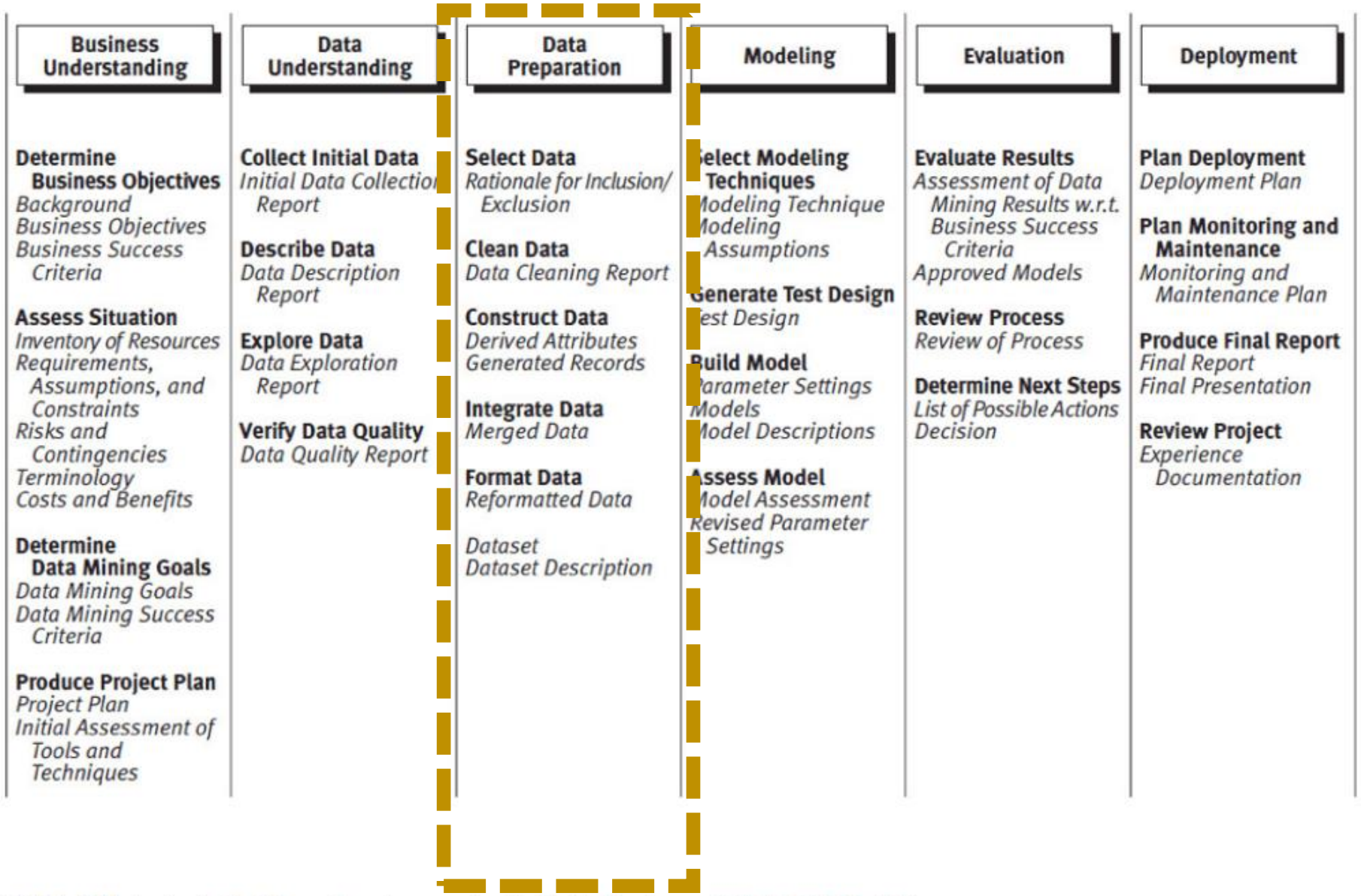
스케일링

스케일링

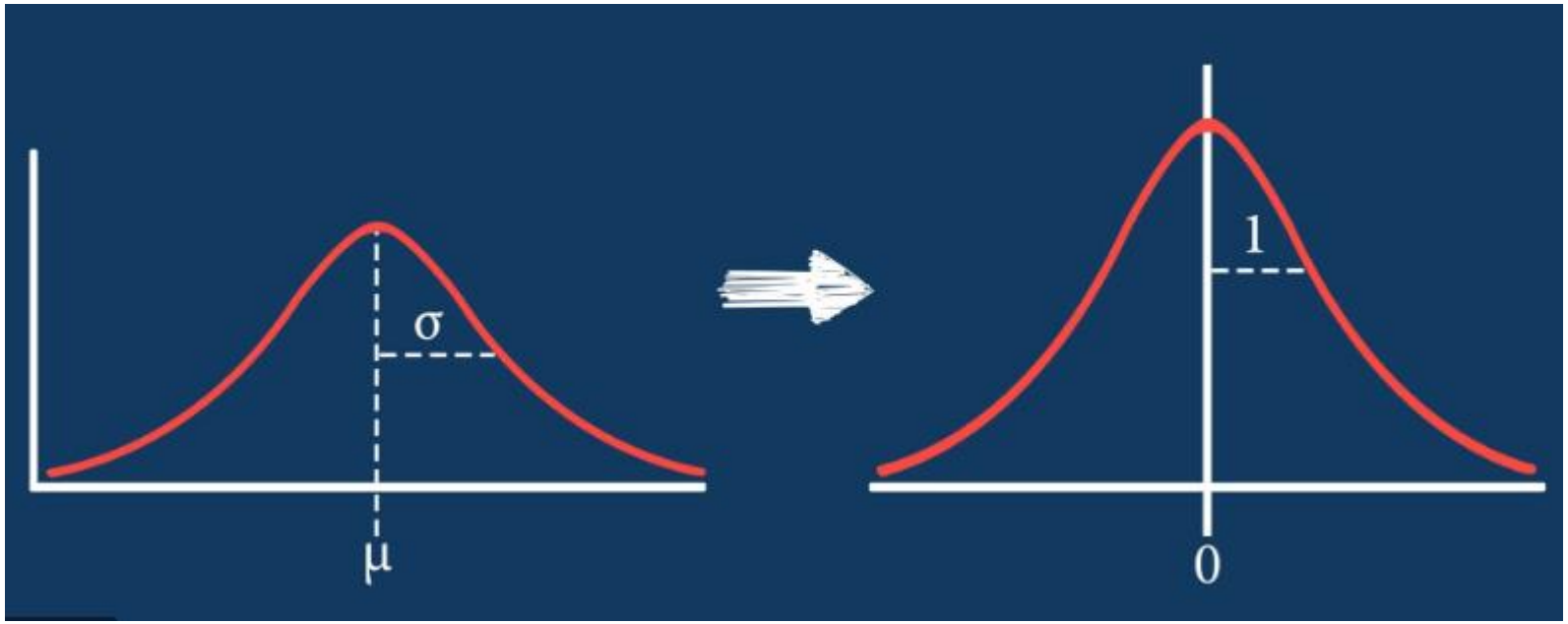


CRISP-DM tasks in **bold**, and outcomes in *italic* (table from CRISP-DM Guide)

스케일링

스케일링 (Data Scaling)

: 데이터의 값의 범위를 조정하는 것

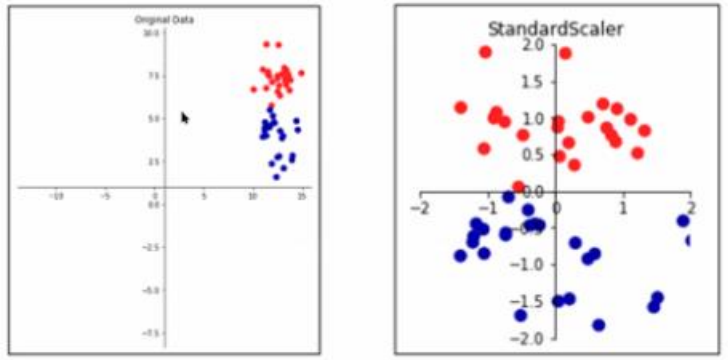


https://cheris8.github.io/data%20analysis/DP-Data-Scaling/

스케일링

스케일링 (Data Scaling)

: 데이터의 값의 범위를 조정하는 것



- 특성(Feature)들의 범위(range)를 **정규화**
- 특성마다 다른 범위를 가지는 경우 모델들이
  제대로 학습되지 않을 가능성이 있음 (KNN, SVM, Neural network 모델, Clustering 모델 등)

- **장점**
  - **특성들을 비교 분석하기 쉽게**
  - **Linear Model, Neural network Model 등에서 학습의 안정성과 속도 개선**

스케일링

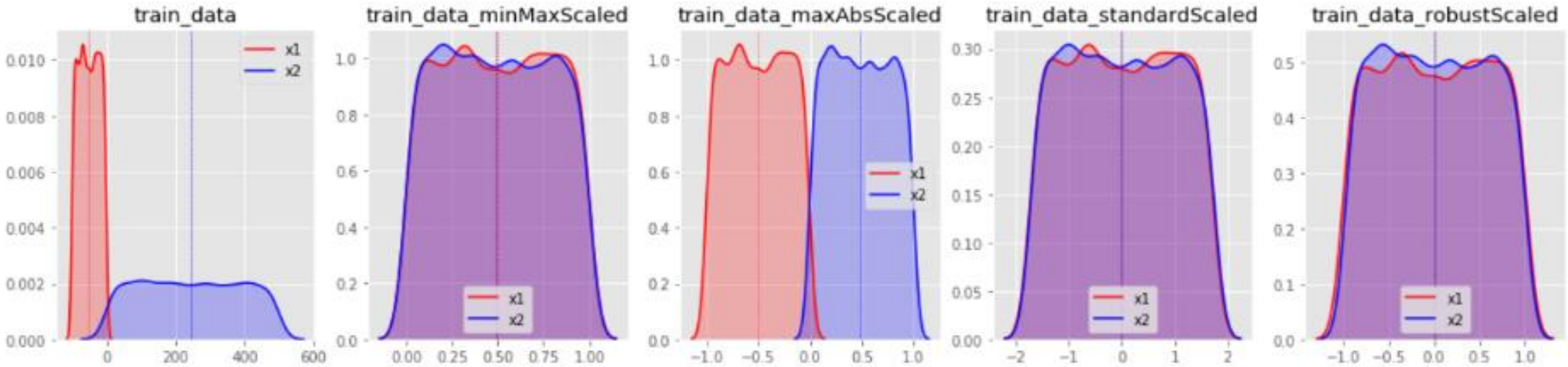
- **스케일링 종류**
  MinMaxScaler, MaxAbsScaler, StandardScaler, RobustScaler, Normalizer

1. MinMaxScaler : 데이터가 **0과 1 사이**에 위치하도록 스케일링
2. MaxAbsScaler : 데이터가 **-1과 1 사이**에 위치하도록 스케일링
3. StandardScaler : 데이터의 **평균 = 0, 분산 = 1**이 되도록 스케일링
4. RobustScaler : 데이터의 **중앙값 = 0, IQR = 1**이 되도록 스케일링
5. Normalizer : **각 행(row)마다 정규화** 진행

   cf) 앞의 방법은 각 피처(feature)의 통계치를 대상으로, 즉 열(columns)을 대상으로 함

스케일링

- **스케일링 종류**
  MinMaxScaler, MaxAbsScaler, StandardScaler, RobustScaler, Normalizer



스케일링

스케일링 종류

MinMaxScaler

- 데이터가 0과 1 사이에 위치하도록 스케일링 (default)
- 최소값 = 0, 최대값 = 1이 되도록 스케일링
  - (x - x의 최소값) / (x의 최대값 - x의 최소값)
- 데이터의 최소값과 최대값을 알 때 사용
- 이상치가 존재할 경우 스케일링 결과가 매우 좁은 범위로 압축될 수

$$\frac{x_i - \min(x)}{\max(x) - \min(x)}$$

MaxAbsScaler

- 데이터가 -1과 1 사이에 위치하도록 스케일링
- 절대값의 최소값 = 0, 절대값의 최대값 = 1이 되도록 스케일링
- 데이터의 값이 양수만 존재할 경우 `MinMaxScaler`와 유사하게 동작
- 이상치가 큰 쪽에 존재할 경우 이에 민감할 수

새로운 값 = 원래 값 / 해당 특성의 최대 절대값

ex) 어떤 특성 값이 [-10, 5, 15]
최대 절대값은 15
따라서, 이 특성의 값은 각각 -10/15, 5/15, 15/15 로 스케일링 연산 실행.
[-0.67, 0.33, 1]로 스케일링

스케일링

스케일링 종류

`StandardScaler`

- 데이터의 평균 = 0, 분산 = 1이 되도록, 즉 데이터가 표준 정규 분포(standard normal distribution)를 따르도록 스케일링
  - (x - x의 평균) / (x의 표준편차)
- 데이터의 최소값과 최대값을 모를 때 사용
- 평균(mean)과 분산(variance)을 사용
- 모든 feature들이 같은 스케일
- 평균과 표준편차가 이상치로부터 영향을 많이 받는다는 점에서 이상치에 민감

$$\frac{x_i - \text{mean}(x)}{\text{stdev}(x)}$$

(Xi - (X의 평균)) / (X의 표준편차)

`RobustScaler`

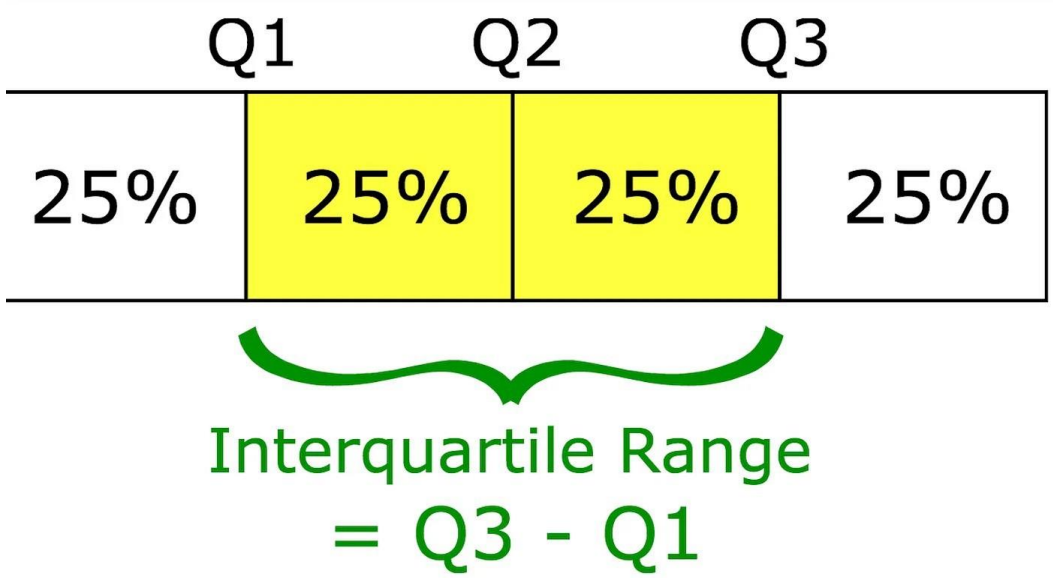
- 데이터의 중앙값 = 0, IQE = 1이 되도록 스케일링
- 중앙값(median)과 IQR(interquartile range)을 사용
  - `RobustScaler`를 사용할 경우 `StandardScaler`에 비해 스케일링 결과가 더 넓은 범위로 분포
- 모든 feature들이 같은 스케일
- 이상치의 영향을 최소화

(데이터 값 - Q2) / (Q3 - Q1)

스케일링

스케일링 종류 – RobusterScaler

(데이터 값 - Q2) / (Q3 - Q1)



1,2,3,4,5,6,100

IQR (Q3 - Q1) : 6 - 2 = 4

스케일링

스케일링 종류

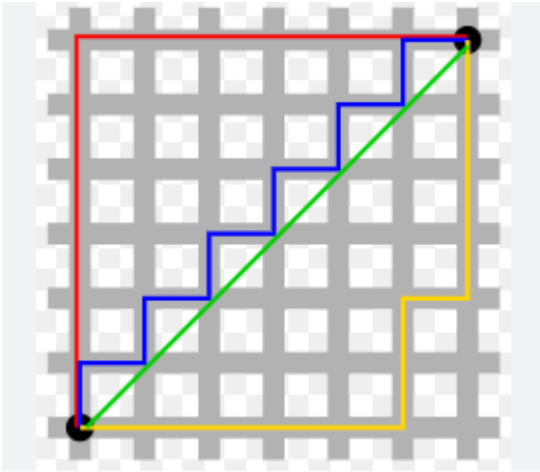
Normalizer()

앞의 4가지 방법은 각 피처(feature)의 통계치를 이용, 즉, 열(columns)를 대상으로.

Normalizer 의 경우 각 **행(row)마다 정규화**가 진행.

한 행의 모든 피처들 사이의 유클리드 거리가 1이 되도록 조정.

-> 더 빠른 학습, 과대적합 Down.



특성 벡터의 유클리디안 길이가 1이 되도록 조정

스케일링

스케일링 종류

Normalizer()

한 행의 모든 피처들 사이의 유클리드 거리가 1이 되도록 조정.

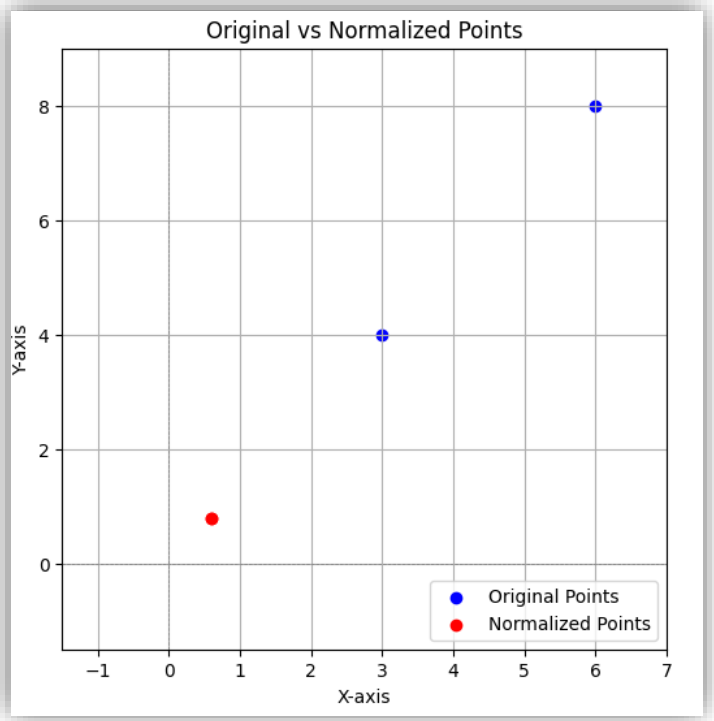
- 데이터 : X1=(3,4), X2=(6,8)
- 유클리디안 길이 계산

  X1 : ||X1||2=32+42=9+16=5
  X2 : ||X2||2=62+82=36+64=10

- Normalizer 적용

$$X_{1,normalized} = \left(\frac{3}{5}, \frac{4}{5}\right) = (0.6, 0.8)$$

$$X_{2,normalized} = \left(\frac{6}{10}, \frac{8}{10}\right) = (0.6, 0.8)$$



스케일링

Normalizer()

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import Normalizer

# 예제 데이터
X = np.array([[3, 4], [6, 8]])

# Normalizer 적용
normalizer = Normalizer(norm='l2')  # L2 정규화
X_normalized = normalizer.transform(X)

# 그래프 설정
plt.figure(figsize=(6,6))
plt.axhline(0, color='gray', linestyle='--', linewidth=0.5)
plt.axvline(0, color='gray', linestyle='--', linewidth=0.5)

# 원래 데이터 점 플로팅 (파란색)
plt.scatter(X[:, 0], X[:, 1], color='blue', label='Original Points')

# 변환된 데이터 점 플로팅 (빨간색)
plt.scatter(X_normalized[:, 0], X_normalized[:, 1], color='red', label='Normalized
Points')

# 축 범위 설정
plt.xlim(-1.5, 7)
plt.ylim(-1.5, 9)
plt.grid(True)

# 범례 추가
plt.legend()
plt.title("Original vs Normalized Points")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")

# 그래프 출력
plt.show()
```

2.06.스케일링.Normalizer.ipynb

스케일링

- 실습

  2.03.스케일링.ipynb

- TASK

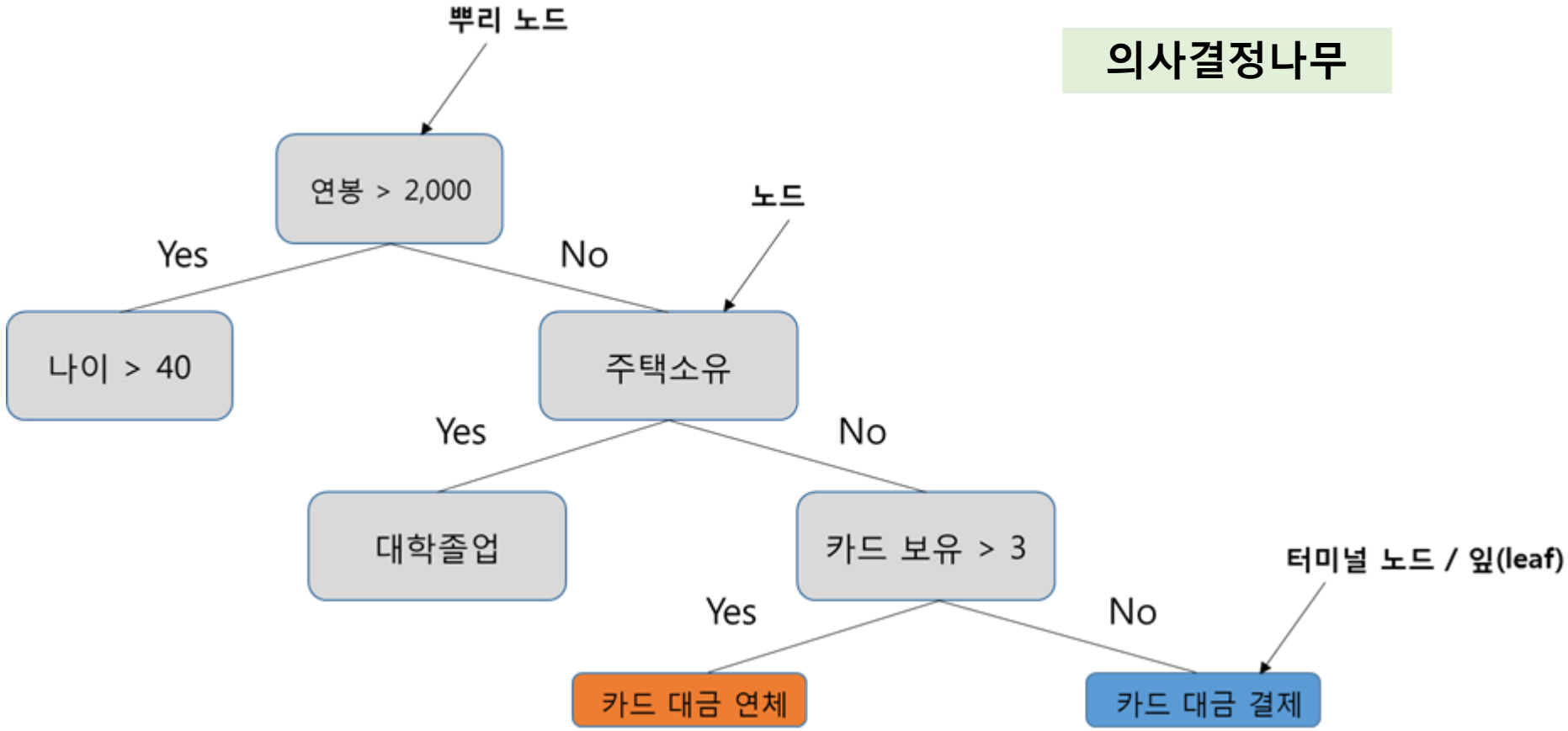
  StandardScaler외 4개의 스케일러를 사용 스케일링 전 후의 모델 정확도 확인

스케일링

- 스케일링의 필요성

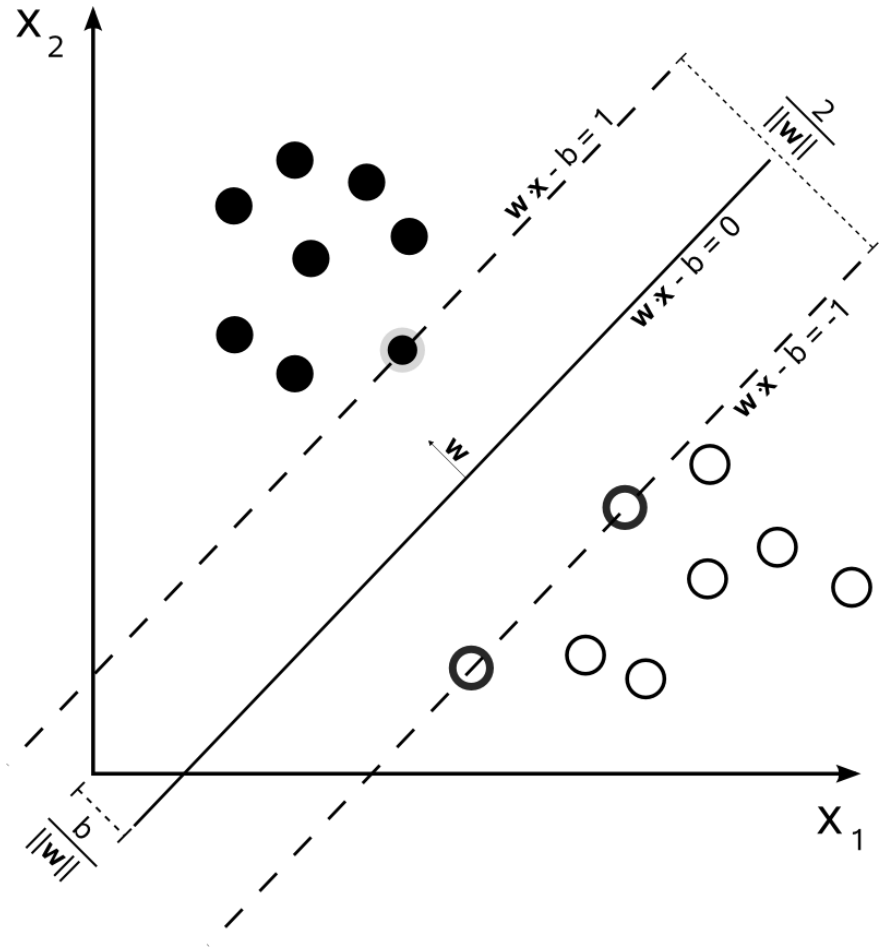
  특성마다 다른 범위를 가지는 경우 모델들이

  제대로 학습되지 않을 가능성이 있음 (KNN, SVM, Neural network 모델, Clustering 모델 등)



스케일링

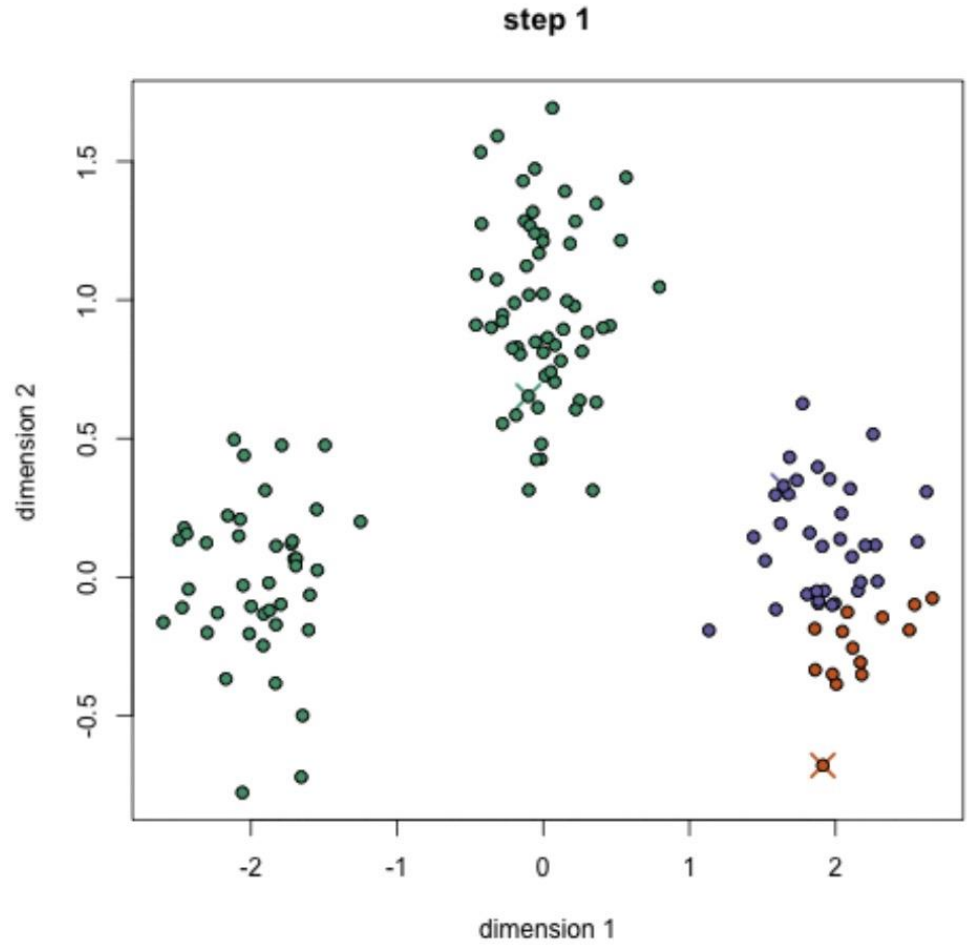
- 특성마다 다른 범위를 가지는 경우 머신러닝 모델들이
  제대로 학습되지 않을 가능성이 있음 (KNN, SVM, Neural network 모델, Clustering 모델 등)



SVM

스케일링

- 특성마다 다른 범위를 가지는 경우 머신러닝 모델들이
  제대로 학습되지 않을 가능성이 있음 (KNN, SVM, Neural network 모델, Clustering 모델 등)



클러스터링

THANK YOU