

핸즈온 머신러닝

1장. 한눈에 보는 머신러닝

박해선(옮긴이)

haesunrpark@gmail.com

<https://tensorflow.blog>

한눈에 보는 머신러닝

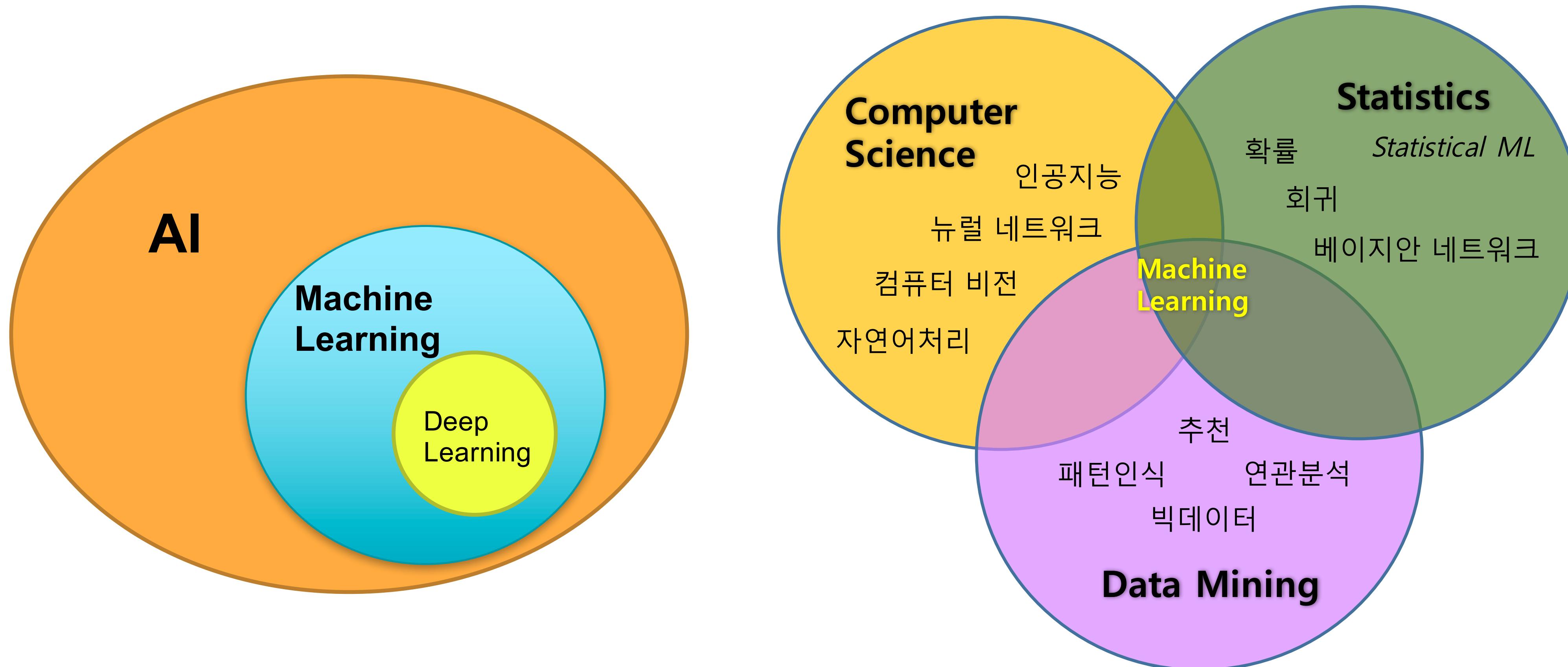
- 머신러닝이란 무엇일까요? (터미네이터나 스카이넷은 아니죠)
- 기계가 배운다는 것은 정확히 무엇을 의미하는 걸까요?
- 머신러닝이 왜 필요한가요?

이 장에서는 머신러닝의 시스템 종류, 주요 도전 과제, 평가와 튜닝 등 머신러닝 전체에 대한 그림을 그려봅니다.

머신러닝이란

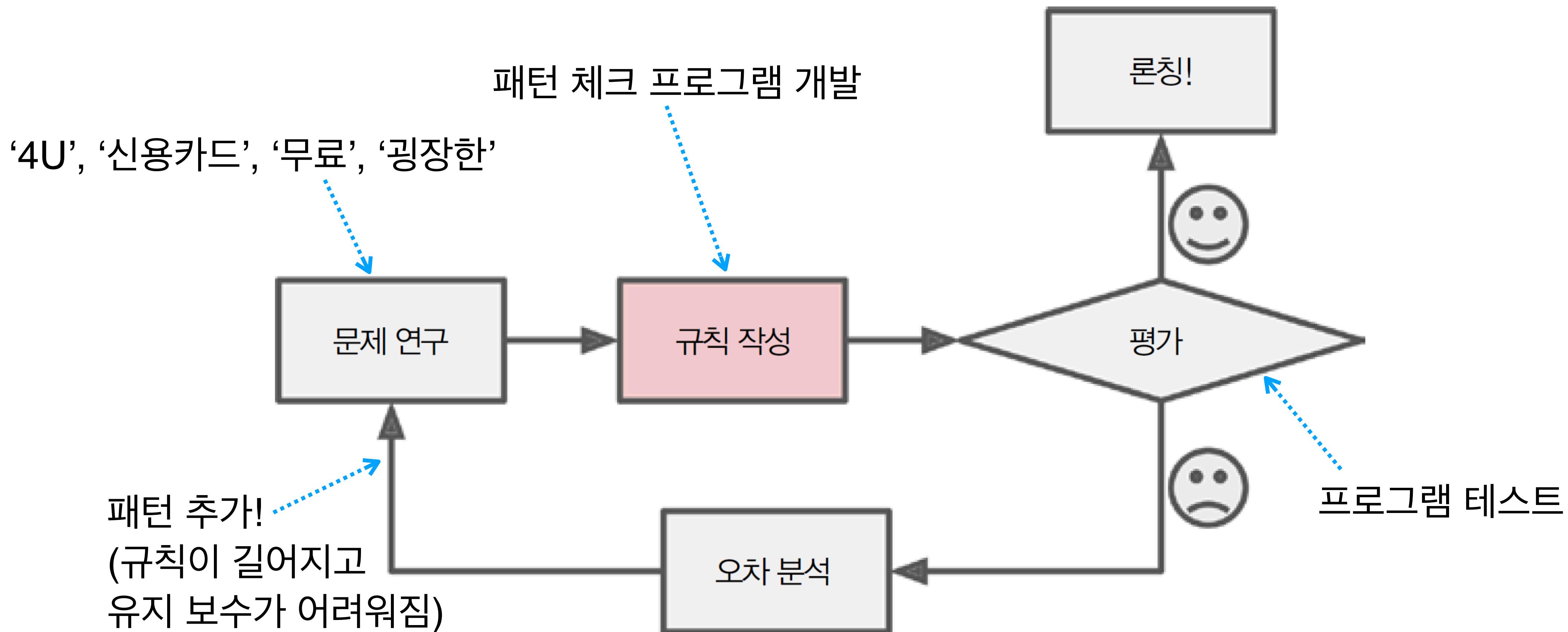
- “머신러닝”은 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야다. (아서 사무엘Arthur Samuel, 1959)
- 어떤 작업 T에 대한 컴퓨터 프로그램의 성능을 P로 측정했을 때 경험 E로 인해 성능이 향상됐다면, 이 컴퓨터 프로그램은 작업 T와 성능 측정 P에 대해 경험 E로 학습한 것이다. (톰 미첼Tom Mitchell, 1997)
 - 스팸 메일 구분하기—작업 T
 - 훈련 데이터^{training data}—경험 E
 - 정확도^{accuracy}—성능 측정 P
- “머신러닝은 데이터로부터 학습하도록 컴퓨터를 프로그래밍하는 과학(또는 예술)입니다.”

인공지능 > 머신러닝 > 딥러닝



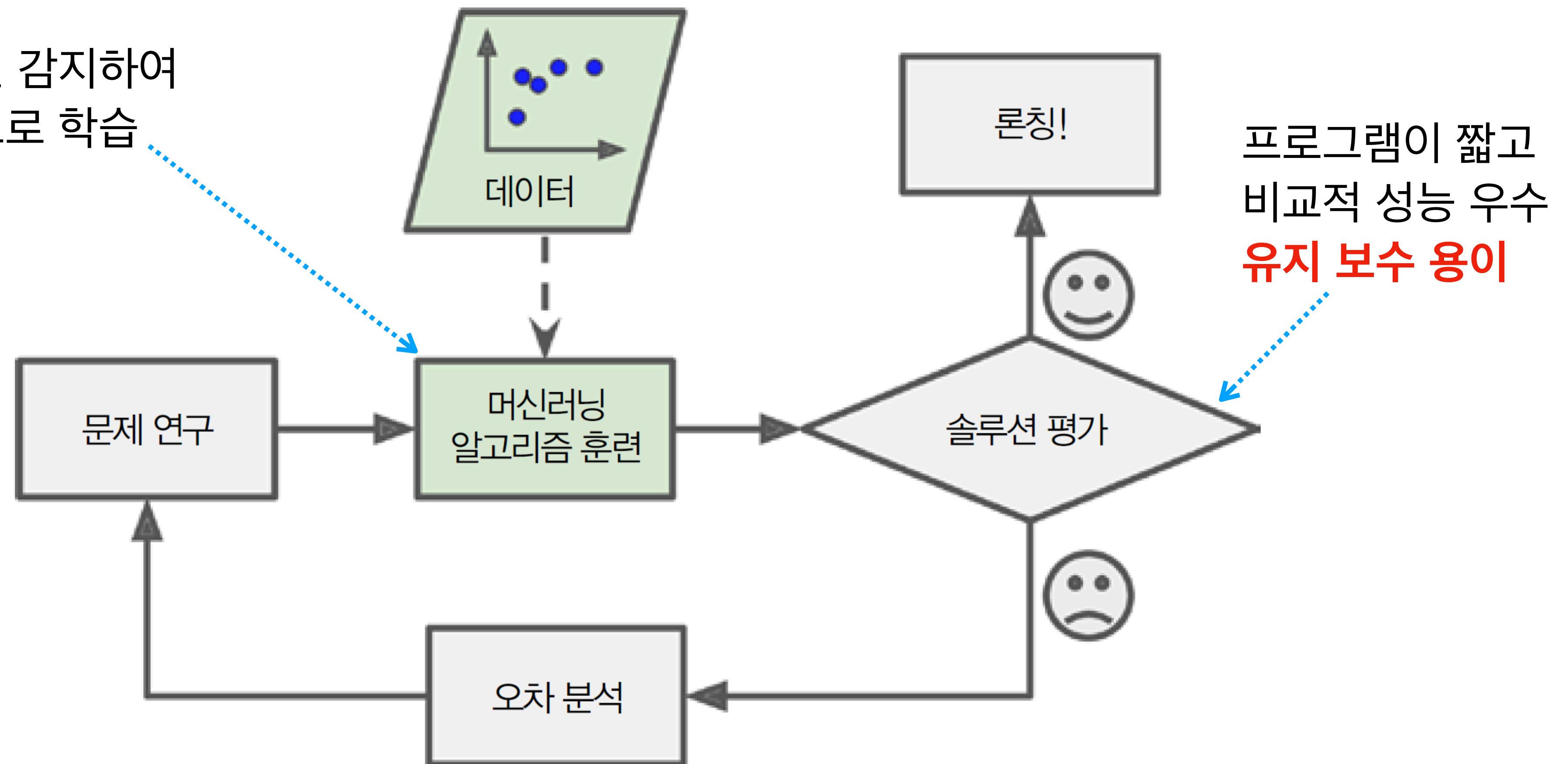
왜 머신러닝을 사용하는가

전통적인 접근 방법: 규칙 기반 시스템(rule based system)

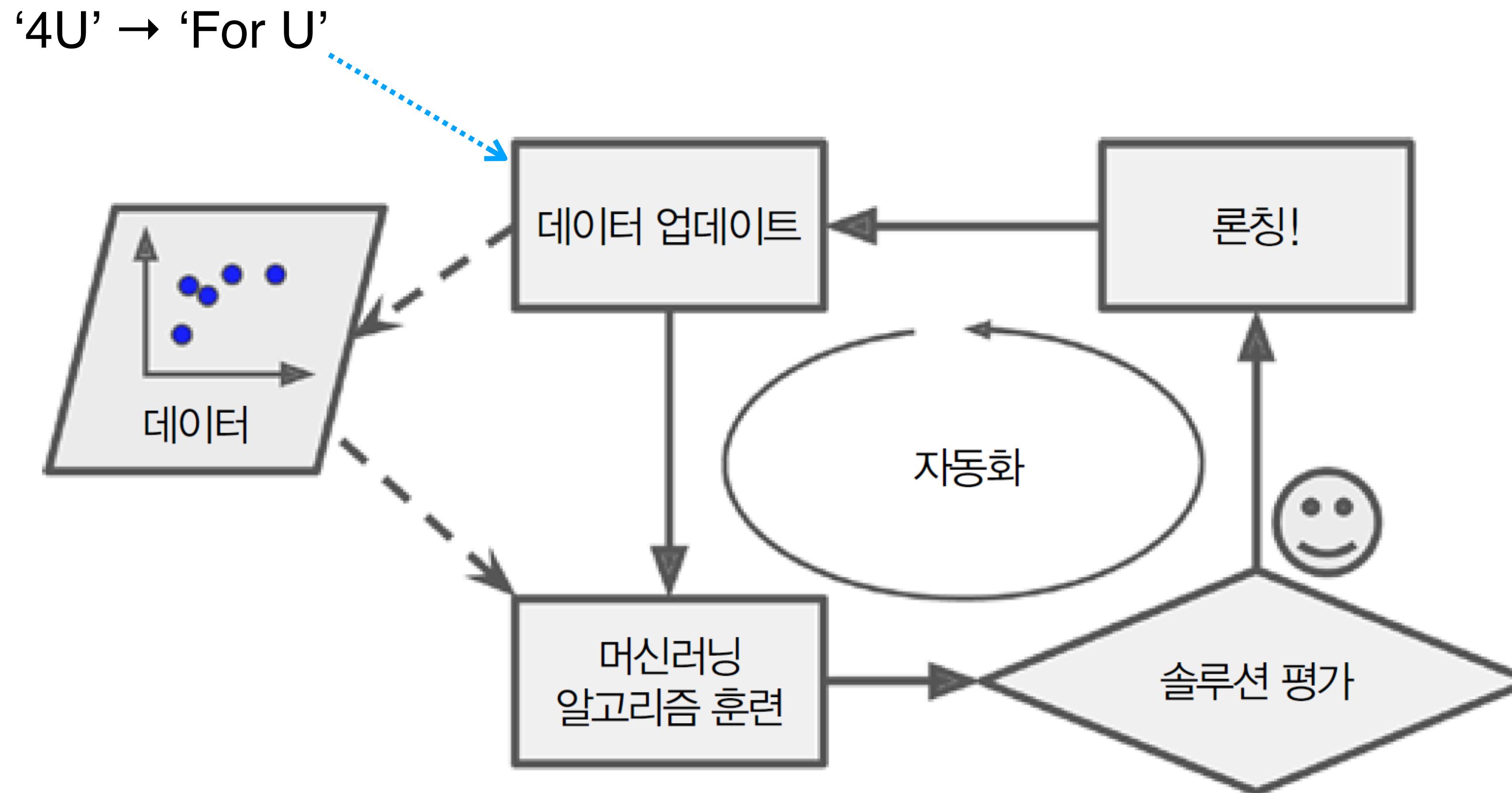


머신러닝 접근 방법

자주 나타나는 패턴 감지하여
좋은 기준을 자동으로 학습

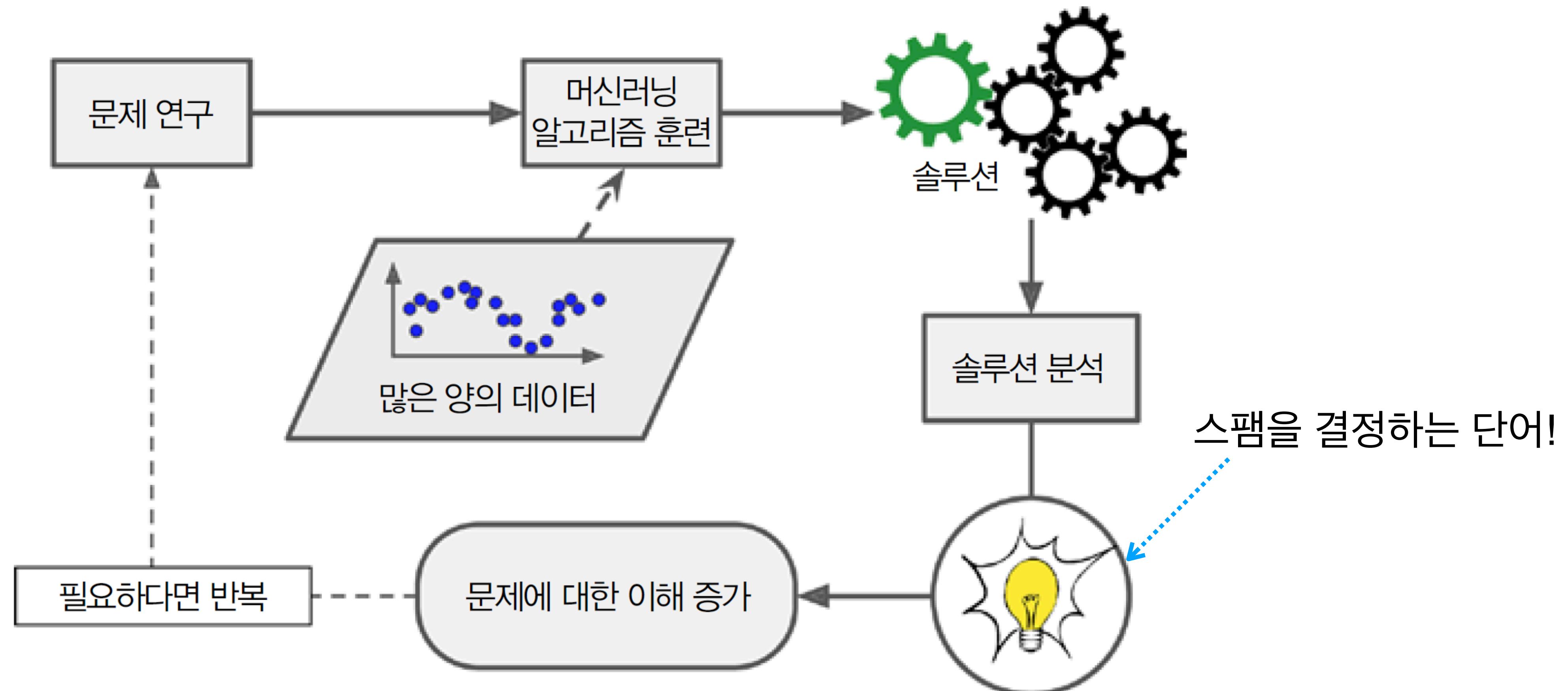


자동으로 변화에 적응



머신러닝에서 통찰을 얻다

AKA 데이터마이닝(data mining)



머신러닝이 유용한 분야

- 많은 수동 조정과 규칙이 필요한 문제
- 해결하기 너무 어렵거나 알려진 해가 없는 문제
 - 음성인식('one' vs 'two')
 - 얼굴인식(눈, 코, 입의 위치)
- 변화하는 환경에 적응해야 하는 문제
- 복잡한 문제와 대량의 데이터에서 통찰 얻기(데이터 마이닝)

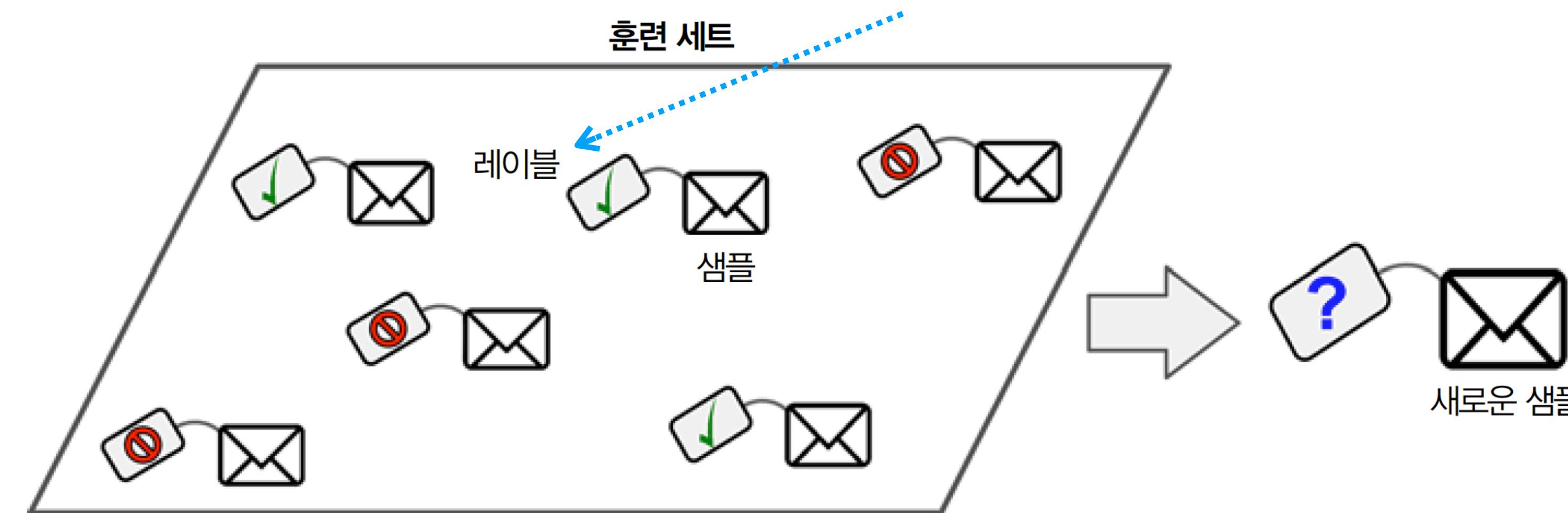
머신러닝 시스템의 종류

- 사람의 감독 여부(지도, 비지도, 준지도, 강화학습, **자기지도**)
- 실시간으로 점진적 학습인지 여부(온라인 학습, 배치 학습, 점진적 학습)
- 훈련 데이터와 단순 비교인지, 패턴을 찾기 위해 예측 모델을 만드는지
(사례 기반, 모델 기반 학습)
- 이 범주들은 배타적이지 않습니다(e.g. 스팸 필터-온라인 모델 기반 지도 학습)

지도 학습

- Supervised Learning (감독 학습)

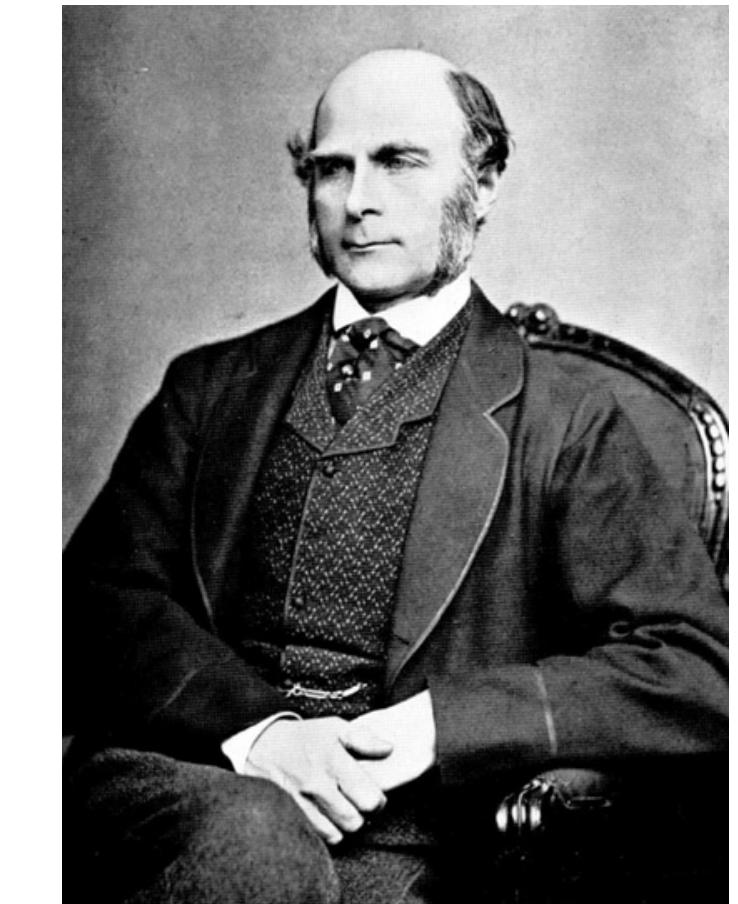
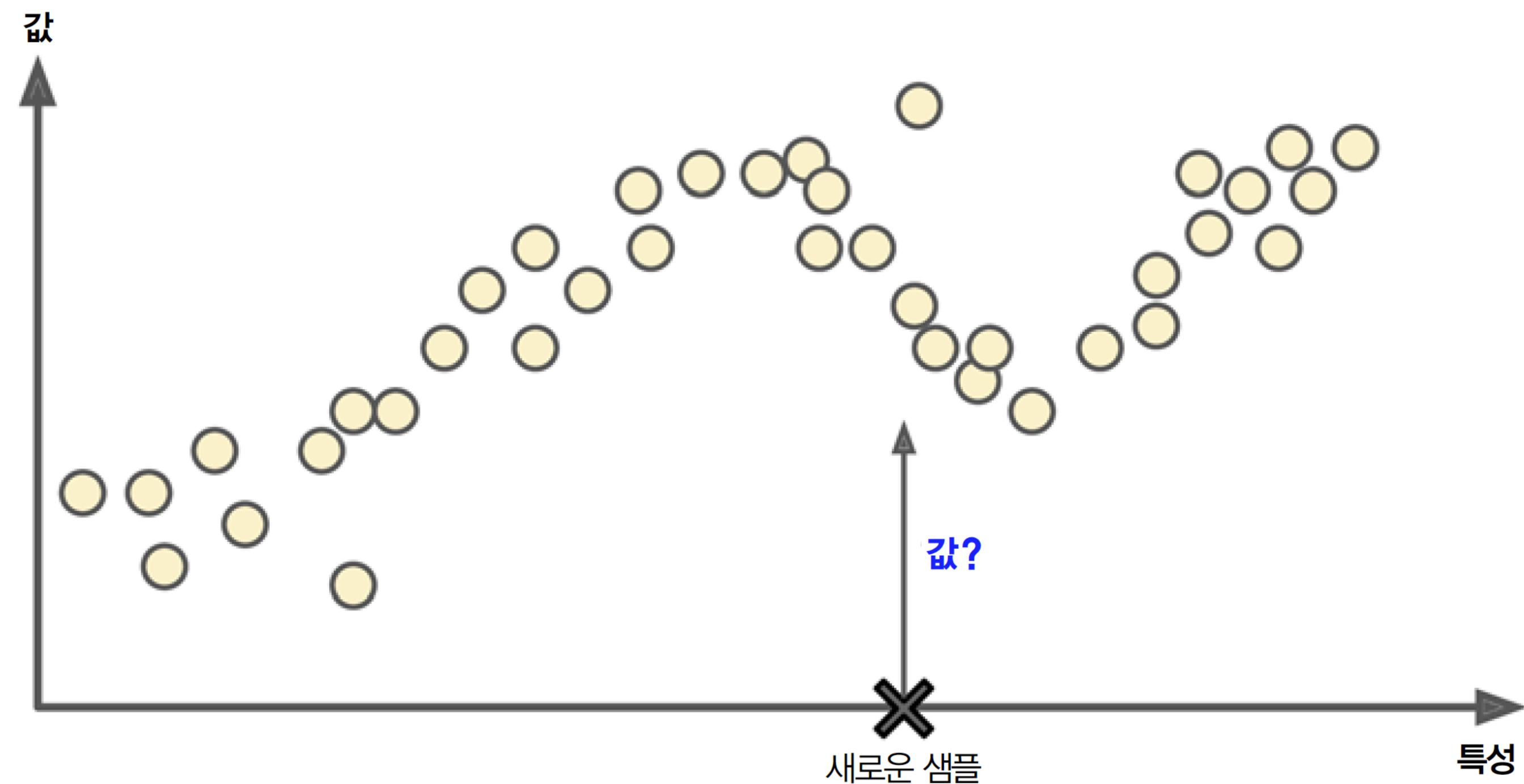
정답(ground truth) 또는 타깃(target): 스팸 or 스팸아님
이진 분류(binary classification)



- 개, 고양이, 새를 분류하는 문제는 다중 분류(multiclass classification)입니다.
- 레이블의 범주(개, 고양이, 새)를 클래스(class)라고 합니다.
- 특성(feature) 또는 속성(attribute): 스팸 메일의 단어, 이미지의 픽셀
독립(independent) 변수, 예측(predictor) 변수—종속(dependent) 변수, 반응(response) 변수

회귀(regression)

- 연속된 숫자를 예측
- 주행거리, 연식, 브랜드 등(특성)을 사용하여 중고차 가격(타깃) 예측



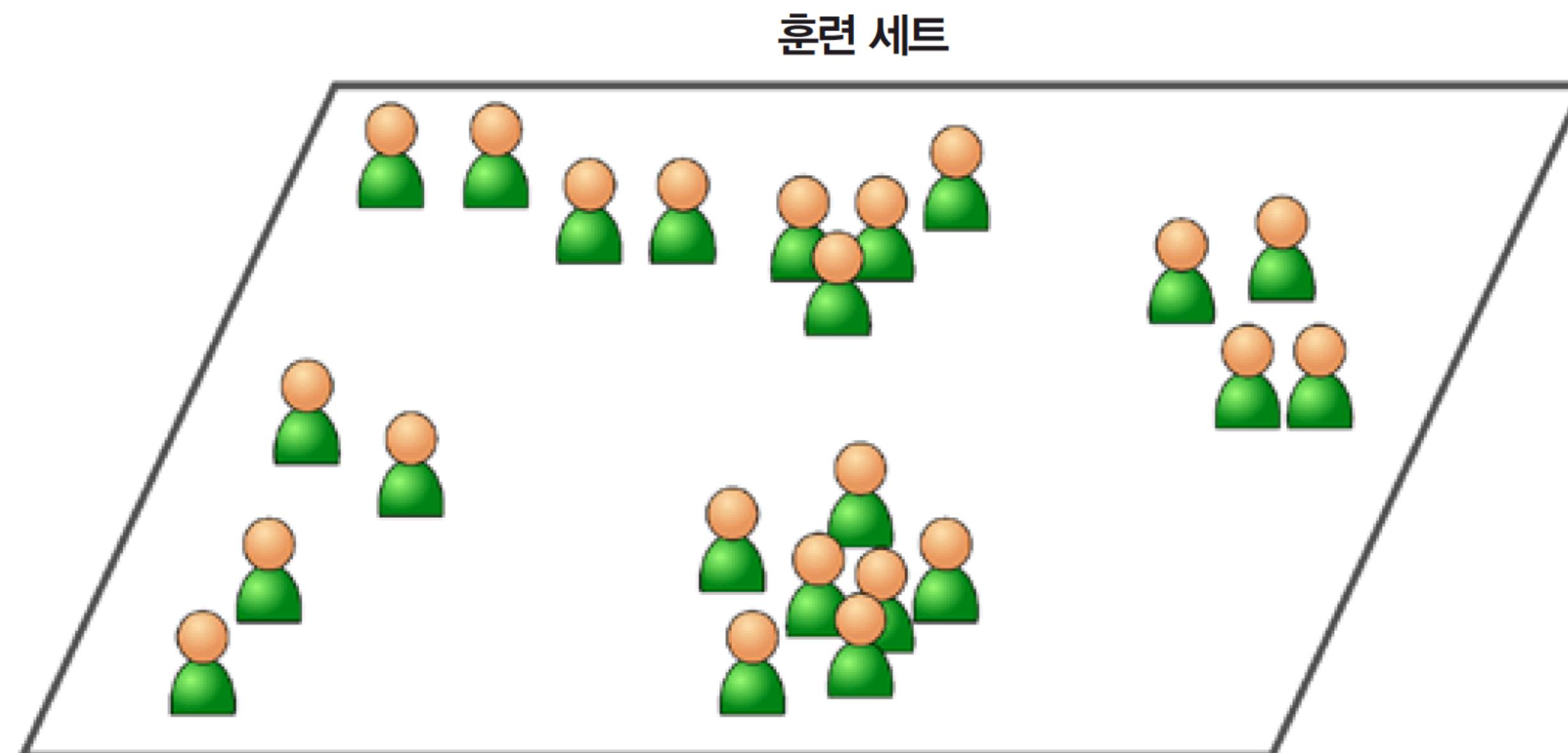
프란시스 갈頓(Francis Galton)
“regression toward the mean”

대표적인 지도 학습 알고리즘

- k-최근접 이웃(k-Nearest Neighbors)
- 선형 회귀(Linear Regression) - 회귀 알고리즘
- 로지스틱 회귀(Logistic Regression) - 분류 알고리즘
- 서포트 벡터 머신(Support Vector Machine, SVM)
- 결정 트리(Decision Tree), 앙상블 학습(Ensemble Learning)
- 신경망(Neural Network)

비지도 학습

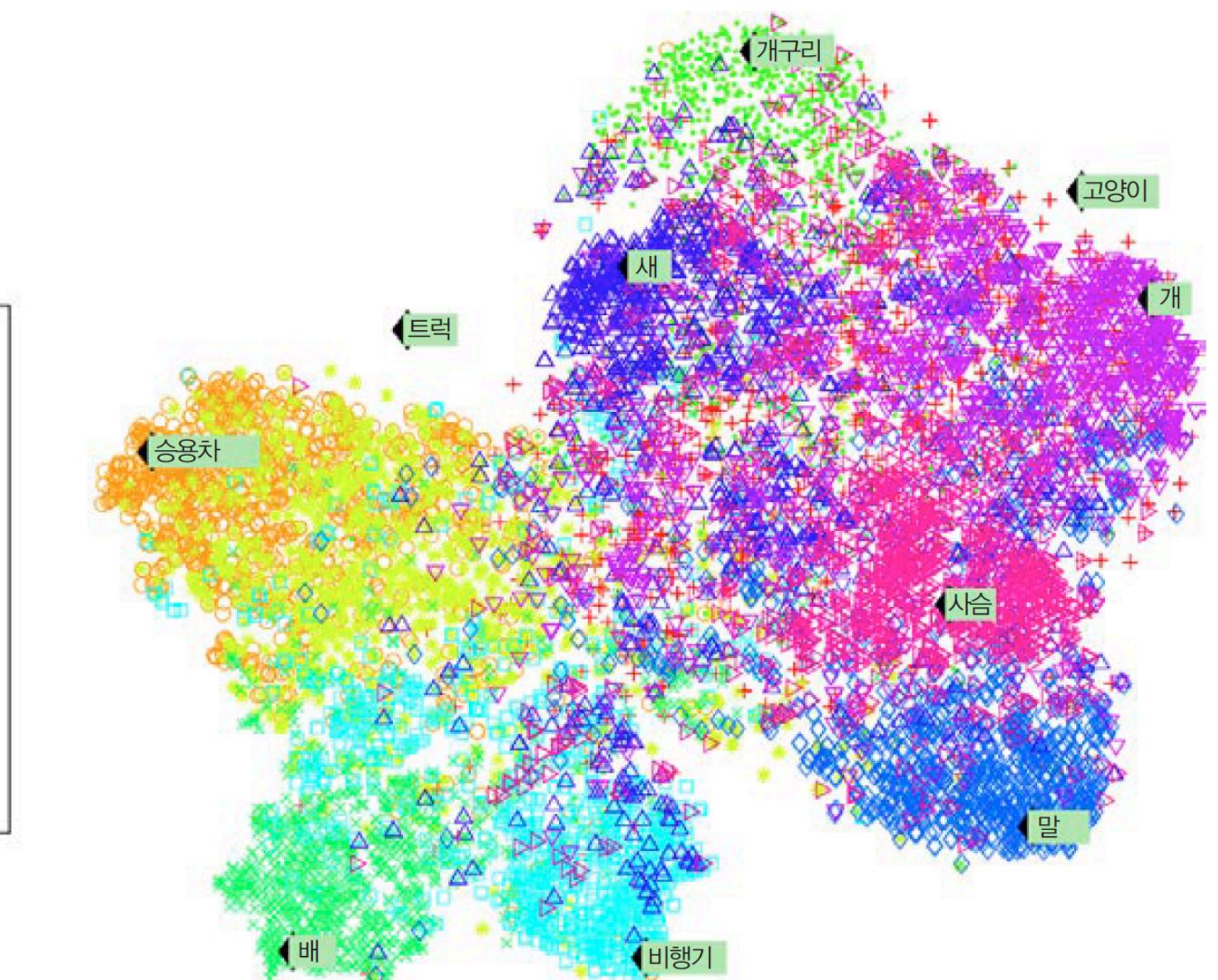
- Unsupervised Learning (비감독 학습)
- 훈련 데이터에 레이블이 없습니다.
- 블로그 방문자를 그룹으로 묶기 (어떻게 40%가 만화 애호가, 20%는 공상가인지 알 수 있을까요?)



대표적인 비지도 학습 알고리즘

- 군집(Clustering)
 - k-평균(k-means)
 - 계층 군집(Hierarchical Clustering): 병합 군집(Agglomerative Clustering)
 - 기댓값 최대화(Expectation Maximization)
- 시각화, 차원 축소-8장
 - 주성분 분석(PCA), 커널 PCA, NMF
 - 지역 선형 임베딩(LLE)
 - t-SNE
- 연관 규칙(Association Rule): 데이터 마이닝
 - 어프라이어리(Apriori), 이클렛(Eclat)

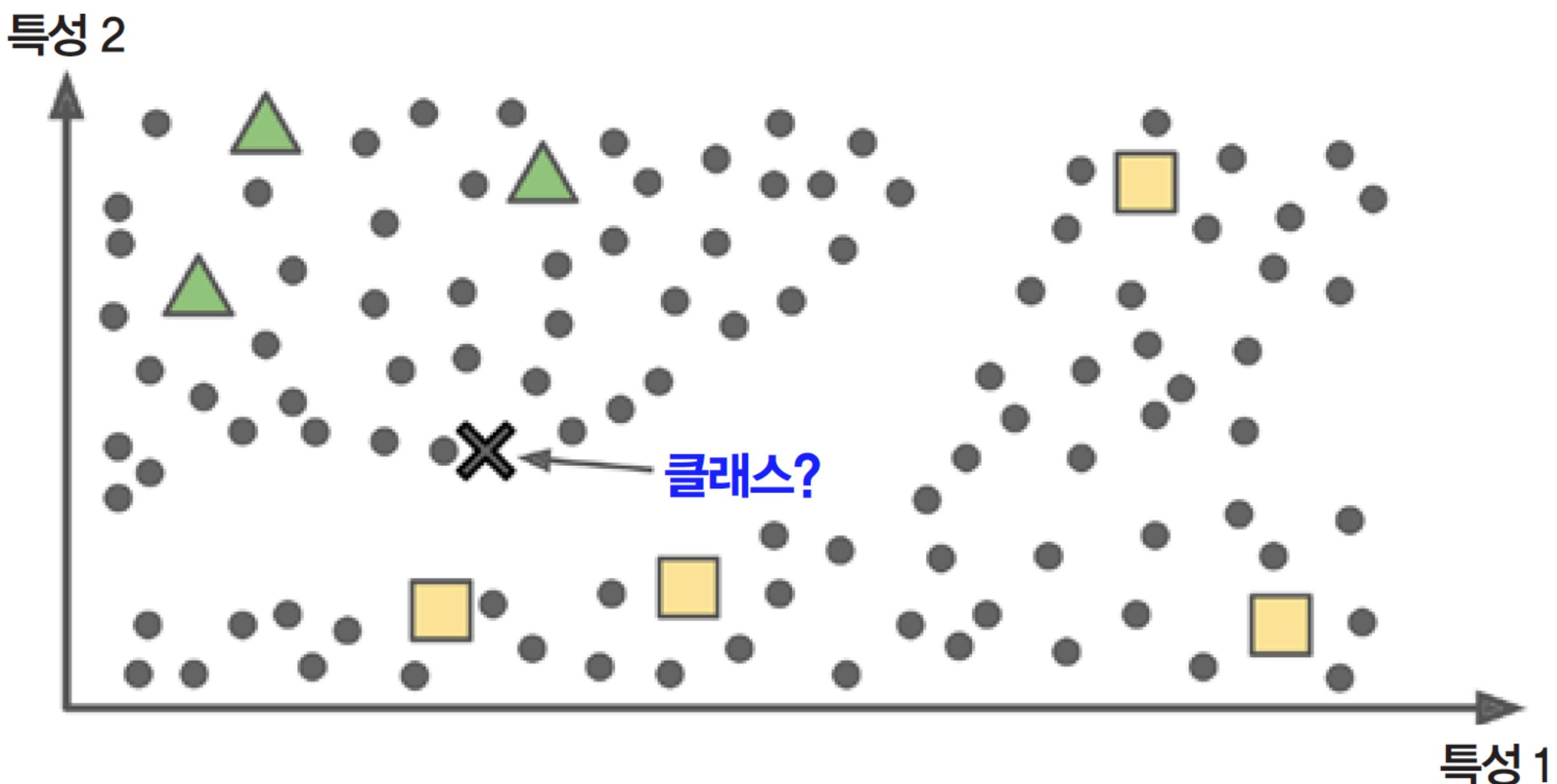
+	고양이
○	승용차
●	트럭
■	개구리
×	배
□	비행기
◇	말
△	새
▽	개
▶	사슴



“파이썬 라이브러리를 활용한 머신러닝”

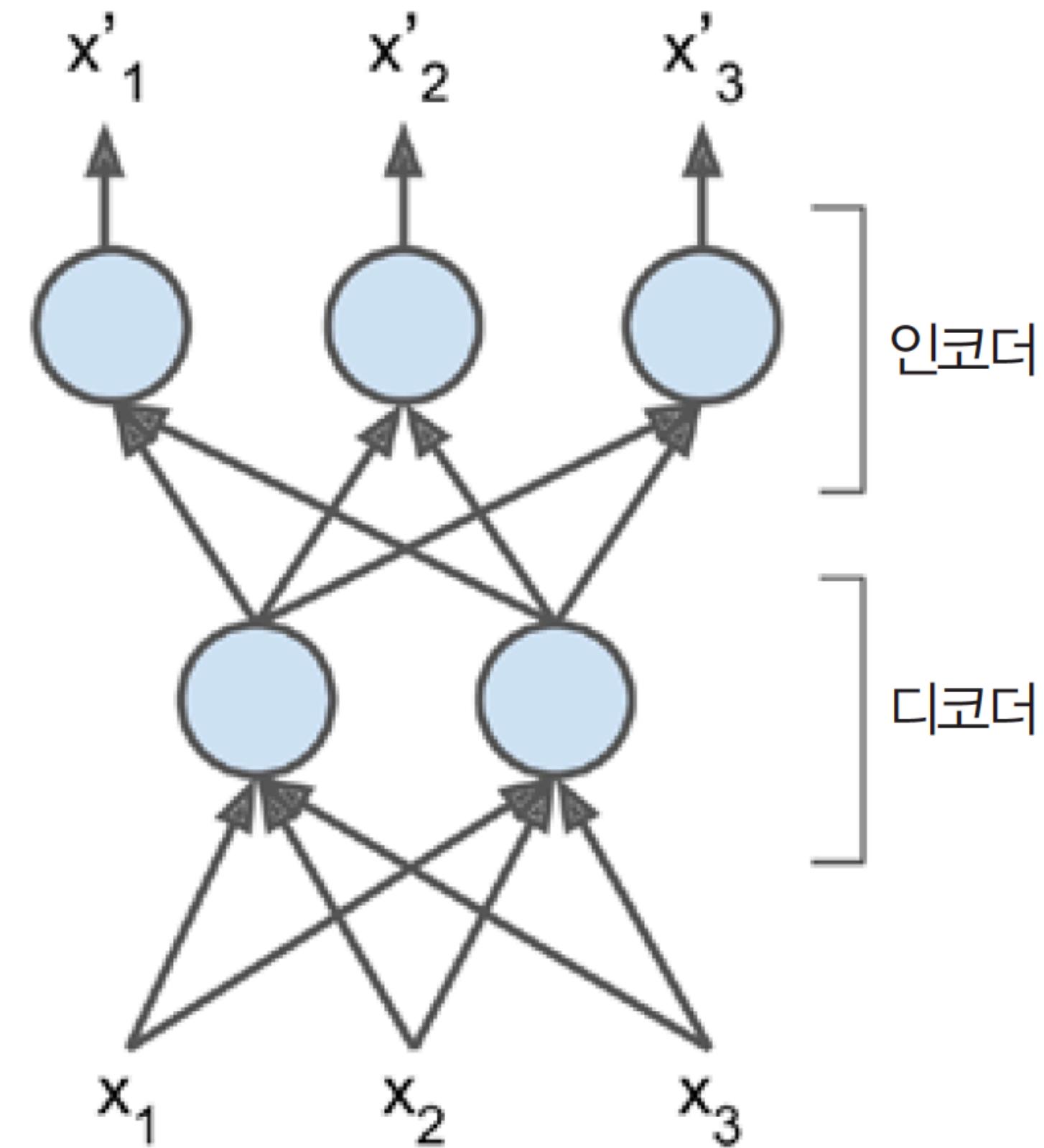
준지도 학습

- Semi-supervised Learning
- 레이블이 있는 데이터(적음)와 레이블이 없는 데이터(많음)를 사용합니다.
- 군집을 통해 데이터를 분할하고 소수 레이블을 이용해 전체 그룹을 인식합니다.
- Google Photos, Facebook 사진 태그
- 제한된 볼츠만 머신(RBM)으로 구성한 심층 신경망(DBN)



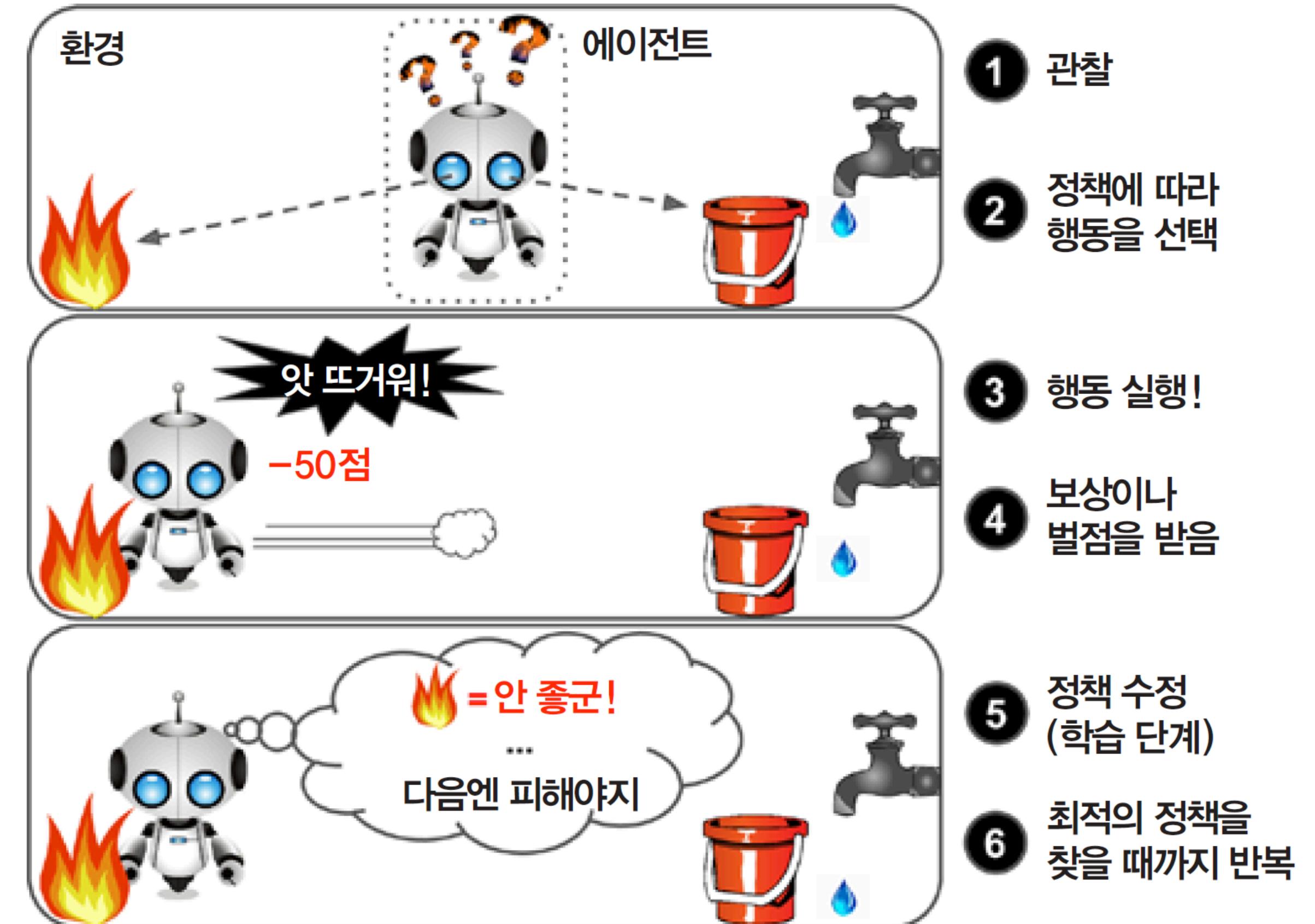
자기지도 학습

- Self-supervised Learning
- 오토인코더(Autoencoder)-15장
- 입력이 타깃이 됩니다.
- 일반적인 레이블이 없기 때문에 비지도 학습, 자기자신이 타깃이기 때문에 자기지도 학습, 어쨌든 타깃이 있으므로 지도 학습으로도 부릅니다.



강화 학습

- 매우 다른 종류의 알고리즘
(사이킷런에 포함되어 있지 않습니다)
- 주어지 환경(environment)에서
에이전트(agent)가 최대의
보상(reward)을 얻기 위해
최상의 정책(policy)을 학습합니다.
- 딥마인드의 알파고(Alpha Go),
아타리(Atari) 게임, 광고?

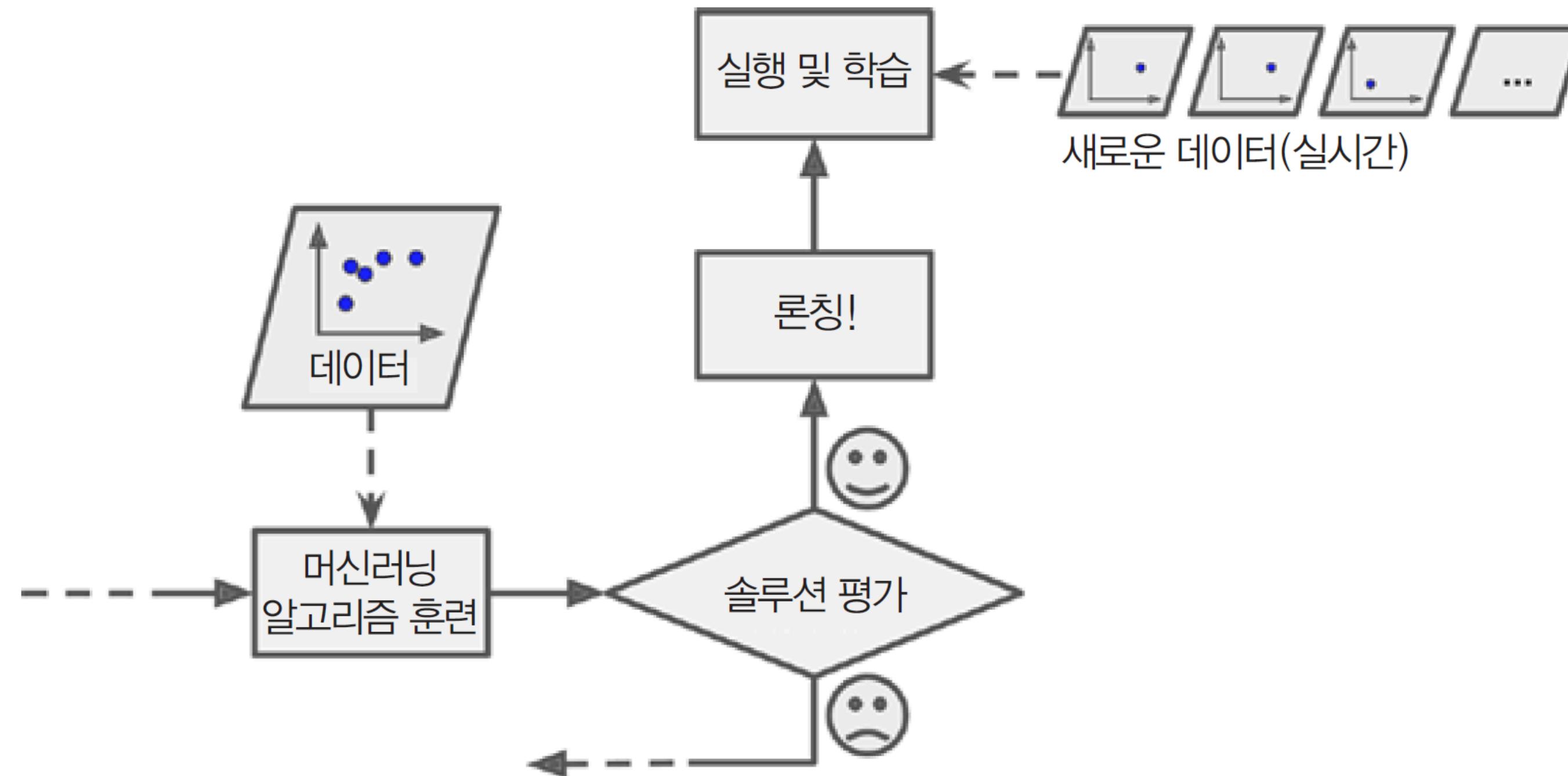


배치(batch) 학습

- 가용한 데이터를 모두 사용하여 훈련시키는 오프라인(offline) 학습법입니다.
- 이전 데이터와 새 데이터를 활용하여 새로운 버전의 모델을 학습합니다.
- 배치 학습도 훈련과 모델 롱칭을 자동화할 수 있습니다.
- 24시간, 혹은 1주일 마다 학습합니다.
- 일반적으로 시간과 자원이 많이 소모됩니다.

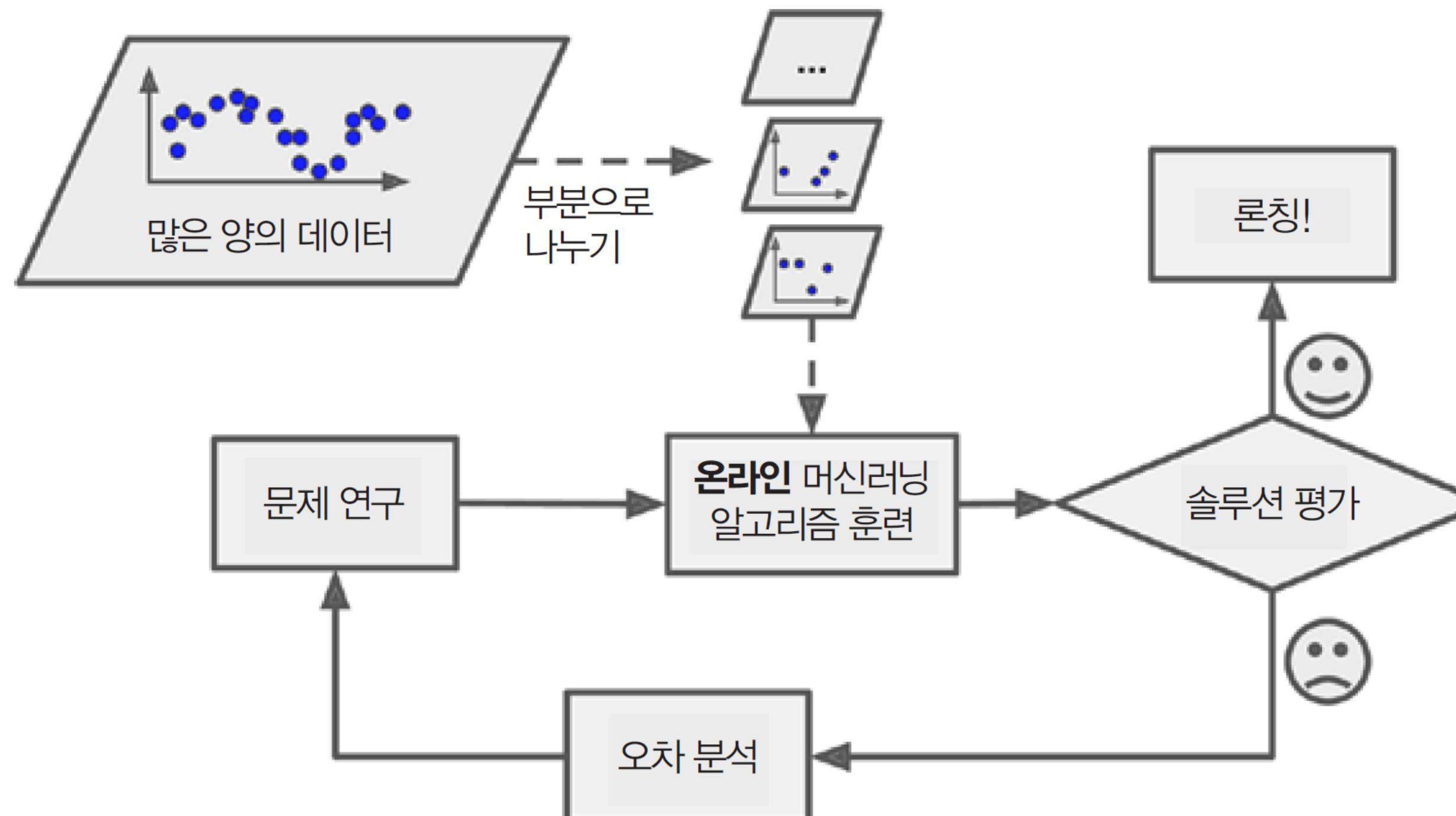
온라인(online) 학습

- 샘플 한 개 또는 미니배치(mini-batch)라 부르는 작은 묶음 단위로 훈련합니다.
- 학습 단계가 빠르고 데이터가 준비되는 대로 즉시 학습할 수 있습니다.(주식 가격?)
- 사용한 샘플을 버릴지 보관할지 결정합니다.



외부 메모리(oob) 학습

- 일부 데이터를 사용하지만 오프라인에서 학습됩니다.
- 점진적 학습(incremental learning)이라고도 부릅니다.

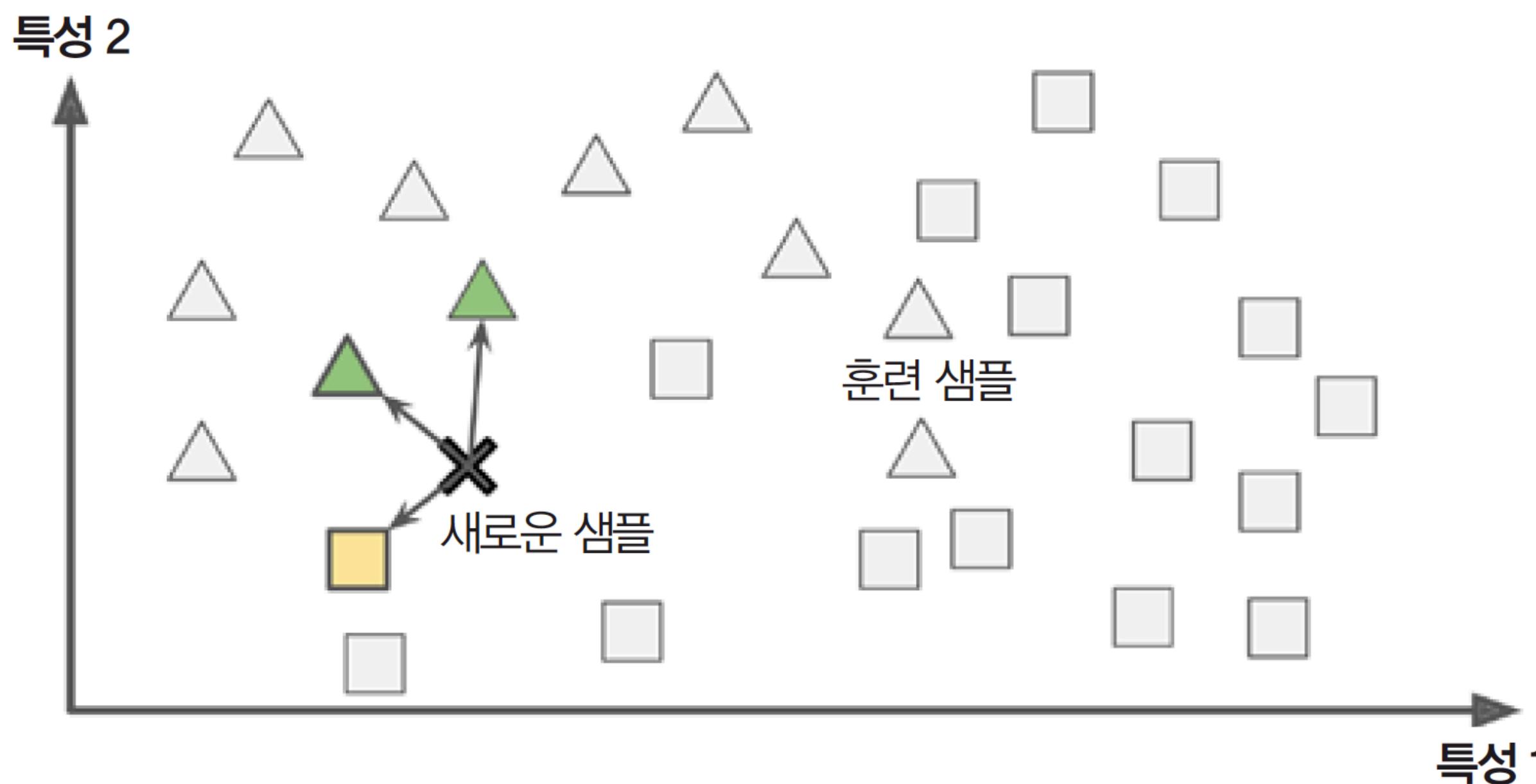


온라인 학습의 주의 사항

- 학습률(learning rate)로 데이터에 얼마나 빠르게 적응할지 제어합니다.
- 나쁜 데이터가 모델 학습에 주입되면 성능이 조금씩 감소합니다.
- 시스템 모니터링이 필요하고 성능 감소가 감지되면 학습을 중지합니다.
- 이전 버전으로 되돌리거나 비정상적인 데이터를 찾습니다.

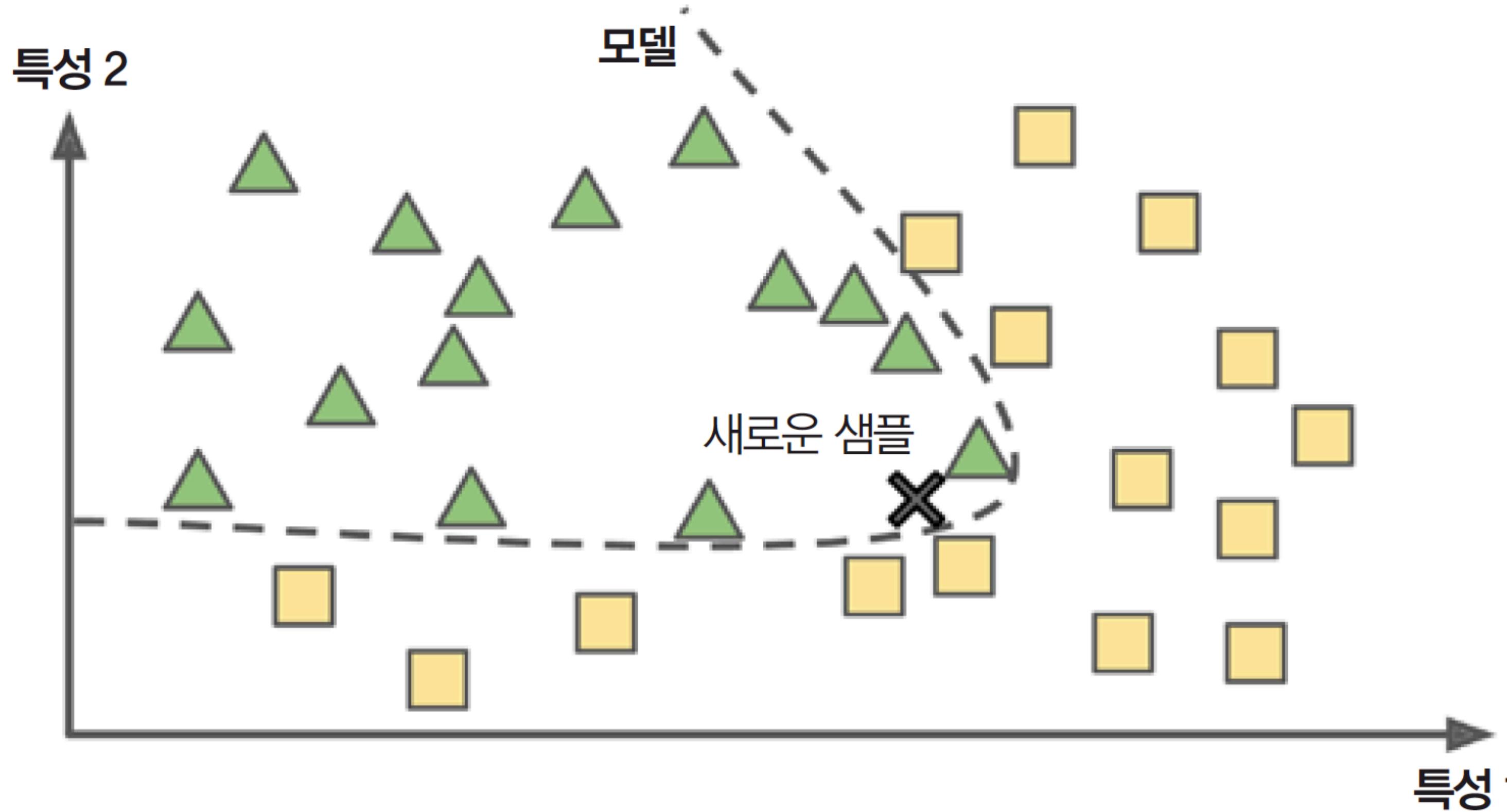
사례 기반 학습

- 일반화 방법에 따라 사례 기반(instance-based)과 모델 기반(model-based) 학습으로 나눕니다.
- 저장된 훈련 데이터에서 가장 가까운 샘플을 찾는 식입니다.(유사도 측정)
- k-최근접 이웃(k-Nearest Neighbors)



모델 기반 학습

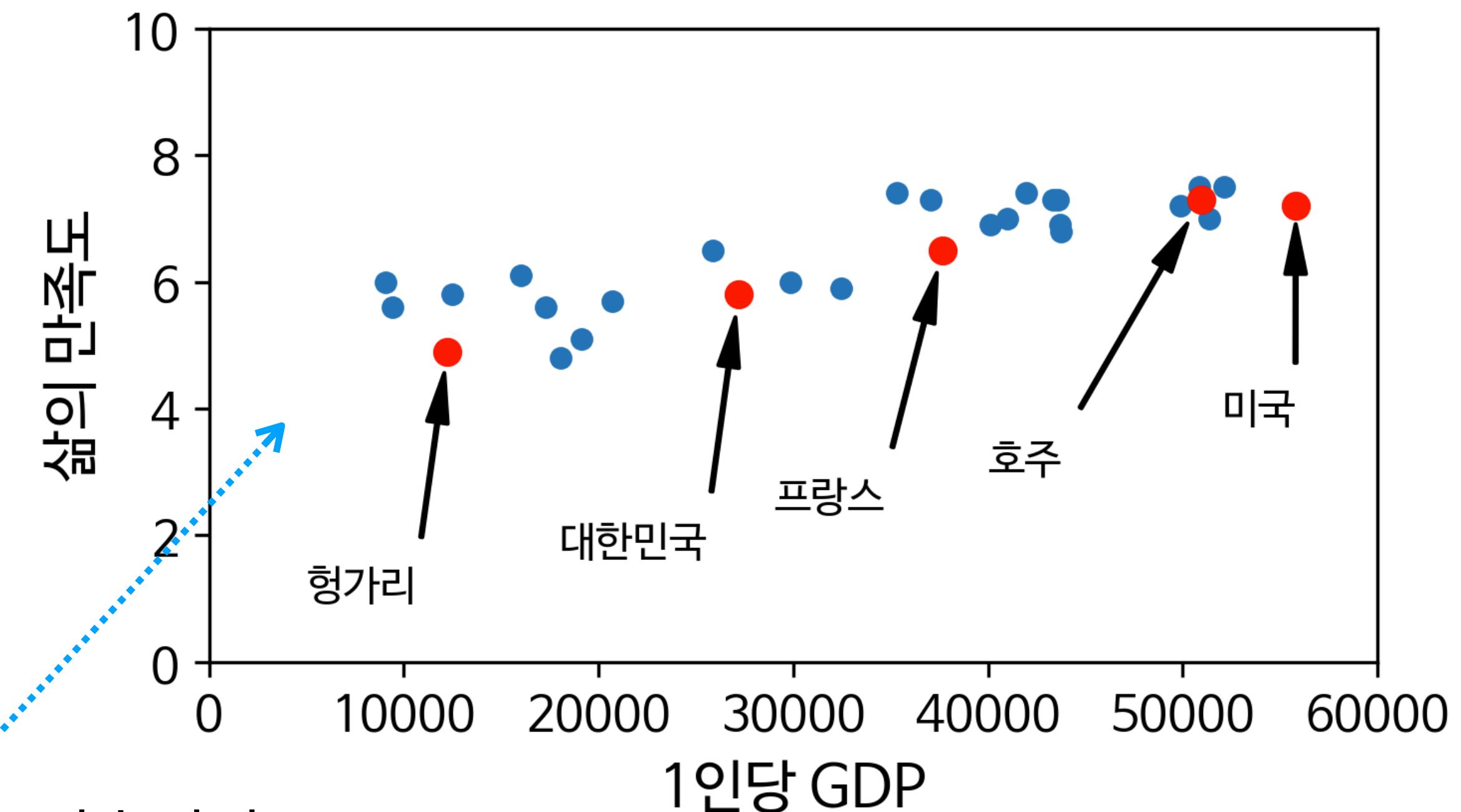
- 모델을 만들어 예측에 사용합니다.(거의 대부분의 머신러닝 모델)



1인당 GDP에 대한 삶의 만족도

- OECD 웹사이트에서 ‘더 나은 삶의 지표’와 IMF 웹사이트에서 ‘1인당 GDP’ 데이터를 다운로드하여 사용합니다.

국가	1인당 GDP(US달러)	삶의 만족도
헝가리	12,240	4.9
대한민국	27,195	5.8
프랑스	37,675	6.5
호주	50,962	7.3
미국	55,805	7.2

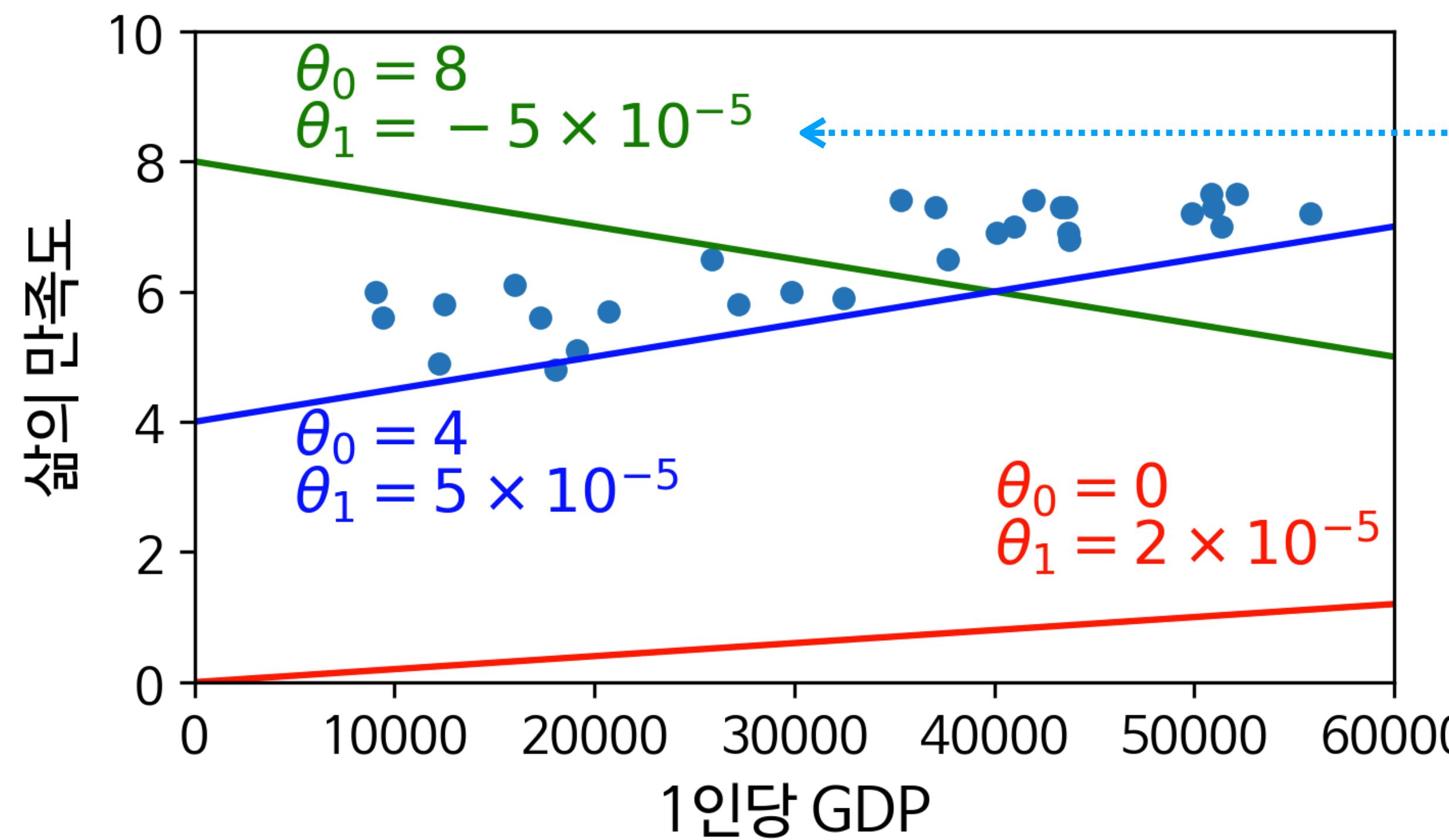


일부 데이터를 제거하고 그렸습니다.
어떤 경향이 보이나요?

모델 선택

- 1인당 GDP의 선형 함수(linear function)로 삶의 만족도를 모델링합니다(선형 회귀).

$$\text{삶의 만족도} = \theta_0 + \theta_1 \times \text{1인당 GDP}$$



모델 파라미터(model parameter)
(β, ω, b)

$$\text{삶의 만족도} = w \times \text{1인당 GDP} + b$$

어떤 모델이 가장 좋은가요?

비용 함수(Cost Function)

- 좋은 모델을 고르기 위해 얼마나 나쁜지를 측정합니다.
- 목적 함수(Object function) > 비용 함수 > 손실 함수(Loss function)
- 선형 회귀에서는 예측과 타깃 사이의 거리가 클수록 나쁜 모델입니다.
- 주어진 데이터에서 비용 함수의 값이 가장 작아지는 모델 파라미터(θ_0, θ_1)를 찾는 과정을 훈련(training) 또는 학습(learning)이라고 합니다.

훈련

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn.linear_model

# 데이터 적재
oecd_bli = pd.read_csv(datapath + "oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv(datapath + "gdp_per_capita.csv", thousands=',', delimiter='\t',
                             encoding='latin1', na_values="n/a")

# 데이터 준비
country_stats = prepare_country_stats(oecd_bli, gdp_per_capita)
X = np.c_[country_stats["GDP per capita"]]
y = np.c_[country_stats["Life satisfaction"]]

# 데이터 시각화
ax = country_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')
ax.set(xlabel="삶의 만족도", ylabel="1인당 GDP")
plt.show()

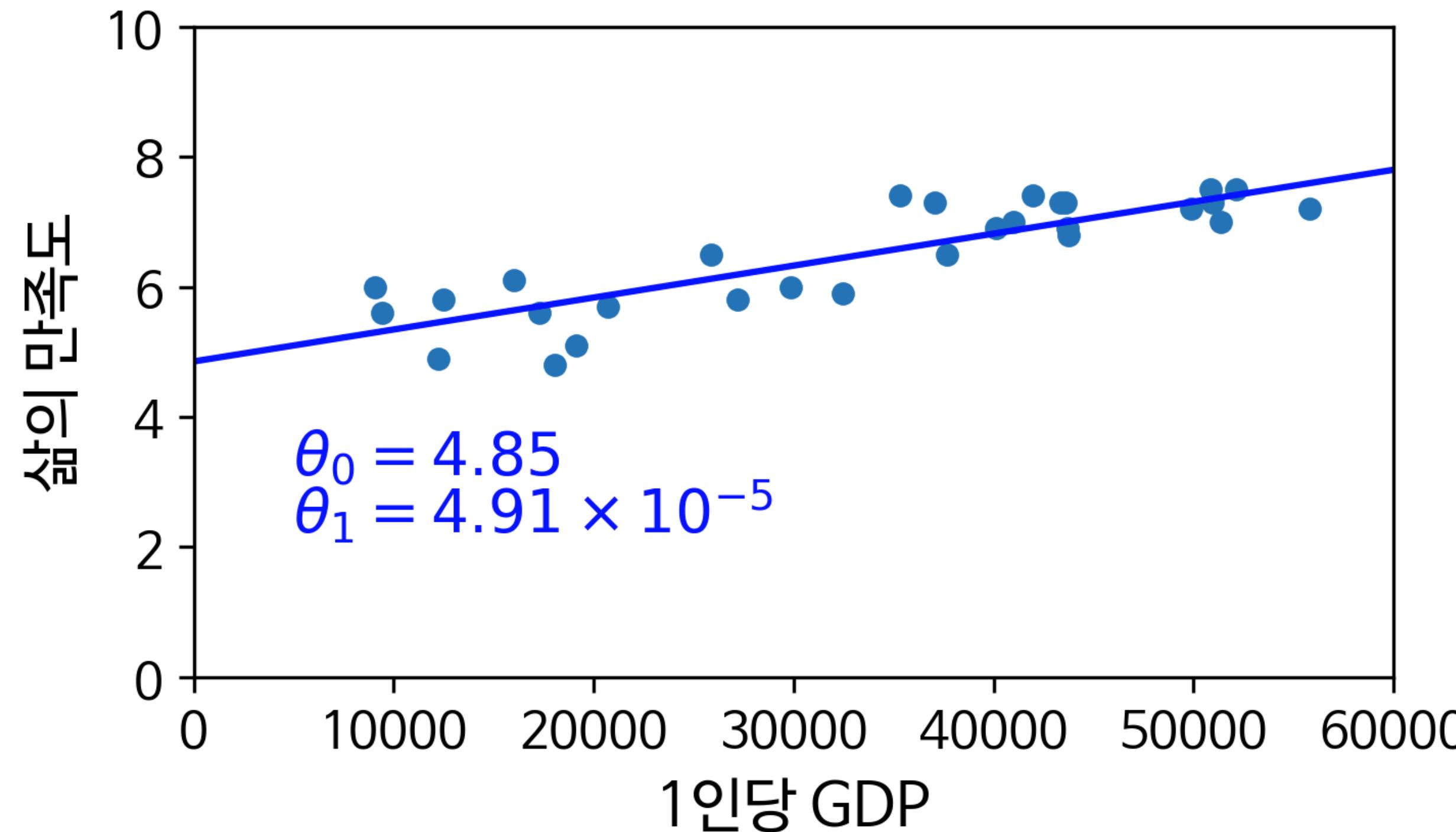
# 선형 모델 선택
model = sklearn.linear_model.LinearRegression()

# 모델 훈련
model.fit(X, y)
```

최적의 모델 파라미터

- `model.intercept_`, `model.coef_`

$$\text{삶의 만족도} = 4.85 + 4.91 \times 10^{-5} \times \text{1인당 GDP}$$



키프로스의 삶의 만족도 예측

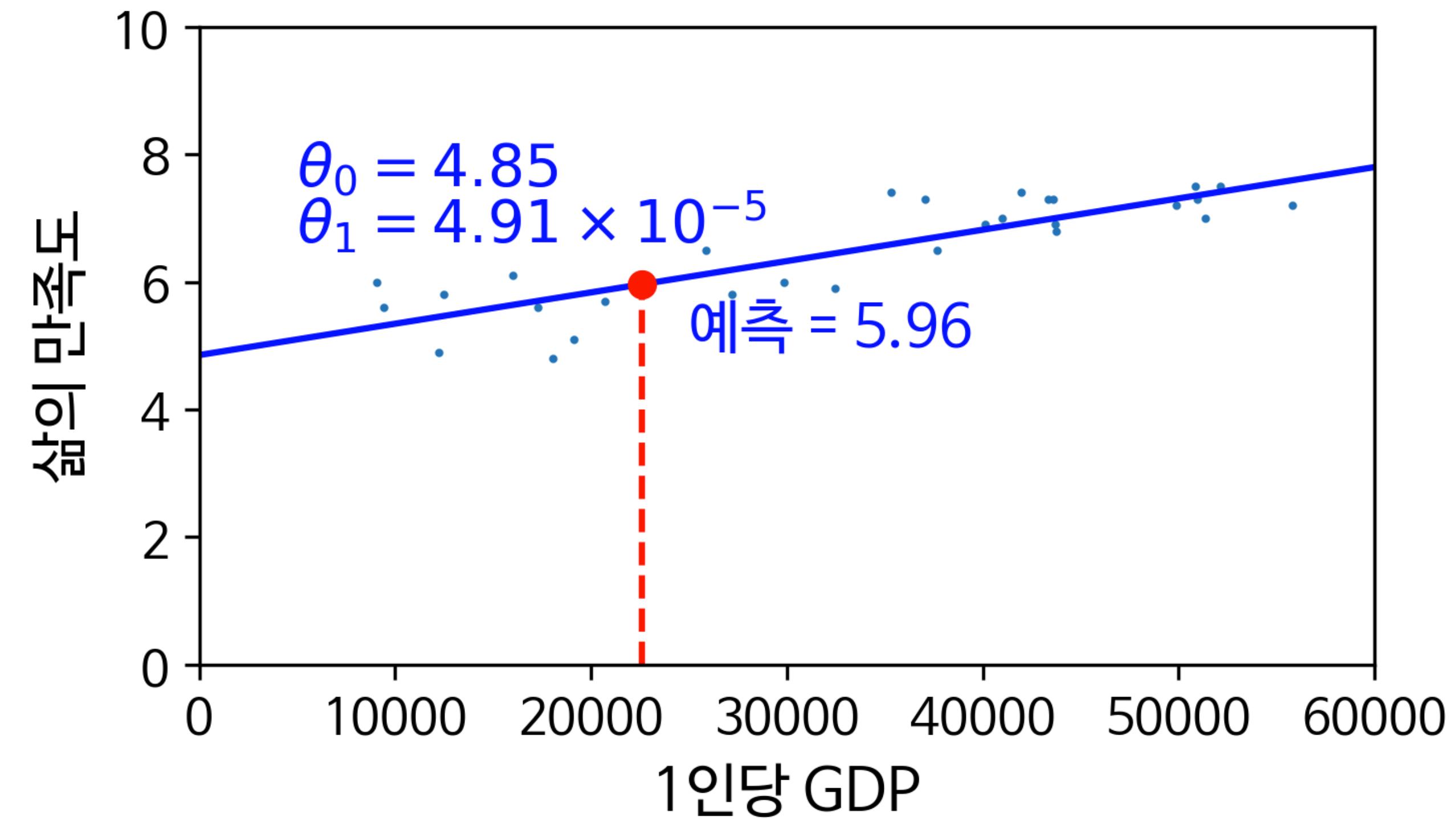
- 키프로스의 1인당 GDP는 \$22,587 입니다.

$$\text{키프로스의 삶의 만족도} = 4.85 + 4.91 \times 10^{-5} \times 22587 = 5.96$$

```
# 키프로스에 대한 예측
x_new = [[22587]] # 키프로스 1인당 GDP
print(model.predict(x_new)) # 결과 [[ 5.96242338]]
```

- Scikit-Learn: `fit(X, y) → predict(X)`

행렬
벡터



k-최근접 이웃을 사용하면?

- k=3일 때, 키프로스와 가장 가까운 슬로베니아, 포르투갈, 스페인의 삶의 만족도를 평균합니다.
- $\text{fit}(X, y) \rightarrow \text{predict}(X)$

```
# 선형 회귀 모델을 k-최근접 이웃 회귀 모델로 교체할 경우
knn = sklearn.neighbors.KNeighborsRegressor(n_neighbors=3)

# 모델 훈련
knn.fit(X, y)

# 키프로스에 대한 예측
print(knn.predict(X_new)) # 결과 [[ 5.76666667]]
```

무언가 잘못 되었다면?

- 더 많은 특성(고용률, 건강, 대기오염 등)을 사용합니다.
- 좋은 훈련 데이터를 더 많이 모읍니다.
- 더 강력한 모델(다항 회귀 모델 등)을 선택합니다.

머신러닝 작업 요약

- 데이터를 분석합니다.
- 모델을 선택합니다.
- 훈련 데이터로 모델을 훈련시킵니다. 즉 비용 함수를 최소화하는 모델 파라미터를 찾습니다.
- 새로운 데이터에 대한 예측(추론)을 만듭니다. 좋은 일반화를 기대합니다.
- 그리고 반복!

추론 vs 예측

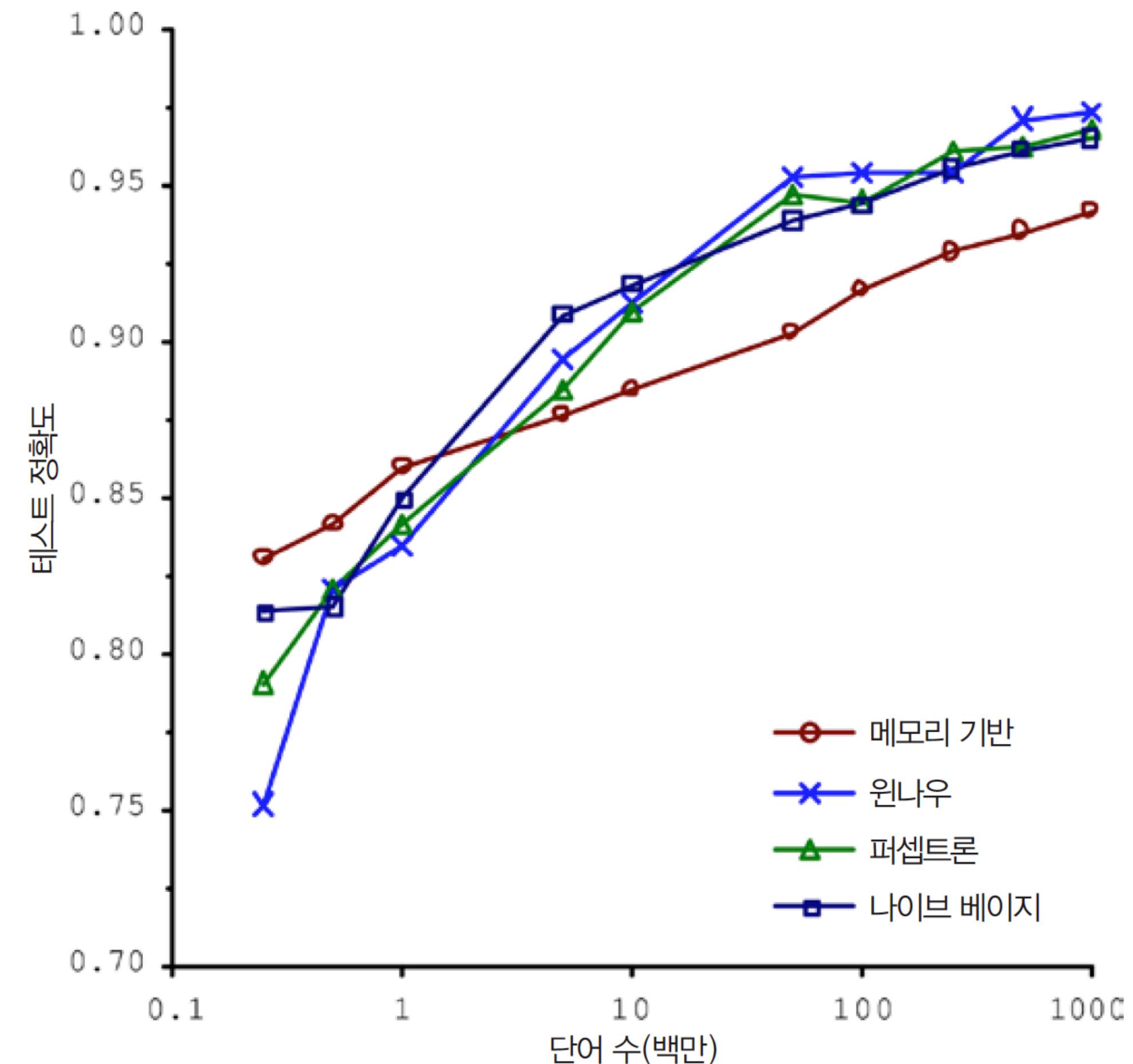
- 추론(inference)과 예측(prediction)은 거의 같습니다.
- 추론은 특성과 타깃 간의 관계에 집중하고 예측은 결과에 관심을 둡니다.
- 종종 두 용어를 특별히 구분하지 않고 혼용합니다.

머신러닝의 도전 과제

- 나쁜 데이터
 - 충분하지 않은 훈련 데이터
 - 대표성 없는 훈련 데이터
 - 낮은 품질의 데이터
 - 관련 없는 특성
- 나쁜 알고리즘
 - 과대적합
 - 과소적합

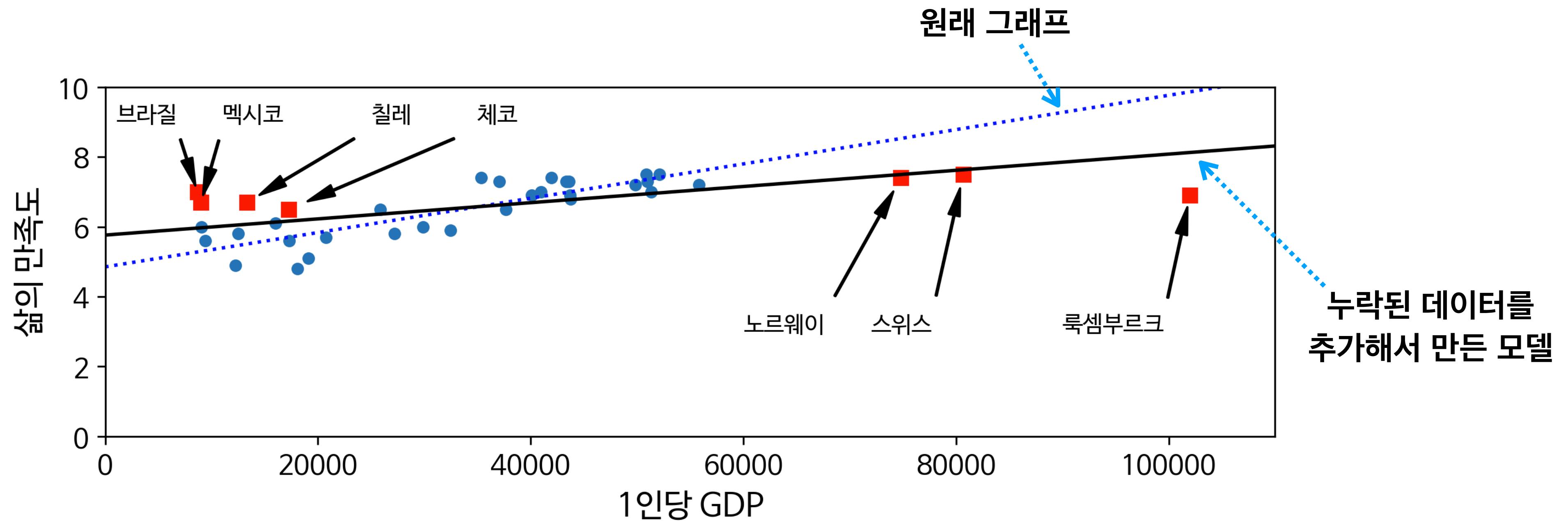
충분하지 않은 훈련 데이터

- 어린 아이는 ‘사과’ 샘플 몇 개를 보고도 모든 종류의 사과를 쉽게 일반화합니다.
- 머신러닝은 아직 많은 데이터가 필요합니다
(정형화된 데이터⇒이미지인식⇒음성인식).
- 2001년 MS 연구자들은 충분한 데이터가 주어지면 알고리즘들이 복잡한 자연어 처리 문제를 거의 비슷하게 잘 처리한다는 것을 보였습니다.
(알고리즘과 말뭉치 사이의 트레이드오프)
- 2009년 피터 노르빅의 “The Unreasonable Effectiveness of Data” 논문으로 널리 알려짐



대표성 없는 훈련 데이터

- 삶의 만족도 모델에 누락된 데이터를 추가하면 그래프가 달라집니다.



- 샘플링 잡음(sampling noise): 샘플이 작거나 이상치
- 샘플링 편향(sampling bias): 표본 추출 방법이 잘못 된 경우

가장 유명한 샘플링 편향 사례

- 1936년 랜던과 루즈벨트의 대통령 선거에서 Literary Digest 사의 여론조사(랜던 57% 예측)
- 실제로는 루즈벨트가 60.8% 득표로 당선되었습니다.
 - 전화번호부, 구독자 명부, 클럽 회원 명부를 사용했습니다.
 - 25%의 응답률로 표본이 편중되었습니다(비응답 편향)
- 다른 예: 구글을 사용한 데이터 수집은 인기 위주의 샘플링 편향을 만듭니다.



낮은 품질의 데이터

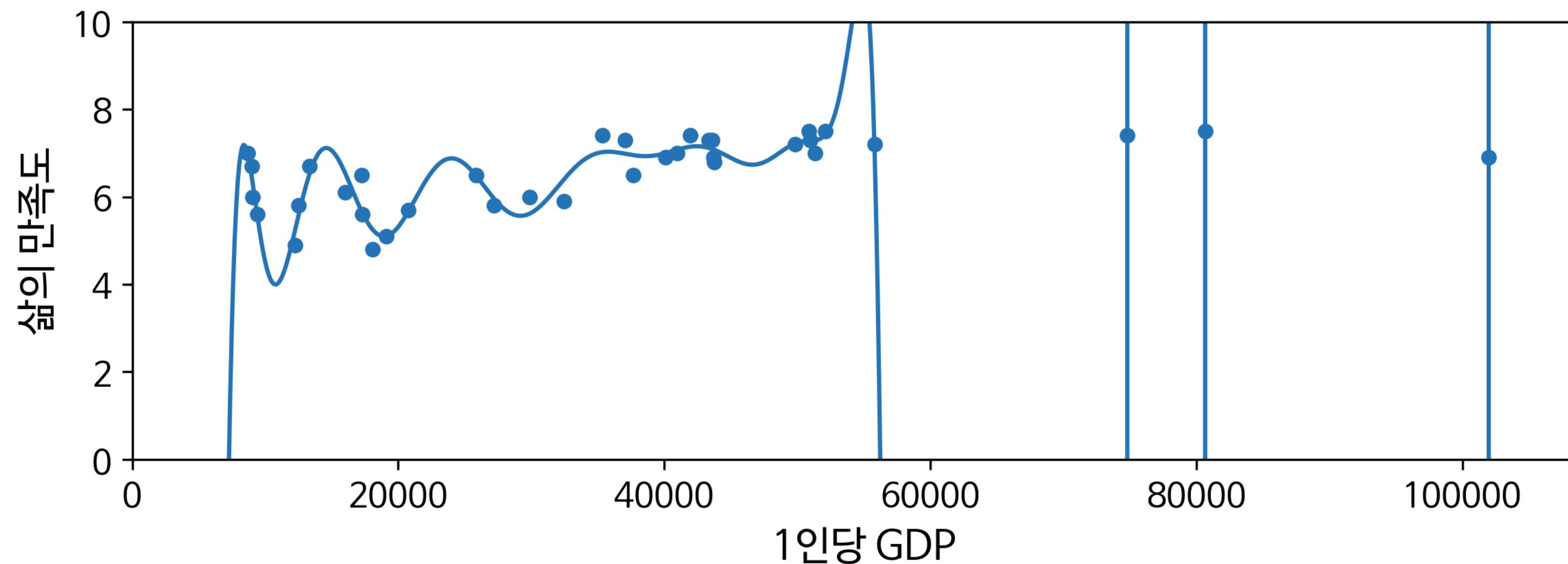
- 일부 샘플이 이상치일 경우 무시하거나 수동으로 고칩니다.
- 일부 샘플에서 특성이 몇 개 빠져 있다면, 특성 전체를 무시할지, 샘플을 무시할지, 빠진 값을 채울지, 이 특성이 넣은 것과 뺀 것을 따로 훈련할지 정해야 합니다.

관련 없는 특성

- 특성 공학(feature engineering)은 주어진 문제와 관련이 높은 특성을 찾습니다.
 - 특성 선택: 가지고 있는 특성 중에서 가장 유용한 특성을 선택합니다.
 - 특성 추출: 특성을 결합하여 더 유용한 특성을 만듭니다(차원 축소, 다행 회귀 등)
 - 새로운 데이터로부터 새로운 특성을 만듭니다.

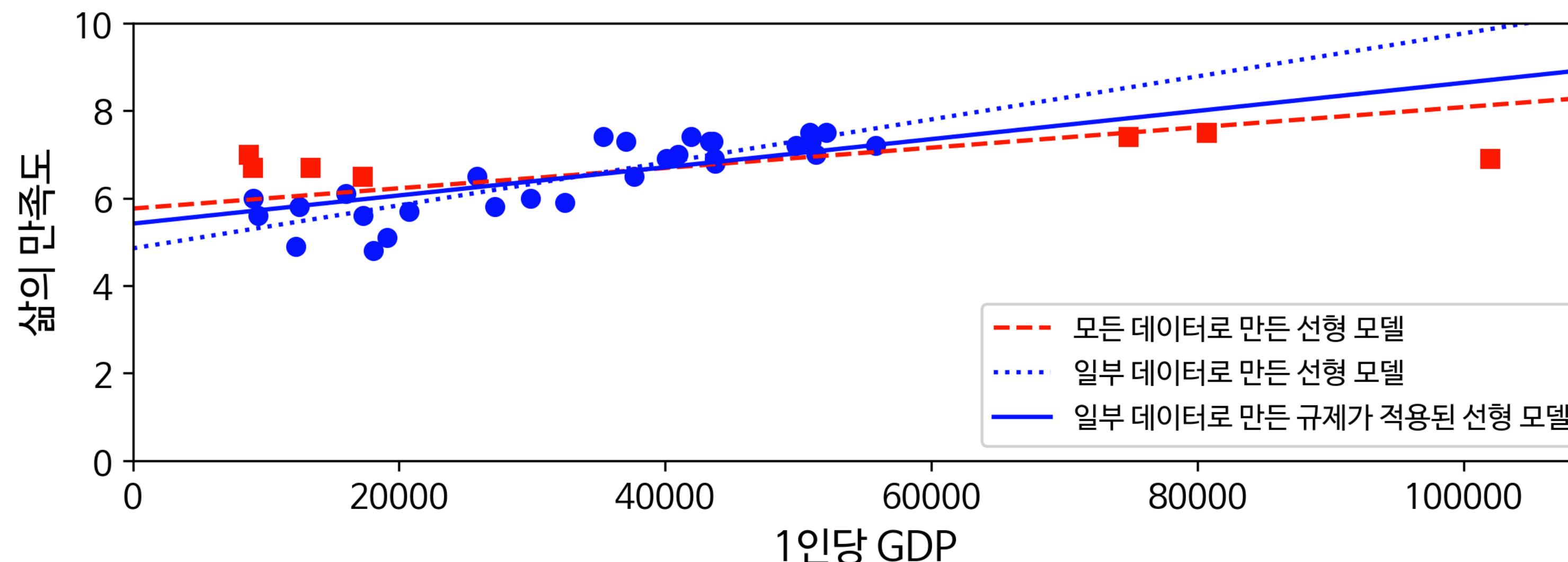
과대적합(overfitting)

- 해외 여행에서 택시 운전사에게 속았다면 그 나라의 모든 택시를 의심하게 됩니다 (일반화의 오류).
- 너무 복잡한 모델이 훈련 데이터에만 잘 들어 맞는 경우를 과대적합되었다고 합니다.
- 잡음이 많거나 샘플이 작으면 일반화가 잘 되지 못합니다.



과대적합을 피하려면

- 모델 파라미터 개수가 적은 모델을 선택하거나, 특성 수를 줄이거나, 모델에 규제(regularization)를 추가합니다.
- 규제는 모델 파라미터의 값(θ_1)을 작게하거나 0으로 만듭니다(하이퍼파라미터(hyperparameter)).
- 훈련 데이터를 더 많이 모으거나 잡음을 줍니다(오류 데이터 수정, 이상치 제거 등)



과소적합(underfitting)

- 과대적합의 반대입니다. 모델이 너무 단순해서 적절한 패턴을 학습하지 못합니다.
- 해결 방법
 - 모델 파라미터가 더 많은 강력한 모델을 선택합니다.
 - 학습 알고리즘에 더 좋은 특성을 제공합니다(특성 공학).
 - 모델의 제약을 줄입니다(규제 하이퍼파라미터를 감소시킵니다).

한걸음 물러서서

- 머신러닝은 명시적인 규칙을 코딩하지 않고 데이터에서 학습합니다.
- 지도 학습 - 비지도 학습, 배치 학습 - 온라인 학습, 사례 기반 학습 - 모델 기반 학습
- 학습 알고리즘은 주입된 데이터를 잘 표현하는 모델 파라미터를 찾습니다.
- 훈련 세트가 너무 작거나, 대표성이 없거나, 잡음이 많거나, 관련 없는 특성이 많다면 올바른 모델을 학습하지 못합니다.
- 모델이 너무 복잡하거나(과대적합), 단순하지(과소적합) 않아야 합니다.

테스트 세트와 검증 세트

- 훈련된 모델을 검증하기 위해 따로 떼어 놓은(훈련에 사용하지 않은) 테스트 세트를 사용합니다.
- 훈련 세트: 80%, 테스트 세트: 20%
- 테스트 세트를 사용하여 하이퍼파라미터를 조정하면 모델이 테스트 세트에 과대적합되어 실제 서비스에서 성능이 기대한 것보다 감소합니다.
 - 훈련 세트, 검증 세트, 테스트 세트로 나누고 검증 세트로 하이퍼파라미터를 조정합니다.
 - 훈련 세트와 검증 세트를 교번하여 여러번 검증 점수를 계산합니다(교차 검증(cross-validation)).

공짜 점심 없음(no free lunch)

- 1996년 데이비드 월퍼트가 어떤 가정도 없다면 특정 모델이 뛰어나다고 판단할 근거가 없다는 이론입니다.
- 주어진 데이터셋에 선형 모델이 잘 맞을지 신경망이 잘 맞을지 경험하기 전에 알 수 없습니다(결국 모두 시도해 보세요).
- 하지만 보편적인 가이드는 정형화된 데이터에는 트리기반 앙상블과 지각에 관련된 데이터(이미지, 텍스트, 사운드)에는 신경망이 좋은 결과를 만듭니다.

감사합니다

-질문-