지난 시간: Laplace Smoothing

Question

조정용 · 6일 전

나이브 베이즈에서 조건부확률에 대한 적용이 와 닿지가 않았는데, 영화 장면에 대한 장르를 기준으로 설명해주셔서 멀티노미날 나이브 베이즈에 대해서 이해가 쉬웠습니다. 그리고 각 단어의 빈도수를 구하고 영화리뷰중 지정한 단어가 나타나는 확률에 대해서 계산할때, P(fats|comedy)*p(furious|comedy)*P(fun|comedy)로 계산시 comedy에는 fun이라는 word가 없어서 마지막에 P(words|comedy)값이 0이 되었습니다. 이러한 문제를 방지하기 위해서raplace smoothing을 사용하여 일반적으로0~1의 사이값을 더하는 추가적인 단계를 넣는다고 하셨는데, 이 경우 0이 안되기만 하는 최소값인 10^-4같은 값을 사용해도 될텐데 1을 대표적으로 사용하는 이유가 있을까요?

Answer

Additive smoothing

From Wikipedia, the free encyclopedia

In statistics, **additive smoothing**, also called **Laplace smoothing**[1] (not to be confused with Laplacian smoothing as used in image processing), or **Lidstone smoothing**, is a technique used to smooth categorical data. Given an observation $\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle$ from a multinomial distribution with N trials, a "smoothed" version of the data gives the estimator:

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

where the "pseudocount" $\alpha > 0$ is a smoothing parameter. $\alpha = 0$ corresponds to no smoothing. (This parameter is explained in § Pseudocount below.) Additive smoothing is a type of shrinkage estimator, as the resulting estimate will be between the empirical probability (relative frequency) x_i/N, and the uniform probability $1/d$. Invoking Laplace's rule of succession, some authors have argued[*citation needed*] that α should be 1 (in which case the term **add-one smoothing**[2][3] is also used)[*further explanation needed*], though in practice a smaller value is typically chosen.

(Source: https://en.wikipedia.org/wiki/Additive_smoothing)

지난 시간: Laplace Smoothing

Question

조정용 · 6일 전

나이브 베이즈에서 조건부확률에 대한 적용이 와 닿지가 않았는데, 영화 장면에 대한 장르를 기준으로 설명해주셔서 멀티노미날 나이브 베이즈에 대해서 이해가 쉬웠습니다. 그리고 각 단어의 빈도수를 구하고 영화리뷰중 지정한 단어가 나타나는 확률에 대해서 계산할때, P(fats|comedy)*p(furious|comedy)*P(fun|comedy)로 계산시 comedy에는 fun이라는 word가 없어서 마지막에 P(words|comedy)값이 0이 되었습니다. 이러한 문제를 방지하기 위해서raplace smoothing을 사용하여 일반적으로0~1의 사이값을 더하는 추가적인 단계를 넣는다고 하셨는데, 이 경우 0이 안되기만 하는 최소값인 10^-4같은 값을 사용해도 될텐데 1을 대표적으로 사용하는 이유가 있을까요?

Answer

Additive smoothing

From Wikipedia, the free encyclopedia

In statistics, **additive smoothing**, also called **Laplace smoothing**[1] (not to be confused with Laplacian smoothing as used in image processing), or **Lidstone smoothing**, is a technique used to smooth categorical data. Given an observation $\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle$ from a multinomial distribution with N trials, a "smoothed" version of the data gives the estimator:

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

where the "pseudocount" $\alpha > 0$ is a smoothing parameter. $\alpha = 0$ corresponds to no smoothing. (This parameter is explained in § Pseudocount below.) Additive smoothing is a type of shrinkage estimator, as the resulting estimate will be between the empirical probability (relative frequency) x_i/N, and the uniform probability $1/d$. Invoking Laplace's rule of succession, some authors have argued[*citation needed*] that α should be 1 (in which case the term **add-one smoothing**[2][3] is also used)[*further explanation needed*], though in practice a smaller value is typically chosen.

(Source: https://en.wikipedia.org/wiki/Additive_smoothing)

지난 시간: Laplace Smoothing

Question

조정용 · 6일 전

나이브 베이즈에서 조건부확률에 대한 적용이 와 닿지가 않았는데, 영화 장면에 대한 장르를 기준으로 설명해주셔서 멀티노미날 나이브 베이즈에 대해서 이해가 쉬웠습니다. 그리고 각 단어의 빈도수를 구하고 영화리뷰중 지정한 단어가 나타나는 확률에 대해서 계산할때, P(fats|comedy)*p(furious|comedy)*P(fun|comedy)로 계산시 comedy에는 fun이라는 word가 없어서 마지막에 P(words|comedy)값이 0이 되었습니다. 이러한 문제를 방지하기 위해서raplace smoothing을 사용하여 일반적으로0~1의 사이값을 더하는 추가적인 단계를 넣는다고 하셨는데, 이 경우 0이 안되기만 하는 최소값인 10^-4같은 값을 사용해도 될텐데 1을 대표적으로 사용하는 이유가 있을까요?

Answer

[1] 아직 훈련하지 못한 케이스가 하나쯤은 있다고 보면 1.
[2] 너무 작을 경우, 가능도 결과가 0에 가까운 값이 될 수도 있다는 의미이니 주의.

Additive smoothing

From Wikipedia, the free encyclopedia

In statistics, **additive smoothing**, also called **Laplace smoothing**[1] (not to be confused with Laplacian smoothing as used in image processing), or **Lidstone smoothing**, is a technique used to smooth categorical data. Given an observation $\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle$ from a multinomial distribution with N trials, a "smoothed" version of the data gives the estimator:

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

where the "pseudocount" $\alpha > 0$ is a smoothing parameter. $\alpha = 0$ corresponds to no smoothing. (This parameter is explained in § Pseudocount below.) Additive smoothing is a type of shrinkage estimator, as the resulting estimate will be between the empirical probability (relative frequency) x_i/N, and the uniform probability $1/d$. Invoking Laplace's rule of succession, some authors have argued[*citation needed*] that α should be 1 (in which case the term **add-one smoothing**[2][3] is also used)[*further explanation needed*], though in practice a smaller value is typically chosen.

(Source: https://en.wikipedia.org/wiki/Additive_smoothing)

Naive Bayes의 실제 코딩은?

Machine learning

Linear Regression

머신러닝의 학습 방법들

- **Probability theory-based learning**
- **Gradient descent-based learning**
- **Information theory-based learning**
- **Distance similarity-based learning**

경사 하강법

Gradient descent based learning

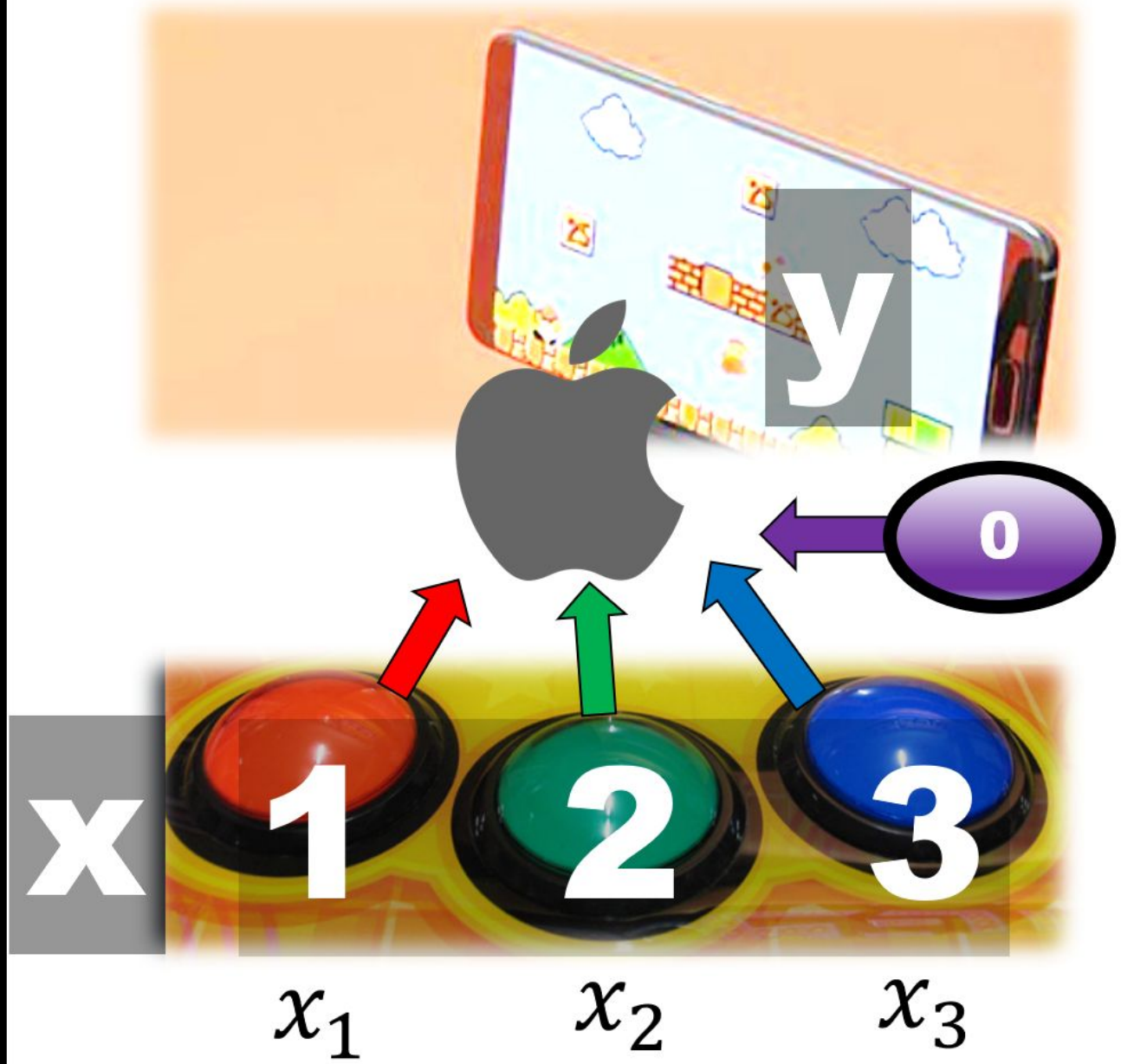
- 실제 값과 학습된 모델 예측치의 오차를 최소화

- 학습을 통해 모델의 최적 파라미터 찾기가 목적

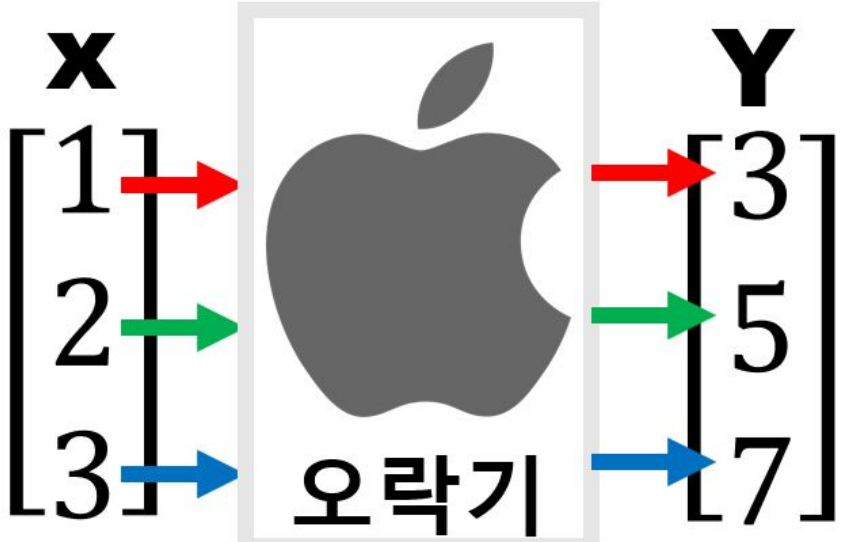
→ 훈련할 때 찾은 최적 파라미터로
진짜 테스트셋에서 분류해본다!

Regression 왜 회귀라 불리는지,

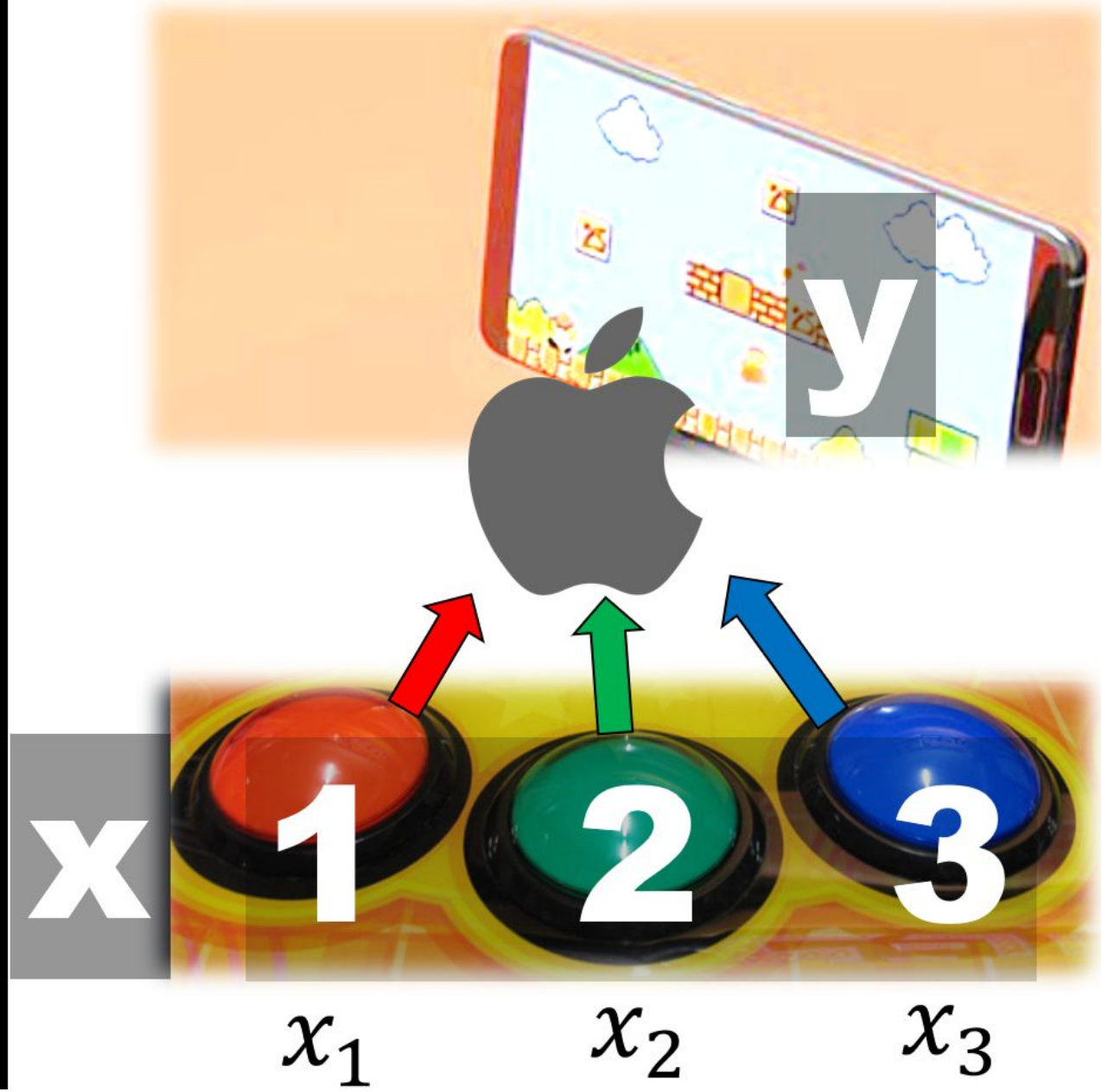
목표 : 오락기 구조를 아는 것
이유 : 새버튼 '0'의 결과 예측!



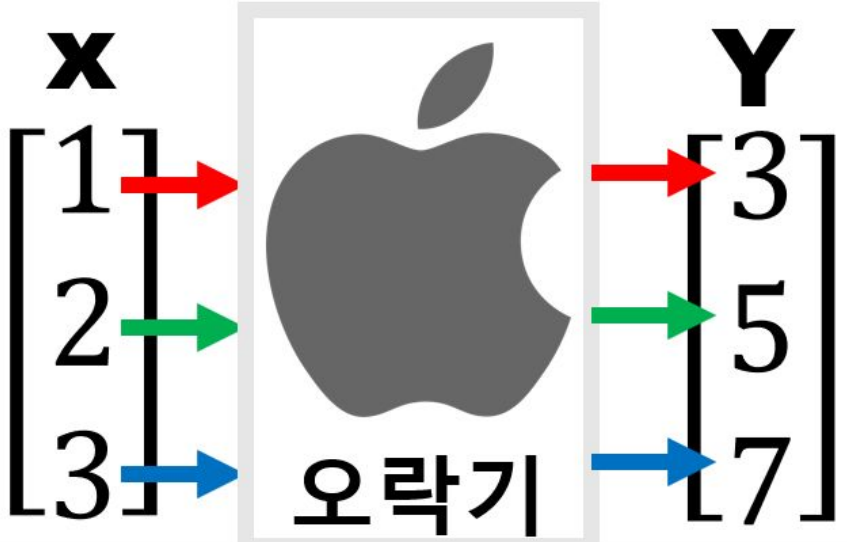
목표 : 오락기 구조를 아는 것
이유 : 새버튼 '0'의 결과 예측!



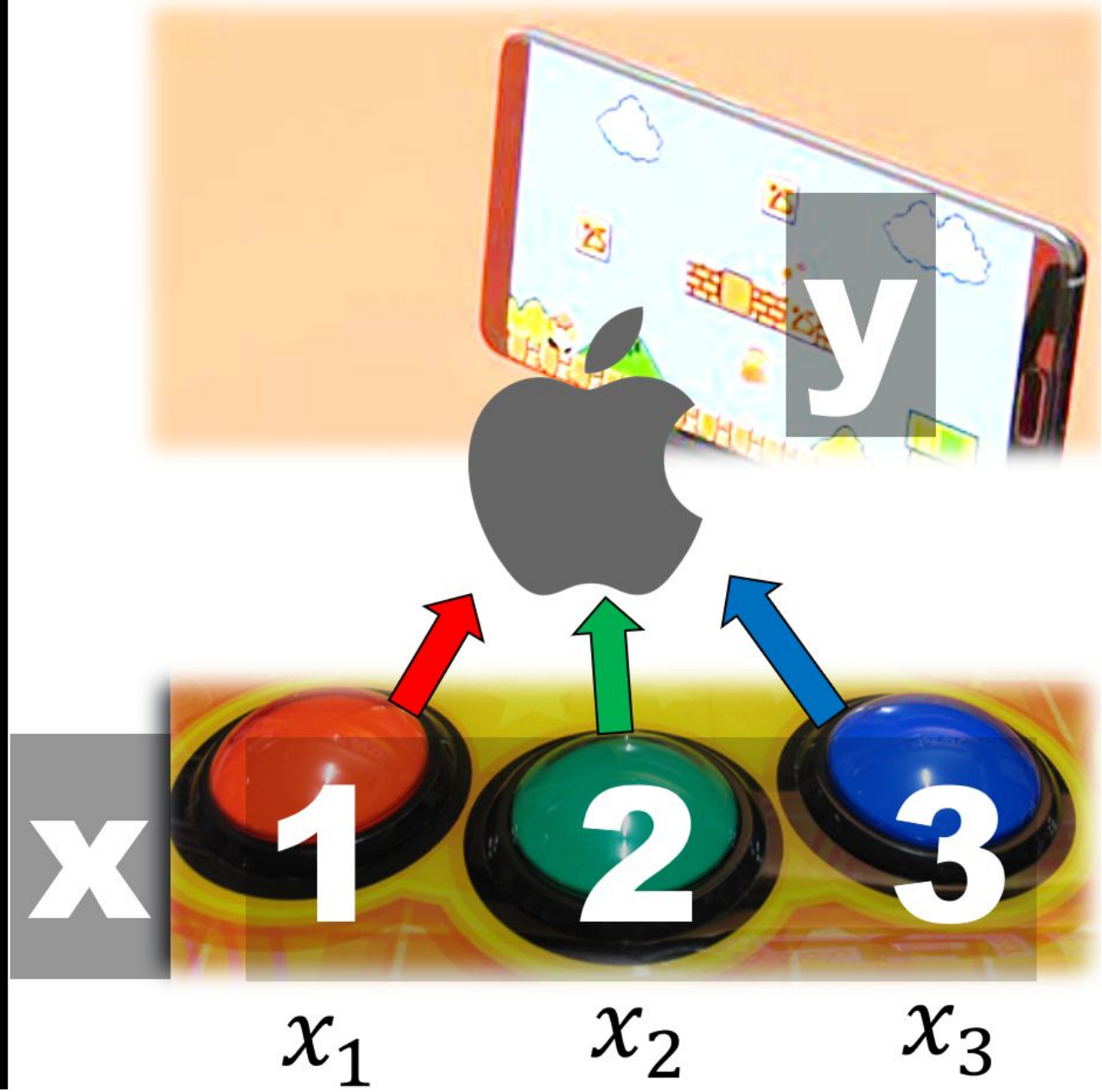
실제, $y_i = @x_i + \$$



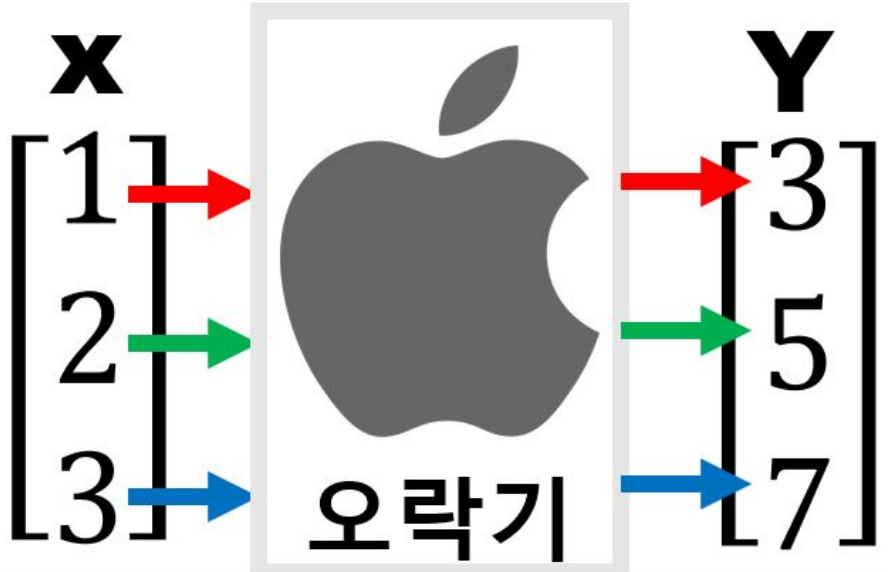
목표 : 오락기 구조를 아는 것
이유 : 새버튼 '0'의 결과 예측!



실제, $y_i = @x_i + \$$



목표 : 오락기 구조를 아는 것
이유 : 새버튼 '0'의 결과 예측!



실제, $y_i = @x_i + \$$

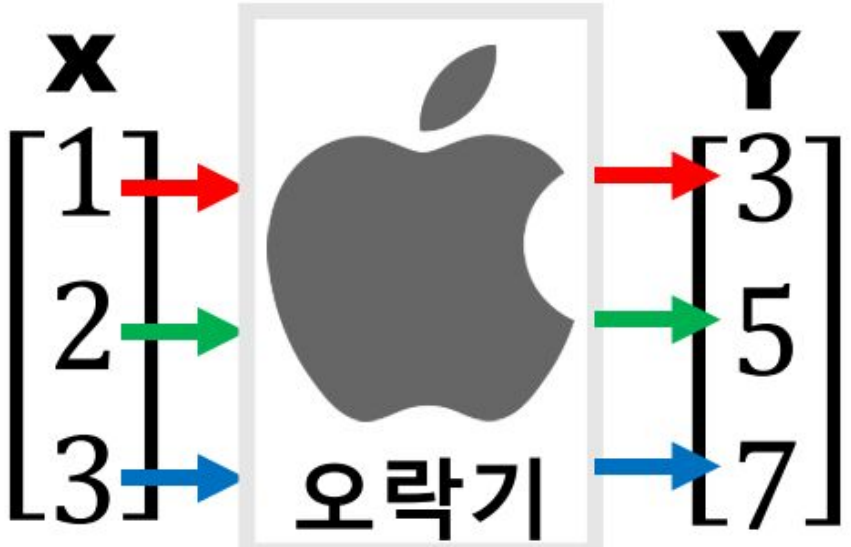
가설, $H(w,b) = Wx_i + b$

$i = 1,2,3$까지는 경험
$i = 4$는 **새로운 입력**

가설, $Hypotheis(W,b)$
$= Weight \cdot x_i + bias$

목표 : 오락기 구조를 아는 것

이유 : 새버튼 '0'의 결과 예측!



실제, $y_i = @x_i + \$$

가설, $H(w,b) = Wx_i + b$

$i = 1,2,3$까지는 경험

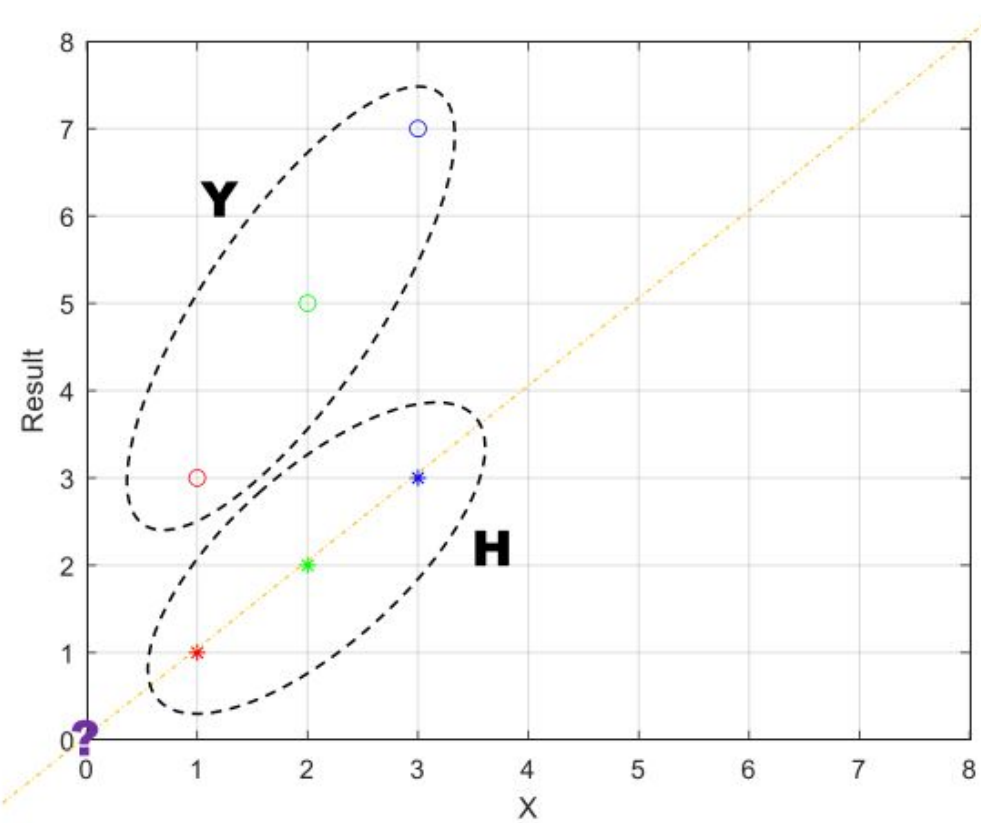
$i = 4$는 **새로운 입력**

가설, $Hypotheis(W,b)$

$= Weight \cdot x_i + bias$

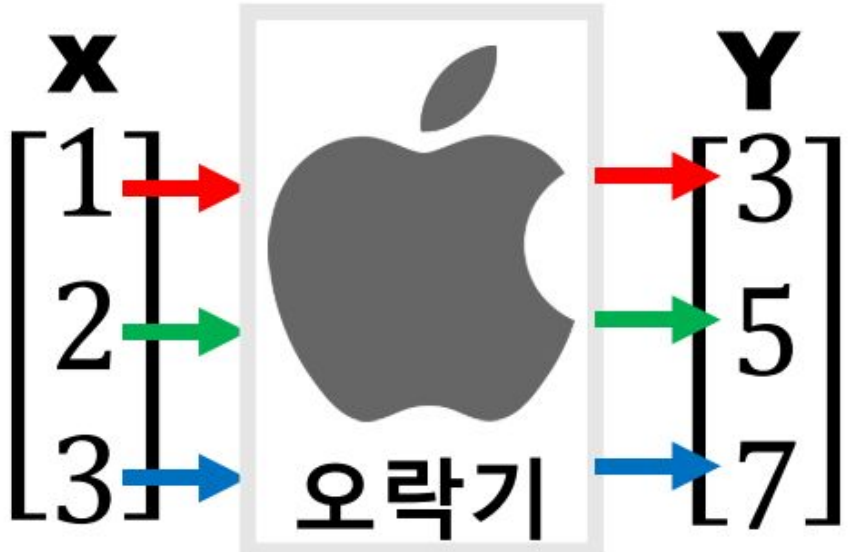
초기값 $W = 1, b = 0$으로 가정하면,

$$H(1,0) = 1 \cdot X + 0$$



목표 : 오락기 구조를 아는 것

이유 : 새버튼 '0'의 결과 예측!



실제, $y_i = @x_i + \$$

가설, $H(w,b) = Wx_i + b$

$i = 1,2,3$까지는 경험

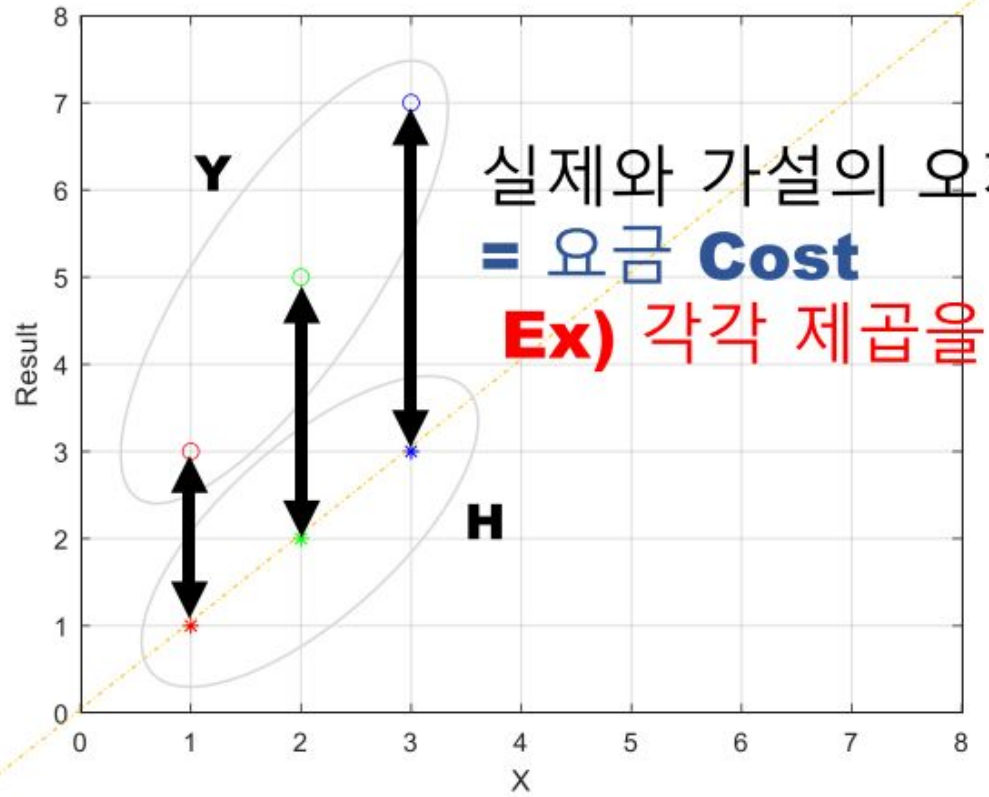
$i = 4$는 **새로운 입력**

가설, $Hypotheis(W,b)$

$= Weight \cdot x_i + bias$

초기값 $W = 1, b = 0$으로 가정하면,

$$H(1,0) = 1 \cdot X + 0$$



실제와 가설의 오차(전체)

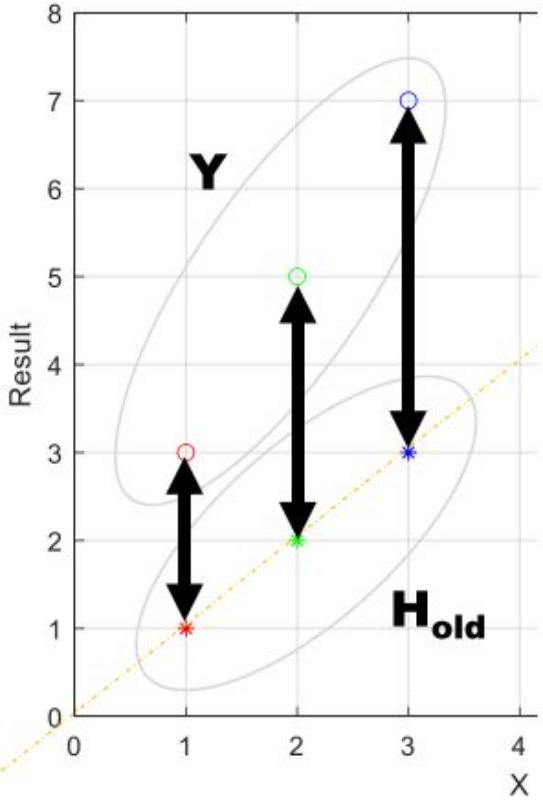
= 요금 **Cost**

Ex) 각각 제곱을 평균

초기값 $W = 1, b = 0$으로 가정하면,

$$H(1,0) = 1 \cdot X + 0$$

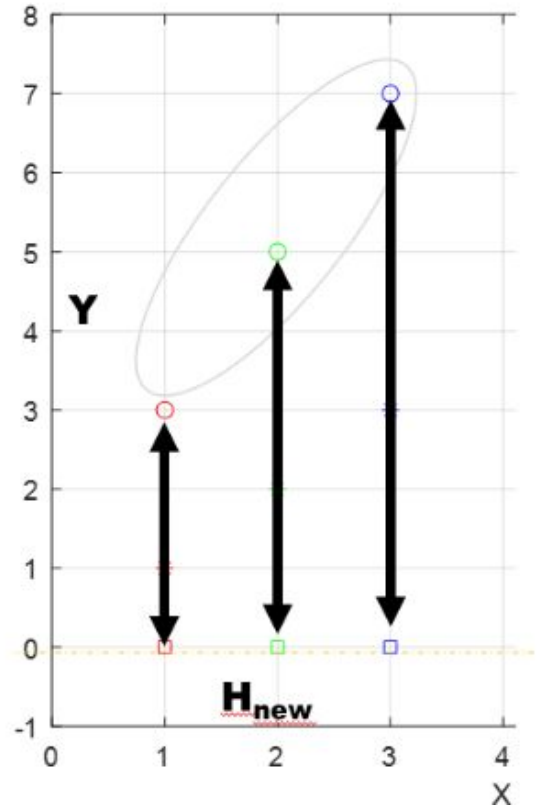
$$cost(1,0) = \frac{2^2+3^2+4^2}{3} = \frac{29}{3}$$



$W = 0, b = 0$으로 바꾸면

$$H(0,0) = 0 \cdot X + 0$$

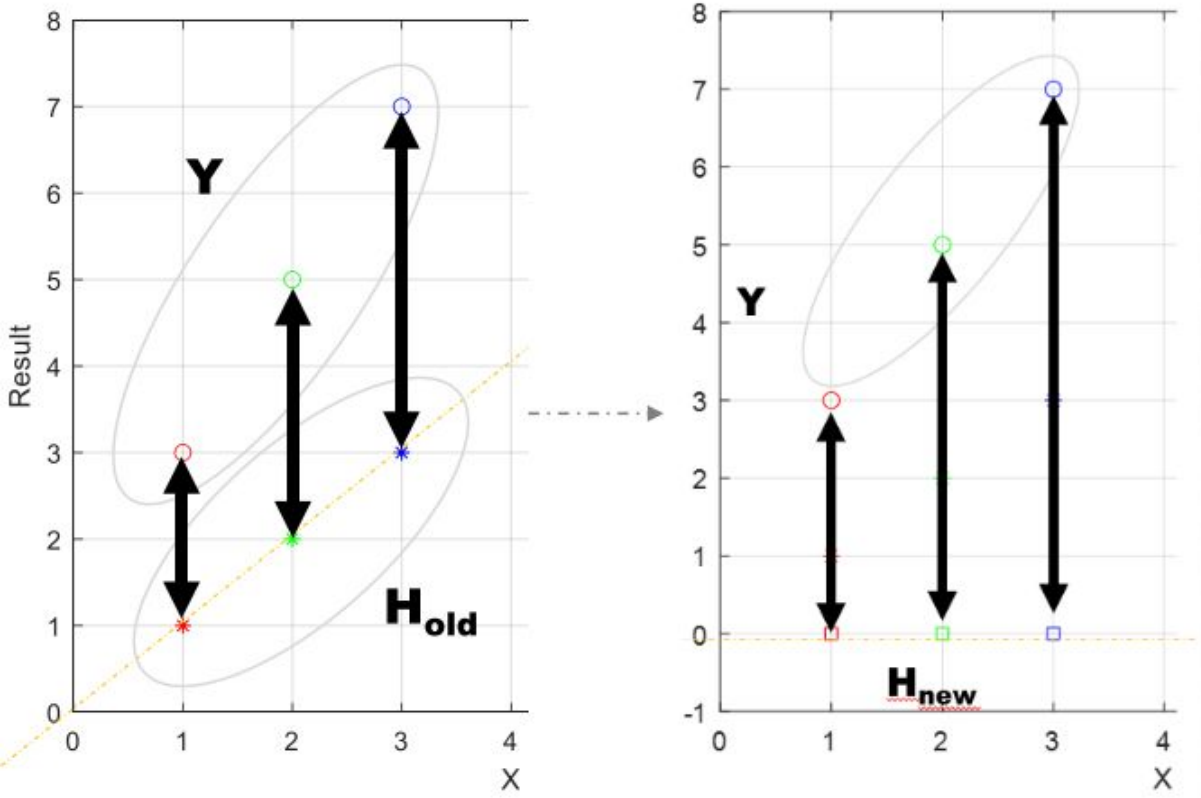
$$cost(0,0) = \frac{3^2+5^2+7^2}{3} = \frac{83}{3}$$ **(>2.8배)**



초기값 $W = 1, b = 0$으로 가정하면, → $W = 0, b = 0$으로 바꾸면

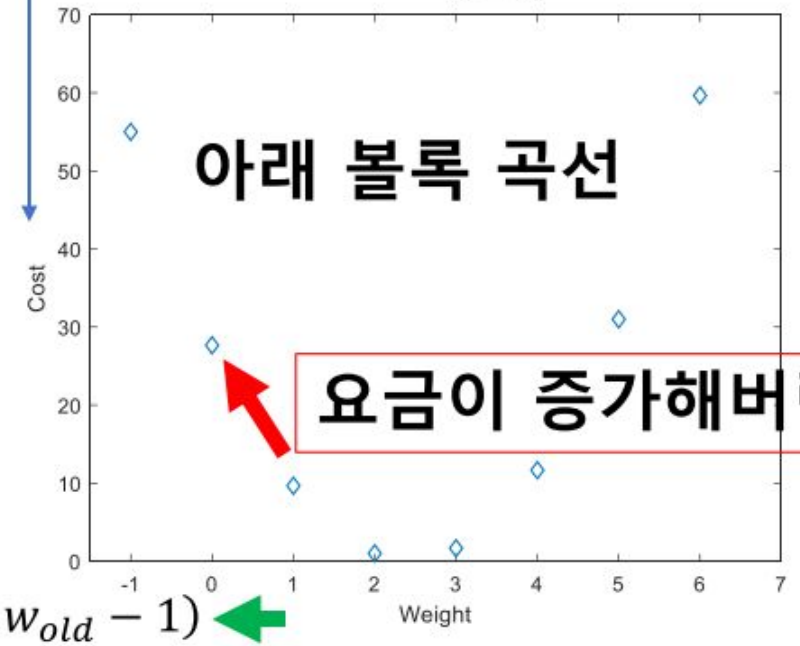
$$H(1,0) = 1 \cdot X + 0 \qquad H(0,0) = 0 \cdot X + 0$$

$$cost(1,0) = \frac{2^2+3^2+4^2}{3} = \frac{29}{3} \rightarrow cost(0,0) = \frac{3^2+5^2+7^2}{3} = \frac{83}{3}$$ **(>2.8배)**



일반화

$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}\{(Wx_i+b) - y_i\}^2$$

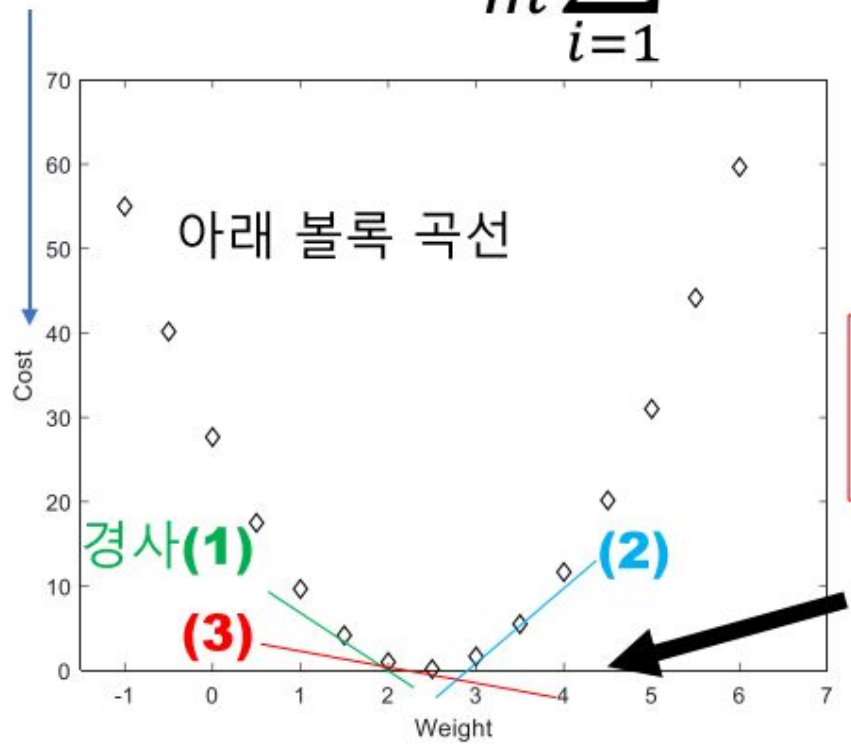


요금이 증가해버린 나쁜 예

$(w_{new} = w_{old} - 1)$

임무: 정확한 모델링
= 비용 최소화.

$$\min cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}\{(Wx_i+b) - y_i\}^2$$



경사 **Gradient**
하강법 **Descent**

경사가 가장 작을 때
비용이 최소화되기 때문

경사 **(1)** 음수일 때, 오른쪽으로 **(W**를 증가**)**
(2) 양수일 때, 왼쪽으로 **(W**를 감소**).**
(3) 가장 작은 경사일 가능성
단, **W**만의 함수이므로, **b**도 고려하면 **3**차원으로 표현

정보 **[Input Data]** → 가설 **H(W,b)** → 비용 **Cost** 경사**[편미분]**

Loop

α만큼 W, b 업데이트 ← 최소**?**

No

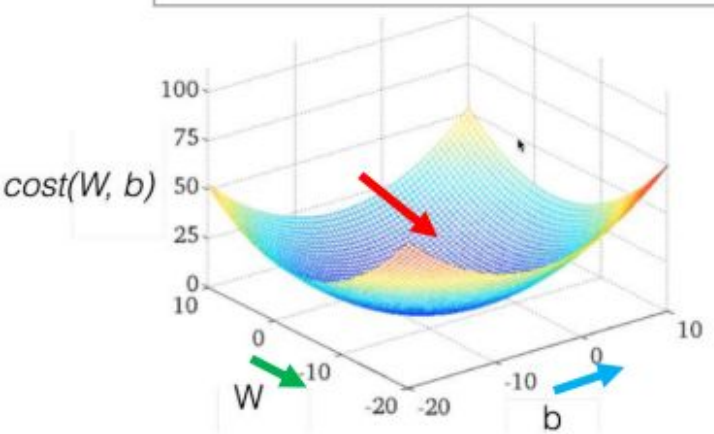
Yes

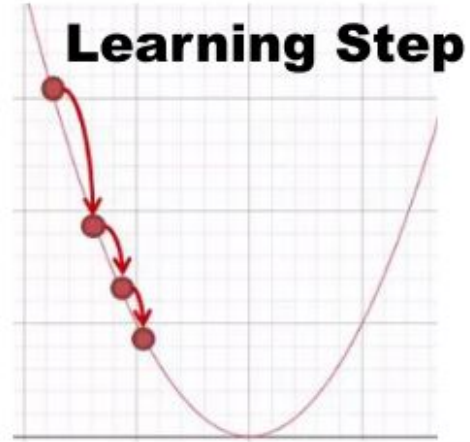
$$W := W - \alpha \frac{\partial cost(W,b)}{\partial w} \quad W \to 2$$

$$b := b - \alpha \frac{\partial cost(W,b)}{\partial b} \quad b \to 1$$

오락기 모델링 완료

$$H(2,1) = 2x_i + 1$$





Step 반복수(Iteration)
: Epoch

정보 **[Input Data]** → 가설 **H(W,b)** → 비용 **Cost** 경사**[편미분]**

Loop

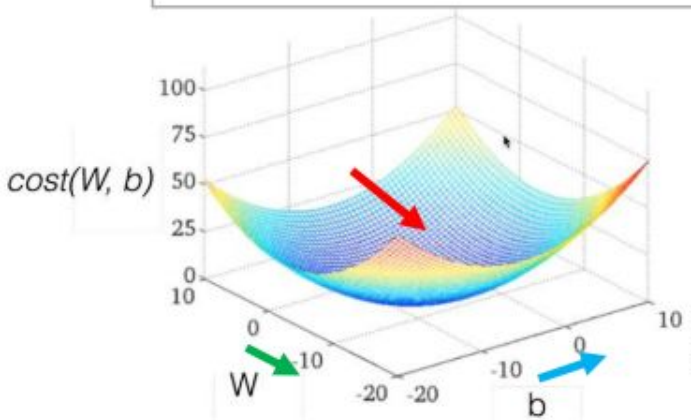
α만큼 W, b 업데이트 ← 최소? **No**

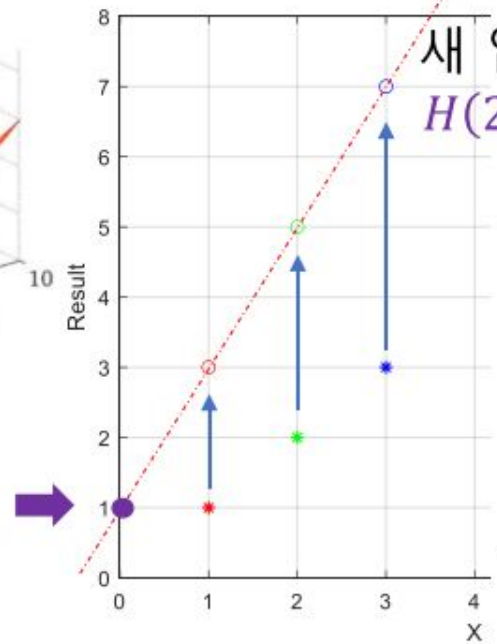
Yes → 오락기 모델링 완료

$$W := W - \alpha \frac{\partial cost(W,b)}{\partial w} \quad W \to 2$$

$$b := b - \alpha \frac{\partial cost(W,b)}{\partial b} \quad b \to 1$$

$$H(2,1) = 2x_i + 1$$





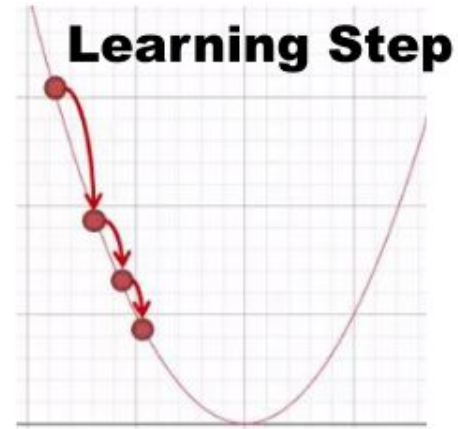
새 입력 **'0'**가 들어오면

$H(2,1) = 1$

Regression

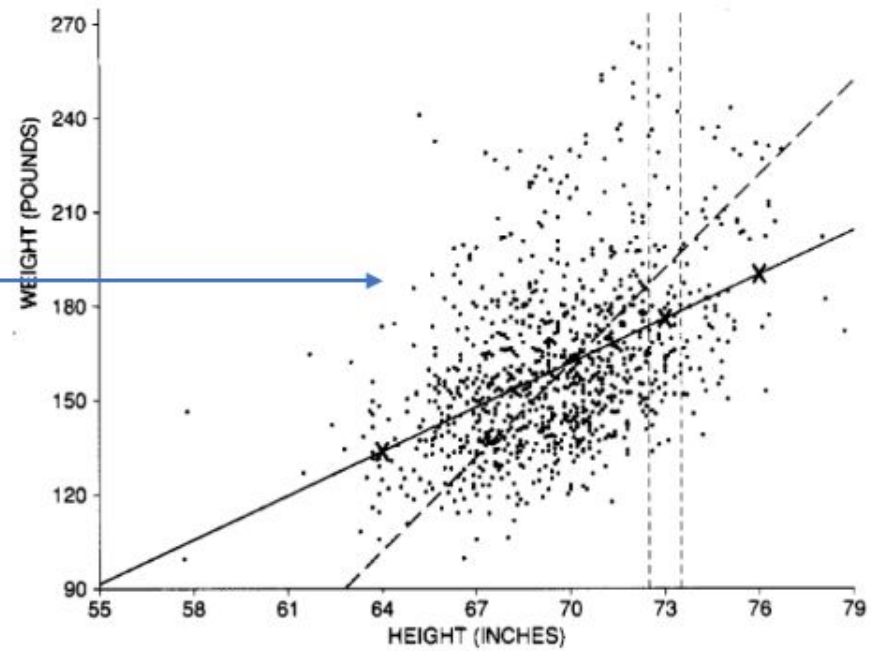
평범함으로 회귀

(다음 수치가 추정식의 평균으로 돌아오는 현상)



Step 반복수**(Iteration)**

: Epoch



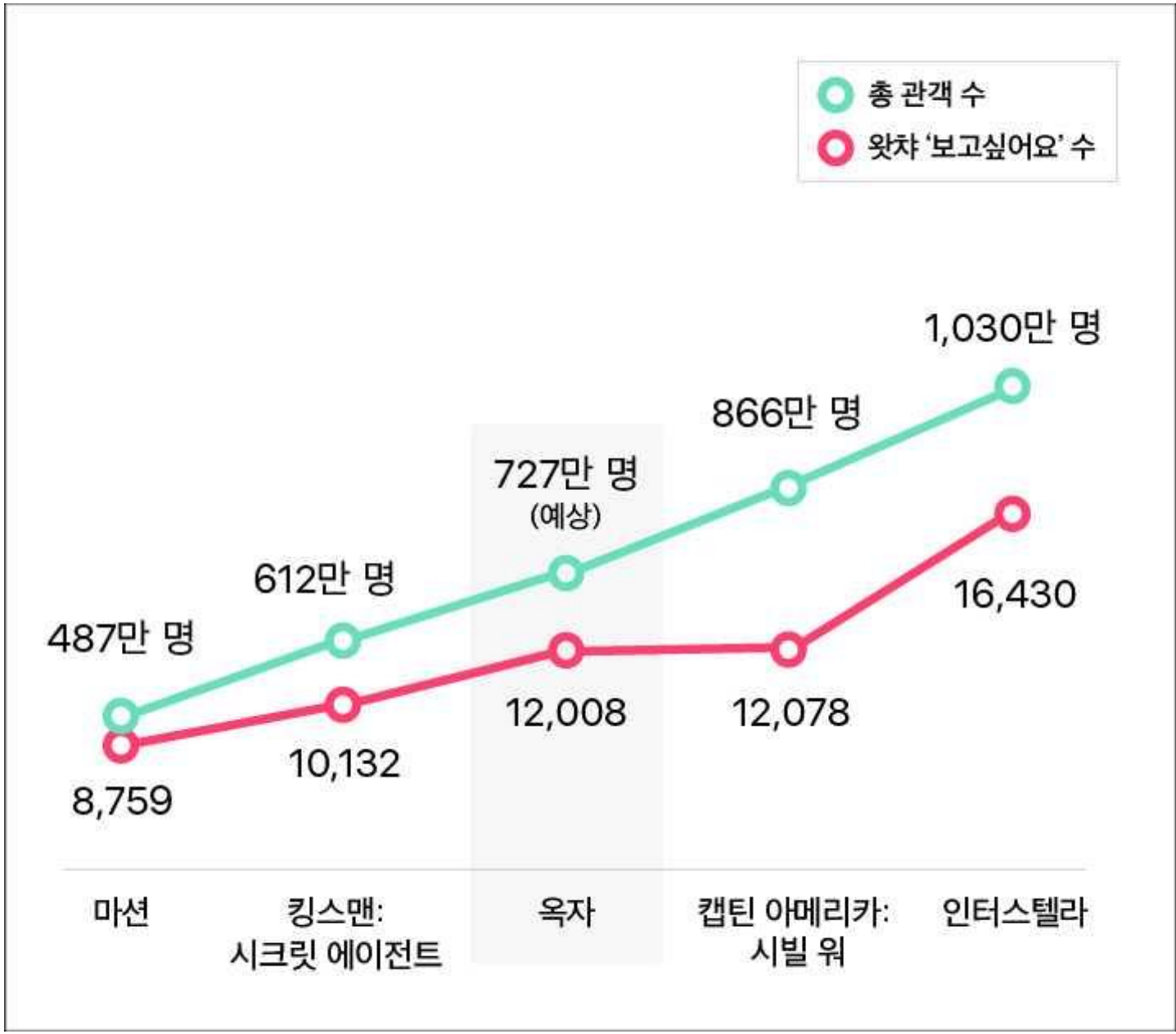
회귀의 사례

추정하고 싶은 수치 (Estimate)

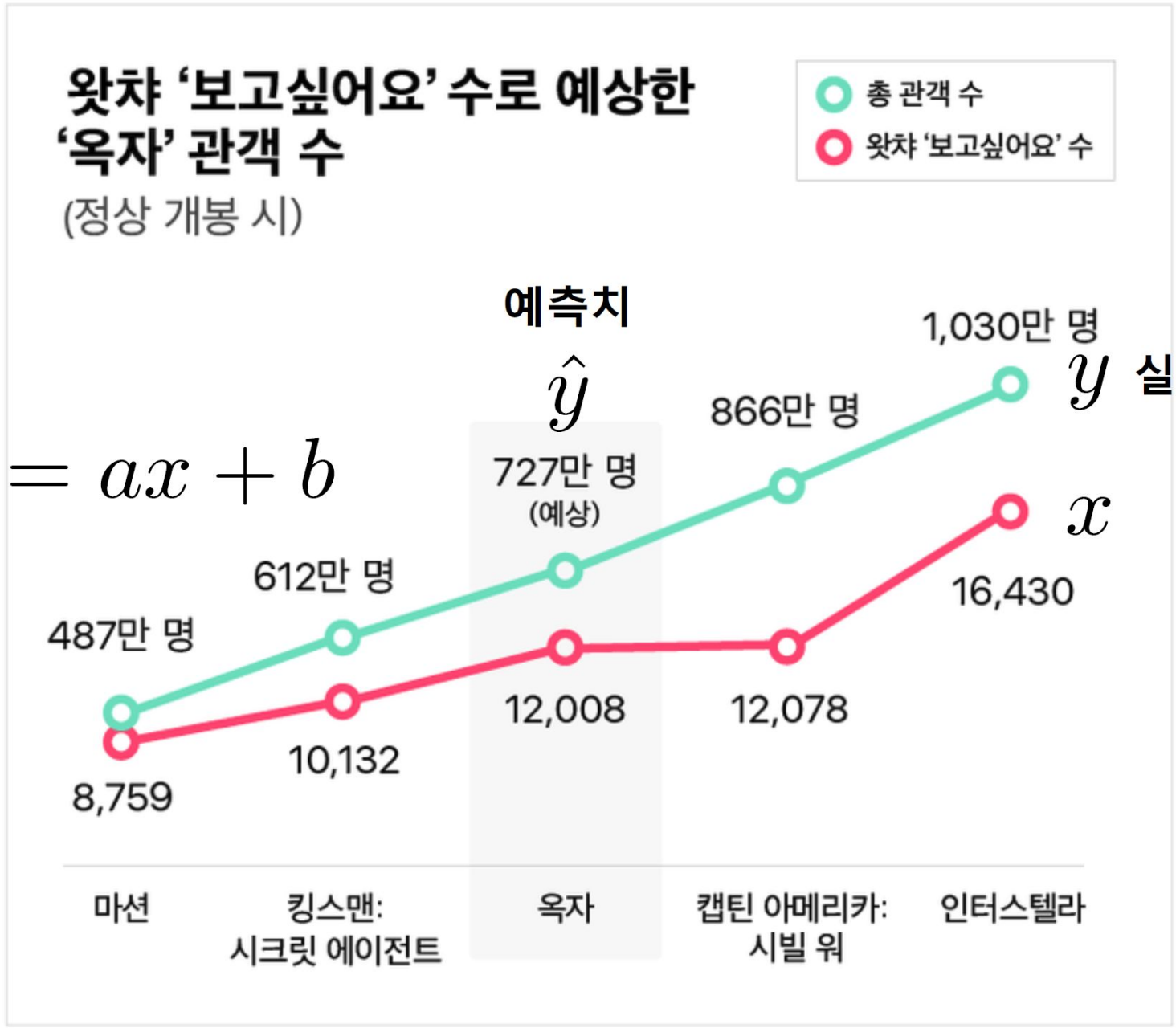
“영화관에서 개봉은 안한 옥자가
만약 극장에서 개봉했다면 예상 관객수는?”

추정에 사용된 정보의 특징 (Feature)

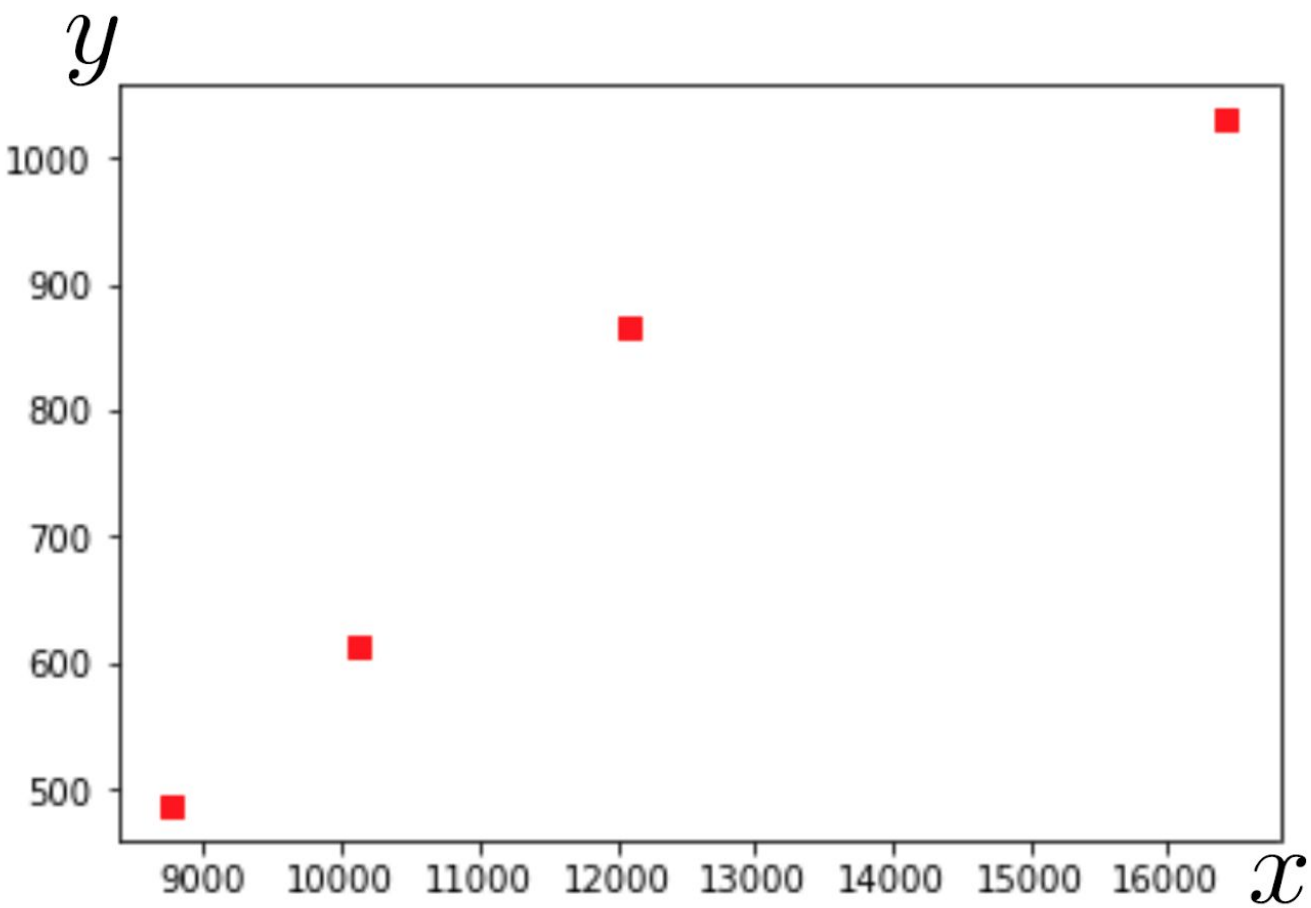
= 왓챠라는 영상앱에서 ‘보고싶어요’ 숫자

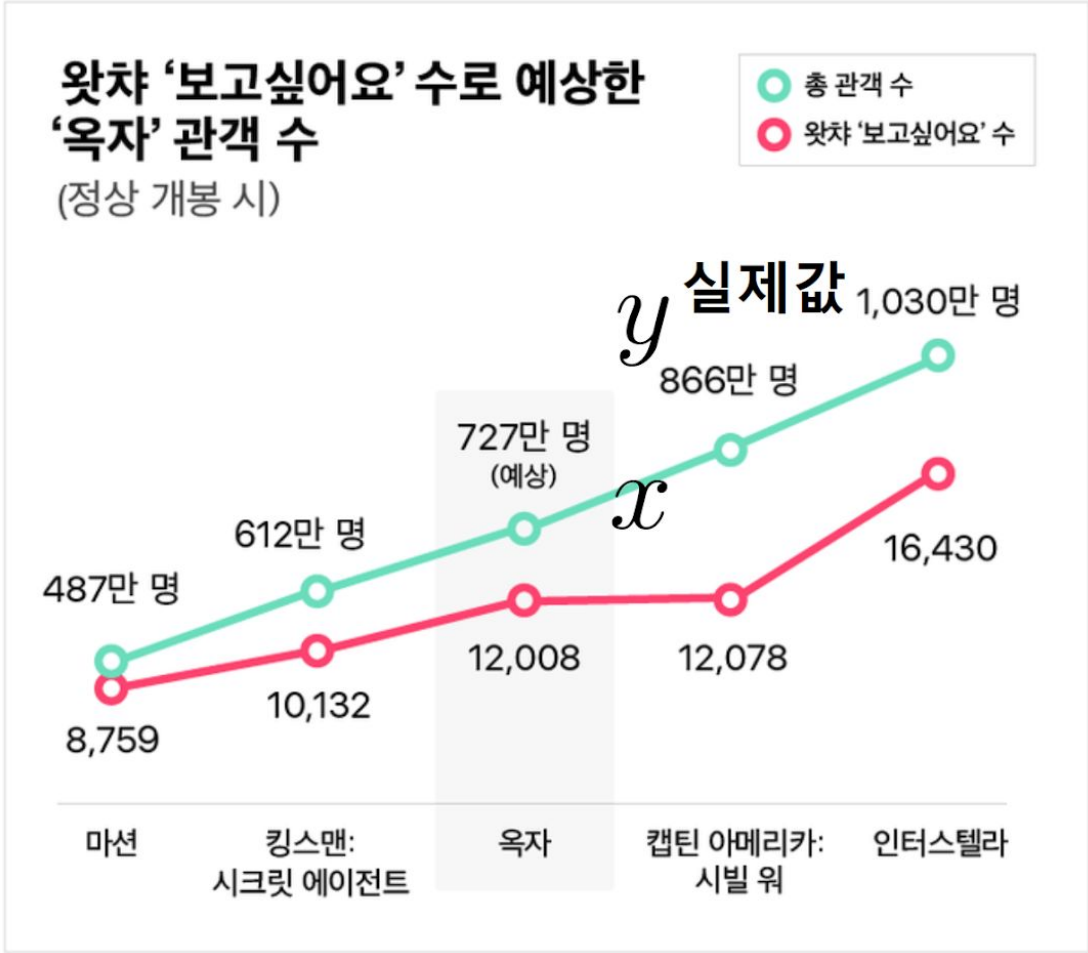


http://platum.kr/archives/83757



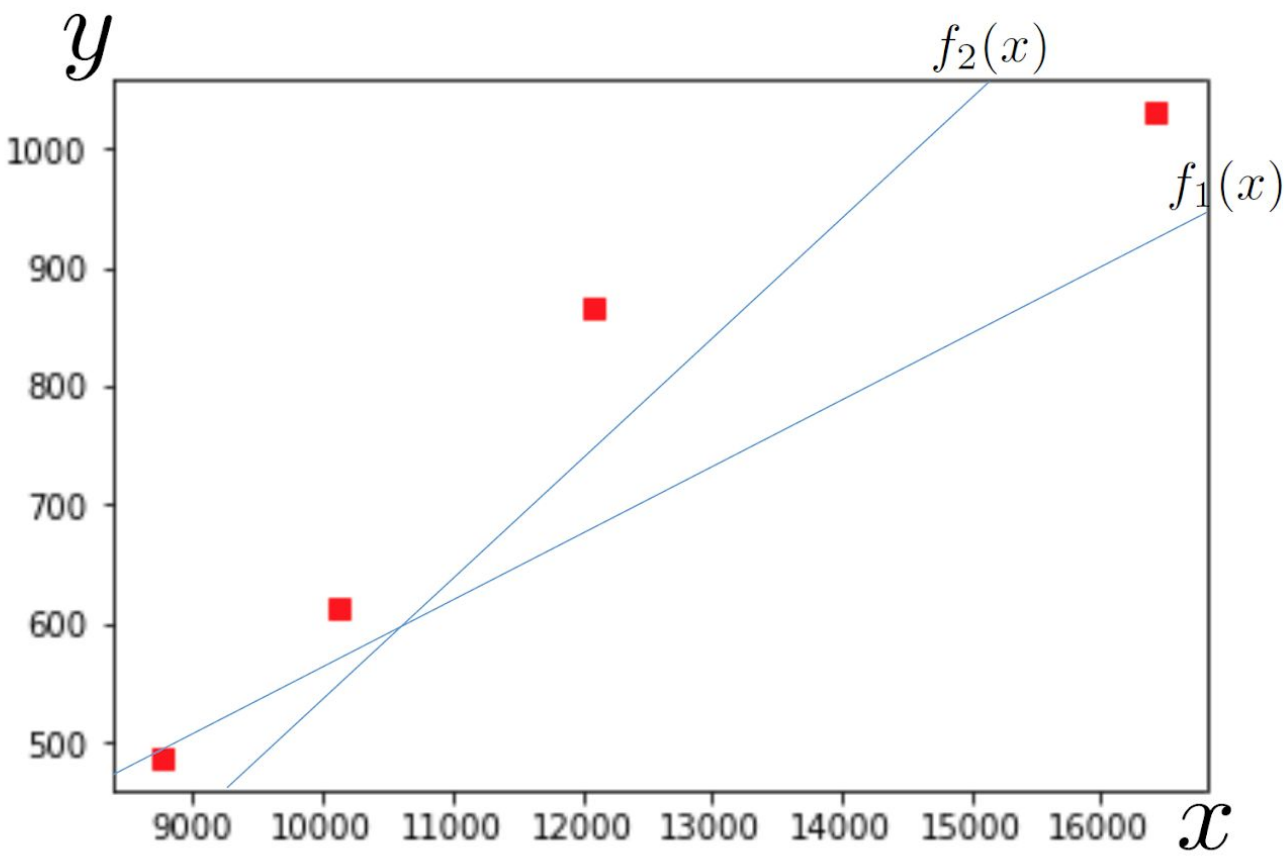
Source: http://platum.kr/archives/83757

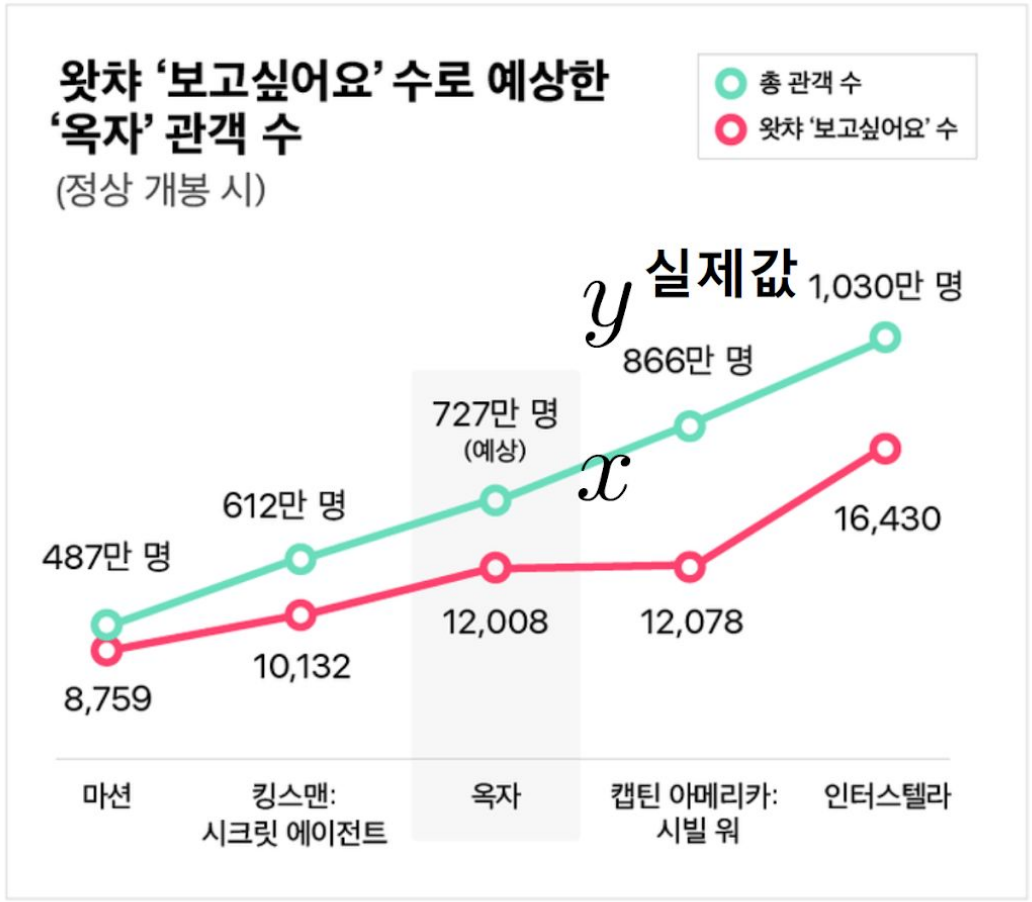




Source: http://platum.kr/archives/83757

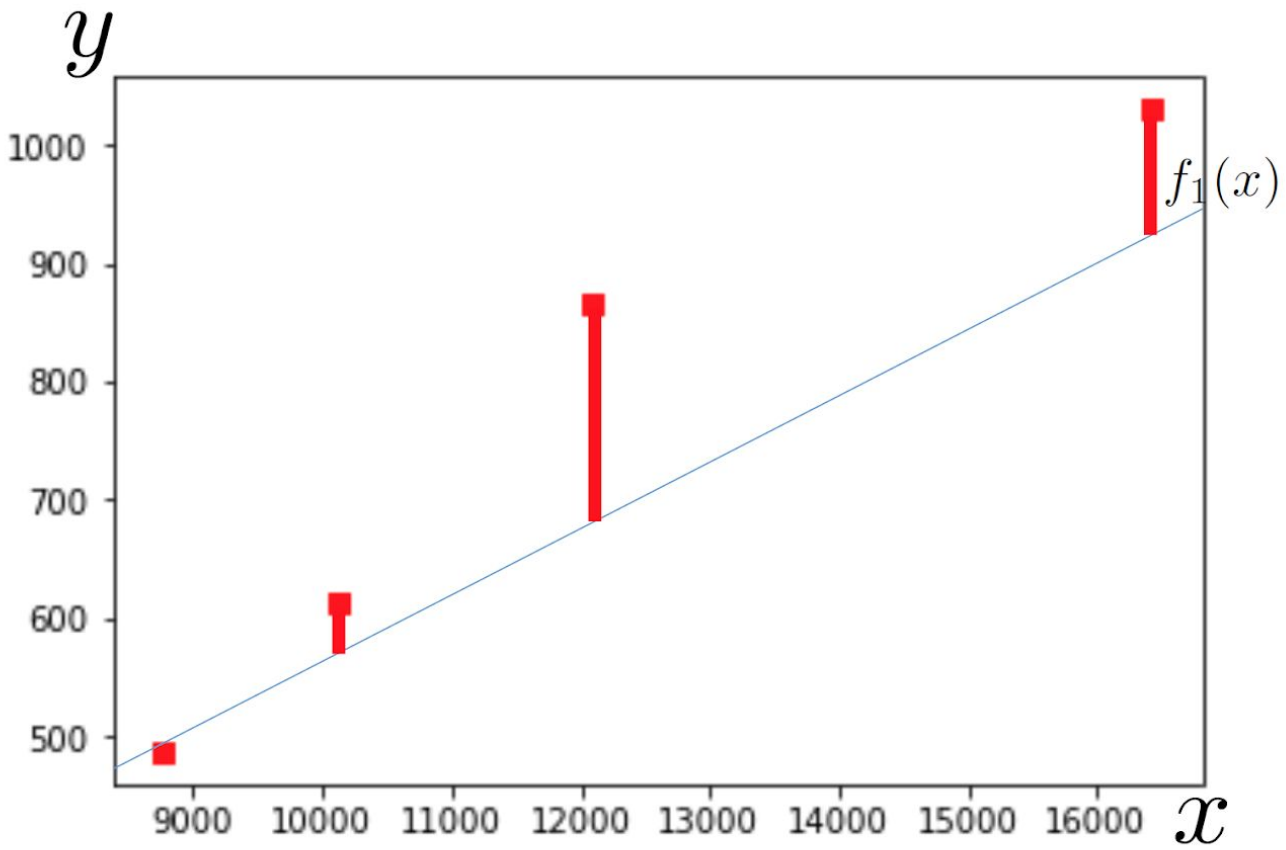
$$f(x) = \hat{y} = ax + b$$

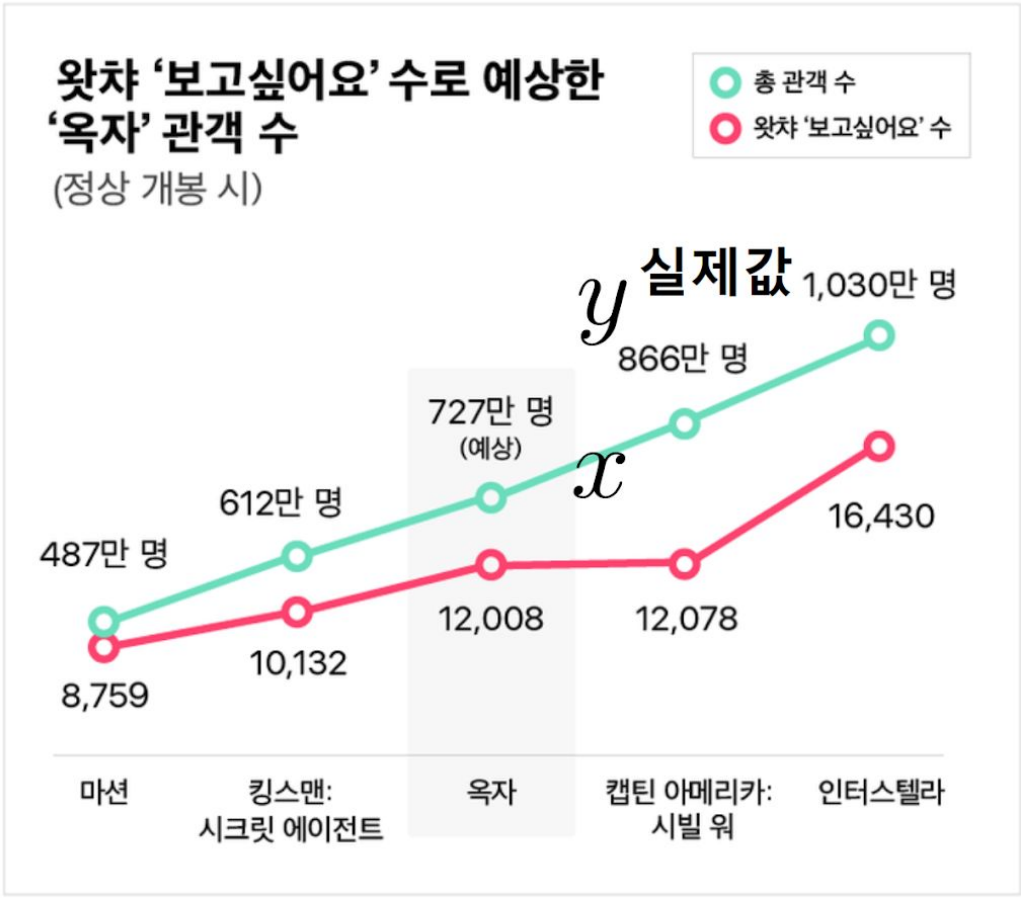




Source: http://platum.kr/archives/83757

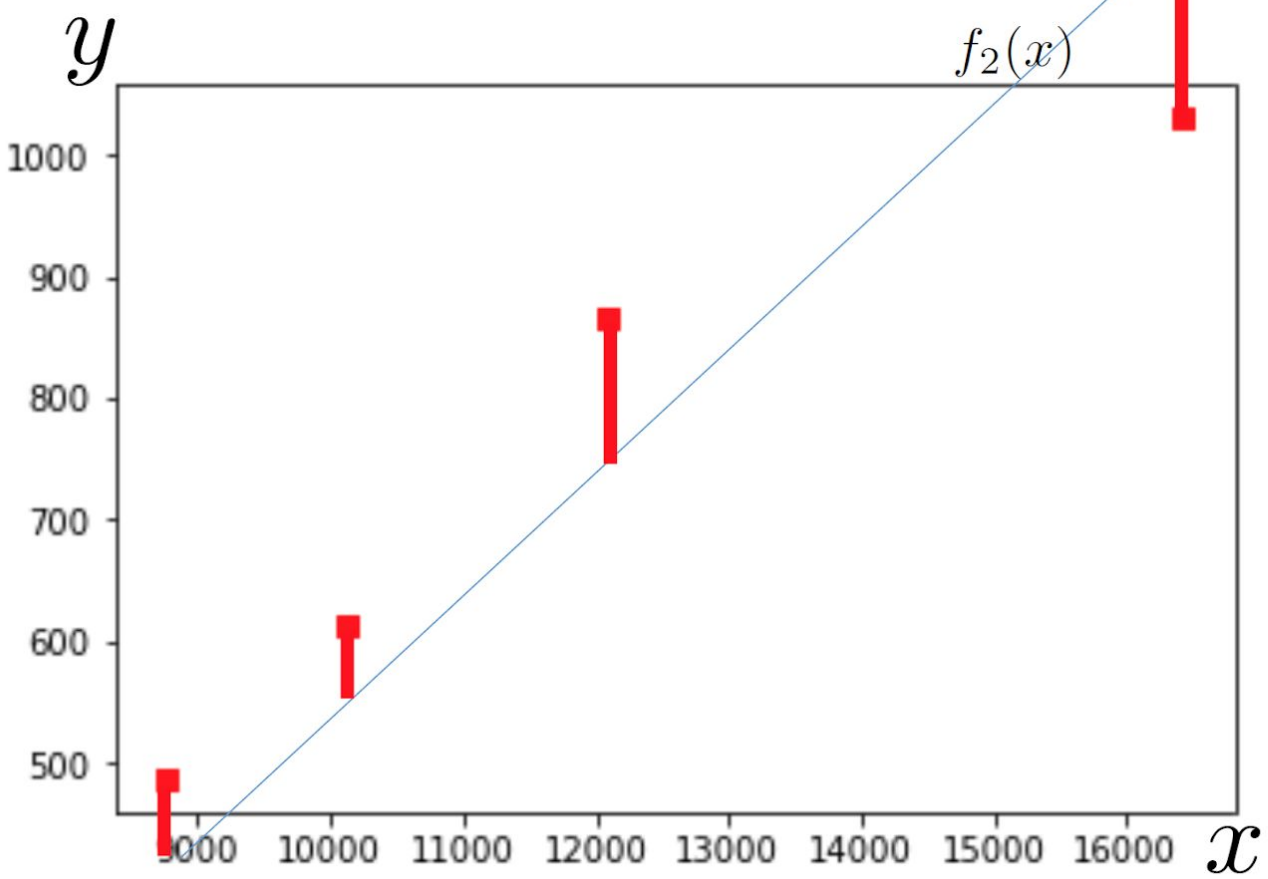
$$f(x) = \hat{y} = ax + b$$

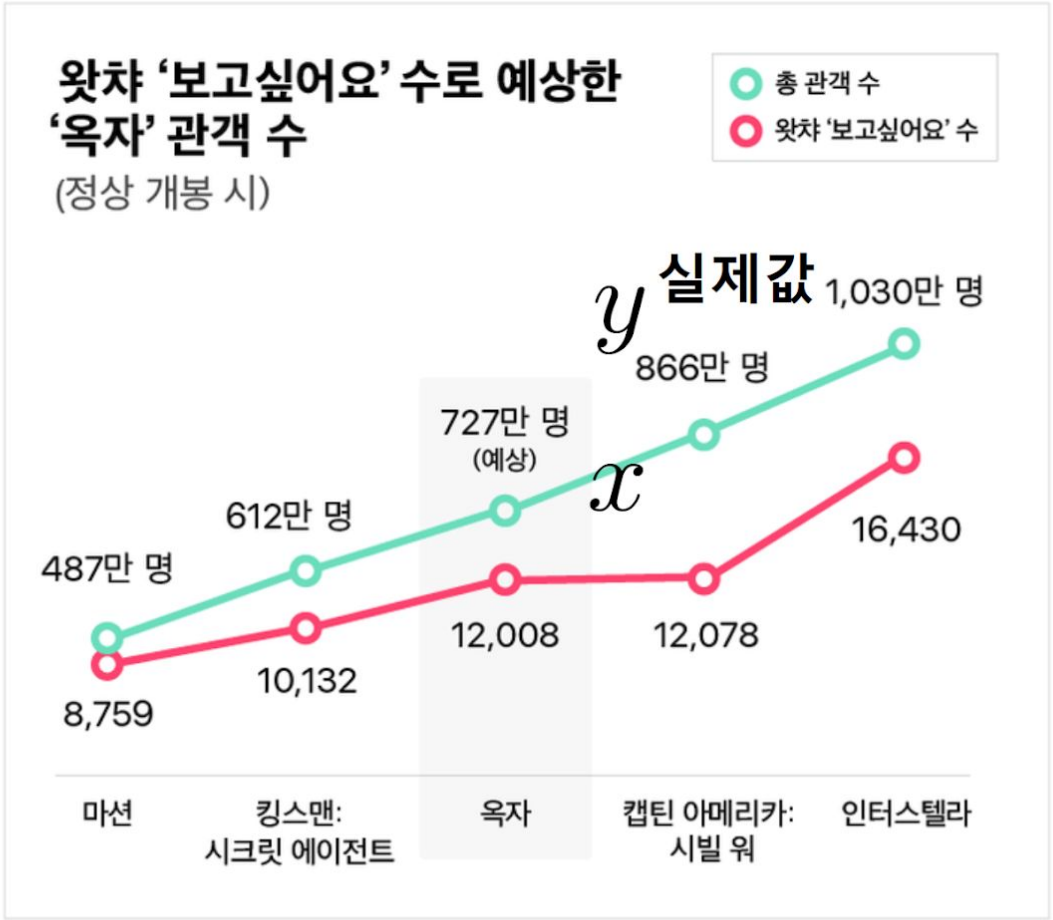




Source: http://platum.kr/archives/83757

$$f(x) = \hat{y} = ax + b$$





Source: http://platum.kr/archives/83757

Linear regression

- **Linear regression의 학습(파라미터 a, b 구하기) 원리**
  - 추정값과 실제값의 오차를 최소화 하자

추정값 $\hat{y} = ax + b$

실제값 y

- Linear regression의 학습 원리
  - 예측값과 실제값의 오차를 최소화 하자

오차의 합

$$(\hat{y}^{(1)} - y^{(1)}) + (\hat{y}^{(2)} - y^{(2)}) + (\hat{y}^{(3)} - y^{(3)}) + (\hat{y}^{(4)} - y^{(4)})$$

오차는 양수 또는 음수 가능 ➔ 상쇄될 수 있음

$$(\hat{y}^{(1)} - y^{(1)})^2 + (\hat{y}^{(2)} - y^{(2)})^2 + (\hat{y}^{(3)} - y^{(3)})^2 + (\hat{y}^{(4)} - y^{(4)})^2$$

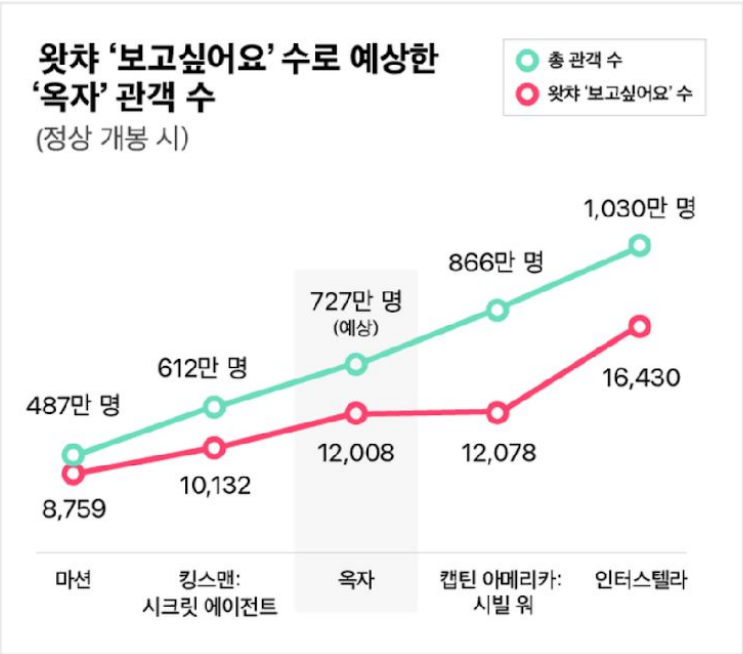
제곱의 합으로 변환

머신러닝 모델의 파라미터

- **파라미터**
  - 학습을 통해서 최적화 해줘야 하는 변수
  - 기호
    - ω_0, ω_1
    - w_0, w_1
    - θ_0, θ_1

- **하이퍼 파라미터**
  - 사람들이 선험적 지식으로 설정을 하거나 또는 외부 모델 메커니즘을 통해 자동으로 설정이 되는 변수

$$\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})^2 \quad \hat{y} = \begin{bmatrix} w_1 \times 8759 + w_0 \\ w_1 \times 10132 + w_0 \\ w_1 \times 12078 + w_0 \\ w_1 \times 16430 + w_0 \end{bmatrix} \quad y = \begin{bmatrix} 487 \\ 612 \\ 866 \\ 1030 \end{bmatrix}$$



$$(\hat{y} - y)^2 = \begin{bmatrix} (w_1 \times 8759 + w_0 - 487)^2 \\ (w_1 \times 10132 + w_0 - 612)^2 \\ (w_1 \times 12078 + w_0 - 866)^2 \\ (w_1 \times 16430 + w_0 - 1030)^2 \end{bmatrix}$$

Squared Error

Linear regression 학습

Squared error를
최소화 하는 파라미터(weight)를
탐색

Loss Function

VS

Cost Function

VS

Objective Function

Loss Function 분실 함수? 손실 함수?

VS

Cost Function 비용 함수

VS

Objective Function 목적 함수

Linear regression 모델

$$\sum_{i=1}^{n}(w_1 x^{(i)} + w_0 \times 1 - y^{(i)})^2$$

최소 또는 최대의 문제 → 미분으로 해결하기

찾고자 하는 값은? w_1, w_0

가설함수

$$f(x) = h_\theta(x)$$

- 추정함수를 가설 **hypothesis** 함수로 부른다.

Loss function

- **한 데이터 포인트에서의**
  - (예측-실제)의 차이 = '놓친 정도!'를 표현한 함수
  - 여러 종류의 함수가 존재할 수 있음

$$(h_\theta(x^{(i)}) - y^{(i)})^2$$

Cost function

- **데이터 전체에서**
  - 예측 값과 실제 값 차이의 평균

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Loss Function 분실 함수

순간순간 데이터 포인트마다 loss function에 의한 loss (놓침) 값

VS

학습이 완료된 후에는 cost function으로 총 cost (비용) 확인

Cost Function 비용 함수

VS

Objective Function 목적 함수

- **Objective function** is the most general term for any function that you optimize during training. For example, a probability of generating training set in maximum likelihood approach is a well defined objective function, but it is not a loss function nor cost function (however you could define an equivalent cost function). For example:
  - MLE is a type of objective function (which you maximize)
  - Divergence between classes can be an objective function but it is barely a cost function, unless you define something artificial, like 1-Divergence, and name it a cost

Long story short, I would say that:

A loss function **is a part of** a cost function **which is a type of** an objective function.