출처가 명시되지 않은 모든 자료(이미지 등)는 조성현 강사님 블로그 및 강의 자료 기반.

<< 머신러닝-분류 >>

# [Logistic Regression]

로지스틱 회귀는 이름은 '회귀' 모형이지만, 분류에서 사용되는 대표적인 회귀 알고리즘이다. 샘플이 특정 클래스에 속할 확률을 추정함으로써 해당 클래스에 속하는지 혹은 속하지 않는지 예측한다.

## 1. 배경

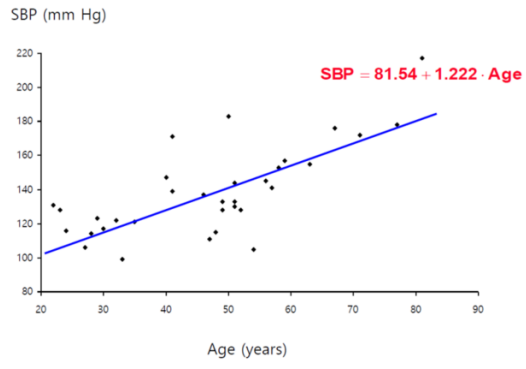
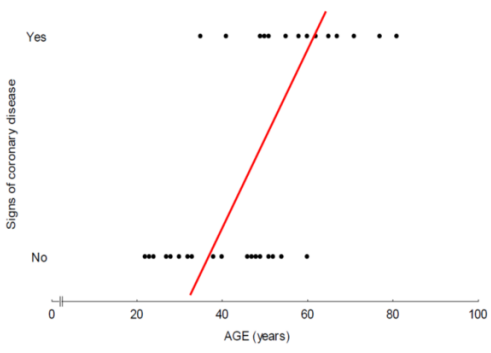
'회귀' 알고리즘인데도 분류에 사용된다고 했다. 그렇다면 선형 회귀 모형은 분류 작업에 사용할 수 없는 것일까?

가능은 하다. 그러나 어려움이 있다.

선형 회귀 모형은 결과값이 연속형 변수일 때 사용하는 알고리즘이다. 범주형 변수를 예측해야 하는 분류 작업의 경우, 타겟 변수에서의 숫자(혹은 라벨)이 아무런 의미를 지니지 않는다. 따라서 선형 회귀 모형으로 모델을 구축하면 적절하지 않다.

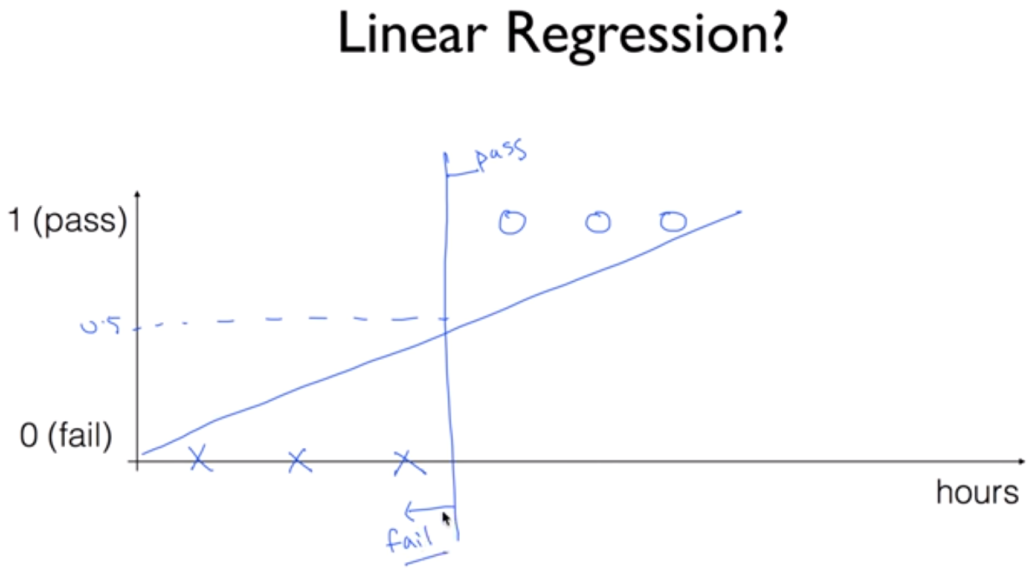
> *참고* : 김성범 교수님 로지스틱 회귀 모델 강의
>
> 선형 회귀 모형에서의 최소제곱법을 이용해 회귀 계수 값을 추정하려면 만족해야 하는 여러 가정이 있다. Y값이 연속형이 아니라 범주형으로 바뀌는 경우, 해당 가정들을 만족할 수가 없게 된다.

범주형 모델을 사용해 회귀 모델을 구축할 수는 있을 것이다. 그런데 그 그래프의 모양이 우스꽝스러운 모양이 될 것이다.(출처: ratsgo's blog)

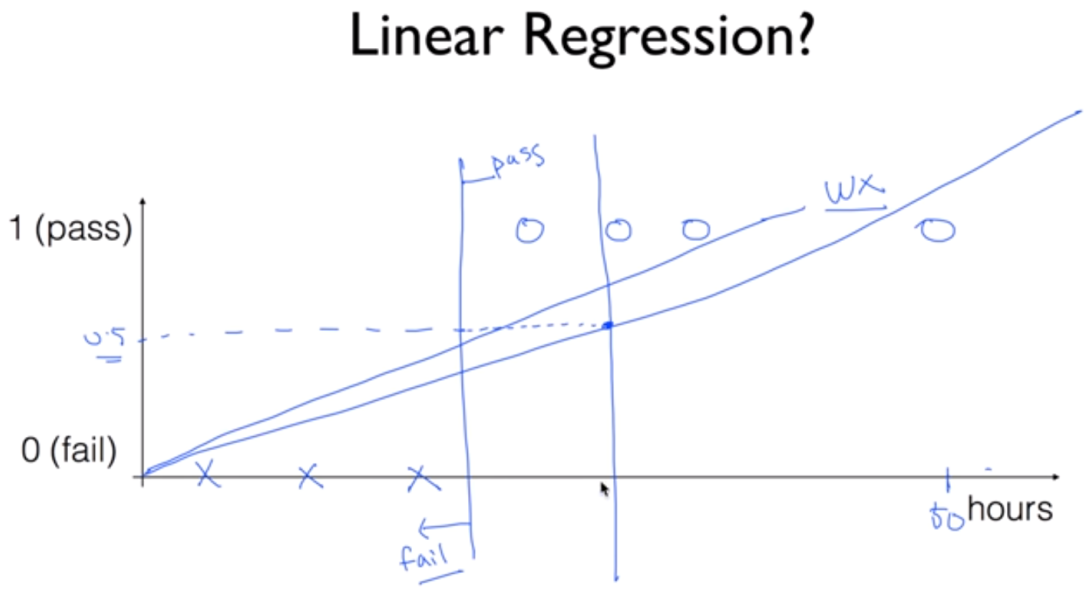
| 연속형 변수 | 범주형 변수 |
| --- | --- |
|  |  |

또 다른 문제도 있다. 입력 데이터 X가 가지는 확률값을 바탕으로 분류 문제를 수행한다고 해 보자.(출처: 김성훈 교수님 모두를 위한 딥러닝 강좌 시즌 1)

공부한 시간에 따라 시험에 합격할 확률을 예측하는 상황이다. 가지고 있는 학습 데이터를 바탕으로 다음과 같은 회귀선을 그었다고 하자.




만약 어떤 학생이 혼자서 굉장히 많은 시간(예컨대 50시간)을 공부하고 시험에 합격했다고 하자. 만약 선형 회귀로서 점수를 예측한다면 위로 올라갈 수 있겠지만, 이 문제에서는 분류 문제 특성 상 50시간을 공부해서 합격했다고 하더라도 라벨이 1이 되어 버린다.



이제 이 상태에서 선형 회귀 모델을 학습시키면, 회귀선이 위와 같이 변화하게 된다. 그러면 분류의 기준이 되는 선이 달라질 수 있다.

> *참고*
>
> 이 문제에 대해 이전 문쌤의 강의에서 코드를 통해 예측이 달라지는 것을 확인한 적도 있으니, 복습하자!

그 외에도 선형 회귀의 예측값이 0보다 매우 작거나 1보다 매우 큰 값이 나올 수 있다는 점, 분류의 기준이 되는 threshold 값이 계속해서 변할 수 있다는 점 등의 문제가 있다.

# 2. 이진 분류

## 기본 원리

위와 같은 배경에서 로지스틱 회귀 알고리즘이 등장하게 되었다. 그렇다면 이 알고리즘은 실제로 어떻게 동작하여 분류 작업을 수행하게 되는 것일까?

0과 1의 두 클래스를 분류하는 이진 분류 문제를 생각해 보자.

로지스틱 회귀는 다음과 같은 Sigmoid 함수를 이용하여 입력 데이터 X의 로짓 값을 출력한다.

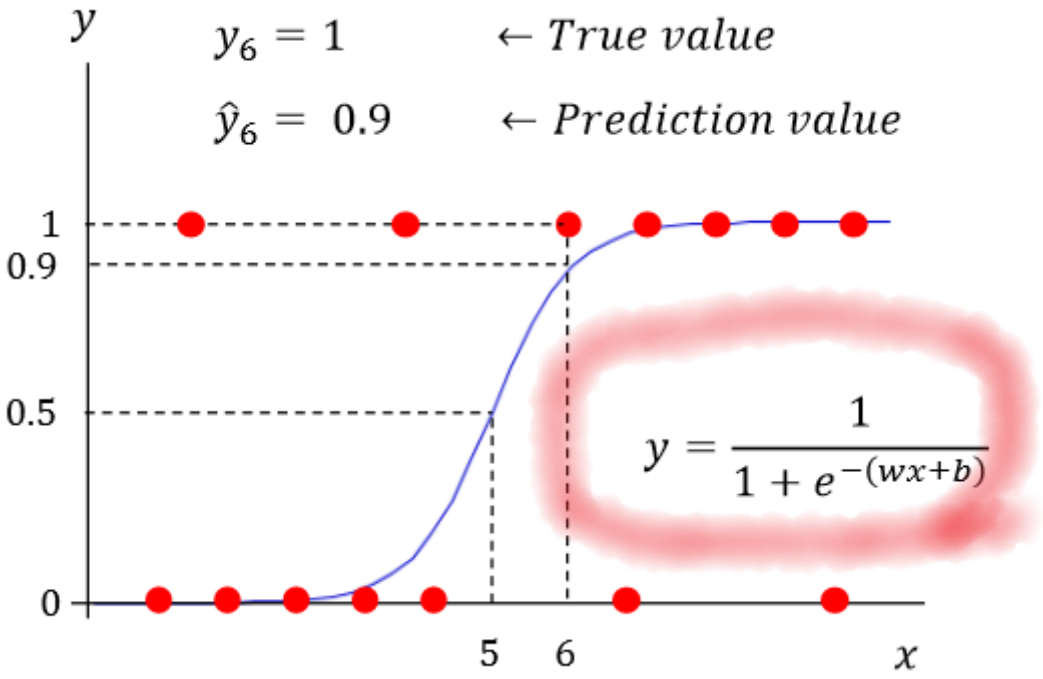
$$\sigma(t) = \frac{1}{1 + exp(-t)}$$

위의 sigmoid 함수에서 input 값으로 사용되는 `t`는 linear regression model의 output값과 같다. 다시 말하면, 다음과 같은 선형 회귀식

$$H(x) = w \cdot x + b$$

으로부터 도출된 `H(x)` 값, 즉, `y`값을 sigmoid 함수에 넣으면 된다.

이 sigmoid 함수는 위 그림과 같이 S자 형태를 띠게 된다. 그리고 결과적으로 선형 회귀식에 의해 도출된 결과 값은 0과 1 사이의 값으로 바뀌게 된다.



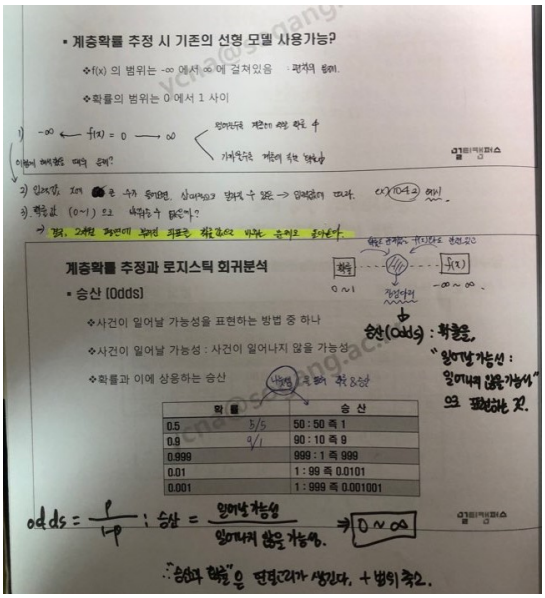
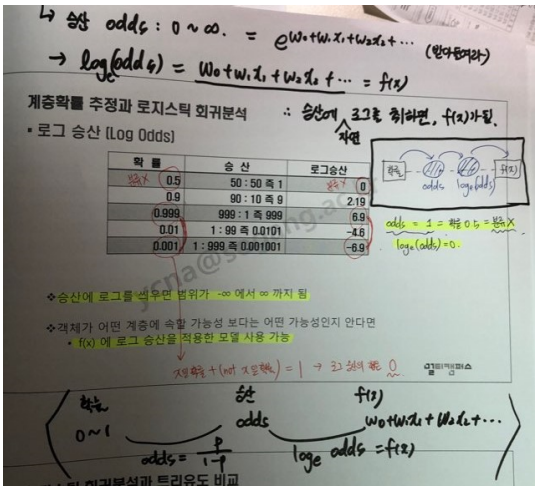

동그라미 친 부분이 바로 sigmoid 함수 식이다.

결과적으로 선형 회귀식에 의해 도출된 결과 값은 0과 1사이의 값으로 바뀌게 된다. 이렇게 도출된 결과 값을 **로짓**이라고 부르며, 로지스틱 회귀 모형은 다음의 원리에 따라 해당 로짓 값이 0.5 이상이면 양성 클래스(혹은 1), 미만이면 음성 클래스(혹은 0)이라고 예측하는 것이다.

$$\hat{y} = \begin{matrix} 0, & \hat{p} < 0.5 \\ 1, & \hat{p} \geq 0.5 \end{matrix}$$

## 자세한 과정

> 참고: 멀티캠퍼스 나용찬 교수님 빌드업 특강(20200127~20200128) 자료

| 승산 | 로그 승산 |
| --- | --- |
|  |  |



1. 목적 : 선형회귀식 `y = ax + b`의 결과값을 확률로 변환해야 한다.
    - 선형회귀식의 결과값 y는 -inf ~ +inf의 값을 갖는다.
    - 따라서 이 값이 0과 1 사이의 값인 확률로 변환해야 한다.

2. 선형 회귀식에서 `y`를 `P`로 바꾼다. `P = ax + b`가 된다.
    - 그러나 여전히 양변의 값의 범위가 다르다.
    - 수학적으로 가정만 했을 뿐, 틀린 수식이다.

3. `P` 자리에 odds를 대입한다.

$$\frac{P}{1-P} = ax + b$$

> *Odds(승산)*
>
> 사건이 일어날 가능성을 표현하는 방법 중 하나로, 사건이 일어날 가능성 대비 사건이 일어날 가능성으로 정의된다.
>
> $$odds = \frac{P}{1-P}$$
>
> 확률을 승산으로 표현하면, 사건이 일어날 가능성을 0에서 양의 무한대 사이의 값으로 표현할 수 있게 된다.

    - 좌변의 값이 0 에서 +inf 사이의 값이 된다.
    - 여전히 양변의 값의 범위가 다르다.

4. 좌변의 승산에 로그를 씌운다.

- 로그 승산은 -inf ~ +inf의 값이다.
- 드디어 양변의 값의 범위가 일치한다.

$$log(odds) = log(\frac{1}{1-P}) = ax + b$$

5. 이제 양변을 P에 대해 정리하자.
   - 우리가 알고 싶은 건 `P`이다. 로지스틱 회귀분석을 통해 입력 데이터 `x`가 갖는 확률 `P`를 구하고 싶어 했음을 잊지 말자.
   - 4.의 식에 `e`를 씌우고 정리하자.

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

   - 역수를 취하자.

$$\frac{1-P}{P} = \frac{1}{e^{ax+b}}$$
$$\frac{1}{P} - 1 = \frac{1}{e^{ax+b}}$$

   - 1을 더하자.

$$\frac{1}{P} = \frac{1+e^{ax+b}}{e^{ax+b}}$$

   - 역수를 취하자.

$$P = \frac{e^{ax+b}}{1+e^{ax+b}}$$

이제 우리가 알고 있는 sigmoid 함수를 도출했다. 위에서 도출한 식의 분자, 분모에

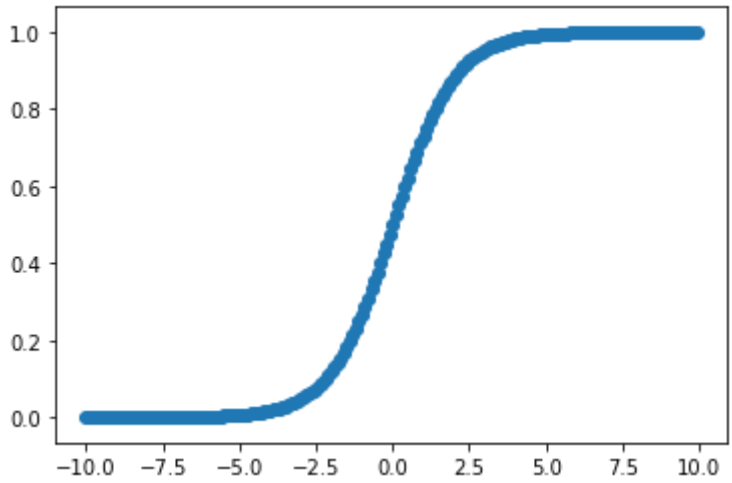
$$e^{-(ax+b)}$$

만 곱해주면 원래 알고 있는 Sigmoid 식이 된다.

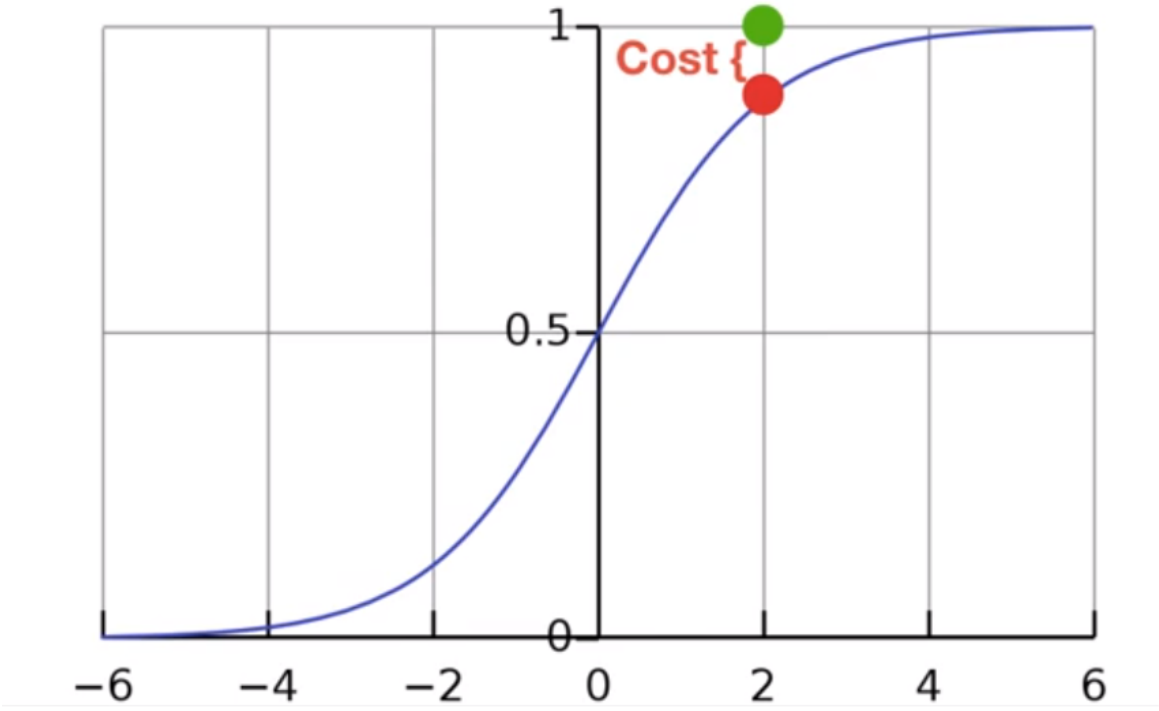
Python을 이용해 위의 sigmoid함수를 그려 보면 S자 형태의 곡선이 나오는 것을 알 수 있다.

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(-10, 10, 0.1)
y = 1 / (1 + np.exp(-x))
plt.scatter(x,y)
```




## 손실 함수




로지스틱 회귀에서도 역시나 손실함수를 최소화하는 것이 중요하다.

출처: https://www.youtube.com/watch?v=zASrGSHoqL4

로지스틱 회귀를 이용한 분류 문제에서는 log loss, 즉, cross entropy를 손실 함수로 사용한다.

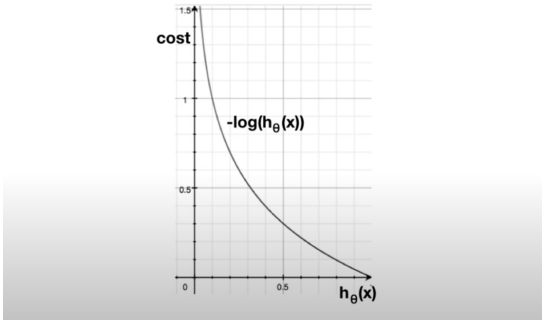
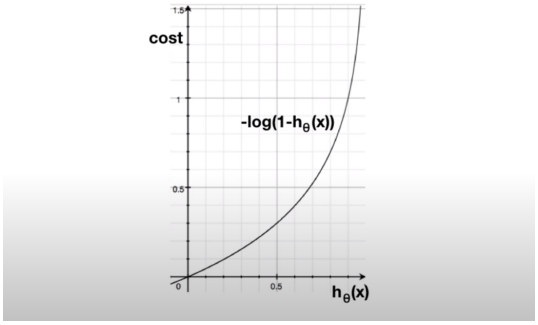
이전 시간에도 배운 크로스 엔트로피 식을 이진 분류 문제에 맞게 다시 써 보자. 클래스가 2개이기 때문에, 이진 분류에서의 크로스 엔트로피 식은 다음과 같다.

$$CE = -\Sigma_{k=1}^{n}\Sigma_{i=1}^{2} y_{i,k} \cdot log\hat{y}$$

위 식을 풀어 쓰면 다음과 같아진다.

$$-\frac{1}{n}\sum_{i=1}^{n}[y_i\ log(y_i) + (1-y_i)\ log(1-\hat{y}_i)]$$

이제 위의 식을 해석해 보자. 실제 레이블 `y`가 1일 때, 식의 뒷 부분이 남고, `y`가 0이 될 때, 식의 앞 부분이 남는다. 이를 그림으로 나타내면 다음과 같다.

| y = 1 | y = 0 |
| --- | --- |
|  |  |



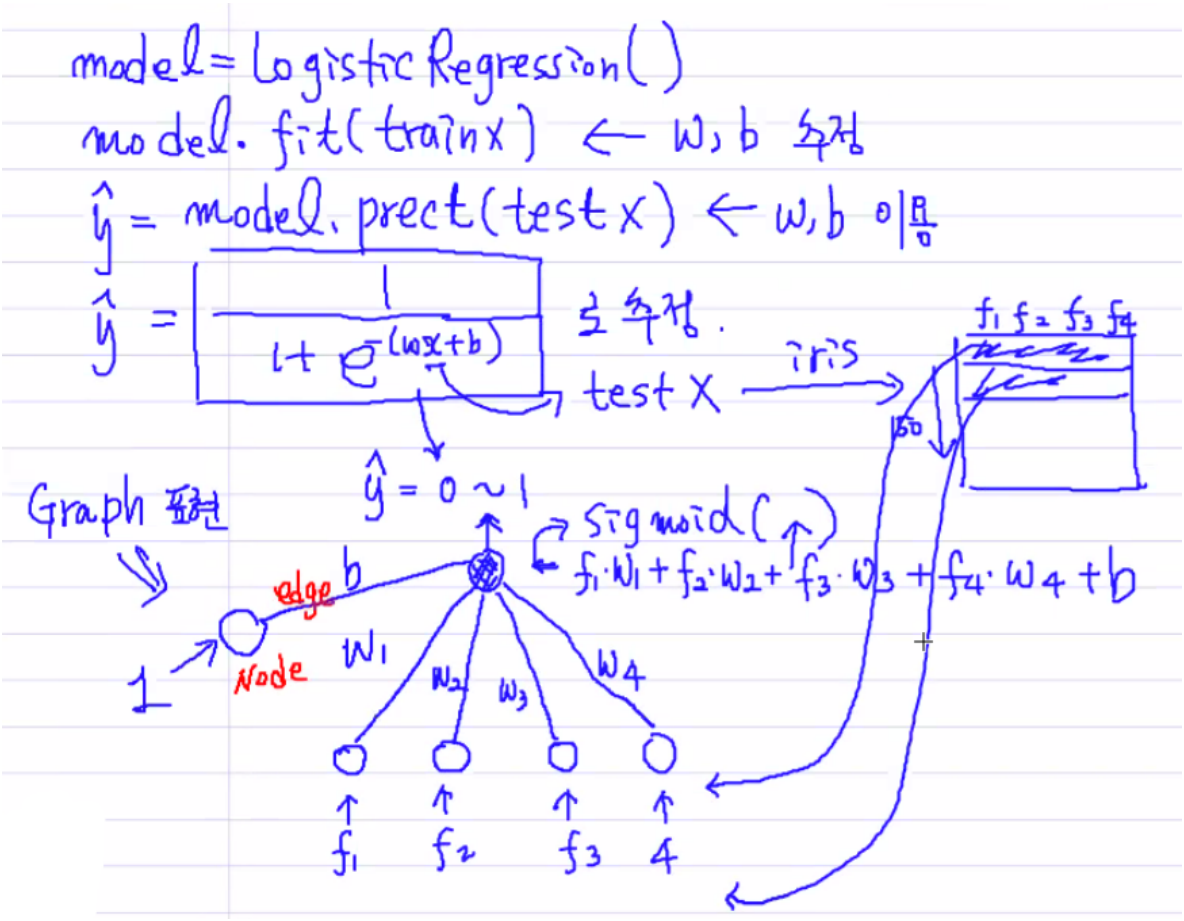
출처: https://www.youtube.com/watch?v=zASrGSHoqL4

각각의 경우, 모두 클래스를 틀리게 예측할 경우 cost 값이 매우 커지게 된다.

## 그래프 표현

로지스틱 회귀 모델도 선형 회귀 모델과 마찬가지로 입력 데이터의 가중치 합을 계산한다. 선형 회귀 모델과 다른 것은 계산한 가중치 값을 그대로 출력하지 않고, 위에서 살펴 본 **Sigmoid** 함수를 활용해 **로짓** 값을 출력한다는 것이다.

이를 그래프로 표현하면 다음과 같다.




하나의 feature 당 하나의 가중치가 설정되고, 이들의 가중치 합이 Sigmoid 함수를 통과하는 구조다.

## 3. 다중 분류

## 4. 규제

이진분류

소프트맥스 회귀