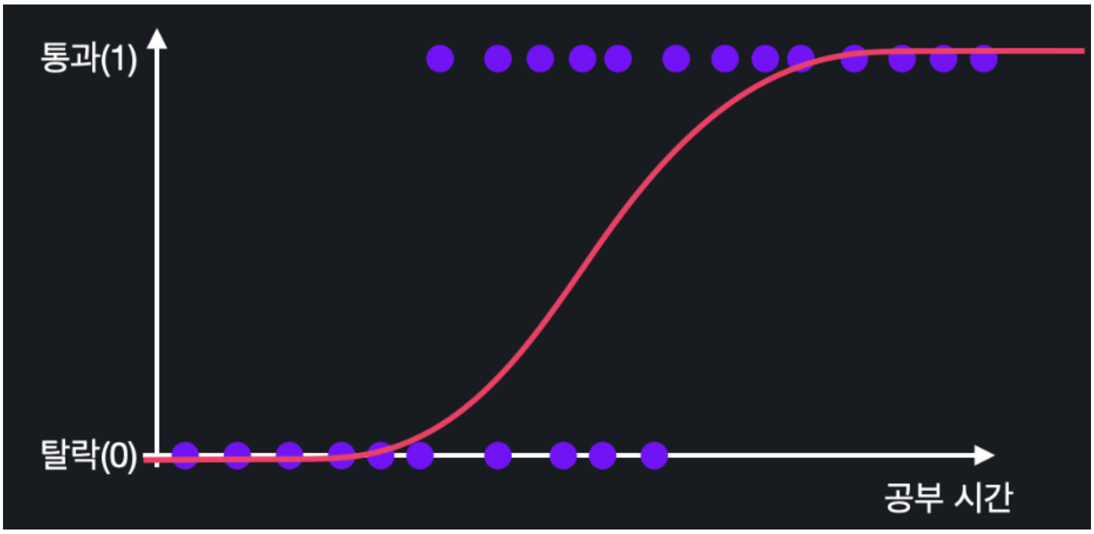
기본 지도 학습 알고리즘들(4)

로지스틱 회귀 (Logistic Regression)

- 회귀 : 연속적인 값을 예측하는 것
    - but 예외적인 데이터 하나하나에 민감하게 반응함
- 분류 : 정해진 몇 개의 값 중에 예측하는 것
    - 시그모이드 함수

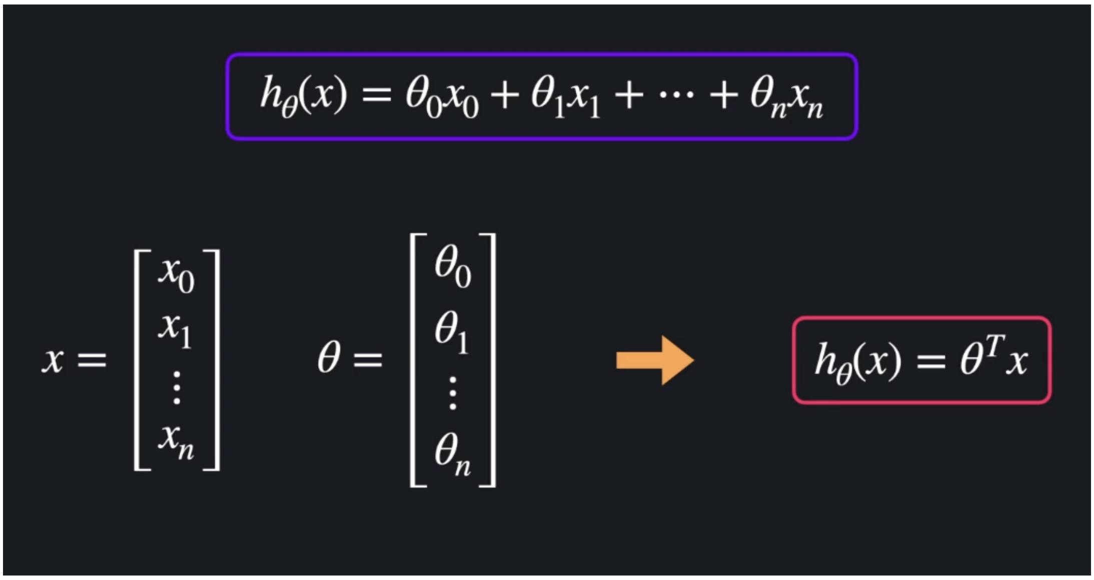
$$S(x) = \frac{1}{1 + e^{-x}}$$

    - 그래프 형태

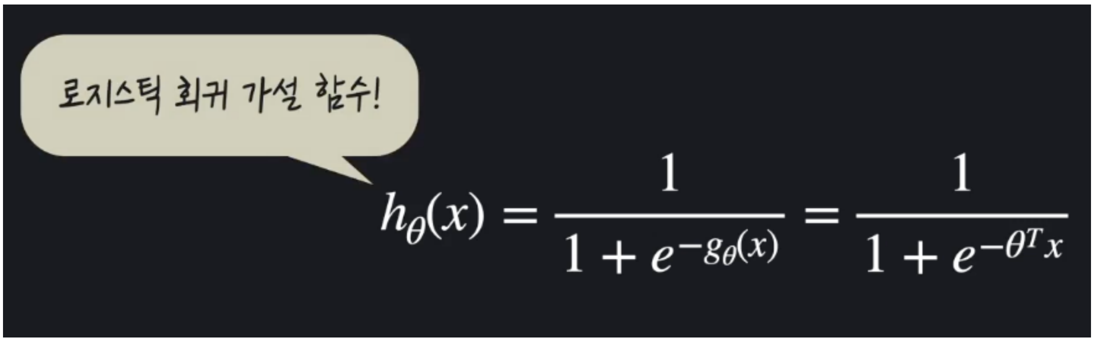


- 선형 회귀는 결과가 범위 없이 얼마든지 크거나 작아질 수 있음
- 무조건 0과 1사이의 값이 나오는 시그모이드 함수가 분류에 더 적합!
- 로지스틱 회귀
    - 결국 리턴 값이 0과 1사이의 연속적인 값
    - 0.5보다 큰지 작은지로 분류

- 가설 함수
  - 입력 변수를 받아서 목표 변수를 예측해주는 함수
  - 선형 회귀에서의 가설 함수

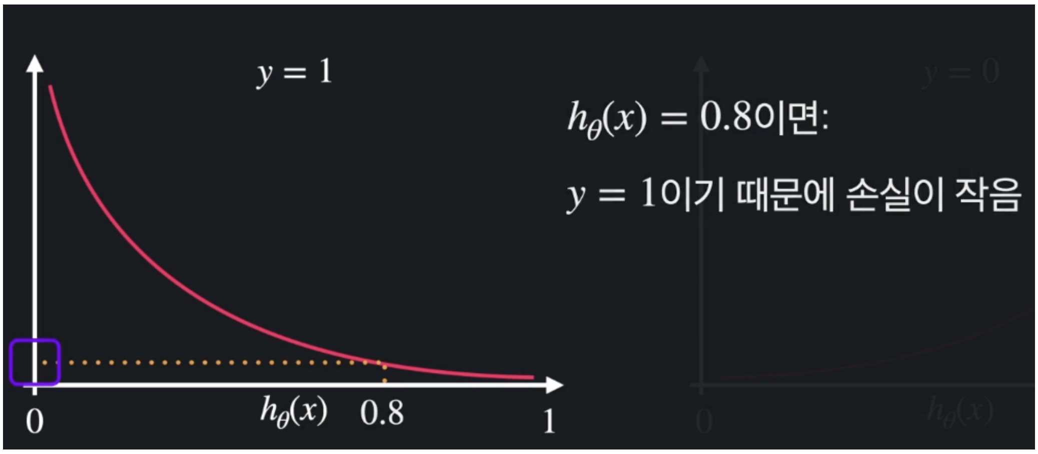


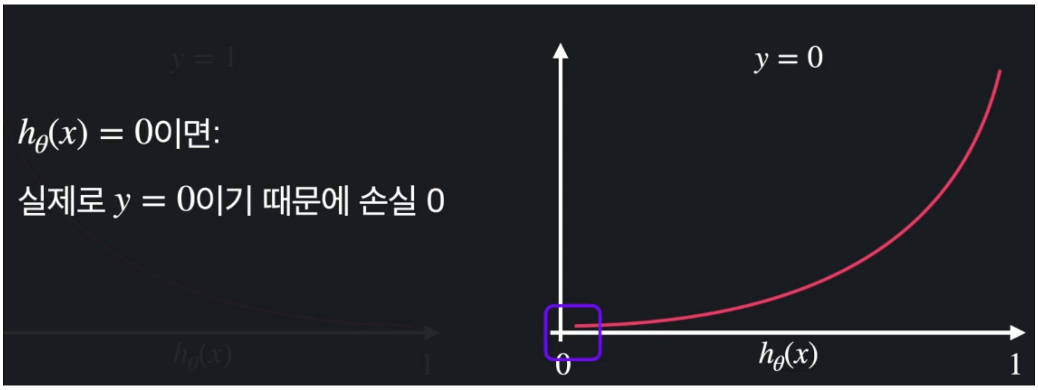
  - 로지스틱 회귀 가설 함수



    - 벡터 x를 가설 함수에 넣어 0.9라는 값이 리턴되었다고 하면 목표 변수가 1일 확률이 0.9 이므로 90%를 의미. 분류를 하자면 이 학생은 통과한 학생으로 생각하면 된다.
    - 분류를 구분하는 경계선 : Decision Boundary
    - 로지스틱 회귀 손실 함수
      - 로그 손실(log-loss / cross entropy) ← 선형 회귀에서 MSE 와 같은 개념

$$logloss(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & y = 1 \\ -\log(1 - h_\theta(x)) & y = 0 \end{cases}$$





- 손실의 정도를 로그 함수로 결정하기 때문에 로그 손실

$$logloss(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

- 로지스틱 회귀 손실 함수

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

- 로지스틱 회귀 경사 하강법
    - 선형 회귀든 로지스틱 회귀든 경사 하강할 때는 항상 이렇게 똑같이 함

$$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta)$$

$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta)$$

$$\theta_2 = \theta_2 - \alpha \frac{\partial}{\partial \theta_2} J(\theta)$$

- 편미분한 결과

$$\theta_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 = \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

선형 회귀랑 똑같음

- 즉, h(세타) 값만 다른 것

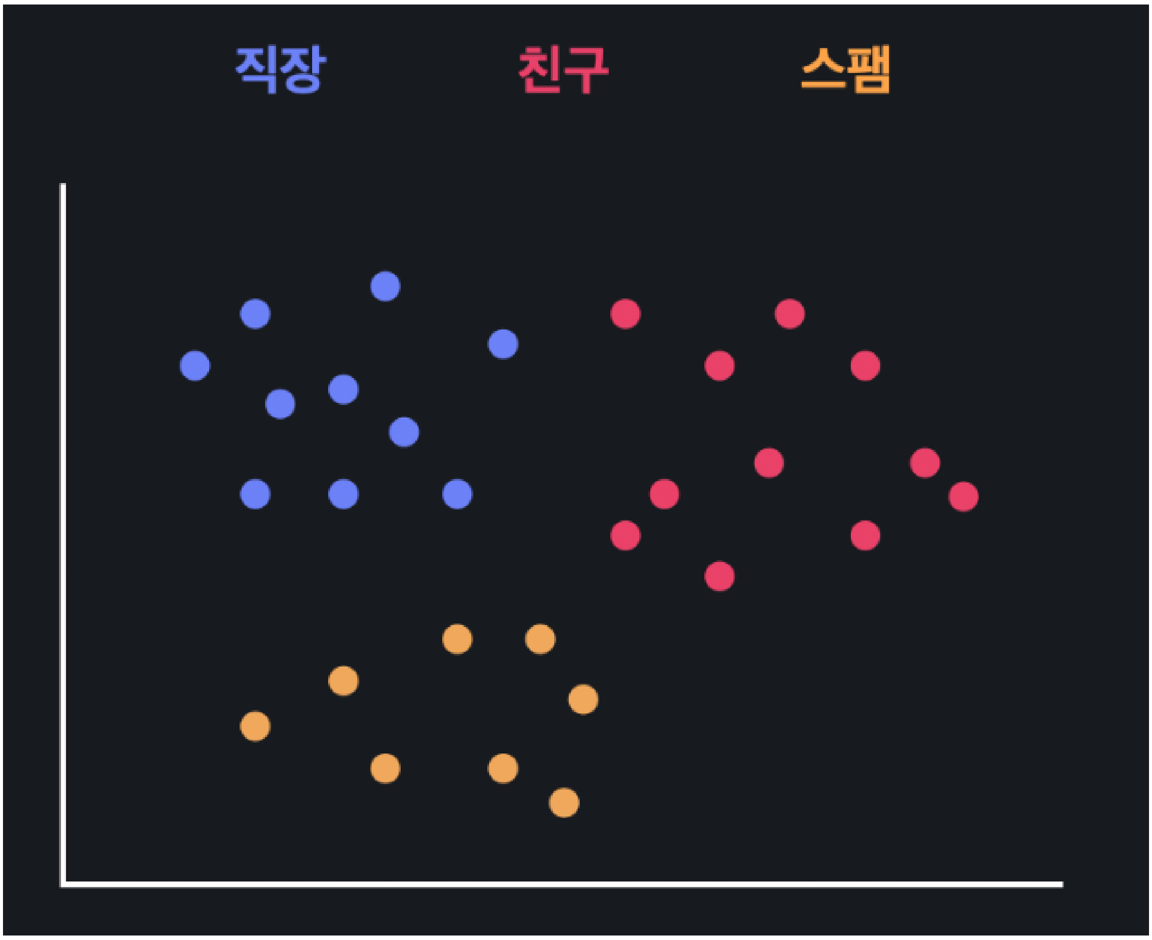
$$\theta_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 = \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

선형 회귀랑 똑같음

- 분류가 3개 이상일 때



  - 아래와 같은 경우는 어떻게 분류할까?
    - 입력 변수 x를 가설 함수 h0, h1, h2에 집어 넣었을 때 가장 큰 값을 나온 것이라고 판단

- 선형 회귀 → 단순 행렬 계산만으로도 최적의 세타 값들을 구할 수 있었음
- but 로지스틱 회귀에서는 정규 방정식과 같은 단순 행렬 연산만으로는 손실 함수의 최소 지점을 찾을 수 없음
    - 선형 회귀는 손실함수 J(MSE)가 아래로 볼록 할 뿐만 아니라, 편미분 원소들을 모두 선형식으로 나타낼 수 있기 때문에 정규 방정식처럼 단순 행렬 연산만으로도 최적의 세타 값들을 구할 수 있음.
    - 로지스틱 회귀에서는 손실 함수 J(로그손실)가 아래로 볼록. 따라서 경사 하강법을 사용하면 항상 최적의 세타 값을 찾아낼 수 있음. 하지만 J에 대한 편미분 원소들이 선형식이 아님. 가장 기본적으로 세타가 e의지수에 포함 → 지수에 포함된 식은 일차식으로만 표현하기가 불가능.

Logistic Regression.ipynb