강의 및 자료 출처 : https://youtu.be/nQqMT0wbthw

Standalone Deep-Learning

#2 ML Basic

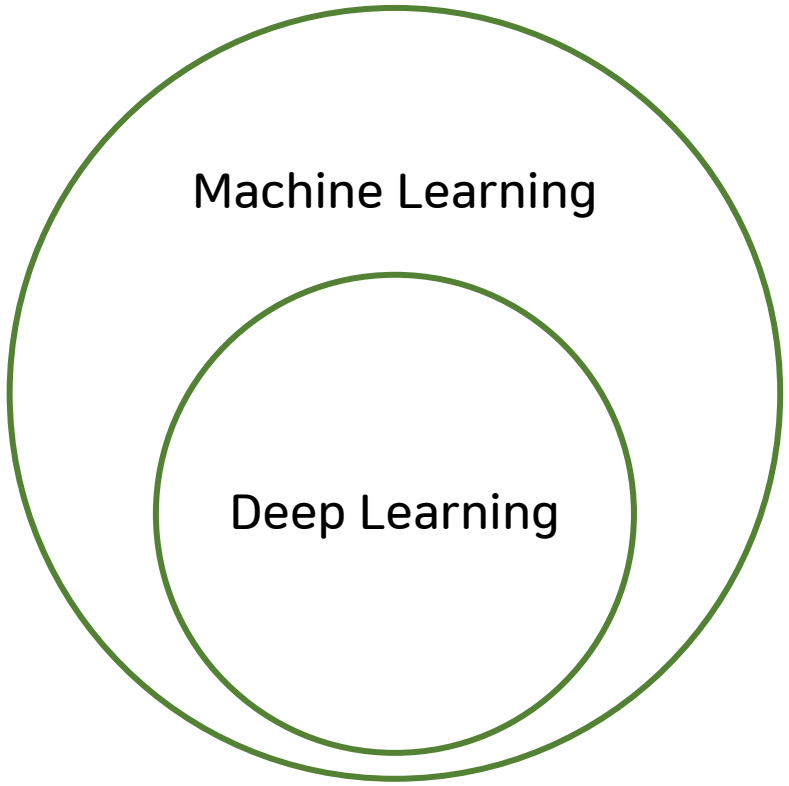
#3 Linear Regression

#4 Linear Regression Practice

1주차_21.07.13

곽민지

Contents



- 머신러닝(Machine Learning)

  : AI의 한 종류.
  학습을 통해 특정 업무를 실행할 수 있는 AI.
  주로 사람이 특징을 정의.

- 딥러닝(Deep Learning)

  : 머신러닝의 한 종류.
  인간 뇌의 신경세포(뉴런)를 따라한 학습법에서 발전.
  주로 기계가 특징을 자동으로 정의.

출처 : 테크월드뉴스(http://www.epnc.co.kr)

- 머신러닝(Machine Learning)

- 딥러닝(Deep Learning)

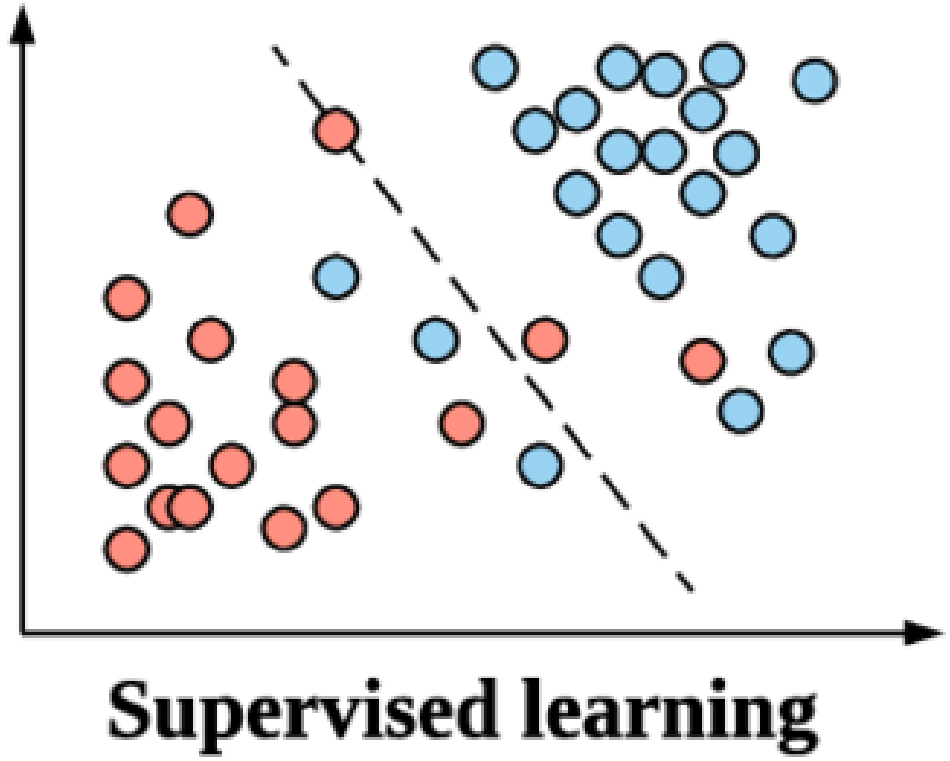
출처 : https://modern-manual.tistory.com/3

	Supervised	Unsupervised	Reinforcement
Discrete	Classification	Clustering	Discrete Action Space Agent
Continuous	Regression	Dimensionality Reduction	Continuous Action Space Agent

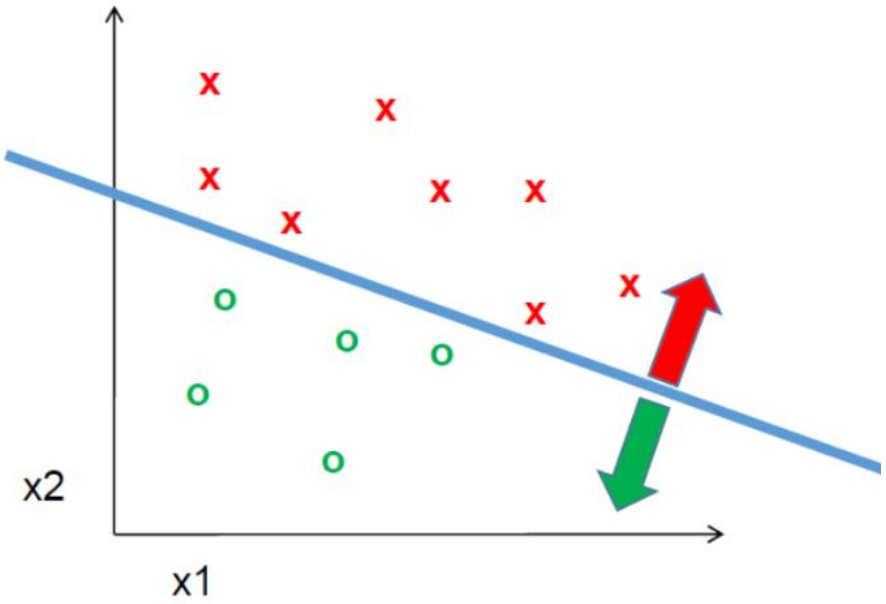
Semi-Supervised Learning

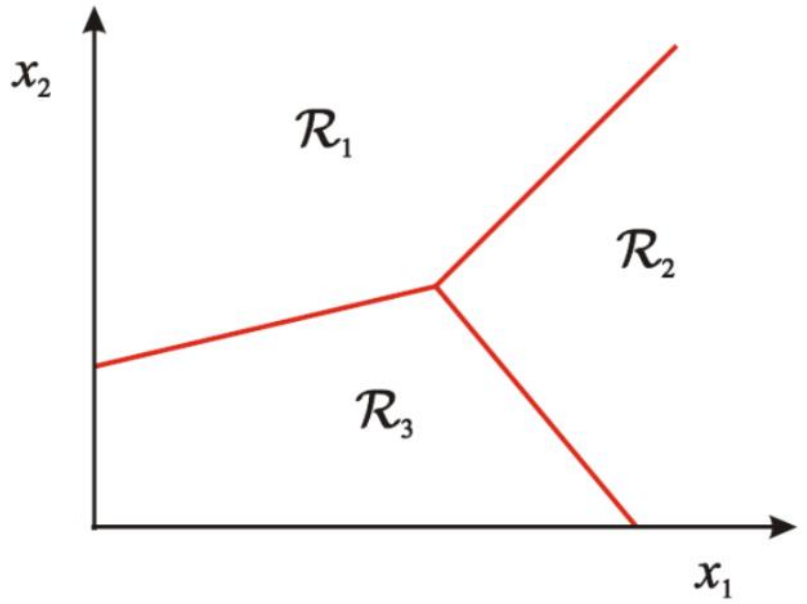
- Supervised Learning

	Supervised
Discrete	Classification
Continuous	Regression

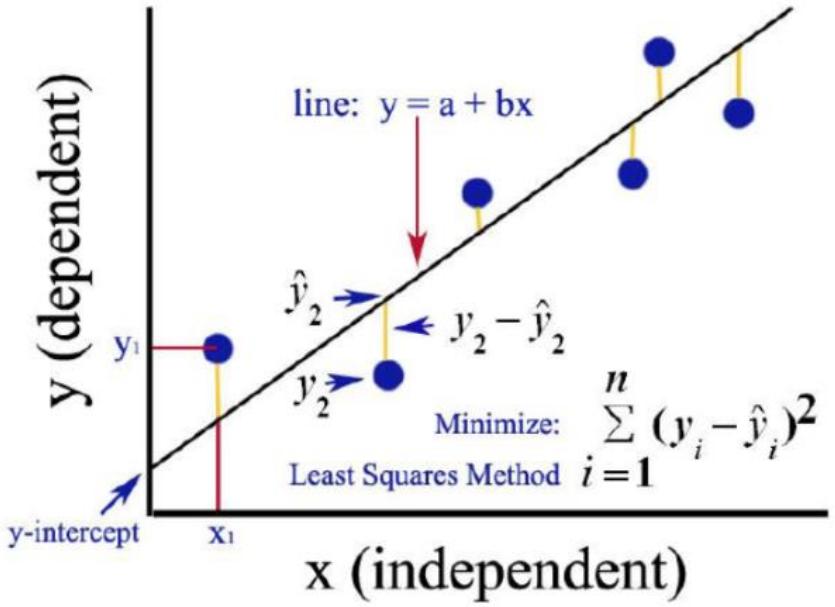


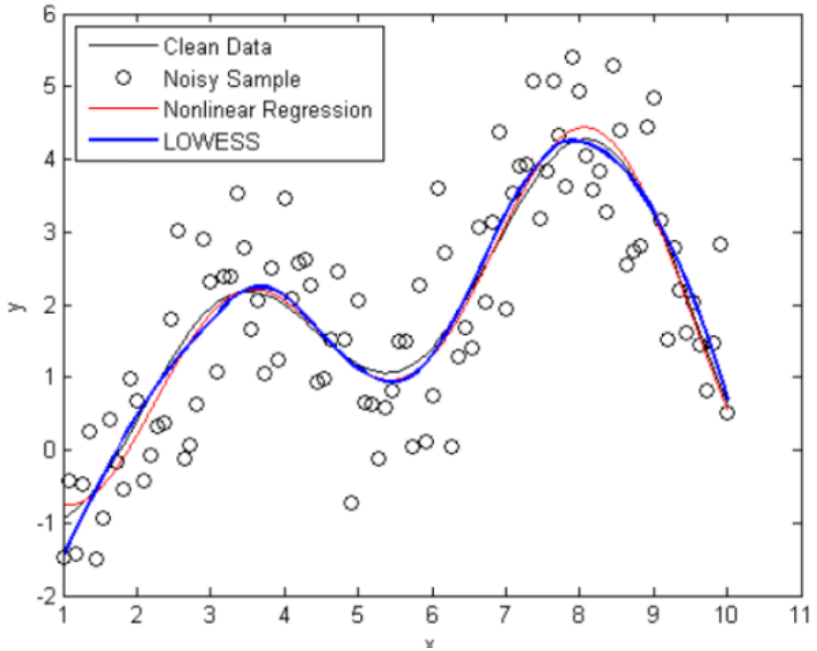
- Classification





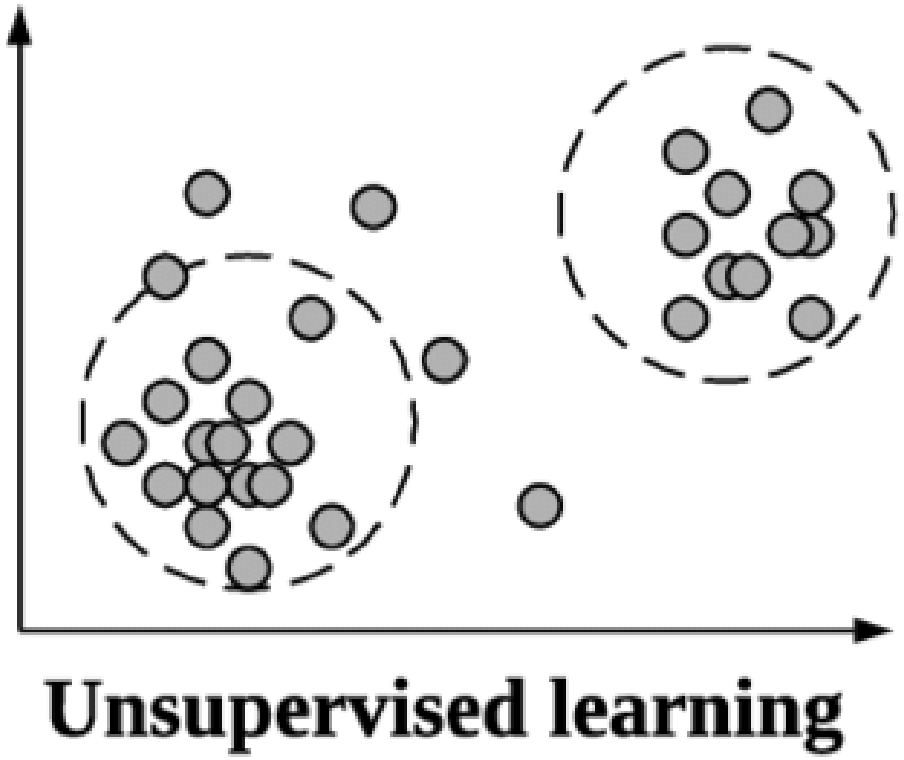
- Regression


line: y = a + bx
y (dependent)
x (independent)
y-intercept
$\hat{y}_2$
$y_2 - \hat{y}_2$
y_2
Minimize: $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
Least Squares Method

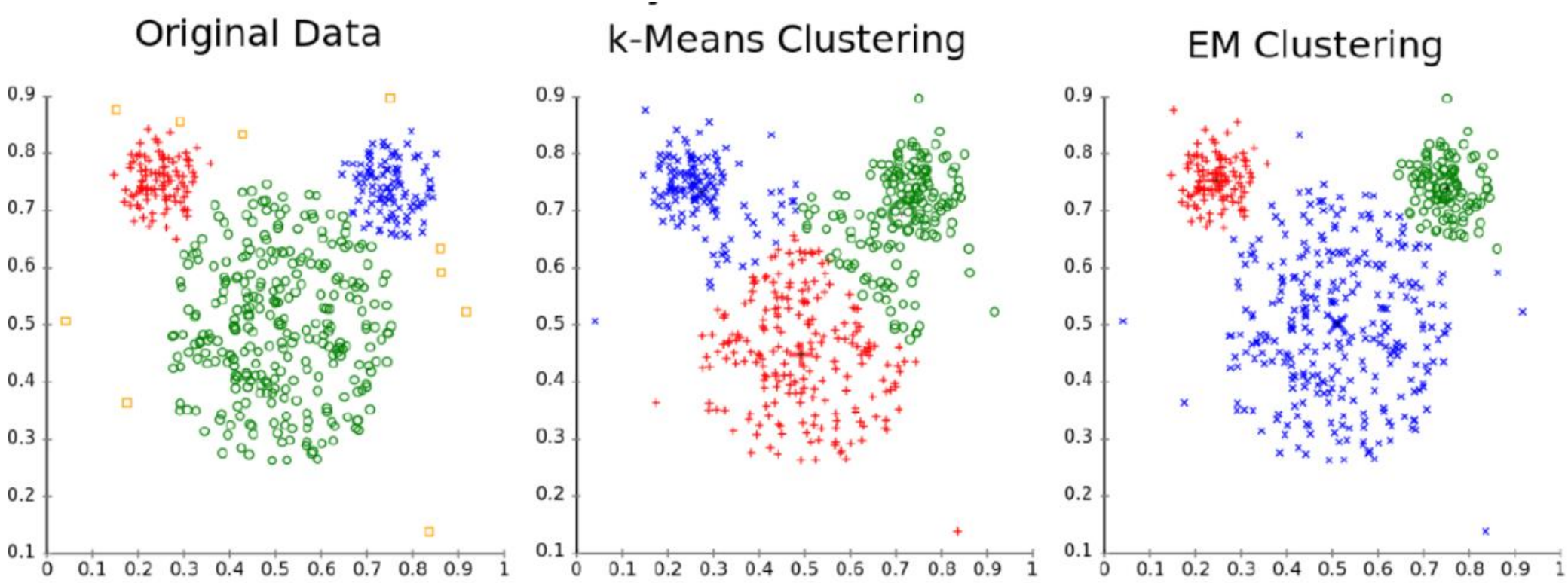

Clean Data
Noisy Sample
Nonlinear Regression
LOWESS

- Unsupervised Learning

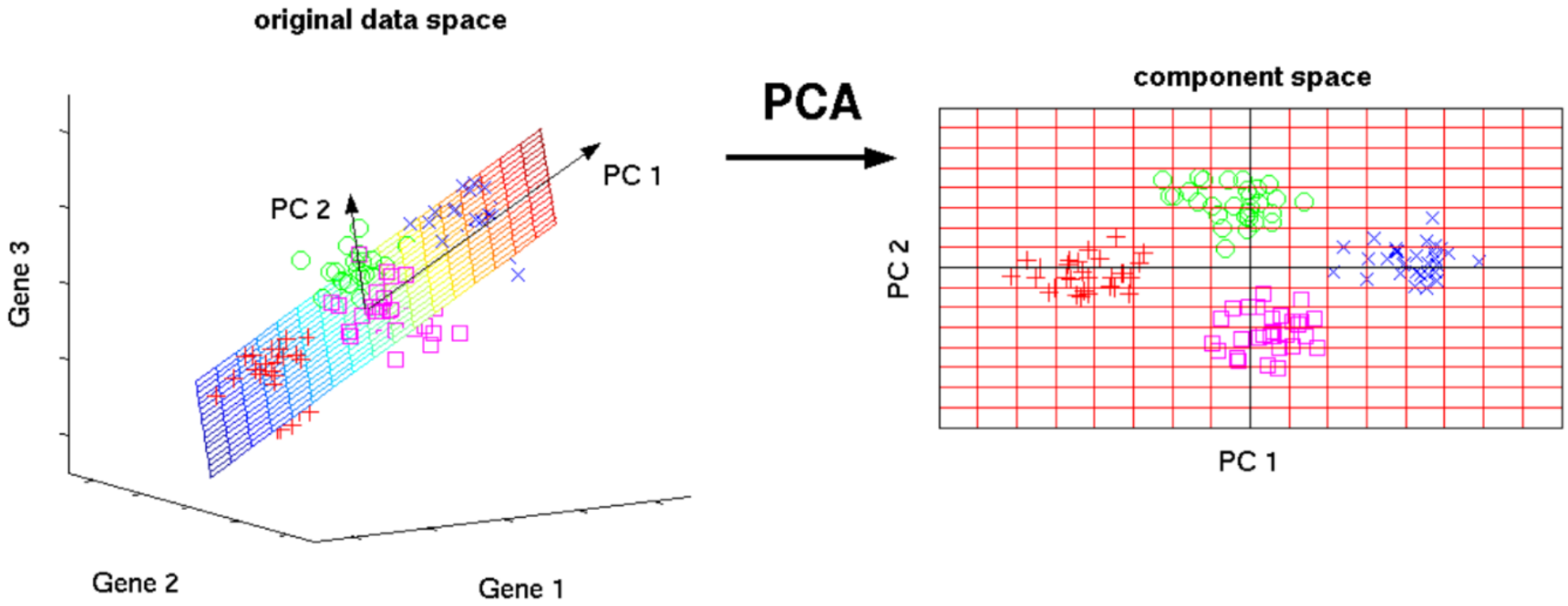
	Unsupervised
Discrete	Clustering
Continuous	Dimensionality Reduction



- Clustering


Original Data
k-Means Clustering
EM Clustering
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

- Dimensionality Reduction

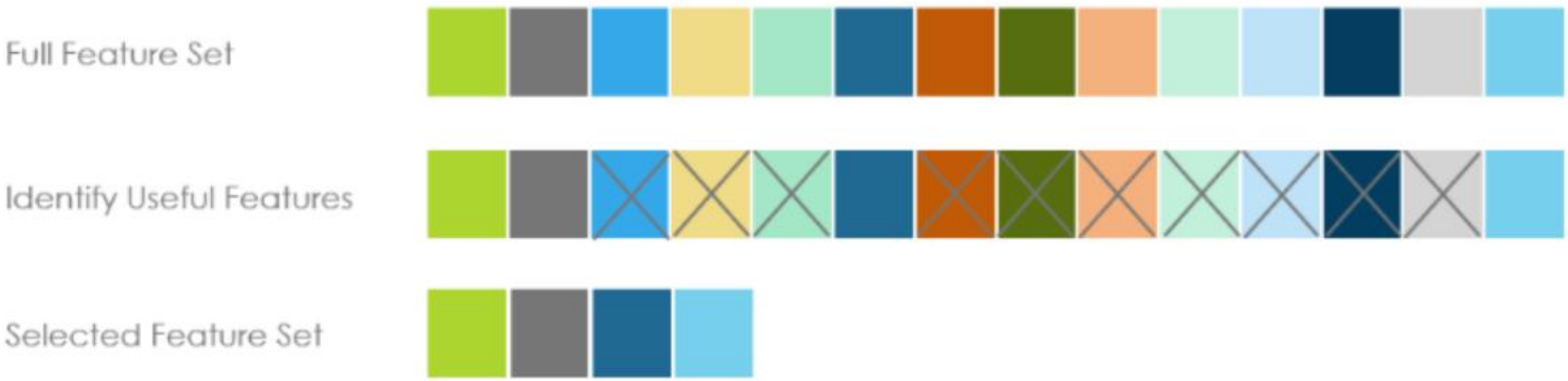


- Dimensionality Reduction

Feature selection	Feature extraction
Feature 중에서 필요한 것만 선택	원본 feature의 조합으로 새로운 feature 생성
Genetic Algorithms	PCA, LDA

- Feature selection

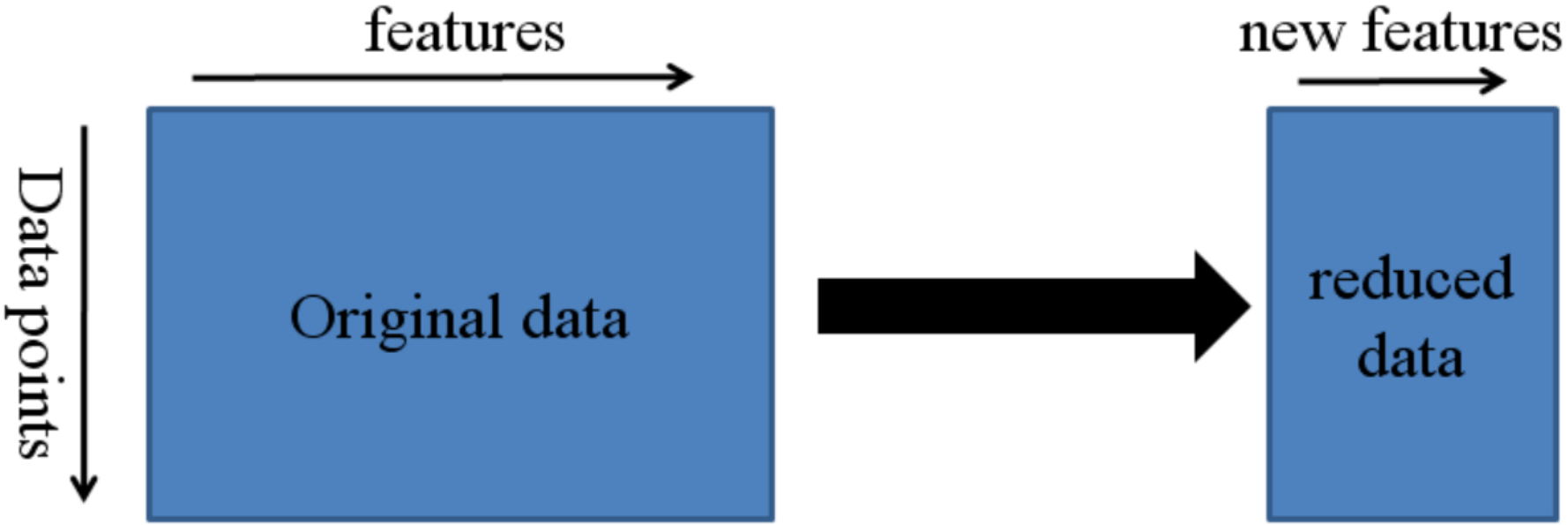
  → Feature 중에서 필요한 것만 선택



출처 : https://bioinformaticsandme.tistory.com/188

- Feature extraction

  → 원본 feature의 조합으로 새로운 feature 생성

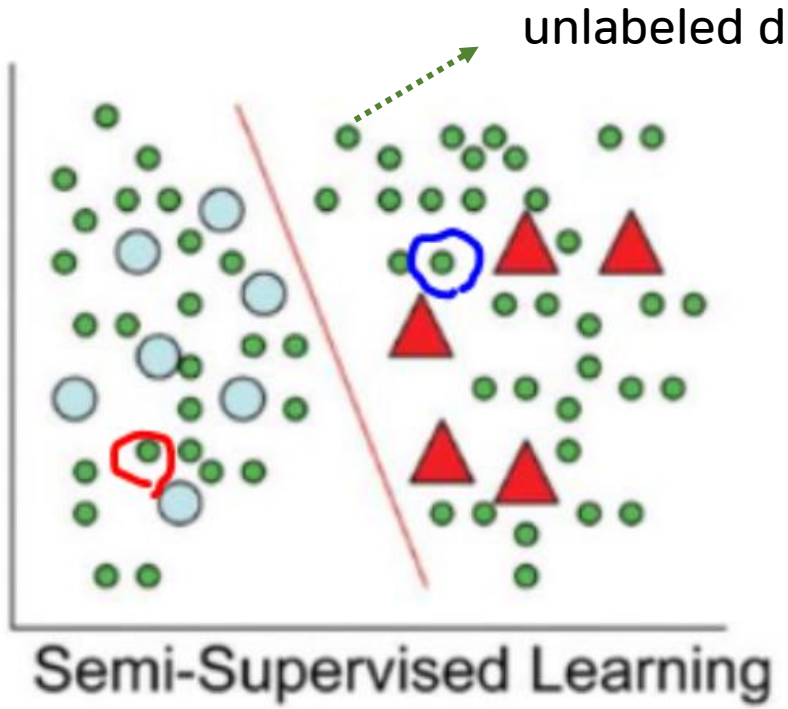


출처 : https://bioinformaticsandme.tistory.com/188

- Semi-Supervised Learning

	Supervised	Unsupervised
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality Reduction

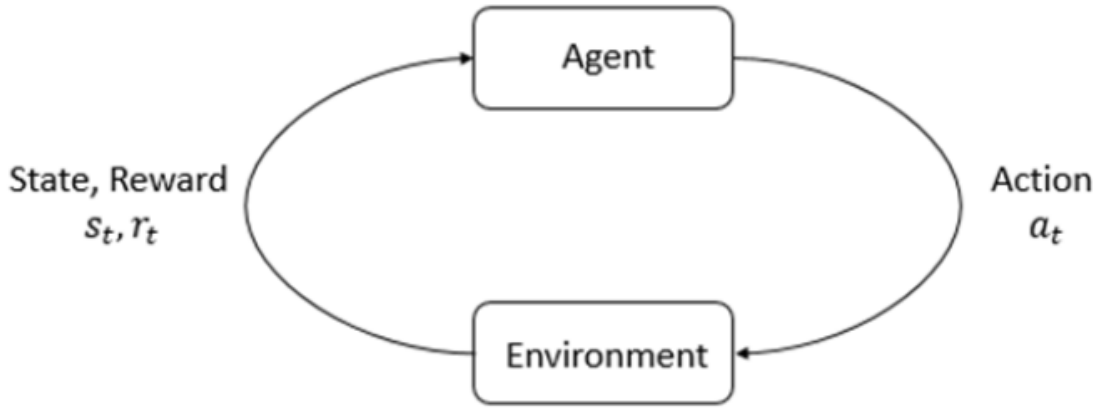
Semi-Supervised Learning



출처 : https://techblog-history-younghunjo1.tistory.com/82

- Reinforcement Learning

	Reinforcement
Discrete	Discrete Action Space Agent
Continuous	Continuous Action Space Agent



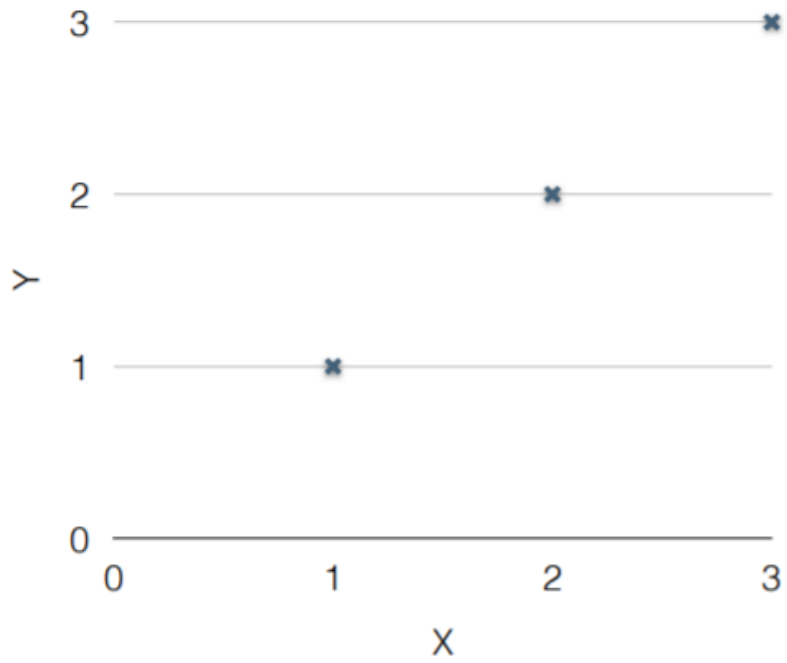
출처 : https://www.secmem.org/blog/2019/12/15/RL-key-concepts/

Linear Regression

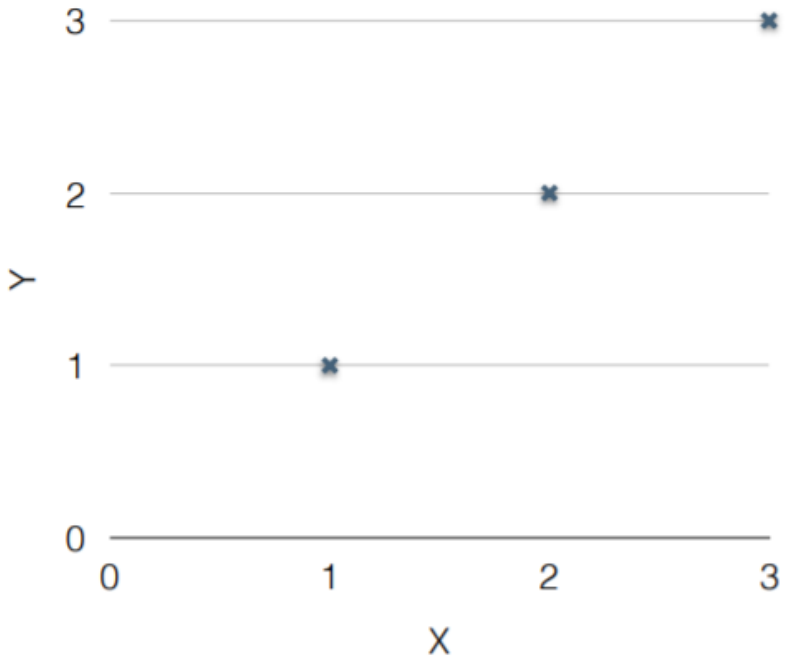
- Regression

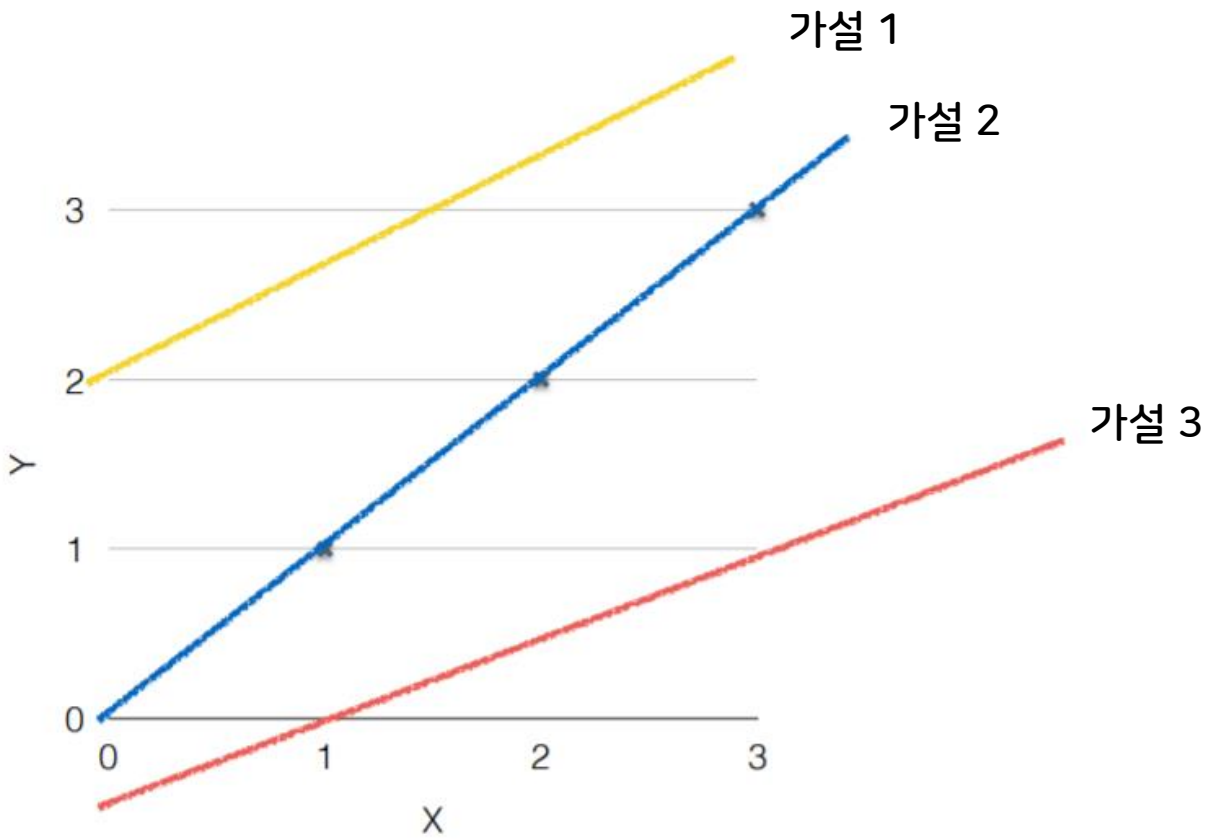
  : 독립변수 x로 종속변수 y를 예측

X	Y
1	1
2	2
3	3



Linear Regression

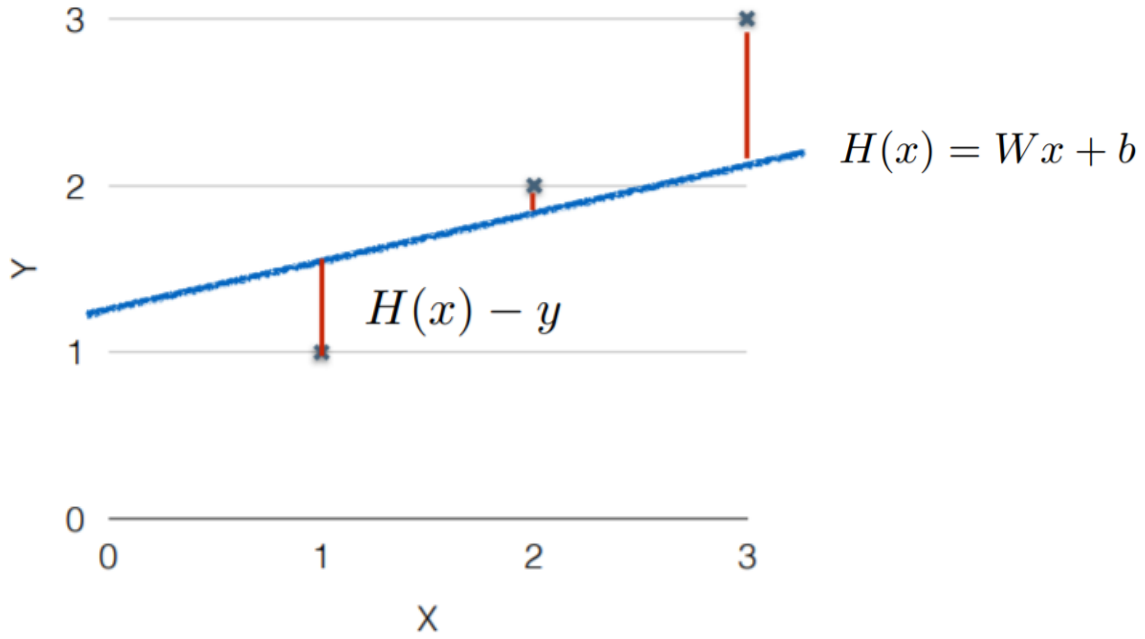




$$H(x) = Wx + b$$

Linear Regression

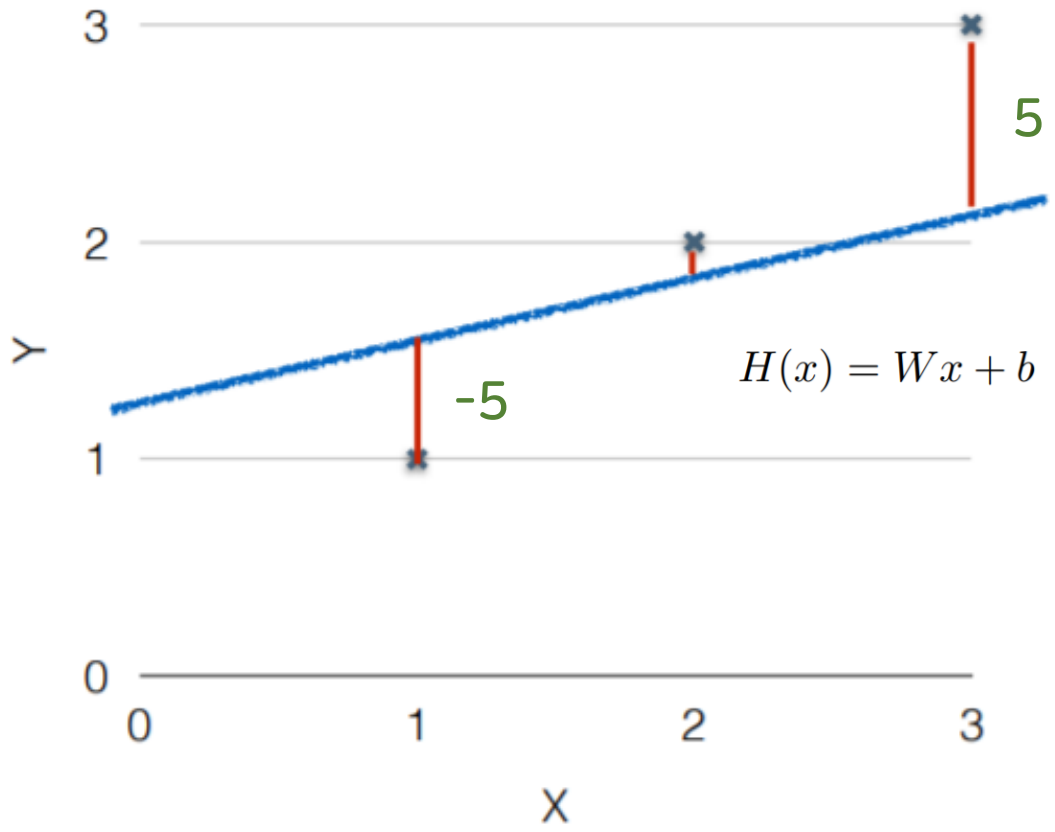
cost function



$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

Linear Regression

cost function



$$H(x) - y \quad \times$$

$$(H(x^{(i)}) - y^{(i)})^2 \quad \bigcirc$$

Linear Regression

- loss function

  : 하나의 input data에 대해서 오차를 계산하는 함수

  → single dataset

$$(H(x^{(i)}) - y^{(i)})^2$$

- cost function

  : 모든 data에 대한 오차를 계산하는 함수

  → entire dataset

$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

Linear Regression

- Minimize Cost

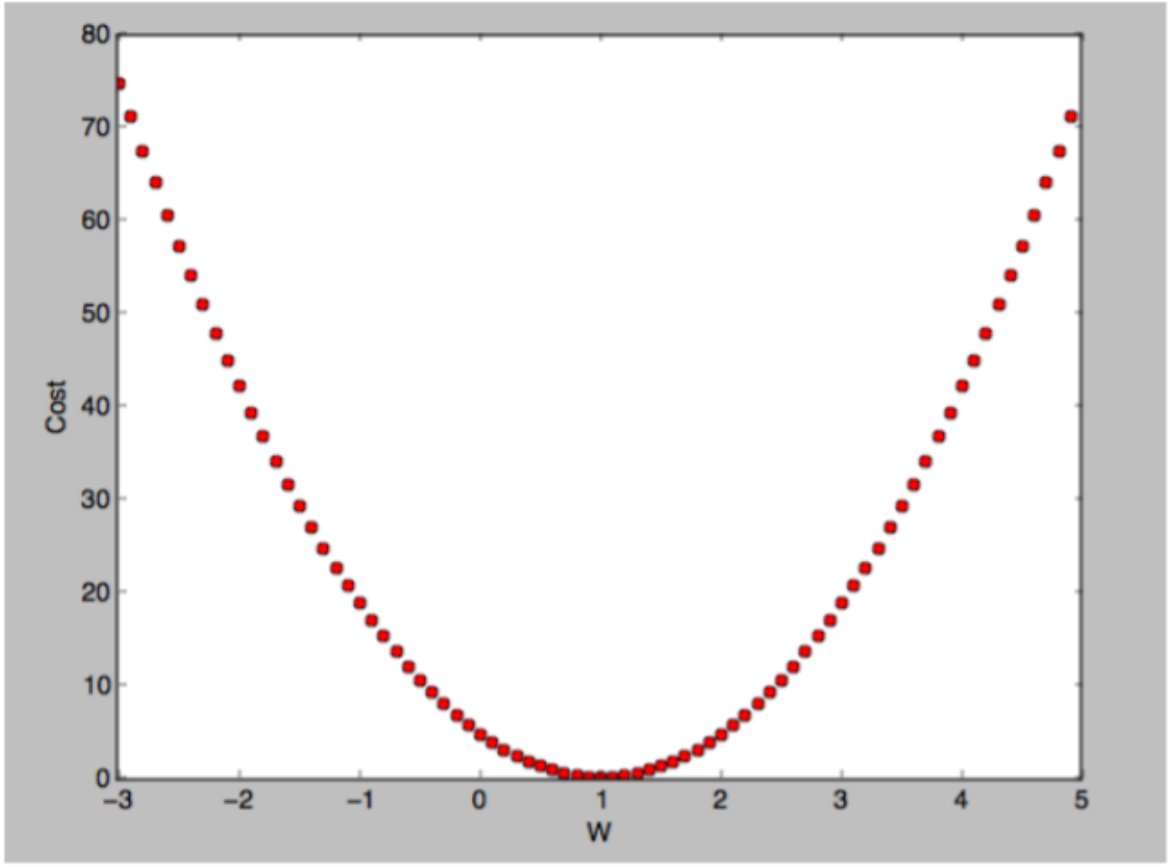
x	Y
1	1
2	2
3	3

$$H(x) = Wx$$

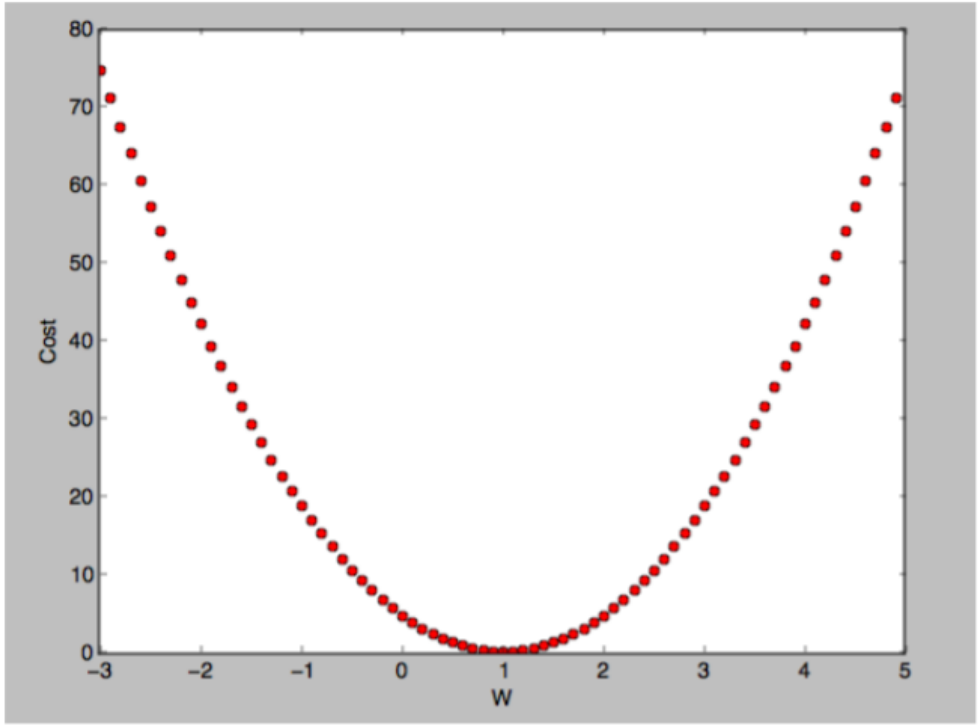
$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

- W=1, cost(W)=0
- W=0, cost(W)=4.67
- W=2, cost(W)=4.67

Linear Regression



Linear Regression



$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

Linear Regression

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$

Linear Regression Practice

- **cost function**

```
def better_cost(pred_y, true_y):
    error = 0
    for i in range(len(x)):
        error += (pred_y[i] - true_y[i])**2
    error = error / len(x)
    return error

pred_y = [h.forward(x[i]) for i in range(len(x))]
print('cost value with better code structure :', better_cost(pred_y, y))
```

```
cost value with better code structure : 222.2
```

Linear Regression Practice

- **gradient**

```
def cal_grad2(w, cost):
    h = H(w)
    grad = 0
    for i in range(len(x)):
        grad += 2 * (h.forward(x[i]) - y[i]) * x[i]
    grad = grad / len(x)
    c = cost(h, x, y)
    return grad, c
```

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$

Linear Regression Practice

- **gradient**

```
for i in range(100):
    grad, mean_cost = cal_grad(w1, cost)
    grad2, mean_cost2 = cal_grad2(w2, cost)

    w1 -= lr * grad
    w2 -= lr * grad2
    list_w1.append(w1)
    list_w2.append(w2)
    list_c1.append(mean_cost)
    list_c2.append(mean_cost2)
```

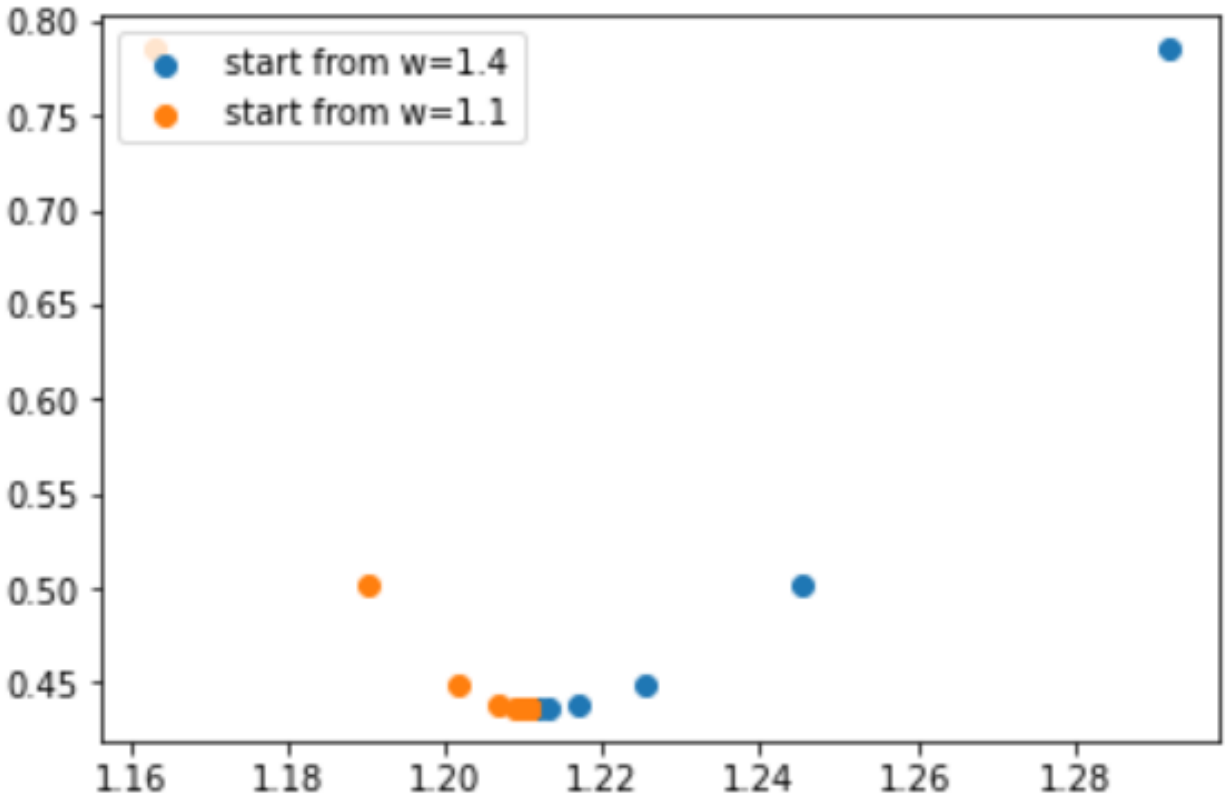
$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$

Linear Regression Practice

- start from w = ?



Q&A