AI의 이해

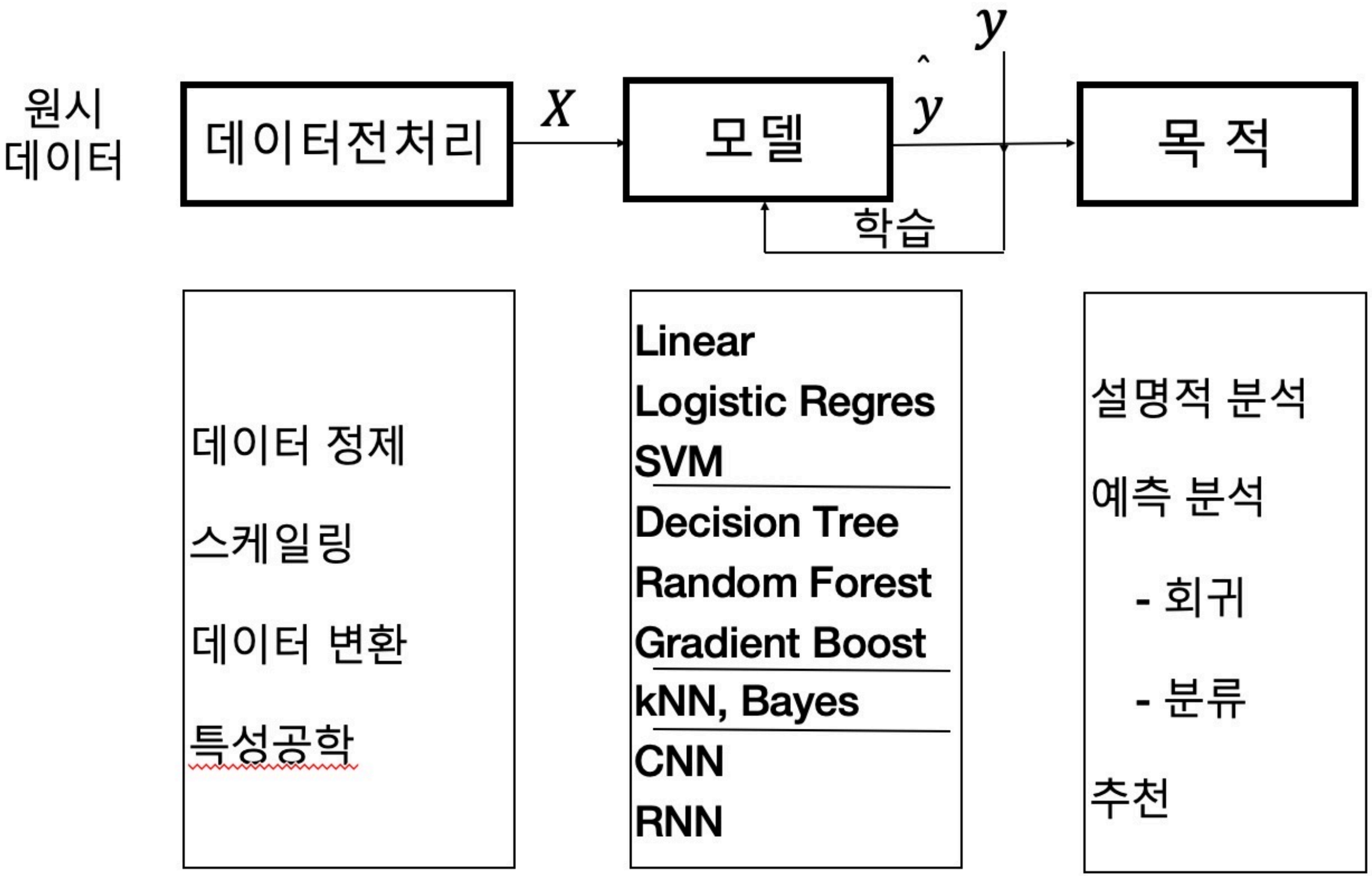
AI의 발전

- 영역별로 인간의 능력을 능가하고 있다.
  - 이미지 인식 – 보는 능력
  - 음성인식, 텍스트 인식 – 듣는 능력
  - 실시간 번역 – 말하는 능력
  - 이미지 캡션, 언어 모델링 – 쓰는 능력
  - 감성 능력까지 ?

AI와 머신러닝 정의

- AI란
  - "컴퓨터가 마치 지능이 있는 것처럼 똑똑하게 동작하는 것"을 말한다
  - 자동차 내비게이터, 검색엔진, 음악, 영화, 상품 추천 앱 등
  - AI가 특정한 기술을 사용해야만 하는 것이 아니다. 예를 들어 신경망 (딥러닝) 을 사용해야만 하는 것은 아니다.
- AI를 구현하는 기술은 1950년대부터 연구됨
  - 초기에는 사람처럼 "생각하는 방법"을 모방하려고 시도했고 이를 위해 언어학, 기호학 기반의 AI를 연구하였으나 모두 성공하지 못했다.
- 머신러닝 기반 AI
  - 현재 동작하는 AI 는, 데이터를 보고, 스스로 학습하여, 성능이 개선되는 방식의 AI가 널리 사용되고 있다.
  - 이를 머신러닝(기계학습) 기반의 AI라고 한다.

머신러닝 기반 AI


원시 데이터
데이터전처리
X
모델
ŷ
y
학습
목 적
데이터 정제
스케일링
데이터 변환
특성공학
Linear
Logistic Regres
SVM
Decision Tree
Random Forest
Gradient Boost
kNN, Bayes
CNN
RNN
설명적 분석
예측 분석
- 회귀
- 분류
추천

파이썬 기초 문법

- 기본 변수
  - 정수, 소수, 문자열, 논리값을 표현한다
- 리스트
  - 임의의 데이터를 목록을 만들어 담을 수 있다. [ ] 로 표현된다
- 튜플
  - 상수화된 리스트이다. 값의 변경이 불가하다. ( ) 로 표현된다
- 사전(dictionary)
  - 모든 항목이 항상 "키(key)"와 "값(value)" 짝으로 구성. { }로 표현된다
- 논리 흐름
  - 조건의 만족을 논리적으로 판단한다 if, else, elif 사용
  - 반복을 위하여 for, while을 사용한다
- 함수
  - 사용자가 임의의 기능을 정의할 수 있다. def를 사용

데이터 탐색과 전처리

데이터 탐색

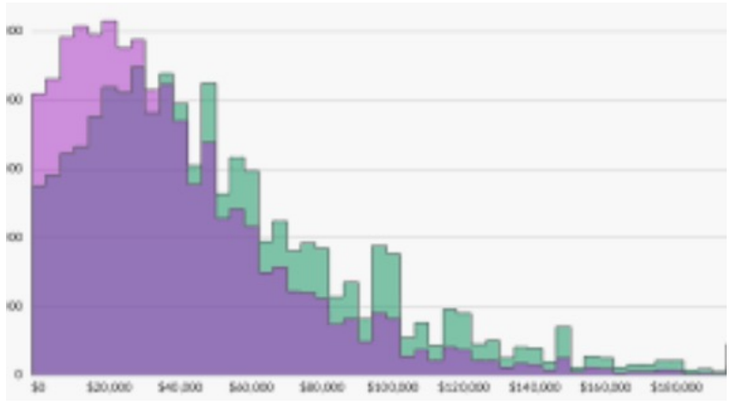
- exploratory data analysis: EDA
- 본격적인 데이터 분석이나 머신러닝을 수행하기에 앞서 테이터의 전체적인 특성을 살펴보는 것
- 수집한 데이터가 분석에 적절한지 알아보는 과정
- 시각화 기술을 주로 사용

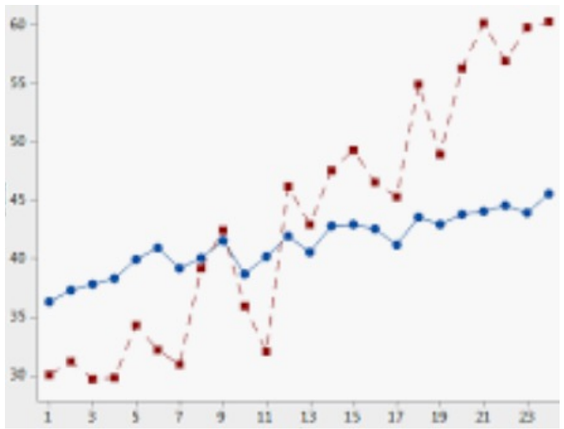
데이터 타입

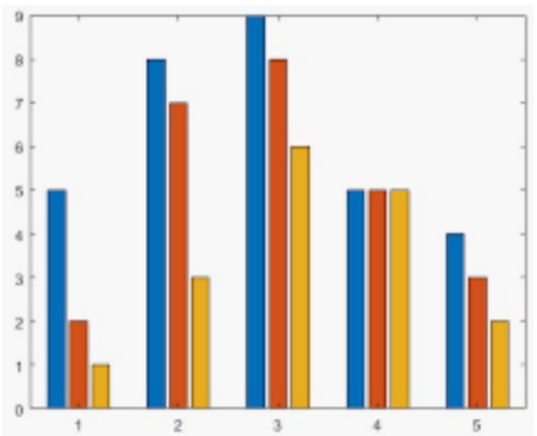
- 정형 데이터
  - 문자형: “Hello World”, “대한민국”, ...
  - 수치형: 1, 5, 10, 3.14, 0.9, ...
  - 바이너리형: 0100100101010101...
  - 논리형: True, True, False, True, ...
    - 수치형 데이터: 범주형(categorical), 순서형(ordinal), 연속형(continuous)
- 비정형 데이터
  - 정형 데이터와 달리 일정한 포맷을 가지지 않은 데이터를 말한다.
  - 블로그, 트윗 등 텍스트와, 오디오나 비디오 데이터, 센서 데이터, 웹데이터 등
  - 컴퓨터는 궁극적으로 숫자로 표현된 데이터만 처리할 수 있으므로 비정형 데이터를 숫자로 표현을 바꾸어야만 처리할 수 있다.

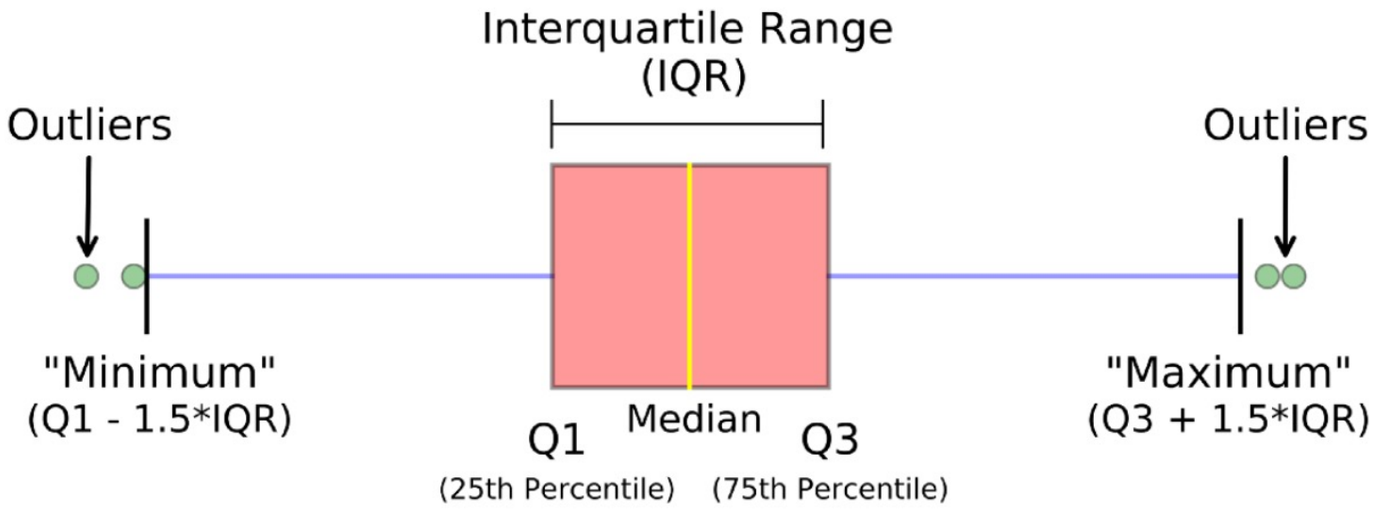
시각화 함수

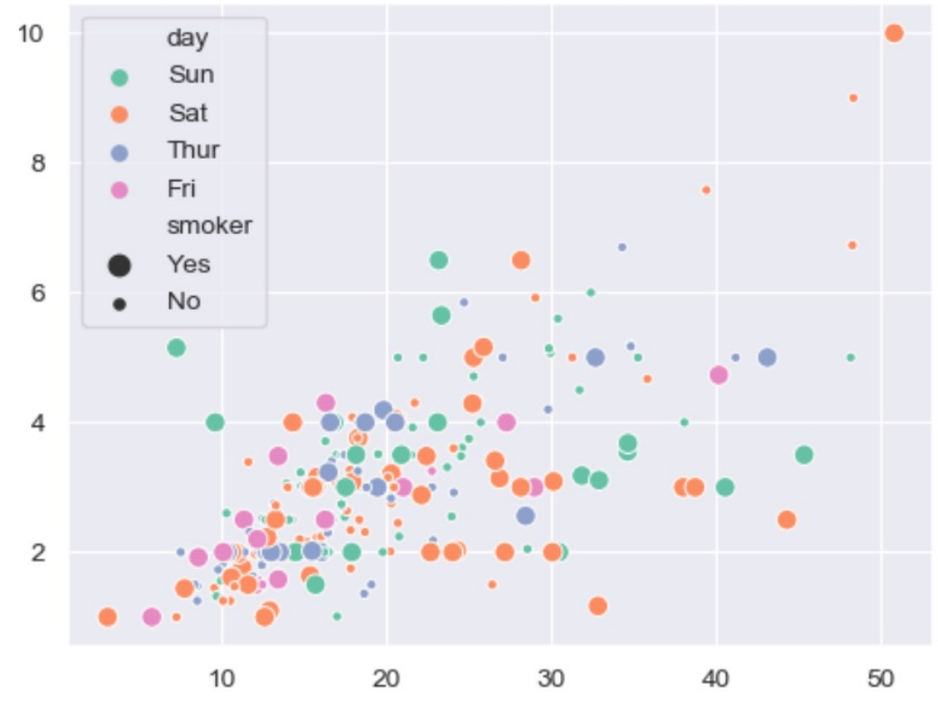
- 히스토그램
- 타임 플롯
- 바플롯
- 박스플롯
- 산포도











데이터 전처리

- 데이터 전처리(preprocessing)란 데이터를 분석에 사용할 때 성능이 더 좋게 나오도록 데이터를 수정하거나 형태를 변형하는 것
- 실제로 수집한 데이터를 머신러닝에 바로 사용하는 경우는 거의 없다.
- 데이터가 비정형이라면 이를 정형 데이터로 바꾸어야 한다.
  - 이미지나 텍스트와 같은 비정형 데이터를 컴퓨터가 바로 다룰 수는 없다.

데이터 전처리 유형

- 데이터 정제 (cleaning)
  - 빠진 값을 처리하거나 오류를 수정하는 것
- 데이터 변형 (transformation)
  - 값에 로그를 적용하거나, 역수를 취하거나, 카테고리 변수를 적용하는 것
- 스케일링 (scaling)
  - 데이터 값의 범위를 조정하는 것
- 특성 선택과 차원 축소
  - feature selection, dimensionality reduction
  - 특성(feature) 중 일부를 선택하여 분석에 사용하는 것
  - 특성을 조합하여 새로운 특성을 만드는 것
  - 특성의 차원의 수를 줄이는 작업

데이터 정제

- 결측치 처리
  - 빠진 값 즉, 결측치(missing value)를 처리하는 것이다.
  - 결측치를 처리하는 방법은 크게 세 가지가 있다.
    - 결측치가 포함된 샘플 항목을 모두 버리는 방법
    - 결측치를 적절한 값으로 대체하는 방법
    - 분석 단계로 결측치 처리를 넘김
- 틀린값 처리
  - 틀린값을 처리하는 방법도 결측치를 처리하는 방법과 같이 세가지이다.
    - 틀린 값이 포함된 항목을 모두 버리는 방법
    - 틀린 값을 적절한 값으로 대체
    - 분석 단계로 틀린 값 처리를 넘김

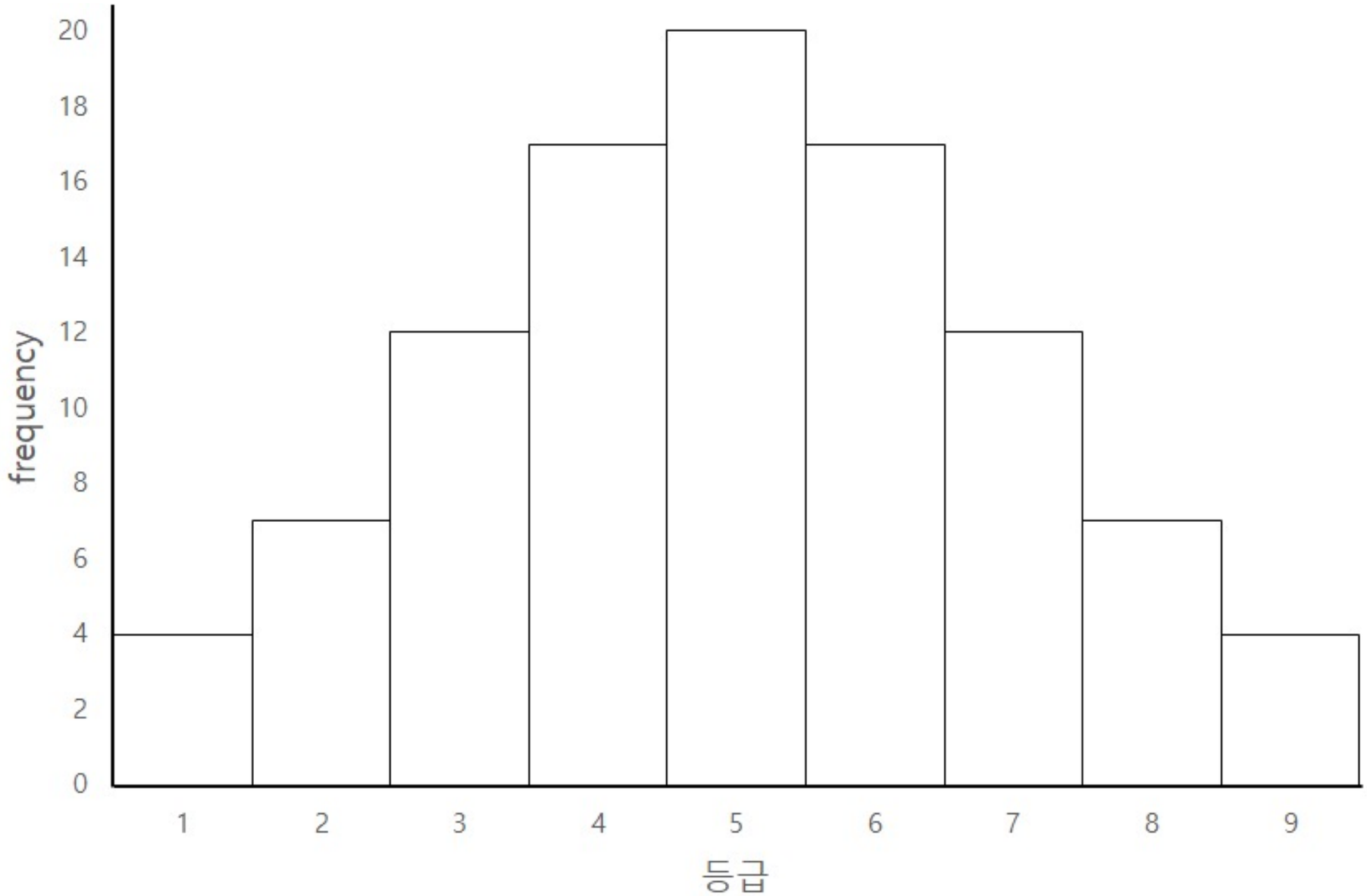
이상치 처리

- 이상치(outlier)란 값의 범위가 일반적인 범위를 벗어나 특별한 값을 갖는 것을 말한다.
  - 이상치는 틀린 값은 아니다
- 이상치를 찾아내는 것을 이상치 검출(detection)이라고 한다
  - 도난당한 카드의 사용을 찾아내거나,
  - 불법으로 보험료를 청구하는 것을 찾거나,
  - 기계가 이상한 동작을 하는 것을 찾는데 사용된다.
- 이상치 검출은 데이터 전처리가 아니라 이 자체가 데이터 분석이다.

데이터 변환

- 데이터를 주어진 그대로 사용하지 않고 다른 형태로 바꾸어 사용하는 것
- 범주형 변환
  - 범주형 즉, 카테고리를 나타내는 표현으로 바꾸는 것
  - 예를 들어 월요일은 1로, 화요일을 2로 코딩
  - 수치 데이터를 범주형으로 변환하여 사용하는 것
    - 나이, 연간 소득, 성적 등급 등
  - get_dummies()를 사용한다.
- 로그변환
- 역수변환
  - 어떤 수치를 그대로 사용하지 않고 역수를 사용하는 것

범주형 변환 예



범주형 변수 코딩

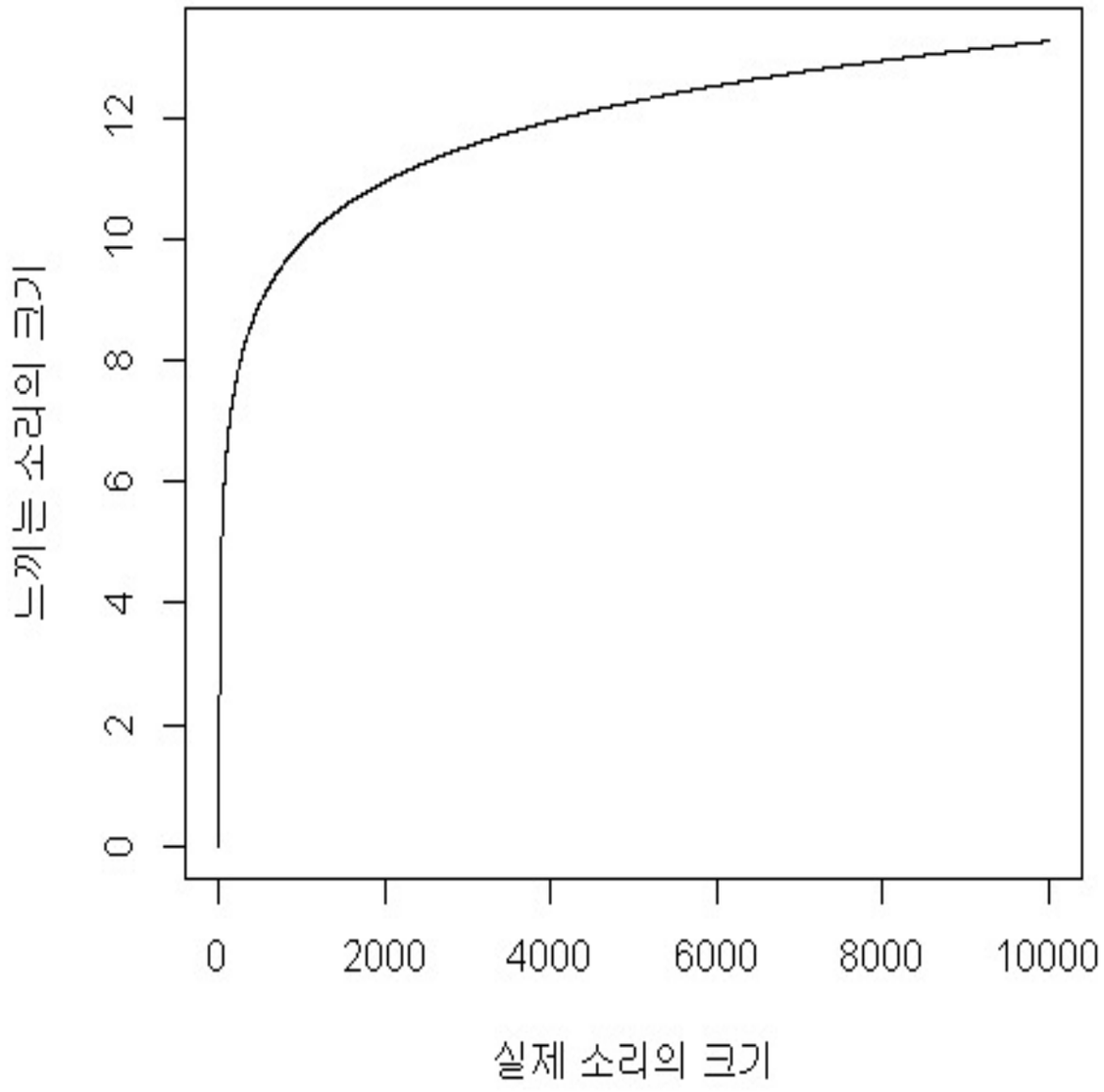
- 예를 들어 요일을 1, 2, 3, 4, 5, 6, 7 등으로 표시한 경우 이 변수를 컴퓨터가 연산(덧셈이나 곱셈)을 할 수 있는 숫자로 인식해서는 안 된다.
- 이 숫자를 범주형(카테고리형)으로 분명하게 처리되어야 한다.
- one hot encoding
  - 하나로 하나의 특성(컬럼)만 1이 될 수 있고 다른 특성은 모두 0으로 코딩하는 방법
  - 판다스가 제공하는 get_dummies()를 사용하면 카테고리형 변수들을 원핫인코딩으로 만들어준다

로그 변환

- 체감형 수치를 선형적으로 표현할 때 사용
  - 사람이 자연적으로 느끼는 느낌의 양을 표현할 때 사용
  - 돈, 소리, 빛, 압력, 냄새 등 생물학적인 자극을 주는 경우
  - 같은 자극을 느끼려면 현재 보유한 양이 많을수록 이에 비례한 더 강한 자극이 필요하다는 것
- 이를 수학적으로 표현하면 로그 함수가 됨
  - 현재 보유한 양이 x이고 이의 변화량, 즉 미분값이 1/x이 되려면 로그 함수를 얻음
  - 로그를 취한 이후의 값에 대해서 사람들이 변화량을 느끼는 것이 선형적이라는 특성

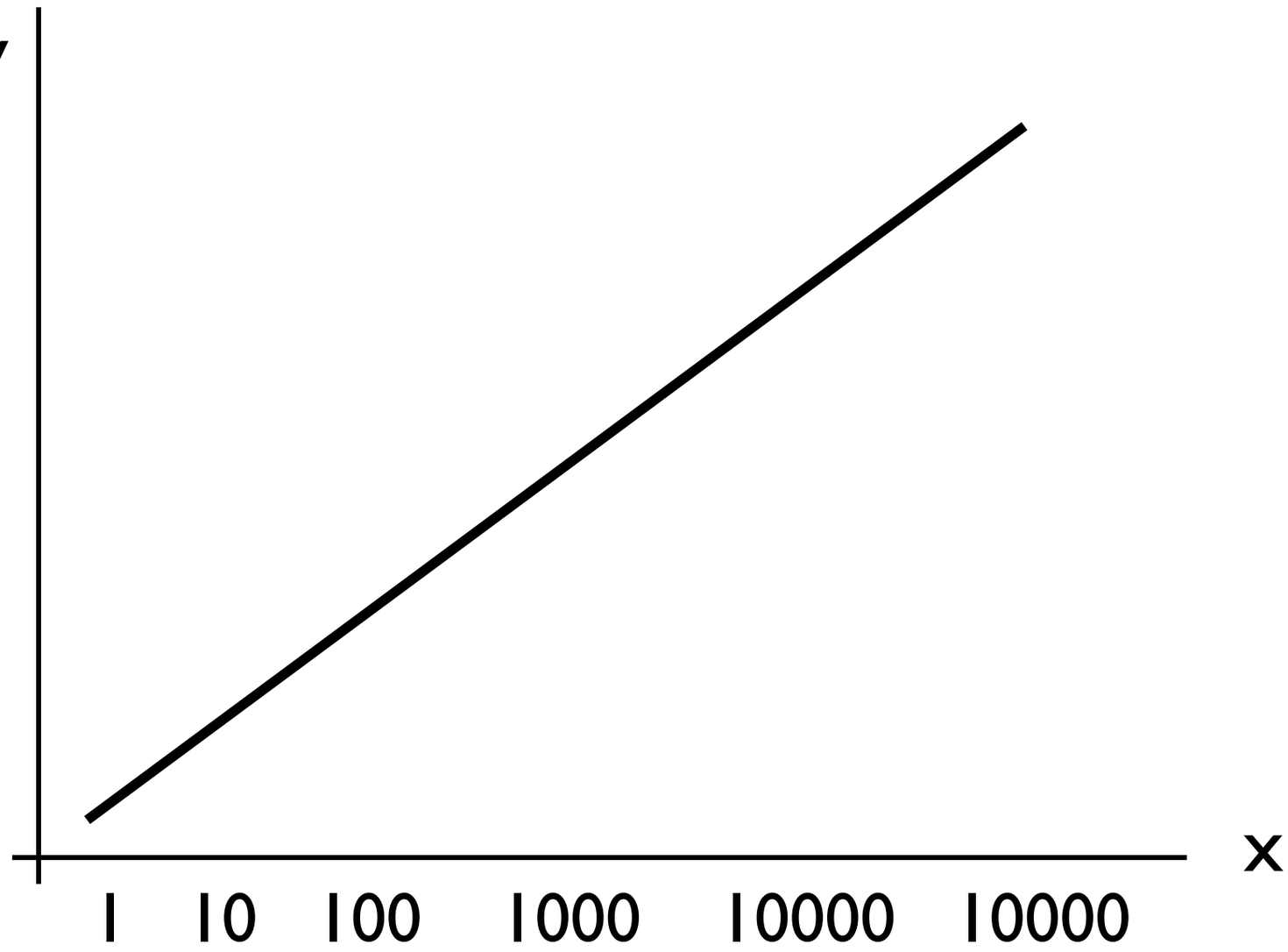
로그 변환

- 원래 값에 로그를 취한 값을 사용하는 것
- 사람이 느끼는 감각, 소리, 빛, 압력, 냄새, 금전적인 수입 등 생물학적인 자극에 대해서는 로그를 취한다.



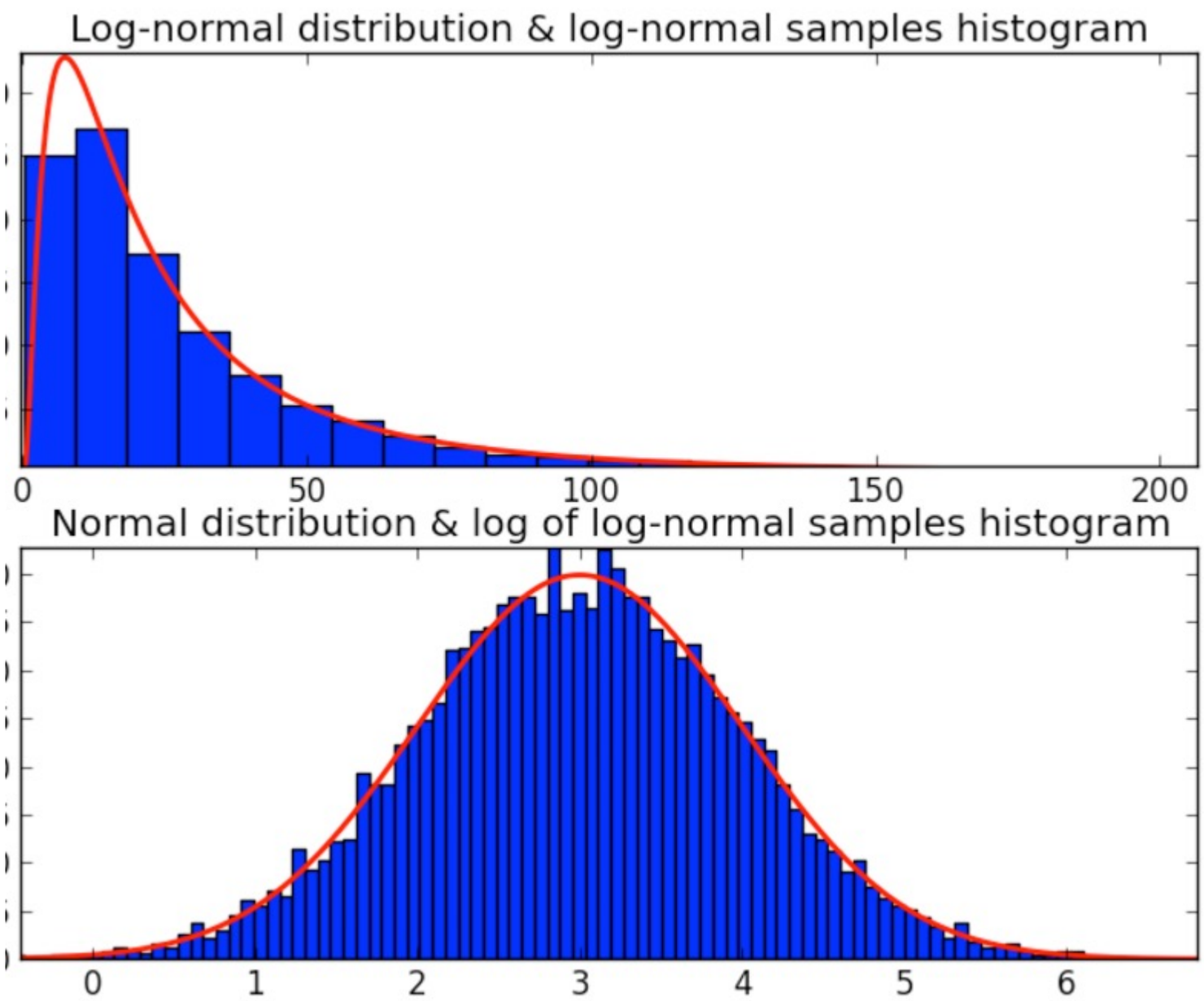
로그 스케일

- 로그 스케일 입력에 대해 선형 특성을 갖는 것



로그-정규 분포

- 로그를 취하고 나면 정규 분포를 이루는 경우



역수 변환

- 역수를 사용하면 선형적인 특성을 가져 분석의 정확도가 높아지는 경우
- 자동차 마일리지(연료 1L로 가는 거리 Km)와 연비(100km 주행하는데 필요한 연료 L)는 모두 자동차의 성능을 나타내지만 서로 역수의 관계
- 측정 목적:
  - 같은 비용을 얼마나 멀리 갈 수 있는가?
  - 같은 거리를 여행하는데 비용이 얼마가 드는가?
- 속도와 시간의 관계는 역함수이다

스케일링

- 스케일링(scaling)이란 원래 데이터가 갖는 값의 범위를 다르게 조정하는 작업
- 각 특성의 중요도를 갖게 맞추기 위해서 필요
- 최소-최대 스케일링
  - 주어진 값의 최소값을 0으로 최대값을 1로 재조정하는 것
  - min-max 스케일링
  - MinMaxScaler() 함수 사용
- 표준 스케일링
  - 주어진 샘플의 평균치가 0이, 표준편차가 1이 되도록 변환하는 것
  - z-score 정규화라고도 한다.
- Robust 스케일링

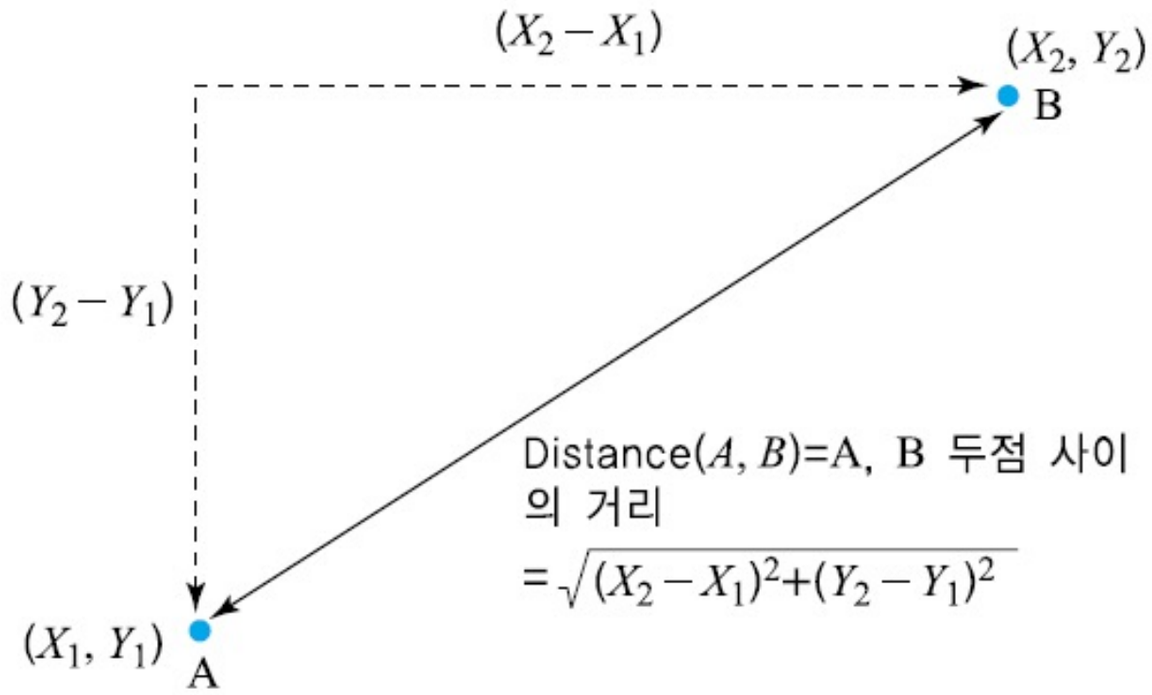
유사도와 거리

유사도와 거리

- 머신러닝은 유사도와 거리를 정의하는 것에서 출발한다
- 특성 공간상의 유사도와 거리를 사용한다
- 거리가 가까운 샘플이 비슷한 성격을 갖는다는 가정을 사용
- 유사도 s(similarity)는 $0 \leq s \leq 1$ (1에 가까울수록 유사도 높음)
- 유사도의 상대 개념으로 거리(distance) 사용

공간 거리

- 기하학의 공간(space) 상의 직선 거리
  - 유클리디언 (Euclidian) 거리

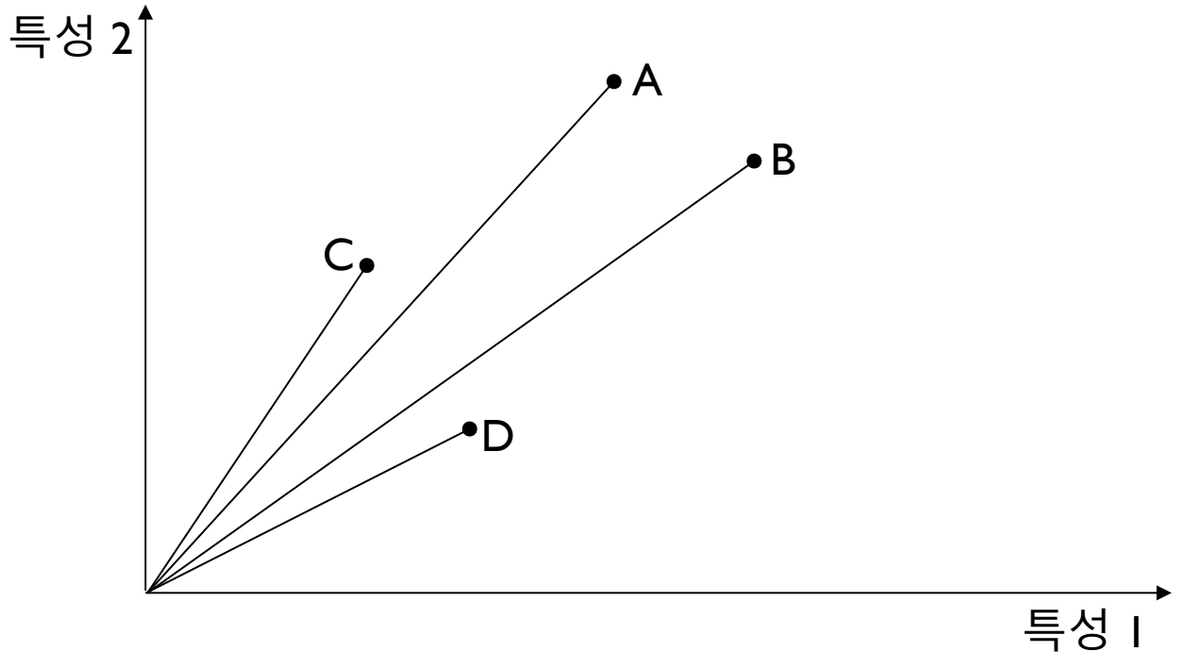


- n 차원 공간상의 두 점의 거리

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

코사인(cosine) 유사도

- 공간상의 두 점이 만드는 각도를 기준으로 유사도를 측정하는 방법 (-1 ~ +1 사이의 값을 갖는다)
- 공간상 거리가 멀어도 두 점이 가리키는 방향이 같으면 서로 비슷하다고 보는 것



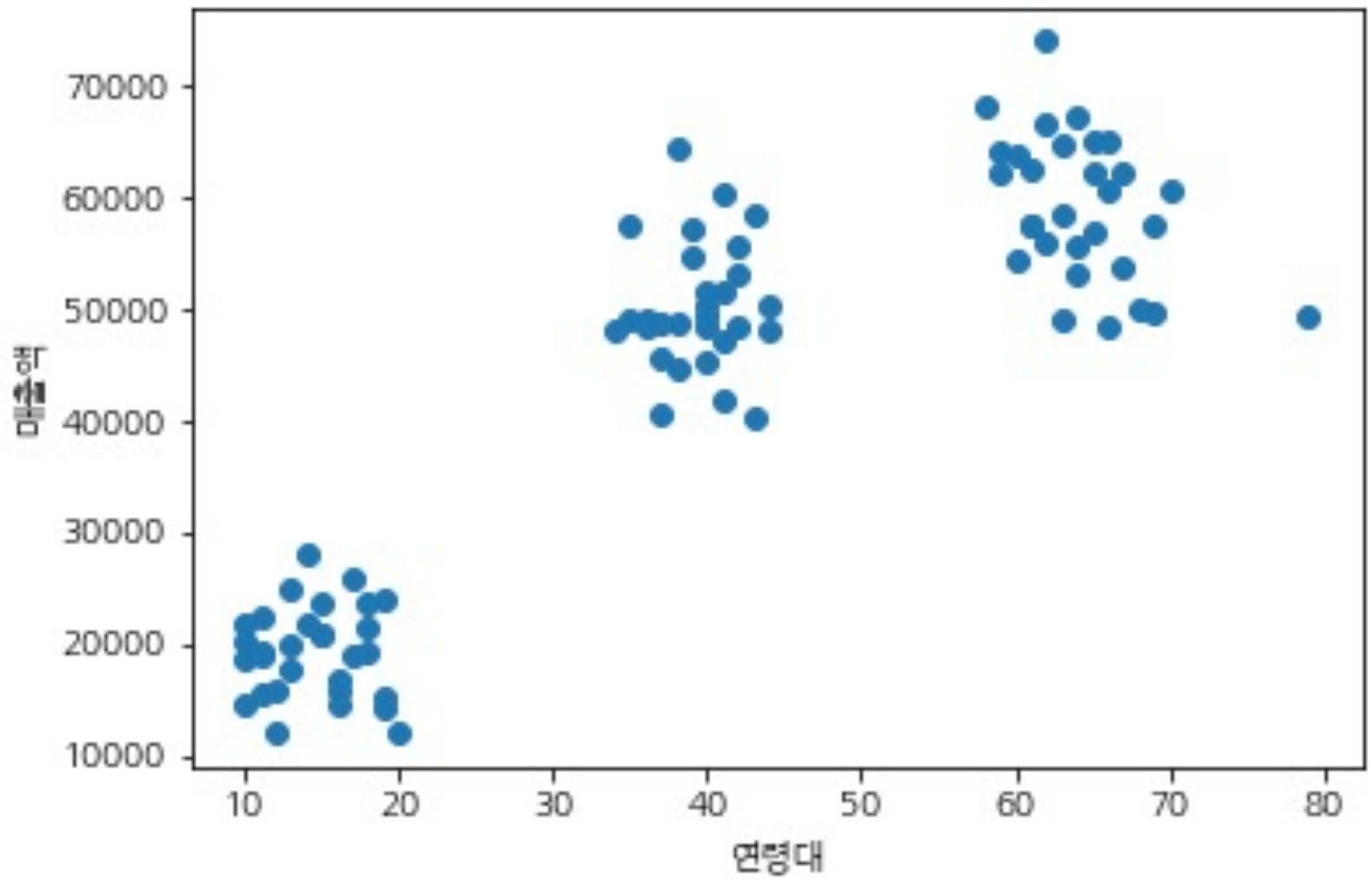
$$s_{\cos}(x,y) = \frac{X \bullet Y}{|X||Y|}$$

- A와 C가 가깝고 B와 D가 가깝다고 정의

클러스터링

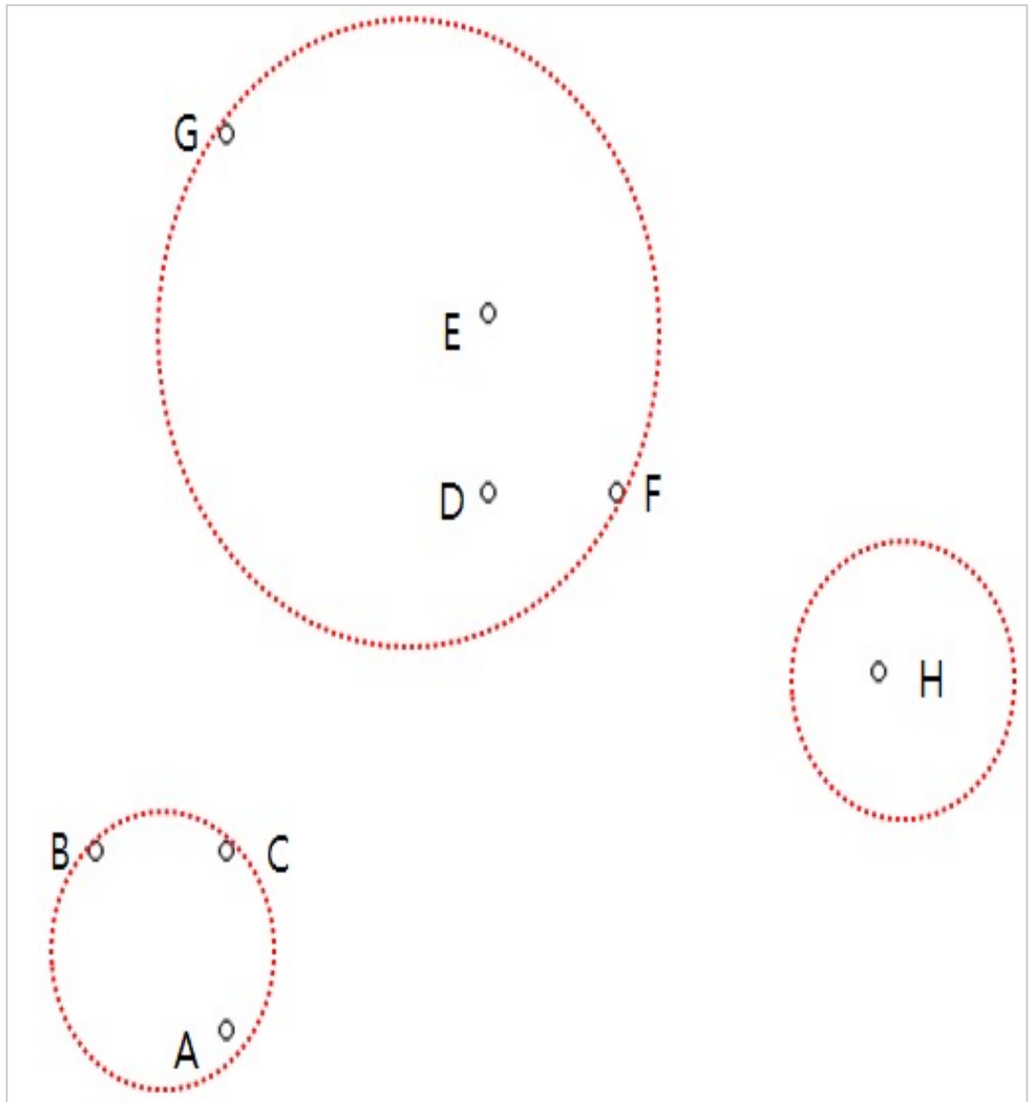
정의

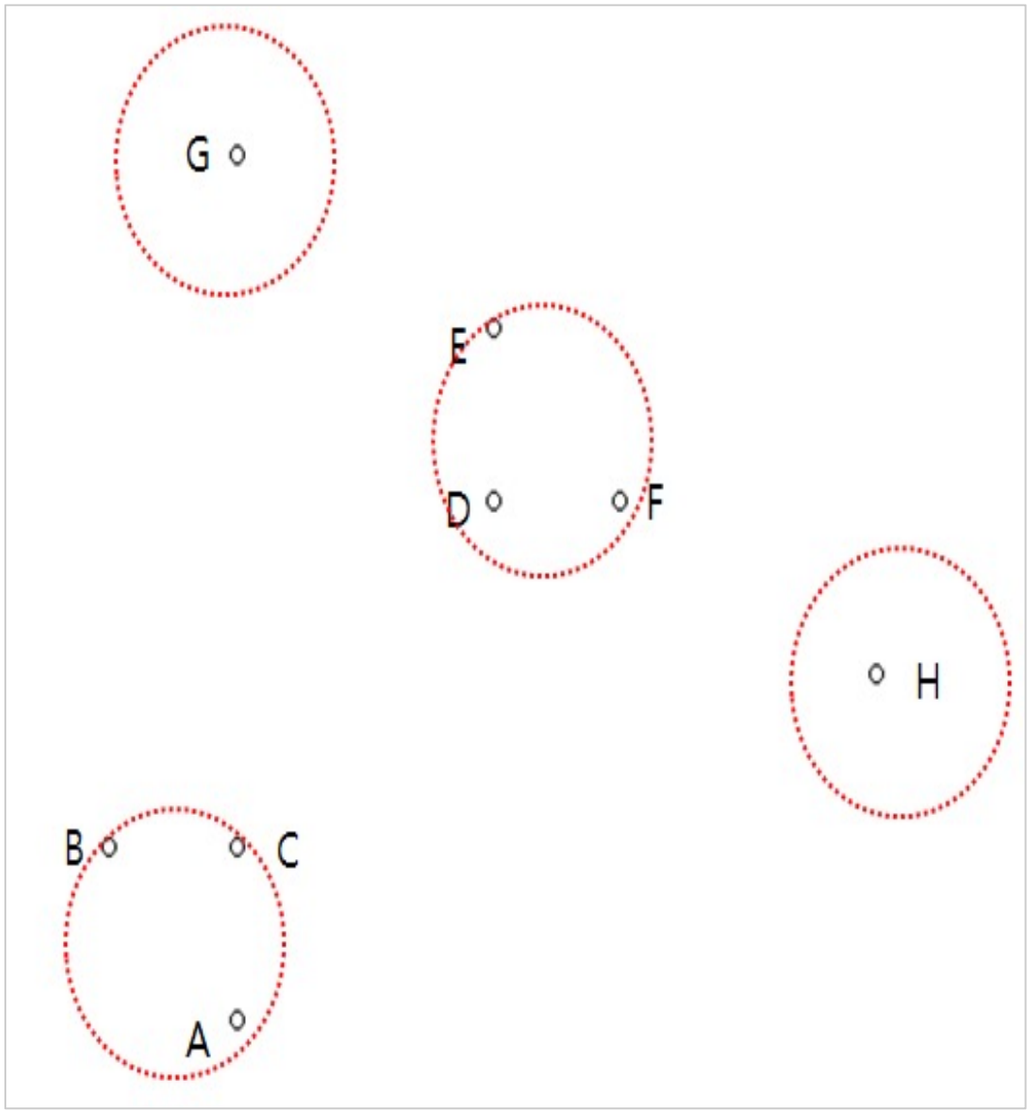
- 클러스터링(군집화: clustering)란 성격이 비슷한 항목들을 그룹으로 묶는 작업을 말한다.
- 군집화는 비지도학습이다.
  - 고객세분화는 고객의 타입을 나누어 마케팅에 대응하기 위해서이다.



클러스터 수, k

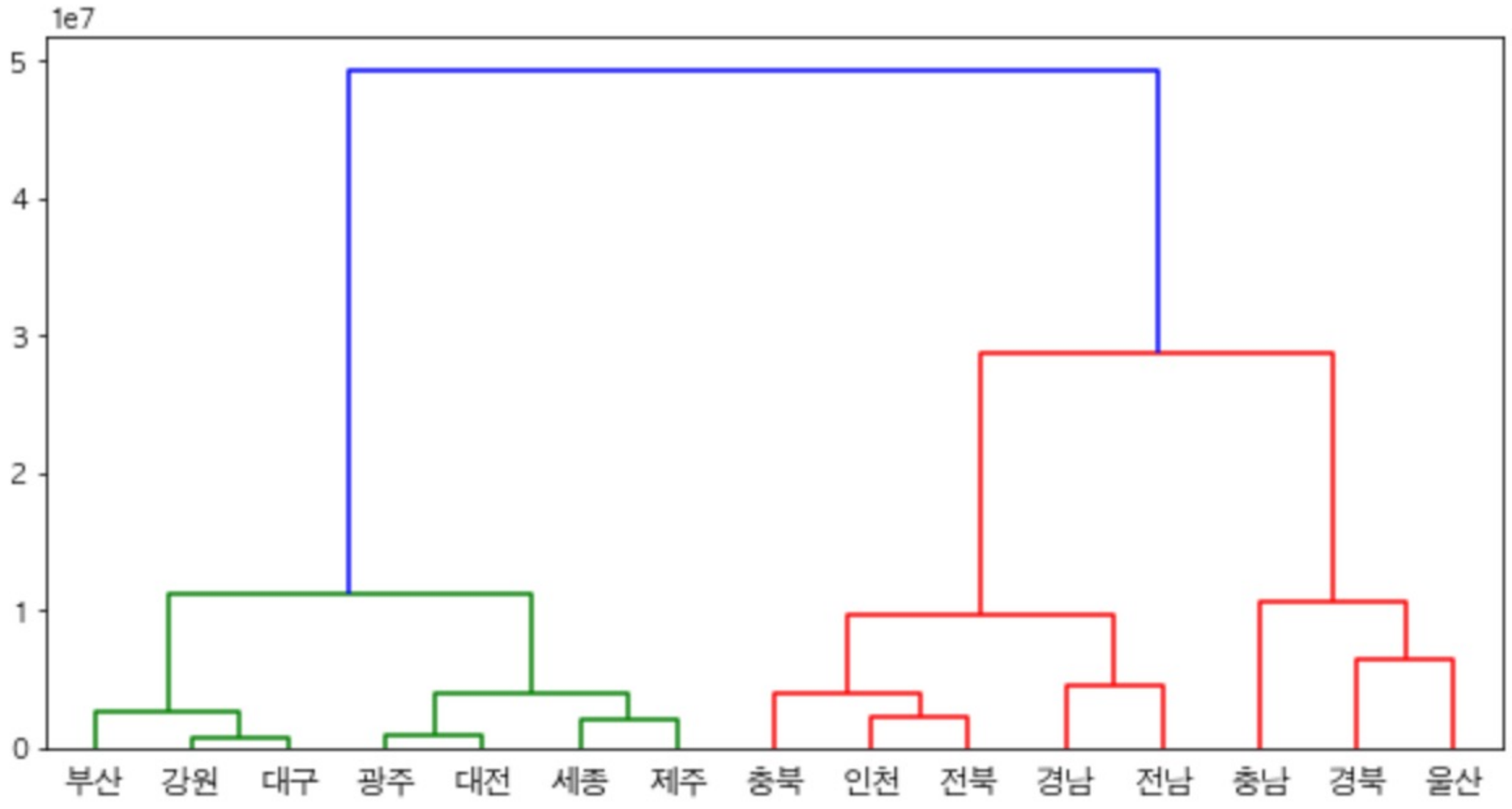
- 적정한 군집의 수(k)를 먼저 찾아야 함





덴드로그램

- 지자체별 전기사용량을 기준으로 유사한 지자체를 그루핑한다

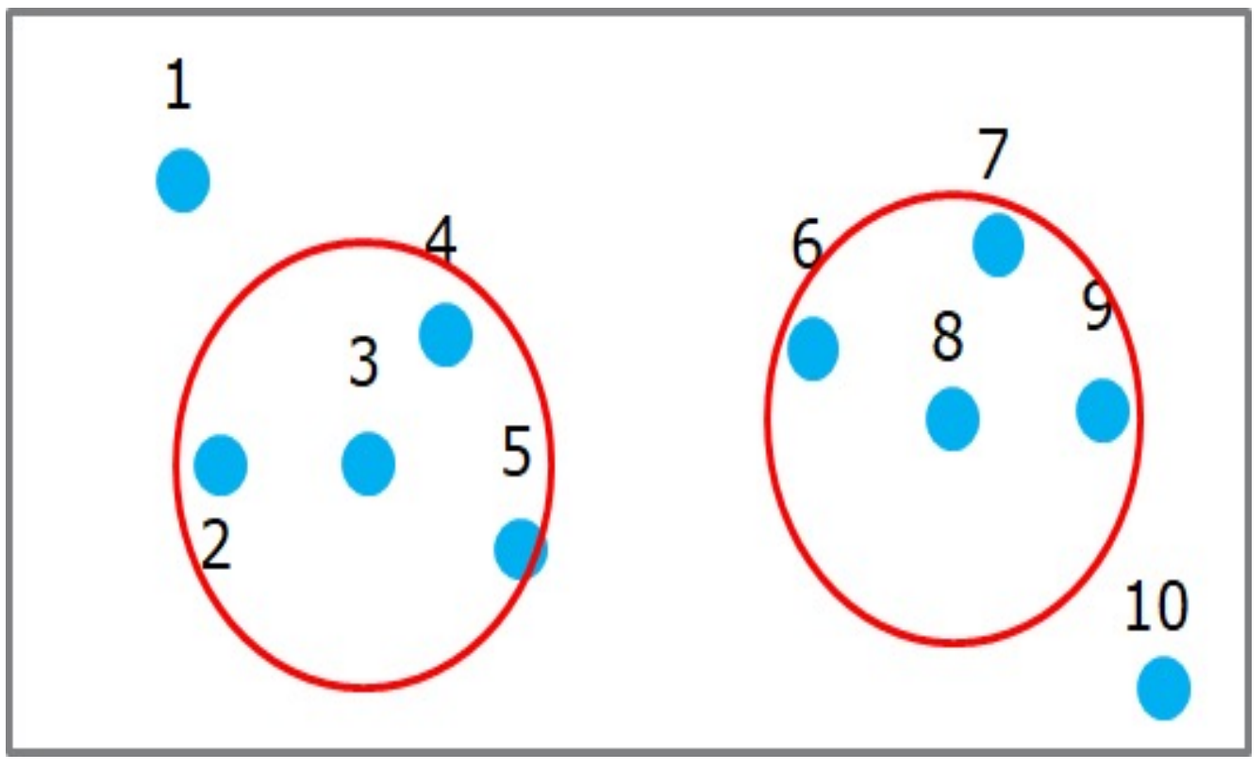


클러스터링 알고리즘

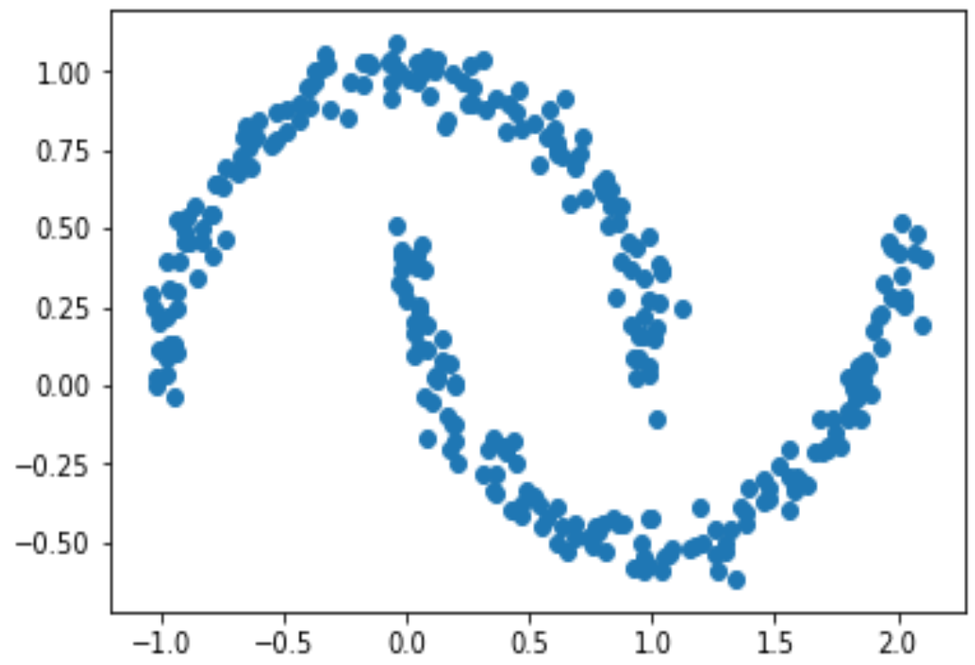
- 클러스터링 구성 조건
  - 같은 그룹 내의 항목들은 서로 속성이 비슷함 (유사도가 큼)
  - 다른 그룹에 속한 항목과는 속성이 서로 다름 (유사도가 작음)
- 비정상 패턴 (이상치) 식별에도 사용된다

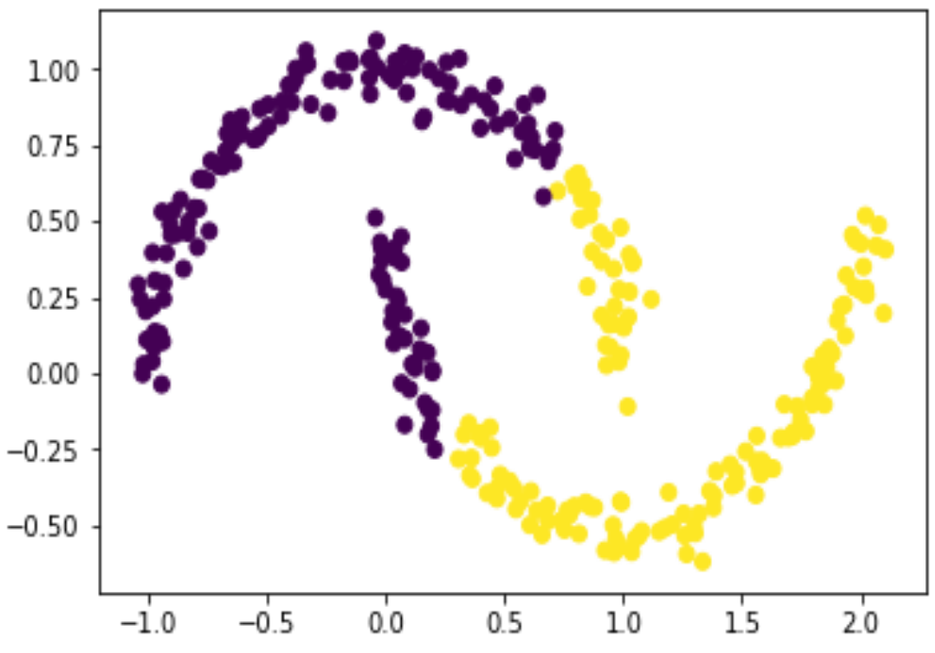
DBSCAN

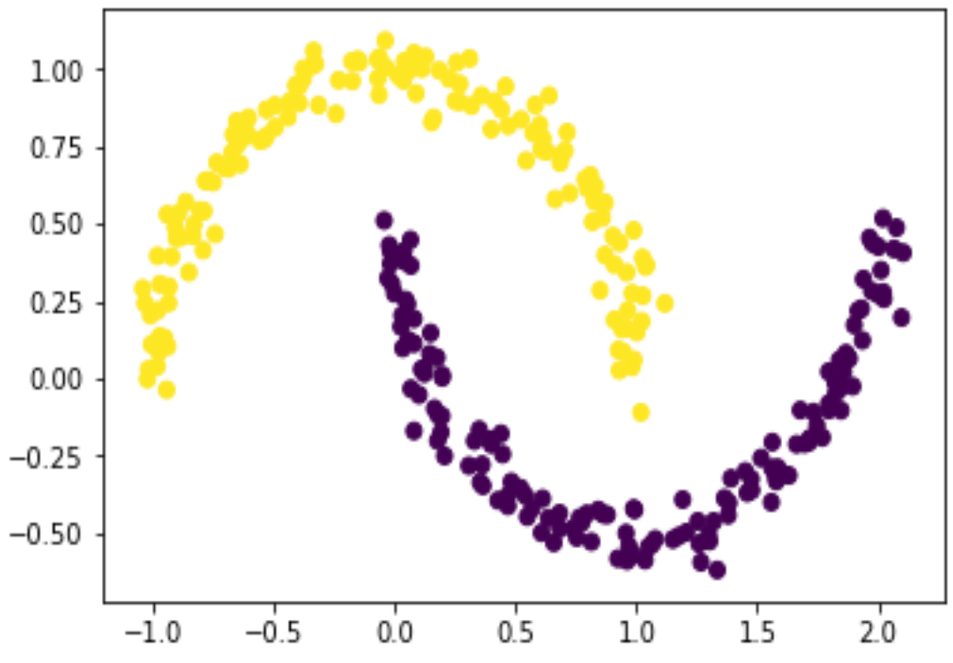
- DBSCAN은 밀도 기반 클러스터링 알고리즘이다.
- k-means처럼 단순히 거리만을 기준으로 군집화를 하는 것이 아니라 "가까이 있는 샘플들은 같은 군집에 속한다"는 원칙으로 군집을 차례로 넓혀가는 방식이다.
- 임의의 점을 기준으로 반경 r내에 점이 n개 이상 있으면 하나의 군집으로 인식



two-moons 데이터



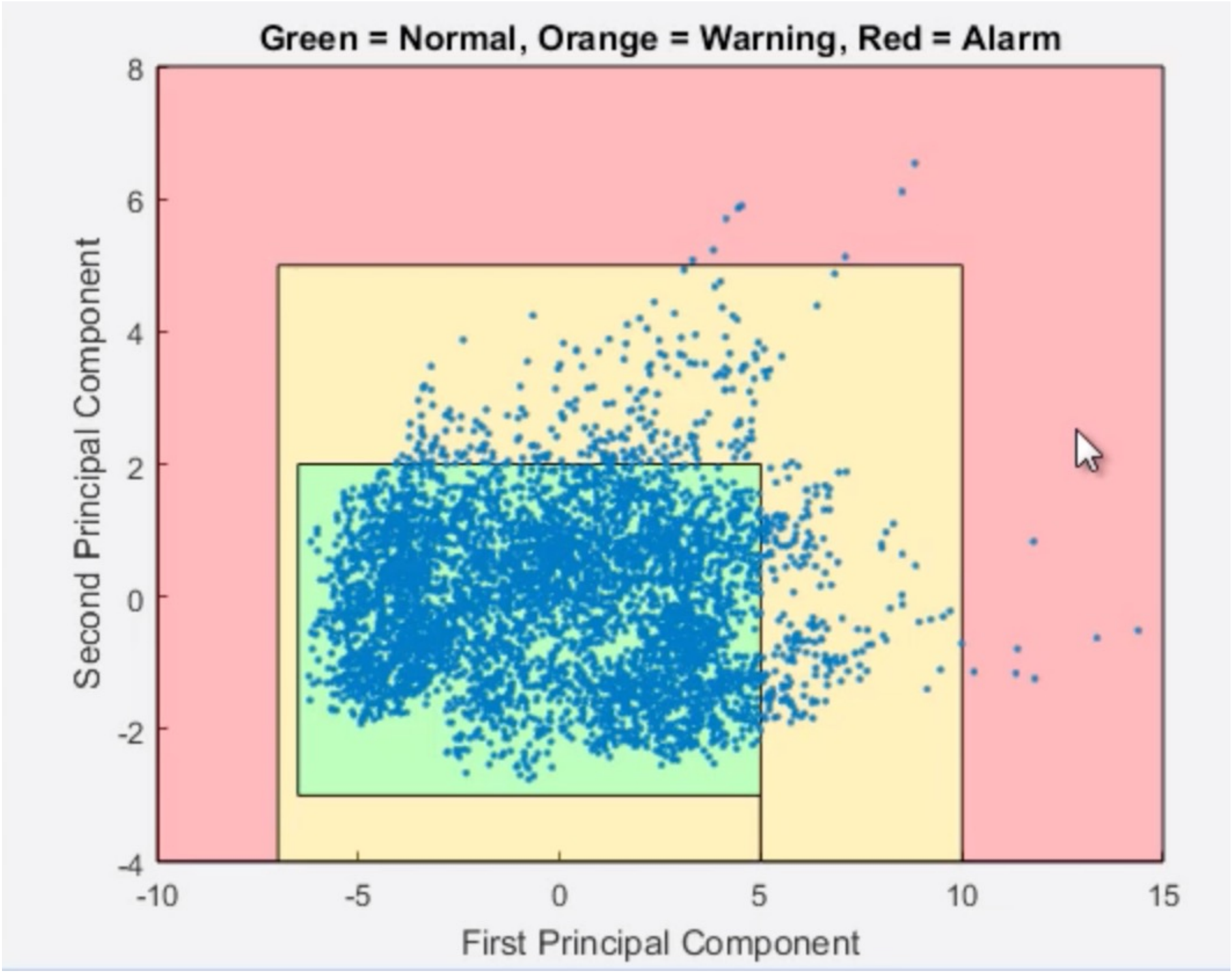




이상치 검출

- Outlier Detection
- 정답이 존재하는 것이 아니라, 특이하게 튀는 값을 찾는 것
- 이상치 타입
  - additive outlier: spikes
  - temporal changes: drops, trend changes
  - (seasonal) level shifts
- 이상치 검출 방법
  - 예측을 하고 이것이 신뢰구간을 크게 벗어나는지를 찾는 방법
  - 6시그마 이상 떨어진 분포를 찾는 방법
  - 이상치를 지도학습으로 훈련하는 방법 (데이터분균형 문제 발생)

PCA를 이용하는 방법


Green = Normal, Orange = Warning, Red = Alarm
Second Principal Component
First Principal Component
8
6
4
2
0
-2
-4
-10
-5
0
5
10
15

머신러닝 기초

머신러닝의 유형

- 설명(description)적 분석
  - 진단(원인, 해석)
  - 클러스터링
  - 비지도 학습
- 예측 (predictive) 분석
  - 회귀(regression)
  - 분류(classification)
  - 지도학습
- 강화학습
  - 학습하는 방법을 습득
- 추천(recommendation)
  - 처방 (prescriptive), 최적화

설명적(Descriptive) 분석

- 데이터로 표현되는 현상, 고객, 비즈니스 프로세스, 서비스, 등을 이해하는 것
  - 어떤 현상의 원인을 설명하는 것을 말한다.
  - 어떤 상품이 많이 팔렸다면 그 이유를 파악하는 것 등이다.
- 탐색적 분석과 시각화를 이용
- 예
  - 유사한 특성을 가진 항목들을 함께 묶는 군집화(Clustering)
  - market research
  - funnel analysis (고객의 행동 전환이 이루어지는 단계 분석)
- 결과는 인사이트를 얻는 것
- 전통적인 통계적 분석의 주된 목적이 설명적 분석이다.

예측(Predictive) 분석

- 예측분석에는 회귀와 분류가 있다.
- 회귀는 수치를 예측하는 것을 말한다
  - 내일의 날씨 예측, 주가 예측, 병에 걸릴 확률이 얼마일지,
  - 다음 달 가게의 매출이 얼마일지 등을 예측하는 것을 말한다
- 분류는 주어진 샘플이 어느 카테고리에 속할지를 예측하는 것
  - 수신한 메일이 스팸인지 아닌지를 구분하는 것,
  - 이번 은행 대출이 부도가 날지 아닐지,
  - 누가 우수 고객인지를 예측하는 것 등

지도 및 비지도학습

- 지도 학습(supervised learning)
  - 정답이 있는 학습
  - 예측 분석
  - 정답을 목적변수 (target variable) 또는 레이블(label)이라고 부른다.
- 비지도 학습 (unsupervised learning)
  - 정답이 없이 데이터로부터 중요한 의미를 찾아내는 머신러닝
  - 데이터의 특성을 기술하는 서술형 분석, 탐색적 분석, 시각화
  - 연관분석
  - 군집화: 유사한 항목들을 같은 그룹으로 묶는것
  - 데이터 변환: 데이터를 분석하기 좋게 다른 형태로 변환하는 것
  - 스케일링: 데이터의 범위를 바꾸는 것
  - 주성분분석(PCA): 머신러닝에 사용할 특성의 수를 줄이는 것
    - 특성공학(feature engineering)도 비지도 학습이다.

특성 공학

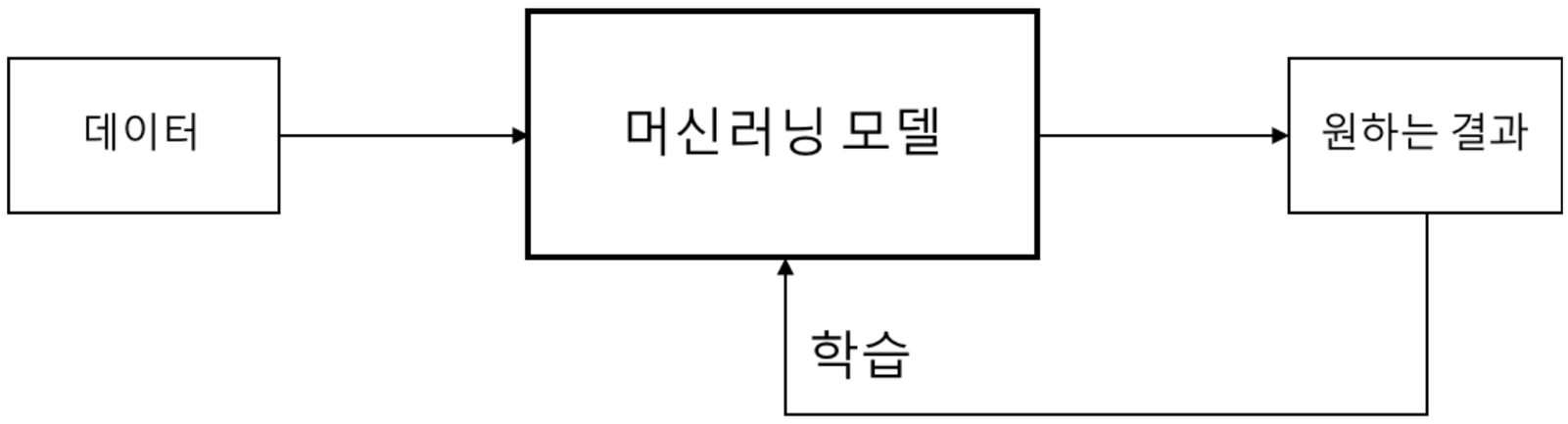
- Feature Engineering
- 분석에 적절하게 특성을 일부 선택하거나 변형하는 것
  - 같은 정보량을 가지면서 데이터의 크기를 줄이는 것
  - 불필요한 데이터를 제거하여 분석 속도와 성능 개선
  - BMI(비만도) 지수 = 몸무게 / 키*키
    - BMI>=30 BMI=1(비만), 아니라면 BMI=0(정상)
  - 사람에 의한 특성 선택
  - 차원 축소
  - 주성분 분석 (PCA)

머신러닝 모델

- 머신러닝에서는 모델을 만드는 것이 핵심이다.
  - 스팸 메일을 찾아내는 모델
  - 수익 예측 모델
  - 도난 카드 사용 검출 모델
- 수식은 가장 명확한 모델이다.
  - 그러나 현실의 많은 현상은 수식으로 간단히 모델링하기 어렵다
  - 데이터 기반의 모델을 만들어야 한다
- 모델은 구조와 파라미터로 구성된다.
  - 모델 구조: 모델의 동작을 정하는 방법
  - 모델 파라미터: 모델이 잘 동작하도록 정한 가중치 등 계수

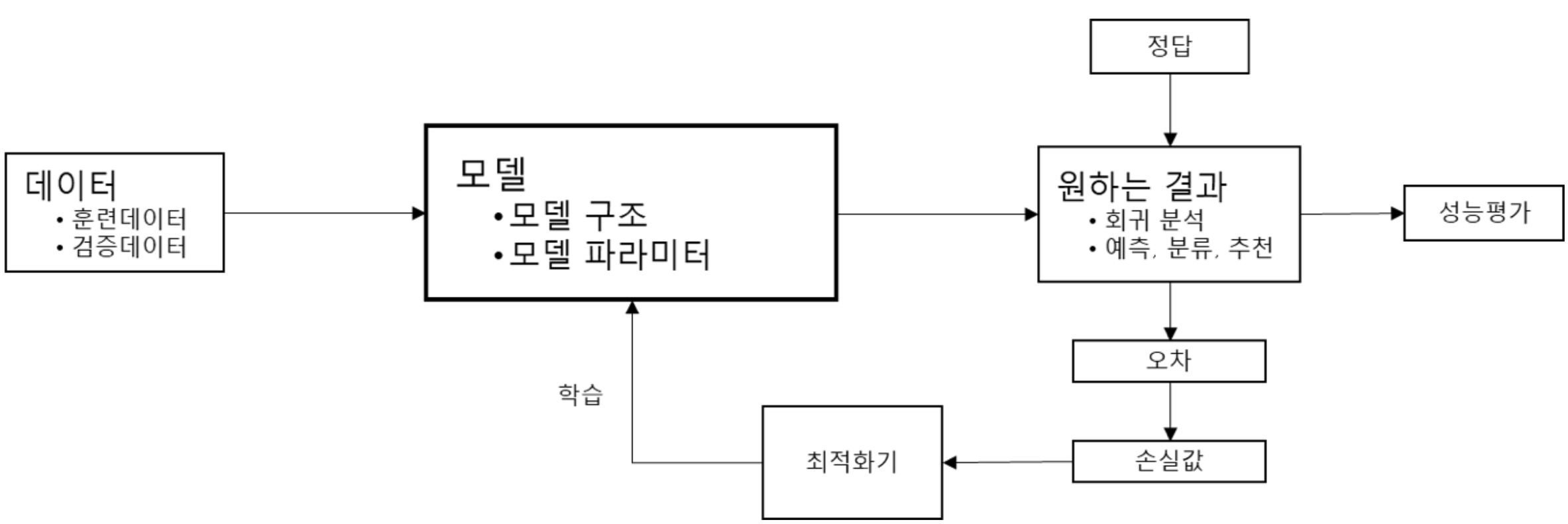
머신러닝 모델

- 모델이 잘 동작한다는 것은 회귀나 분류를 정확하게 하여 원하는 결과를 얻는 것을 말한다.
- 학습을 통하여 모델의 성능을 개선한다



모델의 동작

- 모델은 원하는 결과가 나오는지를 관찰하고 모델의 파라미터를 학습을 통해서 업데이트 한다.
- 모델은 손실함수(Loss Function)를 최소화 하도록 학습한다.
  - 손실함수란 원하는 결과가 나오지 않은 정도, 오차 등으로부터 계산된다.



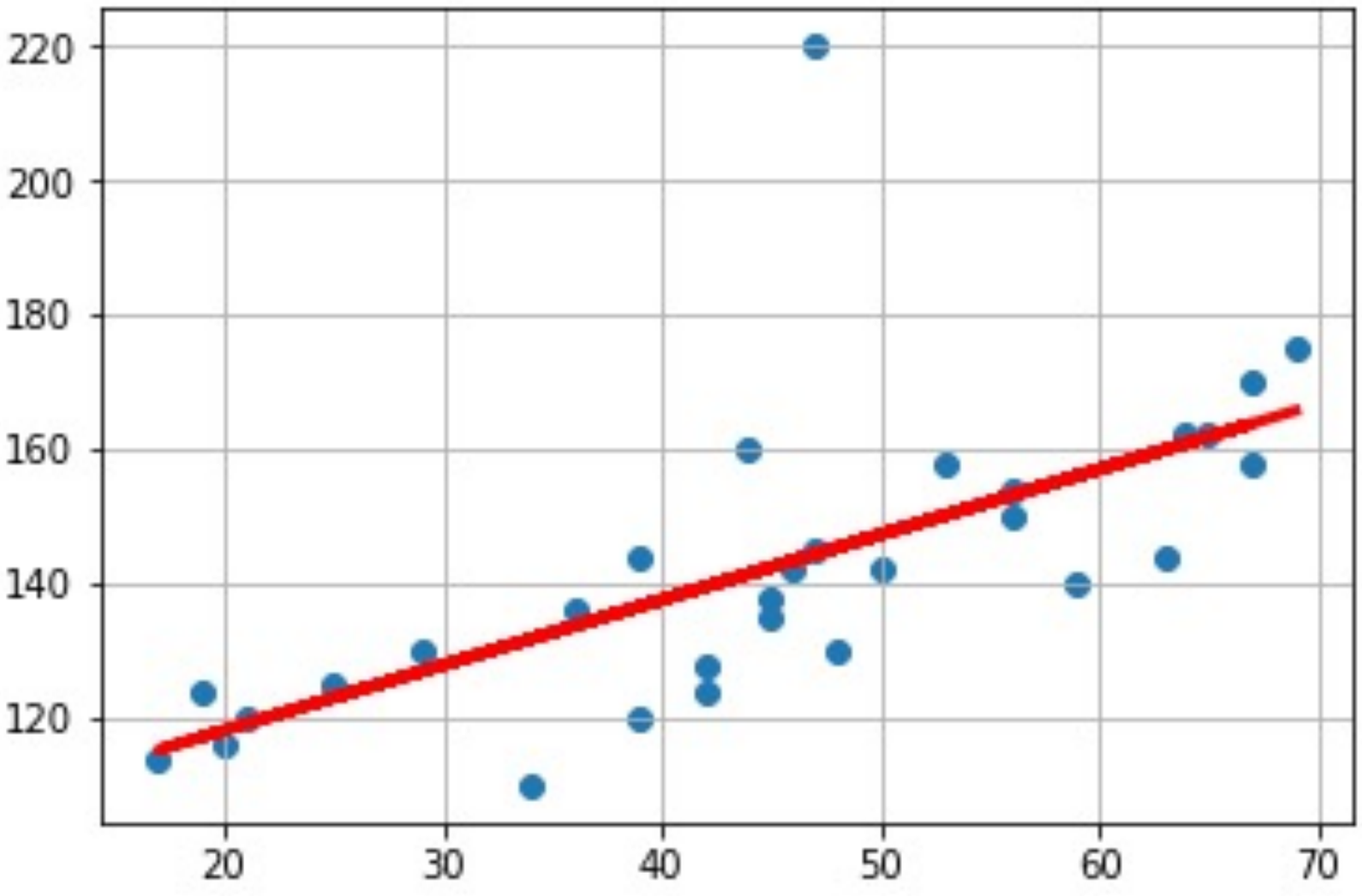
훈련과 검증

- 모델이 데이터를 이용하여 학습하는 과정을 훈련 (training)이라고 한다.
- 모델 파라미터 값 (예를 들어 선형 회귀에서 가중치의 값)의 초기값은 보통 랜덤한 값을 준다. 따라서 초기 모델의 손실함수도 크게 나타난다.
- 모델을 훈련시킨 후에는 모델이 제대로 동작하는지 검증을 해야 하는데 이때는 모델을 만드는 데 사용하지 않은 새로운 데이터로 검증해야 한다.

머신러닝 알고리즘

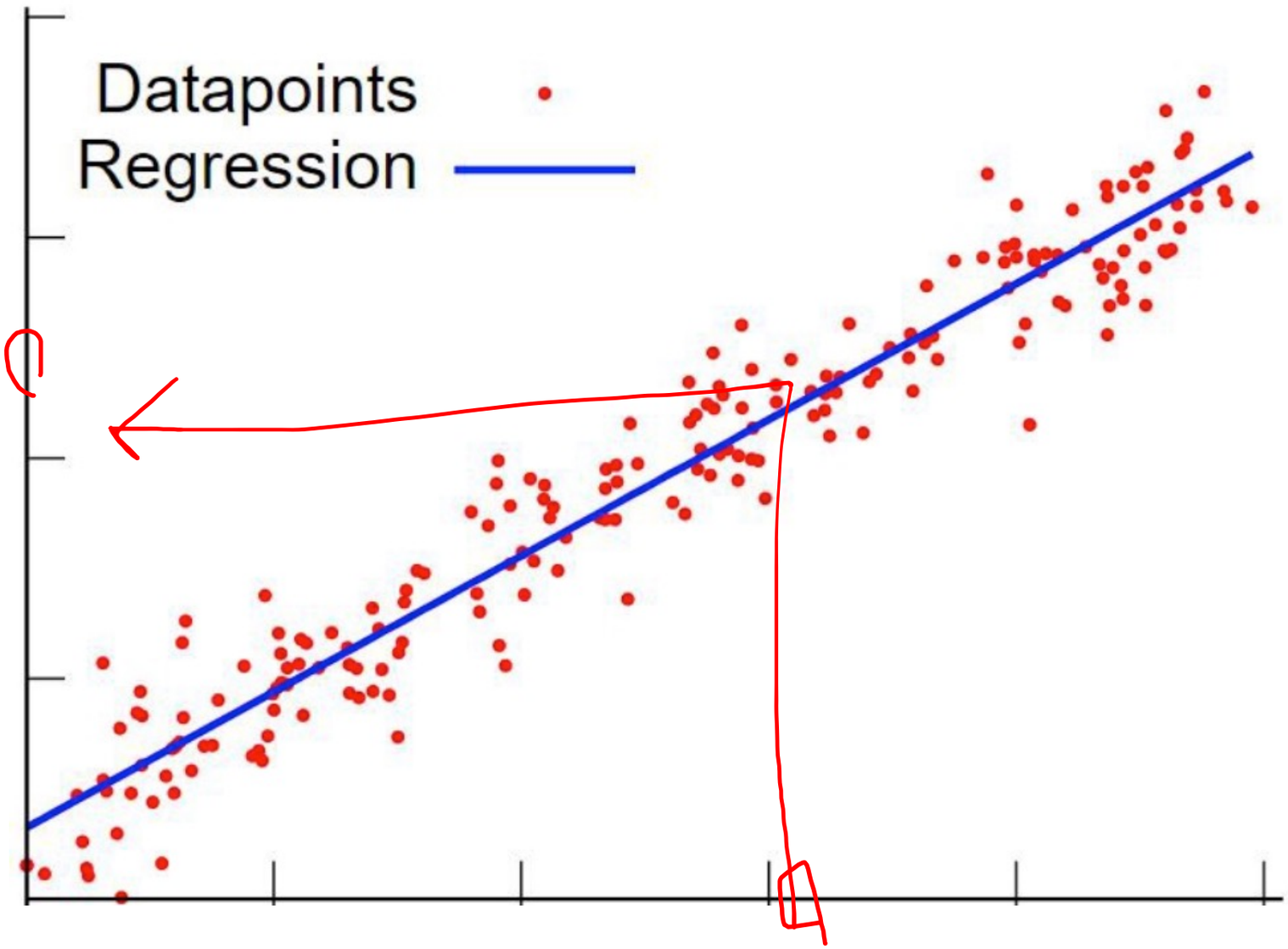
선형 모델

- 선형 회귀 모델
  - 입력 변수의 선형 가중합으로 즉 곱셈과 덧셈만으로 출력변수를 예측하는 모델 (아래는 나이와 혈압의 관계 예측 모델)



선형 회귀

- 선형 회귀(regression) y = wX + b



선형 회귀

- 입력(x)와 출력(y) 변수간의 관계를 대표하는 회귀직선

$$y = ax + b$$

- 입력이 여러 특성값으로 구성되어 있으면 다중 선형 회귀(multiple linear regression)라고 한다.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots . + \beta_p x_p$$

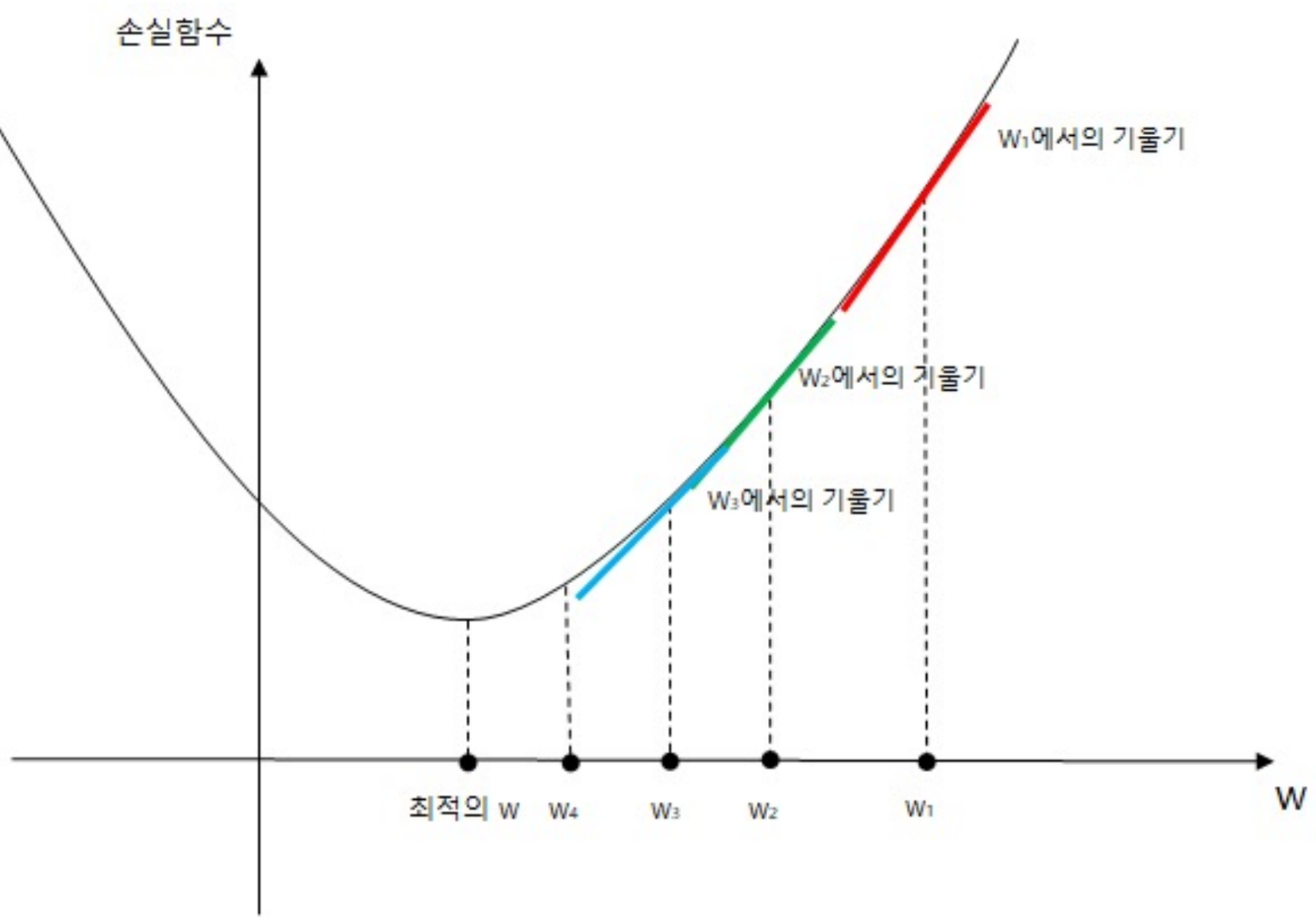
- 출력(outcome) 변수를 종속(dependent) 변수, 목적(target) 변수, response 변수, 또는 label이라고도 부르며,
- 입력(input) 변수를 독립(independent) 변수, 예측(predictor) 변수, 설명 변수(explanatory variable) 또는 feature라고도 한다.

최적화

- 모델의 최적화(optimizer)란 학습을 통하여 파라미터 (예: 선형 회귀 모델의 가중치)를 최적의 값으로 수렴시키는 것을 말한다.
- 여기서 최적의 값이란 손실함수를 가장 줄이는 계수 값을 말한다.
- 최적화 알고리즘으로는 경사하강법(GD: Gradient Descent)이 기본적으로 사용된다

경사하강법

- 경사하강법(Gradient Descent)이란 손실함수 공간에서 기울기(gradient, 미분값)에 비례하여 반대방향으로 이동하는 방식을 말한다.



경사하강법

- 이를 수식으로 표현하며 다음과 같다.

$$W_i = W_{i-1} - \eta Grad(i)$$

- 위에서 Grad(i)는 시점 i에서 손실함수의 기울기이고, η 는 학습 속도를 조절하는 학습률(learning rate)이다. 계수 W의 초기 값은 랜덤하게 준다.
- 경사하강법을 적용하려면 특성 들을 모두 스케일링해야 한다.
- 경사하강법은 크게 배치(Batch)와 통계적(Stochastic) GD (SGD) 두 가지가 있다.

손실함수와 성능지표

손실함수와 성능지표

- **손실함수**를 정하는 목적은 모델을 훈련시킬 때의 기준으로 삼기 위해서이다.
  - 모델은 손실함수를 최소화 하는 방향으로 학습한다.
- **모델의 성능지표**는 이렇게 만든 모델이 궁극적으로 얼마나 잘 동작하는지를 평가하는 척도이다.

회귀모델의 손실함수

- 모델의 예측값과 실제 값과의 차이, 즉 오차로부터 손실함수(loss function)을 계산한다
- 이 손실함수를 줄이는 방향으로 모델을 최적화 (학습) 한다
- 회귀분석에서 많이 사용하는 손실함수로는 오차 자승의 합의 평균치(MSE: mean square error)

$$MSE = \sum_{k=1}^{N} (y - \hat{y})^2$$

  - N: 배치 크기

- 배치 크기 같은 설정 환경 변수를 하이퍼파라미터라고 한다.
  - **하이터파라미터**는 사람이 선택하는 변수이며, 기계 학습으로 자동으로 갱신되는 변수는 "**파라미터**"라고 한다.

기타 회귀 손실함수

- 회귀 모델에서 MSE 외에 다음과 같은 손실함수를 사용할 수 있다.
  - MAE – Mean Absolute Error
  - MAPE– Mean Absolute Percentile Error
  - MSLE– Mean Absolute Log Error
- 사용용도
  - MSE는 큰 오차에 페널티를 많이 부과하는 것
  - MAE는 모든 크기의 오차를 동일하게 다루는 것
  - MALE는 작은 크기의 오차에 더 민감하게 반응하는 것

회귀 성능 평가 지표

- MSE 값은 오차의 크기를 알 수는 있으나 회귀 모델의 최종 성능 평가에 사용하기는 어렵다.
- 회귀모델의 성능 평가에는 R-square가 사용된다.
- R-Sqaure R^2

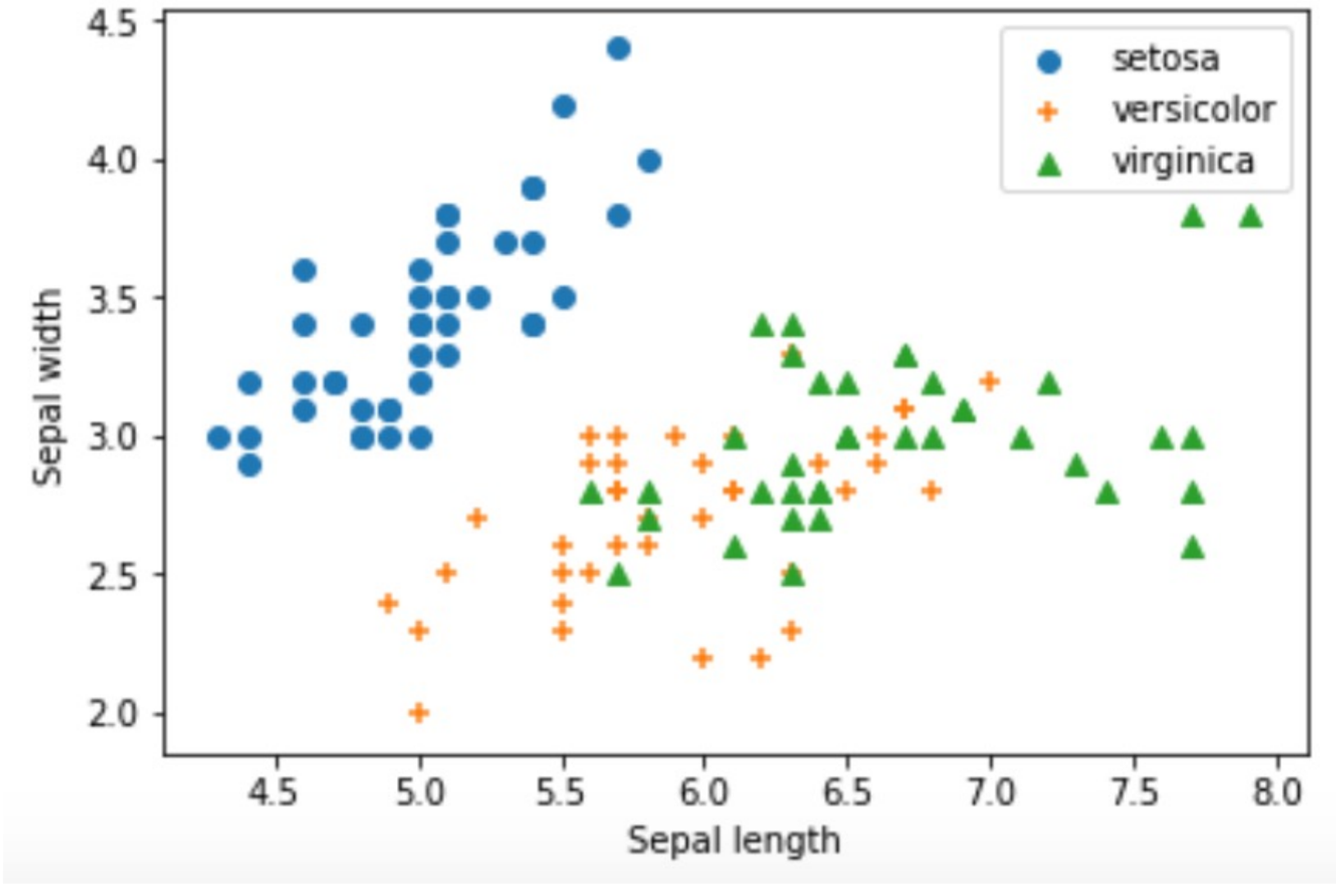
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

- 위에서 y_i 는 실제 값, $\hat{y}_i$ 는 예측한 값, $\bar{y}$ 는 샘플의 평균 값을 나타낸다.

분류

선형 분류 모델

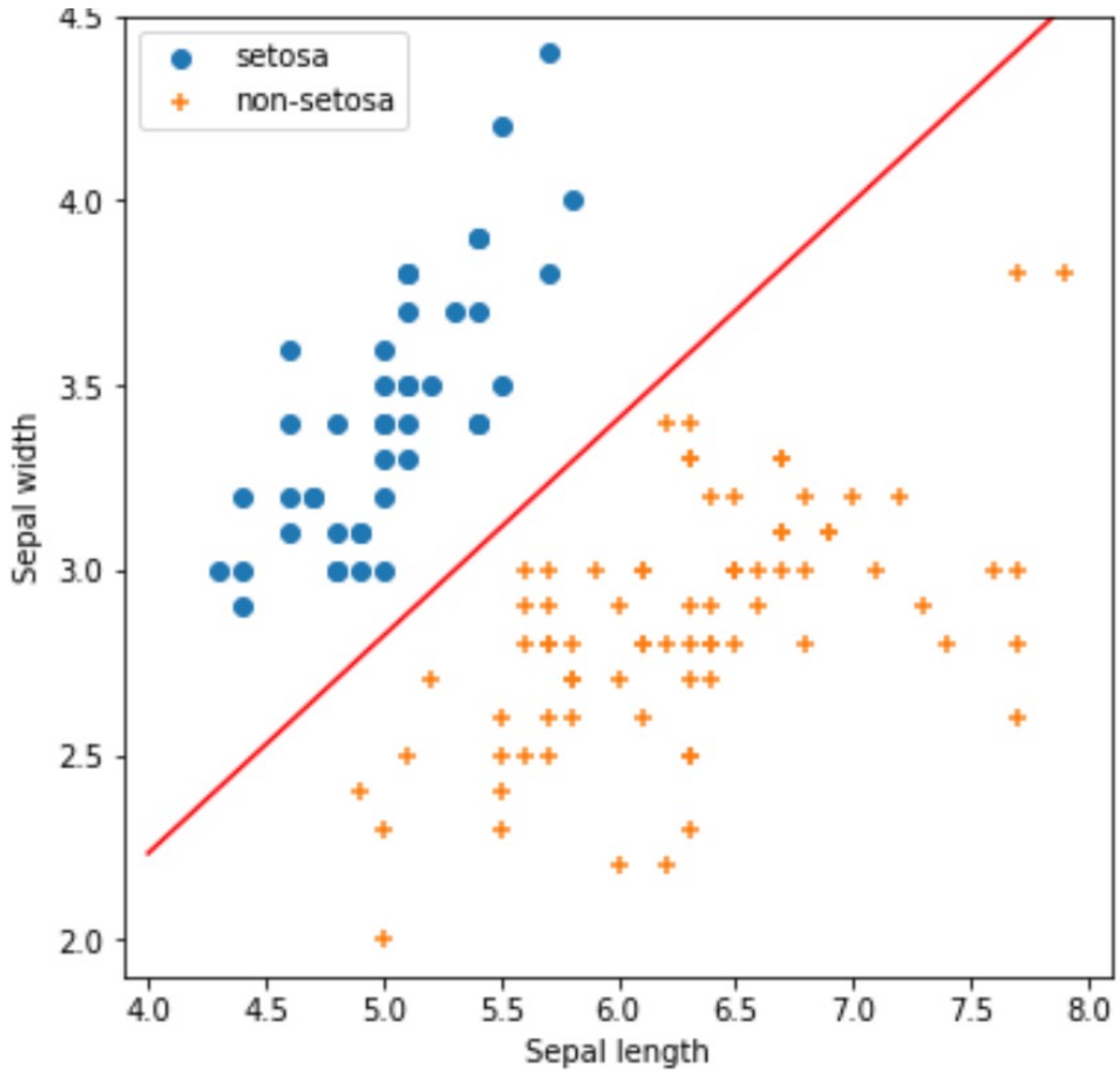
- 이진 분류와 다중 분류
  - 두 가지 카테고리를 나누는 작업을 이진 분류(binary classification)라고 하고 세 개 이상의 클래스를 나누는 작업을 다중 분류(multiclass classification)라고 한다.



결정경계

$$ax + by + c > 0$$

$$y > (-a/b)x - c/b$$



분류의 손실함수

- 분류에서는 손실함수로 MSE를 사용할 수 없다.
- 분류에서 정확도(accuracy)를 손실함수로 사용할 수 있다
  - 예를 들어 100명에 대해 남녀 분류를 시도하였으나 96명을 맞추고 4명을 틀렸다면 정확도는 0.96이다.
  - 그러나 정확도를 손실함수로 사용하는데에는 다음과 같은 문제가 있다.
- **카테고리 분포 불균형**시 문제
  - 남자가 95명, 여자가 5명이 있는 그룹에서 남자는 1명을 잘 못 분류하고 여자는 3명을 잘 못 분류했다고 하면, 정확도는 여전히 0.96이다
  - 손실을 제대로 측정하지 못한다
  - 이를 보완하기 위해서 크로스 엔트로피(cross entropy)를 사용한다.

크로스 엔트로피

$$CE = \sum_i p_i \log(\frac{1}{p_i'})$$

- Pi는 어떤 사건이 일어날 실제 확률이고, Pi' 는 예측한 확률이다

- 남녀가 50명씩 같은 경우

$$CE = -0.5 \times \log(\frac{49}{50}) - 0.5 \times \log(\frac{47}{50}) = 0.02687$$

- 남자가 95명 여자가 5명인 경우

$$CE = -0.95 \times \log(\frac{94}{95}) - 0.05 \times \log(\frac{2}{5}) = 0.17609$$

분류의 성능지표

- 정확도(accuracy)를 기본으로 측정하나 필요에 따라 다음과 같은 값을 측정한다.
- 컨퓨전 매트릭스를 만들어봐야 한다. (뒤에서 설명함)
  - 정밀도 (precision)
  - 재현률 (recall)
  - F-1 점수
  - ROC
  - AUC

분류 성능 지표

- 발생 빈도가 비 대칭적인 경우는 정확도만 봐서는 성능을 파악할 수 없다.
- 분류에서는 정확도 뿐 아니라 정밀도(precision), 재현률(recall), F1점수(F1-score) 등의 지표를 사용한다.
- classification_report()를 사용한다.

	precision	recall	f1-score	support
0	0.75	0.86	0.80	7
1	0.95	0.91	0.93	23
avg / total	0.91	0.90	0.90	30

조화 평균

- 산술 평균과 달리 취약한 점수에 페널티를 더 주는 방식

$$\text{조화 평균:} \quad \frac{1}{c} = \frac{(\frac{1}{a} + \frac{1}{b})}{2}$$

$$c = \frac{2ab}{a+b}$$

혼돈 매트릭스

- 혼돈 매트릭스(confusion matrix)란 분류의 결과가 잘 맞았는지를 평가하는 채점표와 같다.
- 이진 분류의 경우 예측한 내용과 실제 타겟 값이 맞는지를 따져보면 총 네 가지 경우의 수가 존재한다.

실제 \ 예측	p로 예측	n으로 예측
실제로는 p	True positive (TP)	False negative (FN)
실제로는 n	False positive (FP)	True negative (TN)

```
print(metrics.confusion_matrix(y_test, y_test_pred))
[[6 1]
 [ 2 21]]
```

정확도와 정밀도

- 정확도(accuracy)
  - 정확도란 positive 및 negative 두가지에 대해서 전체적으로 정확히 예측한 비율을 말한다.

    정확도 = (positive 또는 negative를 맞춘 수) / (전체 샘플 수 N)

    = (TP+TF) / (TP + FP + FN + TN)

- 정밀도(precesion)
  - 정밀도란 머신러닝 모델이 positive라고 예측한 것 중에 positive인 비율을 말한다.

    정밀도 = (positive를 맞춘 수) / (positive라고 예측한 수)

    = TP/(TP+FP)

재현율과 F1 점수

- 재현률(recall)
  - 재현률은 전체 positive 샘플 중에 모델이 positive라고 찾아낸 비율을 말한다. 재현율은 관심 있는 대상을 얼마나 찾아내는지를 나타낸다.

재현률 = (positve를 맞춘 수) / (positive 전체 수)

= TP / (TP+FN)

- F1 지표
  - 재현률과 정밀도 두 성능평가 지표를 동시에 측정하는 방법
  - 정밀도와 재현률의 조화평균(harmonic average)으로 정의한다.

F1 = 2 x precision x recall / (precision + recall)

동적 성능 평가

- 분류의 최종 성능 평가 뿐 아니라, 예측한 순서 자체를 평가하는 것이 필요하다.
- ROC 곡선
  - ROC(receiver operating characteristics) 곡선은 분류 예측 순서까지 고려하여 평가할 수 있는 수단이다.
  - ROC 커브는 예측한 점수순으로 순으로 한 행씩 내려오면서 정답과 오류를 차례대로 확인하는 방식으로 그린다.
  - x축은 false positive rate를 y축은 true positive rate를 나타낸다
- AUC(Area Under the Curve)
  - ROC 성능을 간단히 수치로 나타내기 위해서 ROC 그래프의 면적을 AUC(Area Under Curve)로 계산한다.

손실함수와 성능지표

- 대표적인 손실함수와 성능 평가 지표
  - 분류에서는 정확도 외에 성능 지표로 정밀도(precesion), 재현률(recall), F1-지수 등이 추가로 사용된다.

	손실함수	성능평가지표
정 의	손실함수를 줄이는 방향으로 모델이 학습을 함	성능을 높이는 것이 머신러닝을 사용하는 목적임
회귀 모델의 대표적인 값	MSE (오차 자승의 편균치)	R^2
분류 모델의 대표적인 값	크로스 엔트로피	정확도, 정밀도, 재현률, F1점수

훈련과 검증

- 모델이 데이터를 이용하여 학습하는 과정을 훈련 (training)이라고 한다.
- 모델을 훈련시킨 후에는 모델이 제대로 동작하는지 검증을 해야 하는데 이때는 모델을 만드는 데 사용하지 않은 새로운 데이터로 검증해야 한다.
- 데이터 구분
  - 훈련 (train)데이터 – 모델 파라미터를 훈련하는데 사용
  - 검증 (validation) 데이터 – 모델이 제대로 동작하는지 즉, 과대적합이나 과소적합을 검사하고 최적 모델 구조를 찾는데 사용
  - 테스트 (test) 데이터 – 모델의 성능을 최종적으로 테스트 하는데 사용

과대적합

- 모델이 훈련 데이터에 대해서만 잘 동작하도록 만들어져서 새로운 데이터에 대해서는 오히려 잘 동작하지 못하는 경우를 과대적합(over fitting)되었다고 한다.
- 과대적합은 주어진 훈련 데이터를 너무 세밀하게 학습에 반영하여 발생하는 현상이다.
- 머신러닝에서는 과대적합을 피해서 일반적으로 잘 동작하게 모델을 만드는 것이 매우 중요하다.
  - 이를 모델의 일반화(generalization)라고 한다.

과대적합 - 규제화

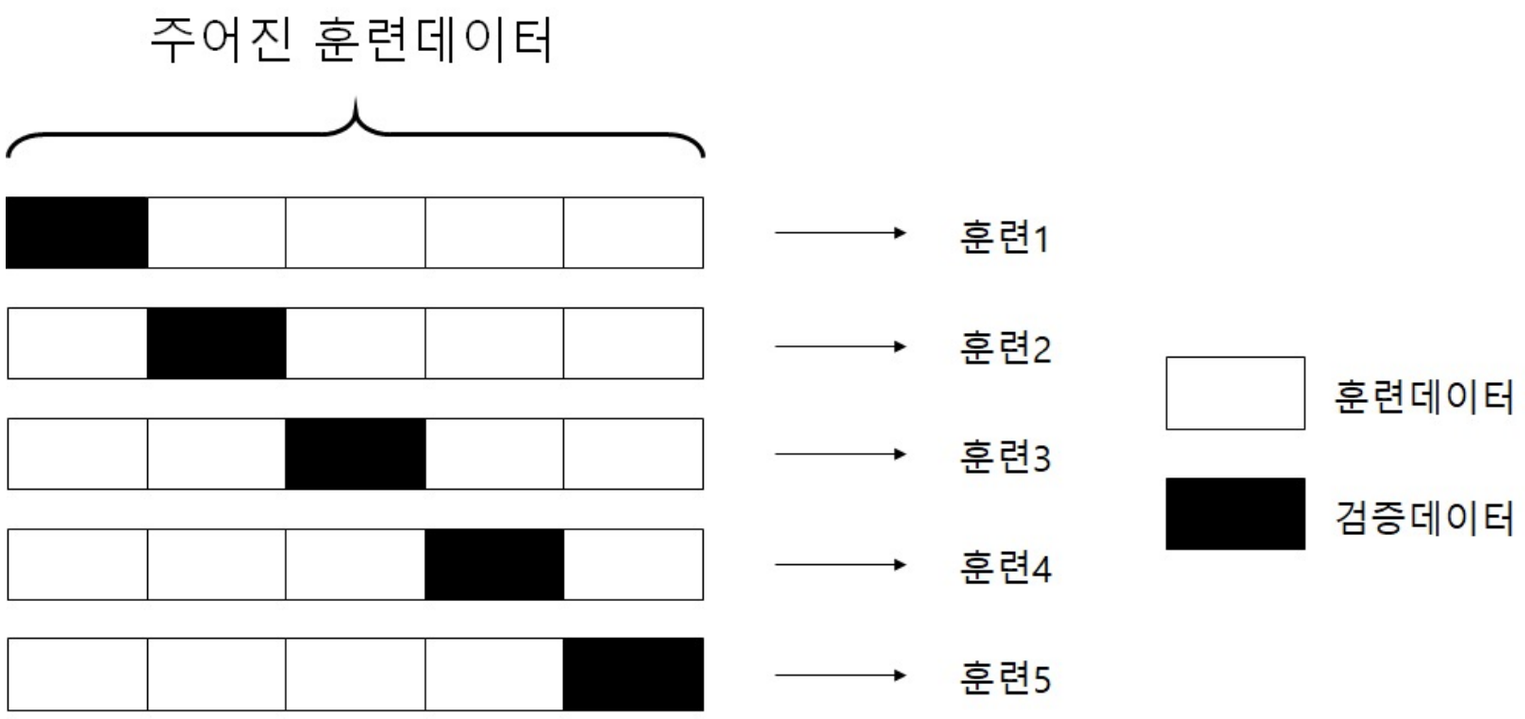
- 과대적합을 줄이려면 훈련 데이터를 많이 확보하여 다양한 경우를 대비하여 일반적인 모델을 만들어야 한다.
- 만일 학습할 데이터가 부족하다면 모델 구조를 좀 단순하게 만들어야 한다.
- 이렇게 모델이 일반성을 갖도록 기능을 제한하는 것을 모델에 제약을 가한다고 하여 규제화(regularization)라고 한다.

과소적합

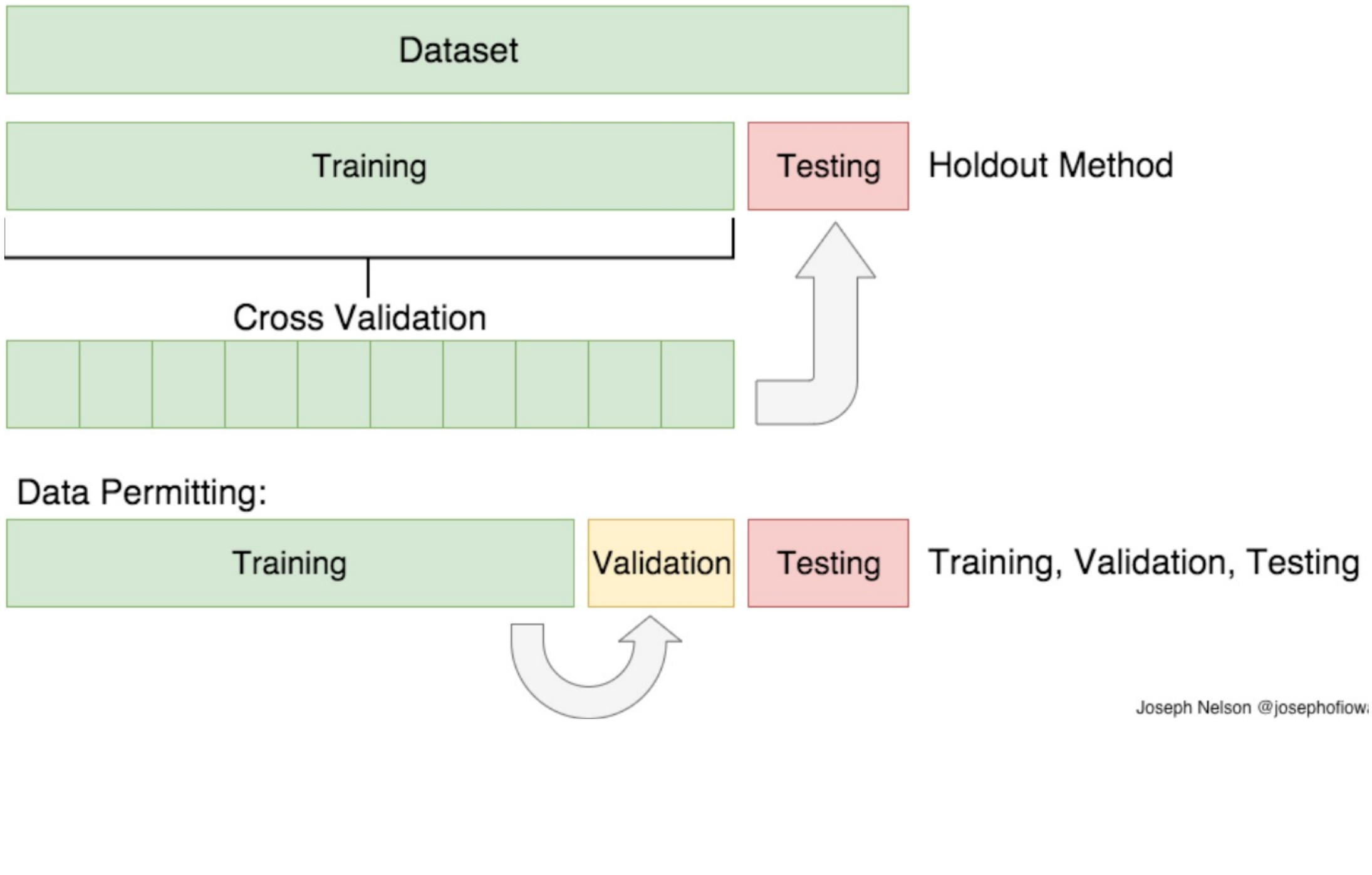
- 과대적합과 반대로 모델이 너무 간단하여 성능이 미흡한 경우를 과소적합(under fitting)이라고 한다.
- 과소적합을 피하려면 좀 더 상세한 모델 구조를 사용해야 한다.
- 머신러닝에서는 과대적합과 과소적합을 모두 피해야 하며 최적의 예측을 수행하는 모델을 만드는 것이 중요하다.'

교차 검증

- cross validation
- 주어진 데이터를 훈련 및 테스트 데이터로 한번만 나누는 방식을 보완
- K-fold 교차 검증이 가장 널리 사용된다.

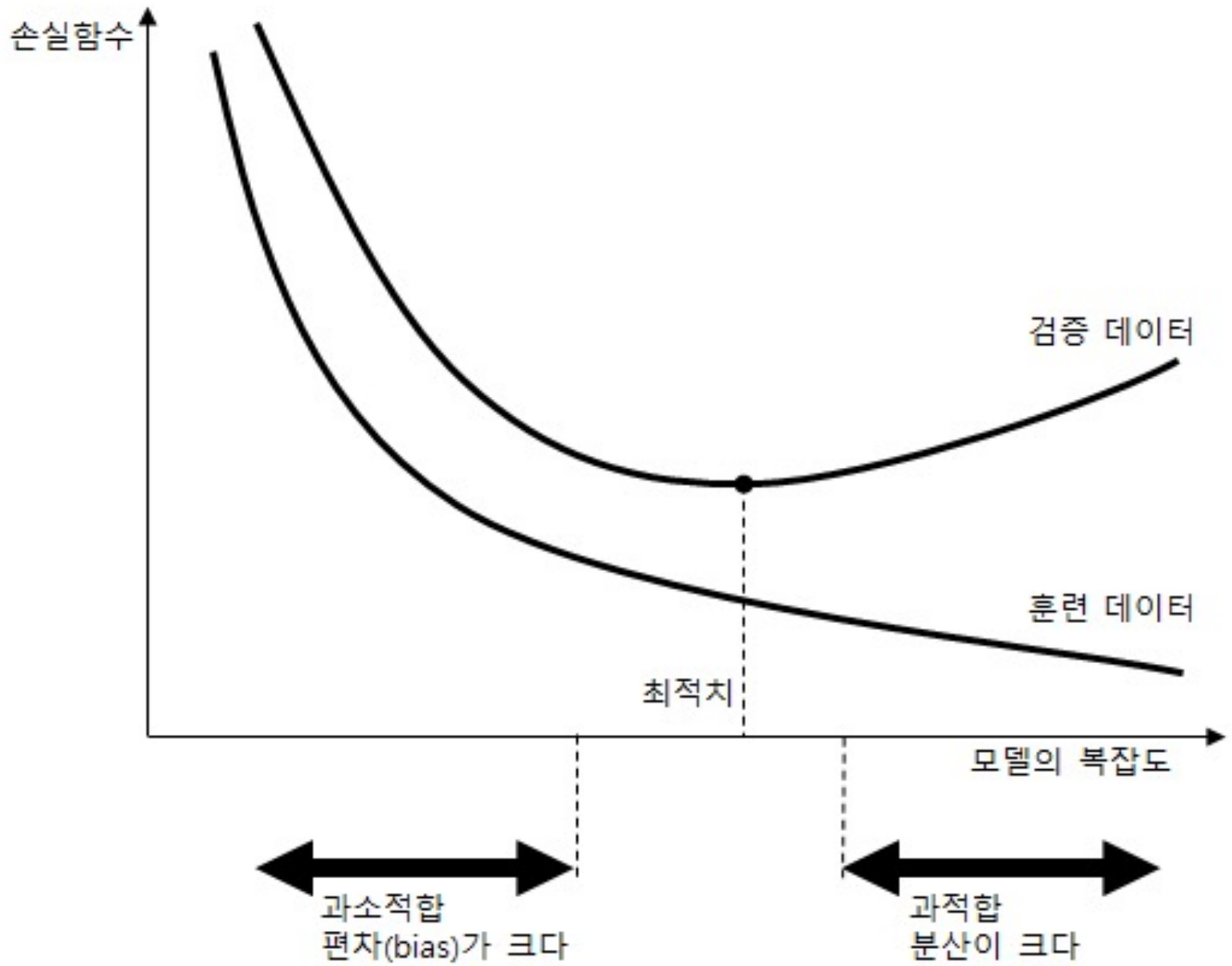


훈련, 검증, 테스트 데이터


Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiow

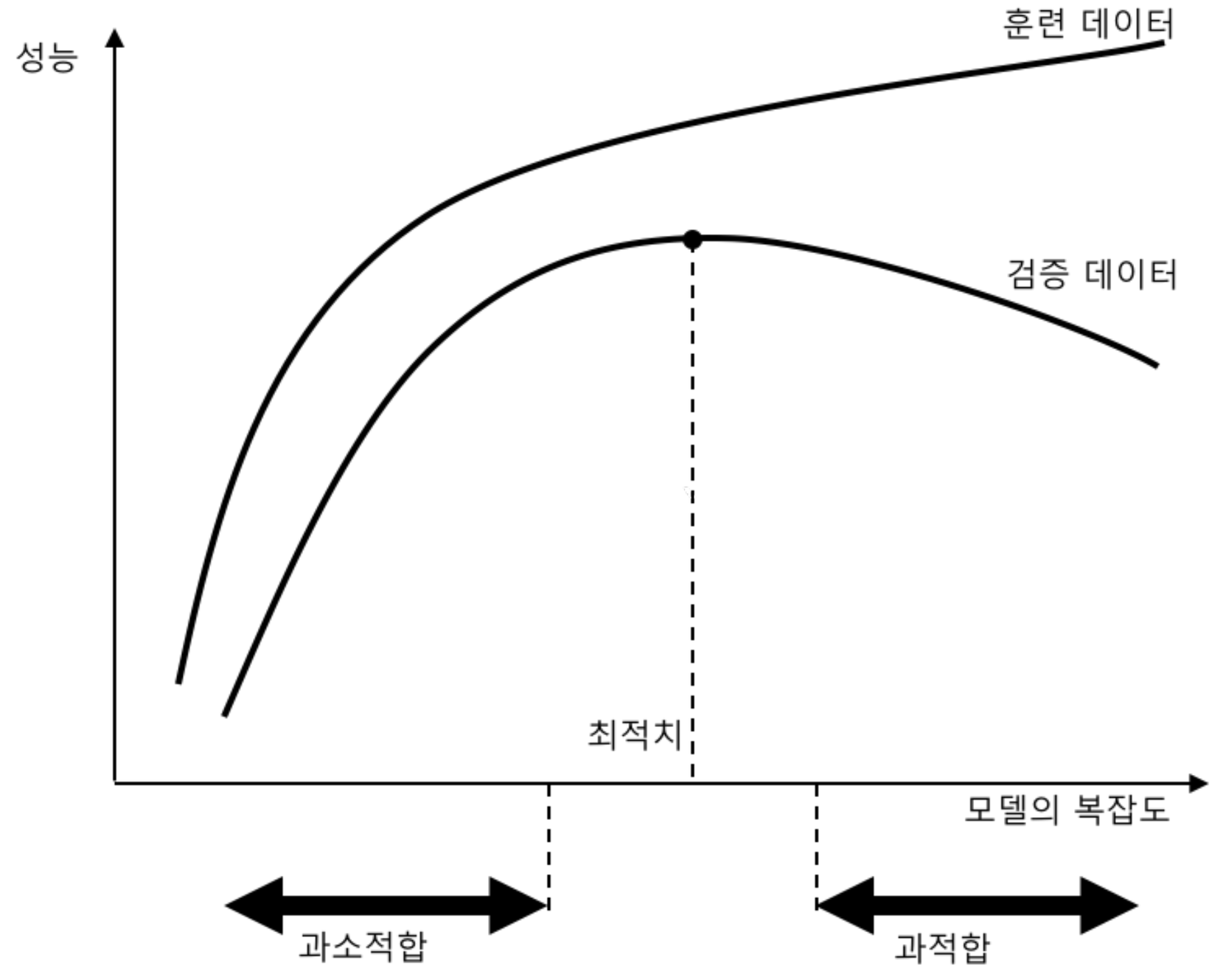
과대적합 검증

- 훈련 데이터에 대한 성능과 검증 데이터에 대한 손실함수를 관찰한다



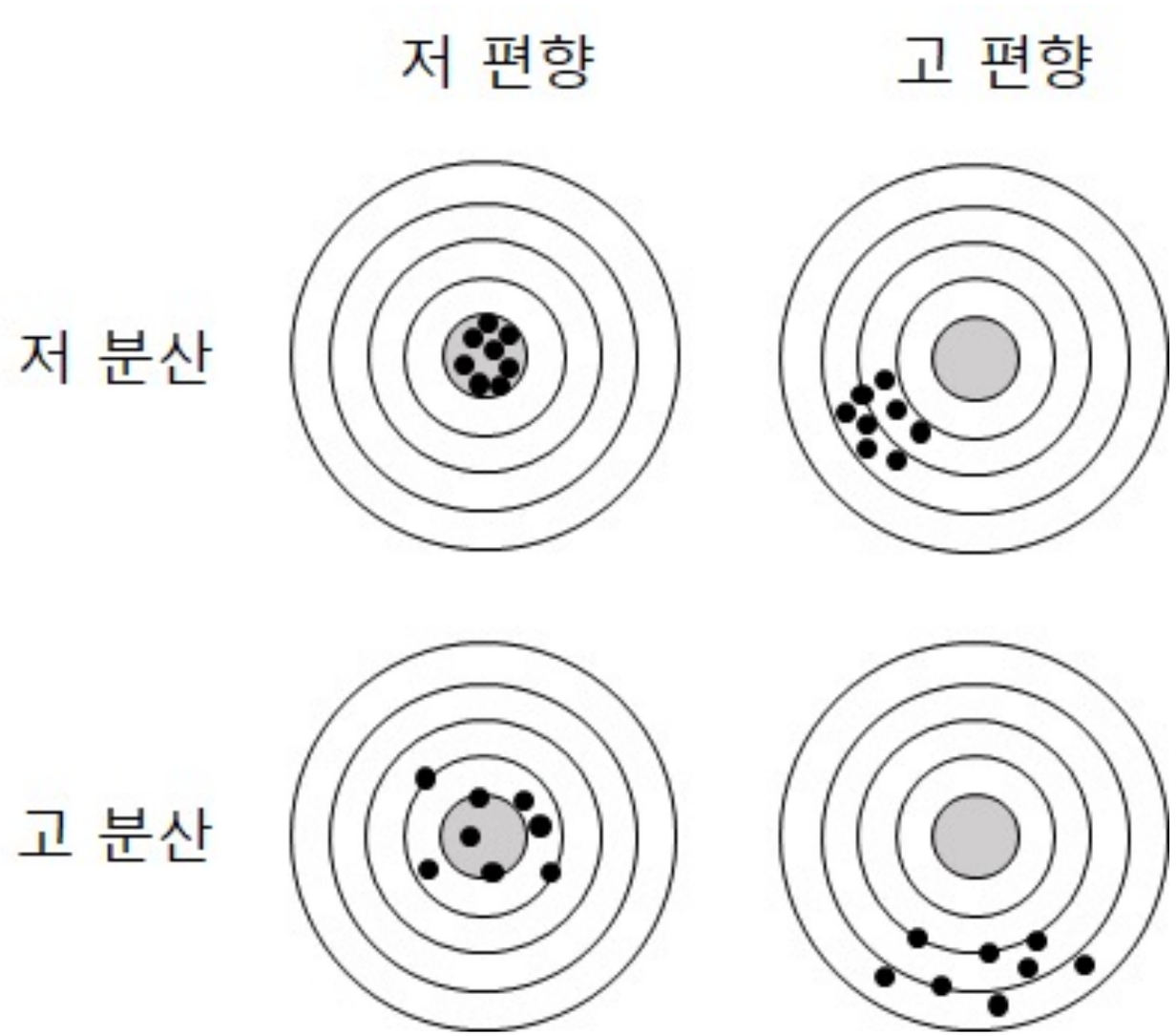
과대적합 검증

- 성능 비교



편향과 분산

- 예측 모델 오차는 분산(variance)과 편향(bias) 두 성분이 있다.
  - 분산이란 모델이 너무 복잡하거나 학습데이터 민감하게 반응하여 예측 값이 산발적으로 나타나는 것이다.
  - 편향이란 모델 자체가 부정확하여 피할 수 없이 발생하는 오차이다



머신러닝 모델 오류

- 바이어스
  - 잘못된 가정에 의한 오류. 바이어스가 크면 머신러닝 모델은 X-y 사이의 중요한 관계를 찾지 못한다(과소적합 상태)
  - 모델 자체가 부정확하여 피할 수 없이 발생하는 오차를 말한다.
- 분산
  - 모델은 과대적합되어 입력의 작은 변화가 큰 오류로 나타날 수 있다. 모델이 훈련 데이터에 대해 일반화되지 못하고 잡음을 신호로 잘못 취급한다.
  - 분산이란 모델이 너무 복잡하거나 학습데이터 민감하게 반응하여 예측 값이 산발적으로 나타나는 것이다.
- 잡음
  - 관측에서 발생하는 오류이다. 랜덤 노이즈나 측정 오류이다. 이는 모델이 찾아내지 못하는 오류이다.

순위 평가

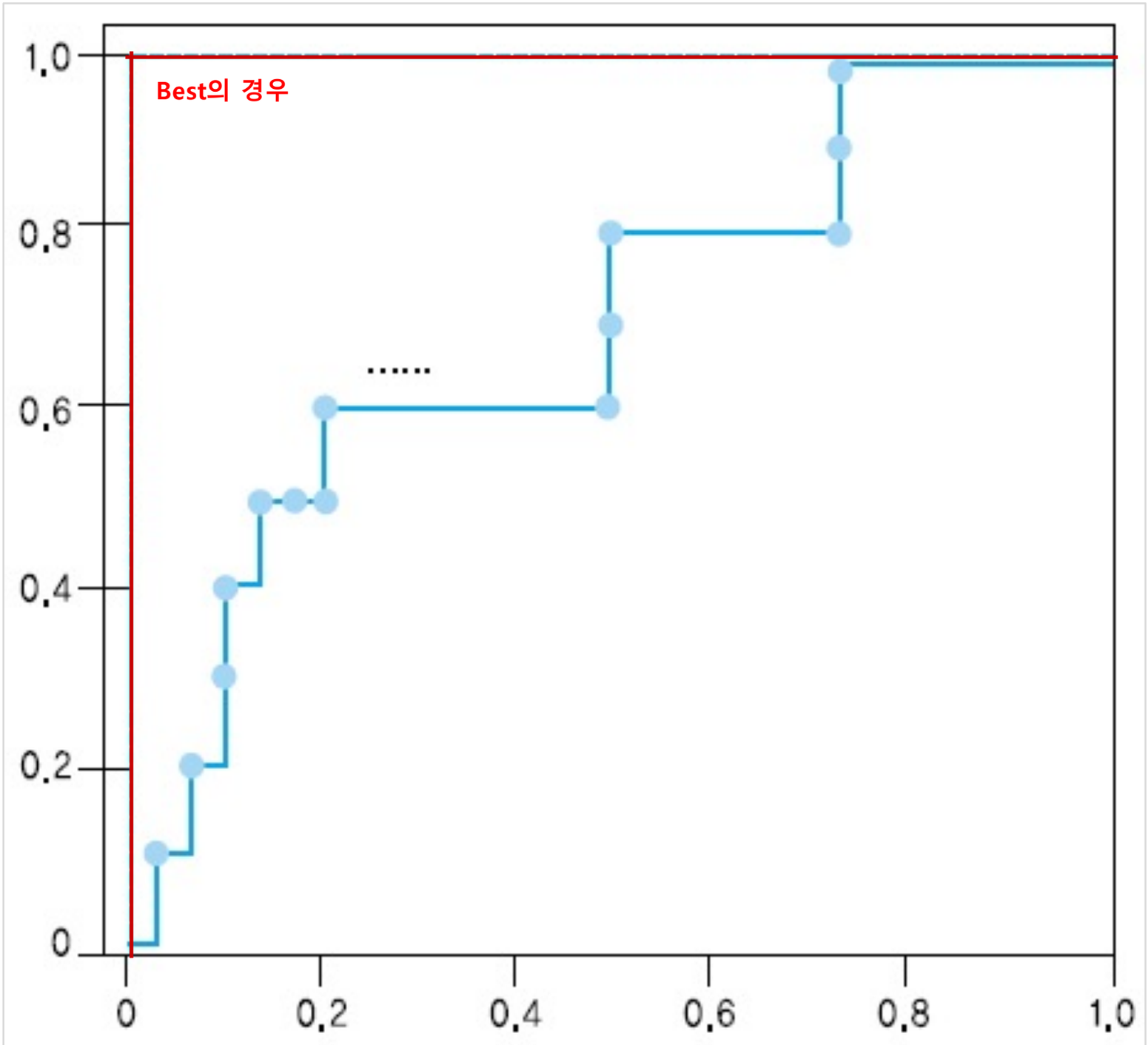
순위 평가

- 분류의 최종 성능 평가 뿐 아니라, 예측한 순서 자체를 평가하는 것이 필요하다.
- ROC 곡선
  - ROC(receiver operating characteristics) 곡선은 분류 예측 순서까지 고려하여 평가할 수 있는 수단이다.
  - ROC 커브는 예측한 점수순으로 순으로 한 행씩 내려오면서 정답과 오류를 차례대로 확인하는 방식으로 그린다.
  - x축은 false positive rate를 y축은 true positive rate를 나타낸다
- AUC(Area Under the Curve)
  - ROC 성능을 간단히 수치로 나타내기 위해서 ROC 그래프의 면적을 AUC(Area Under Curve)로 계산한다.

분류 순서 평가

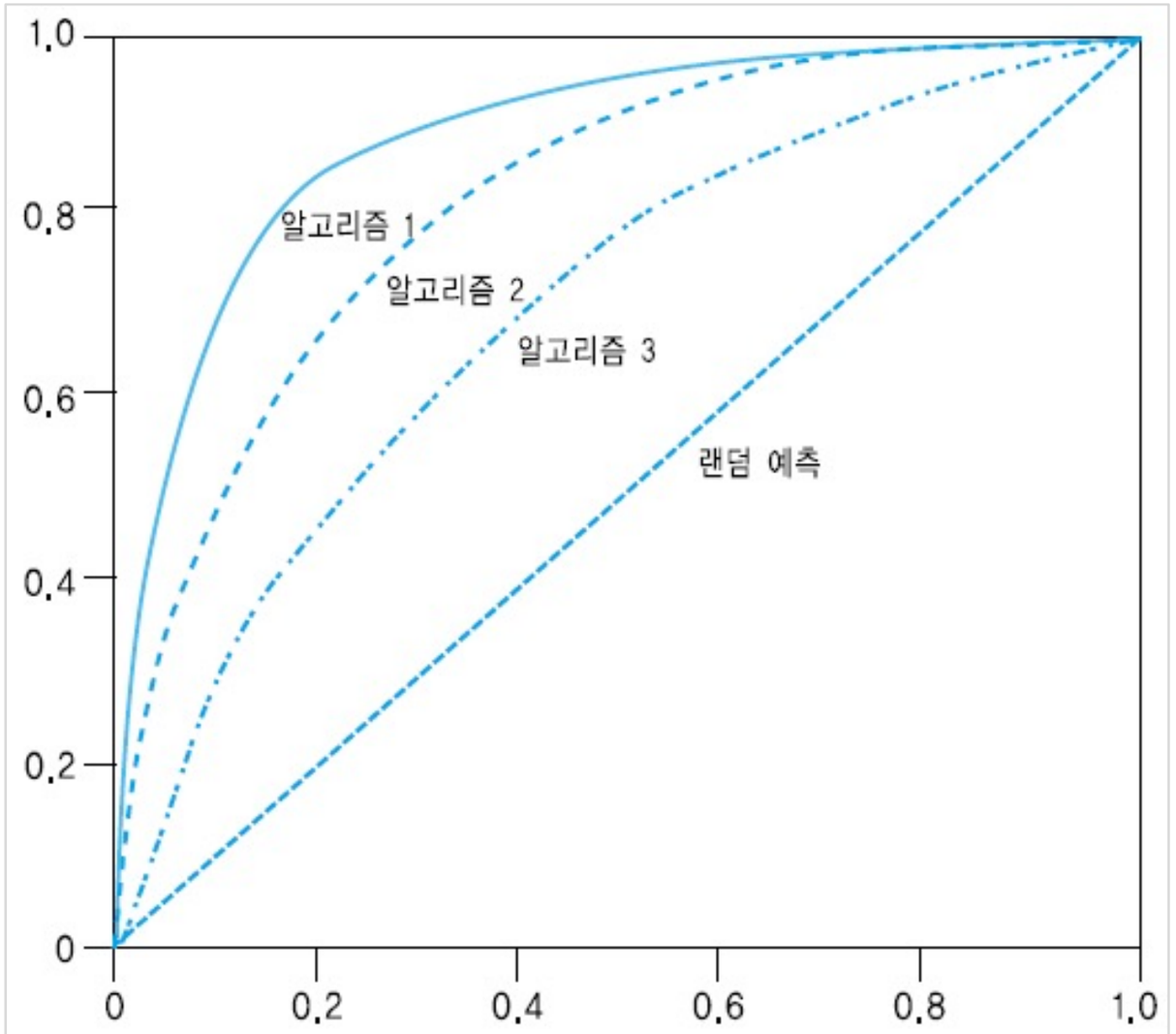
환자번호	성별	점수	순위	실제 값
7	F	0.98	1	N
125	M	0.96	2	C
4	F	0.95	3	N
199	M	0.86	4	C
2	F	0.84	5	N
200	M	0.82	6	C
176	M	0.81	7	C
73	M	0.80	8	N
82	M	0.79	9	C
3	F	0.77	10	N
123	F	0.76	11	N
		...		C
43	F	0.48	198	N
93	M	0.42	199	N
120	F	0.40	200	N

ROC 그래프 예시

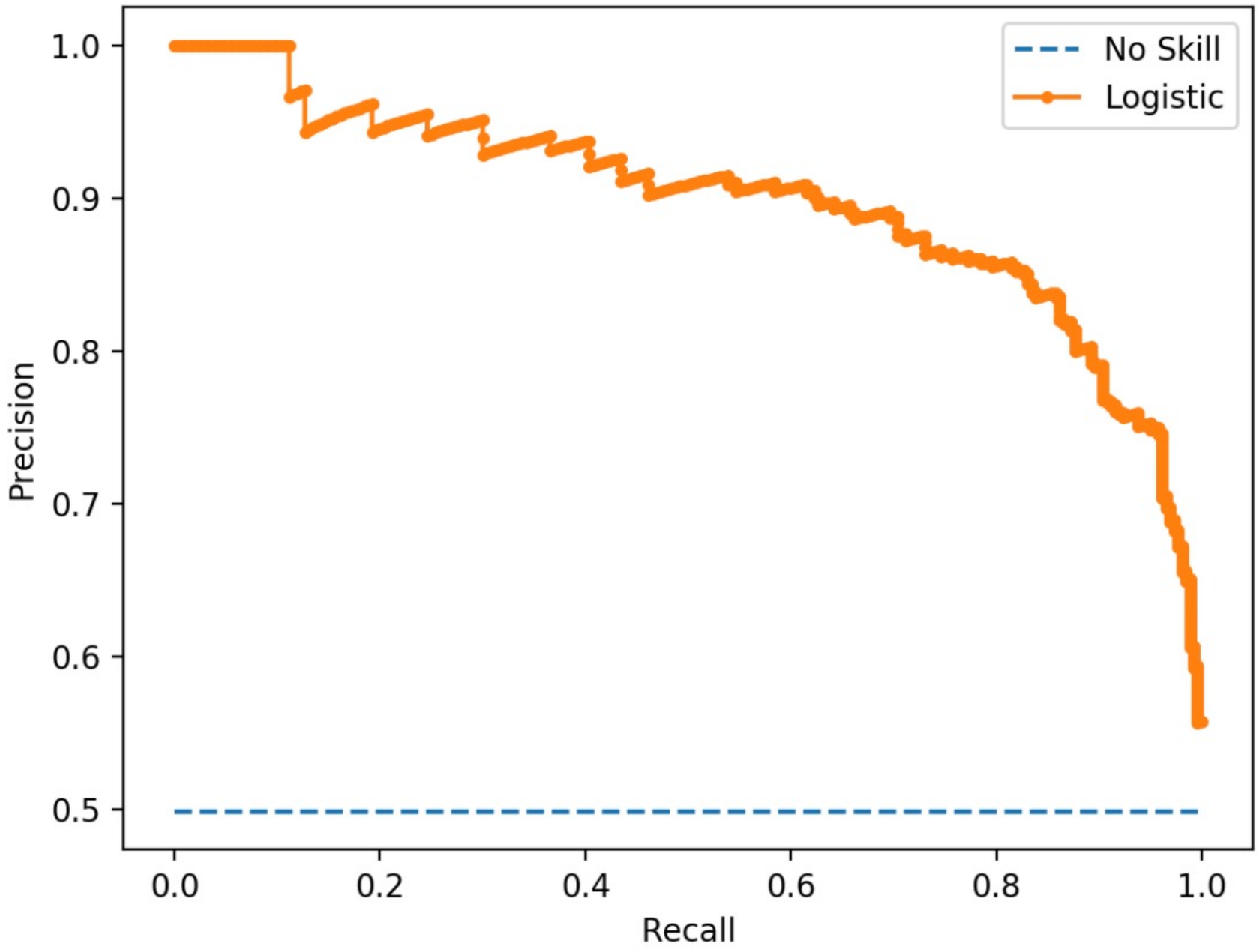


AUC(Area Under Curve)

- 예측 알고리즘의 성능을 간단히 수치로 나타내기 위해서 ROC 그래프의 면적을 계산하는 방법을 사용
- 우수한 알고리즘일수록 초반에 y축 상단 방향으로 이동하므로 ROC 커브의 면적이 넓어짐



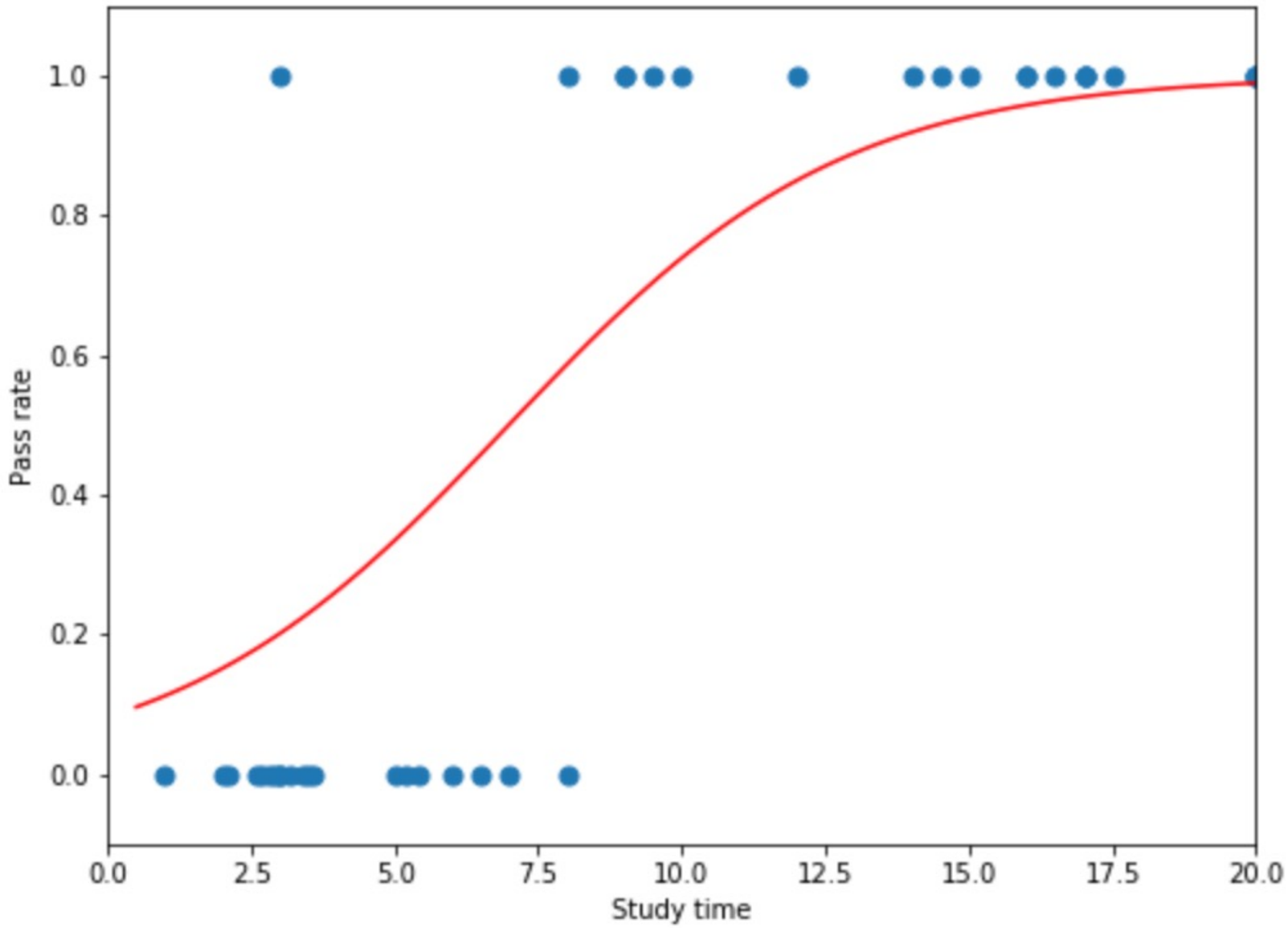
Recall-Presesion 곡선



로지스틱 회귀

로지스틱 회귀

- 로지스틱 회귀분석(logistic regression)은 임의의 범위를 갖는 값으로부터 0과 1사이의 값을 예측하거나 이진 분류에 사용하는 알고리즘이다.



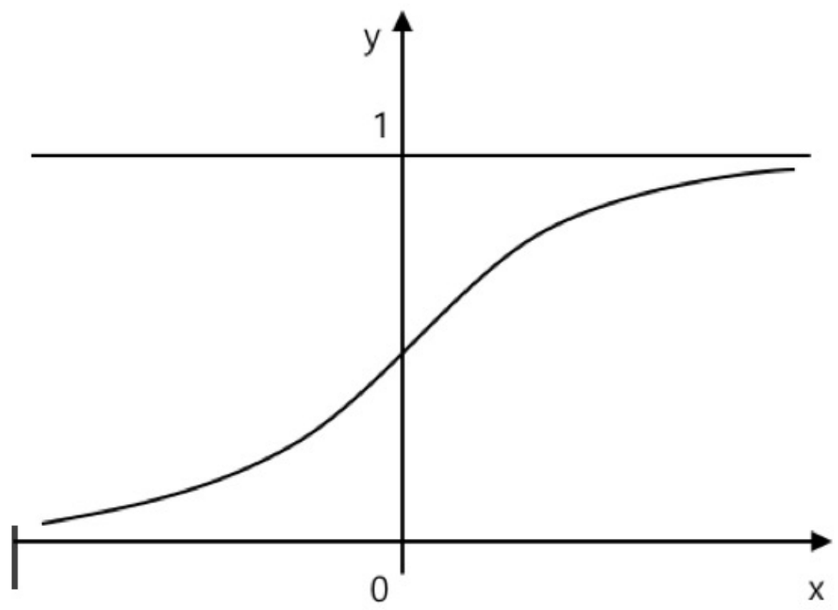
sigmoid

- 입출력의 관계를 직선이 아니라 S형 커브를 사용한다.
- S 커브로는 시그모이드 함수를 사용한다.

$$p = \frac{1}{1+e^{-y}}$$

- 임의의 입력 x에 대해 출력은 0 ~ 1 사이의 값으로 매핑되며 확률 모델에 사용된다

$$p = \frac{1}{1+e^{-(ax+b)}}$$

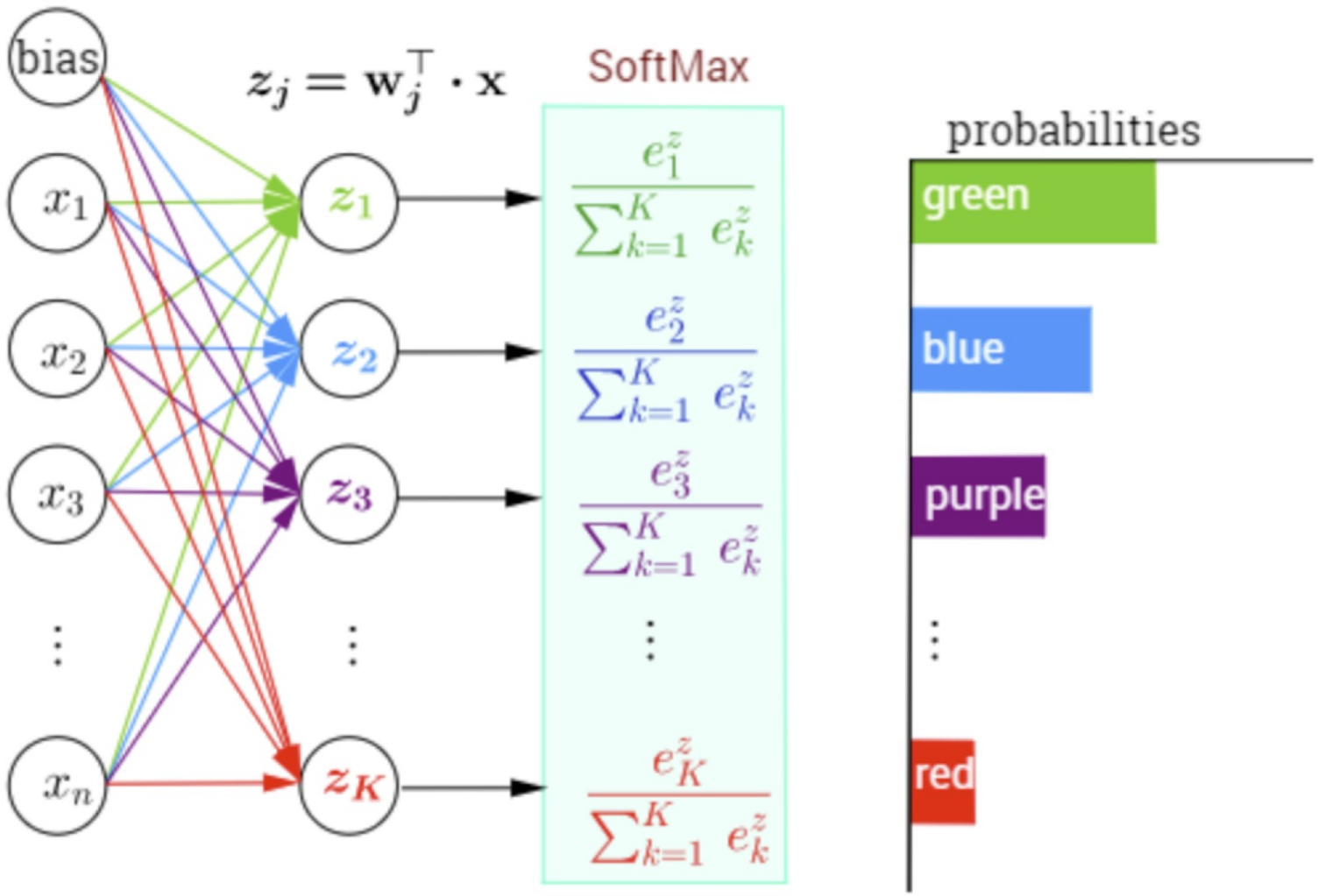


다항 로지스틱 회귀

- 이진 분류가 아니라 3개 이상의 클래스 중에 하나를 예측해야 하는 경우는 로지스틱 회귀를 사용할 수 없다.
- 다항 로지스틱 회귀(multinomial logistic regression) 또는 소프트맥스 (softmax) 함수를 이용한다.
- 다항 로지스틱스에서는 우선 각 클래스로 분류될 가능성을 나타내는 점수를 구하고 이 점수들을 사용하여 상대적인 확률을 구하는 소프트맥스 함수를 적용한다.

소프트맥스

$$\hat{p}_k = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=i}^{K} \exp(s_j(x))}$$



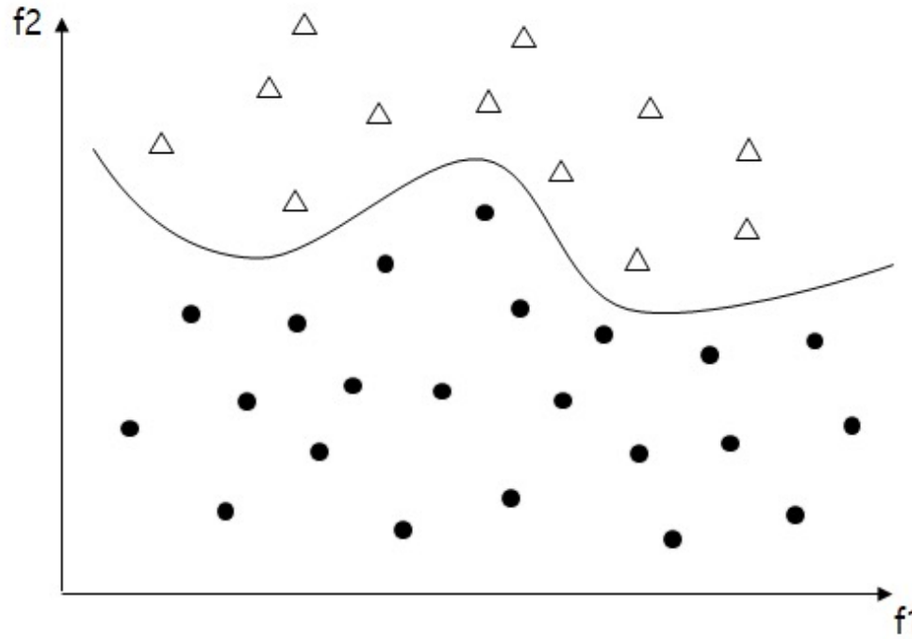
SVM

SVM

- 서포트 벡터 머신(Support Vector Machine, SVM)은 선형 모델을 개선한 것
- SVM의 기본적인 아이디어는 분류시에 경계면을 가능한 일반화 하는 것이다. 가능한 샘플 집단 간의 거리를 멀리 나눌 수 있는 경계면을 찾는 작업을 한다.
- 결정경계 주변에 있는 샘플들과 결정경계와의 거리 (margin)를 최대화 하는 방향으로 설정한다

커널 방식

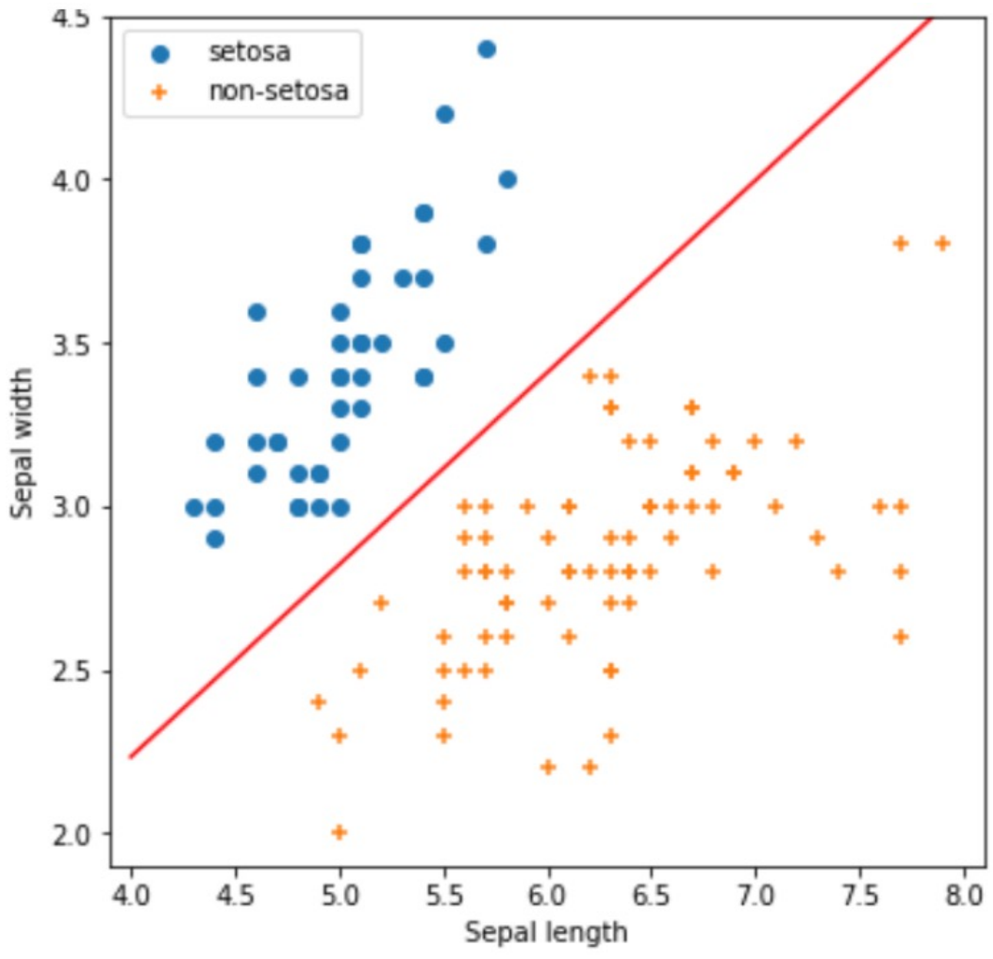
- SVM은 이 외에도 커널 방식을 제공하는데 주어진 특성을 그대로 사용하지 않고 이의 2승, 3승, 4승 등 고차원의 속성을 추가로 사용하는 방식이다.

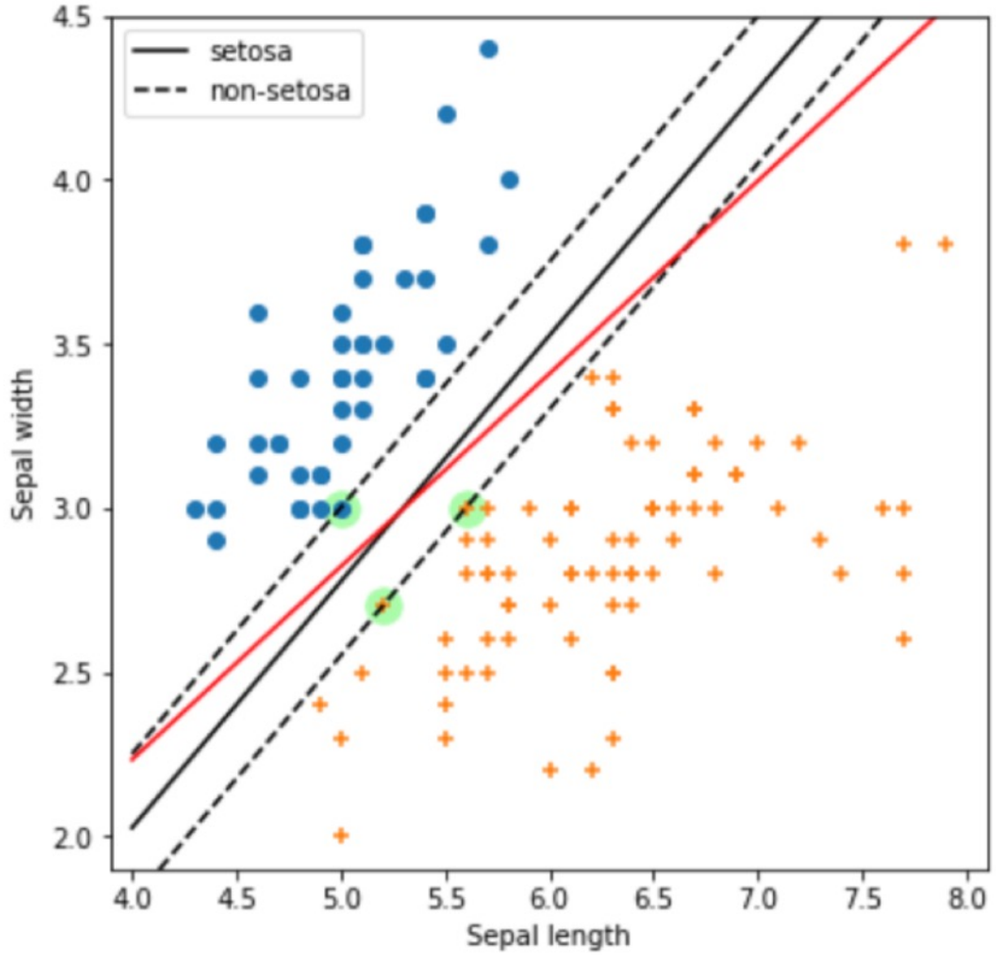


- SVM에 커널 트릭 기법을 도입함으로써 비선형 함수를 도입하고 이것이 선형 모델을 개선하는 주요 요소이다.
- 스케일링을 해야 한다.

SVM (Support Vector Machine)

- 선형 모델의 성능을 개선하였다.
- 분류시에 경계면을 가능한 일반화 하는 것이다.
- 샘플들을 단순히 나누기만 하는 것이 아니라 가능한 거리를 멀리 나눌 수 있는 경계면을 찾는 작업을 한다.



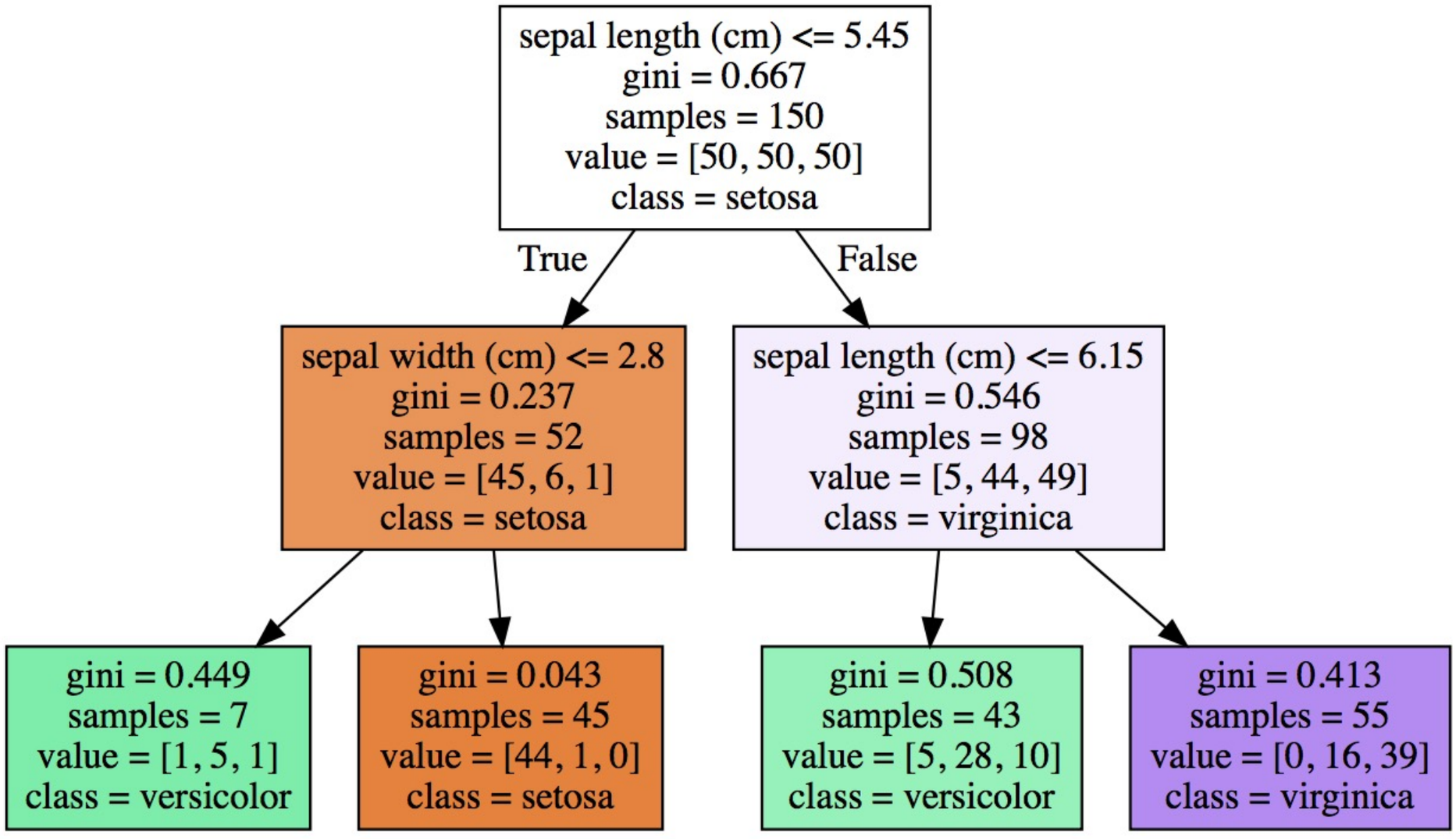


kNN

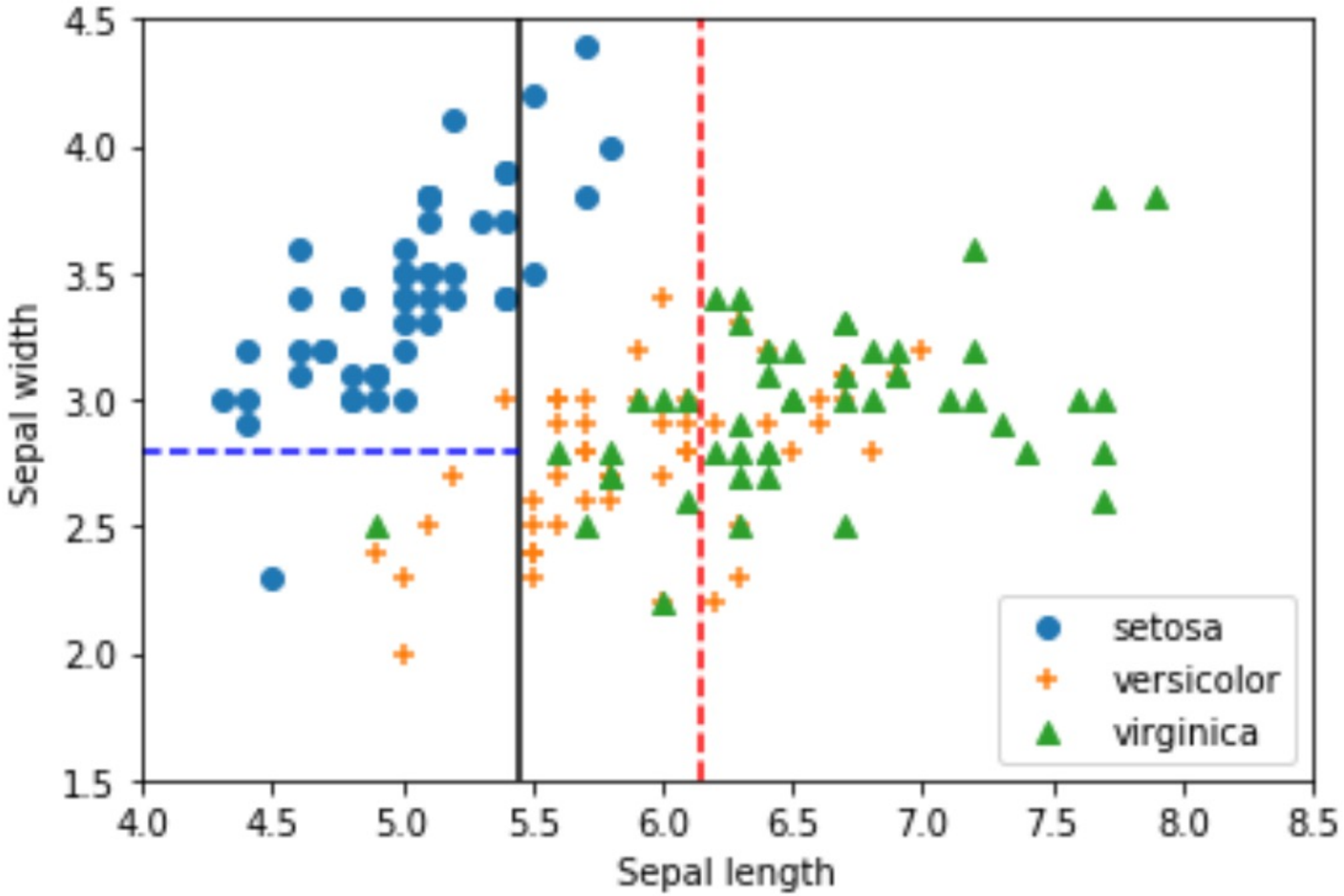
결정 트리

결정트리 동작

- 그룹을 효과적으로 "잘 나누는 것"의 기준은 그룹을 나눈 후에 생성되는 하위 그룹들에 가능하면 같은 종류의 아이템들이 모이는지를 기준으로 삼는다.
- 순도(purity)가 높게 나눈다



결정트리


4.5
4.0
3.5
3.0
2.5
2.0
1.5
Sepal width
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
Sepal length
setosa
versicolor
virginica

결정 트리(decision tree)

- 분류 작업을 수행하기 위해 한번에 한 특성 변수씩 해석
- 결정 트리 모델을 사용하면 동작을 설명하기 수월함
  - 대출 거부 사유
  - 신용도가 낮은 이유
  - 불합격 사유 등

결정트리 판별

- 판별 기준
  - 나누어지는 그룹의 순도를 측정하는 방법
- 엔트로피(entropy)
  - 어떤 사건의 "정보량의 기대치"은 그 사건의 정보량 log(1/p)에 그 사건이 발생할 확률(p)을 곱해주어야 한다.

$$p\log(\frac{1}{p}) = -p\log(p)$$

- 어떤 사건이 갖는 정보량 기대치를 엔트로피(entropy)라고 부른다.
- 한 노드(그룹)에 여러 클래스가 골고루 균일하게 섞여 있을 때는 엔트로피가 가장 높고, 동종의 클래스로 모여 있을수록 엔트로피가 낮다.

지니계수

- 그룹의 순도를 표현하는 데 지니(Gini) 계수도 자주 사용된다.

$$Gini = 1 - \sum_{k=1}^{m} p_k^2$$

- 위에서 Pk 는 카테고리 k에 대한 분포 확률을 말한다.

$$\text{지니}(7:3) = 1 - \left[\left(\frac{7}{10}\right)^2 + \left(\frac{3}{10}\right)^2\right] = 1 - (0.49 + 0.09) = 0.42$$

$$\text{지니}(5:5) = 1 - \left[\left(\frac{5}{10}\right)^2 + \left(\frac{5}{10}\right)^2\right] = 1 - (0.25 + 0.25) = 0.5$$

- 지니와 엔트로피 성능에는 큰 차이가 없다.
- 단, 지니의 계산 속도가 조금 빠르고 엔트로피를 사용하면 트리가 조금 더 잘 나누어진다.

특성 중요도

- feature_importances_
  - 결정 트리 모델을 만든 후에, 어떤 특성이 결정 트리를 생성할 때 중요한 역할을 했는지 비중을 파악할 수 있다.
  - 내부 변수인 feature_importances_에서 확인할 수 있다.
  - 이 결과를 보고 중요하지 않은 특성은 향후에 제외하기도 한다.
- 클래스 확률
  - 분류 모델을 사용하여 예측을 수행하려면 predict()를 사용한다
  - 새로운 샘플이 어느 클래스에 속하는지를 예측하는 것 외에, 이 샘플이 각 클래스에 속할 확률이 각각 얼마인지를 계산하는 것도 가능하다.
  - 예를 들어 다중 분류에서 소프트 투표(soft voting)를 도입하면 보다 정확한 다중 분류를 수행할 수 있다.
  - predict_proba() 함수를 사용한다.

결정 트리의 특징

- 결정 트리의 가장 큰 장점은 알고리즘의 동작을 쉽게 남에게 설명할 수 있다는 것이다.
- 그러나 훈련데이터가 바뀌면 모델의 구조가 달라지는 단점이 있다.
- 결정 트리의 또 다른 장점은 특성 변수의 스케일링이 필요 없다는 것이다.
- 결정 트리 알고리즘을 회귀분석에 사용하려면 DecisionTreeRegressor()를 사용

랜덤 포레스트

랜덤 포레스트

- 결정 트리의 성능을 개선한 방법, Random Forest
- 이 방식은 비교적 간단한 구조의 결정 트리(weak learner)를 수십~수백개를 만들고 각 결정 트리의 동작 결과의 평균치를 구하는 방법이다.
- 이를 앙상블(ensemble) 방법이라고 하며 하나의 모델만 만드는 것보다 좋은 성능을 보인다.
- 동작 원리
  - 랜덤 포레스트를 구성하는 각 결정 트리를 만들 때 훈련 데이터의 일부만 사용하거나 특성의 일부를 무작위로 선택하여 만든다.

간접투표(soft voting)

	P일 확률	Q일 확률	판정결과 (직접투표)
세부 모델 A	**0.9**	0.1	P
세부 모델 B	0.4	**0.6**	Q
세부 모델 C	0.3	**0.7**	Q
확률의 평균 (간접 투표)	(1.6)/3 = **0.533**	(1.4)/3 = 0.456	**P or Q**

앙상블 기법

1. 투표 방식
   - 일반적으로 서로 다른 알고리즘을 사용 -VotingClassifier()
   - 하드 보팅, 소프트 보팅
2. 배깅 방식
   - 샘플링을 다르게 선택하는 방식 (같은 알고리즘 사용)
   - 랜덤 포레스트: 전체 데이터 수과 같게 샘플링하되 중복을 허용
3. 부스팅
   - 그라디언트 부스팅 (예: Ada boosting)
   - Xg (Extra gradient) 부스팅
   - Light GBM
4. 스태킹
   - 앙상블의 앙상블 - 각 모델의 결과로 다시 학습을 수행

- 앙상블에서는 기본적으로 트리 모델을 주로 사용한다
  - 약한 학습기를 만들기가 편리하다

앙상블 알고리즘

- 랜덤 포레스트
  - 데이터를 다양하게 재구성하기 위하여 배깅을 사용
  - 사용하는 특성도 랜덤하게 일부만 사용한다
- 부스팅
  - 순차적으로 학습을 하되 잘 맞추지 못한 부분을 중심으로 재학습
  - Gradient boost (XG Boost, Light GBM)
  - Adaboost (adaptive boosting)

부스팅 알고리즘

- 원리
  - 약한 학습기를 순차적으로 학습-예측하여 성능 개선
- Adaboost
  - 잘 못 예측한 데이터에 가중치를 부여하여 예측을 다시 수행하면서 오류를 개선
  - 최종적으로 모든 학습기를 결합
- GBM (Gradient Boosting Machine)
  - Adaboost와 유사하나 가중치 업데이트에 경사하강법을 사용
- XGBoost
  - GBM의 느린 수행과 과적합 규제 부재 문제를 해결
  - 병렬 학습이 가능하다

Gradiendt 부스팅

- 앞 단에서 발생한 "오차"를 예측하는 모델을 만든다
- 즉, 예측 "오차"를 대상으로 이를 줄이는 작업을 순차적으로 수행한다.
- XGoost는 GB에 규제항을 추가하여 과대적합을 방지한 것이다 (모델을 단순하게)
- LightGBM은 트리 종료 조건에서 level단위가 아니라 leaf 단위로 동작시켜 속도와 성능을 개선한다. 과대적합에 더 민감하여 다량의 데이터가 필요하다
- catboost는 분산도 줄이면서 바이어스도 줄이는 방법으로 카테고리컬 특성을 수치형으로 변환한다 (성능이 가장 좋다고 주장)

LightGBM

- Xgboost 는 학습시간이 길다
- 학습속도와 메모리 사용을 개선한 알고리즘
  - 대용량 데이터 처리 및 GPU 지원
- 성능도 우수하다
- 단점은 적은 데이터 (예, 1만건 이하)에서 과대적합 발생
- 기존의 Level wise 방식이 아니라 Leaf wise 방식으로 동작
  - 기존의 방식은 뎁스를 줄이기 위해서 균형잡힌 트리를 만들기 위해서 계산량이 많다
  - 리프 중심에서는 최대 손실값을 갖는 리프를 지속적으로 분할하여 깊은 트리, 비 대칭적인 트리가 만들어진다
- 원 핫 인코딩을 하지 않아도 된다
  - 카테고리 변수의 자동 변환

특성 공학

- 여러 특성 중에 분석에 가장 영향력이 있는 특성을 선택하거나 신규로 특성을 생성하는 것이 필요할 때가 있다.
- 머신러닝에서 사용할 특성을 잘 선택하는 것을 특성 공학이라고 한다.
  - 특성을 잘 선택하면 학습 속도도 빨라지고 성능도 개선된다.
- 차원축소
  - 머신러닝의 성능을 떨어뜨리지 않으면서 특성의 수를 줄이는 것
  - 특성의 관계를 보기 좋게 시각화하기 위해서도 차원축소가 필요하다.
- 특성 공학은 비지도 학습이다.

특성 선택

- selectPercentile()
  - 어떤 속성을 선택하는 것이 좋을지를 컴퓨터가 자동으로 찾아 주는 방법
  - 가장 널리 사용되는 방법은 목적 변수와 상관관계가 높은 변수들을 찾는 것이다.
  - 상관관계가 높은 순서대로 속성들을 나열하고 상위 몇 %까지의 특성을 찾아준다

차원 축소

- 주성분 분석
  - principal component analysis(PCA)
  - 여러 특성들을 조합하여 이들을 대표할 수 있는 적은 수의 특성을(이를 주성분이라고 한다)를 찾아내는 작업을 말한다.
  - 주성분은 기존의 속성들의 선형 조합으로 새로운 변수를 정의한다.
  - 주성분 분석을 하려면 최종적으로 필요한 주성분의 개수를 알려주어야 한다.
  - 주성분을 만들기 위해서 기존의 특성에 각각 어떤 가중치를 곱하였는지를 알려면 내부변수 pca.components_를 보면 된다.

t-SNE

- 데이터의 특성을 한 눈에 파악하는데 시각화가 매우 유용하다.
  - 사람은 2차원 또는 3차원 공간에서의 시각화만 인식할 수 있다.
- 시각화를 위해서 고차원의 특성을 가진 데이터를 저차원으로 축소하는 기술로 t-SNE가 널리 사용된다.