

임상의를 위한 AI 교육 - 기초과정 2주차

머신 러닝의 개념과 실습

서울대학교병원 융합의학과 김영곤 교수

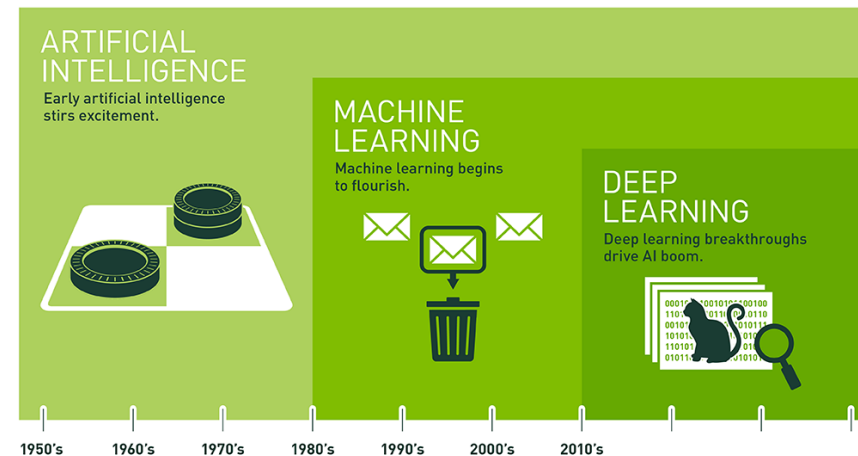
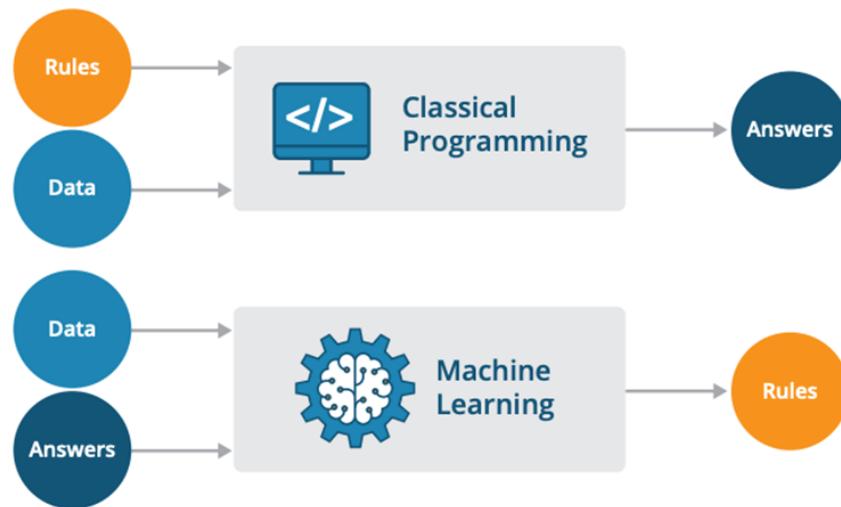


※ 본 수업자료는 “서울대학교병원 데이터사이언스연구부
AI지원실” 학습서기반으로 제작되었습니다.

Part 3. 머신 러닝

- **Machine Learning(기계 학습) :**

- Artificial Intelligence, **AI** : 인간의 학습능력, 추론능력, 지각능력을 인공적으로 구현하려는 컴퓨터과학의 세부분야 중 하나. (Wikipedia)
- Arthur Samuel (1959) – **Machine learning**: "Field of study that gives computers the ability to learn without being explicitly programmed"
- Traditional(Explicit) Programming vs. Machine Learning



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

• 머신 러닝의 종류

• 지도 학습 (Supervised Learning)

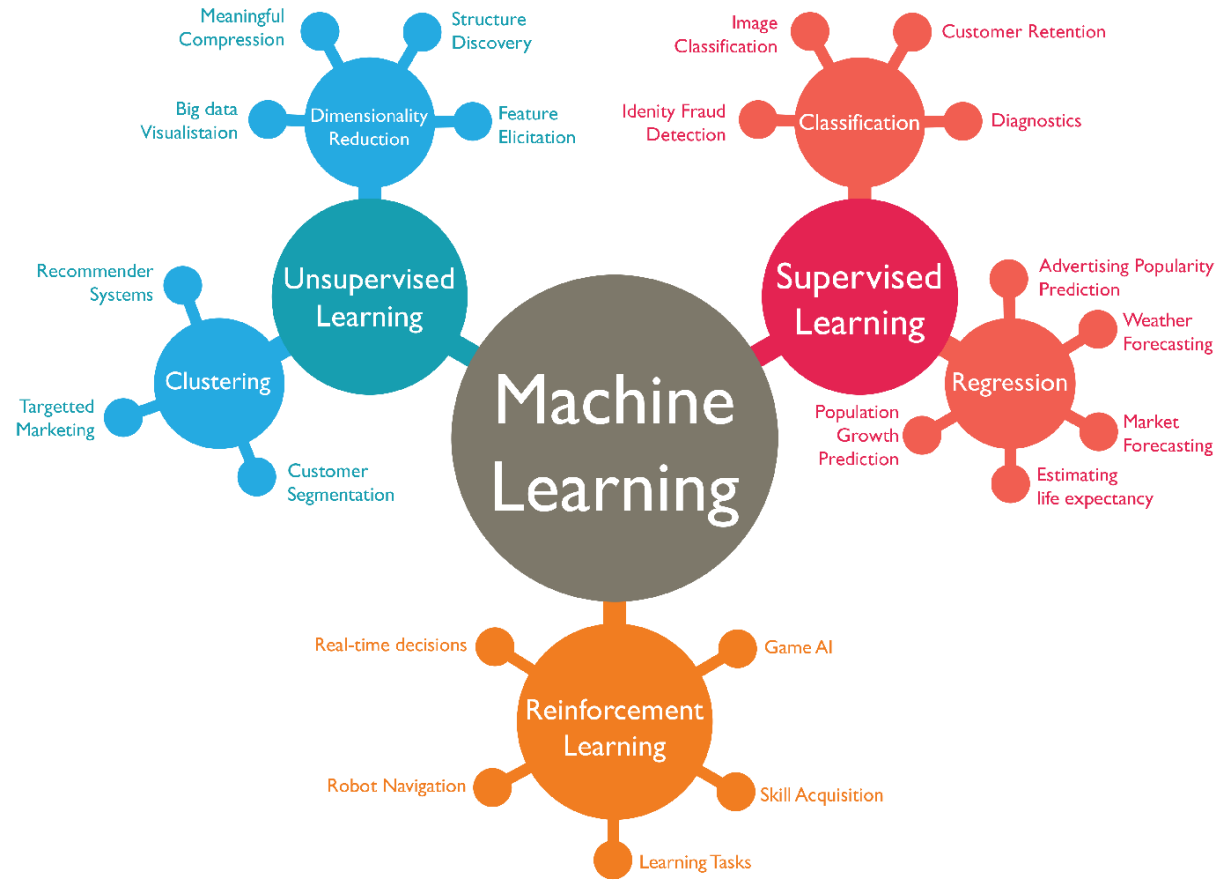
- 정답이 있는 훈련 데이터(Training Data)를 활용하여 학습하는 방법

• 비지도 학습 (Unsupervised Learning)

- 정답이 없는 데이터의 특징을 추출하고 설명하는 방법

• 강화 학습 (Reinforcement Learning)

- 주어진 문제 상황에서 행동을 통해 보상을 얻으며 학습하는 방법



1. 지도 학습 머신 러닝 Master algorithms

3-1-1

Linear Regression

- 선형회귀 (Linear Regression)

- K-NN

X : continuous, Y : categorical

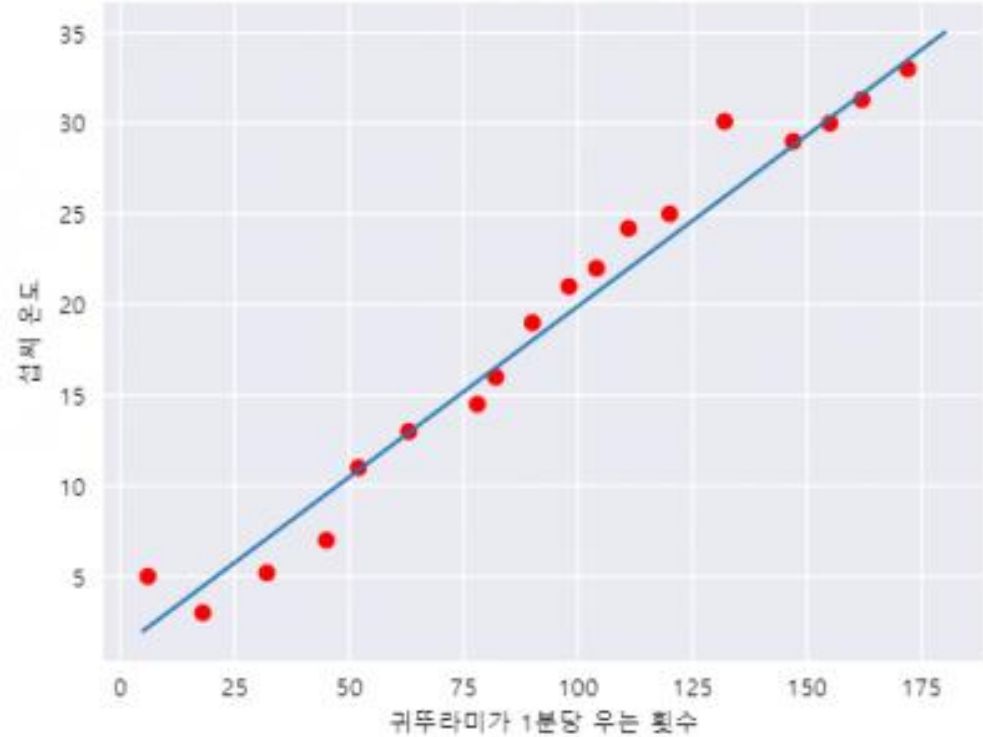
∴ "분류 (Classification)"

- 선형회귀

X : continuous, Y : **continuous**

∴ "회귀 (Regression)"

- 즉, x 와 y 의 선형 관계를 모델링.



$$y = wx + b$$

1. 지도 학습 머신 러닝 Master algorithms

3-1-2

Linear Regression

• 선형회귀 (Linear Regression)

- x 와 y 의 관계를 모델링
- 즉, 독립변수에 따른 종속변수의 변화를 설명

$$y = wx + b$$

↓ ↓
종속변수 독립변수

$$y = wx + b$$

↓ ↓
1차함수 : 기울기 절편
ML : 가중치 편향
 weight bias

- 단순 선형 회귀(Simple Linear Regression)

$$y = wx + b$$

- 다중 선형 회귀(Multiple Linear Regression)

$$y = w_1x_1 + w_2x_2 + \cdots w_nx_n + b$$

1. 지도 학습 머신 러닝 Master algorithms

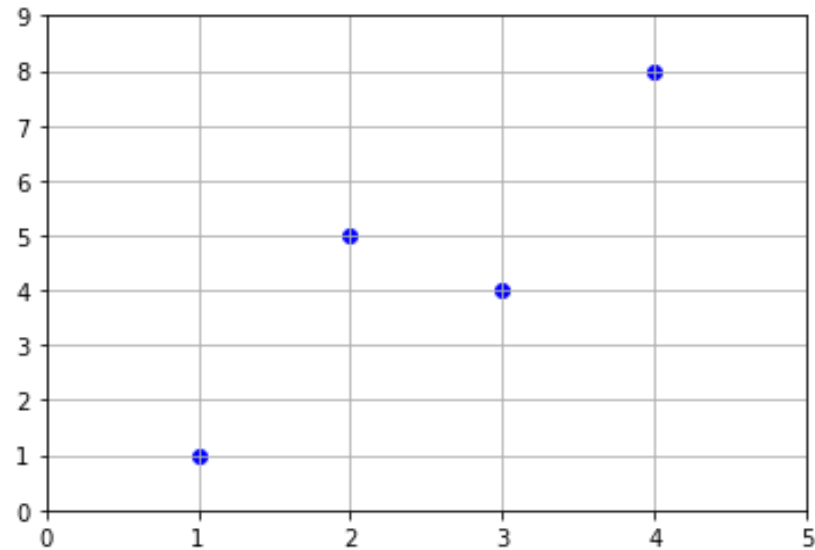
3-1-3

Linear Regression

- 선형회귀 (Linear Regression)

- 예제)

	x	y
0	1	1
1	2	5
2	3	4
3	4	8



1. 지도 학습 머신 러닝 Master algorithms

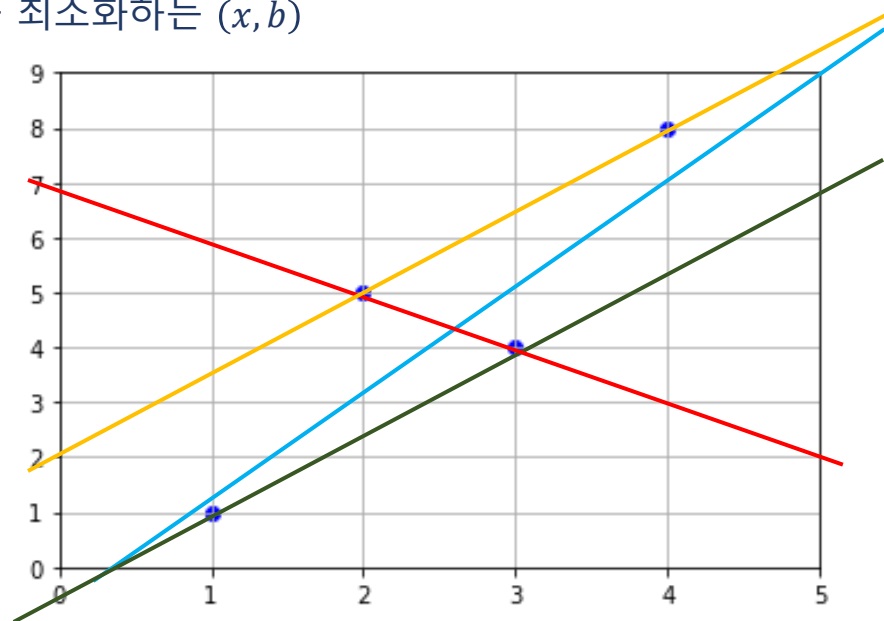
3-1-4

Linear Regression

• 선형회귀 (Linear Regression)

• 최적의 모델 → 최적의 (x, b) → 오차를 최소화하는 (x, b)

	x	y
0	1	1
1	2	5
2	3	4
3	4	8



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

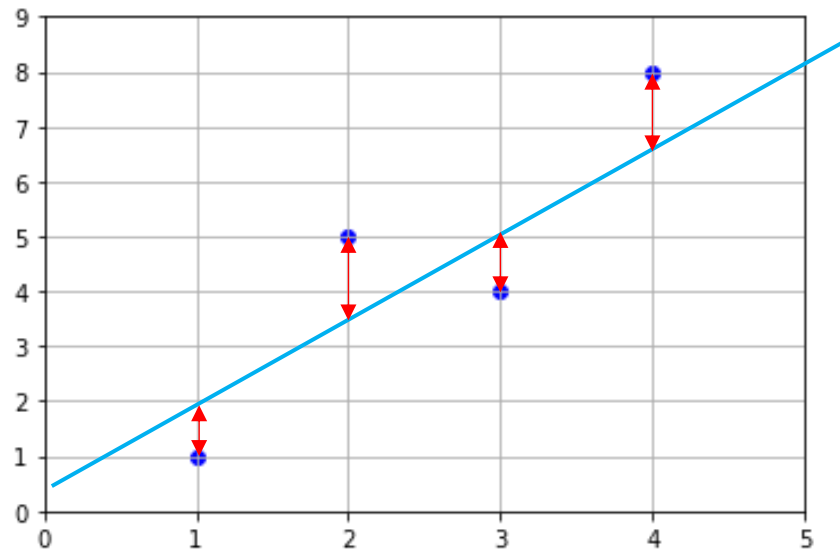
1. 지도 학습 머신 러닝 Master algorithms

3-1-5

Linear Regression

• 선형회귀 (Linear Regression)

- 오차 (Cost, Loss, 손실) : 예측값과 실제값의 차이 $H(x) - y$



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

$$cost = \frac{1}{n} \sum_{i=1}^n \{H(x_i) - y_i\}^2$$

→ 오차 함수(Cost Function)
= 손실 함수(Loss Function)

(MSE, MAE, RMSE, R-squared, ...)

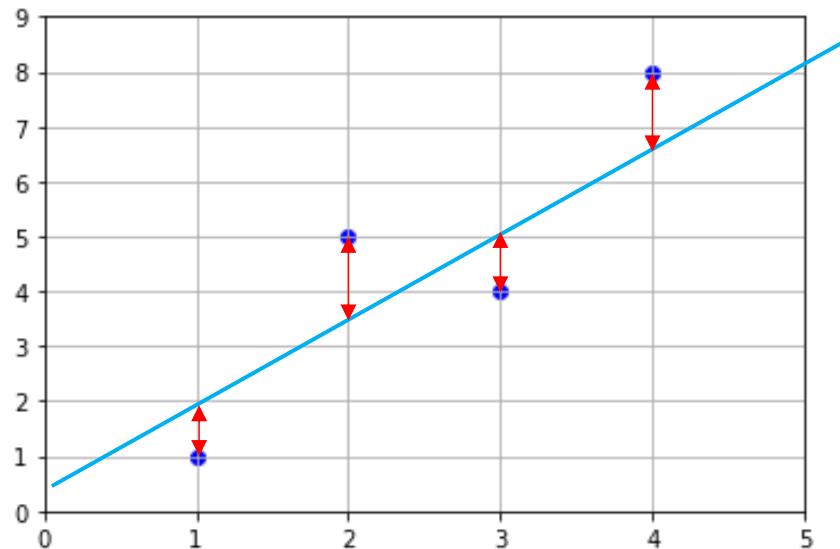
1. 지도 학습 머신 러닝 Master algorithms

3-1-6

Linear Regression

• 선형회귀 (Linear Regression)

- 오차 (Cost, Loss, 손실) : 예측값과 실제값의 차이 $H(x) - y$



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

“오차를 최소화하는 최적의 모델”

minimize cost (where $cost = \frac{1}{n} \sum_{i=1}^n \{H(x_i) - y_i\}^2$)

→ 옵티마이저(Optimizer)

← 훈련/학습
(Learning/Training)

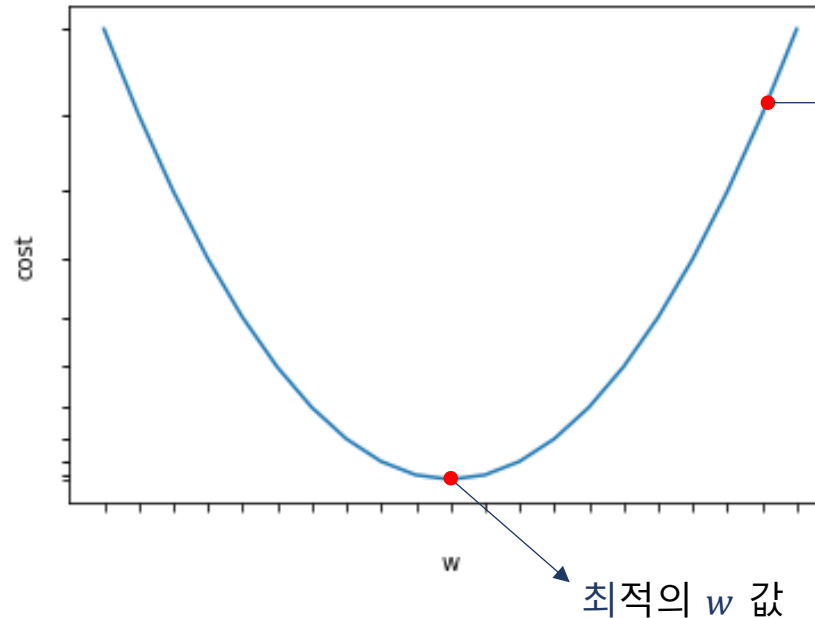
1. 지도 학습 머신 러닝 Master algorithms

3-1-7

Linear Regression

• 경사 하강법 (Gradient Descent)

- Optimizer의 한 방법
- 일반적인 Cost function



→ 최초의 w 값 (randomized)

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

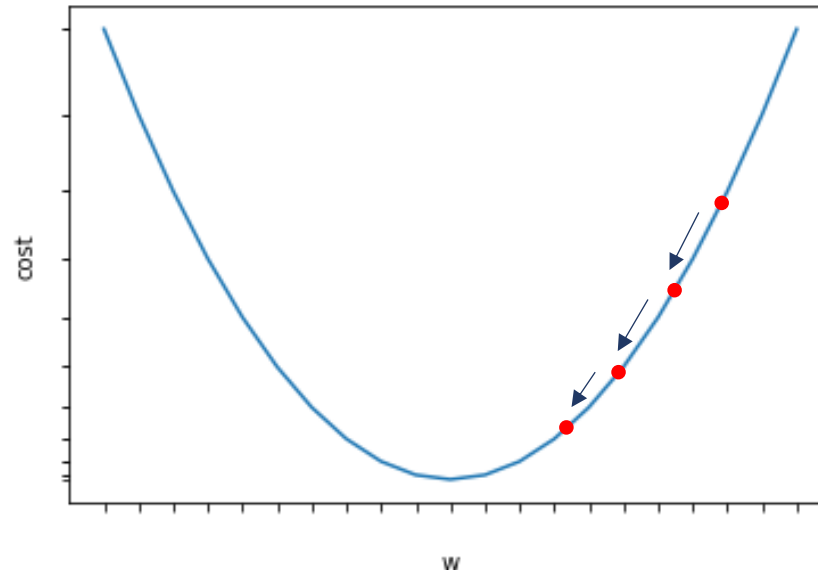
1. 지도 학습 머신 러닝 Master algorithms

3-1-8

Linear Regression

• 경사 하강법 (Gradient Descent)

- Optimizer의 한 방법
- 일반적인 Cost function



- Gradient descent 알고리즘

$$w := w - \alpha \frac{\partial}{\partial w} \text{cost}(w, b)$$

↓
학습률
Learning rate

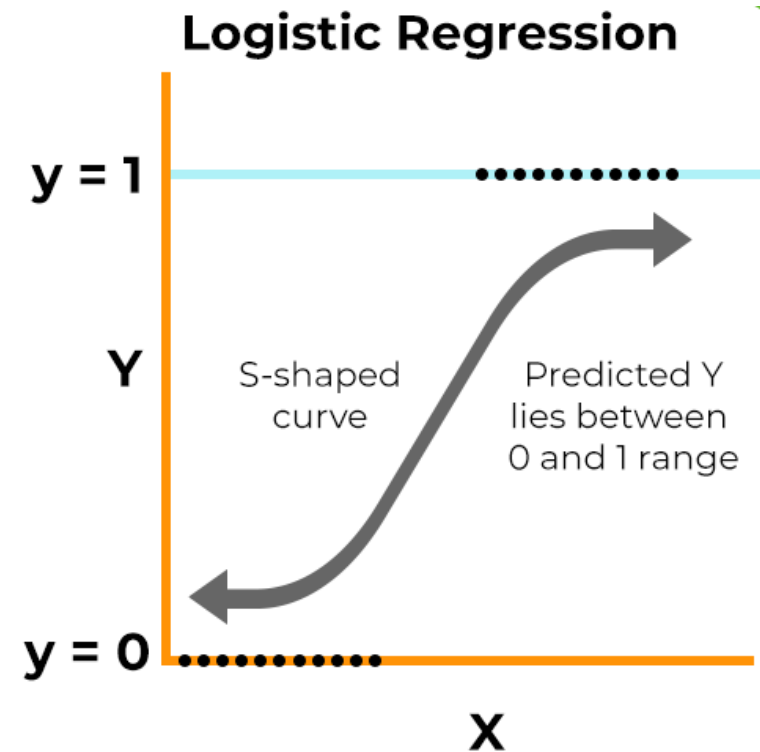
1. 지도 학습 머신 러닝 Master algorithms

3-2-1

Logistic Regression

• 로지스틱 회귀 (Logistic Regression)

- K-NN : 분류(Classification) 문제
 - Ex) 공부시간(X)에 따른 등급(Y) 예측 (A, B, C, D, F)
- Linear Regression : 회귀(Regression) 문제
 - Ex) 공부시간(X)에 따른 점수(Y) 예측 (0~100)
- Logistic Regression : **이진 분류(Binary Classification)** 문제
 - Ex) 공부시간(X)에 따른 합격 여부(Y) 예측 (Pass / Fail)



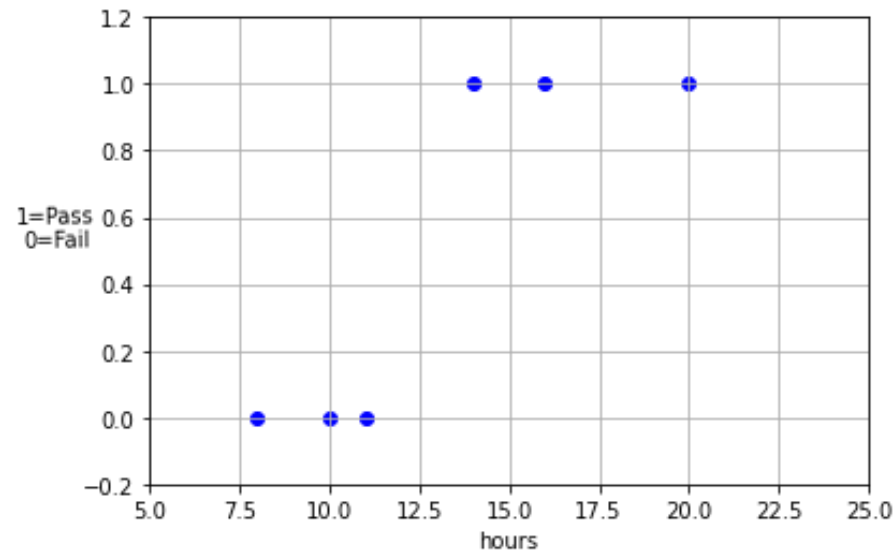
1. 지도 학습 머신 러닝 Master algorithms

3-2-2

Logistic Regression

- 로지스틱 회귀 (Logistic Regression)

- Logistic Regression 예시 – “공부시간에 따른 합격/불합격 예측”



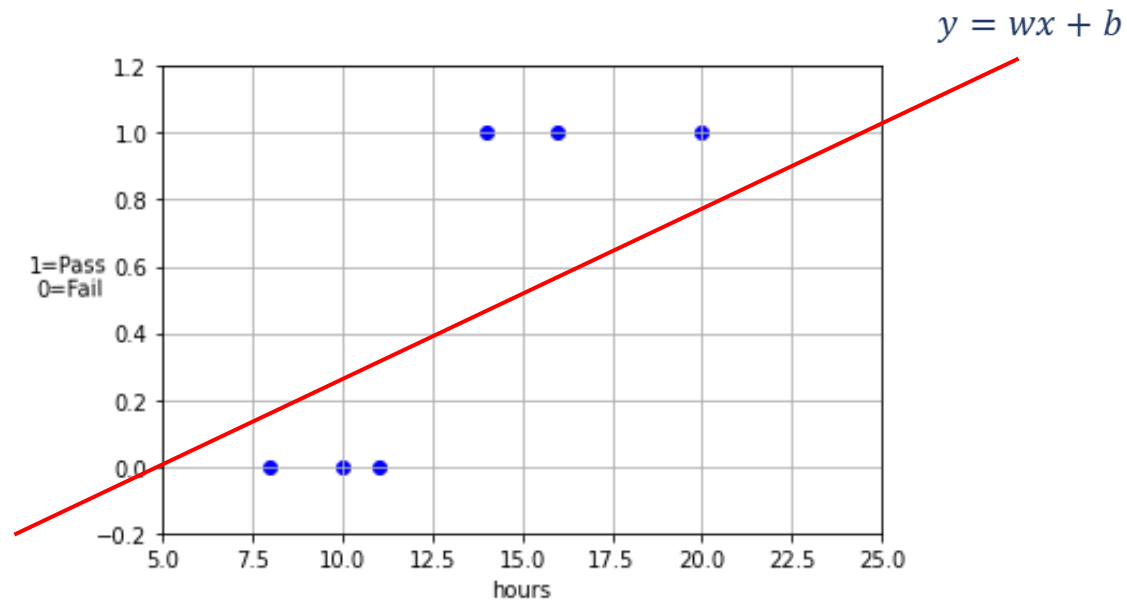
1. 지도 학습 머신 러닝 Master algorithms

3-2-2

Logistic Regression

- 로지스틱 회귀 (Logistic Regression)

- 일반적인 Linear Regression 적용?



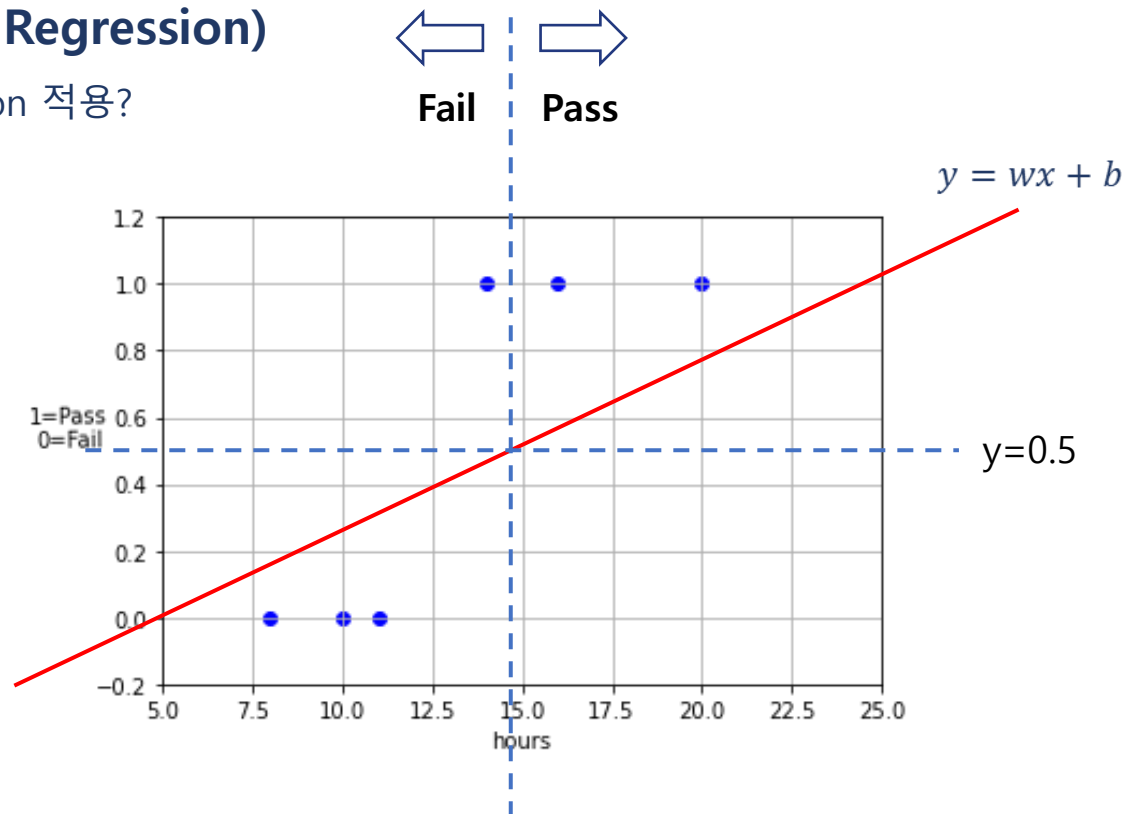
1. 지도 학습 머신 러닝 Master algorithms

3-2-3

Logistic Regression

• 로지스틱 회귀 (Logistic Regression)

- 일반적인 Linear Regression 적용?



1. 지도 학습 머신 러닝 Master algorithms

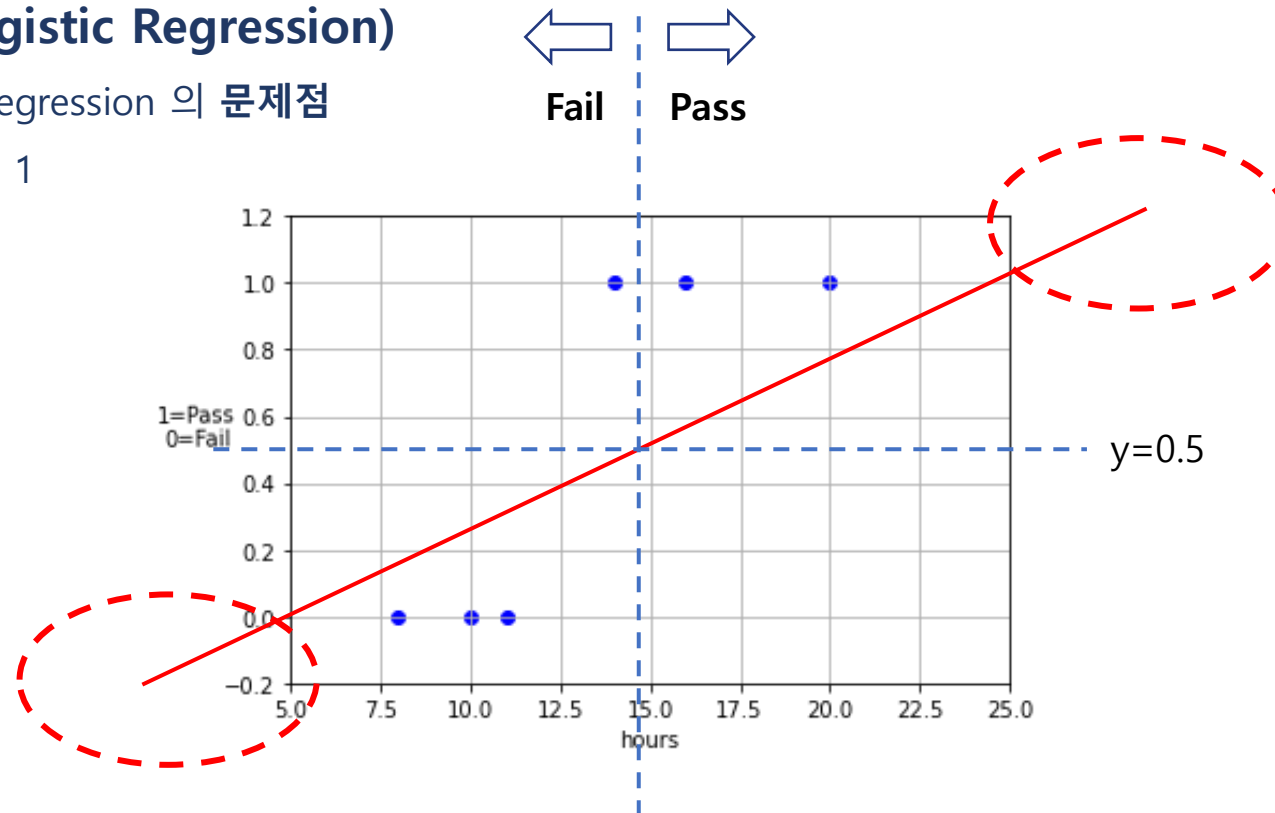
3-2-4

Logistic Regression

• 로지스틱 회귀 (Logistic Regression)

- 일반적인 Linear Regression 의 문제점

1. $Y < 0$ or $Y > 1$



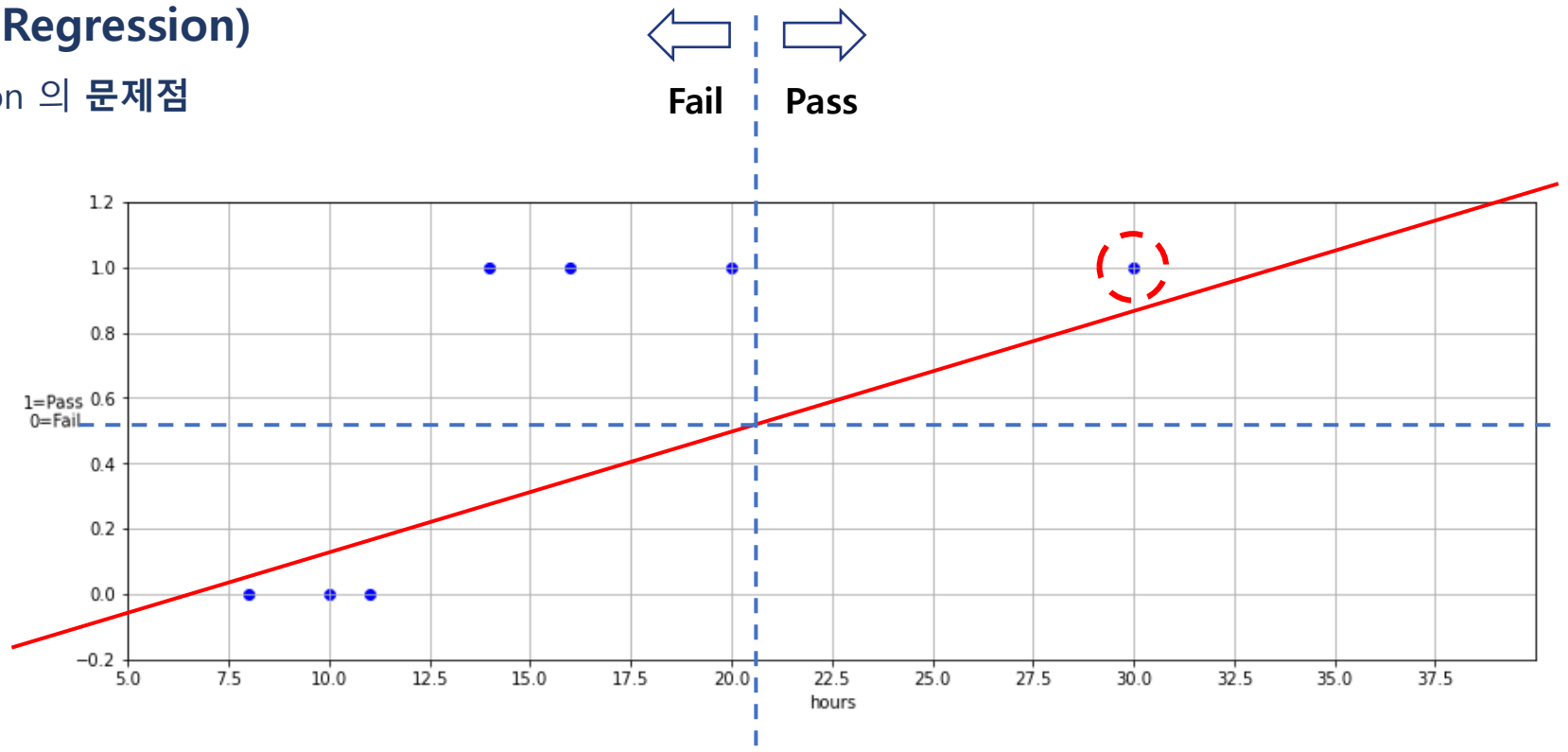
1. 지도 학습 머신 러닝 Master algorithms

3-2-5

Logistic Regression

• 로지스틱 회귀 (Logistic Regression)

- 일반적인 Linear Regression 의 문제점
 - 2. 이상치에 취약



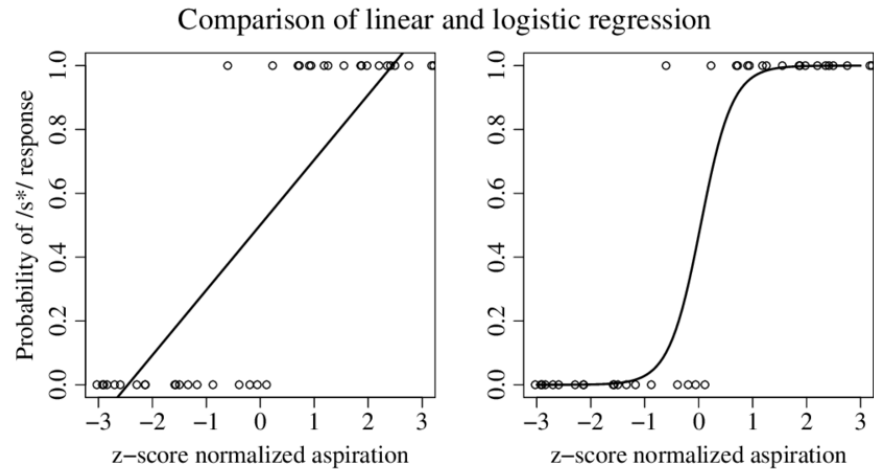
1. 지도 학습 머신 러닝 Master algorithms

3-2-6

Logistic Regression

• 가설 $H(x)$ 의 변경

- Linear regression : $H(x) = Wx + b$
- Logistic regression : $H(x) = \frac{1}{1+e^{-(Wx+b)}}$



Sigmoid function /
Logistic function

1. 지도 학습 머신 러닝 Master algorithms

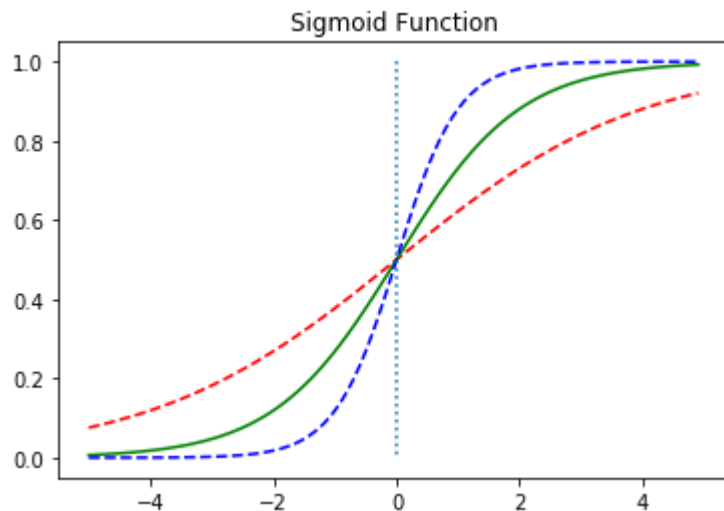
3-2-7

Logistic Regression

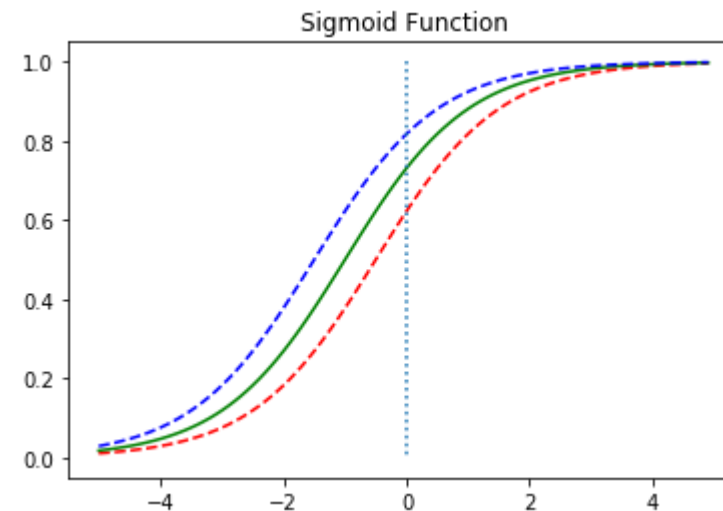
- Sigmoid function

$$H(x) = \frac{1}{1 + e^{-(Wx+b)}}$$

W 에 따른 그래프 변화



b 에 따른 그래프 변화



1. 지도 학습 머신 러닝 Master algorithms

3-2-8

Logistic Regression

• Cost function 의 변경

- Linear regression 의 Cost function 적용

$$H(x) = \frac{1}{1 + e^{-(Wx+b)}}$$

가설

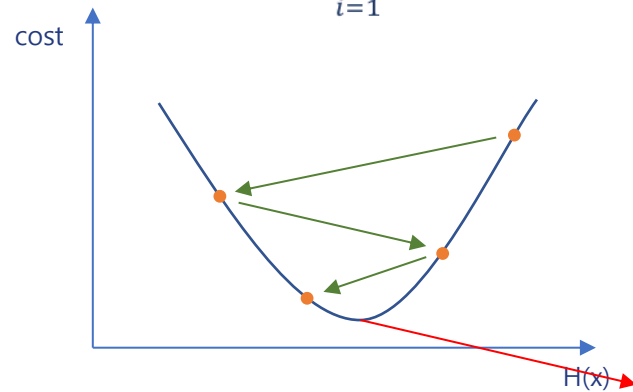
+

Cost function

Gradient Descent
적용

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

$$cost(W, b) = \frac{1}{n} \sum_{i=1}^n \{H(x_i) - y_i\}^2$$

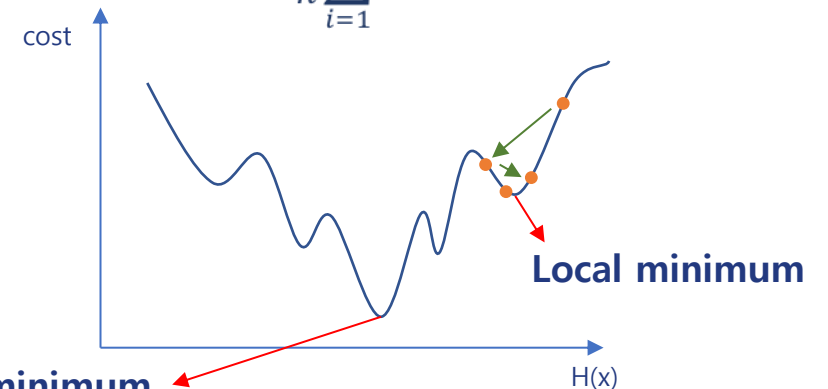


Global minimum

$$H(x) = \frac{1}{1 + e^{-(Wx+b)}}$$

+

$$cost(W, b) = \frac{1}{n} \sum_{i=1}^n \{H(x_i) - y_i\}^2$$



Local minimum

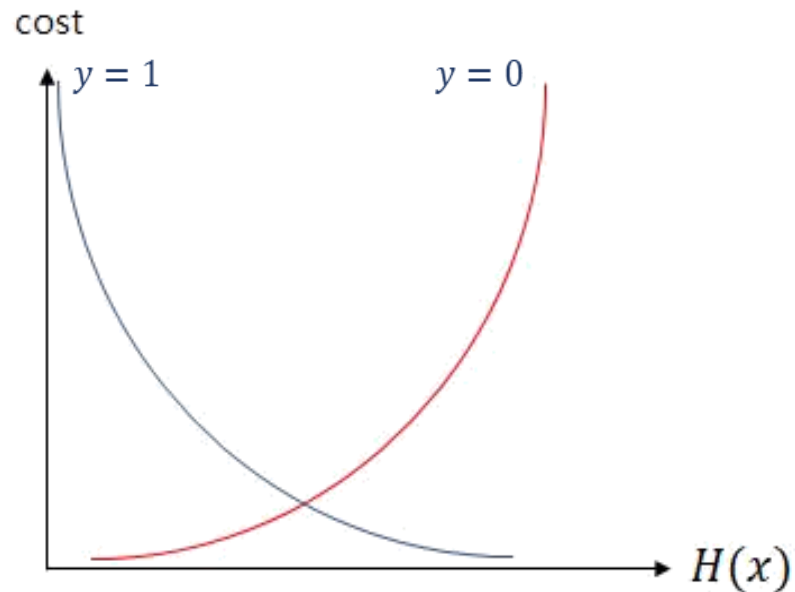
1. 지도 학습 머신 러닝 Master algorithms

3-2-9

Logistic Regression

• Cost function 의 변경

- Sigmoid 를 위한 새로운 Cost function



$$cost(H(x), y) = \begin{cases} -\log H(x) & : y = 1 \\ -\log\{1 - H(x)\} & : y = 0 \end{cases}$$



$$cost(W) = -y \log H(x) - (1 - y) \log\{1 - H(x)\}$$

1. 지도 학습 머신 러닝 Master algorithms

3-3-1

Support Vector Machine

- 서포트 벡터 머신 (Support Vector Machine, SVM)

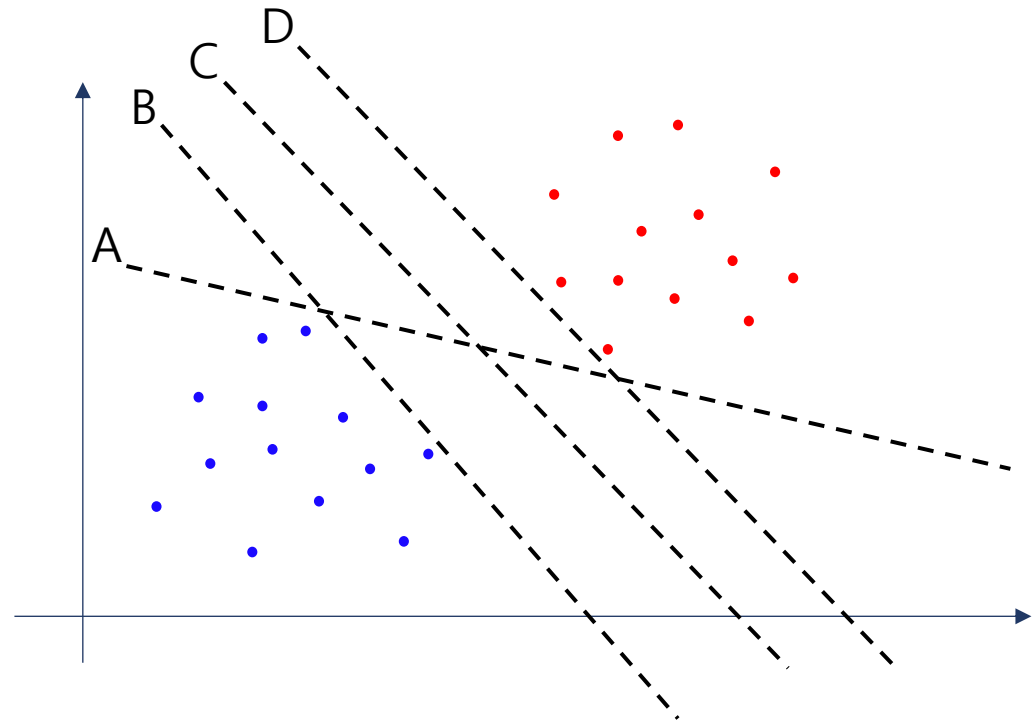
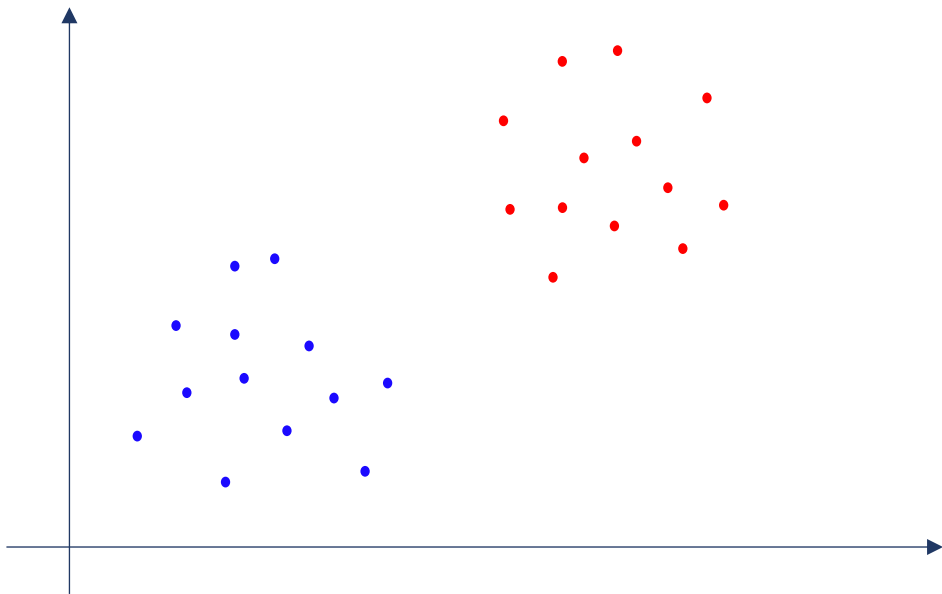
- 분류 (Classification) 혹은 회귀 (Regression) 문제에 사용되는 머신러닝 알고리즘
 - ▶ 서포트 벡터 회귀(Support Vector Regression, SVR)
- 특히 분류에 좋은 성능을 보임
- 그룹과 그룹을 분리하는 "최적의 경계"를 찾아내는 알고리즘

1. 지도 학습 머신 러닝 Master algorithms

3-3-2

Support Vector Machine

- 분류를 위한 최적의 경계?

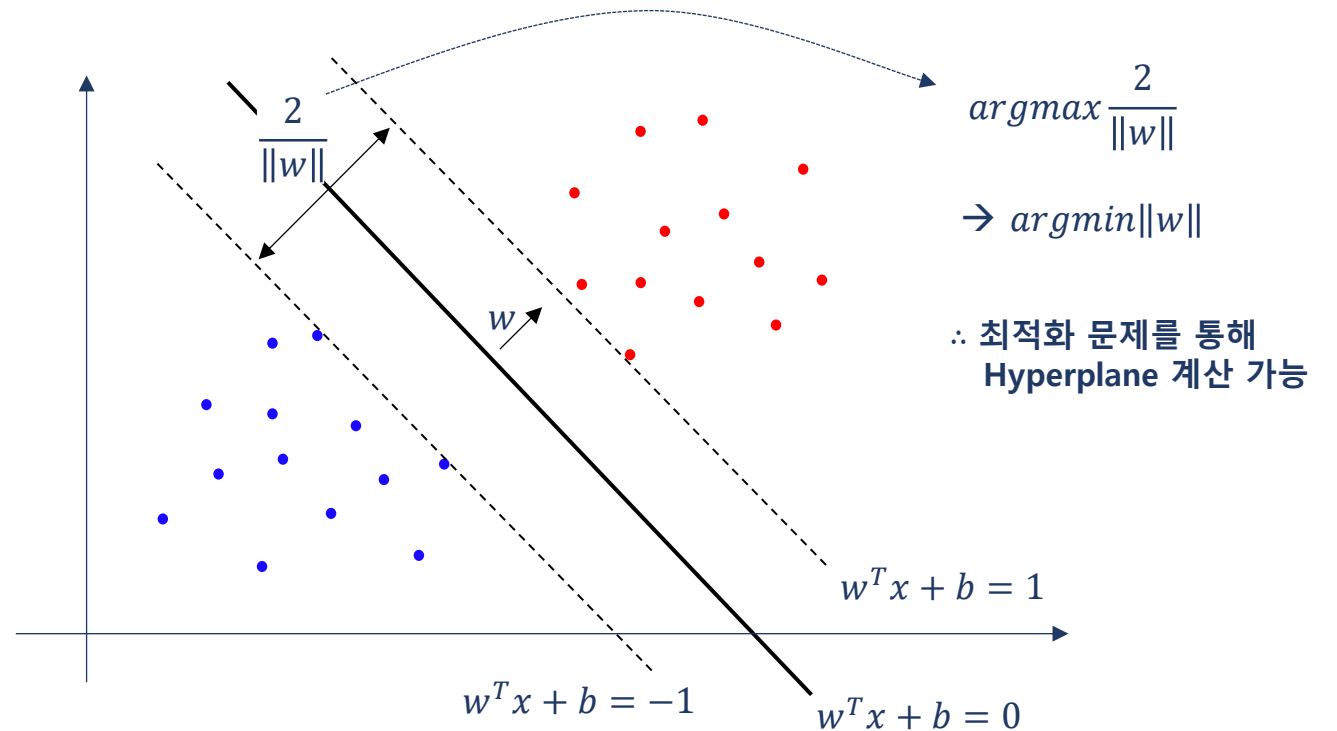
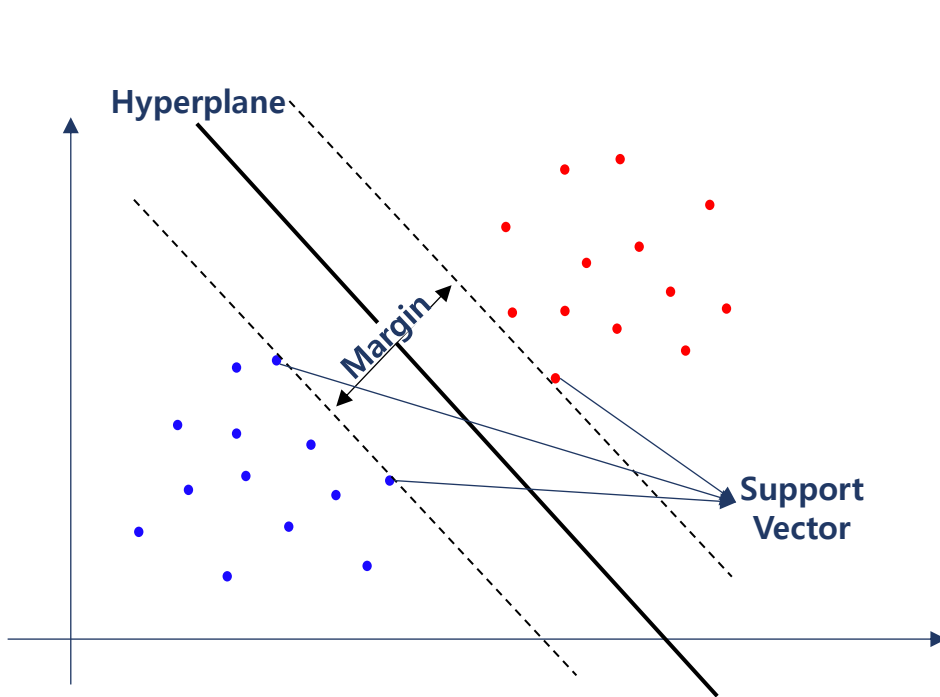


1. 지도 학습 머신 러닝 Master algorithms

3-3-3

Support Vector Machine

- Margin을 최대로 하는 Hyperplane을 찾는 것이 SVM!

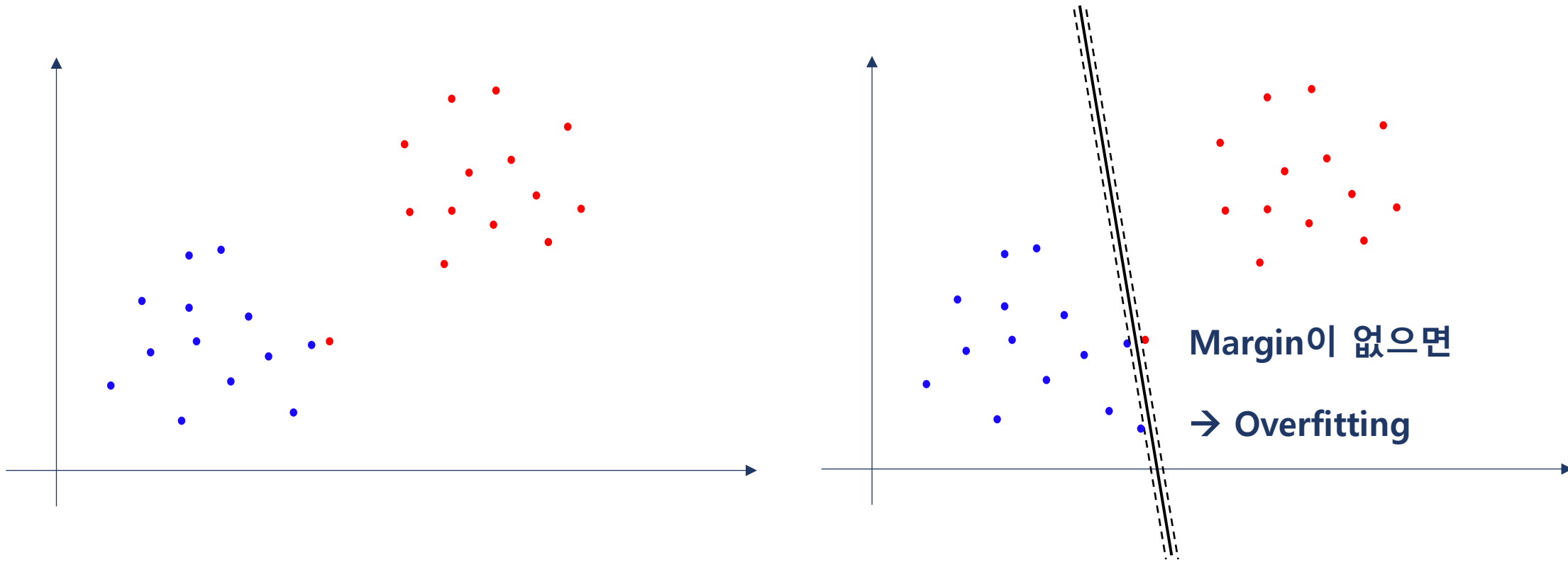


1. 지도 학습 머신 러닝 Master algorithms

3-3-4

Support Vector Machine

- 이상치가 있는 데이터

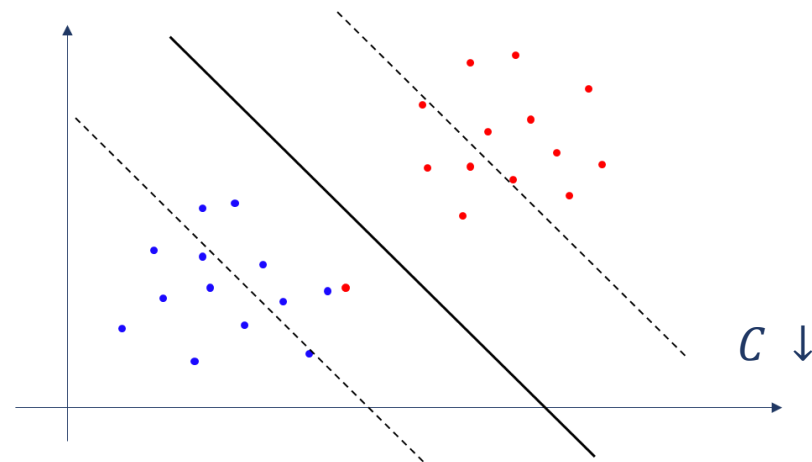
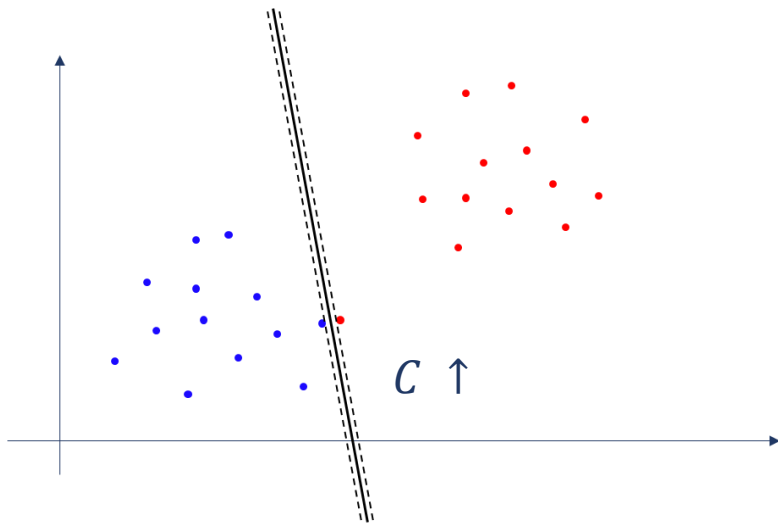


1. 지도 학습 머신 러닝 Master algorithms

3-3-5

Support Vector Machine

- “Soft Margin” → 어느 정도의 오차를 허용
 - Regularization parameter인 C 를 조정하여 오차의 허용 범위를 설정
 - C 를 높이면, 오차의 허용 범위는 작아지고, Overfitting의 가능성이 커진다.
 - C 를 낮추면, 오차의 허용 범위는 커지고, Underfitting의 가능성이 커진다.



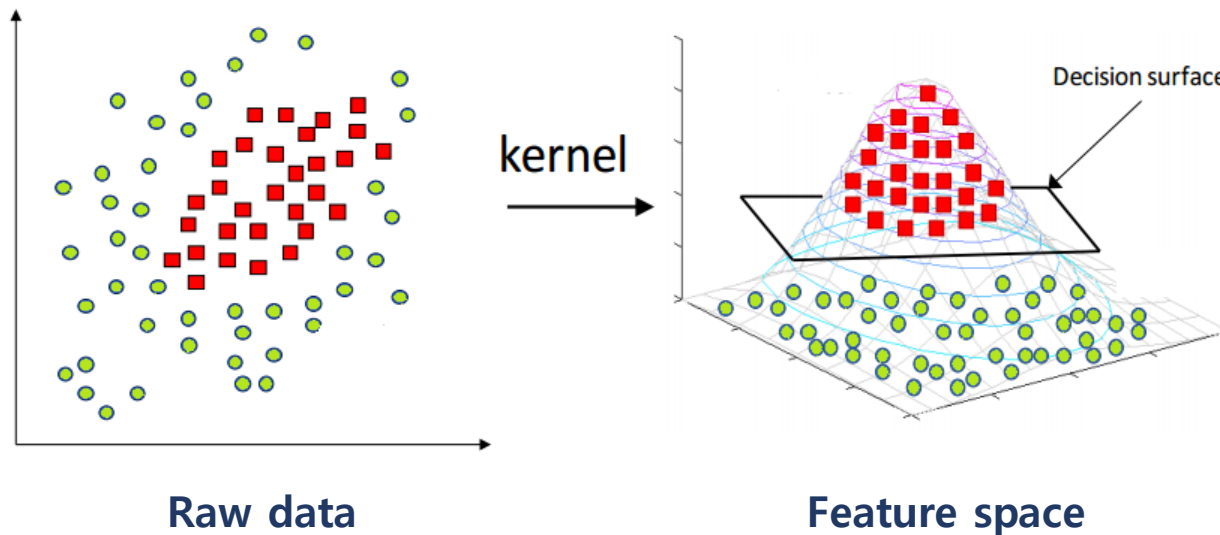
1. 지도 학습 머신 러닝 Master algorithms

3-3-6

Support Vector Machine

• 비선형 데이터의 분류

- "Kernel trick"을 이용하여 raw data를 또 다른 feature space에 mapping
- Mapping된 feature space에서의 데이터를 이용하여 Hyperplane을 계산



- Polynomial kernel

$$K(\vec{x}, \vec{x}_i) = (\vec{x}^T \vec{x}_i + \theta)^d$$

- Gaussian radial basis kernel

$$K(\vec{x}, \vec{x}_i) = e^{-\frac{1}{2\sigma^2} \|\vec{x} - \vec{x}_i\|^2}$$

- Sigmoid (=Hyperbolic tangent) kernel

$$K(\vec{x}, \vec{x}_i) = \tanh(\eta \vec{x} \vec{x}_i + \theta)$$

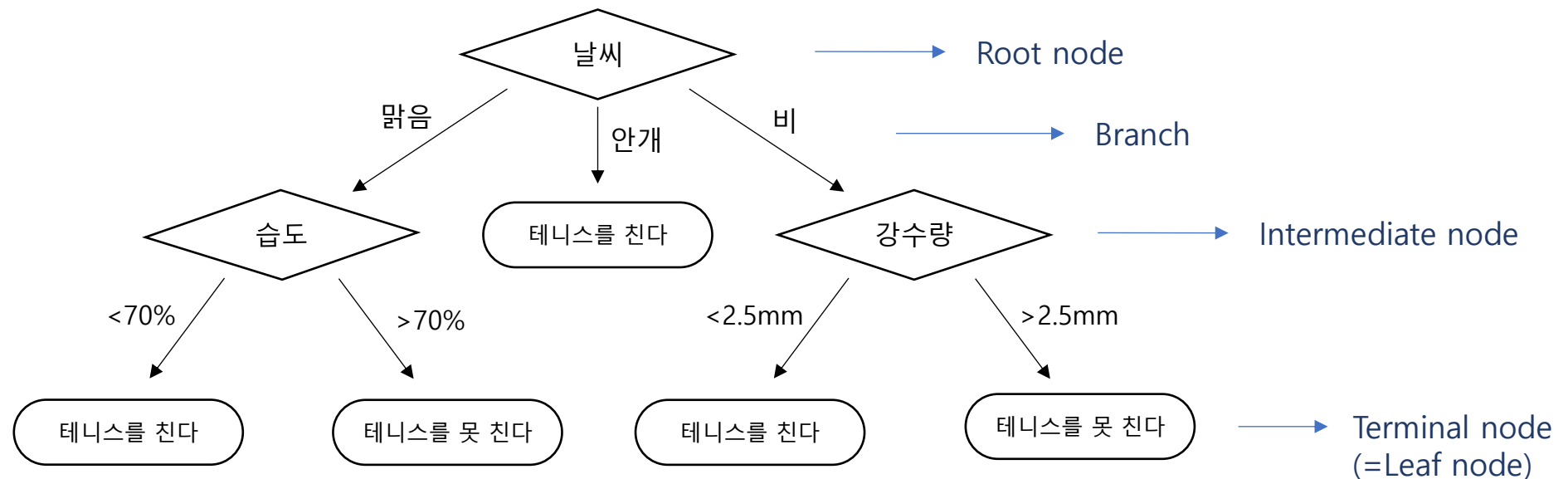
1. 지도 학습 머신 러닝 Master algorithms

3-4-1

Decision Tree

• 의사결정나무

- 분류(Classification) 혹은 회귀(Regression) 문제에 사용되는 머신러닝 알고리즘
- 일반적으로 분류 문제에 이용



Part 3. 머신 러닝

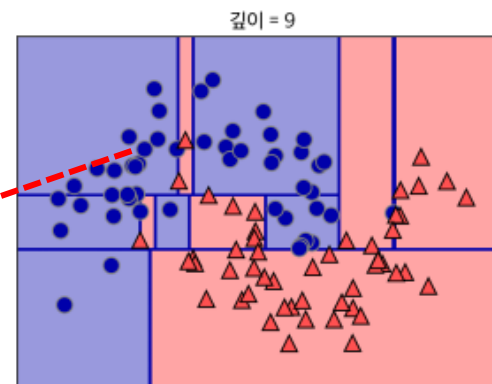
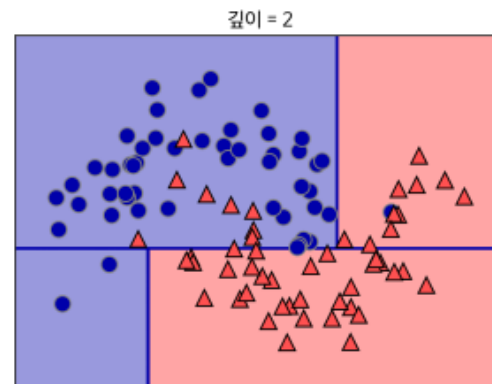
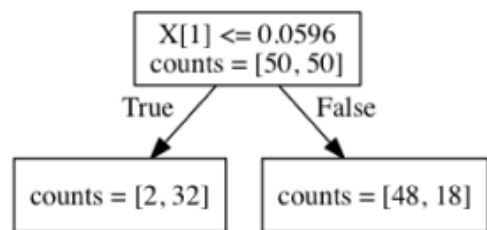
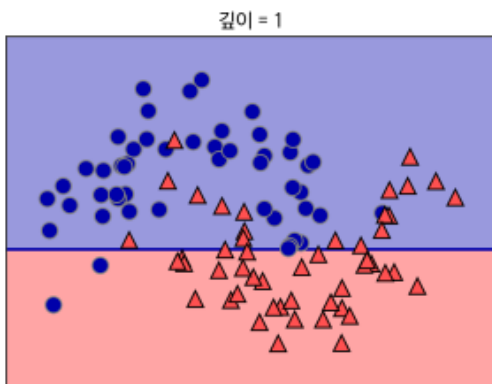
1. 지도 학습 머신 러닝 Master algorithms

3-4-2

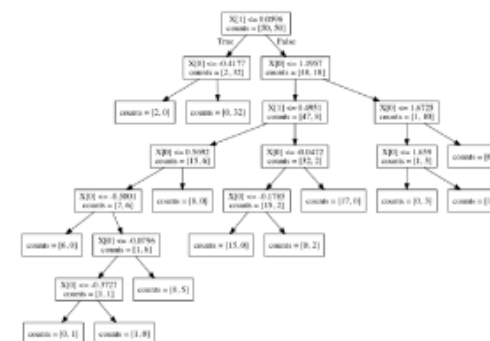
Decision Tree

• 의사결정나무

- 예제)



Overfitting!



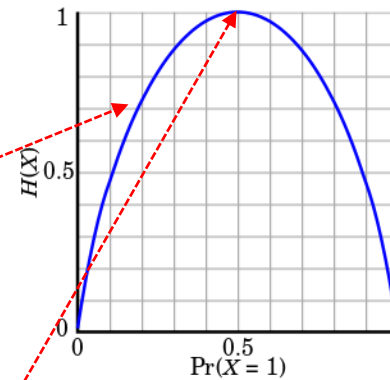
1. 지도 학습 머신 러닝 Master algorithms

3-4-3

Decision Tree

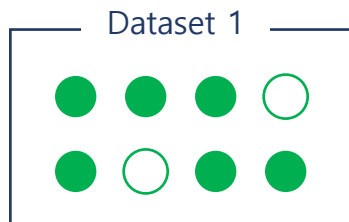
• 엔트로피

- 데이터의 불순도(Impurity)를 나타내는 척도



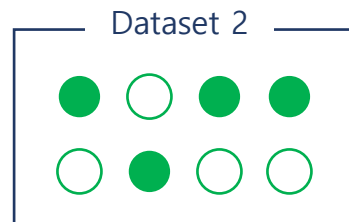
$$E = - \sum_{i=1}^n p_i \log_2 p_i$$

예)



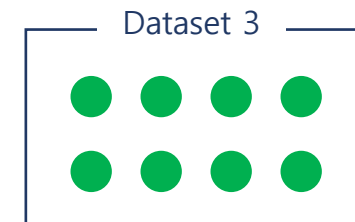
$$E = - \left(\frac{6}{8} \log_2 \frac{6}{8} + \frac{2}{8} \log_2 \frac{2}{8} \right)$$

$$= 0.8113$$



$$E = - \left(\frac{4}{8} \log_2 \frac{4}{8} + \frac{4}{8} \log_2 \frac{4}{8} \right)$$

$$= 1$$



$$E = - \left(\frac{8}{8} \log_2 \frac{8}{8} + \frac{0}{8} \log_2 \frac{0}{8} \right)$$

$$= 0$$

1. 지도 학습 머신 러닝 Master algorithms

3-4-4

Decision Tree

- 정보 획득 (Information Gain)

- 분기 이전 엔트로피와 분기 이후 엔트로피의 차이
- 즉, 분기를 통해 얻은 정보의 획득량

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

분기 X 를 통한 정보 획득 = 분기 X 이전의 엔트로피 - 분기 X 이후의 엔트로피

1. 지도 학습 머신 러닝 Master algorithms

3-4-5

Decision Tree

• 정보 획득 (Information Gain)

- 예제) $Gain(T, X) = Entropy(T) - Entropy(T, X)$

Runny nose	Fever	Headache	Flu
No	Yes	Mild	Yes
No	No	No	No
No	No	Strong	No
Yes	No	Mild	Yes

$$E(Flu) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

Runny nose	Flu(No)	Flu(Yes)	Total
No	2	1	3
Yes	0	1	1

- 변수 "Runny nose"를 통한 Gain

$$E(Flu, R) = -\left\{\frac{3}{4}\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) + \frac{1}{4}\left(\frac{1}{1}\log_2\frac{1}{1} + \frac{0}{1}\log_2\frac{0}{1}\right)\right\} = 0.6887$$

$$Gain(Flu, R) = E(Flu) - E(Flu, R) = 1 - 0.6887 = 0.3113$$

Fever	Flu(No)	Flu(Yes)	Total
No	2	1	3
Yes	0	1	1

- 변수 "Fever"를 통한 Gain

$$E(Flu, F) = -\left\{\frac{3}{4}\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) + \frac{1}{4}\left(\frac{1}{1}\log_2\frac{1}{1} + \frac{0}{1}\log_2\frac{0}{1}\right)\right\} = 0.6887$$

$$Gain(Flu, F) = E(Flu) - E(Flu, F) = 1 - 0.6887 = 0.3113$$

1. 지도 학습 머신 러닝 Master algorithms

3-4-6

Decision Tree

• 정보 획득 (Information Gain)

- 예제) $Gain(T, X) = Entropy(T) - Entropy(T, X)$

Runny nose	Fever	Headache	Flu
No	Yes	Mild	Yes
No	No	No	No
No	No	Strong	No
Yes	No	Mild	Yes

$$E(Flu) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

Headache	Flu(No)	Flu(Yes)	Total
No	1	0	1
Mild	1	1	2
Strong	0	1	1

- 변수 "Headache"를 통한 Gain

$$E(Flu, H) = -\left\{\frac{1}{4}\left(\frac{1}{1}\log_2\frac{1}{1} + \frac{0}{1}\log_2\frac{0}{1}\right) + \frac{2}{4}\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) + \frac{1}{4}\left(\frac{1}{1}\log_2\frac{1}{1} + \frac{0}{1}\log_2\frac{0}{1}\right)\right\} = 0.5$$

$$Gain(Flu, H) = E(Flu) - E(Flu, H) = 1 - 0.5 = 0.5$$

∴ 변수 "Headache"를 통한 Gain이 가장 크기 때문에, 변수 "Headache"를 첫 번째 가지로 선택

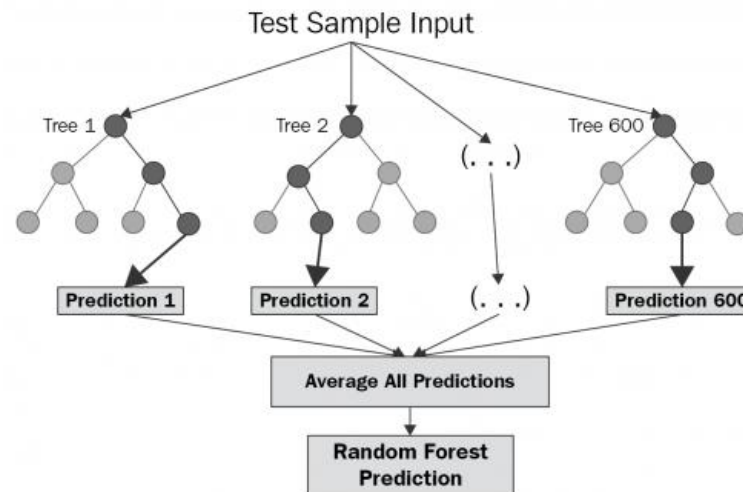
1. 지도 학습 머신 러닝 Master algorithms

3-5-1

Random Forest

• 랜덤 포레스트 (Random Forest)

- 여러 개의 Decision tree들의 예측 결과를 종합하여 예측 정확도를 향상시키는 방법
→ “Ensemble”
- 한 개의 Decision tree로 결정하는 방법보다 **Overfitting**을 방지할 수 있음



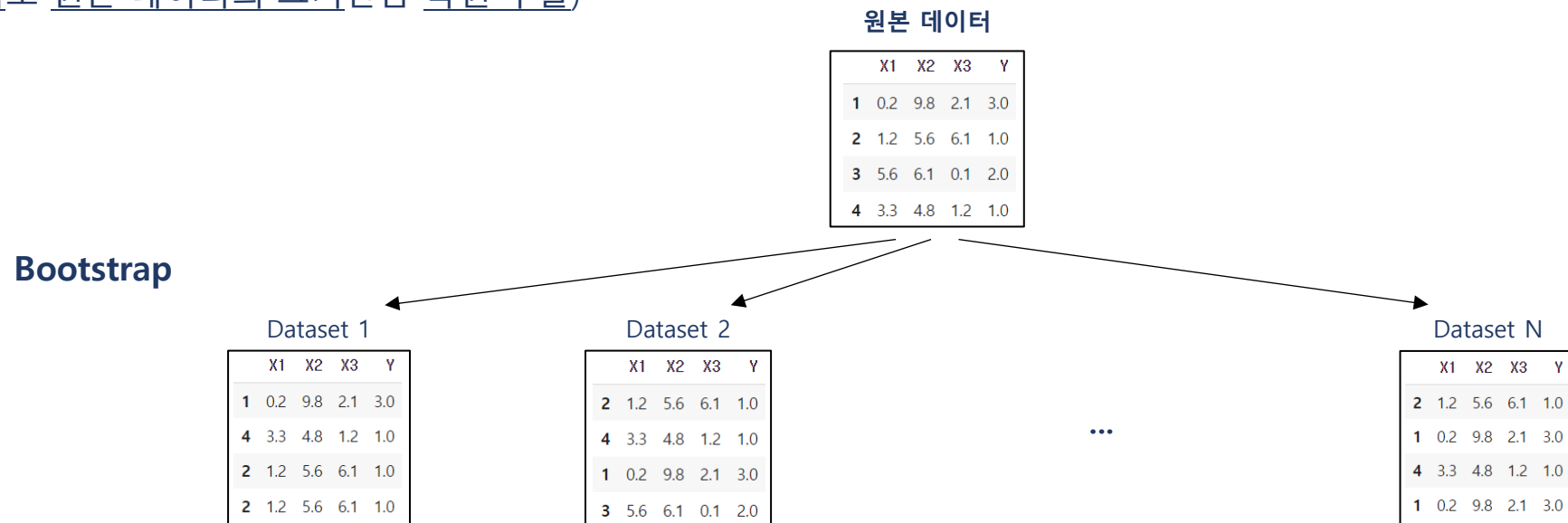
1. 지도 학습 머신 러닝 Master algorithms

3-5-2

Random Forest

• Bootstrap

- 원본 데이터로부터 Bootstrap sampling을 통해 여러 개의 데이터셋 생성
(임의로 원본 데이터의 크기만큼 복원 추출)



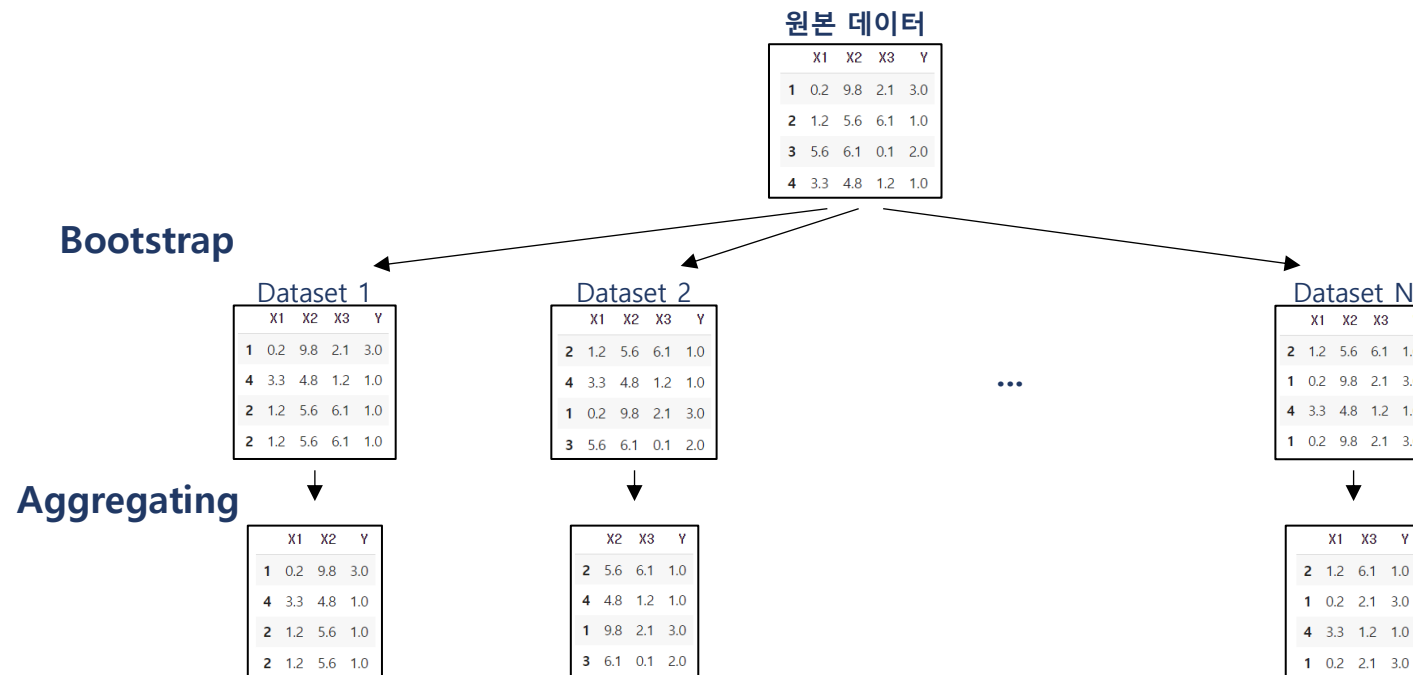
1. 지도 학습 머신 러닝 Master algorithms

3-5-3

Random Forest

• Random Feature Selection (Aggregating)

- Decision tree의 분기로 사용될 특성을 고를 때, 후보가 되는 특성들을 random하게 선택



1. 지도 학습 머신 러닝 Master algorithms

3-6-1

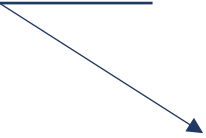
Naïve Bayes Classification

• 나이브 베이즈 분류 (Naïve Bayes Classification)

- 간단하고 효율적인 분류(Classification)를 위한 머신러닝 알고리즘
- 조건부 확률과 베이즈(Bayes) 정리를 이용하며, 특성 값이 서로 독립적임(Naïve)을 가정

- 활용 Task

1. 스팸 분류
2. 감성 분석
3. 이상 감지
4. 질병 진단


$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

1. 지도 학습 머신 러닝 Master algorithms

3-6-2

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단

환자 데이터

Headache	Flu
Mild	No
No	Yes
Strong	Yes
Mild	Yes
No	No
Strong	Yes
Strong	No
Mild	Yes
Strong	Yes
Mild	Yes
No	Yes
Mild	No
Mild	No
No	No
Strong	Yes



새로운 환자

Headache	Flu
Strong	?

Headache	Flu
Strong	?

$$P(Yes|Strong) = ?$$

$$P(No|Strong) = ?$$

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(Yes|Strong) = \frac{P(Strong|Yes)P(Yes)}{P(Strong)}$$

$$P(No|Strong) = \frac{P(Strong|No)P(No)}{P(Strong)}$$

1. 지도 학습 머신 러닝 Master algorithms

3-6-3

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단

환자 데이터

Headache	Flu
Mild	No
No	Yes
Strong	Yes
Mild	Yes
No	No
Strong	Yes
Strong	No
Mild	Yes
Strong	Yes
Mild	Yes
No	Yes
Mild	No
Mild	No
No	No
Strong	Yes



새로운 환자

Headache	Flu
Strong	?

Likelihood Table

Headache	No	Yes	
No	2	2	=4/15 0.2667
Mild	3	3	=6/15 0.4000
Strong	1	4	=5/15 0.3333
	=6/15 0.4000	=9/15 0.6000	

$$P(\text{Yes}|\text{Strong}) = \frac{P(\text{Strong}|\text{Yes})P(\text{Yes})}{P(\text{Strong})} = \frac{4/9 \cdot 9/15}{5/15} = 0.8$$

$$P(\text{No}|\text{Strong}) = \frac{P(\text{Strong}|\text{No})P(\text{No})}{P(\text{Strong})} = \frac{1/6 \cdot 6/15}{5/15} = 0.2$$

1. 지도 학습 머신 러닝 Master algorithms

3-6-4

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단

환자 데이터

Headache	Flu
Mild	No
No	Yes
Strong	Yes
Mild	Yes
No	No
Strong	Yes
Strong	No
Mild	Yes
Strong	Yes
Mild	Yes
No	Yes
Mild	No
Mild	No
No	No
Strong	Yes



새로운 환자

Headache	Flu
Strong	?

Likelihood Table

Headache	No	Yes		
No	2	2	=4/15	0.2667
Mild	3	3	=6/15	0.4000
Strong	1	4	=5/15	0.3333
	=6/15	=9/15		
	0.4000	0.6000		

$$P(Yes|Strong) = 0.8 > P(No|Strong) = 0.2$$

∴ Headache가 strong인 환자는 Flu인 것으로 판단

1. 지도 학습 머신 러닝 Master algorithms

3-6-5

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단 (특성 추가)

환자 데이터

Headache	Chills	Runny nose	Fever	Flu
Mild	Yes	No	Yes	No
No	Yes	Yes	No	Yes
Strong	Yes	No	Yes	Yes
Mild	No	Yes	Yes	Yes
No	No	No	No	No
Strong	No	Yes	Yes	Yes
Strong	No	Yes	No	No
Mild	Yes	Yes	Yes	Yes
Strong	No	Yes	Yes	Yes
Mild	Yes	Yes	No	Yes
No	Yes	Yes	Yes	Yes
Mild	No	No	No	No
Mild	Yes	No	No	No
No	No	No	Yes	No
Strong	Yes	No	No	Yes

Headache	Chills	Runny nose	Fever	Flu
Strong	No	Yes	Yes	?

$$P(Yes|H_s, C_n, R_y, F_y) = ?$$

$$P(No|H_s, C_n, R_y, F_y) = ?$$

새로운 환자

$$P(Yes|H_s, C_n, R_y, F_y) = \frac{P(H_s, C_n, R_y, F_y|Yes)P(Yes)}{P(H_s, C_n, R_y, F_y)}$$

$$= \frac{P(H_s|Yes)P(C_n|Yes)P(R_y|Yes)P(F_y|Yes)P(Yes)}{P(H_s)P(C_n)P(R_y)P(F_y)}$$

서로 독립

1. 지도 학습 머신 러닝 Master algorithms

3-6-6

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단 (특성 추가)

환자 데이터

Headache	Chills	Runny nose	Fever	Flu
Mild	Yes	No	Yes	No
No	Yes	Yes	No	Yes
Strong	Yes	No	Yes	Yes
Mild	No	Yes	Yes	Yes
No	No	No	No	No
Strong	No	Yes	Yes	Yes
Strong	No	Yes	No	No
Mild	Yes	Yes	Yes	Yes
Strong	No	Yes	Yes	Yes
Mild	Yes	Yes	No	Yes
No	Yes	Yes	Yes	Yes
Mild	No	No	No	No
Mild	Yes	No	No	No
No	No	No	Yes	No
Strong	Yes	No	No	Yes

Headache	No	Yes
No	2	2
Mild	3	3
Strong	1	4

Chills	No	Yes
No	4	3
Yes	2	6

Runny nose	No	Yes
No	4	3
Yes	2	6

Fever	No	Yes
No	4	3
Yes	2	6

$$\begin{aligned}
 P(\text{Yes} | H_s, C_n, R_y, F_y) &= \frac{P(H_s | \text{Yes}) P(C_n | \text{Yes}) P(R_y | \text{Yes}) P(F_y | \text{Yes}) P(\text{Yes})}{P(H_s) P(C_n) P(R_y) P(F_y)} \\
 &= \frac{4/9 \cdot 3/9 \cdot 6/9 \cdot 6/9 \cdot 9/15}{5/15 \cdot 7/15 \cdot 8/15 \cdot 8/15} = 0.8928
 \end{aligned}$$

1. 지도 학습 머신 러닝 Master algorithms

3-6-7

Naïve Bayes Classification

• 예시 - Headache 정도에 따른 Flu 여부 판단 (특성 추가)

환자 데이터

Headache	Chills	Runny nose	Fever	Flu
Mild	Yes	No	Yes	No
No	Yes	Yes	No	Yes
Strong	Yes	No	Yes	Yes
Mild	No	Yes	Yes	Yes
No	No	No	No	No
Strong	No	Yes	Yes	Yes
Strong	No	Yes	No	No
Mild	Yes	Yes	Yes	Yes
Strong	No	Yes	Yes	Yes
Mild	Yes	Yes	No	Yes
No	Yes	Yes	Yes	Yes
Mild	No	No	No	No
Mild	Yes	No	No	No
No	No	No	Yes	No
Strong	Yes	No	No	Yes

Headache	No	Yes
No	2	2
Mild	3	3
Strong	1	4

Chills	No	Yes
No	4	3
Yes	2	6

Runny nose	No	Yes
No	4	3
Yes	2	6

Fever	No	Yes
No	4	3
Yes	2	6

$$\begin{aligned}
 P(\text{No} | H_s, C_n, R_y, F_y) &= \frac{P(H_s | \text{No}) P(C_n | \text{No}) P(R_y | \text{No}) P(F_y | \text{No}) P(\text{No})}{P(H_s) P(C_n) P(R_y) P(F_y)} \\
 &= \frac{1/6 \cdot 4/6 \cdot 2/6 \cdot 2/6 \cdot 6/15}{5/15 \cdot 7/15 \cdot 8/15 \cdot 8/15} = 0.1116
 \end{aligned}$$

1. 지도 학습 머신 러닝 Master algorithms

3-6-8

Naïve Bayes Classification

- 예시 - Headache 정도에 따른 Flu 여부 판단 (특성 추가)

환자 데이터

Headache	Chills	Runny nose	Fever	Flu
Mild	Yes	No	Yes	No
No	Yes	Yes	No	Yes
Strong	Yes	No	Yes	Yes
Mild	No	Yes	Yes	Yes
No	No	No	No	No
Strong	No	Yes	Yes	Yes
Strong	No	Yes	No	No
Mild	Yes	Yes	Yes	Yes
Strong	No	Yes	Yes	Yes
Mild	Yes	Yes	No	Yes
No	Yes	Yes	Yes	Yes
Mild	No	No	No	No
Mild	Yes	No	No	No
No	No	No	Yes	No
Strong	Yes	No	No	Yes

$$P(Yes|H_s, C_n, R_y, F_y) = 0.8928 > P(No|H_s, C_n, R_y, F_y) = 0.1116$$

∴ Headache가 Strong, Chills가 No, Runny nose가 Yes, Fever가 Yes인 환자는
Flu인 것으로 판단

1. 지도 학습 머신 러닝 Master algorithms

3-6-9

Naïve Bayes Classification

• 나이브 베이즈 종류

1. 베르누이 나이브 베이즈 분류 (Bernoulli Bayes Classification)

X : 이진변수(0 or 1) ~ Y : 범주형 변수(0,1,2,...)

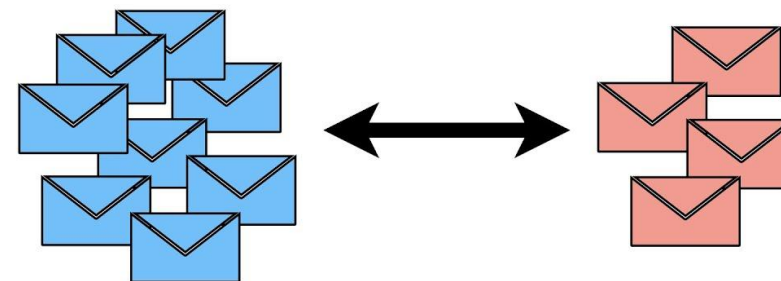
2. 다항분포 나이브 베이즈 분류 (Multinomial Naïve Bayes Classification)

X : 범주형 변수(0,1,2,...) ~ Y : 범주형 변수(0,1,2,...)

3. 가우시안 나이브 베이즈 분류(Gaussian Naïve Bayes Classification)

X : 연속형 변수 ~ Y : 범주형 변수(0,1,2,...)

Naive Bayes....



...Clearly Explained!!!

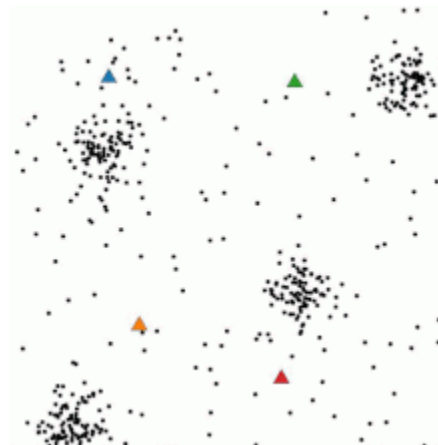
2. 비지도 학습 머신 러닝 Master algorithms

3-7-0

Clustering

- 클러스터링 (Clustering, 군집화)

- 정답(Label)이 주어지지 않은 데이터에서 특성이 비슷한 데이터끼리 클러스터화 하여 나누는 비지도학습 방법



Classification : 그룹에 대한 정보가 있는 데이터에서, 그룹에 대한 정보를 학습하여 새로운 데이터의 그룹을 분류

Clustering : 그룹에 대한 정보가 없는 데이터에서, 데이터의 특성을 학습하여 그룹을 정의하고 데이터를 군집화

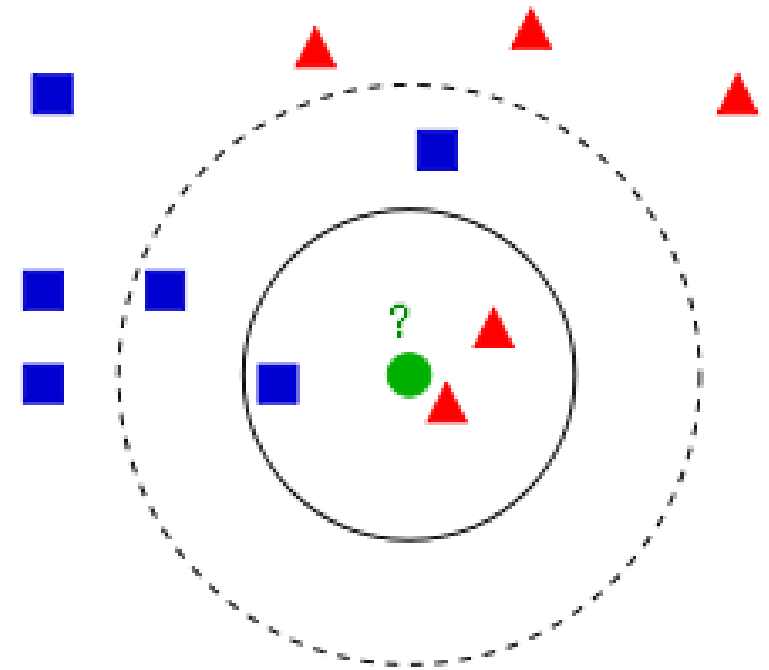
2. 비지도 학습 머신 러닝 Master algorithms

3-7-1

k -NN (Nearest Neighbor)

• K -NN 알고리즘 개념

- 주변 데이터를 탐색하여 주어진 데이터의 범주를 분류하는 알고리즘
- K 개의 가장 가까운 이웃(Neighbor) 데이터의 범주에 따라 결정
- 훈련(train) 과정이 없는 모델 – “Lazy Model”



2. 비지도 학습 머신 러닝 Master algorithms

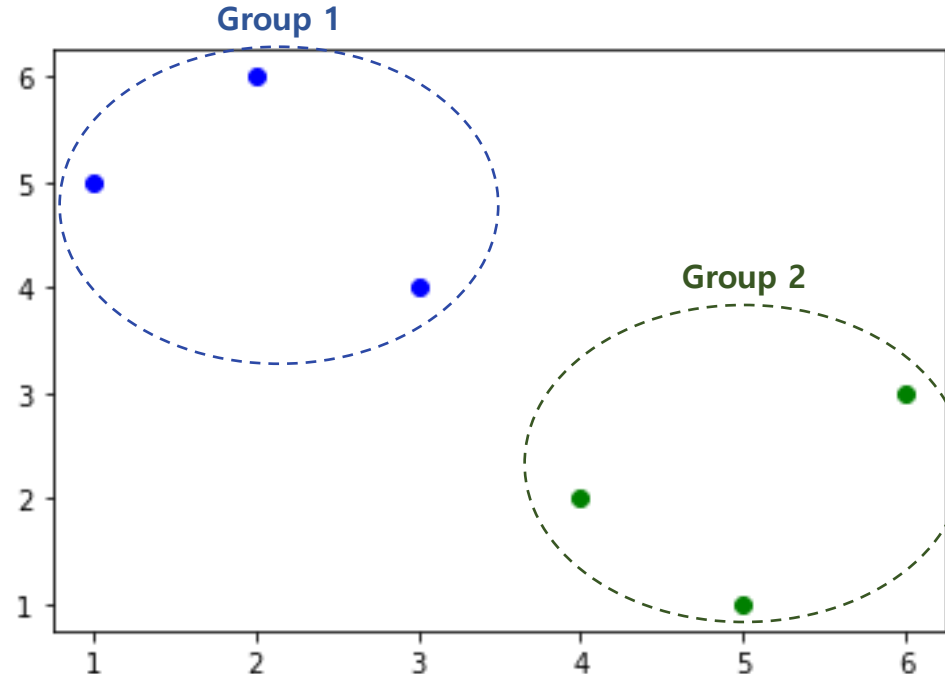
3-7-2

k -NN (Nearest Neighbor)

- K -NN 원리

- 라벨된 데이터(Labeled data)

	x	y	group
0	1	5	1
1	2	6	1
2	3	4	1
3	5	1	2
4	6	3	2
5	4	2	2



2. 비지도 학습 머신 러닝 Master algorithms

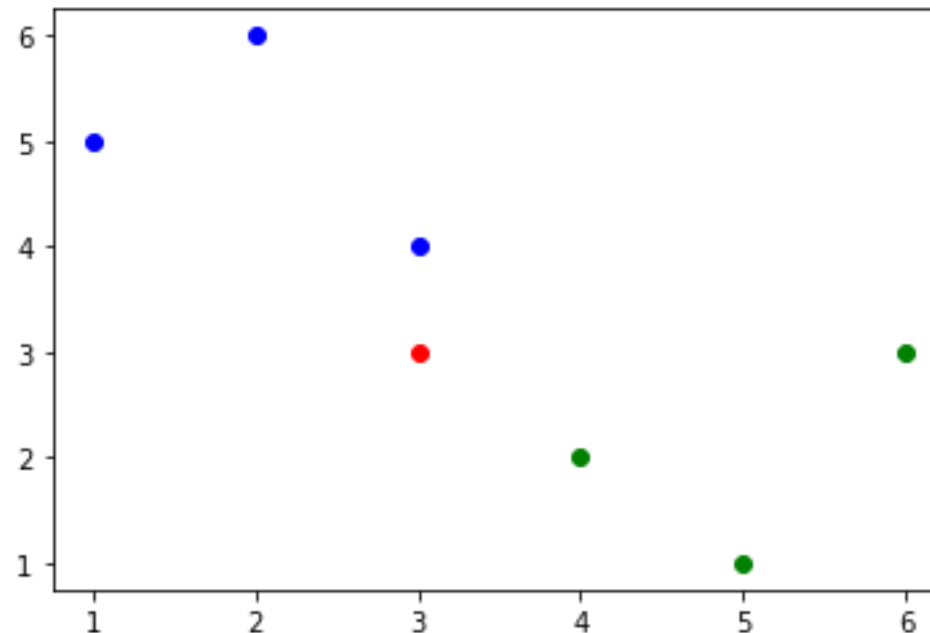
3-7-3

k -NN (Nearest Neighbor)

- K -NN 원리

- 새로운 데이터(Unseen data)

	x	y	group
0	1	5	1
1	2	6	1
2	3	4	1
3	5	1	2
4	6	3	2
5	4	2	2
6	3	3	?



2. 비지도 학습 머신 러닝 Master algorithms

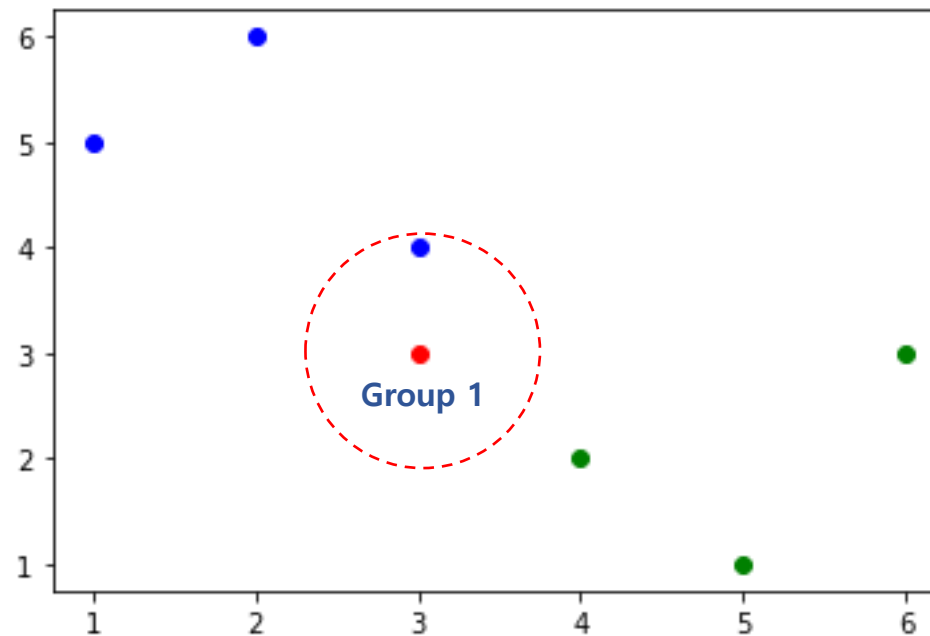
3-7-4

k -NN (Nearest Neighbor)

- K -NN 원리

- $k = 1$

	x	y	group
0	1	5	1
1	2	6	1
2	3	4	1
3	5	1	2
4	6	3	2
5	4	2	2
6	3	3	?



2. 비지도 학습 머신 러닝 Master algorithms

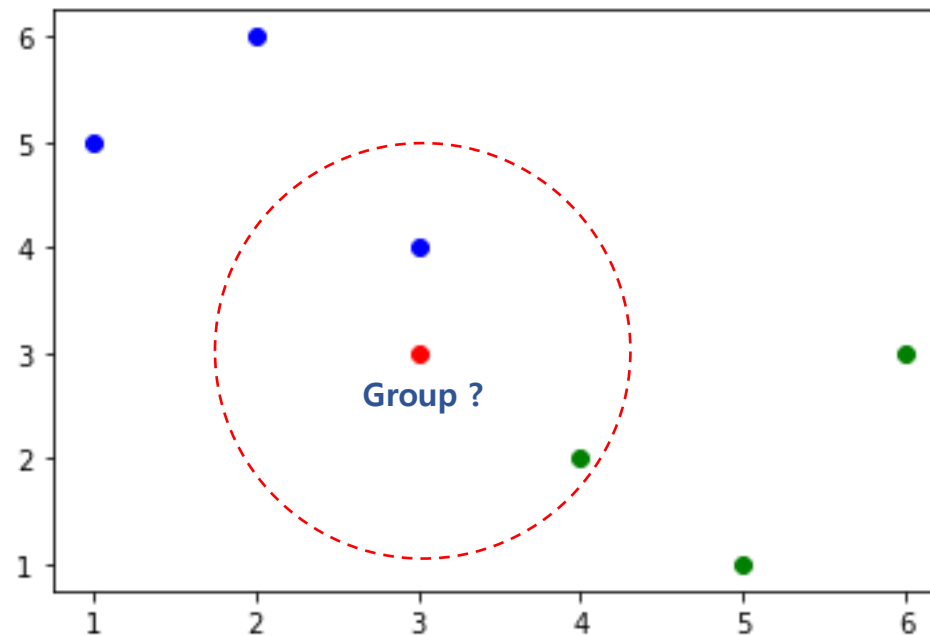
3-7-5

k -NN (Nearest Neighbor)

- K -NN 원리

- $k = 2$

	x	y	group
0	1	5	1
1	2	6	1
2	3	4	1
3	5	1	2
4	6	3	2
5	4	2	2
6	3	3	?



2. 비지도 학습 머신 러닝 Master algorithms

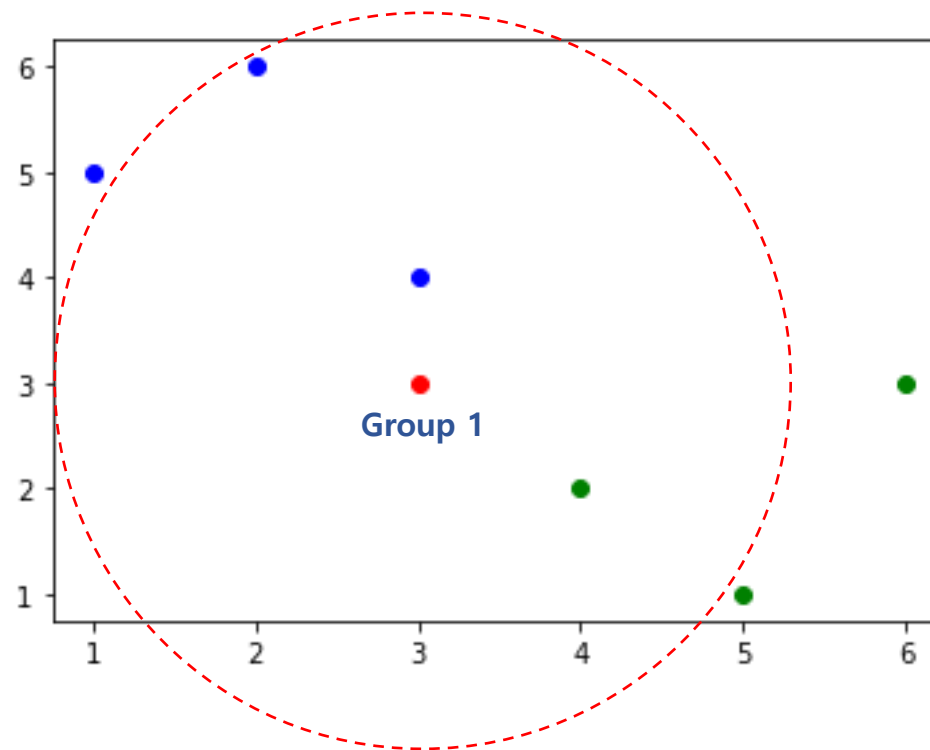
3-7-6

k -NN (Nearest Neighbor)

- K -NN 원리

- $k = 3$

	x	y	group
0	1	5	1
1	2	6	1
2	3	4	1
3	5	1	2
4	6	3	2
5	4	2	2
6	3	3	?



2. 비지도 학습 머신 러닝 Master algorithms

3-7-7

k-NN (Nearest Neighbor)

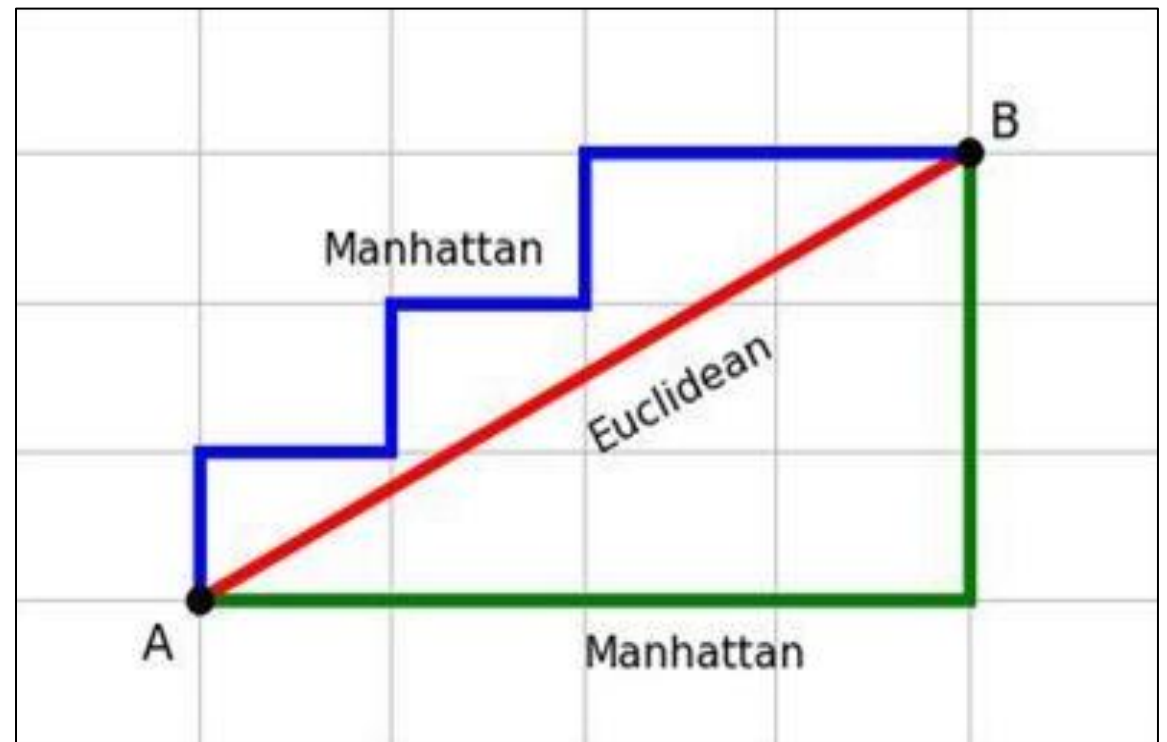
• K-NN 군집화 거리 계산 방법

1. 유클리드 거리 (Euclidean Distance)

$$\text{수식 : } d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

2. 맨해튼 거리 (Manhattan Distance)

$$\text{수식 : } d(A, B) = |x_2 - x_1| + |y_2 - y_1|$$



2. 비지도 학습 머신 러닝 Master algorithms

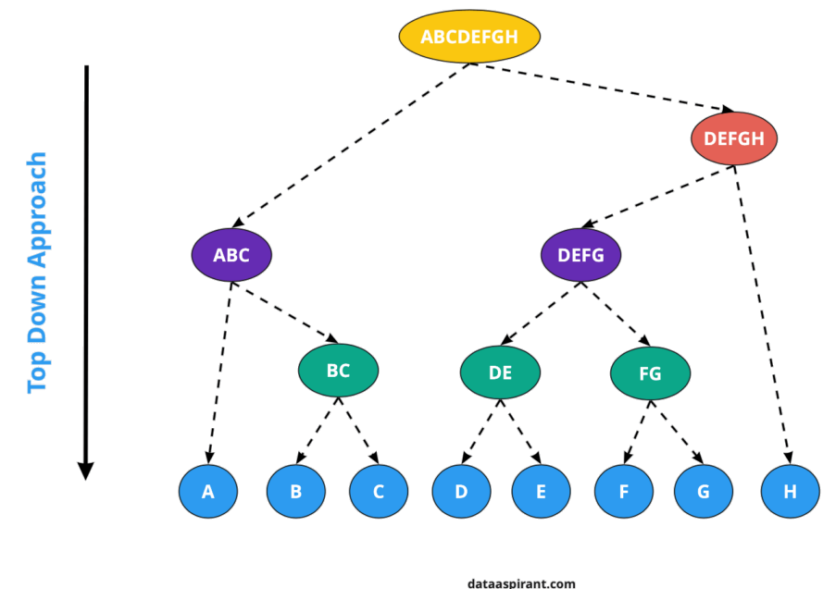
3-8-1

Hierarchical Clustering

• 계층적 군집화(Hierarchical Clustering)

- 클러스터의 계층(Hierarchy)을 만드는 클러스터링 방법
- 가장 가까운 데이터부터 클러스터링하여 순차적으로 모든 데이터를 클러스터링
- 구현이 간단하고, 클러스터의 수를 지정하지 않아도 되며, 명확한 시각화가 가능
- 자료가 많은 경우 시간 소모가 크고, 특성이 많고 복잡한 문제에 부적합

Hierarchical **Divisive** Clustering

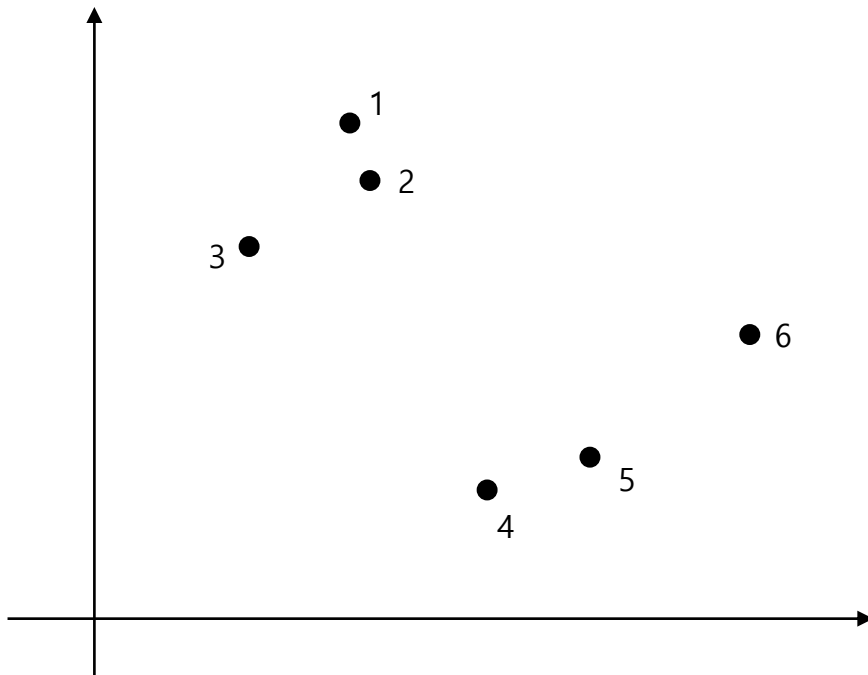


2. 비지도 학습 머신 러닝 Master algorithms

3-8-2

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정



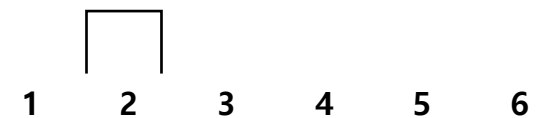
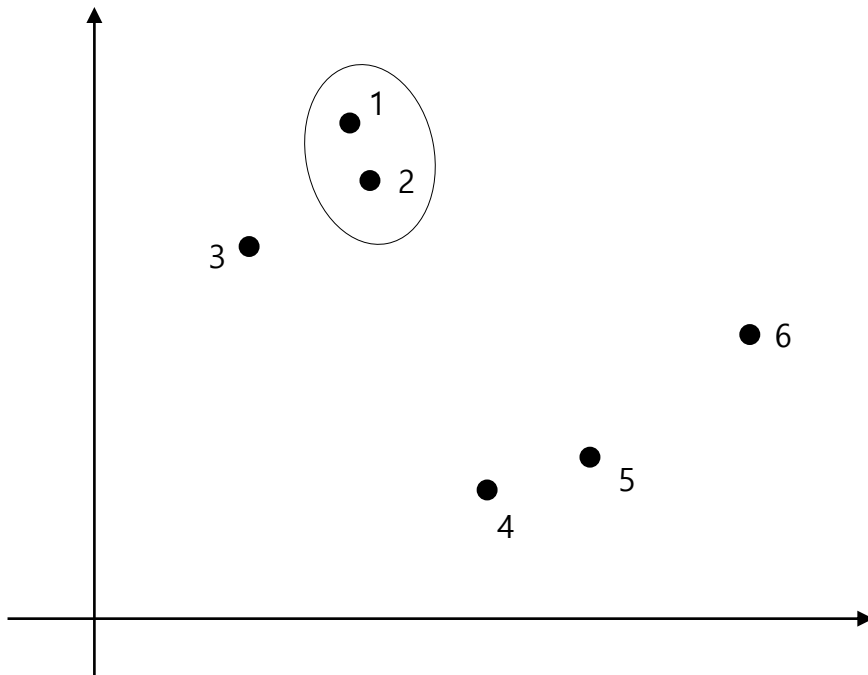
1 2 3 4 5 6

2. 비지도 학습 머신 러닝 Master algorithms

3-8-3

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정

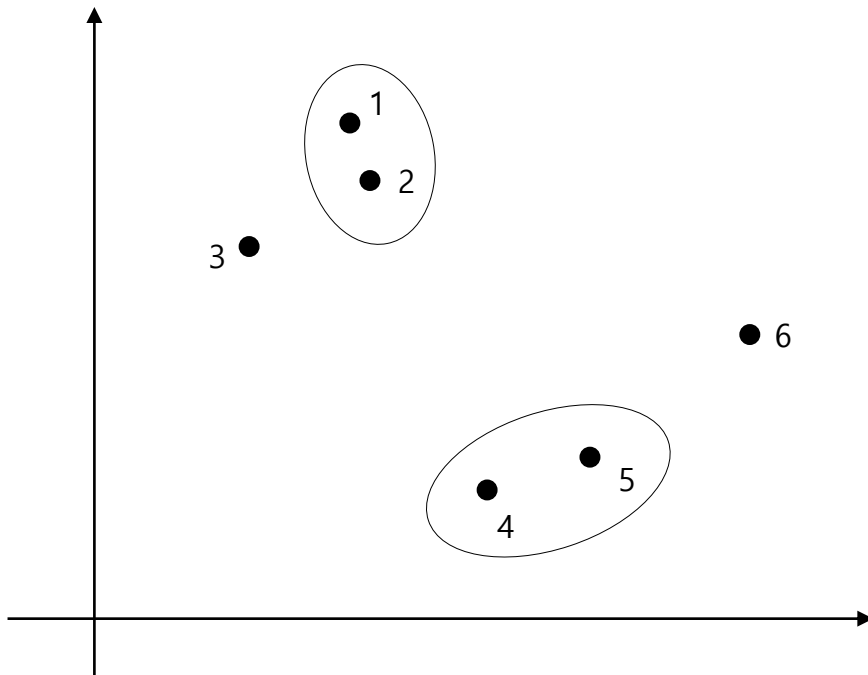


2. 비지도 학습 머신 러닝 Master algorithms

3-8-4

Hierarchical Clustering

• 계층적 군집화(Hierarchical Clustering) 과정

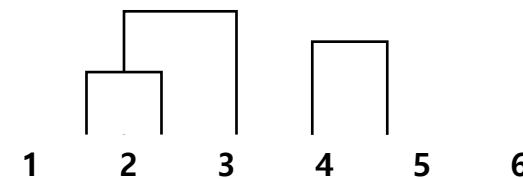
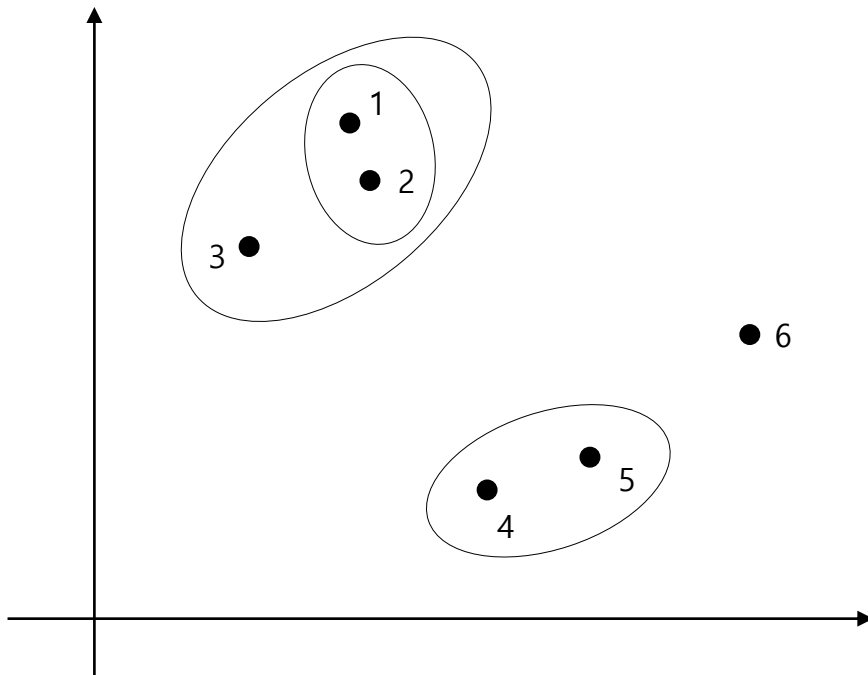


2. 비지도 학습 머신 러닝 Master algorithms

3-8-5

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정

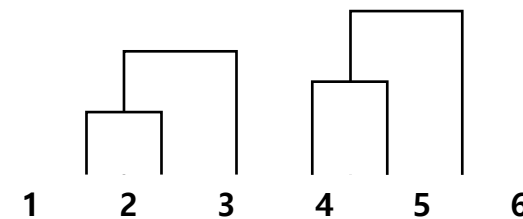
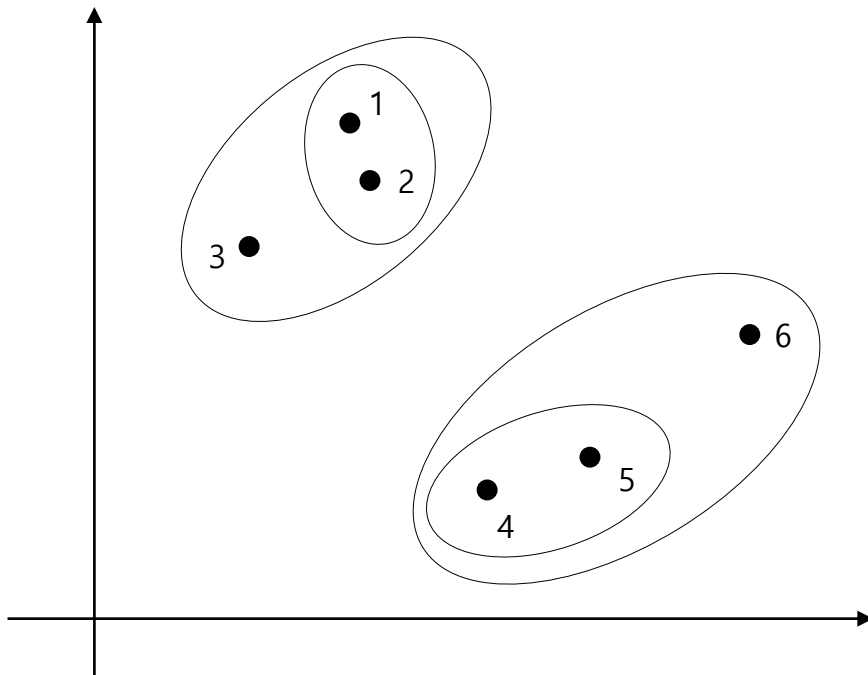


2. 비지도 학습 머신 러닝 Master algorithms

3-8-6

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정

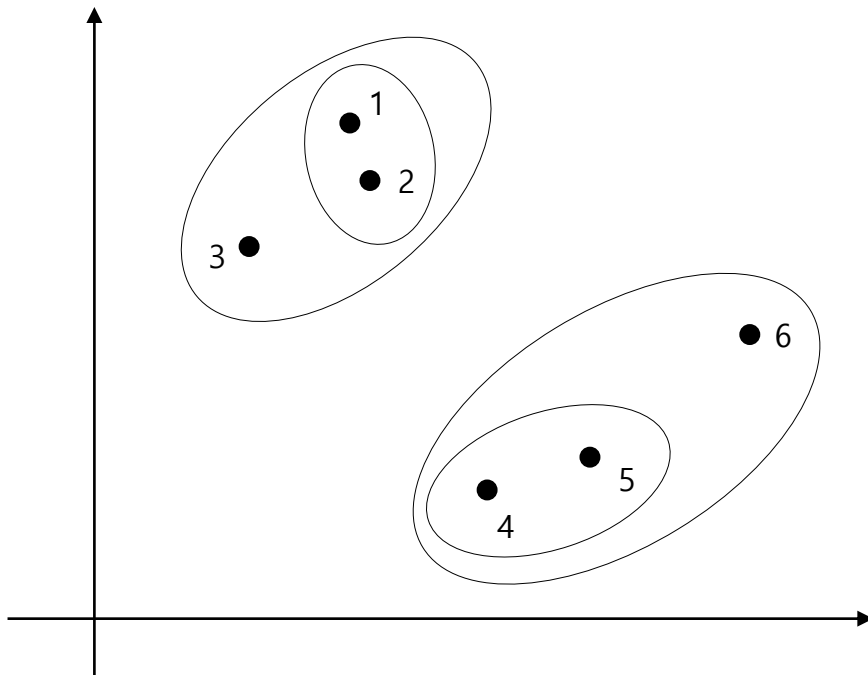


2. 비지도 학습 머신 러닝 Master algorithms

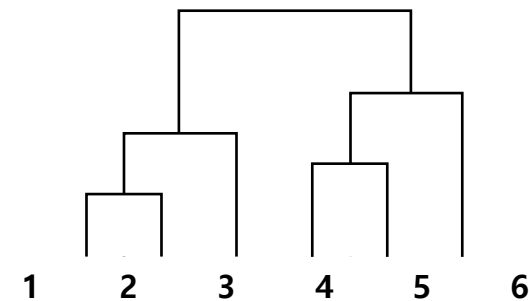
3-8-7

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정



“Dendrogram”



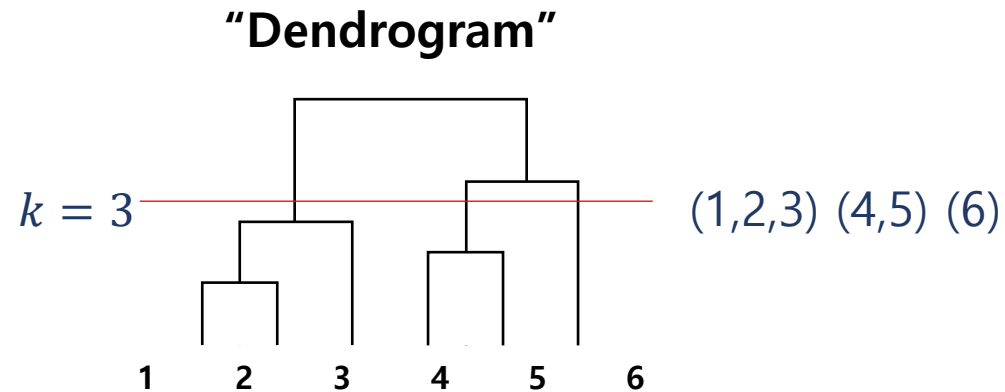
2. 비지도 학습 머신 러닝 Master algorithms

3-8-8

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정

- k 개의 그룹으로 클러스터링



2. 비지도 학습 머신 러닝 Master algorithms

3-8-9

Hierarchical Clustering

- 계층적 군집화(Hierarchical Clustering) 과정

- K 개의 그룹으로 클러스터링

