머신러닝의 개념

머신러닝이란?

- 데이터로부터 학습하도록 컴퓨터를 프로그래밍하는 과학
  - 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야

- 애플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하는 알고리즘

머신러닝이란?

- 현실 세계의 매우 복잡한 조건들로 인해 기존의 소프트웨어 코드만으로는 해결하기 어려웠던 많은 문제점들은 이제 머신러닝을 이용해 해결해 나가고 있다.

- 데이터 마이닝, 영상인식, 음성인식, 자연어 처리 등의 여러 분야에서 사용된다.

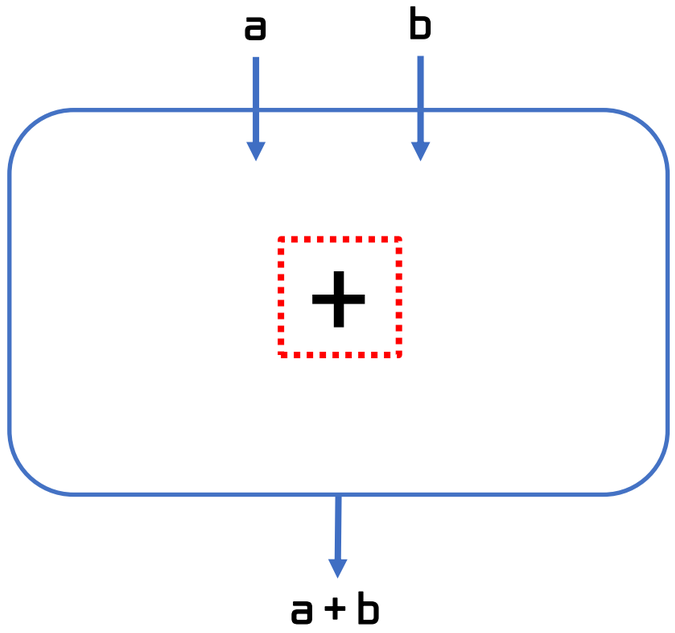
머신러닝이란?

- 현실 세계의 매우 복잡한 조건들로 인해 기존의 소프트웨어 코드만으로는 해결하기 어려웠던 많은 문제점들은 이제 머신러닝을 이용해 해결해 나가고 있다.

- 데이터 마이닝, 영상인식, 음성인식, 자연어 처리 등의 여러 분야에서 사용된다.

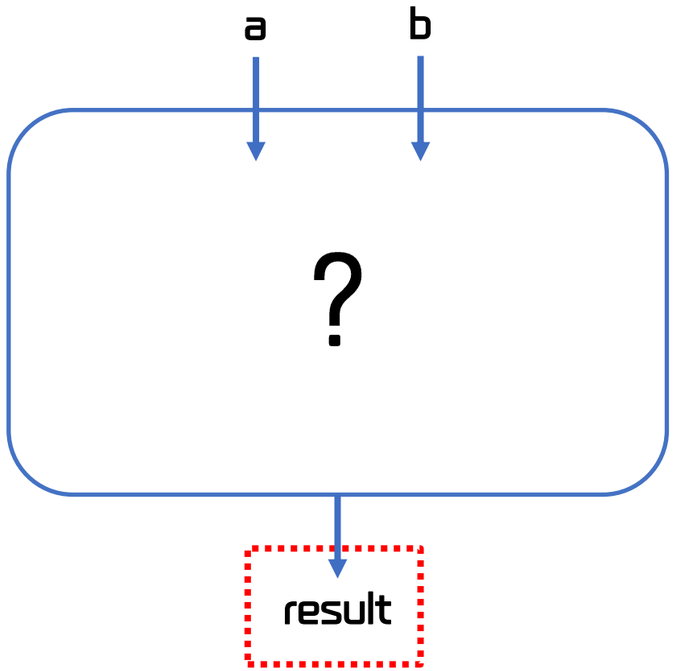
머신러닝과 기존 컴퓨터 사이언스의 차이점

- 기존 컴퓨터 사이언스는 로직을 미리 만들어서 데이터를 받아 결과를 확인하는 방식



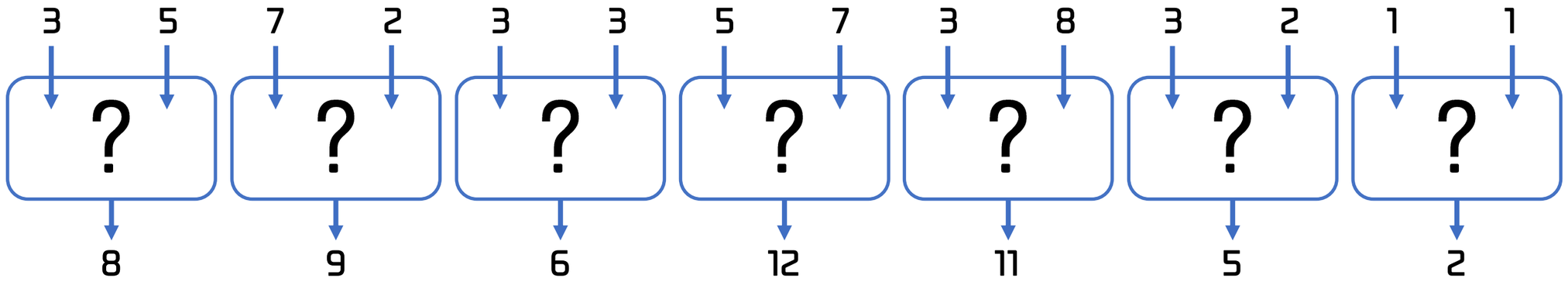
머신러닝과 기존 컴퓨터 사이언스의 차이점

- 머신러닝은 특성 데이터와 결과를 넣어 로직을 머신이 학습하는 방식



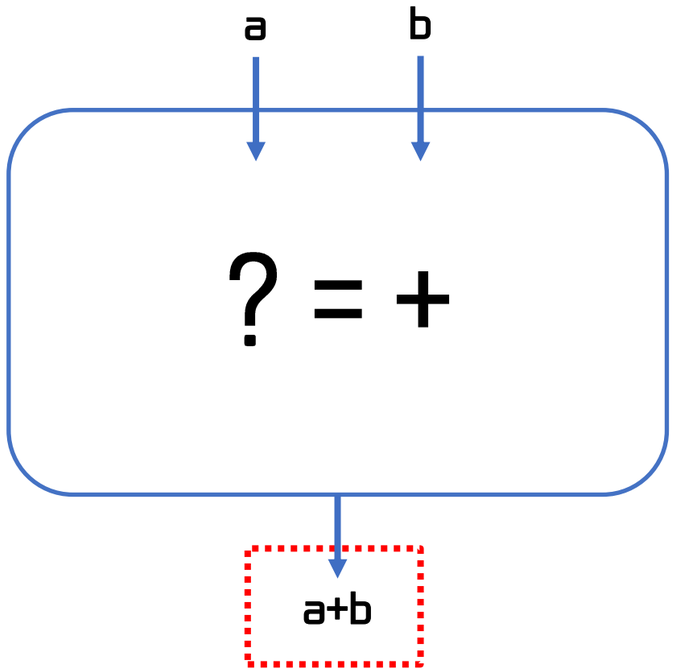
머신러닝 알고리즘에 문제와 답을 입력하면?

- 머신러닝 모델에 문제 데이터와 정답 데이터를 반복적으로 넣어주면



머신러닝과 기존 컴퓨터 사이언스의 차이점

- 입력된 데이터의 패턴을 파악하여 어떤 연산이 이루어 질지를 학습하게 된다.



머신러닝의 유형

- 지도 학습(Supervised Learning)
  - 분류, 회귀, 추천 시스템, 시각/음성 감지 인지 등
  - 지도학습은 머신러닝 모델에게 문제(feature)와 답(label)을 모두 제공

- 비지도 학습(Un-Supervised Learning)
  - 군집화(클러스터링), 차원 축소, 토픽 모델링, 문서 군집 등
  - 비지도 학습은 머신러닝 모델에게 문제(feature)만 제공

지도학습 - 분류(Classification)

- Discreate(Categorical) Valued Output을 예측하는 문제
  - Yes/No 문제일 때 : Binary Classification
    - 음성(Negative), 양성(Positive)를 예측한다.
    - 당신은 비만인가요?
  - 다지선다형 문제일 때 : Multi-Class Classification
    - 여러 카테고리 중 하나를 예측한다.
    - 이 사진은 고양이, 강아지, 말 중 어떤 동물의 사진인가요?
- 예시
  - 스팸 메일 판단하기
  - 신용카드 부정사례 검출하기
  - 코로나 검사

지도학습 - 회귀(Regression)

- Continuous Valued Output 예측하기
- 예시
  - 공부 시간으로 시험점수 예측하기
  - 집값 예측하기

머신러닝의 단점

- 데이터에 너무 의존적이다. (Garbage In, Garbage Out)
- 학습시에 최적의 결과를 도출하기 위해 수립된 머신러닝 모델은 실제 환경 데이터 적용 시 과적합 되기 쉽다.
- 복잡한 머신러닝 알고리즘으로 인해 도출된 결과에 대한 논리적인 이해가 어려울 수도 있다.
- 데이터만 집어 넣으면 자동으로 최적화된 결과를 도출할 것이라는 것은 환상!
  - 끊임없이 모델을 개선하기 위한 노력이 필요
  - 데이터의 특성을 파악하고 최적의 알고리즘과 파라미터를 구성할 수 있는 고급 능력이 필요하다.

사이킷런

사이킷런 소개

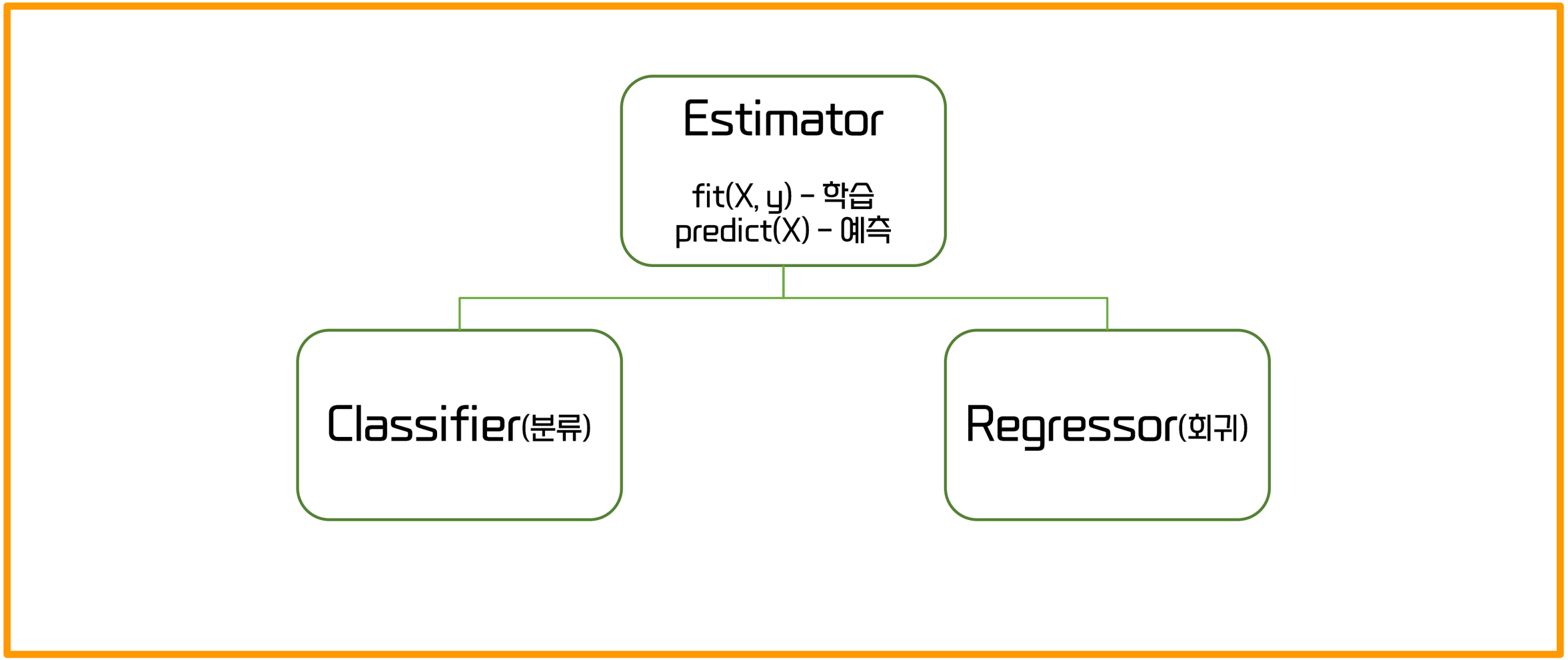
- 파이썬 기반의 다른 머신러닝 패키지도 사이킷런 스타일의 API를 지향할 정도로 쉽고, 가장 파이썬다운 API를 제공
- 머신러닝을 위한 매우 다양한 알고리즘과 개발을 위한 편리한 프레임워크와 API를 제공
- 오랜 기간 실전 환경에서 검증됐으며, 매우 많은 환경에서 사용되는 성숙한 라이브러리
- 주로 Numpy와 Scipy 기반 위에서 구축된 라이브러리

머신러닝을 위한 용어 정리

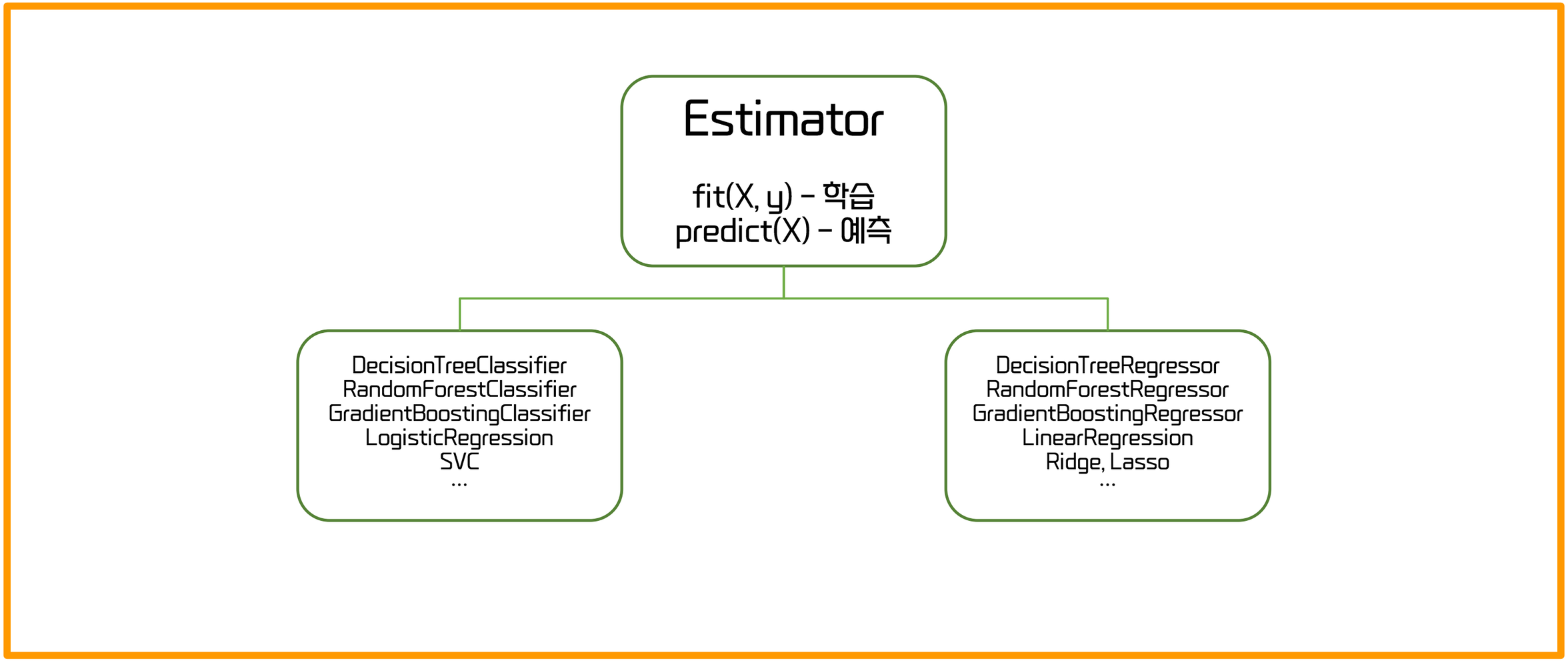
- Feature (X)
  - Feature는 데이터 세트의 일반 속성
  - 머신러닝은 2차원 이상의 다차원 데이터에서도 많이 사용되므로 타겟값을 제외한 나머지 속성을 모두 Feature로 지칭

- Label, Class, Target (y)
  - Target은 지도 학습 시 데이터의 학습을 위해 주어지는 정답 데이터
  - 지도 학습 중 분류의 경우에는 이 타겟을 레이블로 지칭
  - 클래스는 분류 문제에서 레이블의 종류를 의미

사이킷런 기반 프레임워크 – Estimator



사이킷런 기반 프레임워크 – Estimator



사이킷런 주요 모듈 소개

분류	Module	설명
예제 데이터	sklearn.datasets	사이킷런에 내장되어 예제로 제공하는 데이터 세트
데이터 분리, 검증 파라미터 튜닝	sklearn.model_selection	교차 검증을 위한 학습용/테스트용 분리, 그리드 서치(Grid Search)로 최적 파라미터 추출 등의 API 제공
피처 처리	sklearn.preprocessing	데이터 전처리에 필요한 다양한 가공 기능 제공 (인코딩, 정규화, 스케일링 등)
	sklearn.feature_selection	알고리즘에 큰 영향을 미치는 피처를 우선순위 대로 셀렉션 작업을 수행하는 다양한 기능을 제공
	sklearn.feature_extraction	텍스트 데이터나 이미지 데이터의 벡터화된 피처를 추출하는데 사용 텍스트 데이터 피처 추출(sklearn.feature_selection.text) 이미지 데이터 피처 추출(sklearn.feature_selection.image)
피처 처리 및 차원축소	sklearn.decomposition	차원 축소와 관련한 알고리즘을 지원하는 모듈 PCA, NMF, Truncated SVD 등을 통한 차원 축소 기능 수행

사이킷런 주요 모듈 소개

분류	Module	설명
평가	sklearn.metrics	분류, 회귀, 클러스터링 등에 대한 다양한 성능측정 방법 제공 Accuracy, Precision, Recall, ROC-AUC, RMSE 등 제공
머신러닝 알고리즘	sklearn.ensemble	앙상블 알고리즘 제공 RandomForest, AdaBoost, GradientBoosting 등
	sklearn.linear_model	주로 선형 회귀, Ridge, Lasso 및 LogisiticRegression 등 회귀 관련 알고리즘 지원. SGD(Stochastic Gradient Descent) 관련 알고리즘도 지원
	sklearn.naive_bayes	나이브 베이즈 알고리즘 제공. 가우시안 NB, 다항 분포 NB 등 지원
	sklearn.neighbors	최근접 이웃 알고리즘 제공. KNN 등
	sklearn.svm	서포트 벡터 머신 알고리즘 제공
	sklearn.tree	의사 결정 트리 알고리즘 제공
	sklearn.cluster	비지도 클러스터링 알고리즘 제공 (K-Mean, 계층형, DBSCAN 등)
유틸리티	sklearn.pipeline	피처 처리 등의 변환과 ML 알고리즘 학습, 예측 등을 함께 묶어서 실행할 수 있는 유틸리티 제공

사이킷런 붓꽃 데이터 살펴보기

붓꽃 예제 데이터 확인하기

- sklearn.datasets.load_iris 모듈에서 붓꽃 예제 제공

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0

붓꽃 예제 데이터 확인하기

- sklearn.datasets.load_iris 모듈에서 붓꽃 예제 제공

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
data	5.1	3.5	1.4	0.2	0
	4.9	3.0	1.4	0.2	0
	4.7	3.2	1.3	0.2	0

붓꽃 예제 데이터 확인하기

- sklearn.datasets.load_iris 모듈에서 붓꽃 예제 제공

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0

target

붓꽃 예제 데이터 확인하기

- sklearn.datasets.load_iris 모듈에서 붓꽃 예제 제공

feature_names

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0

실습 1-1

사이킷런 내장 예제 데이터 살펴보기

머신러닝 모델링 및 예측 프로세스

데이터 세트 분리	데이터를 학습 데이터, 테스트 데이터로 분리
모델 학습(fit)	학습 데이터를 기반으로 머신러닝 알고리즘을 적용해 모델을 학습
예측 수행(predict)	학습된 머신러닝 모델을 이용해 테스트 데이터로 예측
평가(evaluate)	예측된 결괏값과 테스트 데이터의 실제 결괏값을 비교해 머신러닝 모델 성능을 평가

실습 1-2

머신러닝 모델링 프로세스

훈련 데이터와 테스트 데이터

학습(훈련) 데이터 세트와 테스트 데이터 세트

훈련 데이터 세트

- 머신러닝 알고리즘의 학습을 위해 사용
- 데이터의 속성들과 결정값(레이블) 모두를 가지고 있음
- 훈련 데이터를 기반으로 머신러닝 알고리즘이 데이터 속성과 결정값의 패턴을 인지하고 학습

테스트 데이터 세트

- 훈련 데이터 세트에서 학습된 머신러닝 알고리즘을 테스트
- 테스트 데이터는 속성 데이터만 머신러닝 알고리즘에 제공하며, 머신러닝 알고리즘은 제공된 데이터를 기반으로 결정값을 예측
- 테스트 데이터는 학습 데이터와 별도의 데이터 세트로 제공되어야 함

훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

만약 가지고 있는 모든 데이터를 머신러닝 모델 훈련을 위해 사용한다면?

데이터 세트

↓ 훈련

머신러닝 모델

↓ 예측

평가

훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

만약 가지고 있는 모든 데이터를 머신러닝 모델 훈련을 위해 사용한다면?

데이터 세트

↓ 훈련

머신러닝 모델

↓ 예측

이미 알고 있는 데이터에 대한 평가가 의미가 있을까?

훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

만약 가지고 있는 모든 데이터를 머신러닝 모델 훈련을 위해 사용한다면?

데이터 세트

↓ 훈련

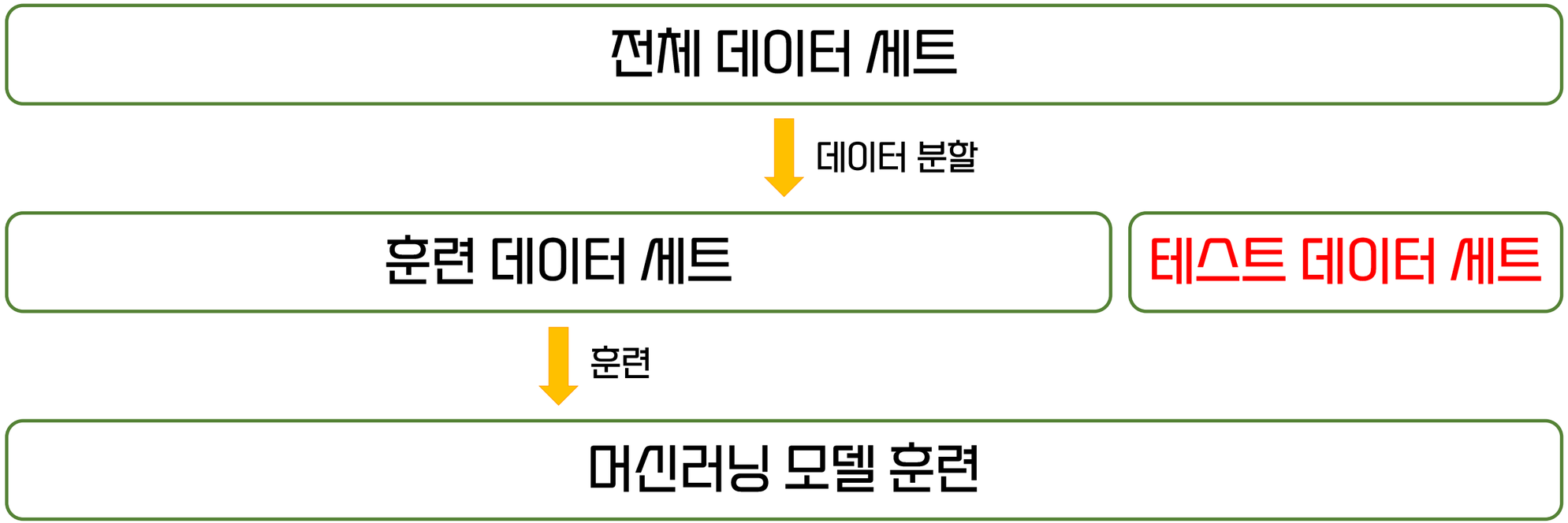
머신러닝 모델

↓ 예측

새로운 데이터에 대해 예측을 잘 할 수 있다는 근거가 없다!

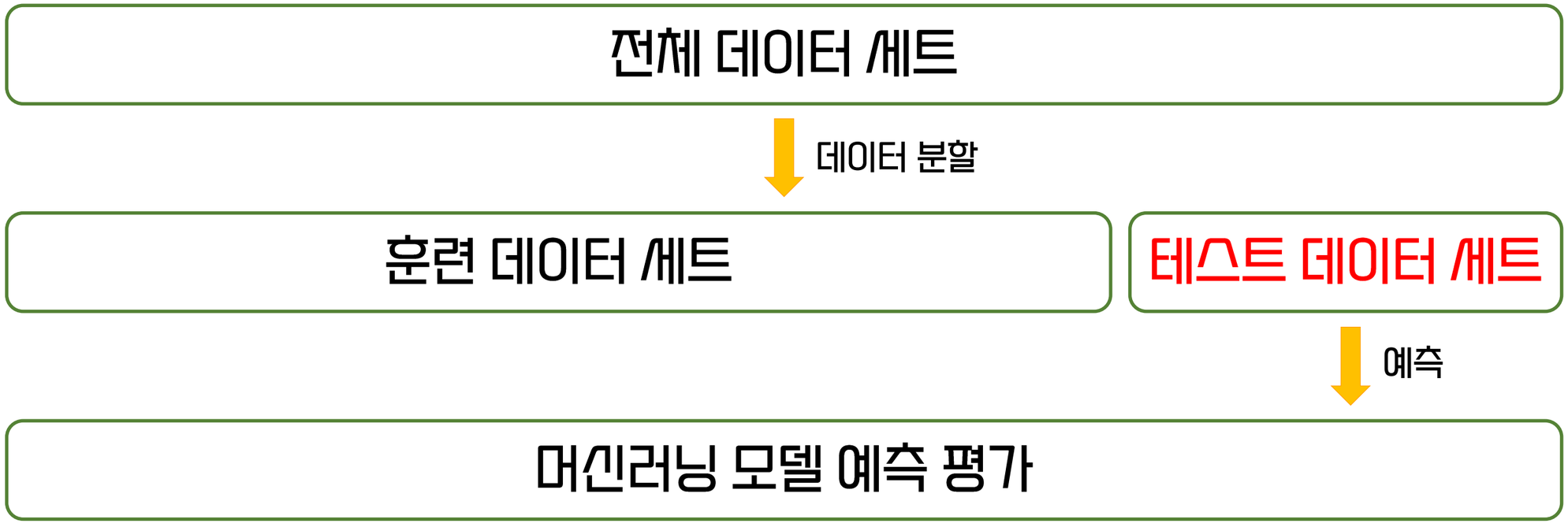
훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

데이터를 분할하여 새로운 데이터에 대해 얼마나 잘 예측 할지를 가늠해 볼 수 있다.



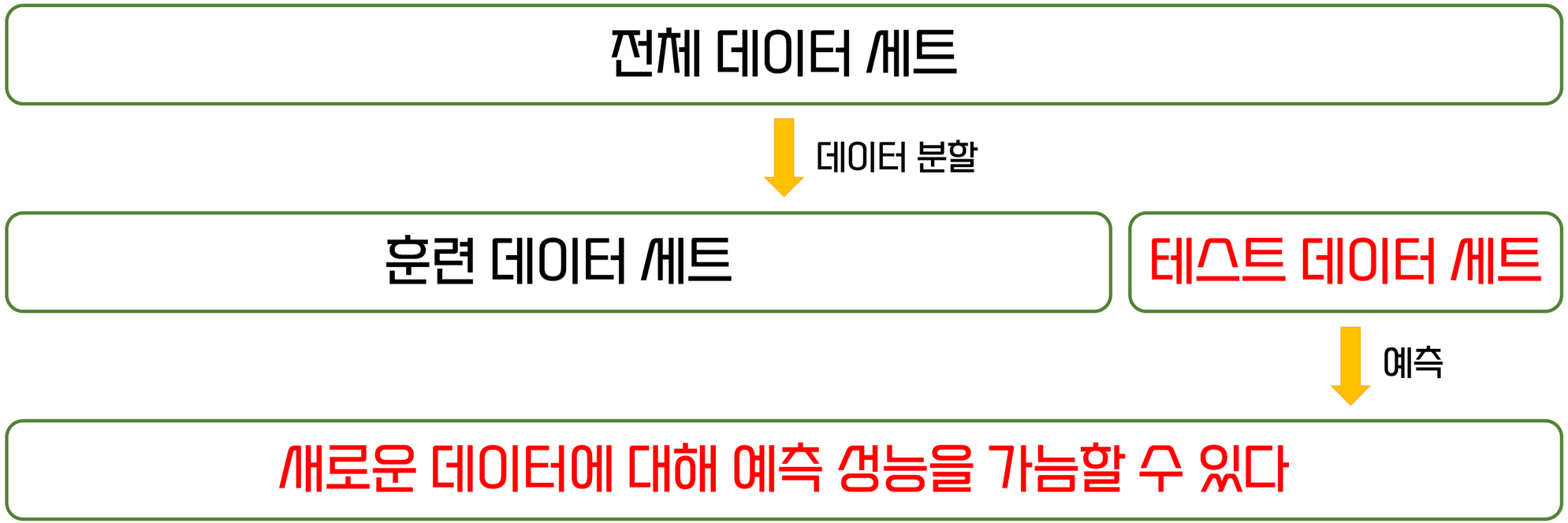
훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

데이터를 분할하여 새로운 데이터에 대해 얼마나 잘 예측 할지를 가늠해 볼 수 있다.



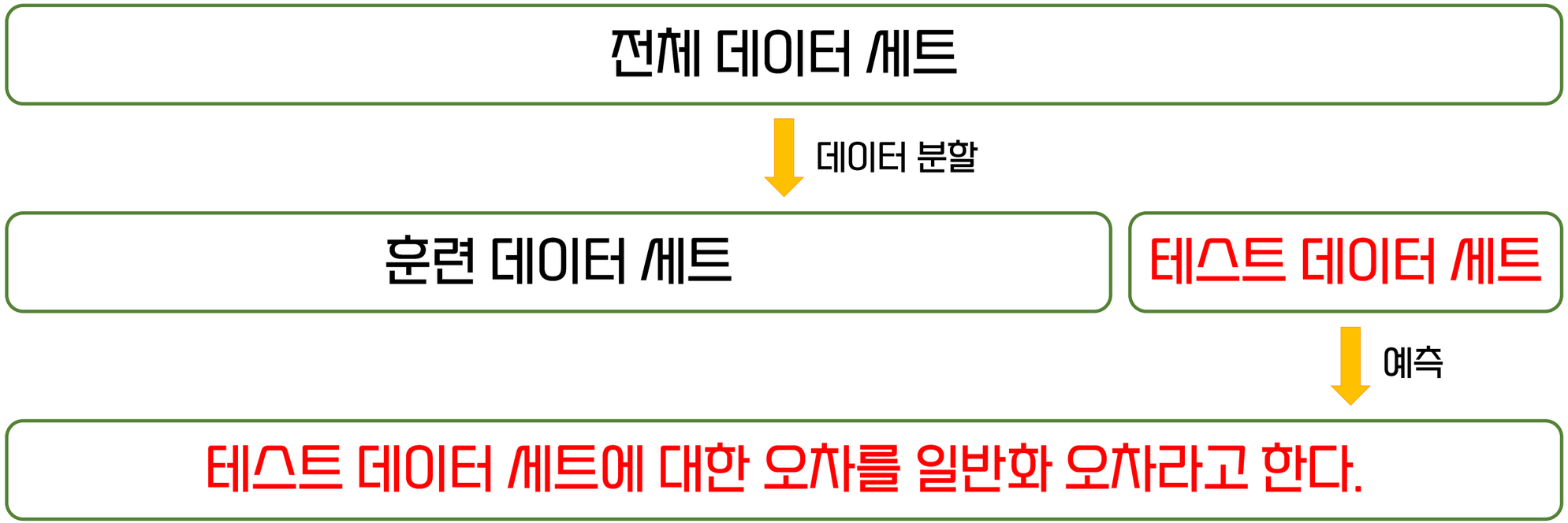
훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

데이터를 분할하여 새로운 데이터에 대해 얼마나 잘 예측 할지를 가늠해 볼 수 있다.



훈련 데이터 세트와 테스트 데이터 세트가 따로 존재하는 이유

데이터를 분할하여 새로운 데이터에 대해 얼마나 잘 예측 할지를 가늠해 볼 수 있다.



train_test_split

sklearn.model_selection의 train_test_split 알아보기

X_train, X_test, y_train, y_test = train_test_split(data, target, test_size, random_state, stratify)

- data
  - feature 데이터
- target
  - label 데이터
- test_size
  - 전체 데이터에서 테스트 데이터 세트의 비율.
    기본값은 0.25로서 25%를 테스트 세트로 사용
- random_state
  - train_test_split은 모든 데이터를 랜덤하게 섞고(shuffle) 분할(split)하기 때문에, 수행 시 마다 다른 데이터 세트가 생성될 수 있으므로 random_state를 이용해 난수 값을 고정
- stratify
  - 데이터 분할 시 원본 데이터의 비율과 동일하게 테스트 세트를 생성하기 위해 지정하는 데이터 기준

왜 랜덤이 필요한가?

만약 target 데이터가 다음과 같은 상황이라면?

[0,0,0,0,0,1,1,1,1,1,2,2,2,2,2]

- 만약 위 데이터를 단순하게 앞에서 부터 10개, 5개로 끊어서 훈련 / 테스트 데이터 세트로 만든다면 다음과 같다.
  - 훈련 데이터 세트 : [0,0,0,0,0,1,1,1,1,1]
  - 테스트 데이터 세트 : [2,2,2,2,2]
- 훈련 데이터 세트에는 0과 1번 레이블만 학습하게 되고, 테스트 시에는 2번만 테스트하기 때문에 머신러닝 모델이 0, 1, 2 클래스를 골고루 학습할 수 없다.
- train_test_split은 랜덤성을 이용해 train/test 데이터 세트를 적절히 섞어서 잘라준다

계층적 분할(Stratified Split) 방식

만약 다음과 같이 데이터가 구성이 되어있을 때

[0,0,0,0,0,1,1,1,1,1,2,2,2,2,2]

- train_test_split을 이용하여 위 데이터를 분할 했을 때 다음과 같게 될 수 있다.
  - train set : [0,0,0,0,1,1,1,2,2]
  - test set : [0,1,1,2,2,2]
- 원본 데이터의 비율은 1: 1: 1 이지만 랜덤하게 데이터를 분할할 때 위와 같이 한쪽 데이터가 많아지거나 적어질 수도 있게 된다.
- 원본 데이터 비율에 맞게 훈련 세트와 테스트 세트를 분할하는 것을 계층적 분할이라고 한다..
  - train set : [0,0,0,1,1,1,2,2,2]
  - test set : [0,0,1,1,2,2]

실습 1-3

train_test_split 알아보기

교차 검증

검증(Validation) 세트

전체 데이터 세트

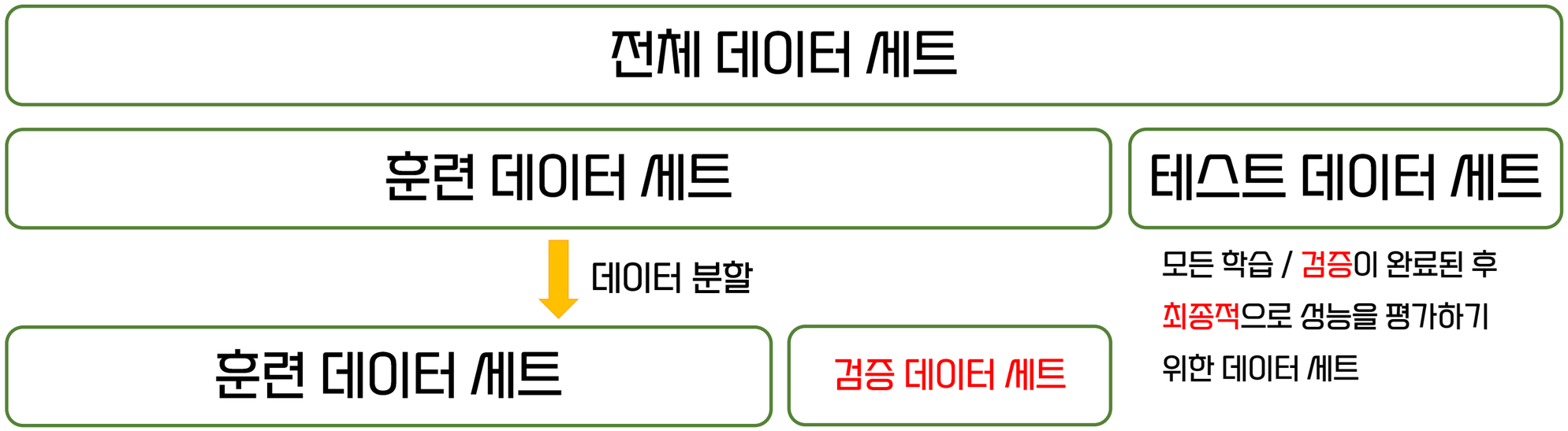
훈련 데이터 세트

테스트 데이터 세트

머신러닝 모델을 훈련하기 위한 데이터 세트

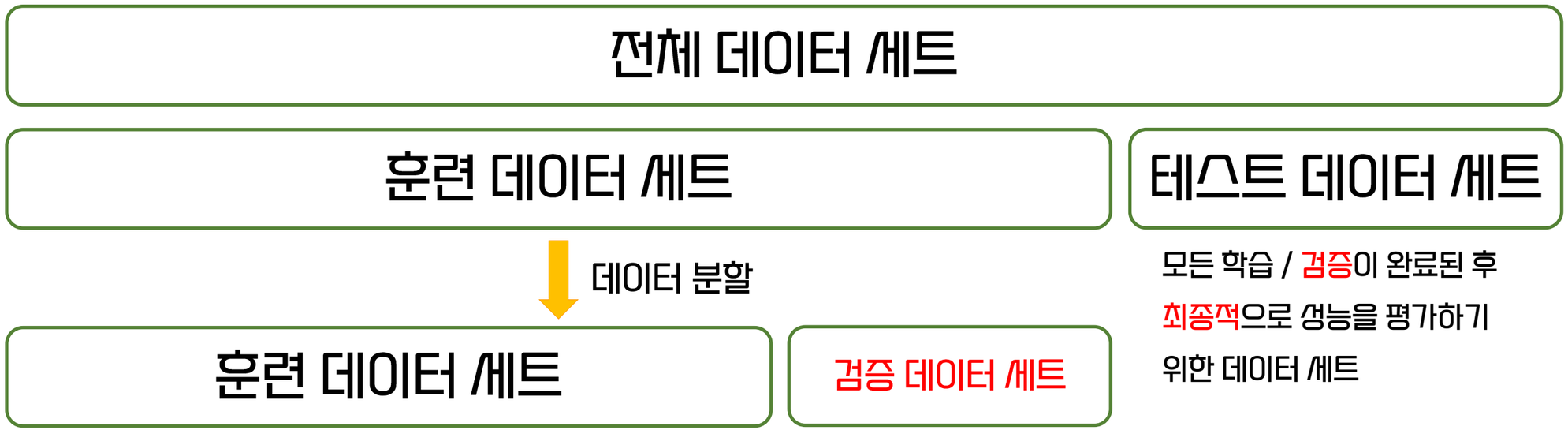
모든 학습이 완료된 후
최종적으로 성능을 평가하기
위한 데이터 세트

검증(Validation) 세트



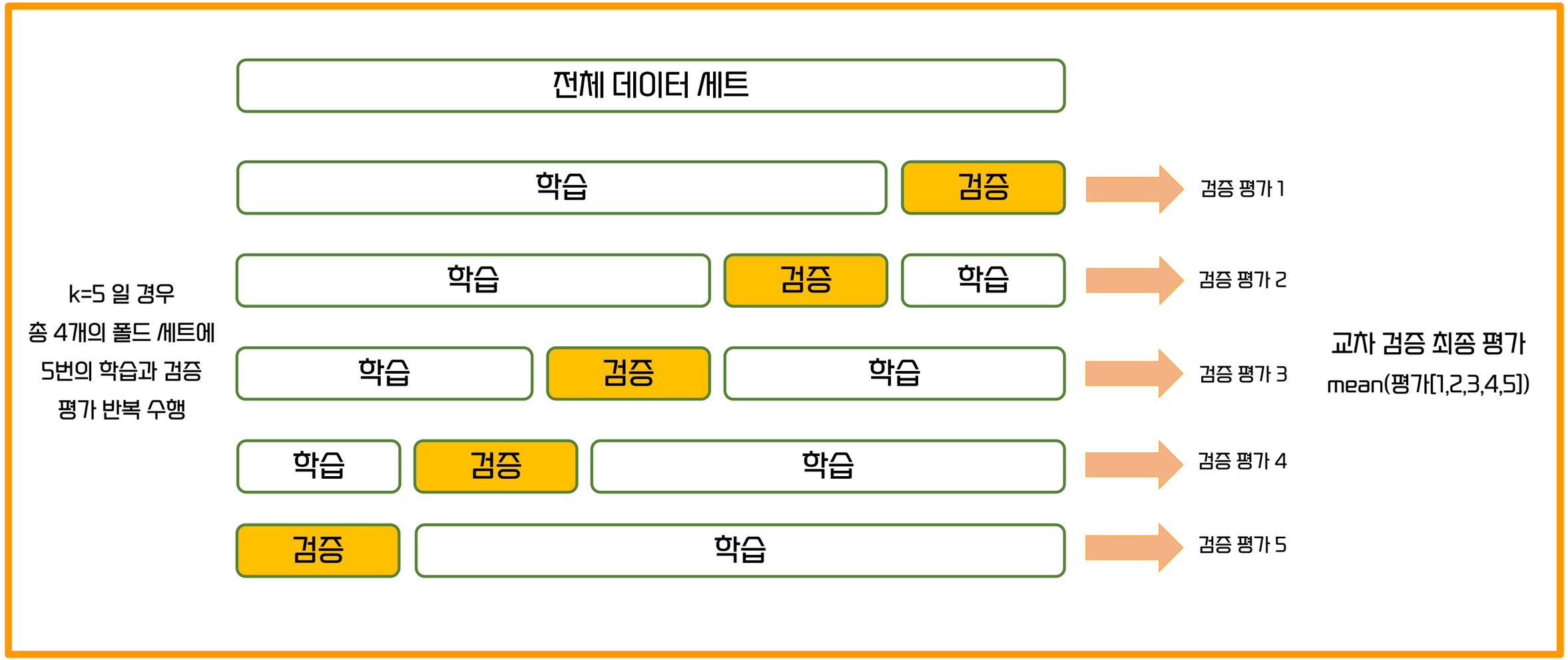
학습 데이터를 다시 분할하여 학습 데이터와 학습된 모델의
성능을 일차적으로 평가하는 검증 데이터 세트로 나눔

검증(Validation) 세트



이 검증 데이터 세트를 여러 번 바꿔서 미리 성능 검증을 하는 것을 교차 검증이라고 한다.

K 폴드(Fold) 교차 검증



Stratified K 폴드

- 일반 K 폴드(K Fold)
  - 기계적으로 지정한 폴드의 개수 k 만큼 데이터를 나누는 방식으로서, 데이터의 분포를 고려하지 않는다.

- 계층적 K 폴드(Stratified K Fold)
  - 불균형한(imbalanced) 분포도를 가진 레이블(결정 클래스) 데이터 집합을 위한 K 폴드 방식
  - 학습 데이터와 검증 데이터 세트가 가지는 레이블 분포도가 유사하도록 검증 데이터 추출

실습 1-3

K-Fold, Stratified K-Fold를 이용한 교차 검증

간편한 교차 검증 - cross_val_score

- K-Fold 클래스를 이용한 교차 검증 방법을 간편화 한 사이킷런의 검증 함수
  - 폴드 세트 추출, 학습/예측, 평가를 한번에 수행할 수 있다.

```
cross_val_score(estimator, X, y=None, scoring=None, cv=None, n_jobs=1)
```

- estimator : 모델
- X : feature
- y : target
- scoring : 예측 성능 평가 방식
- cv : 폴드의 개수

교차 검증과 최적 하이퍼 파라미터 튜닝을 한번에 – GridSearchCV

- 사이킷런은 GridSearchCV를 이용해 분류, 회귀 모델 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터를 도출할 수 있다.

- 하이퍼 파라미터란?
  - 머신러닝의 개별적인 모델에 입력해야 하는 값을 의미한다.
  - 즉 모델이 학습하는 값이 아닌 개발자가 직접 넣어줘야 하는 값을 지칭한다.
  - 하이퍼 파라미터에 의해 모델의 성능이 조절되기 때문에 모델 알고리즘의 최적 튜닝을 할 수 있다.

교차 검증과 최적 하이퍼 파라미터 튜닝을 한번에 – GridSearchCV

- 사이킷런은 GridSearchCV를 이용해 분류, 회귀 모델 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터를 도출할 수 있다.

```
GridSearchCV(estimator, param_grid, cv, refit=True, return_train_score=True)
```

- estimator : 모델
- param_grid : 하이퍼 파라미터의 목록이 들어있는 딕셔너리, 여러 개의 딕셔너리를 이용할 수도 있다.
- cv : 폴드의 개수
- refit : True로 설정하면 가장 좋은 파라미터 설정으로 재학습 시킨다.
- return_train_score : 훈련 결과 점수를 확인할 수 있다.

데이터 전처리

데이터 전처리(Preprocessing)

- 데이터 클린징
- 결손값 처리(Null, NaN 처리)
- 데이터 인코딩(레이블, 원-핫 인코딩)
- 데이터 스케일링
- 이상치(Outlier) 제거
- Feature 선택, 추출 및 가공

데이터 인코딩

- 머신러닝 알고리즘은 문자열 데이터 속성을 입력 받지 않는다.
- 문자형 카테고리형 속성은 모두 숫자값으로 변환/인코딩 되어야 한다.

- 레이블(Label) 인코딩
- 원-핫(One-Hot) 인코딩

레이블(Label) 인코딩

원본 데이터

상품 분류	가격
청바지	50,000
치마	25,000
원피스	35,000
원피스	35,000
청바지	50,000
치마	25,000

상품 분류를 레이블 인코딩한 데이터

상품 분류	가격
0	50,000
1	25,000
2	35,000
2	35,000
0	50,000
1	25,000

[청바지, 원피스, 청바지, 치마, 청바지] → [0, 2, 0, 1, 0]

원-핫(One-Hot) 인코딩

- 원-핫 인코딩은 Feature의 유형에 따라 새로운 Feature를 추가해 고유 값에 해당하는 컬럼에만 1을 표시
- 나머지 컬럼에는 0을 표시

원본 데이터

상품 분류
청바지
치마
원피스
원피스
청바지
치마

원-핫 인코딩

상품 분류_청바지	상품 분류_치마	상품 분류_원피스
1	0	0
0	1	0
0	0	1
0	0	1
1	0	0
0	1	0

원-핫(One-Hot) 인코딩

- 원-핫 인코딩은 Feature의 유형에 따라 새로운 Feature를 추가해 고유 값에 해당하는 컬럼에만 1을 표시
- 나머지 컬럼에는 0을 표시

원본 데이터

상품 분류
청바지
치마
원피스
원피스
청바지
치마

원-핫 인코딩

상품 분류_청바지	상품 분류_치마	상품 분류_원피스
1	0	0
0	1	0
0	0	1
0	0	1
1	0	0
0	1	0

Feature Scaling

- 표준화
  - 데이터의 feature 각각이 평균이 0이고 분산이 1인 가우시안 정규 분포를 가진 값으로 변환
  - $x_i = \frac{x_i - \mu}{\sigma}$ σ : 표준 편차, μ : 평균
  - [20, 30, 40] → [-1.22, 0, 1.22]
  - 데이터 분포의 중심을 0으로 변경.(Zero-Centered)
- 정규화
  - $x_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$
  - [20, 30, 40] → [0, 0.5, 1]
  - 데이터의 최소값을 0으로, 최대값을 1로 변경

실습 1-4

데이터 정규화 실습

실습 1-5

타이타닉 생존자 예측