실습을 통해 기초부터 배우는 머신러닝

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

Week 4

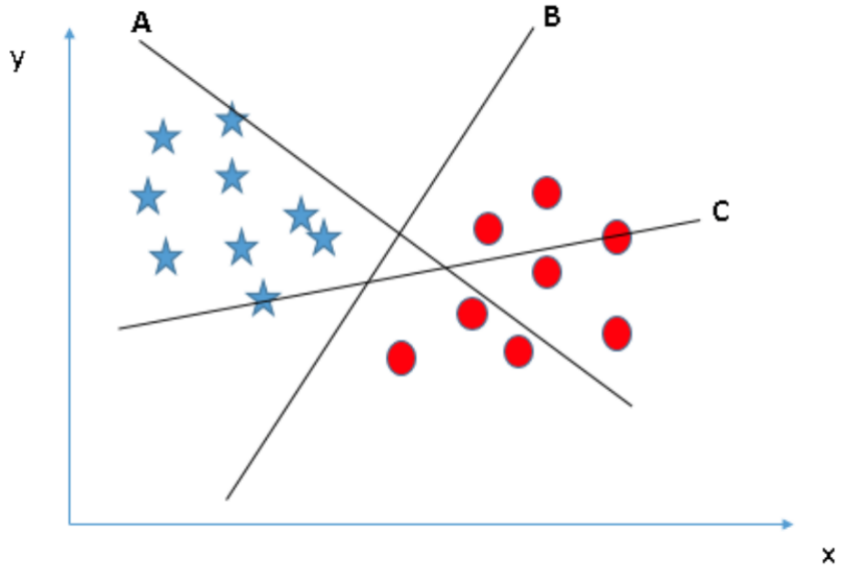
Agenda

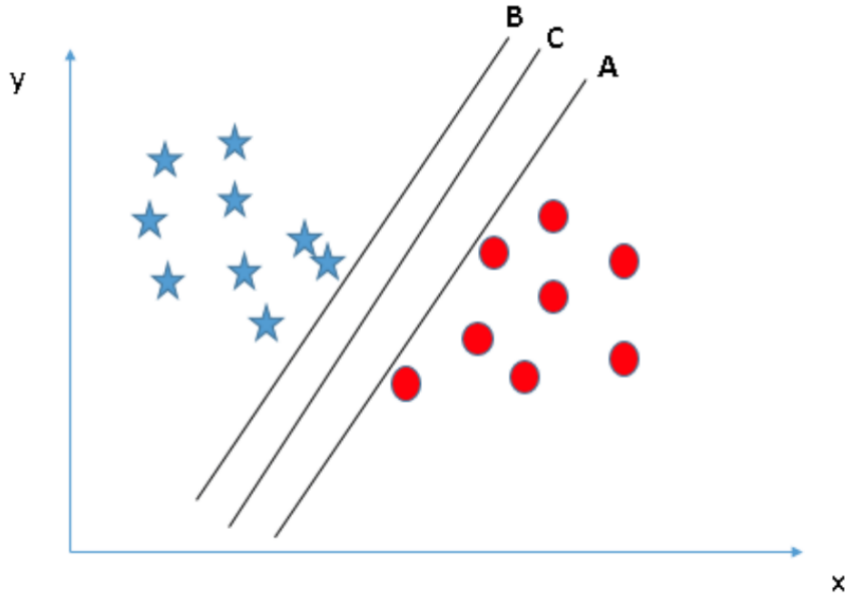
1. Review
2. SVM
3. Decision Tree
4. PCA
5. K-means
6. Summary

Support Vector Machine(SVM)

- 지도 학습 모델 (주로 분류 문제에 사용)
- 뛰어난 성능으로 딥러닝 이전에 많은 주목을 받은 알고리즘

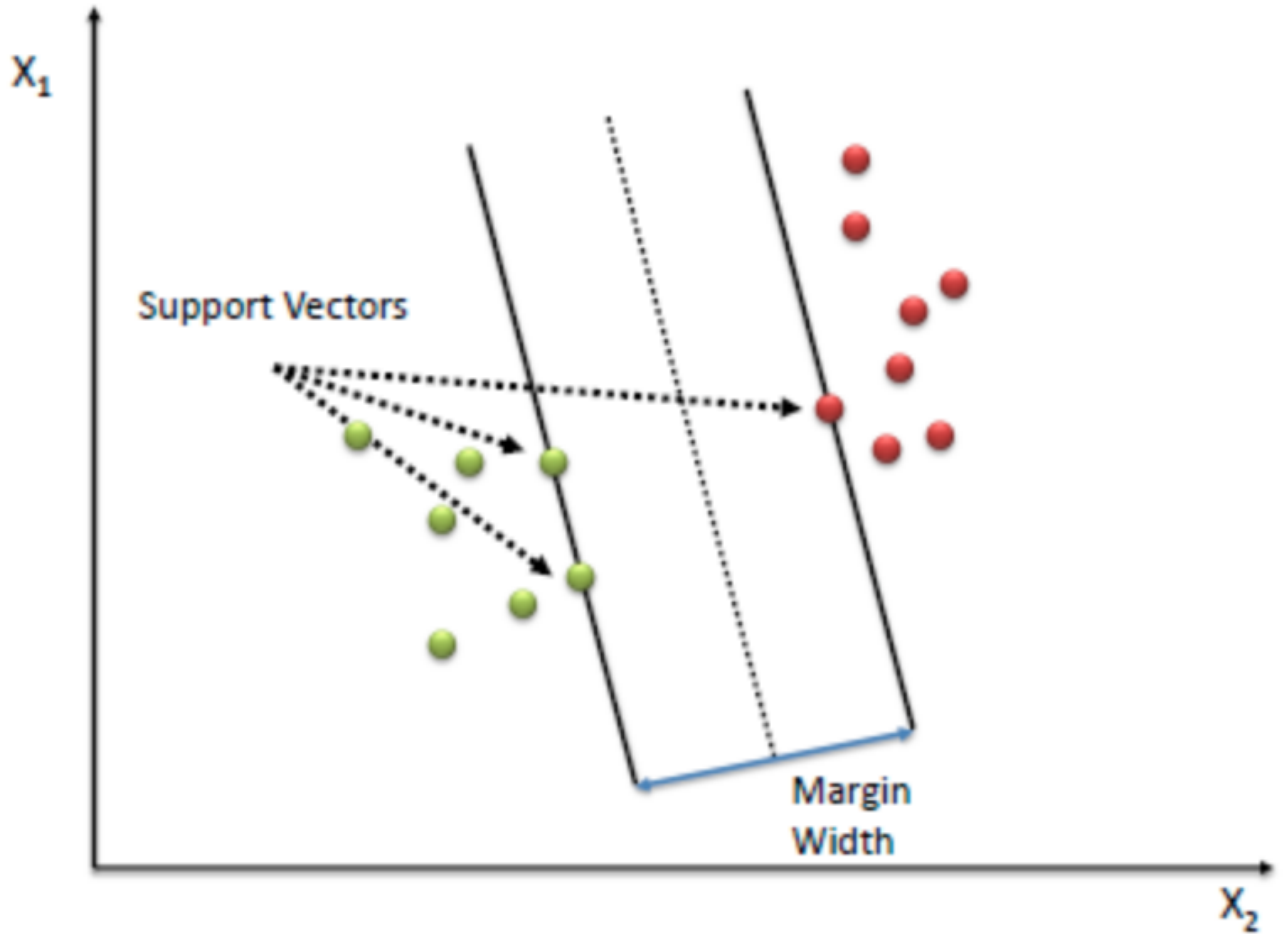
Q) A, B, C 중 다음 데이터를 가장 잘 분리하는 것은 ?



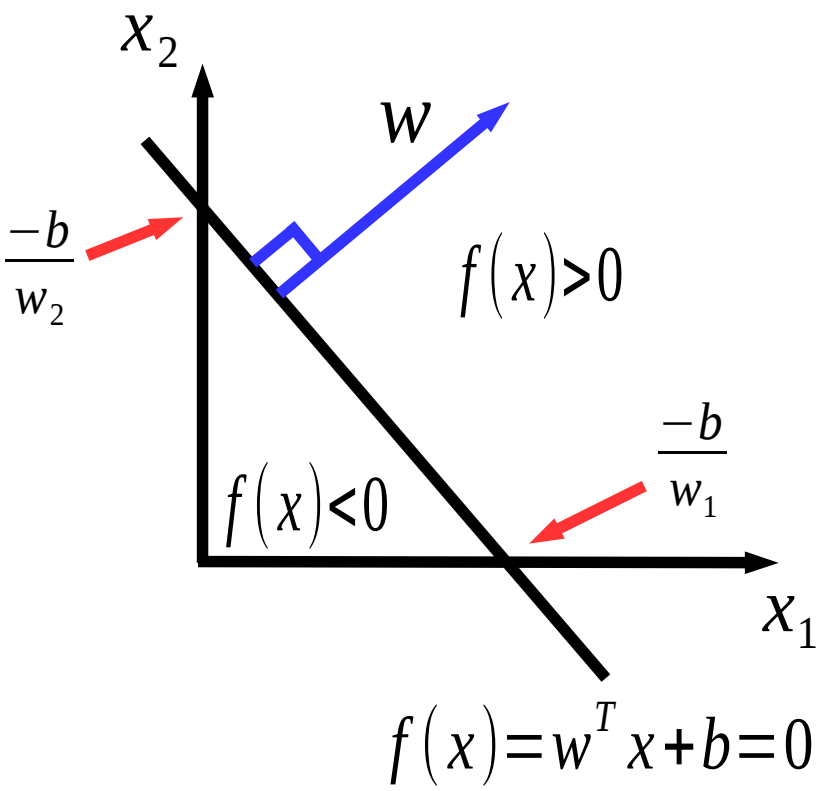


Support Vector Machine(SVM)

- **Margin 을 최대로 하는 분류 경계면 (선) 을 찾는 방법**



Support Vector Machine(SVM)



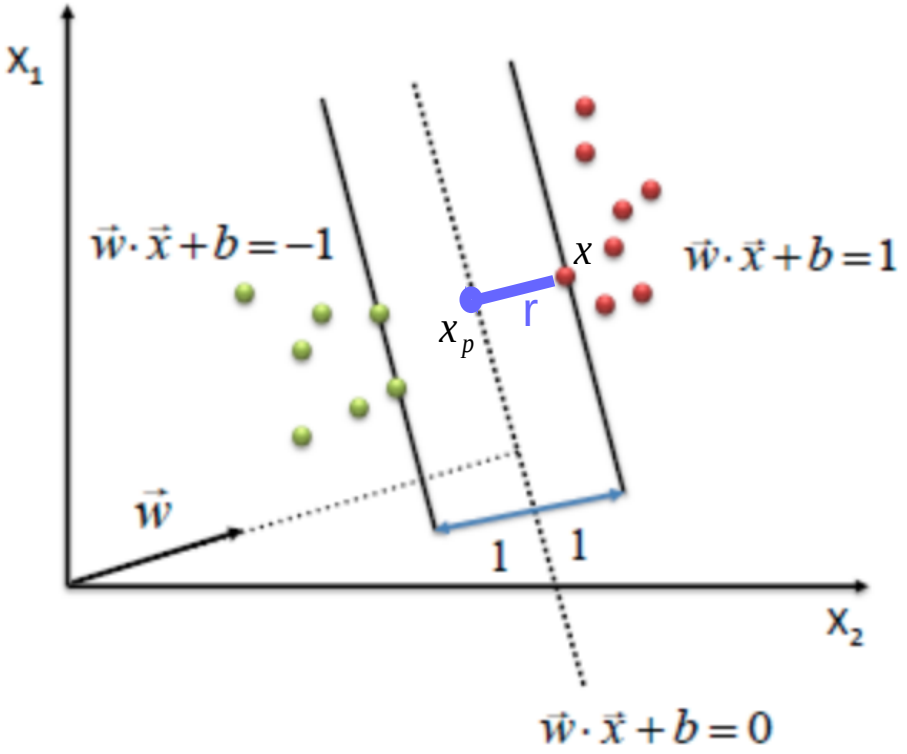
$$f(x)=w^T x+b=w_1 x_1+w_2 x_2+b=0$$

ex) $f(x)=x_1+x_2-1=0$

직선 f(x) 에 수직 방향의
벡터 u 를 찾아보자 .

$$\left(\frac{-b}{w_1},\frac{-b}{w_2}\right)\cdot(u_1,u_2)=0 \qquad (u_1,u_2)=(w_1,w_2)$$

Support Vector Machine(SVM)



$$f(x_p)=w^T x_p+b=0$$

$$x=x_p+r\frac{w}{\|w\|}$$

$$f(x)=w^T\left(x_p+r\frac{w}{\|w\|}\right)+b=w^T x_p+b+\frac{w^T w}{\|w\|}r=r\|w\|$$

$$r=\frac{f(x)}{\|w\|}=\frac{a}{\|w\|}(a>0)$$

Margin (2r) 을 최대화 하는 것이 목표

Support Vector Machine(SVM)

$$r = \frac{f(x)}{\|w\|} = \frac{a}{\|w\|} (a>0)$$

$$max_{w,b} 2r = \frac{2a}{\|w\|}$$

$$(wx_i + b) y_i \geq a \text{ , i = 1...N}$$

y_i : +1 또는 -1 (class)

a = 1

제약 조건 만족하는 w 최적화 문제

$$\min \frac{1}{2}\|w\|^2$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i$$

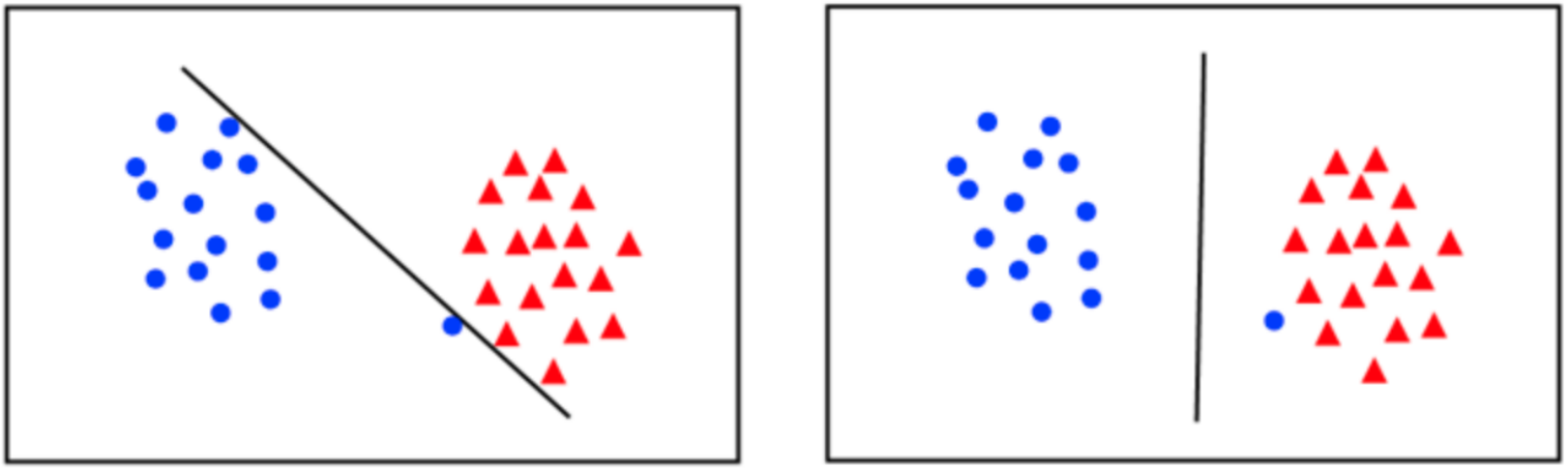
Objective Function

Quadratic optimization problem

Margin (2r) 을 최대화 하는 것이 목표

SVM - Soft Margin

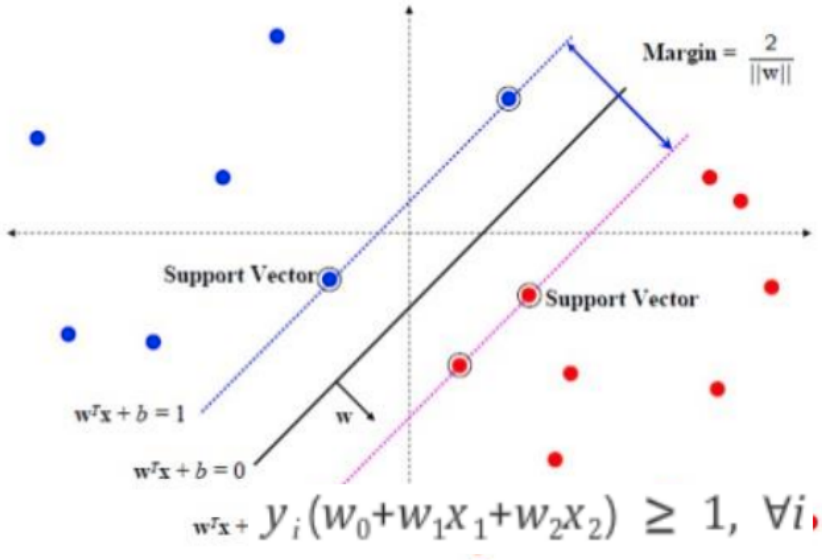
- **Which one is better?**



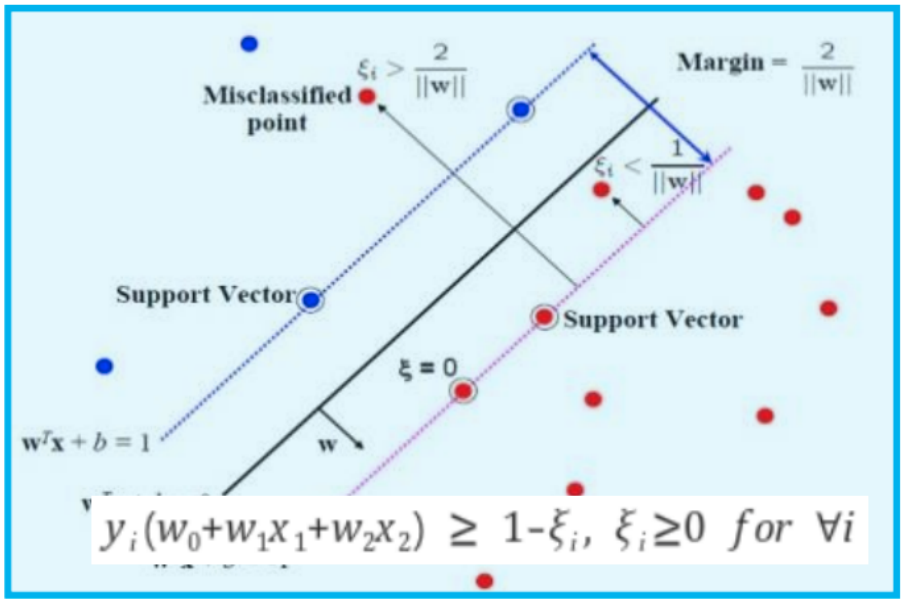
Margin 과 학습 오류의 개수는 Trade-Off 관계
(일반화 VS 정확도)

SVM - Soft Margin

Hard Margin SVM



Soft Margin SVM



ξ_i 만큼의 오류를 허용해 준다.

(오차 허용 변수)

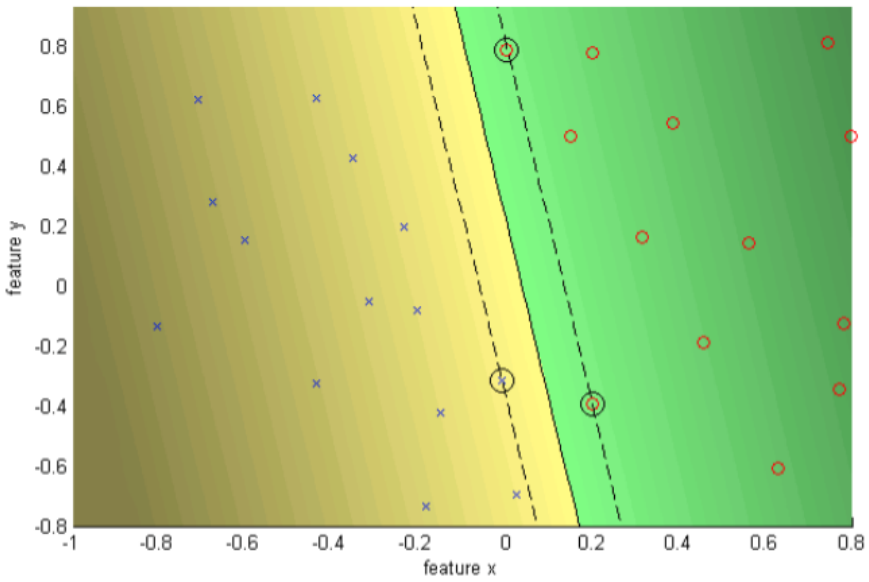
$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

Objective Function

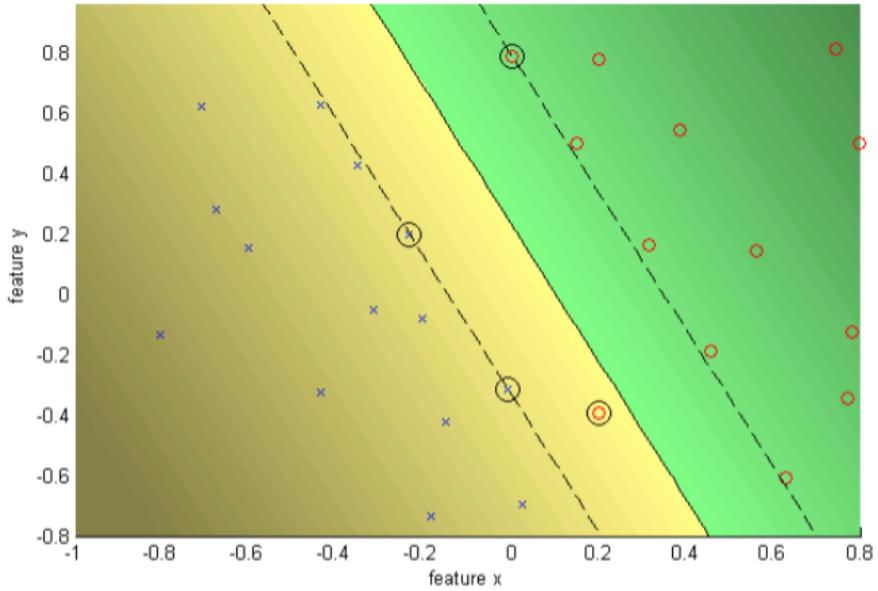
C 는 Hyperparameter 로 Margin 폭을 조절하여 얼마나 오류를 허용할 지 결정

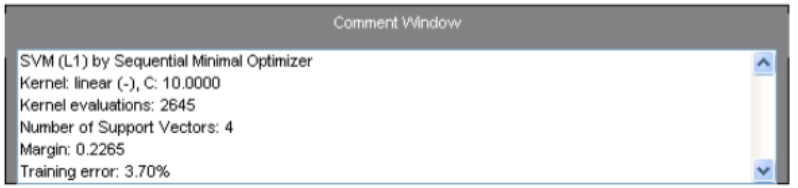
SVM - Soft Margin

C = infinity (hard margin)



C = 10 (soft margin)

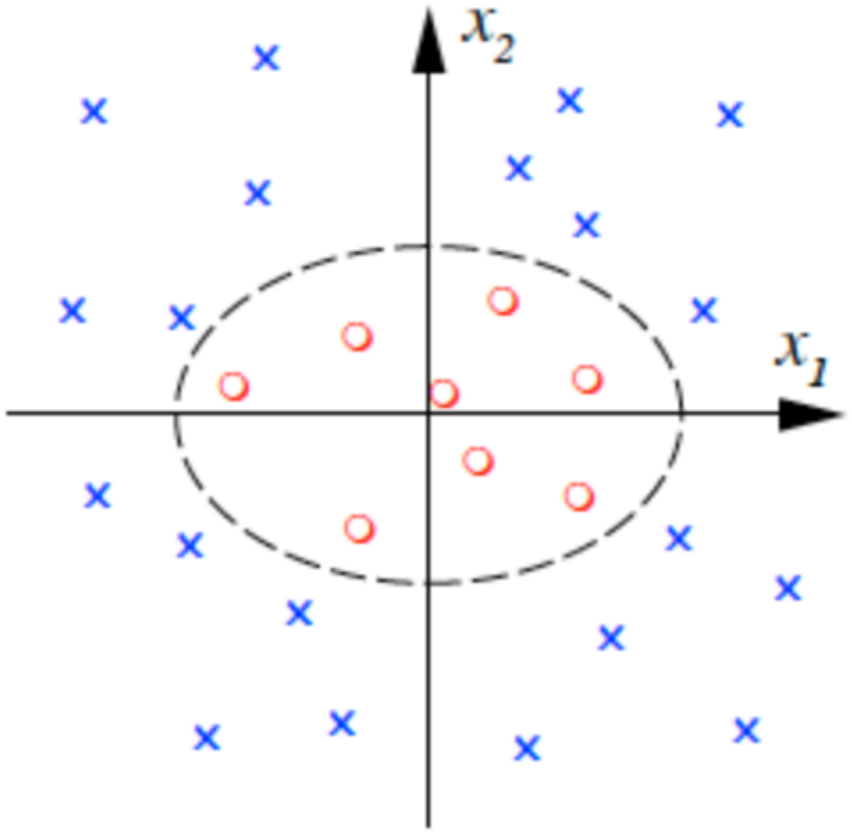




C 가 클수록 Margin 의 폭은 줄어든다 .

SVM

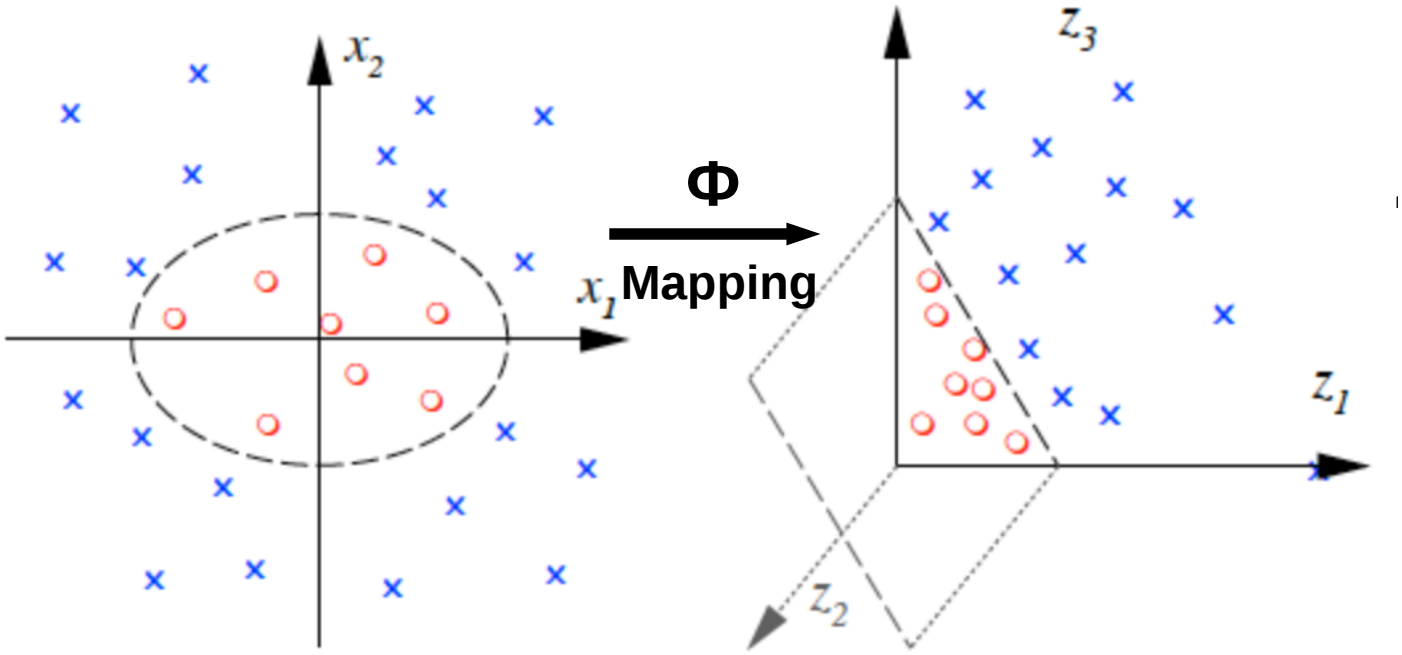
- 만약 데이터가 다음과 같다면 ?



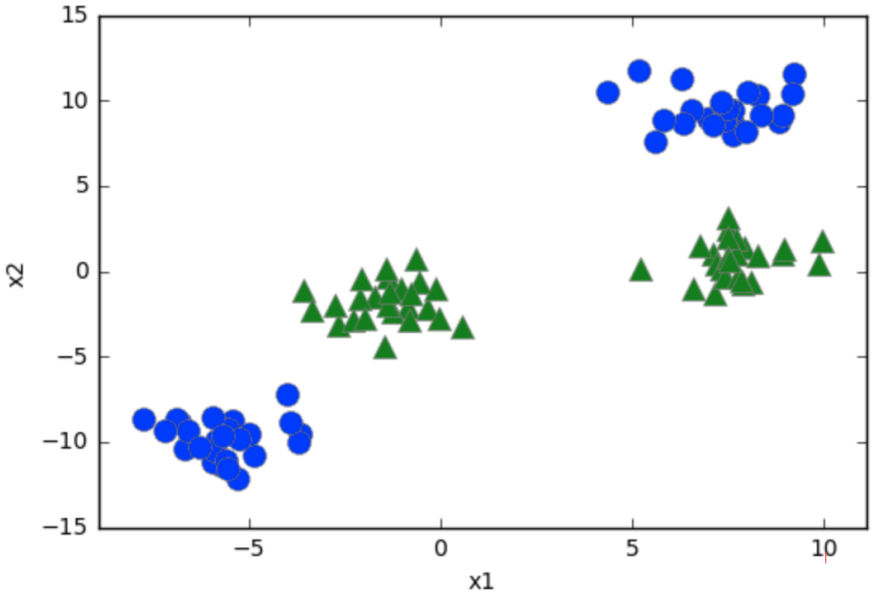
SVM – Kernel trick (dim:low → high)

$$\Phi : R^2 \to R^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{(2)}x_1x_2, x_2^2)$$

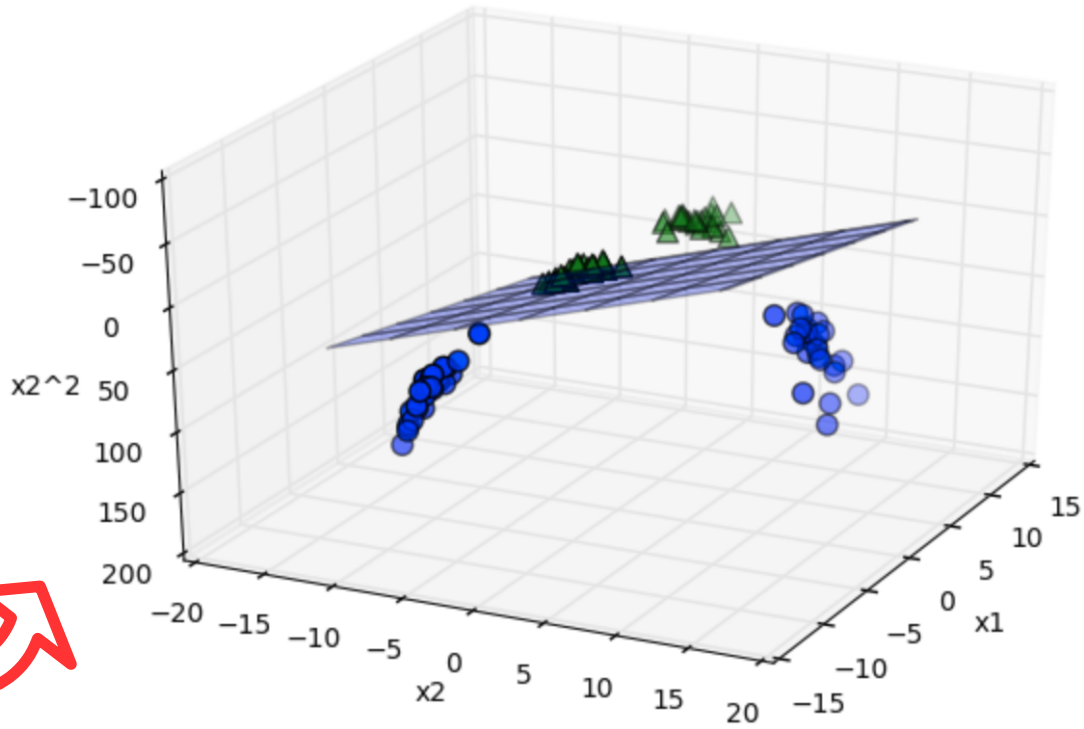


SVM – Kernel trick (dim:low → high)

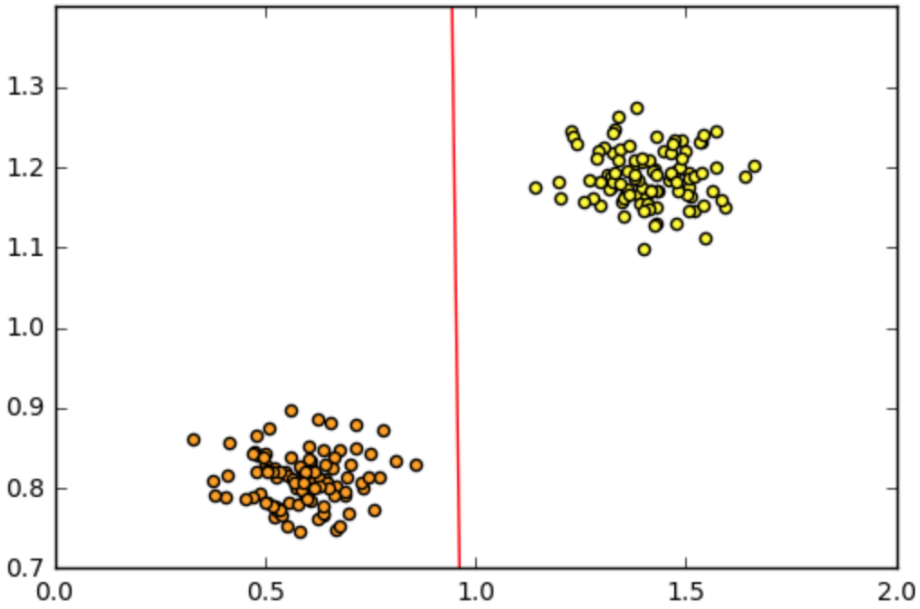


NOT SEPARABLE PROBLEM

SEPARABLE PROBLEM

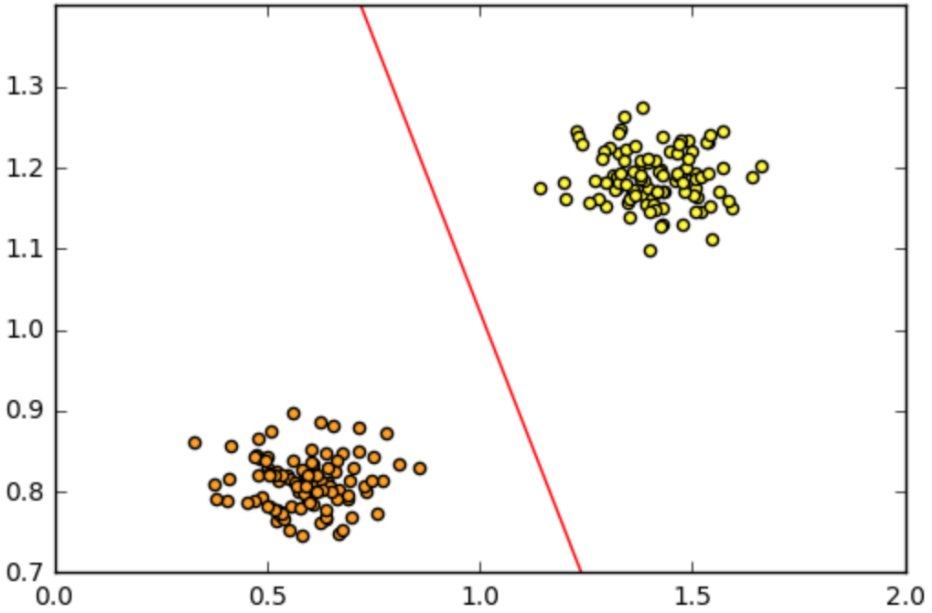


Logistic Regression VS SVM



Logistic Regression

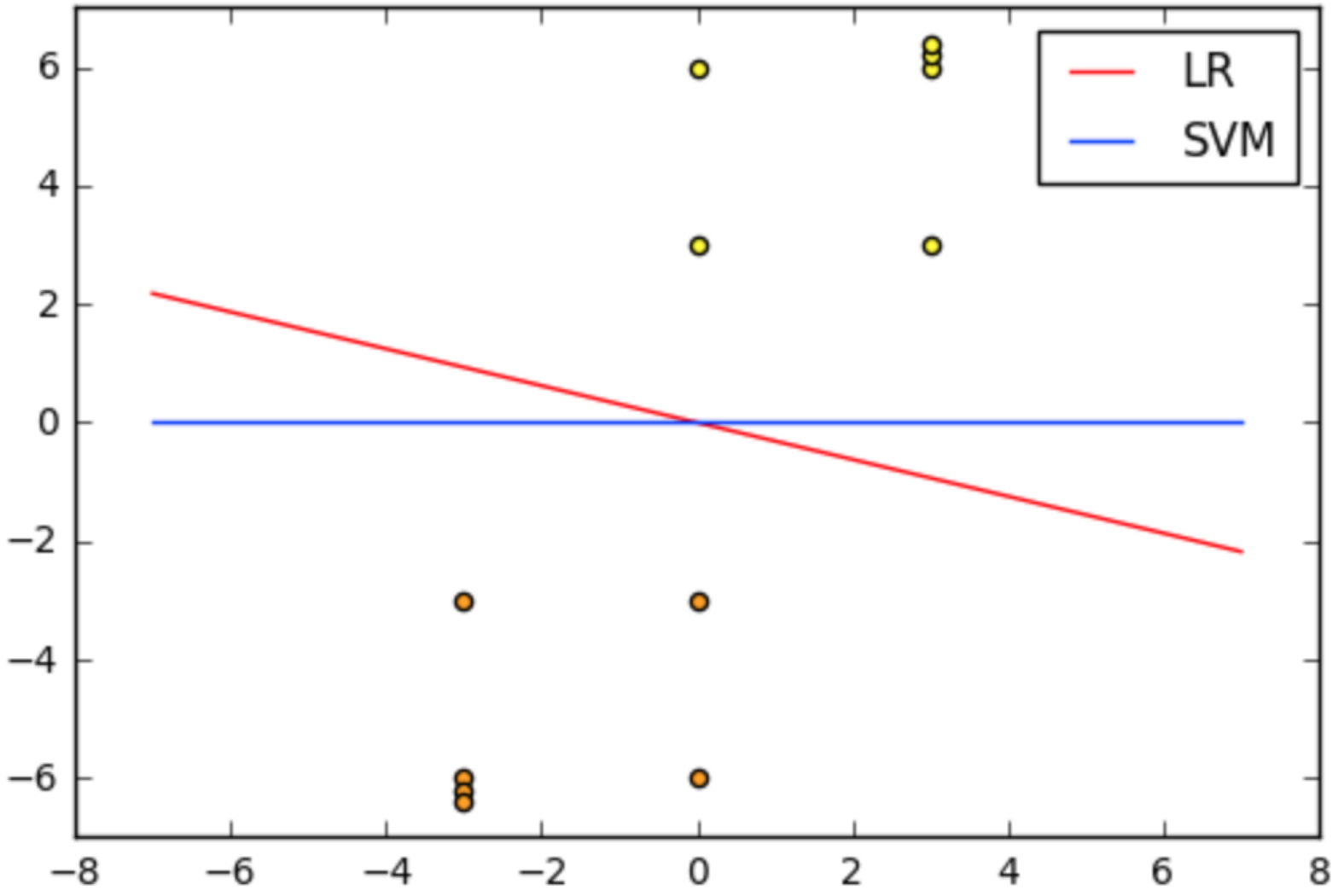
- 데이터의 분포 고려
- 확률 값을 높이는 결정 경계
- 통계적인 접근 방식



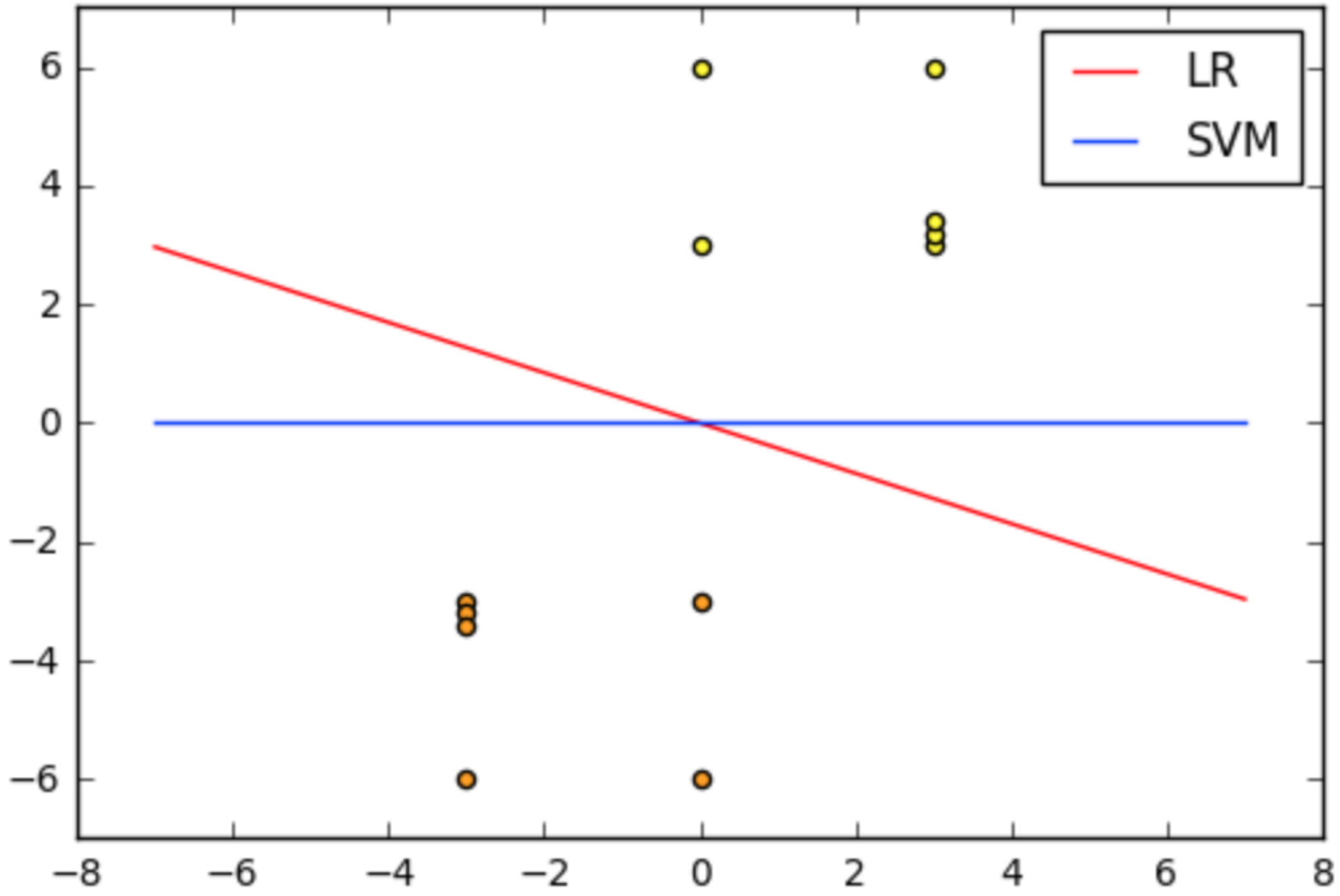
Support Vector Machine

- 데이터의 분포 고려 X
- 마진을 높이는 결정 경계
- 기하학적 접근 방식 (직관)

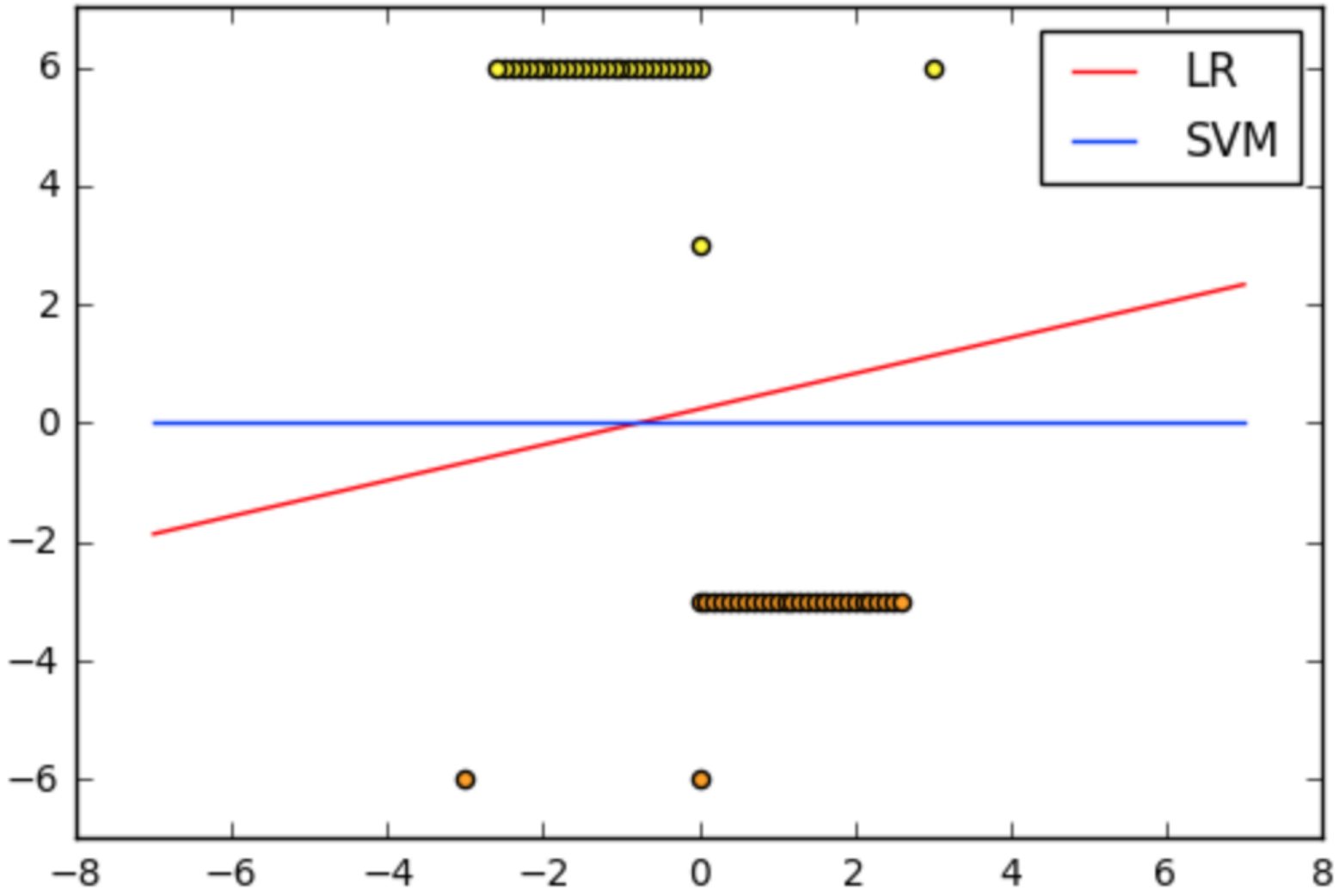
Logistic Regression VS SVM



Logistic Regression VS SVM



Logistic Regression VS SVM

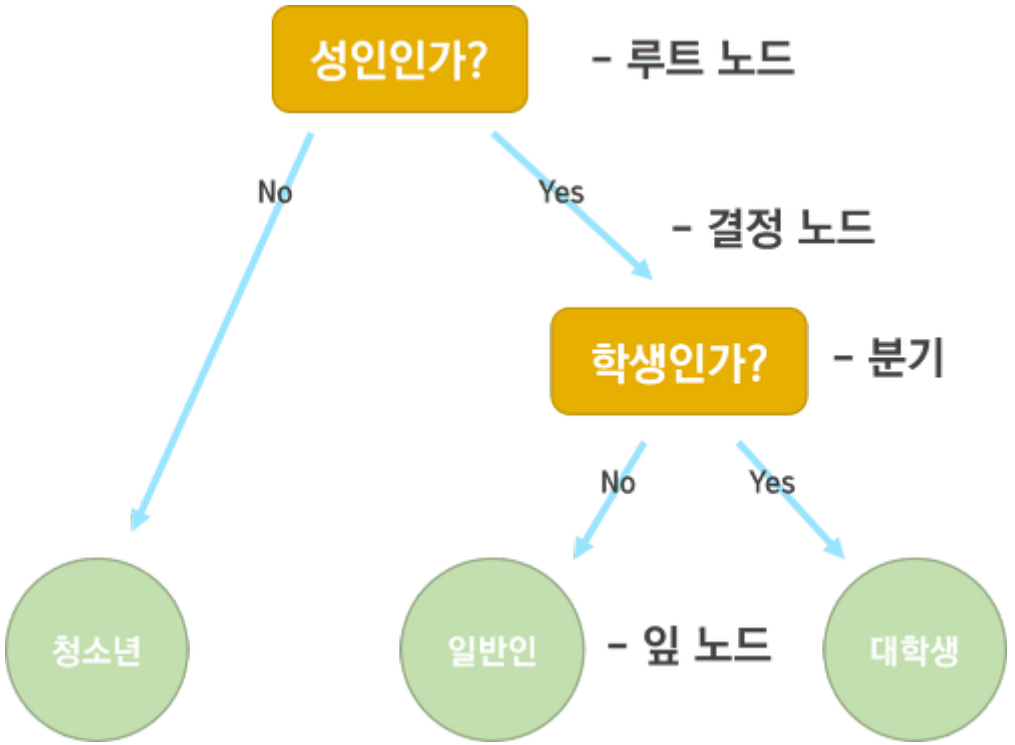


SVM 실습

Decision Tree(Supervised)

- 분류 또는 회귀 문제를 풀기 위한 모델

  ex) 스무고개

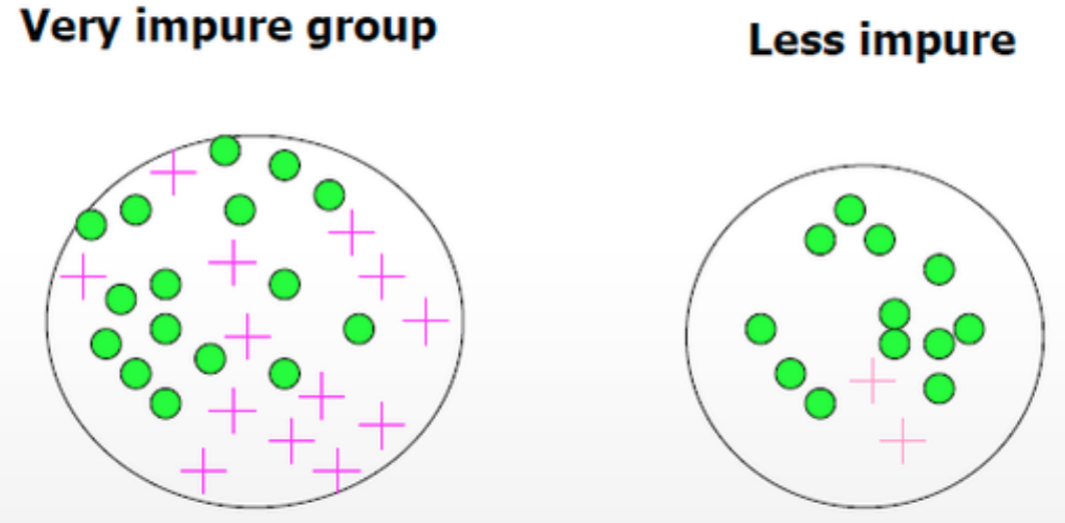


How to construct the decision tree?

- 어떤 순서로 트리를 구성할 것인가 ?
- 엔트로피 (Entropy) : 불순도 (Impurity) 를 측정

$$Entropy = \sum_i - p_i \log_2 p_i$$

p_i is the probability of class i

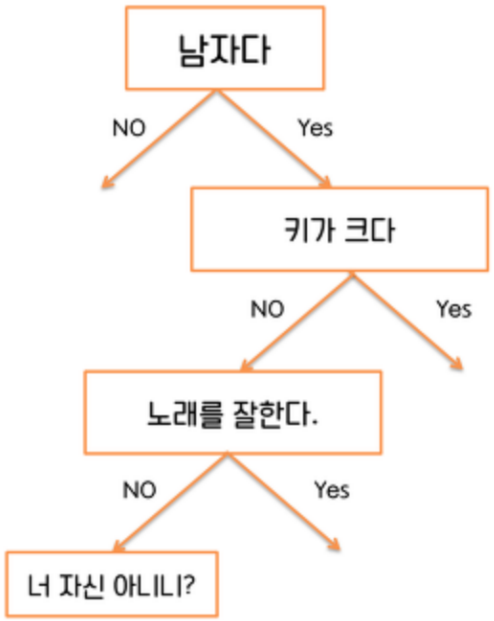


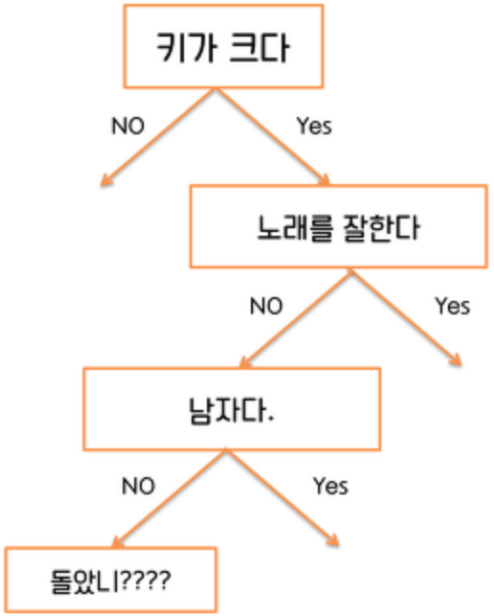
How to construct the decision tree?

- 정보 이득 (Information Gain) : 분기 전 / 후 엔트로피 차이 비교
- 정보 이득이 클수록 변별력이 크다 !

Example 1

Q) 당신의 이상형은 ?





잘 생겼어?	노래 잘해?	키가 커?	이상형
네	아니	네	아니
네	네	네	네
네	네	아니	네
아니	아니	아니	아니

잘 생겼다 : $p(t)H(t) = -\frac{3}{4}(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}) = 0.689$

못 생겼다 : $p(t)H(t) = -\frac{1}{4}(1 * \log_2 1) = 0$

정보 이득 : 1 – 0.689 = 0.311

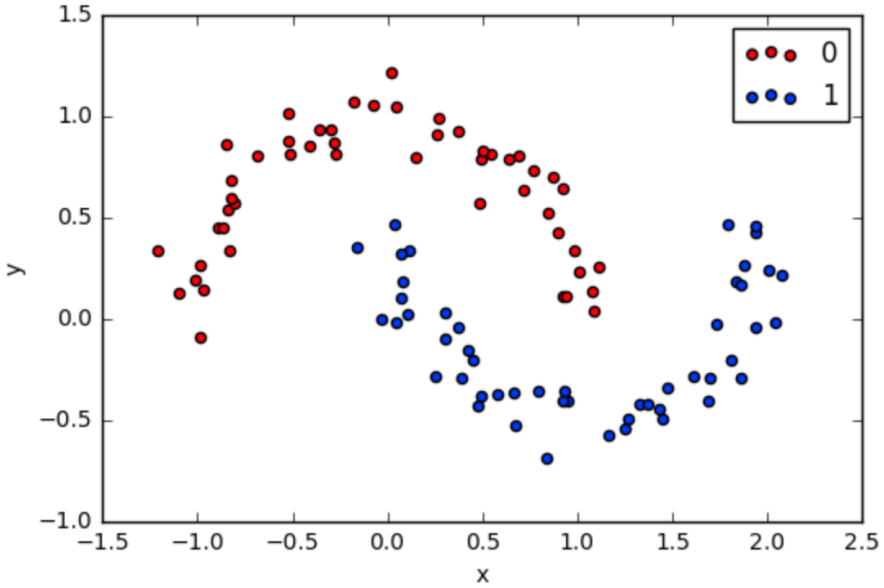
잘 생겼어?	노래 잘해?	키가 커?	이상형
네	아니	네	아니
네	네	네	네
네	네	아니	네
아니	아니	아니	아니

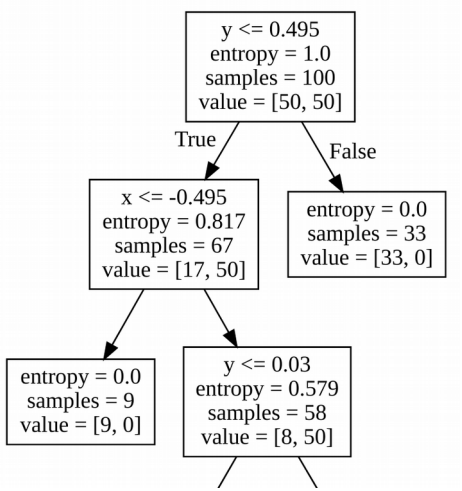
잘 해 : $p(t)H(t) = -\frac{2}{4}(\frac{1}{1}\log_2\frac{1}{1}) = 0$

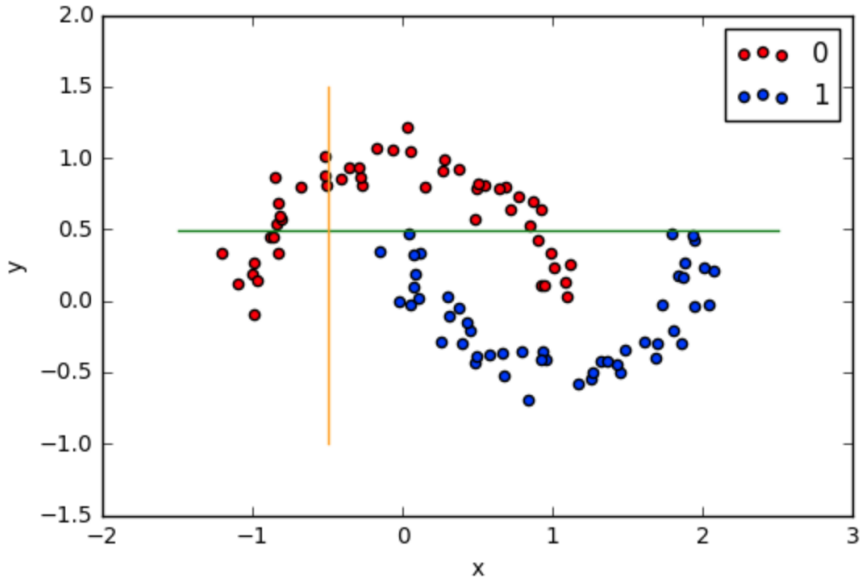
못 해 : $p(t)H(t) = -\frac{2}{4}(1 * \log_2 1) = 0$

정보 이득 : 1 – 0 = 1

Example 2


0
1
x
y


y <= 0.495
entropy = 1.0
samples = 100
value = [50, 50]
True
False
x <= -0.495
entropy = 0.817
samples = 67
value = [17, 50]
entropy = 0.0
samples = 33
value = [33, 0]
entropy = 0.0
samples = 9
value = [9, 0]
y <= 0.03
entropy = 0.579
samples = 58
value = [8, 50]


0
1
x
y

Decision Tree(Supervised)

- 장점

  1) 결과물을 이해하기 쉽다

  2) 스케일에 대한 영향이 적다 (전처리 X)

- 단점

  과적합이 잘 된다 --> 가지치기 & 앙상블 방법

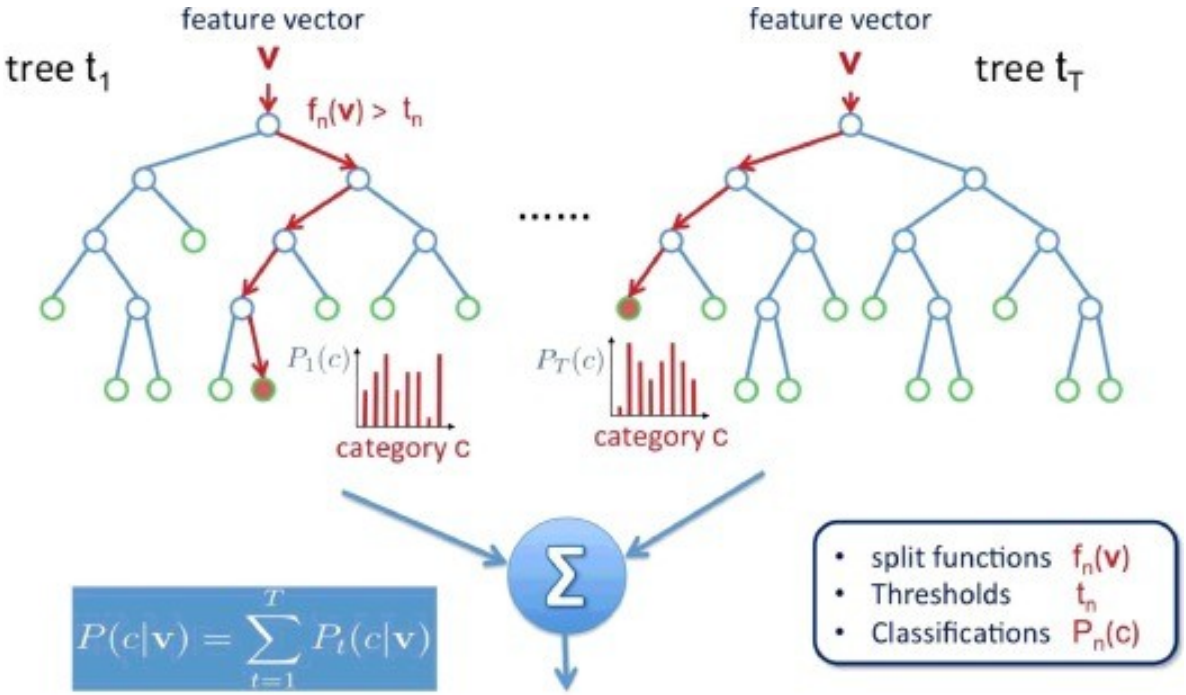
What's ansemble learning?

- 앙상블 학습 : 여러 개의 다른 예측 모델을 결합하여 더 나은 예측 모델을 만드는 기법

- 배깅 (bagging) VS 부스팅 (boosting)
  - 배깅 : 서로 독립된 여러 모델 사용
  - 부스팅 : 이전 모델의 오류를 고려

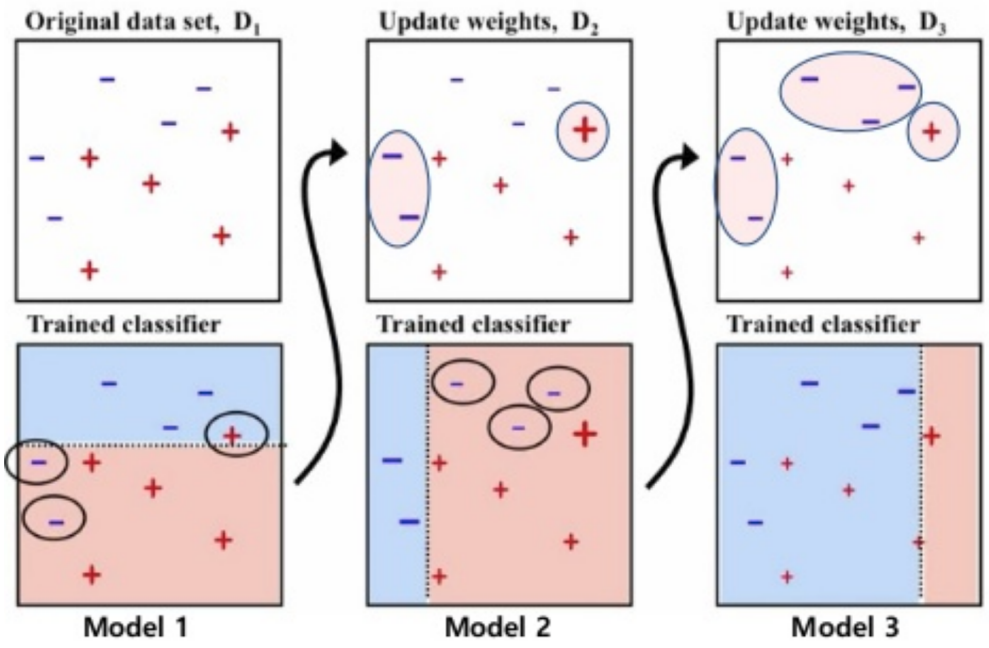
Random Forest

- 무작위 데이터를 추출 (중복 / 누락) 하고 , 일부 특성 만을 사용 (max_features) 하여 여러 개의 트리 생성 (n_estimators)

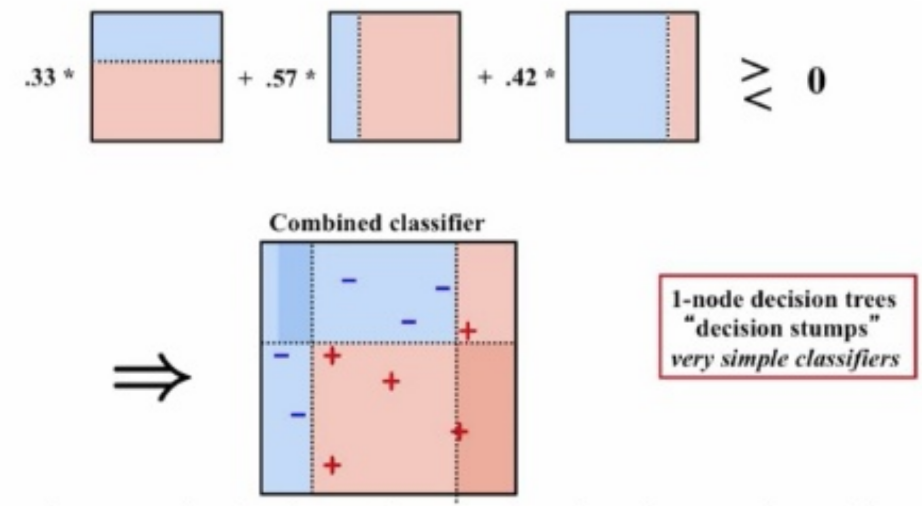


AdaBoost (Adaptive Boosting)

- **weak learner(얕은 트리) 의 오차를 보완하여 순차적으로 트리 생성 --> strong learner**



- 3개의 모델별로 계산된 가중치를 합산하여 최종 모델을 생성

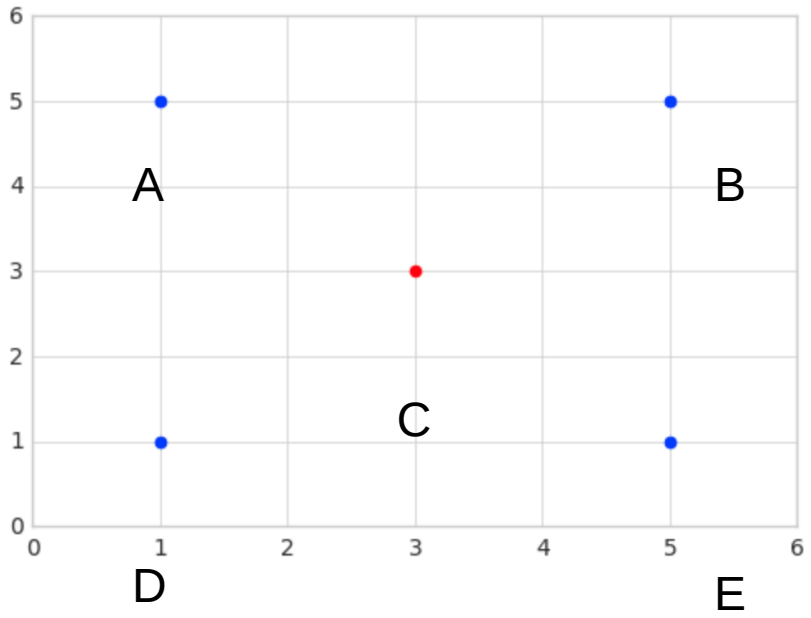


https://www.youtube.com/watch?v=ix6IvwbVpw0&list=PL4zv4UkoVTPflyFDJdJtz248-8Wdz_nct&index=3

Popular boosting tree models

- **XGBoost : 2016 년 3 월에 워싱턴대학교에서 박사로 재학 중인 Tianqi Chen 이 발표한 boosting tree 모델**
- **LightGBM : 2016 년 12 월 Microsoft 에서 발표한 boosting tree 모델**
- **속도가 빠르고 좋은 성능 - 캐글 경진대회에서 많이 사용**

Example - Adaboost



$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

H(x) : Strong Classifier
α : Voting Power
h(x) : Weak Classifier

1. 각 점의 가중치 (w) 초기화

2. 각 h 에 대한 에러 (ε) 계산

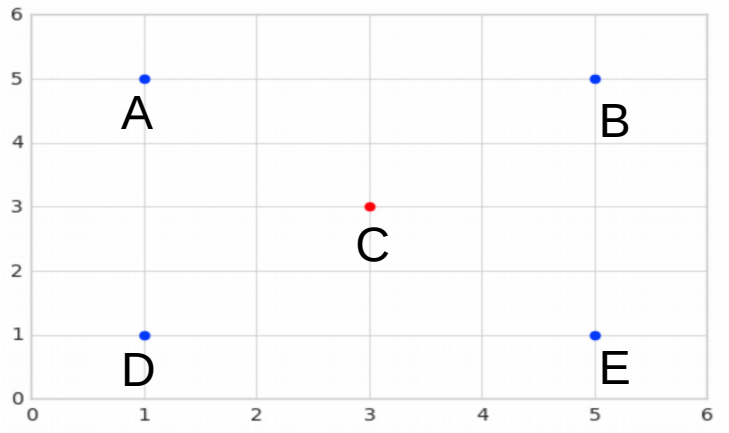
3. 가장 좋은 h 선정 (가장 작은 에러)

4. Voting power(α) 계산
   A = ½ *ln(1-ε/ε)

5. 끝낼 것인가 ?
   - 충분히 좋은 분류기
   - 충분한 반복 횟수
   - 가장 좋은 h 의 ε = ½

6. (5 → NO) 가중치 (w) 업데이트
   w_new = ½ * 1/(1-ε) * w_old (if right)
   = ½ * 1/ε * w_old (if wrong)

Example - Adaboost



H(x) = sign(½ln4(x<6) + ½ln3(x<2) + ½ln5(x>4))

A : ½ln4(B) + ½ln3(B) + ½ln5(R) → (B)

B : ½ln4(B) + ½ln3(R) + ½ln5(B) → (B)

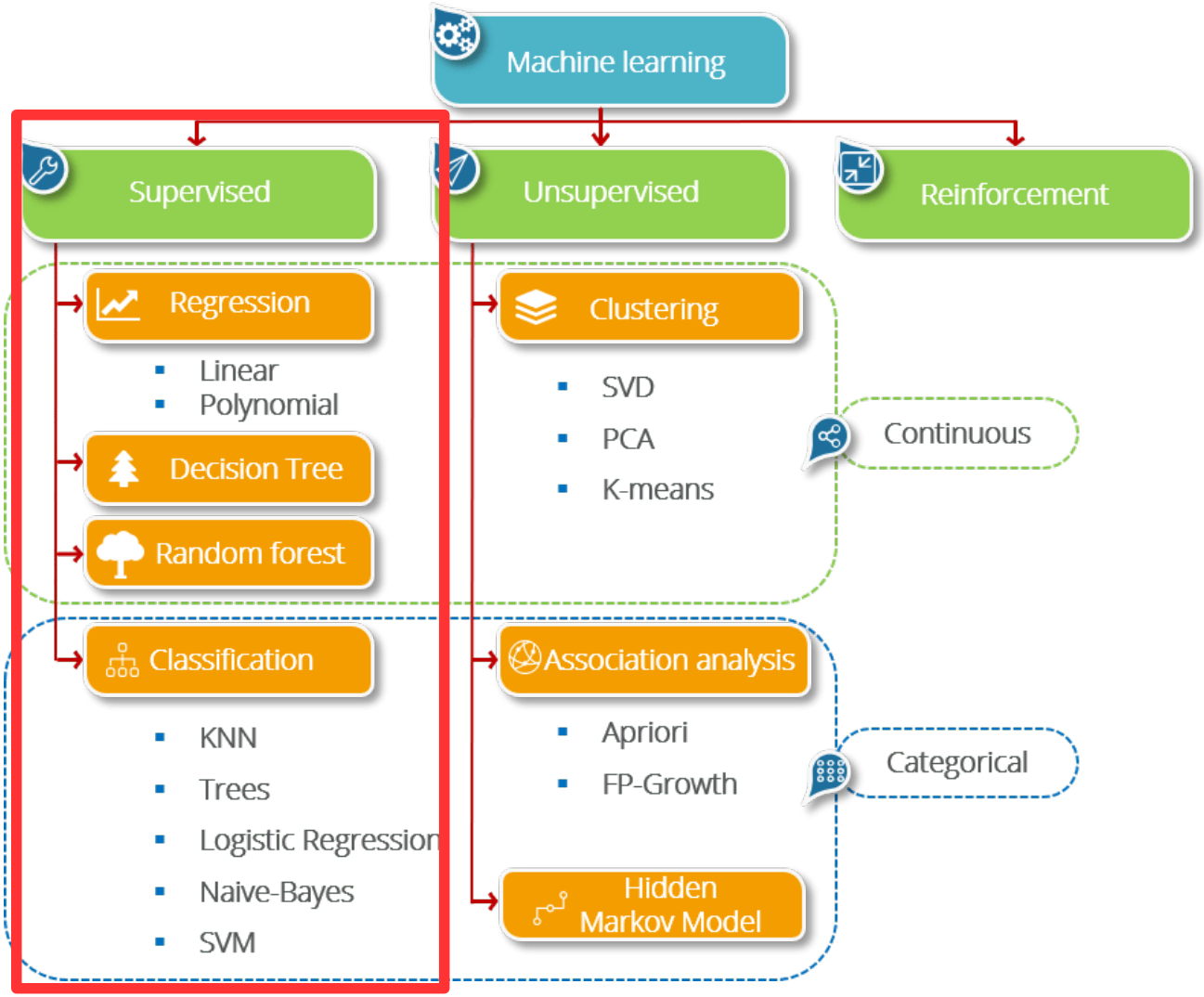
C : ½ln4(B) + ½ln3(R) + ½ln5(R) → (R)

D : ½ln4(B) + ½ln3(B) + ½ln5(R) → (B)

E : ½ln4(B) + ½ln3(R) + ½ln5(B) → (B)

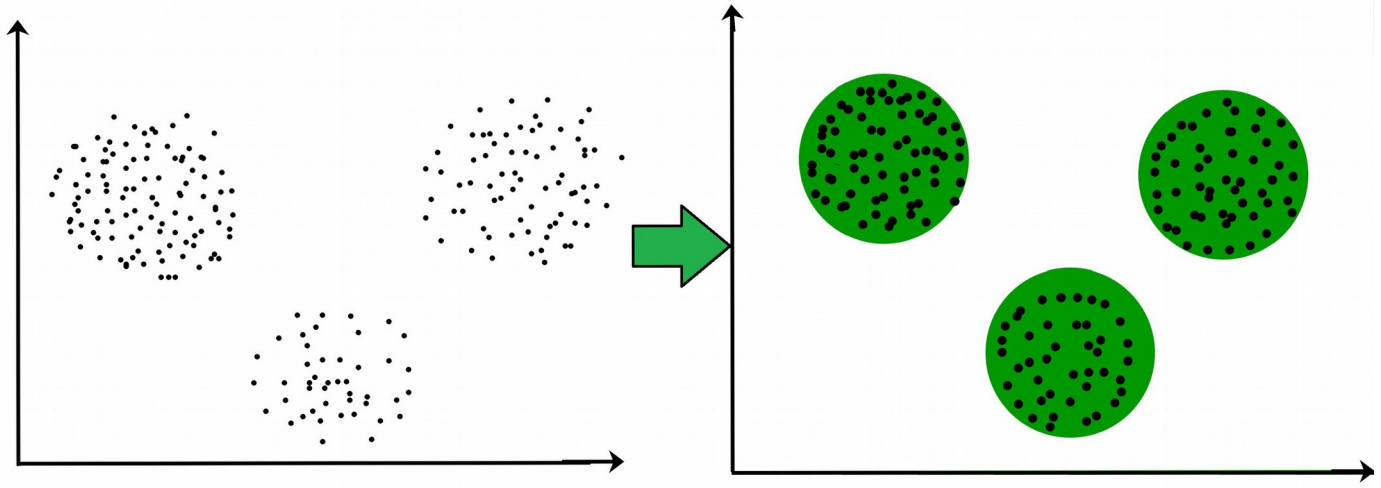
			Round1	Round2	Round3
		w_A	1/5	1/8	1/12
		w_B	1/5	1/8	3/12
		w_C	1/5	1/2	4/12
		w_D	1/5	1/8	1/12
	misclassified	w_E	1/5	1/8	3/12
x<2	B,E	ε	2/5	2/8	1/2
x<4	B,C,E	ε	3/5	6/8	10/12
x<6	C	ε	1/5	4/8	4/12
x>2	A,C,D	ε	3/5	6/8	1/2
x>4	A,D	ε	2/5	2/8	2/12
x>6	A,B,D,E	ε	4/5	4/8	8/12
		h	x<6	x<2	x>4
		ε	1/5	1/4	1/6
		α	1/2*ln4	1/2*ln3	1/2*ln5

Supervised Learning



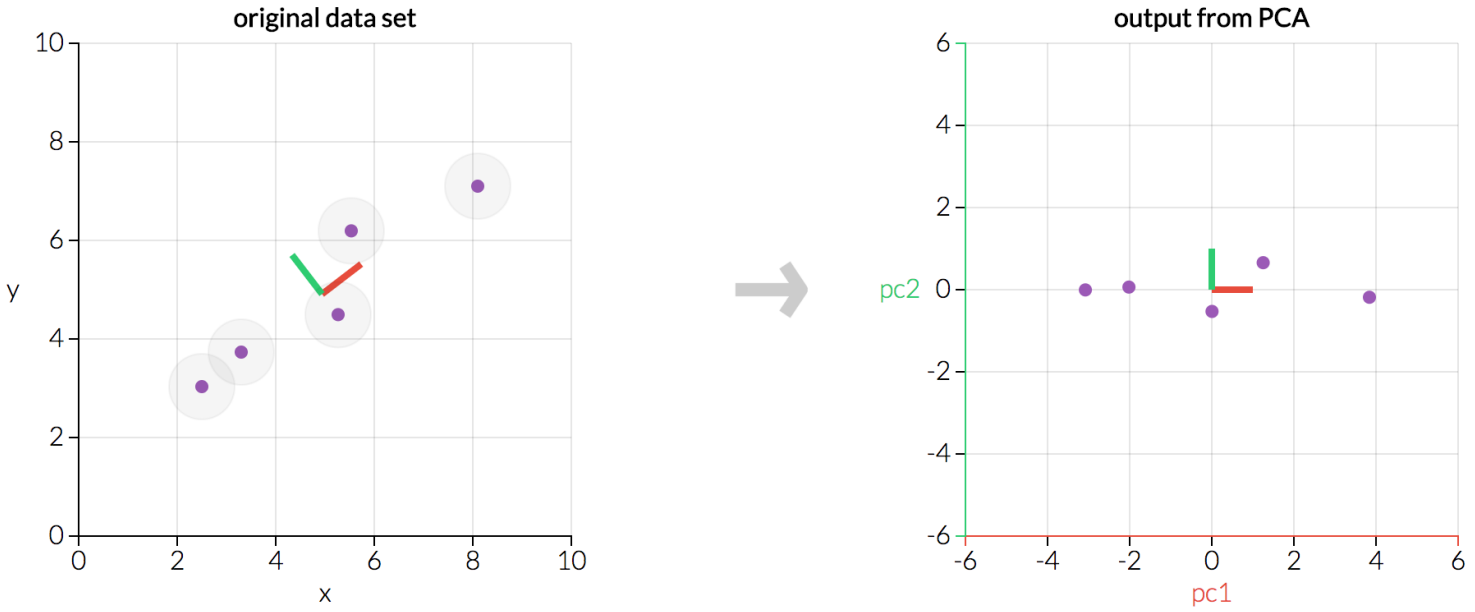
Unsupervised learning

- 데이터 (입력) 에 대한 레이블 (정답) 이 존재하지 않음
- 데이터로부터 특징 또는 구조를 찾음 (데이터를 잘 설명하는 표현)
- 예) 군집화 (clustering)



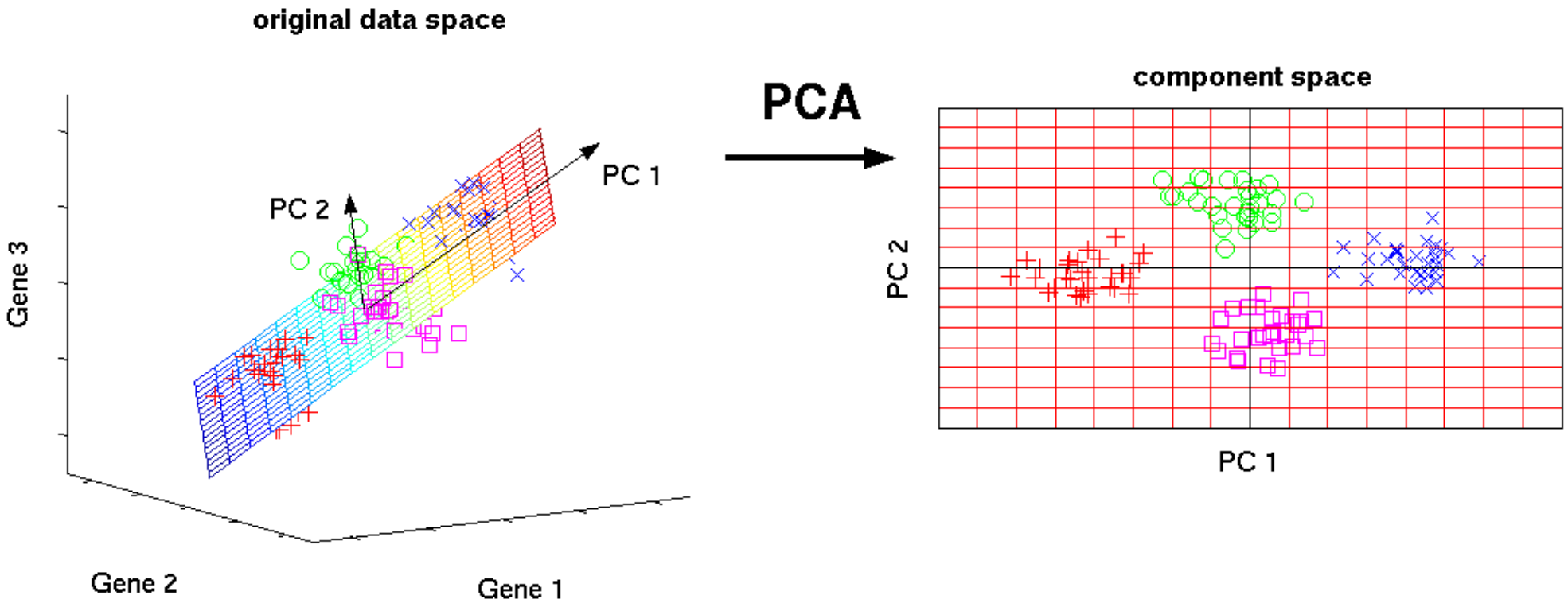
PCA(Principle Component Analysis)

- 차원 축소 기법 (선형 변환)
- 분포된 데이터들의 주성분 (분산이 큰 방향) 을 찾음
- 대부분의 정보를 보존

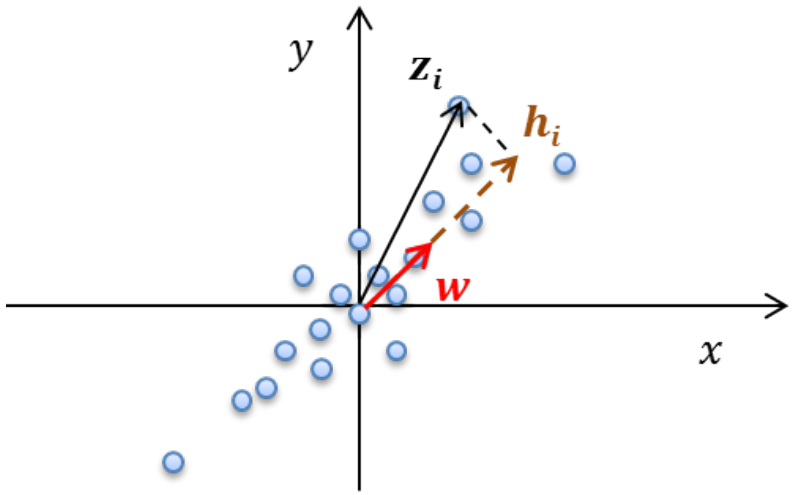


PCA – dimensionality reduction

- 다음 데이터 (3 차원) 의 분포 특성을 2 개의 벡터 (2 차원) 로 표현할 수 있을까 ?



Find a vector to maximize variance



$$\vec{h}_i = (\vec{z}_i \cdot \vec{w})\vec{w}$$

z : 데이터

w : 크기가 1 인 임의의 단위 벡터

(분산이 가장 큰 방향)

h : w 방향의 정사영 벡터

데이터 $\vec{z}_i$의 분산을 최대화 = $\vec{h}_i$의 크기 $(\vec{z}_i \cdot \vec{w})$ 의 분산을 최대화 !!

Find a vector to maximize variance

$$\sigma_w^2 = \frac{1}{n}\sum_i (z_i \cdot w)^2 - \left(\frac{1}{n}\sum_i (z_i \cdot w)\right)^2$$

($\vec{z}_i$ 의 평균 = 0)

$$= \frac{1}{n}\sum_i (z_i \cdot w)^2$$

$$= \frac{1}{n}(Zw)^T(Zw)$$

$$= \frac{1}{n} w^T Z^T Z w$$

$$= w^T \frac{Z^T Z}{n} w$$

$$= w^T C w$$

$\sigma_w^2 : (\vec{z}_i \cdot \vec{w})$ 의 분산

Covariance Matrix of Data features $\vec{z}_i$

$$u = w^T C w - \lambda(w^T w - 1)$$

$$\frac{\partial u}{\partial w} = 2Cw - 2\lambda w = 0$$

$$Cw = \lambda w$$

분산을 최대로 하려는 w 를 구하기 위해 라그랑지안 승수법 사용

Covariance Matrix

$$X = \begin{pmatrix} | & | & | & & | \\ X_1 & X_2 & X_3 & \cdots & X_d \\ | & | & | & & | \end{pmatrix} \in \mathbb{R}^{n \times d}$$

: 행렬의 각 열 (feature) 의 평균 값 = 0

$$X^T X = \begin{pmatrix} — & X_1 & — \\ — & X_2 & — \\ & \cdots & \\ — & X_d & — \end{pmatrix} \begin{pmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{pmatrix}$$

$$\frac{X^T X}{n} = \frac{1}{n} \begin{pmatrix} dot(X_1, X_1) & dot(X_1, X_2) & \cdots & dot(X_1, X_d) \\ dot(X_2, X_1) & dot(X_2, X_2) & \cdots & dot(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d, X_1) & dot(X_d, X_2) & \cdots & dot(X_d, X_d) \end{pmatrix}$$

PCA

- 우리가 구하려는 것은 분산이 가장 큰 방향 !
- 분산을 최대로 하는 방향 벡터

  = 공분산 행렬의 **eigenvector**
- 분산의 크기 = 공분산 행렬의 **eigenvalue**

PCA

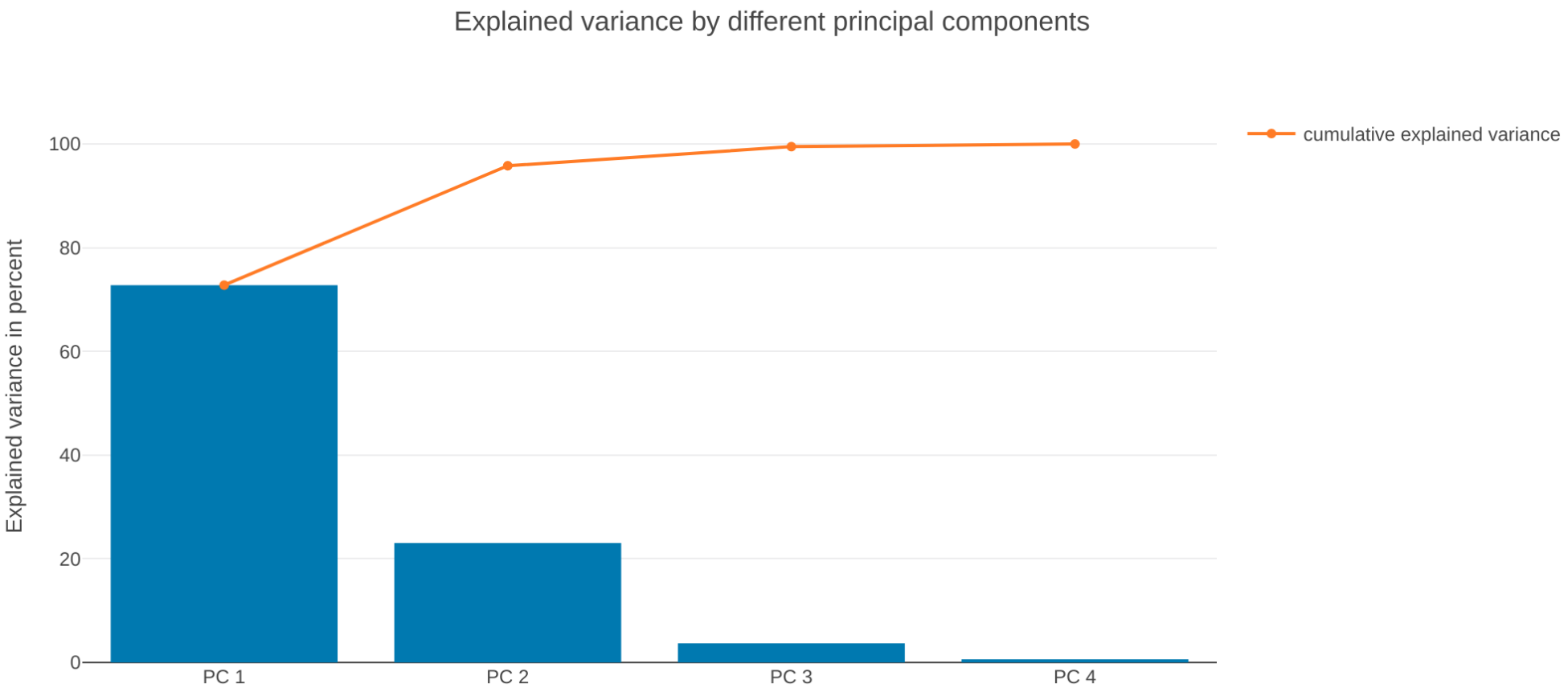
PCA

- C : covariance matrix of x
- $C = P\Sigma P^T$ (P: orthogonal, Σ: diagonal)

$$C = \begin{pmatrix} e_1 & \cdots & e_n \end{pmatrix} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \begin{pmatrix} e_1^T \\ \vdots \\ e_n^T \end{pmatrix}$$

- P : n×n orthogonal matrix
- Σ : n×n diagonal matrix
- $Ce_i = \lambda_i e_i$
  - e_i : eigenvector of C, direction of variance
  - λ_i : eigenvalue, e_i 방향으로의 분산
  - $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$
- e_1: 가장 분산이 큰 방향
- e_2: e_1에 수직이면서 다음으로 가장 분산이 큰 방향
- e_k: $e_1, \ldots, e_{k-1}$에 모두 수직이면서 가장 분산이 큰 방향

How many dimensions?



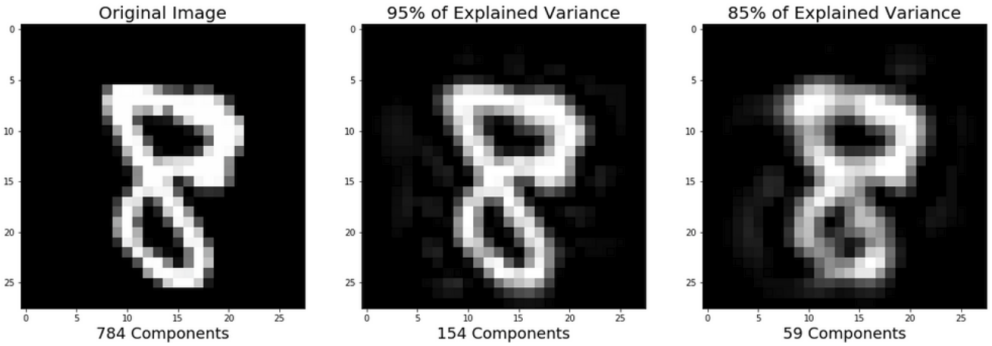
Application of PCA

1) Data Visualization

2) Speed-up ML Algorithms

Variance Retained	Number of Components	Time (seconds)	Accuracy
1.00	784	48.94	0.9158
0.99	541	34.69	0.9169
0.95	330	13.89	0.9200
0.90	236	10.56	0.9168
0.85	184	8.85	0.9156

3) Data Compression

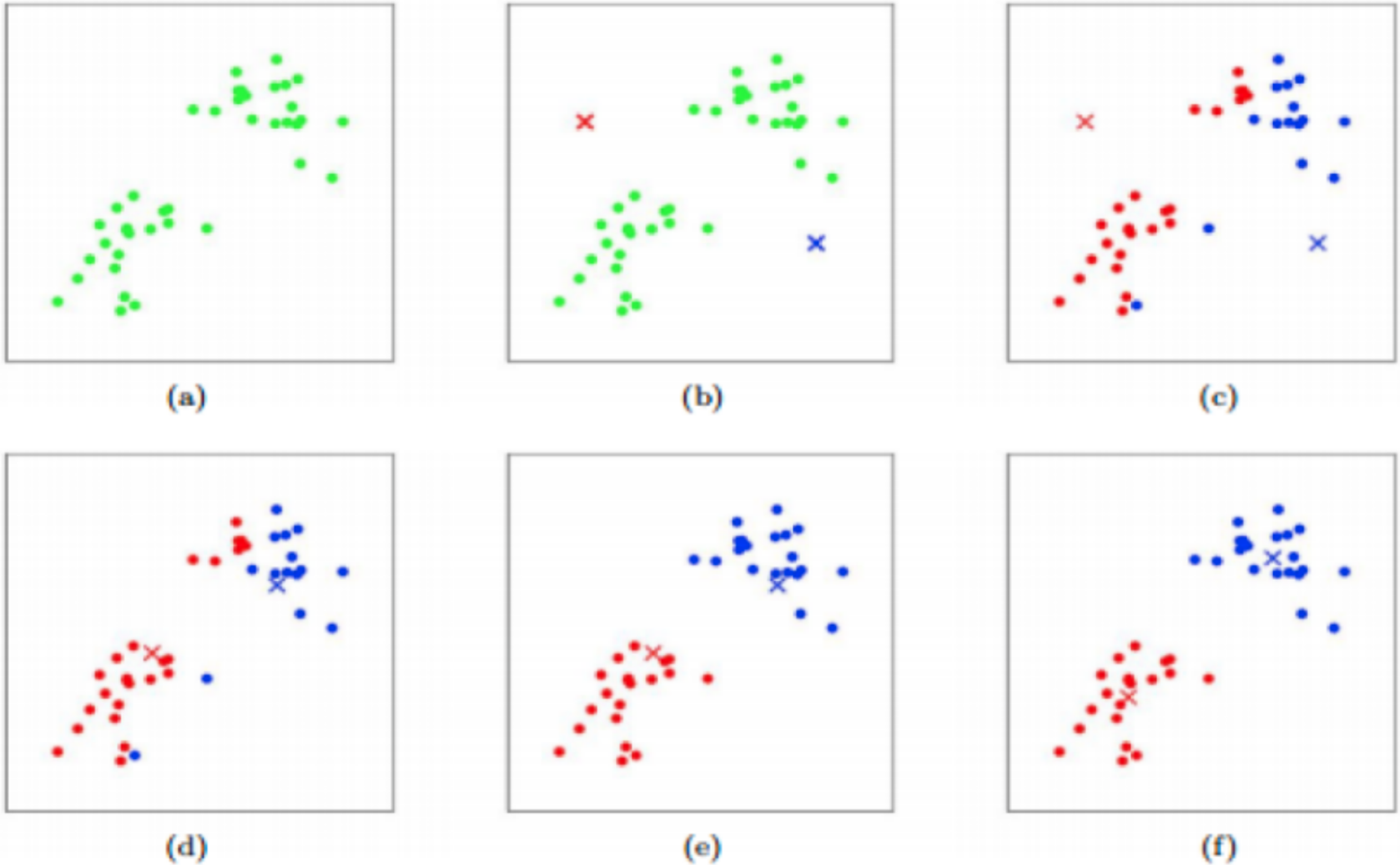


K-means clustering(Unsupervised)

- 대표적인 군집화 알고리즘

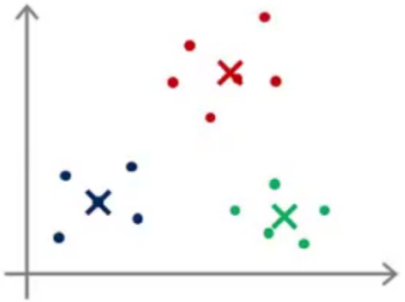
  1. 데이터 집합으로부터 K 개의 초기 대표 벡터 (centroid) 를 선택한다 .
  2. 데이터를 거리가 가장 가까운 centroid 의 cluster 에 속하게 한다 .
  3. 각 cluster 의 새로운 centroid 를 뽑는다 .
  4. 위의 과정을 centroid 가 수렴할 때까지 반복한다 .

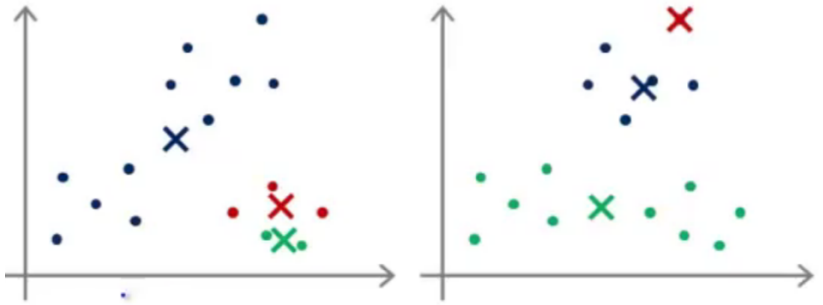
K-means clustering(Unsupervised)



Problem

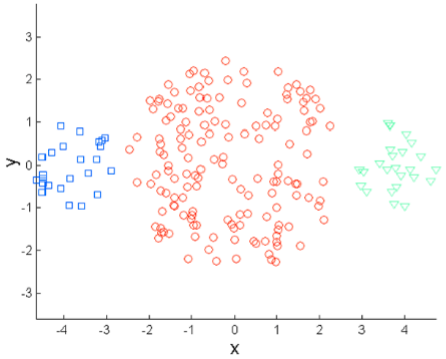
- 적절한 K 를 찾는 일 (군집의 갯수)
- 랜덤으로 초기화 (Random Initialization)

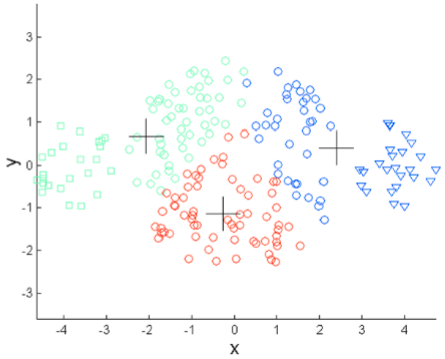


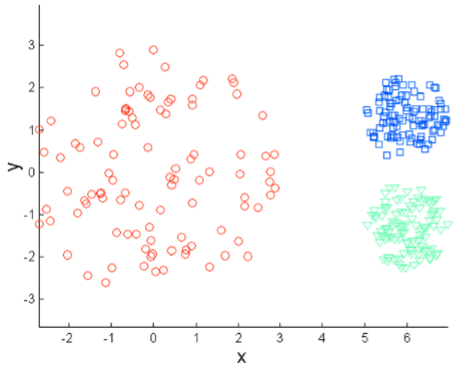


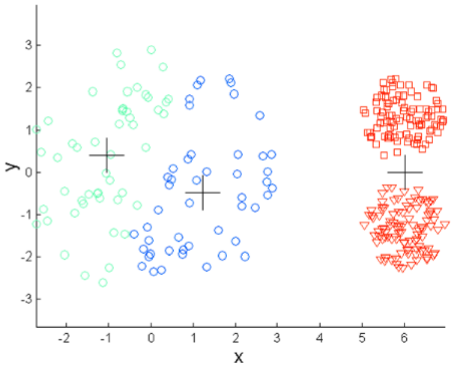
Problem

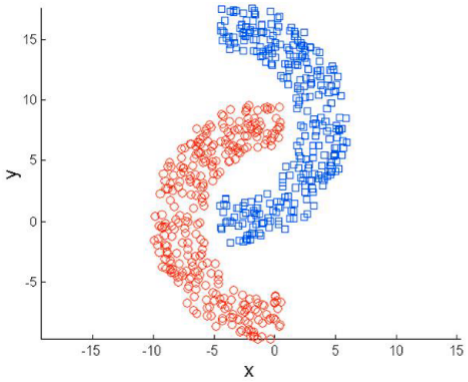
- 그 외 (크기 / 밀도 / 분포)

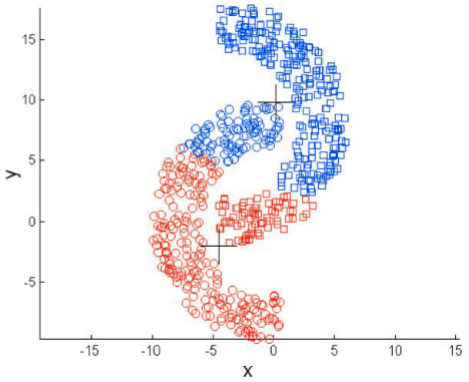




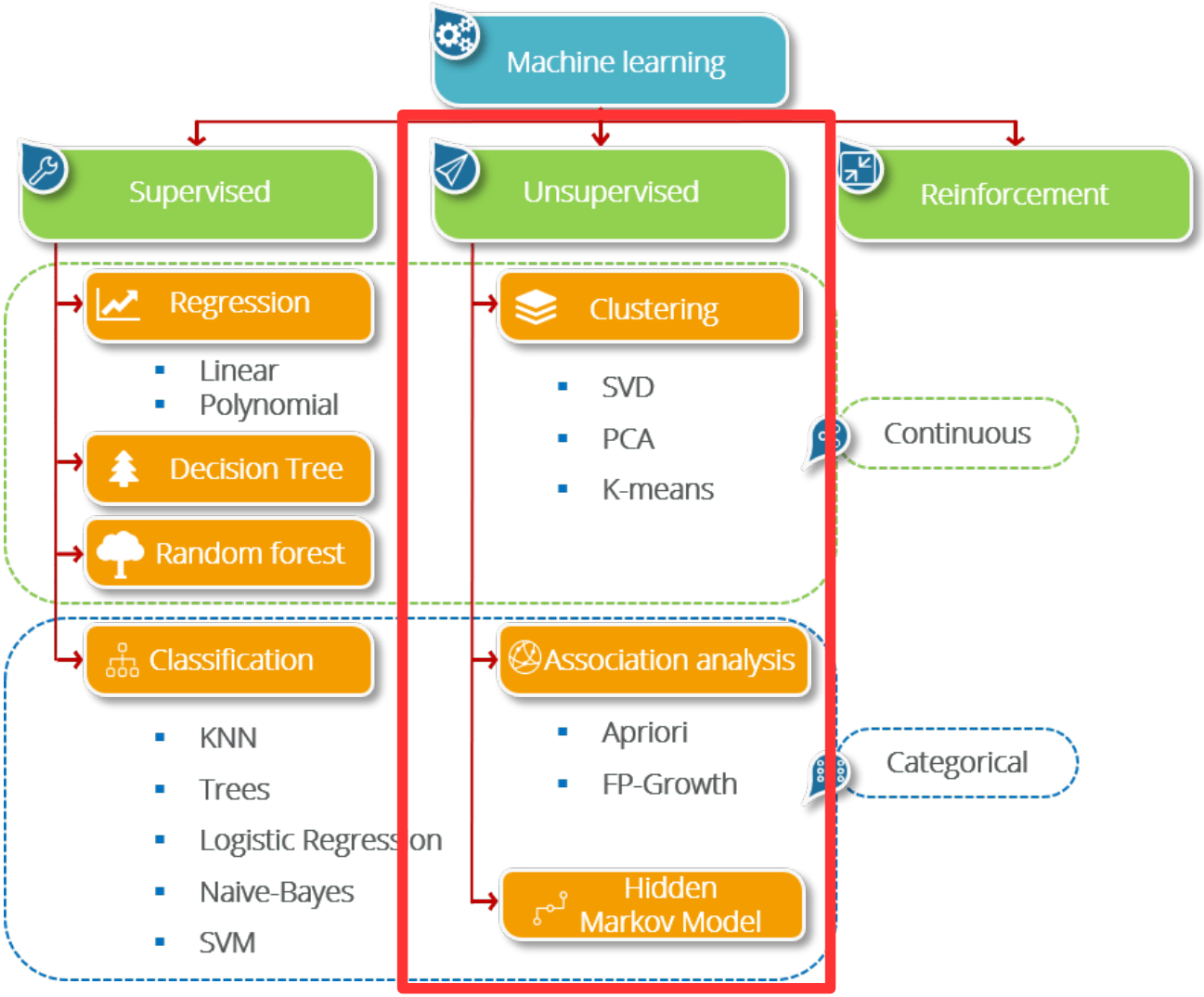






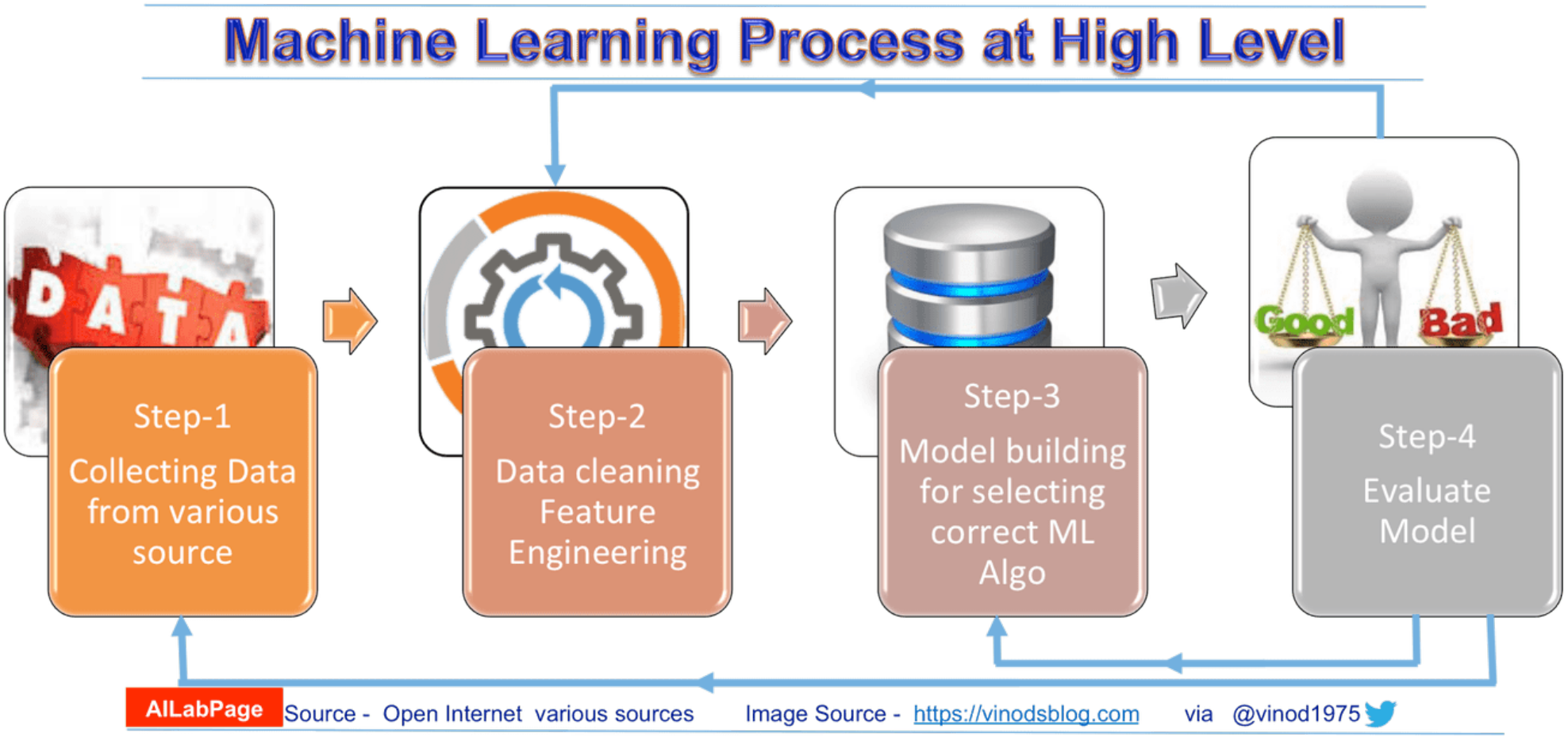


Unsupervised Learning



PCA 실습

Machine Learning Pipeline



Collecting data

Santander Product Recommendation

Can you pair products with people?

$60,000 · 1,787 teams · 2 years ago

	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antiguedad	indrel	ult_fec_cli_1t	indrel_1mes	tiprel_1mes	indresi	indext	conyuemp	ca
2	2015-01-28	1375586	N	ES	H	35	2015-01-12	0	6	1		1.0	A	S	N		KH
3	2015-01-28	1050611	N	ES	V	23	2012-08-10	0	35	1		1	I	S	S		KH
4	2015-01-28	1050612	N	ES	V	23	2012-08-10	0	35	1		1	I	S	N		KH
5	2015-01-28	1050613	N	ES	H	22	2012-08-10	0	35	1		1	I	S	N		KH
6	2015-01-28	1050614	N	ES	V	23	2012-08-10	0	35	1		1	A	S	N		KH
7	2015-01-28	1050615	N	ES	H	23	2012-08-10	0	35	1		1	I	S	N		KH
8	2015-01-28	1050616	N	ES	H	23	2012-08-10	0	35	1		1	I	S	N		KH
9	2015-01-28	1050617	N	ES	H	23	2012-08-10	0	35	1		1	A	S	N		KH
10	2015-01-28	1050619	N	ES	H	24	2012-08-10	0	35	1		1	I	S	N		KH
11	2015-01-28	1050620	N	ES	H	23	2012-08-10	0	35	1		1	I	S	N		KH
12	2015-01-28	1050621	N	ES	V	23	2012-08-10	0	35	1		1	I	S	N		KH
13	2015-01-28	1050622	N	ES	H	23	2012-08-10	0	35	1		1	I	S	N		KH
14	2015-01-28	1050623	N	ES	H	23	2012-08-10	0	35	1		1	A	S	N		KH
15	2015-01-28	1050624	N	ES	H	65	2012-08-10	0	35	1		1	A	S	N		KH
16	2015-01-28	1050625	N	ES	V	23	2012-08-10	0	35	1		1	A	S	N		KH
17	2015-01-28	1050626	N	ES	V	23	2012-08-10	0	35	1		1	A	S	N		KH
18	2015-01-28	1050610	N	ES	V	24	2012-08-10	0	35	1		1	I	S	N		KH
19	2015-01-28	1050627	N	ES	H	23	2012-08-10	0	35	1		1	I	S	N		KH
20	2015-01-28	1050609	N	ES	H	22	2012-08-10	0	35	1		1	I	S	N		KF
21	2015-01-28	1050605	N	ES	V	23	2012-08-10	0	35	1		1	I	S	N		KH
22	2015-01-28	1050582	N	ES	V	28	2012-08-10	0	35	1		1	I	S	N		KH
23	2015-01-28	1050586	N	ES	V	23	2012-08-10	0	35	1		1	A	S	N		KH
24	2015-01-28	1050588	N	ES	H	22	2012-08-10	0	35	1		1	I	S	N		KH
25	2015-01-28	1050589	N	ES	V	23	2012-08-10	0	35	1		1	I	S	N		KH
26	2015-01-28	1050591	N	ES	H	23	2012-08-10	0	35	1		1	A	S	N		KH
27	2015-01-28	1050592	N	ES	H	22	2012-08-10	0	35	1		1	I	S	N		KH
28	2015-01-28	1050595	N	ES	V	25	2012-08-10	0	35	1		1	A	S	N		KH
29	2015-01-28	1050596	N	ES	H	25	2012-08-10	0	35	1		1	A	S	S		KH
30	2015-01-28	1050597	N	ES	H	23	2012-08-10	0	35	1		1	A	S	N		KH

산탄데르 은행 : 금융 상품 판매를 통한 매출 상승 - 고객에게 적절한 금융 상품 추천하기 위해 고객이 신규로 구입할 제품을 예측 (24 개의 금융 제품 중 예상되는 상위 7 개)

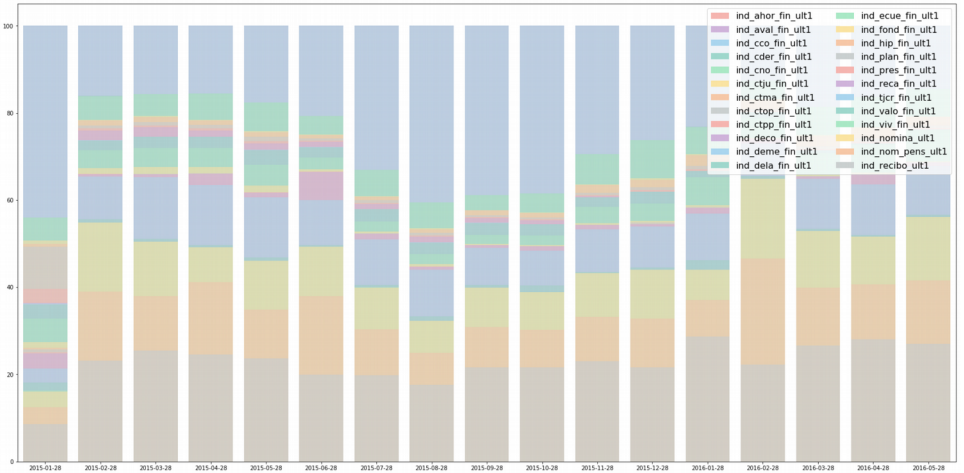
Explore Data Analysis(EDA)

- 본인 나름대로 데이터로부터 의미를 찾는 것

  (데이터 기초 통계 분석 , 시각화 분석 , 변수 아이디어 , 예측 변수에 대한 특징)

	ncodpers	ind_nuevo	indrel	tipodom	cod_prov	ind_actividad_cliente	renta
count	1.364731e+07	1.361958e+07	1.361958e+07	13619574.0	1.355372e+07	1.361958e+07	1.085293e+07
mean	8.349042e+05	5.956184e-02	1.178399e+00	1.0	2.657147e+01	4.578105e-01	1.342543e+05
std	4.315650e+05	2.366733e-01	4.177469e+00	0.0	1.278402e+01	4.982169e-01	2.306202e+05
min	1.588900e+04	0.000000e+00	1.000000e+00	1.0	1.000000e+00	0.000000e+00	1.202730e+03
25%	4.528130e+05	0.000000e+00	1.000000e+00	1.0	1.500000e+01	0.000000e+00	6.871098e+04
50%	9.318930e+05	0.000000e+00	1.000000e+00	1.0	2.800000e+01	0.000000e+00	1.018500e+05

	fecha_dato	ind_empleado	pais_residencia	sexo	age	fecha_alta	antiguedad	ult_fec_cli_1t	indrel_1mes	tiprel_1mes	indresi	indext
count	13647309	13619575	13619575	13619505	13647309	13619575	13647309	24793	13497528.0	13497528	13619575	13619575
unique	17	5	118	2	235	6756	507	223	13.0	5	2	2
top	2016-05-28	N	ES	V	23	2014-07-28	0	2015-12-24	1.0	I	S	N
freq	931453	13610977	13553710	7424252	542682	57389	134335	763	7277607.0	7304875	13553711	12974839



Data preprocessing

- 결측치 (NA) 처리
- 이상치 (outlier) 처리
- 잘못 입력된 데이터 처리 (ex. 나이 : 466)
- 데이터 변환 (범주형 → 수치형)
- Etc.

Feature Engineering

- 스케일링
- 변수 삭제
- 파생 변수 생성 가능

  ex. 주중 / 주말 , 계절 , 시간 , 과거 데이터 활용

  (시계열 데이터)
- Etc.

Model building

- 교차 검증
- 모델 튜닝 : 최적의 파라미터 찾기

```
# 사용하기 위한 컬럼
f_to_use = ['user_total_orders', 'user_total_items',
       'user_average_days_between_orders', 'user_average_basket', 'user_
       'aisle_id', 'department_id', 'product_orders', 'ave_dow', 'ave_ho
       'product_reorder_rate', 'dow', 'order_hour_of_day', 'UP_num_item'
       'UP_frequency', 'UP_average_add_cart', 'UP_reorder_rate', 'UP_F_L
```

```
d_train = lgb.Dataset(df_train[f_to_use],
                      label=labels, # 주문 : 1, 주문X : 0
                      categorical_feature=['aisle_id', 'department_id'])
```

```
params = {
    'task': 'train',
    'boosting_type': 'gbdt',
    'objective': 'binary',
    'metric': ['binary_logloss','auc'],
    'num_leaves': 120,
    'max_depth': 7,
    'learning_rate': 0.05,
    'feature_fraction': 0.9,
    'bagging_fraction': 0.9,
    'bagging_freq': 10
}
ROUNDS = 300

bst = lgb.train(params, d_train, ROUNDS)
```

```
preds = bst.predict(df_test[f_to_use])
preds
```

```
array([ 0.33109674,  0.07060704,  0.11511062, ...,  0.02278696,
        0.01209684,  0.0373361 ])
```

```
TRESHOLD = 0.15

d = dict()
for row in df_test.itertuples():
    if row.pred > TRESHOLD:
        try:
            d[row.order_id] += ' ' + str(row.product_id)
        except:
            d[row.order_id] = str(row.product_id)

for order in test_orders.order_id:
    if order not in d:
        d[order] = 'None'
```

```
d
```

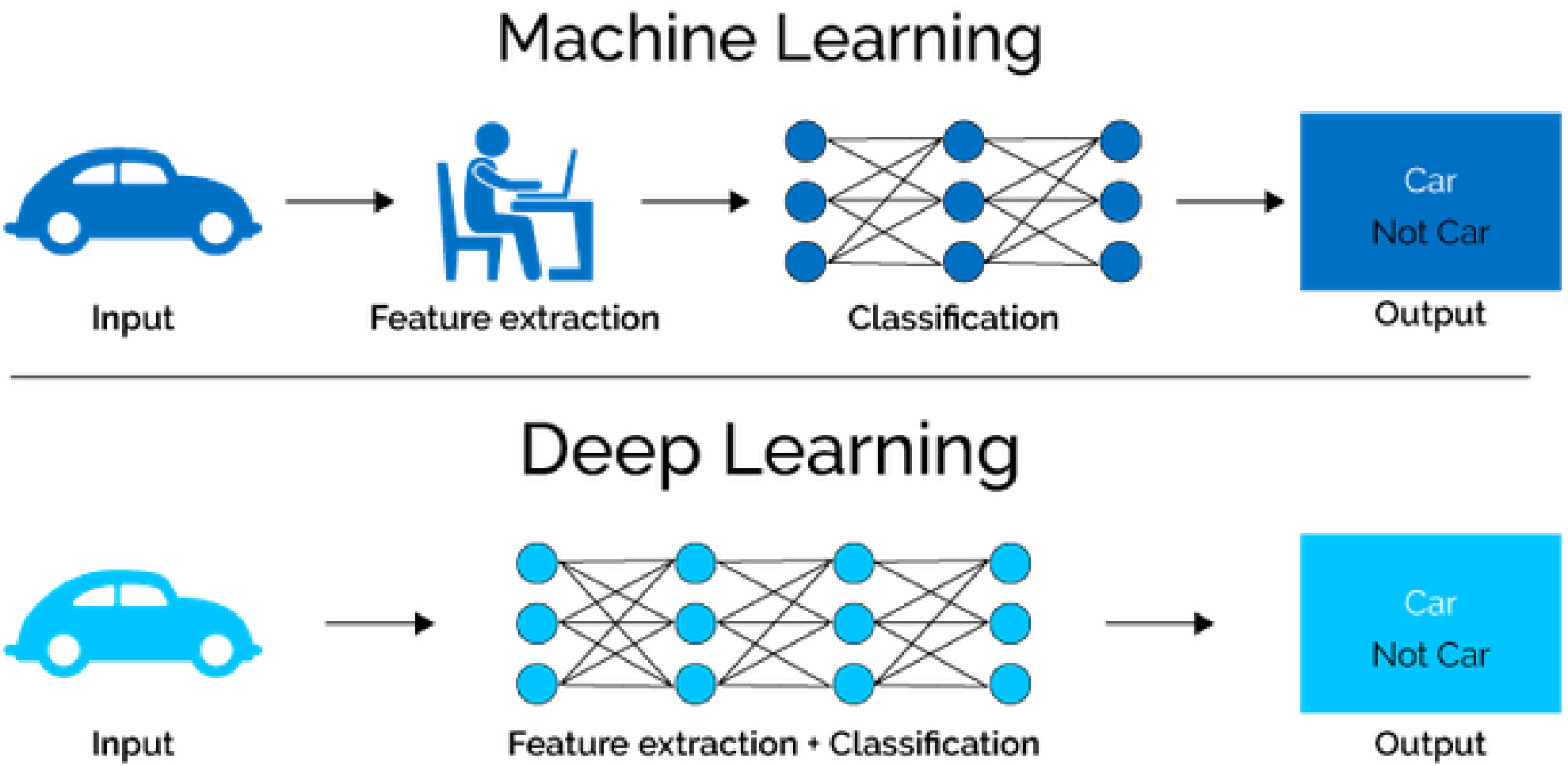
```
{2774568: '17668 21903 32402 22035 39190 47766 16797 18599 43961 23650 24810',
 1528013: '27521 8424 21903 38293 25659',
 1376945: '17794 33572 17706 28465 27959 44632 24799 34658 14947 30563 35948 8309 13176',
 1356845: '11520 14992 7076 22959 37687 7120 28134 10863 21616 13176',
 2161313: '11266 196 10441 12427 37710 48142 1747 14715 27839',
 1416320: '15872 28289 43014 5134 21903 21137 24852 48283 17948 21405 28985 41950 21616 24561',
 1735923: '5183 17008 2192 9387 196 15599 31487 15131 35123 12108 33122 34690 42913',
 1980631: '13575 6184 9387 46061 13914 41400 22362',
 139655: '22935 27845 13176',
 1411408: '26452 22008',
 2940603: '30592 19894 7355 44632 10339 14947 18531 31615',
 1192143: '47626 49683 27307 24759 8424 24852',
 280888: '35413 13966 12440 19213 36144 32566 41406',
 3202221: '46979 21137 24852 2452 49175 9637 21927 24489 43692 45364 6069 4793 13629 27966 49215 20168 45774 10831
17630 7781 39911 48110 9203 11130',
 3222866: '40706 13187 37131 32912 7969 38690 33198 8501 32441 14283 18894 34254 35921 14947 27903 15718 49517 703
9',
 707453: '45066 42585 26209 48230 44142 21137 47766 694 46802 18150 24852 21267 32030 44830 2846 7969 8518 4942 39
275 37766 21903 42450 6111 28156',
```

Evaluate model

- **다수의 모델을 앙상블 한 것이 좋은 성능**

순위	팀명 (점수)	모델
1	idle_speculation (0.031409)	인공신경망 x 12 + XGBoost x 8
2	Tom Van de Wiele (0.031317)	XGBoost x 26
3	Jack (0.0313156)	제품별 다른 XGBoost 모델
...	...	...
8	Alejo y Miro (0.0311226)	LightGBM + XGBoost

ML VS DL(Representation learning)




Thank
you