SVM

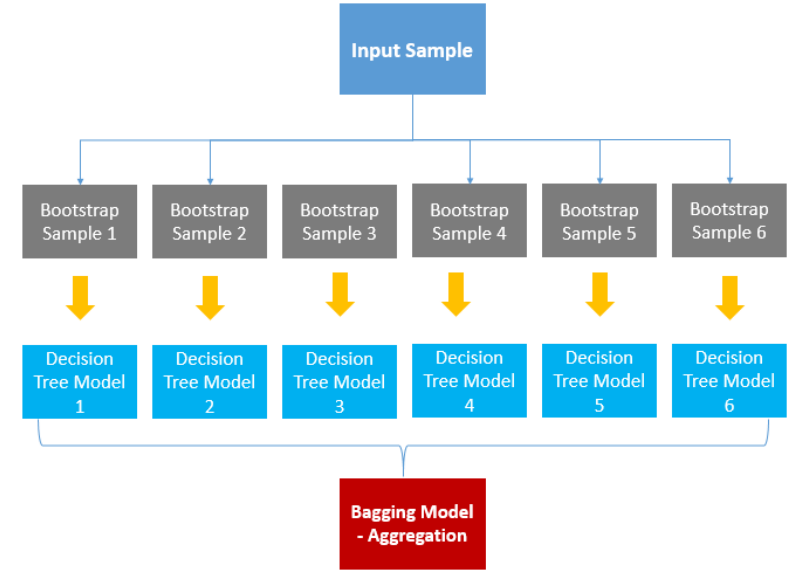
Boosting , Stacking

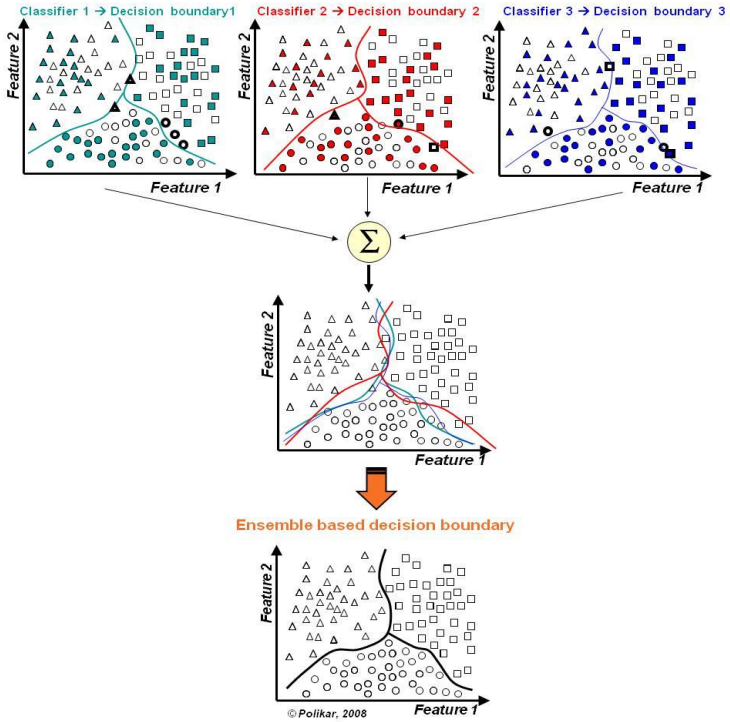
Boosting

- Ensemble
  - 일반적으로 한 개의 모델을 활용하여 결과를 예측
  - Weak Learner를 여러 개 결합한다면 Single Learner보다 더 나은 성능을 얻을 수 있다는 아이디어에서 출발
  - Bagging, Boosting, 그리고 Stacking 모두 이 아이디어에서 출발

Boosting

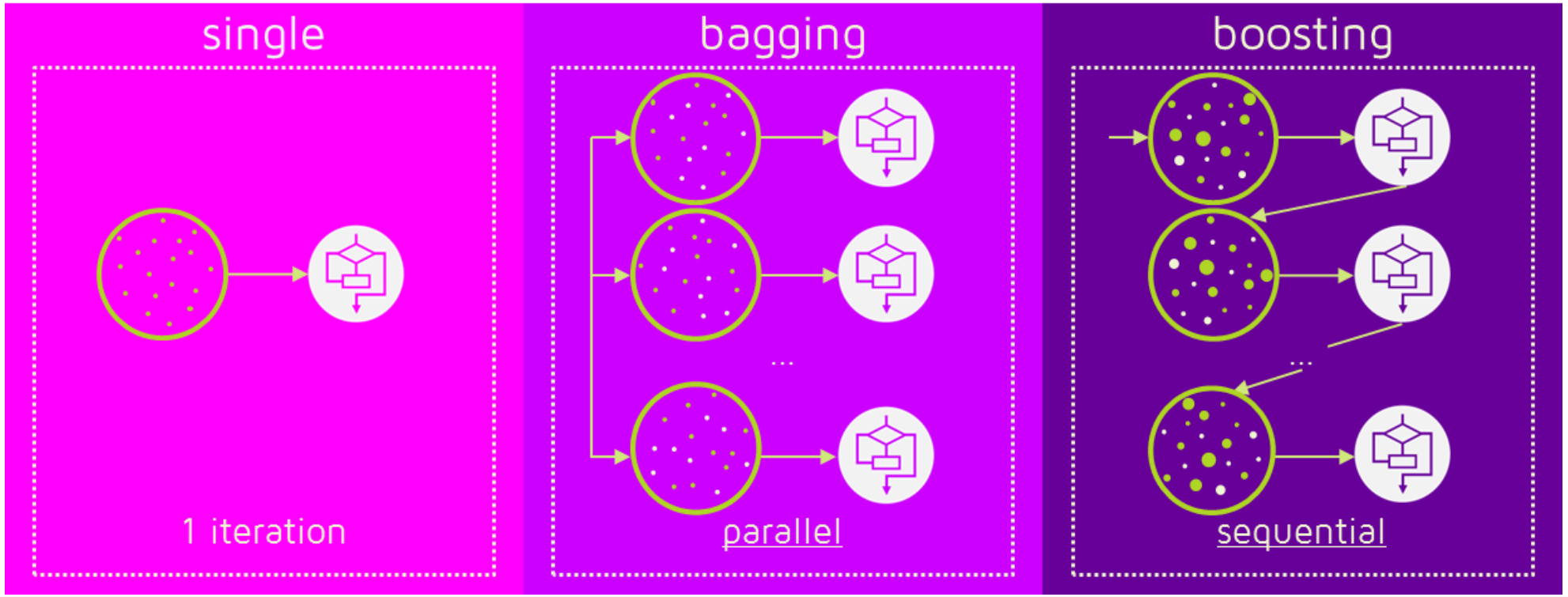
- Bagging이란?
  - 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과물을 집계하는 방법
  - 복원 추출(Bootstrap) + 집계(Aggregation) -> Bagging
  - Categorical Data : Voting
  - Continuous Data : Average





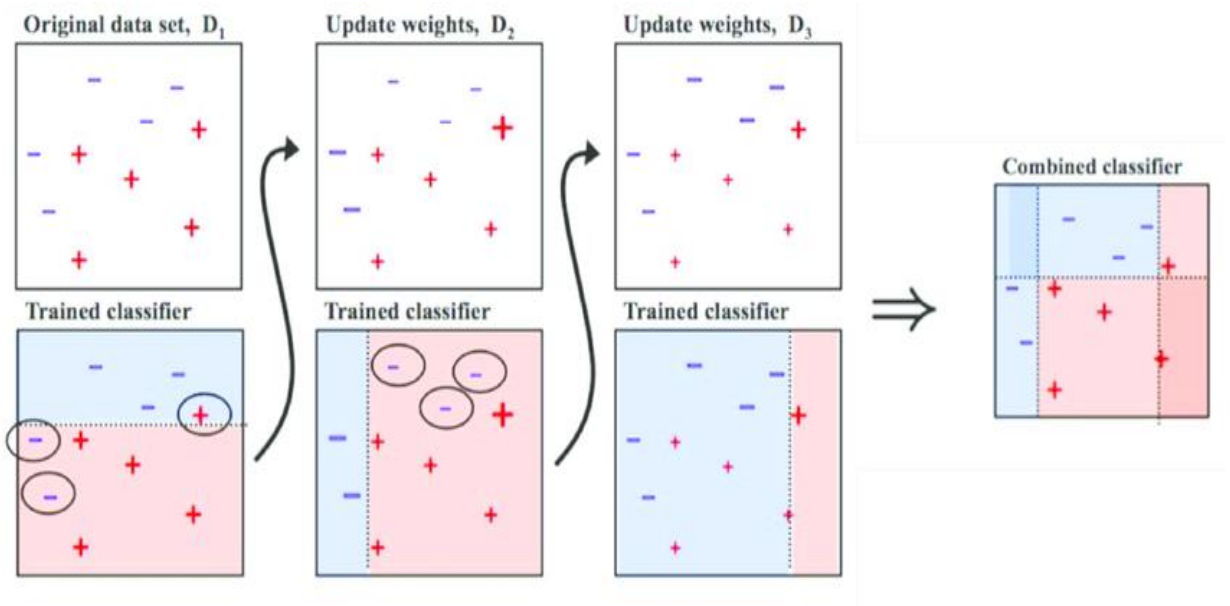
Boosting

- Boosting이란?
  - Bagging과는 다르게 순차적 방법을 통해 이전 학습의 결과가 다음 학습의 결과에 영향



Boosting

- Sample을 Bagging과 마찬가지로 복원 추출
- 첫번째 Sample의 예측 결과에 따라 오답에 대해 가중치를 적용하여 다음 Sample을 순차적으로 학습

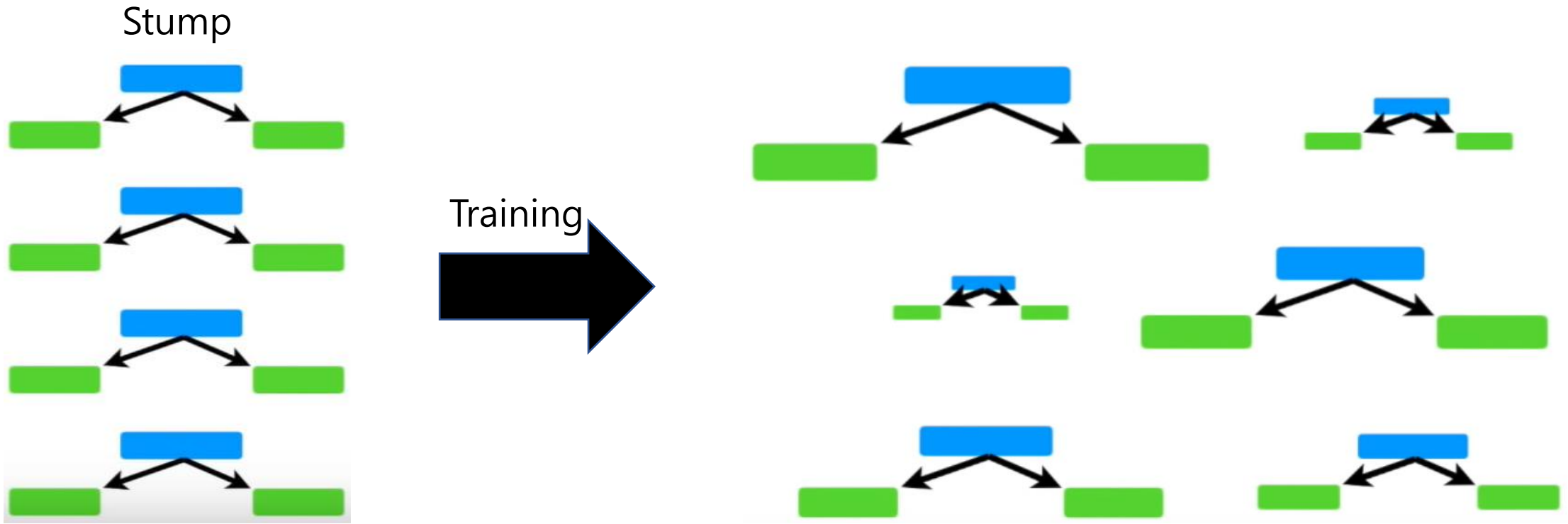


Boosting

- 장점
  - Error에 대해 가중치를 주어 학습하기 때문에 작은 Error
- 단점
  - 순차적으로 진행되기 때문에 느린 속도
  - 높은 확률의 Overfitting
- 종류
  - Boosting 계열 방법론에는 대표적으로 AdaBoost, Gradient Boost 등이 존재

AdaBoost(=Adaptive Boosting)

- AdaBoost는 Stump를 이용하여 학습을 진행



➢ 각각의 Stump에 가중치를 두는 방향으로 학습을 진행

AdaBoost(=Adaptive Boosting)

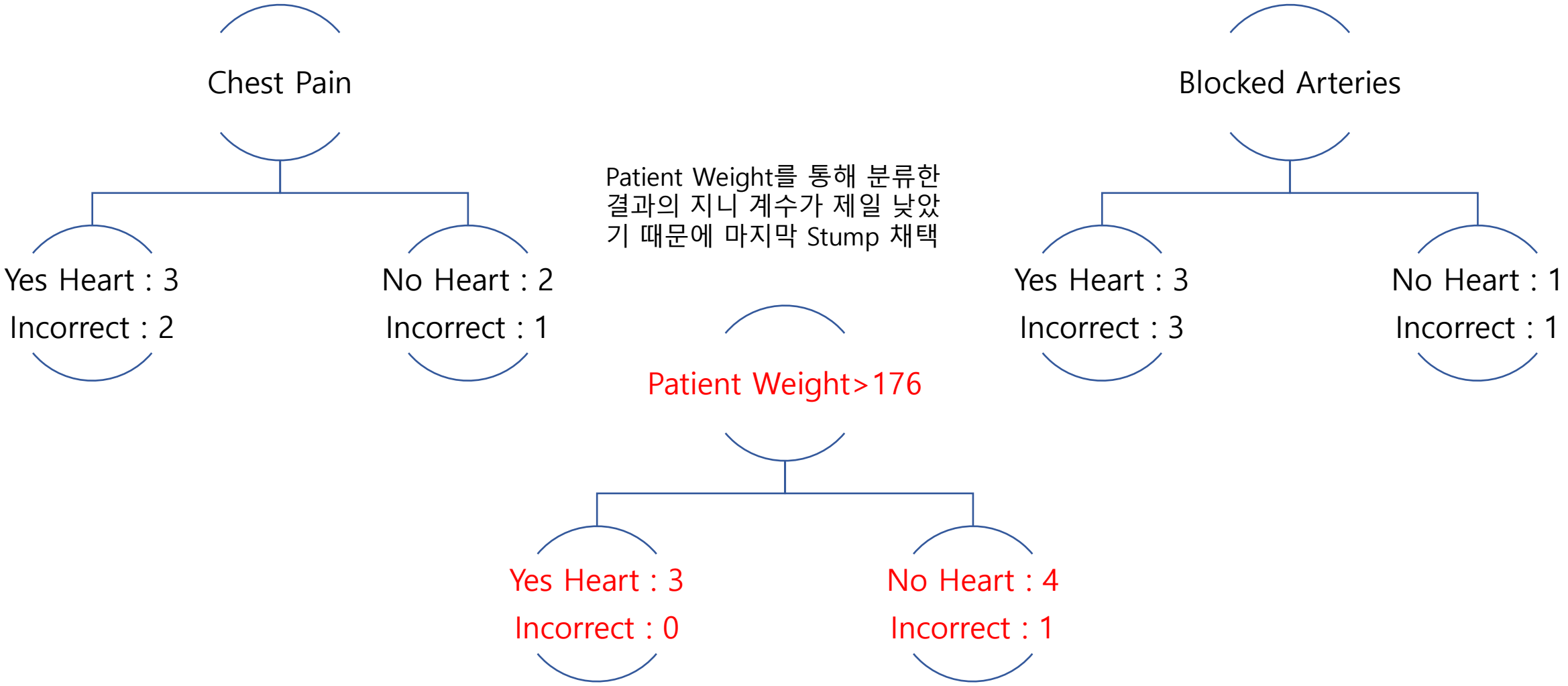
첫 샘플링할 때 각각의 데이터가
뽑힐 확률은 $\frac{1}{8}$로 모두 같다.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

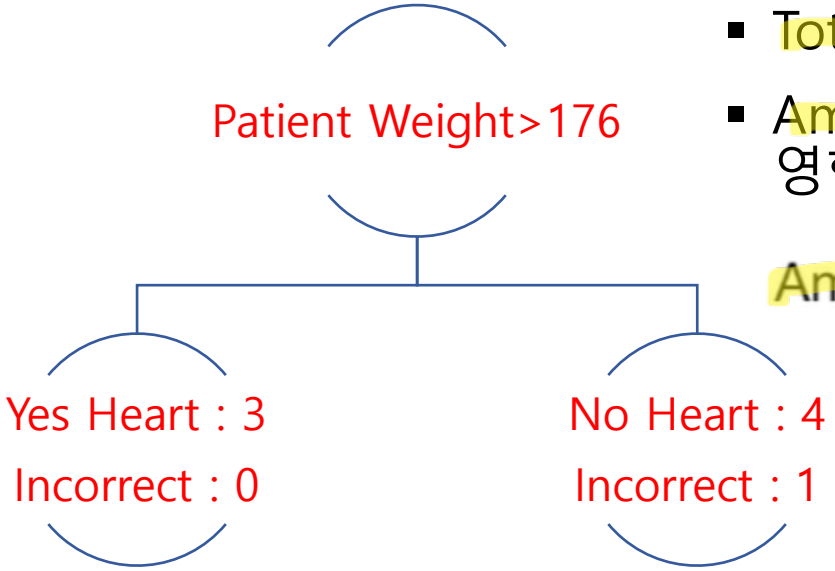
Chest Pain, Blocked Arteries, Patient Weight를 통해 Heart Disease의 발병 유무를 분류하는 Task

Stump를 활용하기 때문에 Heart Disease를 분류하기 위해 하나의 Column을 활용

AdaBoost(=Adaptive Boosting)



AdaBoost(=Adaptive Boosting)



- Total Error = 1/8
- Amount of Say란 최종 분류에 있어 해당 Stump가 얼마만큼의 영향을 주는지에 대한 지표

$$\text{Amount of Say} = \frac{1}{2}\log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

AdaBoost(=Adaptive Boosting)

- Total Error = 1/8
- Amount of Say란 최종 분류에 있어 해당 Stump가 얼마만큼의 영향을 주는지에 대한 지표

$$New\ Sample\ Weight = Sample\ Weight * e^{Amount\ of\ Say}$$

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	**Yes**	**167**	**Yes**	**1/8**
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

오분류된 데이터 -> 더 높은 가중치

$$New\ Sample\ Weight = 0.125 * e^{0.97} = 0.33$$

AdaBoost(=Adaptive Boosting)

- Total Error = 1/8
- Amount of Say란 최종 분류에 있어 해당 Stump가 얼마만큼의 영향을 주는지에 대한 지표

$$New\ Sample\ Weight = Sample\ Weight * e^{Amount\ of\ Say}$$

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8

제대로 분류된 데이터 -> 더 낮은 가중치

$$New\ Sample\ Weight = 0.125 * e^{-0.97} = 0.05$$

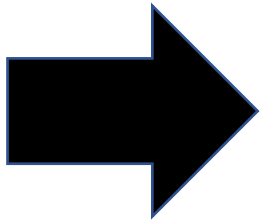
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Gradient Boosting

- Gradient Boosting은 말 그대로 **Gradient**를 이용해서 Boosting하는 방법론
- Target을 예측하는 것이 아닌 **Residual**을 줄여 나가는 방식으로 학습

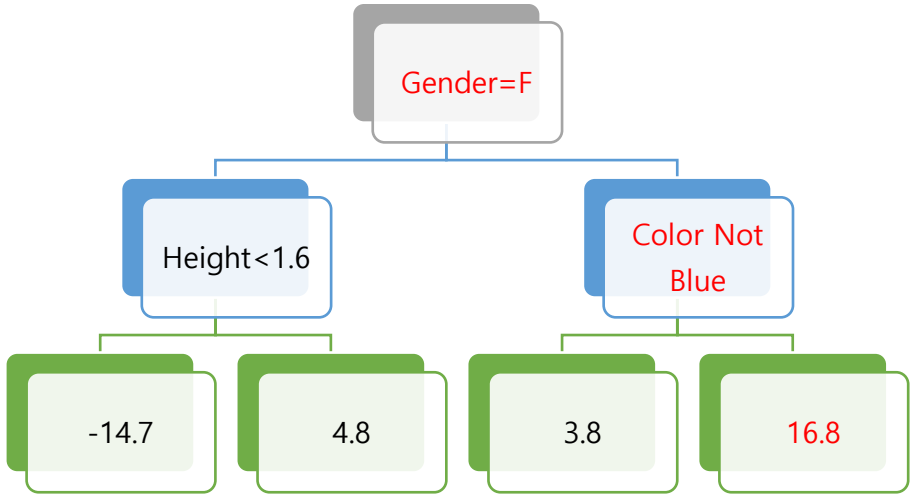
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Mean = 71.2



Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

Gradient Boosting



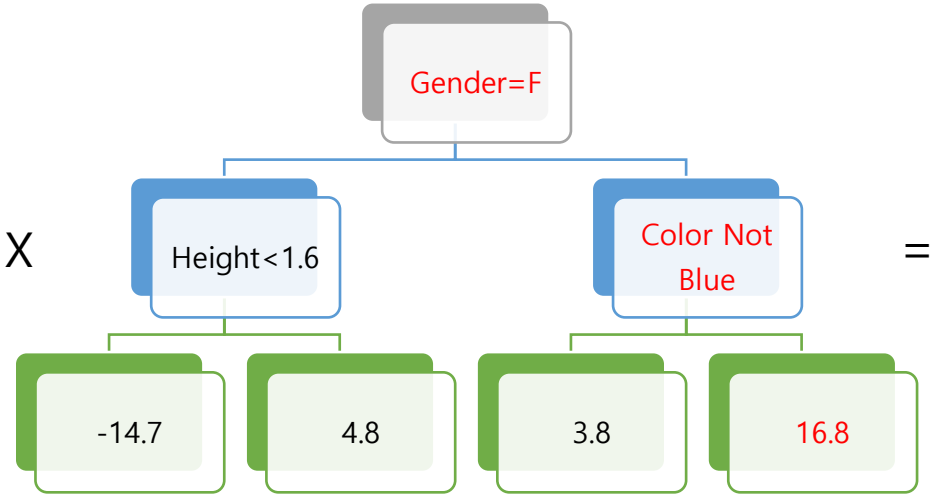
- Tree를 진행시키며 Leaf Node에 Residual를 기록
- 이 때, Gender=M, Color=Blue를 만족하는 데이터의 Residual를 구해보면 16.8이 나오는데 이 때 Residual와 처음 예측값을 더하면 원래 예측해야 되는 값인 88이 나오게 되는 것을 확인
- Train data에 과적합

71.2

Gradient Boosting

- Learning Rate를 이용해 Residual를 줄여 나가는 방향으로 학습을 진행

71.2 + Learning Rate X 0.1



=

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	15.1
1.6	Green	Female	76	4.3
1.5	Blue	Female	56	-13.7
1.8	Red	Male	73	1.4
1.5	Green	Male	77	5.4
1.4	Blue	Female	57	-12.7

- 업데이트된 Residual가 처음 존재하던 Residual와 비교했을 때 더 작아졌음을 확인

Gradient Boosting

- 순차적인 학습으로 인해 Bagging에 비해 많은 시간 소요
- Residual를 줄여 나가는 방식으로 학습이 진행되기 때문에 Overfitting이 일어나기 쉽고, 이런 Overfitting을 막기 위해 정밀한 Hyperparameter 조정이 필요
- Gradient Boosting을 활용한 모델로는 XGBoost, LGBM와 같은 방법론이 존재

XGBoost(=eXtreme Gradient Boost)

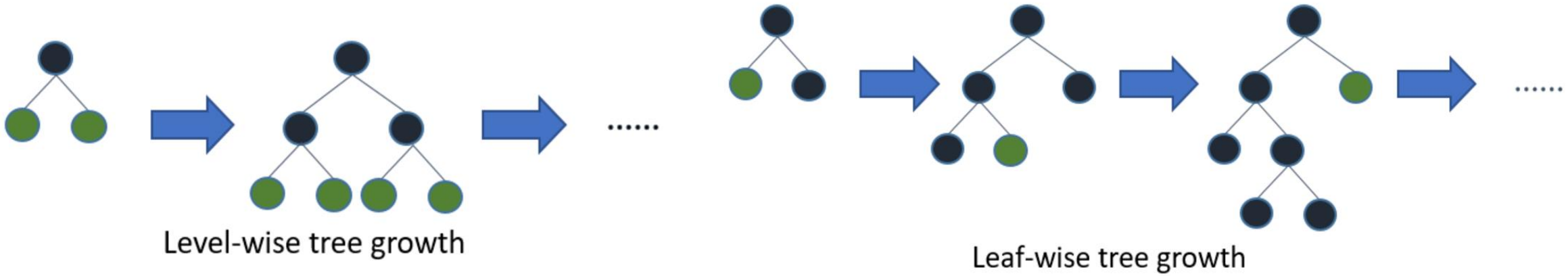
- 장점
  - 기존의 Gradient Boosting Model보다 빠른 속도
    - Tree Pruning
    - 내부적으로 병렬 처리 가능-> 속도 증가
  - Overfitting을 막기 위한 Hyperparameter들이 존재
    - GBM에는 Regularize 할 수 있는 장치가 존재 X
  - Early Stopping
  - CART(Classification And Regression Tree) 기반
    - 분류, 회귀 모두 사용 가능
  - 결측치에 민감 X

LGBM(=Light GBM)

- XGBoost 또한 여전히 시간이 오래 걸리는 문제가 발생
  - 또한 Grandient Boosting 계열 특성 상 Overfitting에 취약해 여전히 정밀한 Hyperparameter 조정이 필요
  - GridSearch와 같이 Hyperparameter를 여러 개 넣어 매번 XGBoost를 돌리게 되면 학습시간이 늘어나는 문제가 발생

LGBM(=Light GBM)

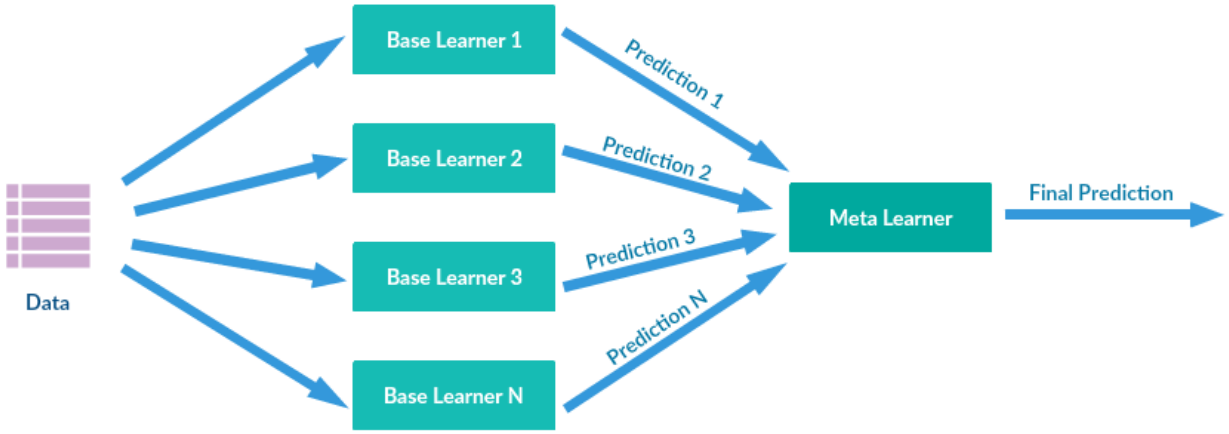
- 장점
  - XGBoost에 비해 훨씬 빠른 속도
    - 기존의 Gradient Boosting Model은 Level-wise 방식 활용
    - 트리의 균형을 맞추지 않고 최대 손실값(max delta loss)를 가지는 리프 노드를 지속적으로 분할
    - 트리의 깊이가 깊어지고 비대칭적 규칙 트리 생성
    - 학습을 반복할 수록 Level-wise보다 예측 오류 손실을 더 줄일 수 있다.
    - 빠르게 학습하기 때문에 대용량 데이터에 적합



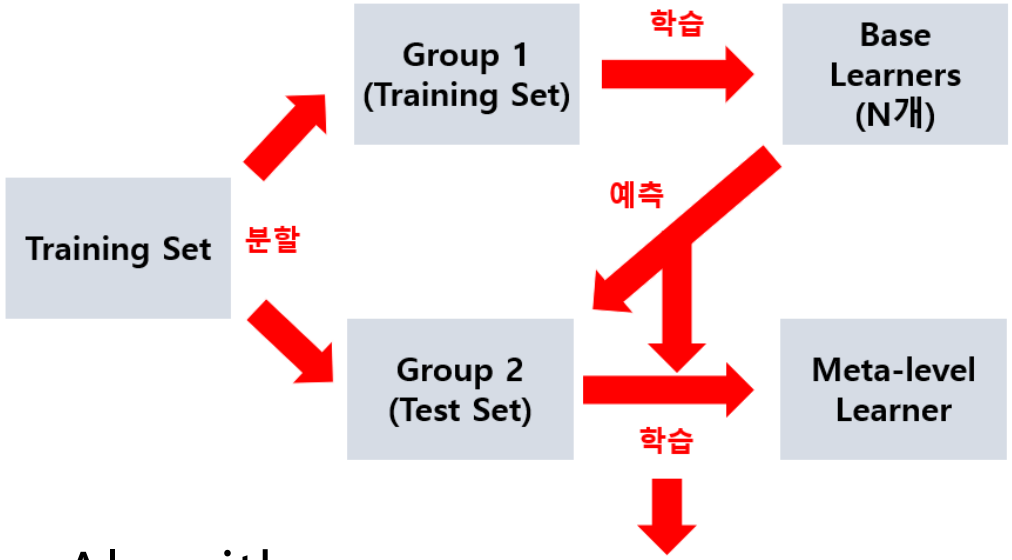
- 단점
  - 적은 데이터셋에 대해서 과적합이 발생하기 쉽다.

Stacking

- Tree 기반의 Bagging이나 Boosting 계열 모델과 달리 서로 다른 모델을 적용 가능
- 각각의 Learner를 통해 나온 결과를 다음 Meta Learner의 Input으로 적용



Stacking

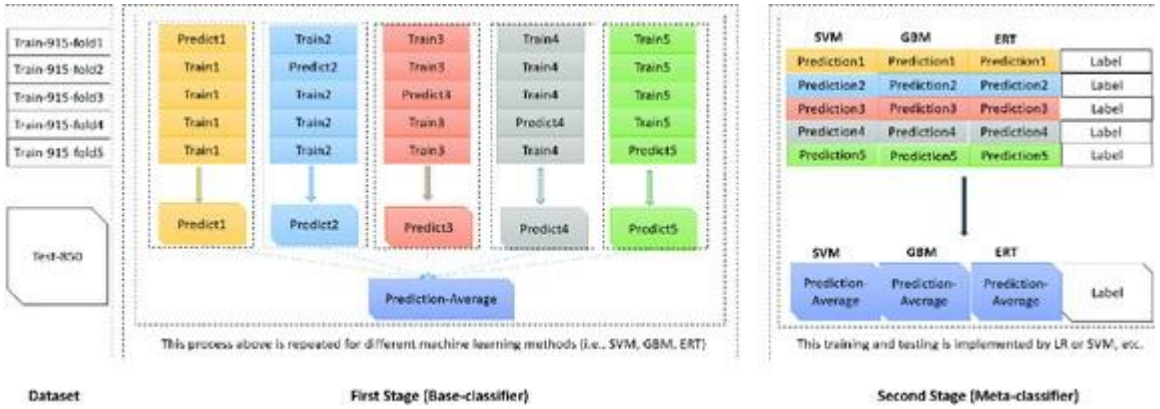


과적합의 문제가 발생해 K-Fold CV를 활용

- Algorithm
  - Training Data를 서로 겹치지 않게 2개의 집단으로 분리
  - 첫번째 그룹을 N개의 Base Learner를 통해 학습
    - 이 때 Bagging이나 Boosting에서는 Tree 기반 모델을 활용했지만 Stacking에서의 Base Learner들은 서로 다른 모델들을 이용할 수 있습니다.(Ex. SVM, 선형 회귀, 신경망 등)
  - 첫번째 그룹을 통해 나온 Output을 두번째 그룹의 Input으로 넣어 두번째 그룹의 실제 Output을 예측하도록 Meta Learner를 학습

Stacking

K-Fold Cross Validation



- K-Fold CV를 통해 K개의 결과를 평균내 사용
- Train, Test 2개로만 분리하는 Hold-out 방식에 비해 과적합 방지하는데 효과적

Boosting, Stacking 참고 자료

Boosting	https://www.youtube.com/watch?v=LsK-xG1cLYA (Adaboost) https://www.youtube.com/watch?v=3CC4N4z3GJc (Gradient Boosting) https://www.youtube.com/watch?v=OtD8wVaFm6E (XGBoost)
Stacking	https://lsjsj92.tistory.com/559 (Stacking Ensemble CV)

BOAZ 분석 15 이재철 , 분석 15 정예림

SVM

Boosting , Stacking

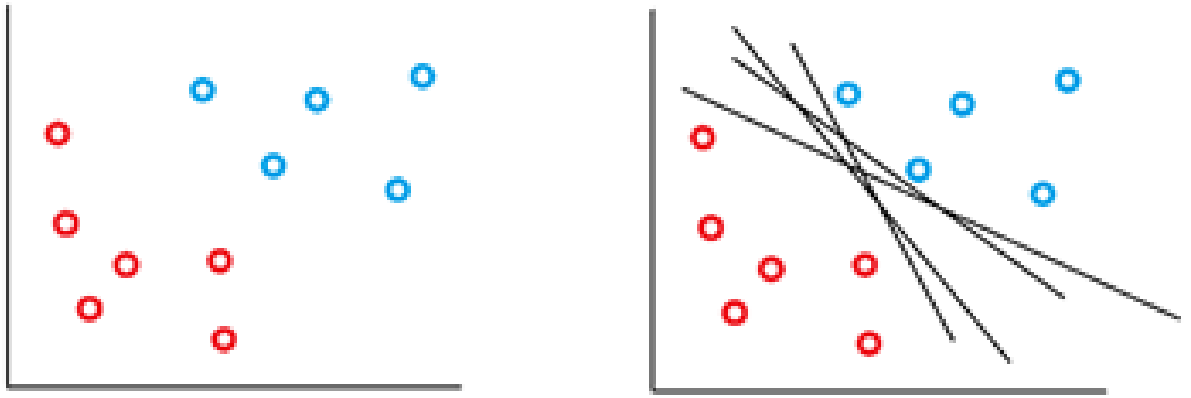
SVM 개요

Support Vector Machine , (*SVM*)

- 지도 학습 모델
- 분류, 회귀 문제
- 탄탄한 이론을 바탕으로 높은 성능을 보여
딥러닝 이전 인기가 많았던 모델

SVM 개요

문제 상황

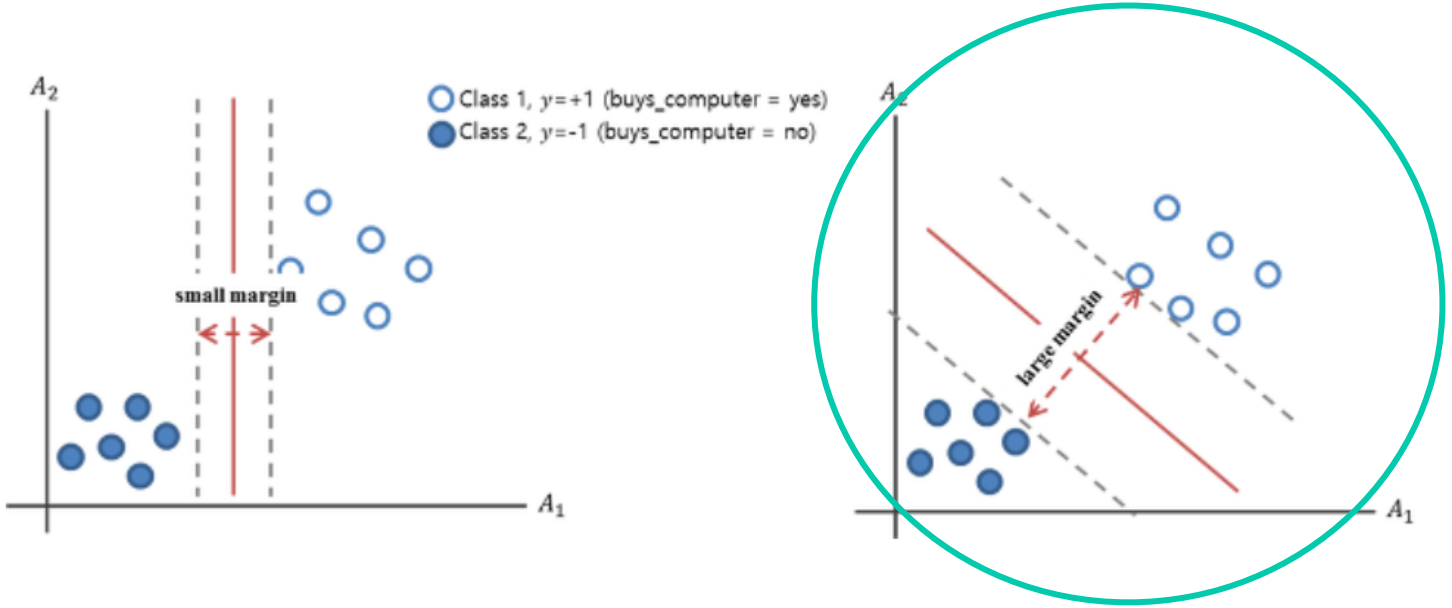


- Two Classification 문제

- 빨간점과 파란점을 나누는 수많은 경계면이 존재함

SVM 개요

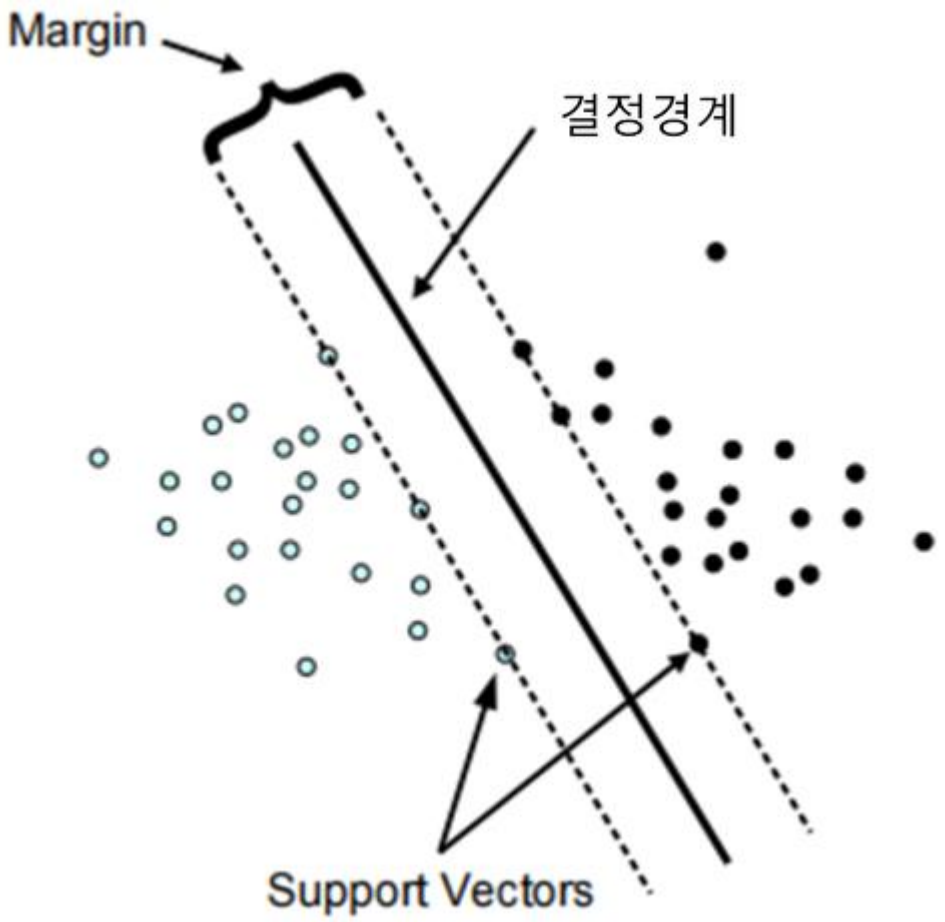
그렇다면, 최적의 경계면은 무엇일까?



"Wide street strategy"

- 마진의 넓이를 최대로 하는 경계면!

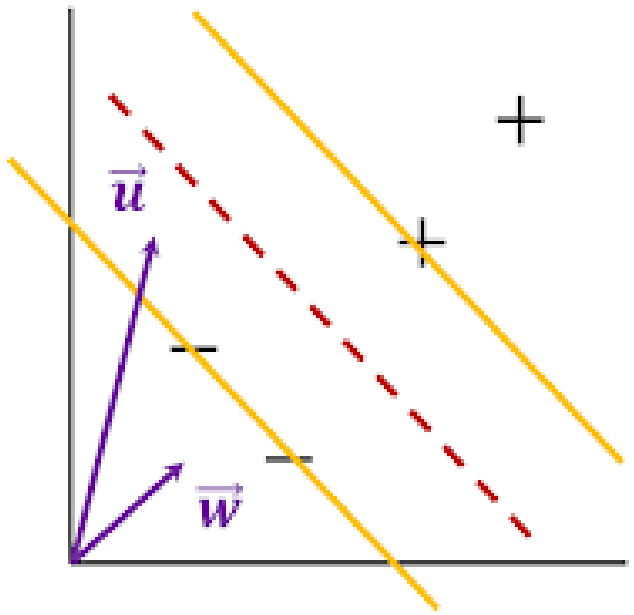
SVM 모델



결정 경계 (Decision Boundary)	데이터의 분류 기준이 되는 경계
서포트 벡터 (Support Vector)	학습 데이터 중에서 결정 경계와 가장 가까이에 있는 데이터들의 집합
마진 (Margin)	결정 경계에서 서포트 벡터까지의 거리

SVM 수식

Decision Rule



$\vec{w}$: 경계면에 직교하는 벡터

$\vec{u}$: 임의의 벡터

[문제]

$\vec{u}$ 가 경계면을 기준으로 어디에 (+,-) 속하는가?

[해결]

$\vec{w}$ 와 $\vec{u}$ 를 내적한 값이 상수 C 보다 큰가?

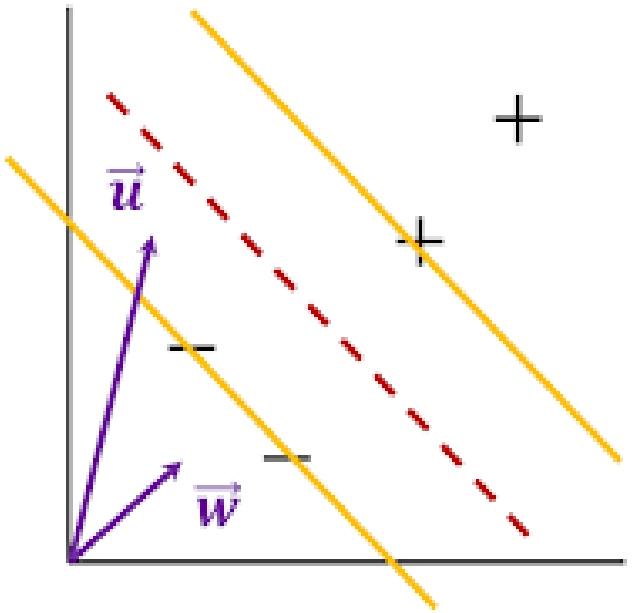
크다면 , '+' 작다면, '-'

$$\vec{w} \cdot \vec{u} \geq C$$

$$\vec{w} \cdot \vec{u} + b \geq 0 \quad \text{then} \quad '+'$$

SVM 수식

Decision Rule



$\vec{w}$: 경계면에 직교하는 벡터
x_+: '+' 샘플
x_-: '-' 샘플

1. 어떤 샘플에 대해

$$\vec{w} \cdot x_+ + b \geq 1$$

$$\vec{w} \cdot x_- + b \leq -1$$

2. 수학적 편리성을 위해 변수를 추가

$$y_i \begin{cases} 1 & for \quad '+' \\ -1 & for \quad '-' \end{cases}$$

3.1에 곱해서 수식을 하나로 정리

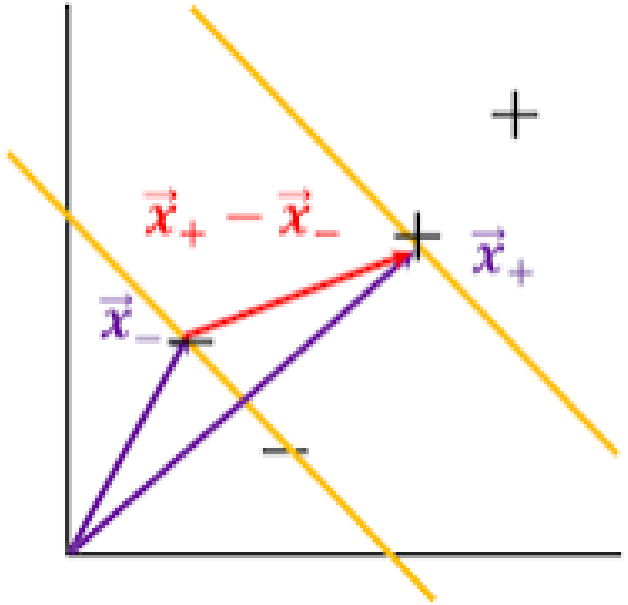
$$y_i(\vec{w} \cdot x_i + b) \geq 1$$

정리하면

$$y_i(\vec{w} \cdot x_i + b) - 1 \geq 0$$

* 등호가 성립할 때 : 샘플이 노란 선에 걸쳐질 때

SVM 수식



$\vec{w}$: 경계면에 직교하는 벡터
$\vec{x}_+$: '+' 샘플
$\vec{x}_-$: '-' 샘플

우리의 목적 : 마진의 폭을 최대로

마진의 폭 : 두 벡터 $\vec{x}_+$, $\vec{x}_-$ 사이의 폭

$$WIDTH = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{||\vec{w}||}$$

이때,

$$\vec{w} \cdot x_+ + b \geq 1$$
$$\vec{w} \cdot x_- + b \leq -1$$

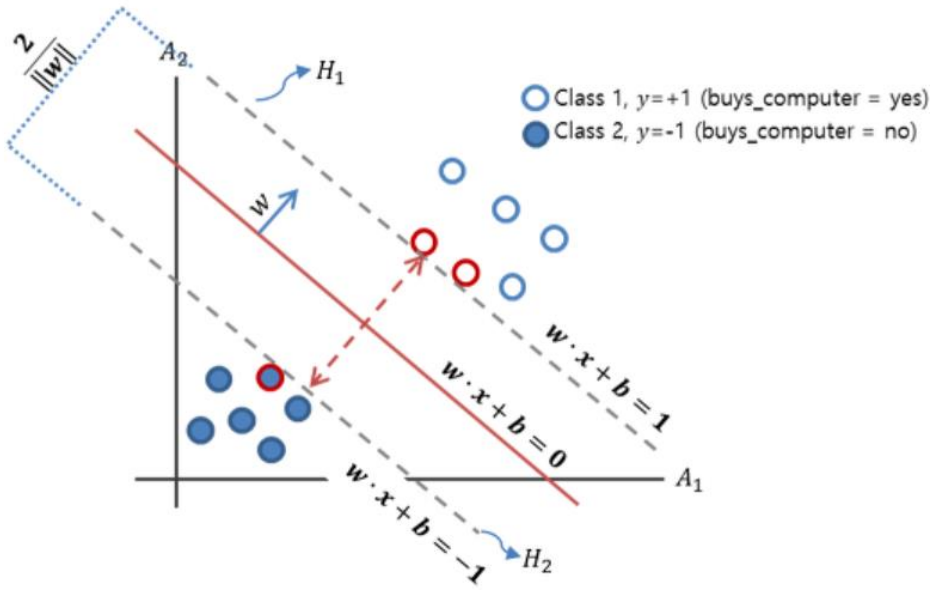
를 이용해서 정리

마진은

$$WIDTH = \frac{2}{||\vec{w}||}$$

SVM 수식

목적식, SVM의 최적화



마진의 넓이 : $WIDTH = \frac{2}{||\vec{w}||}$

제약식 : $y_i(\vec{w} \cdot x_i + b) - 1 \geq 0$

우리의 목적 : 마진의 폭을 최대로

$$max \frac{1}{||\vec{w}||} \leftrightarrow \min||\vec{w}|| \leftrightarrow min\frac{1}{2}||w||^2$$

라그랑주 승주법 적용

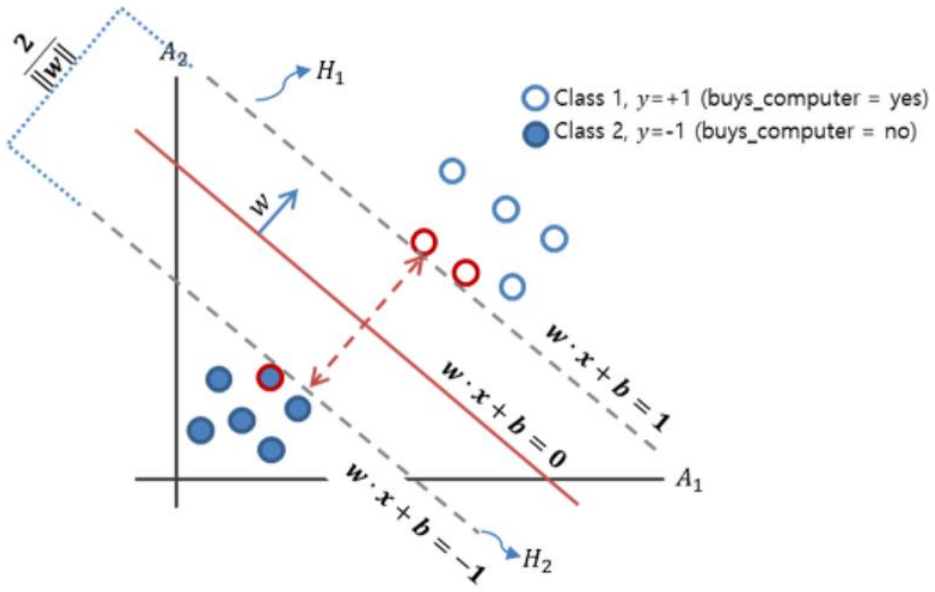
*제약 조건이 있는 최적화 문제를 제약 조건이 없는 문제로 바꾸기 위해

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{N} \alpha_i[y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

$$minimize \quad w.r.t. \quad \vec{w} \; and \; b \qquad maximize \quad w.r.t. \quad \alpha_i \geq 0 \quad \forall i$$

SVM 수식

목적식, SVM의 최적화



마진의 넓이 : $WIDTH = \frac{2}{||\vec{w}||}$

제약식 : $y_i(\vec{w} \cdot x_i + b) - 1 \geq 0$

라그랑주 승주법 적용

*제약 조건이 있는 최적화 문제를 제약 조건이 없는 문제로 바꾸기 위해

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{N} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

$$minimize \quad w.r.t. \quad \vec{w} \; and \; b \qquad maximize \quad w.r.t. \quad \alpha_i \geq 0 \quad \forall i$$

$\vec{w}$ 와 b 에 대해 각각 편미분

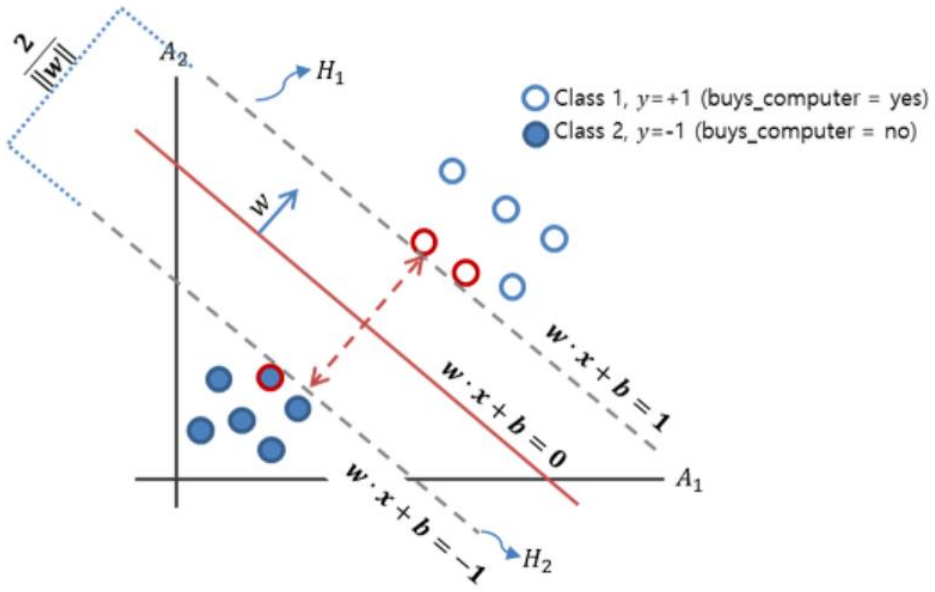
$$\nabla_{\vec{w}} \mathcal{L} = \vec{w} - \sum_i \alpha_i y_i \vec{x}_i = 0$$

$$\nabla_b \mathcal{L} = -\sum_i \alpha_i y_i = 0$$

$$\vec{w} = \sum_i \alpha_i \, y_i \vec{x}_i$$

SVM 수식

목적식, SVM의 최적화



마진의 넓이 : $WIDTH = \frac{2}{||\vec{w}||}$

제약식 : $y_i(\vec{w} \cdot x_i + b) - 1 \geq 0$

α 에 대한 maximization 문제로 정리

$$\sum_{i=1}^{N} \alpha_i + \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{N} \alpha_i y_i \vec{w}^T \vec{x}_i \iff \sum_{i=1}^{N} \alpha_i + \frac{1}{2}\vec{w}^T \vec{w} - \vec{w}^T \vec{w}$$

$$\iff \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\vec{w}^T \vec{w}$$

$$\iff \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$$

$$\iff \mathcal{L}(\alpha).$$

1. α를 구하면 $\vec{w}$ 를 구할 수 있음

2. α값이 0이 아니라는 것은 해당 $\vec{x}$가 경계선을 정하는 샘플

즉, Support Vector 라는 뜻

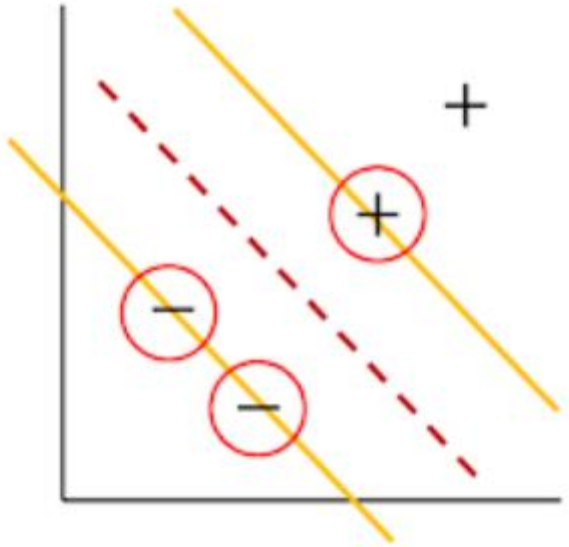
3. $L(\alpha)$ 식 의 성질에 의해

" SVM 으로 구한 해는 항상 (현재 SVM)이 줄 수 있는 최적의 해"

임을 이론적으로 보장. -> local minima에 빠지지 않음

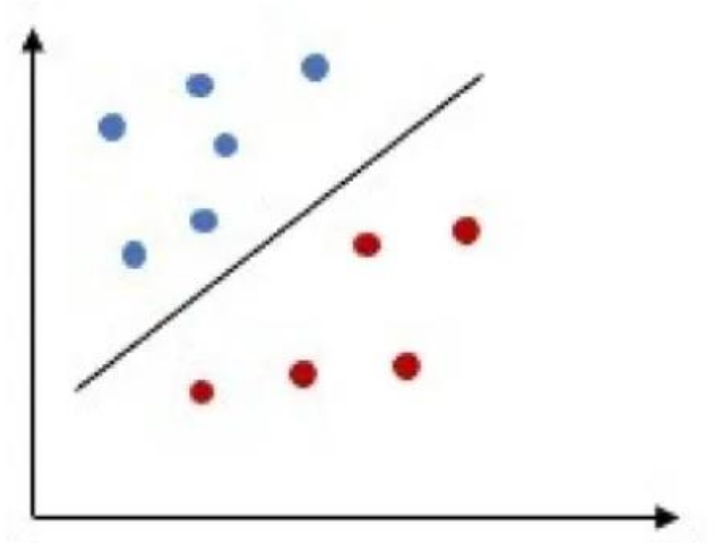
SVM 수식

결론



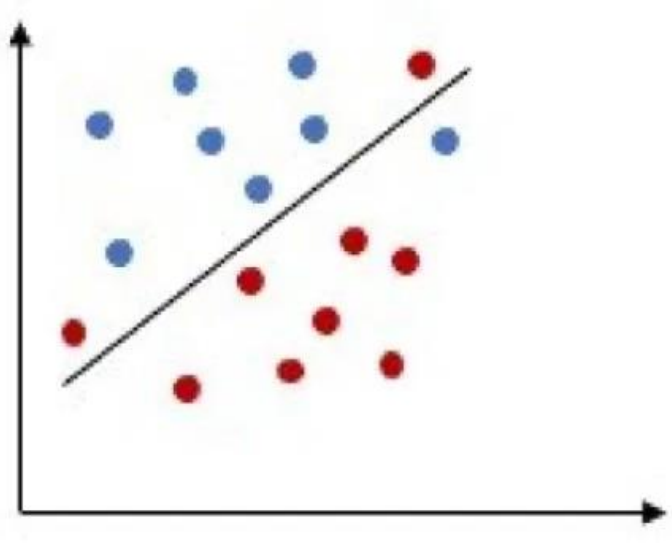
support vector들로 정해진 decision boundary가 가장 최적의 boundary이며,
새로운 샘플이 들어왔을 때 일반화를 가장 잘 할 수 있는
decision rule을 찾을 수 있게 된 것

SVM 종류



하드 마진 SVM (Hard Margin SVM)

- 오분류를 허용하지 않는 SVM
- 노이즈로 인해 최적의 결정 경계를 잘 못 구하거나, 못 찾을 수 있음

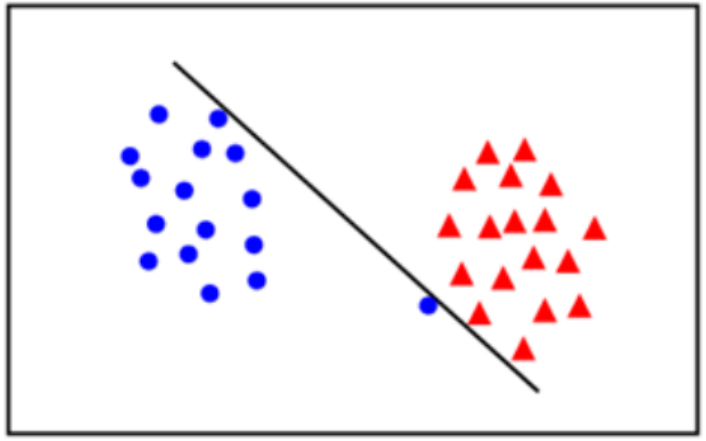


소프트 마진 SVM (Soft Margin SVM)

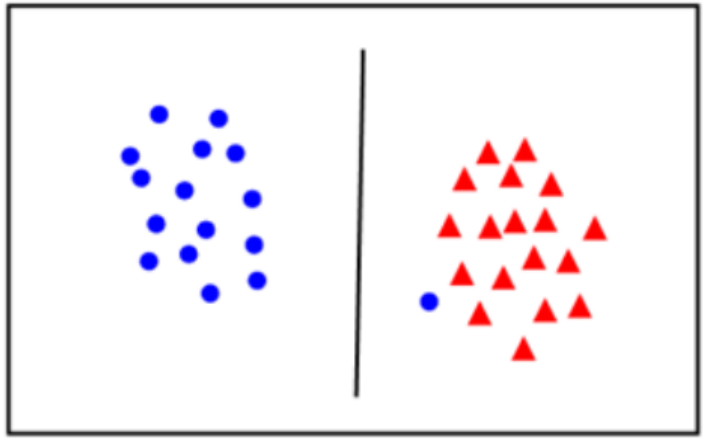
- 오분류를 허용하는 SVM
- 하드 마진 SVM은 적용하기가 어려우므로 어느 정도의 오류를 허용하는 소프트 마진 SVM을 주로 이용

Soft Margin SVM – 슬랙 변수

둘 중 최적의 경우는?



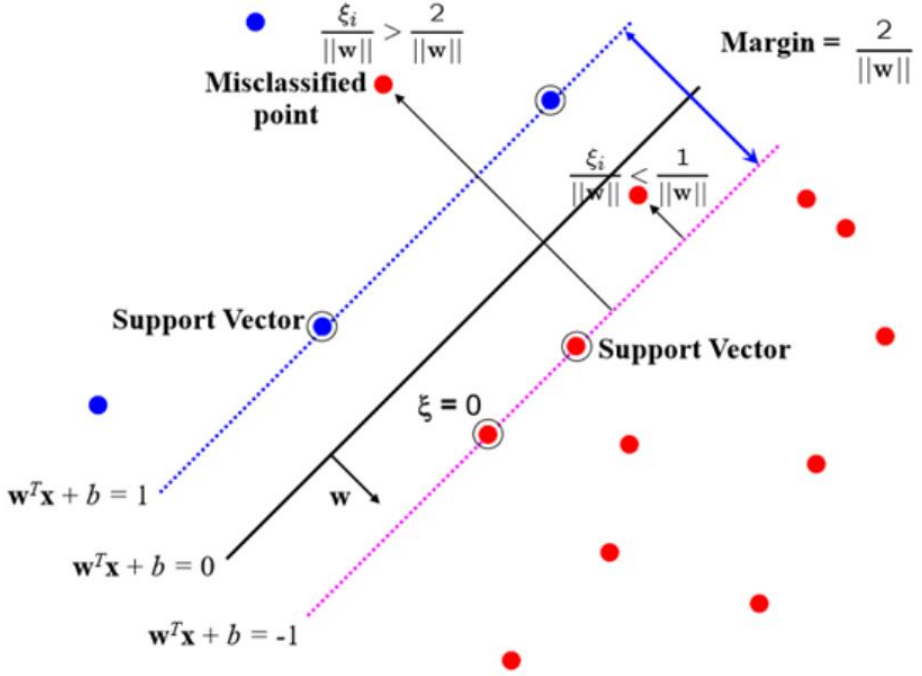
마진 ↓, 오분류 X



마진 ↑, 오분류 O

일반적으로 마진과 학습 오류의 개수는 Trade-Off 관계임

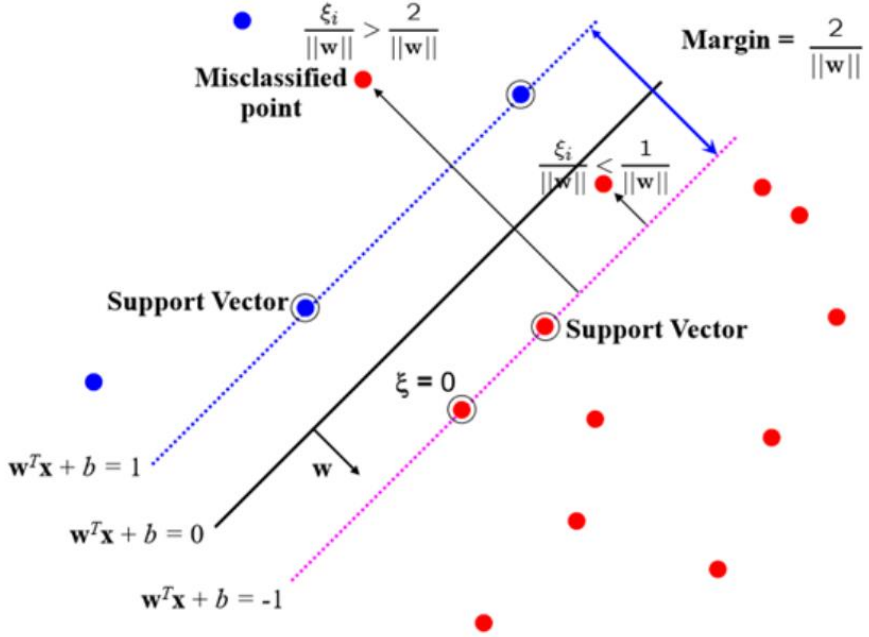
Soft Margin SVM – 슬랙 변수



슬랙 변수 (Slack Variables)

데이터의 완벽한 분리가 불가능 할 때, 선형적 분류를 위해 허용된 오차를 위한 변수

C-SVM



C – SVM

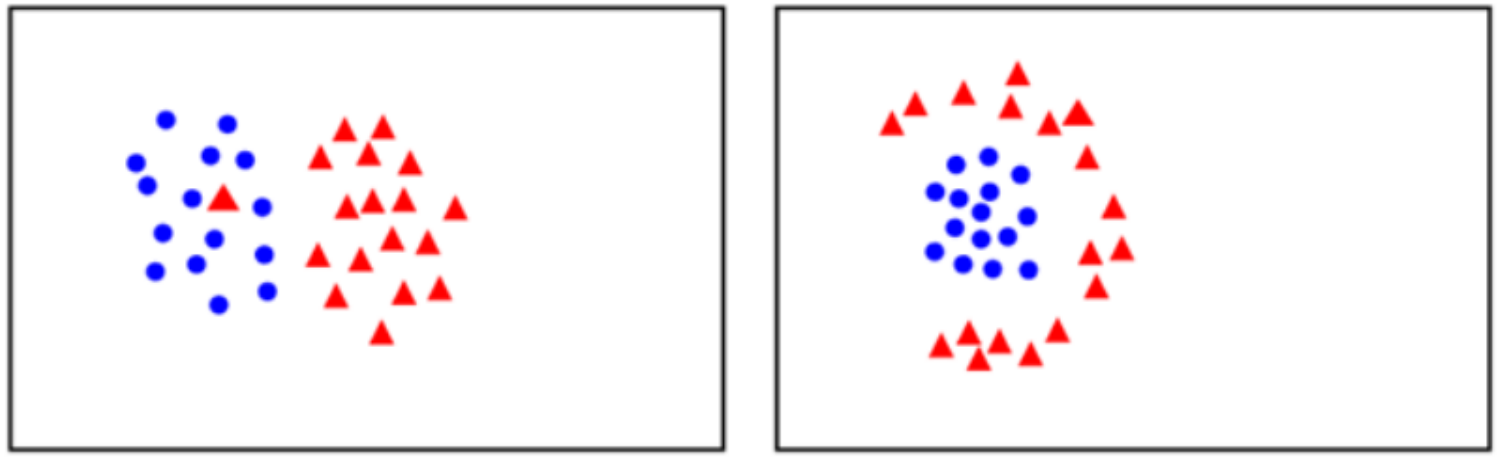
- Soft Margin SVM
- 엡실론만큼 패널티를 부과
- C의 크기로 마진의 폭을 조절

목적식 변화

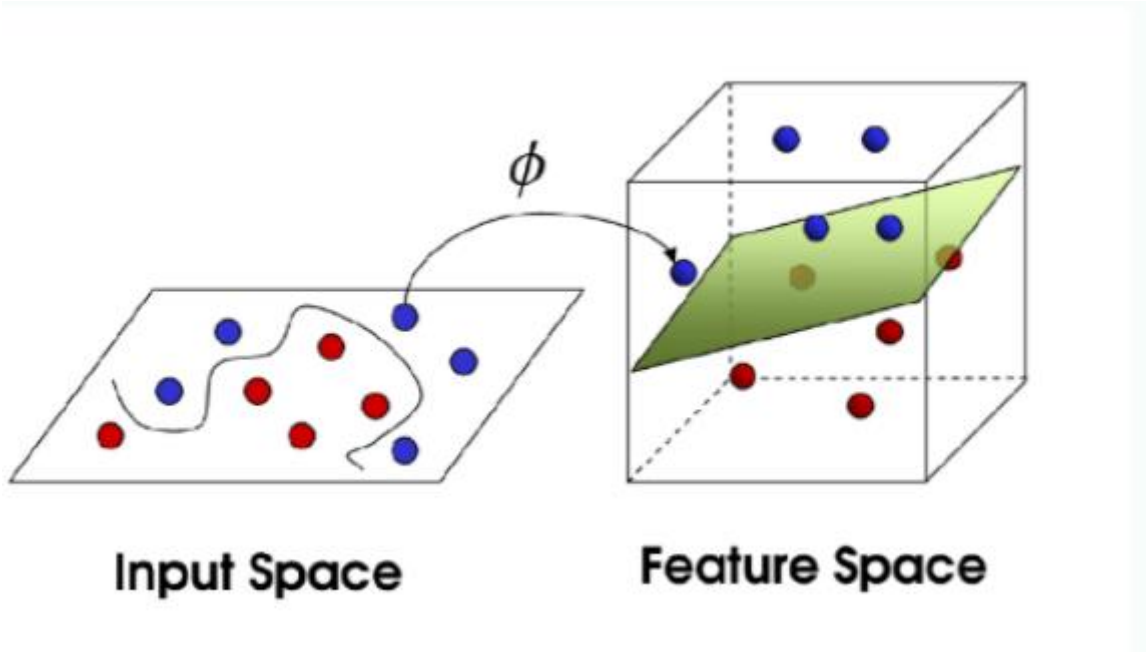
$$min\frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}\varepsilon_i$$

데이터 분류가 비선형일 경우

not
linearly
separable



Kernel Trick (비선형일 경우)



" Kernel trick "

비선형인 경우 저차원 공간을 고차원 공간으로 매핑하는 방법
"커널 함수"를 이용하여 고차원 공간으로 매핑하는 경우에
증가하는 연산량의 문제를 해결하는 기법

커널 함수

주로 사용하는 커널 함수

Type of Kernel	Inner product kernel $K(\vec{x}, \vec{x}_i),\ i = 1, 2, \ldots, N$	Comments
Polynomial Kernel	$K(\vec{x}, \vec{x}_i) = \left(\vec{x}^T \vec{x}_i + \theta\right)^d$	Power p and threshold θ is specified a priori by the user
Gaussian Kernel	$K(\vec{x}, \vec{x}_i) = e^{-\frac{1}{2\sigma^2}\|\vec{x}-\vec{x}_i\|^2}$	Width σ^2 is specified a priori by the user
Sigmoid Kernel	$K(\vec{x}, \vec{x}_i) = tanh(\eta\, \vec{x}\, \vec{x}_i + \theta)$	Mercer's Theorem is satisfied only for some values of η and θ
Kernels for Sets	$K(\chi, \chi') = \sum_{i=1}^{N_\chi} \sum_{j=1}^{N_{\chi'}} k(x_i, x'_j)$	Where $k(x_i, x'_j)$ is a kernel on elements in the sets χ, χ'
Spectrum Kernel for strings	count number of substrings in common	It is a kernel, since it is a dot product between vectors of indicators of all the substrings.

이중, 가우시안 커널 을 가장 많이 사용함
가우시안 커널 (= RBF , Radial Basis Fuction)

SVM 사용하기

Sklearn 라이브러리는 다양한 SVM 모델을 지원함

상황	모델	
분류 문제	SVC	Classification의 C
	LinearSVC	SVC의 linear kernel 특화 버전
회귀 문제	SVR	Regression의 R
	LinearSVR	SVR의 linear kernel 특화 버전

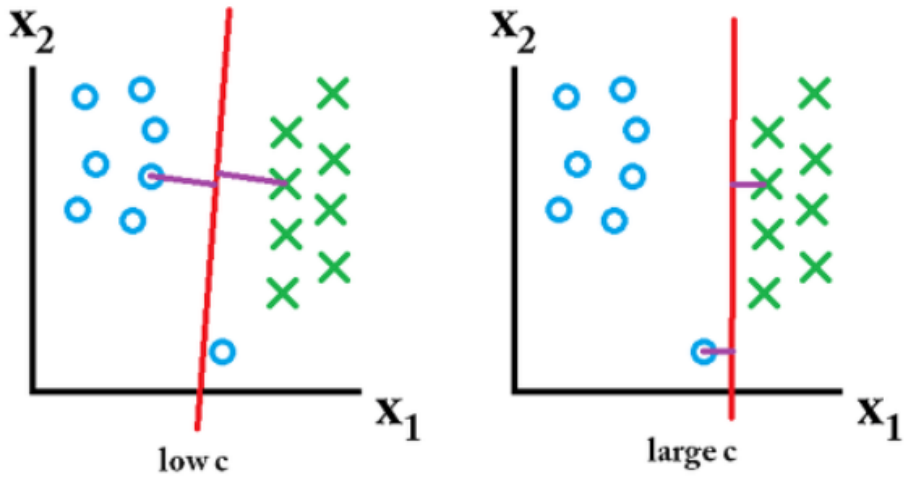
SVM 사용하기

SVM의 매개변수

C , gamma

SVM의 매개변수 - C

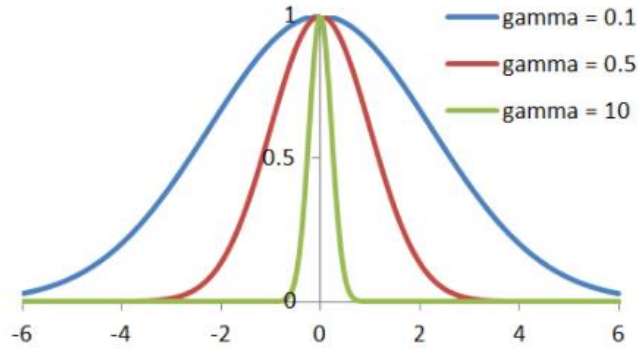
C : 소프트 마진 SVM의 슬랙변수



C가 너무 낮으면 과소 적합
C가 너무 높으면 과대 적합

SVM의 매개변수 - gamma

gamma : 가우시안 함수의 표준편차와 관련 있는 매개변수



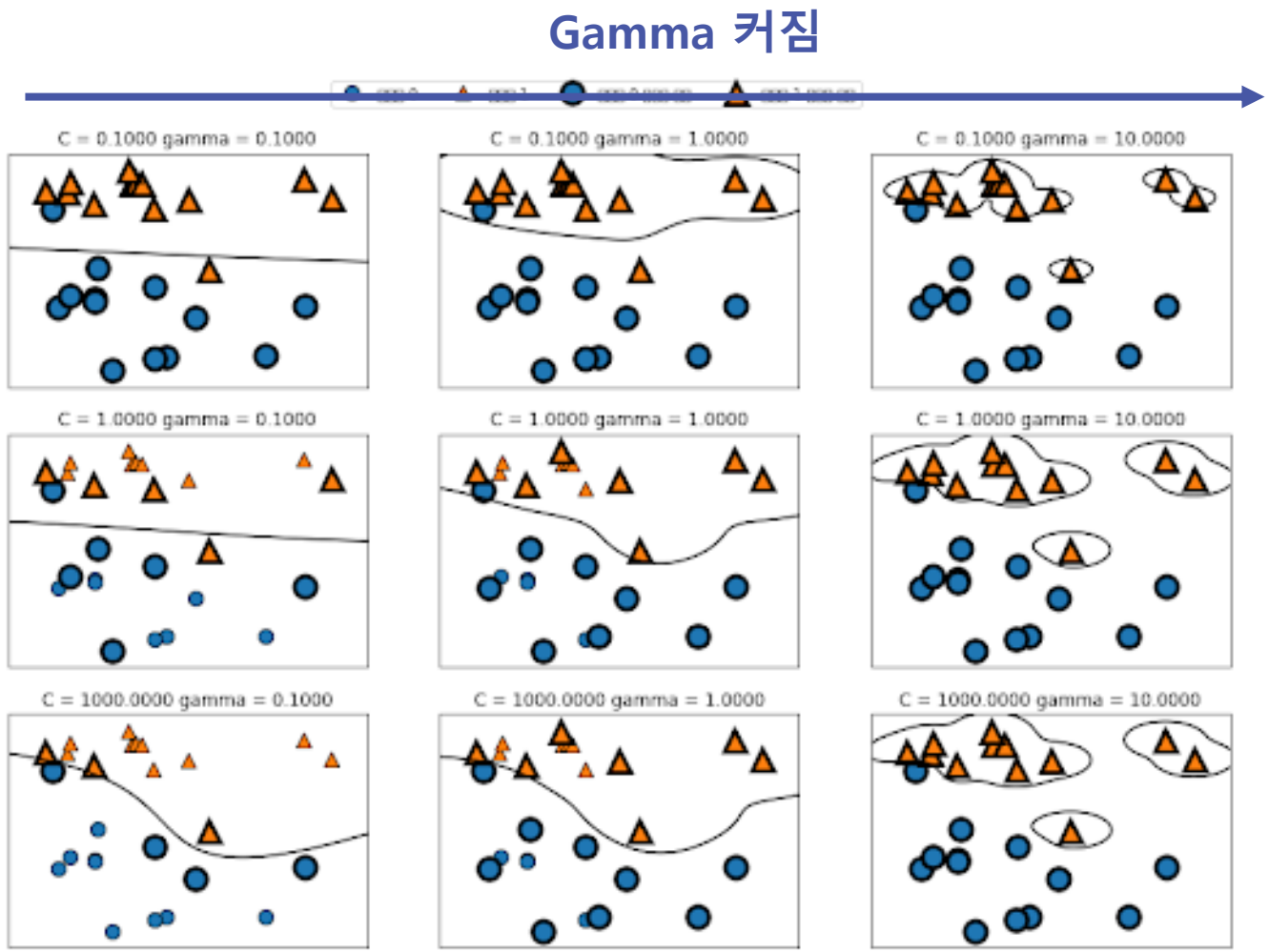
Gamma 는 RBF 커널에서
하나의 데이터 샘플이 영향력을 행사하는 거리를 나타냄

Gamma가 크면 데이터 포인터들이 영향력을 행사하는 거리가 짧아지고
Gamma가 낮으면 영향력을 행사하는 거리가 커짐

Gamma가 너무 낮으면 과소 적합
Gamma가 너무 크면 과대 적합

SVM 사용하기

SVM의 매개변수 조절

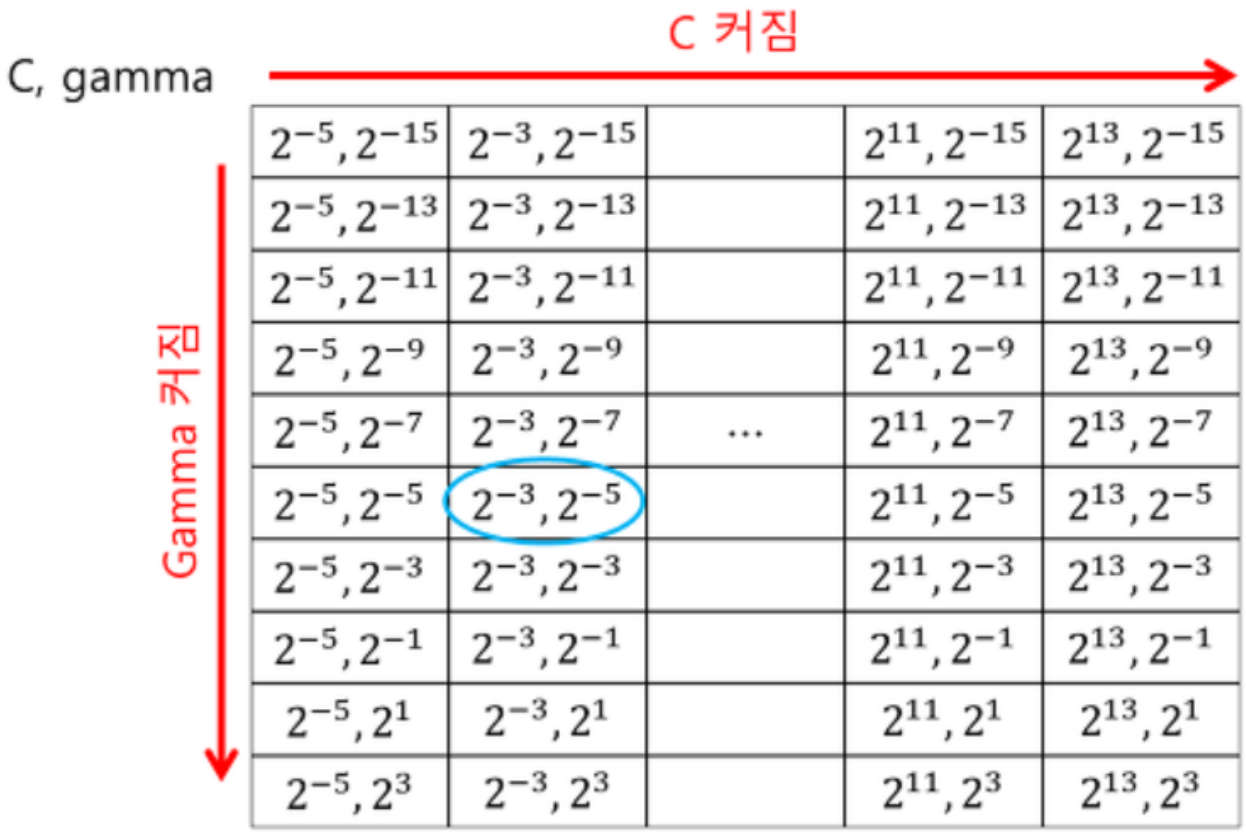


Gamma가 동일한 경우
C가 커질 수록 오분류 인정 X

C 가 동일한 경우
gamma가 커질 수록 곡률 높음

SVM 사용하기

SVM의 매개변수 조절



Grid search

매개변수들의 여러 조합들을 테스트 해서 가장 좋은 성능을 내는 매개변수를 찾아내는 것

SVM 참고 자료

SVM 수식 이해 참고자료	Youtube - **MIT OpenCourseWare** ★★★★★ https://www.youtube.com/watch?v=_PwhiWxHK8o&t=6s Blog - **JaeJun Yoon's Playground** ★★★★☆ http://jaejunyoo.blogspot.com/2018/01/support-vector-machine-1.html
SVM 종류	Blog – 라온피플 ★★☆☆☆ https://blog.naver.com/laonple/220847975603 Blog – 귀퉁이 서재 ★★★☆☆ https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-2%EC%84%9C%ED%8F%AC%ED%8A%B8-%EB%B2%A1%ED%84%B0-%EB%A8%B8%EC%8B%A0-SVM
SVM 사용하기	Blog – bskyVision ★★★☆☆ https://blueskyvision.tistory.com/163 Blog – 코딩생활 커딩왕 ★★★★★ https://chancoding.tistory.com/67