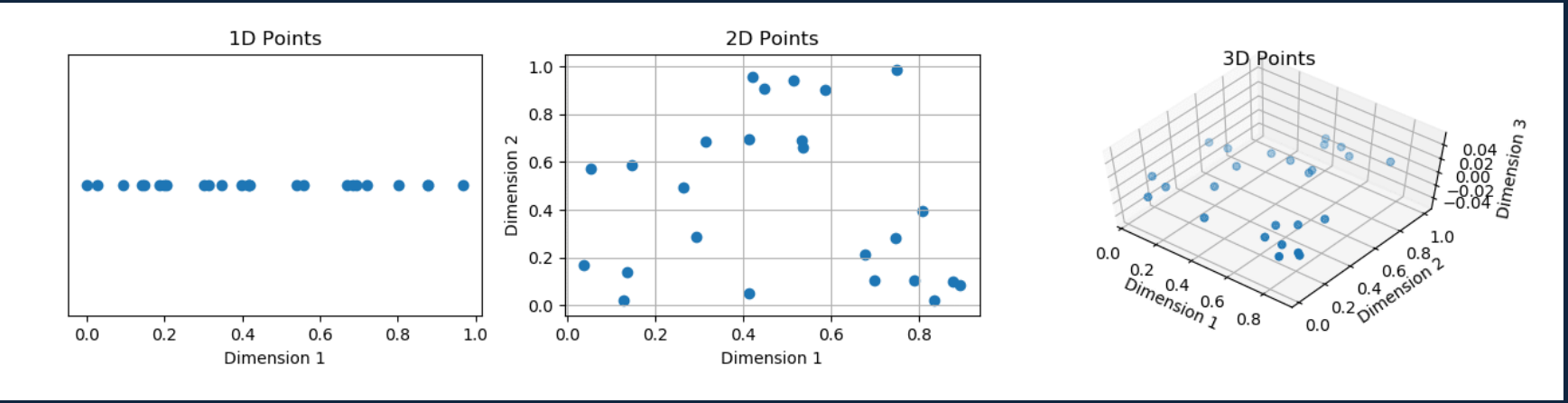


모든 Feature가 중요하지는 않음
적당한 Feature 선택 or 기존 Feature의 특징 추출 등 차원을 축소해 사용

5 Feature Selection

Dimension Reduction

<차원의 저주>



데이터 양은 일정한데 차원이 너무 커지면 데이터의 밀도가 떨어짐

→ 원하는 정보를 찾는데 Computing Cost가 많이 소요됨

→ 따라서 데이터 차원을 낮춰서 분석을 진행

5 Feature Selection

Feature selection 종류

Model based FS

특성 중요도를 제공하는 알고리즘에서 ex) Tree Model
특성 중요도가 기준치보다 높은 특성을 선택

Univariate FS

각각의 특성을 하나만 사용했을 때 예측모델의 성능평가 후
정확도 상관관계가 가장 좋은 특성만을 선택

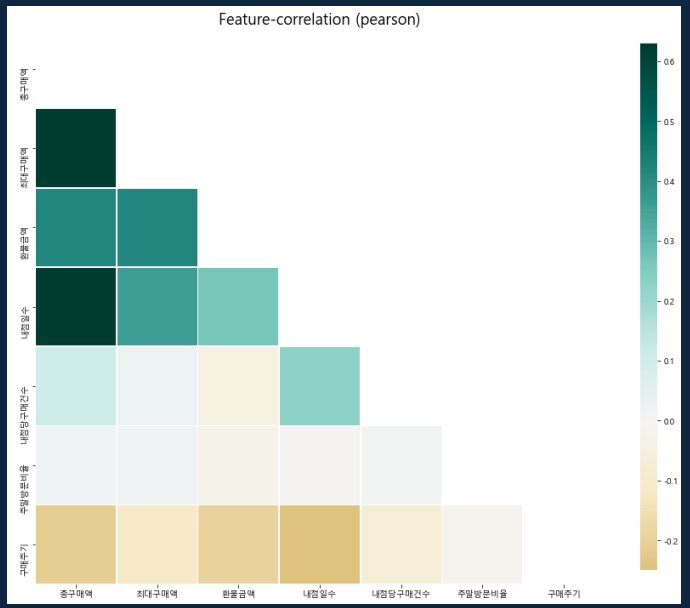
Recursive feature elimination

모든 조합을 다 시도해보고 가장 좋은 set을 찾는 방법
1개 ~ 모든 특성, 모든 특성 ~ 1개의 2가지 방법이 존재

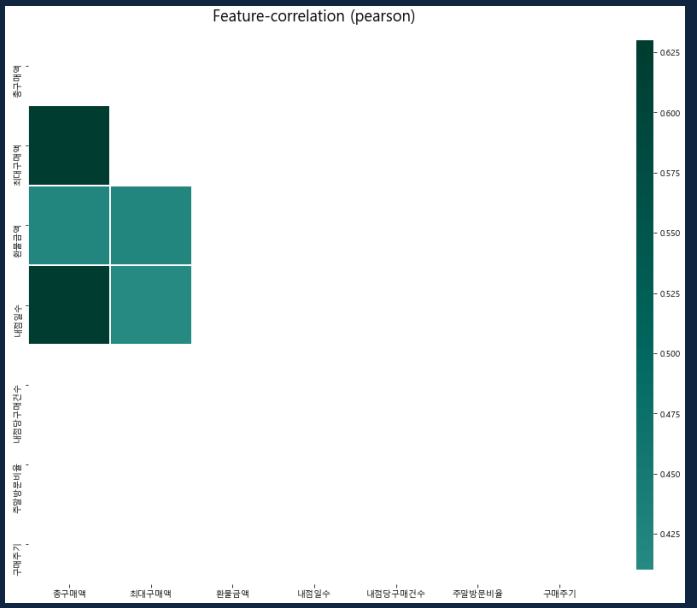
5 Feature Selection: Correlation

상관관계

통계적 접근으로 독립변수 x(Feature) 사이의 상관계수를 파악하고
상관계수가 높은, 즉 연관성이 높은 Feature 둘 중의 하나를 제거하는 방법



상관 계수가 높은 것 들만 시각화



6 Feature Extraction: PCA

PCA(주성분 분석)

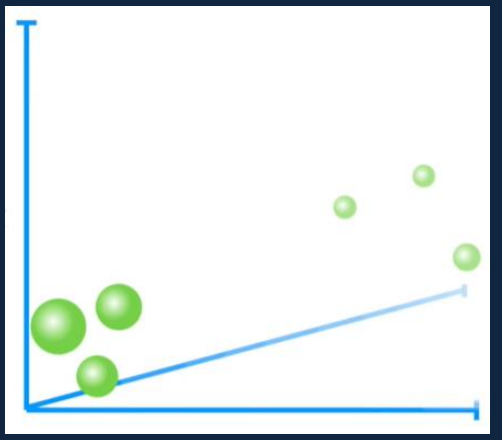
- 고차원의 데이터를 저차원으로 축소시키는 방법 중 하나
- 4차원 이상의 데이터에 대해 시각화 할 수 있게끔 하는 방법

PCA 사용 이유

1. 시각화 (Visualization)
   3차원이 넘어간 시각화는 우리 눈으로 볼 수 없음
   따라서 차원 축소를 통해 시각화를 하여 데이터 패턴을 쉽게 인지 가능
2. 노이즈 제거 (Reduce Noise)
   쓸모없는 feature를 제거함으로써 노이즈 제거 가능
3. 메모리 절약 (Preserve useful info in low memory)
   쓸모없는 feature가 제거되니 메모리도 절약 가능
4. 퍼포먼스 향상
   불필요한 feature들을 제거해 모델 성능 향상에 기여

6 Feature Extraction: PCA

<Visualization>



	국어	수학	영어
학생1	10	4	7
학생2	5	6	3
학생3	2	2	8.2
학생4	4	5.3	5
학생5	3.7	9	6
학생6	8	10	4.3

4차원 변환

	국어	수학	영어	사회
학생1	10	4	7	1
학생2	5	6	3	6
학생3	2	2	8.2	2.1
학생4	4	5.3	5	8
학생5	3.7	9	6	4
학생6	8	10	4.3	5

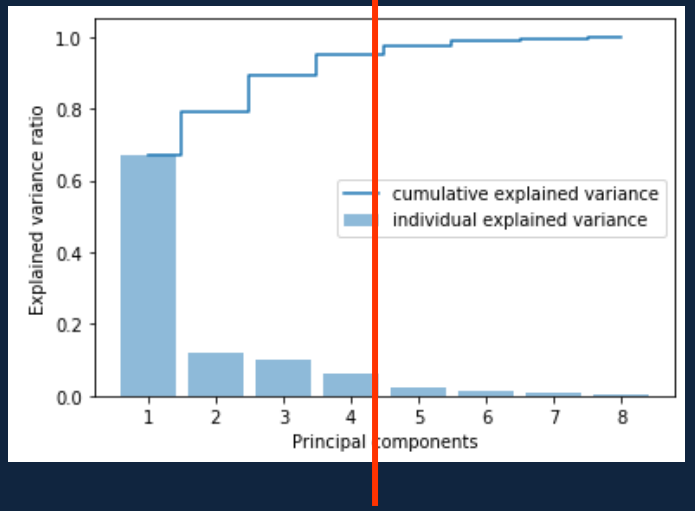
?

6 Feature Extraction: PCA

<100 차원 Data>

PCA 변환

90% 이상



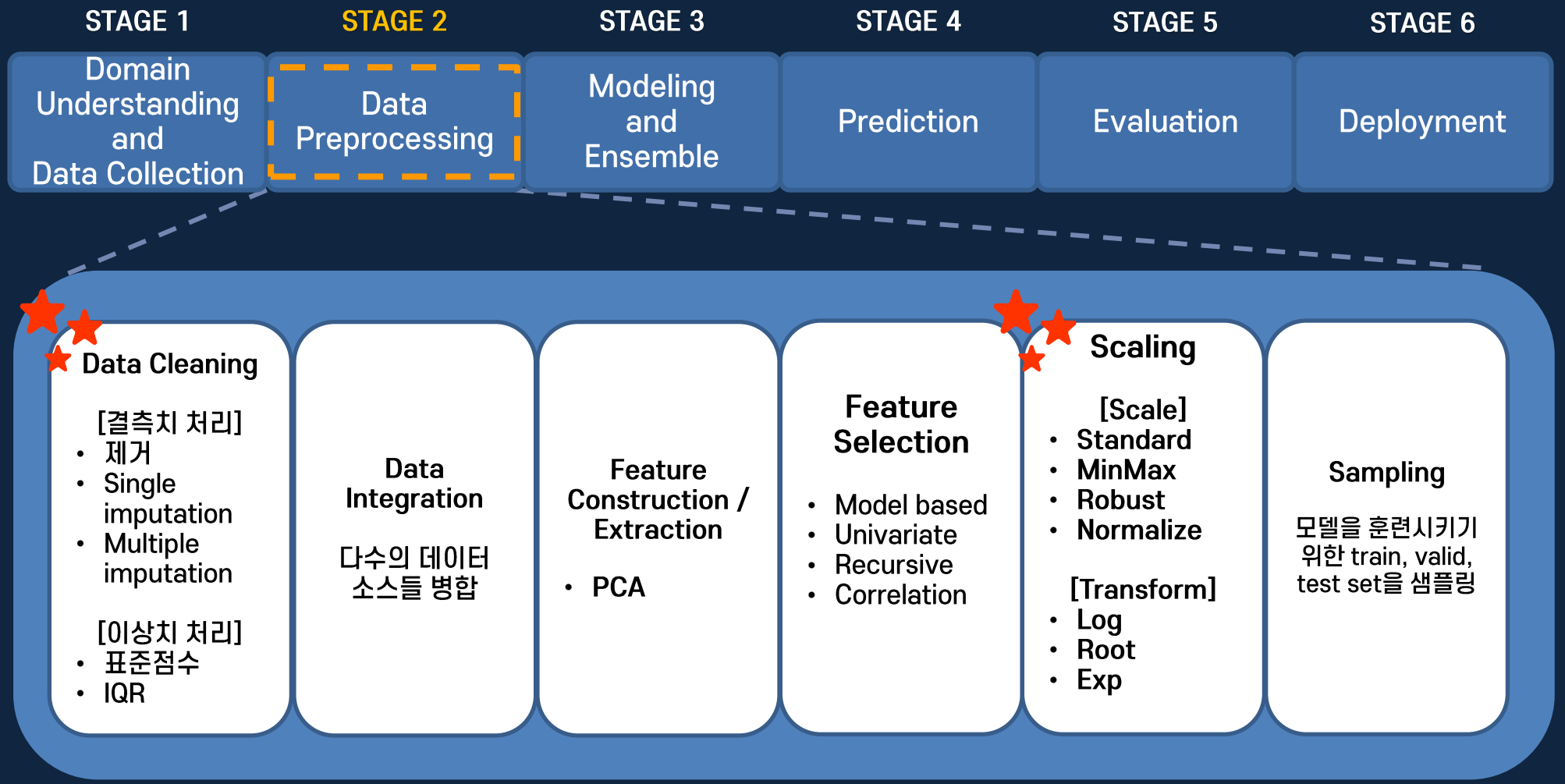
100차원 data를 4차원으로 축소 시킬 경우

4가지 PC column으로 전체 데이터의 90% 정도가 설명이 가능

PCA 변환이 되면 원본 data가 아닌 새로운 data로 변환됨 → Feature Extraction

고유값과 고유벡터에 대한 이해 필요 → 더 자세한 설명은 정여진 교수님 다변량 참고

Summary


STAGE 1
Domain Understanding and Data Collection
STAGE 2
Data Preprocessing
STAGE 3
Modeling and Ensemble
STAGE 4
Prediction
STAGE 5
Evaluation
STAGE 6
Deployment
Data Cleaning
[결측치 처리]
• 제거
• Single imputation
• Multiple imputation
[이상치 처리]
• 표준점수
• IQR
Data Integration
다수의 데이터 소스들 병합
Feature Construction / Extraction
• PCA
Feature Selection
• Model based
• Univariate
• Recursive
• Correlation
Scaling
[Scale]
• Standard
• MinMax
• Robust
• Normalize
[Transform]
• Log
• Root
• Exp
Sampling
모델을 훈련시키기 위한 train, valid, test set을 샘플링

과제

각 column에 적절한 결측치 대체 방법, 이상치 처리방법, Scaling 등을 고려하여 적용하고 feature selection과 PCA 등 도 다양하게 사용하여 Model 성능 올리기
(가능하다면 시각화도 시도해 볼 것)

Q n A

THANK YOU