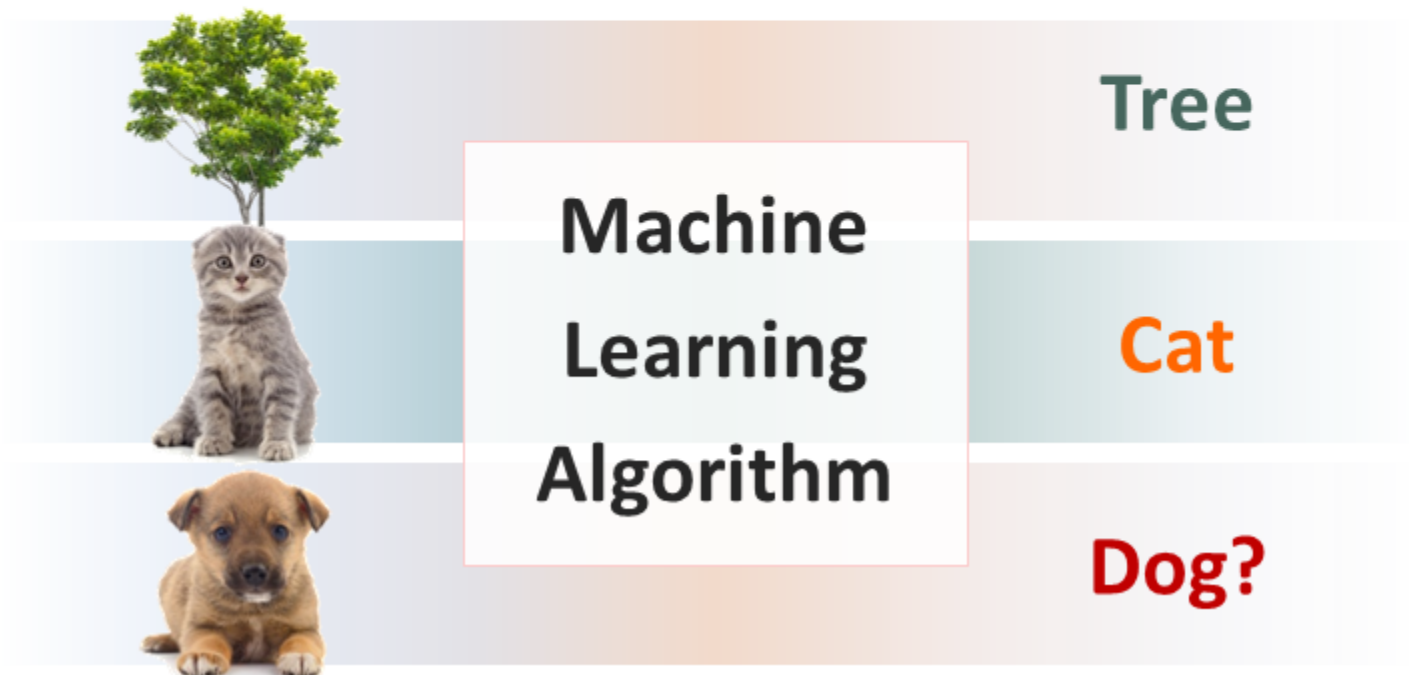


머신러닝

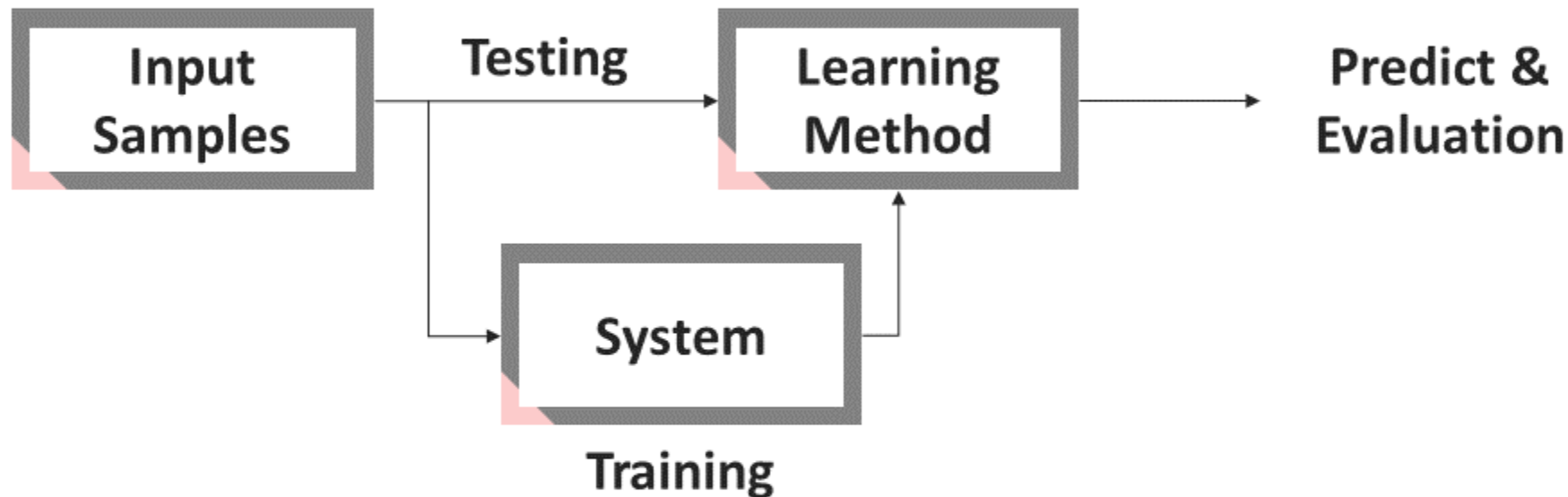


머신러닝이란 무엇인가?

- A branch of **artificial intelligence**, concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.
- As intelligence requires knowledge, it is necessary for the computers to acquire knowledge.

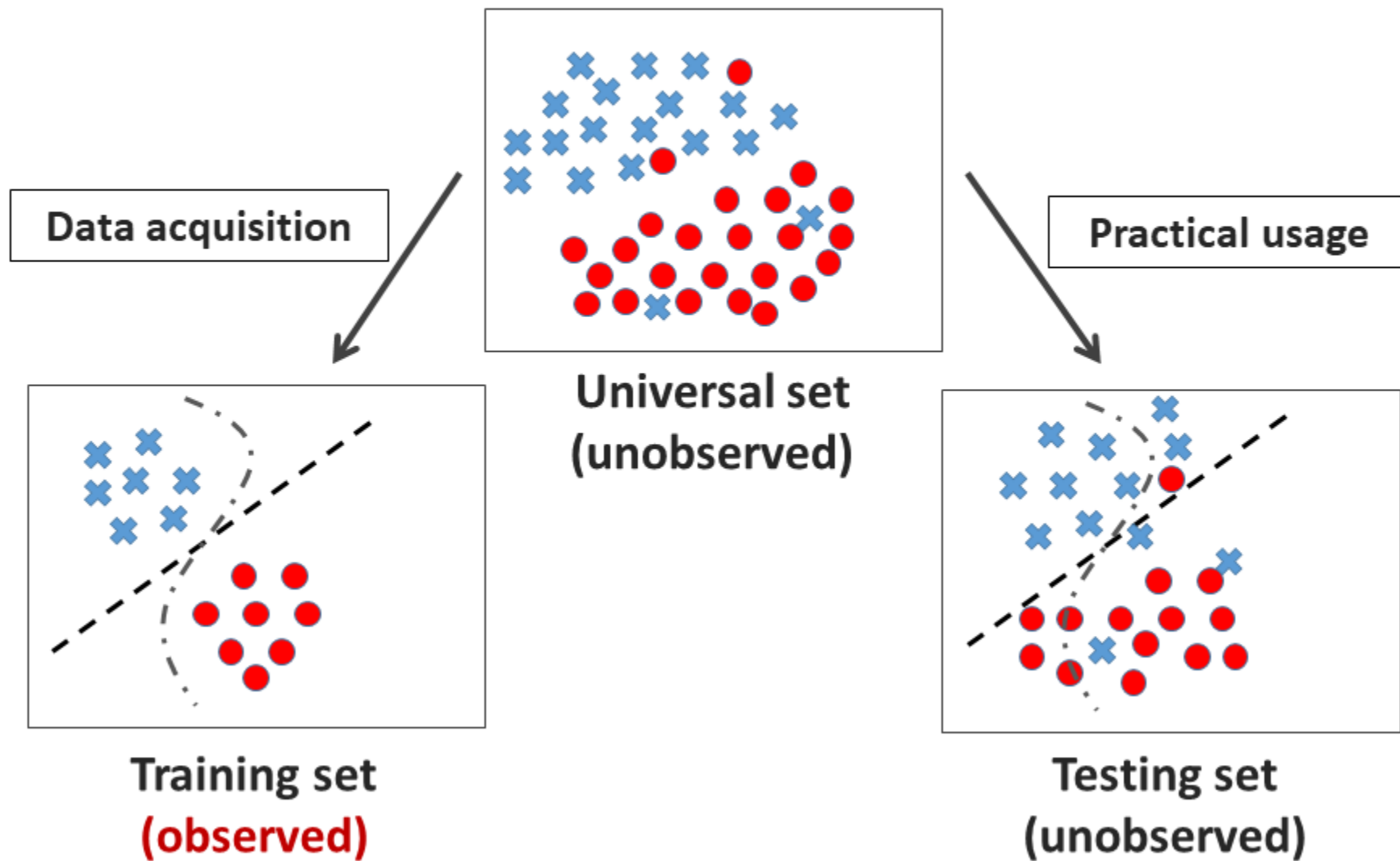


머신러닝이란 무엇인가?

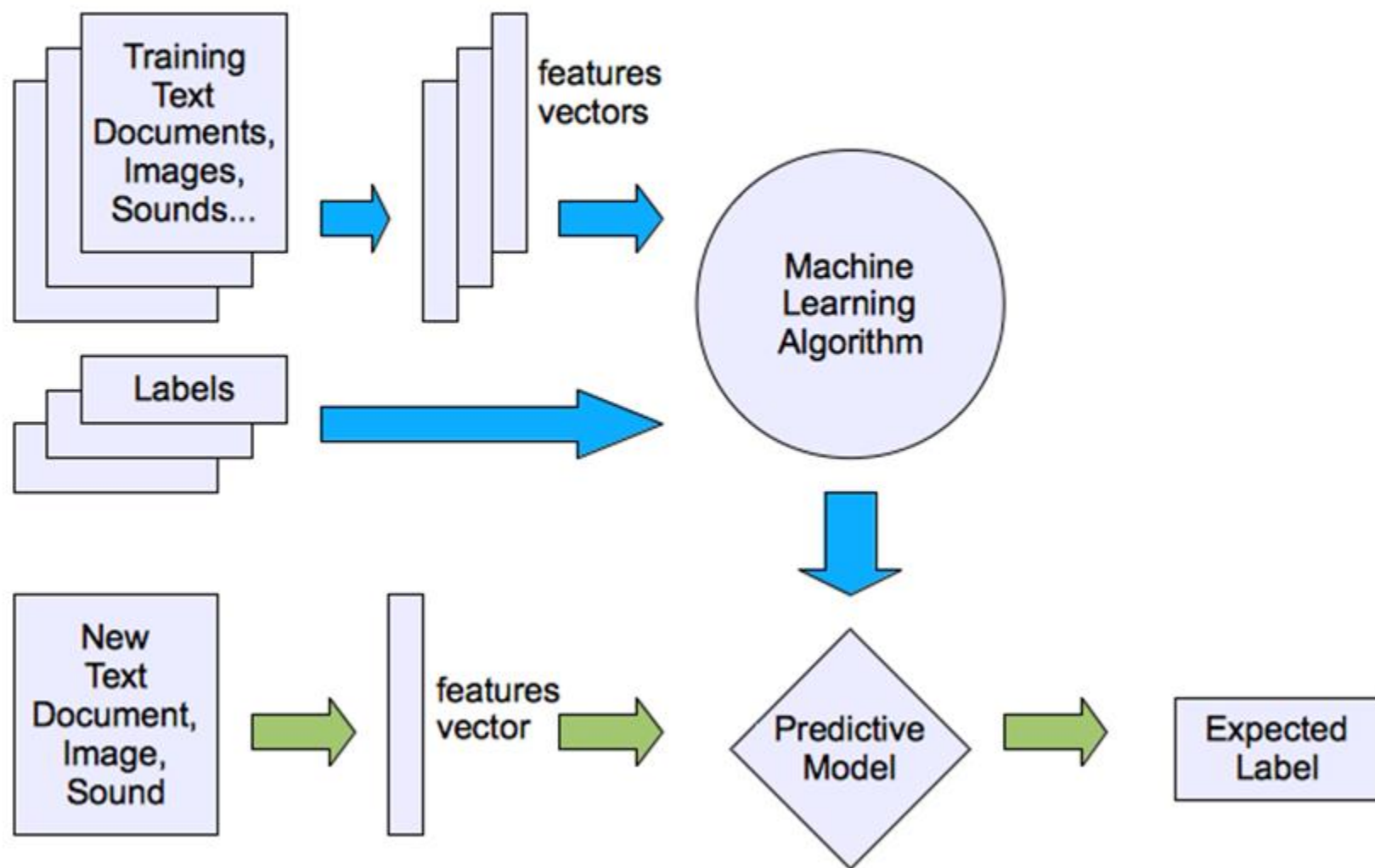


- Training is the process of making the system able to learn.
- Training set and testing set come from the same distribution.
- Need to make some assumptions or bias.

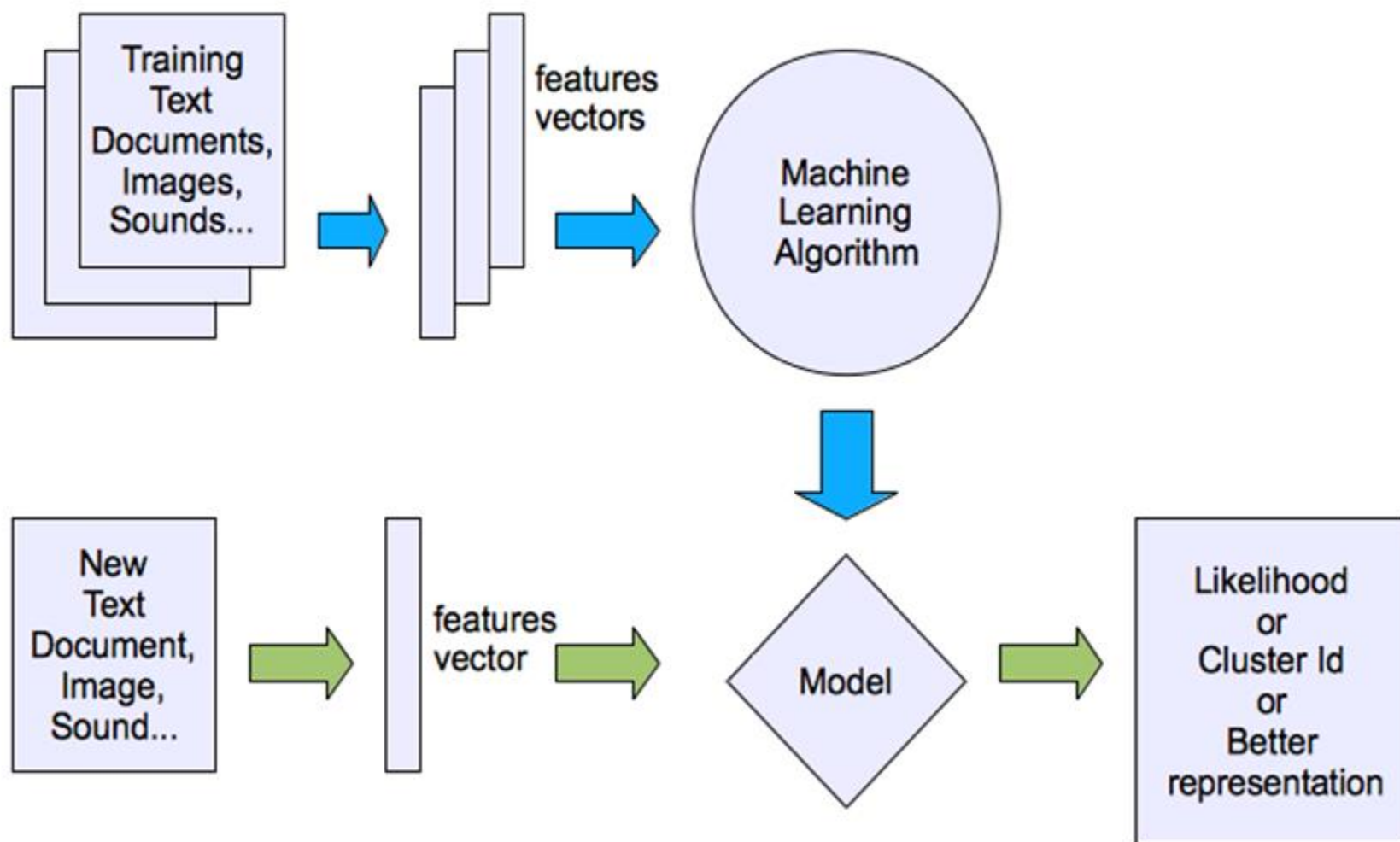
Training 과 Testing



Supervised Learning

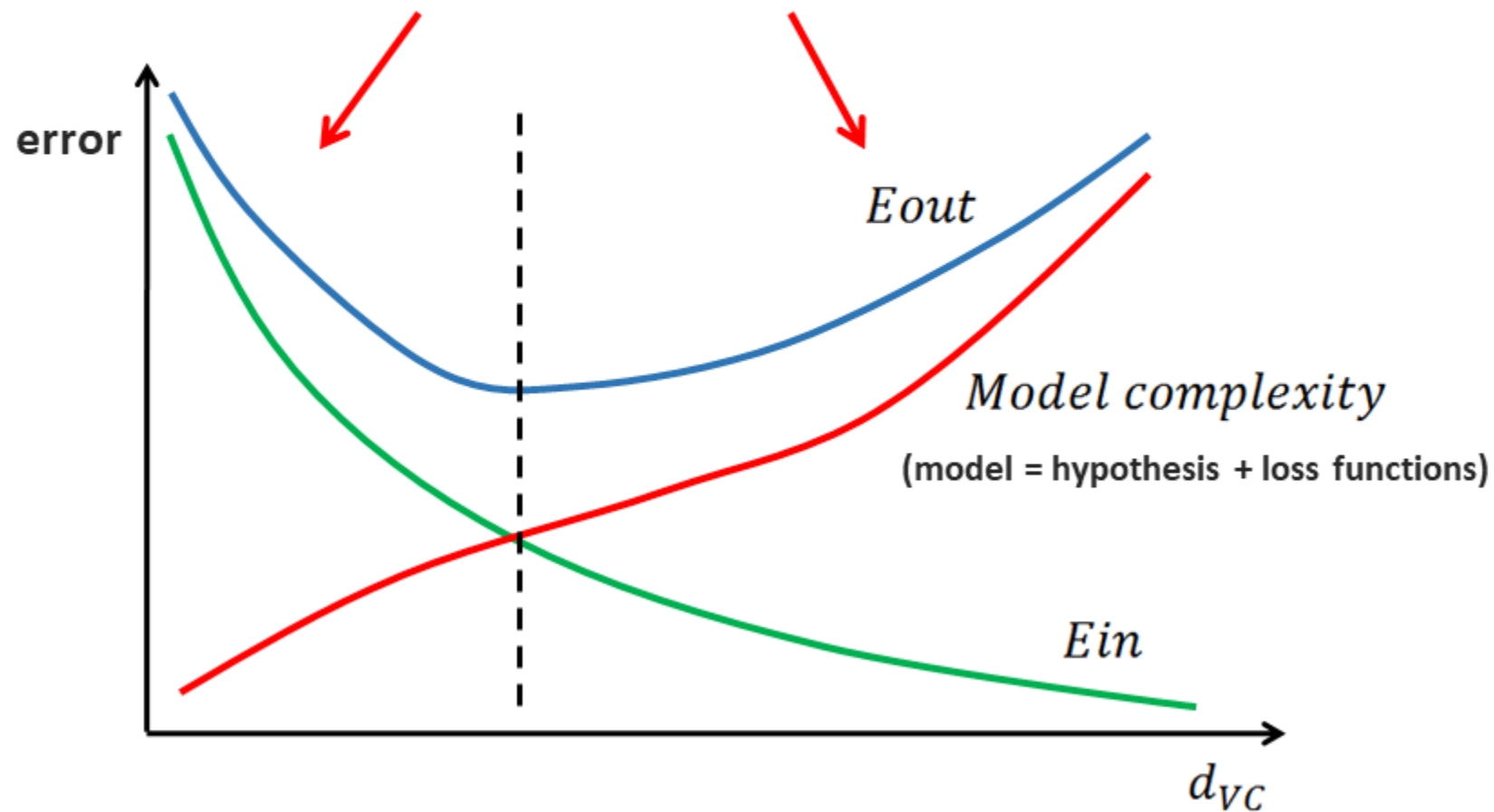


Unsupervised Learning



Overfitting, Underfitting

Under-fitting VS. Over-fitting (fixed N)



손실 함수(Loss Function) - MSE

- 머신 러닝 모델을 학습시키기 위해서는 적절한 파라미터 값을 알아내기 위해 서 현재 파라미터 값이 우리가 풀고자 하는 목적_{Task}에 적합한 값인지를 측정 할 수 있어야 합니다.
이를 위해 손실 함수_{Loss Function} $J(\theta)$ 를 정의합니다.
- 손실 함수는 여러가지 형태로 정의될 수 있습니다. 그 중 가장 대표적인 손실 함수 중 하나는 평균제곱 오차_{Mean of Squared Error(MSE)}입니다.
- MSE는 다음 수식으로 정의됩니다.

$$MSE = \frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

손실 함수(Loss Function) - MSE

- 예를 들어, 정답이 $y = [1, 10, 13, 7]$ 이고 우리 모델의 예측 값이 $\hat{y} = [10, 3, 1, 4]$ 와 같이 잘못된 값을 예측한다면 MSE 손실함수는 아래와 같이 35.375라는 큰 값을 갖게 될 것입니다.

$$MSE = \frac{1}{2 \times 4} \{ (10 - 1)^2 + (3 - 10)^2 + (1 - 13)^2 + (4 - 7)^2 \} = 35.375$$

- 하지만, 정답이 $y = [1, 10, 13, 7]$ 이고 우리 모델의 예측 값이 $\hat{y} = [10, 3, 1, 4]$ 와 같이 비슷한 값을 예측한다면 MSE 손실함수는 아래와 같이 1.5 라는 작은 값을 갖게 될 것입니다.

$$MSE = \frac{1}{2 \times 4} \{ (2 - 1)^2 + (10 - 10)^2 + (11 - 13)^2 + (6 - 7)^2 \} = 1.5$$

손실 함수(Loss Function) - MSE

- 이처럼 손실 함수는 우리가 풀고자 하는 목적에 가까운 형태로 파라미터가 최적화 되었을 때(즉, 모델이 잘 학습되었을 때) 더 작은 값을 갖는 특성을 가져야만 합니다.
- 이런 특징 때문에 손실 함수를 다른 말로 **비용 함수** Cost Function라고도 부릅니다.

scikit-learn

- 방금 살펴본 선형 회귀(Linear Regression)을 포함한 다양한 머신러닝 모델을 쉽고 간편하게 구현할 수 있도록 도와주는 라이브러리가 **scikit-learn**입니다.
- <https://scikit-learn.org/>



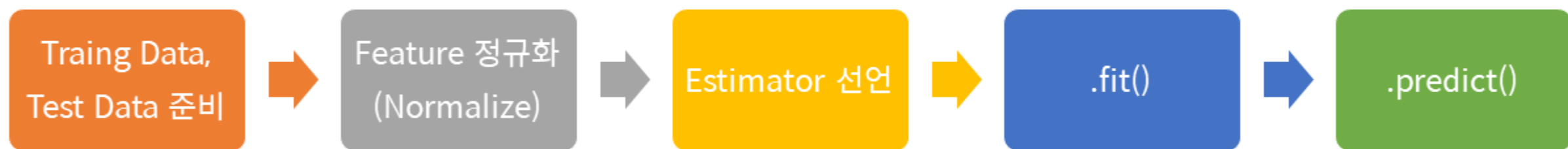
scikit-learn 기본 사용법

- scikit-learn의 기본 사용법은 다음과 같습니다.

- ① Estimator 선언 (e.g. LinearRegression)
- ② .fit() 함수 호출을 통한 트레이닝 (Training)
- ③ .predict() 함수 호출을 통한 예측 (Predict)

scikit-learn을 이용한 예측 모델 구성방법

- scikit-learn을 이용해서 예측모델을 생성하는 방법은 다음과 같습니다.



scikit-learn을 training data, test data 나누기

- scikit-learn을 이용해서 training data, test data를 나누는 방법은 다음과 같습니다.

```
from sklearn.model_selection import train_test_split  
  
# 80%는 트레이닝 데이터, 20%는 테스트 데이터로 나눕니다.  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

scikit-learn을 선형회귀(Linear Regression) Estimator 선언하기

- Scikit-learn을 이용해서 선형회귀(Linear Regression) Estimator를 선언하는 방법은 다음과 같습니다.

```
from sklearn.linear_model import LinearRegression  
  
# 선형회귀(Linear Regression) 모델 선언하기  
lr = LinearRegression()
```

Regression 알고리즘의 성능 평가 지표- RMSE(Root Mean Squared Error)

- MSE는 차이를 제공해서 더하므로 차이가 증폭되는 문제가 있을 수 있습니다.
- 따라서 MSE에 Root를 씌운 형태의 **RMSE (Root Mean Squared Error)**도 많이 사용하는 지표중 하나입니다.
- **RMSE**는 다음 수식으로 정의됩니다.

$$RMSE = \sqrt{\frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

Regression 알고리즘의 성능 평가 지표- MAE(Mean Absolute Error)

- 또한 예측값과 정답간의 차이에 절대값을 취한 **MAE(Mean Absolute Error)**도 Regression 알고리즘의 성능 평가 지표로 활용될 수 있습니다.
- **MAE**는 다음 수식으로 정의됩니다.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

scikit-learn을 이용해서 MSE, RMSE 정의하기

- MSE는 차이를 제공하기 때문에 제곱에 의해서 생기는 오차를 보정하기 위해서 **RMSE(Root Mean Square Error)**를 이용해서 성능을 측정하기도 합니다.
- scikit-learn에서 MSE와 RMSE를 정의하는 방법은 다음과 같습니다.

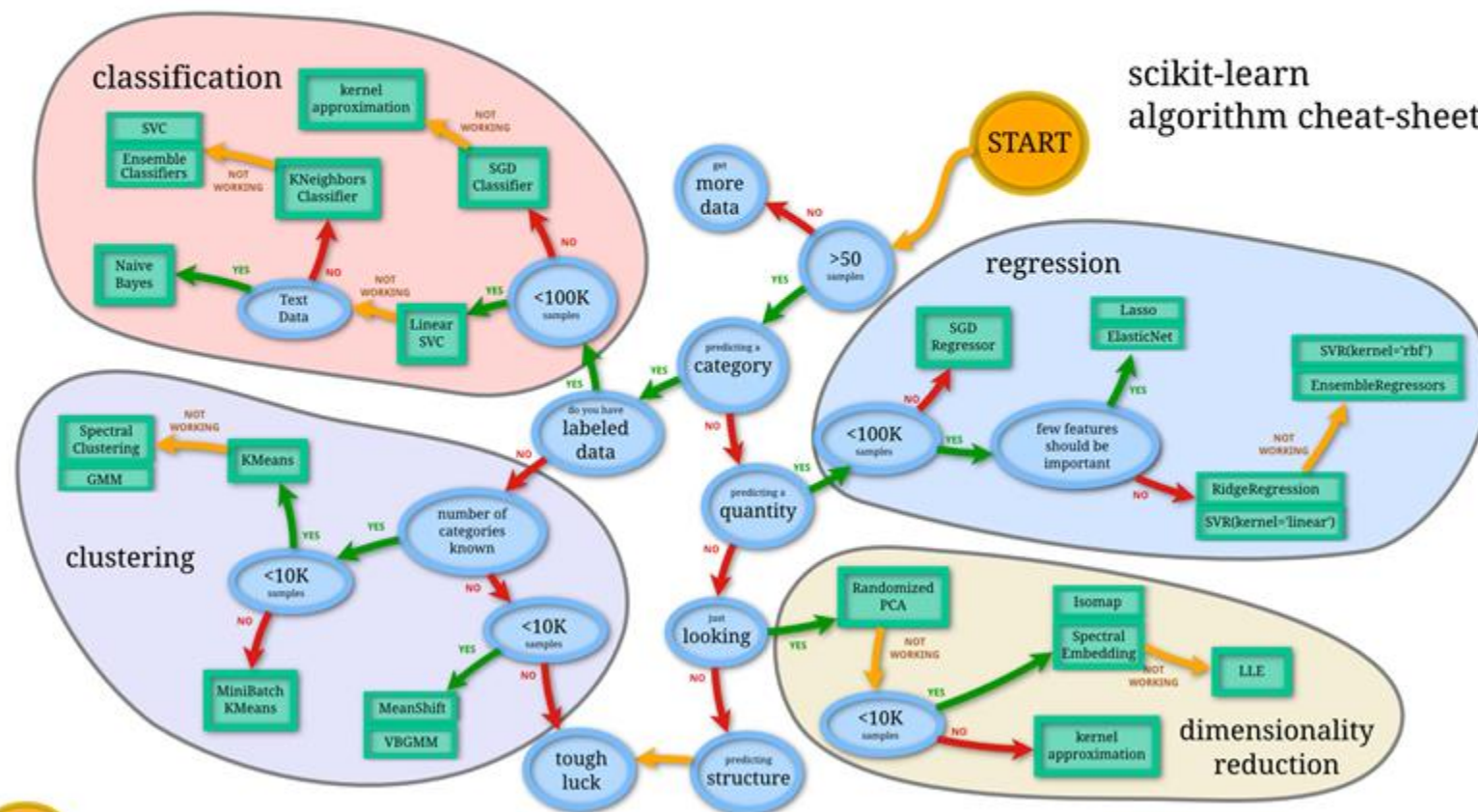
```
from sklearn.metrics import mean_squared_error

# MSE(Mean Squared Error)를 측정합니다.
MSE = mean_squared_error(y_test, y_preds)

# RMSE(Root Mean Squared Error)를 측정합니다.
RMSE = np.sqrt(MSE)
```

scikit-learn을 이용해서 상황에 따른 적절한 모델(Estimator) 선택하기

- https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html





THANK YOU :)