KNN과 로지스틱 회귀로 실습

다중회귀분석

여러개의 독립 변수가 종속 변수에 영향을 주는 경우 다중회귀분석을 사용

즉, 독립변수 X의 개수가 2개 이상일 경우 다중회귀분석이라 한다.

$y = b + a_1X_1 + a_2X_2 + a_3X_3 + \dots + a_nX_n$와 같이 표현할 수 있다.

김시신

보강 안내

- 이번주 중에 신청 인원에 한해
  시간을 맞추어서 2시간 정도 보강이 있을 예정 (대면, AI과방)

- 목적
  - 기본 언어에 익숙해 지고 로직을 손에 익숙해 지는 시간을 갖기 위해
- 내용
  - 여러 데이터 불러와서 분석하고 모델링 실습
  - 코드 리뷰(코드 읽고 이해할 수 있도록 여러 예제 코드 보고 해석)

Contents

- 복습
- KNN 알고리즘과 실습
- 로지스틱 회귀로 이진 분류와 실습
- 로지스틱 회귀로 다중 분류와 실습
- 결정트리 알고리즘으로 다중 분류 실습

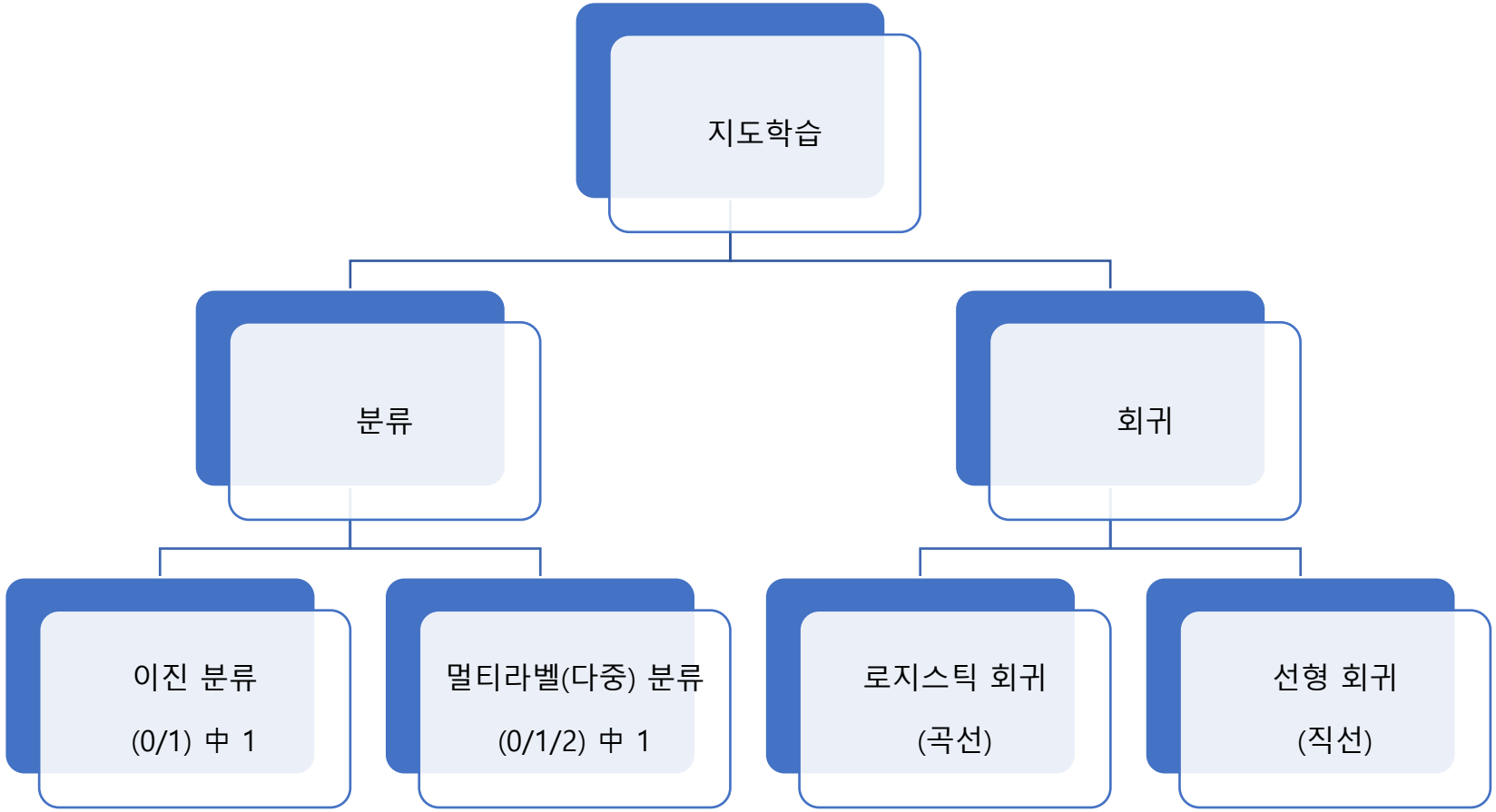
복습

- X_train,X_test,y_train,y_test=train_test_split(X,y)

- X-data의 feature(target값을 제외한 나머지 칼럼의 데이터)
- y-data의 target(예측하려는 목표),class,label

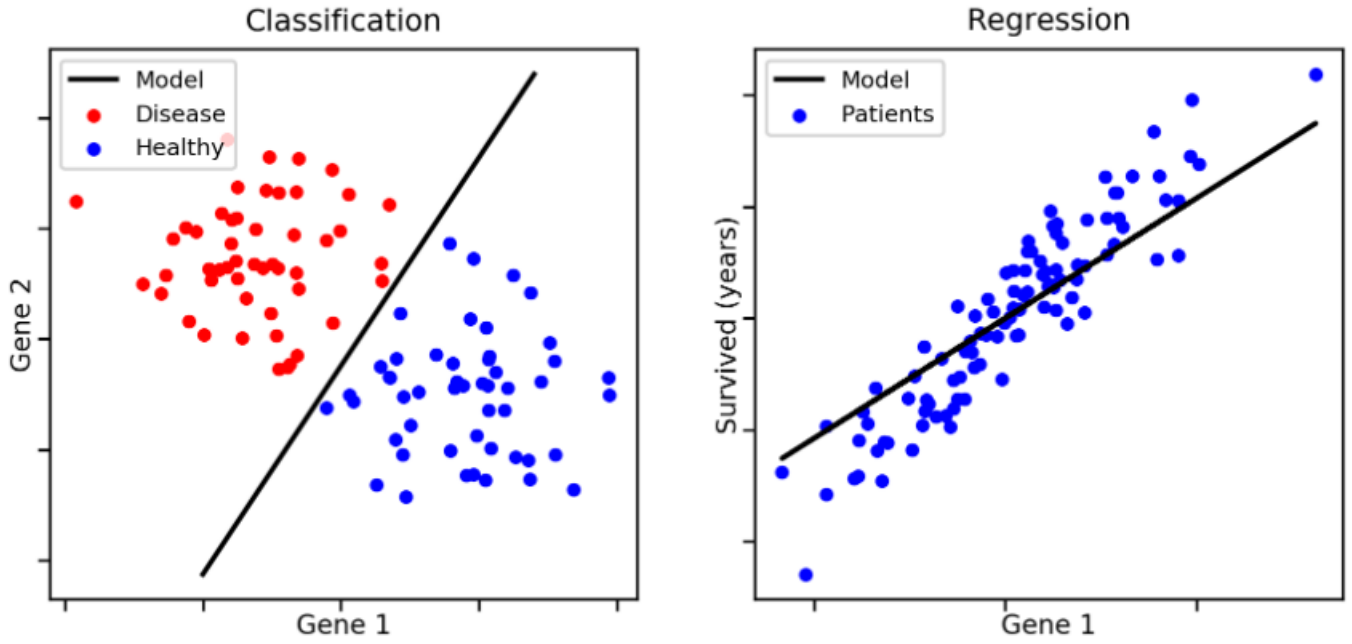
(단, X_train 등의 변수는 바뀔 수 있으나 위치의 의미는 동일)

복습



로지스틱 "회귀 " 로 "분류"를?

- 회귀 – 연속적 (ex. 1.1, 2.1, 5.33,~)
- 분류 – 이산적 (ex. 0,1,2)



https://velog.io/@jeromecheon/ML-3%ED%8E%B8-Linear-tasks-classification%EB%B6%84%EB%A5%98-2-%EC%83%9D%EC%84%B1%EB%AA%A8%EB%8D%B8-%EC%8B%9D%EB%B3%84%EB%AA%A8%EB%8D%B8-%EC%A0%91%EA%B7%BC%EB%B2%95

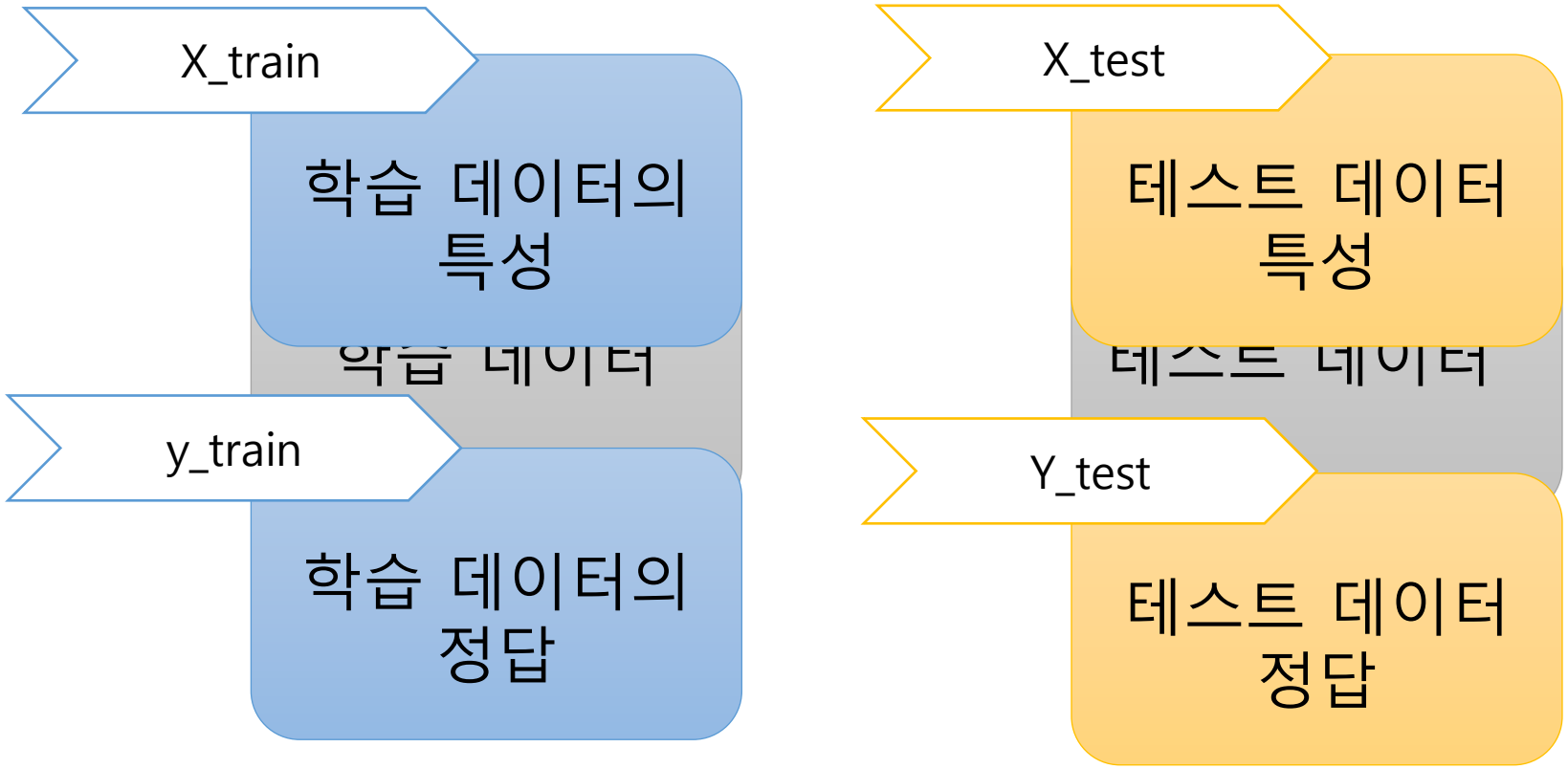
로지스틱 “회귀 " 로 “분류”를?

- 로지스틱 함수는 이름은 회귀지만 분류 모델
- 로지스틱 회귀 알고리즘은 선형회귀와 동일하게 선형 방정식 학습

- 로지스틱 회귀로 분류를 한다는 것을 연속적인 결과를 표준화시켜서 임곗값을 기준으로 “분류 " 하는 것
- (사이킷런은 0.5 기준)

- 연속적인 결과 -> 클래스(X)
- 연속적인 결과를 이용하여 라벨을 이산적으로 분류(O)

4가지 인자


X_train
학습 데이터의 특성
y_train
학습 데이터의 정답
X_test
테스트 데이터 특성
Y_test
테스트 데이터 정답

흐름

- 학습 데이터와 테스트 데이터를 분리
- 학습데이터의 특성(X_train)과 그에 대한 정답(y_train)을 가지고 모델에 학습시킴
- 테스트 데이터의 특성(X_test)을 넣어서 정답을 예측하고
- 모델에서 예측한 정답과 실제 정답(y_test)를 비교해서 정확도 측정

+변수명 X_train,y_train은 바뀔 수 있으나 그 **위치에 있는 것의 의미는 동일함 즉 fit 메서드의 첫번째 인자 두 번째 인자 -~~**

모델 학습의 주된 패턴

- 객체=모델() // 객체에 모델을 할당
- 객체.fit(X_train,y_train) // 객체에 접근해 학습데이터의 feature,target을 주어 학습

- Y_pred=객체.predict(X_test) // 테스트 데이터를 학습한 모델에 입력해 예측하게 하기
- 객체.accuracy(y_pred,y_test)

테이블에서 예시

인덱스	꽃받침 길이	꽃받침 너비	꽃잎 길이	품종
0	5.1	3.5	1.4	0
1	4.9	3.0	1.4	1
2	6.7	3.0	5.2	1
3	6.3	2.5	5.0	2
4	5.0	3.6	1.4	2
5	4.6	3.1	5.4	1
6	5.0	3.0	1.6	2
7	4.7	3.2	1.6	0
8	6.7	2.5	5.0	1
9	6.3	3.0	5.2	1

Train data

Feature - X

Test data

Target - y
(Label, Class)

실제 데이터에서의 흐름

인덱스	꽃받침 길이	꽃받침 너비	꽃잎 길이	품종
0	5.1	3.5	1.4	0
1	4.9	3.0	1.4	1
2	6.7	3.0	5.2	1
3	6.3	2.5	5.0	2
4	5.0	3.6	1.4	2
5	4.6	3.1	5.4	1
6	5.0	3.0	1.6	2
7	4.7	3.2	1.6	0
8	6.7	2.5	5.0	1
9	6.3	3.0	5.2	1

Train X
-data feature

Train y
-data target

Test X
-data feature

Test Y
- Data target

실제 데이터에서의 흐름

인덱스	꽃받침 길이	꽃받침 너비	꽃잎 길이	품종
0	5.1	3.5	1.4	0
1	4.9	3.0	1.4	0
2	6.7	3.0	5.2	1
3	6.3	2.5	5.0	2
4	5.0	3.6	1.4	2
5	4.6	3.1	5.4	1
6	5.0	3.0	1.6	2
7	4.7	3.2	1.6	
8	6.7	2.5	5.0	
9	6.3	3.0	5.2	

Train X
-data feature

Train y
-data target

학습

Test X
-data feature

예측

정확도 측정

실제 데이터에서의 흐름

인덱스	꽃받침 길이	꽃받침 너비	꽃잎 길이	품종
0	5.1	3.5	1.4	0
1	4.9	3.0	1.4	1
2	6.7	3.0	5.2	2
3	6.3	2.5	5.0	2
4	5.0	3.6	1.4	2
5	4.6	3.1	5.1	1
6	5.0	3.0	1.6	2
7	4.7	3.2	1.6	0
8	6.7	2.5	5.0	1
9	6.3	3.0	5.2	1

Train X
-data feature

Train y
-data target

예측

정확도 측정

Test X
-data feature

Test Y
- Data target

모델을 돌린다?

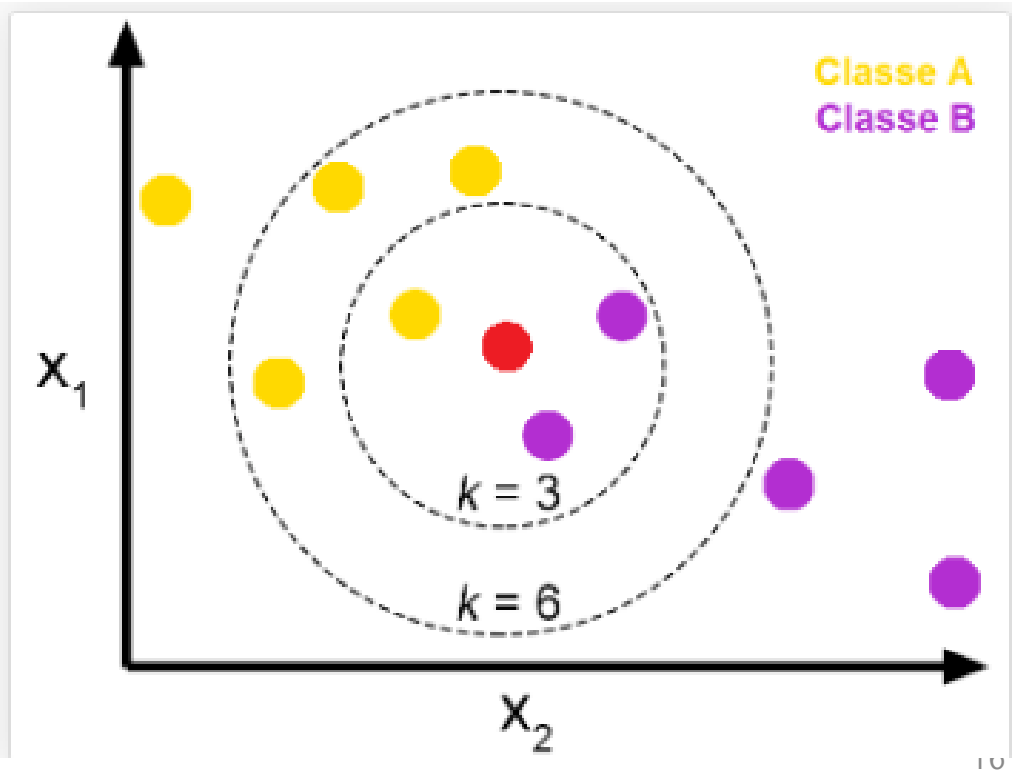
- 어렵지 않다(사이킷런 내장 알고리즘의 모델)
- 어려운 건 어떻게 좋은 성능을 내는 방법론(알고리즘)을 택하느냐
- 전처리와 feature 엔지니어링 (문장 칼럼 -> 키워드 단어 -> 인코딩)
- 데이터를 train_test_split을 기준으로 나누고 reshape같은걸로 형식에 맞춰서 넣어주기만 하면 된다
- 따라서 배울 때도 직접 구현해 보는게 좋은거같다
- 따라서 오늘 배울 내용도 모델의 역할과 목적 원리같은걸 전달하려고 한다.
- 데이터 분석 후 처리가 중요하다
- 기반 지식을 잘 배워놓아서 이후에 활용할 수 있도록하는게 중요하다.
- Stratified K Fold -> 언제쓰이냐 -> 어떤 데이터에서 모델을 학습시켜서 결과를 냈는데 스코어가 낮다. 근데 라벨의 분포가 불균형하다. -> 쓸 타이밍
- 머신러닝을 잘한다

-> 어떤 상황의 데이터에 어떤 모델을 돌리지 알고 feature engeneering을 해서 잘 예측한다

KNN 이란?

- 거리기반 분류 분석 모델
- 새로운 데이터가 들어왔을 때 기존 데이터의 어떤 그룹에 속하는지 분류
- 하이퍼 파라미터 K로 범위 조정

  (K=1,3,5 ~)



로지스틱 회귀(RogisticRegression)

- 로지스틱 회귀란 이름은 회귀이지만 분류 알고리즘
- 아래 다중 회귀 수식을 이용하여 여러 개의 독립 변수를 입력 받고
- Y값을 Sigmoid 함수의 입력으로 넣어 정답을 예측하는 방식

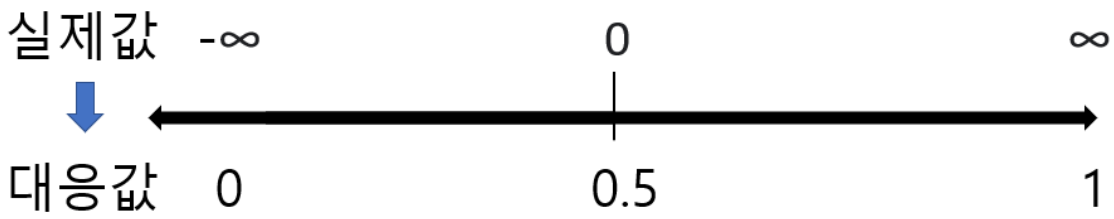
$$y = w_1 x_1 + w_2 x_2 + \ldots w_n x_n + b$$

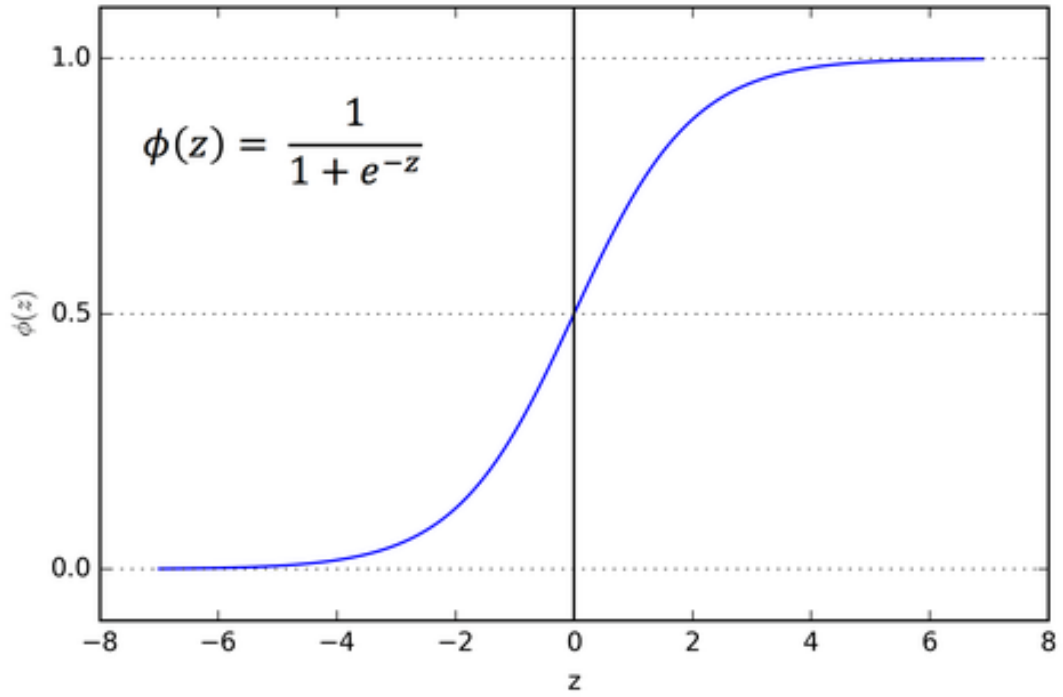
Y: 예측하려는 변수
B(bias): 편향
X: 입력 feature
W(weight): 가중치/회귀 계수

시그모이드(Sigmoid) 함수

- 시그모이드 함수는 출력값을 0~1사이의 값으로 압축해준다.

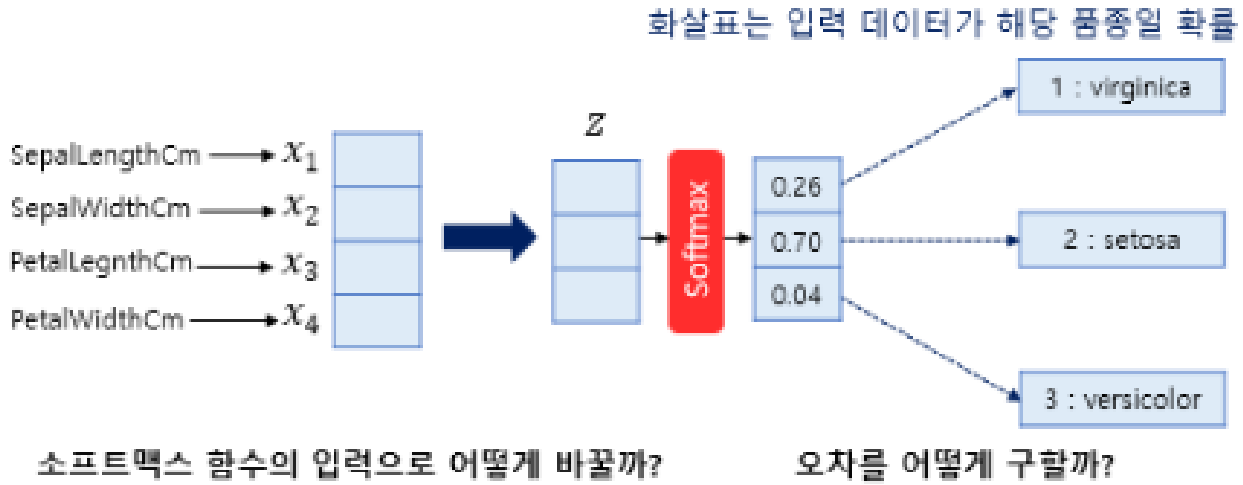
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$





소프트맥스(Softmax) 함수

- 소프트맥스 함수는 여러 개의 선형 방정식의 출력값을 0~1 사이로 압축하고 전체 합이 1이 되도록 만들어줌
- 다중 분류 모델에서 사용



로지스틱 회귀 함수의 인자

- LogisticRegression(C=규제 값, max_iter=최대 반복 횟수)
  - C - 기본값: 1
  - 규제와 C는 반비례,
    -> C가 크면 규제가 작은 것
    -> C가 작으면 규제가 큰 것

  예시) 기본값 1->20은 규제를 줄이는 것(기본값 증가)

  - Max_iter – 최대 반복 횟수
  기본값: 100
  - Predict_proba(): 예측 확률 반환
  - Decision_function():선형 방정식의 출력을 반환

실습 총 정리

- 로지스틱 회귀 모델(LogisticRegression)를 직접 구현해보며 계산되는 과정을 봄
- 데이터의 feature들이 X(1,2,3~)으로 가중치 W(1,2,3~)과 편향(b)를 모델이 구해주면 z값 구할 수 있음
- Z값은 범위가 다양하게 나와서 이를 시그모이드 또는 소프트맥스 함수로 0에서 1사이의 값으로 정규화해줌
  -> 정규화한 값이 확률이 되어서 그 확률이 가장 높은 값의 클래스로 예측

감사합니다.