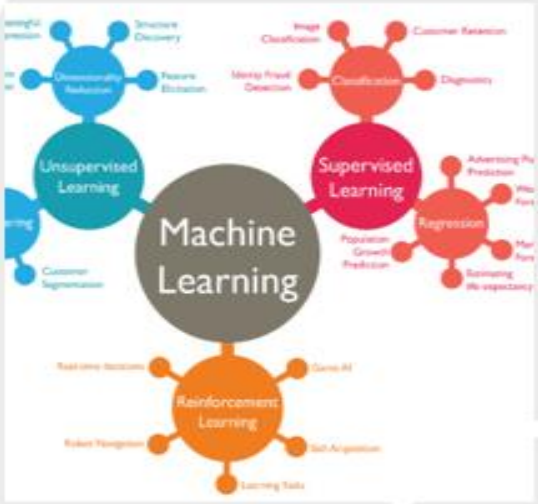


Machine Learning

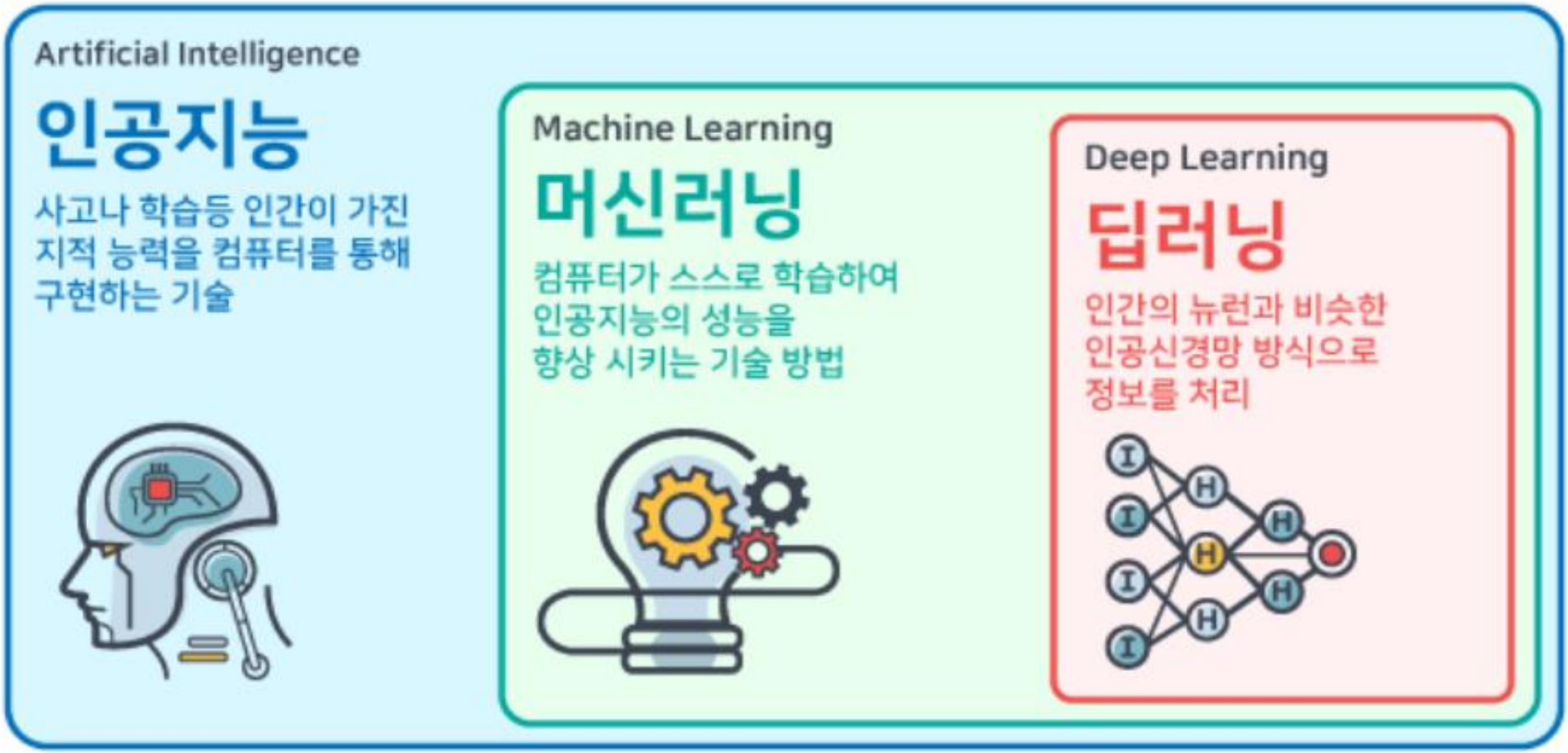
김진숙

KMR
한국경영인증원

머신 러닝
(Machine Learning)

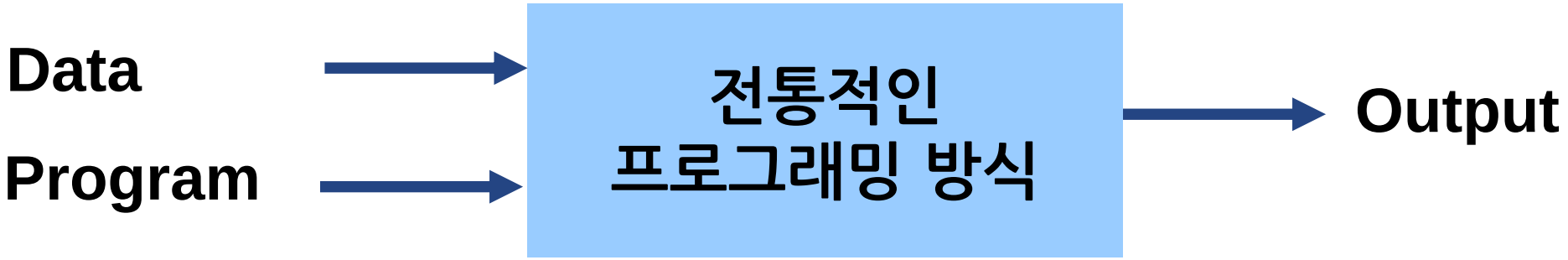
머신러닝(Machine Learning)

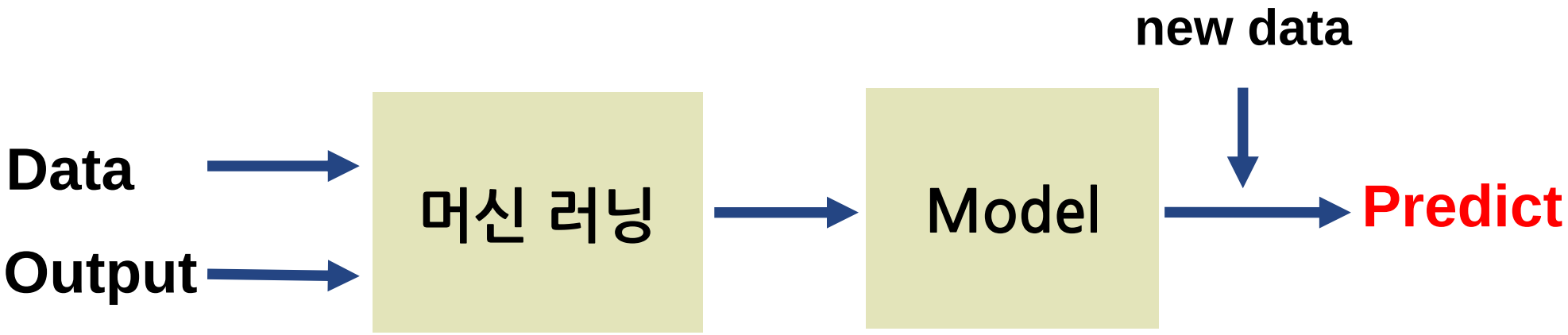
■ 인공지능(AI), 머신러닝(ML), 딥러닝(DL)



머신러닝(Machine Learning) 개념

- 전통적인 프로그래밍 방식과 머신 러닝과의 차이점





머신러닝(Machine Learning) 종류

기계학습
(Machine Learning)

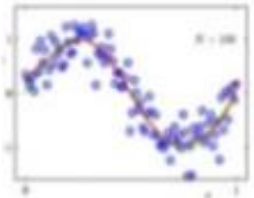
지도학습
(Supervised Learning)

입력(input)과 결과값(label)을 이용한 학습

분류
(classification)

- MNIST 필기체 인식
- 스팸 메일 분류

회귀
(regression)



- 주가 예측

비지도학습
(Unsupervised Learning)

입력(input)만 가지고 학습

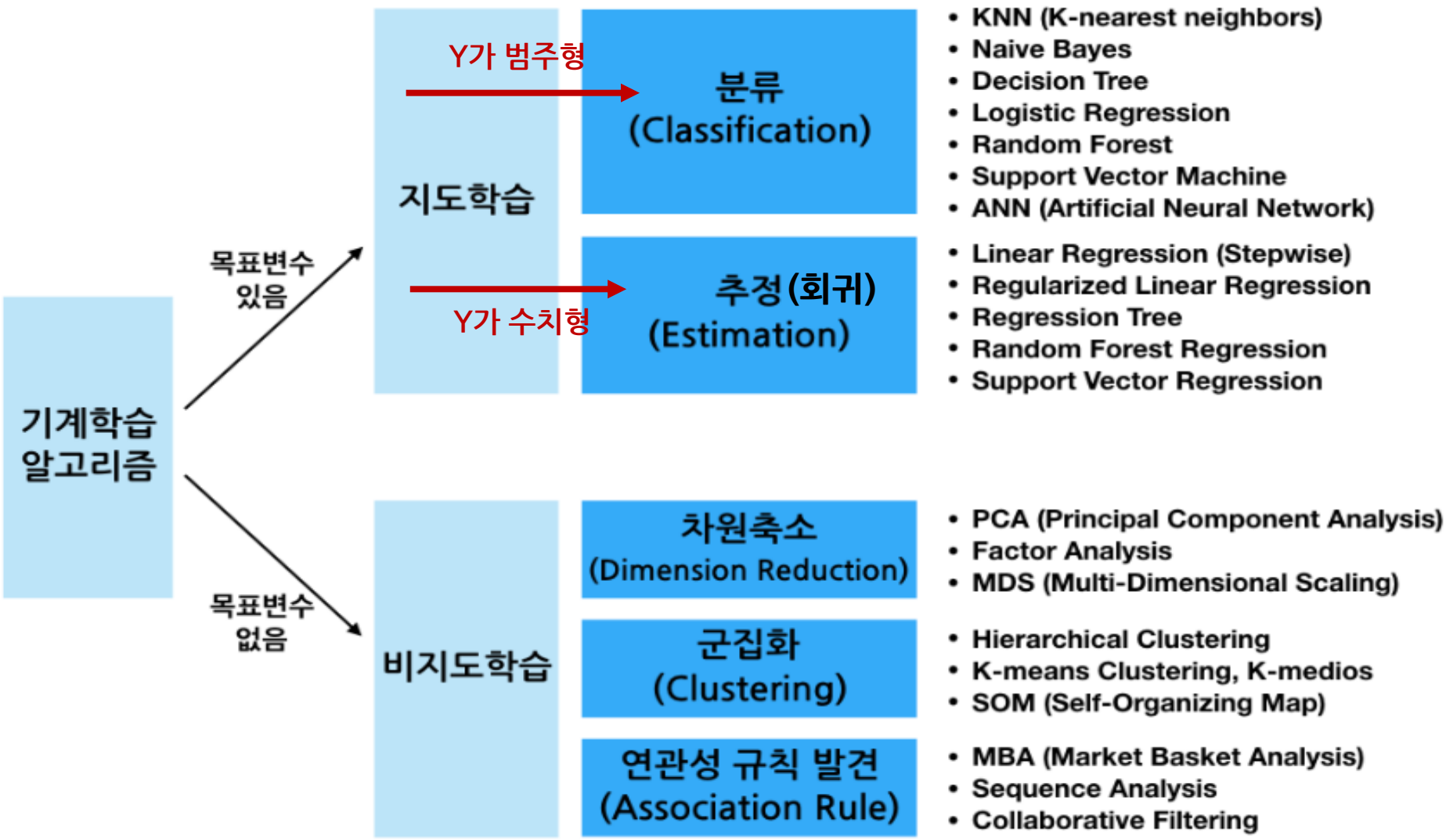
클러스터링
(clustering)

- 마케팅 고객 그룹화

강화학습
(Reinforcement Learning)

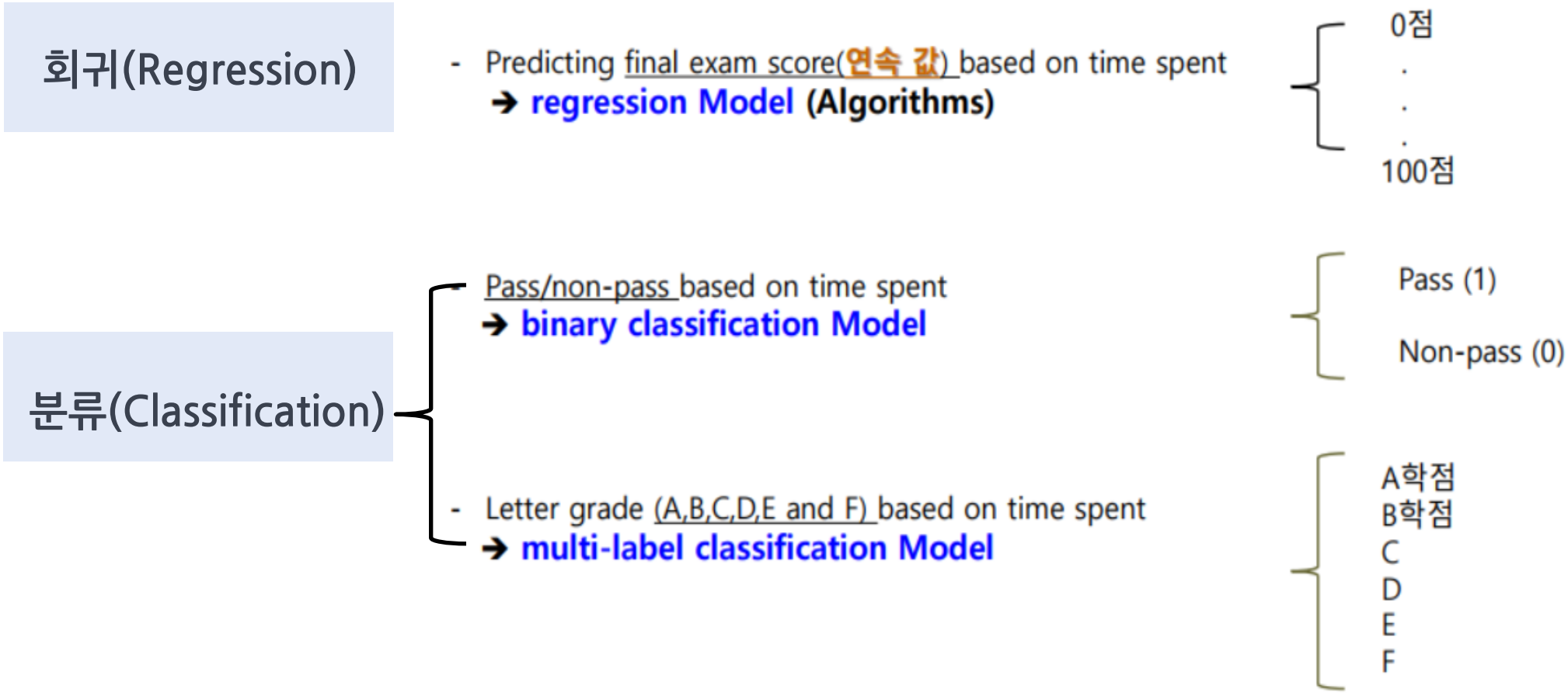
- 보상 시스템으로 학습
- 의사 결정을 위한 최적의 액션 선택

머신러닝(Machine Learning)의 종류



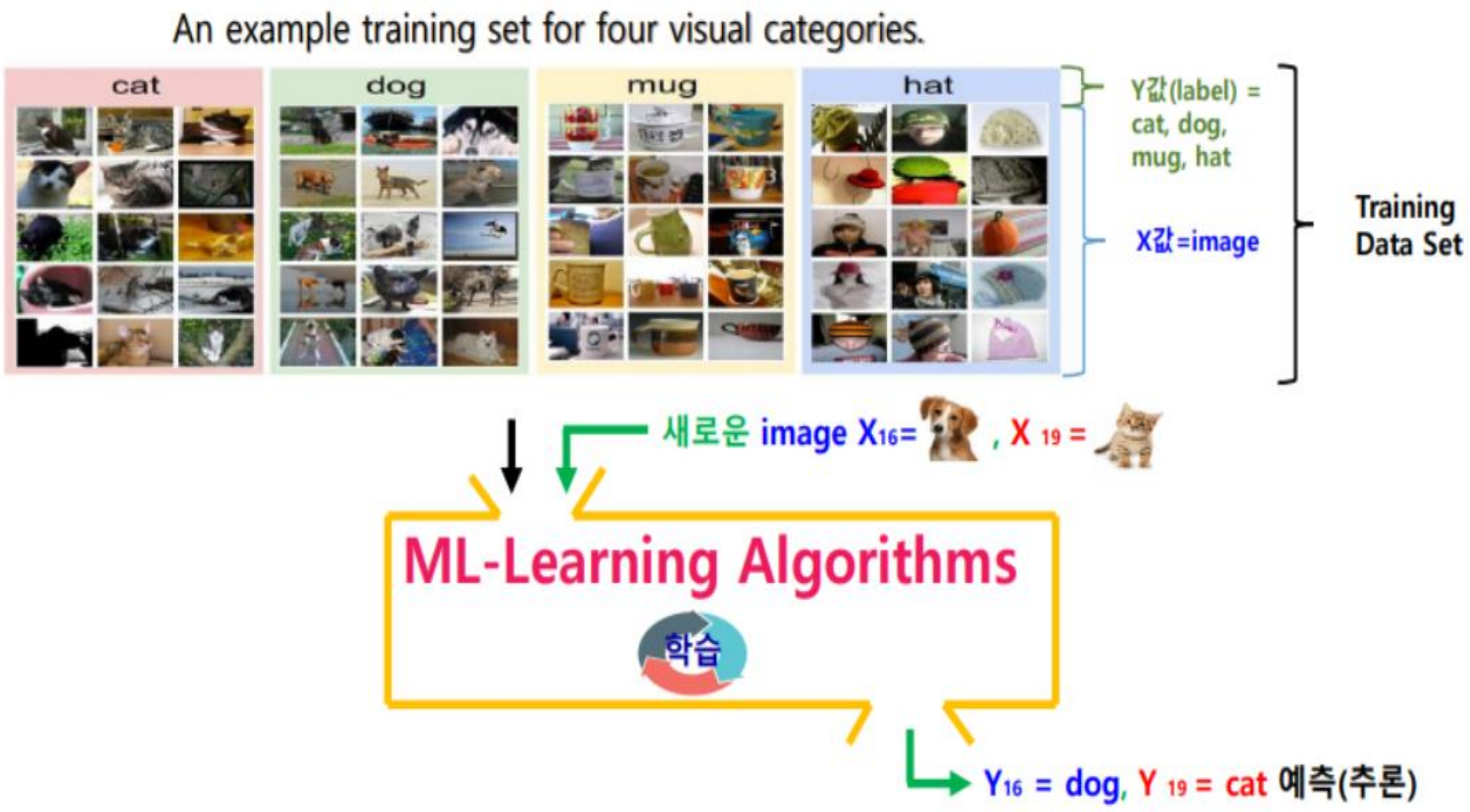
머신러닝의 종류

- **지도학습(Supervised Learning) 종류**



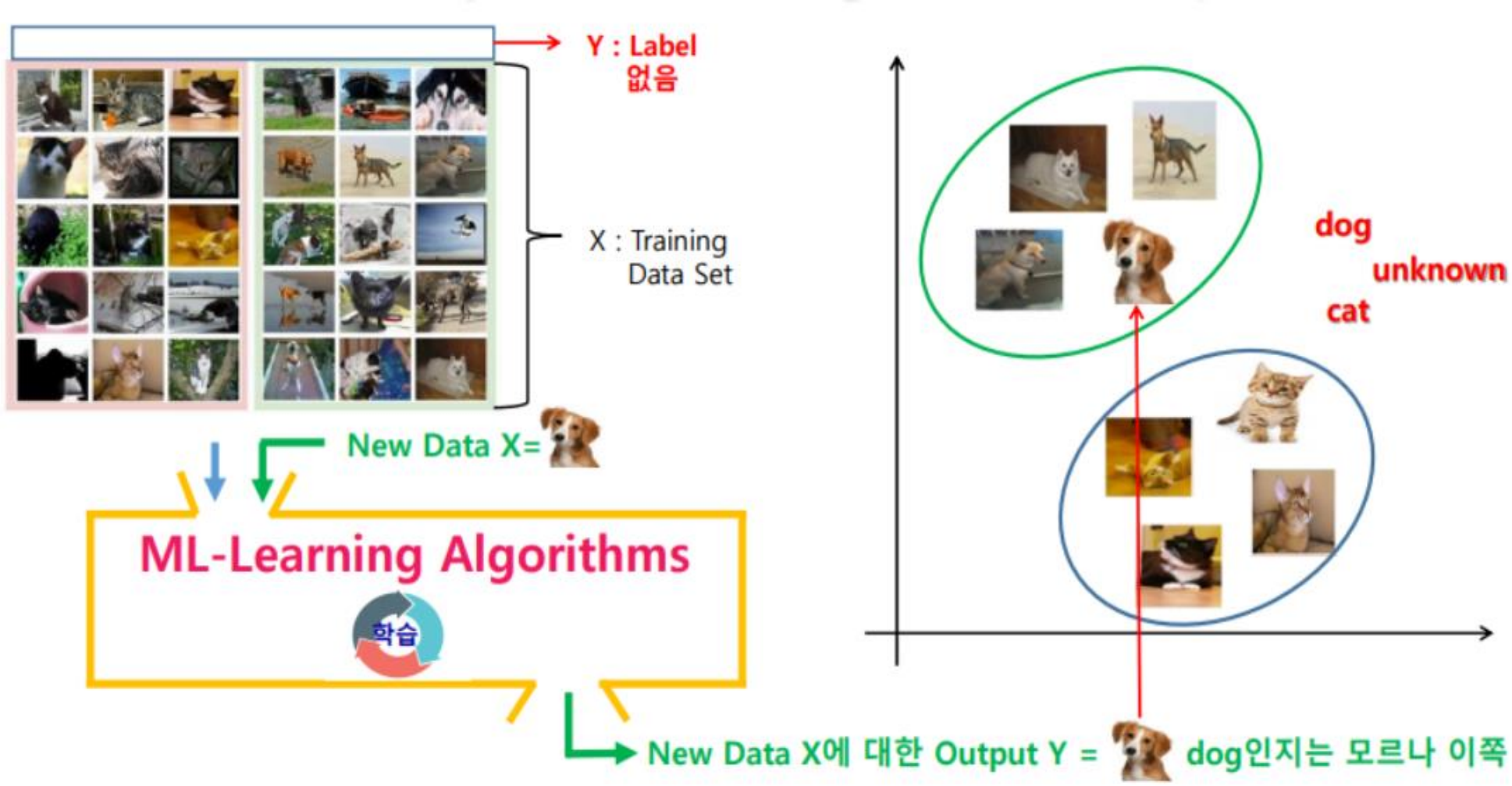
머신러닝(Machine Learning) 개념

- **Supervised Learning(예시 : image label)**



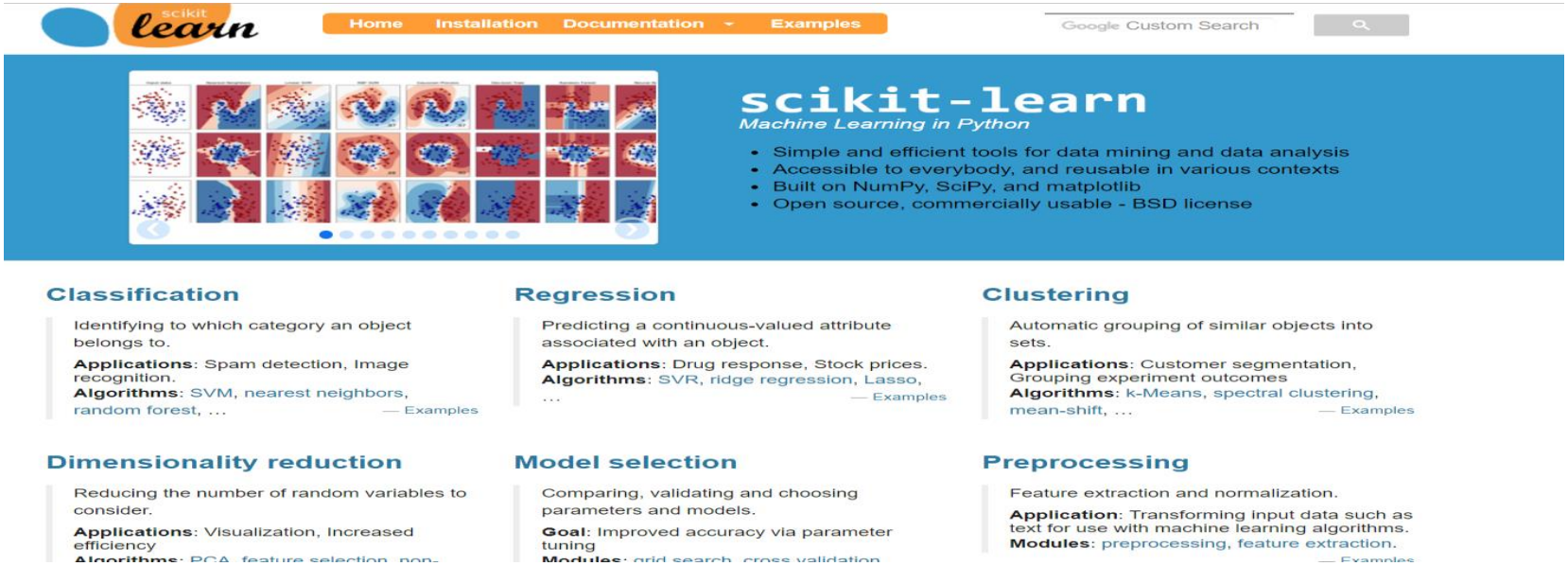
머신러닝(Machine Learning) 개념

- **Unsupervised Learning(예시 : 군집화(Clustering))**



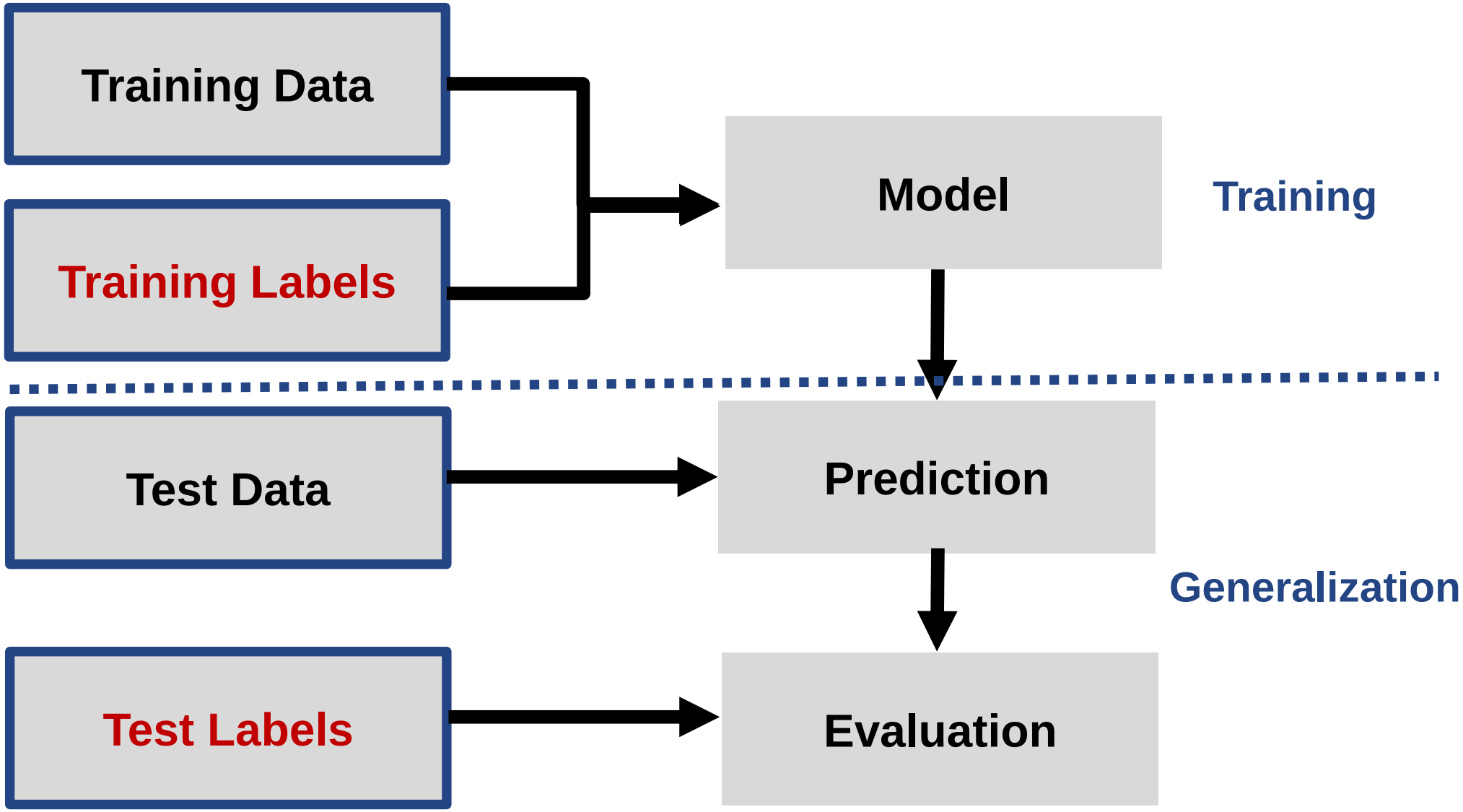
사이킷런 소개와 특징

- Scikit-learn 소개 : https://scikit-learn.org/stable/
  - 파이썬 머신러닝 라이브러리 중 가장 많이 사용되는 라이브러리

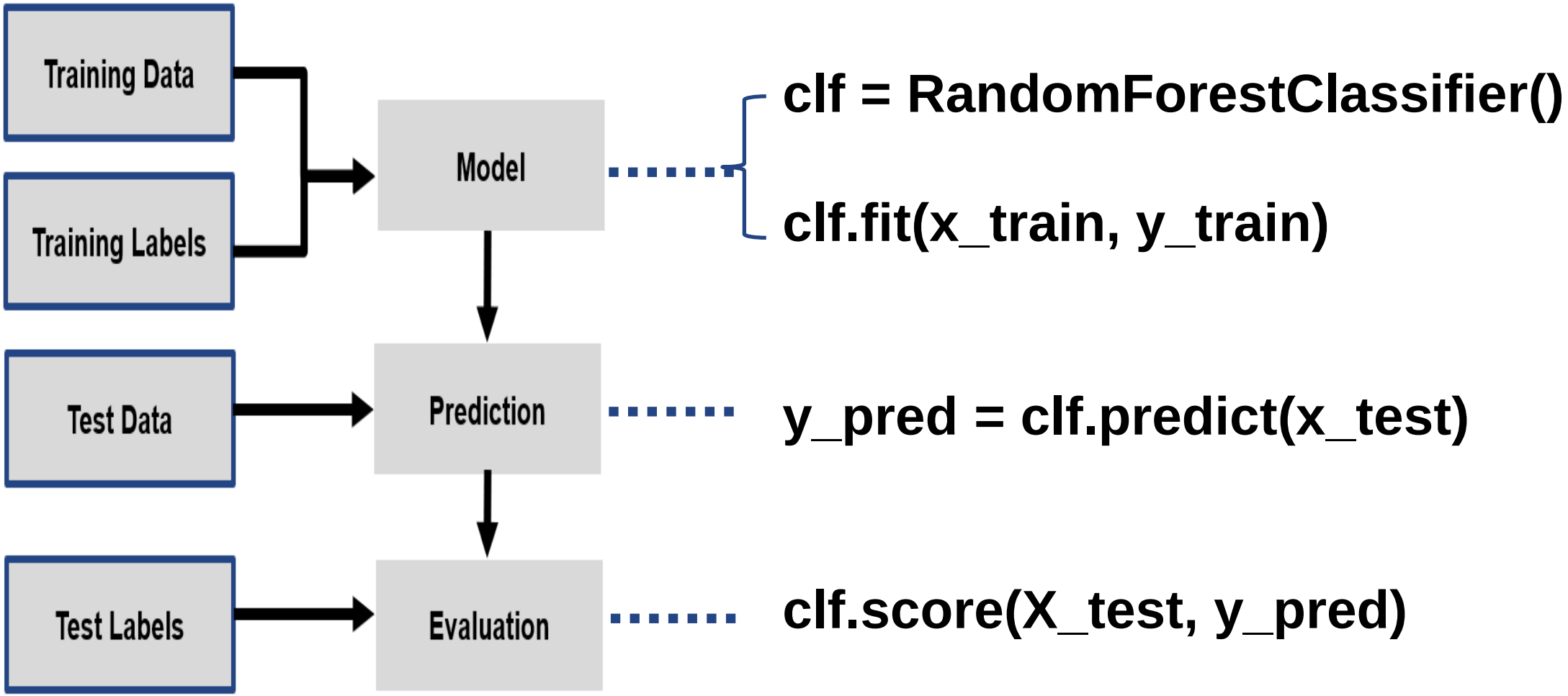


- 사이킷런의 특징
  - 파이썬 기반의 다른 머신러닝 패키지도 사이킷런 스타일의 API를 지향할 정도로 가장 파이썬스러운 API 제공
  - 머신러닝을 위한 매우 다양한 알고리즘과 개발을 위한 편리한 프레임워크와 API 제공
  - 오랜 기간 실전 환경에서 검증되었으며, 매우 많은 환경에서 성숙한 라이브러리
  - 주로 Numpy와 Scipy 기반 위에서 구축된 라이브러리

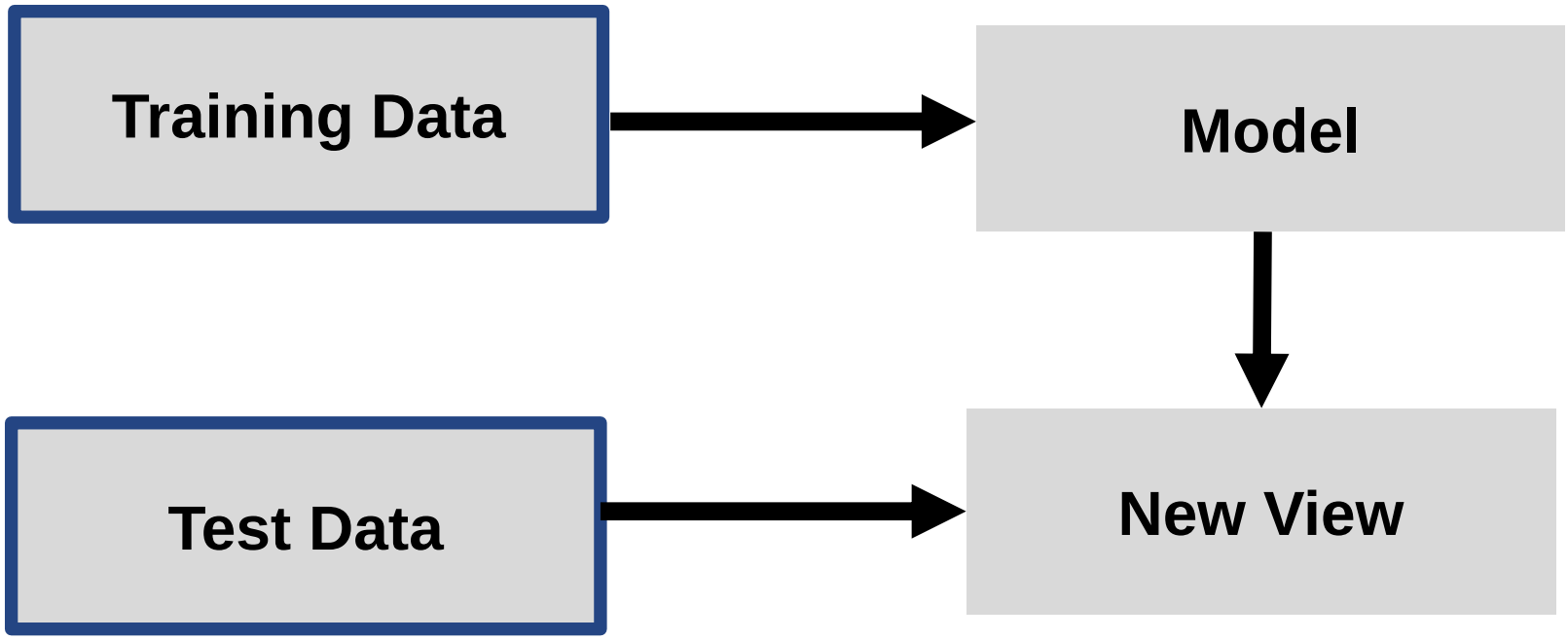
- 사이킷런의 지도 학습(Supervised Machine Learning)



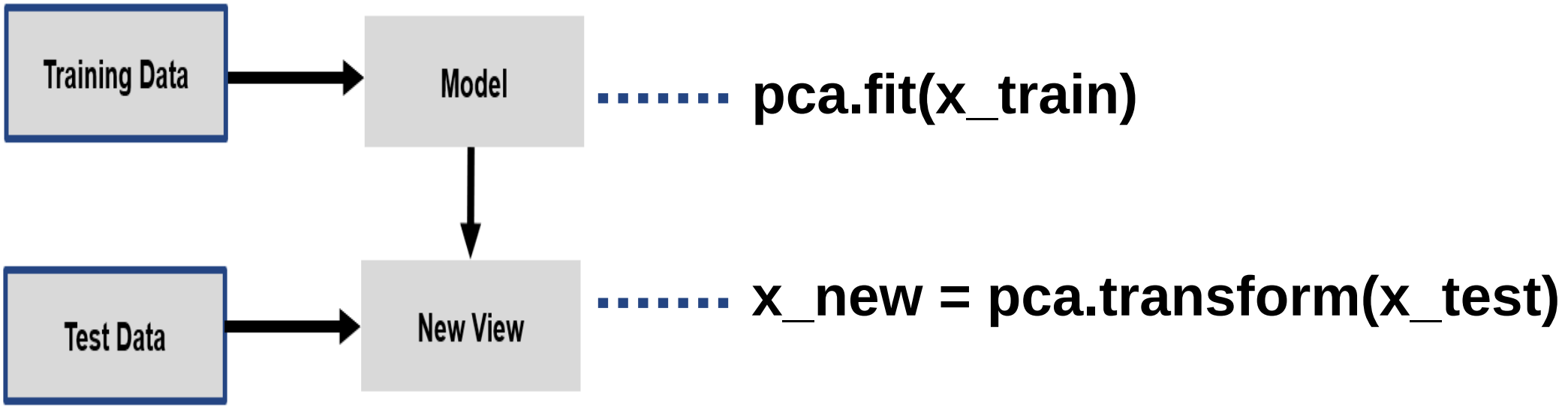
- 사이킷런의 지도 학습(Supervised Machine Learning)



- 비지도 학습(Unsupervised Machine Learning)

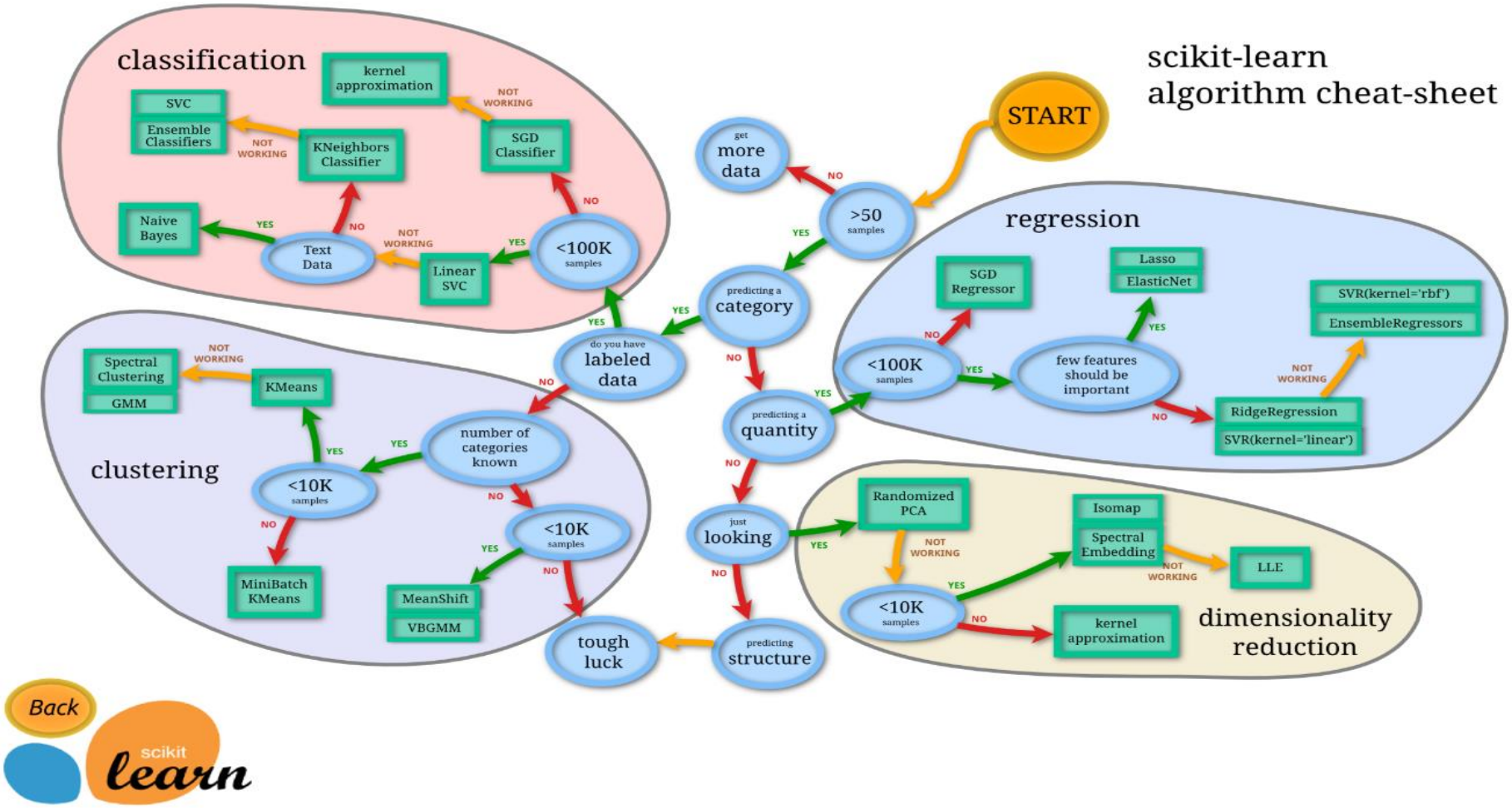


- 사이킷런의 비지도 학습(Unsupervised Machine Learning)



사이킷런 소개와 특징

- **Scikit-learn algorithm cheat-sheet**
  - https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

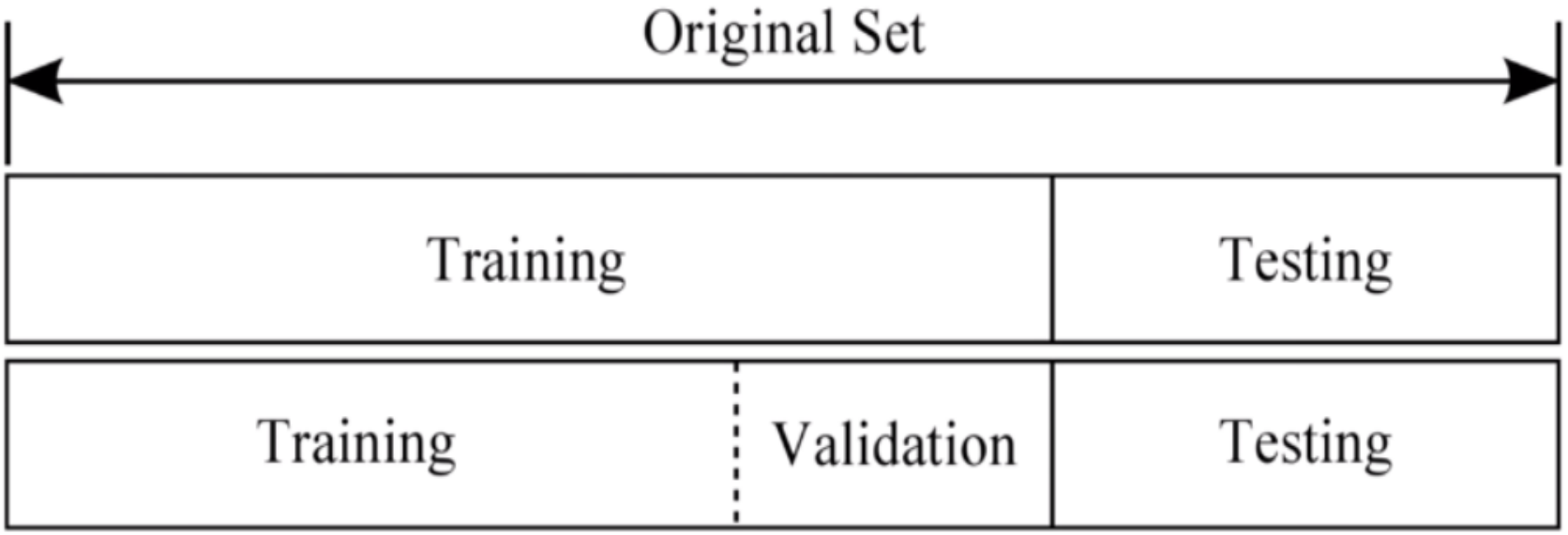


사이킷런 소개와 특징

- 내장 예제 데이터 셋 구성
  - data, target, feature_names, taget_names

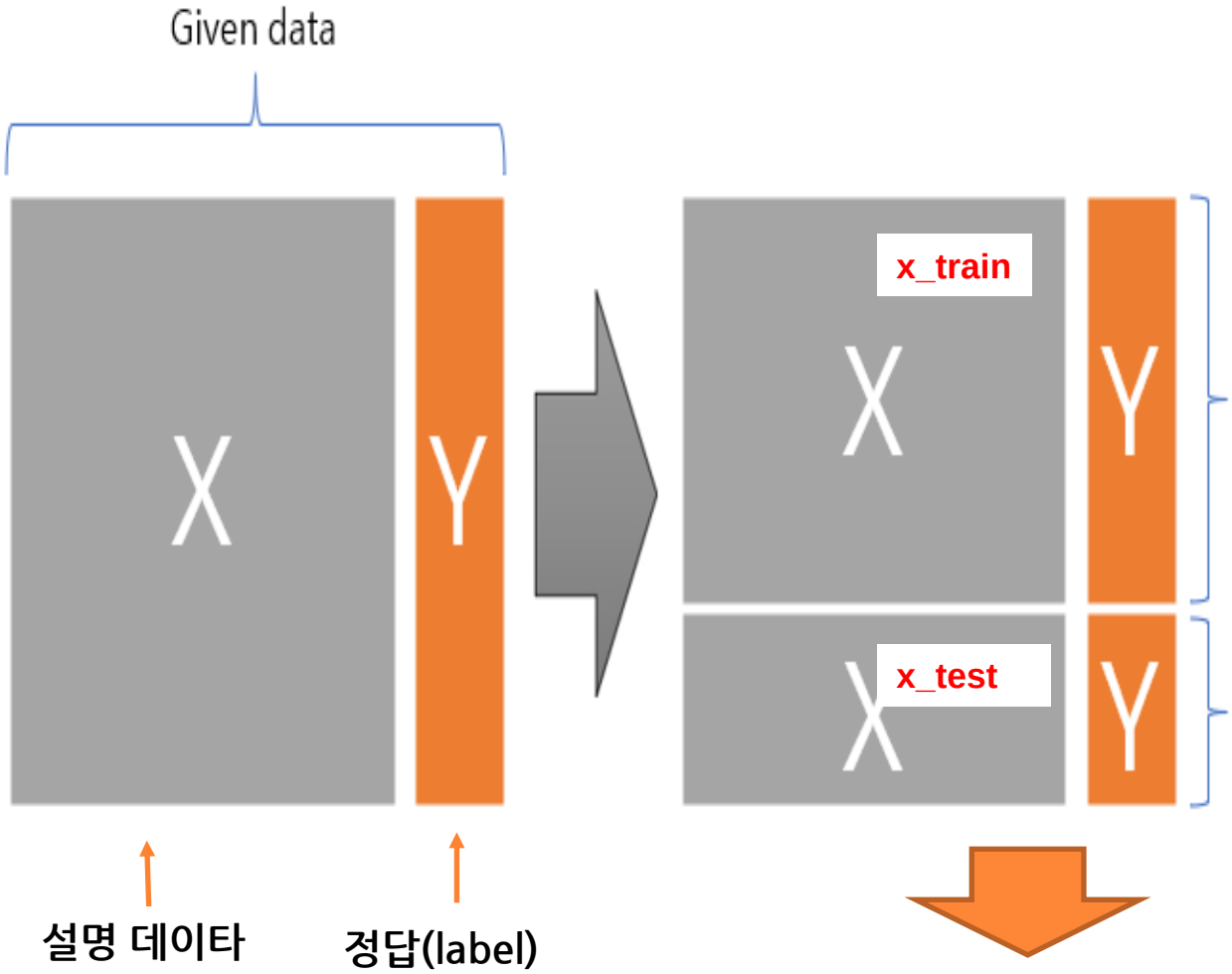
홀드아웃(Hold Out)

- 홀드아웃(Hold Out)
  - 데이터를 훈련 데이터와 테스트 데이터로 나눔
  - 모형의 최종 성능을 객관적으로 측정하기 위한 방법으로 트레이닝에 사용되지 않은 새로운 데이터(테스트 데이터)를 사용해서 예측한 결과를 기반으로 성능을 계산
  - 일정한 비율로 Train/Test의 비율로 나누어 사용(7:3, 8:2, 6:4)



홀드아웃(Hold Out)

- 홀드아웃(Hold Out)



사이킷런의 데이터 분류 : x_train, x_test, y_train, y_test

사이킷런 모듈

- **홀드아웃 : 학습 데이터와 테스트 데이터 분리 - train_test_split()**
  - sklearn.model_selection의 train_test_split() 함수

```
X_train, X_test, y_train, y_test = train_test_split( iris_data.data,
                                                     iris_data.target,
                                                     test_size=0.3,
                                                     random_state=2020)
```

- test_size : 전체 데이터에서 테스트에서 테스트 데이터 세트 크기를 얼마로 샘플링 할 것인가를 결정, 디폴트는 0.25, 즉 25% 입니다.
- train_size : 전체 데이터에서 학습용 데이터 세트 크기를 얼마로 샘플링 할 것인가를 결정, test_size parameter를 통상적으로 사용하기 때문에 train_size는 잘 사용되지 않는다.
- shuffle : 데이터를 분리하기 전에 데이터를 미리 섞을지를 결정, 디폴트는 True, 데이터를 분산시켜서 좀 더 효율적인 학습 및 테스트 데이터 세트를 만드는 데 사용
- random_state : random_state는 호출할 때마다 동일한 학습/데스트용 데이터 세트를 생성하기 위해 주어지는 난수 값, train_test_split()는 호출 시 무작위로 데이터를 분리하므로 random_state를 지정하지 않으면 수행할 때마다 다른 학습/테스트용 데이터를 생산한다.

데이터 분석의 실수

- 과소적합(Under Fitting), 과대적합(Overfitting)

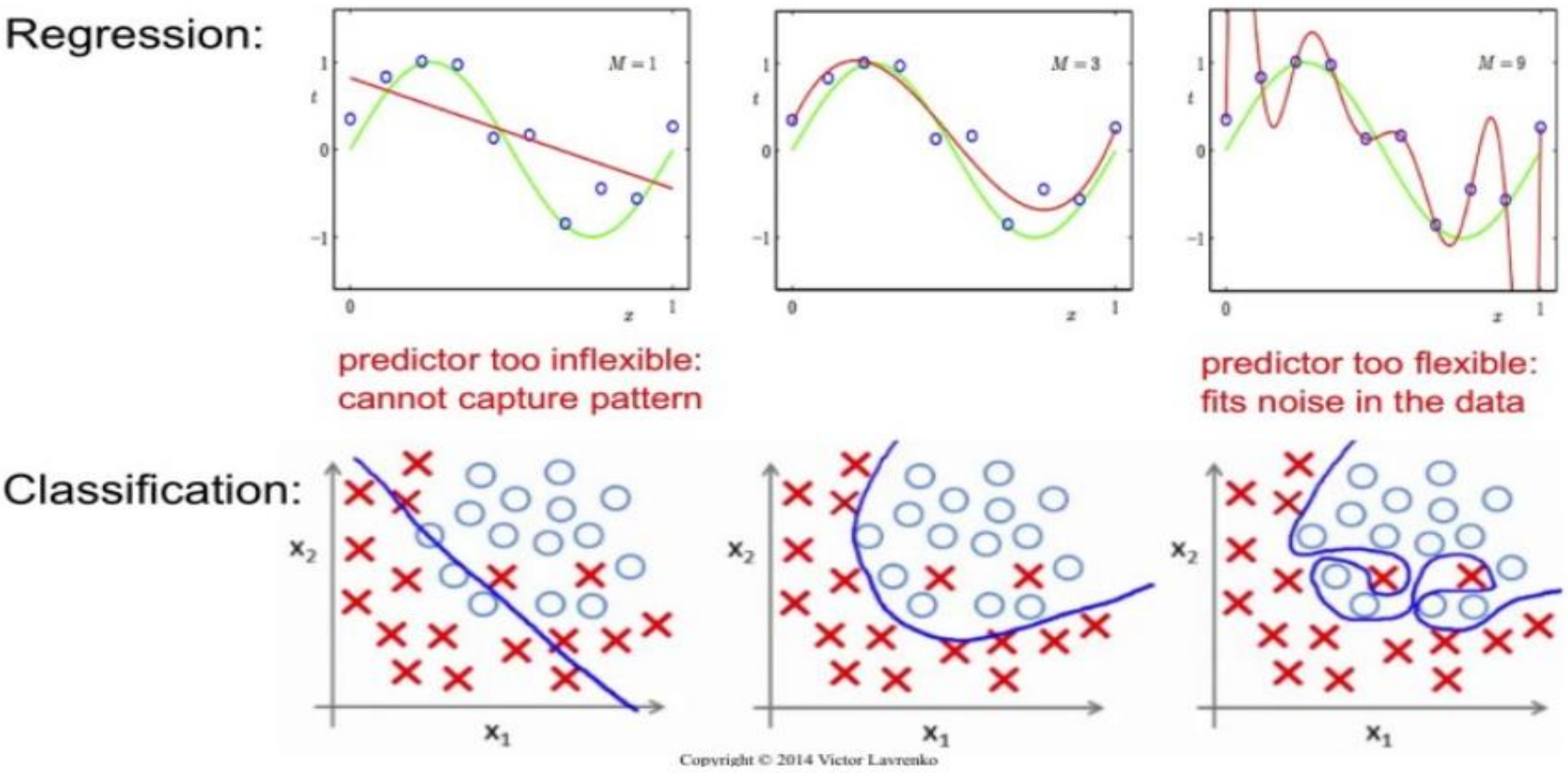


사진 출처: https://www.youtube.com/watch?v=dBLZg-RqoLg

■ 지도학습 – 분류 평가

- **혼동행렬**
  - ✓ 주로 분류 알고리즘이나 모델의 성능을 평가할 때 많이 사용

		Actual Class: Positive	Actual Class: Negative
Predicted Class	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

- **정확도**
  - ✓ 가장 기본이 되는 지표
  - ✓ 단순하게 전체 데이터 중에서 실제 데이터의 정답과 모델이 예측한 정답이 같은 비율

$$\text{accuracy} = \frac{\text{정확한 예측수}}{\text{총 예측수}} = \frac{TP + TN}{TP + TN + FP + FN}$$

지도 학습 : 분류

지도학습 : 분류

- **분류(Classification)란?**
  - 지도학습은 레이블(Label, 명시적인 정답)이 있는 데이터가 주어진 상태에서 학습하는 머신러닝
  - 주어진 데이터의 피처(Feature)와 레이블 값을 머신러닝 알고리즘으로 학습해 모델을 생성
  - 모델에 새로운 데이터 값이 주어지면 이 알 수 없는 레이블 값을 예측

- **분류 알고리즘**
  - 로지스틱 회귀(Logistic Regression) : 독립변수와 종속변수의 선형 관계성
  - 결정 트리(Decision Tree) : 데이터 균일도에 따른 규칙
  - 나이브 베이즈(Naïve-Bays) : 베이즈 통계와 생성 모델
  - 최소 근접(Nearest Neighbor) : 근접 거리를 기준으로 하는 모델
  - 신경망(Neural Network) : 심층 연결
  - 서포트 벡터 머신(Support Vector Machine) : 개별 클래스 간의 최대 마진을 효과적 활용

- **분류 모델 평가**
  - 정확도(Accuracy)

지도학습 : 분류

- **분류 모델링**
  - 머신러닝 모델 – 선형회귀모델, 랜덤포레스트모델
  - 딥러닝 모델 – 합성곱 신경망(CNN), 순환 신경망(RNN)

- **회귀 모델**
  - **선형 회귀 모델(Linear Regression)**
    - 종속변수(Y)와 한 개 이상의 독립변수(X)와의 선형 상관 관계를 모델링하는 회귀 분석 기법

    1) 단순 선형 회귀: $y = wx + b$ w : 계수(가중치), b : 절편(편향)

    2) 다중 선형 회귀: $y = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$

  - **선형 회귀의 비용 함수**

$$\text{Cost}_{tr} = \sum_i (y_I - \hat{y}_i)^2$$

$$\hat{y}_i = b + wx_i$$

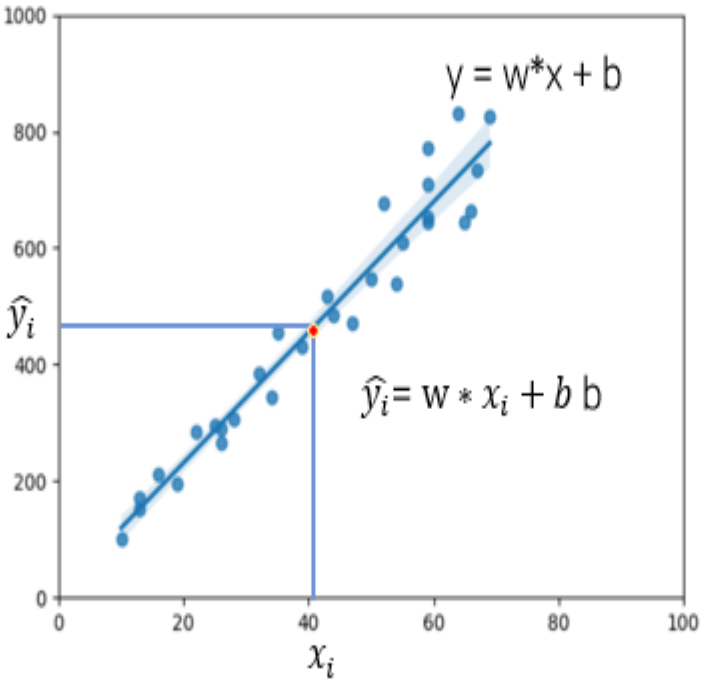
    - 실제 참값과 회귀 모델이 출력한 예측값 사이의 잔차의 제곱의 합을 최소화하는 w(계수)를 구하는 것이 목적

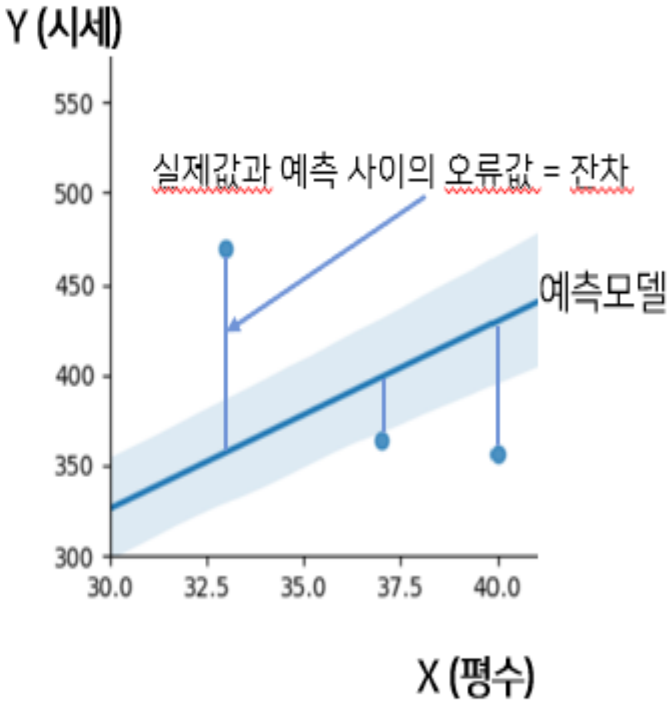
      -> Least Square, 최소 제곱법

지도학습 : 분류

- **선형 회귀 모델**
  - 주택 가격 예측: 주택 시세가 평수로만 결정된다고 가정

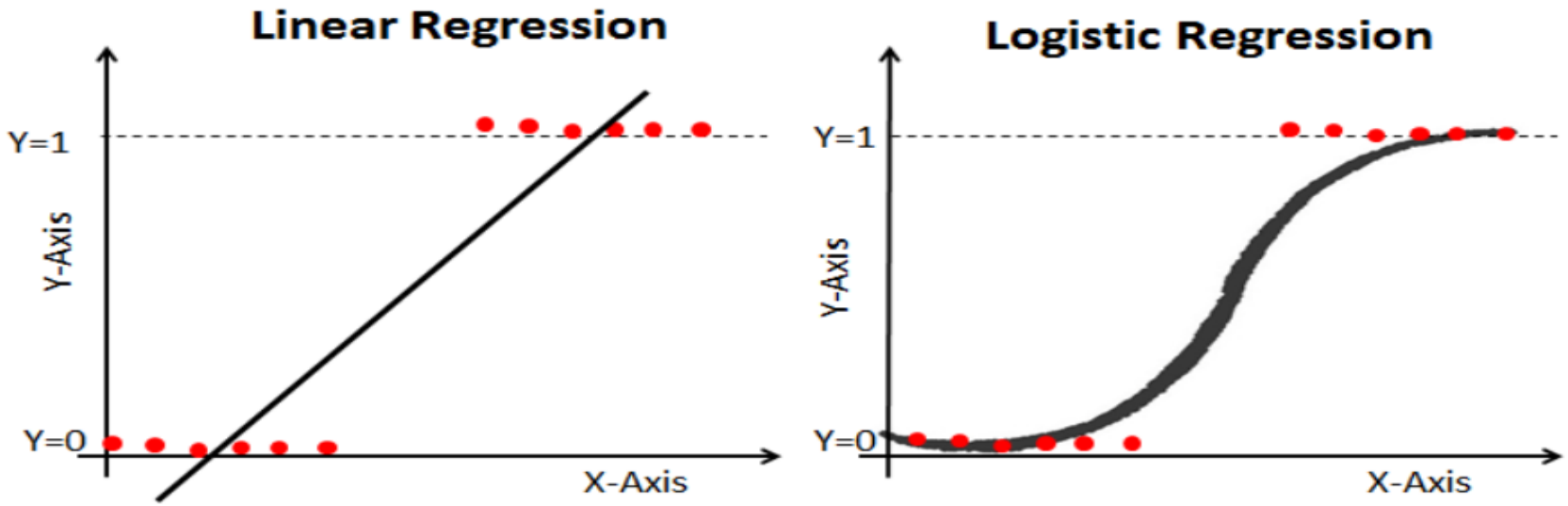
	Xi(평수)	Yi(시세)
0	10	102
1	22	284
2	32	384
3	64	832
4	50	547
NewData	Xi	Yj





지도학습 : 분류

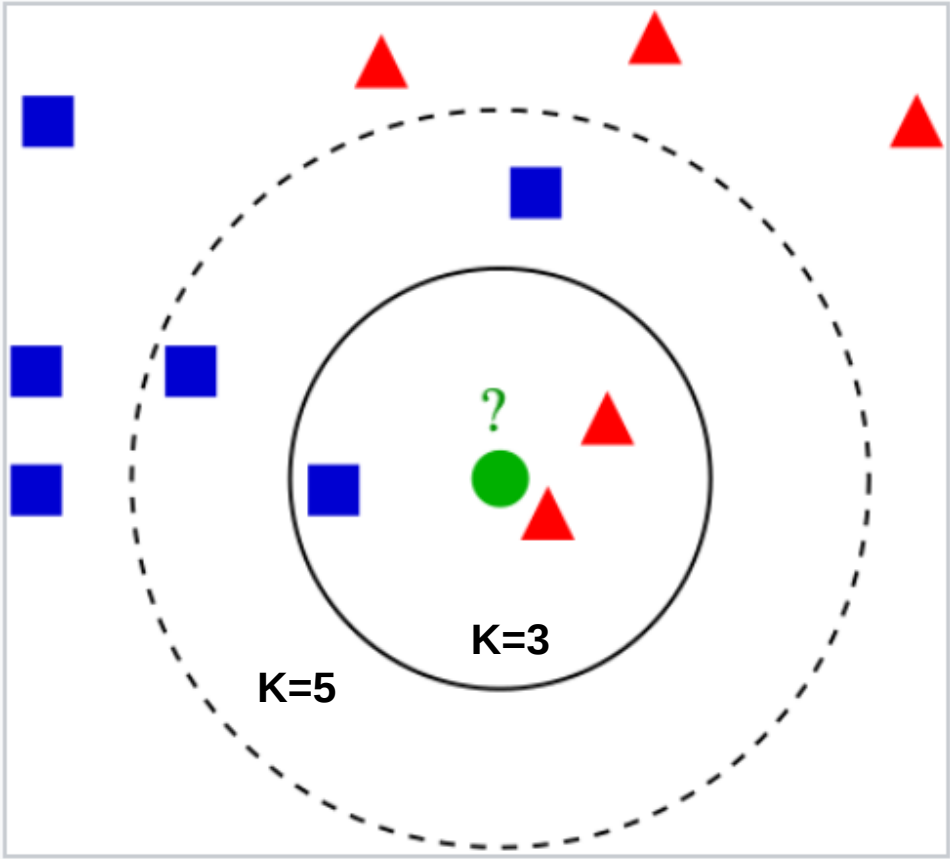
- **로지스틱 회귀 모델**
  - 선형 모델의 결과값에 로지스틱 함수를 적용하여 0 ~ 1 사이의 값을 갖게 해서 확률로 표현한 모델
  - 이렇게 나온 결과를 통해 1에 가까우면 정답이 1이라고 예측하고, 0에 가까울 경우 0으로 예측
  - 로지스틱 회귀 모델에 가지고 텍스트 분류 진행
  - 선형회귀 분석과 유사하지만 해당 데이터 결과가 특정 분류(Classification)로 나눔
  - 로지스틱 회귀에는 종속변수가 이항적 문제에 사용



$$\text{logistic function} = \frac{e^{\beta \cdot X_i}}{1 + e^{\beta \cdot X_i}}$$

지도학습 : 분류

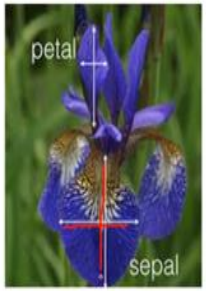
- K-최근접 이웃 알고리즘
  - 알고리즘의 훈련단계는 오직 훈련 표본이 특징 벡터와 항목 분류명으로 저장하는 것
  - k의 역할은 몇 번째로 가까운 데이터까지 살펴볼 것인가를 정한 숫자



지도학습 : 분류

- 실습 예제) 붓꽃 데이터 분류 - iris dataset

붓꽃 데이터 피처



- Sepal length
- Sepal width
- Petal length
- Petal width

붓꽃 데이터 품종(레이블)

피처 / 레이블

학습 데이터

번호	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
.....					....
50	6.4	3.5	4.5	1.2	Versicolor
.....					.....
150	5.9	3.0	5.0	1.8	Virginica

학습 데이터로 모델 학습

모델 학습

테스트 데이터

번호	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류는?
1	5.1	3.5	1.4	0.2	?
.....					?
50	6.4	3.5	4.5	1.2	?

학습 모델 통해 테스트 데이터의 레이블 값 예측

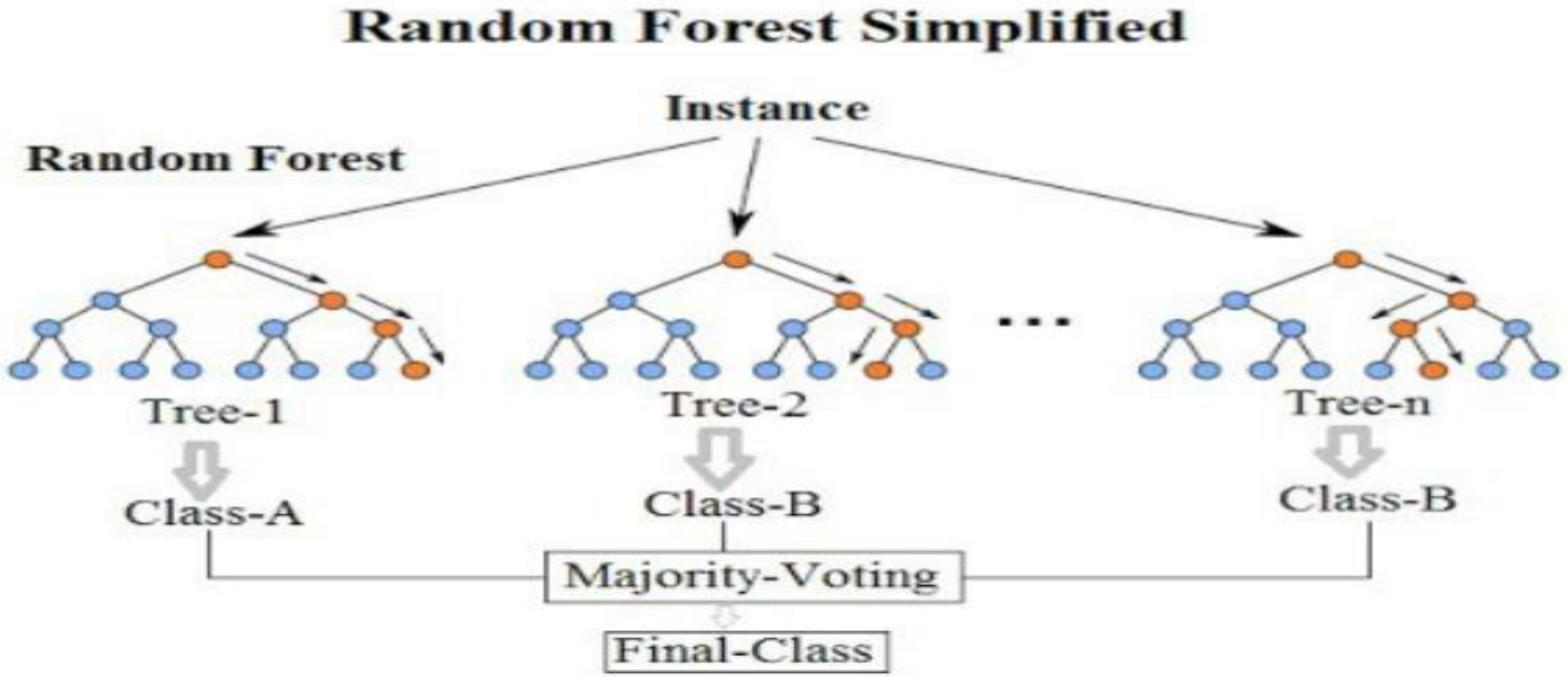
학습된 모델로 테스트 데이터 예측

예측된 레이블 값과 실제 레이블 값 예측 정확도 평가

평가

지도학습 : 분류 앙상블 학습

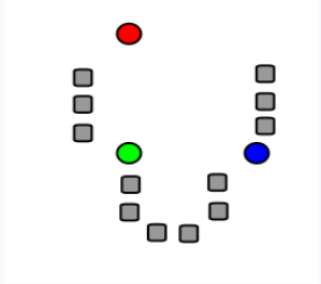
- 랜덤 포레스트(Random Forest)
  - 배깅의 가장 대표적인 알고리즘
  - 앙상블 알고리즘 중 비교적 빠른 수행 속도
  - 다양한 영역에서 높은 정확도
  - 결정 트리 기반의 알고리즘



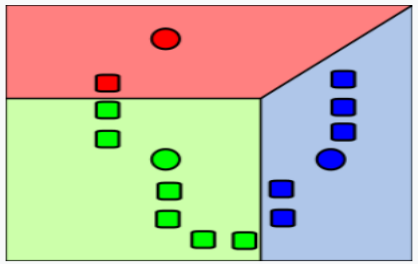
비지도 학습 : 군집화

비지도학습 : 군집화(Clustering)

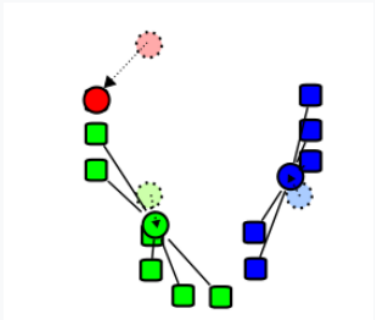
- K- 평균(K-means) 군집화



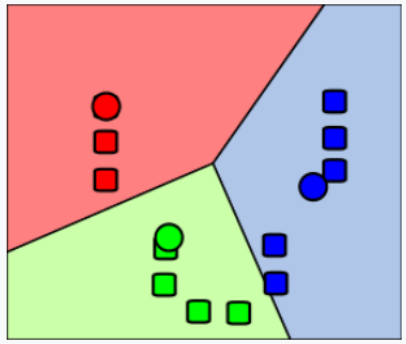
1) 초기 k "평균값" (위의 경우 k=3) 은 데이터 오브젝트 중에서 무작위로 뽑힌다. (색칠된 동그라미로 표시됨).



2) k 각 데이터 오브젝트들은 가장 가까이 있는 평균값을 기준으로 묶인다. 평균값을 기준으로 분할된 영역은 보로노이 다이어그램 으로 표시된다..



3) k개의 클러스터의 중심점을 기준으로 평균값이 재조정된다.



4) 수렴할 때 까지 2), 3) 과정을 반복한다.