Introduction to Artificial Intelligence [AICS223]

Supervised Learning-Classification I (W04)

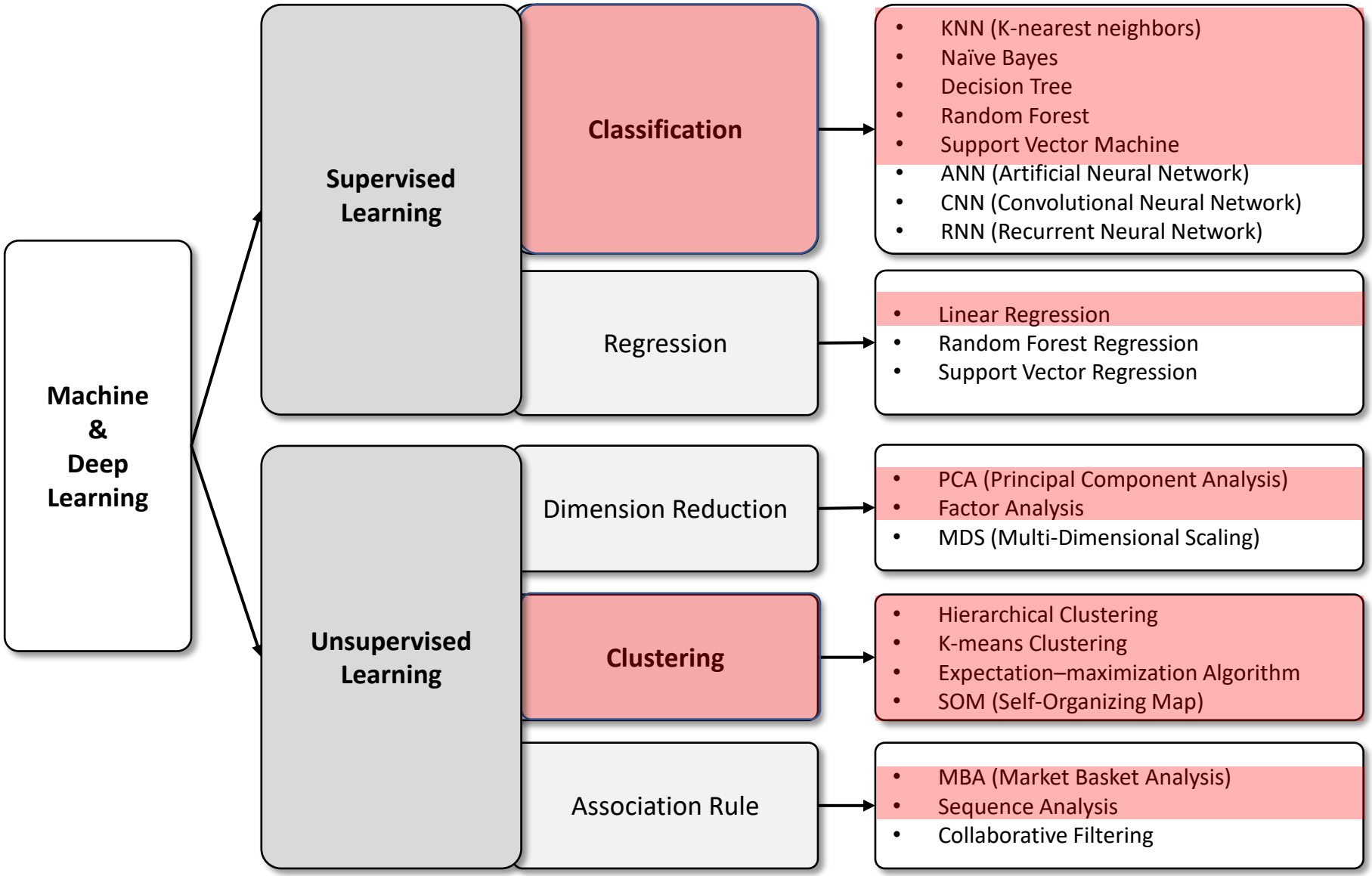
Prof. Mee Lan Han (aeternus1203@gmail.com)

고려대학교

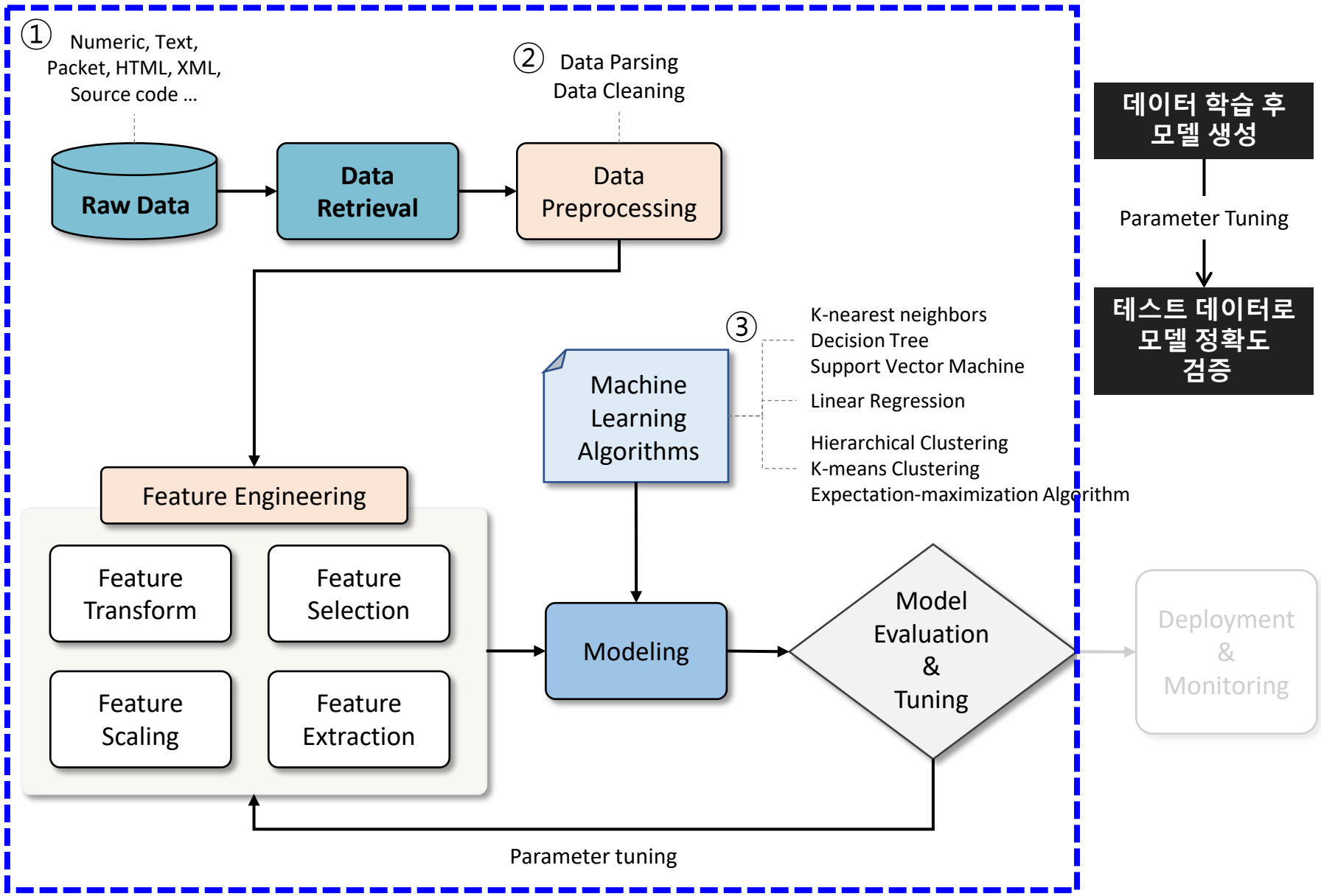
인공지능사이버보안학과

KOREA UNIVERSITY

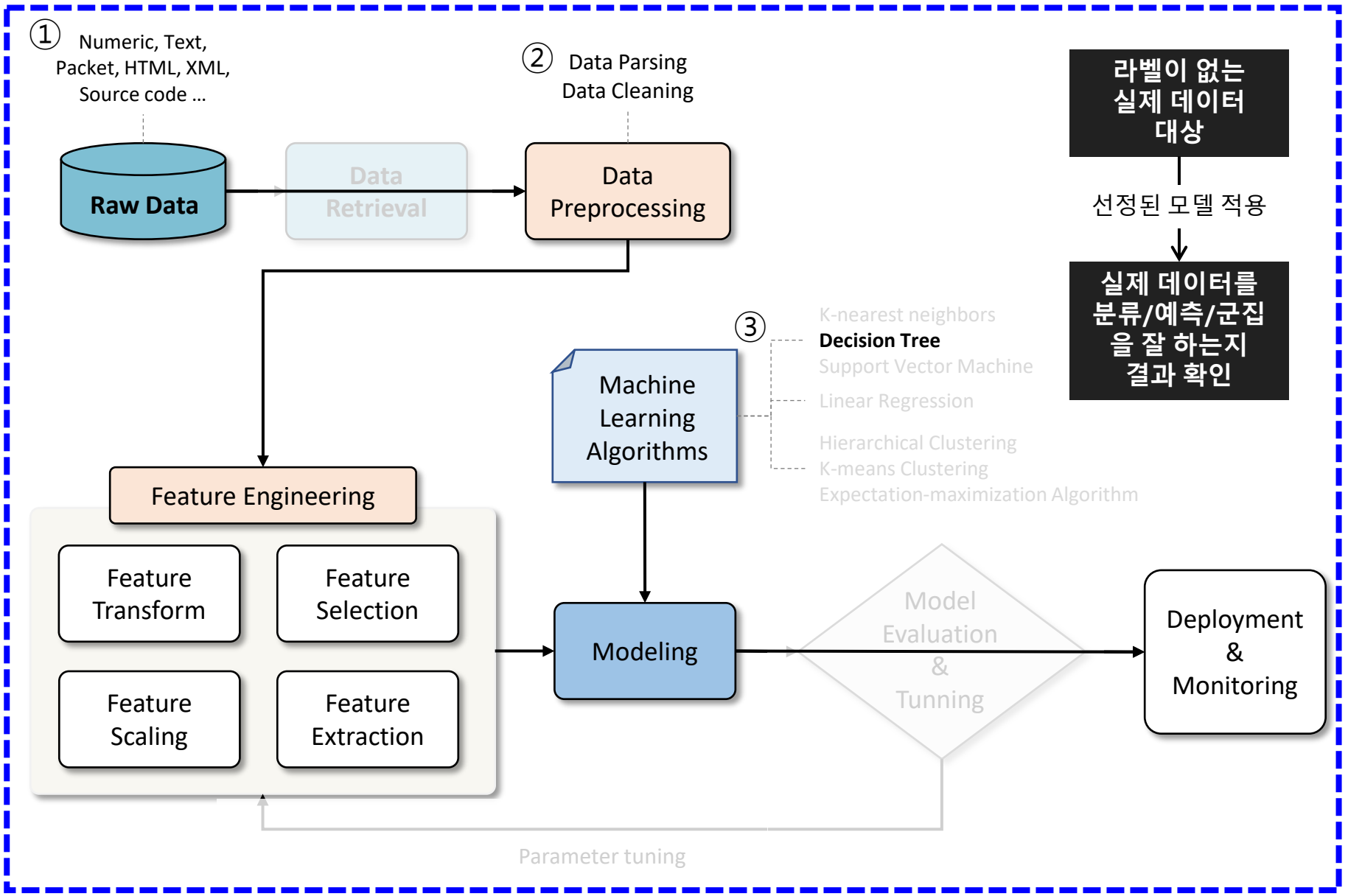
Review – AI algorithms



Review - Machine Learning Pipeline



Review - Machine Learning Pipeline



Review

■ Data Preprocessing Methods

□ Normalization & Standardization & One-Hot Encoding

□ Missing Value Treatment: 단순삭제, 다른값으로 대체, 예측값 삽입

□ Outlier Treatment: 단순삭제, 다른값으로 대체, 리샘플링, Case분리 멀티분석

■ Feature Engineering

□ Feature Treatment: Scaling, Binning (구간화), Transform, Dummy

□ Feature Engineering

□ Underfitting/Overfitting: 과소적합, 과대적합

CONTENTS

- **Classification**
  - ✓KNN (K-nearest neighbors)
  - ✓SVM(Support Vector Machine)
  - ✓One-class Classification
  - ✓Decision Tree
  - ✓Random Forest
  - ✓XGBoost

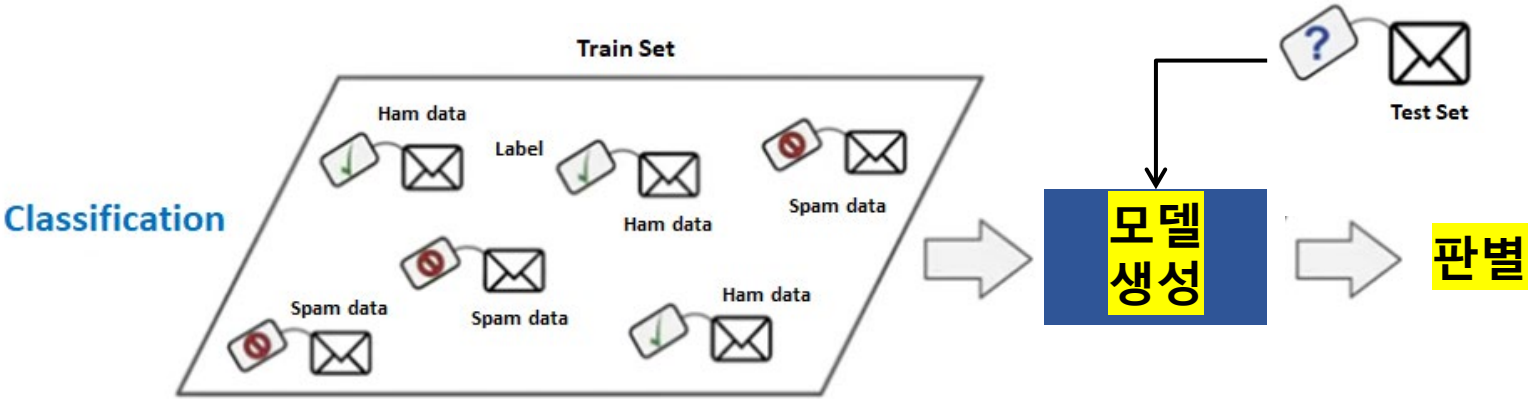
Machine Learning

■ Supervised Learning - Classification

□ 분류 모델은 **입력 데이터를 사전에 정의된 클래스 레이블로 분류하는 작업을 수행하는 모델**

□ 입력 데이터와 해당 데이터의 클래스 레이블이 주어지는 학습 데이터를 사용하여 모델을 학습

□ **학습된 모델을 사용하여 새로운 입력 데이터의 클래스 레이블을 예측하거나 분류**



□ **빈도/거리기반 분류 알고리즘**

- ① K-Nearest Neighbor (K-NN) / ② Support Vector Machine (SVM) / ③ One-class Classification

□ **트리기반 분류 알고리즘**

- ① Decision Tree / ② Random Forest(Bagging) / ③ XGBoost (Boosting)

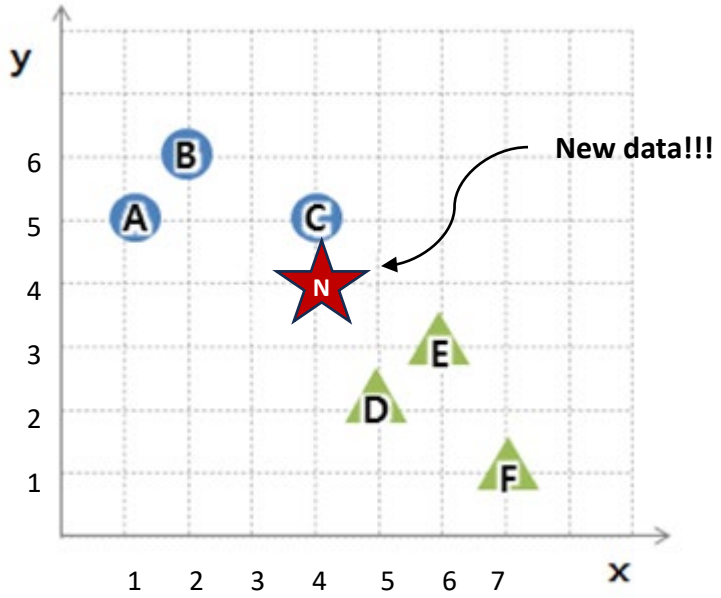
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

□ 특정공간 내 새로 들어온 입력값이 어떤 그룹의 데이터와 가장 가까운가 분류하는 알고리즘

□ **K의 역할**은 **몇 번째로 가까운 데이터까지 살펴볼 것인가를 설정한 수**

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	??



□ K 의 default 값은 5

□ 일반적으로 **K 는 홀수를 사용** (짝수일 경우 동점이 되어 하나의 결과를 도출할 수 없음)

K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

□ 장점

- **높은 정확도**
- **단순하며 효율적 (모델 생성을 미리 하지 않음)**
- **가까운 상위 k 개의 데이터만 활용**하기 때문에 오류 데이터가 결과에 영향을 미치지 않음
- 데이터 분산에 대한 추정을 만들 필요가 없음

□ 단점

- 모든 데이터를 비교해야 하기 때문에, **데이터가 많아질수록 처리 시간이 증가**
- 메모리를 많이 사용하게 되어 고사양의 하드웨어 필요
- **모델 생성이 미리 되지 않아 새로운 데이터에 대한 학습 시간보다 분류/예측 시간이 더 걸림**

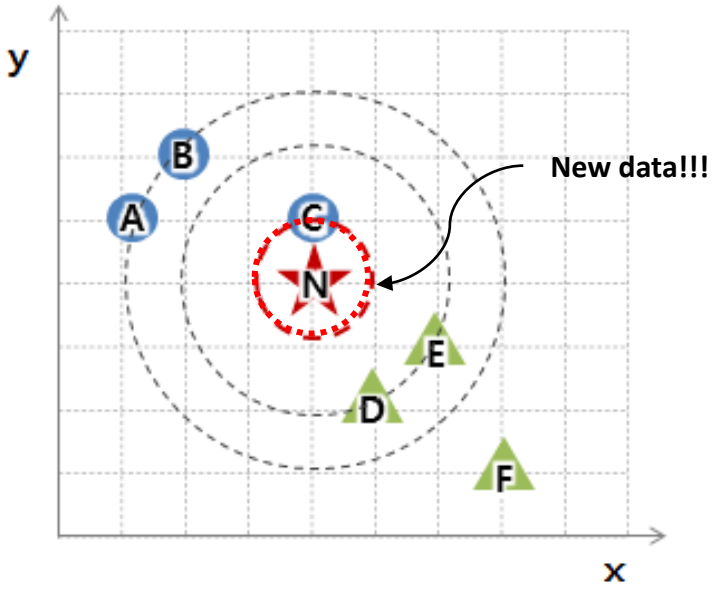
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 1** 이라면, 거리가 첫 번째로 가까운 데이터만 (그림에서는 C 부분)을 가지고 신규 데이터를 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	●



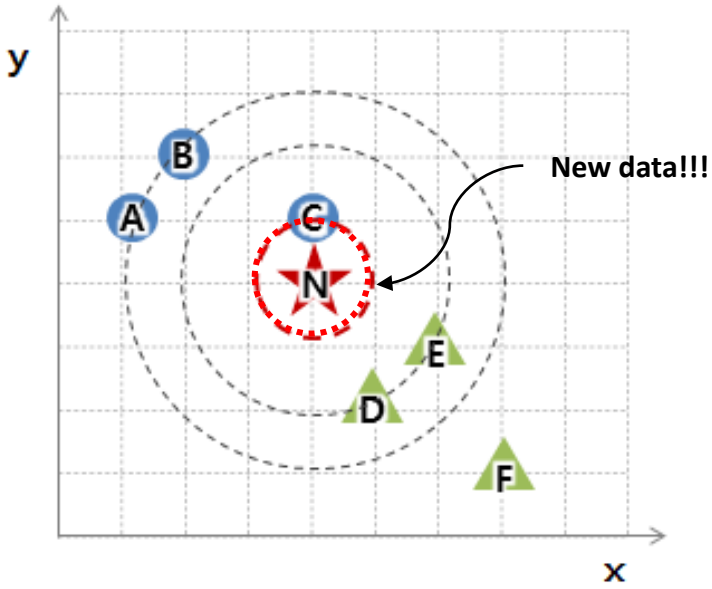
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 1** 이라면, 거리가 첫 번째로 가까운 데이터만 (그림에서는 C 부분)을 가지고 신규 데이터를 분류
- 새로운 입력값 N은 C 와 같은 그룹인 ●로 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	●



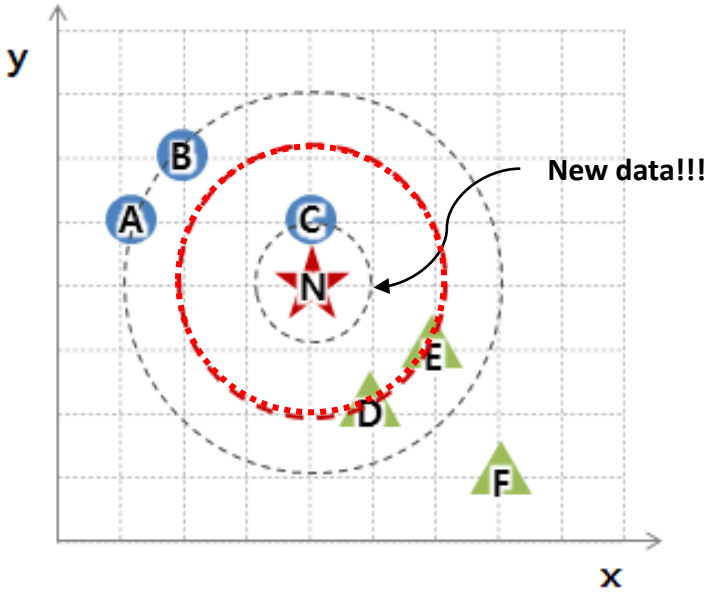
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 3** 이라면, 거리가 세 번째까지 가까운 C, D, E까지 보고 신규 데이터를 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	▲



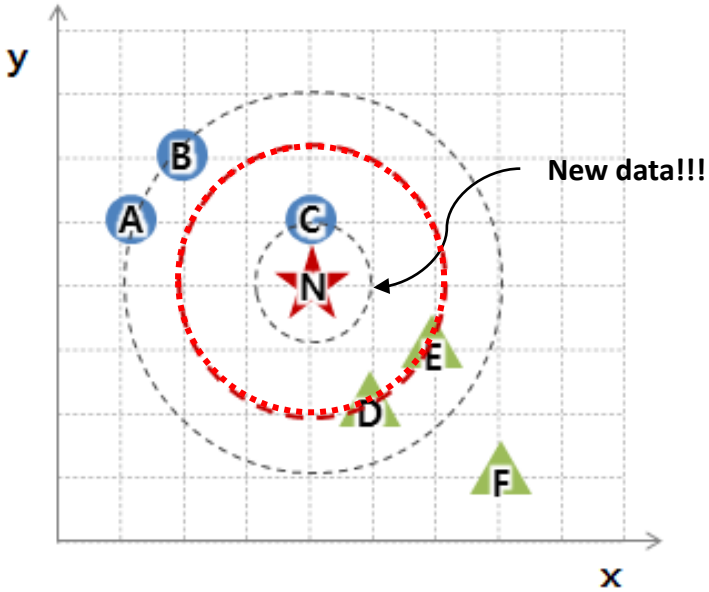
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 3** 이라면, 거리가 세 번째까지 가까운 C, D, E까지 보고 신규 데이터를 분류
- 다수결의 원칙에 따라 새로운 입력값 N은 D, E 와 같은 그룹인 ▲ 로 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	▲



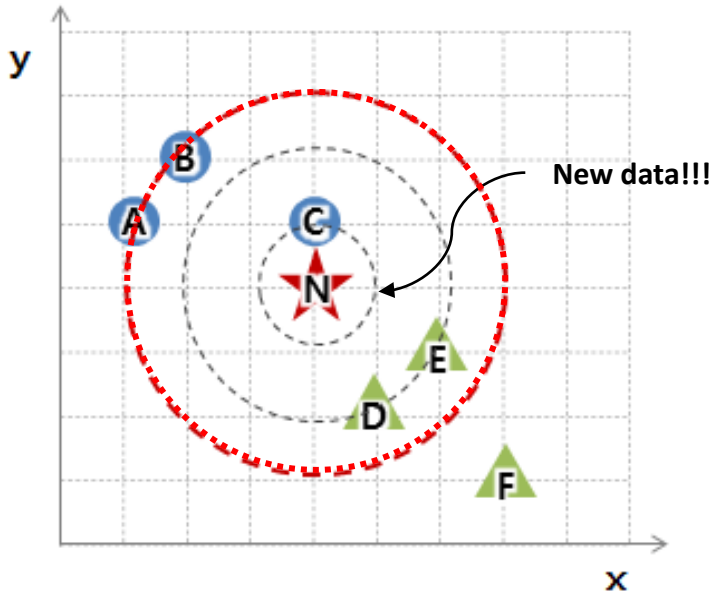
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 5** 이라면, 거리가 다섯 번째까지 가까운 C, D, E, B, A까지 보고 신규 데이터를 분류
- 다수결의 원칙에 따라 새로운 입력값 N은 A, B, C 와 같은 그룹인 ●로 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	●



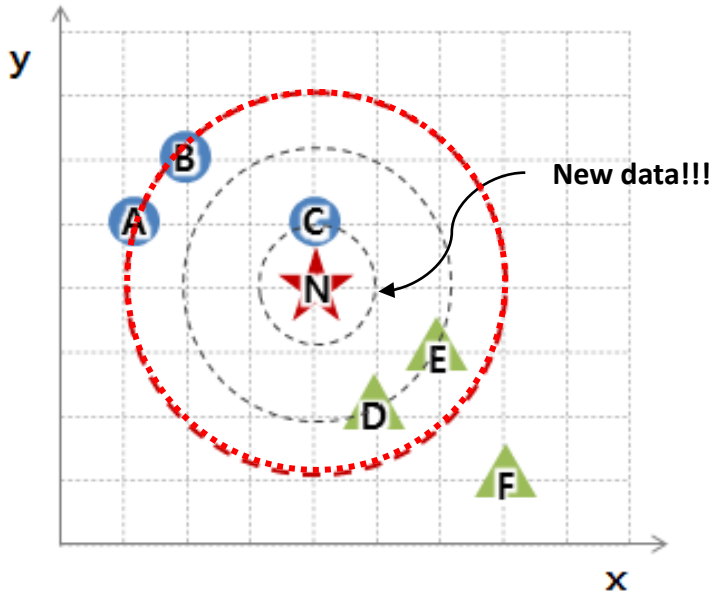
K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

1) 빈도 기준 (k 의 개수)

- 만약에 **K = 5** 이라면, 거리가 다섯 번째까지 가까운 C, D, E, B, A까지 보고 신규 데이터를 분류
- 다수결의 원칙에 따라 새로운 입력값 N은 A, B, C 와 같은 그룹인 ●로 분류

Data	(x, y)	Class
A	(1, 5)	●
B	(2, 6)	●
C	(4, 5)	●
D	(5, 2)	▲
E	(6, 3)	▲
F	(7, 1)	▲
N (입력값)	(4, 4)	●



K 값에 따라 New data가 ● 로 분류되기도 하고 ▲ 로 분류되기도 함

적절한 K 를 정해주는 게 중요

K-Nearest Neighbor

■K-최근접 이웃 알고리즘 (K-NN)

2) 거리 기준

- 유클리디안 거리 (Euclidean Distance) 계산식 사용

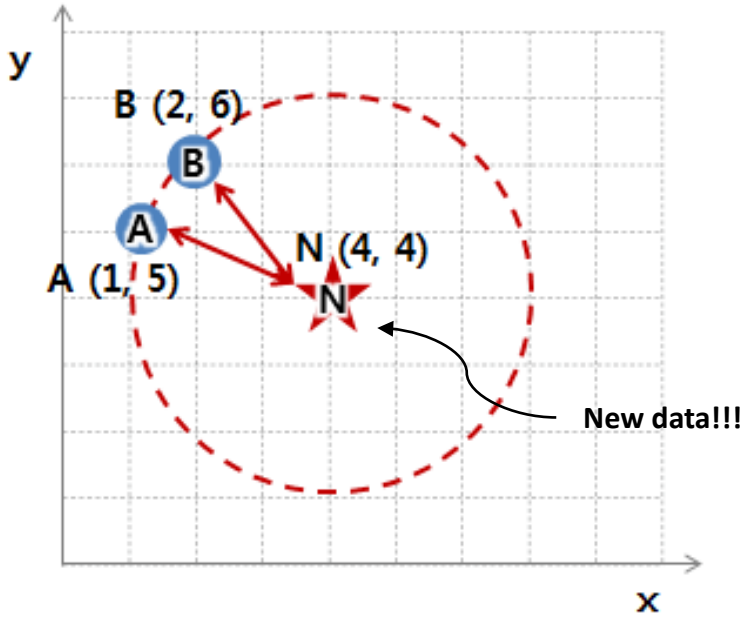
$$Distance(A, N) = \sqrt{(x_A - x_N)^2 + (y_A - y_N)^2} = 3.16227 \ldots$$

$$Distance(\boldsymbol{B, N}) = \sqrt{(x_B - x_N)^2 + (y_B - y_N)^2} = 2.82842 \ldots$$

□ B와 N 사이가 더 가까움

□ 가중합 (Weighted voting)

- 거리가 가까운 (유사도가 높음) 이웃의 정보에 가중치를 더 주는 방식

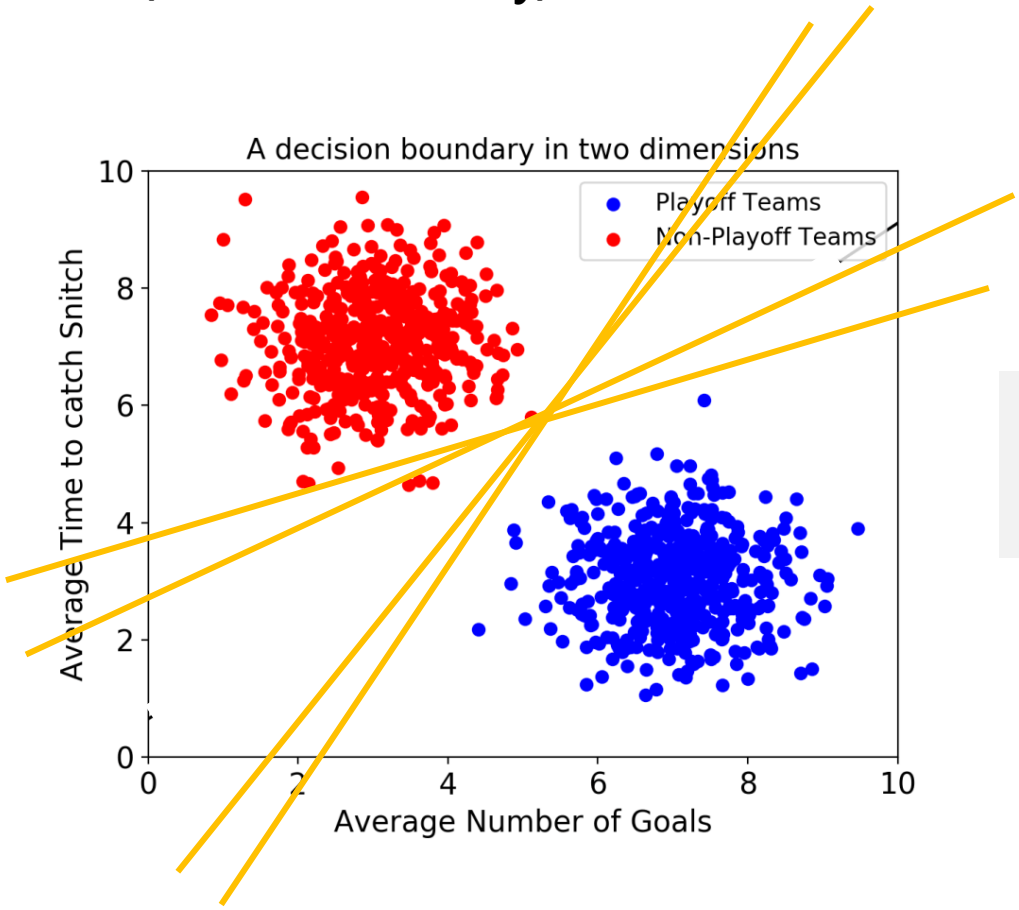


SVM (Support Vector Machine)

■SVM 이란?

□ 주어진 데이터가 어느 카테고리에 속할지 판단하는 **이진 선형 분류** 모델

□ **결정 경계 (Decision Boundary)**, 즉 분류를 위한 기준선을 정의하는 모델



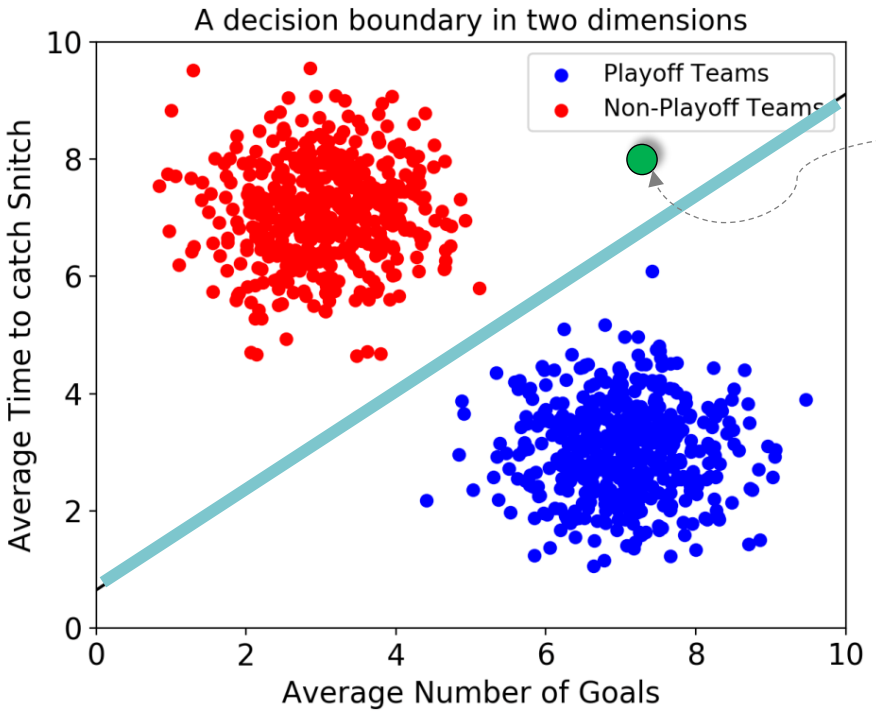
경계의 어느 쪽에 속하는지 확인해서 분류 과제를 수행

SVM (Support Vector Machine)

■SVM 이란?

□ 주어진 데이터가 어느 카테고리에 속할지 판단하는 **이진 선형 분류** 모델

□ **결정 경계 (Decision Boundary)**, 즉 분류를 위한 기준선을 정의하는 모델



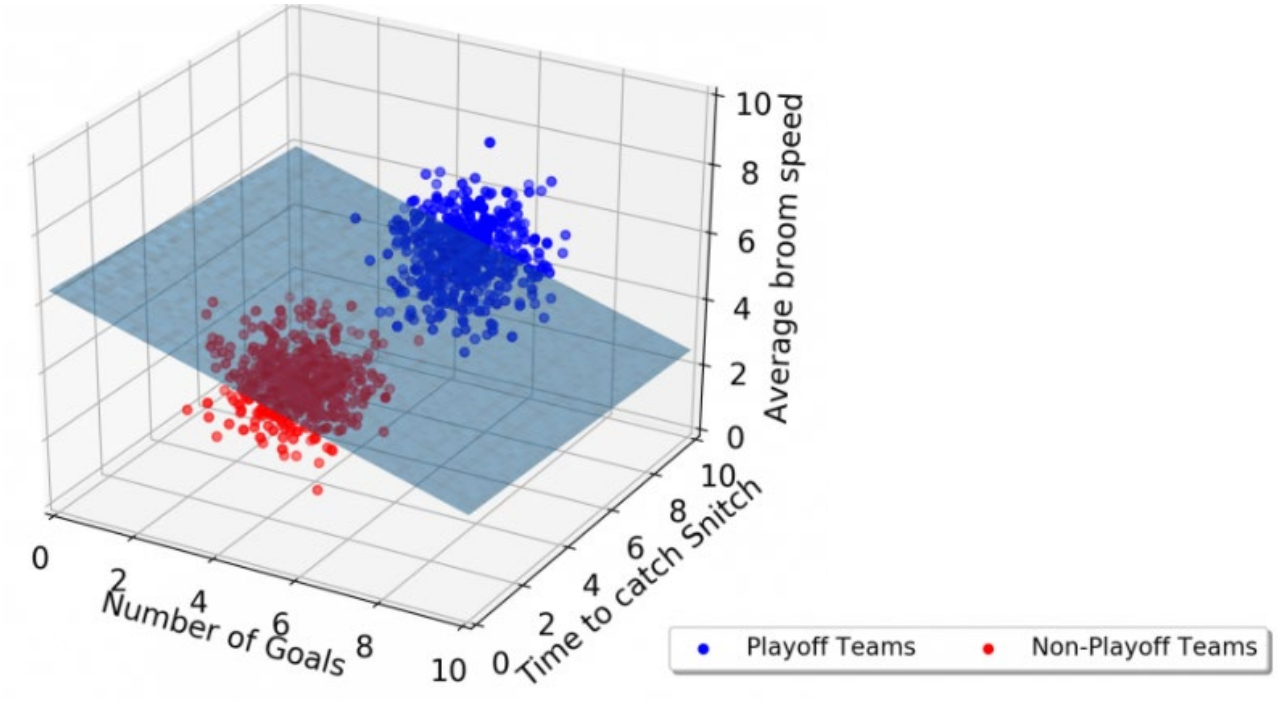
새로운 데이터

경계의 어느 쪽에 속하는지 확인해서 분류 과제를 수행

SVM (Support Vector Machine)

■SVM 이란?

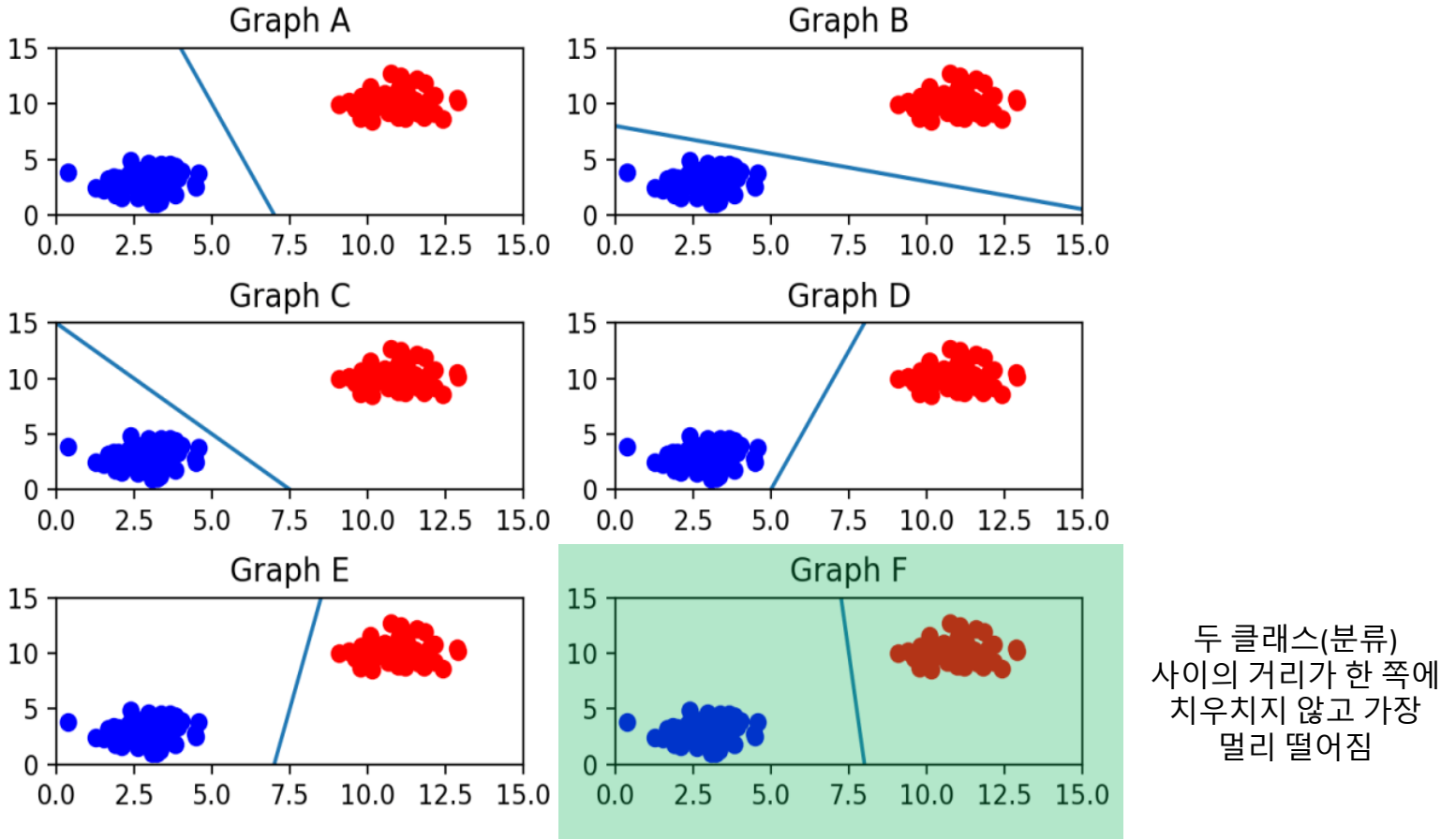
□ Feature (차원, 속성)가 3개 일 경우 3차원으로 표현
□ 3차원일 때, 결정 경계는 선이 아닌 **평면**
□ Feature (차원, 속성)의 개수가 늘어나면, 결정 경계도 단순한 평면이 아닌 고차원이 됨
□ 이를 초평면 (hyperplane) 이라고 함



SVM (Support Vector Machine)

■최적의 결정 경계 (Decision Boundary)

- □ 어떤 결정 경계가 가장 적절할까?
- □ 결정 경계는 데이터 각각의 군집으로부터 최대한 멀리 떨어지는 것이 좋음



두 클래스(분류) 사이의 거리가 한 쪽에 치우치지 않고 가장 멀리 떨어짐

SVM (Support Vector Machine)

■마진 (Margin)

□ **Support Vectors**: 결정 경계와 가까이 분포된 데이터를 의미

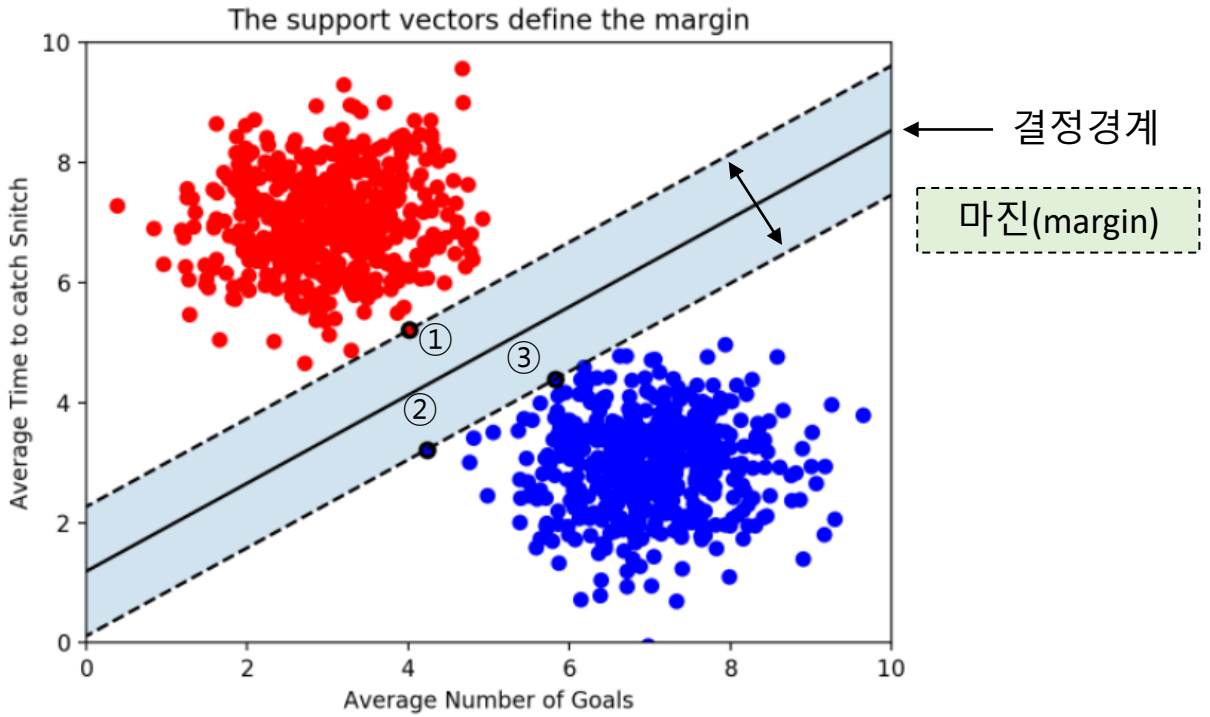
□ **Margin**: 결정 경계와 서포트 벡터 사이의 거리를 의미 → 경계를 정의하는 역할

□ 최적의 결정 경계는 마진을 최대화함 (명확하게 데이터를 분류한다는 의미)

□ n개의 Feature (속성)을 가진 데이터에는 **최소 n+1개의 서포트 벡터가 존재**

→ **3차원 (Feature 3개라면, 서포트 벡터는 4개 존재)**

Support Vectors
①②③



← 결정경계

마진(margin)

SVM (Support Vector Machine)

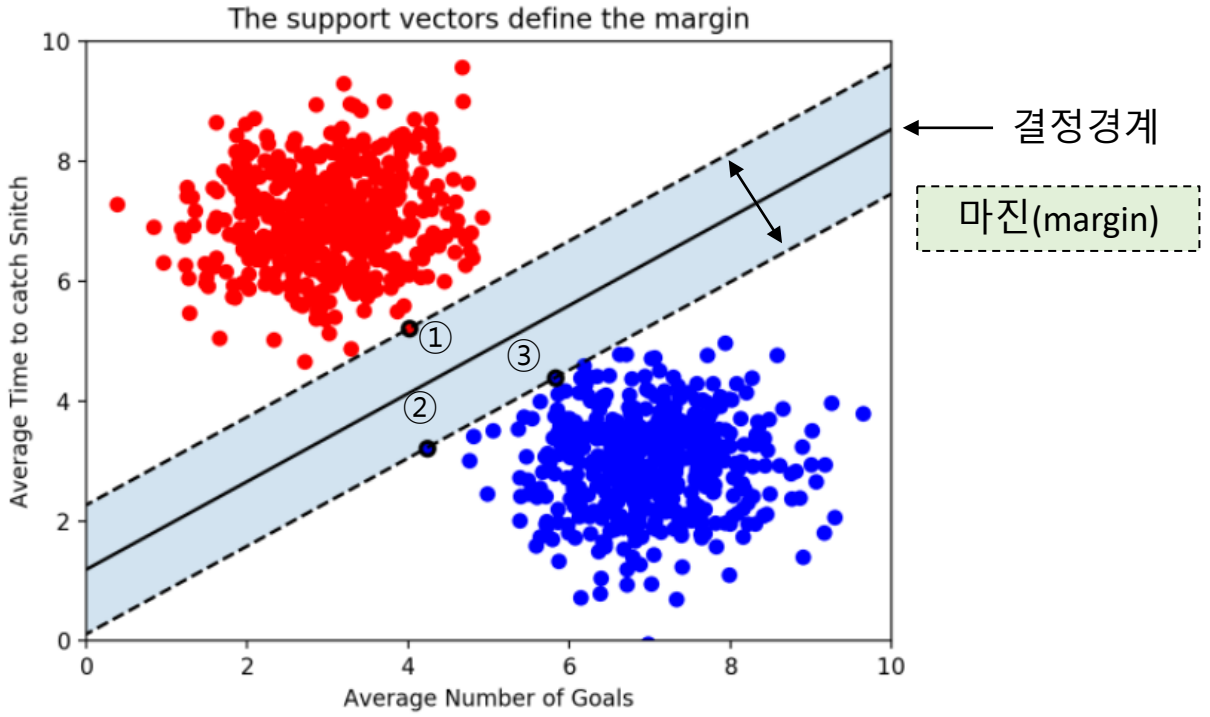
■SVM의 장점

□ 대부분의 머신러닝 알고리즘은 학습 데이터 모두를 사용하여 모델을 학습

□ **SVM은 결정 경계를 정의하는 서포트 벡터만 잘 골라내면 됨**

□ 서포트 벡터만으로 학습 → 모든 데이터를 확인할 필요가 없어 속도가 매우 빠름

Support Vectors
①②③

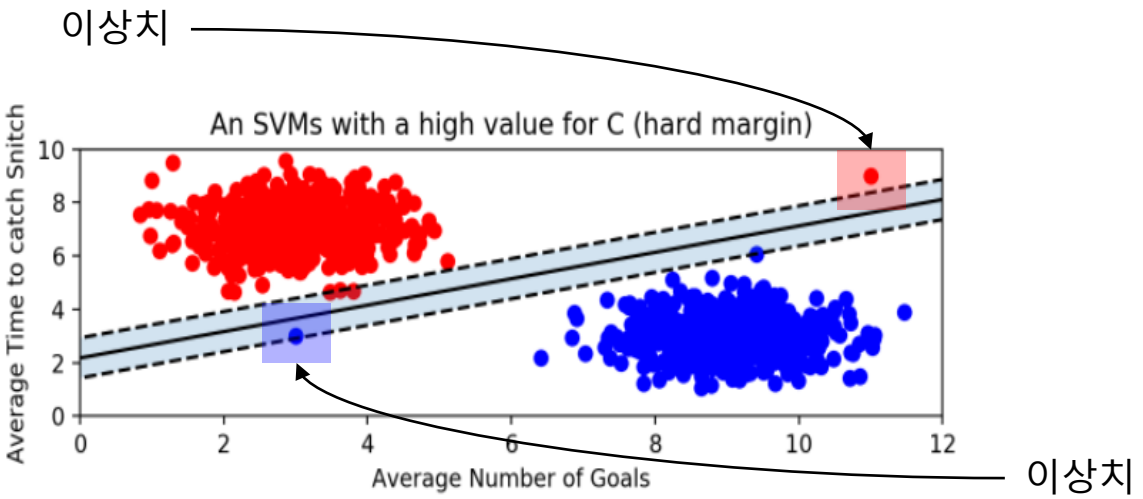


SVM (Support Vector Machine)

■이상치 (Outlier) 허용 기준

□ SVM은 데이터를 분류 시 마진 (Margin) 의 크기를 최대화 하도록 수행됨

□ 이상치 (outlier) 데이터를 잘 다루는 것이 중요!!



하드 마진 (hard margin)

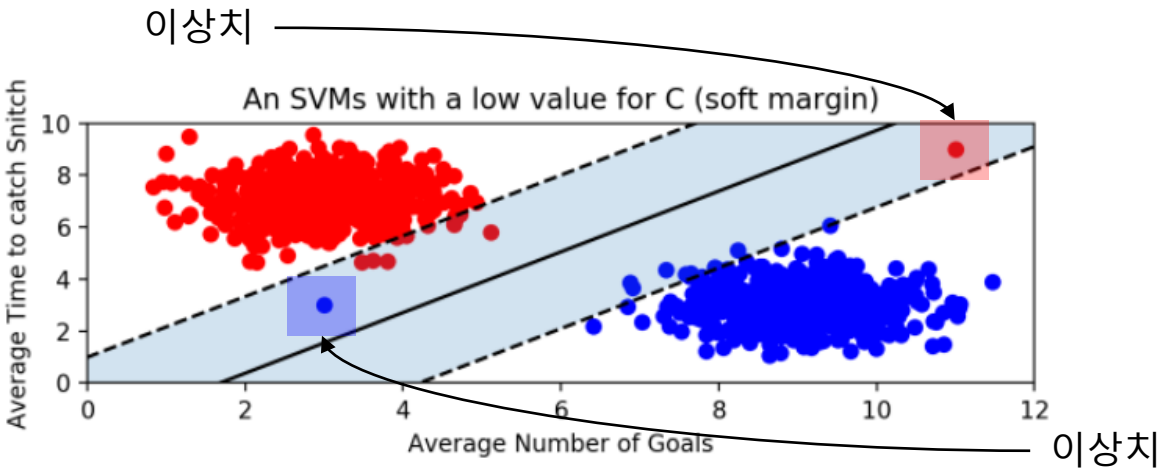
- 서포트 벡터와 결정 경계 사이의 거리가 매우 좁음 (마진이 작음)
- 개별적인 학습 데이터들을 다 놓치지 않으려고 이상치를 허용하지 않는 기준으로 결정 경계 설정
- 오버피팅 (Overfitting) 문제가 발생

SVM (Support Vector Machine)

■이상치 (Outlier) 허용 기준

□ SVM은 데이터를 분류 시 마진 (Margin) 의 크기를 최대화 하도록 수행됨

□ 이상치 (outlier) 데이터를 잘 다루는 것이 중요!!



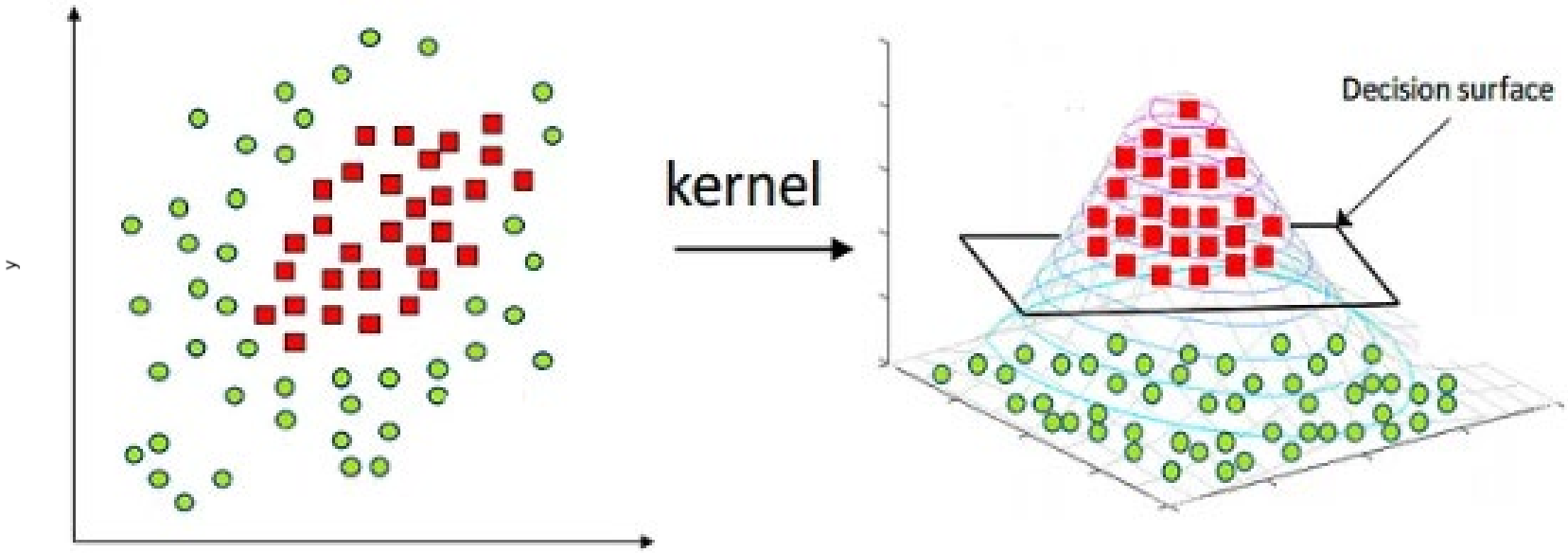
소프트 마진 (soft margin)

- 서포트 벡터와 결정 경계 사이의 거리가 넓음 (마진이 커짐)
- 아웃라이어들이 마진 안에 어느정도 포함되도록 너그럽게 기준을 잡음
- 너무 대충대충 학습하는 경우라 언더피팅 (Underfitting) 문제가 발생

SVM (Support Vector Machine)

■커널 (Kernel)

- □ 선형으로 분류할 수 없는 데이터 특성을 갖고 있는 경우가 존재
- □ 커널 (Kernel)은 원래 가지고 있는 데이터를 더 높은 차원의 데이터로 변환
- □ **선형에서 구분하지 못하는 구조를 Kernel을 사용해 데이터를 변환해 구분**



SVM (Support Vector Machine)

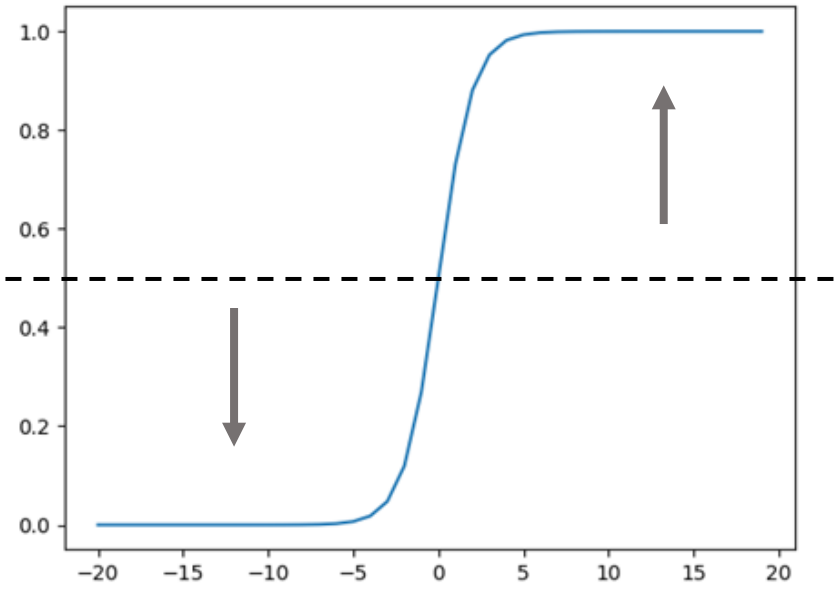
■커널 (Kernel)의 종류

□ 다항식 (Polynomial) 함수

- 2차원의 점으로 나타낼 수 있는 데이터를 3차원으로 변환
- 다항식 (polynomial) 커널을 사용하면 데이터를 더 높은 차원으로 변형하여 나타냄
- 초평면 (hyperplane)의 결정 경계를 얻을 수 있음

□ 시그모이드 (Sigmoid) 함수

- 0~1사이의 값을 갖는 함수
- S자 모양의 형태로 0부터 1 사이의 값을 갖기 때문에 이진 분류 가설을 표현하기 좋음



- 예측값이 0.5이상이면 1 로 분류
- 예측값이 0.5 이하면 0으로 분류

SVM (Support Vector Machine)

■커널 (Kernel)의 종류

□ 방사 기저 함수 (RBF: Radial Bias Function)

- 2차원의 점으로 나타낼 수 있는 데이터를 무한의 차원으로 변환
- Parameter: Cost, **Gamma**
- **Gamma**
  - 결정 경계를 얼마나 유연하게 그려줄 것인지 정해주는 파라미터
  - 학습 데이터에 얼마나 민감하게 반응할 것인지 모델을 조정하는 부분

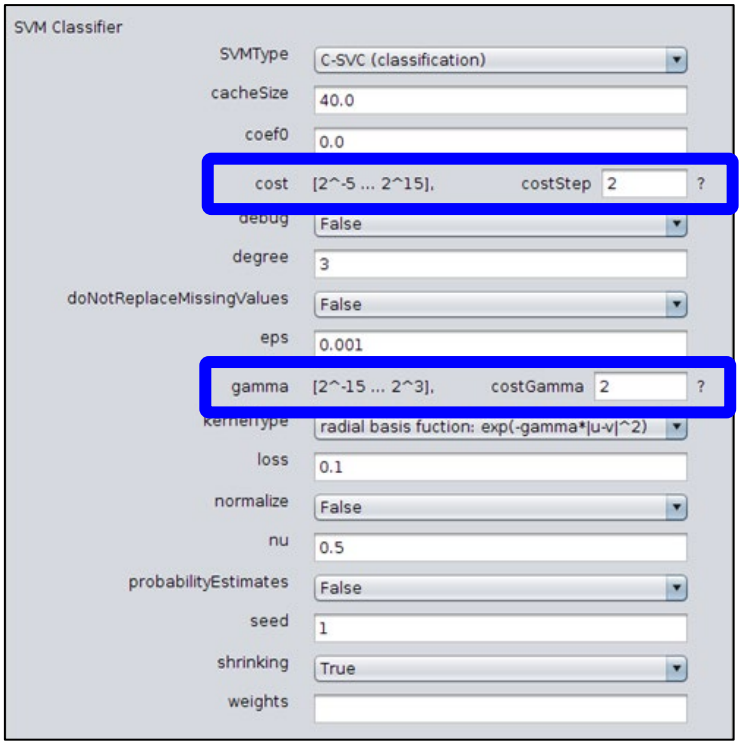
```
from sklearn.svm import SVC

classifier = SVC(kernel = 'linear')
```

```
from sklearn.svm import SVC

classifier = SVC(kernel='rbf')
```

```
classifier = SVC(kernel = "rbf", C = 2, gamma = 0.5)
```



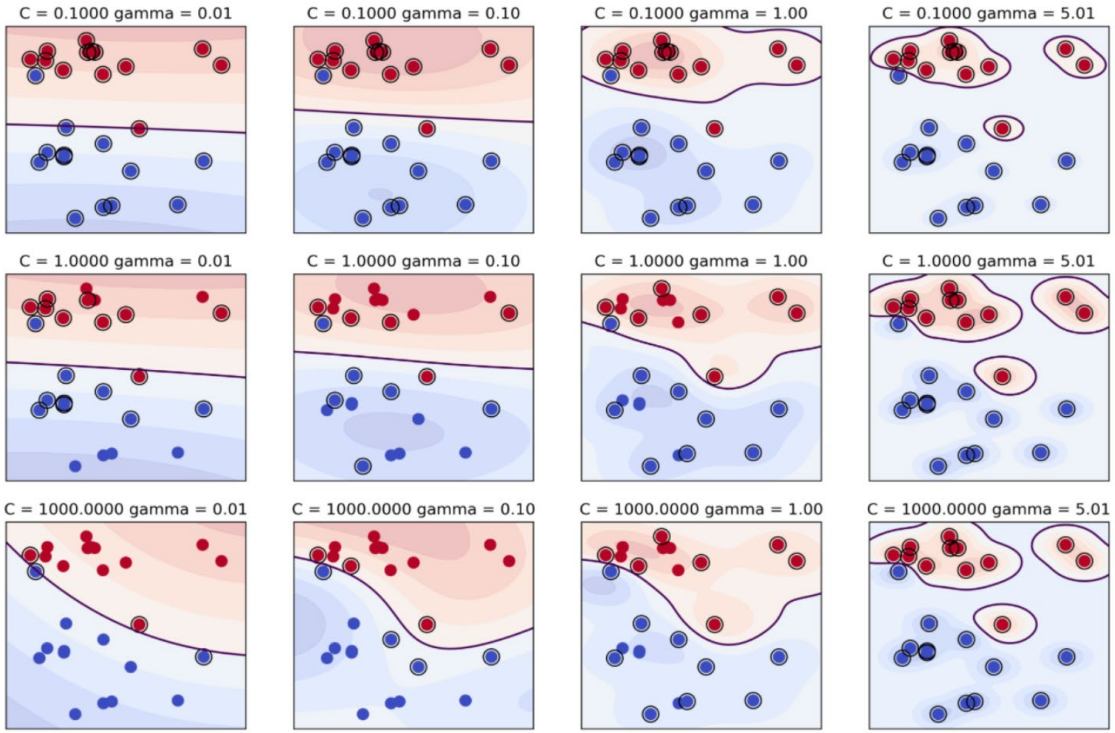
SVM (Support Vector Machine)

■Gamma

□ 감마 값은 모델이 생성하는 경계의 복잡성 정도를 결정함

- 감마 값이 낮을수록 모델은 학습 데이터의 속성에 의존하지 않음
- 결정 경계는 직선으로 생성
- **Underfitting**

- 감마 값이 높을수록 모델이 학습 데이터의 속성에 더 많이 의존
- 결정 경계는 복잡하게 생성
- **Overfitting**

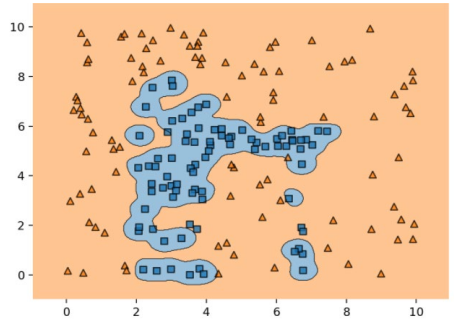


https://amueller.github.io/COMS4995-s18/slides/aml-09-021418-support-vector-machines/#1

SVM (Support Vector Machine)

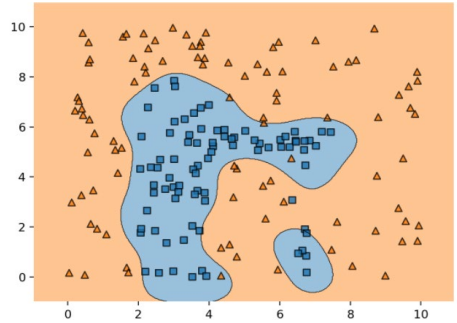
■Gamma

□ 모델이 생성하는 경계가 복잡해지는 정도



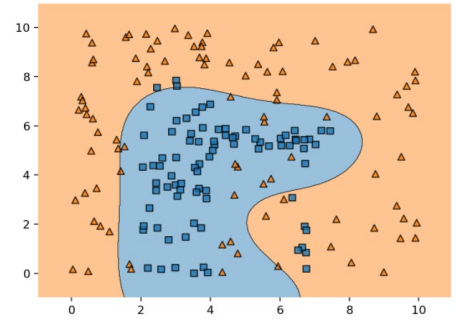
$gamma = 10$

- gamma값을 높이면 학습 데이터에 많이 의존
- 결정 경계를 구불구불하게 설정하게 됨
- Overfitting 을 초래



$gamma = 1$

- 적절한 경계



$gamma = 0.1$

- gamma를 낮추면 학습 데이터에 별로 의존하지 않음
- 결정 경계를 직선에 가깝게 단순해짐
- Underfitting 이 발생

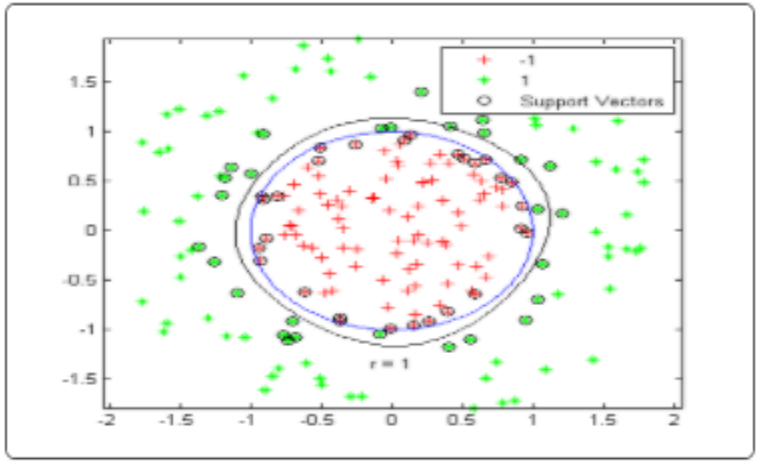
One Class Classification

■One-class Classification

□ Binary classification을 할 때, class 한 쪽의 데이터만 있고 나머지 데이터는 없는 경우

□ 정상 샘플만을 이용해 모델을 학습

□ 정상 샘플과 거리가 먼 샘플을 이상치로 탐지하는 준지도(Semi-supervised) 학습 방법



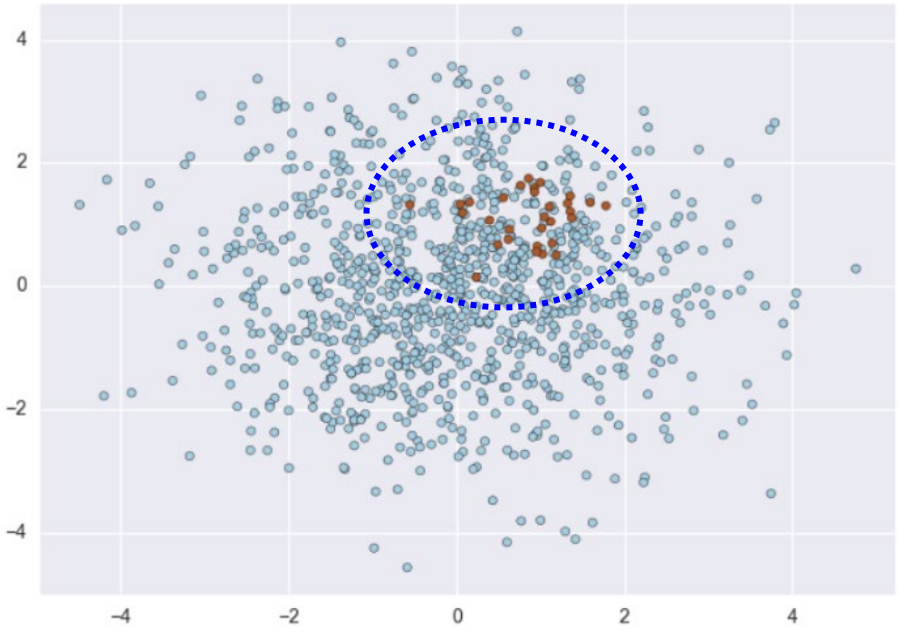
One-Class Classification Boundary

One Class Classification

■One-class Classification

□ 한 개의 클래스와 그 이외의 클래스로 나뉘어야 하는 경우 그 이외의 클래스의 범주가 너무 많아서 명확하게 데이터를 모으기가 어려운 경우

-> **하나의 Class 만 학습시켜서 분류를 할 수 있도록 하고 나머지는 outlier로 분류함**



데이터가
불균형한 상태

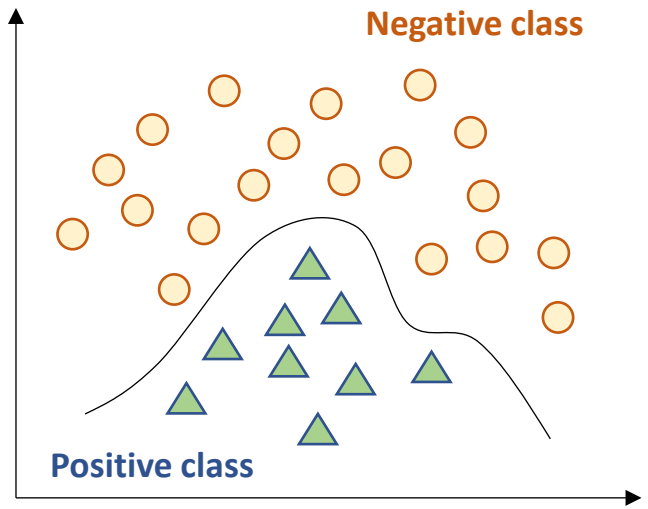
One Class Classification

■One-class SVM

□ One-class SVM은 SVM과는 달리 **비지도 학습 (unsupervised learning)**

□ 주어진 데이터를 잘 설명할 수 있는 최적의 support vector를 구하고 이 영역 밖의 데이터들은 outlier로 간주

- 분류를 할 때 Positive class에 대해서 수행하므로 Positive class 가 A라면 Negative class는 ~A가 됨



Two-class SVM

One-class SVM

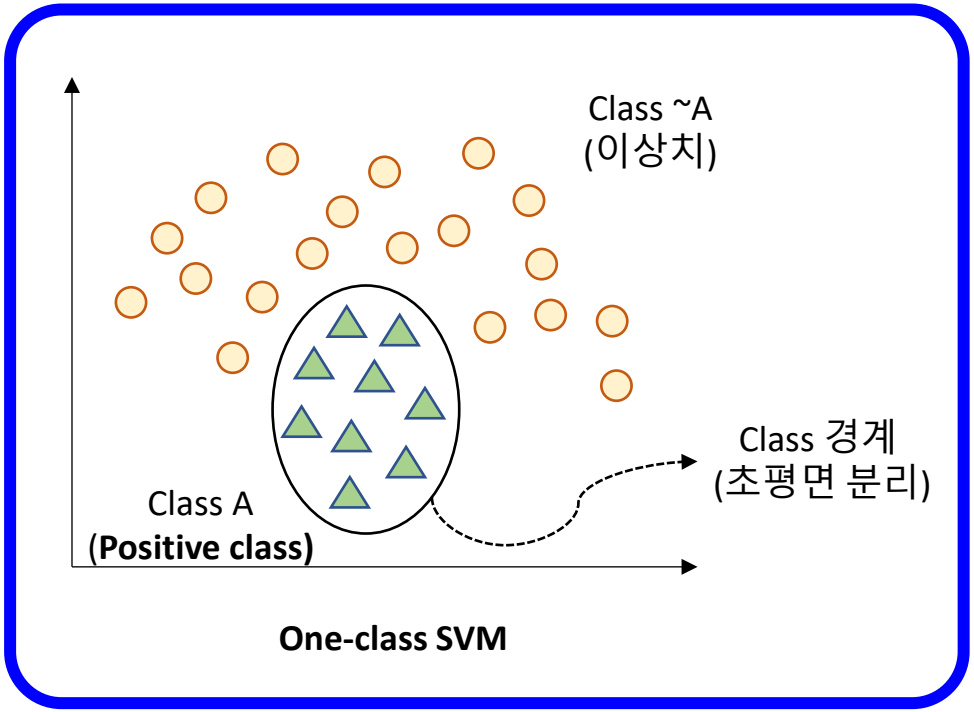
One Class Classification

■One-class SVM

□ One-class SVM은 SVM과는 달리 **비지도 학습 (unsupervised learning)**

□ 주어진 데이터를 잘 설명할 수 있는 최적의 support vector를 구하고 이 영역 밖의 데이터들은 outlier로 간주

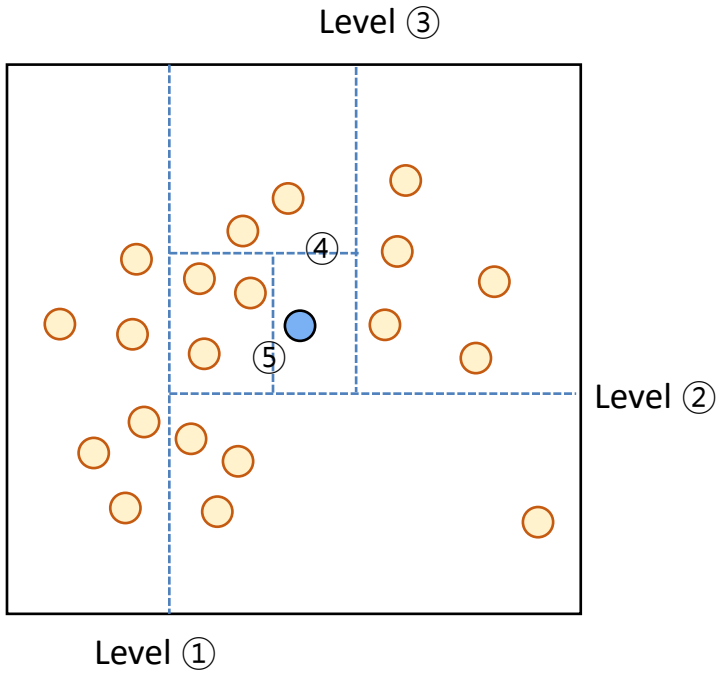
- 분류를 할 때 Positive class에 대해서 수행하므로 Positive class 가 A라면 Negative class는 ~A가 됨

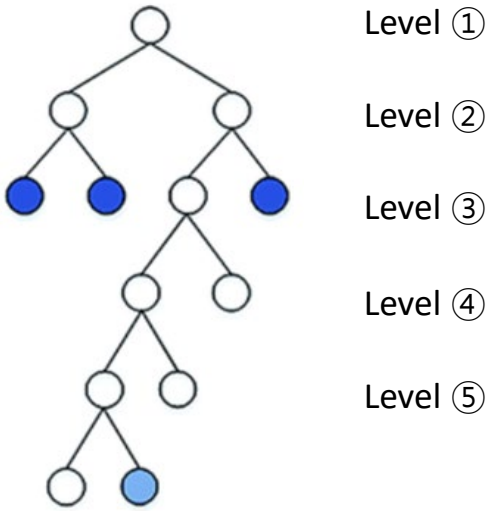


One Class Classification

■Isolation Forest

□ Isolation forest는 기본적으로 데이터셋을 **<u>의사결정나무 (Decision Tree) 형태로 표현</u>**
□ 랜덤으로 데이터를 split하여 모든 관측치를 고립 (분리) 시키며 구현
□ 변수가 많은 데이터에서도 효율적으로 작동할 수 있는 장점



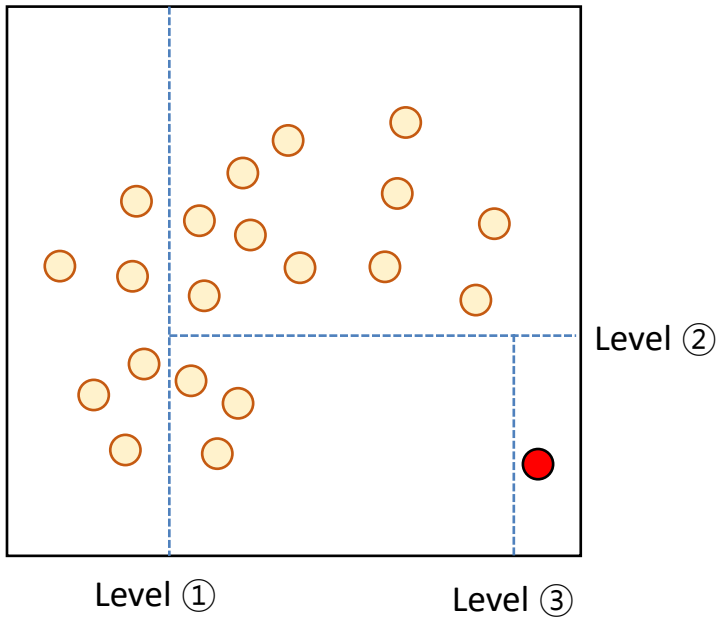


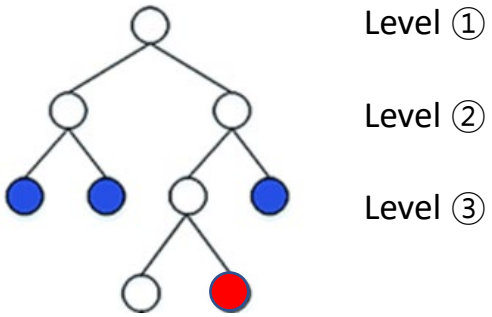
정상 데이터를 분리하는 경우, 5번의 split 필요

One Class Classification

■Isolation Forest

□ Isolation forest는 기본적으로 데이터셋을 **<u>의사결정나무 (Decision Tree) 형태로 표현</u>**
□ 랜덤으로 데이터를 split하여 모든 관측치를 고립 (분리) 시키며 구현
□ 변수가 많은 데이터에서도 효율적으로 작동할 수 있는 장점





이상치를 분리하는 경우, 3번의 split 필요

One Class Classification

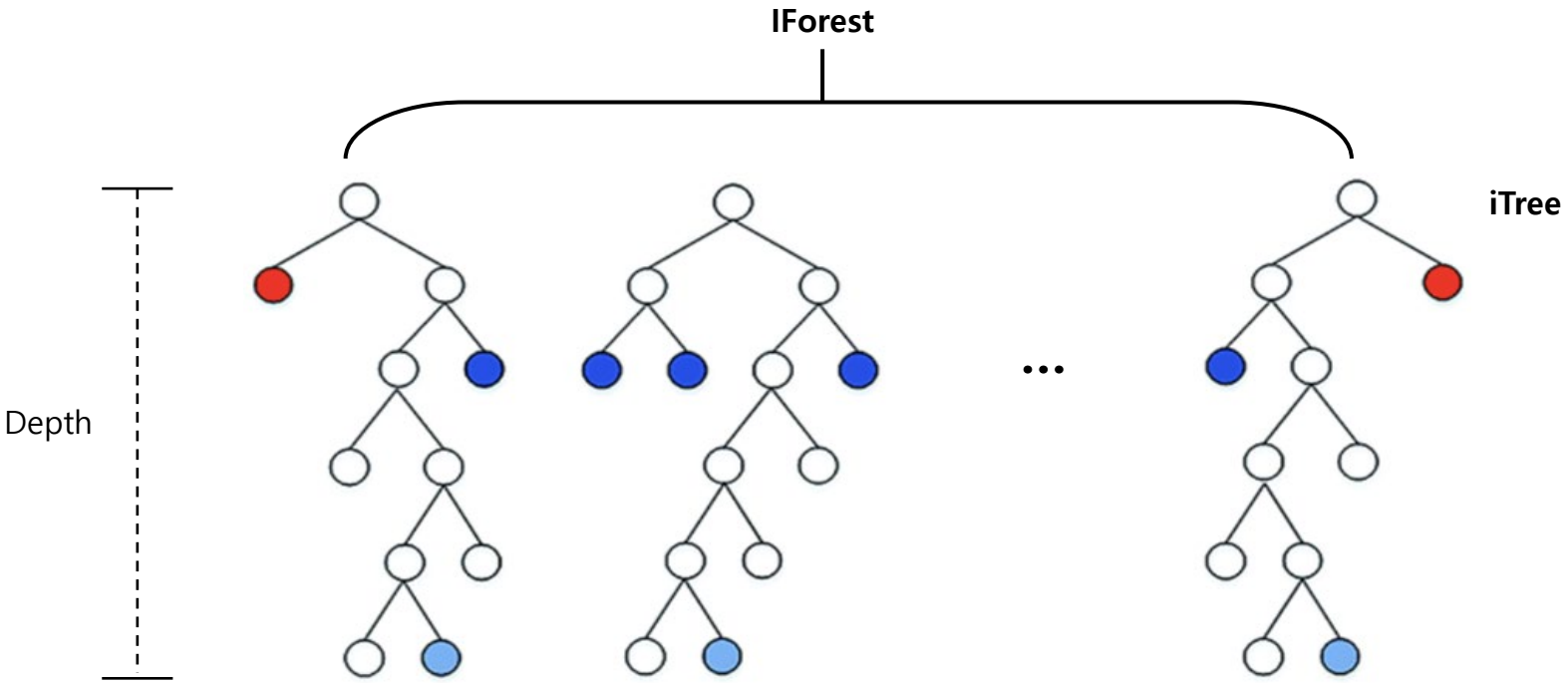
■Isolation Forest 학습 방법

□ 데이터의 Depth를 기준으로 정상값과 이상값을 분리

- **정상 데이터:** tree의 terminal node와 근접하며, 경로길이 (Depth)가 큼
- **이상치**: **tree의 root node와 근접**하며, **경로길이 (Depth)가 작음**

□ Random Forest와 비슷하게 작동

□ 여러개의 iTree로 구성



Thank you

KOREA
UNIVERSITY