1주차 강의자료

PUSAN NATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Software & System Security Laboratory

주차 별 교육 계획

1주차	<연구 내용 소개> 연구실 소개, 수업 내용 소개, 기계학습 알고리즘에 대한 전반적 소개 - 연구실 핵심 연구 분야 소개 - 파이썬 주요 과학 라이브러리(넘파이, 판다스 등) 소개 - 실습환경 구축
2주차	<교과 및 과제 질의 응답> 수업 내용 및 제출 과제에 대하여 질의 응답 수행 (선택) - 연구실 개별 방문 및 질의 응답
3주차	<기계학습 알고리즘 이론> 기계학습 알고리즘 종류, 동작 원리와 특징 설명 - 다양한 기계학습 종류 (지도, 비지도, 준지도, 강화 학습) 소개 - 기계학습 주요 도전 과제 설명
4주차	<기계학습 프로젝트 수행하기> 예제 프로젝트를 활용하여 처음부터 끝까지 기계학습 프로젝트 프로그램 수행 설명 - 시스템 설계(문제 정의, 성능 측정 지표 선택, 가정 검사) - 공개 데이터 가져오기, 데이터 탐색 및 데이터 수집 방법 - 모델 선택 및 훈련 방법 (선형 회귀 모델, 결정 트리 등) - 모델 튜닝 (그리드 탐색, 랜덤 탐색, 앙상블 방법 등) **[실습 과제물 제공]**
5주차	<딥러닝 프로젝트 수행하기> 심층 신경망을 활용한 예제 딥러닝 프로젝트 프로그램 작성 방법 설명 - 인공 신경망 개요 - 심층 신경망 훈련 코드 작성 방법 **[실습 과제물 수거]**

요약

❖ 실습 환경 구축

- Anaconda 및 Jupyter 설치

❖ 파이썬 언어의 기초인 다양한 데이터 타입 활용

- Built-in Data Types
  - 숫자, 시퀀스, 매핑, 파일, 클래스, 인스턴스 및 예외 데이터
- Numeric Types
  - 정수형, 실수형, 복소수형, 연산자
- Sequence Types
  - strings, unicode strings, lists, tuples, bytearrarys, buffers, and xrange objects
- Mapping Types
  - Dictionary
- 제어문
  - if, while, for, try
- 파이썬 함수의 구조
  - def 키워드, 입력 매개변수, 모듈

요약

- ❖ 머신러닝 이란?
- ❖ 왜 머신러닝을 사용하는가?
- ❖ 머신러닝 시스템의 종류 : 지도 학습 / 비지도 학습 /준지도 학습 /강화 학습
- ❖ 배치 학습과 온라인 학습
- ❖ 사례 기반 학습과 모델 기반 학습
- ❖ 주요 도전 과제

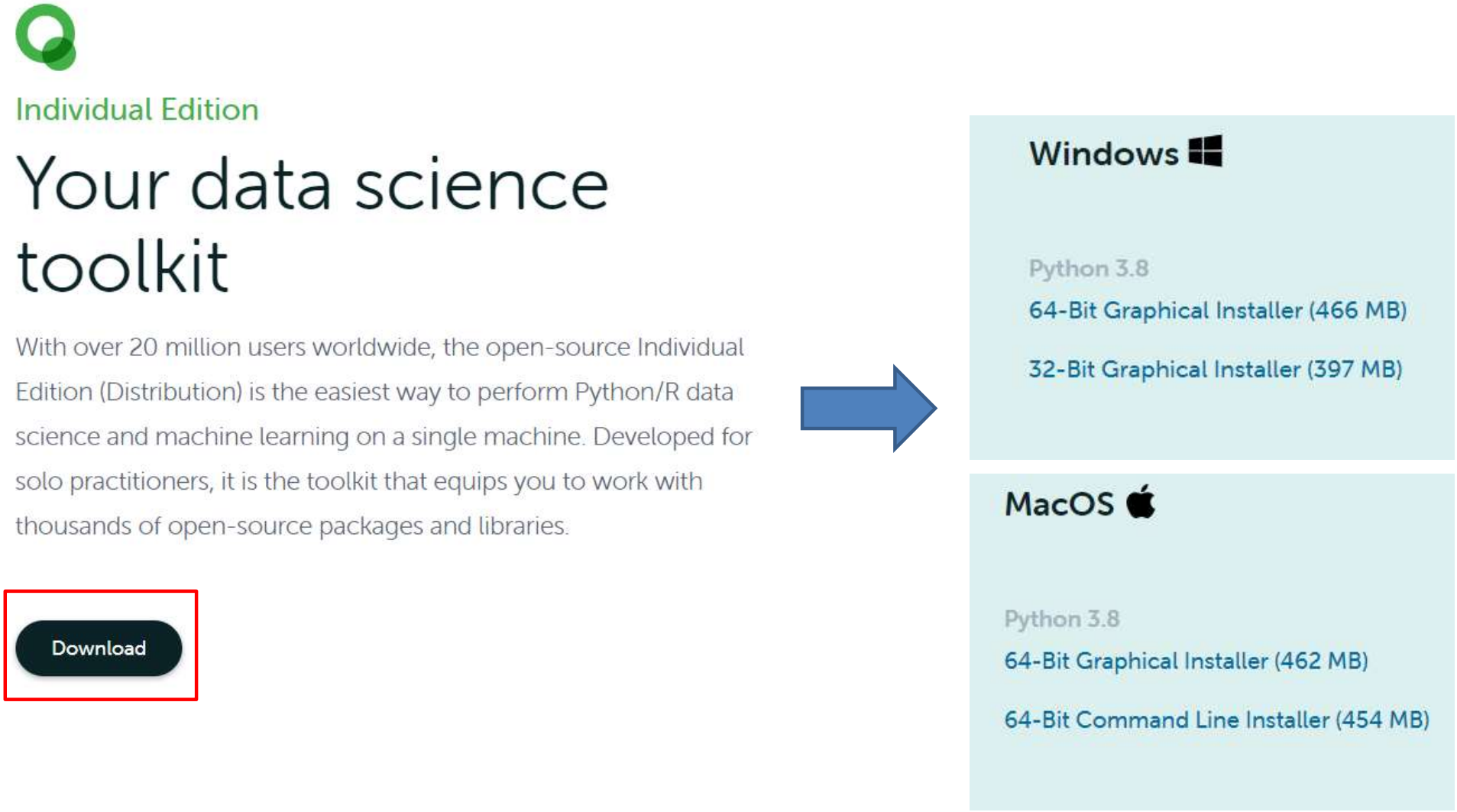
Introduction to Python and Machine Learning Environment

PUSAN NATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Software & System Security Laboratory

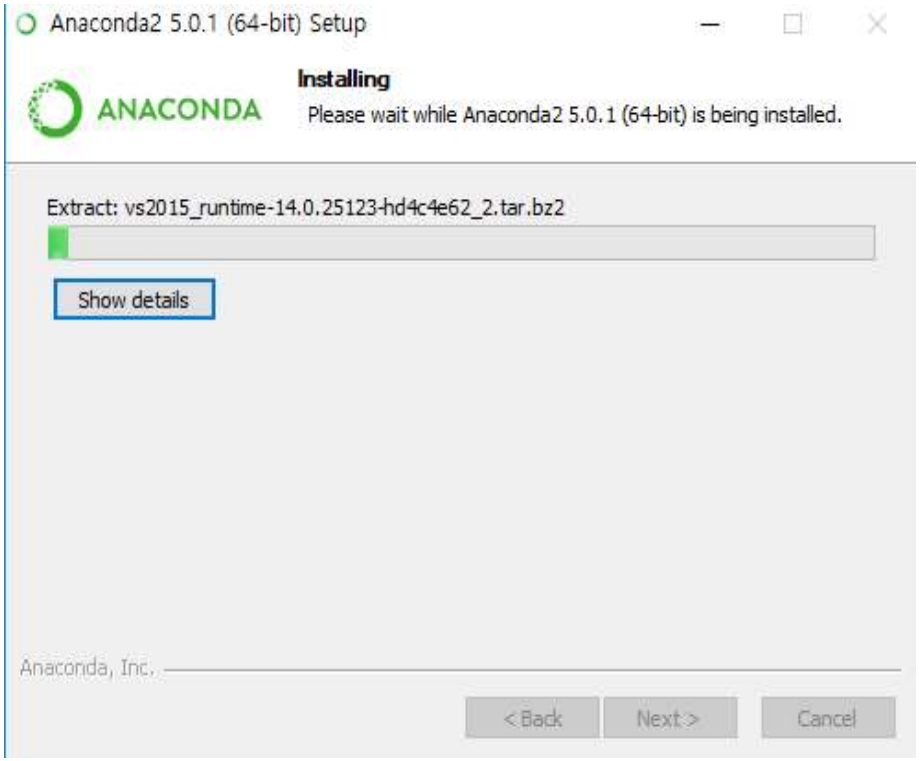
실습 환경 구축 (1/3)

❖ 1단계: https://www.anaconda.com/products/individual 에서 IPython Notebook (Jupyter Notebook)을 다운받아 설치한다.



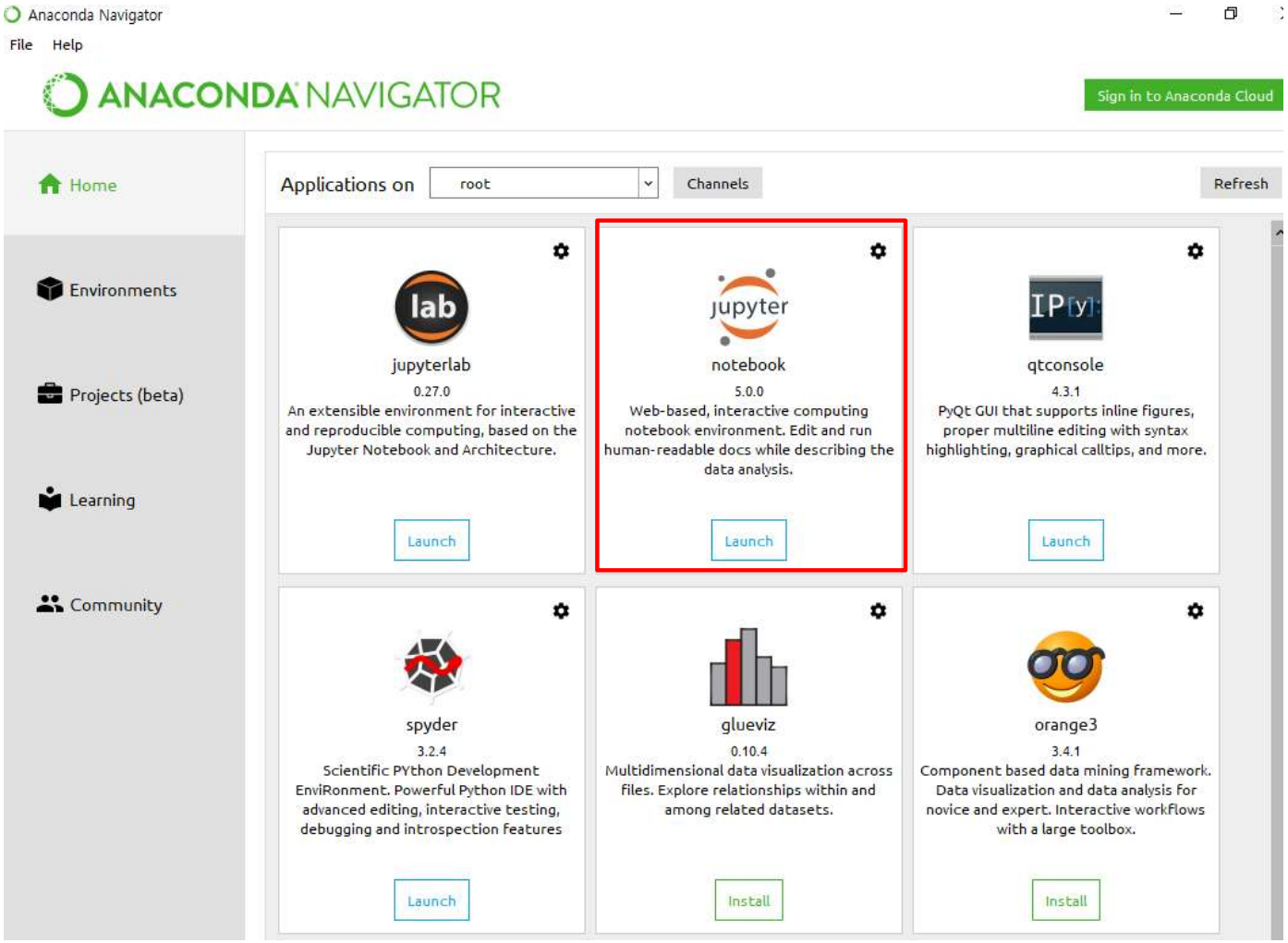
실습 환경 구축 (1/4)

❖ 2단계: 다운로드 한 Anaconda 파일 실행 후 설치 지침에 따라 Anaconda를 설치한다.



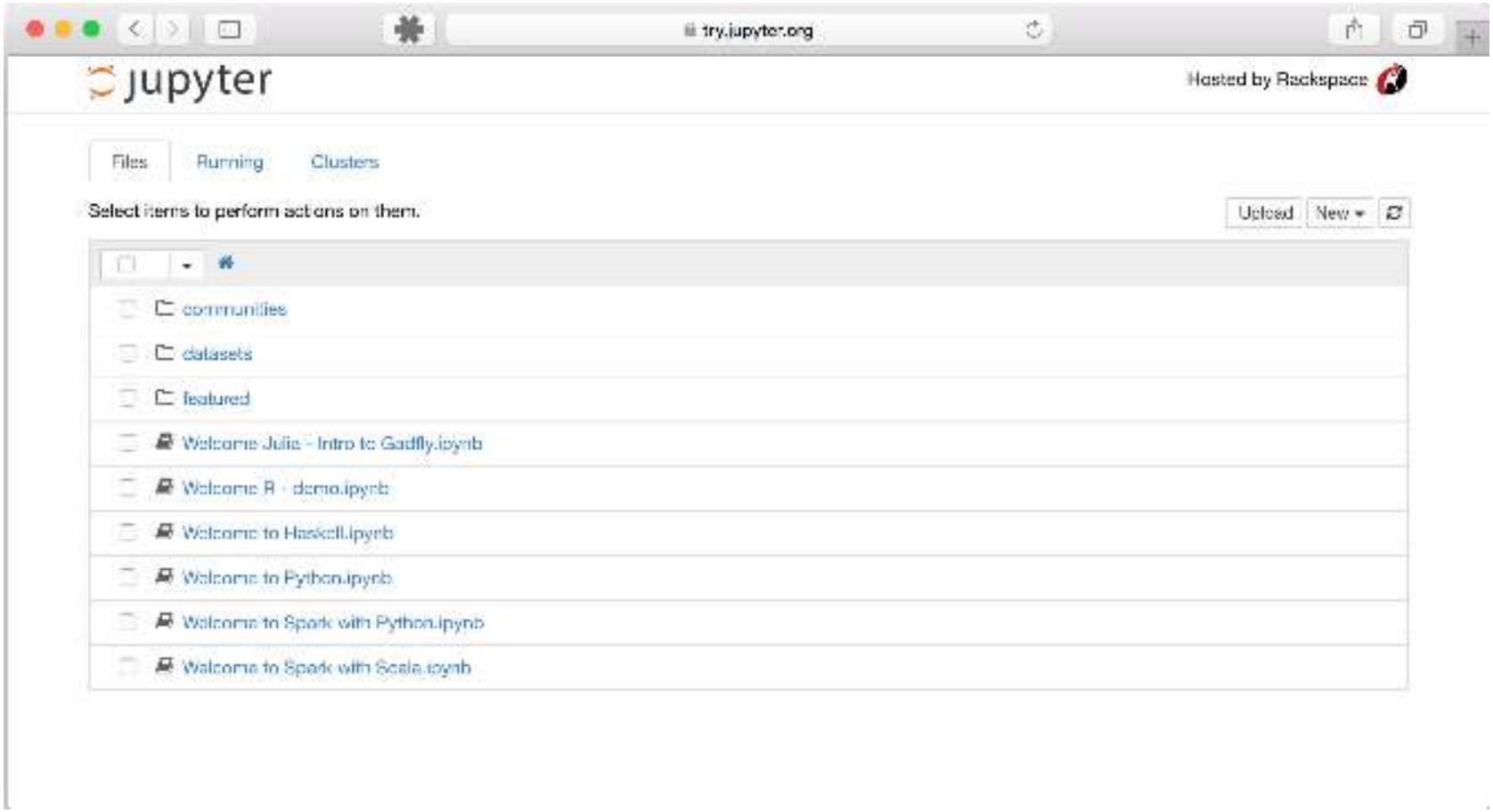
실습 환경 구축 (2/4)

❖ 3단계: IPython Notebook (Jupyter Notebook)을 컴퓨터에 설치하면 프로그램 목록에서 “Anaconda Navigator”를 실행하고 “Jupyter Notebook” 아이콘을 클릭하여 Notebook 서버를 실행



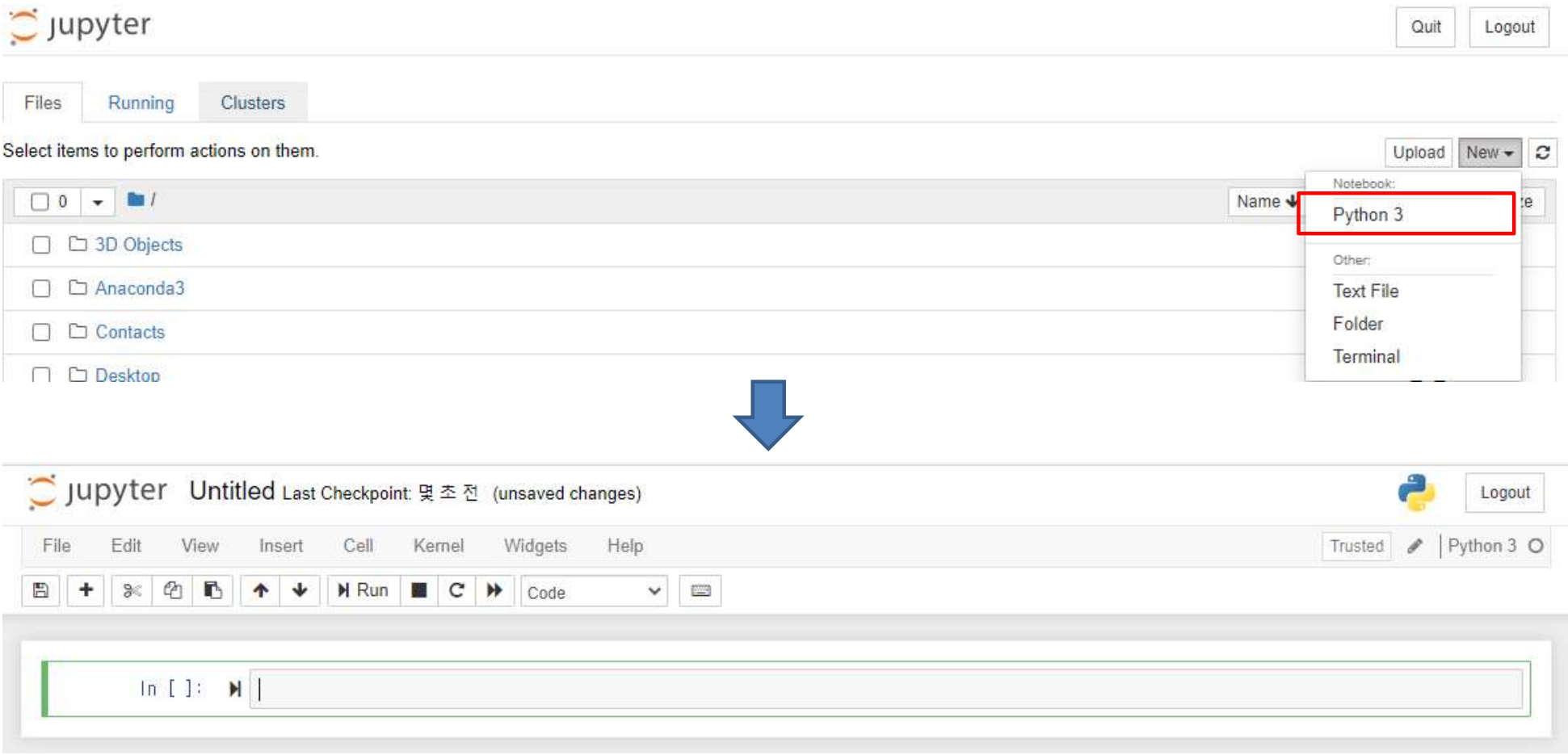
실습 환경 구축 (3/4)

❖ 정상적으로 Notebook 서버를 시작하면 웹 응용 프로그램의 URL (기본적으로 http://localhost:8889/tree)을 포함하여 노트북 서버에 대한 정보가 인쇄되면서 다음과 같은 Notebook Dashboard가 열림



실습 환경 구축 (4/4)

❖ Notebook Dashboard가 열리면 우측의 New 버튼을 누른 후, Python 3 탭을 클릭하여 코드 타이핑이 가능한 Jupyter 환경을 확인 가능



필수 라이브러리와 도구들

❖ 파이썬의 주요 과학 라이브러리

- 넘파이(NumPy), http://www.numpy.org
  - 파이썬으로 과학 계산을 하려면 꼭 필요한 패키지: 선형 대수 연산, 푸리에 변환, 유사 난수 생성기 포함
  - Numpy 배열(ndarrary 클래스)은 Scikit-learn의 기본 데이터 구조
- 판다스(Pandas), http://pandas.pydata.org
  - 데이터 처리와 분석을 위한 패키지
  - 엑셀의 스프레드시트와 비슷한 테이블 형태인 DataFrame을 사용
- 맷플롯립(Matplotlib), http://matplotlib.org
  - 파이썬의 대표적인 과학 계산용 그래프 라이브러리

❖ 미적분, 선형대수, 확률, 통계 등의 수학 지식

❖ 파이썬 프로그래밍

❖ 주피터(Jupyter) Notebook

- 프로그램 코드를 브라우저에서 실행해주는 대화식 환경
- 탐색적 데이터 분석에 적합함

필수 라이브러리와 도구들

❖ 넘파이(NumPy), http://www.numpy.org

```
In [1] :    import numpy as np

            x = np.array( [[1, 2, 3], [4, 5, 6]])
            print("x:₩n{}".format(x))

Out [2] :   x:
            [[1 2 3]
             [4 5 6]]
```

❖ 판다스(Pandas), http://pandas.pydata.org

```
In [1] :    from IPython.display import display
            import pandas as pd

            data = {'Name': ["John", "Anna", "Peter", "Linda"],
                    'Location': ["New York", "Paris", "Berlin", "London"],
                    'Age': [24, 13, 53, 33]
                   }
            data_pandas = pd.DataFrame(data)
            display(data_pandas)
```

	Name	Location	Age
0	John	New York	24
1	Anna	Paris	13
2	Peter	Berlin	53
3	Linda	London	33

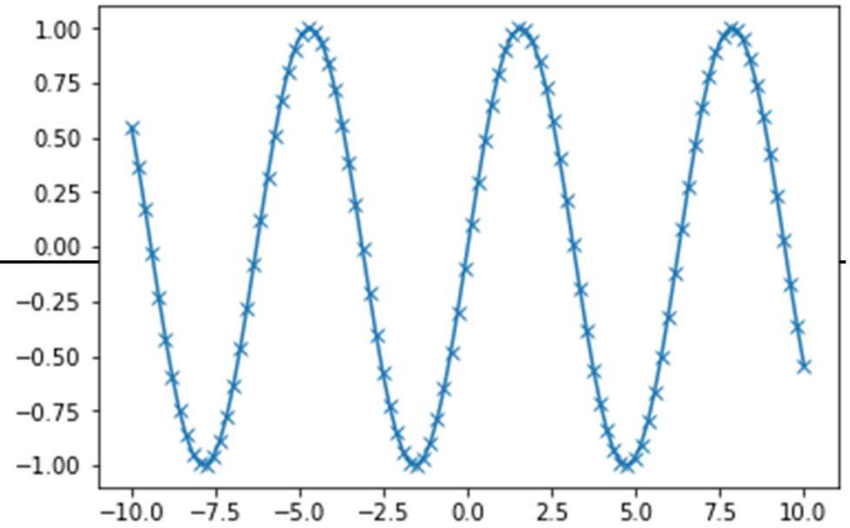
❖ 라이브러리 설치가 되어 있지 않은 경우 코드 상단에 '!pip install 라이브러리이름 ' 을 입력하여 주피터에서 설치가능함 ex) !pip install pandas

필수 라이브러리와 도구들

❖ 맷플롯립(Matplotlib), http://matplotlib.org

```
In [1] :    %matplotlib inline
            import numpy as np
            import matplotlib.pyplot as plt

            x = np.linspace(-10, 10, 100)
            y = np.sin(x)
            plt.plot(x, y, marker="x")
```



❖ SciPy, https://www.scipy.org/scipylib

- 과학 계산용 함수를 모아놓은 파이썬 패키지
- 고성능 선형 대수, 함수 최적화, 신호 처리, 특수한 수학 함수 등을 제공함
- Scikit-learn 알고리즘에서 사용함
  - 예: scipy.sparse를 이용한 희소 행렬 기능

참고 URL

❖ 파이썬 학습

- http://learnpython.org
- https://docs.python.org/ko/3/tutorial

❖ 소스코드 다운로드

- https://github.com/rickiepark/introduction_to_ml_with_python
- https://github.com/rickiepark/handson-ml

❖ 머신러닝 학습

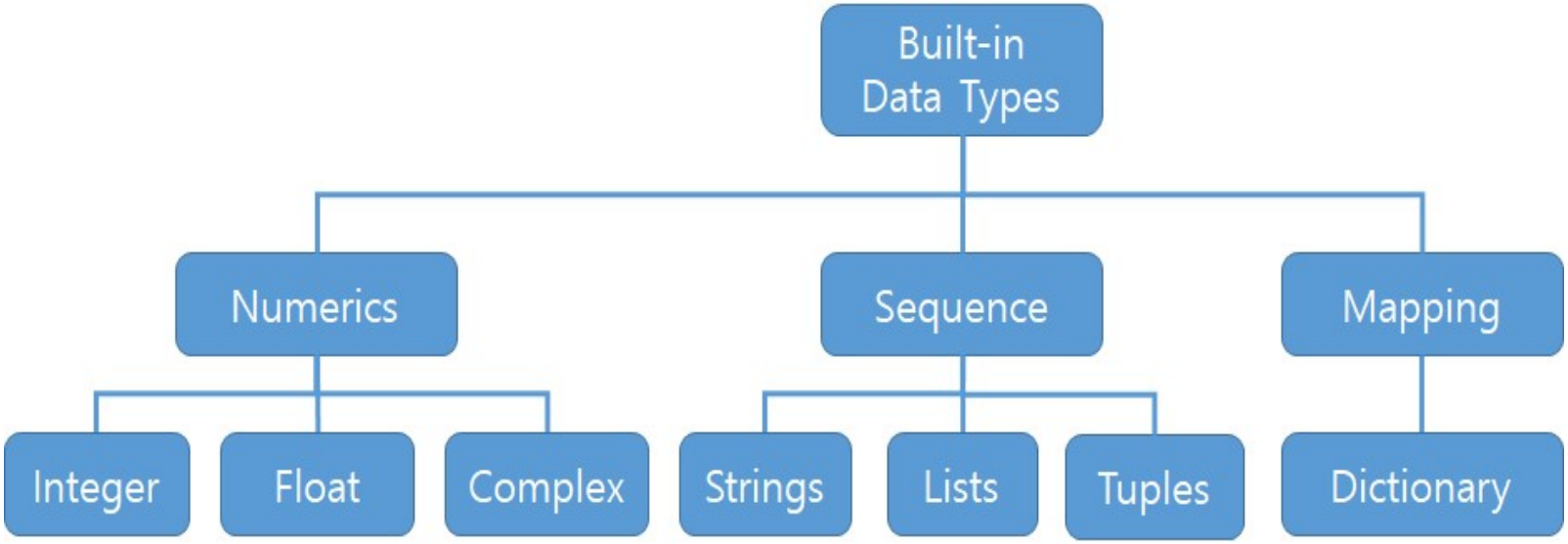
- http://scikit-learn.org/stable/user_guide.html
- https://www.Dataquest.io
- http://goo.gl/GwtU3A - Quora
- http://deapearning.net
- http://goo.gl/7PEkVy - 모두의 머신러닝/딥러닝
- https://goo.gl/WnHK7a - Andrew Ng’ s Coursera Lecture

파이썬 학습하기

- Built-in Data Types
- Numeric Types
- Sequence Types
- Mapping Types
- 제어문
- 파이썬 함수의 구조

Built-in Data Types

❖ 기본 내장 데이터 유형은 숫자, 시퀀스, 매핑, 파일, 클래스, 인스턴스 및 예외 데이터로 구분



❖ 변수

- 변수는 데이터를 저장하는 메모리의 특정 위치에 주어진 이름

변수 선언 및 값 대입 형식	변수명= 값

Numerics Types (1/2)

❖ 정수형

- 정수를 표현하는 자료형

```
1:  val = 123
2:  val = -123
3:  val = 0
```

❖ 실수형

- 소수점이 포함된 숫자를 표현하는 자료형

```
1:  val = 123.0
2:  val = -123.0
3:  val = 0.0
```

- sys.float_info
  - 실수형에 대한 정보가 들어있는 구조체

❖ 복소수형

- 실수와 허수의 합의 꼴로서 표현하는 실수의 개념을 확장한 수

```
1:  val = 1+2i
2:  val = 1-2i
```

Numerics Types (2/2)

❖ 연산자

Types	Operators	Example	Result
Arithmetic	+, −, *, /	x / y	quotient of x and y
	**	x**	x to the power y
	%	x % y	remainder of x / y
	floor division: //	x // y	(floored) quotient of x and y
Comparison	==	x == y	if x is equal to y, then 1, else 0
	!=	x != y	if x is not equal to y, then 1, else 0
	<, >, <=, >=	x < y	if x is less than y, then 1, else 0
Assignment	=	x = y	assignment of y to x
	+=,−+,*=,/=,%=,//=	x += y	assignment of x+y to x
Logics	and	x and y	if x is false, then x, else y
	or	x or y	if x is false, then x, else y
	not	not x	if x is false, then True, else False
bitwise	& (AND)	x & y	bitwise and of x and y
	| (OR)	x | y	bitwise or of x and y
	^ (XOR)	x ^ y	bitwise exclusive or of x and y
	~ (Complement)	~x	the bits of x inverted
	<<(Left Shift)	x << n	x shifted left by n bits
	>> (Right Shift)	x >> n	x shifted right by n bits
Membership	in, not in	x in s	True if an item of s is equal to x, else False

Sequence Types (1/5)

- 여러 요소들로 구성된 집합 자료형으로 각각을 구성하는 원소는 순서를 가짐
- 파이썬은 strings, unicode strings, lists, tuples, bytearrarys, buffers, and xrange objects의 7가지 sequence 자료형을 지원함
- strings
  - 큰 따옴표나 작은 따옴표로 문자열 리터럴을 표현하는 자료형

```
1: val = 'I love you.'
2: val = "I love you."
3: val = '''I
4: love
5: you.'''
6: val = "I love you.\n I love U."
```

Sequence Types (2/5)

❖ Strings

▪ 문자열 포맷팅

- 문자열 리터럴을 화면에 출력할 때는 주로 일정한 포맷으로 문자열 리터럴을 조합하여 출력

Conversion Specifier	Meaning
%s	데이터를 str()을 사용하여 변환
%r	데이터를 repr()을 사용하여 변환
%c	character
%d or %i	integer
%f or %F	float
%e or %E	exponential float
%o or %O	Octal
%x or %X	Hexadecimal

```
1:  val = “answer: %s” % “I love you.”
2:  print(val)
3:  val = “answer: %1.3f” % 3.141592
4:  print(val)
```

Sequence Types (3/5)

❖ lists

- 표 연산자로 항목을 구분하는 대괄호로 표현하는 집합 자료형으로, 새로운 요소를 추가하거나 갱신할 수 있는 Mutable types
- 서로 다른 데이터 형일 수 있으며 원소가 없는 빈 리스트는 square bracket만 사용하여 "[]" 와 같이 표현

리스트 인덱싱	리스트명[인덱스값] #인덱스값: 0, 1, 2, ..., 크기-1

```
1:  val = ["love", 1]
2:  print(val[0])
3:  print(val[1])
4:  print(val[0:1])
5:  val[1] = val[1] + 1
6:  del val[0]
7:  print(val)
```

Sequence Types (4/5)

❖ Tuples

- 괄호를 포함하거나 포함하지 않고 쉼표 연산자로 표현하는 자료형
- 빈 튜플에는 paranthesis '()' 가 있어야하며, 단일 항목 튜플에는 '(d,)' 와 같이 후행 쉼표가 있어야 함
- 원소를 변경할 수 없는 immutable 자료형으로 고정된 원소 값을 저장하는 경우 사용함

튜플 인덱싱	튜플명[인덱스값] #인덱스값: 0, 1, 2, ..., 크기-1

```
1:  val = ("love", 1)
2:  print(val[0])
3:  print(val[1])
4:  print(val[0:1])
5:  print(val)
```

Sequence Types (5/5)

❖ 연산자

Operators	Example	Result
in	x in s	True if an item of s is equal to x, else False
not in	x not in s	False if an item of s is equal to x, else True
+	s + t	the concatenation of s and t
*	s * t	equivalent to adding s to itself n times
[]	s[i]	ith item of s, origin 0
[:]	s[i:j]	slice of s from i to j
[: :]	s[i:j:k]	slice of s from i to j with step k
len()	len(s)	length of s
min()	min(s)	smallest item of s
max()	max(s)	largest item of s
.index()	s.index(x)	index of the first occurrence of x in s
.count()	s.count(x)	total number of occurrences of x in s

Mapping Types

❖ 매핑 객체인 dictionary는 해시 가능한 값을 임의의 객체에 매핑하여 키로 신속하게 값을 찾아내는 해시 테이블 구조

- dictionary는 { 'data': 1, 'data': 2} 혹은 {1 : 'data', 2 : 'data'}와 같이 curly brace “{ }” 안에 쉼표로 구분 된 ‘key : value’ 쌍의 목록을 배치하여 표현
- 빈 dictionary는 “{ }” 로 표현
- dict 자료형으로 표현

dictionary 인덱싱	dictionary명[인덱스값] #인덱스값: 0, 1, 2, …, 크기-1

```
1:  val1 = {“math”: 90, “computer”: 95}
2:  print(val1[“math”])
3:  scores = [(math”, 90), (computer”, 95)]
4:  val2 = dict(scores)
5:  print(val2[“math”])
```

제어문 (1/4)

- if, while 및 for 문이 제어 흐름을 변경하기 위해 사용됨
- The if statement

if statement	"if" expression ":" suite ("elif" expression ":" suite) ["else" ":" suite]

```
1:  x = 12
2:  if x % 3 == 0 :
3:  print(“3의 배수”)
4:  elif x % 5 == 0 :
5:  print(“5의 배수”)
6:  else :
7:  print(“3과 5의 배수가 아님”)
```

제어문 (2/4)

❖ The while statement

while statement	"while" expression ":" suite

- while 키워드 뒤에 위치한 expression이 참일 경우 콜론(:) 뒤에 있는 suite를 반복적으로 실행

```
1:  x = 1
2:  while x < 10 :
3:  print(“한자리 수”)
4:  x++ # x=x+1
```

제어문 (3/4)

❖ The for statement

- 주로 시퀀스의 요소 (예 : 문자열, 튜플 또는 목록)를 반복하는 데 사용

for statement	"for" target_list "in" expression_list ":" suite

```
1:  sum=0
2:  for x in range(1, 11) :
3:  if x%2 == 0:
4:  sum=sum+x # sum+=x
```

제어문 (4/4)

❖ The try statement

- 명령문 그룹에 대한 예외 처리 코드를 지정하는데 사용

try statement 사용형식 1	"try" ":" suite ("except" [expression [("as" | ",") identifier]] ":" suite)

```
1:  x=10
2:  try:
3:  x / 0
4:  except ZeroDivisionError as e:
5:  print(e)
```

try statement 사용형식 2	"try" ":" suite "finally" ":" suite

```
1:  f = open('text.txt', 'w')
2:  try:
3:  ....
4:  finally:
5:  f.close()
```

파이썬 함수의 구조 (1/2)

- def는 함수를 선언하기 위해 사용되는 파이썬의 키워드
- 함수이름은 사용자가 원하는 이름으로 임의로 설정함

함수 선언	def 함수이름(입력매개변수리스트): 수행할 코드#1 수행할 코드#2 ...

```
>>> def hello_world(name):
... print('hello world, ' + name)
...
>>> name = 'byoung-dai'
>>> hello_world(name)
hello world, byoung-dai
```

- 입력매개변수의 개수
  - 입력매개변수 앞에 '*' 를 이용하여 함수를 선언할 경우 해당 함수는 가변 크기의 입력매개변수를 가질 수 있음

가변크기의 입력매개변수를 가지는 함수 선언	def 함수이름(*입력매개변수): 수행할 코드#1 수행할 코드#2 ...

```
1:  def add(*numbers):
2:  sum = 0
3:  for number in numbers:
4:  sum = sum + number
5:  return sum
```

파이썬 함수의 구조 (2/2)

❖ 모듈

- 함수나 변수 등 파이썬 프로그램의 일부를 저장하고 있는 독립적인 파일

모듈 import 방법	import 모듈 이름

```
# my_module.py
def add(a, b):
return a+b
```

```
>>> sum = my_module.add(10, 20)
>>> print add
30
```

혹은

```
>>> from my_module import add
>>> add(10, 20)
30
```

Q&A

Machine Learning

PUSAN NATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Software & System Security Laboratory

머신 러닝 개요 (1/5)

❖ 머신러닝이란?

- 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야 – 아서 사무엘, 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 미첼, 1997
  - 어떤 작업 T: 문제 (예: 스팸 필터)
  - 경험 E: 훈련 데이터 (Training Set)
  - 성능 측정 P: 정확도 등
- 명시적인 규칙을 코딩하지 않고 기계가 데이터로부터 학습하여 어떤 작업을 더 잘하도록 만드는 거

❖ 왜 머신러닝을 사용하는가?

- 전통적인 접근 방법에서는 문제가 단순하지 않아 규칙이 점점 길고 복잡해지므로 유지 보수가 매우 힘든 경우
- 전통적인 접근 방법에서는 너무 복잡하거나 알려진 알고리즘이 없는 경우
- 대용량의 데이터를 분석하여 겉으로는 보이지 않던 패턴을 발견함
- (데이터가) 유동적인 환경

머신 러닝 개요 (2/5)

❖ 왜 머신러닝을 사용하는가?

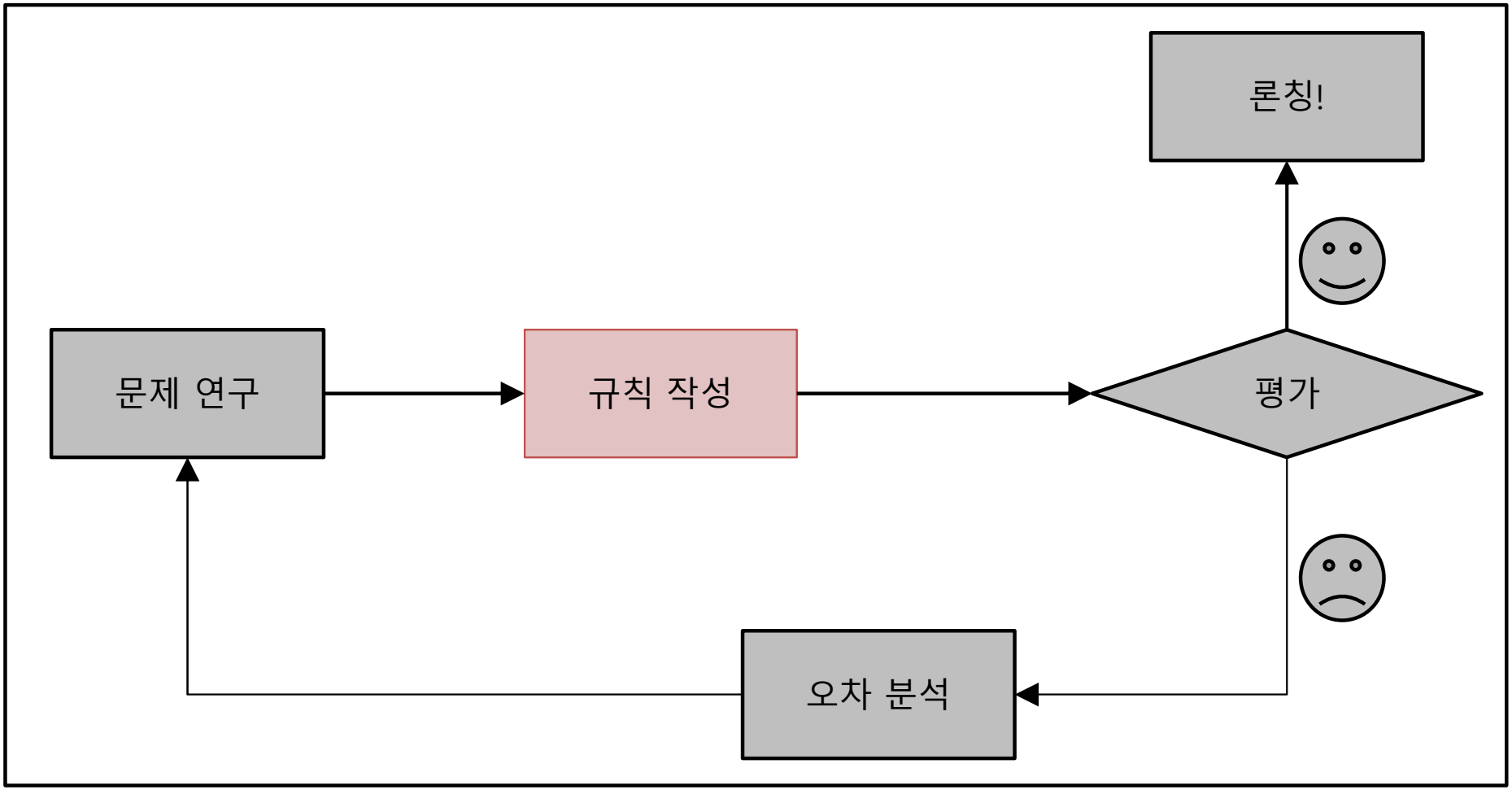


그림 1-1. 전통적인 접근 방법

머신 러닝 개요 (3/5)

❖ 왜 머신러닝을 사용하는가?

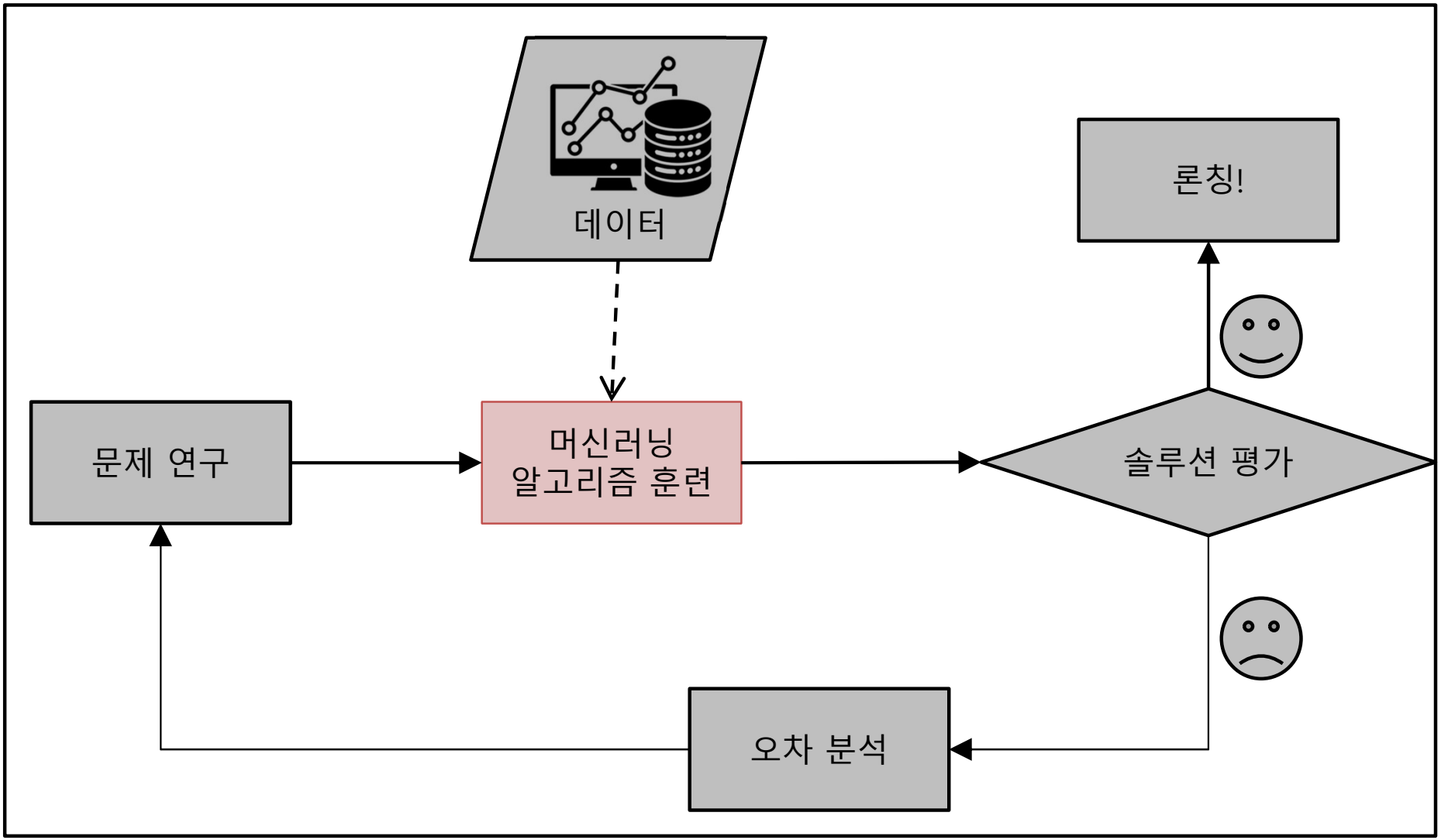


그림 1-2. 머신러닝 접근 방법

머신 러닝 개요 (4/5)

❖ 왜 머신러닝을 사용하는가?

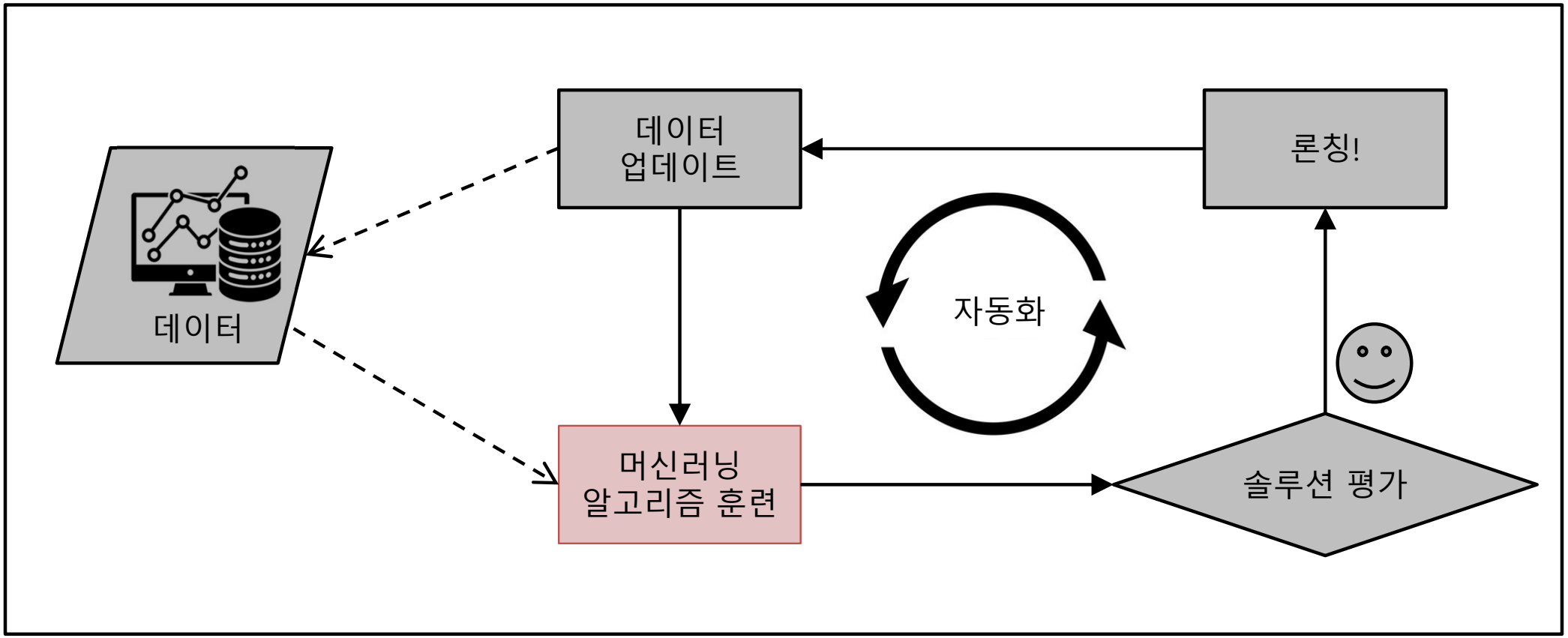


그림 1-3. 자동으로 변화에 적응함

머신 러닝 개요 (5/5)

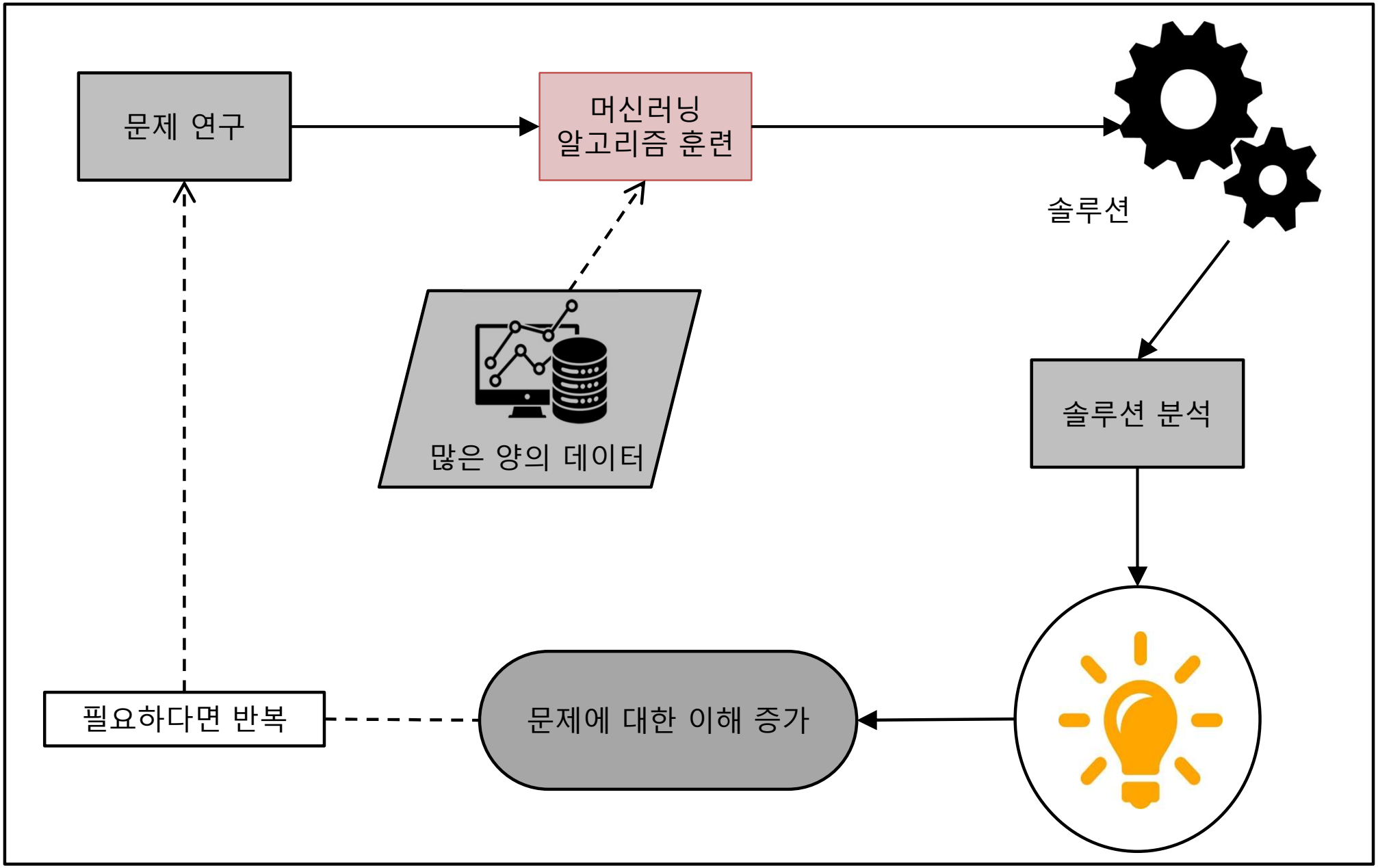


그림 1-4. 머신러닝을 통해 배웁니다.

머신러닝 시스템의 종류

❖ 머신러닝 시스템의 종류

- 지도, 비지도, 준지도, 강화 학습
  - 학습하는 동안의 감독 형태나 정보량에 따라 분류
  - k-nearest neighbors, linear regression, logistic regression, support vector machine (SVM), decision tree & random forests, neural networks
- 온라인 학습과 배치 학습
  - 입력데이터의 스트림으로부터 실시간으로 점진적인 학습을 하는지 아닌지
    - Clustering: k-means, hierarchical cluster analysis (HCA), expectation maximization
    - Visualization & dimensionality reduction: Principal component analysis (PCA), Kernel PCA, Locally-Linear Embedding (LLE), t-distributed stochastic neighbor embedding (t-SNE)
    - Association rule learning: Apriori, Eclat
- 사례 기반 학습과 모델 기반 학습
  - 일반화 방법에 따른 분류
    - 단순하게 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 아니면 훈련 데이터 셋에서 과학자들처럼 패턴을 발견하여 예측 모델을 만드는지

❖ 머신 러닝 동작 과정

- 데이터 분석 → 모델 선택 → 훈련 데이터를 이용한 모델 학습 → 새로운 데이터를 이용한 예측(추론(inference)) → 일반화 오류 최소화

지도, 비지도, 준지도, 강화 학습

❖ 지도 학습 (Supervised Learning)

- 알고리즘에 주입하는 훈련 데이터에 레이블(label 혹은 class)이라는 원하는 답이 포함됨
- 분류(Classification)와 회귀(Regression)

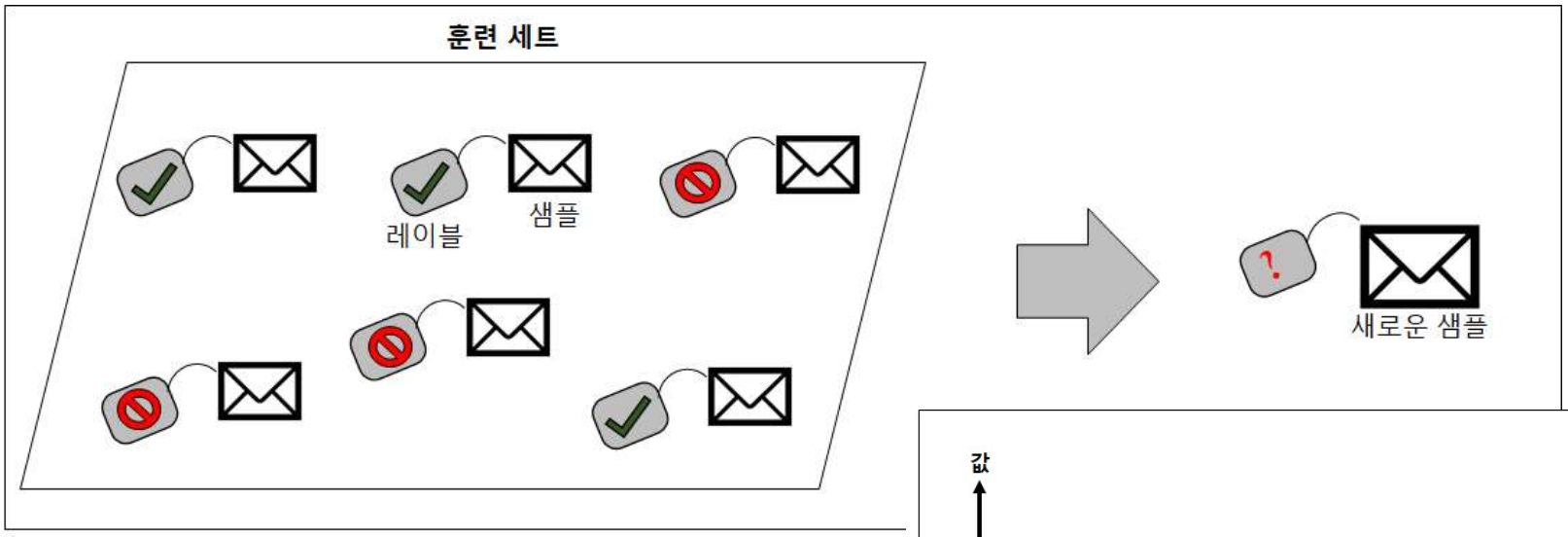


그림 1-5. 지도학습에서 레이블된 훈련 세트(예: 스팸 분류)

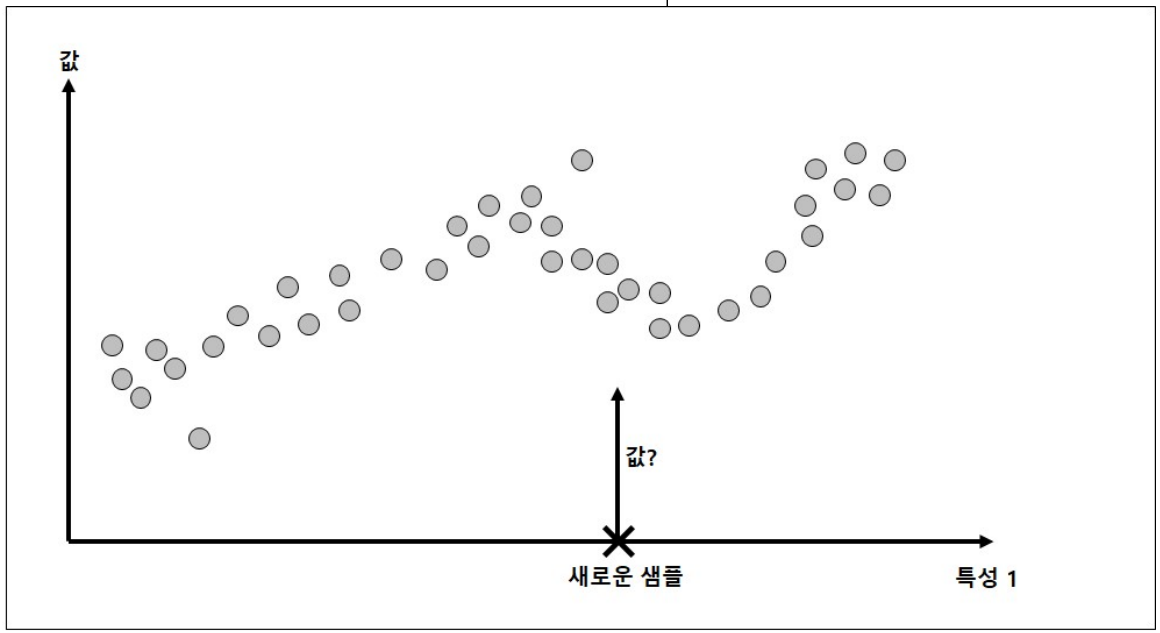


그림 1-6. 회귀

지도, 비지도, 준지도, 강화 학습

❖ 비지도 학습 (Unsupervised Learning)

- 훈련 데이터에 레이블이 없는 경우의 학습 방법
- 군집(clustering)

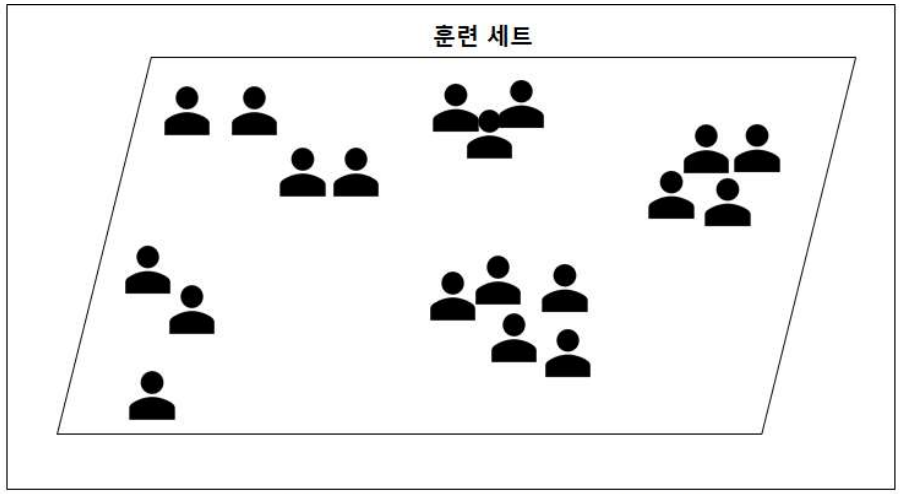


그림 1-7. 비지도 학습에서 레이블이 없는 훈련 세트 (예: 블로그 방문자)

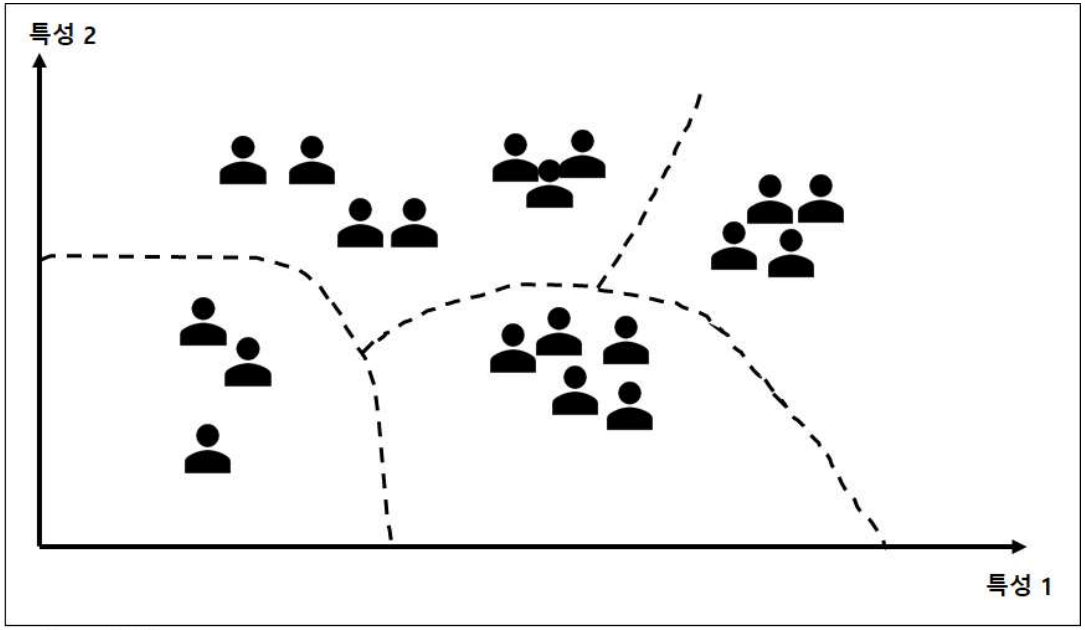


그림 1-8. 군집

지도, 비지도, 준지도, 강화 학습

❖ 비지도 학습 (Unsupervised Learning)

- 시각화(visualization)

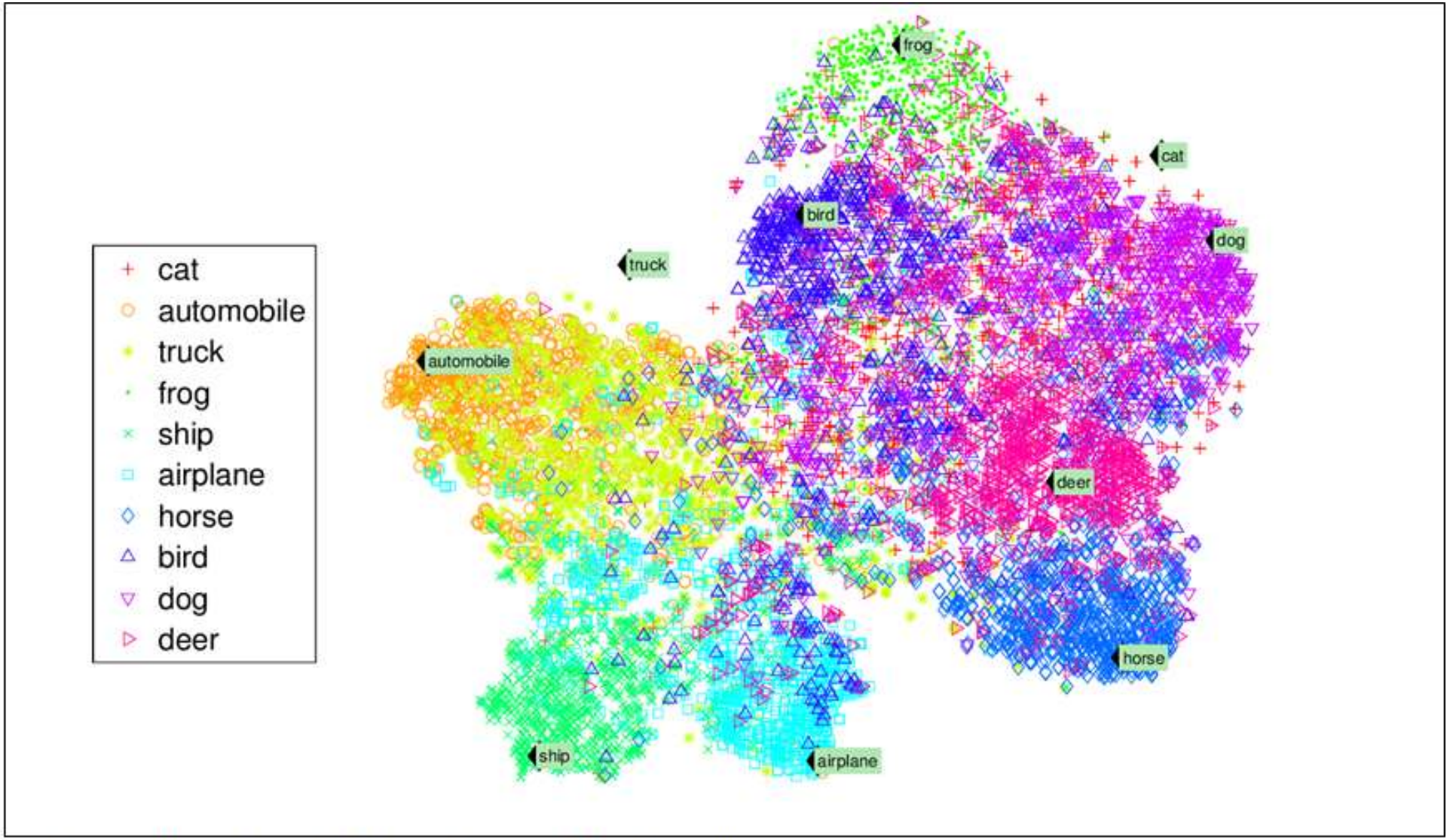


그림 1-9. 의미 있는 군집을 강조한 t-SNE 시각화의 예

지도, 비지도, 준지도, 강화 학습

❖ 비지도 학습 (Unsupervised Learning)

- 이상 탐지(Anomaly detection)

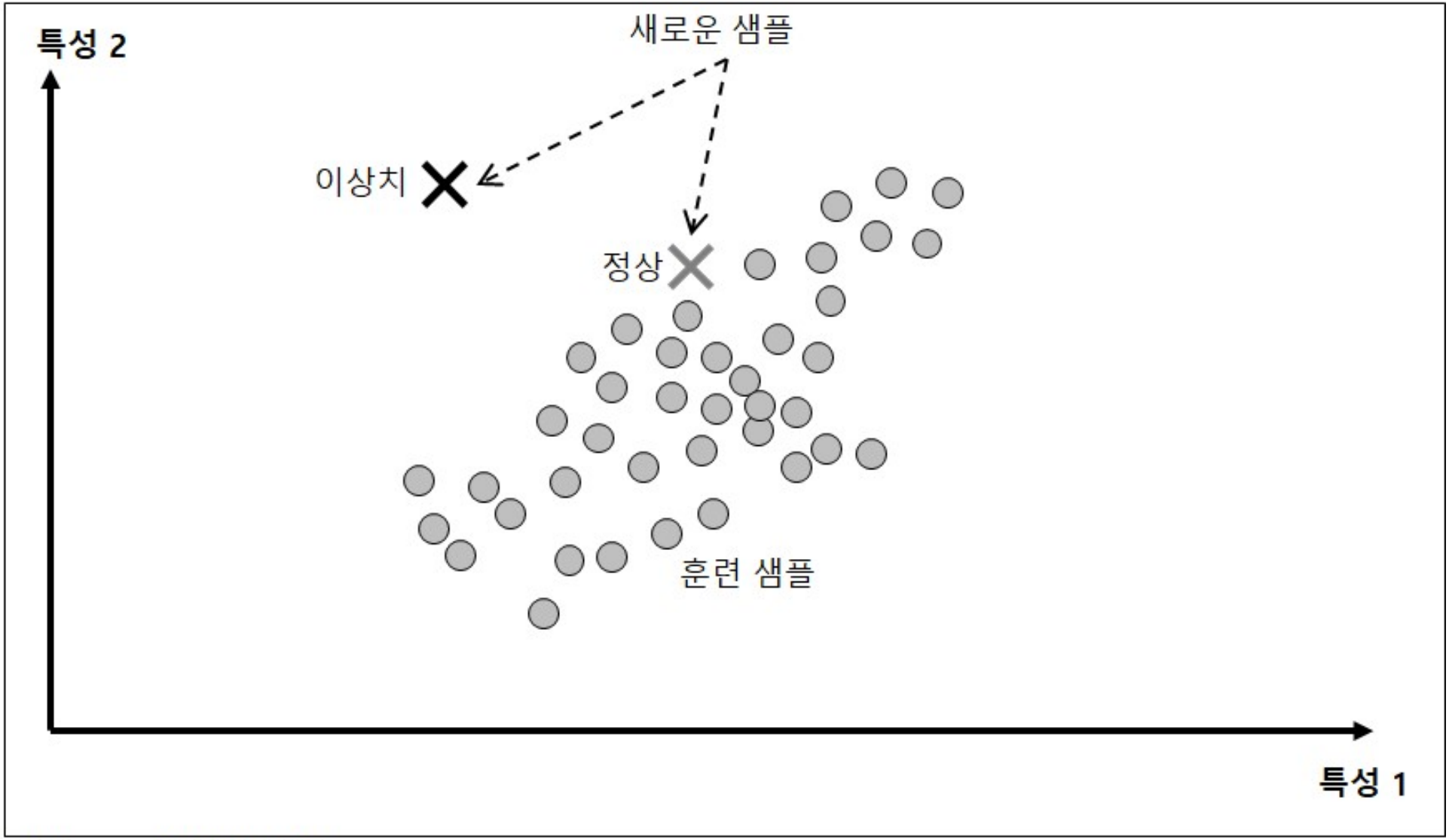


그림 1-10. 이상치 탐지

지도, 비지도, 준지도, 강화 학습

❖ 준지도 학습 (Semi-supervised Learning)

- 레이블이 없는 많은 데이터 + 레이블이 있는 일부 데이터
- 예: 구글 호스팅 서비스

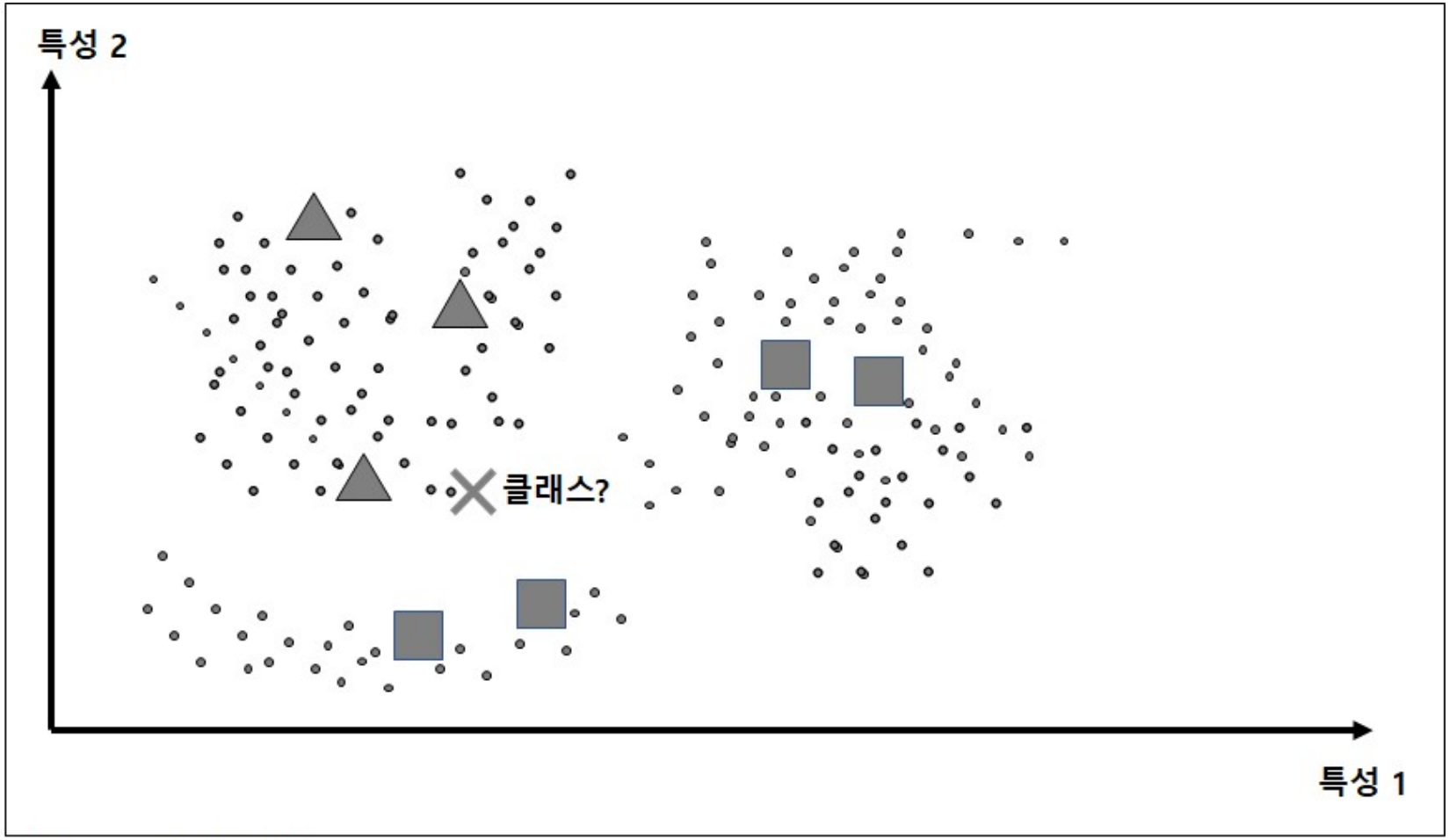


그림 1-11. 준지도 학습

지도, 비지도, 준지도, 강화 학습

❖ 강화 학습

- 학습하는 시스템인 agent가 환경(environment)을 관찰해서 행동(action)을 실행하고, 그 결과로 보상(또는 벌점)을 받음
- 가장 큰 보상을 얻는 주어진 상황에서 agent가 선택할 행동인 정책(policy)을 학습

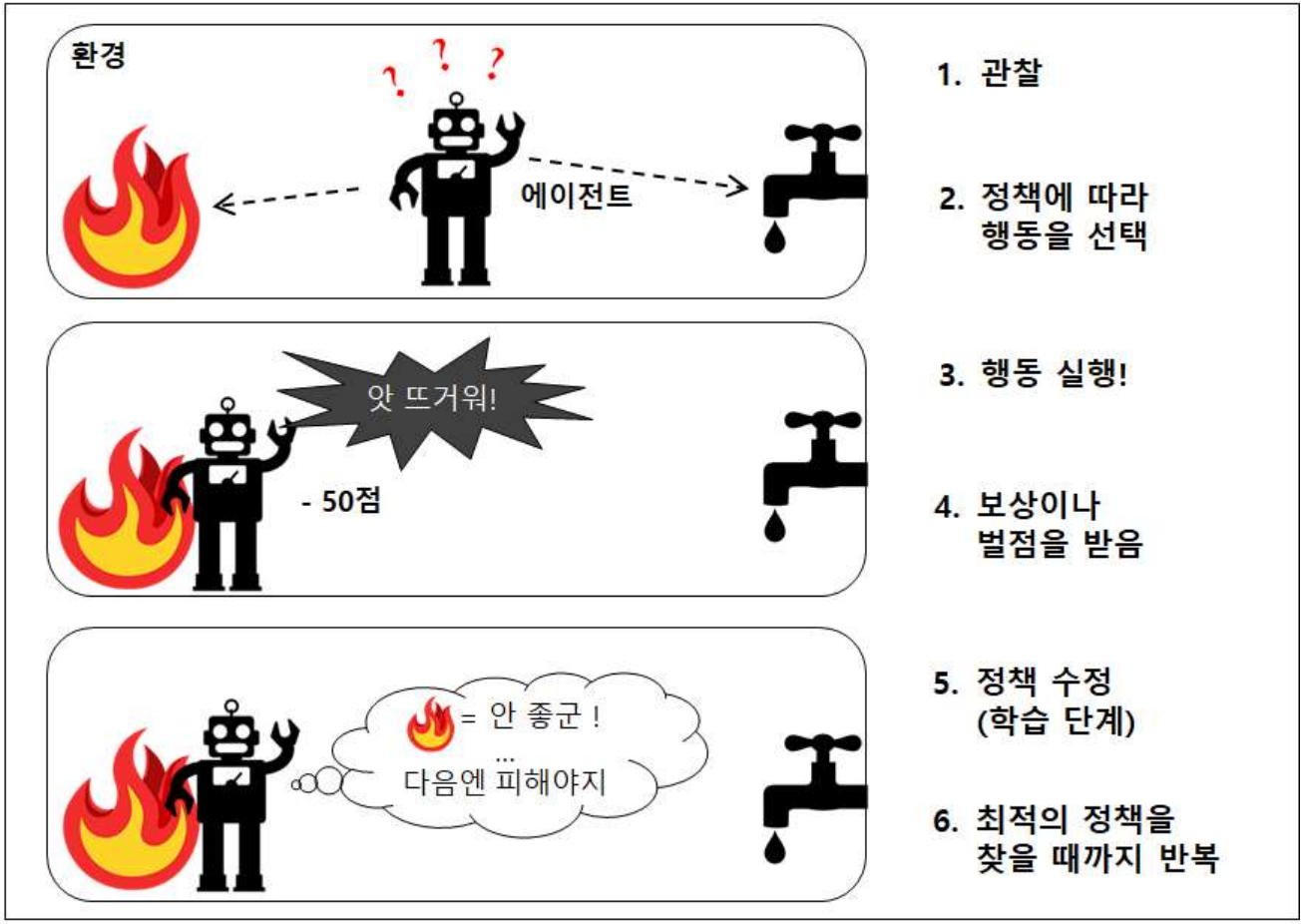


그림 1-12. 강화학습

배치 학습과 온라인 학습

❖ 배치 학습 (batch learning)

- 시스템이 점진적으로 학습할 수 없는 오프라인 학습
- 가용한 데이터를 모두 사용해 훈련시켜야 함

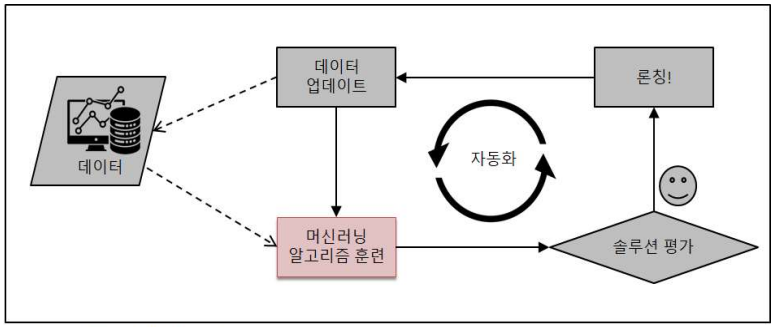


그림 1-3. 자동으로 변화에 적응함

❖ 온라인 학습 (online learning 혹은 incremental learning)

- 데이터를 순차적으로 한 개씩 또는 미니 배치 단위로 주입하여 시스템을 훈련시킴
- 주식 가격과 같이 연속적으로 데이터를 받고 빠른 변화에 스스로 적응해야 하는 경우 적합함
- 단점: 시스템에 나쁜 데이터가 주입될 때 시스템 성능이 점진적으로 감소함

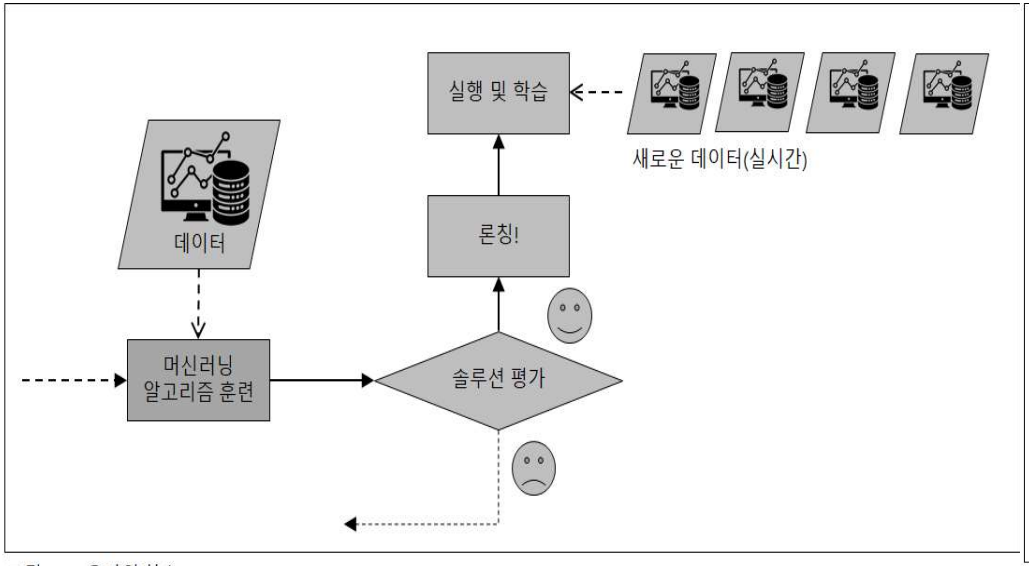


그림 1-13. 온라인 학습

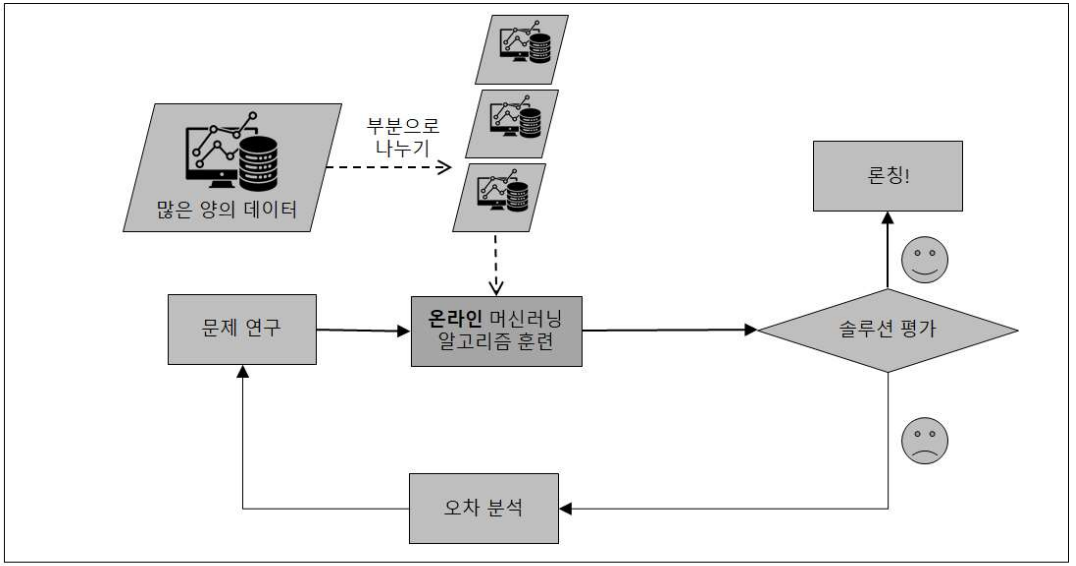


그림 1-14. 온라인 학습을 사용한 대량의 데이터 처리

사례 기반 학습과 모델 기반 학습

❖ 사례 기반 학습 (instance-based learning)

- 시스템에 사례를 기억함으로써 학습함
- 유사도(similarity)를 측정하여 새로운 데이터에 일반화 함

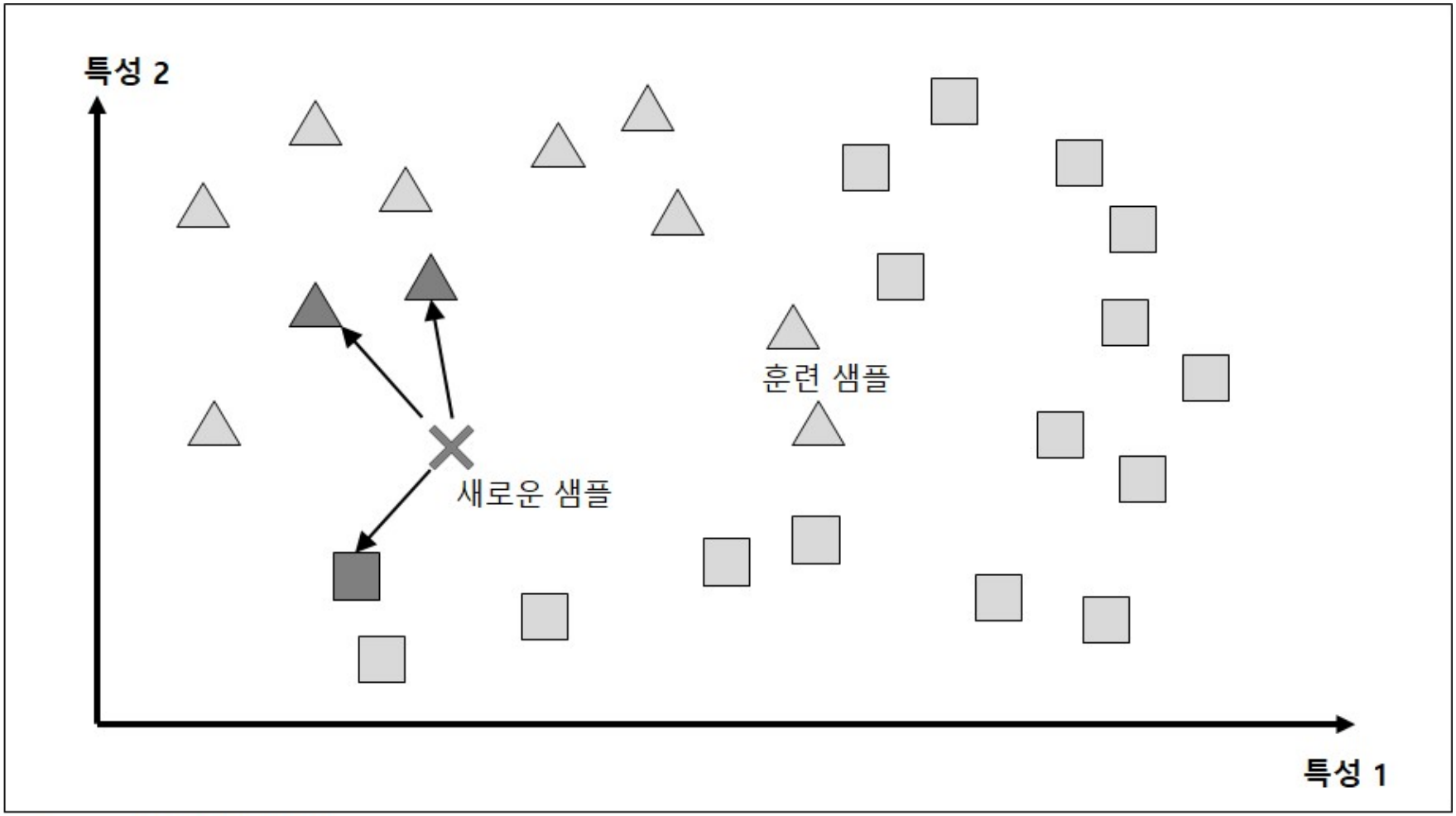


그림 1-15. 사례 기반 학습

사례 기반 학습과 모델 기반 학습

❖ 모델 기반 학습 (model-based learning)

- 샘플들의 모델을 만들어 예측에 사용함

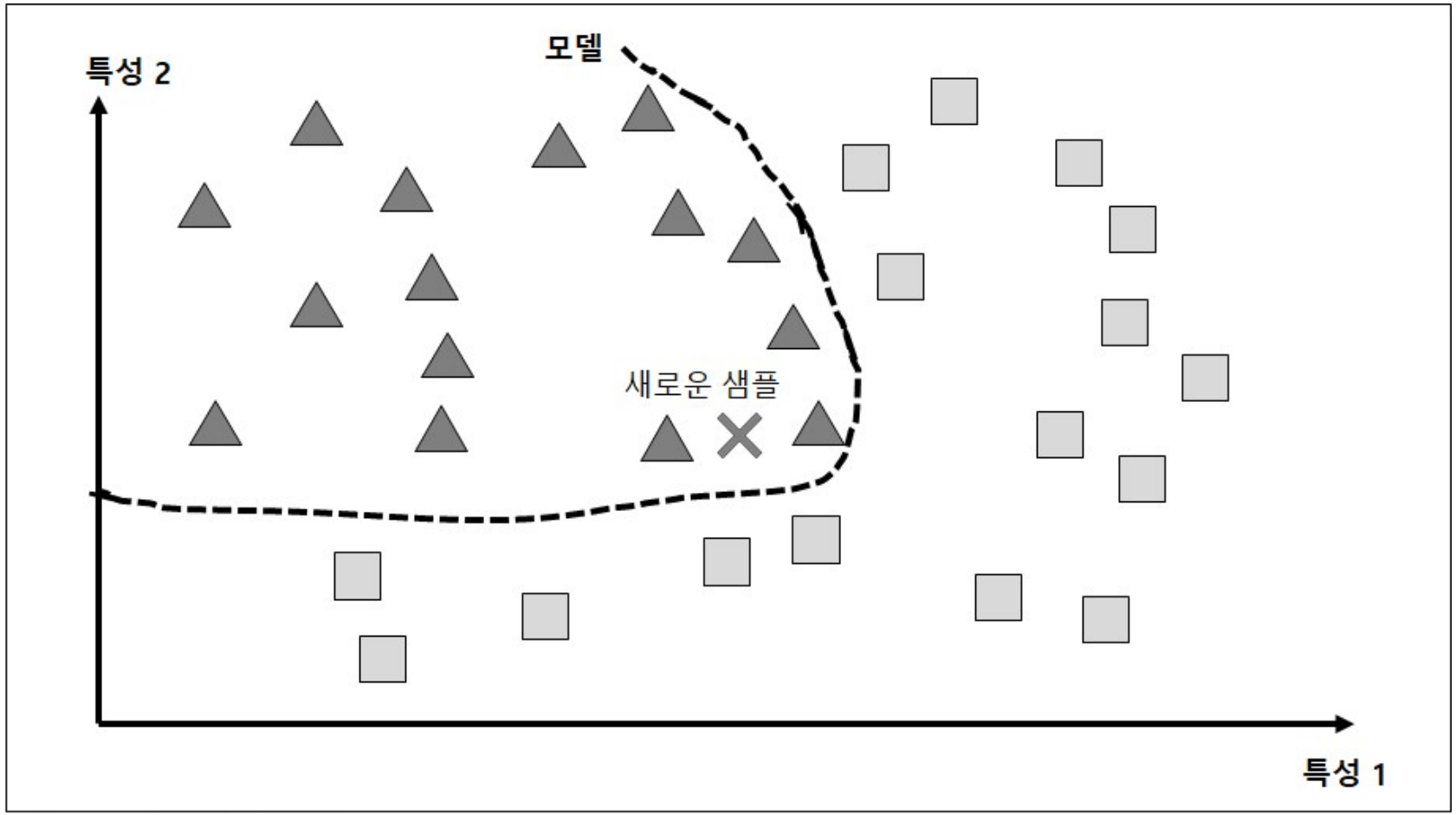


그림 1-16. 모델 기반 학습

주요 도전 과제

❖ 나쁜 데이터

- 충분하지 않은 양의 훈련 데이터
- 대표성 없는 훈련 데이터
- 낮은 품질의 데이터
- 관련 없는 특성
  - Feature selection과 feature extraction

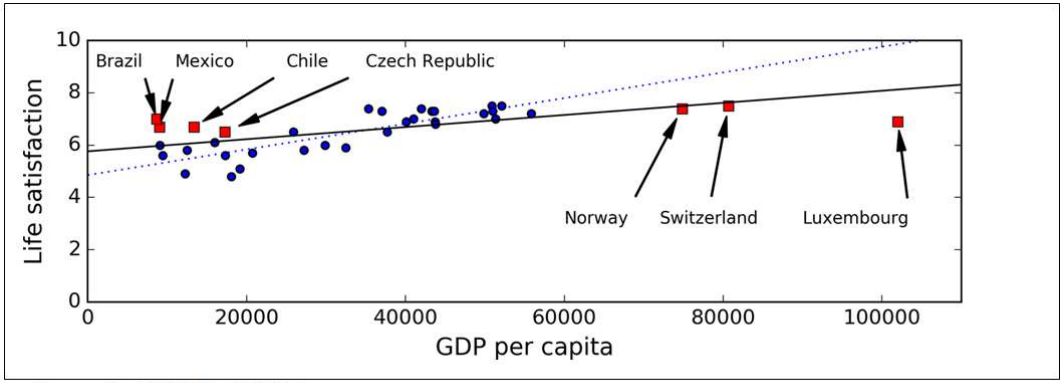


그림 1-21. 대표성이 더 큰 훈련 샘플

원인: sample noise 혹은 sampling bias

❖ 나쁜 알고리즘

- 훈련 데이터 과대적합 (overfitting)
  - 파라미터 수가 작은 모델을 사용
  - 훈련 데이터를 더 많이 모음
  - 훈련 데이터의 잡음을 제거
- 훈련 데이터 과소적합 (underfitting)
  - 파라미터가 더 많은 모델을 사용
  - 학습 알고리즘에 더 좋은 특성을 제공
  - 모델의 제약을 감소시킴

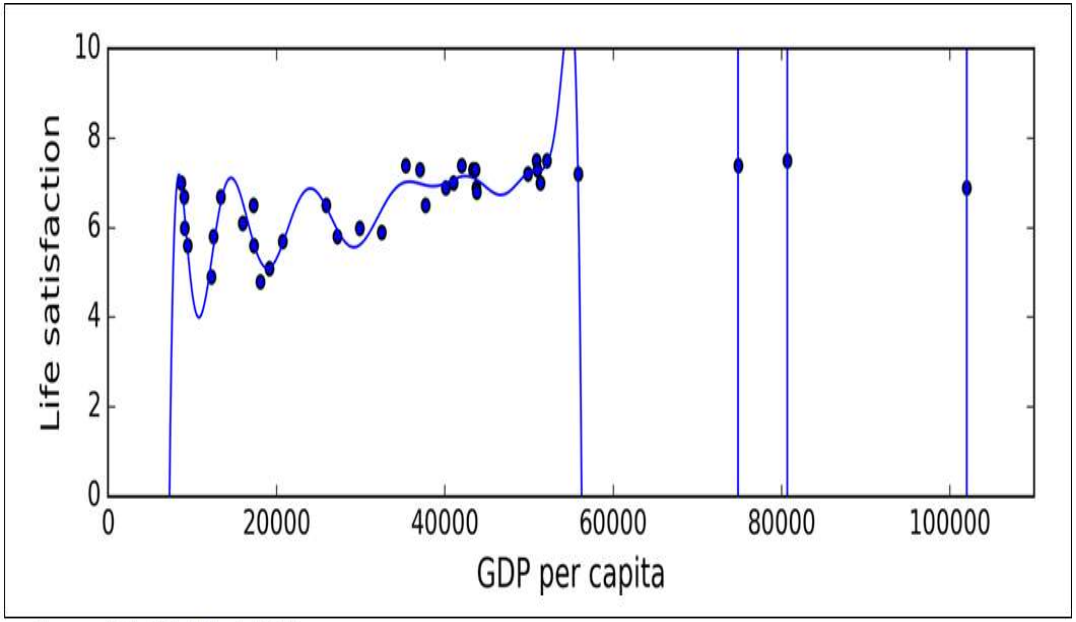


그림 1-22. 훈련 데이터에 과대적합

Q&A