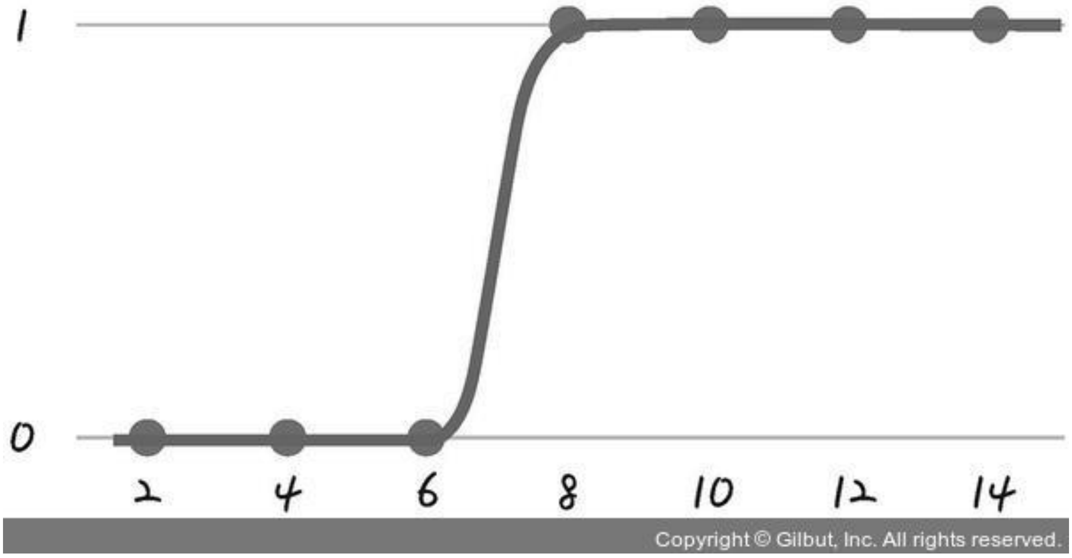
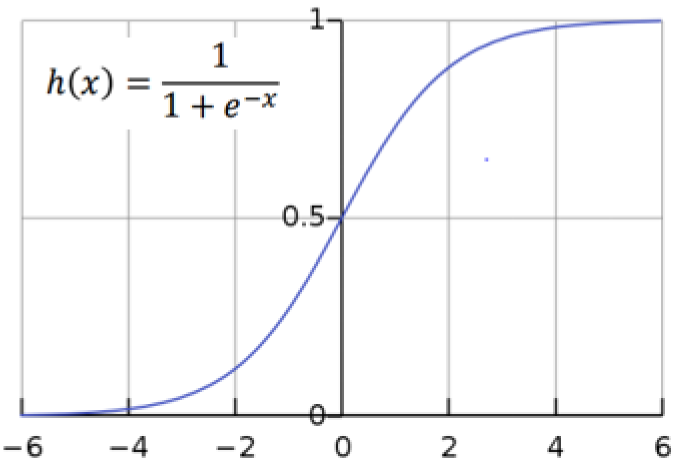
1. 로지스틱 회귀의 정의

합격, 불합격과 같이 참(1)과 거짓(0)값을 갖는 점들의 특성을 담아낼 수 있는 선을 긋는 작업이다. 1 과 0 사이의 값이 없으므로 직선으로 그리기 어렵다. 따라서 아래와 같이 S 자 형태로 그려진다.



2. 시그모이드 함수

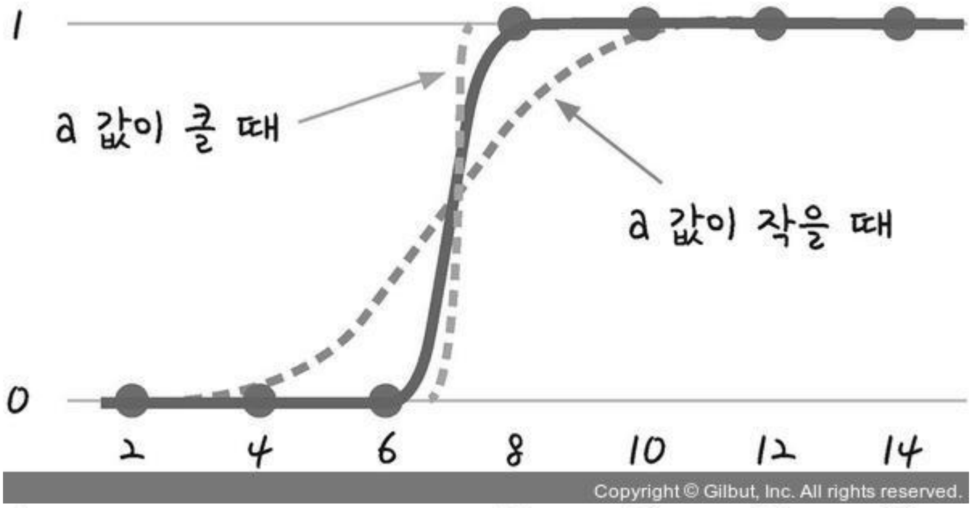


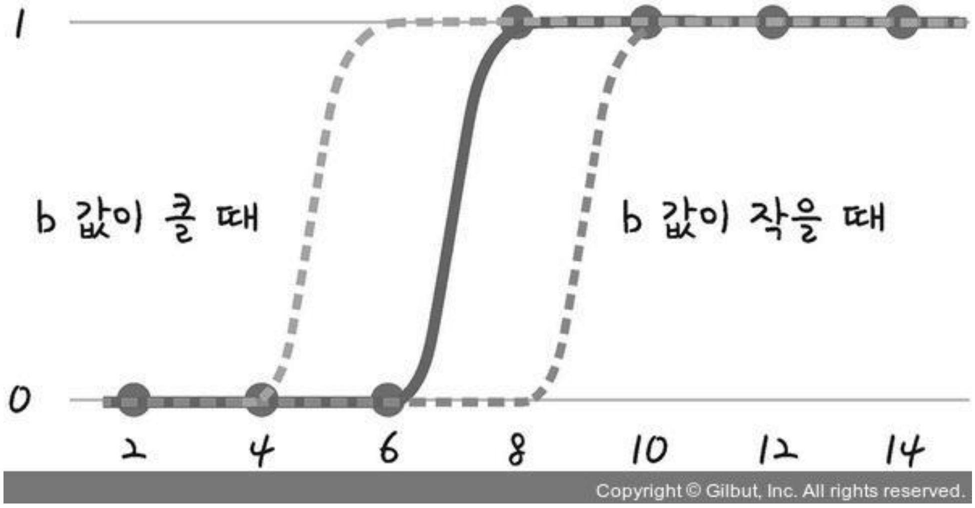
Sigmoid Function

시그모이드 함수는 y 값이 0 과 1 사이이다.

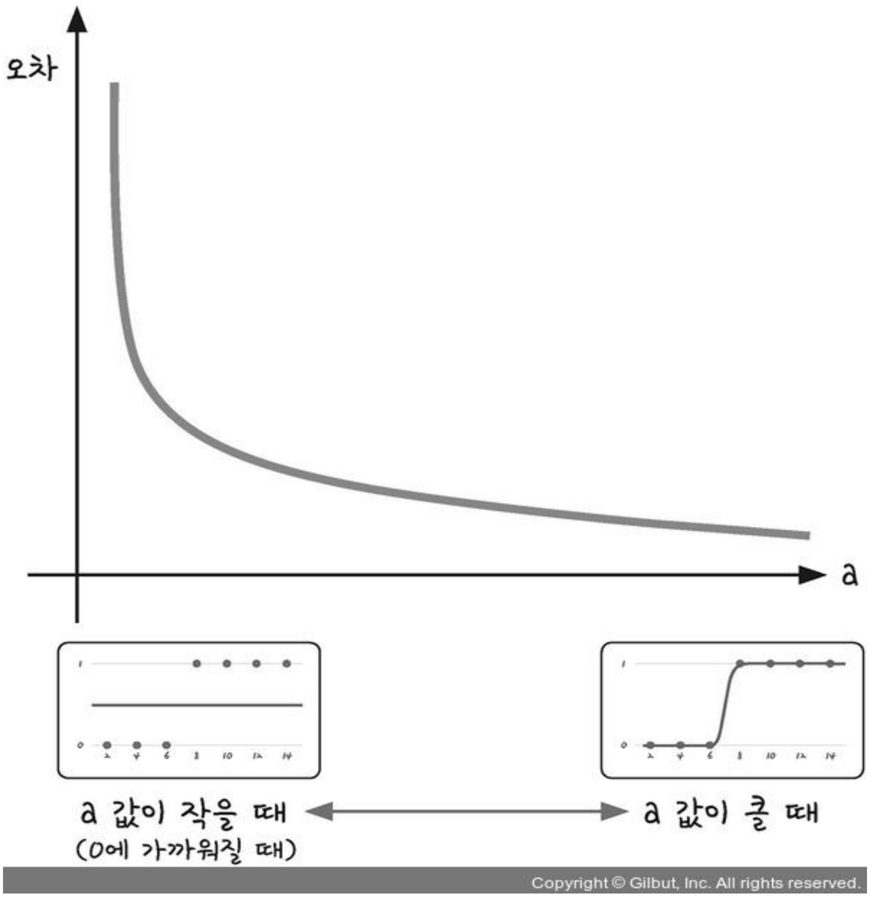
$$y = \frac{1}{1 + e^{(-ax+b)}}$$

이 식에서 a 는 그래프의 경사도를, b 는 그래프의 좌우 이동을 의미한다.



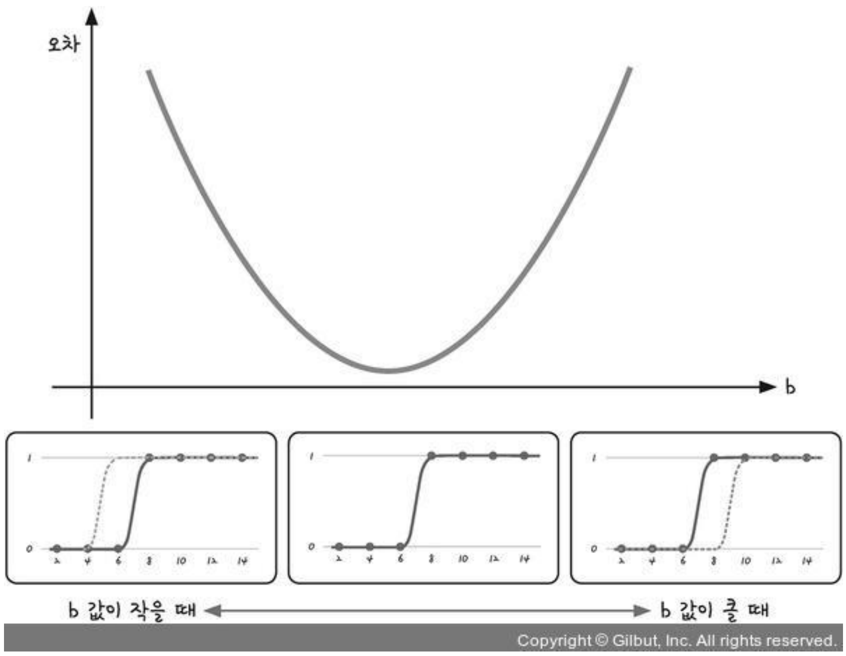


따라서 a 와 b 의 값에 따라 오차가 변한다. a 값에 따라 변화하는 오차를 그래프로 나타내면 아래와 같다.



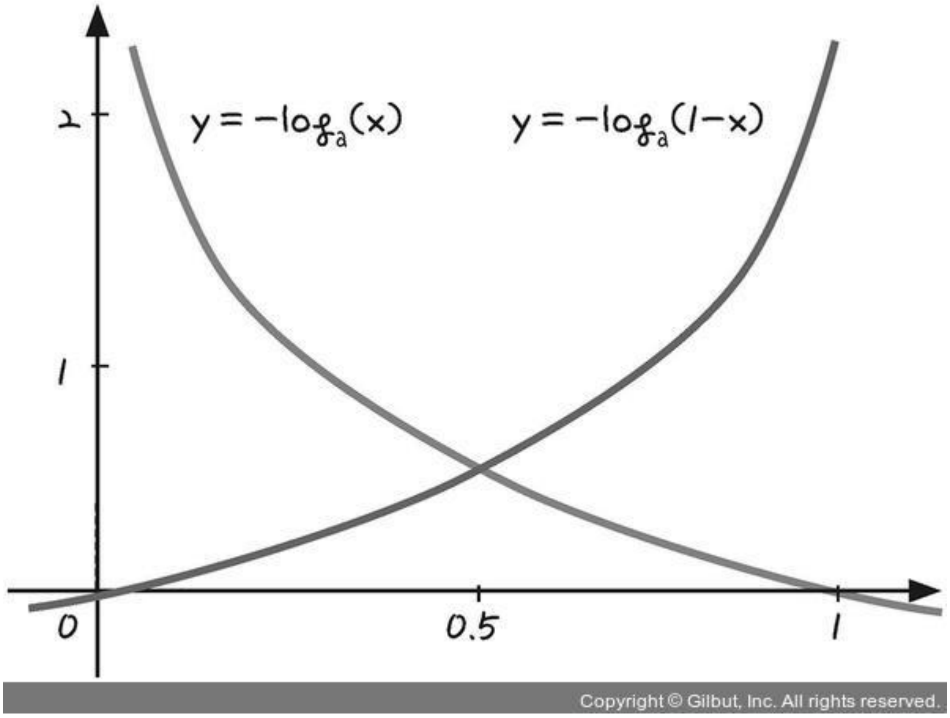
a 값이 작을 때는 0 또는 1 의 값을 아예 나타내지 않는 직선이 된다. 따라서 오차가 무한대로 증가한다.

하지만 a 값이 크다고 해서 오차가 사라지지는 않는다.



b 값이 너무 크거나 작을 경우 오차는 위와 같이 이차함수 그래프와 유사한 형태로 나타난다.

3. 로그 함수



실제 값이 1 일 때 예측값이 0 에 가까워지면 오차가 커진다. (-logh)
반대로, 실제 값이 0 일 때 예측 값이 1 에 가까워지면 오차가 커진다. (-log(1-h))
아래의 식을 통해 이를 해결할 수 있다.

$$-\{\underbrace{y\_data\,\log h}_{A} + \underbrace{(1-y\_data)\log(1-h)}_{B}\}$$

4. 코딩으로 확인하는 로지스틱 회귀

```
# 시그모이드 식에 대입하여 return

def sigmoid(x):

        return 1/(1+np.e**(-x))

# 경사 하강법을 이용해 a 와 b 의 최적값 구하기
```

```
for i in range(2001):

          for x_data, y_data in data:

          a_diff = x_data*(sigmoid(a*x_data+b) - y_data)

          b_diff = sigmoid(a*x_data+b) - y_data

          a = a - lr*a_diff

          b = b - lr*b_diff
```

5. 로지스틱 회귀에서 퍼셉트론으로

정리해보면 입력 값을 통해 출력값을 구하는 함수 y 는 다음과 같다.

이때 x1,x2 는 입력값, y 는 출력값이다.

$$y = a_1x_1 + a_2x_2 + b$$