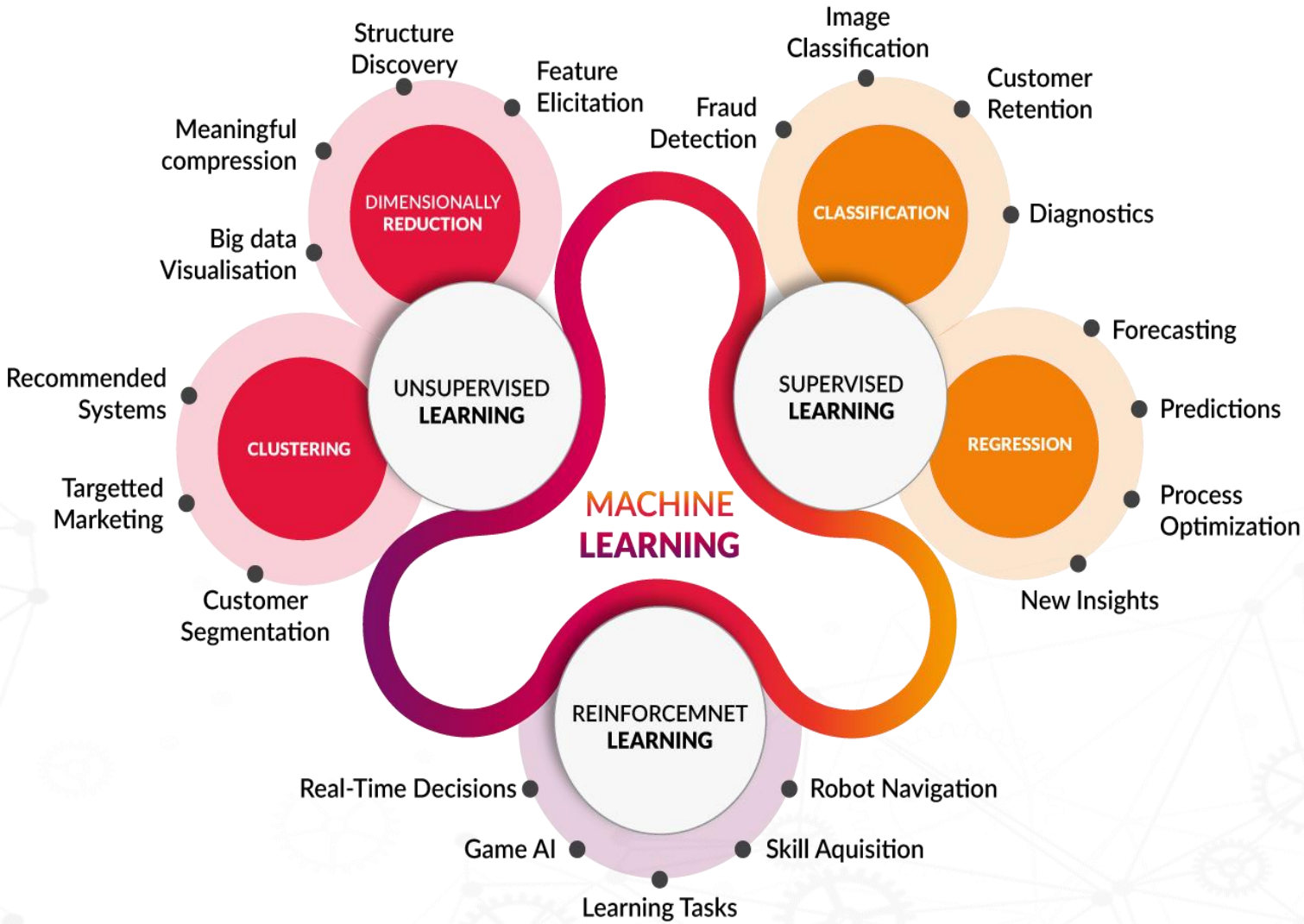
Classification

Preview

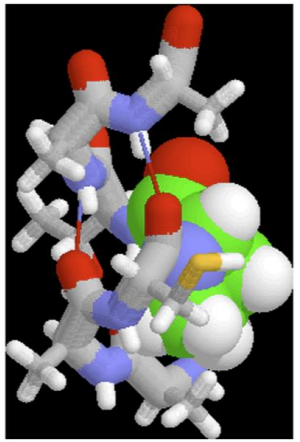

MACHINE LEARNING
UNSUPERVISED LEARNING
SUPERVISED LEARNING
REINFORCEMNET LEARNING
DIMENSIONALLY REDUCTION
CLUSTERING
CLASSIFICATION
REGRESSION
Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
Image Classification
Customer Retention
Fraud Detection
Diagnostics
Recommended Systems
Targetted Marketing
Customer Segmentation
Forecasting
Predictions
Process Optimization
New Insights
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

Classification

종양세포(tumor cells)가 양성인지 음성(악성)인지 판별

신용카드 거래 트랜잭션이 정상인지 사기인지 구분한다.

단백질(protein)의 2차 구조가 alpha-helix인지, beta-sheet인지, random coil인지 분류한다.



신문 기사를 경제, 날씨, 연예, 스포츠 등으로 구분한다.

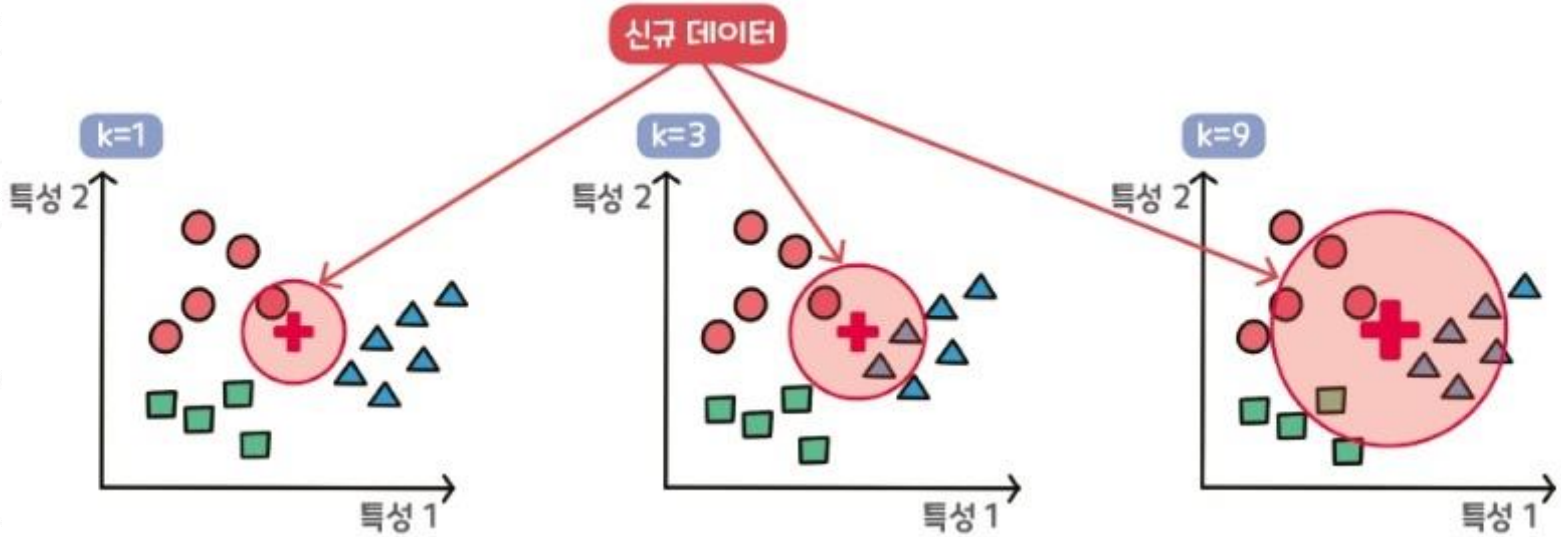
Classification

- Examples
  - K-Nearest Neighbor, KNN
  - Decision Tree
  - Support Vector Machine, SVM
  - Linear Discriminant Analysis
  - Ensemble Model
  - Random Forest
  - Probabilistic Graphical Model
  - Naïve Bayes classifier
  - Multiple Perceptron
  - Deep learning

K-Nearest Neighbor

❖ Principles

- Algorithms to classify which of the existing groups of data (K groups) belongs to when new data comes in
- (Example) When new data is entered when K=1, new data is classified as a red circle, when K=3, and when K=9, it is classified as a blue triangle



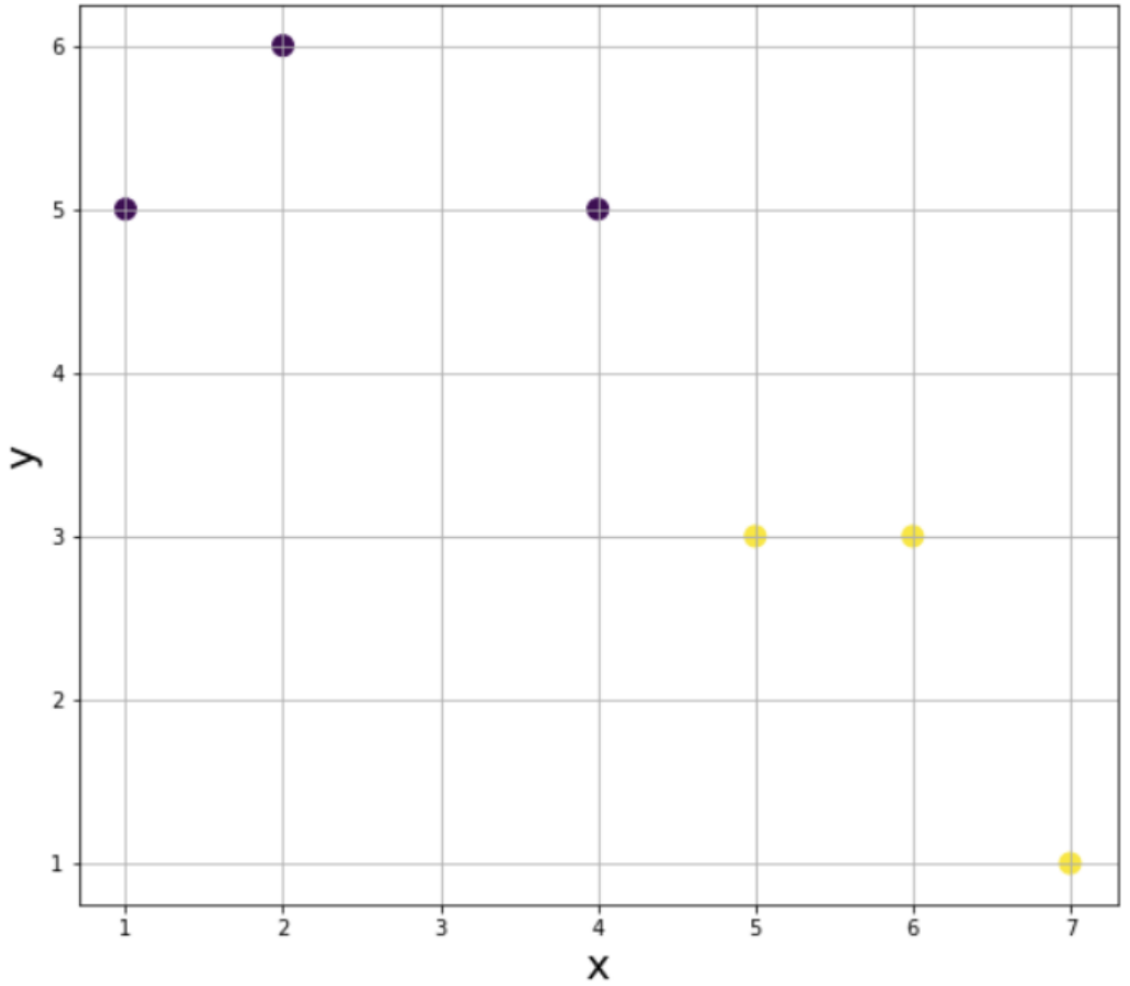
K-Nearest Neighbor

❖ Principles

- KNNs are not significantly affected by the noise present in the learning data and are quite effective when the number of learning data is large
- However, it is unclear which hyperparameters are suitable for analysis, so there is a disadvantage that researchers should randomly select according to each characteristic of the data

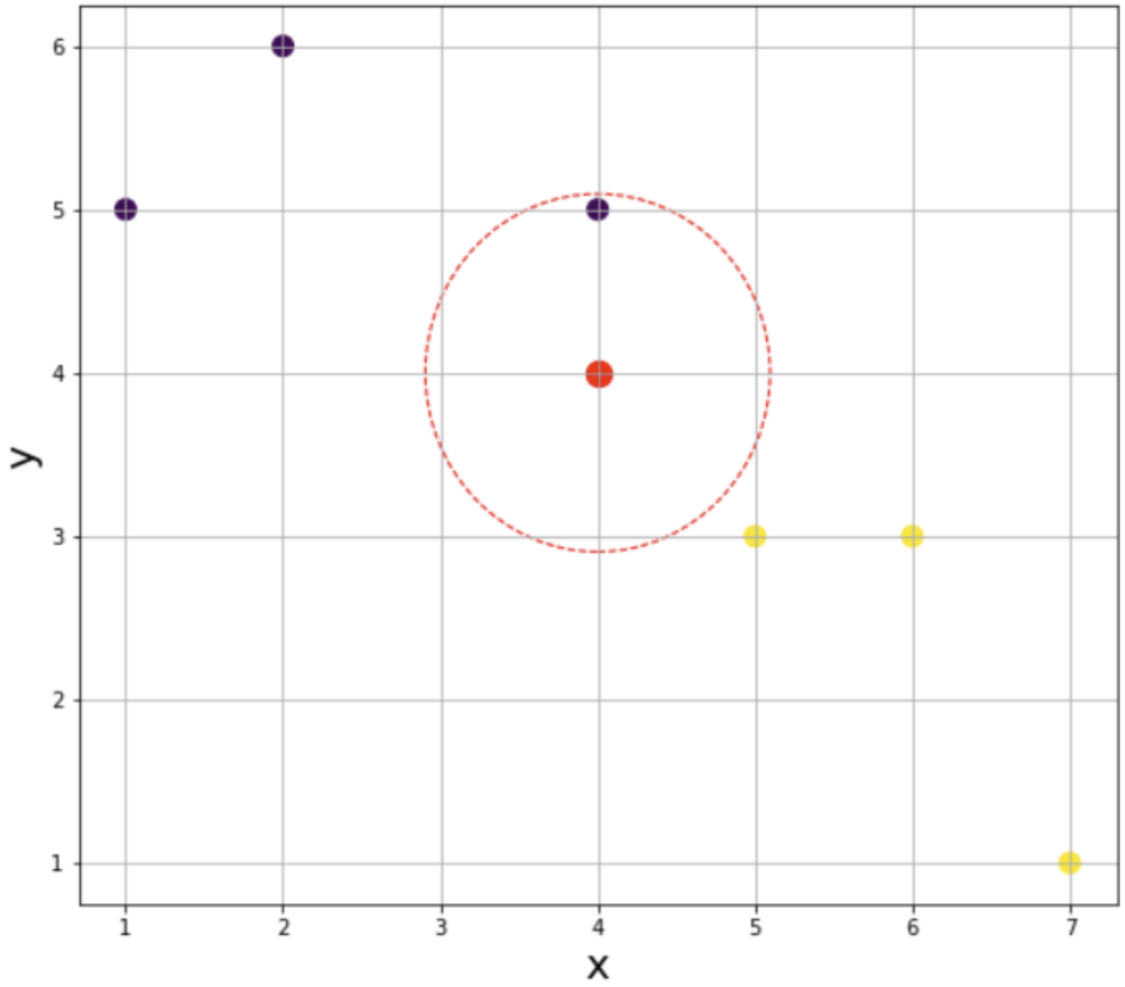
K-Nearest Neighbor

	x	y	label
0	1	5	0
1	2	6	0
2	4	5	0
3	5	3	1
4	6	3	1
5	7	1	1



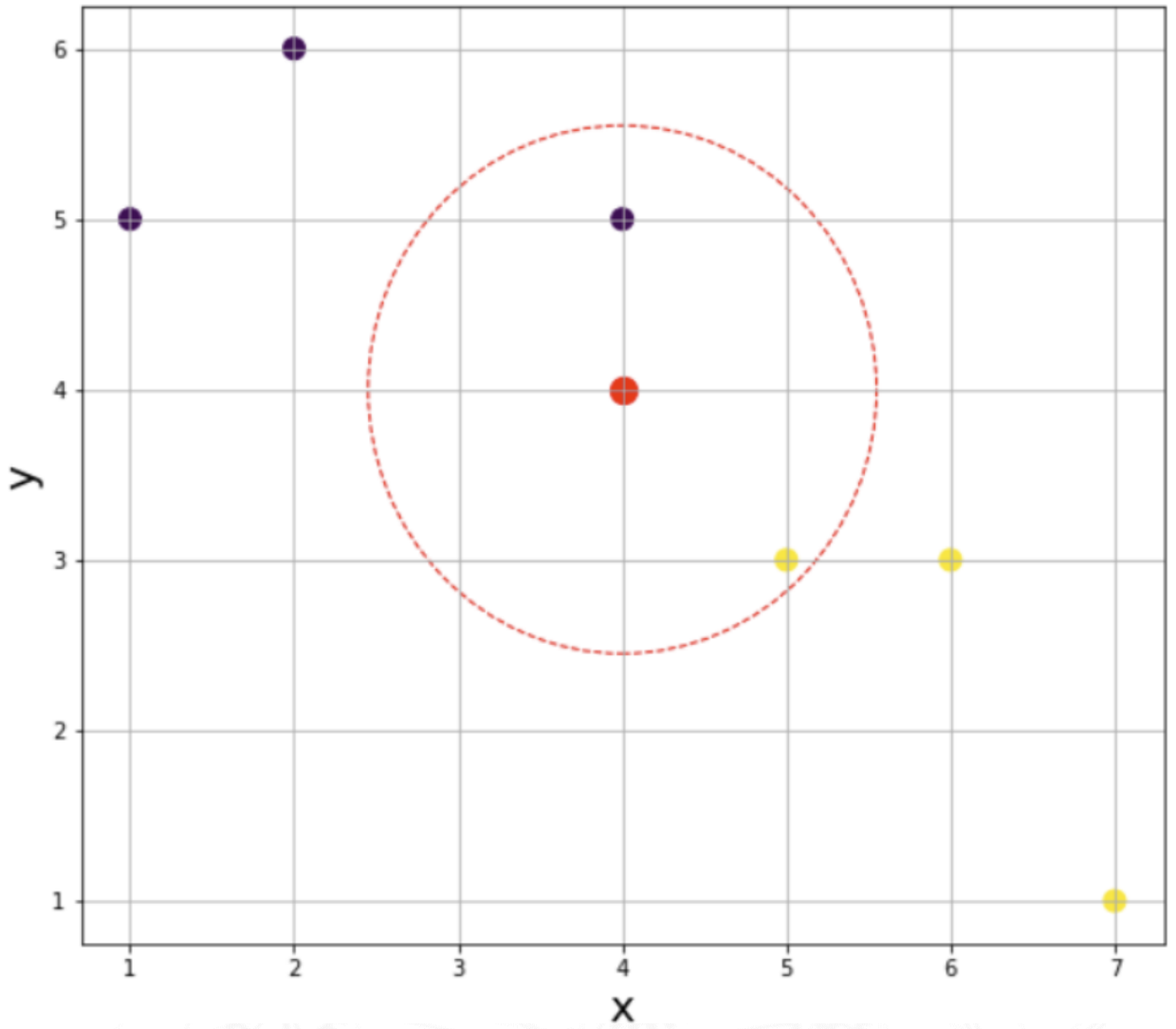
K-Nearest Neighbor

	x	y	label
0	1	5	0
1	2	6	0
2	4	5	0
3	5	3	1
4	6	3	1
5	7	1	1



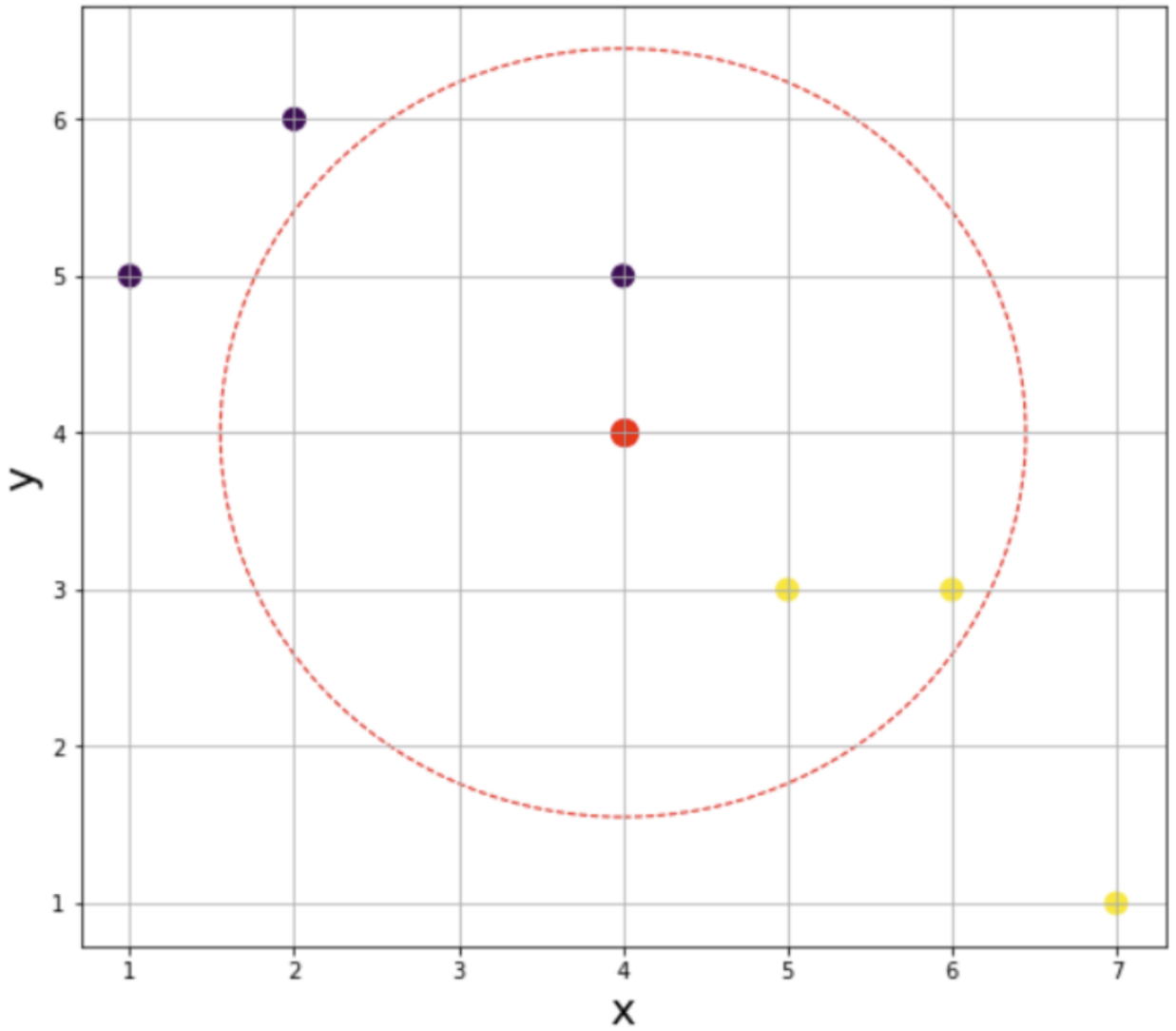
K-Nearest Neighbor

	x	y	label
0	1	5	0
1	2	6	0
2	4	5	0
3	5	3	1
4	6	3	1
5	7	1	1



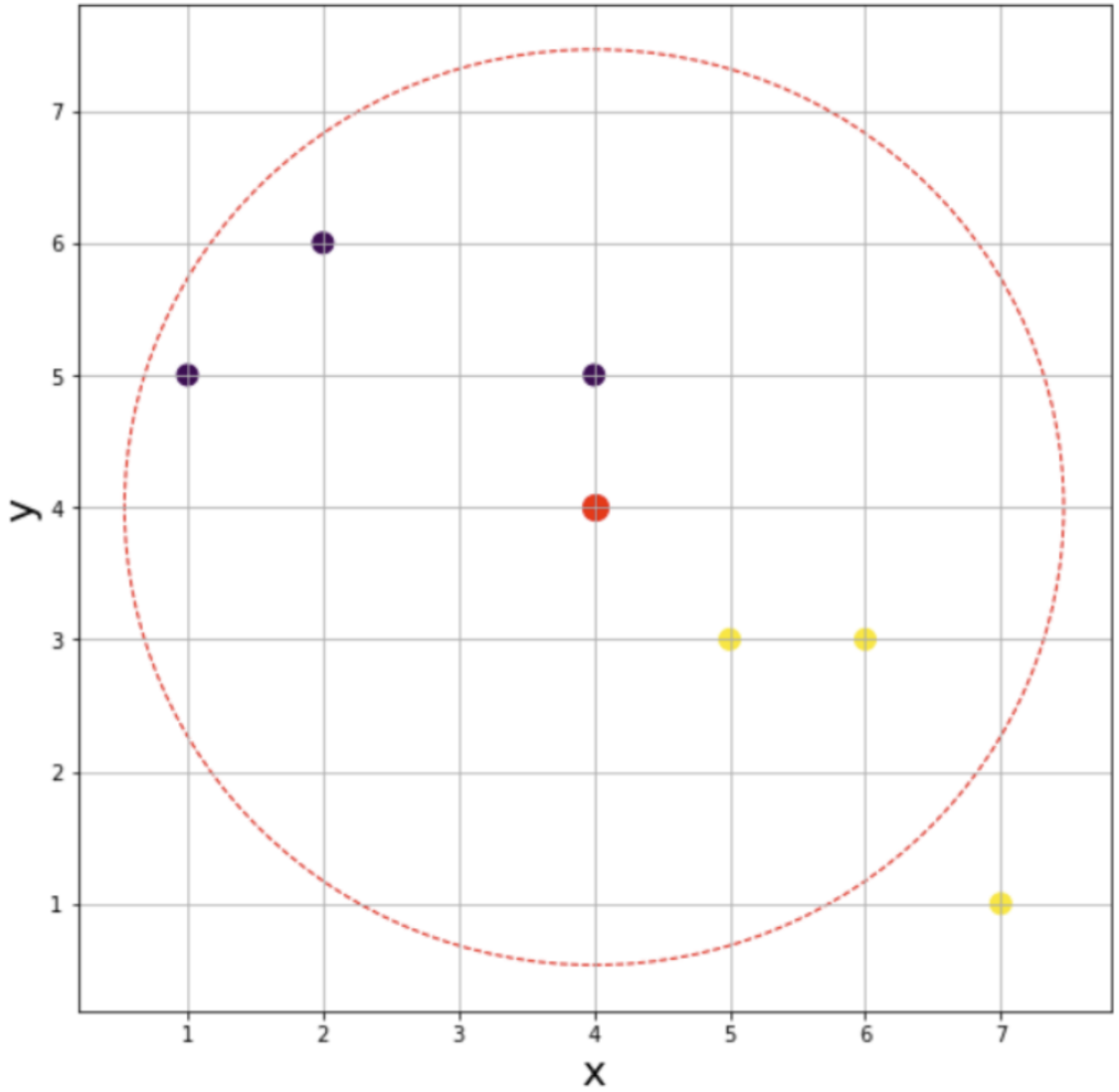
K-Nearest Neighbor

	x	y	label
0	1	5	0
1	2	6	0
2	4	5	0
3	5	3	1
4	6	3	1
5	7	1	1



K-Nearest Neighbor

	x	y	label
0	1	5	0
1	2	6	0
2	4	5	0
3	5	3	1
4	6	3	1
5	7	1	1

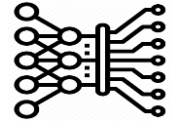


K-Nearest Neighbor

❖ How to Calculate??

- To give a weight that is inversely proportional to the distance (the closer it is to the data, weighted)
- A large number of points (each point's influence is Uniform) by determining only the number (K)
  - 범주형 변수일 경우, K-Nearest Neighbors은 가장 많이 나타나는 범주로 타겟 = Y를 추정하고, 50:50으로 K가 나눠지는 경우가 발생하는 문제를 막기 위해, K는 홀수로 지정을 권장
  - 연속형 변수일 경우, K-Nearest Neighbors은 대표값(ex, 평균)으로 타겟 = Y를 추정되고, 거리에 반비례하는 가중치(inverse distance weighted average)를 사용

K-Nearest Neighbor



❖ Categorical Variable

- Hamming Distance
  - Hamming 거리는 같으면 거리가 0이고 다르면 거리가 1씩 증가

$$HammingDistance = \sum_{j=1}^{J} I(x_j \neq y_j)$$

  - What is distance of this length? “1011111” vs “1001001”

K-Nearest Neighbor: Pseudo-code

❖ Categorical Variable

- 점 (x,y), N개의 Training 관측치 (X_i,Y_i), i = 1,...,N에 대하여, $m = 1, ..., M$
- 좌표상에 (X_1,Y_1) 부터 (X_n,Y_n)까지 있는 데이터를 $(X_{(1)}, Y_{(1)}), ..., (X_{(n)}, Y_{(n)})$
- 거리에 따른 오름차순으로 분류한다. 즉 (X_1,Y_1)이 예측하고자 하는 변수와 가장 가까운 변수 $d(X_{(1)}, Y_{(1)}) \leq ... \leq d(X_{(n)}, Y_{(n)})$
- Y를 예측하기 위해서, K번째까지 즉 (X_k,x)까지를 사용하여, 그 중에 가장 많은 범주를 선택

$$d(X_{(1)}, x) \leq ... \leq d(X_{(n)}, x)$$

- k개 중에서 선택된 범주(m)와 같은 변수 $Y_{(i)}$의 개수가 곧 확률로 정의 $\hat{p}_m = \frac{\sum_{i=1}^{k}(Y_{(i)=m})}{k}$

$$\hat{y} = argmax\, \hat{p}_m \qquad m = 1, ..., M$$

K-Nearest Neighbor

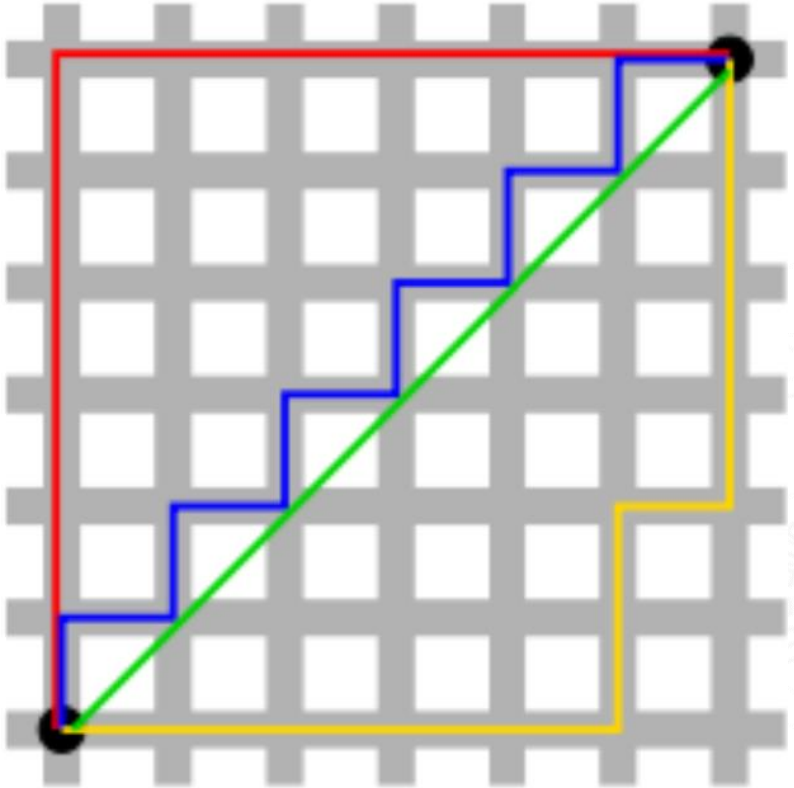
❖ Continuous Variable

- **Euclidian distance**

$$EuclidianDistance = \sqrt{\sum_{j=1}^{J}(x_j - y_j)^2}$$

- **Manhattan Distance**

$$ManhattanDistance = \sum_{j=1}^{J}|x_j - y_j|^2$$



K-Nearest Neighbor: Pseudo-code

❖ Continuous Variable

- 거리에 따른 오름차순으로 분류한다. 즉 (X_1,Y_1)이 예측하고자 하는 변수와 가장 가까운 변수

$$d(X_{(1)}, Y_{(1)}) \leq \ldots \leq d(X_{(n)}, Y_{(n)})$$

- Y의 변수들을 k개만 가져와서, 표본평균 $\hat{y} = \sum_{i=1}^{k} \frac{Y_{(i)}}{k}$

- 가까운 변수에 대해서 가중치를 부여할 경우, 거리의 역수를 곱해주어 가까운 거리일 수록 분자가 커지도록 계산하여 평균

$$\hat{y} = \frac{\sum_{i=1}^{k} \frac{1}{d(X_{(i)},x)} Y_{(i)}}{k}$$

K-Neighbor Nearest

- Need for rebalancing variable ranges
  - min-max normalization
    - Minimum-maximum normalization represents a range of variables from 0% to 100%

$$Z=(X-\min(X))/(\max(X)-\min(X))$$

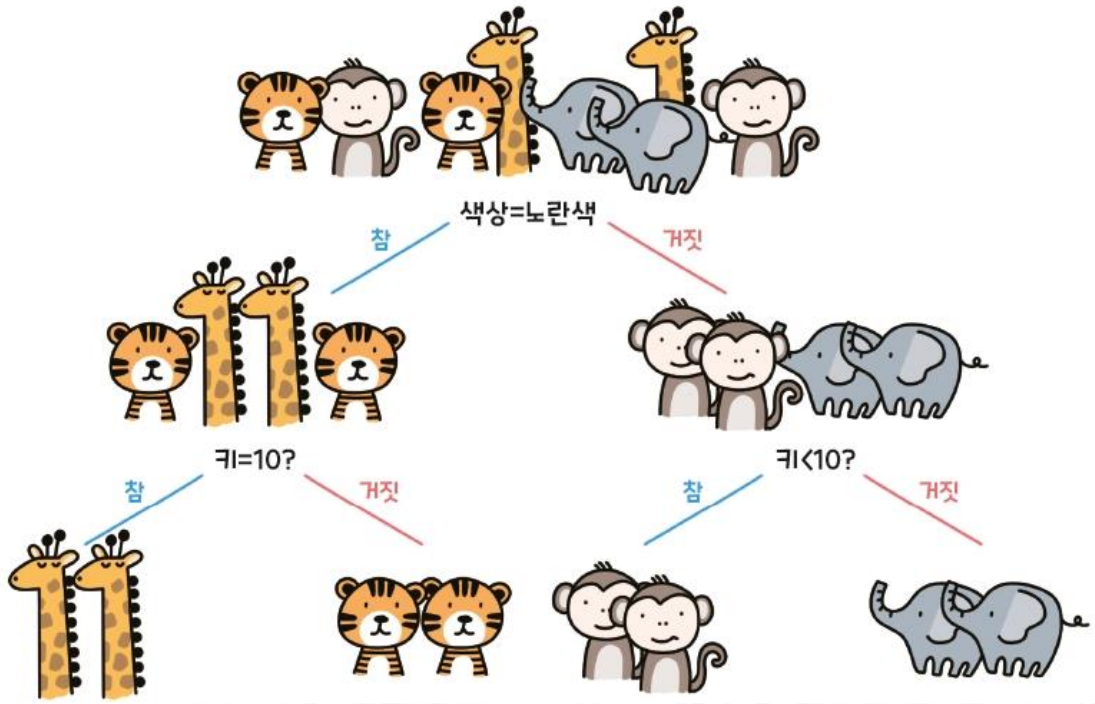
  - z-score standardization
    - The z-score standardization normalizes the range of variables so that the mean is 0 and the coverage deviation is 1

$$Z=(X-\text{mean})/\text{standard deviation}$$

Decision Tree

❖ Definition

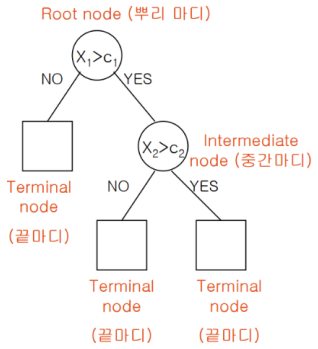
- An analysis method for classifying decision-making rules into tree forms
- It is called 'decision tree' because the method of starting from the upper node and expanding to the lower node according to the classification criteria resembles 'tree'

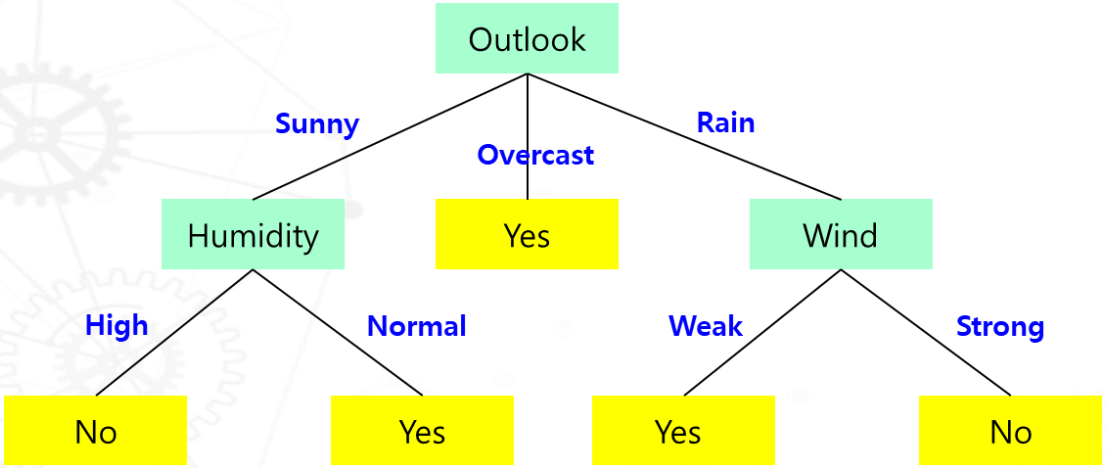


Decision Tree

❖ Representing of decision-making knowledge in the form of a tree

- Internal node : comparison property
- Edge : property value
- Terminal node : class, representative value



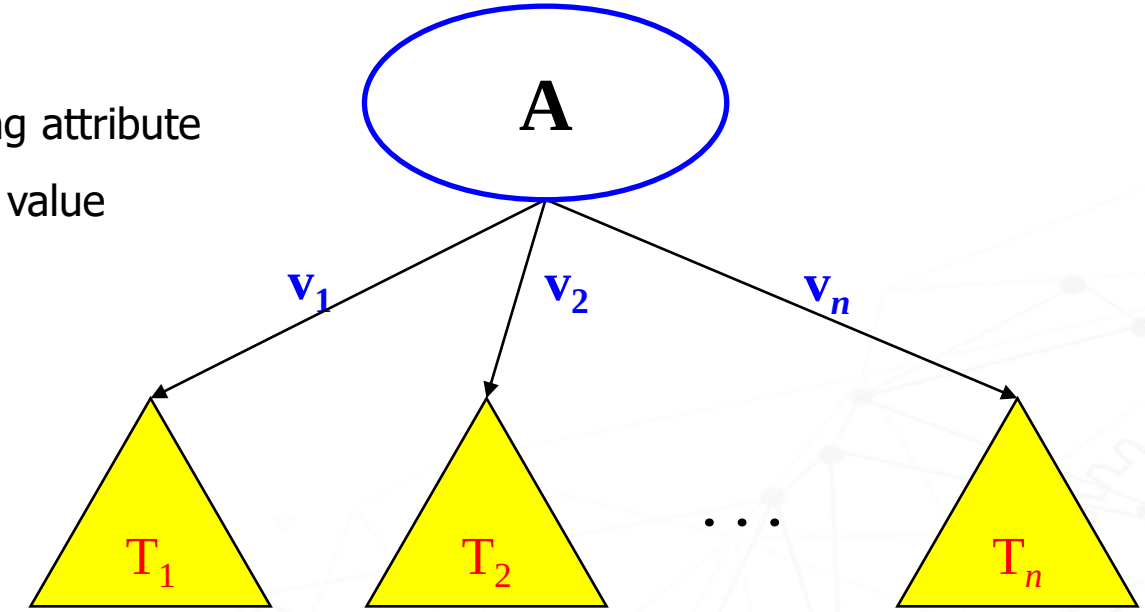


Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

IF Outlook = Sunny **AND** Humidity = High **THEN** Answer = No

Decision Tree

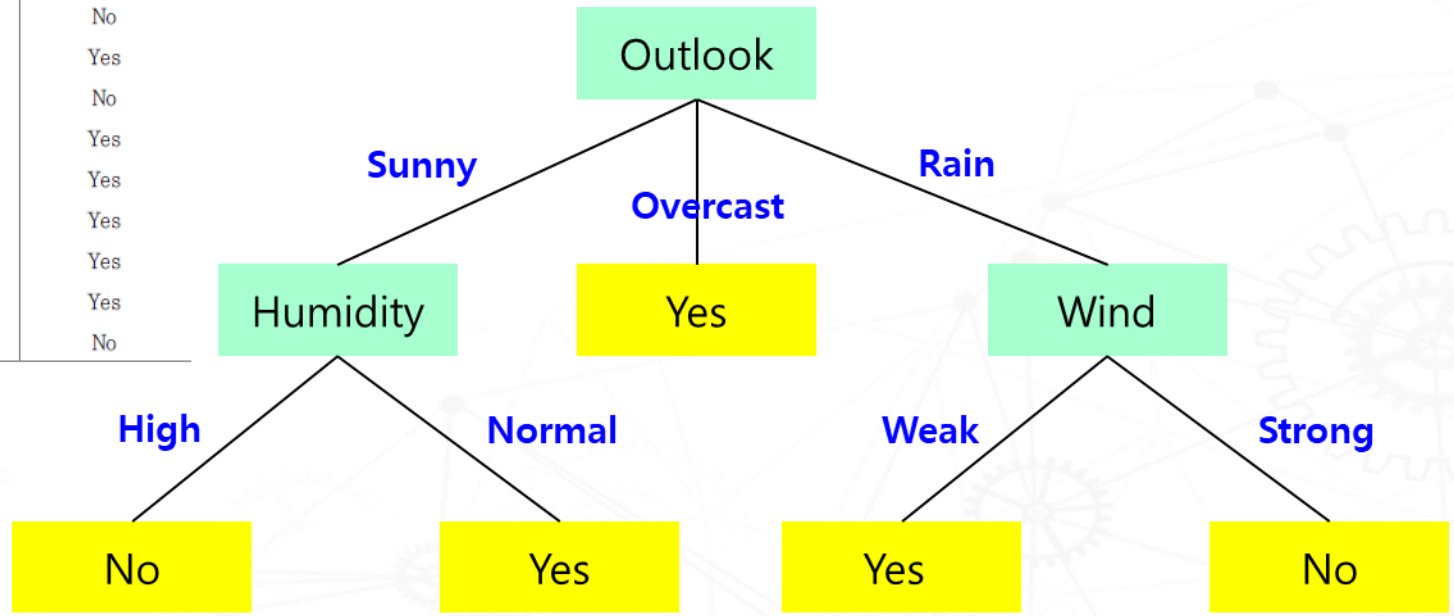
- Start from a tree consisting of one node, including all data
- Repeat the node segmentation process
  1. Select the splitting attribute
  2. Generate the subtree according to splitting attribute
  3. Distribute the data according to attribute value



Decision Tree

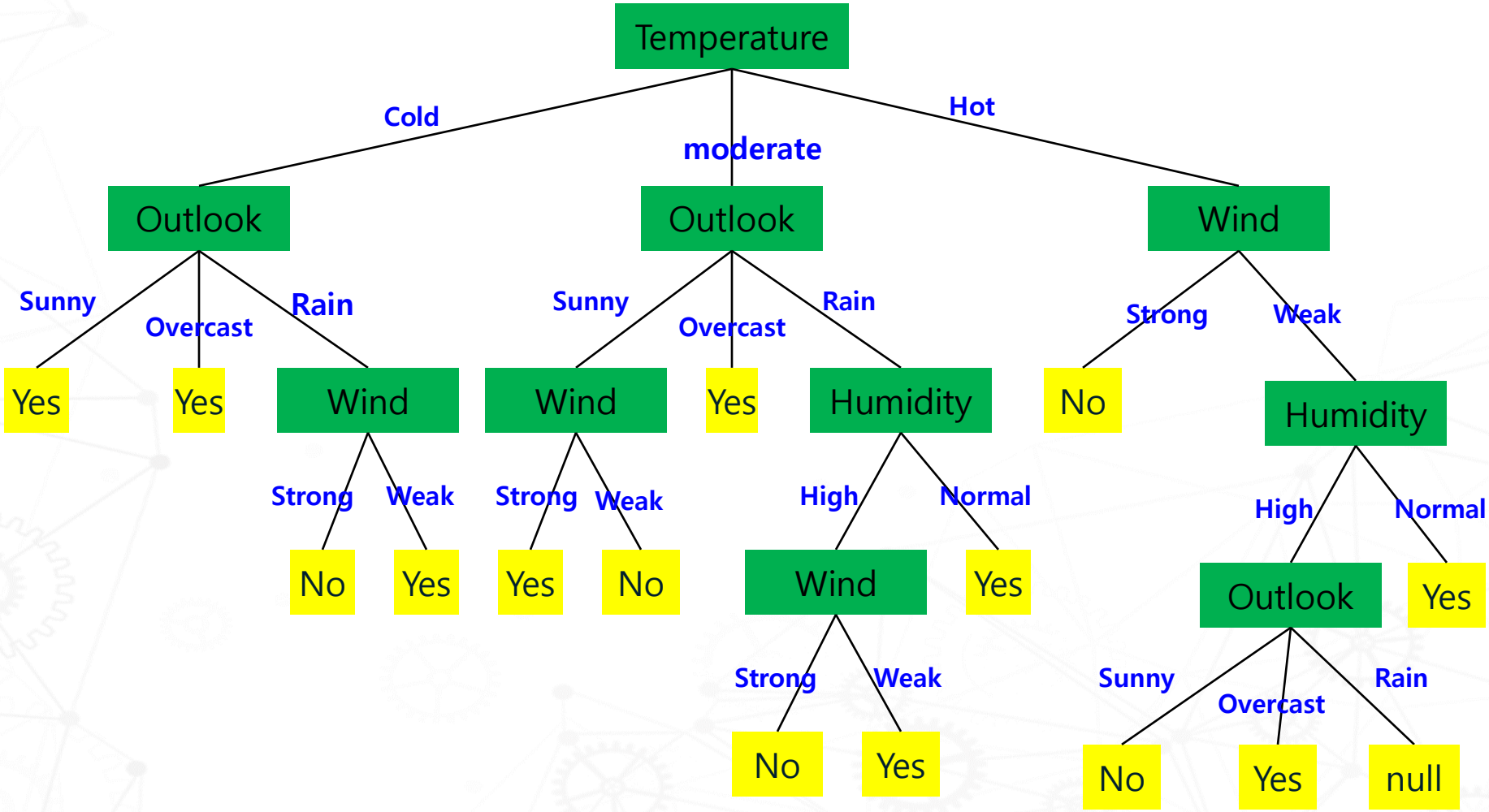
❖ Simple tree

Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No



Decision Tree

❖ **Complex tree**

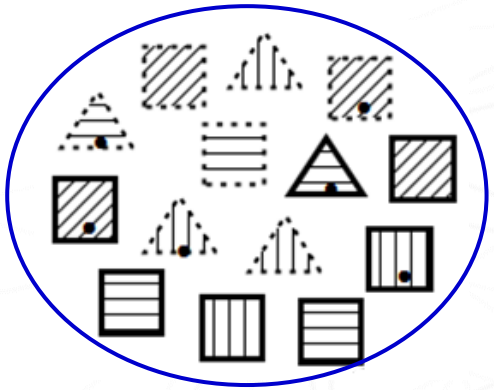


Decision Tree

❖ Example of training data

- Data with class label

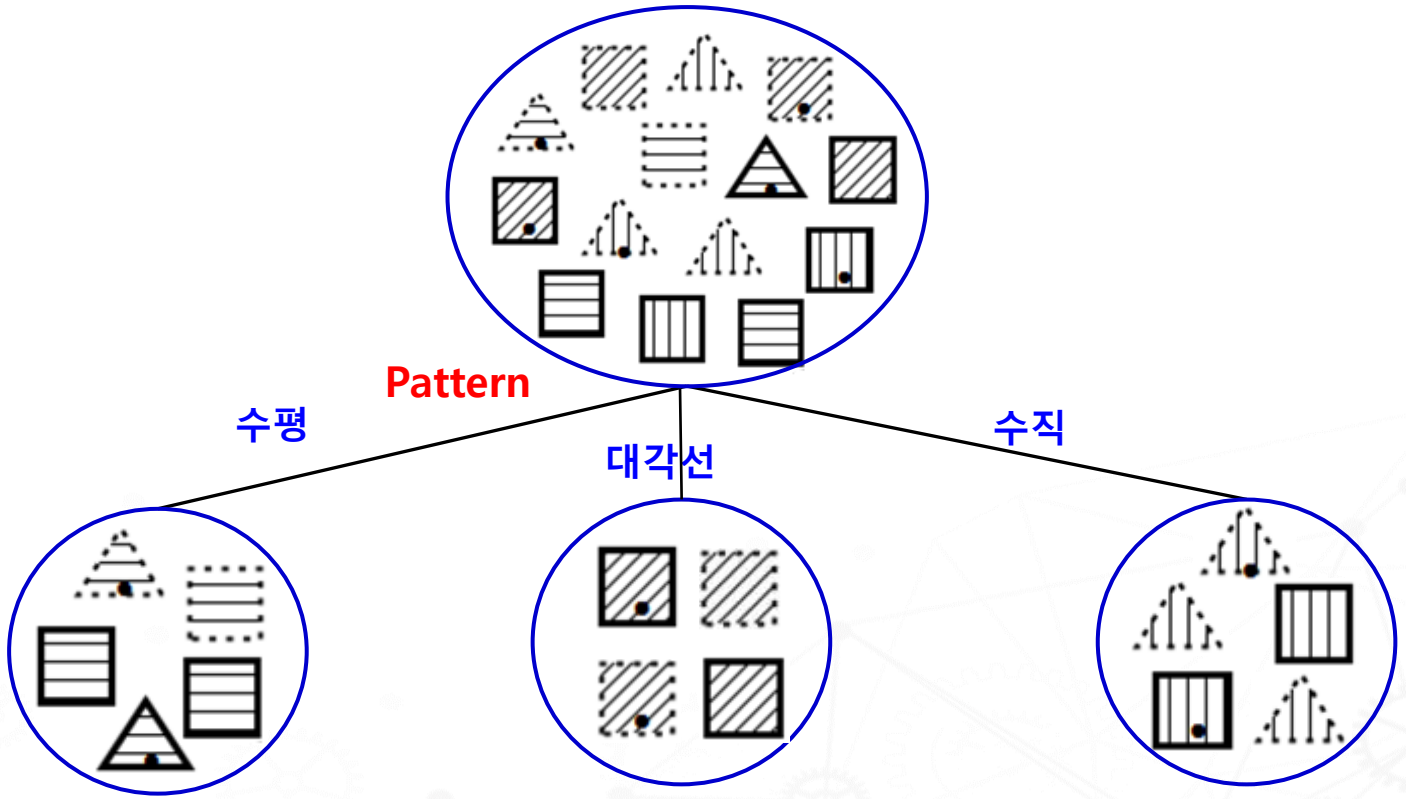
	속성			부류
	Pattern	*Outline*	*Dot*	*Shape*
1	수직	점선	무	삼각형
2	수직	점선	유	삼각형
3	대각선	점선	무	사각형
4	수평	점선	무	사각형
5	수평	실선	무	사각형
6	수평	실선	유	삼각형
7	수직	실선	무	사각형
8	수직	점선	무	삼각형
9	대각선	실선	유	사각형
10	수평	실선	무	사각형
11	수직	실선	유	사각형
12	대각선	점선	유	사각형
13	대각선	실선	무	사각형
14	수평	점선	유	삼각형



Decision Tree

❖ Dataset segmentation and information retrieval

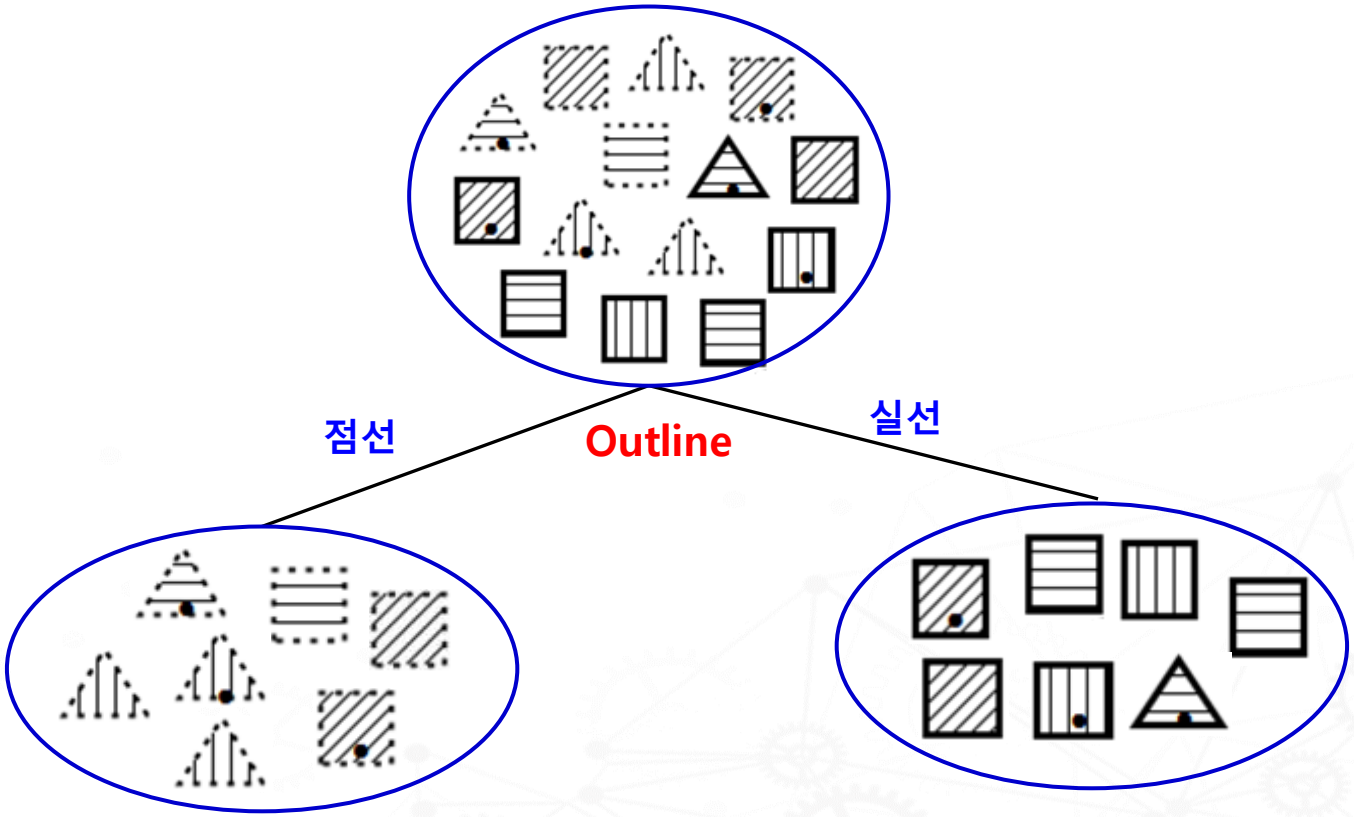
- ‘Pattern’ criteria



Decision Tree

❖ Dataset segmentation and information retrieval

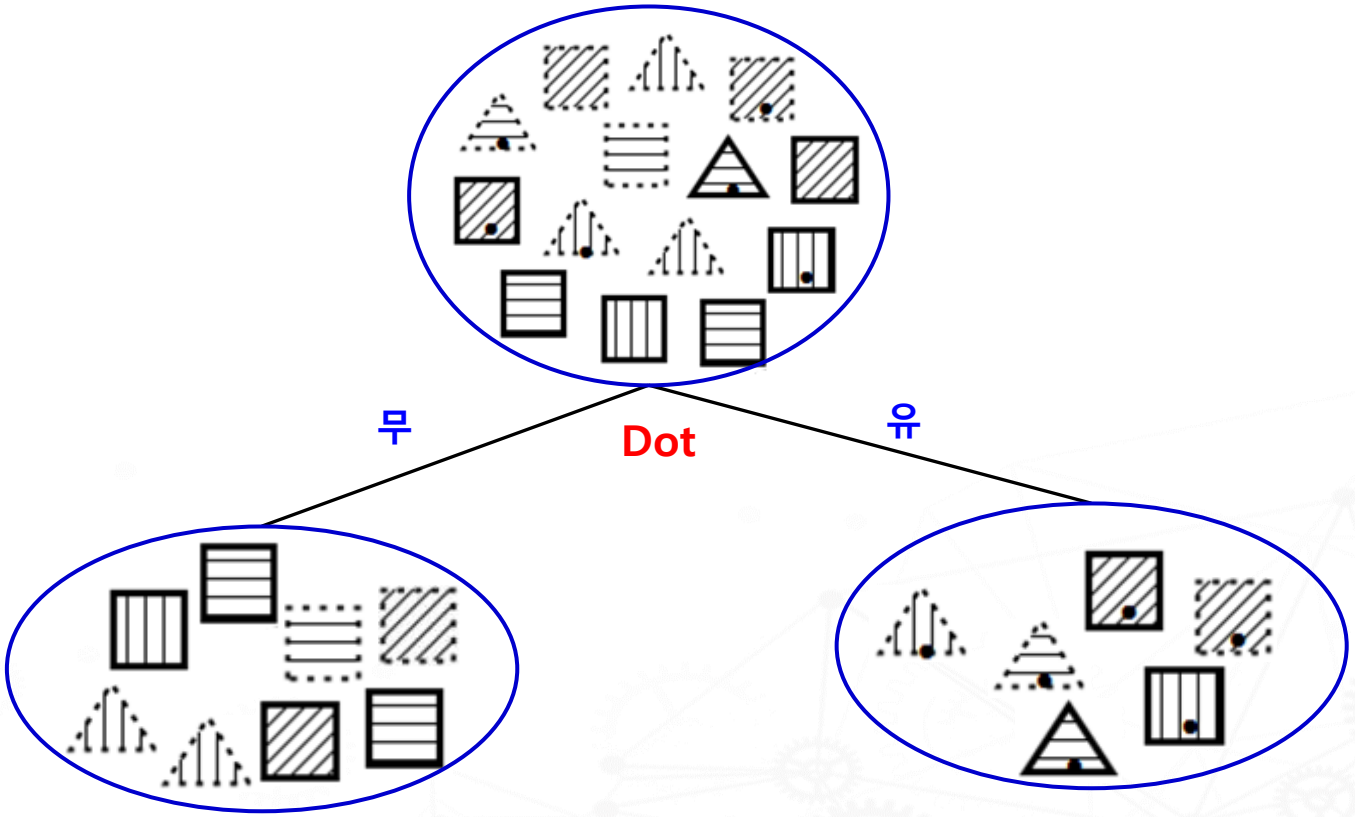
- ‘Outline’ criteria



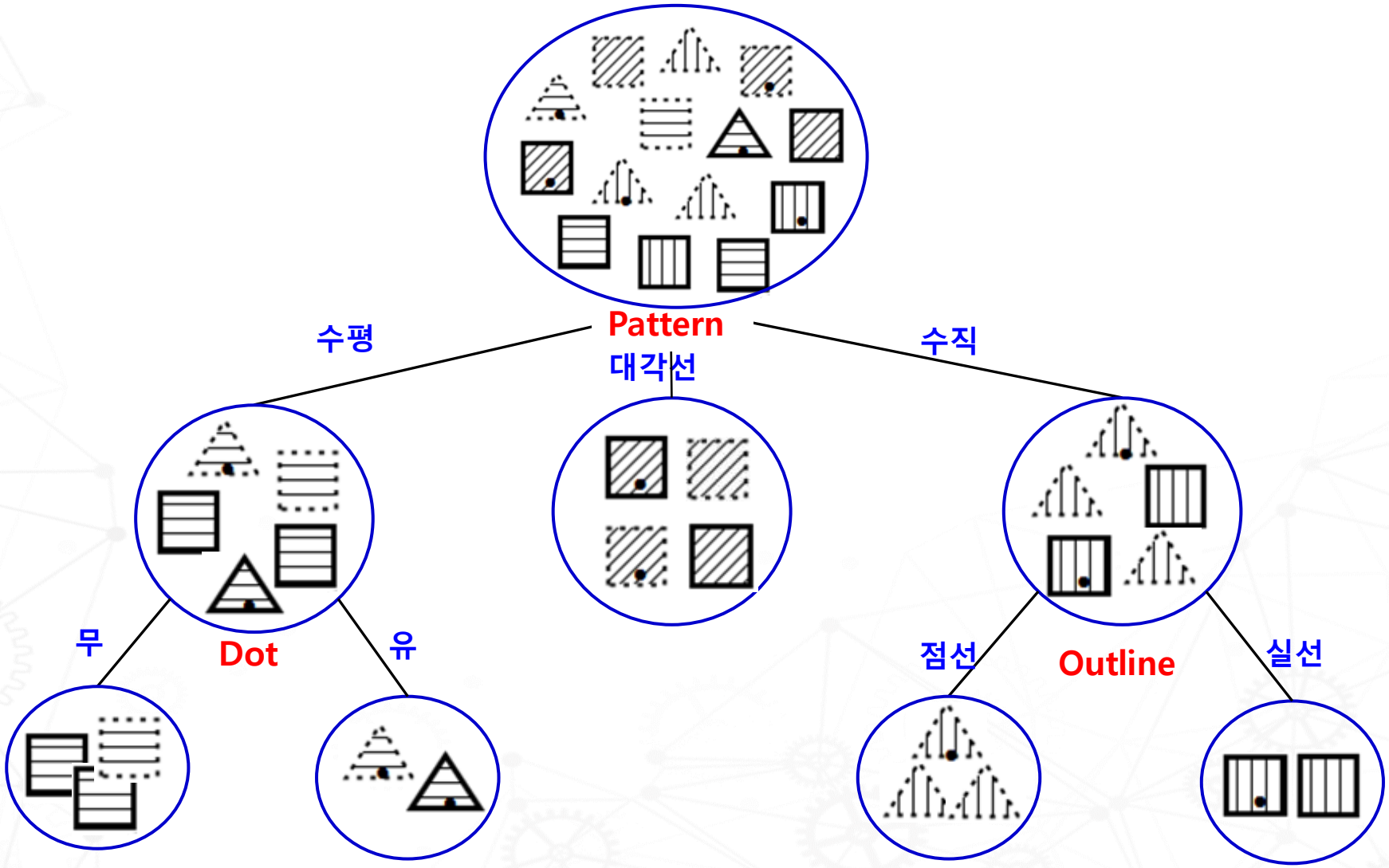
Decision Tree

❖ Dataset segmentation and information retrieval

- ‘Dot’ criteria

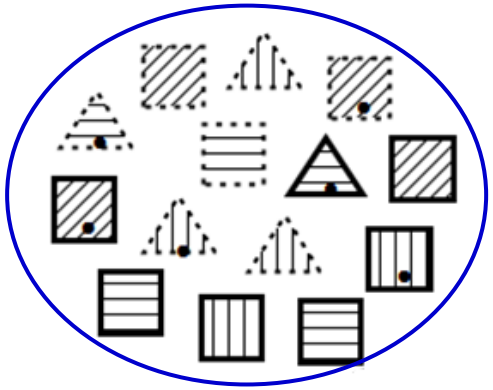


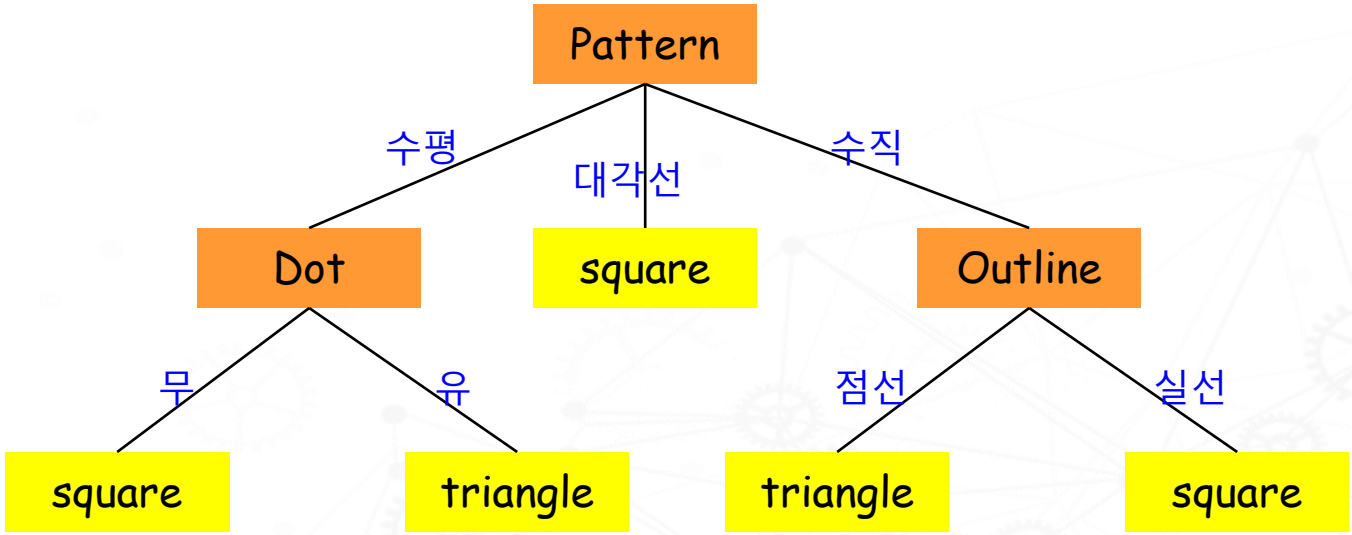
Decision Tree



Decision Tree

❖ **Final decision tree**

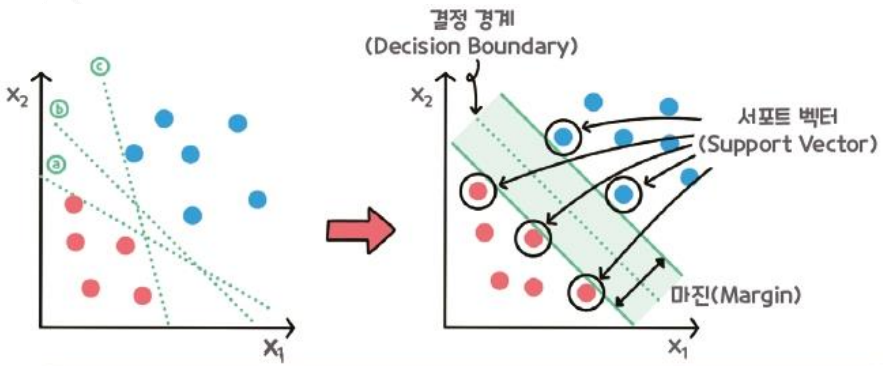




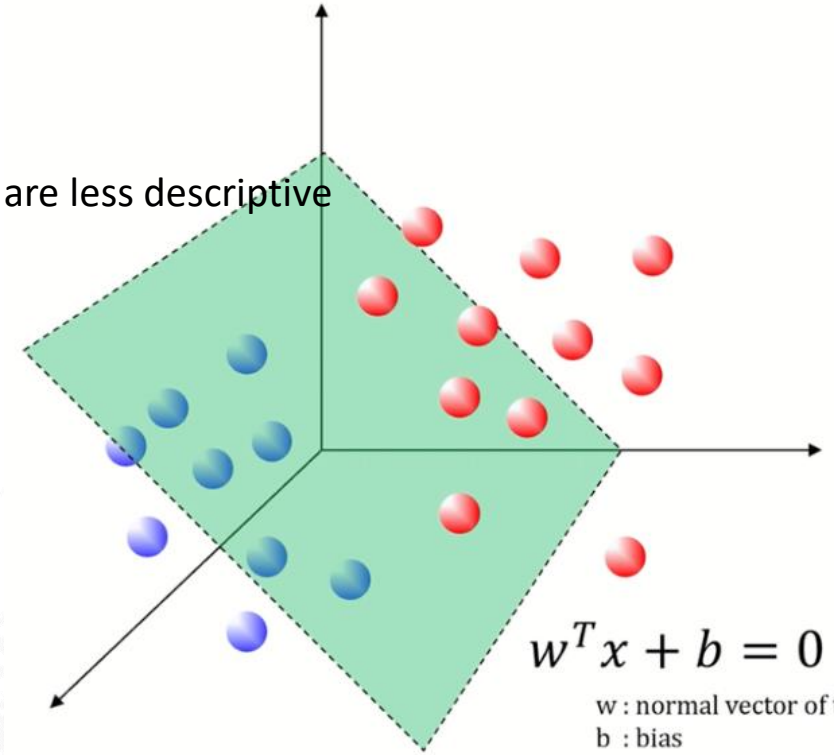
Support Vector Machine

❖ Definition

- Categorize data in the direction of maximizing margin, which means margin between two categories
- SVMs find and classify lines that maximize margins, so larger margins are more likely to be classified even if new data comes in
- SVM is easy to use and highly predictive
  - However, it takes time to build a model and the results are less descriptive



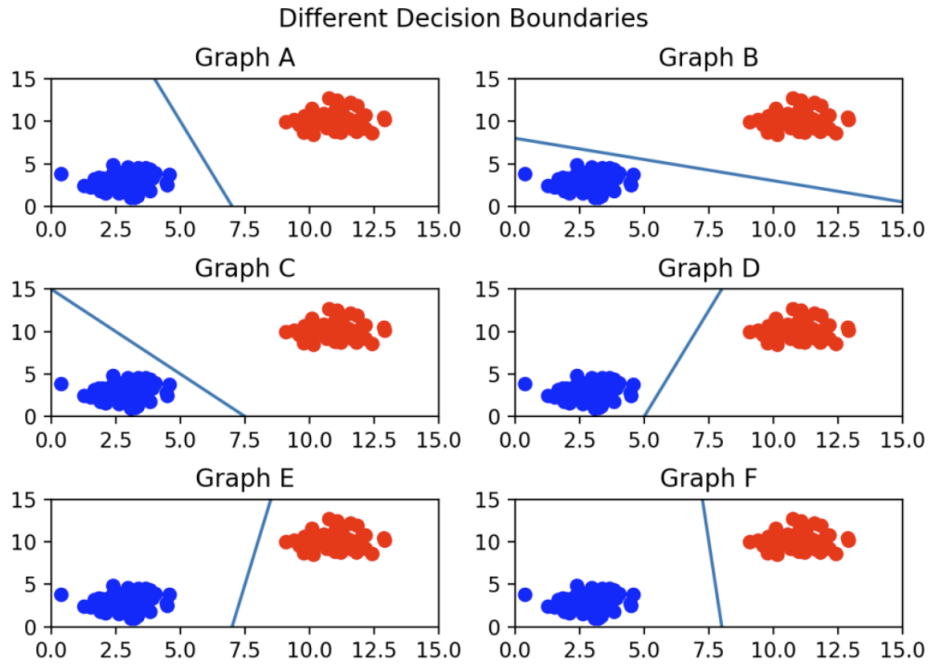
· 결정 경계(Decision Boundary) : 분류를 위한 기준선
· 서포트 벡터(Support Vector) : 결정 경계와 가장 가까운 위치에 있는 데이터
· 마진(Margin) : 결정 경계와 서포트 벡터 사이의 거리

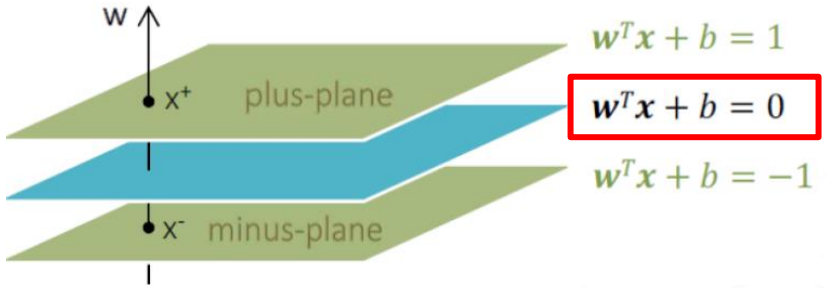


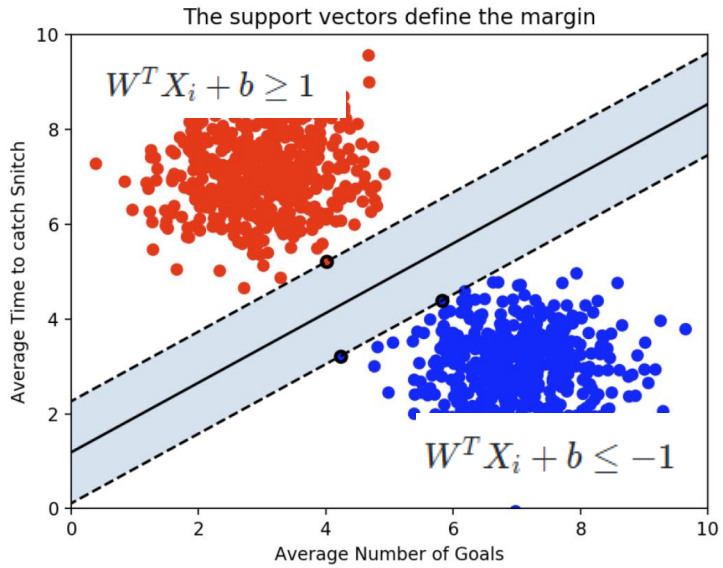
Support Vector Machine

❖ Decision boundary

- SVMs find and classify lines that maximize margins, so larger margins are more likely to be classified even if new data comes in
- The optimal decision boundary maximizes the margin

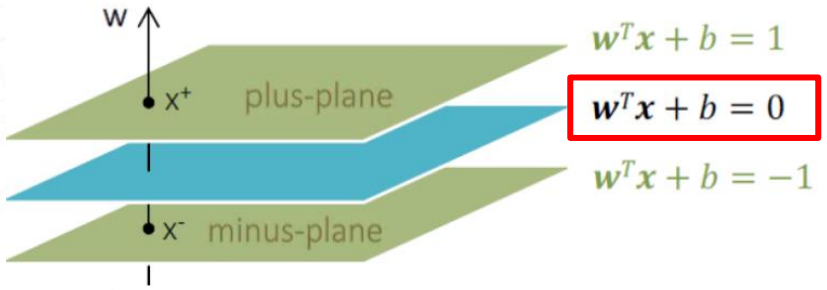


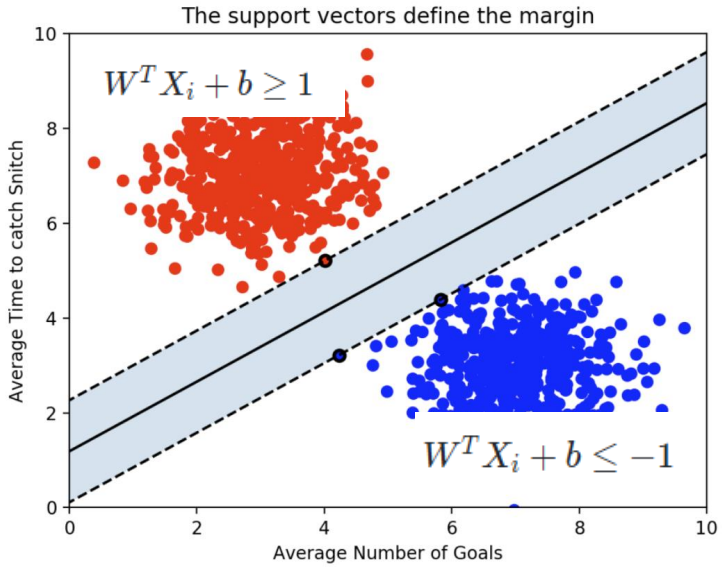




Support Vector Machine

❖ Decision boundary





$$x^+ = x^- + \lambda w$$

$$w^T x^+ + b = 1$$

$$\Rightarrow w^T(x^- + \lambda w) + b = 1$$

$$\Rightarrow w^T x^- + b + \lambda w^T w = 1$$

$$w^T x^- + b = -1$$

$$\Rightarrow -1 + \lambda w^T w = 1$$

$$\lambda = \frac{2}{w^T w}$$

$$Margin$$

$$= distance(x^+, x^-)$$

$$= ||x^+ - x^-||_2$$

$$= ||x^- + \lambda w - x^-||_2$$

$$= ||\lambda w||_2$$

$$= \lambda\sqrt{w^T w}$$

$$= \frac{2}{\sqrt{w^T w}}$$

$$= \frac{2}{||w||_2}$$

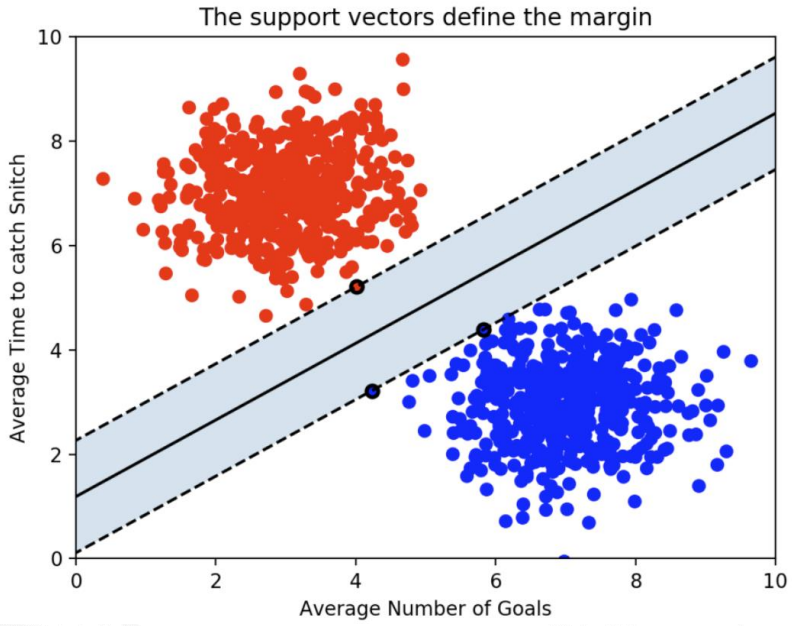
$$\text{max margin} = \max \frac{2}{||w||_2} \Rightarrow \min \frac{||w||_2}{2}$$

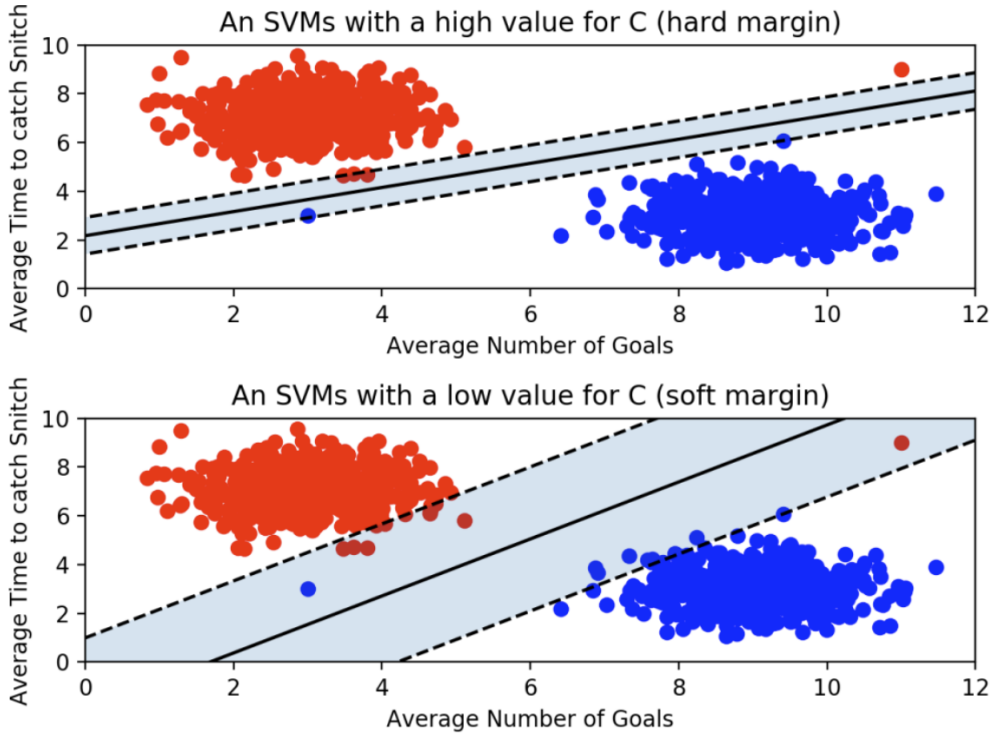
$$max_a min_{w,b} L(w, b, \alpha) = \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

Support Vector Machine

❖ Support vector

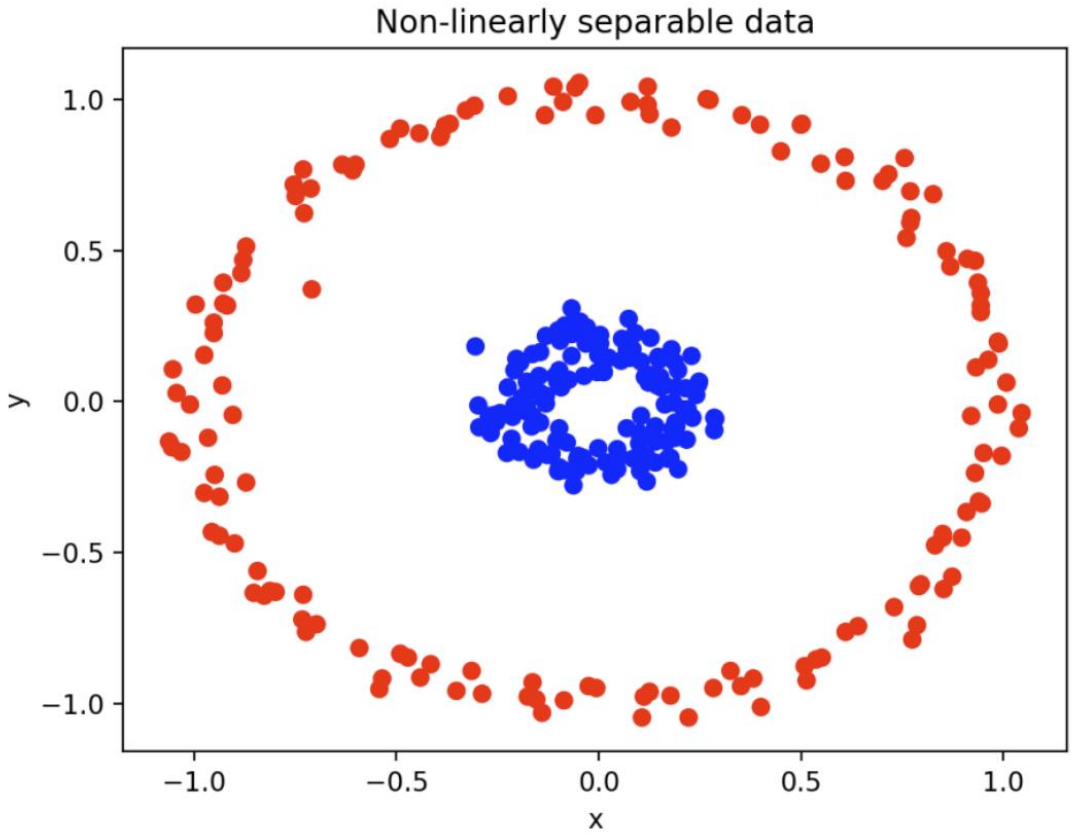
- In SVM, defining the decision boundary is ultimately a support vector, so if you select a support vector well among the data points, you can ignore the remaining numerous useless data points





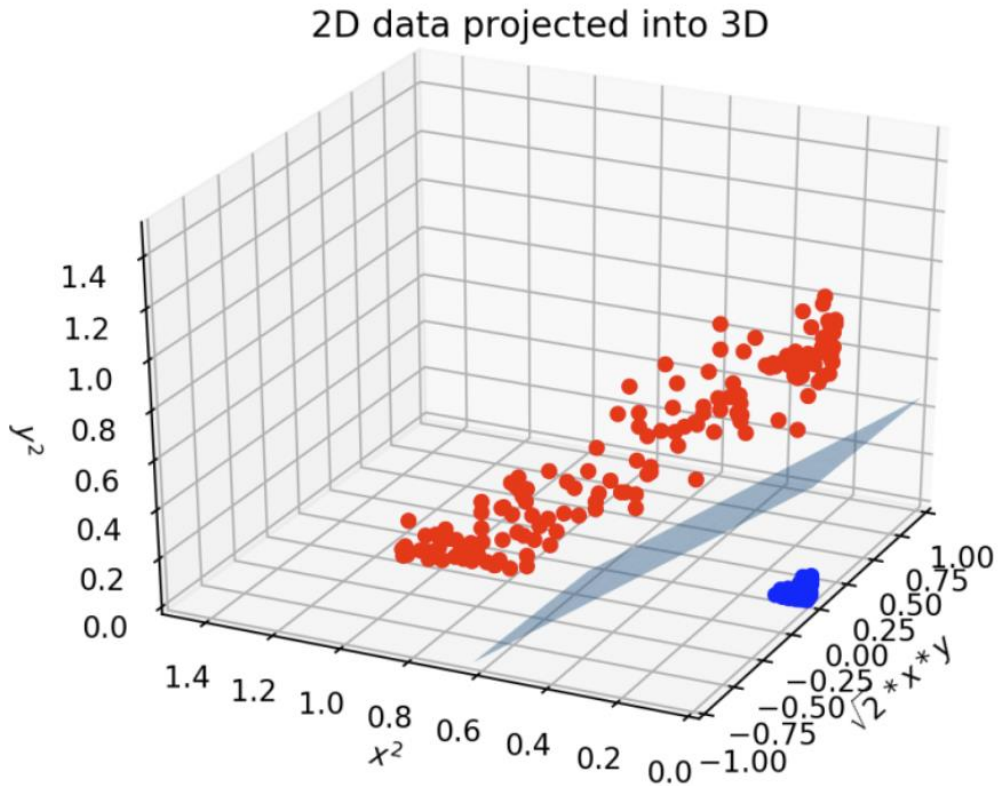
Support Vector Machine

- Kernel
  - What about non-linear problem



Support Vector Machine

- Kernel
  - Non-linear problem using polynomial kernel



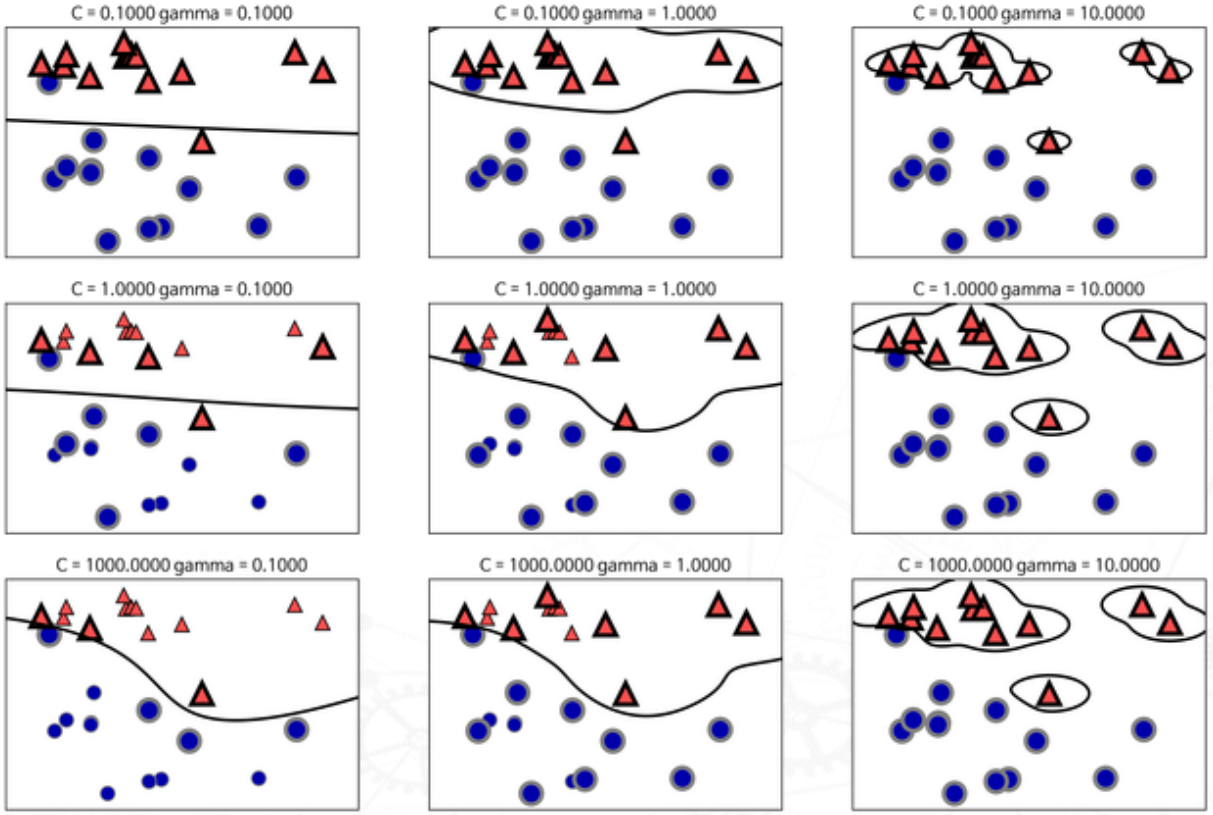
$$(x,\ y) \rightarrow (\sqrt{2} \cdot x \cdot y,\ x^2,\ y^2)$$

$$(1,\ 2) \rightarrow (2\sqrt{2},\ 1,\ 4)$$

Support Vector Machine

❖ Gamma(γ)

- Defines how far the influence of a single training point reaches
  - 값이 커짐에 따라 c는 두 데이터를 정확히 구분하는 것에 초점
  - Gamma는 개별 데이터마다 decision boundary를 만드는 것에 초점



Linear Classifier

❖ **Linear Discriminant Analysis, LDA**

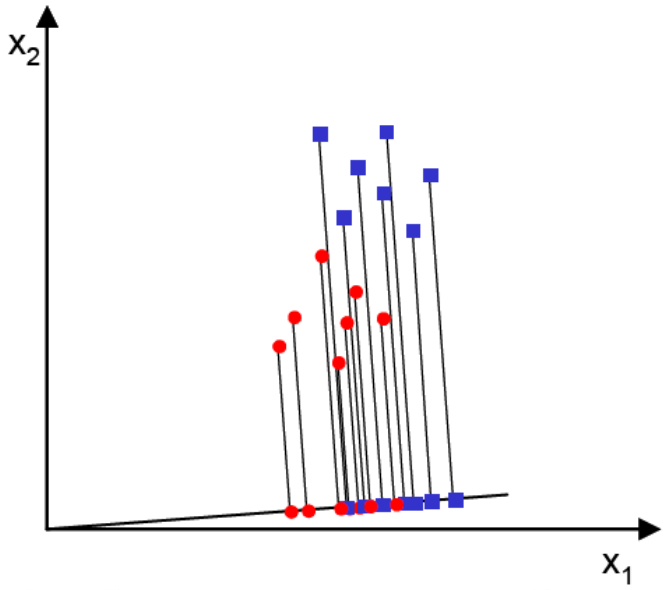
- Principal Component Analysis (PCA) is a method of reducing data from the perspective of optimal representation of data,
- LDA is a method of reducing data from the perspective of optimal classification of data,
  - i.e., reducing dimensions to linear subspaces by thinking of it as a main axis that maximizes class separation on feature space
- Reducing dimensions while maintaining the classification information between classes as much as possible
  - Given the D-dimensional sample dataset X={x(1, x(2,..., x(N}), if N1 belongs to class ω1 and N2 belongs to class ω2, to obtain scalar y by projecting x along any line
  - Maximizing the separation of these scalar values among all possible lines

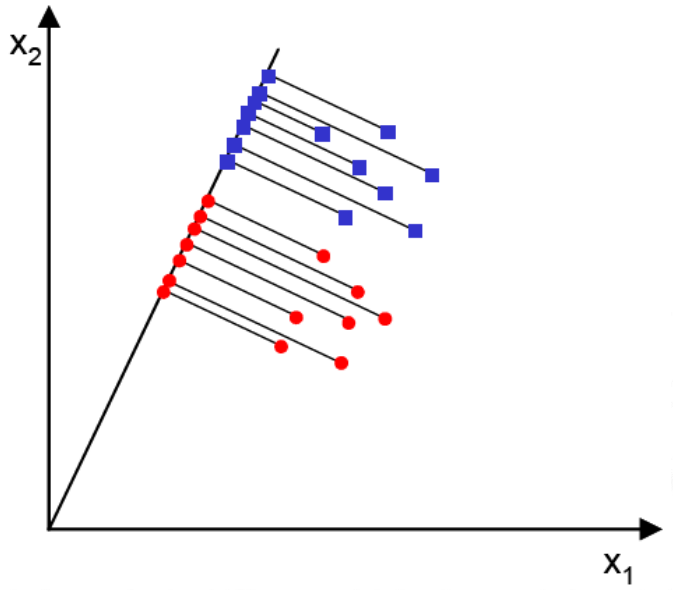
$$y = W^T x$$

$W : D \times 1$ 행렬
$x : D \times 1$ 차원 데이터
y : 1차원의 스칼라 값

Linear Discriminant Analysis

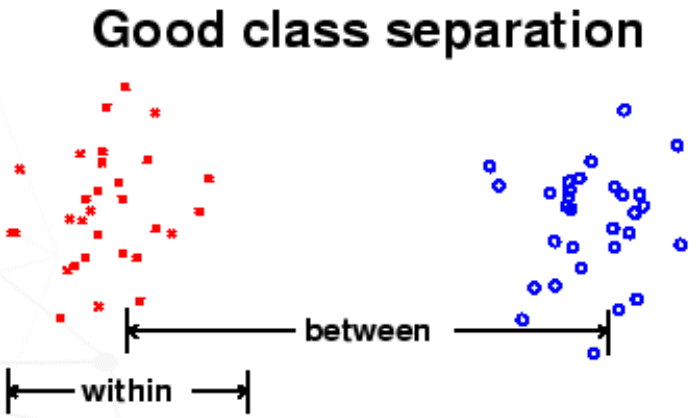
- Maximizing the separation of these scalar values among all possible lines



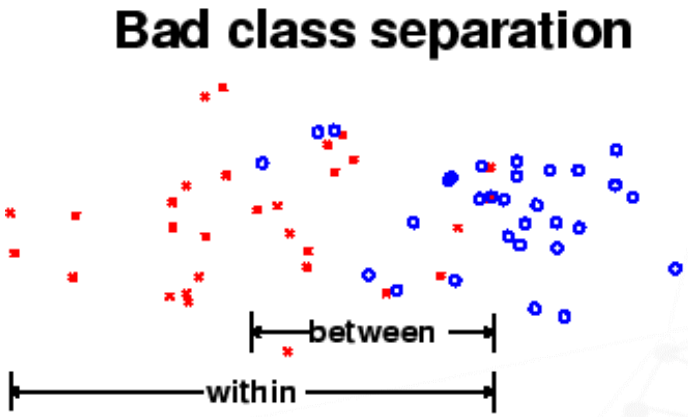


Linear Discriminant Analysis

❖ Reduce the dimension of feature vectors for data by maximizing the ratio of between-class scatter and within-class scatter



판별하기 용이한 분포



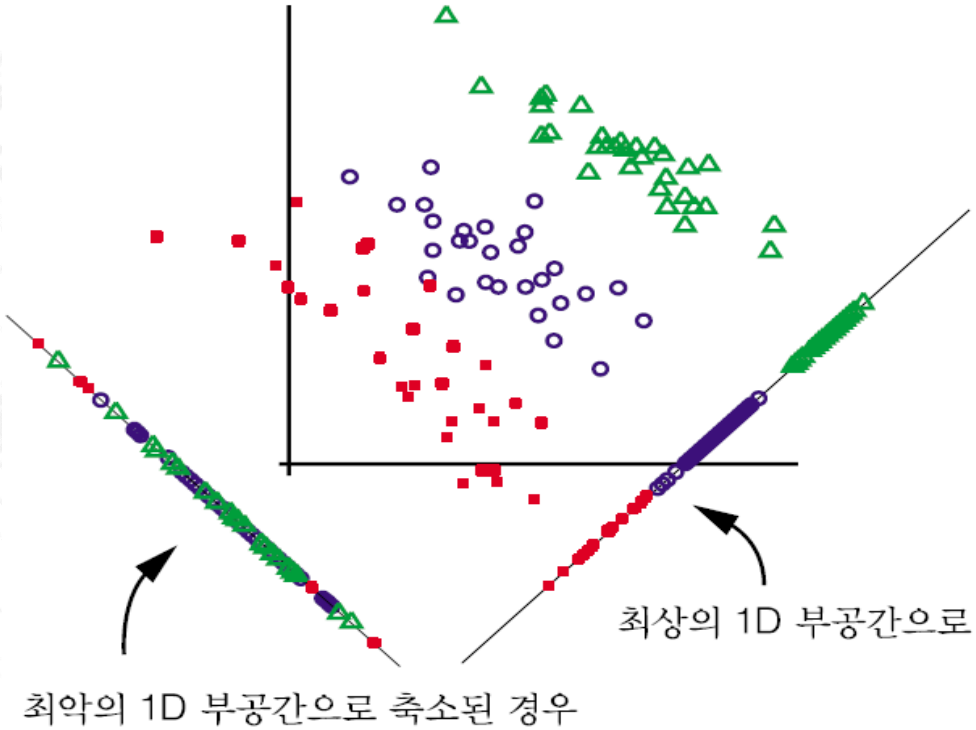
판별하기 어려운 분포

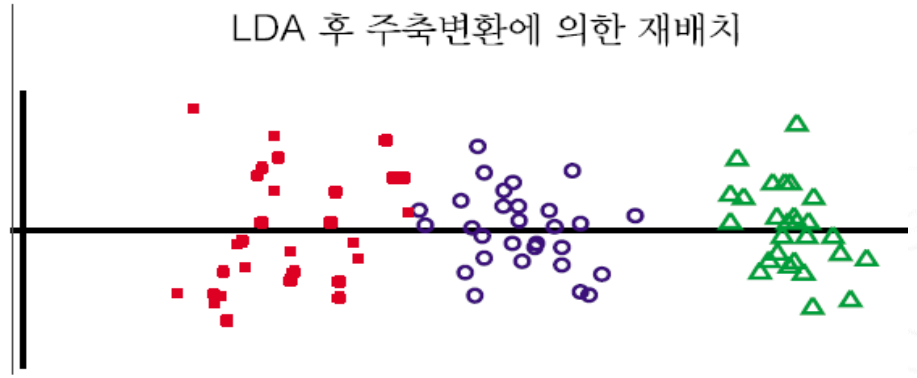
클래스간 분산 (between-class Scatter)

크다 ← ---------- → 작다

클래스내 분산 (within-class Scatter)

Linear Discriminant Analysis



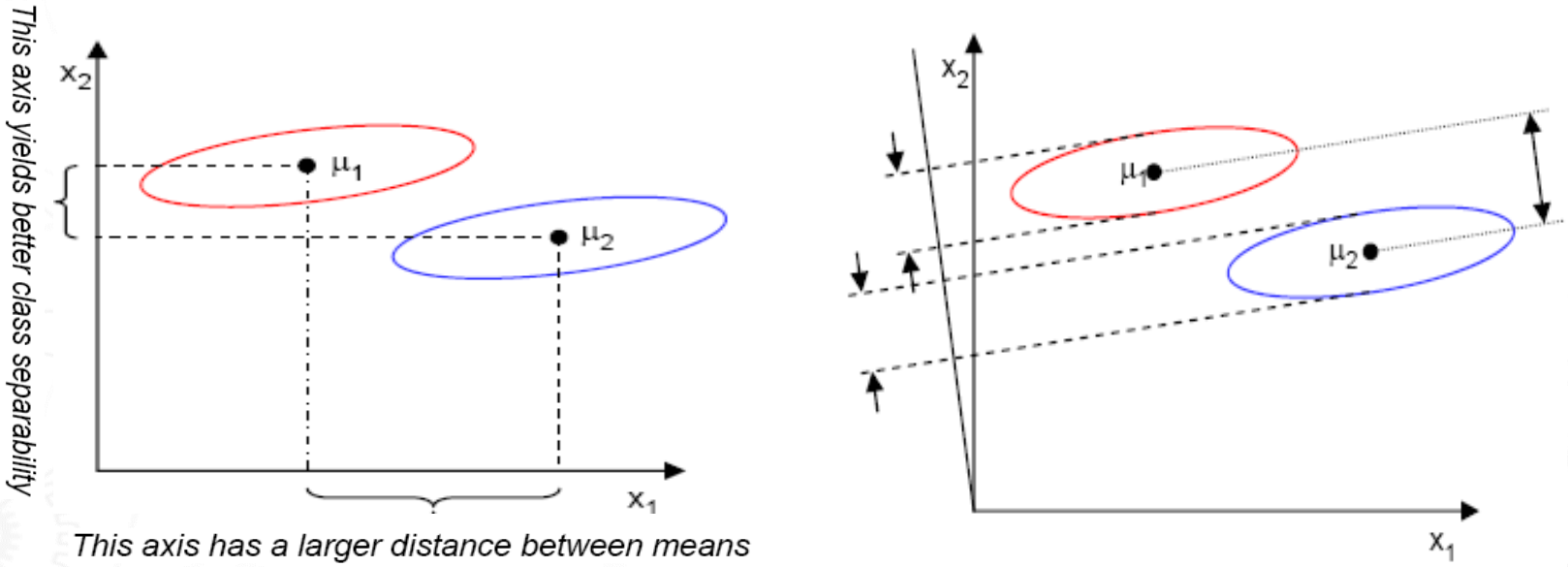


Linear Discriminant Analysis

어느 사영을 취하는 것이 좋을 것인가? ➔ 좋은 사영을 찾기 위해서는 사영들 간의 분리 정도를 측정할 수 있어야 한다.

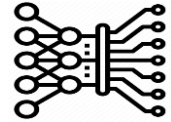
선형변환에 의한 사영 ➔ $y = w^T x$

where $w : D \times 1,\ x : D \times 1$



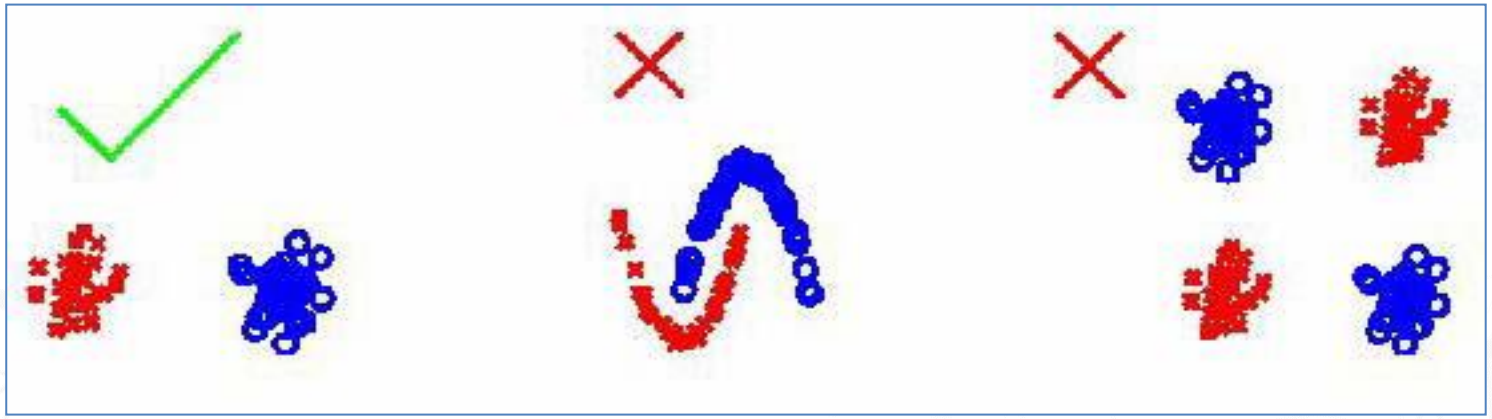
Fisher 에 의해서 제안된 방법은 **클래스내(within-class)의 스캐터로 정규화한 평균들 간의 차이**로 표현된 **함수를 최대화**시키는 것이다.

Linear Discriminant Analysis



- Class-dependent transformation
  - To maximize the ratio of variance between classes to intra-class variance
    - $\frac{\text{각 클래스간 분산}}{\text{클래스내 분산}}$
- Class-independent transformation
  - To maximize the ratio of total to intra-class variance
    - $\frac{\text{전체 분산}}{\text{클래스내 분산}}$
- Choosing an LDA Approach
  - Depending on the purpose of the data set and classification problem
  - For good classification purposes: class-dependent transformation
  - When generalization is important: class-Independent transformation

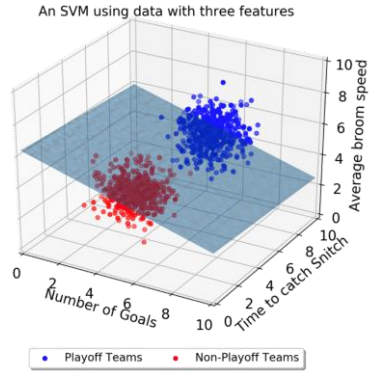
Limitations



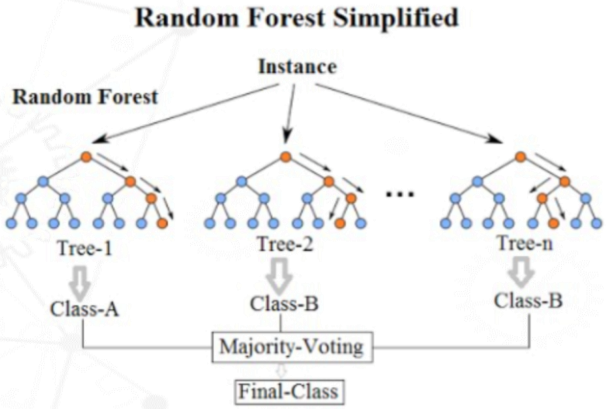
Limitation of LDA

Ensemble Classifier

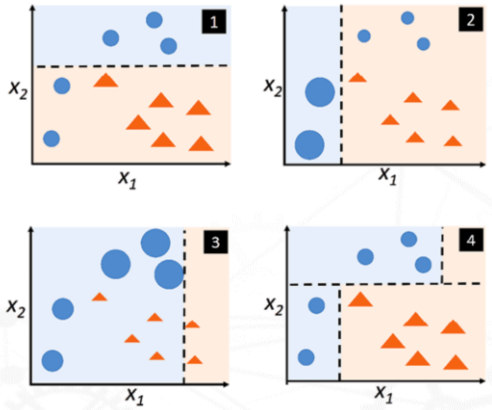
- Various algorithms for pattern recognition
  - Questioning the superiority of the algorithm
  - Existence of universally good algorithms
- Ensemble model
  - A model that combines multiple algorithms
  - Limitations of a single algorithm
    - The need for algorithms that can be solved with the highest performance given a particular problem
  - Explore universally superior algorithms



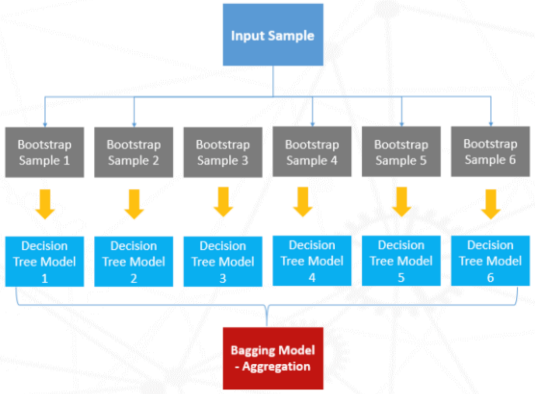
SVM



Random Forest



Adaboost



Bagging

Ensemble Classifier

❖ Motivation

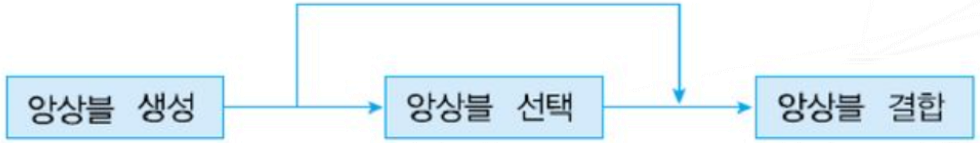
- Imitating the way people make decisions
- Final decisions are made by listening to and synthesizing the opinions of various experts

❖ Kinds of ensemble model

- Combining algorithms
  - Multiple classifier methods that create multiple classifiers and combine their outputs
- Computational shares
  - More than one algorithm intervention

❖ Reason

- Performance improvements
- Incremental learning is possible
- Efficiency in multi-sensor systems
- Effective when decision boundaries are complex
- Overcoming the challenges of quantity and quality of data



Three problems with classifier ensemble systems

Ensemble Classifier

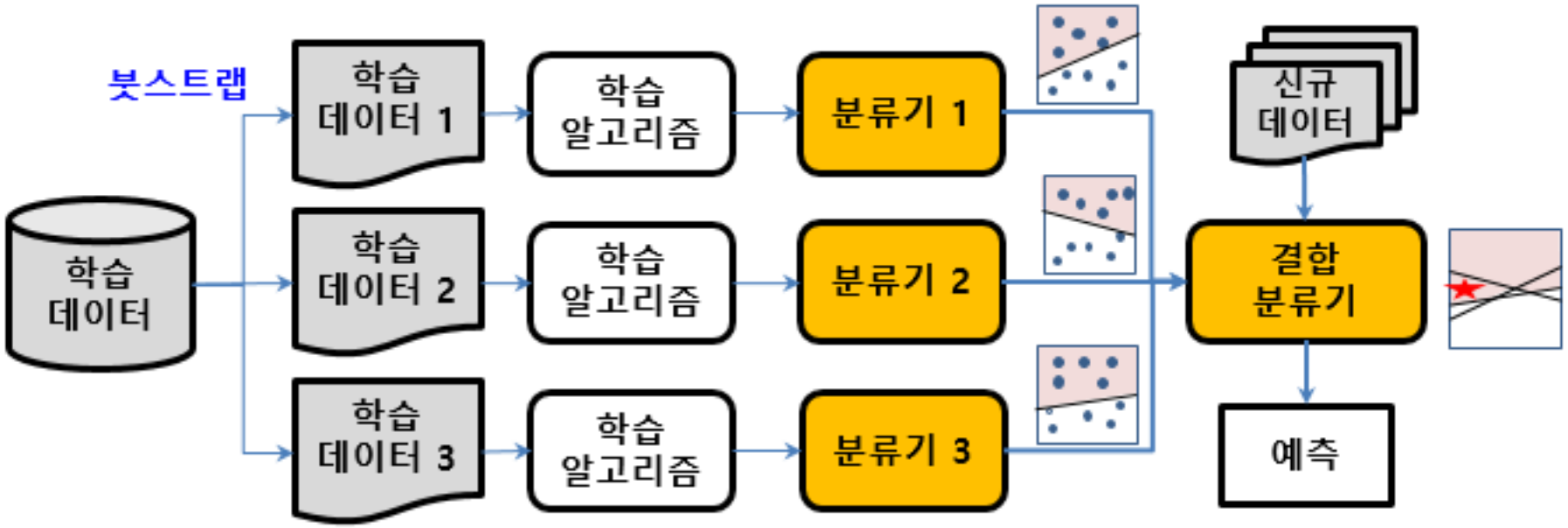
❖ Ensemble classifier

- Create several different classifiers for a given set of learning data, and combine the results of these classifiers in a voting method or a weighted voting method
- Bootstrap
  - A technique that creates a large number of learning datasets by resampling with replacement from a given learning dataset
- Bagging, bootstrap aggregating
- Boosting

Bagging Algorithm

❖ **Bagging**, **b**ootstrap **agg**regat**ing**

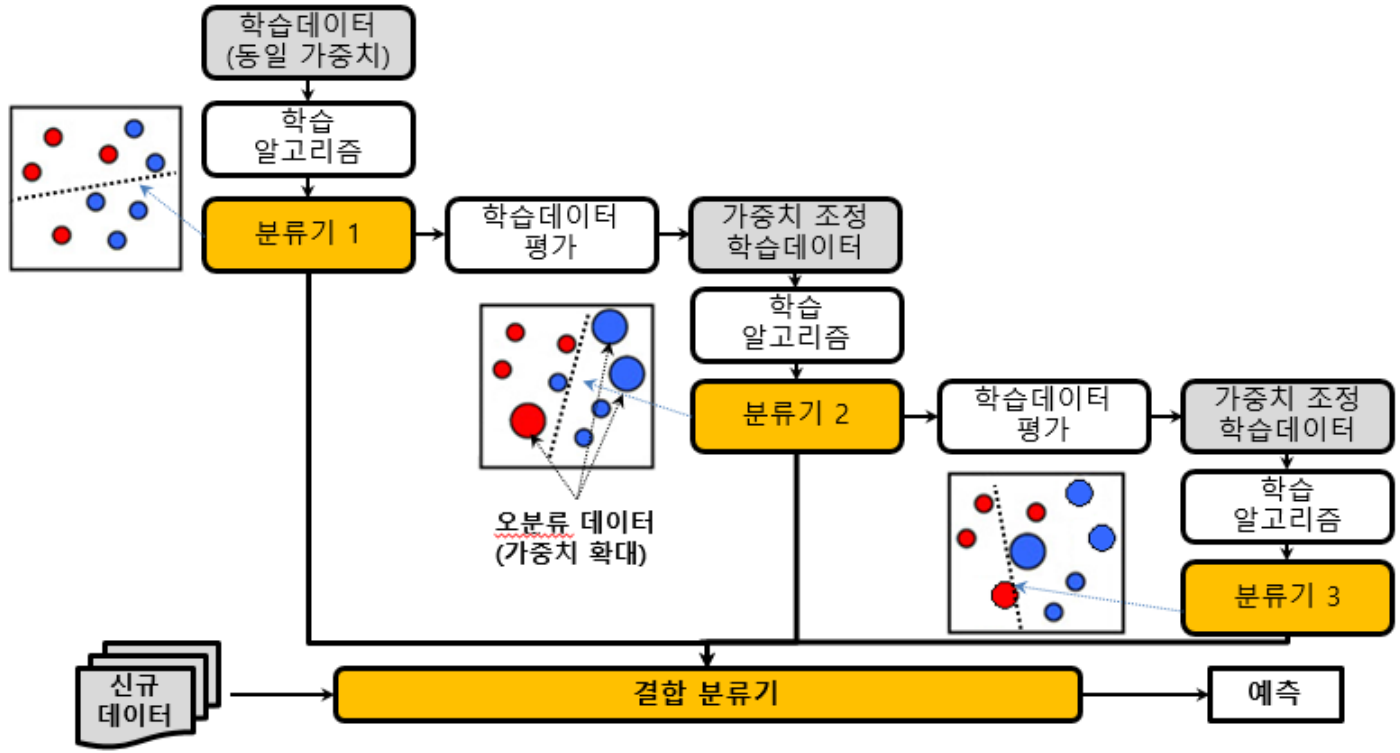
- The technique of creating multiple learning datasets through bootstrap, creating classifiers for each learning dataset, and making final decisions by voting or weighting votes



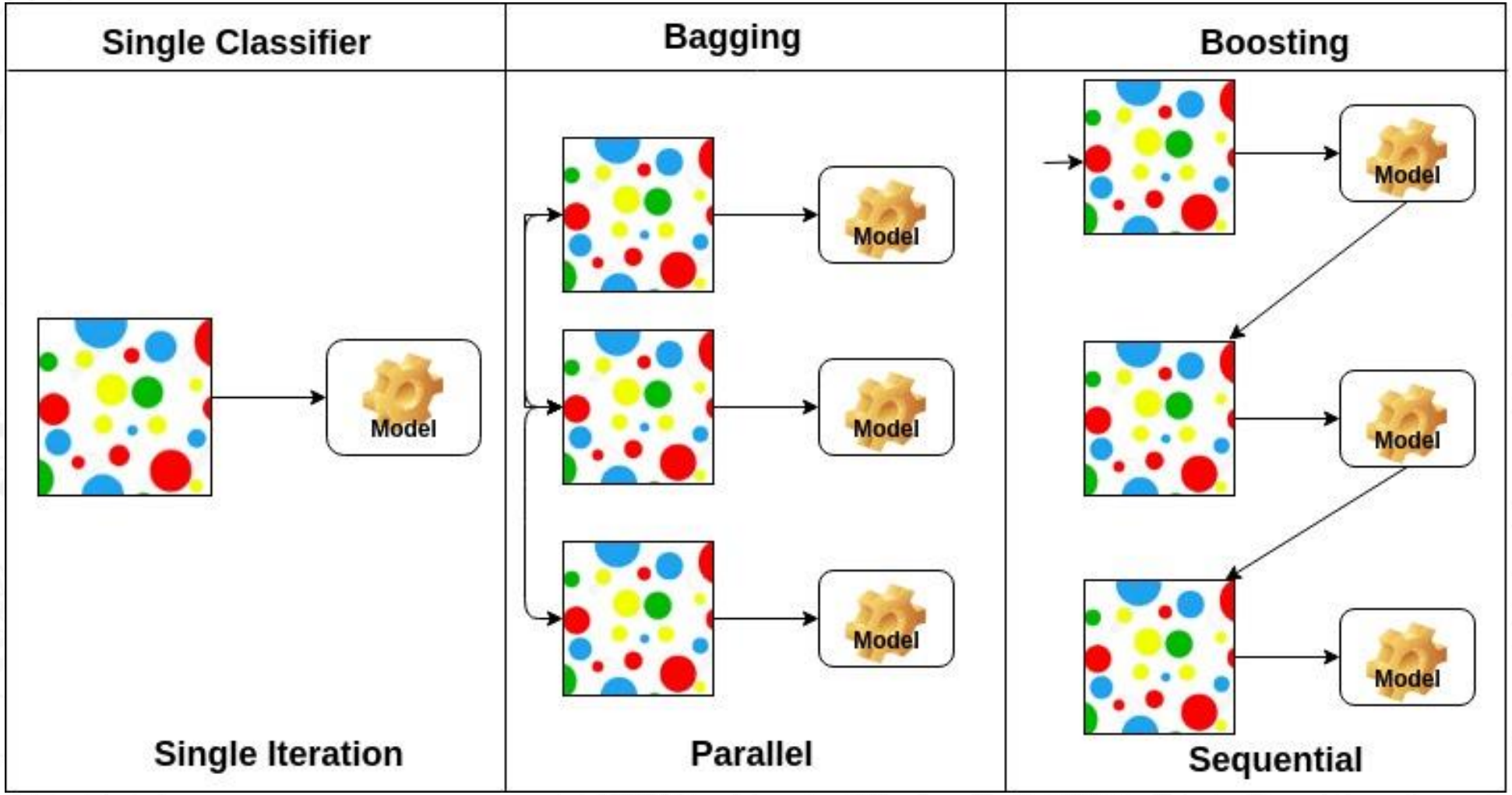
Boosting Algorithm

❖ Boosting

- An ensemble classifier generation method that creates k classifiers sequentially
- Generate a classifier by changing weights in the learning data based on classification accuracy

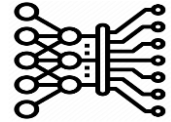


Boosting Algorithm



참조: https://python.plainenglish.io/adaboost-classifier-in-python-8d34a9f20459

Performance Evaluation



❖ Resampling techniques

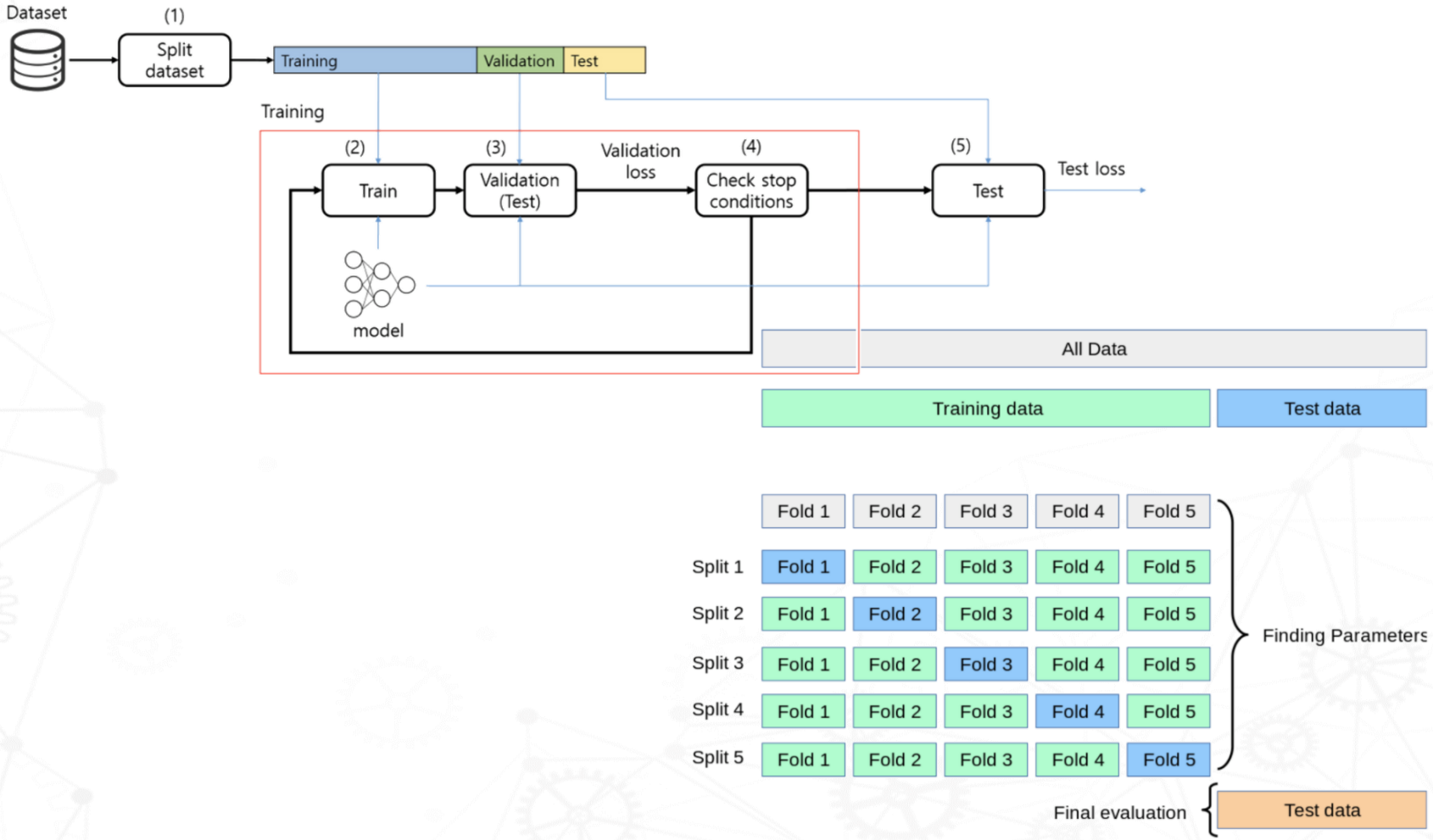
- Repeat use of the same sample
- Increased statistical reliability of performance measurements
- The need for resampling
  - Quality/quantitative quality of the database depends on the performance of the recognizer
  - When selecting a classifier model, a separate verification set is required in addition to the training set
  - Database acquisition cost issues

❖ Cross-validation

- Avoid overfitting specific data and underfitting due to lack of data
- Enables generation of generalized models based on evaluation results
- Increased training and evaluation time due to increased number of iterations

Performance Evaluation

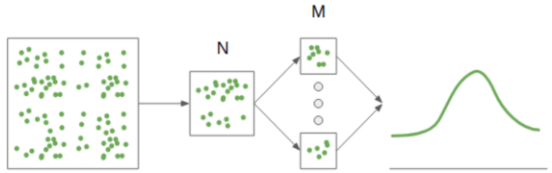
❖ Cross-Validation



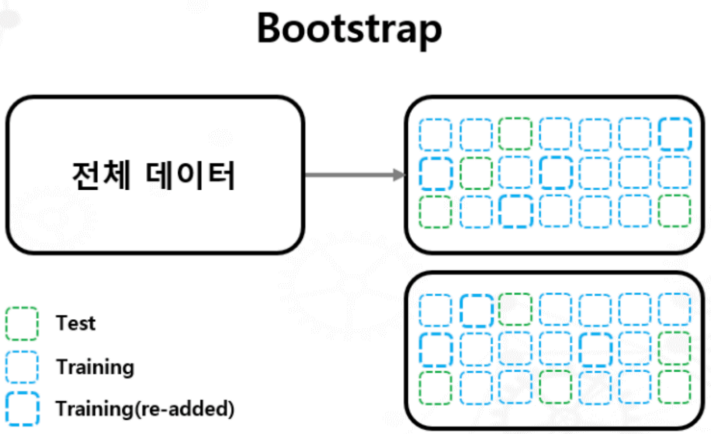
Performance Evaluation

❖ Bootstrap

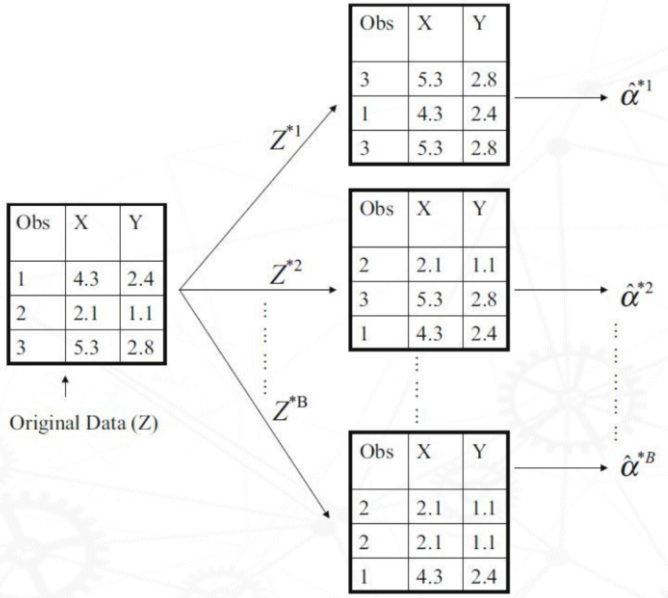
- To measure the performance of a classifier using resampling
- Random sampling techniques using restored extraction method
- Algorithms
  - 1) N개의 샘플을 가진 집합 X에서 pN개의 샘플을 임의로 선택
  - 2) 한 번에 하나씩 샘플을 선택, 선택한 것은 다시 넣음
  - 3) 2)에서 얻은 샘플 집합으로 성능 측정
  - 4) 1)~3)과정을 독립적으로 T번 수행
  - 5) 4)과정을 통해 얻어진 결과의 평균치를 최종 성능으로 결정



Example of sampling distribution

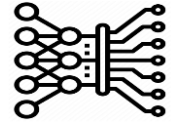


Example of bootstrap



Bootstrap sampling

Applications



- Sentiment analysis
  - It is a machine learning text analysis technique that assigns sentiment (opinion, feeling, or emotion) to words within a text, or an entire text, on a polarity scale of *Positive*, *Negative*, or *Neutral*
  - It can automatically read through thousands of pages in minutes or constantly monitor social media for posts about you
    - The tweet below, for example, about the messaging app, *Slack*, would be analyzed to pull all of the individual statements as *Positive*
    - This allows companies to follow product releases and marketing campaigns in real-time, to see how customers are reacting

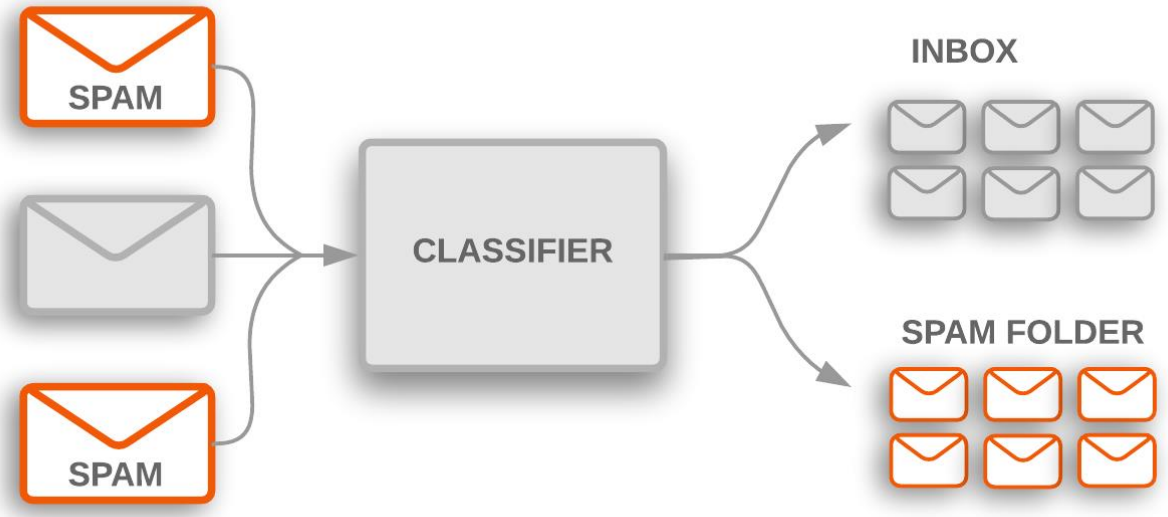
Applications

- Sentiment analysis
  - Using advanced machine learning algorithms, sentiment analysis models can be trained to read for things like sarcasm and misused or misspelled words
  - Once properly trained, models produce consistently accurate results in a fraction of the time it would take humans

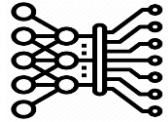
Applications

❖ Email spam classification

- Working non-stop and with little need for human interaction, email spam classification saves us from tedious deletion tasks and sometimes even costly phishing scams
- Email applications use the above algorithms to calculate the likelihood that an email is either not intended for the recipient or unwanted spam



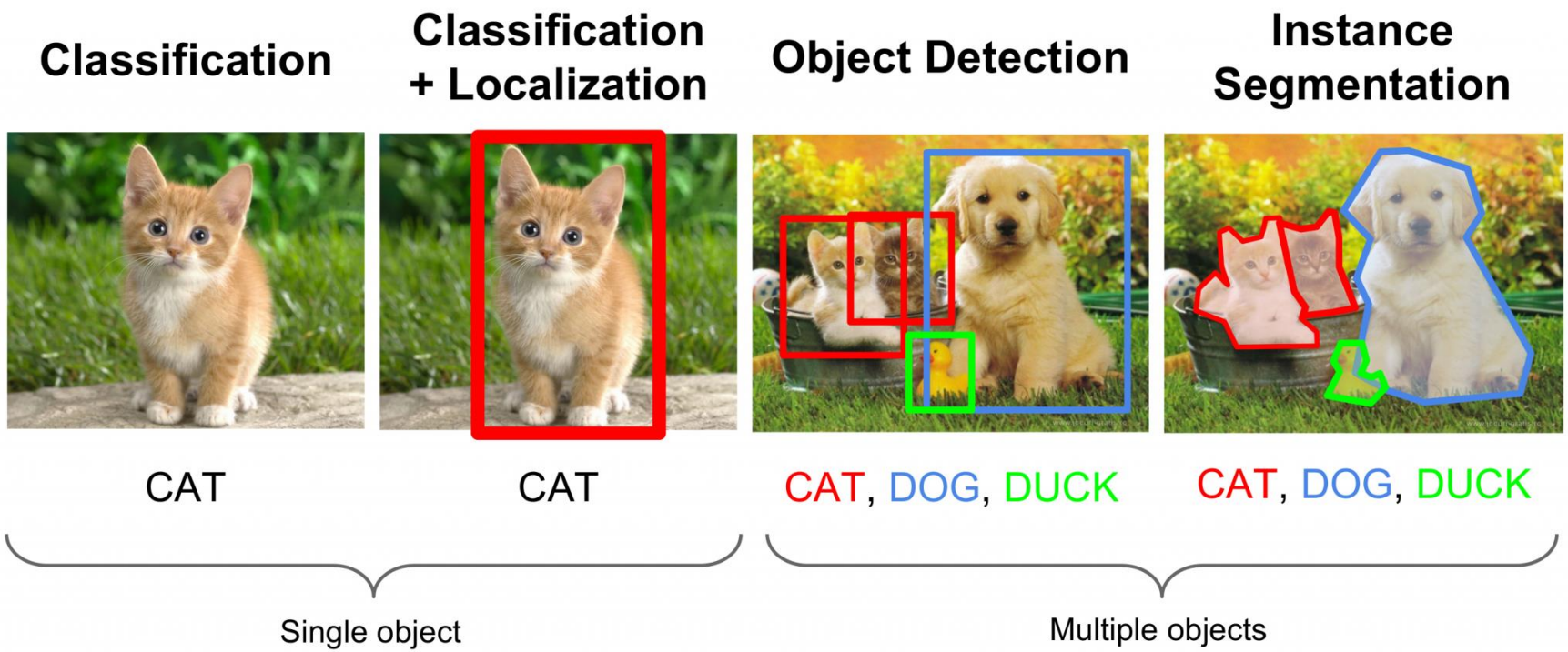
Applications



- Image classification
  - Image classification assigns previously trained categories to a given image. These could be the subject of the image, a numerical value, a theme, etc.
  - Image classification can even use multi-label image classifiers, that work similarly to multi-label text classifiers, to tag an image of a stream, for example, into different labels, like “stream,” “water,” “outdoors,” etc.
  - Using supervised learning algorithms, you can tag images to train your model for appropriate categories. As with all machine learning models, the more you train it, the better it will work

Applications

- Image classification
  - Using supervised learning algorithms, you can tag images to train your model for appropriate categories. As with all machine learning models, the more you train it, the better it will work



https://paperswithcode.com/task/image-classification

Apply What You Learned

- Select and apply one of the KNN/LDA/Decision tree models using MNIST data and explain the concept of the model
  - The concept of a model is explained in relation to the data
- Practice and interpret confusion matrices using SVM model for MNIST data
  - Please caption the main parameters