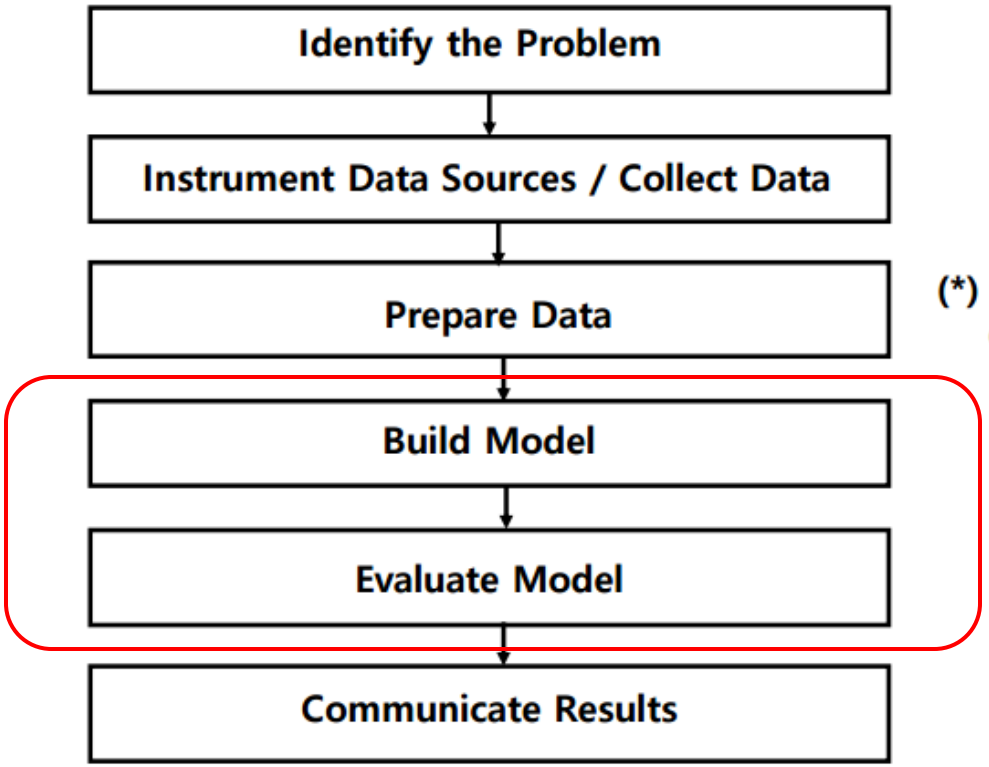
선형 회귀

데이터 분석 모델

Data Analysis Model (Jeff Hammerbacher)



(*) integrate, transform,
clean, aggregate, alter

데이터 분석 모델

- 데이터 분석 방법
  - 수학적 모델에 기반한 방법
  - 통계적 모델링
  - 기계학습적인 모데링
- 모델
  - 데이터를 발생시킨 원래 시스템을 설명하기 위해 설정한 구조
- 모델을 이용
  - 연구 대상의 본질적인 특성을 설명
  - 미래의 어떤 값을 예측
- 비록 모델이 정확하지 않더라도 실전에서 유용하게 쓰일 수 있고 데이터를 보고 점차 정교하게 개선해 나갈 수 있다.

데이터 분석 모델

- 모델이 필요한 또 다른 중요한 이유
  - 분석 과정에서 사람들 간의 오해를 줄이기 위해서는 객관적인 기준이 필요하기 때문
  - 데이터 분석 결과를 객관적으로 설명하기 위해서는 공통으로 사용할 수 있는 모델이 필요하다

Machine Learning(기계학습) 모델

- 머신 러닝은 모델(model)을 사용
  - 스팸 메일을 찾아내는 모델
  - 누가 게임에서 이길지 예측하는 모델
  - 내일 날씨를 예측하는 모델
- 과학에서는 어떤 현상을 설명하는 모델로 수식을 주로 사용
  - 모든 질량을 가진 모든 물체는 서로 끌어당긴다는 만유인력 법칙은 두 물체의 질량에 각각 비례, 두 물체의 거리의 자승에 반비례하는 수식으로 표현
- 머신 러닝, AI 모델은 데이터 기반의 모델을 사용

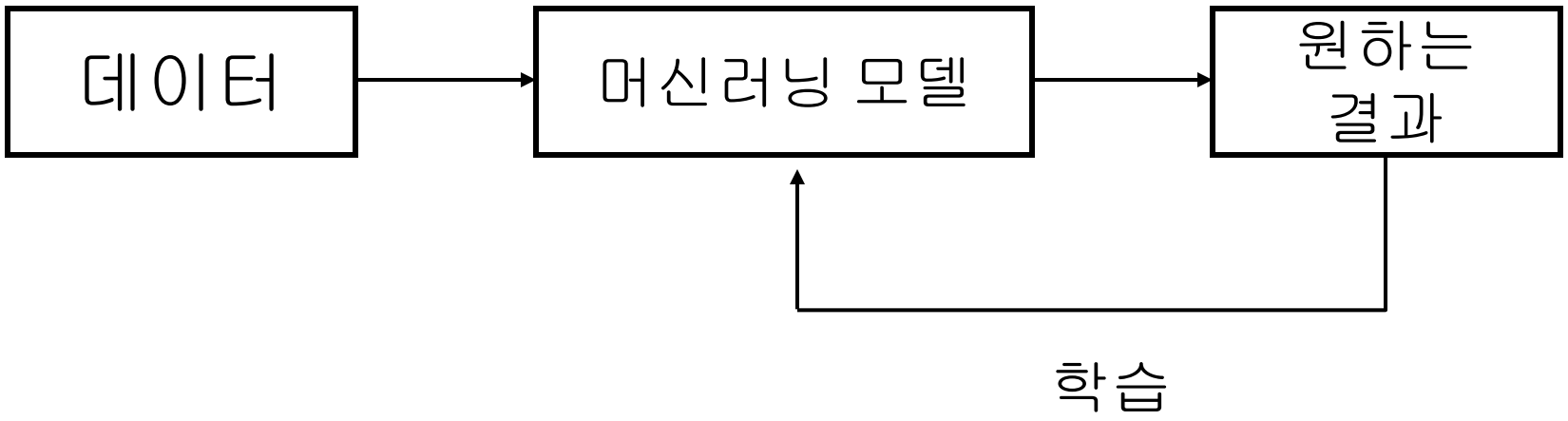
모델의 가치

- 와인 품질 = 12.145 + (0.00117x겨울철 강수량)
  + (0.064x 재배철 평균기온) – (0.00386x 수확기 강수량)

모델의 특징

- 머신 러닝에서는 데이터에 기반한 모델을 사용 (학습)
- 현실 세계의 많은 현상
  - 수식으로 간단히 모델링하기 어렵고, 과학적으로 증명할 수도 없다.
- 그러나 머신 러닝은 성능이 꽤 유용

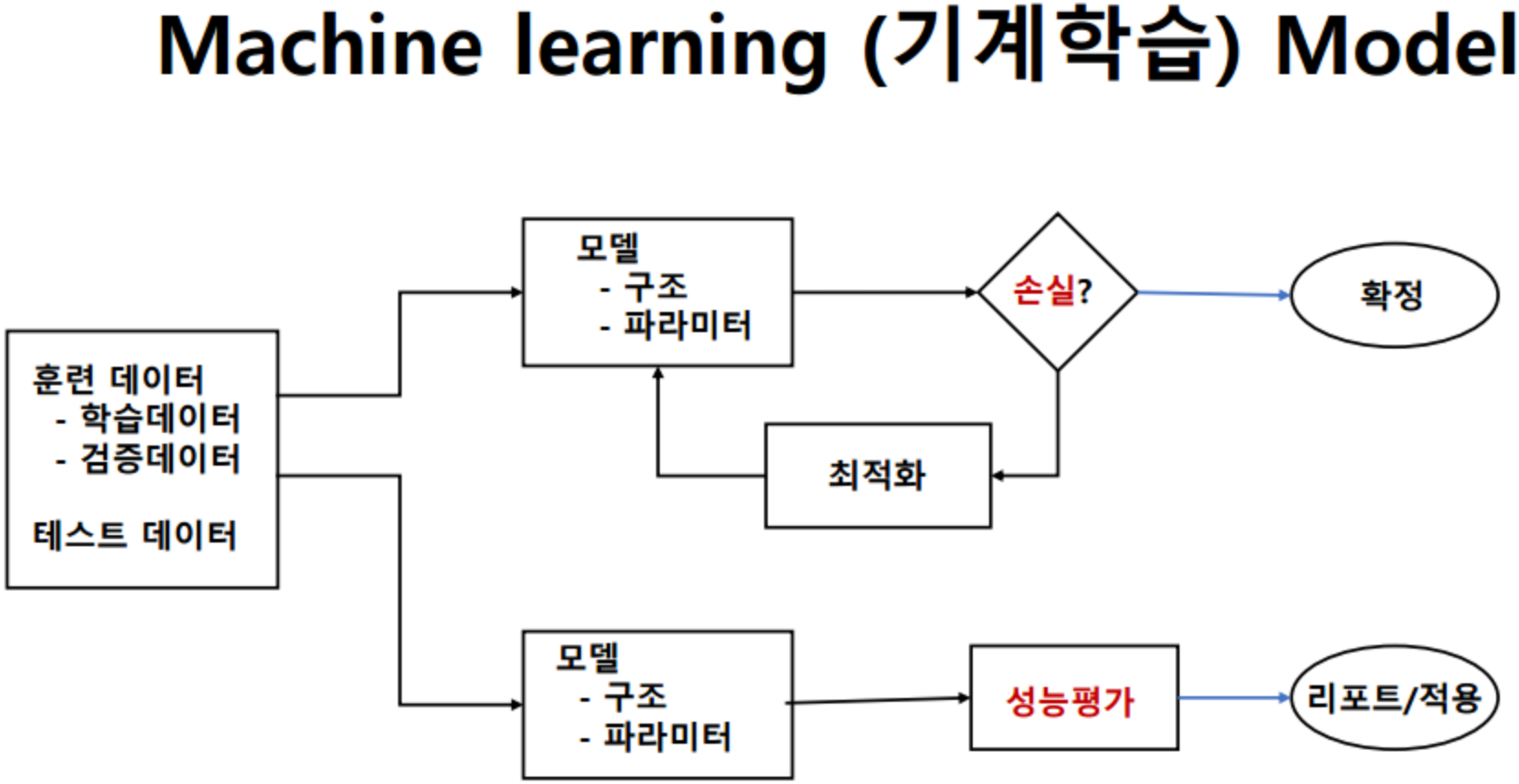
머신 러닝의 기본 동작



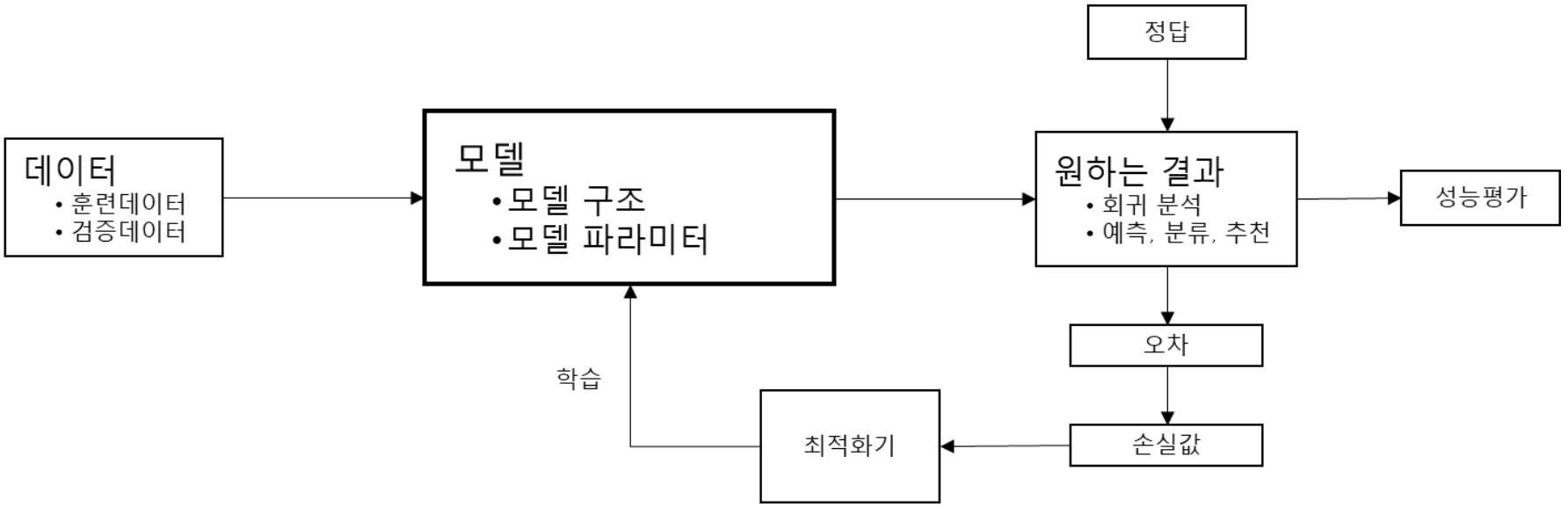
모델의 특징

- 모델 구조
  - 모델의 동작을 규정하는 방법
  - Programmer가 선택 : hyper parameter
- 모델 파라미터
  - 모델이 잘 동작하도록 학습하는 매개 변수(parameter)
    - 모델 계수 (신경망의 경우, 가중치)
- 모델 학습
  - 주어진 데이터에 가장 적절한 parameter를 찾는 작업
  - 머신 러닝 : 학습을 통하여 찾는다

Machine Learning(기계학습) 모델

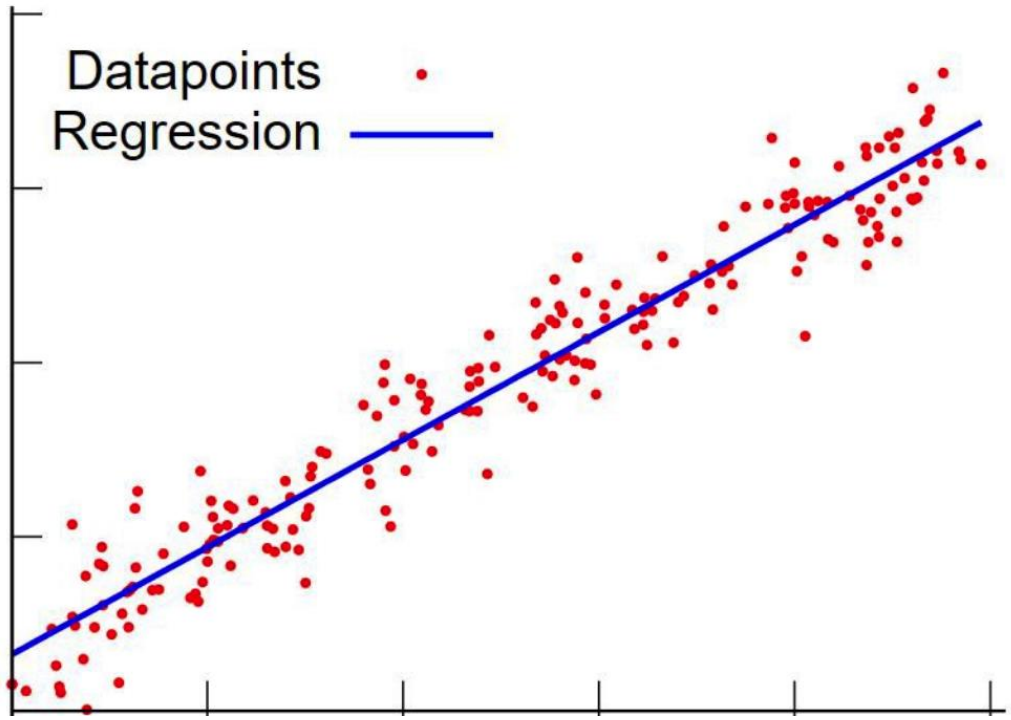


Machine Learning(기계학습) 모델



선형 회귀(Linear Regression)

- 선형 회귀(regression) y = wX + b



다중 선형 회귀

- 여러 개의 특성을 이용한 회귀 예측 모델
  - multiple linear regression
  - 예) 혈압을 예측 : 연령 뿐 아니라, 몸무게를 같이 고려
    - 종속 변수 : 혈압
    - 독립 변수 : 연령, 몸무게
  - 다중 선형 회귀식

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

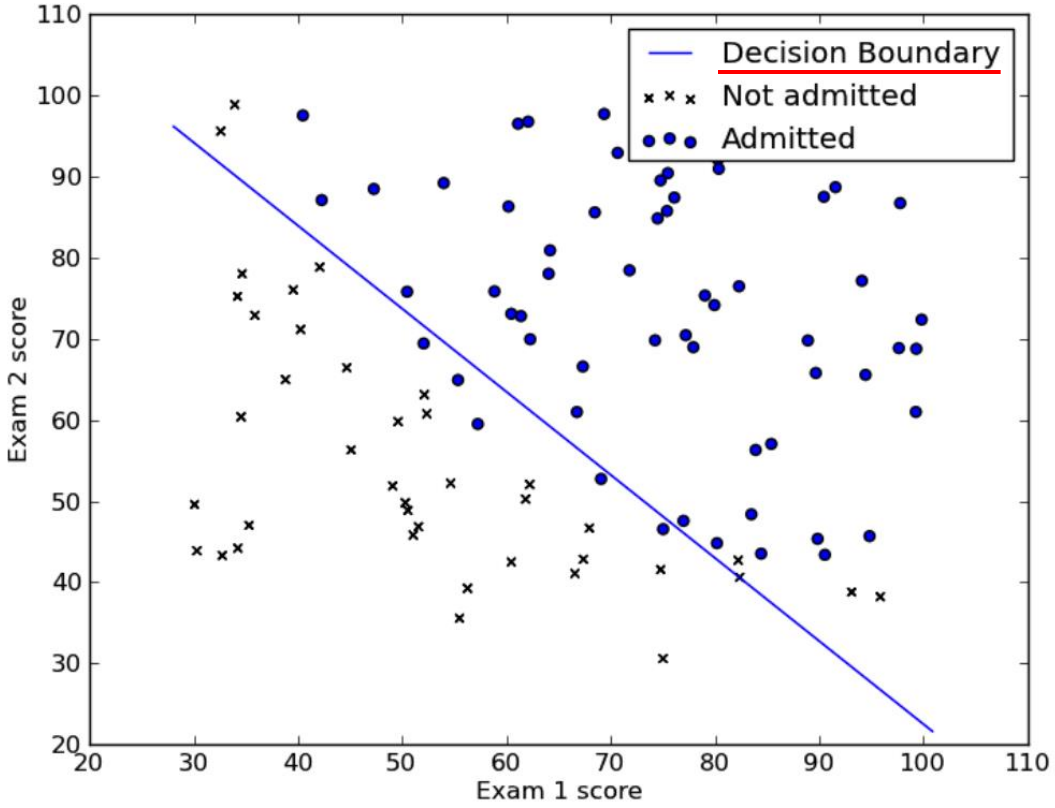
  - p : 회귀 분석에 사용되는 변수의 총 갯수

- 종속 변수 : 목적(target) 변수, outcome 변수, response 변수, label
- 독립 변수 : predictor, 설명 변수(explanatory variable), 특징(fearure)

선형 분류

선형 분류(Linear Classification)

- 선형 분류(classification) $ay + bx > c$



손실 함수(Loss Function)

- 손실함수(loss function)
  - 모델의 예측값과 실제 값과의 차이, 즉 오차(error)를 계산
- 이 오차를 줄이는 방향으로 모델을 최적화(학습) 한다
- 회귀분석에서 많이 사용하는 손실함수
  - 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 hyper parameter라고 한다.
  - **hyper parameter** : 사람이 선택하는 변수
  - **parameter** : 기계 학습으로 자동으로 갱신되는 변수

분류의 손실 함수(Loss Function)

- 분류에서는 손실함수로 MSE를 사용할 수 없다
- 대신, 분류에서 정확도(accuracy)를 손실함수로 사용할 수 있다
  - 예) 100명에 대해 남녀 분류 문제
    - 96명을 맞추고 4명을 오 분류 : 정확도 0.96
  - 그러나 정확도를 손실함수로 사용하는 데에는 다음과 같은 문제가 있다

- **Category 분포 불균형**시 문제
  - 예)
    - Group : 남자 95명, 여자 5명
    - 오 분류 케이스 - 남자 1명, 여자 3명
    - 정확도는 여전히 0.96 :
    - 문제 : 여자의 경우, 5명 중 3명을 오 분류 ➔ 결과 심각
  - 데이터 분포가 비대칭인 상황 : 질병 진단의 경우 자주 발생
  - 손실을 제대로 측정하지 못함
  - 이를 보완하기 위해서 크로스 엔트로피(cross entropy)를 사용
    - Category가 둘 이상인 경우에도 동일한 개념으로 적용 가능

크로스 엔트로피(Cross Entropy)

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

- p_i : 어떤 사건이 일어날 실제 확률, p_i' : 예측한 확률
- 남녀가 50명씩 같은 경우

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

- 남자가 95명 여자가 5명인 경우

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

훈련(Training)

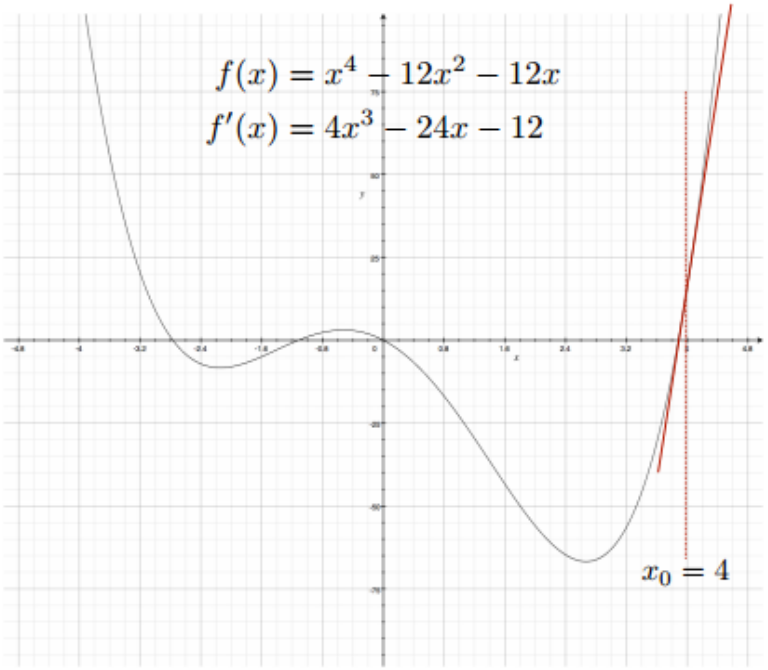
- 훈련 (training)
  - 모델이 데이터를 이용하여 학습하는 과정
  - 최적화 알고리즘에 의해서 parameter(가중치 등)를 계속 갱신하여 모델의 예측 값이 실제 값에 수렴하도록 학습

- 검증(validation)
  - 훈련된 모델이 잘 동작하는지 확인하는 과정

훈련 방법 : 최적화 – 경사하강법

- 경사 하강법(Gradient Descent)
  - 가장 일반적인 최적화 알고리즘
  - 손실함수를 계수에 관한 그래프로 그렸을 때 최소값으로 빨리 도달하기 위해서는 현재 위치에서의 기울기(미분값)에 비례하여 반대방향으로 이동

$f(x): error$



$x : weight$

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

모델의 성능

- 모델의 성능을 평가하는 척도 필요

- 분류에서는 성능 척도로 정확도(accuracy)를 주로 사용
  - (참고) 분류에서 손실함수로 크로스 엔트로피를 주로 사용

- 손실함수와 성능 지표의 차이점
  - **손실함수** :
    - 모델을 훈련시킬 때의 기준
    - 모델은 손실함수를 최소화 하는 방향으로 학습
  - **성능 지표**
    - 이렇게 만든 모델이 궁극적으로 얼마나 잘 동작하는지를 평가하는 척도

대표적인 손실함수와 성능지표

	손실함수	성능 지표
정 의	손실함수를 줄이는 방향으로 학습	성능을 높이는 것이 머신러닝을 사용하는 최종 목적
회귀 모델	MSE (오차 자승의 평균)	R^2
분류 모델	크로스 엔트로피	정확도, 정밀도, 재현률, F1점수

Regression (회귀) – 예측, 분류

❖ **What to reduce? (Loss Function: 손실함수)**

- **MSE** (Mean Square Error)

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

❖ **How Good it is? (Performance: 성능지표)**

- **R^2** (R-Squared)

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

Classification (분류)

❖ **What to reduce? (Loss Function: 손실함수)**

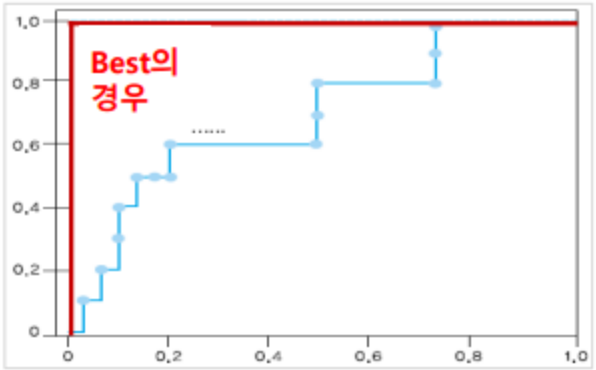
- **Cross Entropy (CE)**
- **Gini (지니계수)**

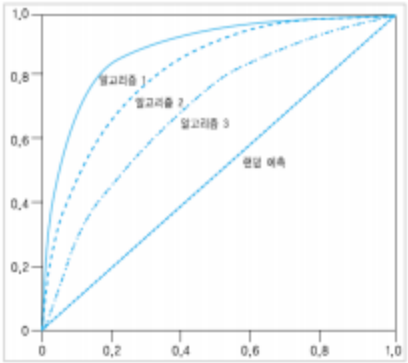
$$CE = \sum_i p_i \log\left(\frac{1}{p_i}\right)$$

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

❖ **How Good it is? (Performance: 성능지표)**

- **Confusion Matrix**: Accuracy, Recall, Precision, F-1 Score
- **Ranking(순서)**: ROC (Receiver Operating Characteristic), AUC (Area Under Curve)





모델의 성능 지표

- 정확도(accuracy): 정확하게 예측한 비율을 의미
  - accuracy = (TP+TN) / 전체 경우의 수(N)

실제 / 예측	암(예측)	정상(예측)	합계
암환자(실제)	6 (TP)	4 (FN)	10
정상(실제)	2 (FP)	188 (TN)	190
합계	8	192	200

  - 암진단 정확도 = (6 + 188)/200 = 194/200 = 0.97 => 97%
  - 오류율 = 1-accuracy = 0.03 => 오진율은 3%
- 리콜(recall): 관심 대상을 얼마나 잘 찾아내는가
  - recall = TP / (TP+FN)
  - 실제 암 환자 발견률 = 6 / (6+4) = 0.6 => 60%
- 정밀도(precision): 예측의 정확도
  - precision = TP / (TP+FP) = 6 / (6+2) = 0.75 => 75%

모델의 성능 지표

- recall과 precision의 두 가지 지표를 동시에 높이는 것은 어려움,
- F1은 이러한 두 요소를 동시에 반영한 새로운 지표임
- F1은 recall과 precision의 조화 평균을 구한 것

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

- 두 지표의 값이 각각 0.5와 0.7일 때
  - 산술 평균 c=(a+b)/2=(0.5)+(0.7)/2=0.6
  - 조화 평균 c=2ab/(a+b)=0.7/1.2=0.58

- 두 지표의 값이 각각 0.9와 0.3일 때
  - 산술 평균 c=(a+b)/2=(0.9)+(0.3)/2=0.6
  - 조화 평균 c=2ab/(a+b)=0.54/1.2=0.45

모델의 동작 속도

- 일반적으로 모델이 정교하고 복잡할수록 성능은 좋아지지만 모델을 만들거나 적용하는데 시간이 오래 걸린다.

  - 학습 시간 : 모델을 만드는데 걸리는 시간
  - 동작 속도 : 모델을 적용하는데 걸리는 시간

훈련과 검증

- 검증 데이터(validation data)
  - 모델 동작이 얼마나 우수한 지를 검증할때는 훈련 데이터로 해서는 안됨
  - 훈련에 사용하지 않은 새로운 데이터를 사용

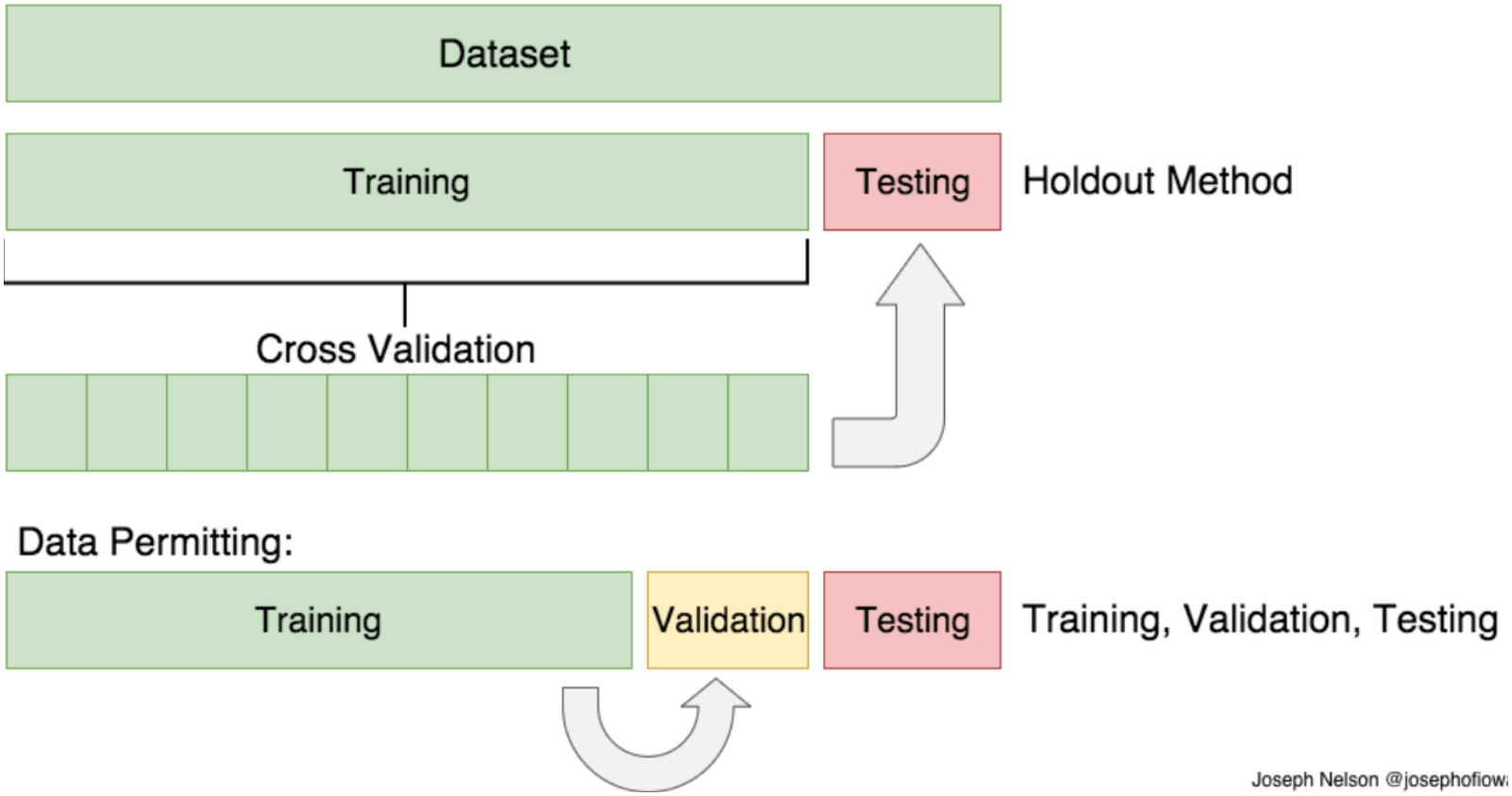
- 보통 검증 데이터를 따로 제공하지 않으므로 훈련에 사용할 데이터의 일부를 검증용으로 미리 확보해야 한다

- 훈련에 사용하지 않고 남겨 두었다가 모델이 제대로 동작하는지 테스트할 때 사용하는 데이터를 hold-out 데이터라고 한다.

훈련, 검증, 테스트 데이터

- 훈련(Training) 데이터
  - 모델 parameter를 학습시키는데 사용
- 검증(Validation) 데이터
  - 모델의 학습 중에 과소적합, 과대적합을 검사하고 최적 모델 구조(hyper parameter 등)를 찾는데 사용
  - 훈련 데이터 중의 일부를 학습에 참여시키지 않고 남겨 둔 데이터
- 테스트(Test) 데이터
  - 모델의 성능을 최종적으로 시험하는데 사용

훈련, 검증, 테스트 데이터

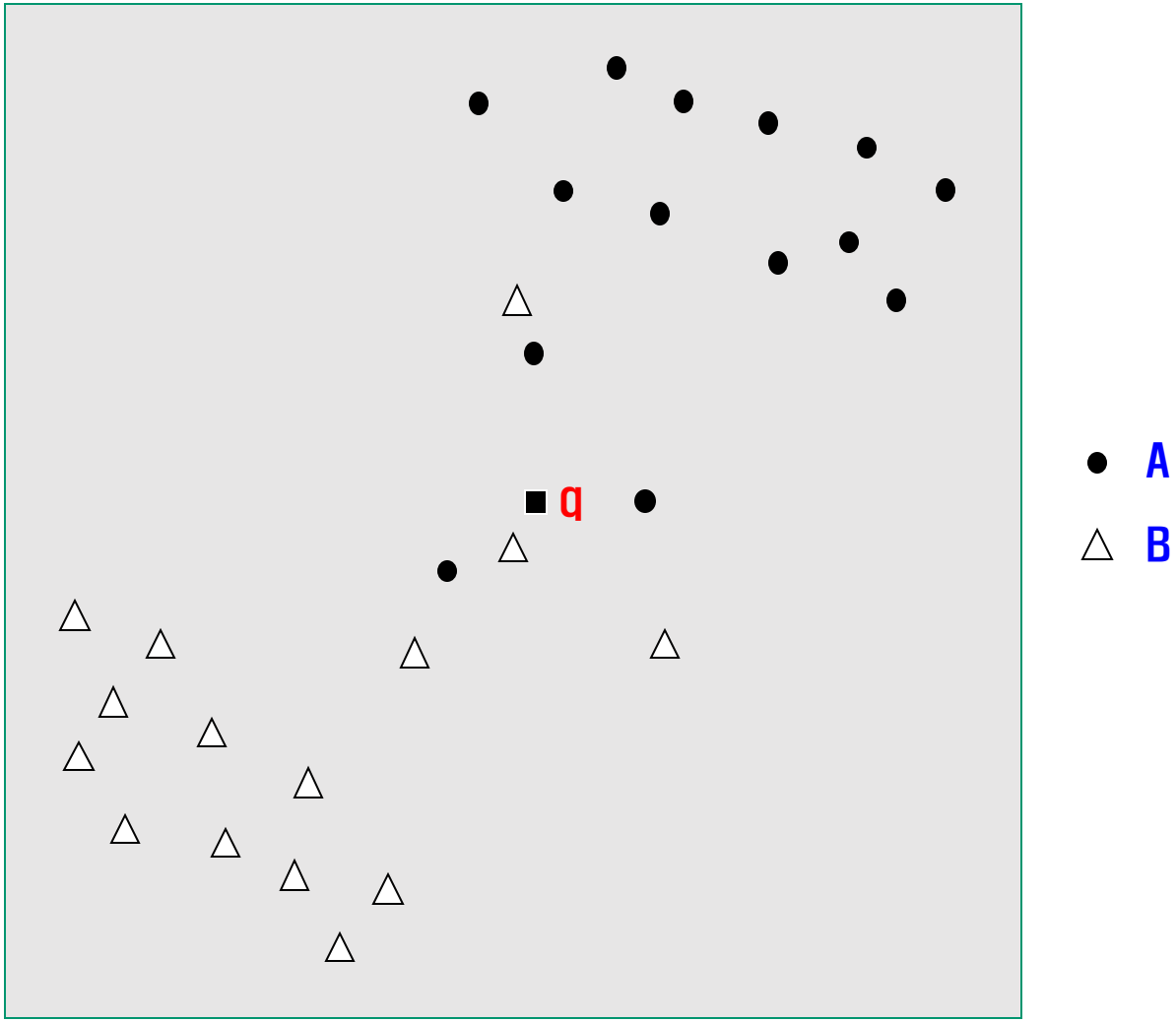

Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiow

k-fold 교차 검증

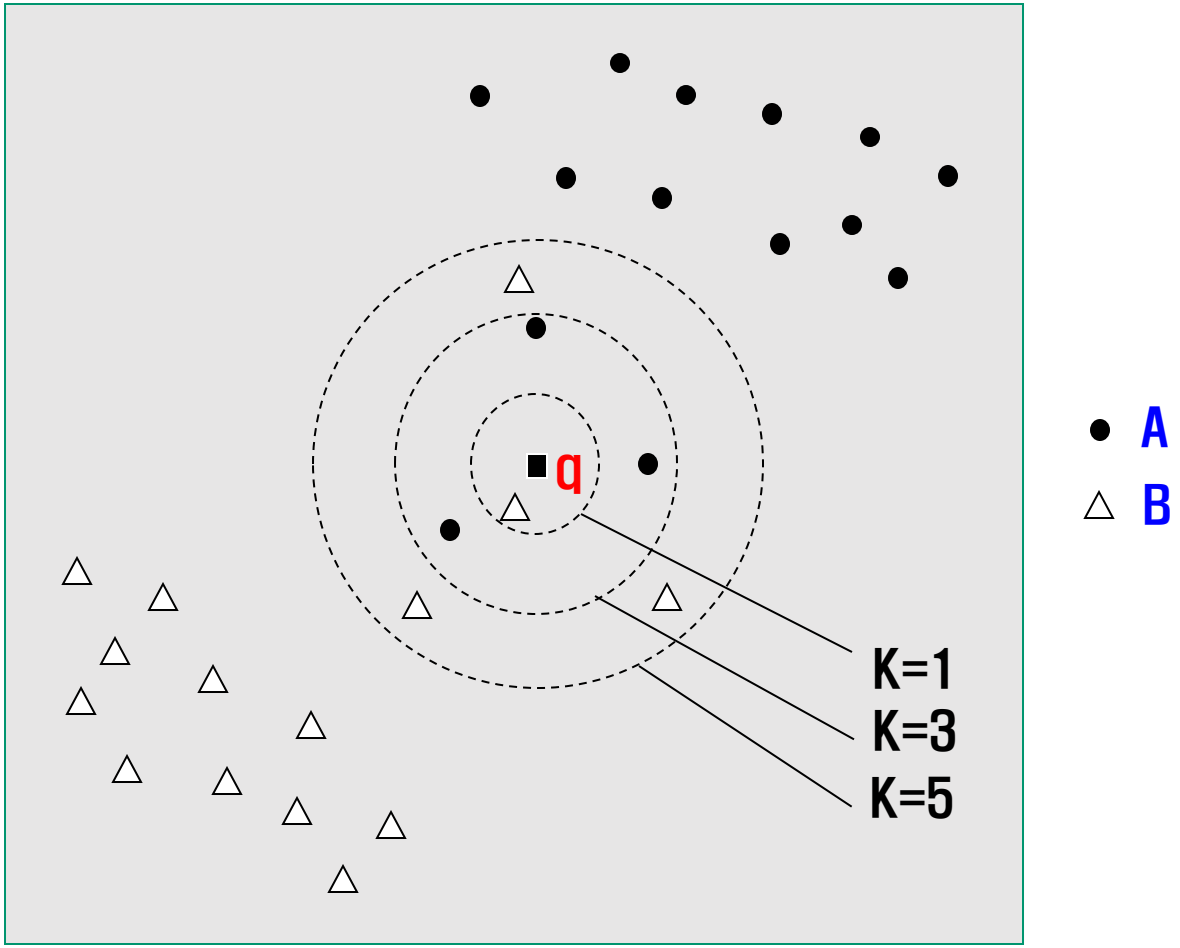
k-NN(k nearest neighbor)

- 주어진 샘플의 특성 값을 보고 가장 가까운 특성을 가지는 이웃(neighbor)을 k개 선택하고 이들 레이블의 평균치로 이 샘플이 속할 분류를 예측
- kNN은 직관적으로 이해하기 쉬운 분류 알고리즘으로서 추천 시스템에서 많이 사용
  - 적절한 추천을 하기 위해서 추천을 요청한 사람의 성향을 특성들로 파악하고 그 사람과 가장 성향이 유사한 k명의 사람들이 좋아하는 품목을 추천하는 방식을 사용
- kNN알고리즘을 협업 필터링(collaborative filtering)이라고도 부른다.

k-NN(k nearest neighbor)



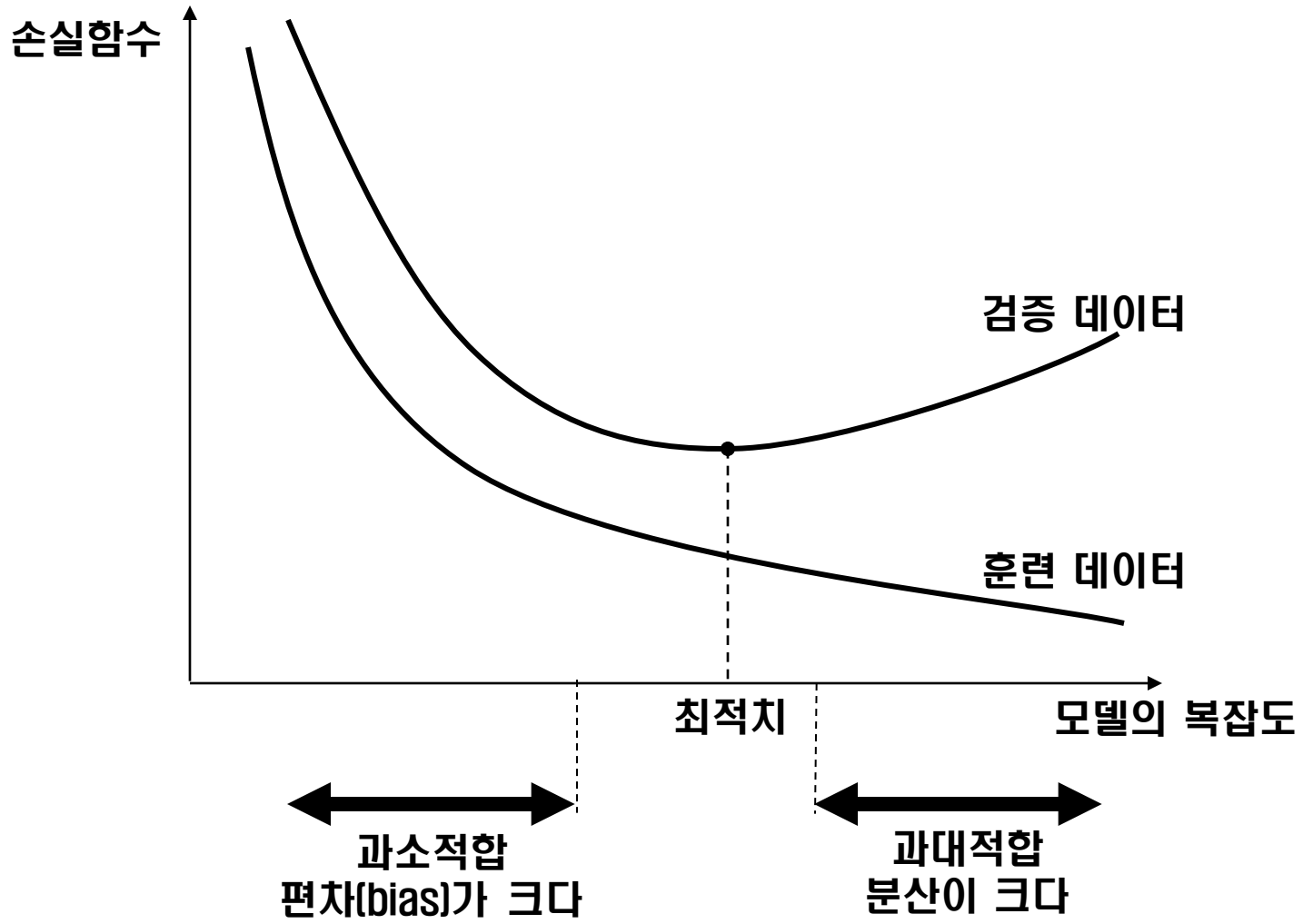
k-NN(k nearest neighbor)



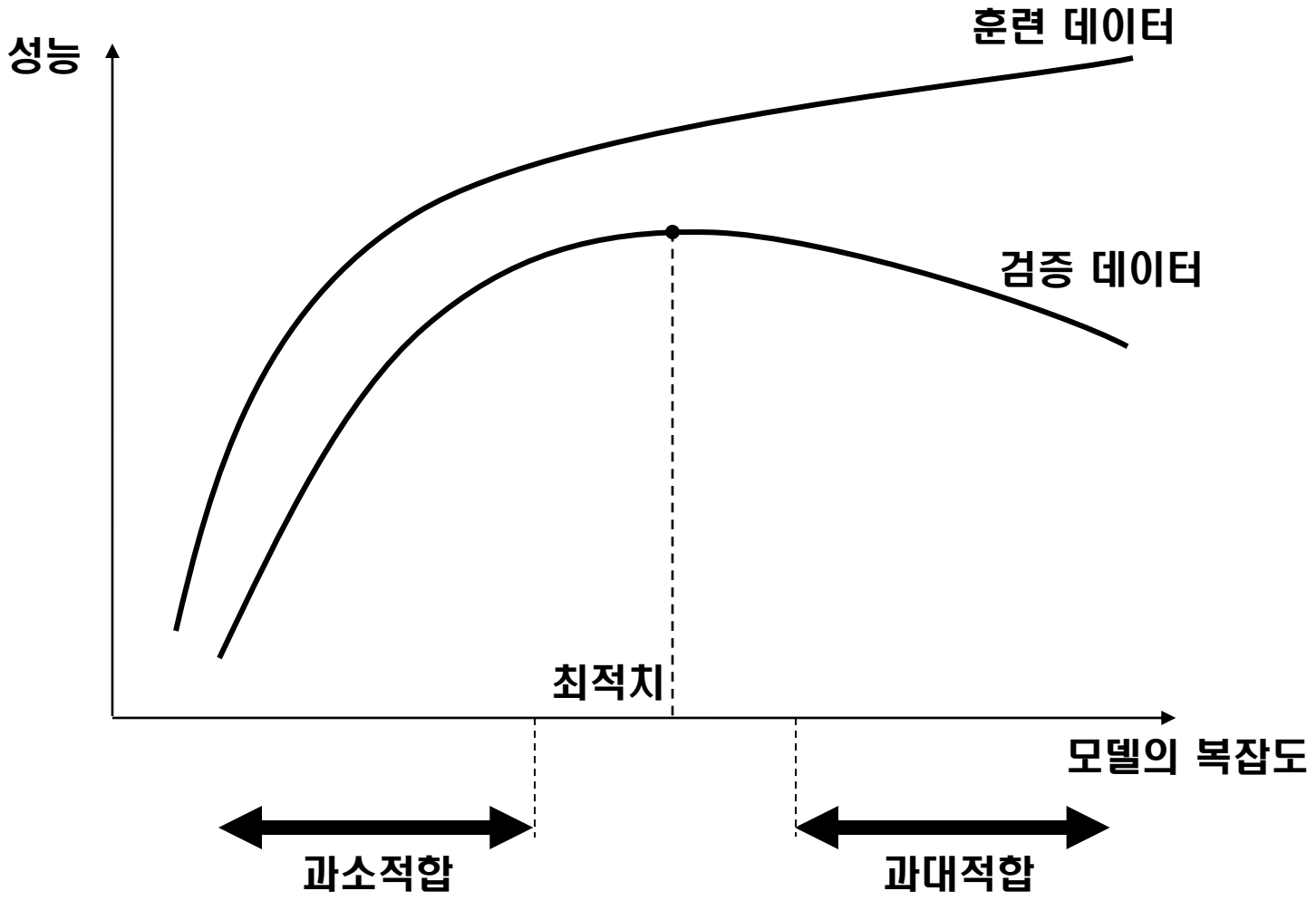
k-변화

- k 값을 너무 작게 잡으면 주변 데이터에 너무 예민하게 반응하고 k 값을 너무 크게 잡으면 주변에 너무 많은 데이터의 평균치를 사용하므로 분류가 무뎌진다.
- 극단적으로 k=N (전체 샘플 수)로 잡으면 항상 전체 데이터의 평균치 값을 예측하게 된다.
  - 영화 추천에서 k=N으로 한다면 이는 평균적으로 가장 많은 사람들이 본 영화 즉, 종합 베스트셀러를 추천하는 것과 같다.
- k값을 작게 잡으면 노이즈에 민감하나 정확도는 올라가고 k를 크게 잡을수록 노이즈에 강하나 정밀한 예측이 어렵다.
- kNN의 단점은 훈련시간이 거의 없는 것에 비해 분류를 처리하는 시간, 즉 알고리즘을 수행하는 시간이 길다는 것이다.

과소 적합과 과대적합 판단 – 손실함수



과소 적합과 과대 적합 판단 – 성능



편향과 분산

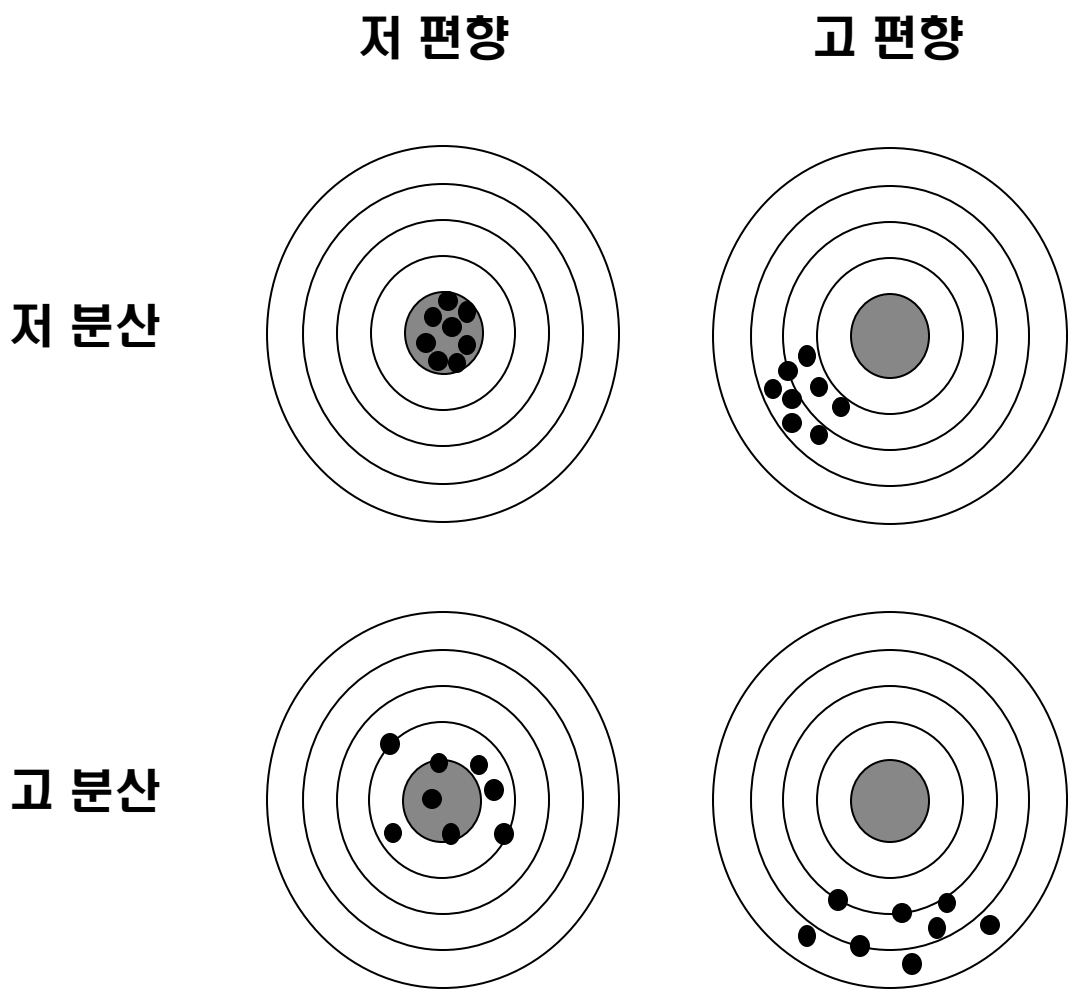
- 예측 모델에서 발생하는 오차는 분산(variance)과 편향(bias) 두가지 성분으로 설명할 수 있다.
- 분산이란 모델이 너무 복잡하거나 학습데이터 민감하게 반응하여 예측 값이 산발적으로 나타나는 것이다.
- 편향이란 모델 자체가 부정확하여 피할 수 없이 발생하는 오차를 말한다.

편향과 분산

- 예를 들어 kNN알고리즘에서 k=1인 경우는 모델이 학습 데이터에 민감하게 반응하여 예측의 변화가 커지므로 분산 성분이 증가한다.
  - 그러나 편향은 발생하지 않는다.
  - 이러한 경우를 과대적합되었다고 한다.

- 반면에 k=N인 경우는 모델은 항상 평균치로 동일하게 예측하므로 분산은 거의 발생하지 않는다.
  - 그러나 모델 자체가 부정확하여 편향이 커진다.
  - 모든 사람의 키를 평균치로 예측하면, 예측치는 평균치로 동일하므로 분산은 없어지지만 편향 오차가 클 수 밖에 없다.
  - 이는 과소적합의 현상이다.

편향과 분산



편향과 분산

- 모델이 훈련 데이터에 너무 종속적이거나 정교하면 (k=1 인 경우) 분산이 늘어나고 편향은 줄어든다 (위 그림에서 좌 하)

- 모델이 너무 단순하면 (k=N인 경우) 분산을 줄어드나 모델이 부족하여 생긴 편향이 증가한다.

- 뒤에서 설명할 결정 트리에서는 트리의 깊이(depth)에 따라서 편향과 분산 오류가 달라진다

정보량

- 데이터(이벤트)가 포함하고 있는 정보의 총 기대치, 정보의 가치
- 정보량을 표현
  - 해당 사건이 발생할 확률(probability)을 사용
- 사건 발생 확률에 따른 정보 가치
  - 확률 = 1 : 정보가 주는 가치가 없다
  - 사건 발생 확률이 낮을수록 : 정보가 주는 가치가 높음
- 정보량
  - 일어날 확률의 역수에 비례
- 정보량 정의

$$log\left(\frac{1}{p}\right)$$

정보량과 엔트로피

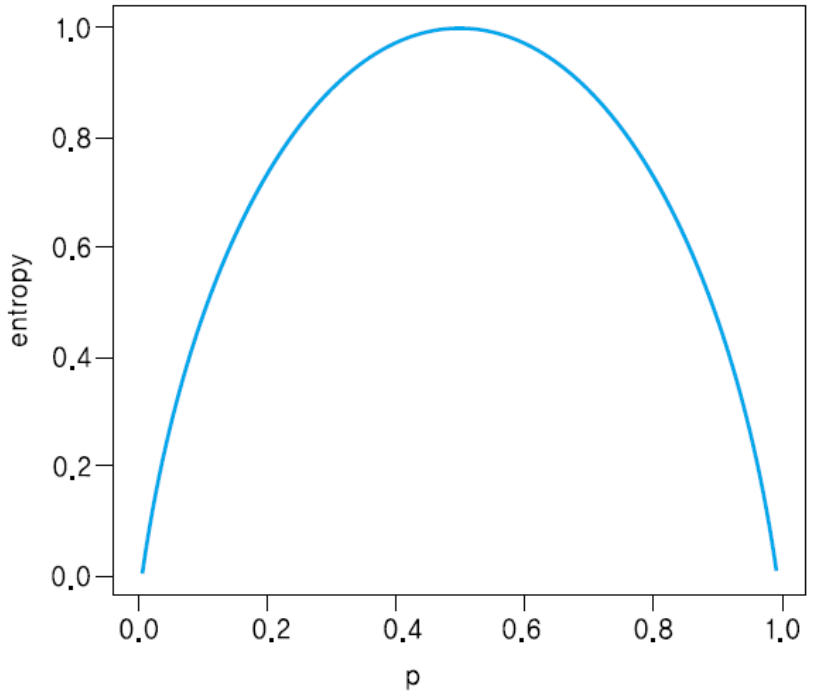
- 정보량의 기대치
  - 어떤 사건이 갖는 가치와 그 사건이 발생할 확률의 곱
  - 이를 엔트로피(entropy)라고 함

- 엔트로피(정보량의 기대치)

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

정보량과 엔트로피

- 바이너리(binary) 사건의 경우
  - 엔트로피는 p가 0.5 일 때 가장 높음
- 즉, 불확실성이 가장 높을 때 엔트로피가 가장 높음

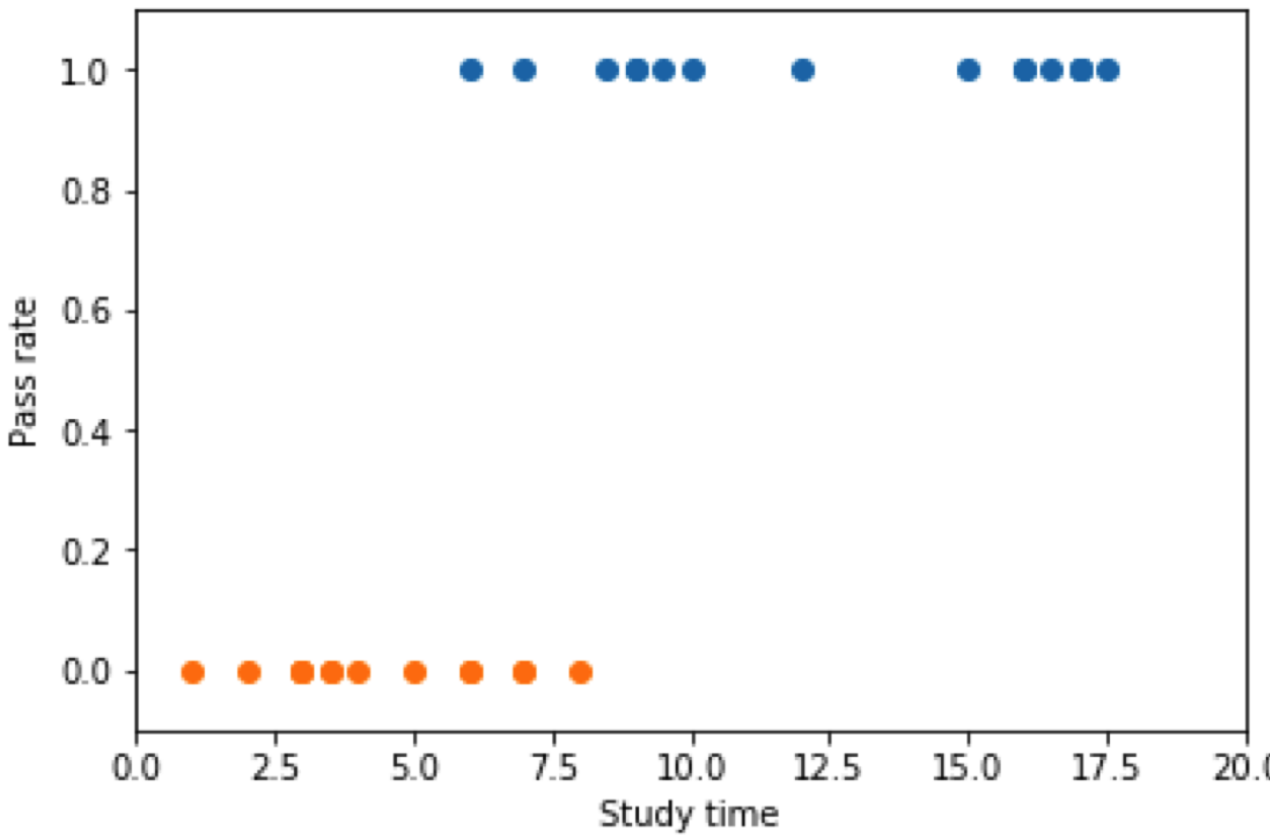


로지스틱 회귀

로지스틱 회귀분석(logistic regression)

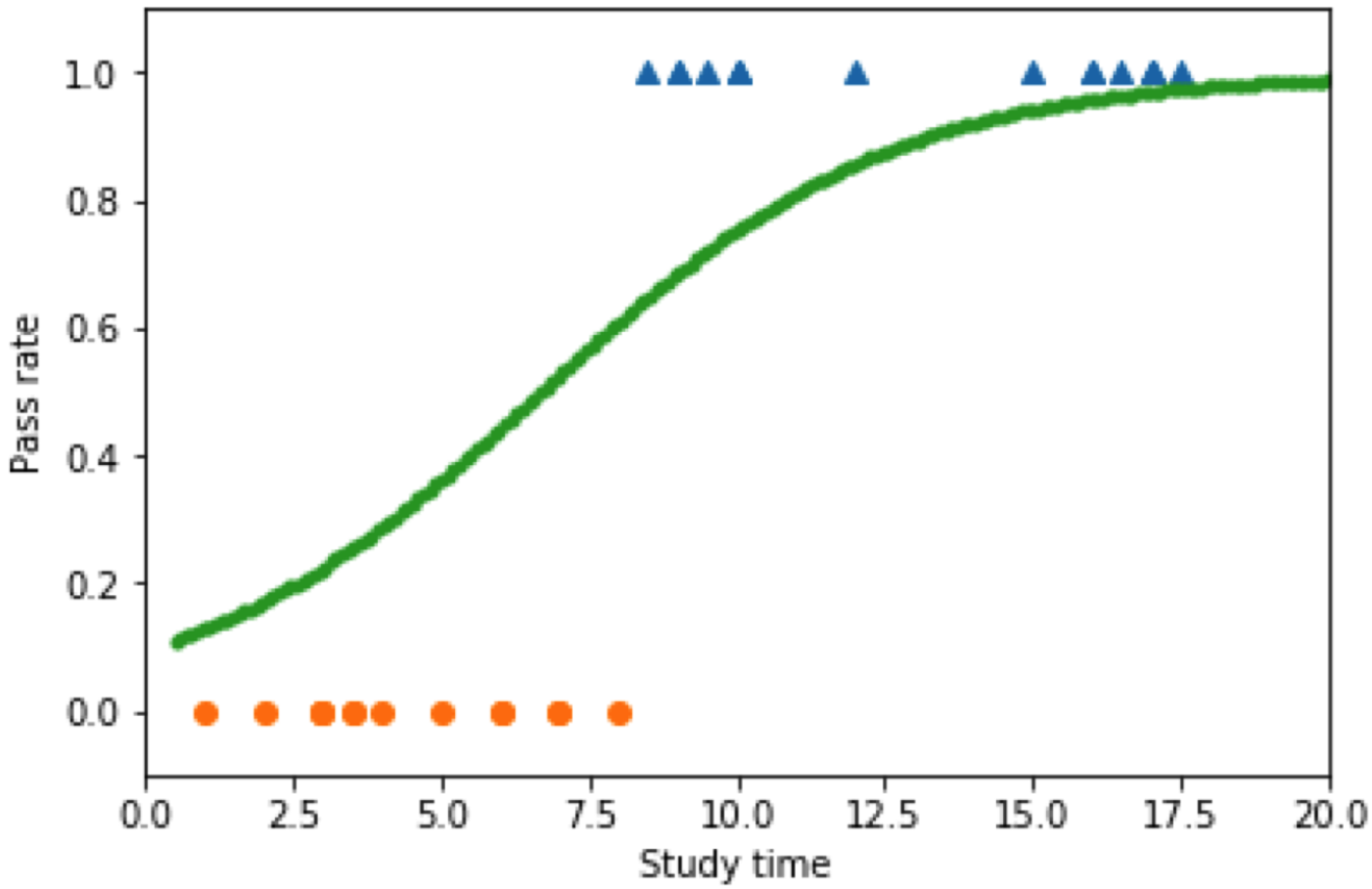
- 임의의 범위를 갖는 값으로부터 0과 1사이의 값을 예측하거나 이진 분류에 사용하는 알고리즘
- 로지스틱 회귀분석은 보통 독립 변수와 종속 변수의 관계를 S형 커브로 매핑함(선형 회귀분석 사용이 불가한 경우)
- 응용 예
  - 신용도 판단
  - 연간 구매량 기준 우수 고객 여부 판단
  - 평가 지표 기준 합격 여부 판단
  - 건강 지표에 따른 건강 여부
  - 팀의 승리/패배 여부 예측
  - 등 여러 경우에 사용

로지스틱 회귀

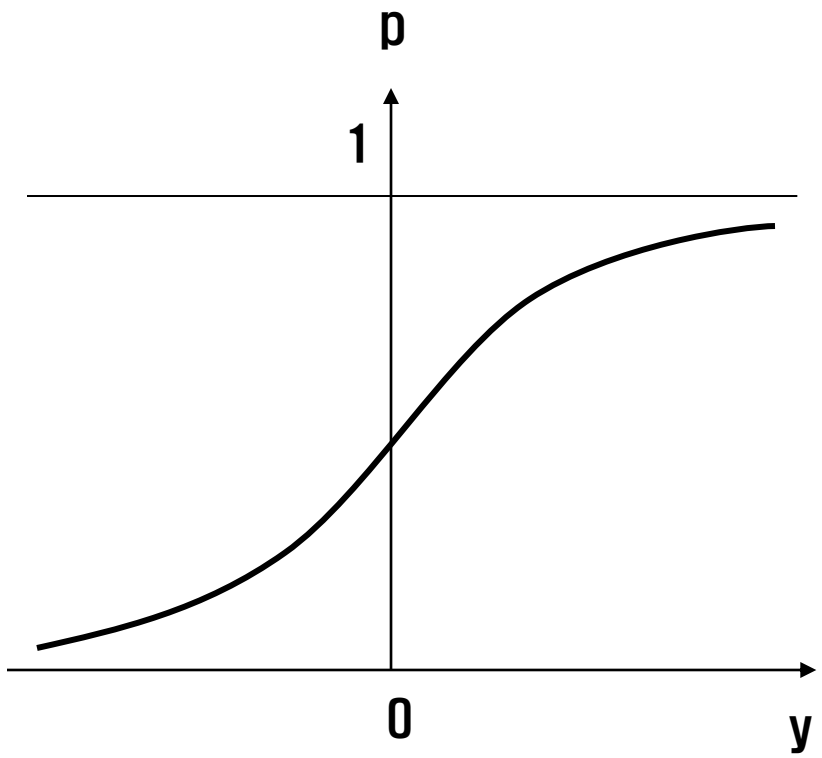


공부 시간과 합격 여부

로지스틱 회귀 모델링


Pass rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Study time

시그모이드 함수



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

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

다항 로지스틱 회귀

- 3개 이상의 클래스 중에 하나를 예측해야 하는 경우
  - 로지스틱 회귀를 그대로 사용할 수 없다
  - 다항 로지스틱 회귀(multinomial logistic regression)를 이용
    - 다중 분류 (multiclass classification)
    - Softmax 함수 사용

- 이진 분류를 이용한 다중 분류 (예, A, B, C 분류)
  - A, {B, C}
  - B, {A, C}
  - C, {A, B}

- 한번에 다중 분류 가능 : 랜덤 포레스트, 나이브 베이즈 등
- 소프트맥스(softmax) 함수를 사용

$$\sigma(j) = \frac{\exp(\mathbf{w}_j^\top \mathbf{x})}{\sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})} = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$$

다항 로지스틱 회귀

- 소프트맥스(softmax) 함수를 사용

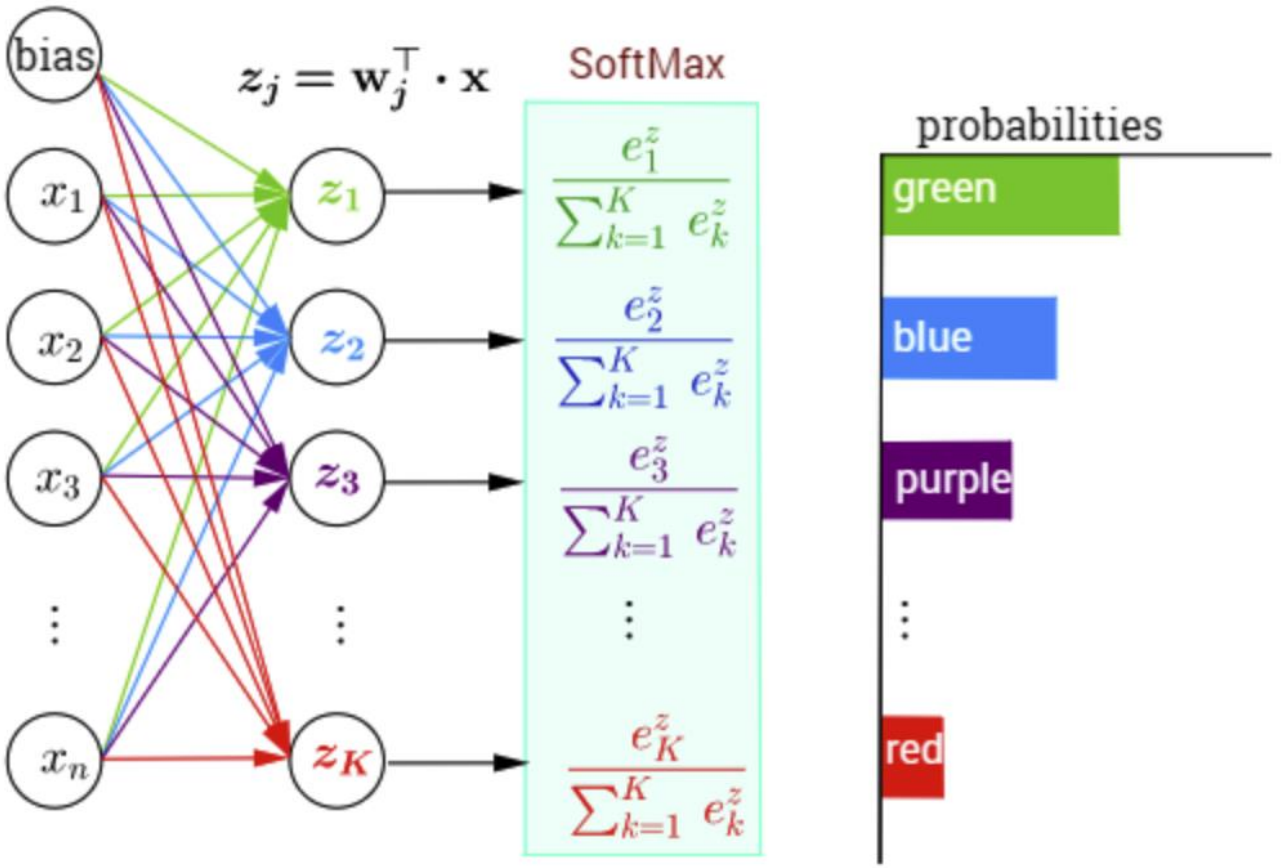
$$\hat{p}_k = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=i}^{K} \exp(s_j(x))}$$

  - $\widehat{p_k}$: 클래스 k에 속할 확률
  - x : 주어진 샘플
  - $S_k(x)$: 소프트맥스 회귀 모델이 각 클래스k에 대한 점수

- 예) 얼굴을 보고 한국인, 중국인, 일본인 3 class로 구분
  - 어떤 샘플에 대해서 모델의 예측 값이 1.5, 2.0, 1.8 로 가정
  - 소프트맥스 적용 : 0.23, 0.43, 0.34 (합 = 1.0)
    - 각각의 점수를 확률처럼 사용 가능
    - 모델 예측 값이 음수(-)이어도, 소프트맥스 출력 값은 0~1 사이 값

소프트맥스

- 상대적인 점수 비교 : 확률처럼 0~1 사이 값으로 매핑



수고하셨습니다

Q & A