

실습을 통해 기초부터 배우는 머신러닝

18.12.01 – 18.12.29 (토 , 13:00 ~ 18:00)



Week 3



Agenda

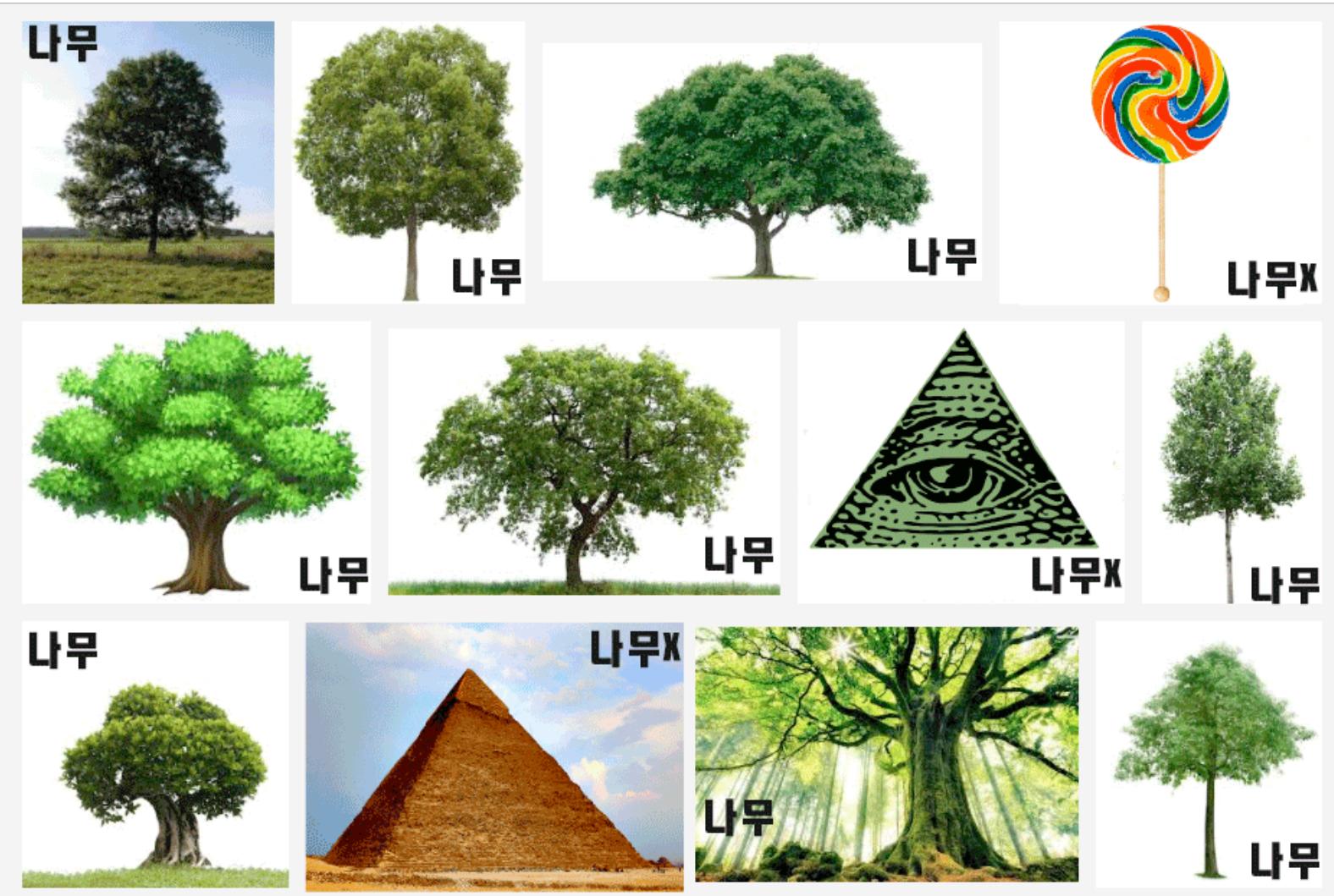
- 1. Introduction**
- 2. What's & Why ML?**
- 3. Categorization of ML**
- 4. Linear Regression**
- 5. Logistic Regression**
- 6. Practice**



How do we learn?



Learning process



Can you guess what it is?



There are many fruits here!



딸기



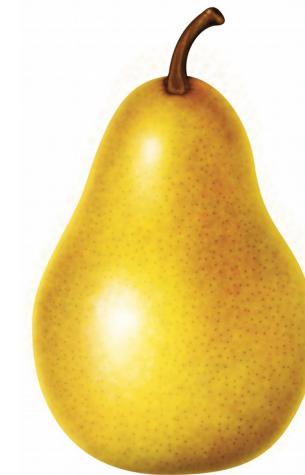
포도



바나나



사과



배

What's machine learning?

Machine learning \subseteq artificial intelligence

ARTIFICIAL INTELLIGENCE

Design an intelligent agent that perceives its environment and makes decisions to maximize chances of achieving its goal.
Subfields: vision, robotics, machine learning, natural language processing, planning, ...

MACHINE LEARNING

Gives "computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

SUPERVISED LEARNING

Classification, regression

UNSUPERVISED LEARNING

Clustering, dimensionality reduction, recommendation

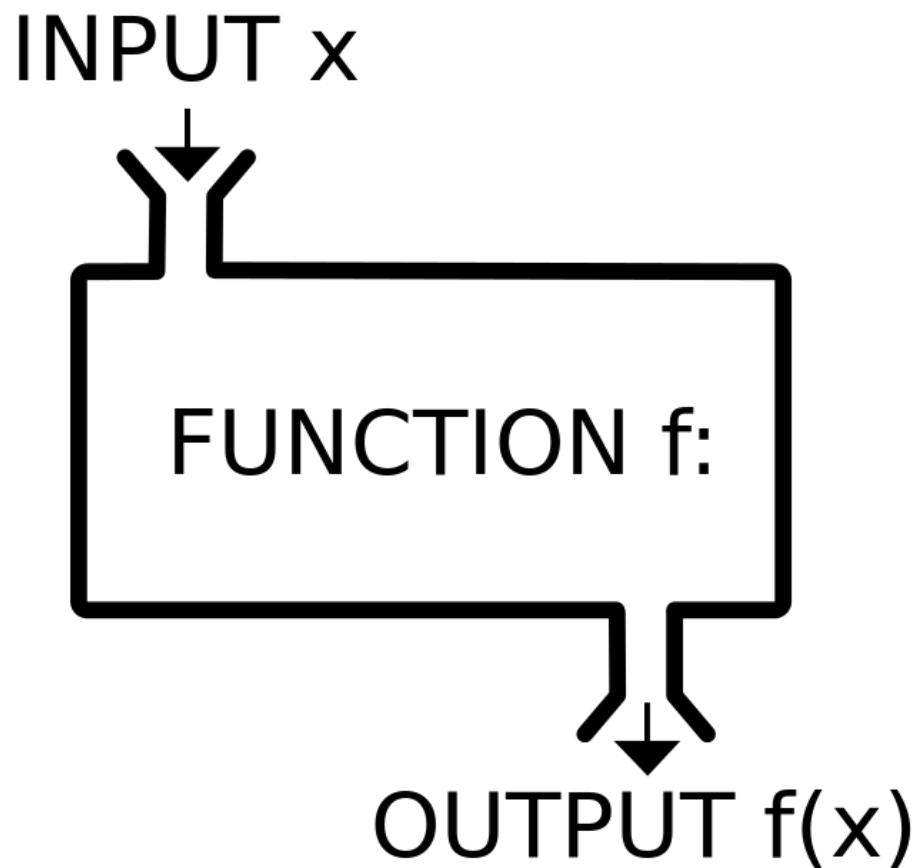
REINFORCEMENT LEARNING

Reward maximization

What's machine learning?

- 기계가 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구 분야
– Arthur Samuel, 1959
- 만약 작업 T 에 대해 기준 P 로 측정한 성능이 경험 E 로 인해 향상되었다면, 그 프로그램은 작업 T 에 대해 기준 P 의 관점에서 경험 E 로부터 배웠다 – Tom Mitchell, 1998
- 데이터로부터 패턴을 학습하여 새로운 데이터에 대해 적절한 작업을 수행하는 일련의 알고리즘이나 처리 과정

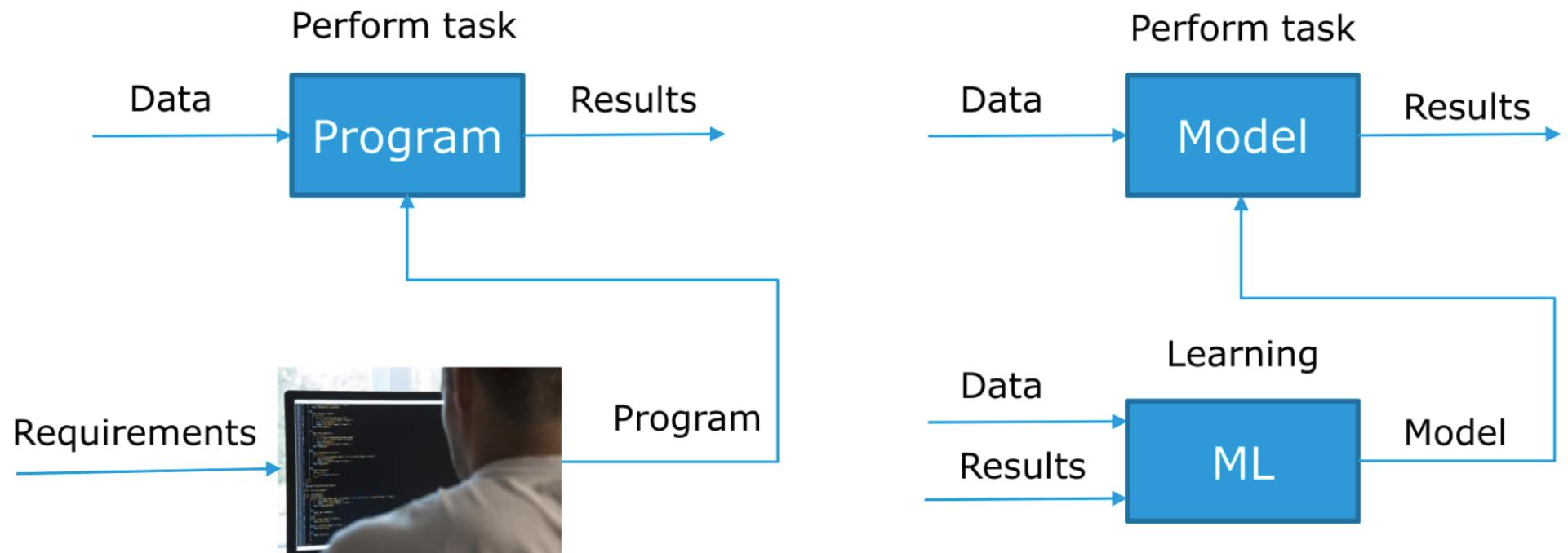
What's machine learning?



Through Machine Learning ?

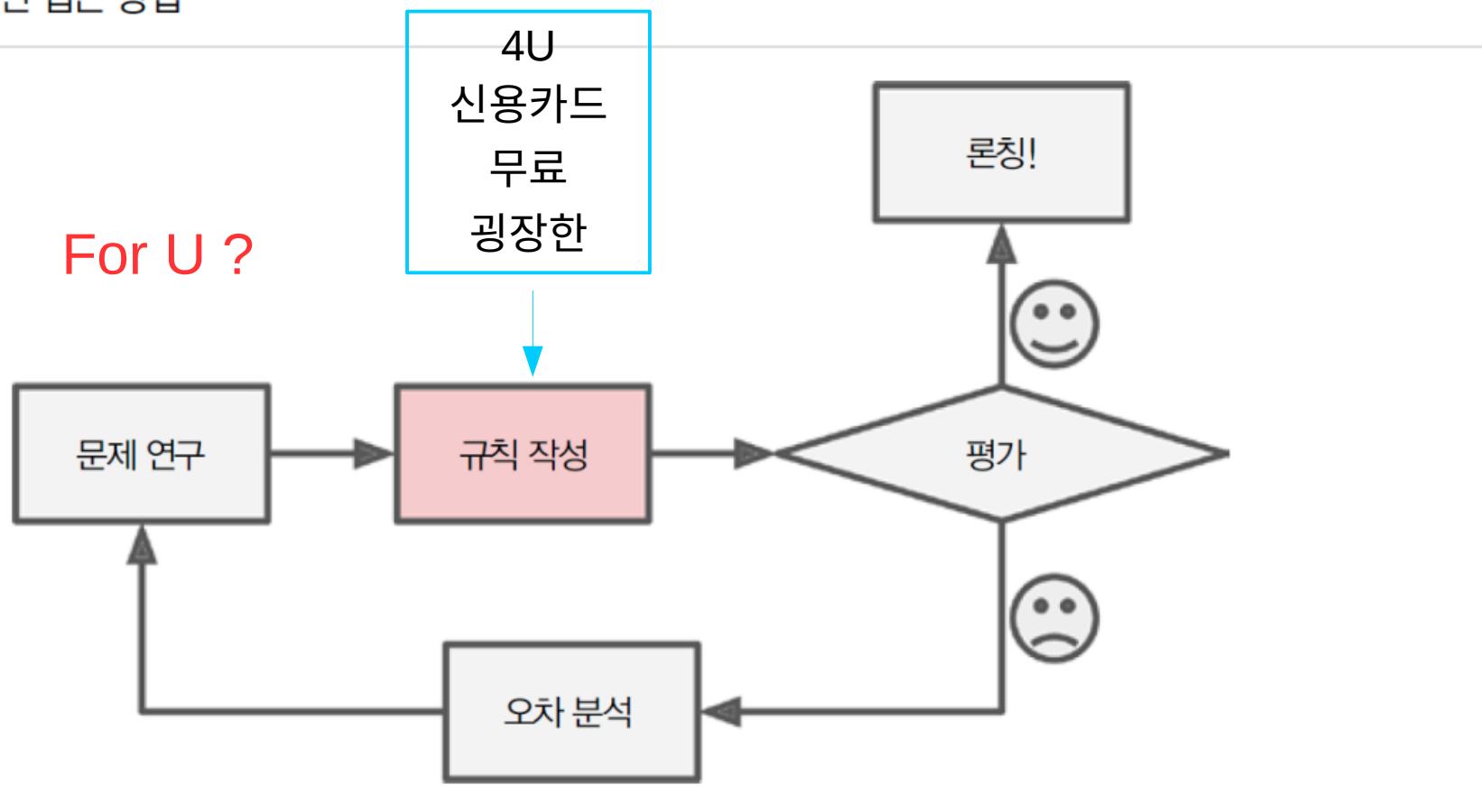
입력 값으로부터 원하는 결과 값을
출력해주는 함수를 얻는 것 !!!

What's different from programming



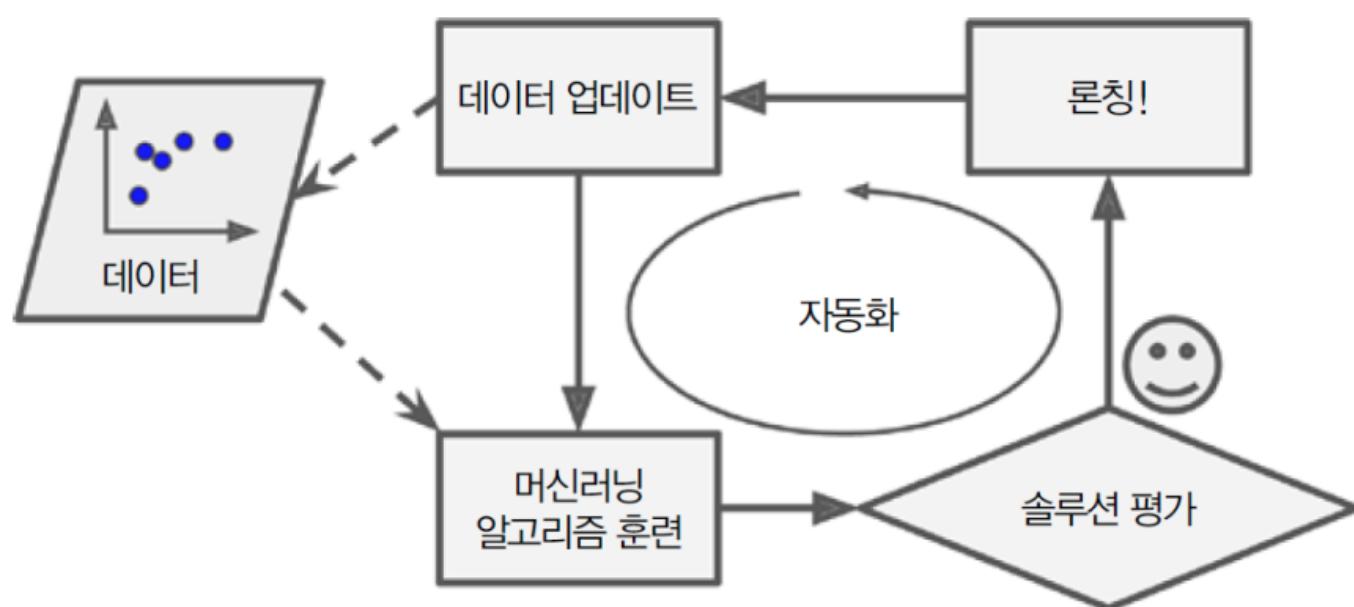
Example - spam filter

그림 1-1 전통적인 접근 방법



Example - spam filter

그림 1-3 자동으로 변화에 적응함



ML is good for...

- 기존 솔루션으로는 많은 수동 조정과 규칙이 필요한 문제

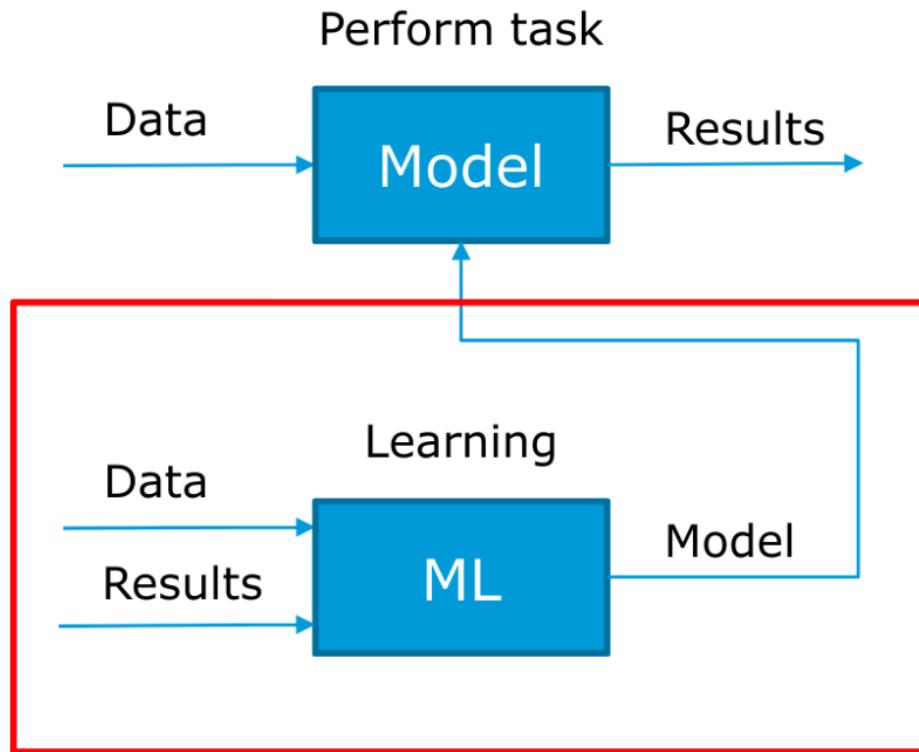
ex) 스팸 필터

- 전통적인 방식으로는 해결하기 어려운 복잡한 문제

ex) 이미지 분류 , 음성 인식



What's learning?

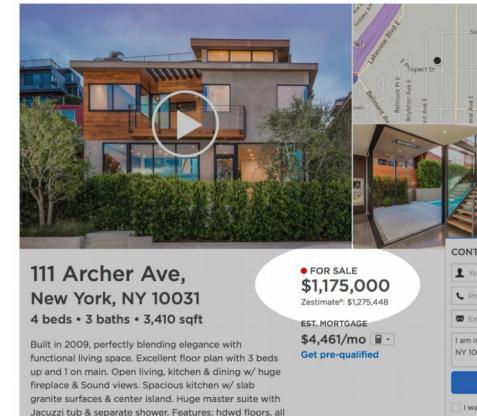


- Different Machine Learning techniques for different kinds of tasks
- Learning is finding which model's parameters represent best the input-output mapping

Which tasks can ML perform?



Spam Filter



Real Estate

Predict at which price
a property will be sold

A screenshot of an e-commerce website. At the top, it says 'Customers who bought this also bought'. Below are four product cards: 1. 'The Get Up Kids: Band Camp Pullover Hoodie' (green hoodie, \$30.00). 2. 'Akio Dresser' (wooden dresser, \$399.99). 3. 'Barcelona Bamboo Platform Bed' (bed frame, \$2,299.00). 4. 'Canon Digital Rebel XT 8MP Digital SLR Camera' (camera, \$550.00). Each card has an 'Add to Cart' button and links to 'Add to Wishlist' and 'Add to Compare'.

Recommendation Engine

A Google search results page for 'android mobile phones'. The top navigation bar shows 'All', 'Shopping', 'Images', 'News', 'Videos', 'More', 'Settings', and 'Tools'. The search bar contains 'android mobile phones'. Below the bar, it says 'About 9,340,000 results (0,68 seconds)'. A section titled 'Shop for android mobile phones' displays five sponsored ads for different phones: Sony Xperia X Compact Zwart, Samsung Galaxy J7 2017 Goud, Nieuwe Samsung A3 2017 Zwart, Samsung Galaxy C210 2017 Zwart, and LG K10 (2017) - €159.00. Each ad includes a small image of the phone, its name, price, and a link to KPN.com.

Mobile Phones bij KPN | Abonnement geldig in de EU | KPN.com
(Ad) mobileshop.kpn.com/MobilePhones
Bestel nu jouw mobile phone bij KPN. Voor 23:59 besteld = Morgen in huis!
Razendsnel 4G internet - Gratis Thuisbezorgd - Korting voor KPN-Klanten - Gratis Nummerbehoud

Samsung Android telefoons | Voor iedereen een smartphone
(Ad) www.samsung.com/Smartphones/Android
Bekijk alle Smartphones en ontdek welke Samsung Galaxy bij jou past!
2 jaar garantie - Water- en Stoofbestendig - Topcamera - Gratis verzending

Advertising

Predict which ads you are
more likely to click on

Which tasks can ML perform?

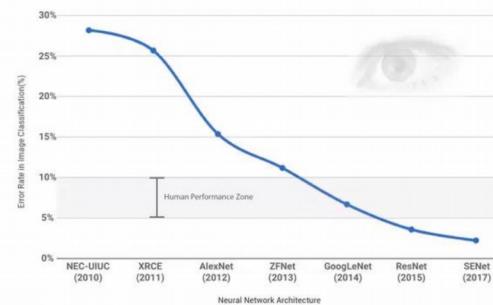
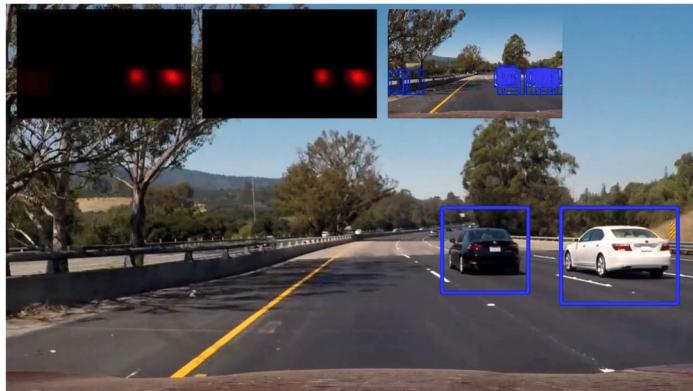
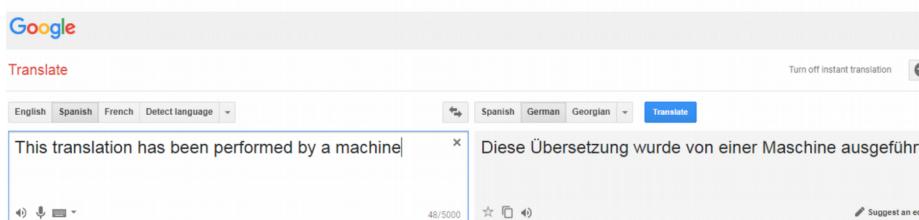


Image Classification



Self-driving cars



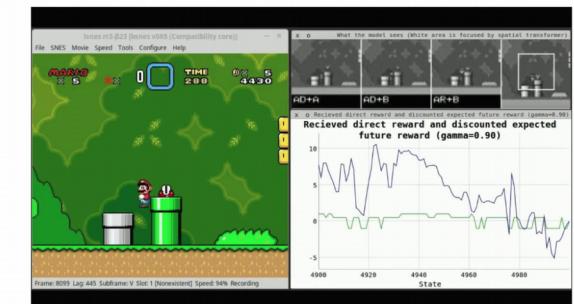
This translation has been performed by a machine | Diese Übersetzung wurde von einer Maschine ausgeführt



Speech Recognition & Synthesis



Playing Games



Why machine learning now?

- **Amount of data(Big data)**
- **Computing power(GPU)**
- **Improved Algorithms**



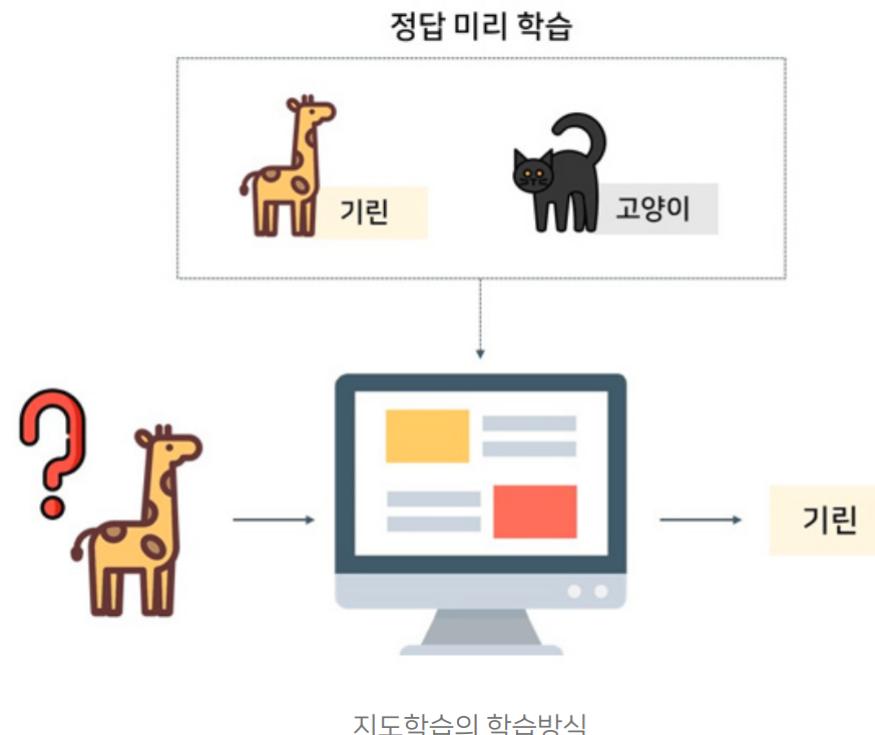
Categorization of ML



머신러닝의 분류

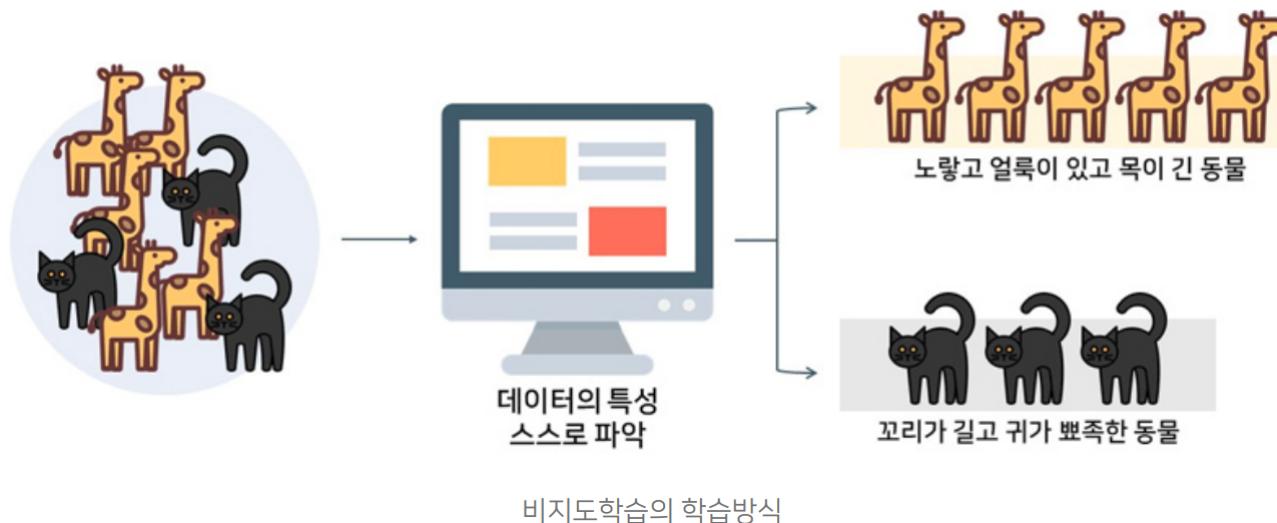
Categorization of ML

- 지도학습 (Supervised Learning)
: 정답이 주어진 상태에서 학습하는 알고리즘



Categorization of ML

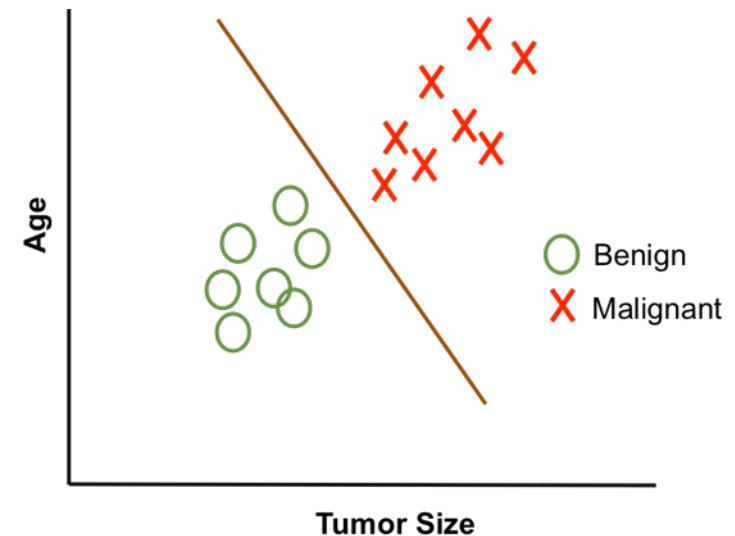
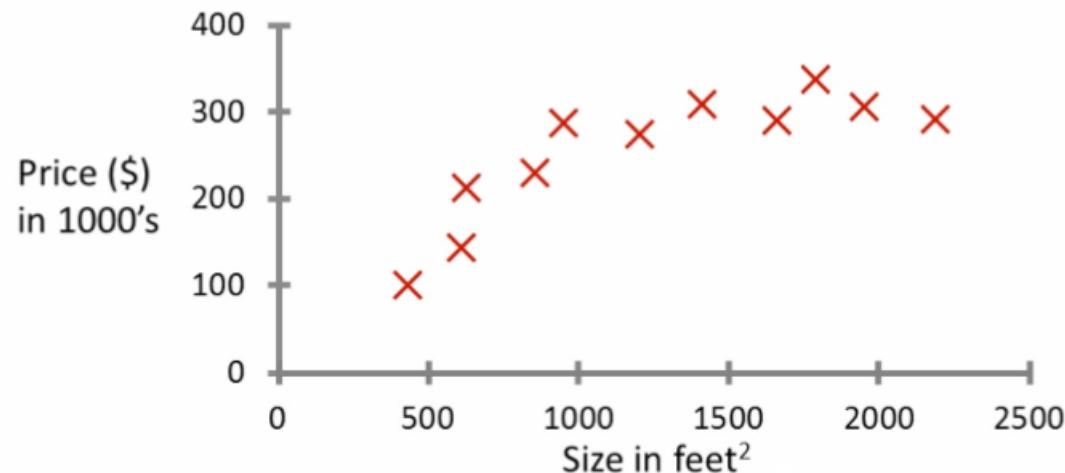
- 비지도 학습 (**Unsupervised Learning**)
: 정답이 주어지지 않은 상태에서 학습하는 알고리즘



Supervised learning

- 데이터 (입력)에 대한 레이블 (정답)이 존재
- 정답 종류에 따라 분류 또는 회귀

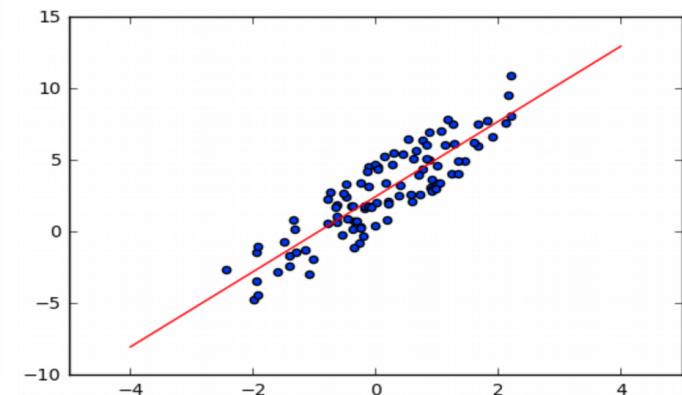
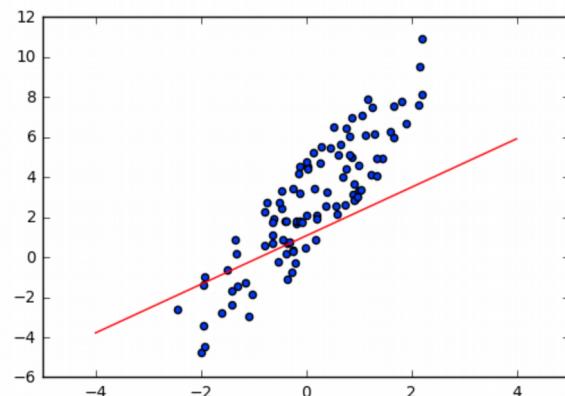
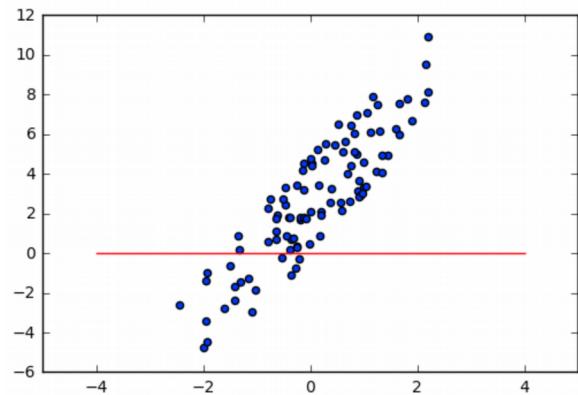
Housing price prediction.



Linear Regression(Supervised)

- 다음 중 데이터를 가장 잘 표현하는 함수(직선)는?

ex) 키와 몸무게



What's linear regression?

- 연속형 (수치) 값을 예측하기 위한 모델
- 주어진 데이터로부터 입력 (x) 와 출력 (y) 사이의 선형 관계를 분석하는 것

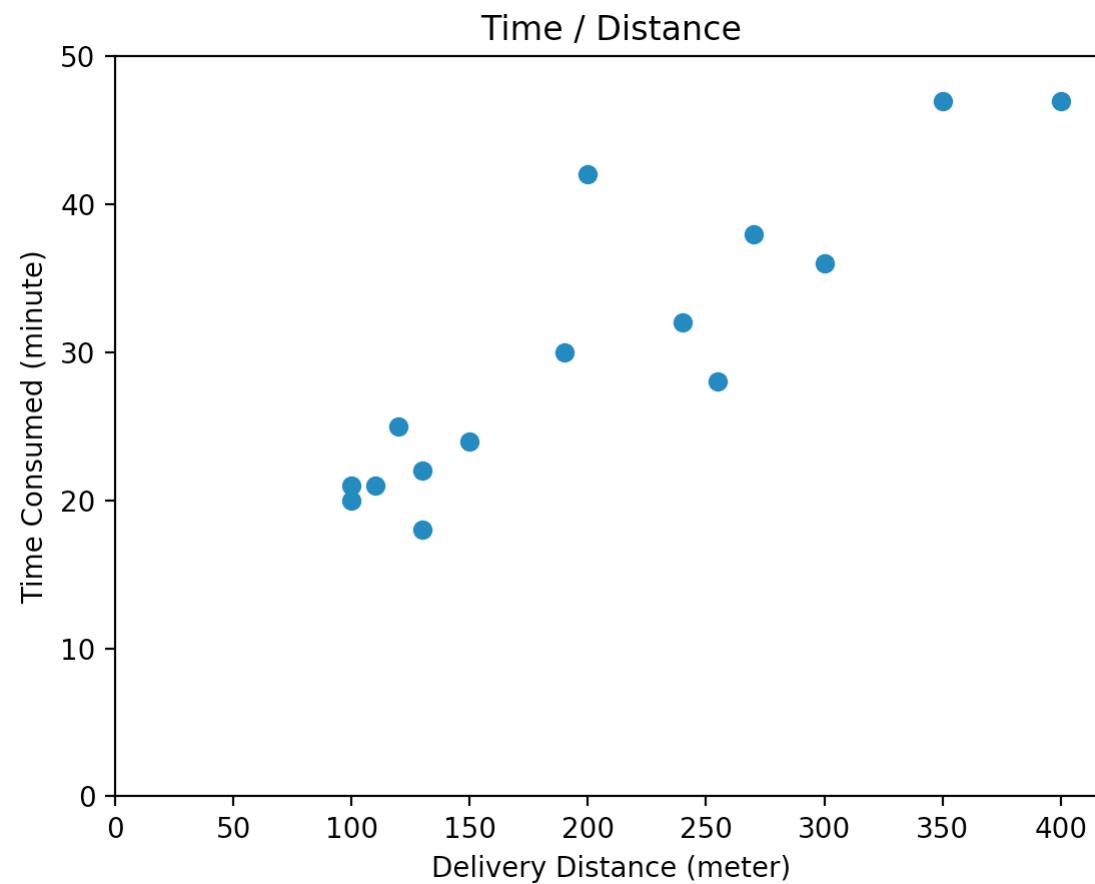
$$Y(\text{키}) = a * X(\text{몸무게}) + b$$

$$y = f(\theta_0 x_1 + \theta_1 x_2 + \dots + \theta_n x_n)$$



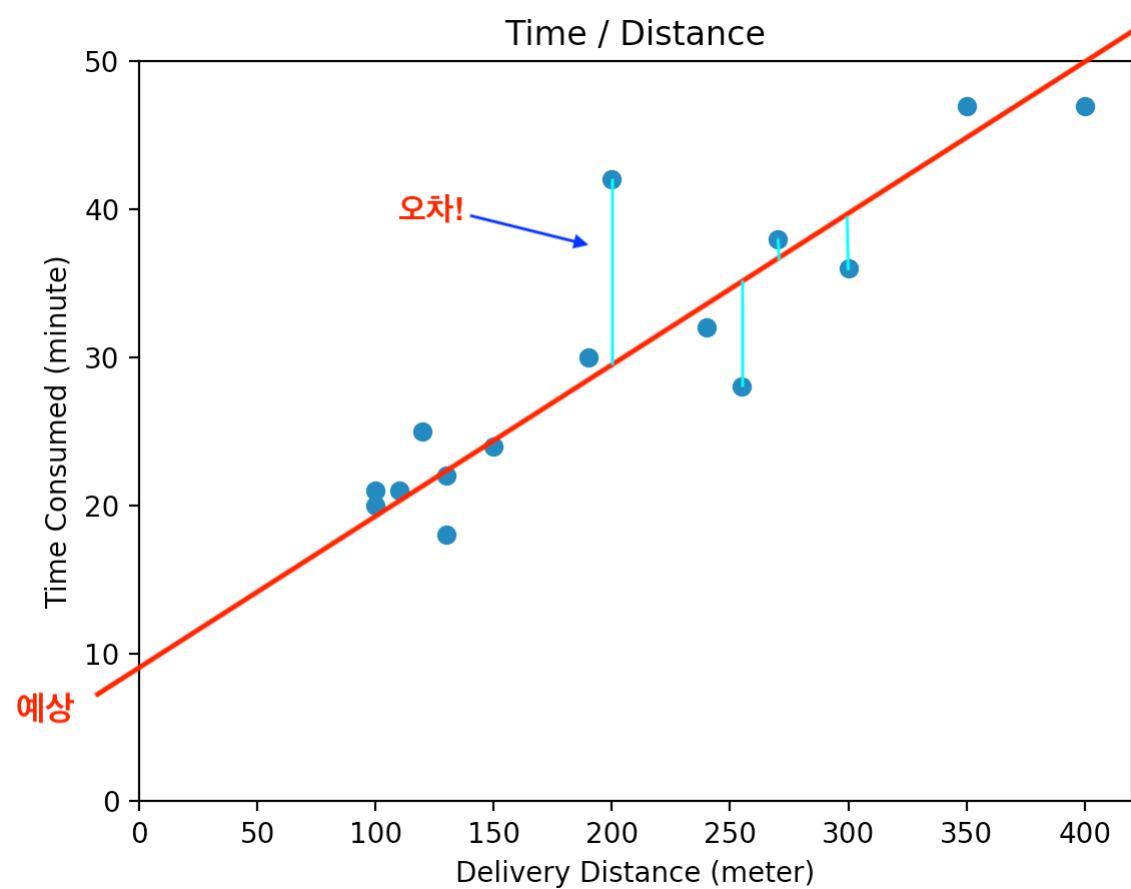
Linear Regression(Supervised)

- (배달 시간) 50m 거리 일 때는 ?



Linear Regression(Supervised)

- 예상 : 약 13 분 ?



오차가 최소가 되는
직선 찾기 !

$$Y(\text{배달시간}) = a * X(\text{거리}) + b$$

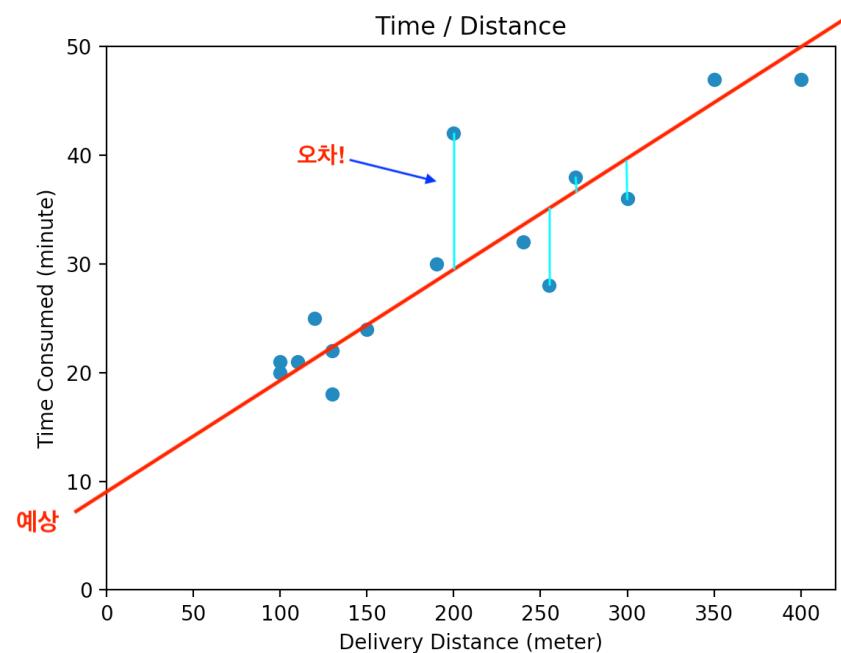
Linear Regression(Supervised)

- 가설 (hypothesis)

$$h_{\theta}(x) = \theta_0 + \theta_1 x \quad (\theta_i : \text{Parameter})$$

- 비용 함수 (cost function)

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



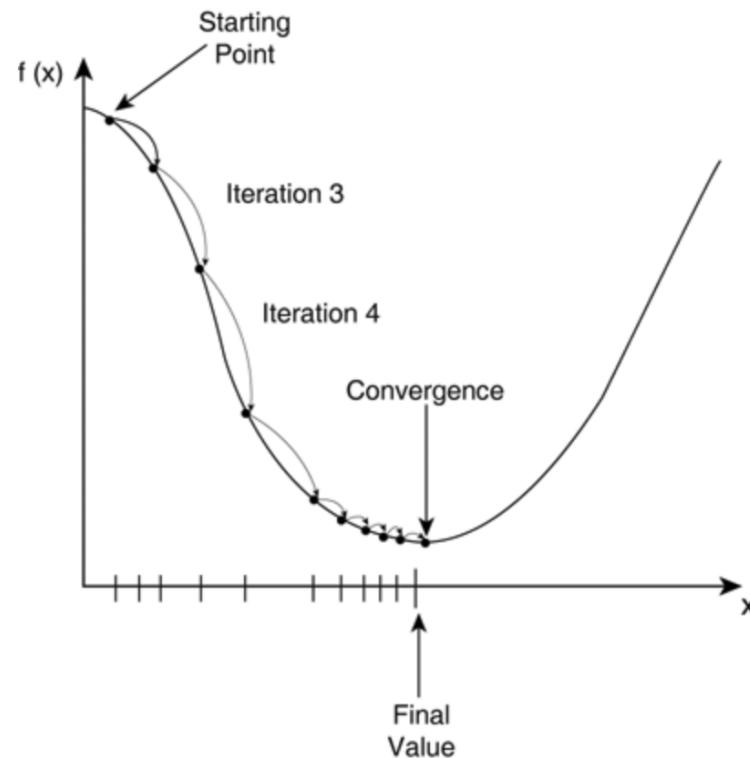
오차를 최소화하는 파라미터 (Θ) 를 찾는 것이 목적 !

= 데이터를 가장 잘 설명하는 모델을 찾는다 (model fitting)

How to minimize cost function?

- 경사하강법 (Gradient descent)

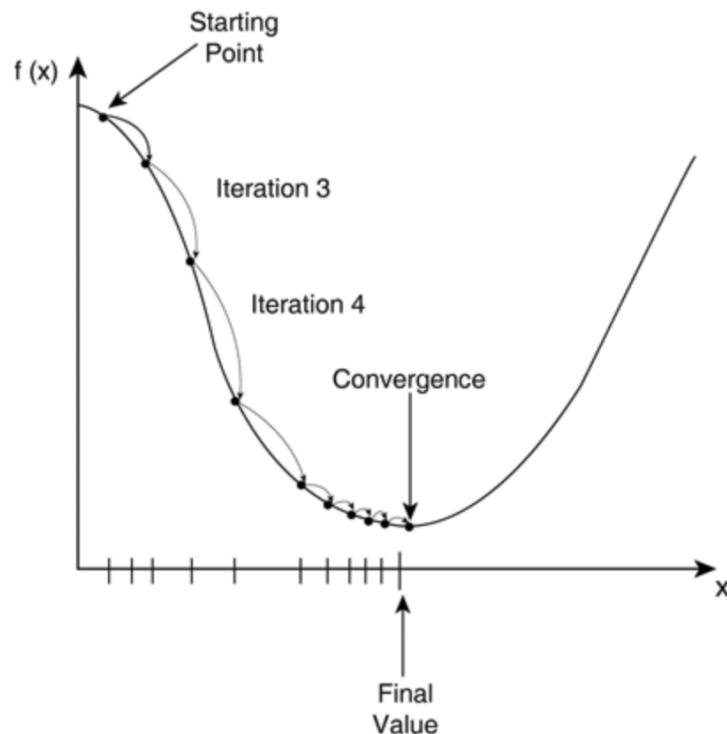
: 비용 함수를 최소화하기 위한 최적화 알고리즘 (기울기로 함수의 최솟값을 찾는다)



Gradient Descent Algorithm

$$h_{\theta}(x) = \theta_0 + \theta_1 x \quad (\theta_i : \text{Parameter})$$

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



1. 파라미터 초기화
2. 해당 파라미터에 대한 기울기 계산
3. 파라미터 업데이트
4. 비용 함수가 수렴할 때까지
(최솟값을 가질 때까지) 반복

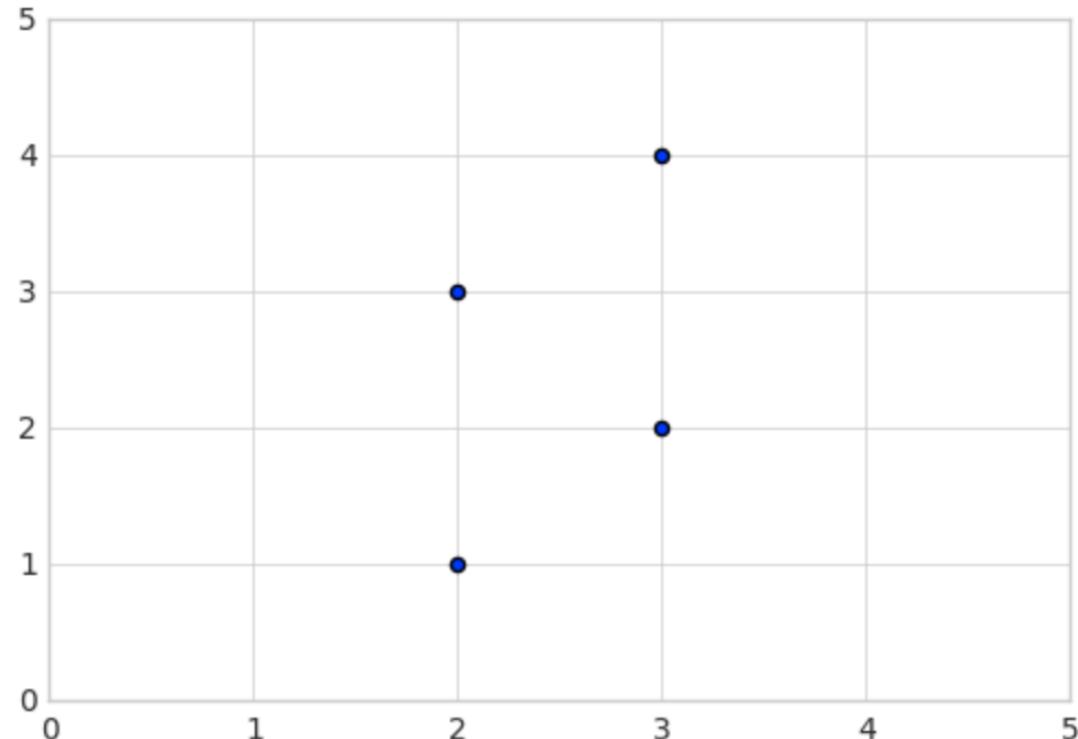
repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j=0 \text{ and } j=1)$$

}

α : learning rate(학습률)

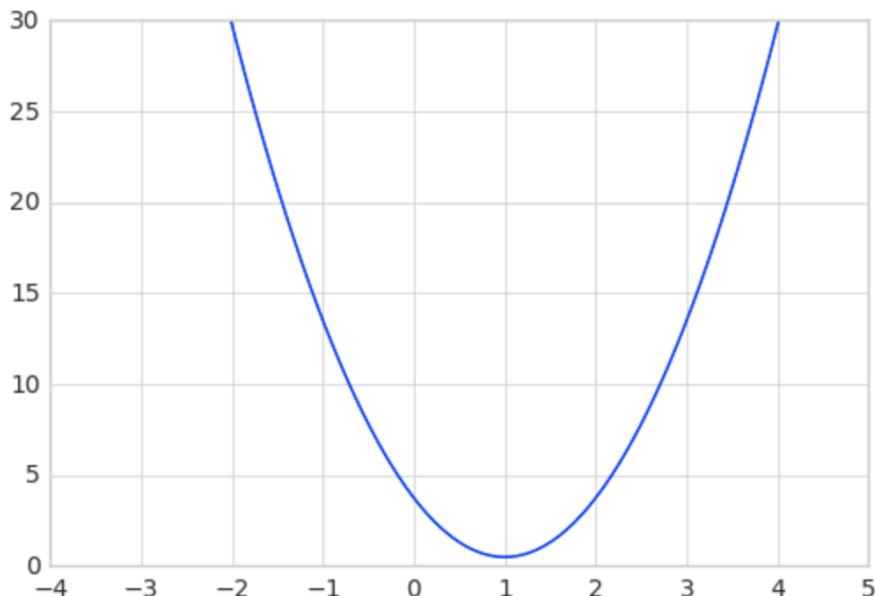
Gradient descent example



$$h_{\theta}(x) = \theta_1 x$$

$$J(\theta_1) = \frac{1}{2*4} \sum_{i=1}^4 (h_{\theta}(x^i) - y^i)^2 = \frac{1}{8} (26\theta_1^2 - 52\theta_1 + 30)$$

Gradient descent example



$$J(\theta_1) = \frac{1}{8}(26\theta_1^2 - 52\theta_1 + 30)$$

θ_1 의 초기값 = 0 , α = 0.1

```
for i in range(38):
    theta = update_theta(theta, alpha)
    print (theta)
```

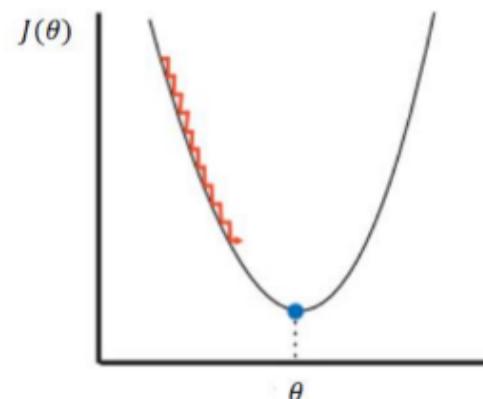
```
0.65
0.8775000000000001
0.957125
0.98499375
0.9947478125
0.998161734375
0.99935660703125
0.9997748124609375
0.9999211843613282
0.9999724145264649
0.9999903450842628
0.9999966207794919
0.9999988172728221
0.9999995860454878
0.9999998551159207
0.9999999492905722
0.9999999822517003
0.999999937880951
0.999999978258333
0.999999992390416
0.999999997336646
0.999999999067826
0.999999999673739
0.99999999985809
0.999999999960033
0.999999999986011
0.99999999995104
0.99999999998286
0.9999999999994
0.99999999999979
0.99999999999927
0.999999999999974
0.999999999999991
0.999999999999997
0.999999999999999
1.0
1.0
1.0
```

Learning rate(학습률)

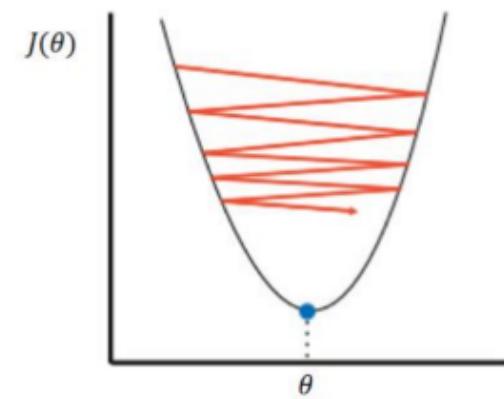
- 얼마나 많이 이동할 것인가 ?

```
theta = 0
alpha = 0.5
for i in range(10):
    theta = update_theta(theta, alpha)
    print (theta)
```

3.25
-4.0625
12.390625
-24.62890625
58.6650390625
-128.746337890625
292.92926025390625
-655.8408355712891
1478.8918800354004
-3324.256730079651

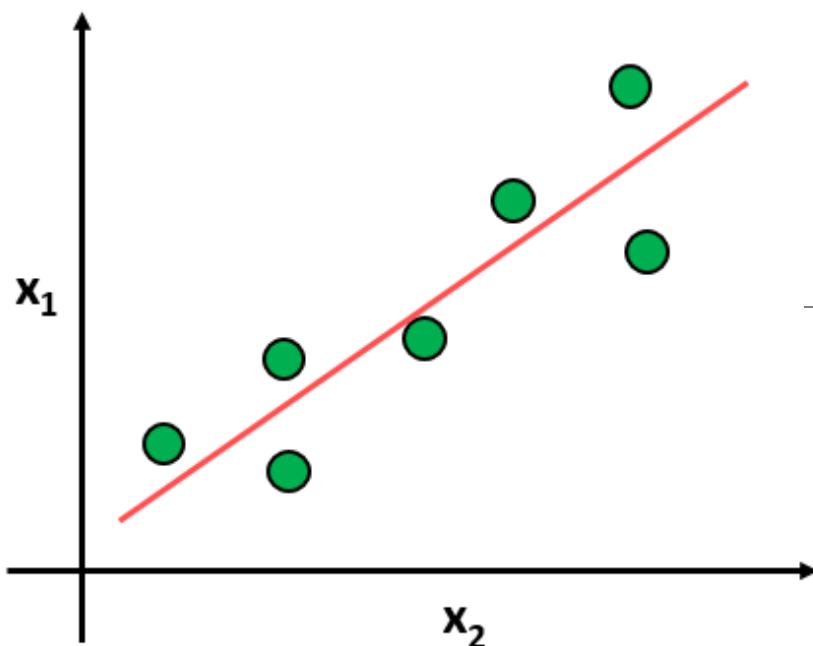


Too small: converge very slowly

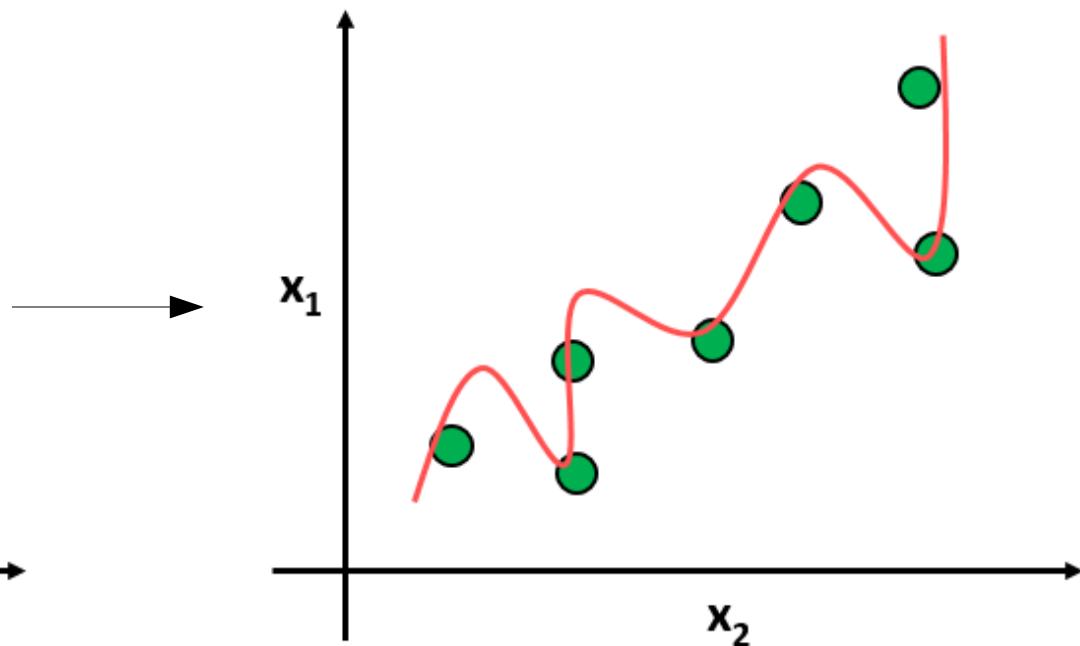


Too big: overshoot and even diverge

Why not?

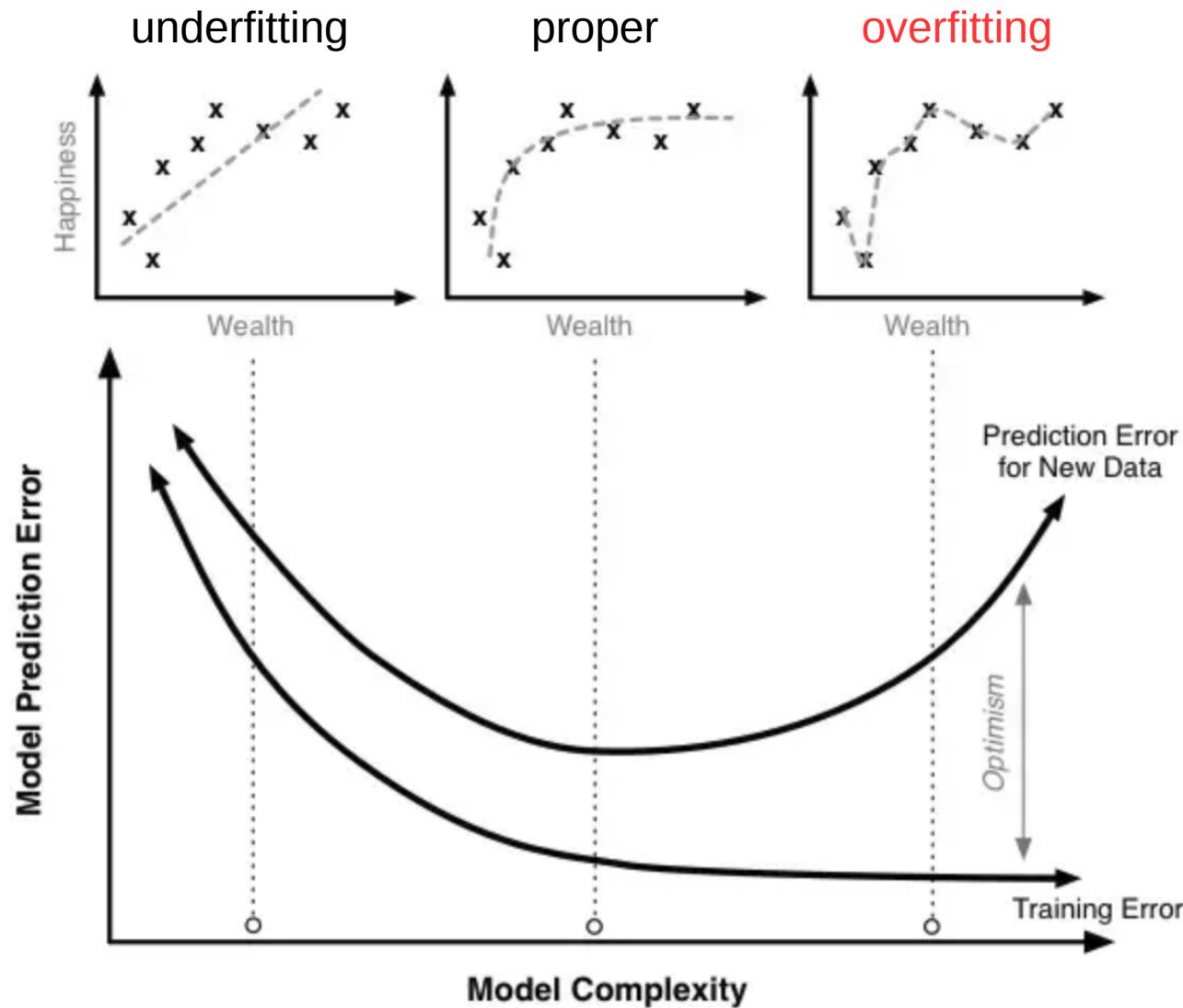


$$y = wx + b$$

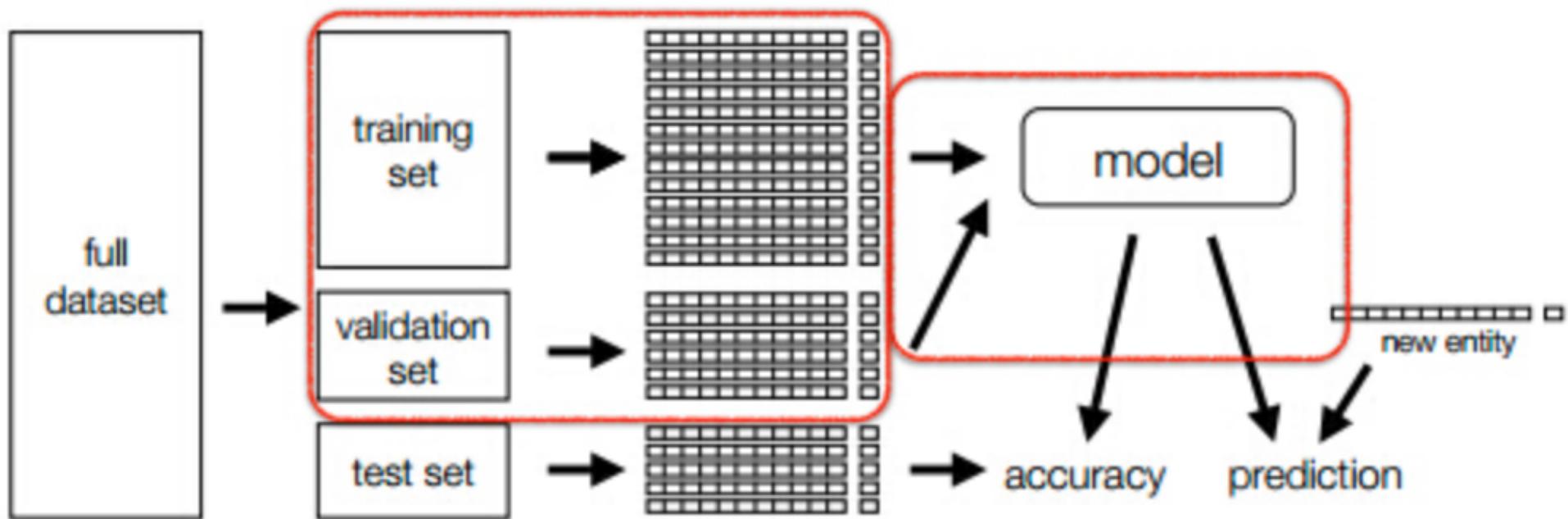


$$y = w_1x_1 + w_2x_2 + \dots + w_9x_9 + b$$

Model complexity



Data set includes ...



Training set : 모델 학습

Validation set : 모델 선택 / Hyper parameter 최적화

Test set : 오직 평가

Logistic Regression(Supervised)

- 분류 문제를 해결하기 위한 모델 (주로 이진 분류)
ex) 스팸 메일 분류 , 질병 양성 / 음성 분류 등
- 결과 값을 0 또는 1로 표현
- 특정 범주 (0 또는 1)에 속할 확률을 계산

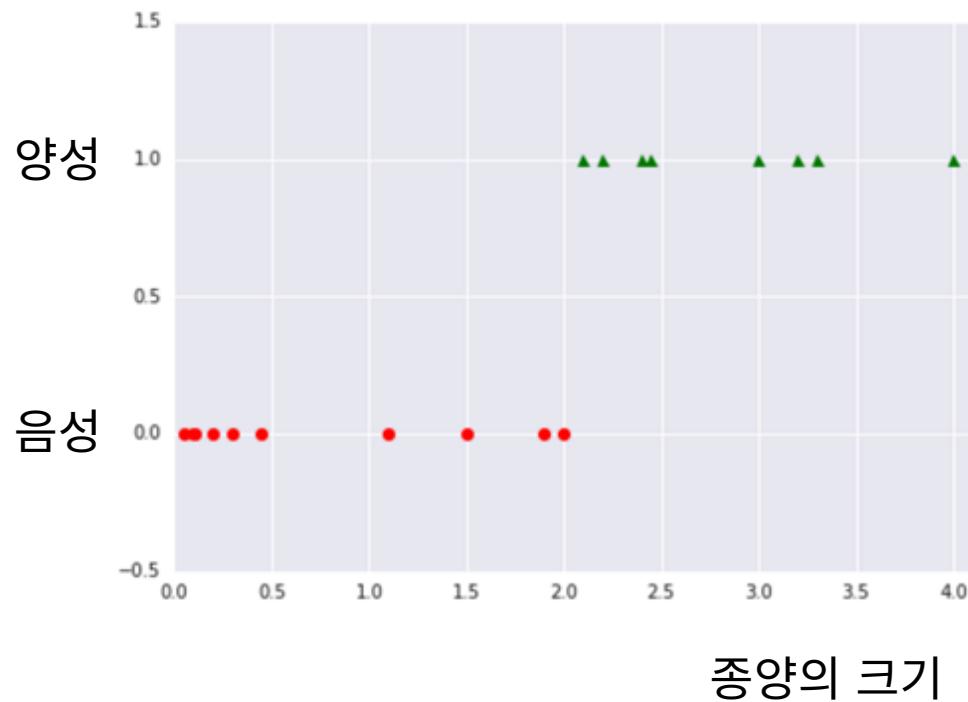
Logistic Regression(Supervised)

- 고객이 5년 내에 사망할 확률 ?

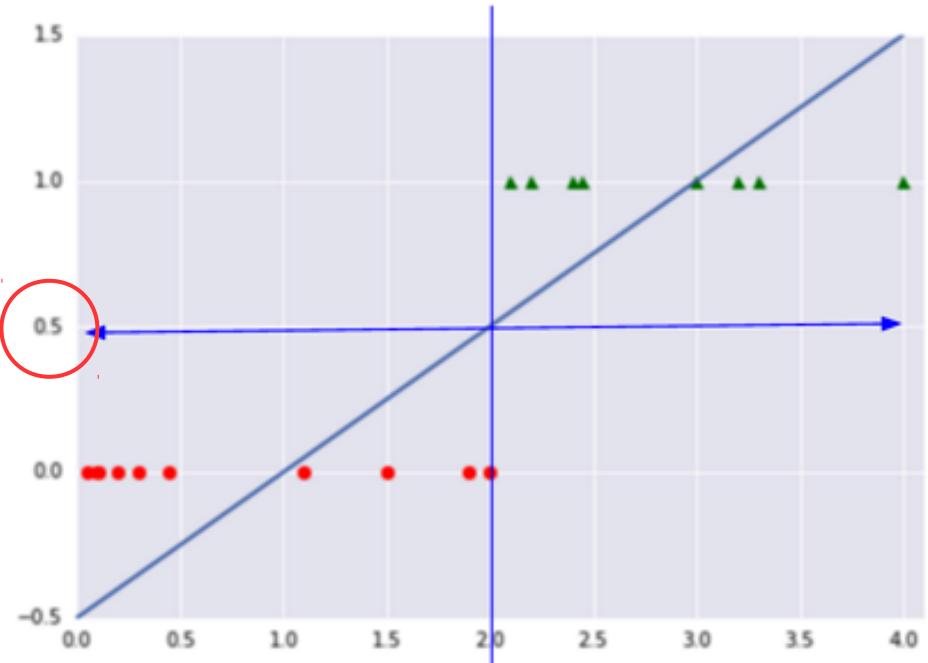
Sex	Age	Diabetes	...	Life span
Male	20	Yes	...	65
Female	40	Yes	...	68
Male	60	No	...	64



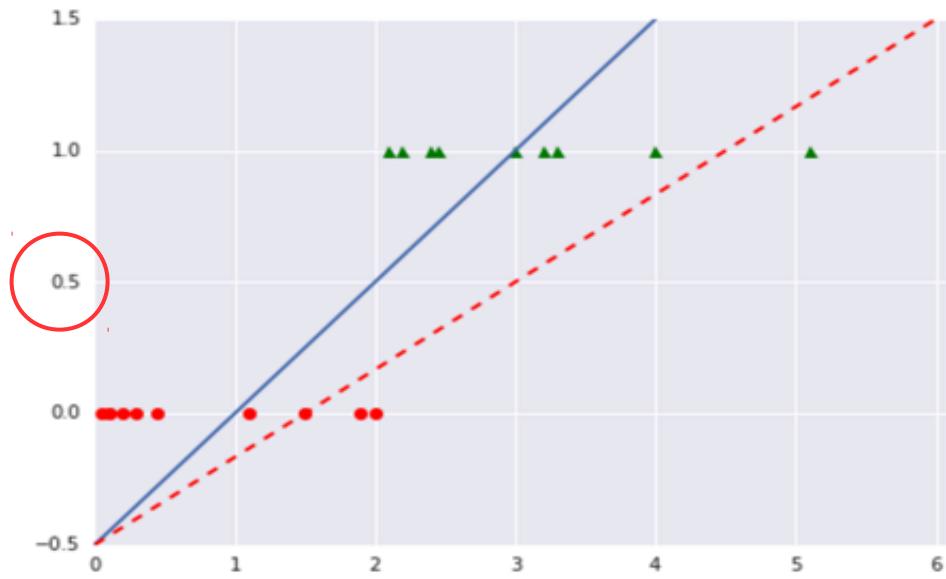
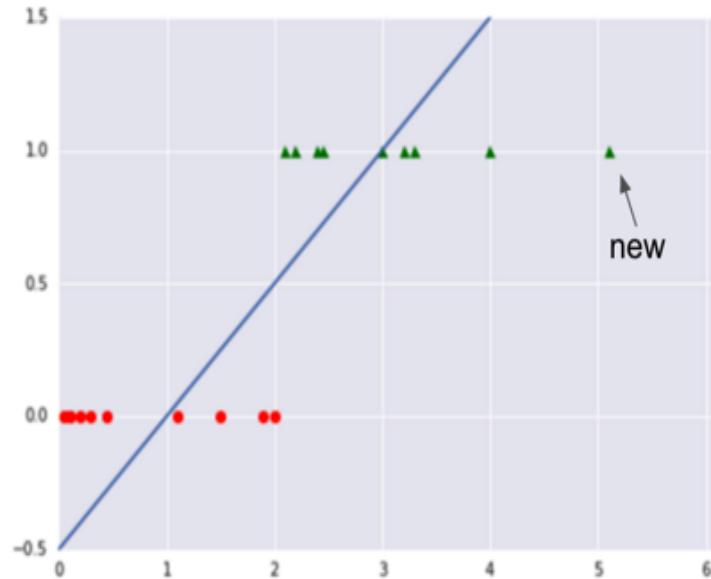
Logistic Regression(Supervised)



Linear Regression?



Logistic Regression(Supervised)



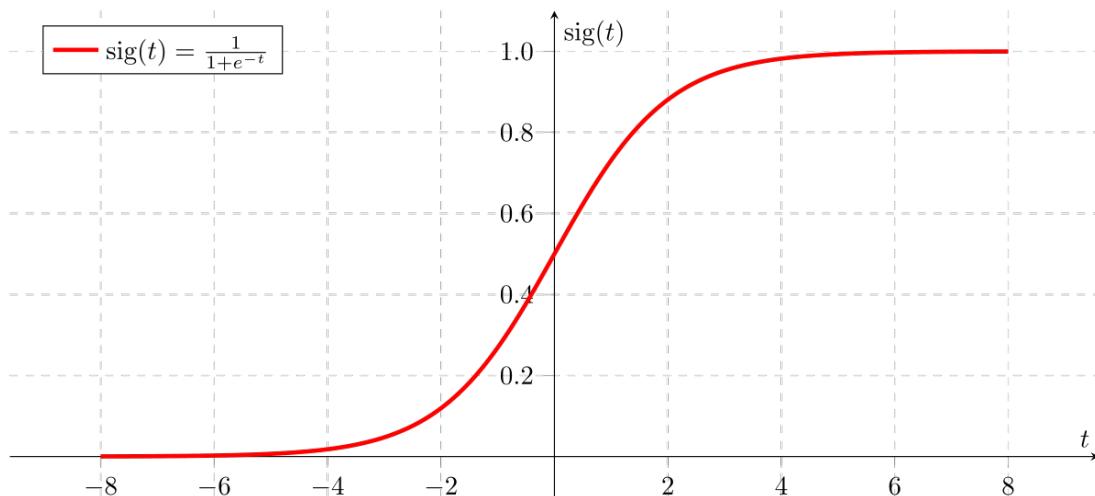
Linear regression for classification?

Problem

1) Threshold

2) Range

Logistic Regression(Supervised)



$$t = a^*X + b$$
$$h = \text{sigmoid}(t)$$

Q. 양성 VS 음성 ?

$$h = p(Y=\text{양성} | X=\text{데이터})$$

If $h > 0.5 \rightarrow \text{양성}$

If $h < 0.5 \rightarrow \text{음성}$

\downarrow

$$t > 0 \rightarrow \text{양성 } (y=1)$$
$$t < 0 \rightarrow \text{음성 } (y=0)$$


Logistic Regression(Supervised)

- 가설 (hypothesis)

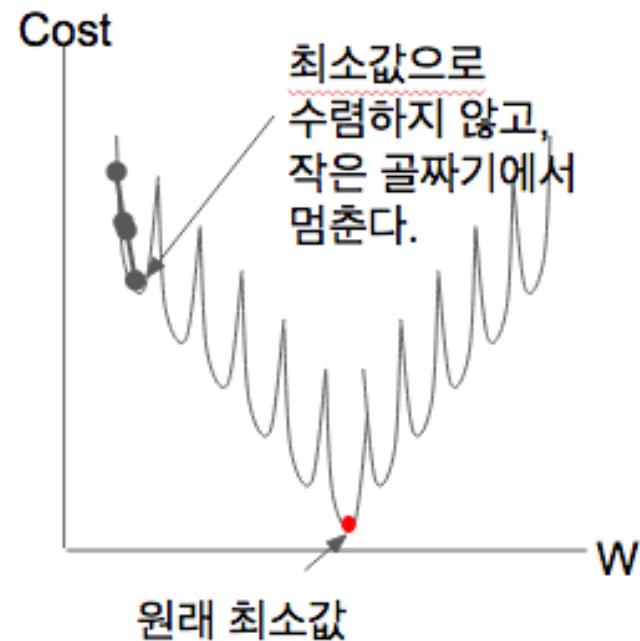
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- 비용 함수 (cost function)

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



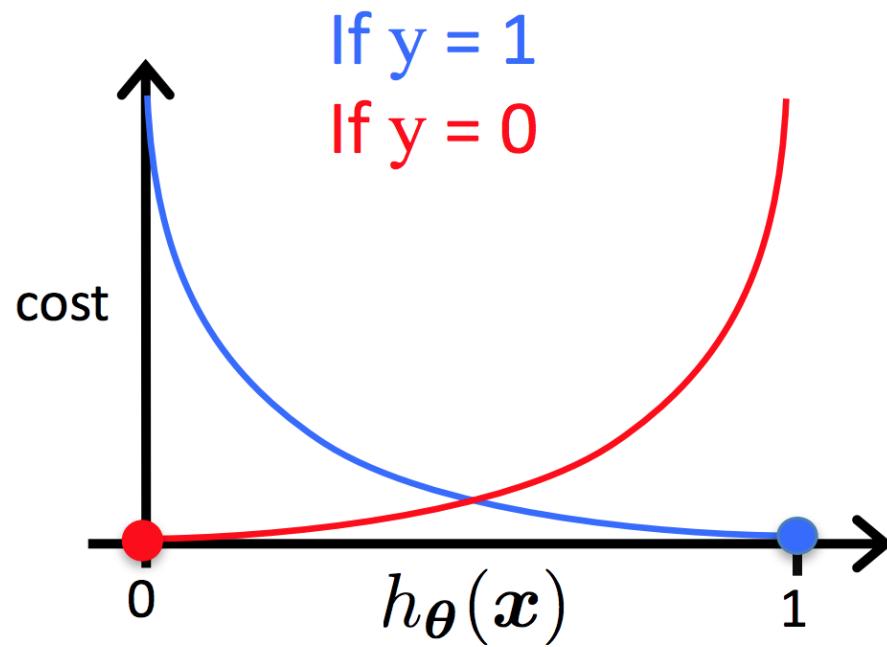
새로운 비용 함수 필요



Logistic Regression(Supervised)

- 비용 함수 (cost function)

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



Logistic Regression(Supervised)

- 비용 함수 (cost function)

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$
$$-\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)})) \right]$$

비용 함수 (오차) 를 최소화 하는 파라미터 (θ) 를 찾는다 .

→ 경사 하강법 (Gradient descent)

Calculating Gradient

$$h_{\theta}(x^{(i)}) = \frac{1}{1+e^{-(\theta^T x^{(i)} + b)}}$$

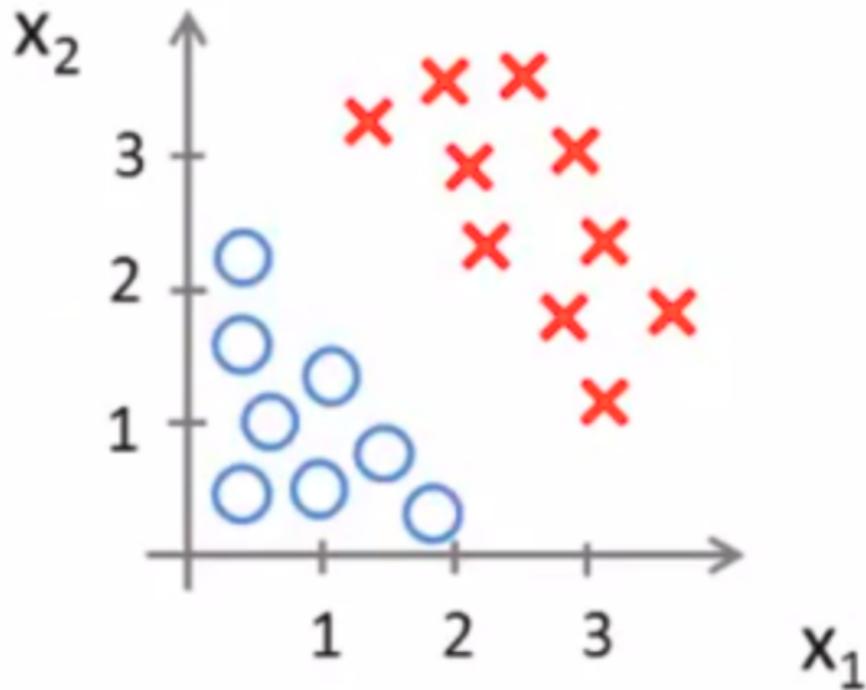
$$\begin{aligned}
J(\theta) &= -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] \\
&= \frac{1}{m} [\sum_{i=1}^m -y^{(i)} \log(\frac{1}{1+e^{-(\theta^T x^{(i)} + b)}}) - (1 - y^{(i)}) \log(1 - \frac{1}{1+e^{-(\theta^T x^{(i)} + b)}})] \\
&= \frac{1}{m} [\sum_{i=1}^m -y^{(i)} (\log 1 - \log(1 + e^{-(\theta^T x^{(i)} + b)})) - (1 - y^{(i)}) \log(\frac{e^{-(\theta^T x^{(i)} + b)}}{1+e^{-(\theta^T x^{(i)} + b)}})] \\
&= \frac{1}{m} [\sum_{i=1}^m y^{(i)} \log(1 + e^{-(\theta^T x^{(i)} + b)}) - (1 - y^{(i)}) \log(e^{-(\theta^T x^{(i)} + b)}) + (1 - y^{(i)}) \log(1 + e^{-(\theta^T x^{(i)} + b)})] \\
&= \frac{1}{m} [\sum_{i=1}^m (1 - y^{(i)}) (\theta^T x^{(i)} + b) + \log(1 + e^{-(\theta^T x^{(i)} + b)})] \\
&= \frac{1}{m} [\sum_{i=1}^m \log e^{\theta^T x^{(i)} + b} - y^{(i)} (\theta^T x^{(i)} + b) + \log(1 + e^{-(\theta^T x^{(i)} + b)})] \\
&= \frac{1}{m} [\sum_{i=1}^m \log(e^{\theta^T x^{(i)} + b} + 1) - y^{(i)} (\theta^T x^{(i)} + b)]
\end{aligned}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} [\sum_{i=1}^m \frac{x_j e^{\theta^T x^{(i)} + b}}{e^{\theta^T x^{(i)} + b} + 1} - y^{(i)} x_j] = \frac{1}{m} x_j \sum_{i=1}^m [\frac{1}{1+e^{-(\theta^T x^{(i)} + b)}} - y^{(i)}] = \frac{1}{m} x_j \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}]$$

$$\frac{\partial J(\theta)}{\partial b} = \frac{1}{m} [\sum_{i=1}^m \frac{e^{\theta^T x^{(i)} + b}}{e^{\theta^T x^{(i)} + b} + 1} - y^{(i)}] = \frac{1}{m} \sum_{i=1}^m [\frac{1}{1+e^{-(\theta^T x^{(i)} + b)}} - y^{(i)}] = \frac{1}{m} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}]$$



Decision Boundary



$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

predict " $y = 1$ " if $-3 + x_1 + x_2 \geq 0$
 $\Leftrightarrow x_1 + x_2 \geq 3$

the line $x_1 + x_2 = 3$ we call that "Decision boundary"