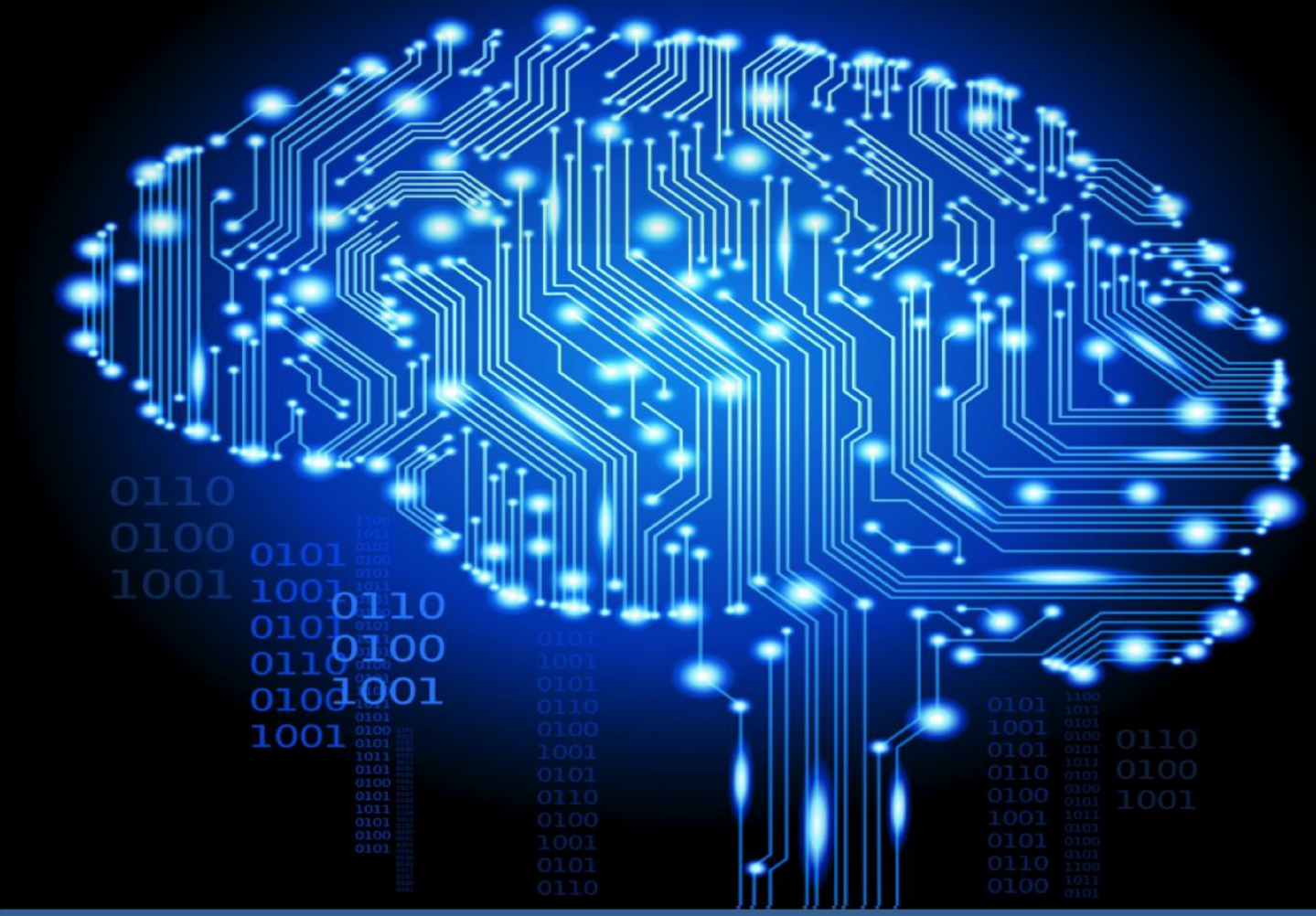


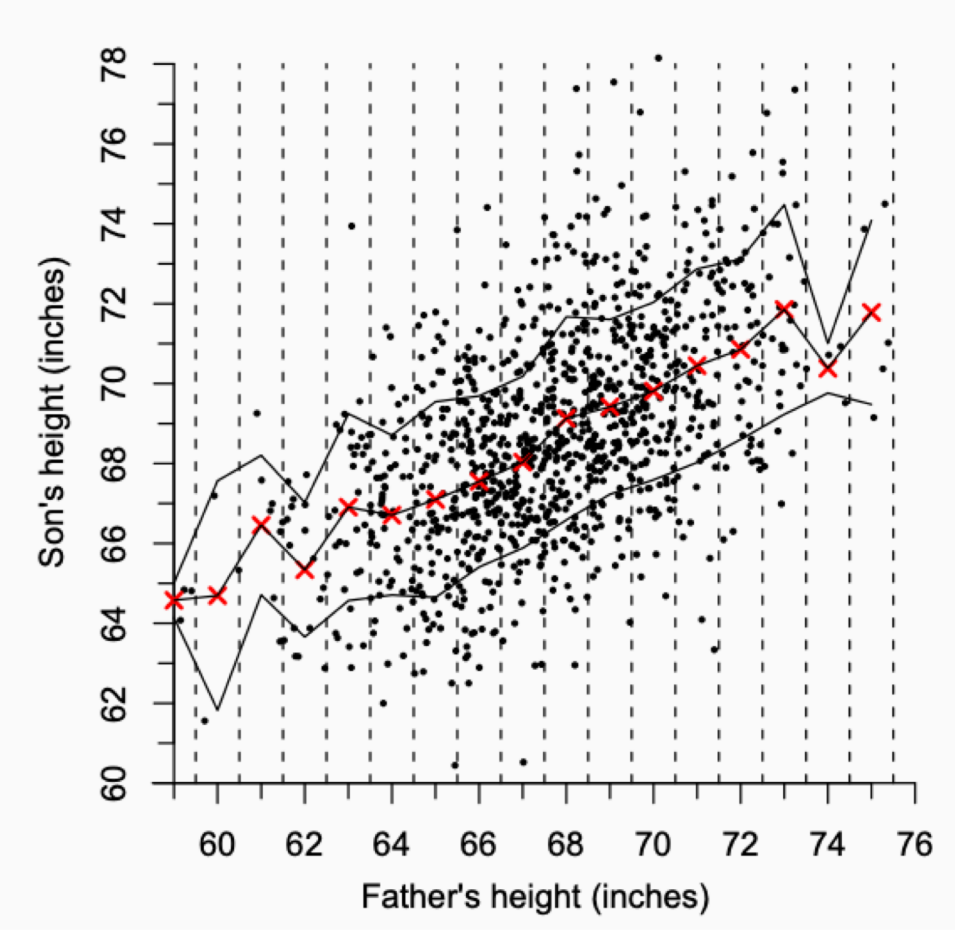
Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
http://slcf.snu.ac.kr

Lecture 8: Regression, Decision Tree, SVM, PCA

Regression

Linear Regression

Example : Pearson's Father-and-Son Data



아버지의 키와 아들의 키와의 관계?

$y = \beta_0 + \beta_1 x$ (x : father's height, y : son's height)

Linear Regression

Regression Model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

$$\begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Find $\boldsymbol{\beta}$ => minimize $\boldsymbol{\varepsilon^T \varepsilon}$

Linear Regression

Minimize $\boldsymbol{\varepsilon^T \varepsilon}$

Choose $\hat{\boldsymbol{\beta}}$ to minimize

$$\text{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_p x_{ip} \right)^2$$

$$= \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

To solve the problem

$$\frac{\partial SSE}{\partial \boldsymbol{\beta}} = 0 \quad implies \quad 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0} \Rightarrow \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y} \quad \text{(the normal equations)}$$

$$Solving\ for\ \hat{\boldsymbol{\beta}},\ we\ get\ \text{(analogous to } \hat{\beta} = \frac{S_{xy}}{S_{xx}}\text{)}$$

$$\boxed{\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}}$$

Ridge Regression, Lasso Regresion

β 값 자체에 대해 Penalty 를 줘서 모델을 단순하게 만들 수 있음

Linear Regression : $(Y - X\beta)^T(Y - X\beta)$ 을 최소화하는 β를 찾자

Ridge Regression : $(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$ 을 최소화하는 β 를 찾자

Lasso Regression : $(Y - X\beta)^T(Y - X\beta) + \lambda|\beta|_1$ 을 최소화하는 β 를 찾자

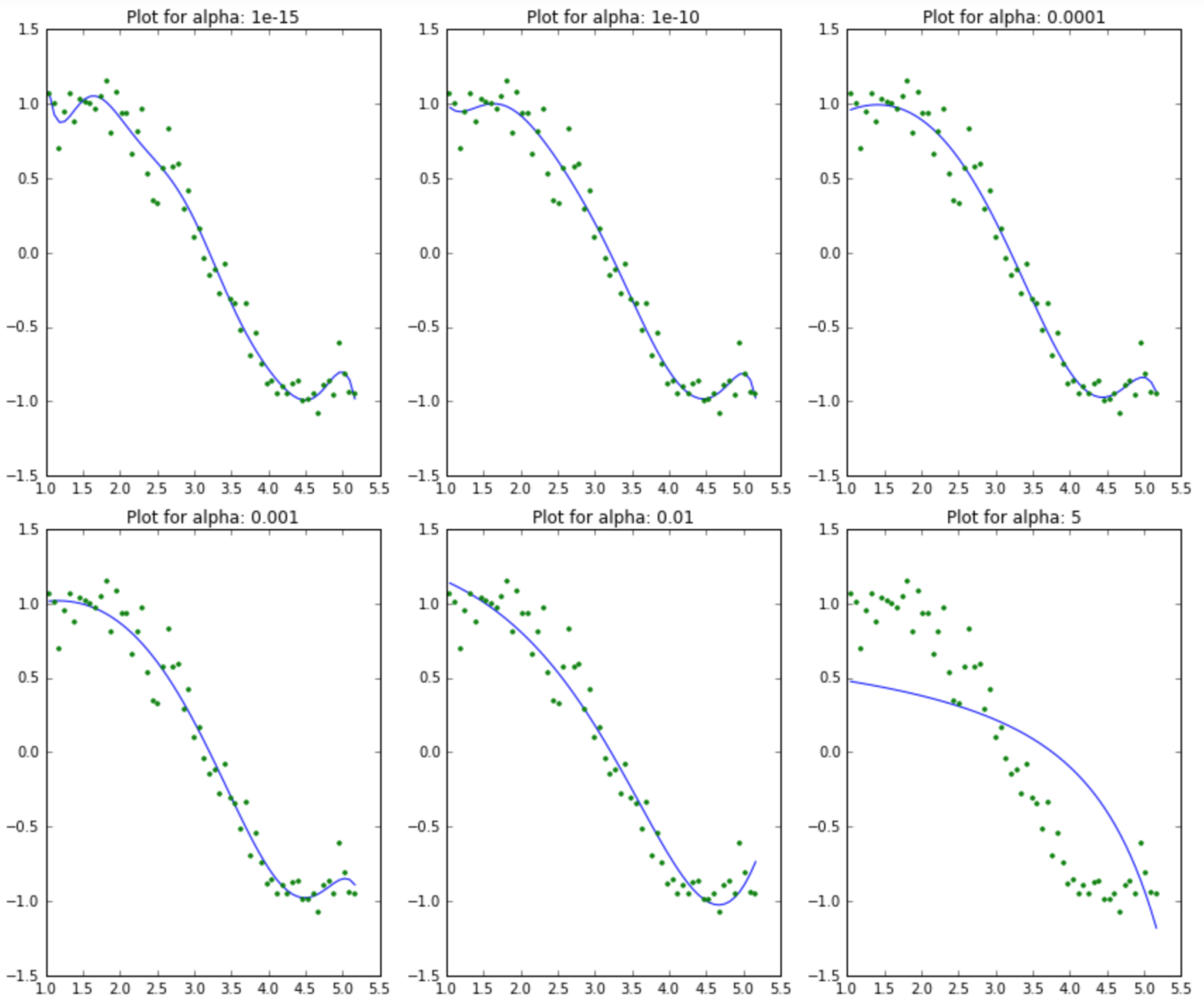
=> β가 지나치게 커지는 것을 방지

Ridge Regression, Lasso Regresion

λ 크기에 따른 모델 복잡도 변화

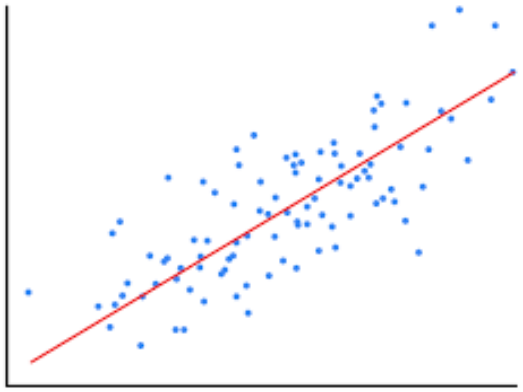
λ 크다 => β 커지는걸 막아준다 => 모델 단순화

λ 작다 => β 커지는걸 허용한다 => 모델 복잡해짐

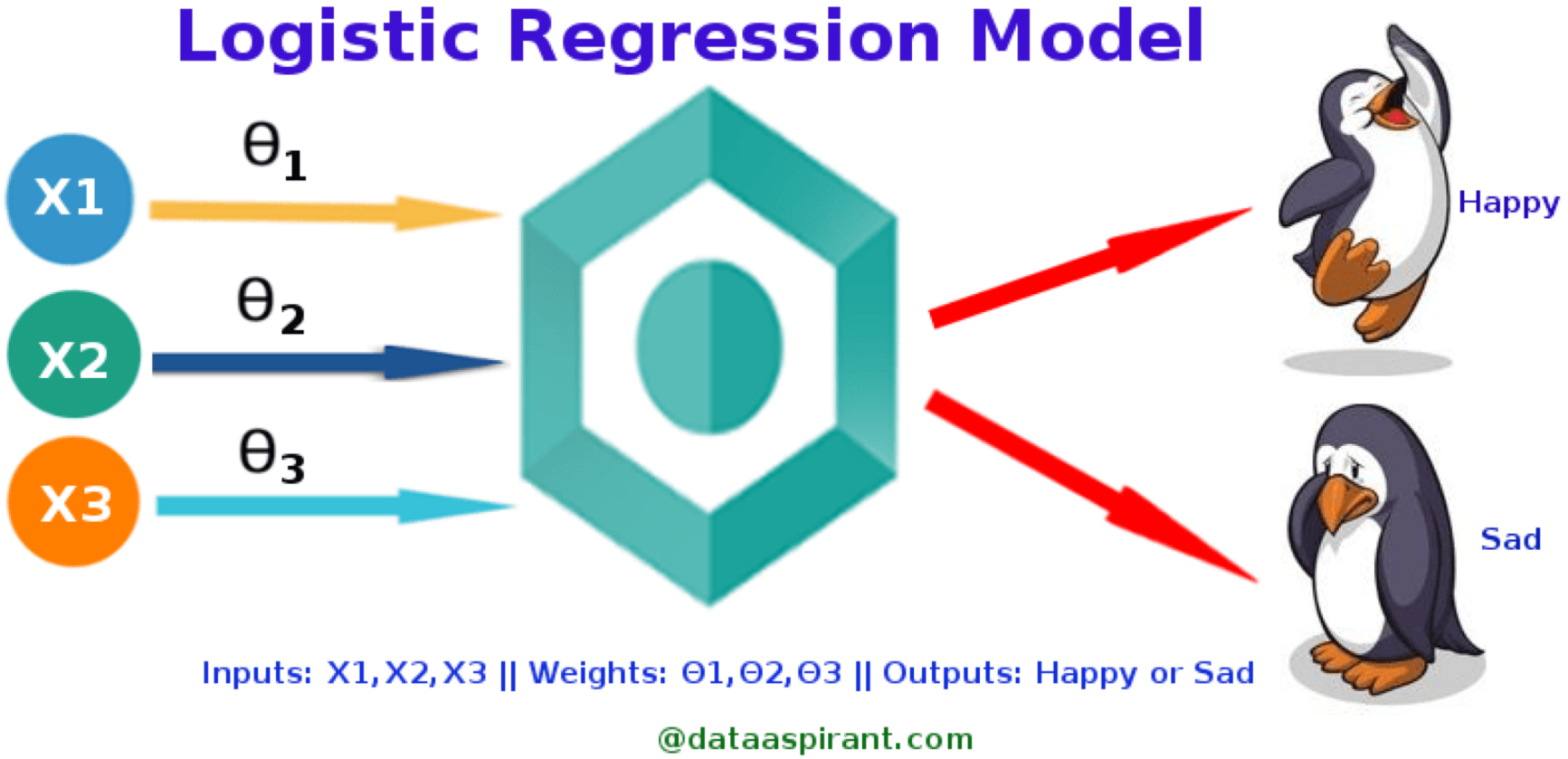


Logistic Regression

Linear Regression is not suitable for classification



Linear Regression



Source:http://dataaspirant.com/2017/03/02/how-logistic-regression-model-works/

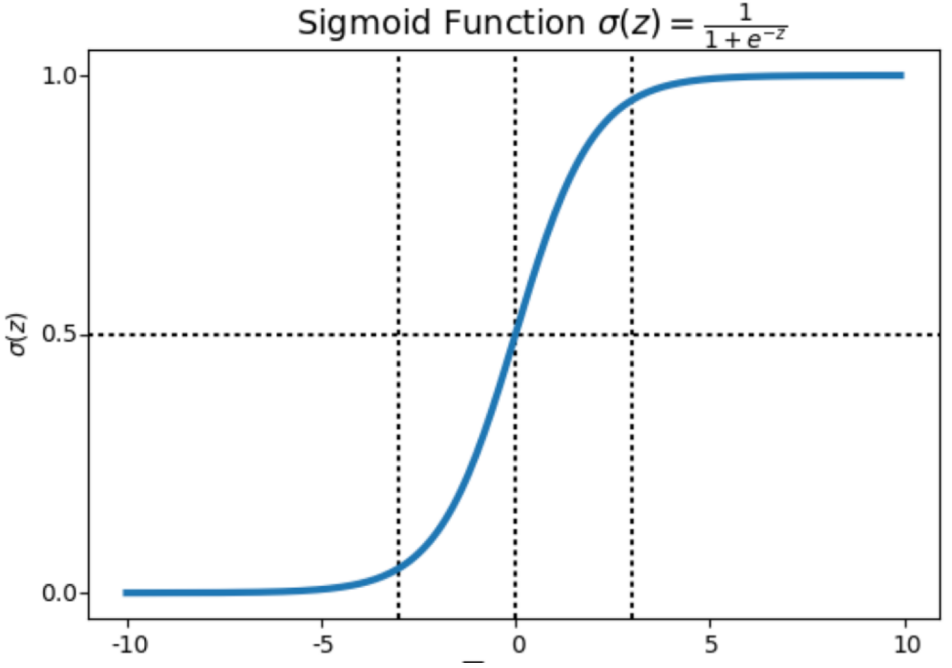
Logistic Regression

간단한 Logistic Regression 예시 : Binary Classification

Linear Regression의 경우 output의 범위 제한이 없음 $(-\infty, \infty)$

Regression의 output이 0과 1 사이의 확률이면?

Sigmoid function : $\frac{1}{1+e^{-z}}$

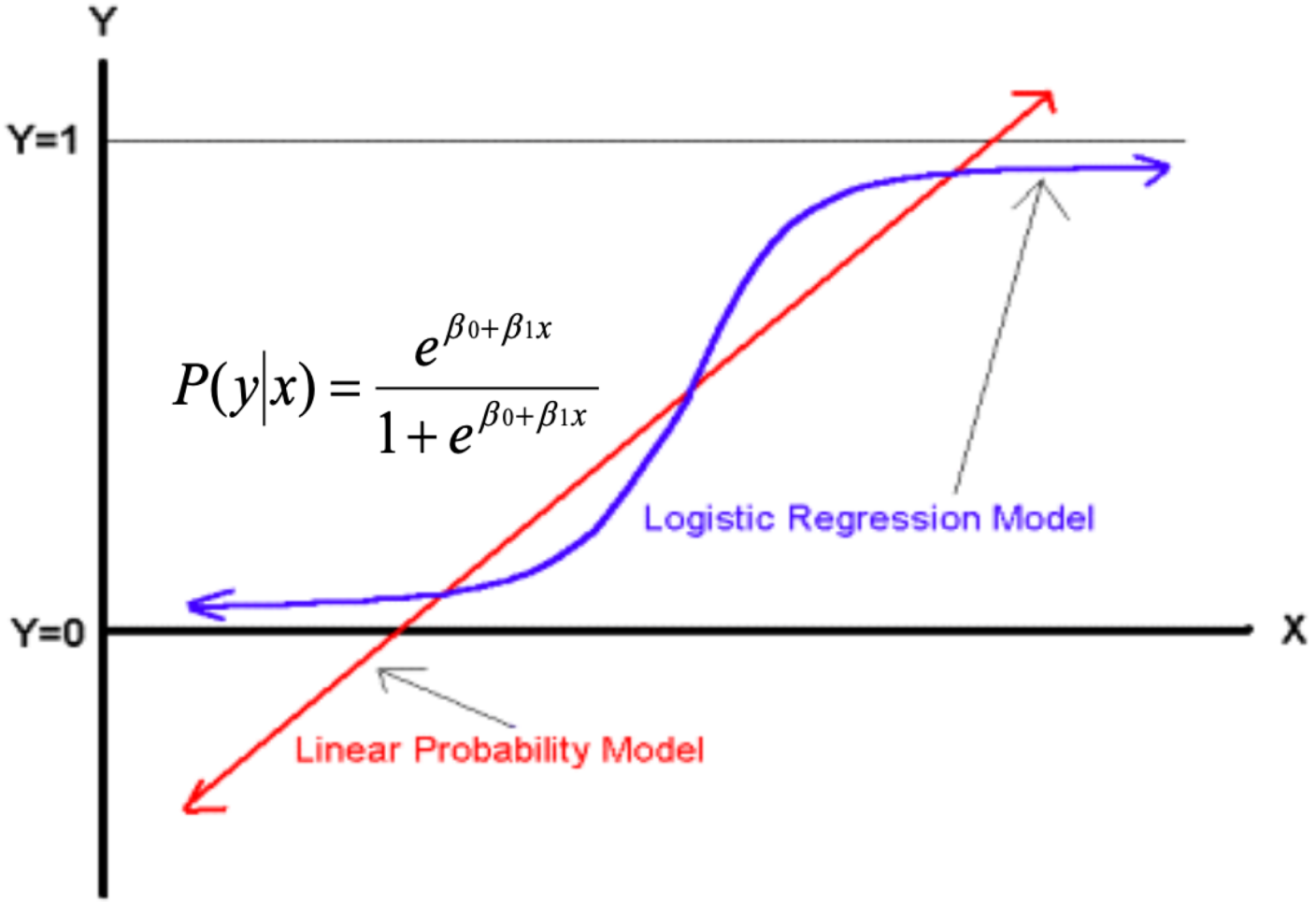


Logistic Regression

x가 주어졌을 때 y=1 클래스에
해당할 확률로 생각!

$$P(y=1|\boldsymbol{x}) = \sigma(\boldsymbol{\beta}^T\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}}$$

Linear Regression과 마찬가지로,
β값을 구하는 것이 목표



Practice

https://github.com/SLCFLAB/Data-Science-Python/blob/main/Day%208/8_1.Regression.ipynb

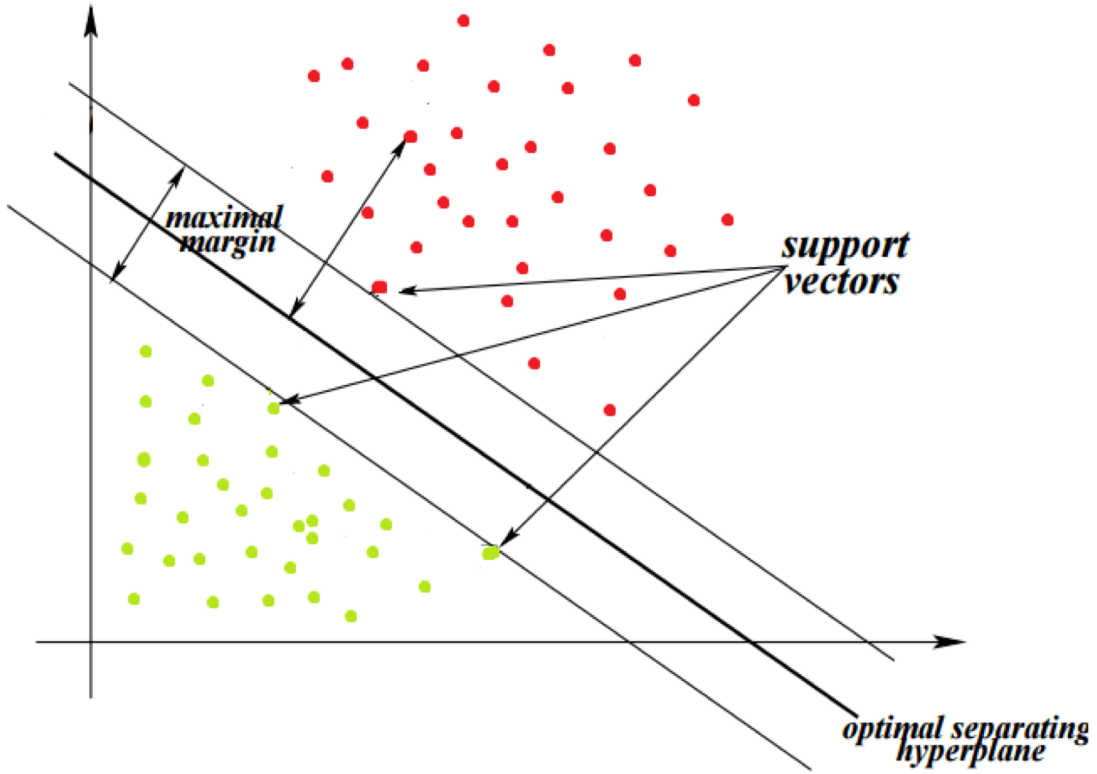
Support Vector Machine

Support Vector Machine

What is SVM (Support Vector Machine)?

Supervised learning models for classification and regression problems

Find a hyperplane that maximizes the margin of the classifier



Source: tinyurl.com/43544syy

Support Vector Machine

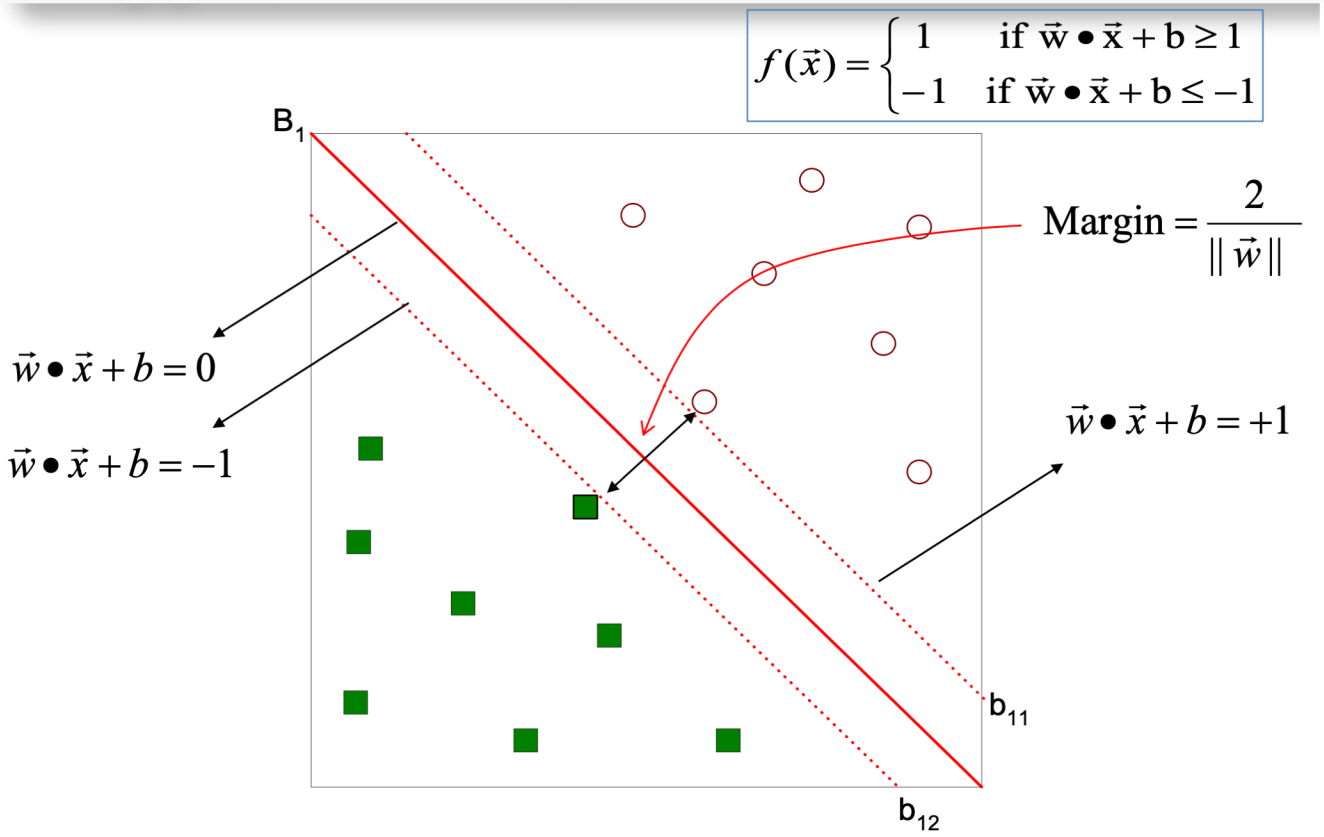
$w \cdot x + b = -1$

$w \cdot x + b = +1$

$w \cdot x + b = 0$ 은 평면

$w \cdot x + b = \pm 1$ 의 바깥쪽을 지나게 함

$w \cdot x + b = \pm 1$ 를 지나는 점들 – "Support vector"



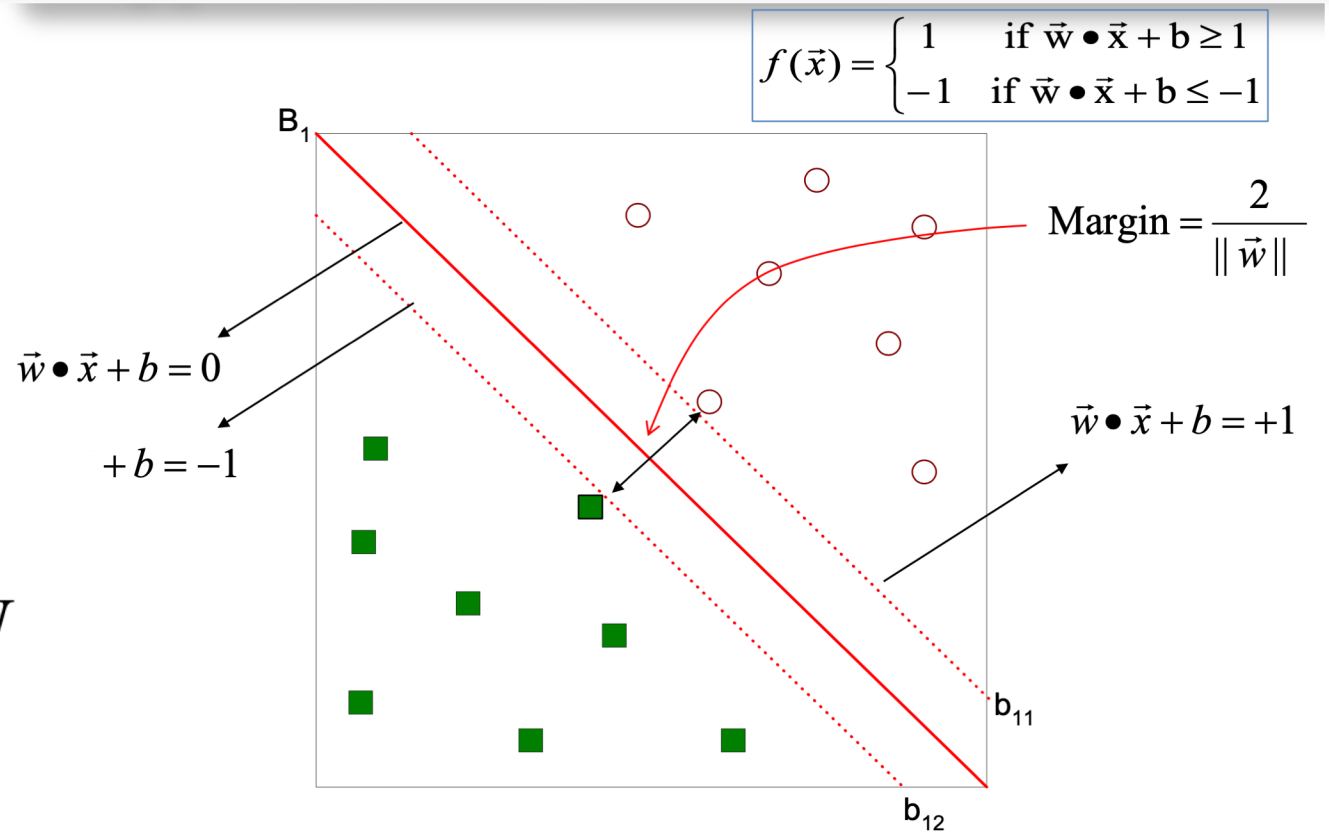
Support Vector Machine

Margin = $\frac{2}{\|w\|}$

maximize margin ⇔ minimize $\|w\|$

$$\min_{w,b} \quad \Phi(w) = \frac{1}{2} w^T w$$

$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1, \quad i = 1, \ldots, N$$



Support Vector Machine

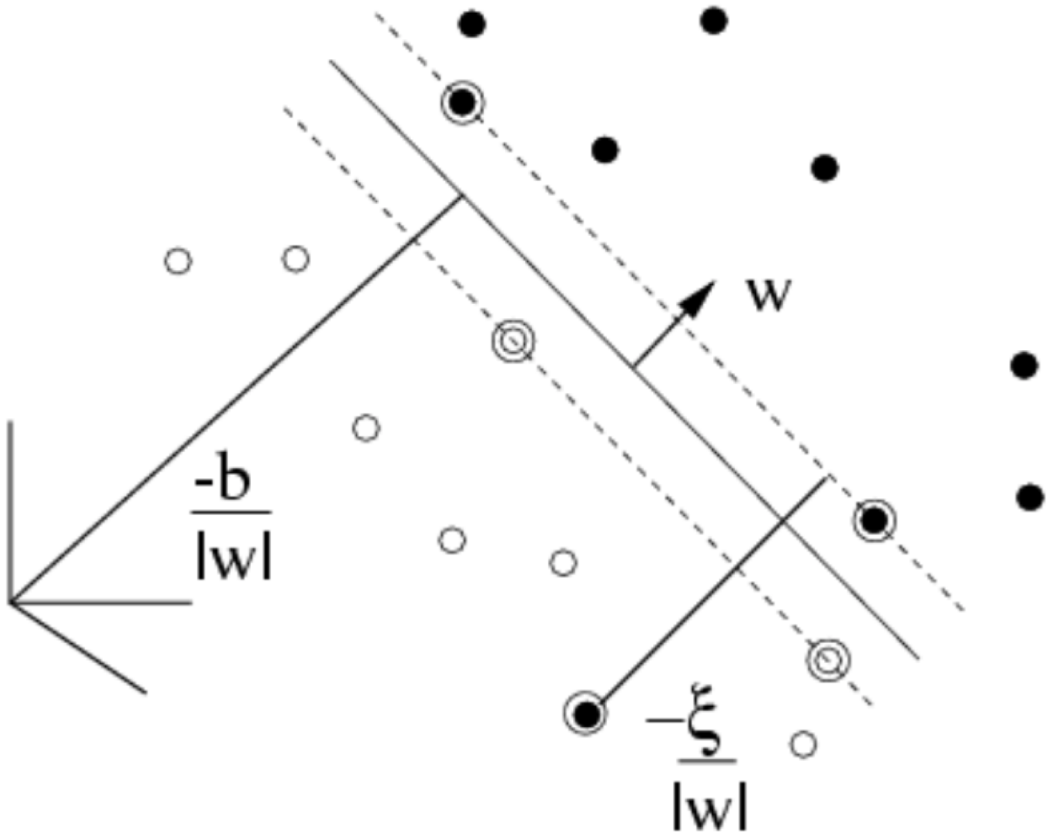
Not linearly separable ?

Consider slack variables :ξ

Allow for outliers

Objective function

$$\frac{\|w\|^2}{2} \longrightarrow \frac{\|w\|^2}{2} + C\left(\sum_i \xi_i\right)^k$$



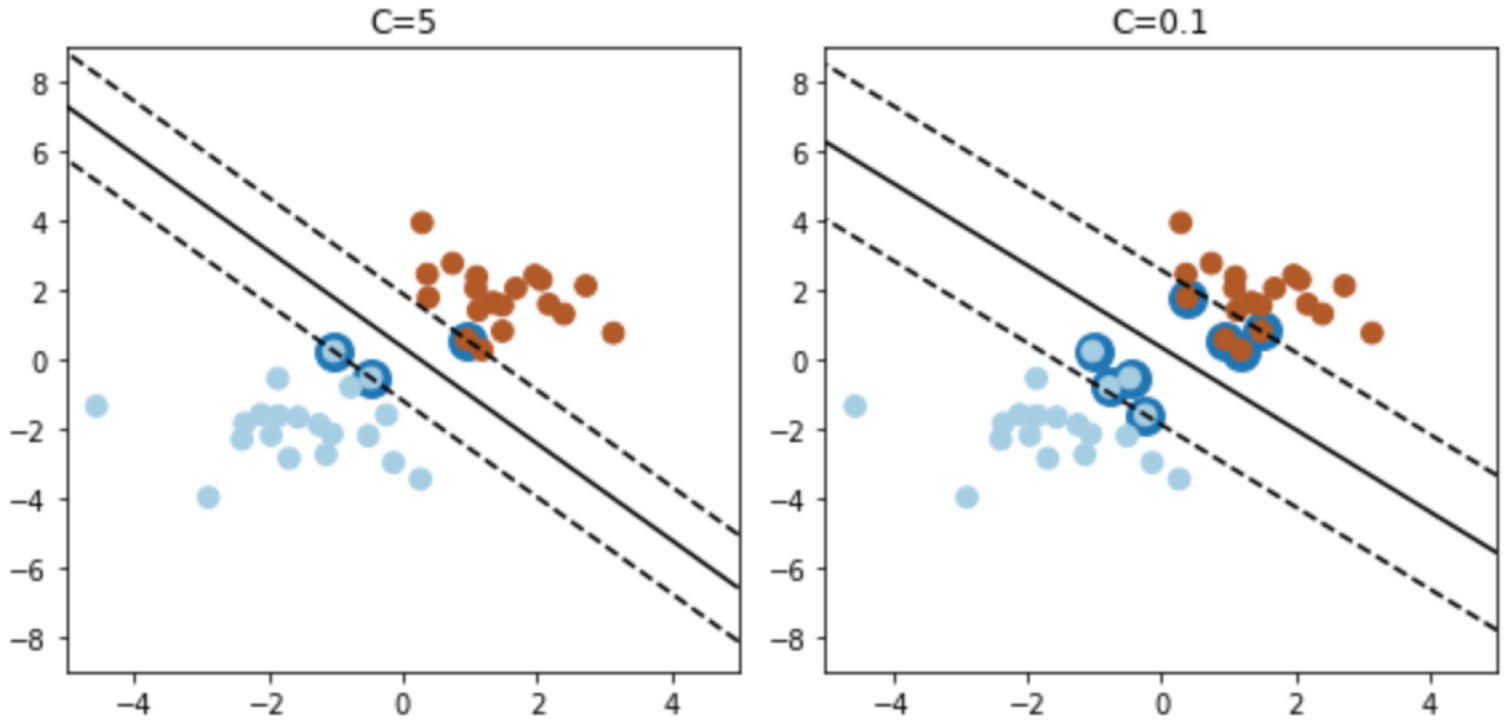
Support Vector Machine

Objective function

$$\frac{\|w\|^2}{2} \longrightarrow \frac{\|w\|^2}{2} + C\left(\sum_i \xi_i\right)^k$$

C가 크다 => slack variable들 허용하지 않겠다 => outlier들의 수가 적다

C가 작다 => slack variable들 허용하겠다=> outlier들의 수가 많다

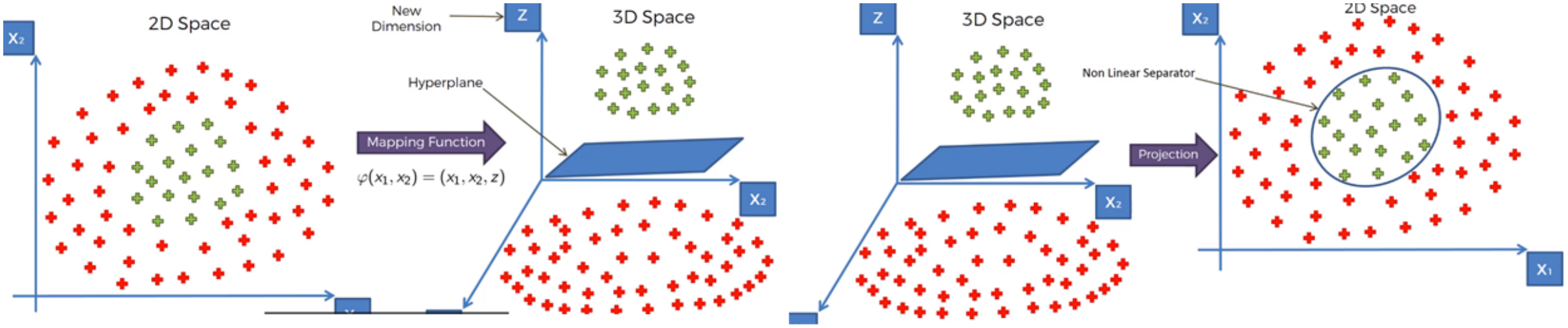


Support Vector Machine

Kernel SVM

고차원 공간으로 바꾸어서 linearly separable 하게끔 만듬

kernel 종류에 따라 다양 : polynomial kernel, rbf kernel, exponential kernel, ...



Practice

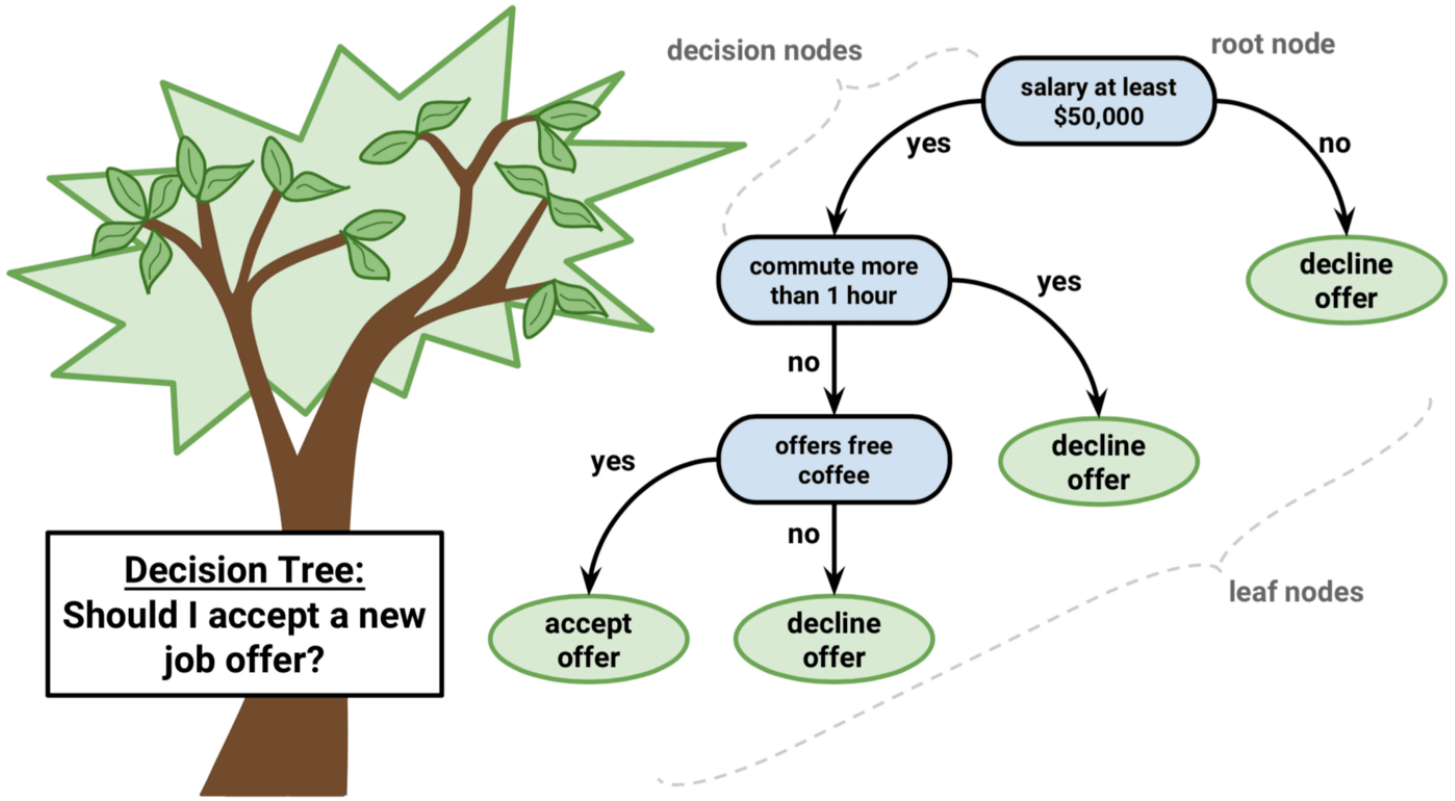
https://github.com/SLCFLAB/Data-Science-Python/blob/main/Day%208/8_2.SVM.ipynb

Decision Tree

Decision Tree

Non-parametric supervised learning method for classification

predicts the value of a target variable by learning simple decision rules inferred from the data features



Decision Tree

Advantages

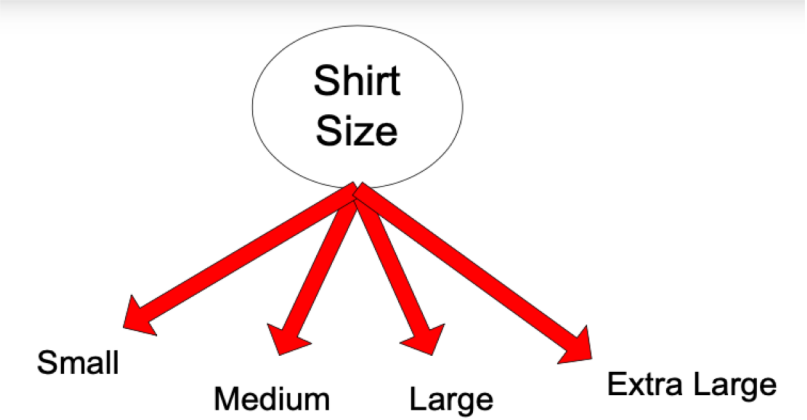
- Simple to understand and to visualize
- Cost of using the tree is logarithmic in the number of data points
- Able to handle both numerical and categorical data

Disadvantages

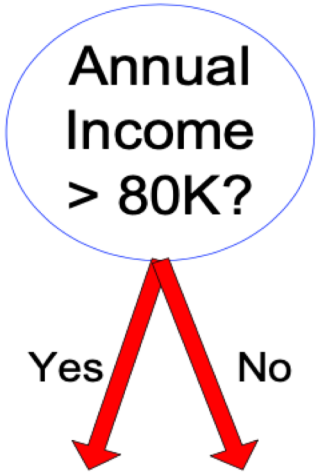
- Overfitting problem
- Does not take into account interactions between attributes

Decision Tree

Multi way split



Binary split



Decision Tree

How to determine the best split?

Calculate information gain

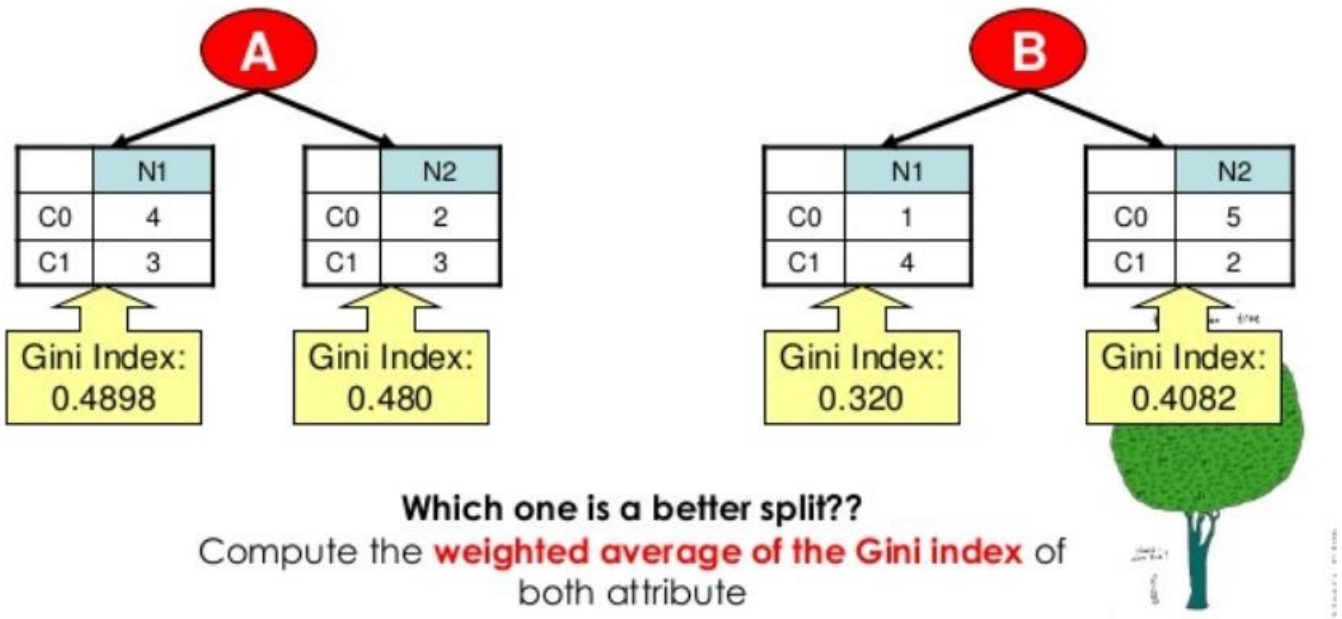
Gini index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

Entropy

Misclassification error

Suppose there are two ways (A and B) to split the data into smaller subset.



Which one is a better split??

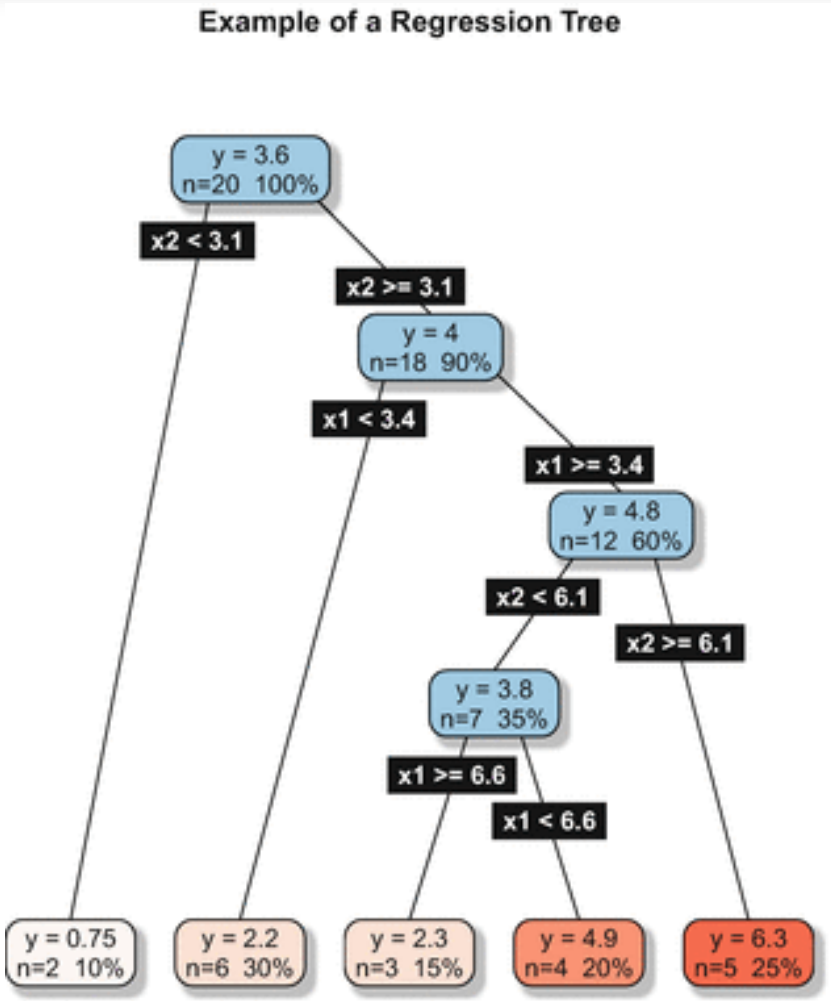
Compute the **weighted average of the Gini index** of both attribute

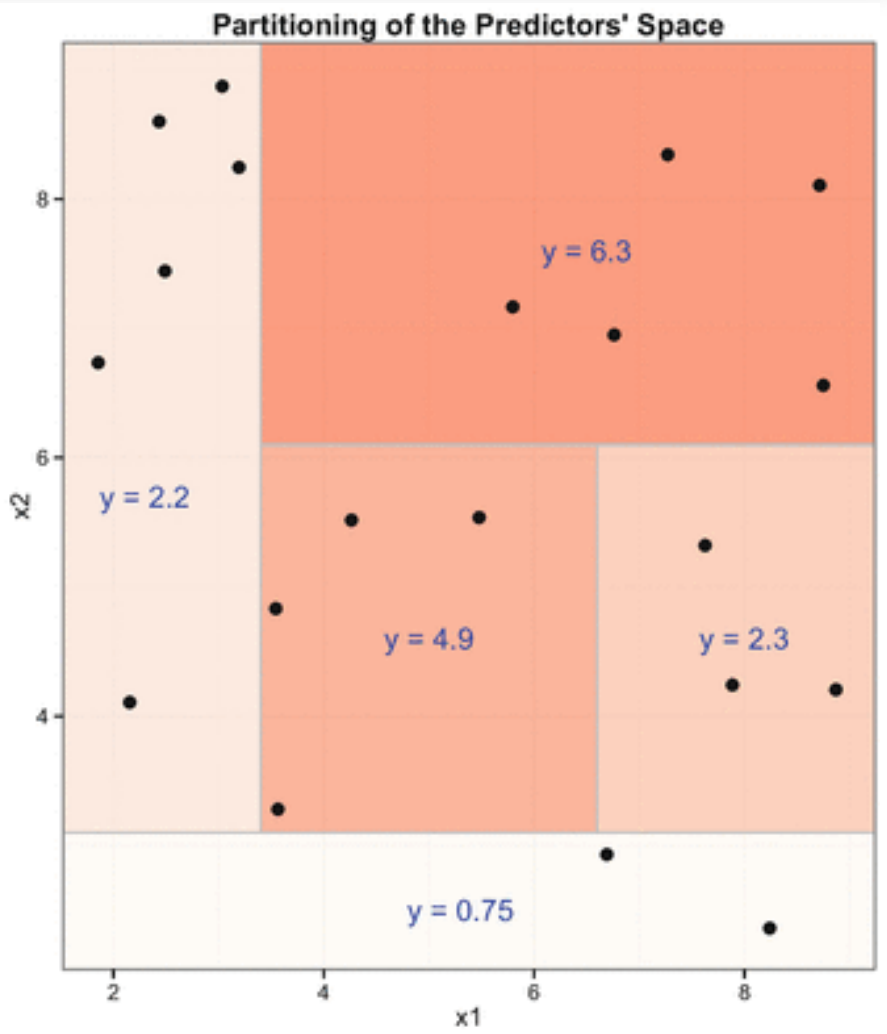
Regression Tree

Regression case

$Average(y|x_i \in R_m)$

속한 그룹에 있는 다른 데이터들의 y값의 평균으로 근사시킴

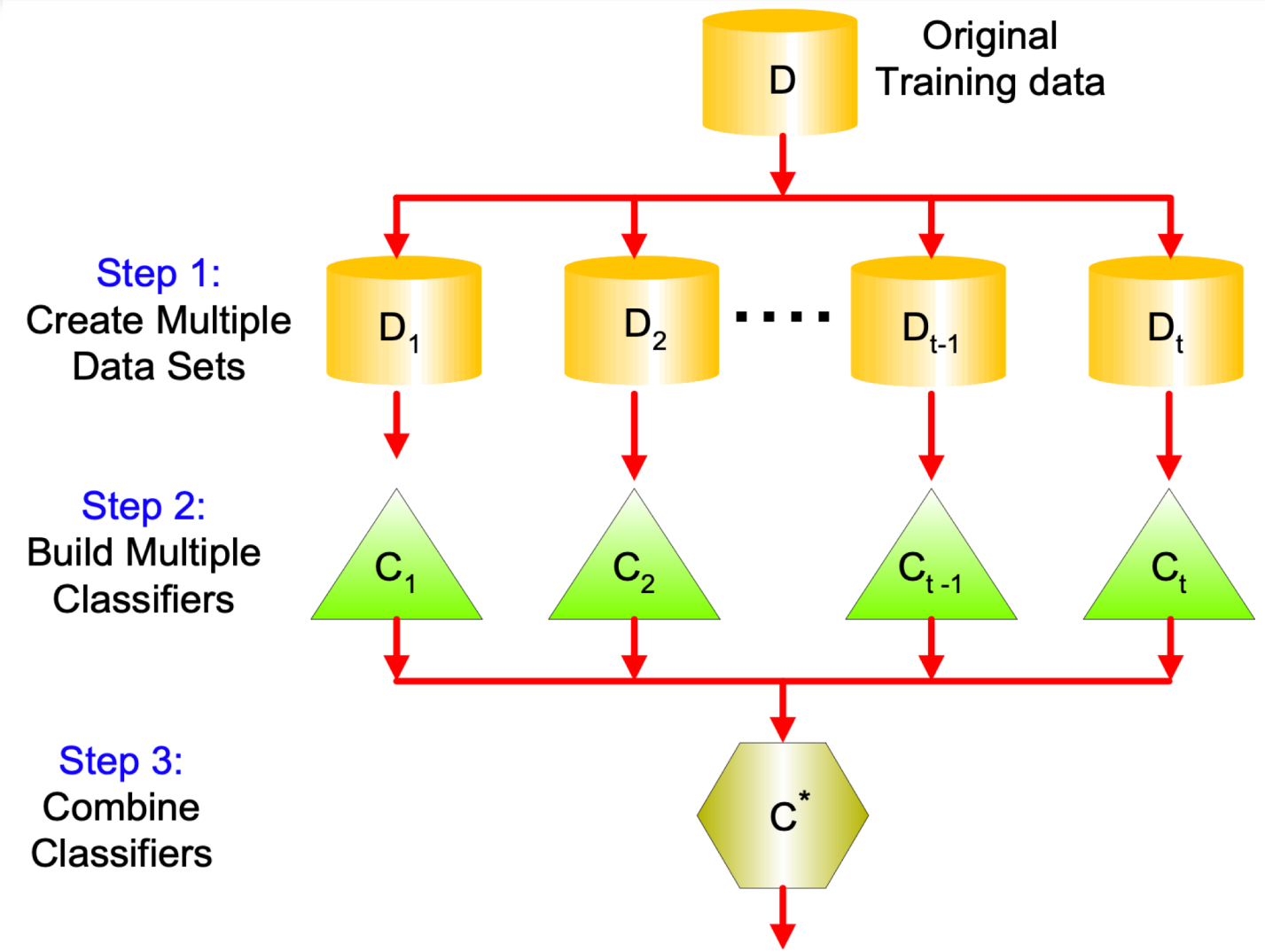




Ensemble

트레이닝 데이터를 여러 개로 나누어 각 데이터셋에 대해 모델 여러 개 학습

학습된 분류 모델들이 내는 결과를 종합 (평균, 다수결 등) 하여 최종 결과 선정



Bagging

Bootstrap Aggregation

입력 데이터를 모델 수만큼 나눈 뒤 각각 학습시킴

출력되어 나온 예측 값들로 voting하여 majority vote를 받은 값이 최종 예측값이 됨

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

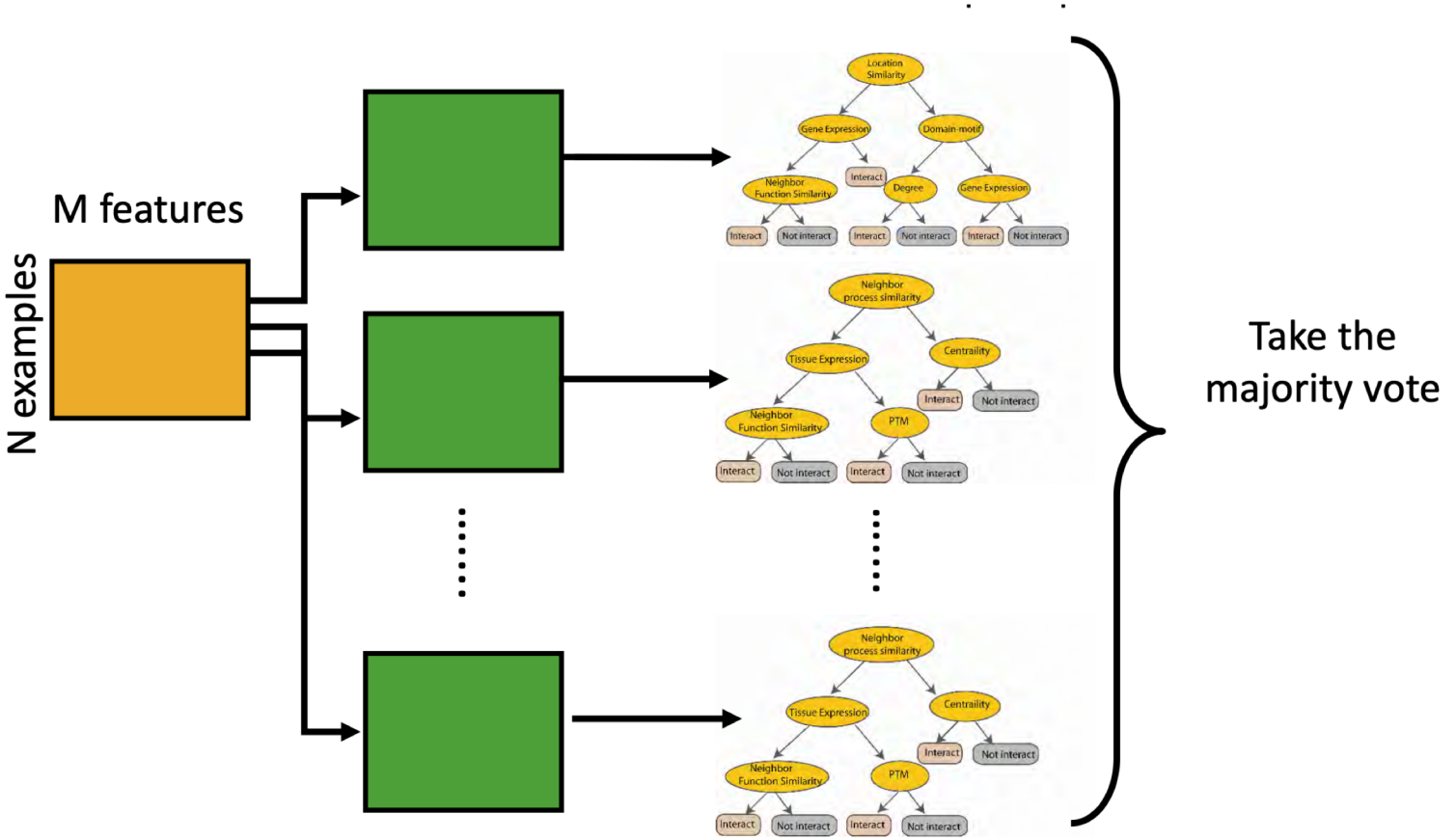
x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

Random Forest

bagging의 extension

각 데이터셋에 대해 모든 feature를 다 쓰는 것이 아니라, feature들 중 일부만을 랜덤하게 골라서 사용



Boosting

AdaBoost

Records that are wrongly classified will have their weights increased

Records that are classified correctly will have their weights decreased

e.g. data 4 is hard to classify during Round 1 and Round 2 => weights increased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

Boosting

AdaBoost

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

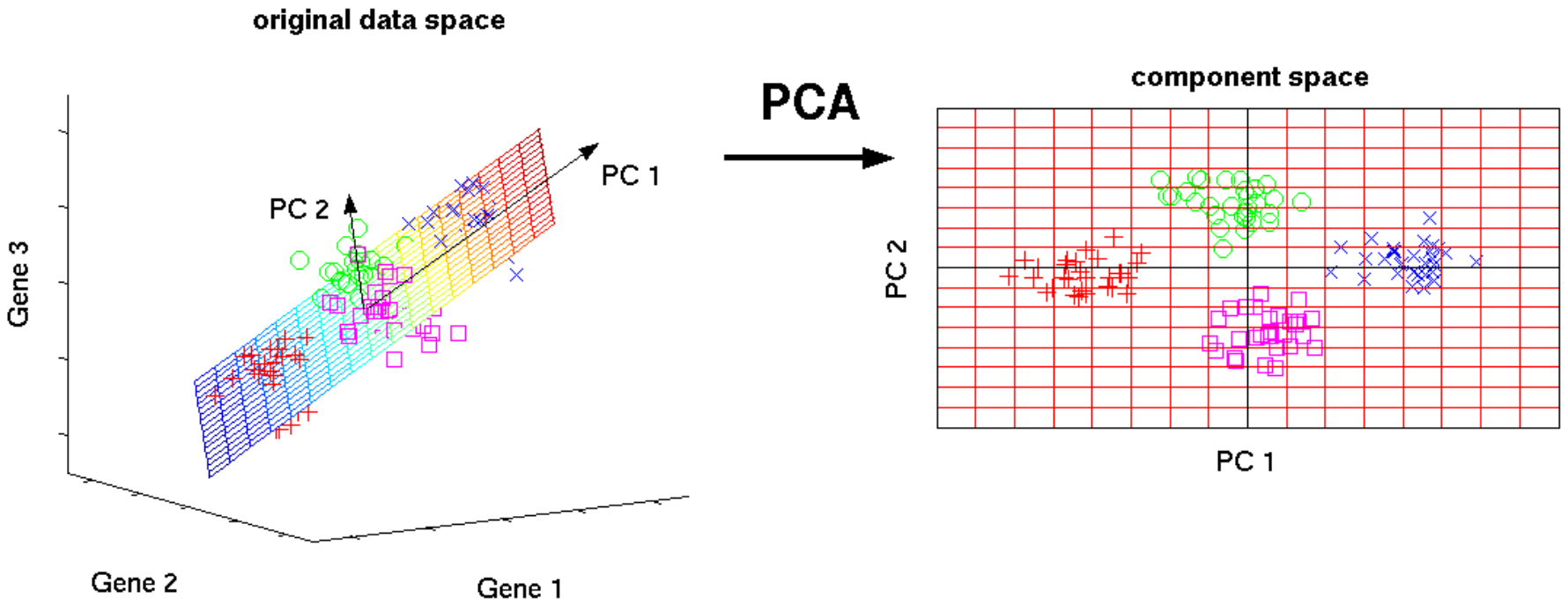
x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

Principal Components Analysis

PCA

Dimensionality-reduction technique

"Curse of Dimensionality" – how to handle it?



PCA

Variance를 최대화하는 축을 찾자 (SVD 분해와 관련)

$$\mathbf{X}V = U\Sigma \;\rightarrow\; \mathbf{Z_c} = \mathbf{X}V_{:,1:c} = U\Sigma_{:,1:\mathbf{c}}$$

$$\mathbf{Z}_c = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_N^T \end{bmatrix} = \mathbf{X}V_{:,1:c} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{v}_1 & \mathbf{x}_1^T\mathbf{v}_2 & \cdots & \mathbf{x}_1^T\mathbf{v}_c \\ \mathbf{x}_2^T\mathbf{v}_1 & \mathbf{x}_2^T\mathbf{v}_2 & \cdots & \mathbf{x}_N^T\mathbf{v}_c \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_N^T\mathbf{v}_1 & \mathbf{x}_N^T\mathbf{v}_2 & \cdots & \mathbf{x}_N^T\mathbf{v}_c \end{bmatrix} = U\Sigma_{:,1:c} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \cdots & \sigma_c\mathbf{u}_c \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$

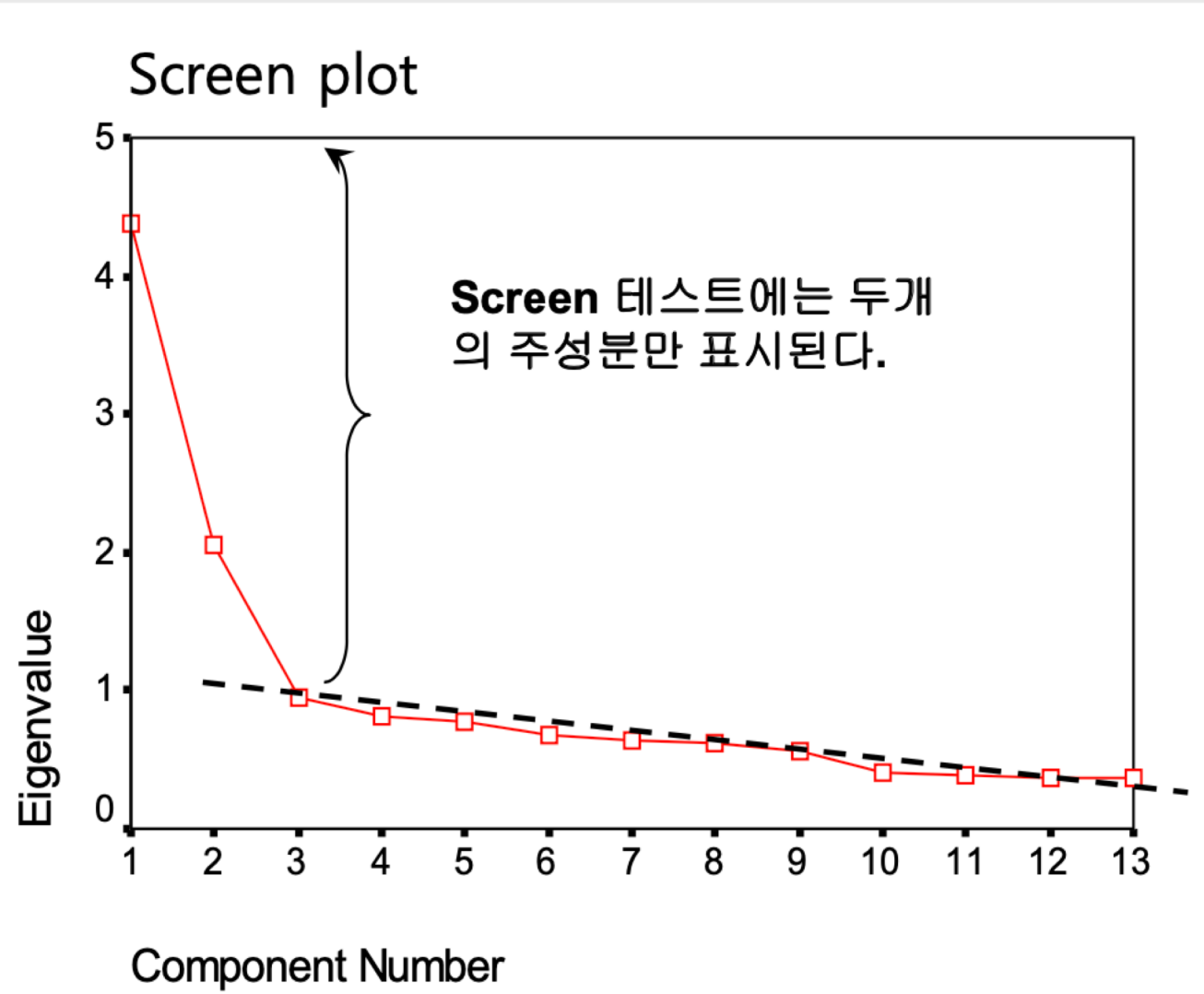
Variance를 최대화하는 축 순서대로 : SVD 분해 후 나오는 $\sigma_1 u_1$ (첫번째 축), $\sigma_2 u_2$ (두번째 축) …

축들의 Variance의 합은 곧 "설명력"

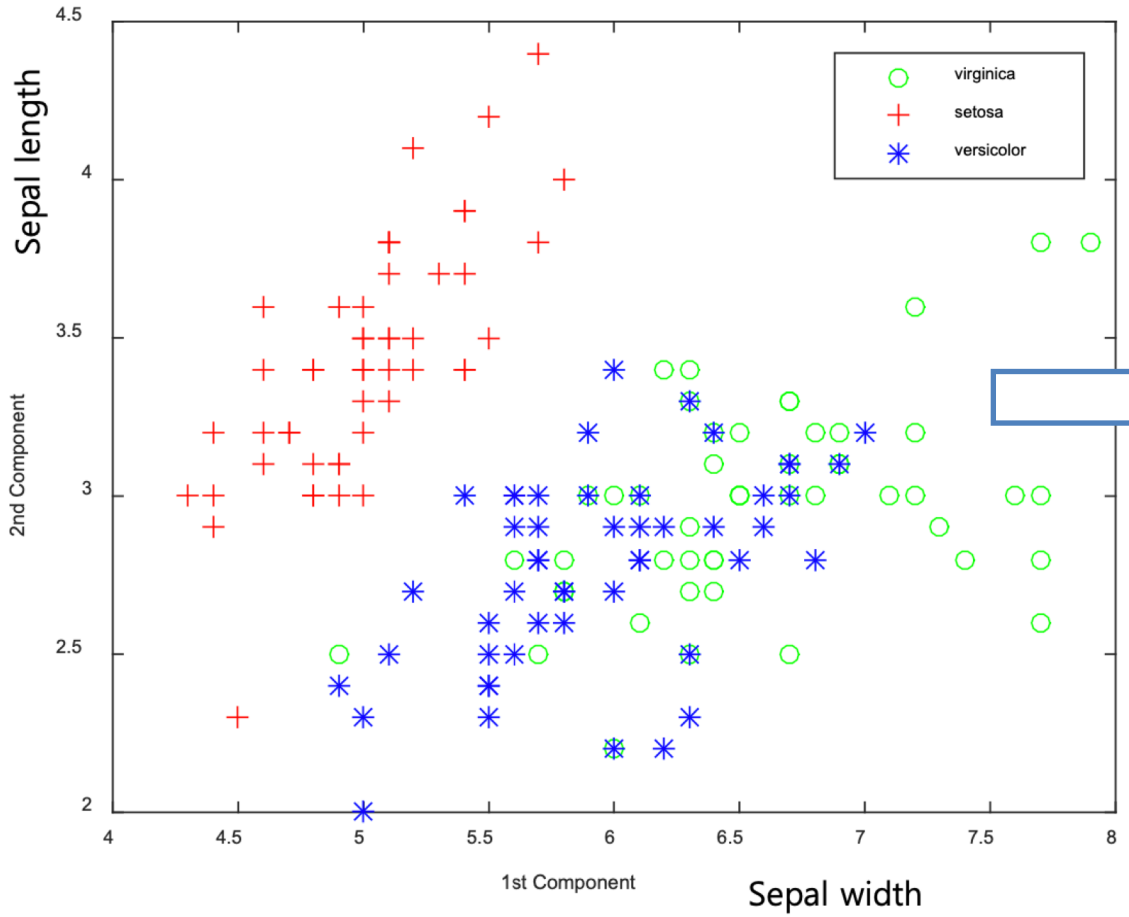
$var(z_1 + z_2 + \cdots)$ =Eigenvalue의 합과 같음

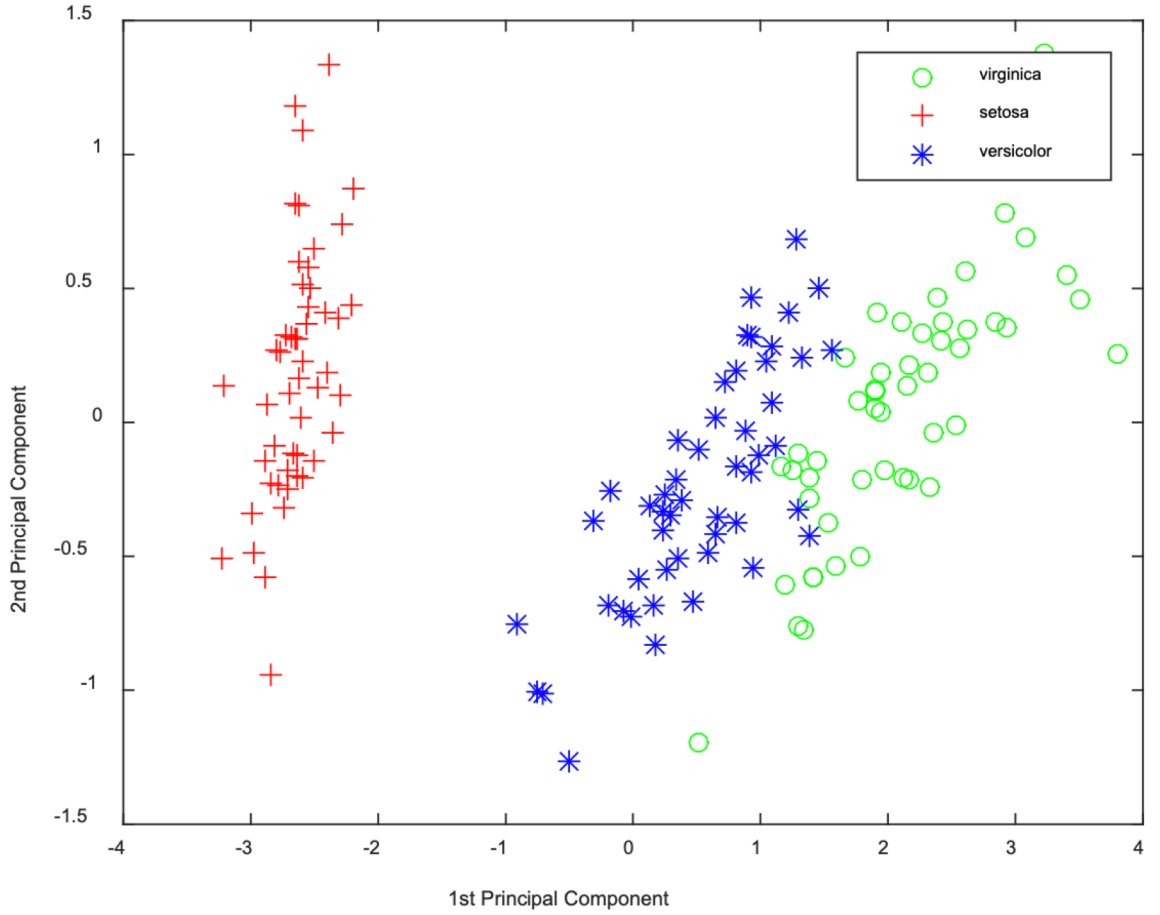
구한 각 축은 x_k 들의 lienar combination

PCA


Screen plot
Screen 테스트에는 두개의 주성분만 표시된다.
Eigenvalue
0
1
2
3
4
5
1
2
3
4
5
6
7
8
9
10
11
12
13
Component Number

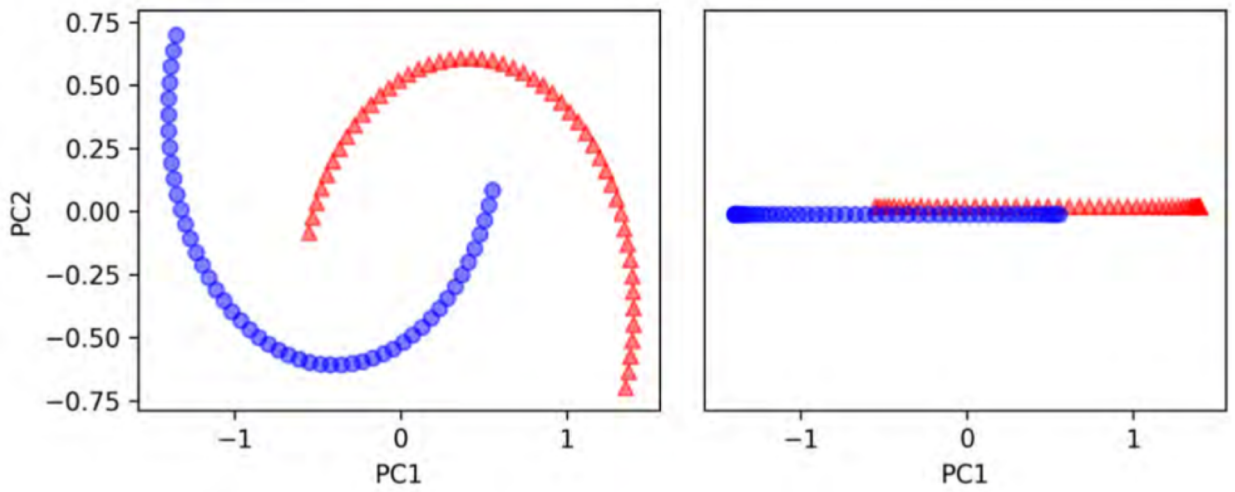
PCA


virginica
setosa
versicolor
Sepal length
2nd Component
1st Component
Sepal width
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8

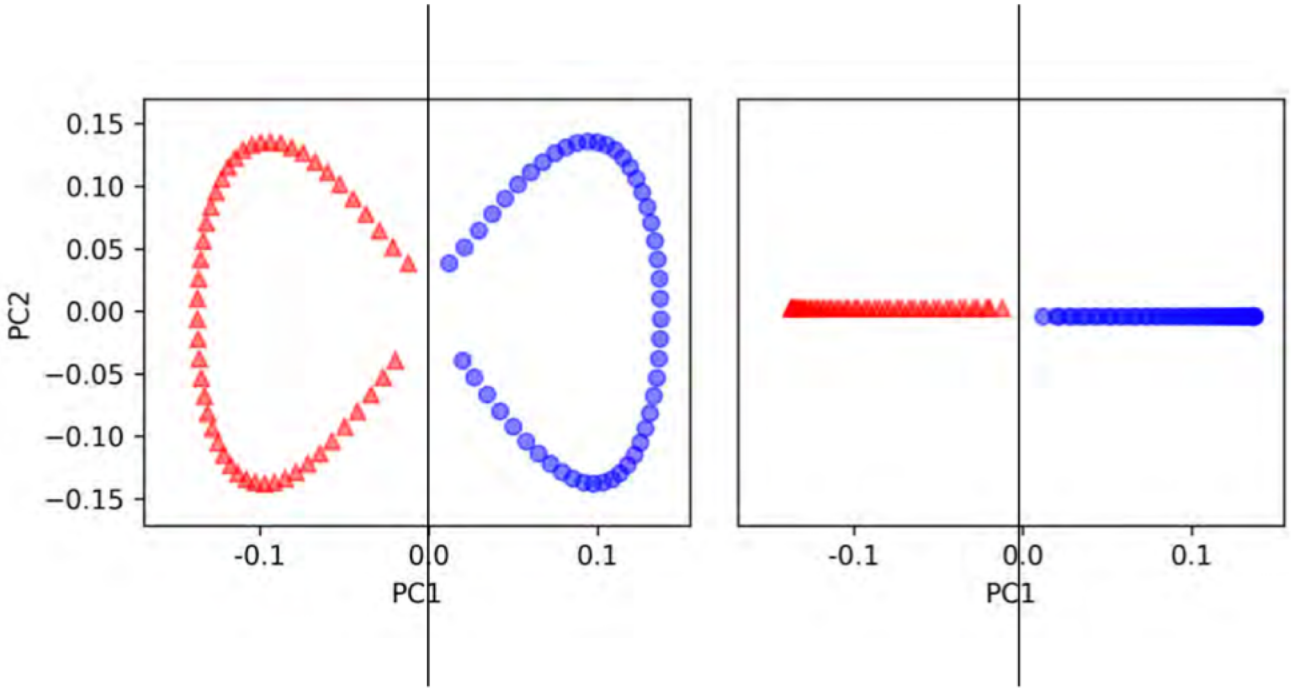

virginica
setosa
versicolor
2nd Principal Component
1st Principal Component
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4

Kernel PCA

Standard PCA로 분리하기 어려운 경우



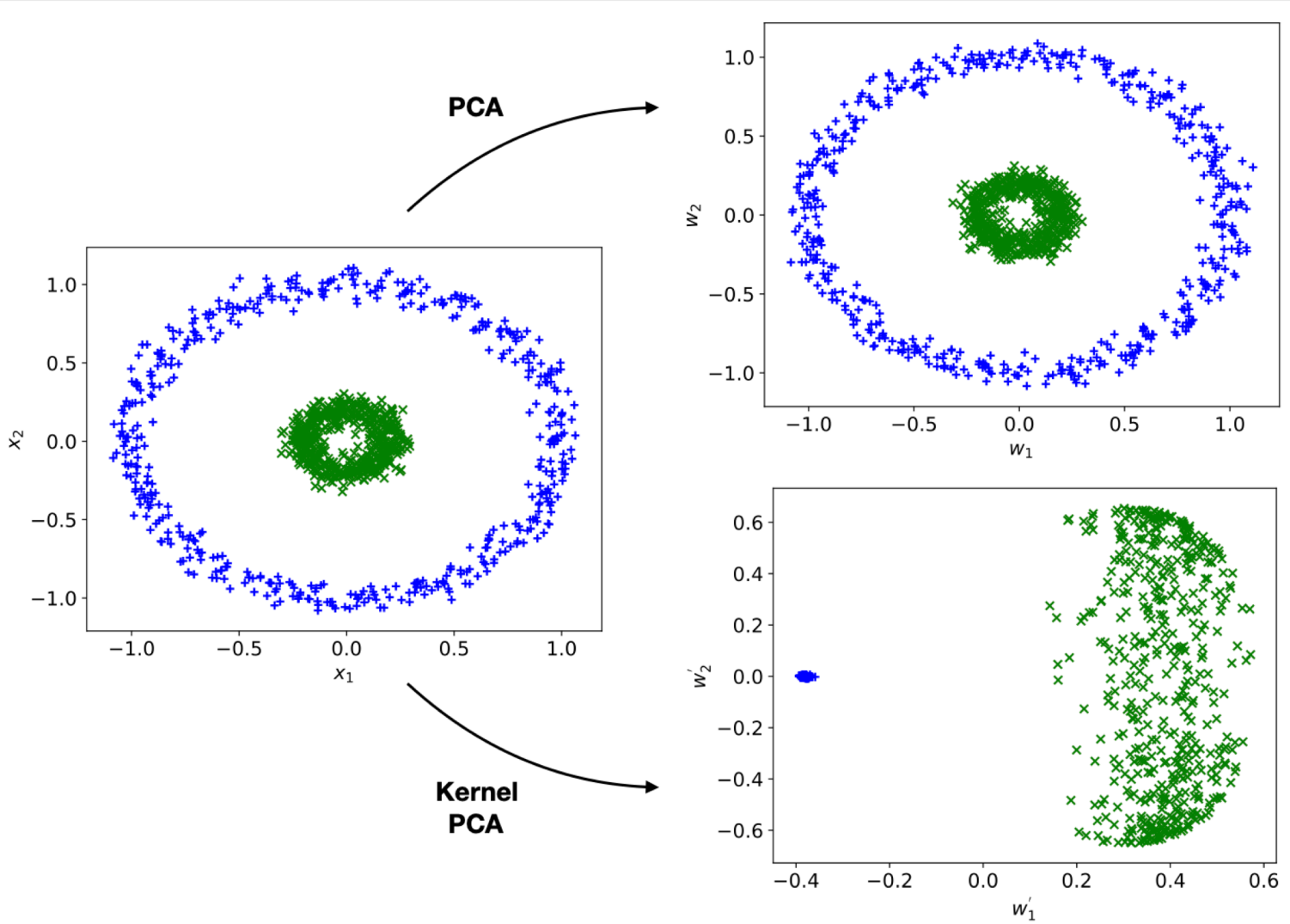
Kernel PCA



Idea : Kernel SVM과 유사하게, kernel 사용하여 고차원으로 mapping 시킨 후 고차원에서 PCA 진행

Kernel PCA

Other examples



References

https://galton.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L24.pdf

https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Hands-on machine learning

https://medium.com/pursuitnotes/day-12-kernel-svm-non-linear-svm-5fdefe77836c