Machine Learning

Kim Hye Kyung

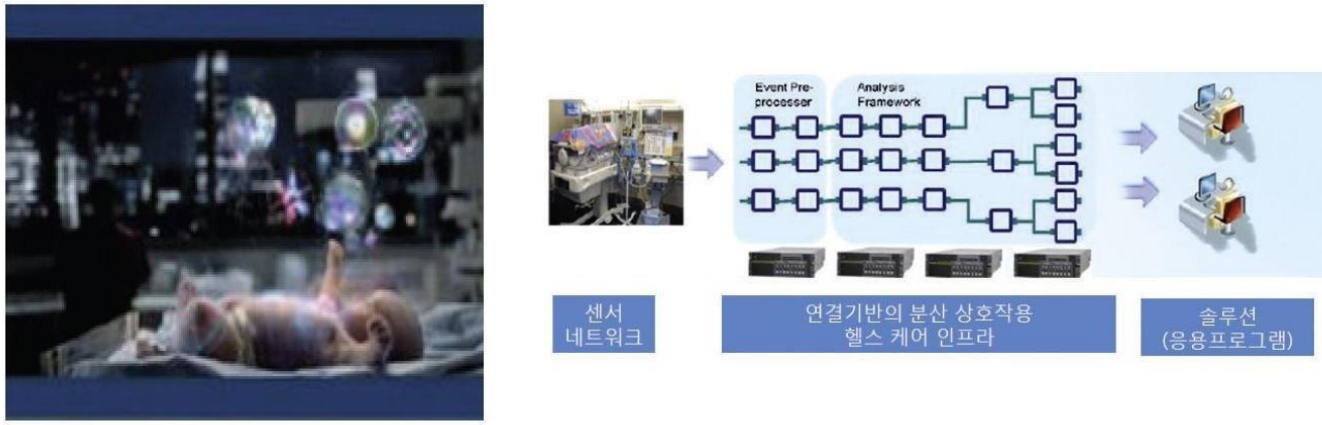
topickim@naver.com

데이터 사이언스

데이터 사이언스 활용 분야

- 미래의 예측, 진단 및 탐지
  - 신생아의 혈압, 체온, 심전도, 혈중산소포화도 등 미숙아 모니터링 장비에서 생성되는 환자당 일 9,000만 건 이상의 생리학 데이터스트림 실시간 분석
  - 의료진보다 24시간 전에 감염 사실을 밝혀냄으로써 조기 치료 가능

캐나다 온타리오 공과대병원의 신생아 모니터링



자료 : Anjul Bhambhri, Smarter Analytics for Big Data, IBM, 2011.6.7

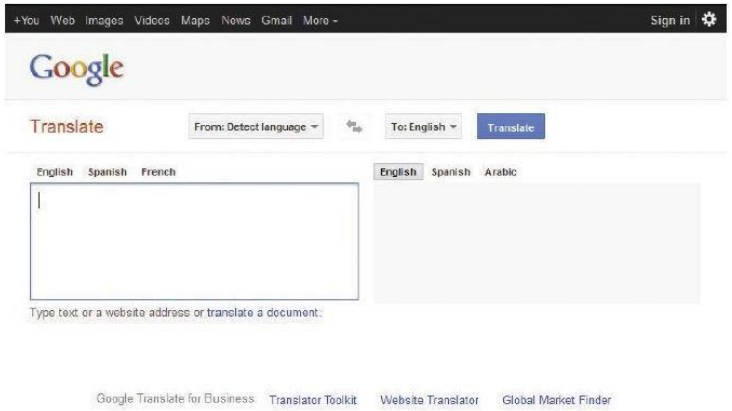
데이터 사이언스 활용 분야

- 일상생활에서의 의사결정 지원

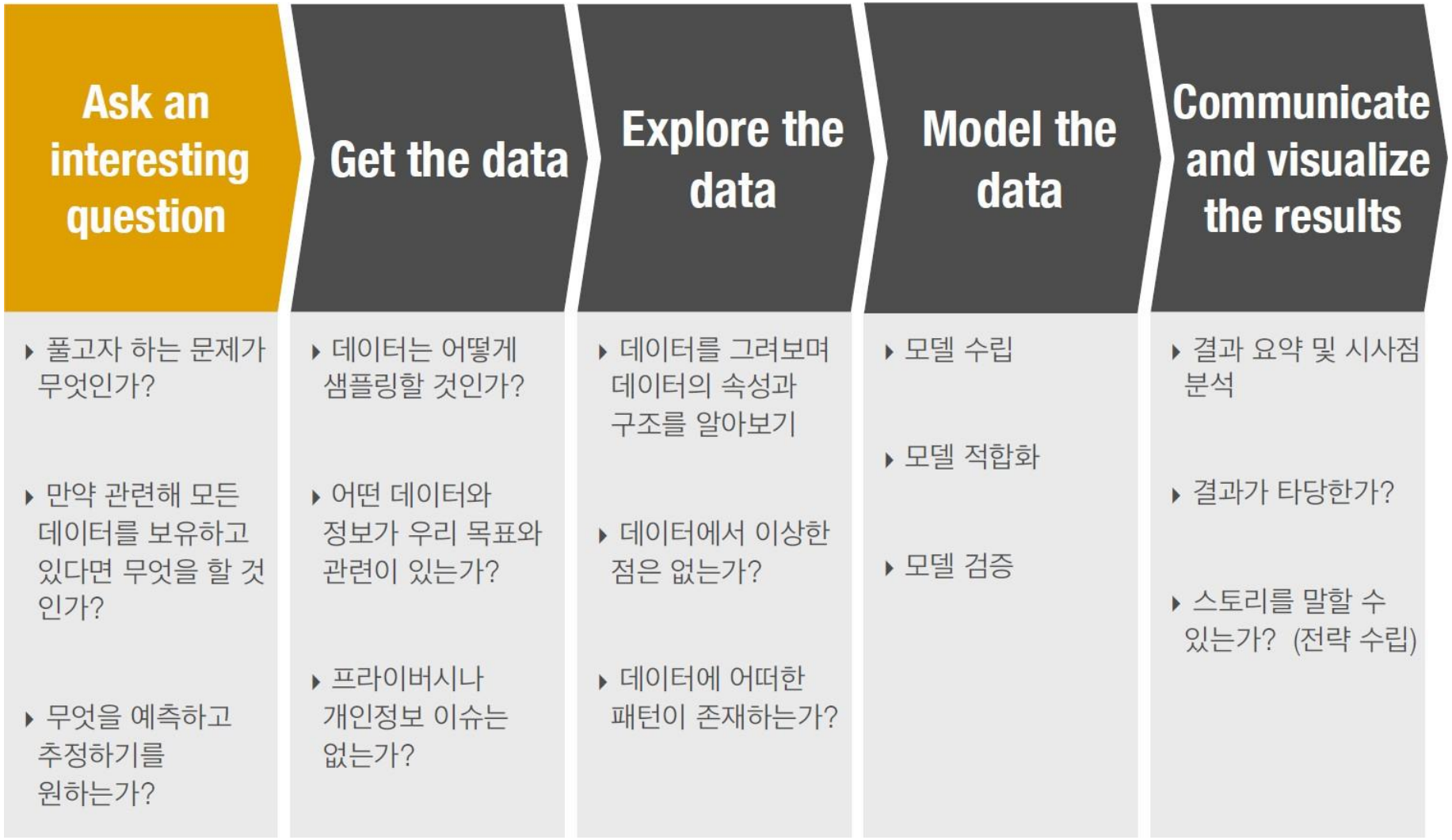
데이터 사이언스 활용 분야

- 삶의 질 향상
  - 언어의 문법적 구조를 컴퓨터가 이해할 수 있는 로직으로 변환하는 기존의 방법 탈피
  - 6개 국어로 번영되는 UN회의록과 23개 국어로 번역된 유럽의회 회의록을 번역 엔진에 입력한 뒤 통계적 추론 기법을 학습

구글 번역 홈페이지와 애플리케이션



데이터 사이언스 프로젝트 절차



데이터 사이언스 프로젝트 절차

1 단계: 흥미로운 질문을 하라

구조물 검사 문서로부터 검사 프로세스를 표준화할 수 있는가?

2단계: 분석에 적합한 데이터를 수집하라

Garbage in, garbage out

3단계: 성급한 모델링 이전에 충분히 데이터를 탐색하라

데이터 시각화 툴을 사용 권장

4단계: 모델 구축

질문의 속성, 데이터의 특징, 결과의 설명력 포함 유무 등을 고려하여
적합한 분석 알고리즘 선택

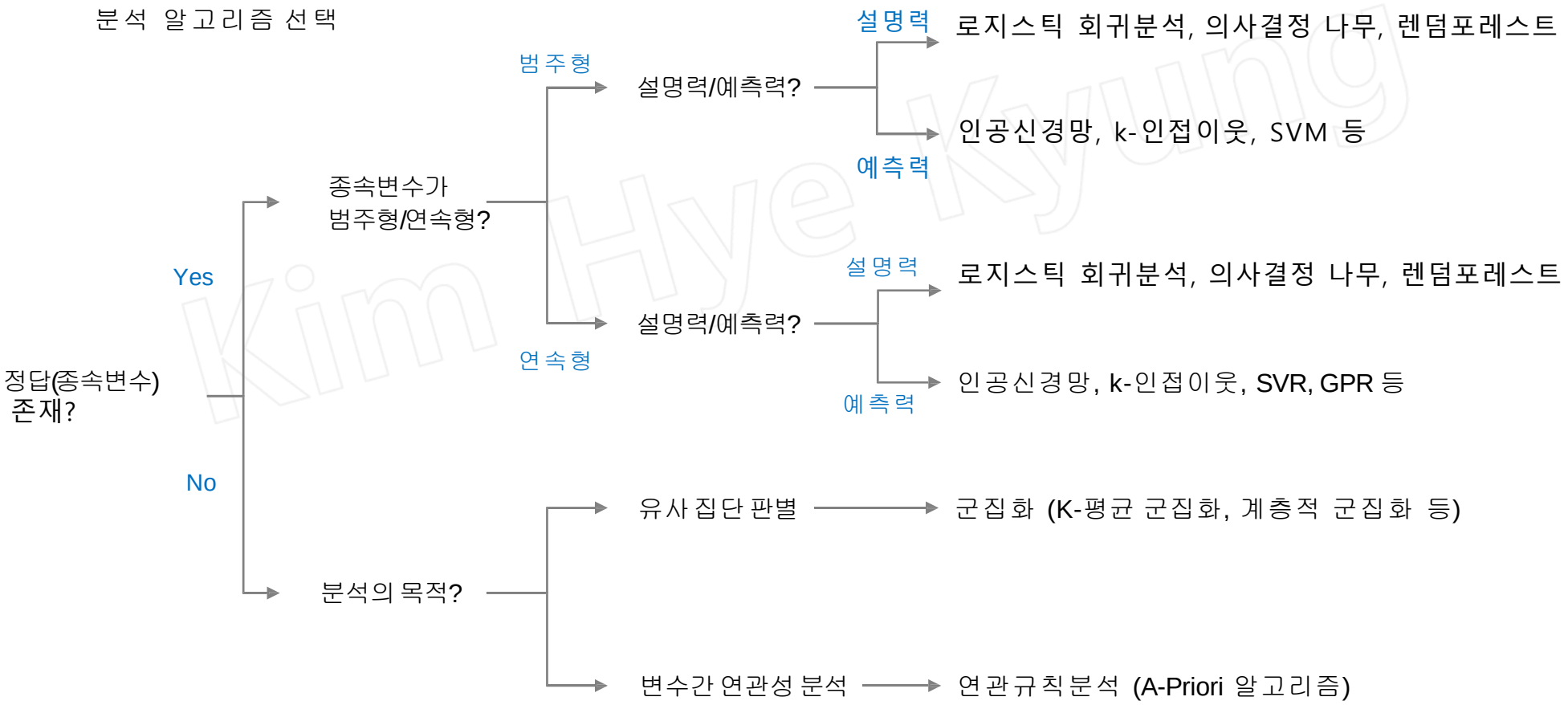
5단계: 결과 적용

구축된 모델의 시스템 탑재, 예측된 결과를 통한 테스트,
시간에 따른 성능 모니터링 및 업데이트 결정 등

데이터 사이언스 프로젝트 절차

4단계: 모델 구축

질문의 속성, 데이터의 특징, 결과의 설명력 포함 유무 등을 고려하여
적합한 분석 알고리즘 선택



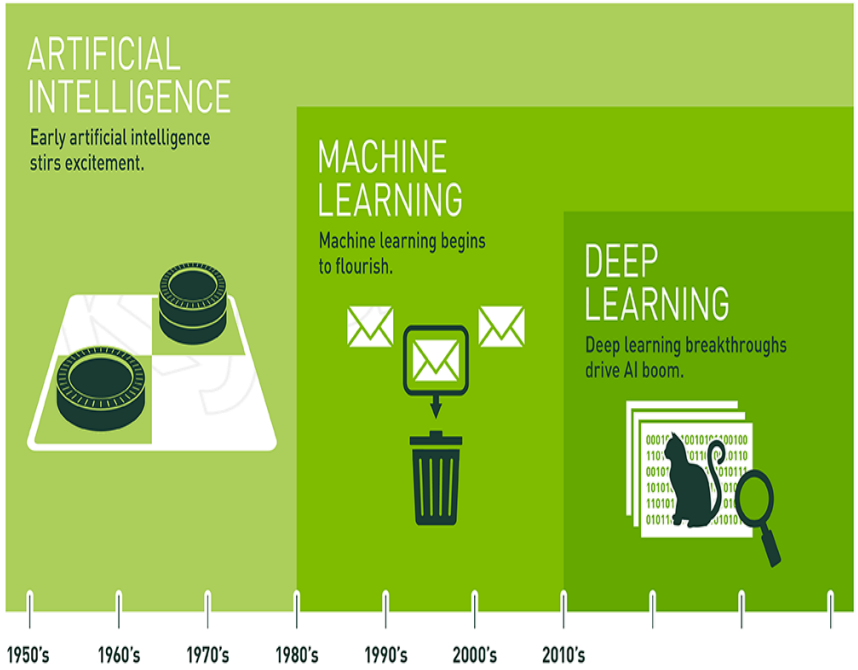
고려 사항

- 각 단계별 주요 과업 및 산출물

	목적 및 문제 정의	데이터 수집/검증/수정	데이터 전처리	모형 구축	평가 및 해석
주요 활동	• 데이터분석을통해 달성하고자하는목표 구체화	• 데이터원천확인 • 독립변수/종속변수 정의 • 변수별이상치/결측치 탐지및제거	• 불필요한 변수삭제 • 변수변환 • 비지도 방식의 변수 선택 및 추출 • 데이터분할	• 모델학습 • 최적파라미터선택	• 모델링결과평가 • 개선안수립
주 사용 기법			• 기초통계분석을 포함한EDA • 주성분분석	• 분류알고리즘 • 회귀알고리즘 • 군집화알고리즘 • 이상치탐지알고리즘	
산출물	• 문제기술서 • 모형의유형(분류/회귀 등)	• 행렬형태의모델링 기초데이터(행: 레코드,열:변수)	• 정제된모델링용 데이터	• 구축된모형 • 성능평가결과	• 모델결과평가표 • 개선아이디어리스트
고려 사항	• 현재보유데이터로 달성가능한목적인가?	• 최대한많은레코드와 변수를이단계에서 수집	• 사용모형에따른 데이터분할비율 • 문제에따른적절한 변수수	• 다양한알고리즘시도 • 최적파라미터선택 시 충분한영역탐색	• 모델의결과가 현장에서수용가능한 수준인가?

인공지능 vs. Machine Learning vs. 딥러닝

- 인공지능 vs. Machine Learning vs. 딥러닝
  - 인공지능
    - 특정 분야를 지칭하는 것이 아닌, 지능적 요소가 포함된 기술을 총칭
  - Machine Learning
    - 학습 전용 데이터에서 규칙성 등을 '학습' 하고 미지의 데이터를 판별할 수 있는 알고리즘
    - 스스로 규칙을 수정
  - 딥러닝
    - 심층 신경망을 이용한 Machine Learning 기법
    - Machine Learning의 기초 필수
    - 수술 환자의 사망률 예측
    - 손으로 쓴 글씨 판별
    - 꽃 품종 판별

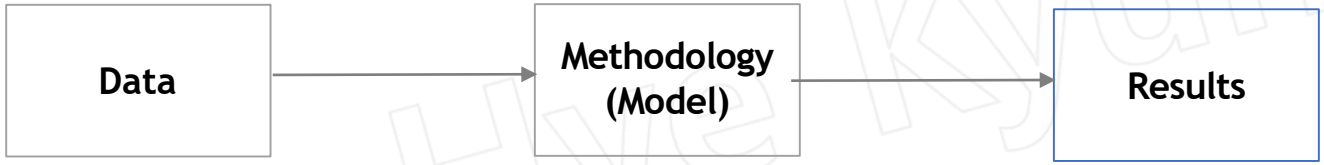


Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Machine Learning

- 특정한 과업(**T**ask)을 달성하기 위해 경험(**E**xperience)이 축적 될수록 과업 수행의 성능(**P**erformance)이 향상되는 컴퓨터 프로그램 또는 에이전트를 개발하는 것



- 얼굴 인식 및 자율 주행등에도 사용

Machine Learning

- Machine Learning 과 딥러닝

Deep Learning
적합한 데이터

Machine Learning
적합한 데이터

비정형 데이터

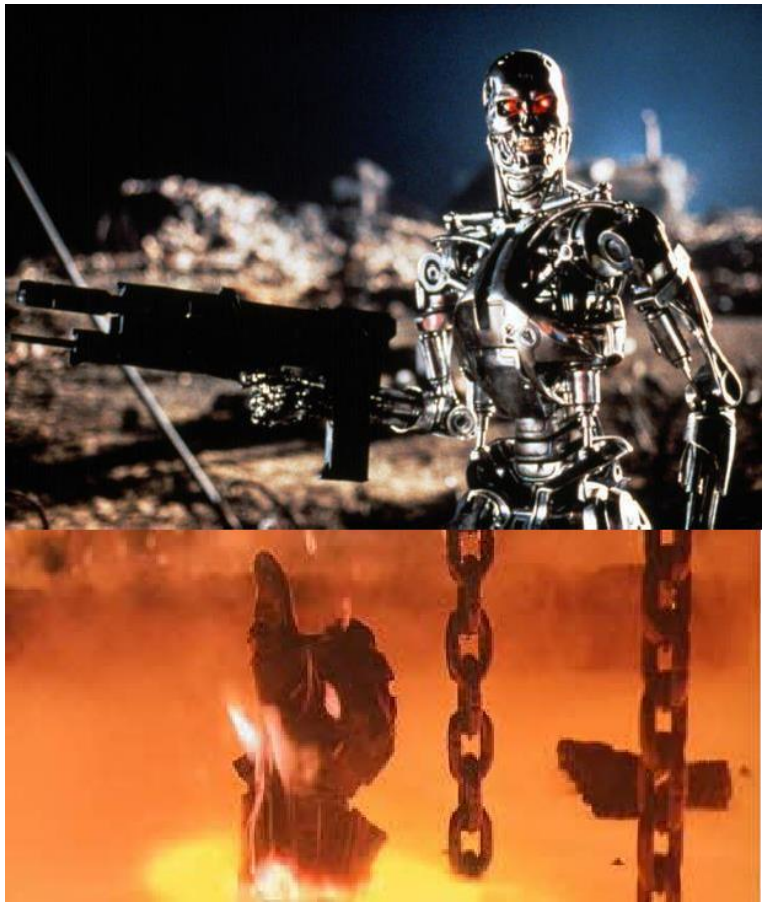
이미지, 영상, 음성, 텍스트, 번역 등

정형 데이터

데이터베이스, 레코드 파일, 엑셀, CSV 등

인공 지능 (Artificial Intelligence)

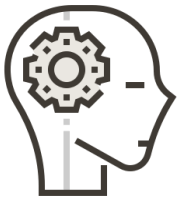
- 우리가 상상하는 인공지능



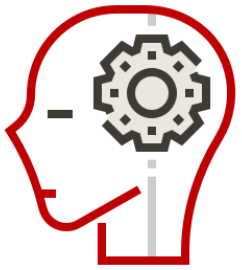
인공 지능 (Artificial Intelligence)

- 우리가 구현한 인공지능

인공 지능 (Artificial Intelligence)



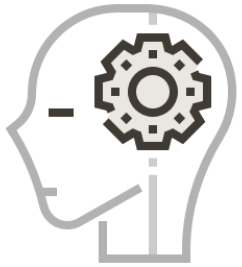
인공지능
(AI)



강 인공지능
(Strong AI)

사람과 구분이 안 될 정도로 강한 성능을 가진 인공지능

예 : 아이언맨의 자비스등
현 시점에 개발 가능? 그렇다면 언제쯤?



약 인공지능
(Weak AI)

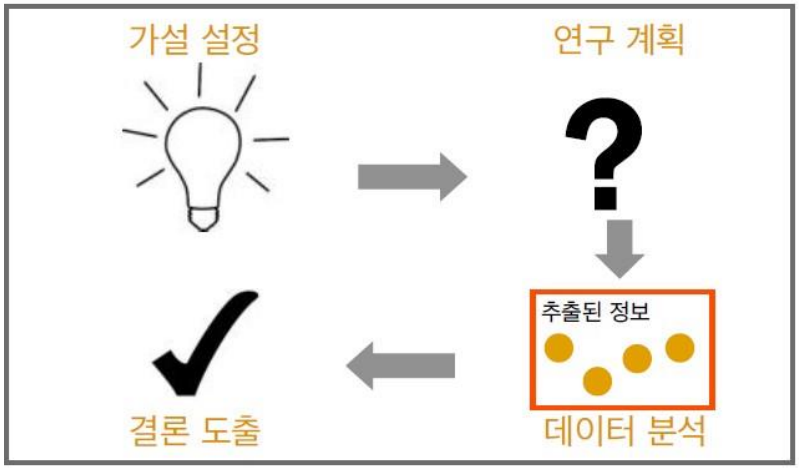
특정 영역에서 작업을 수행하는 인공지능

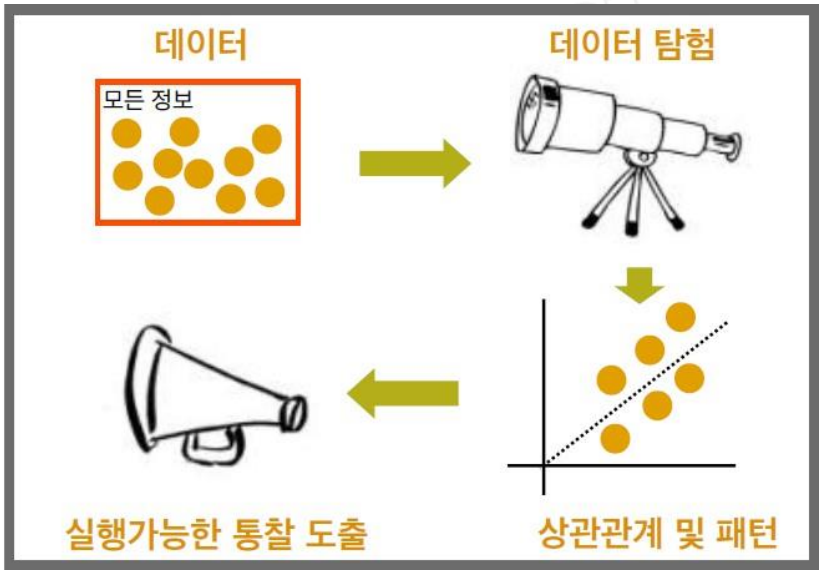
운전 보조, 질문 답변, 검색 수행
예 : 테슬라의 자율 주행 자동차,
애플의 아이폰에 포함된 음성 비서 시리등

데이터

데이터 기반의 의사결정

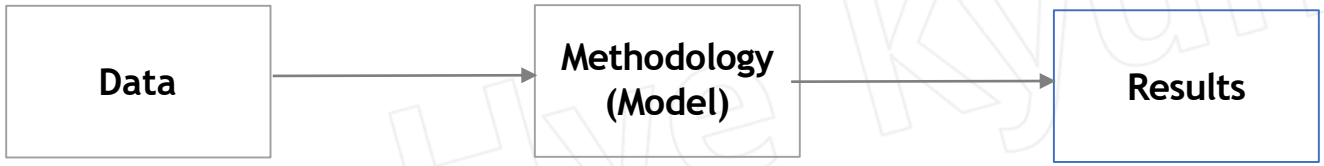
- 기존 학술 연구(좌) vs. 데이터 사이언스(우)





데이터 마이닝

- Definitions
  - 대량의 데이터로부터 의미 있는 규칙이나 패턴을 추출하는 일련의 활동



Ⅰ Machine Learning

Ⅱ Machine Learning 필요성 체감하기

Ⅲ Machine Learning Framework

Machine Learning 개발 환경

Machine Learning 개발을 위한 소프트웨어

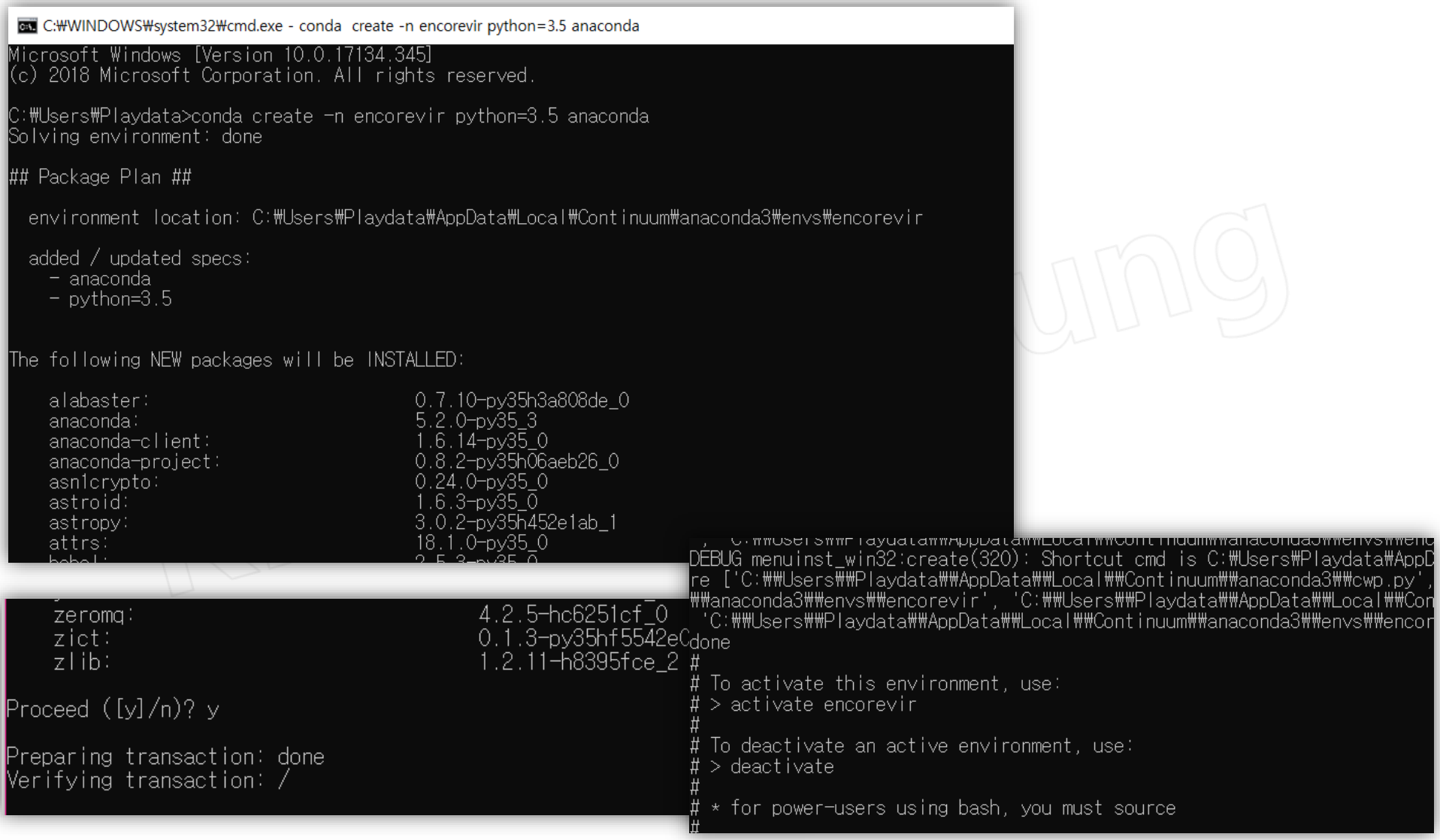
Anaconda

파이썬 기반의 Machine Learning에 필요한 package들을 일괄적으로 설치 할 수 있게 함

Microsoft Visual Studio Build Tool
2015이상 버전

윈도우 환경에서 서드파티 패키지를 설치할때 필요한 tool

가상 환경 구축


C:₩WINDOWS₩system32₩cmd.exe - conda create -n encorevir python=3.5 anaconda
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.
C:₩Users₩Playdata>conda create -n encorevir python=3.5 anaconda
Solving environment: done
Package Plan
environment location: C:₩Users₩Playdata₩AppData₩Local₩Continuum₩anaconda3₩envs₩encorevir
added / updated specs:
- anaconda
- python=3.5
The following NEW packages will be INSTALLED:
alabaster: 0.7.10-py35h3a808de_0
anaconda: 5.2.0-py35_3
anaconda-client: 1.6.14-py35_0
anaconda-project: 0.8.2-py35h06aeb26_0
asn1crypto: 0.24.0-py35_0
astroid: 1.6.3-py35_0
astropy: 3.0.2-py35h452e1ab_1
attrs: 18.1.0-py35_0
zeromq: 4.2.5-hc6251cf_0
zict: 0.1.3-py35hf5542eC
zlib: 1.2.11-h8395fce_2
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: /
DEBUG menuinst_win32:create(320): Shortcut cmd is C:₩Users₩Playdata₩AppD
re ['C:₩₩Users₩₩Playdata₩₩AppData₩₩Local₩₩Continuum₩₩anaconda3₩₩cwp.py',
₩₩anaconda3₩₩envs₩₩encorevir', 'C:₩₩Users₩₩Playdata₩₩AppData₩₩Local₩₩Con
'C:₩₩Users₩₩Playdata₩₩AppData₩₩Local₩₩Continuum₩₩anaconda3₩₩envs₩₩encor
done
#
To activate this environment, use:
> activate encorevir
#
To deactivate an active environment, use:
> deactivate
#
* for power-users using bash, you must source
#

가상 환경 구축

- 1단계 : 가상 환경 설치
  - (base) >conda create –n encore python=3.6 anaconda
- 2단계 : 가상 환경 활성화
  - (base) >conda activate encore
  - (encore) >python --version
- 3단계 : python package install
  - (encore) >pip install scikit-learn scipy scikit-image mglearn xlrd
- 4단계 : 가상 환경 비활성화
  - encore> conda deactivate

참고

- 개발 환경 셋팅
  - win10에서 발생된 문제
  - import sklearn 인식 불가
  - 해결책
    - 삭제
      - 문제 발생
      - 삭제시 많을 모듈 삭제
      - 아나콘다 콘솔창도 안 보임
    - 재 설치
    - conda 콘솔창에서 jupyter note 실행

```
C:₩Users₩Kimhyekyung>conda install scikit-learn
Collecting package metadata: done
Solving environment: |
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

  - defaults/win-64::anaconda==2018.12=py37_0
done

## Package Plan ##

  environment location: C:₩Users₩Kimhyekyung₩Anaconda3

  added / updated specs:
    - scikit-learn


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    conda-4.7.5                |           py37_0         3.0 MB
    conda-package-handling-1.3.10|           py37_0         280 KB
    joblib-0.13.2              |           py37_0         365 KB
    mkl-2019.4                 |             245       157.5 MB
    mkl-service-2.0.2          |   py37he774522_0          63 KB
    scikit-learn-0.21.2        |   py37h6288b17_0         5.9 MB
    ------------------------------------------------------------
                                           Total:       167.2 MB

The following NEW packages will be INSTALLED:

  conda-package-han~ pkgs/main/win-64::conda-package-handling-1.3.10-py37_0
  joblib             pkgs/main/win-64::joblib-0.13.2-py37_0

The following packages will be UPDATED:

  conda                                    4.6.14-py37_0 --> 4.7.5-py37_0
  mkl                                    2019.1-144 --> 2019.4-245
  mkl-service                  1.1.2-py37hb782905_5 --> 2.0.2-py37he774522_0
  scikit-learn                0.20.1-py37h343c172_0 --> 0.21.2-py37h6288b17_0


Proceed ([y]/n)? y


Downloading and Extracting Packages
mkl-service-2.0.2    | 63 KB     | ######################################## | 100%
joblib-0.13.2        | 365 KB    | ######################################## | 100%
mkl-2019.4           | 157.5 MB  | ######################################## | 100%
scikit-learn-0.21.2  | 5.9 MB    | ######################################## | 100%
conda-4.7.5          | 3.0 MB    | ######################################## | 100%
conda-package-handli | 280 KB    | ######################################## | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Machine Learning 개요

Machine Learning 이란?

애플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하고

결과를 예측하는 알고리즘 기법을 통칭

Machine Learning이란

- 인간의 뇌가 자연스럽게 수행하는 "학습"이라는 능력을 컴퓨터로 구현하는 방법
- 인공지능 연구 과제 중 하나로 수많은 데이터를 학습시켜 그에 맞는 패턴을 찾아내는 것
- 데이터로 부터 학습하도록 컴퓨터를 프로그래밍 하는 과학(예술)

규모가 있는 sample 데이터를 자원으로 분석

데이터에서 일정한 규칙을 찾아냄

찾아낸 규칙을 기반으로 다른 데이터를 분류하거나 미래를 예측 하는 것

문자 인식, 음성 인식, 바둑 또는 장기 등의 게임 전략, 의료 진단, 로봇 개발등에 사용

데이터로부터 학습하도록 컴퓨터가 프로그래밍하는 과학(예술)

Machine Learning

전통적인 프로그램

규칙을 사람이 수정

규칙 정하기

프로그래밍

프로그램

습도가 70%면
그 다음 날은 비,

습도가 30%면
그 다음 날은
아주 맑음 ...

습도가 75%이므로
내일은 비가 옵니다

Machine Learning

Machine Learning 프로그램

프로그래밍

Machine Learning이
오늘의 습도를 통해
다음 날의 날씨를 스스로
판단할 수 있도록
프로그래밍 하자!

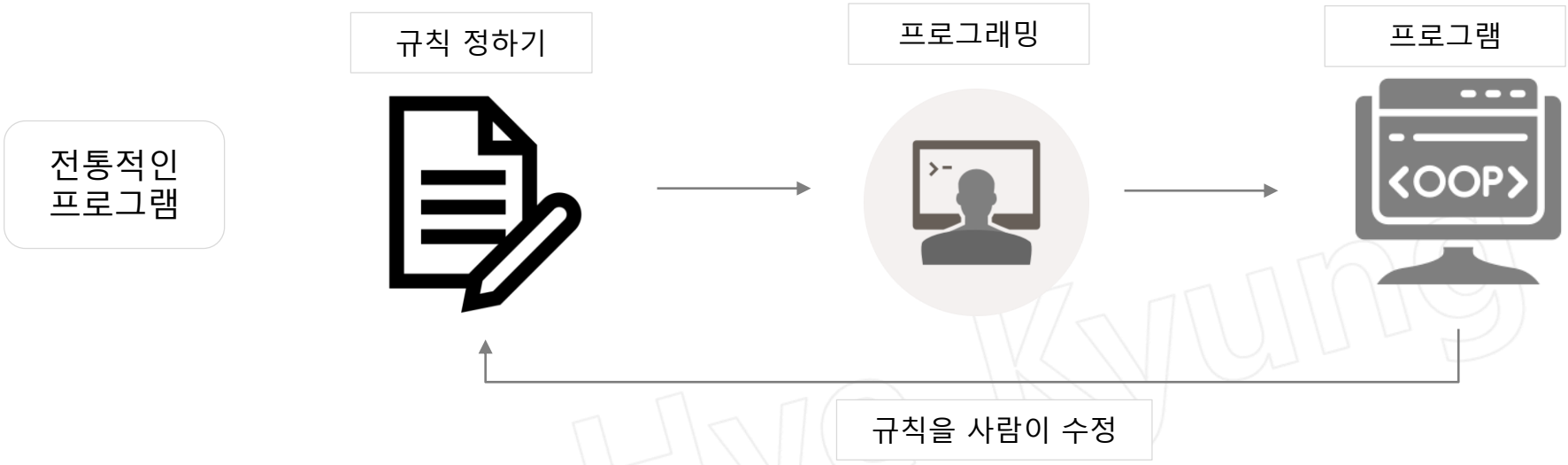
Machine Learning

규칙을 스스로 수정

지금까지의 비 온 날의
데이터를 종합해서 분석해
보니
오늘의 습도가 55%면 다음
날에 비가 왔습니다.

새로운 데이터가 유입되면
규칙을 다시 조정합니다

Machine Learning



분류

지도 학습

입력과 타깃으로 모델을
훈련하는 학습

비지도 학습

입력은 있으나 타깃이 없는
훈련 데이터를 사용하는 학습

날씨 예측을 통한 훈련 데이터 및 모델 이해하기

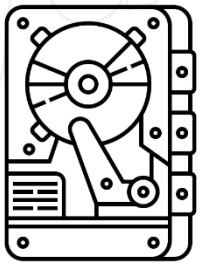
훈련 데이터

습도	비가 왔는지?
0.542	o
0.675	o
0.375	x
...	...

입력 타깃

훈련

모델



습도가 0.6 이상이면
다음날에 비가 왔네

Machine Learning

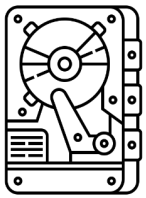
훈련 데이터

습도	비가 왔는지?
0.542	o
0.675	o
0.375	x
...	...

입력 타깃

훈련

모델



습도가 0.5이상이면
다음날에 비가 왔네

훈련 데이터 : 모델을 훈련시키기 위해 사용하는 데이터

입력 : 모델이 풀어야 할 일종의 문제와 같은 개념의 데이터

타깃 : 모델이 맞춰야 할 정답과 같은 개념의 데이터

문제에 대한 답을 주는 방식으로 모델을 훈련

모델 : 학습을 통해 만들어진 프로그램
모델은 새로운 입력에 대한 예측을 만듦

훈련 데이터로 학습된 Machine Learning 알고리즘

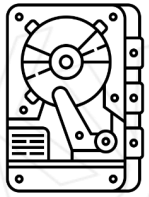
Machine Learning

새로운 입력 데이터

오늘의 습도가 0.87

훈련

모델



오늘은 비가 올거 같네

Machine Learning

지도 학습

측정된 데이터의 특징(feature)과 관련된 레이블(label) 사이의 관계를 모델링

명확한 정답이 주어진 데이터를 먼저 학습한 뒤 미지의 정답을 예측하는 방식

비지도 학습

레이블을 참조하지 않고 데이터 세트의 특징을 모델링 하는 것

'데이터 세트가 스스로 말하게 하는 것' 이라고 설명

Machine Learning

- Machine Learning은 스스로 규칙을 찾음
- 규칙이란?

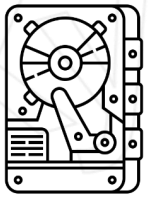
훈련 데이터

습도	비가 왔는지?
0.542	o
0.675	o
0.375	x
...	...

입력 타깃

훈련

모델



습도가 0.5이상이면
다음날에 비가 왔네

규칙

- 규칙을 표현하는 Machine Learning 알고리즘
  - 훈련 데이터와 규칙의 관계를 식으로 표현

$1.5 \times x + 0.1 = y$	y가 1 이상이면 다음날 비가 온다고 예측

- 규칙을 표현하는 Machine Learning 알고리즘
  - 훈련 데이터와 규칙의 관계를 식으로 표현

$1.5 \times x + 0.1 = y$	y가 1 이상이면 다음날 비가 온다고 예측

기호	표현	설명
x	입력	입력 데이터
y	타깃	타깃 데이터
1.5	가중치	입력 데이터에 곱하는 수 의미
0.1	절편	더하는 수

모델 파리미터(model parameter) : 가중치와 절편을 합친 의미

Machine Learning

- 수많은 데이터를 학습시켜 패턴을 찾아내는 것
- 패턴을 찾으면 그러한 패턴을 기반으로 데이터를 분류하거나 미래를 예측

- 학습 목적에 따른 구분

분류 (Classification)

회귀 (Regression)

군집 (Clustering)

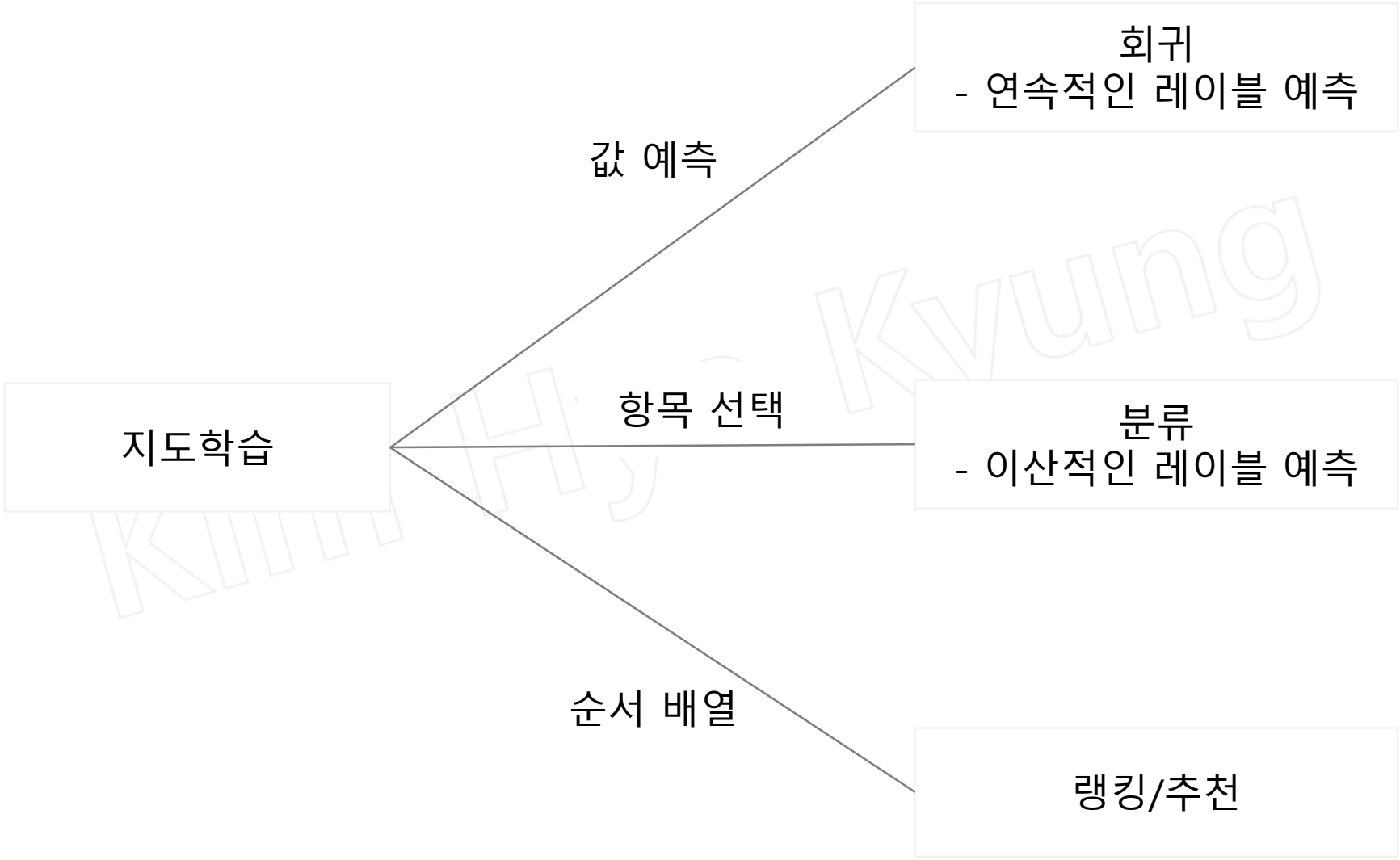
Machine Learning - 학습 목적에 따른 구분

분류 (Classification)	예) 손글씨 인식, 스팸메일 분류, 증권 사기, 자동차의 길이, 너비, 높이, 바퀴 크기, 엔진 마력 등의 특징을 보고 경차, 준중형차, 대형차 중 한가지로 분류 입력데이터들을 주어진 항목들도 구분 가령 어떤 문서가 도서관 어떤 분류에 해당하는지 선택하는 경우 이산적인 레이블 예측하기
회귀 (Regression)	과거 데이터를 기반으로 미래를 예측 연속형 변수를 예측하는 방법론 예) 판매 예측, 주가 변동 예측 불완전한 데이터의 값을 알아내기 위한 문제 예) 어제의 온도와 구름의 양으로 내일의 날씨가 좋을지 안 좋을지 예측하는 분류시스템 구현시 날씨가 좋을 경우 1, 날씨가 나쁠 경우 0을 두어 회귀를 이용하여 날씨가 좋을 확률을 0부터 1사이의 값으로 구하고 이 값이 0.8이상이면 좋음, 아니면 나쁨으로 분류 가능
군집 (Clustering)	유사한 개체들의 집단을 판별하는 방법론 예) 마케팅을 위해 고객을 군집화. 사용자의 취향을 그룹으로 묶어 사용자 취향에 맞는 광고 제공

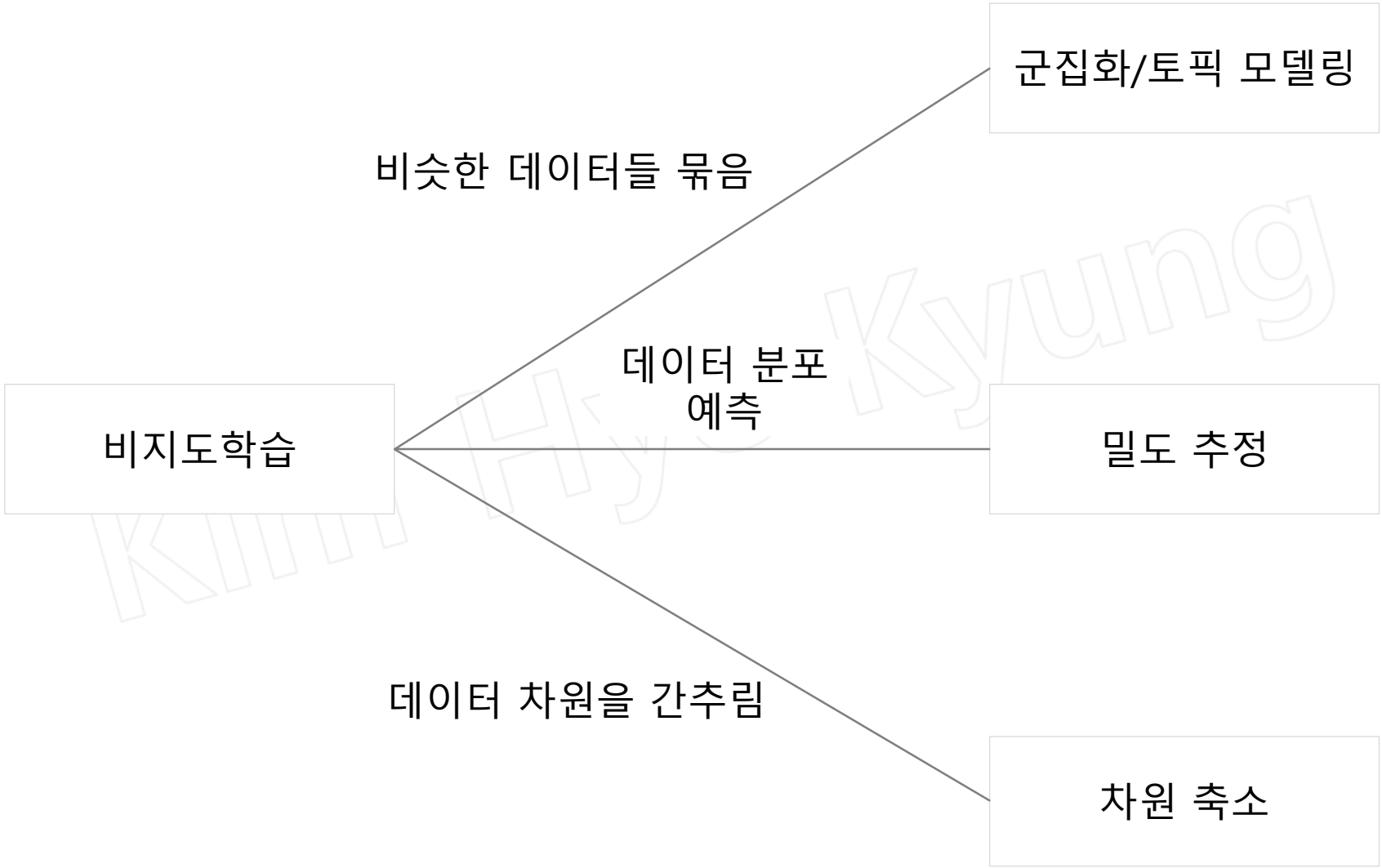
Machine Learning 범주

종류	설명
교사(지도) 학습 (supervised learning)	데이터와 답(레이블)을 입력 다른 데이터의 값을 예측 컴퓨터가 자동으로 빠르고 정확하게 분류와 판별을 수행
비교사(비지도) 학습 (unsupervised learning)	데이터는 입력하지만 답은 입력하지 않음 미지의 데이터를 이해하기 쉽게 하기 위한 분석 방법 때론 지도 학습을 위한 전처리로 사용됨 다른 데이터의 규칙성을 찾음 최종적인 답이 정해져 있지 않음 군집화 방법으로 K-평균 알고리즘(K-means)이 유명, 신경망에서는 오토 인코더(Auto Encoder)라는 방법이 있음 예 : 클러스터 분석(Cluster analysis), 주성분 분석(Principal component analysis), 벡터 양자화(Vector quantization), 자기 조직화(Self organization)등
강화 학습 (reinforcement learning)	Machine Learning을 이용한 프로그램에 데이터를 주고 어떤 데이터가 출력되었을 때 그 출력이 어느정도 좋은지 평가해서 그 에 따른 보상을 주고 그 보상을 좀 더 얻기 위해 조정해 가는 학습 방법 입력 데이터에 대해 직접적인 정답 데이터를 줄 필요가 없음

지도 학습의 세부 분류



비지도 학습의 세부 분류



Machine Learning 작업시 주요 단계

- 특징 공학
  - Feature engineering
  - 문제에 대해 가지고 있는 정보를 모두 취해 특징 행렬을 구축하는데 사용 할 수 있는 숫자로 변환하는 것

특징 공학

- 특징을 성격에 따라 표현해 보기

범주 데이터(categorical data)를 표현하는 특징

텍스트를 표현하는 특징

이미지를 표현하는 특징

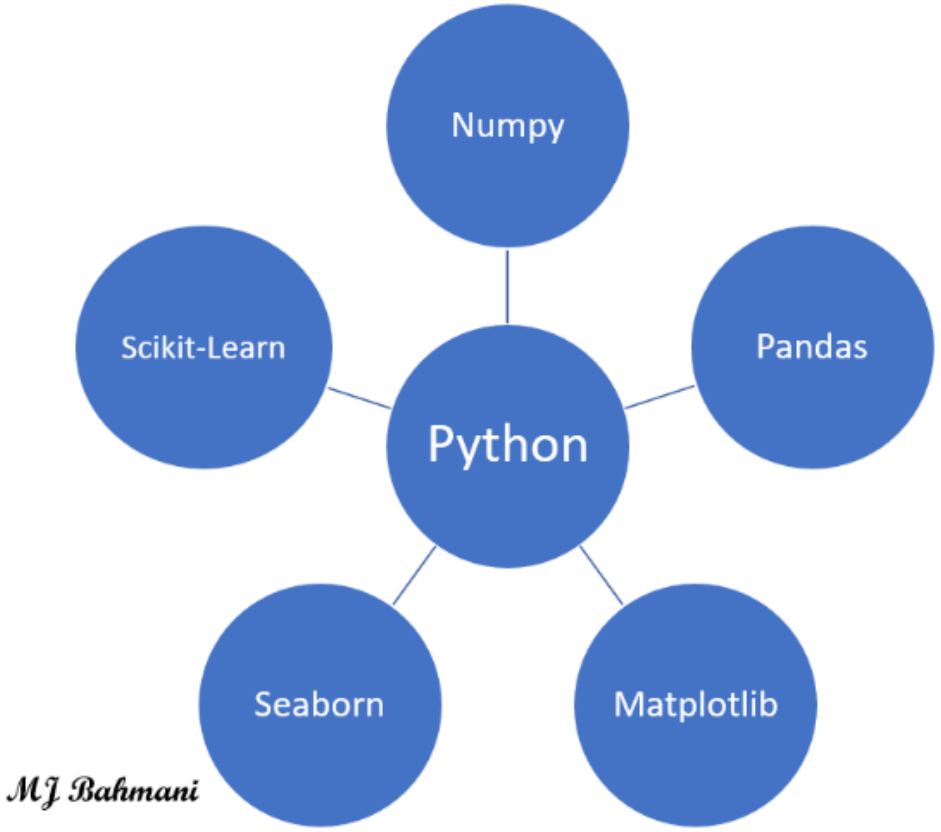
특징 공학

- 특징을 성격에 따라 표현해 보기

범주 데이터(categorical data)를 표현하는 특징

원-핫 인코딩(one-hot encoding)

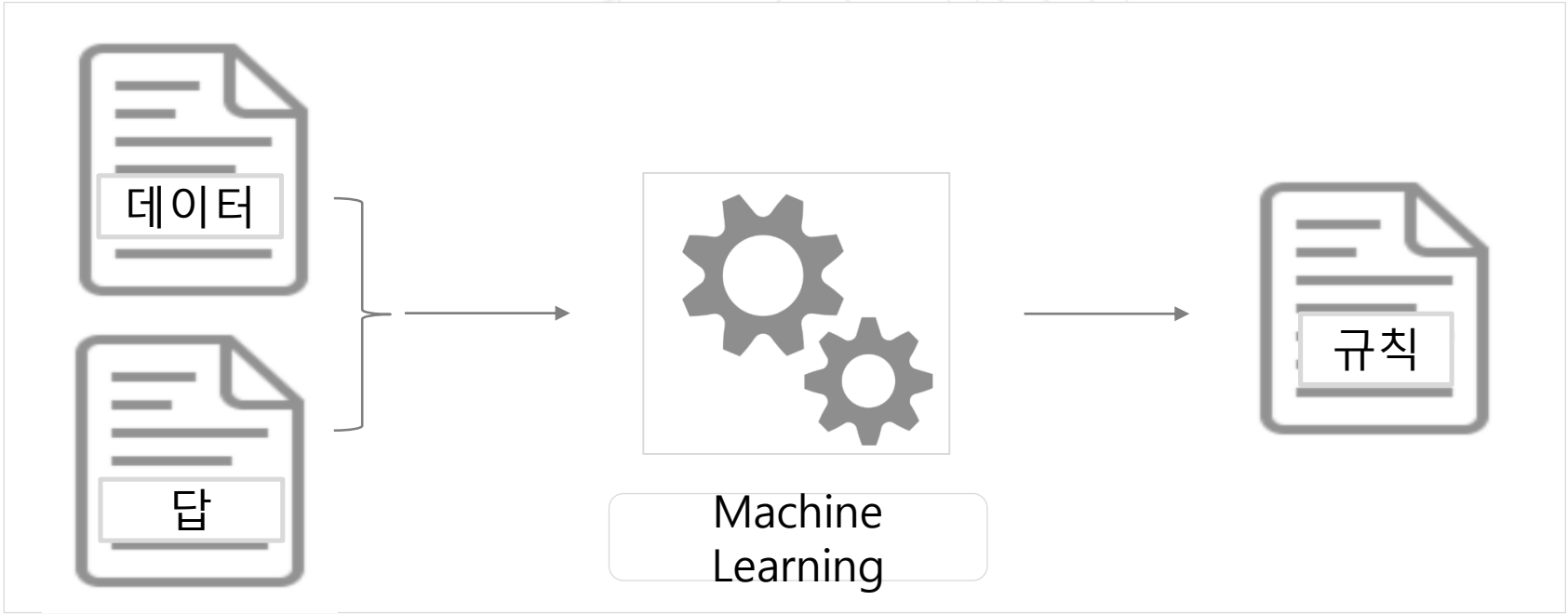
Machine Learning을 위해 필요한 학문



Python Machine Learning 생태계를 구성하는 주요 package

Machine Learningpackage	Scikit-Learn 데이터 마이닝 기반의 Machine Learning에서 독보적
행렬/선형대수/통계 package	Machine Learning의 이론적 백그라운드는 선형대수와 통계로 구성 파이썬의 대표적인 행렬과 선형대수를 다루는 package로 Numpy
데이터 핸들링	Pandas 는 파이썬 세계의 대표적인 데이터 처리 package 2차원 데이터 처리에 특화되어 있음
시각화	Matplotlib은 python의 대표적인 시각화 library Seaborn은 Matplotlib의 간소화한 library로 Matplotlib을 기반으로 개발

Machine Learning과 일반 프로그래밍 차이점



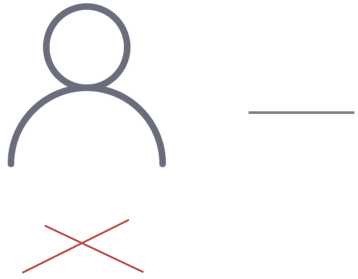
Machine Learning 필요성 체감하기

단순한 식별

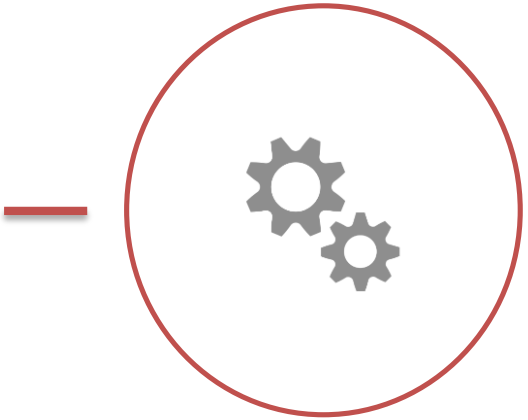
- 특징 벡터, 식별 경계
- Machine Learning 사용 없이 단순한 데이터를 기반으로 설계자가 직접 정한 규칙으로 데이터 식별해 보기
- 식별함수 개발해 보기

생산된 다양한 제품의 품질 확인을 사람이 아닌 기계가 대신 하려 하게 한다

기계가 x0과 x1의 값 만으로 y의 값을 판정할 수 있게 만들려면?
x1이 x0 보다 클 경우 : **합격**



입력 x0	입력 x1	결과 y	
0.2	0.7	**1**	**합격**
0.6	0.3	0	불합격
0.1	0.3	**1**	**합격**
0.3	0.2	0	불합격
…	…	…	



단순한 식별

- 규칙 기반 설계
  - 품질 관리 메니저라 가정
  - 필요 데이터
    - 특징량 또는 특징 벡터
      - x0 또는 x1과 같은 입력 데이터
    - 클래스
      - y 결과 즉 레이블
  - 품질 결과
    - 1 : 합격
    - 0 : 불합격
    - x1이 x0 보다 클 경우 : **합격**

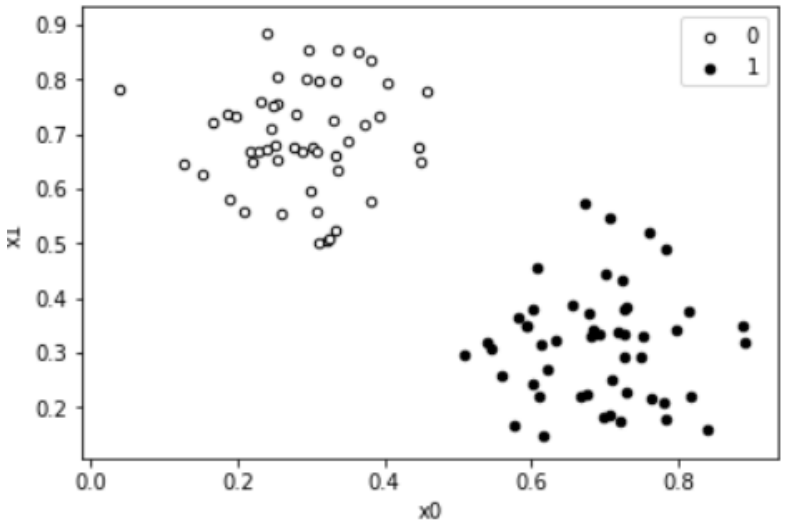
입력 x0	입력 x1	결과 y	
0.2	0.7	1	합격
0.6	0.3	0	불합격
0.1	0.3	1	합격
0.3	0.2	0	불합격
…	…	…	

단순한 식별

생산 데이터

입력 x0	입력 x1	결과 y	
0.2	0.7	1	합격
0.6	0.3	0	불합격
0.1	0.3	1	합격
0.3	0.2	0	불합격
…	…	…	

생산 데이터 분포

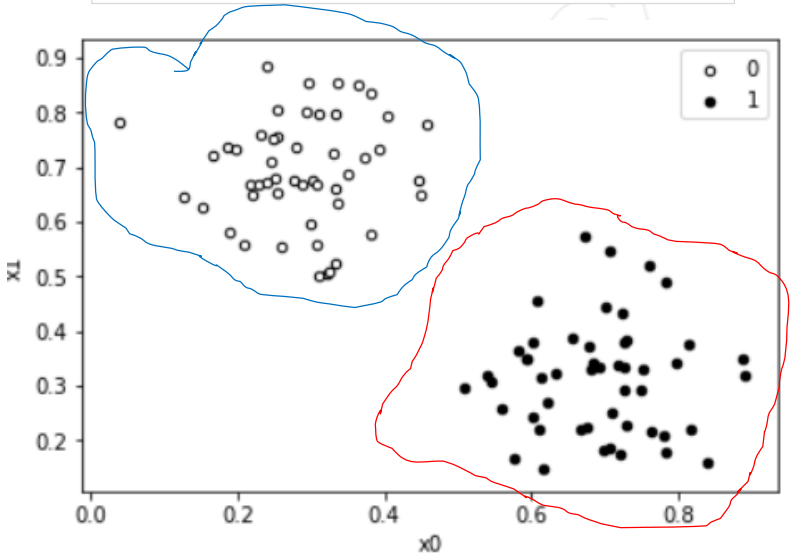


단순한 식별

생산 데이터

입력 x0	입력 x1	결과 y	
0.2	0.7	1	합격
0.6	0.3	0	불합격
0.1	0.3	1	합격
0.3	0.2	0	불합격
…	…	…	

생산 데이터 분포



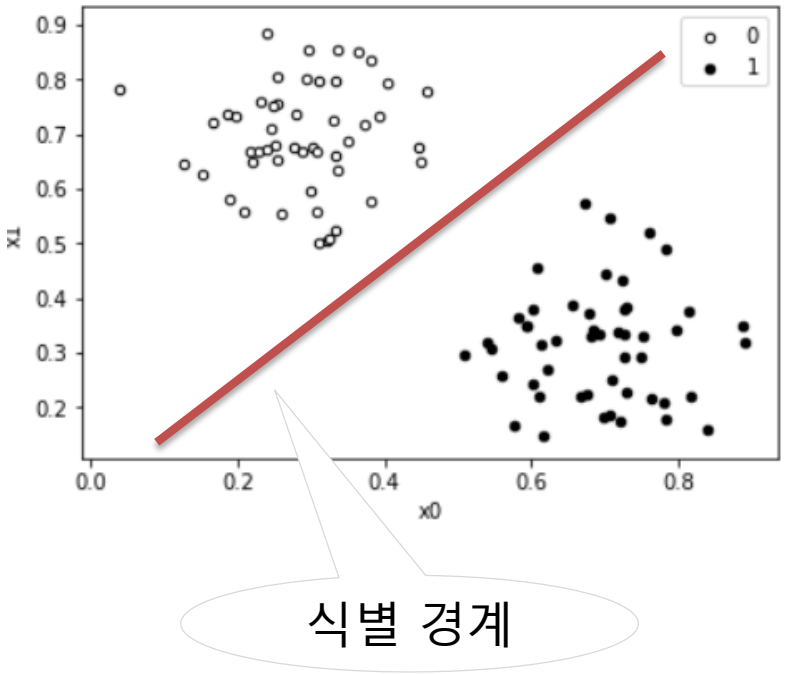
단순한 식별

- 합격 불합격의 특징 찾아보기

생산 데이터

입력 x0	입력 x1	결과 y	
0.2	0.7	1	합격
0.6	0.3	0	불합격
0.1	0.3	1	합격
0.3	0.2	0	불합격
…	…	…	

생산 데이터 분포



- 식별 경계를 활용하여 새로운 데이터 검증시 기계적으로 판정 가능

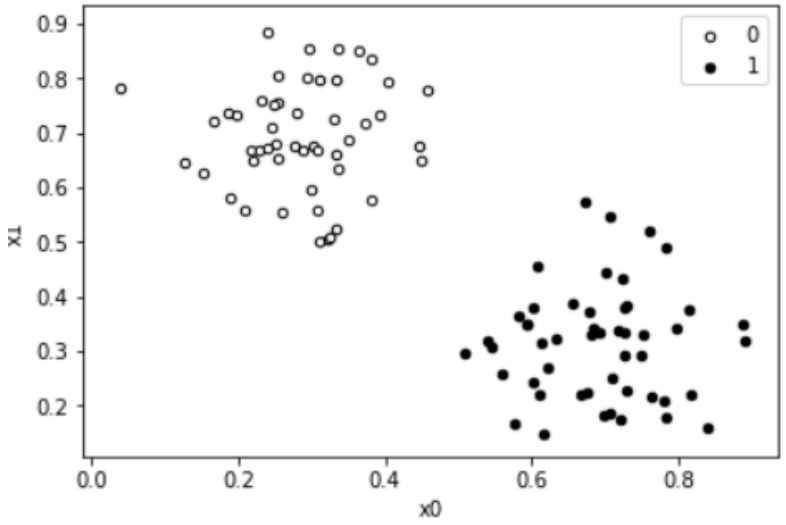
단순한 식별

- x1이 x0 보다 클 경우 : **합격**
- 문제 1 : x0=0.8, x1=0.4인 경우 합격? 불합격?
- 문제 2 : x0=0.5, x1=0.5인 경우 합격? 불합격?

생산 데이터

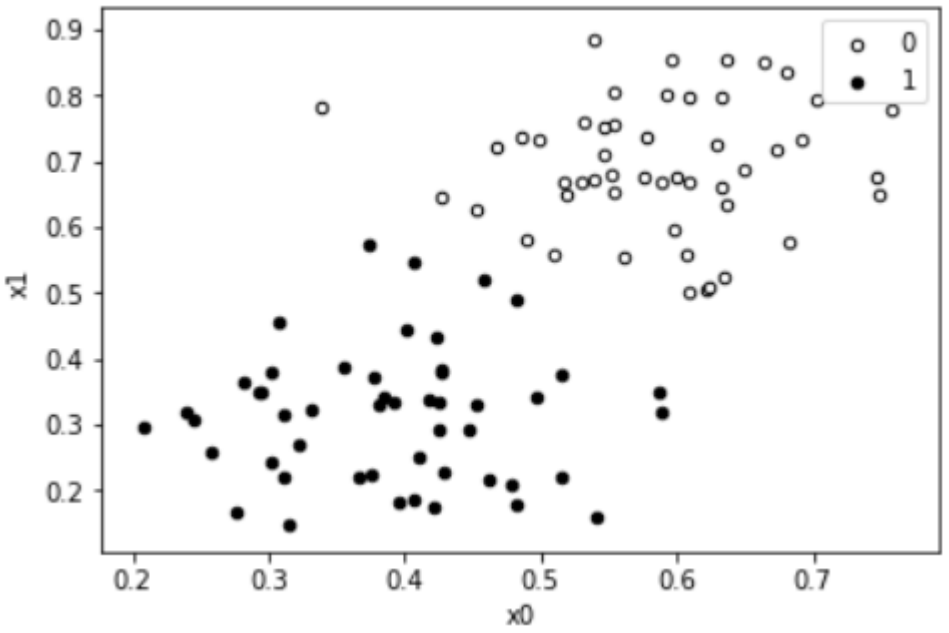
입력 x0	입력 x1	결과 y	
0.2	0.7	1	합격
0.6	0.3	0	불합격
0.1	0.3	1	합격
0.3	0.2	0	불합격
…	…	…	

생산 데이터 분포



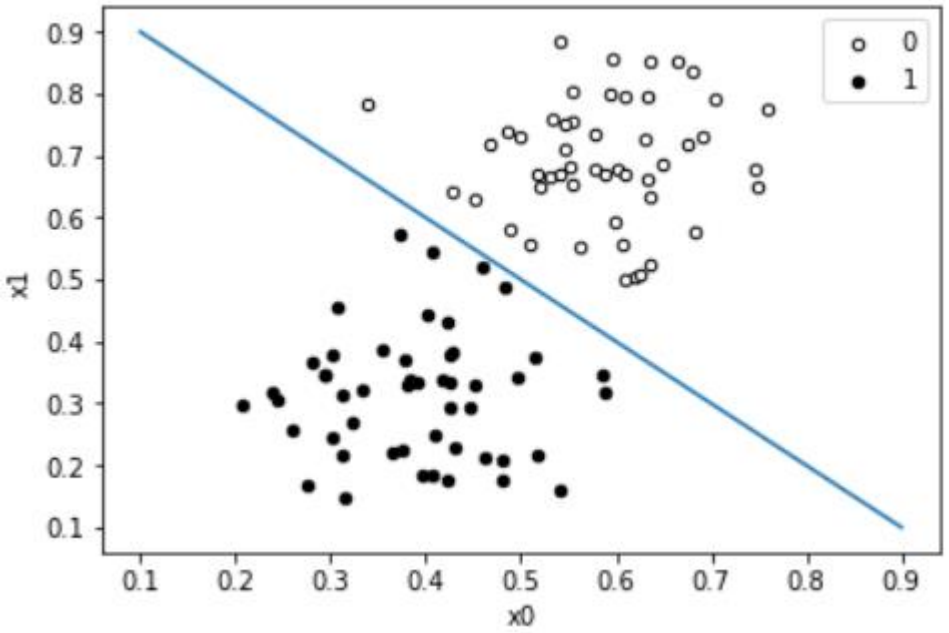
단순한 식별

- 다음 결과치를 보고 눈대중으로 식별 경계선 표기해 보기

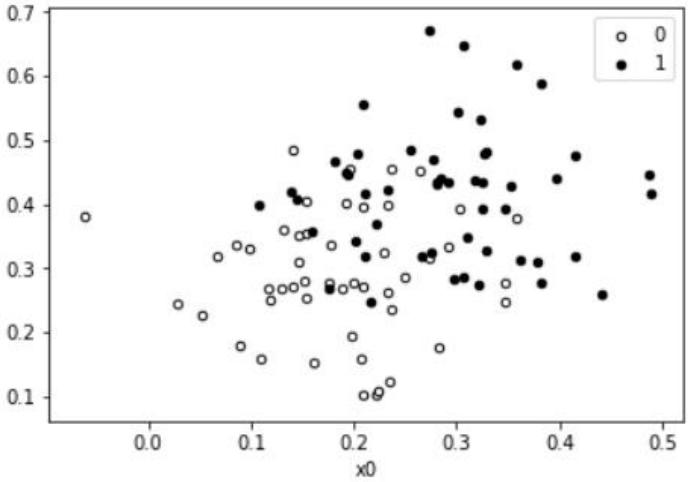


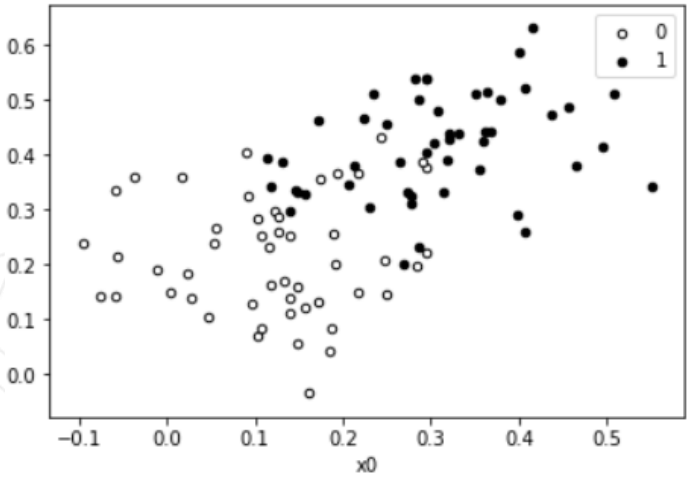
단순한 식별

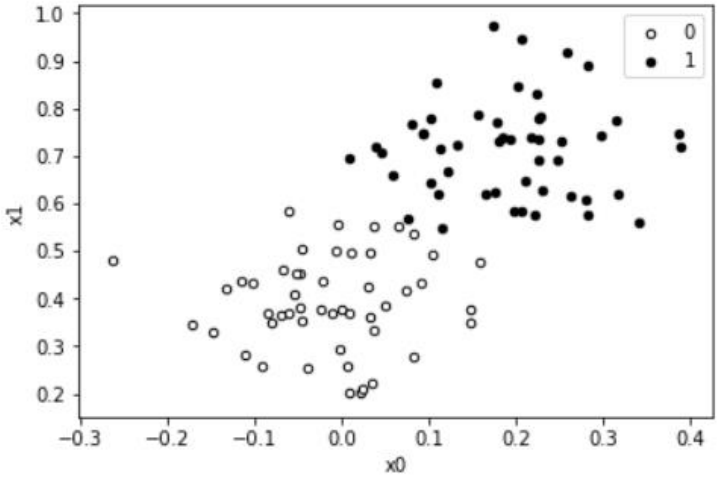
- 다음 결과치를 보고 눈대중으로 식별 경계선 표기해 보기



문제 : 식별 경계를 어떻게 구분할 것인가?







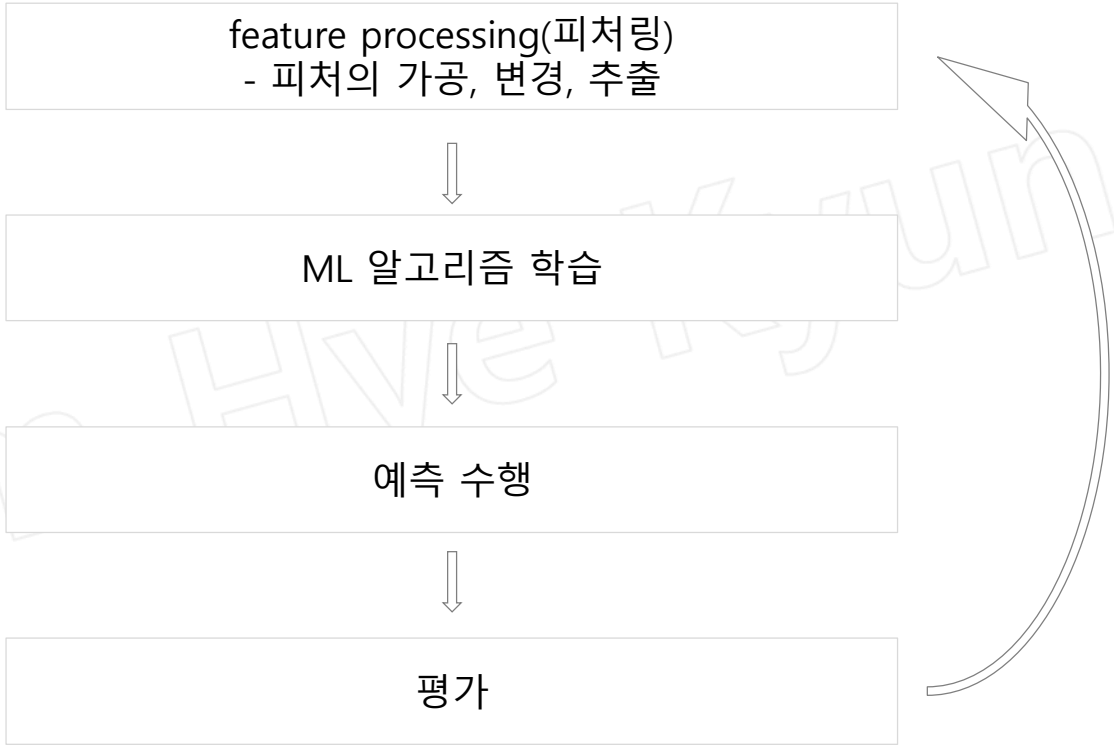
답안 :

- 2차원 평면 데이터인 경우 사람이 눈으로 어느정도 판단 가능
- 그러나 데이터가 100가지의 종류라면 100차원을 인식해야 함
- 규칙 기반으로 식별하는 방법의 한계

- **해결책 : 기계가 자동으로 식별 경계를 표현하게 하는 Machine Learning 기술 필요**

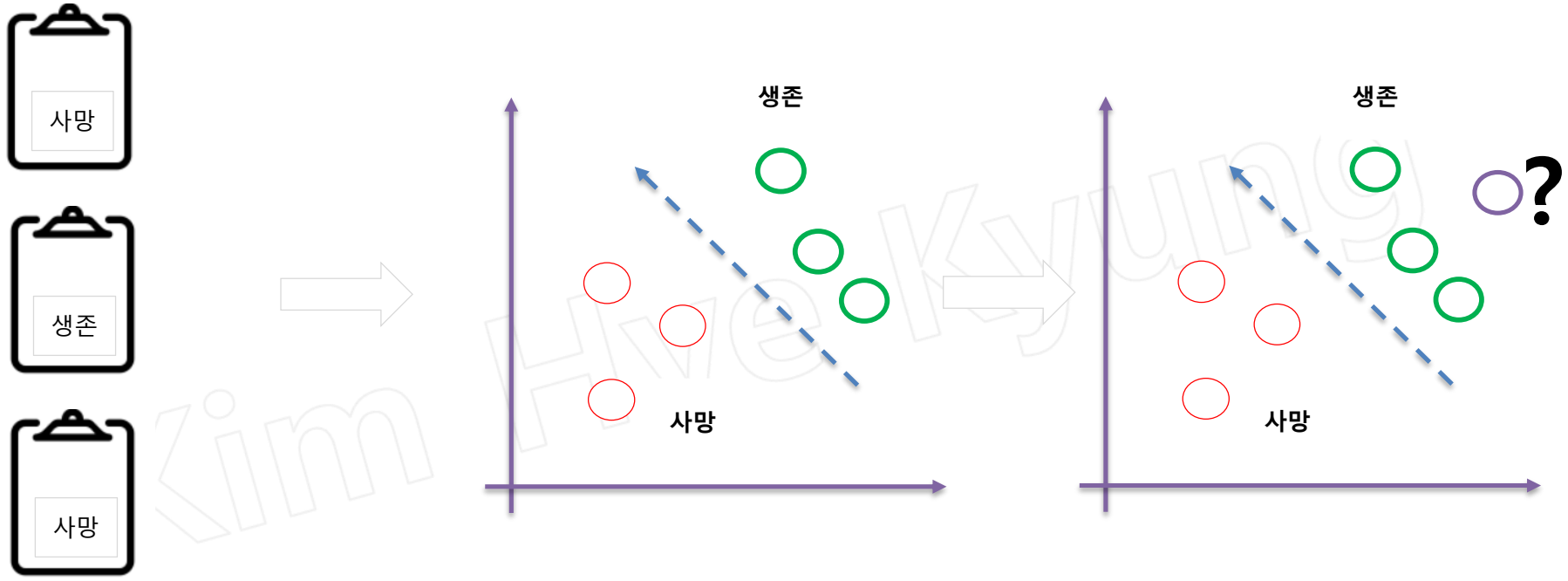
Machine Learning

Machine Learning 구축을 위한 주요 Process



Machine Learning 학습 및 예측 과정

- 환자들의 분포도를 기반으로 한 예측



기존 환자 데이터

Machine Learning 으로 학습

새로운 환자 예측

- 학습
  - 데이터가 입력되고 패턴이 분석되는 과정

개념 및 용어 정리

종류	설명
학습	모델을 **학습**시킨다는 것은 단순히 말하자면 라벨이 있는 데이터로부터 올바른 가중치와 편향값을 학습(결정)하는 것입니다 지도 학습에서 Machine Learning 알고리즘은 다양한 예를 검토하고 손실을 최소화 하는 모델을 찾아봄으로써 모델을 만들어내는데, 이 과정을 **경험적 위험 최소화**라고 합니다. 모델의 예측이 완벽하면 손실은 0이고 그렇지 않으면 손실은 그보다 커짐
학습방법	모델을 학습하려면 모델의 손실을 줄이기 위한 좋은 방법이 필요합니다. 반복 방식은 손실을 줄이는 데 사용되는 일반적인 방법 중 하나로 매우 간편하고 효율적
미니 배치 확률적 경사하강법(미니 배치 SGD)	전체 배치 반복과 SGD 간의 절충안입니다. 미니 배치는 일반적으로 무작위로 선택한 10개에서 1,000개 사이의 예로 구성됩니다. 미니 배치 SGD는 SGD의 노이즈를 줄이면서도 전체 배치보다는 더 효율적

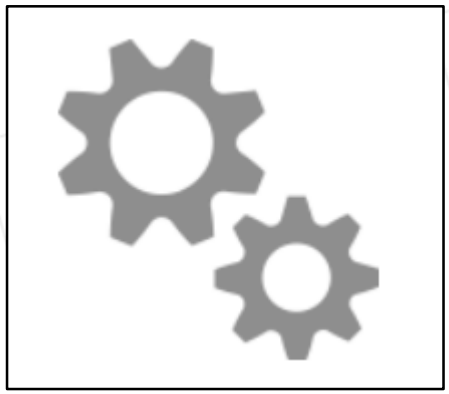
개념 및 용어 정리

종류	설명
가중치(Weight)	평균치—산출할 때 각각의 개별에 주어지는 중요도
데이터마이닝이란	Machine Learning 기술을 적용해서 대용량의 데이터를 분석하면 겉으로는 보이지 않던 패턴을 발견할 수 있는 기술
하이퍼 파라미터	**Machine Learning 알고리즘별로 최적의 학습을 위해 직접 입력하는 파라미터들의미** **하이퍼 파라미터를 통해 Machine Learning 알고리즘의 성능을 튜닝 할 수 있음**
학습 데이터 셋트	학습을 위해 주어진 데이터 세트
테스트 데이터 세트	Machine Learning 모델의 예측 성능을 평가하기 위해 별도로 주어진 데이터 세트

Machine Learning의 예측 성공률을 높이기 위한 방법들

서포트 벡터 머신
(support vector machines)

랜덤 포레스트
(random forest)



얼마나 정확한 경계선을 긋느냐?

Machine Learning의 예측 성공률을 높이기 위한 방법들

- 서포트 벡터 머신(support vector machine, SVM)
  - 기계 학습의 분야 중 하나로 패턴 인식, 자료 분석을 위한 지도 학습 모델

  - 주로 분류와 회귀 분석을 위해 사용
  - 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만듬
  - 만들어진 분류 모델은 데이터가 사상된 공간에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘
  - SVM은 선형 분류와 더불어 비선형 분류에서도 사용될 수 있음
  - 비선형 분류를 하기 위해서 주어진 데이터를 고차원 특징 공간으로 사상하는 작업이 필요한데, 이를 효율적으로 하기 위해 커널 트릭을 사용하기도 함

Machine Learning의 예측 성공률을 높이기 위한 방법들

- 서포트 벡터 머신(support vector machine, SVM)
  - 기계 학습의 분야 중 하나로 패턴 인식, 자료 분석을 위한 지도 학습 모델
  - 주로 분류와 회귀 분석을 위해 사용
  - 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만듦
  - 만들어진 분류 모델은 데이터가 사상된 공간에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘
  - SVM은 선형 분류와 더불어 비선형 분류에서도 사용될 수 있음
  - 비선형 분류를 하기 위해서 주어진 데이터를 고차원 특징 공간으로 사상하는 작업이 필요한데, 이를 효율적으로 하기 위해 커널 트릭을 사용하기도 함

- 랜덤 포레스트
  - 여러 개의 결정 트리들을 임의적으로 학습하는 방식의 앙상블 방법
  - 방법
    - 크게 다수의 결정 트리를 구성하는 학습 단계와 입력 벡터가 들어왔을 때, 분류하거나 예측하는 테스트 단계로 구성
    - 랜덤 포레스트는 검출, 분류, 그리고 회귀 등 다양한 애플리케이션으로 활용되고 있음

데이터 표현

- 특성 추출
  - 기존의 프로그래밍에서는 코드에 중점을 둠
  - Machine Learning 프로젝트에서는 표현에 중점을 둠
  - 즉, 개발자는 특성을 추가하고 개선하여 모델을 다듬어 나감
  - Machine Learning이나 딥러닝 알고리즘은 수치로 된 데이터만 학습

숫자값 매핑	정수 및 부동 소수점인 경우 특수한 인코딩을 하지 않음
문자열값 매핑	모델은 문자열 값을 학습할 수 없으므로 특성 추출을 수행하여 추출한 값을 숫자로 변환 **레이블 인코딩 또는 원핫 인코딩**
범주형(열거값) 매핑	각 범주형 데이터는 **원-핫 인코딩** 형태로 변환

데이터 표현

- 범주형 데이터 다루기
- 원핫인코딩
  - 예 : 과일이라는 컬럼에 사과, 배, 귤이 들어있다고 가정, 이 때 각각의 과일인 사과, 배, 귤로 컬럼을 만들어 주고 해당 되는 과일에만 1로 표기를 해주고 나머지 과일은 0으로 표기

원핫인코딩 전

과일
사과
배
귤
사과
포도

원핫인코딩 후

과일	과일_사과	과일_배	과일_귤	과일_포도
사과	1	0	0	0
배	0	1	0	0
귤	0	0	1	0
사과	1	0	0	0
포도	0	0	0	1

데이터 표현 – 발생 가능한 경우의 수

- 데이터를 숫자로 변경할 때 주의 사항
  - 연속형 자료로 인식 할 수도 있다…?
  - 예시
    - 데이터 : red, blue, yellow
  - 데이터를 숫자로 할당했을 때 문제점
    - red=1, blue=2, yellow=3 scikit-learn에서는 연속형 자료로 인식하여 yellow는 blue나 red 보다 높다 로 받아들임
    - 그 결과, 퍼포먼스는 형편 없이 떨어지게 됨
    - 따라서 **one hot encoding** 후, 학습에 활용하면 연속형 자료로 인식하는 문제는 해결
    - **데이터의 용량은 늘어나지만 색끼리 전혀 관련성이 없을 경우 one hot encoding 권장**

Machine Learning 작업을 위한 필요사항

- 데이터가 어떤 특징을 보유하고 있는지 찾고 벡터로 변환
  - 특징 추출이라 함
  - 특징량이란?
    - 어떤 요소가 모여 있는 것 의미
- 회귀 분석 이해하기
  - 통계 용어
    - 통계학에서, 회귀 분석(回歸 分析, 영어: regression analysis)은 관찰된 연속형 변수들에 대해 두 변수 사이의 모형을 구한뒤 적합도를 측정해 내는 분석 방법
  - Y가 연속적인 값일 때 Y=f(x)와 같은 모델로 표현
  - Y : 연속 측정의 종속 변수, X : 독립 변수
  - X가 1차원 : 단순 회귀
  - X가 2차원 : 다중 회귀

Machine Learning의 특징

데이터

데이터를 기반으로 함
여러 규칙을 단순 조합하는 '고전적인 인공지능 시스템 ' 과 달리 알고리즘이 아닌 데이터 학습을 통해 실행 동작이 바뀜

데이터를 기반으로 하기 때문에 통계학에 가까움

패턴인식

통계학 및 딥러닝을 이용한 데이터 패턴을 유추하는 방법이 주축

데이터를 보고 패턴을 추리는 것

컴퓨터를 이용한 계산

Machine Learning은 데이터를 처리하고 패턴을 학습하고 계산하는데 컴퓨터 사용

실제 데이터에 대해 계산해서 결과를 만들어 낸다는 점에서 Machine Learning은 전산학의 한 분야로 볼수 있음

Machine Learning 유형

지도학습	레이블이 있는 훈련데이터를 기반으로 레이블을 예측할 수 있는 모델
분류	둘 이상의 이산적인 범주로 레이블을 예측하는 모델
회귀	연속적인 레이블을 예측하는 모델
비지도학습	레이블이 없는 데이터의 구조를 식별하는 모델
군집화	데이터의 개별 그룹을 탐지하고 식별하는 모델
차원 축소	고차원 데이터의 저차원 구조를 탐지하고 식별는 모델

Machine Learning 응용 분야

클래스 분류

특정 데이터에 레이블을 붙여 분류
(스팸 메일 분류, 필기 인식, 증권사기등)

클러스터링[그룹 나누기]

값의 유사성을 기반으로 데이터를 여러 그룹으로 구분
(사용자의 취향별 그룹으로 구분후 취향에 맞는 광고 제공)

차원 축소

데이터의 특성을 유지하면서 데이터의 양을 감소하는 것
고차원의 데이터를 저차원의 데이터로 변환
(데이터 시각화, 구조 추출, 용량 감소, 메모리 절약등)

회귀[regression]
과가의 데이터를 기반으로 미래의 데이터 예측
(판매 예측, 주가 변동 예측 등)

추천[recommendation]

특정 데이터를 기반으로 다른 데이터 추천
(이미 인터넷 서점에서 구매한 책을을 기반으로 다른책 추천)

Machine Learning 개발 process

데이터 수집

데이터 클리닝 및 가공

학습 & 예측

평가

추론

Machine Learning 개발 process

- 데이터 수집
- 데이터 클리닝 및 가공
  - 클리닝이란?
    빠진 데이터를 채우거나 틀린 데이터를 제거하는 과정 의미
  - 가공이란?
    문자 데이터를 숫자로 변환 하는 것등 의미
- 학습 & 예측
  - 학습 전용 데이터
  - 학습 방법 선택 | 매개변수 조정 | 모델 학습
- 평가
  - 테스트 전용 데이터
- 추론
  - 새로운 데이터에 모델을 적용해 예측(추론)을 하고 이 모델을 일반화

Machine Learning 개발 process

- 데이터 평가 방법과 튜닝시 고려해야 할 사항

클리닝과 가공

학습, 평가, 추론

Machine Learning 개발 process

- 클리닝과 가공
  - Machine Learning은 단순 fit() 으로 학습만 하는게 아님
  - 데이터 수집, 수집한 데이터를 클리닝하고 가공해야 함

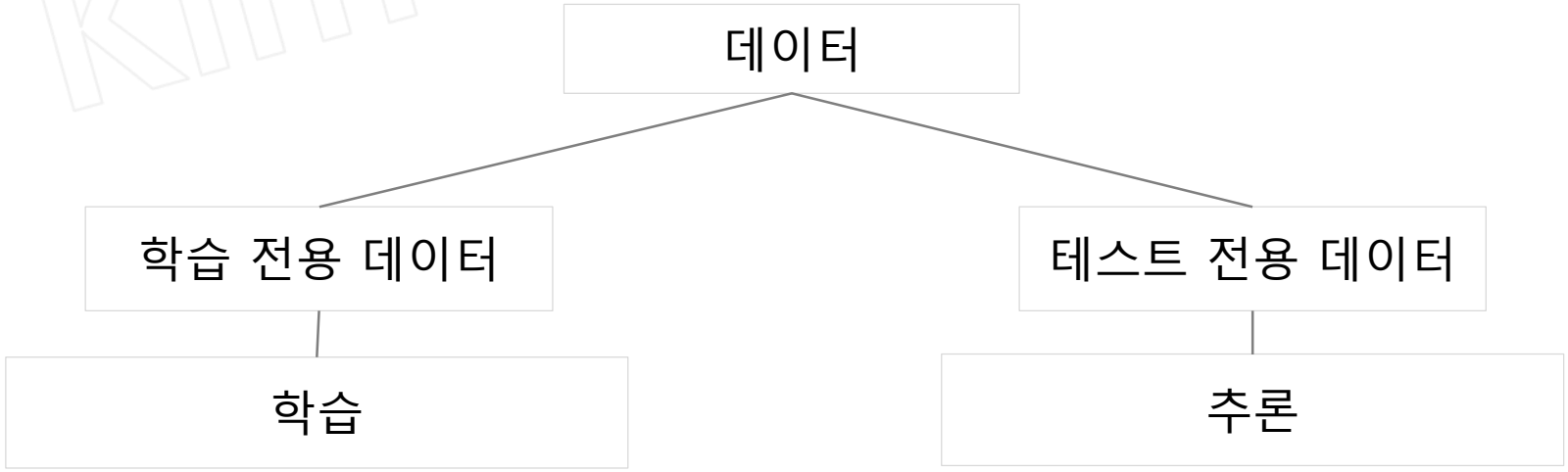
  - 클리닝이란?
    - 빠진 데이터를 채우거나 틀린 데이터를 제거하는 과정등을 의미
  - 가공이란?
    - 문자 데이터를 숫자 데이터로 변환 하는 것등을 의미

식별기는 문자열을 인식하지 못하므로 고객의 '성별 ' 이라는 특징량을 '남성 ' 과 '여성 ' 이라는 문자열 그대로는 사용 불가

남자는 '0', 여자는 '1'로 변환해야 하는 과정 필요

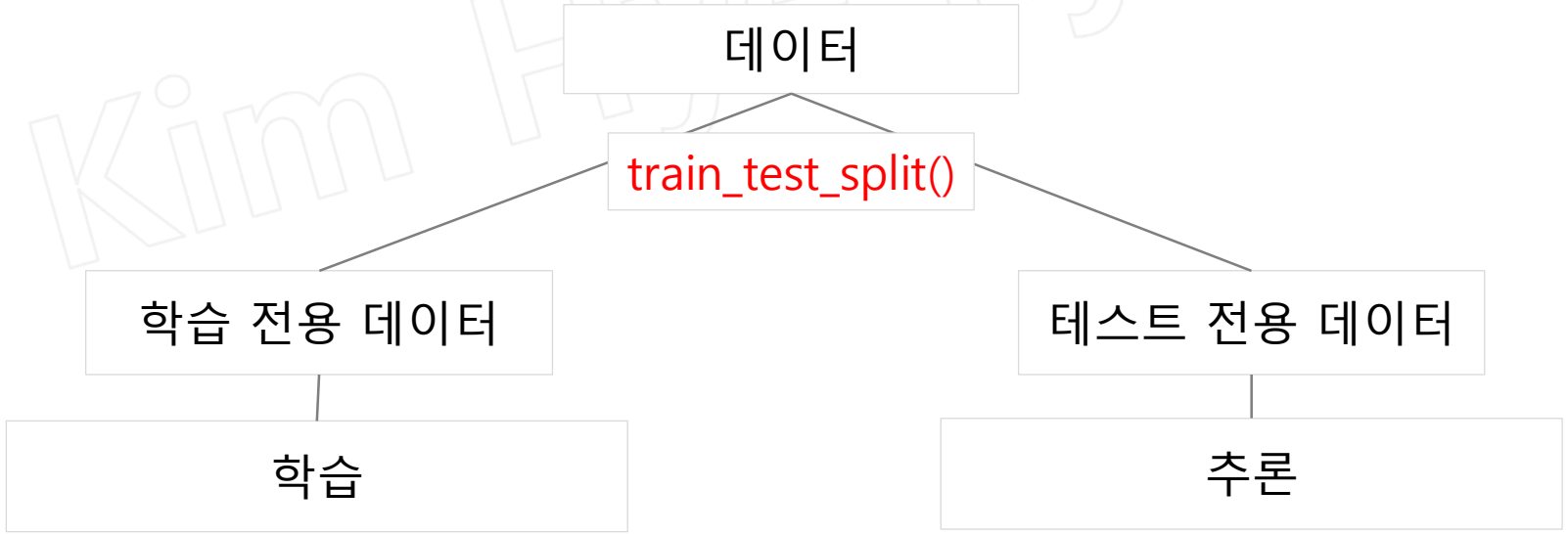
Machine Learning 개발 process

- 학습, 평가, 추론
  - 수집하고 클리닝한 데이터만으로 학습 할 경우 발생 가능한 문제
    - 추론하고 싶은 미지의(실제 데이터)만으로는 학습이 제대로 되었는지 판정 불가
  - 해결책
    - 데이터 분리를 **학습 전용 데이터**와 **테스트 전용 데이터**로 구분



Machine Learning 개발 process

- 학습, 평가, 추론
  - 데이터 분리를 **학습 전용 데이터**와 **테스트 전용 데이터**로 구분
  - train_test_split(특징량, 레이블 [,test_size=테스트 전용 데이터 비율]) 함수 사용

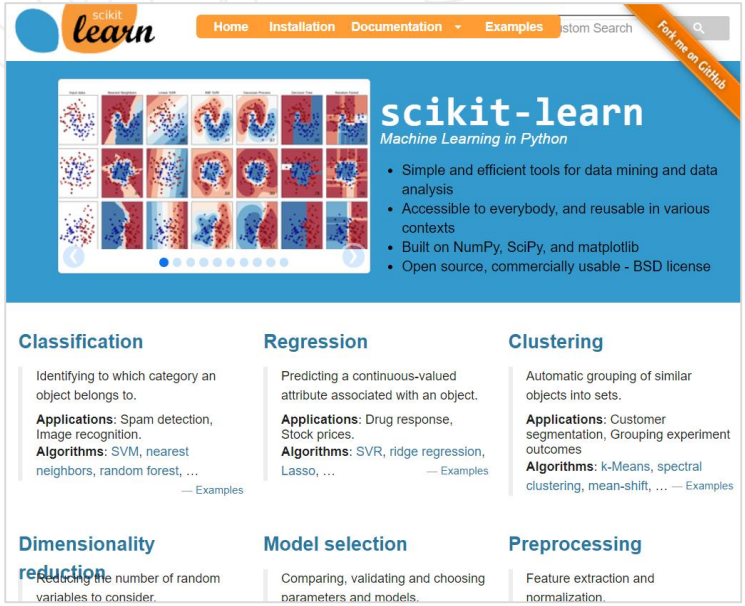


Machine Learning Framework

Scikit-learn 개요

Scikit-Learn

- 파이썬 Machine Learning 라이브러리의 정석
- 파이썬 Machine Learning library중 가장 많이 사용되며, 가장 쉽고 효율적인 개발 library 제공
- 특징
  - 깔끔하고 일관되고 간결한 API와 매우 유용한 온라인 문서 제공
  - 다양한 분류기 지원
  - 일관성 유지
    - 한 가지 유형의 모델에 대한 Scikit-learn 기본 사용법과 구문을 익히고 나면 새로운 모델이나 알고리즘으로 전환하는 것이 매우 쉬움



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

Scikit-Learn 데이터 표현 방식

- Machine Learning은 데이터로부터 모델을 만듬
- 컴퓨터가 이해할 수 있는 데이터로 표현하는 방식이 우선

- **Scikit-Learn에서는 데이터를 테이블 관점으로 이해하는 것이 우선**

테이블로서의 데이터

특징 행렬(feature matrix)

대상 배열

Scikit-Learn 데이터 표현 방식

- 테이블로서의 데이터
  - 기본 테이블 : 2차원 데이터 grid
  - Row : 데이터 세트의 개별 요소
  - Column[feature] : 각 열은 각 표본을 설명하는 특정 수량 정보 의미

표본(sample) : 행렬의 행

Feature : 행렬의 열의 특징

```
1 import seaborn as sns
2 import numpy as np
```

```
1 iris = sns.load_dataset('iris')
2 iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Scikit-Learn 데이터 표현 방식

- 특징 행렬(feature matrix)
  - 정보를 2차원 수치 배열이나 행렬로 봄

```
1 import seaborn as sns
2 import numpy as np
```

```
1 iris = sns.load_dataset('iris')
2 iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

표본(sample) : 행렬의 행

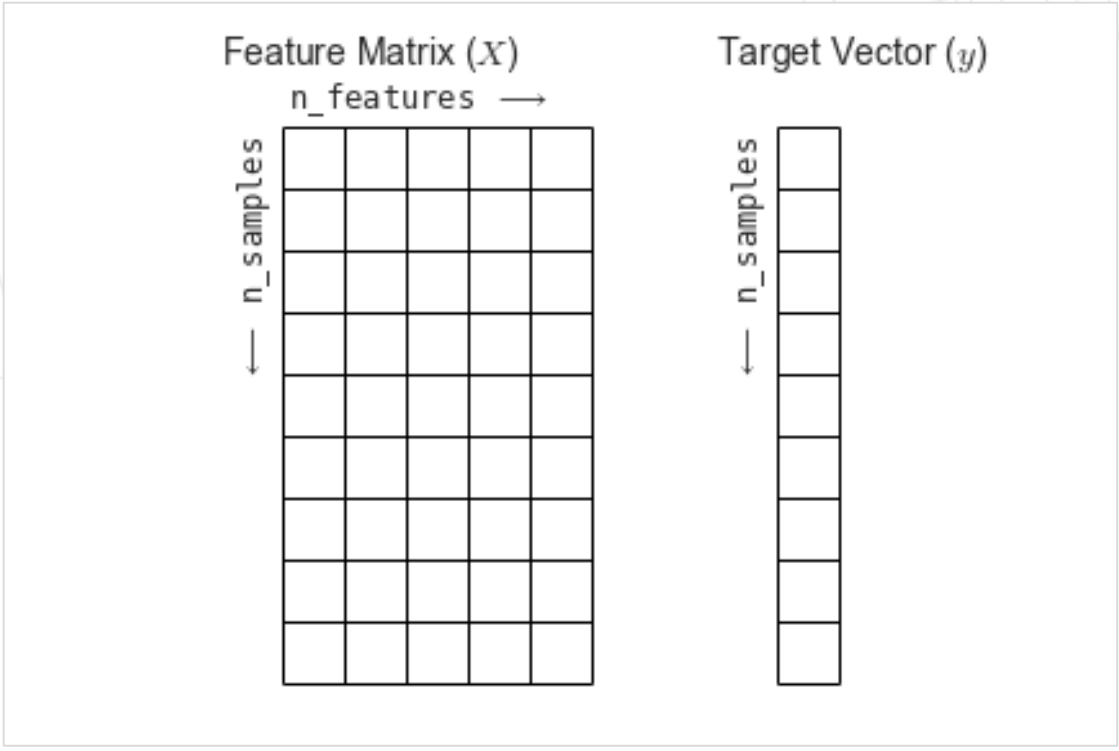
- 데이터 세트가 설명하는 개별 객체
- 꽃, 사람, 문서, 이미지, 음성파일, 동영상등 정량적 수치로 설명할 수 있는 모든 것

Feature : 행렬의 열의 특징

- 각 표본을 정량적 방식으로 표현하는 개별 관측치
- 대개가 실측치에 따라 Boolean 또는 이산값일수도 있음

Scikit-Learn 데이터 표현 방식

- 대상 배열
  - Scikit-Learn의 데이터 레이아웃
  - 특징 행렬 + 관례상 y라고 부르는 레이블



Machine Learning Framework

Scikit-Learn API

Scikit-learn

- 다양한 분류기 지원
- Machine Learning의 결과를 검증하는 기능도 포함
- 분류, 회귀, 클러스터링, 차원 축소처럼 Machine Learning에서 자주 사용되는 다양한 알고리즘 지원

- 설치 명령어

```
pip install –U scikit-learn scipy matplotlib scikit-image

pip install pandas
```

Scikit-Learn 주요 모듈

분류	모듈명	특징
예제 데이터	sklearn.datasets	사이킷런에서 예제로 제공하는 데이터 셋
피처 처리	sklearn.preprocessing	데이터 전처리에 필요한 다양한 가공 기능 제공(문자열을 숫자로 인코딩, 정규화, 스케일링 등)
피처 처리 & 차원 축소	sklearn.decomposition	차원 축소 알고리즘 지원 모듈
데이터 분리, 검증 & 파라미터 튜닝	sklearn.model_selection	교차 검증을 위한 학습용/테스트행 분리 GridSearch로 최적 파라미터 추출등의 API 제공
평가	sklearn.metrics	분류, 회귀, 클러스터링등에 대한 다양한 성능 측정 방법 제공

Scikit-Learn 주요 모듈

분류	모듈명	특징
ML 알고리즘	sklearn.ensemble	앙상블 알고리즘 제공 렌덤 포레스트, 그레디언트 부스팅 등
	sklearn.linear_model	선형회귀, 라쏘(Lasso) 및 로지스틱 회귀등 회귀 관련 알고리즘 지원
	sklearn.svm	서포트 벡터 머신 알고리즘 제공
	sklearn.tree	의사 결정 트리 알고리즘 제공
	sklearn.cluster	비지도 클러스터링 알고리즘 제공(k-평균, DBSCAN등)
…		

Scikit-Learn 데이터 셋

- 분류나 회귀 연습용 예제 데이터

API 명	특징
datasets.load_boston()	회위 용도 미국 보스턴의 집 피처들과 가격에 대한 데이터 셋
datasets.load_breast_cancer()	분류 용도 위스콘시 유방암 피처들과 악성/음성 레이들 데이터 셋
datasets.load_diabetes()	회귀 용도 당뇨 데이터 셋
datasets.load_digits()	분류 용도 0~9까지 숫자의 이미지 픽셀 데이터 셋
datasets.load_iris()	분류 용도 붓꽃에 대한 피처를 가진 데이터 셋

Scikit-Learn 데이터 셋

- 분류와 클러스터링을 위한 표본 데이터 생성기

API 명	특징
datasets.make_blobs()	클러스터링을 위한 데이터 셋을 무작위로 생성해 줌 군집 지정 개수에 따라 여러가지 클러스터링을 위한 데이터 셋을 쉽게 구성해 줌
datasets.make_classifications()	분류를 위한 데이터 셋 특히 높은 상관도, 불필요한 속성등의 노이즈 효과를 위한 데이터를 무작위로 생성해 줌
…	

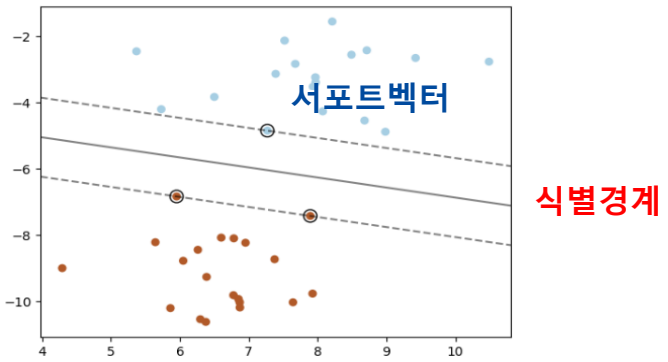
Scikit-Learn 데이터 셋

- 분류와 회귀를 위한 연습용 데이터 셋 특징
  - 딕셔너리 형태로 제공

API 속성명	특징	
data	피처의 데이터 셋 의미	numpy 배열 타입
target	분류시 레이블 값, 회귀일 때는 숫자 결과값 데이터 셋 의미	numpy 배열 타입
target_names	개별 레이블 이름	numpy 배열 또는 list 타입
feature_names	피처의 이름	numpy 배열 또는 list 타입
DESCR	데이터 셋에 대한 설명과 각 피처의 설명	문자열 타입

Scikit-Learn의 Estimator(추정기 예시) - SVM(Support Vector Machine) - 분류기

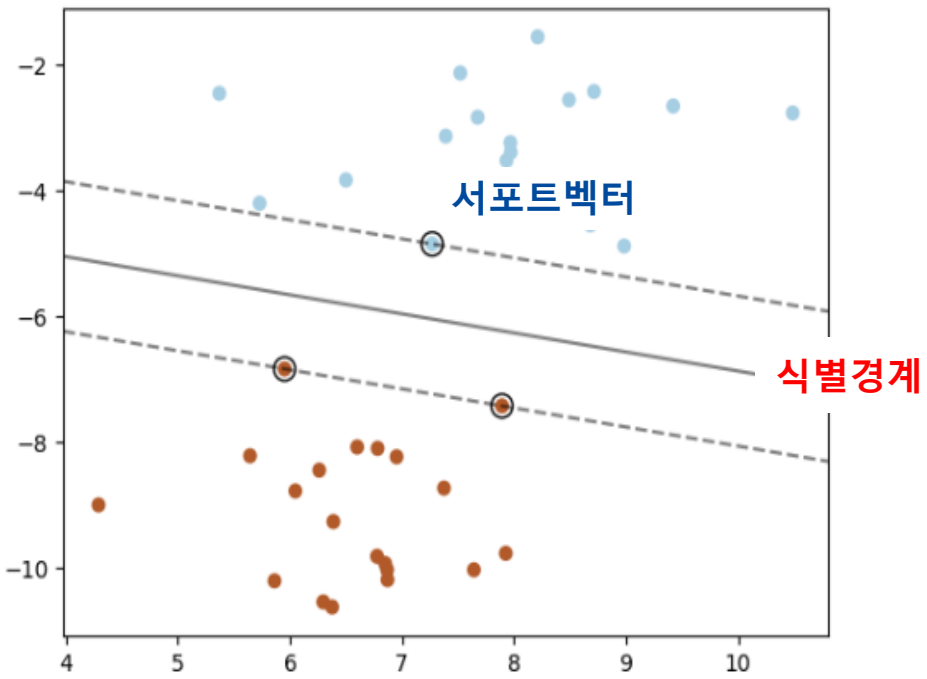
- Machine Learning 방법 중 하나로 선을 구성하는 매개변수를 조정해서 요소들을 구분하는 선을 찾고, 이를 기반으로 패턴을 인식하는 방법
- 구분선을 확실하게 정할 수 있으면 그 이후 새로운 패턴이 나타났을 때도 쉽게 분류가 가능
  - 가령, A와 B라는 두 가지 패턴이 있을 경우 A와 B를 구분하는 방법을 찾는 것이 패턴 인식의 목표
  - 식별 평면이란?
    - A와 B를 벡터로 나타내서 평면 위에 올리고 구분선을 그리게 됨
    - 이때 패턴의 경계가 되는 것 의미



Scikit-Learn의 Estimator(추정기 예시) - SVM(Support Vector Machine) - 분류기

- 미지의 데이터를 제대로 식별하기 위함
- 이 미지의 데이터에 대한 성능을 '일반화 능력' 이라고 표현
- SVM은 일반화 능력을 향상시키기 위해 훈련 데이터의 각 점으로 부터 식별 경계까지의 거리가 최대한 멀어지게

- 서포트 벡터란?
  - 훈련 경계에서 식별 경계에 가장 거리가 가까운 것



Scikit-Learn의 Estimator API

Scikit-Learn의 Estimator(추정기) API

- Scikit-Learn Estimator API 특징
  - 일관성
    - 모든 객체는 일관된 문서를 갖춘 제한된 메서드 집합에서 비롯된 공통 인터페이스를 공유함
  - 검사(inspection)
    - 모든 지정된 모수(parameter) 값은 공개(public) 속성으로 노출
  - 제한된 객체 계층구조
    - 알고리즘만 파이썬 클래스에 의해 표현되고 데이터 세트는 표준 포멧(NumPy 배열, Pandas DataFrame, SciPy 행렬등)으로 표현되며 매개 변수명은 표준 파이썬 문자열을 사용
  - 구성
    - 많은 Machine Learning 작업은 기본 알고리즘의 시퀀스로 나타낼 수 있으며, Scikit-Learn은 가능한 곳이라면 어디서든 이 방식을 사용
  - 합리적인 기본값
    - 모델이 사용자 지정 모수를 필요로 할 때 라이브러리가 적절한 기본값을 정의

Scikit-Learn의 Estimator API

- API 이용 단계

Scikit-Learn으로부터 적절한 추정기(estimator) 클래스를 import해서 모델의 클래스 선택

이 클래스를 원하는 값으로 인스턴스화해서 모델의 초모수(hyperparameters) 선택

데이터를 특징 배열과 대상 벡터로 배치

모델 인스턴스 fit() 메서드를 호출해 모델을 데이터에 적합시킴

모델을 새 데이터에 적용

1. 지도 학습인 경우
   predict() 메서드를 사용해 알려지지 않은 데이터에 대한 레이블 예측

2. 비지도 학습인 경우
   transform()이나 predict() 메서드를 사용해 데이터의 속성을 변화하거나 추론

Scikit-Learn의 Estimator API

- 1단계

Scikit-Learn으로부터 적절한 추정기(estimator) 클래스를 import해서 모델의 클래스 선택

- Scikit-Learn에선 모두 python 클래스로 표현
- 다양한 모델 클래스들 제공

Scikit-Learn의 Estimator API

- 2단계

이 클래스를 원하는 값으로 인스턴스화 해서 모델의 초모수(hyperparameters) 선택

- 모델 클래스가 모델 인스턴스와 같지 않음
- 모델 클래스를 결정했더라도 몇 가지 선택해야 할 옵션 필요

- 작업하는 모델 클래스에 따라 다음 옵션중에 선택해야 함
  - 오프셋 즉 절편에 적합한가?
  - 모델을 정규화 할 것인가?
  - 모델 유연성을 높이기 위해 특징을 사전 처리할 것인가?
  - 모델에서 어느 정도의 정규화를 사용할 것인가?
  - 얼마나 많은 모델 성분을 사용할 것인가?

Scikit-Learn의 Estimator API

- 2단계

이 클래스를 원하는 값으로 인스턴스화해서 모델의 초모수(hyperparameters) 선택

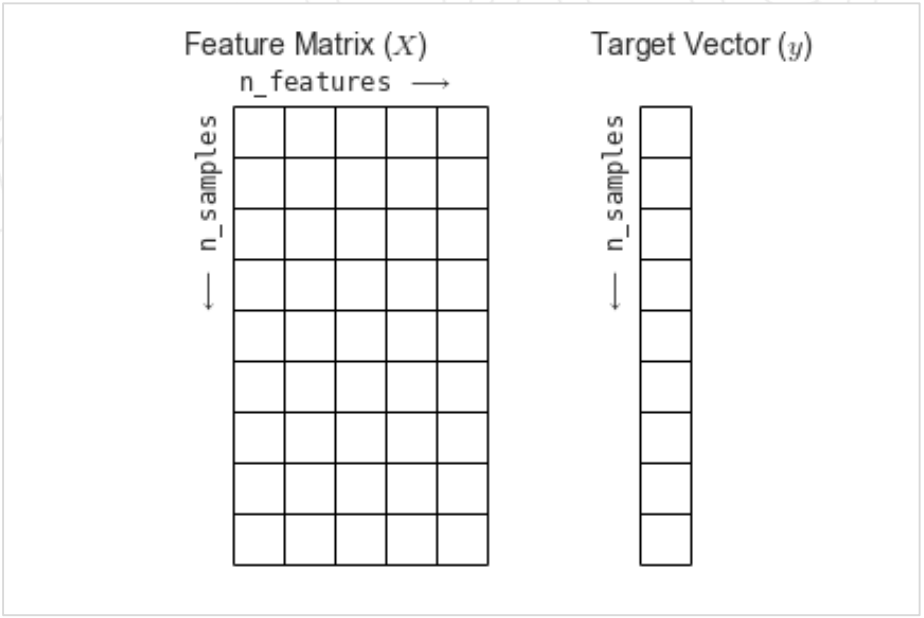
  - 모델 클래스가 정해지고 나면 선택해야 할 중요 항목이 있음
    - 선택 사항 – 초모수 또는 모델을 데이터에 적합시키기 전에 설정돼야 할 모수로 표현
    - Scikit-Learn에선 모델 인스턴스화 시점에 값을 전달함으로써 초모수를 선택
    - Scikit-Learn API는 모델을 선정하는 것과 모델을 데이터에 적용하는 것을 명확히 구분해야 함

Scikit-Learn의 Estimator API

- 3단계

데이터를 특징 배열과 대상 벡터로 배치

  - 데이터 표현시 2차원 특징 행렬과 1차원 대상 배열 필요



Scikit-Learn의 Estimator API

- 4단계

모델 인스턴스 fit() 메서드를 호출해 모델을 데이터에 적합시킴

  - 모델을 데이터에 적용(적합) 시키기
    - fit()

- 5단계

예측 수행 : 모델을 새 데이터(test 데이터)에 적용

  - 알려지지 않은 데이터에 대한 레이블을 예측함
  - predict() 이용

Scikit-Learn의 Estimator API

- 6단계

정확도 측정 :

  - 예측 결과가 실제 레이블 값과 얼마나 정확하게 맞는지 평가하는 지표
  - accuracy_score(실제 레이블 데이터 세트, 예측 레이블 데이터 세트) 사용

Scikit-Learn의 주요 API

API	특징	상세 설명
fit()	모델 학습	
predict()	학습된 모델의 예측	

train_test_split()	학습 데이터와 테스트 데이터 셋을 분리	`X_train, X_test, y_train, y_test ₩`
		`= train_test_split(iris_data, iris_label, ₩`
		`test_size=0.2, random_state=11)`

…		

Scikit-Learn의 주요 API

- 각 단계별 sample 예제
  - 지도 학습 기반의 붓꽃 분류

```
1  from sklearn.model_selection import train_test_split
2  #1단계 : 모델 클래스 선택
3  from sklearn.naive_bayes import GaussianNB
4
5
6  #2단계 : 모델 인스턴스화
7  model = GaussianNB()
8
9  #3단계 : 데이터를 특징 행렬과 대상 벡터로 배치
10 Xtrain, Xtest, ytrain, ytest = train_test_split(X_iris, y_iris, random_state=1)
11
12 #4단계 : 데이터를 특징 행렬과 대상 벡터로 배치, 모델을 데이터에 적합
13 model.fit(Xtrain, ytrain)
14
15 #5단계 : 새 데이터에 대해 예측
16 y_model = model.predict(Xtest)
```

Model Selection 모듈

Scikit-Learn의 주요 API

- sklearn.model_selection 모듈
  - 학습 데이터와 테스트 데이터 셋 분리
  - 교차 검증 분할 및 평가
  - Estimator의 하이퍼 파라미터를 튜닝하기 위한 다양한 함수와 클래스 제공

Scikit-Learn의 주요 API - sklearn.model_selection 모듈

- 학습 / 테스트 데이터 셋 분리

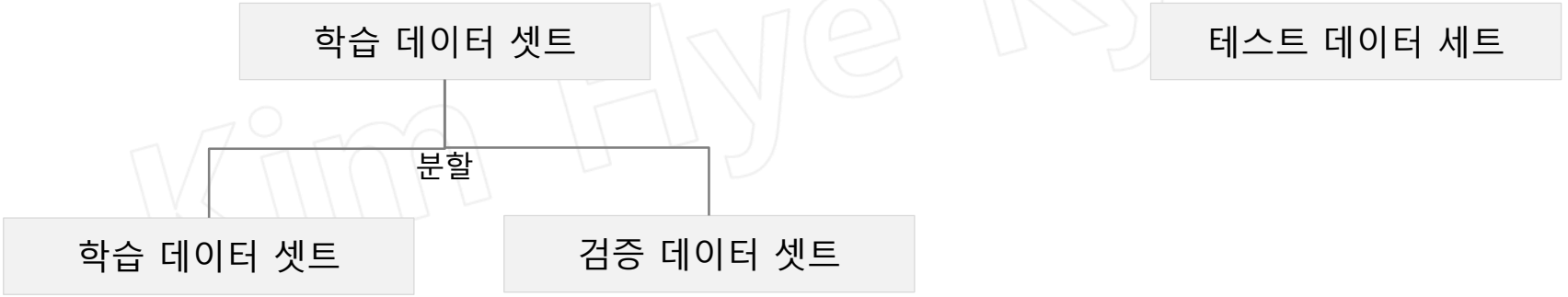
API	특징	parameter 상세 설명
train_test_split() 학습 데이터와 테스트 데이터 셋을 분리 반환값은 tuple	test_size	전체 데이터에서 테스트 데이터 세트 크기를 얼마로 샘플링 할 것인가 결정 default 0.25(25%)
	train_size	전체 데이터에서 학습용 데이터 세트 크기를 얼마로 샘플링 할 것인지 결정 잘 사용하지는 않음
	shuffle	데이터를 분리하기 전 미리 섞을지를 결정 default = True 데이터를 분산시켜서 좀더 효율적인 학습 및 테스트 데이터 세트를 만드는데 사용
	random_state	동일한 학습/테스트용 데이터 세트를생성하기 위한 설정 생략시 무작위로 데이터 분리하기 때문에 지정하지 않으면 수행할 때마다 다른 학습/테스트 용 데이터를 생성

Scikit-Learn의 주요 API - sklearn.model_selection 모듈

- 교차 검증
  - 과적합이란? 모델이 학습 데이터에만 과동하게 최적화되어, 실제 예측을 다른 데이터로 수행할 경우에는 예측 성능이 과도하게 떨어지는 것 의미
  - 고정된 학습 데이터와 테스트 데이터로 평가할 경우 발생되는 문제점
    - 테스트 데이터에만 최적의 성능을 발휘 할 수 있도록 편향되게 모델을 유도하는 경향 발생
    - 해당 테스트 데이터에만 과접합되는 학습 모델이 만들어져 다른 테스트용 데이터가 유입될 경우 성능 저하 야기
    - 해결책 : 교차 검증을 적용해서 다양한 학습과 평가를 수행
    - 예시 : 본 고사가 테스트 데이터 셋에 대한 평가하는 거라면 모의고사는 교차 검증에서 많은 학습과 검증 세트에서 알고리즘 학습과 평가를 수행하는 것

Scikit-Learn의 주요 API - sklearn.model_selection 모듈

- 교차 검증
  - ML 모델의 성능 평가는 교차 검증 기반으로 1차 평가를 한 뒤에 최종적으로 테스트 데이터 셋에 적용해 평가하는 프로세스

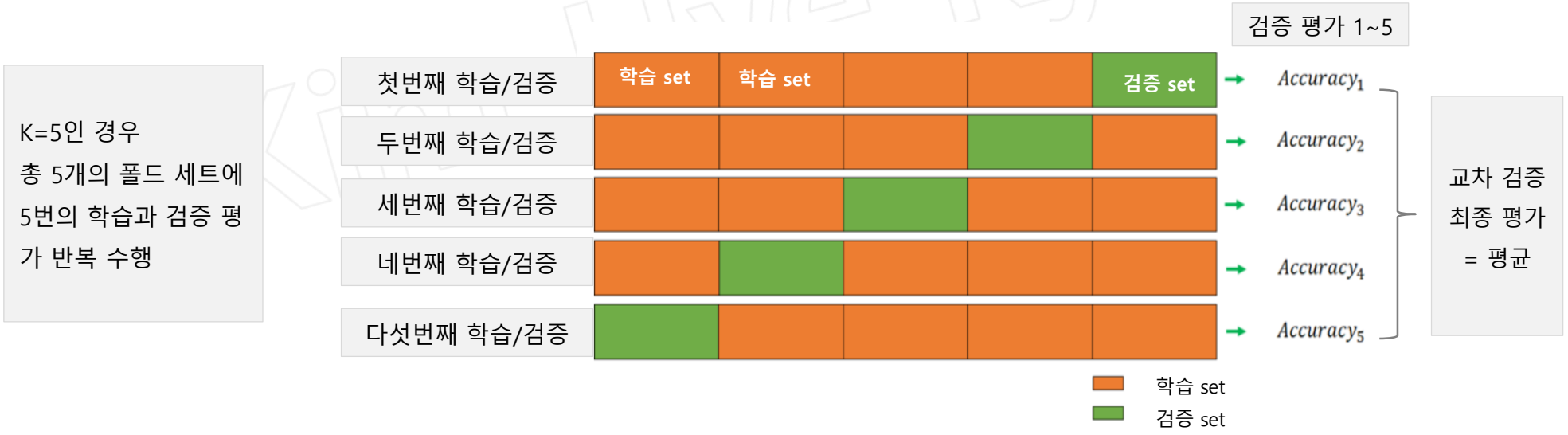


학습 데이터 분할
학습 데이터와 학습된 모들의 성능을 1차 평가하는 검증 데이터로 구분

모든 학습과 검증 과정이 완료된 후
최종적으로 성능 평가를 위한 데이터 셋트

Scikit-Learn의 주요 API - sklearn.model_selection 모듈

- 교차 검증
  - K 폴드 교차 검증
    - 가장 보편적으로 사용되는 교차 검증 기법
    - K 개의 데이터 폴드 셋트를 만들어서 K 번만큼 각 폴드 셋트에 학습과 검증 평가를 반복 수행



Scikit-Learn의 주요 API - sklearn.model_selection 모듈

- 교차 검증과 최적 하이퍼 파라미터 튜닝을 한번에
- GridSearchCV
  - 분류(Classifier)와 회귀(Regressor)와 같은 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터 도출
  - 교차 검증을 기반으로 하이퍼파라미터의 최적값을 찾음
  - 생성자의 parameter

하이퍼 파라미터

머신러닝 알고리즘을 구성하는 주요 요소

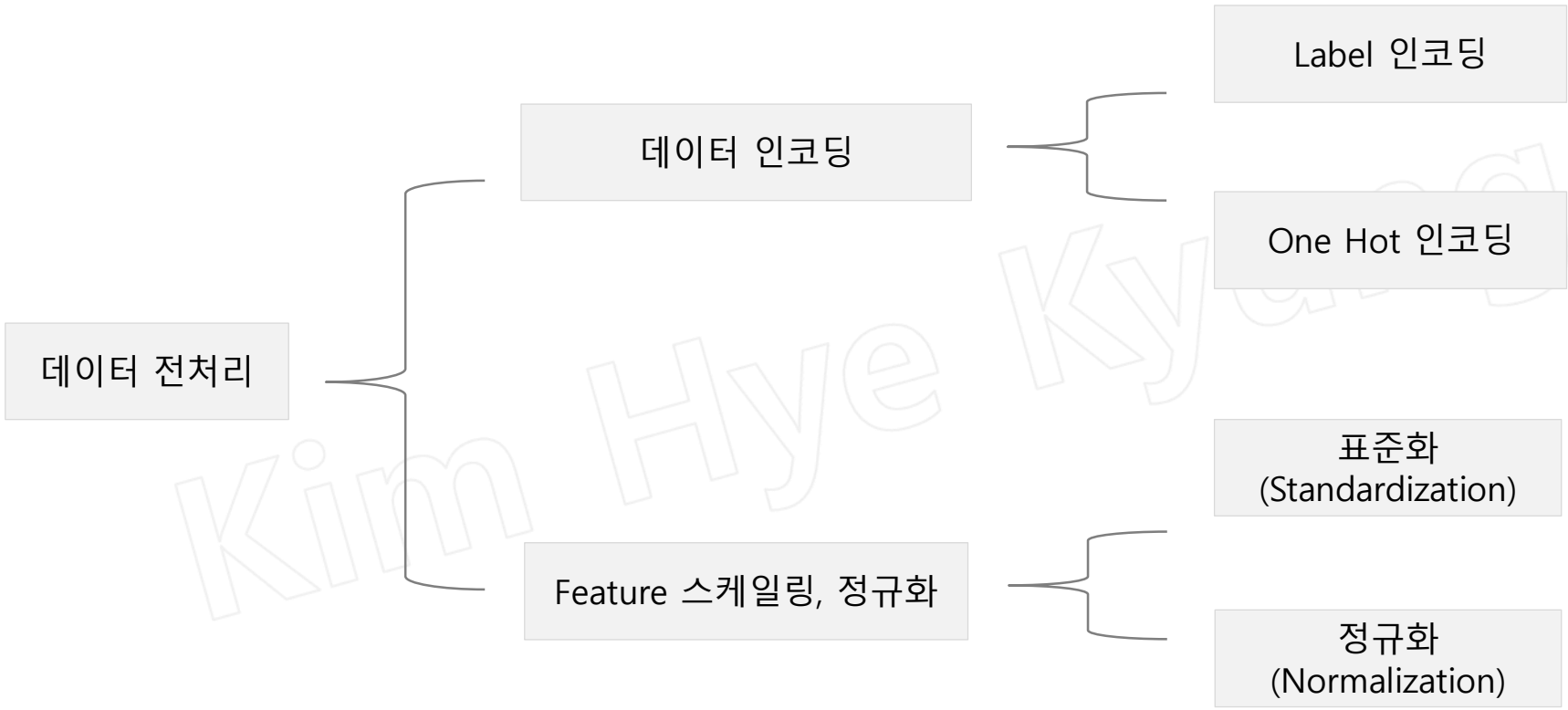
이 값을 조정해 알고리즘의 예측 성능을 개선할 수 있음

```
GridSearchCV(dtree, param_grid=parameters, cv=3, refit=True)
```

parameter	특징
estimator	검증기
param_grid	key+list 값을 보유한 dict estimator의 튜닝을 위해 파라미터명과 사용될 여러 파라미터값 지정
cv	교차 검증을 위해 분할되는 학습/테스트 세트의 개수를 지정
refit	default=True, 가장 최적의 하이퍼파라미터를 찾은 뒤 입력된 estimator 객체를 해당 하이퍼파라미터로 재학습

데이터 전처리

데이터 전처리



데이터 인코딩

Data Preprocessing은 ML 알고리즘만큼 중요

결손값 불허용

문자열 값을 입력 값으로 허용하지 않음
인코딩해서 숫자 형으로 변환 해야 함

데이터 인코딩

- 결손값 불허용
  - NaN, Null -> 고정된 다른 값으로 변환해야 함
  - feature에 포함된 Null 값 비율에 따른 처리
    - 소량 : 값이 얼마 되지 않는다면 피처의 평균값 등으로 간단히 대체
    - 다수 : 해당 feature는 drop 권장
    - 발생 가능한 경우의 수
      - 중요도가 높은 feature이고 Null을 단순히 feature의 평균값으로 대체 할 경우 예측 왜곡이 심한 현상 발생
    - 고려사항
      - 업무 로직 등을 상세히 검토 후 정밀한 대체 값으로 선정

데이터 인코딩

- 문자열 값을 입력 값으로 허용하지 않음
  - 문자열
    - 카테고리형 featur와 텍스트형 feature 의미
  - 카테고리형
    - 코드 값으로 표현 권장
  - 텍스트형
    - 픽처 벡터화등의 기법으로 벡터화 또는 불필요한 경우 삭제 권장

데이터 인코딩

- 머신러닝의 대표적인 인코딩 방식

Label 인코딩

카테고리 피처를 코드형 숫자값으로 변환

주의사항
: 숫자값으로 변환 따라서 "크다, 작다" 개념이 적용되어 큰 숫자에 가중치가 더 부여 될 수도 있음 따라서 선형회귀 같은 ML 알고리즘엔 부적합

숫자의 특성이 반영되지 않는 트리 계열의 ML 알고리즘에 반영

One Hot 인코딩

피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1 표시, 나머지는 0을 표시하는 방식

데이터 인코딩

- 문자열 인코딩 방식 – Label 인코딩

Label 인코딩

```
: 1  from sklearn.preprocessing import LabelEncoder
  2
  3  items=['구글','삼성','네이버','카카오','애플', '애플', 'MS','MS']
  4
  5  # LabelEncoder를 객체로 생성
  6  encoder = LabelEncoder()
  7
  8  # fit( ) 과 transform( ) 으로 label 인코딩 수행
  9  encoder.fit(items)
 10  labels = encoder.transform(items)
 11
 12  print('인코딩 변환값:',labels)
```

```
인코딩 변환값: [1 3 2 5 4 4 0 0]
```

```
: 1  print('인코딩 클래스:', encoder.classes_)
```

```
인코딩 클래스: ['MS' '구글' '네이버' '삼성' '애플' '카카오']
```

```
: 1  print('디코딩 원본 값:',encoder.inverse_transform([4, 5, 2, 0, 1, 1, 3, 3]))
```

```
디코딩 원본 값: ['애플' '카카오' '네이버' 'MS' '구글' '구글' '삼성' '삼성']
```

데이터 인코딩

- 문자열 인코딩 방식 – One Hot 인코딩

One Hot 인코딩

```
from sklearn.preprocessing import OneHotEncoder
import numpy as np

items=['구글','삼성','네이버','카카오','애플', '애플', 'MS','MS']

# 먼저 숫자값으로 변환을 위해 LabelEncoder로 변환
encoder = LabelEncoder()
encoder.fit(items)
labels = encoder.transform(items)

# 2차원 데이터로 변환
labels = labels.reshape(-1, 1)
print(labels)

# 원-핫 인코딩을 적용
oh_encoder = OneHotEncoder()
oh_encoder.fit(labels)
oh_labels = oh_encoder.transform(labels)
print('원-핫 인코딩 데이터')
print(oh_labels.toarray())
print('원-핫 인코딩 데이터 차원')
print(oh_labels.shape)
```

```
[[1]
 [3]
 [2]
 [5]
 [4]
 [4]
 [0]
 [0]]
원-핫 인코딩 데이터
[[0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1. 0.]
 [1. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0.]]
원-핫 인코딩 데이터 차원
(8, 6)
```

데이터 인코딩

- 문자열 인코딩 방식 – One Hot 인코딩
  - Pandas 지원 API
    - get_dummies() : 숫자형 값으로 변환 없이 바로 변환 가능

pandas에서 원핫 인코딩 적용을 지원하는 API - get_dummies()

```
1  import pandas as pd
2
3  df = pd.DataFrame({'item' : ['구글','삼성','네이버','카카오','애플', '애플', 'MS','MS'] })
4  pd.get_dummies(df)
```

	item_MS	item_구글	item_네이버	item_삼성	item_애플	item_카카오
0	0	1	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	0	0
3	0	0	0	0	0	1
4	0	0	0	0	1	0
5	0	0	0	0	1	0
6	1	0	0	0	0	0
7	1	0	0	0	0	0

Feature 스케일링과 정규화

- Feature Scaling이란?
  - 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
- 종류
  - 표준화(Standardization)
    - 데이터의 피처 각각이 평균이 0, 분산이 1인 가우시안 정규 분포를 가진 값으로 변환
    - SVM, 선형회귀(Linear Rregression), 로지스틱 회귀(Logistic Regression)
      - **데이터가 가우시안 분포를 가지고 있다고 가정하고 구현되어 있기 때문에 사전에 표준화를 적용하는 것은 예측 성능 향상에 중요한 요소**
  - 정규화(Normalization)
    - 서로 다른 피처의 크기를 통일하기 위해 크기를 변환해주는 개념

Feature 스케일링과 정규화 : 표준화(Standardization)

- 표준화(Standardization) 주요 API
  - StandardScaler
    - 표준화를 지원 클래스
    - 개별 피처를 평균이 0, 분산이 1인 값으로 변환

```python
from sklearn.preprocessing import StandardScaler

# StandardScaler객체 생성
scaler = StandardScaler()

# fit( ) 과 transform( )으로 표준화
scaler.fit(iris_df)
iris_scaled = scaler.transform(iris_df)

# 데이터 셋이 numpy ndarry로 반환되어, DataFrame으로 변환
iris_df_scaled = pd.DataFrame(data=iris_scaled, columns=iris.feature_names)

print('feature 들의 평균 값')
print(iris_df_scaled.mean())

print('₩nfeature 들의 분산 값')
print(iris_df_scaled.var())
```

```
feature 들의 평균 값
sepal length (cm)   -1.690315e-15
sepal width (cm)    -1.842970e-15
petal length (cm)   -1.698641e-15
petal width (cm)    -1.409243e-15
dtype: float64

feature 들의 분산 값
sepal length (cm)    1.006711
sepal width (cm)     1.006711
petal length (cm)    1.006711
petal width (cm)     1.006711
dtype: float64
```

평균이 0에 가까운 값으로,
분산이 1에 가까운 값으로 변환

Feature 스케일링과 정규화 : 정규화(Normalization)

- 정규화(Normalization)
  - MinMaxScaler
    - 데이터값을 0과 1사이의 범위값으로 변환
    - 음수 값이 있으면 -1~1 값으로 변환
    - 데이터 분포가 가우시안 분포가 아닐 경우 Min, Max Scale 적용

```
1  from sklearn.preprocessing import MinMaxScaler
2
3  # MinMaxScaler객체 생성
4  scaler = MinMaxScaler()
5
6  # MinMaxScaler 로 데이터 셋 변환
7  scaler.fit(iris_df)
8  iris_scaled = scaler.transform(iris_df)
9
10 # 변환된 데이터 셋이 numpy ndarry로 반환되어 이를 DataFrame으로 변환
11 iris_df_scaled = pd.DataFrame(data=iris_scaled, columns=iris.feature_names)
12
13 print('feature들의 최소 값')
14 print(iris_df_scaled.min())
15 print('₩nfeature들의 최대 값')
16 print(iris_df_scaled.max())
17
```

```
feature들의 최소 값
sepal length (cm)    0.0
sepal width (cm)     0.0
petal length (cm)    0.0
petal width (cm)     0.0
dtype: float64

feature들의 최대 값
sepal length (cm)    1.0
sepal width (cm)     1.0
petal length (cm)    1.0
petal width (cm)     1.0
dtype: float64
```

평가

정확도

- 정확도란?
  - 실제 데이터에서 예측 데이터가 얼마나 같은지 판단하는 지표

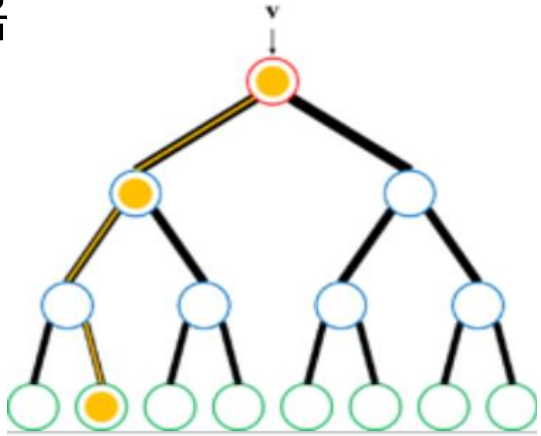
$$\text{정확도(Accuracy)} = \frac{\text{예측 결과가 동일한 데이터 건수}}{\text{전체 예측 데이터 건수}}$$

  - 직관적으로 모델 예측 성능을 나타내는 평가 지표
  - 이진 분류 – 데이터의 구성에 따라 ML 모델의 성능을 왜곡 할수 있기 때문에 정확도 수치 하나만으로 성능을 평가하지는 않음
-

랜덤 포레스트

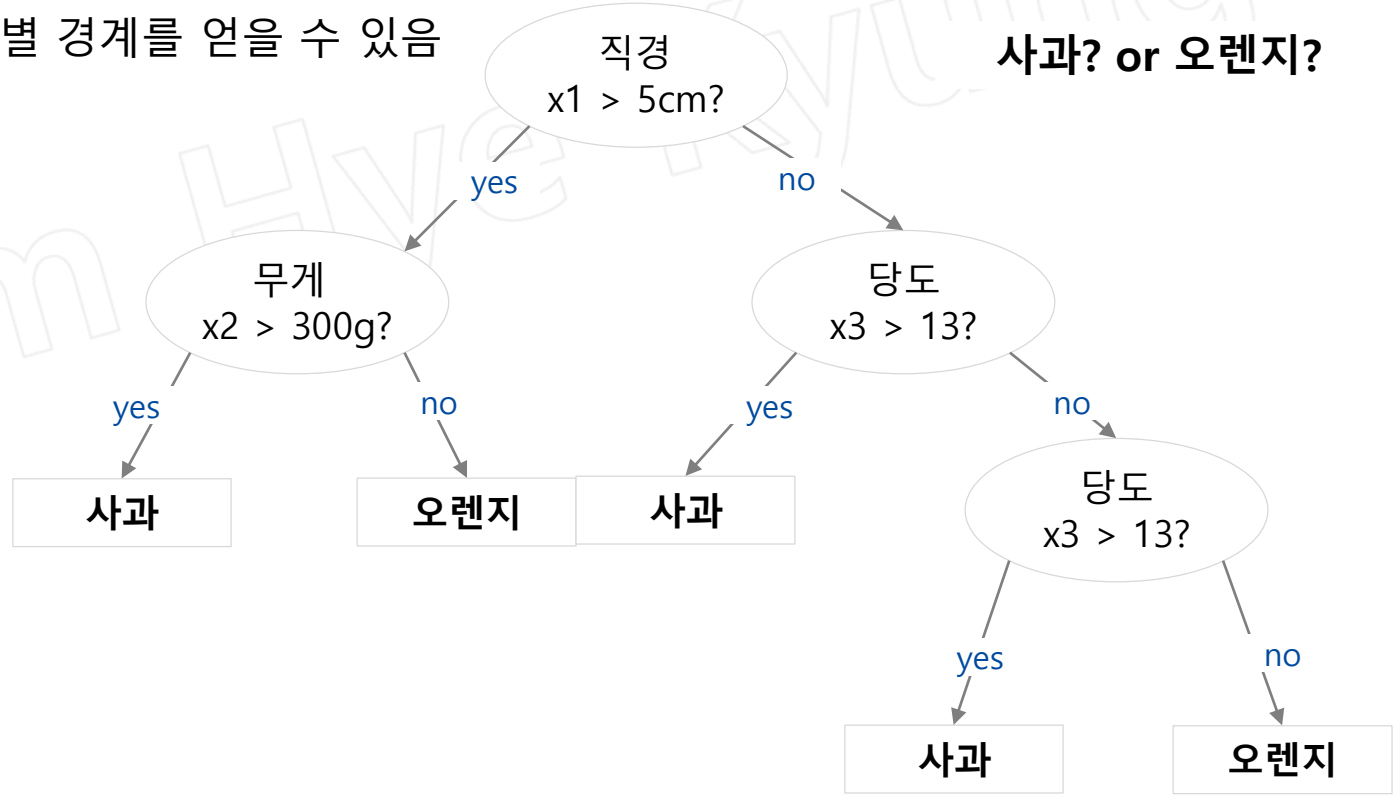
랜덤 포레스트 – 분류기(학습기)

- 분류, 회귀 분석 등에 사용되는 앙상블 학습(ensemble learning) 방법의 일종
  - 학습기를 여러 개 조합해서 전체의 성능을 끌어 올리는 방법
- 이때 결정 트리(Decision Tree)라는 학습기를 사용
- 특징을 무작위로 골라 만든 결정 tree를 여러 개 뭉쳐 사용하기 때문에 랜덤 포레스트(숲)이라는 이름이 붙음


결정 트리 테스트 단계

랜덤 포레스트 – 분류기(학습기)

- 결정 트리란?
  - 단순한 식별 규칙을 여러 개 조합해 놓은 식별기
  - 질문을 여러 개 모아 놓은 것으로 간주
    - 비선형 식별 경계를 얻을 수 있음



랜덤 포레스트 – 분류기(학습기)

- 랜덤 포레스트의 장점
  - 딥러닝과 비교했을 때 사람이 직관적으로 이해하기 쉬움
  - 성능이 우수
    - 딥러닝이 성능이 우수하다고 하나 많은 데이터가 있는 경우에 해당
    - 데이터가 적은 경우에는 오히려 랜덤 포레스트가 더 좋은 성능을 발휘하기도 함
  - 클래스가 많은 경우 사용시 성능이 좋지 않음
  - classification과 regression이 모두 가능
- 사용 방법

```
clf = RandomForestClassifier()          # 랜덤포레스트 학습하기
clf.fit(data_train, label_train)        # 학습
clf.predict(data_test)                  # 예측
```

선형 회귀

선형 회귀란?

- 회귀분석
  - 결정론적 모형(deterministic Model)
  - 확률적 모형(probabilistic Model)
- 결정론적 회귀분석 모형
  - 독립 변수 x에 대해 대응하는 종속 변수 y와 가장 비슷한 값 y^를 출력하는 함수 H(x)를 찾는 과정

- 선형 회귀분석(linear regression analysis) 이란?
  - 독립 변수 x와 이에 대응하는 종속 변수 y간의 관계가 다음과 같은 선형 함수 H(x) 의미 $\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D = w_0 + w^T x$

선형 회귀

- 딥러닝은 자그마한 통계의 결과들이 무수히 얽히고 설켜 이루어지는 복잡한 연산의 결정체
- 가장 훌륭한 예측선 긋기 – 선형 회귀(linear regression)
- Machine Learning – 제대로 된 선을 긋는 작업부터 시작

선형 회귀란?

독립 변수 x 를 사용해서 종속 변수 y 의 움직임을 예측하고 설명하는 작업

선형 회귀

- 정의

학생들의 중간 고사 성적이 다 다르다

학생들의 중간 고사 성적이 [ ] 에 따라 다 다르다

- [ ] 에 들어갈 내용 = 정보

- Machine Learning과 딥러닝은 이 정보가 필요
- 정보를 정확히 준비해 놓으면 성적 예측이 가능

선형 회귀

- 수학적인 관점에서의 정보

학생들의 중간 고사 성적이 [ ] 에 따라 다 다르다

- 성적을 변하게 하는 '정보' 요소 = x
- x 값에 의해 변하는 '성적' = y
- x 값이 변함에 따라 y 값도 변함
  - x
    - 독립 변수
    - 독립적으로 변할 수 있는 값을 x를 독립 변수
  - y
    - 종속 변수
    - 독립 변수에 따라 종속적으로 변하는 변수

선형 회귀란?

독립 변수 x 를 사용해서
종속 변수 y 의 움직임을 예측하고
설명하는 작업

선형 회귀

선형 회귀란?

독립 변수 x 를 사용해서 종속 변수 y 의 움직임을 예측하고 설명하는 작업

단순 선형 회귀
(simple linear regression)

하나의 x 값만으로도 y 값을 설명할 수 있는 경우

다중 선형 회귀
(multiple linear regression)

x 값이 여러 개 필요한 경우

선형 회귀

- 독립 변수가 하나뿐인 단순 선형 회귀의 예
  - 성적을 결정하는 여러 요소 중에 '공부한 시간' 한 가지만 놓고 예측하는 경우

  - x : 공부한 시간
  - y : 성적
  - 집합 x와 집합 y의 표

공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

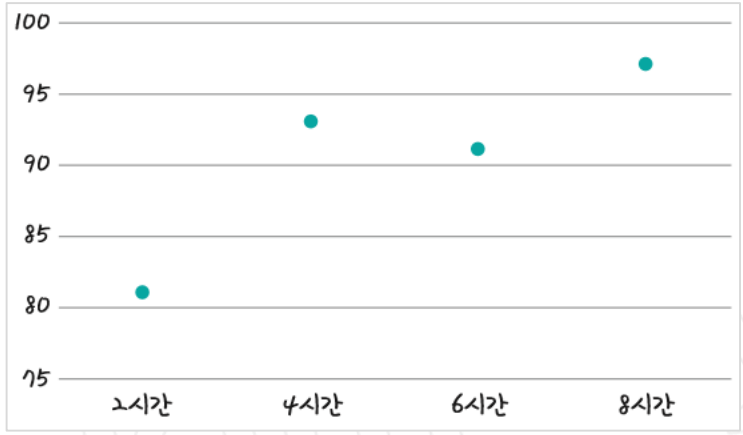
- 공부한 시간을 x라 하고 성적을 y라 할 때 집합 x와 집합 y를 다음과 같이 표현할 수 있음

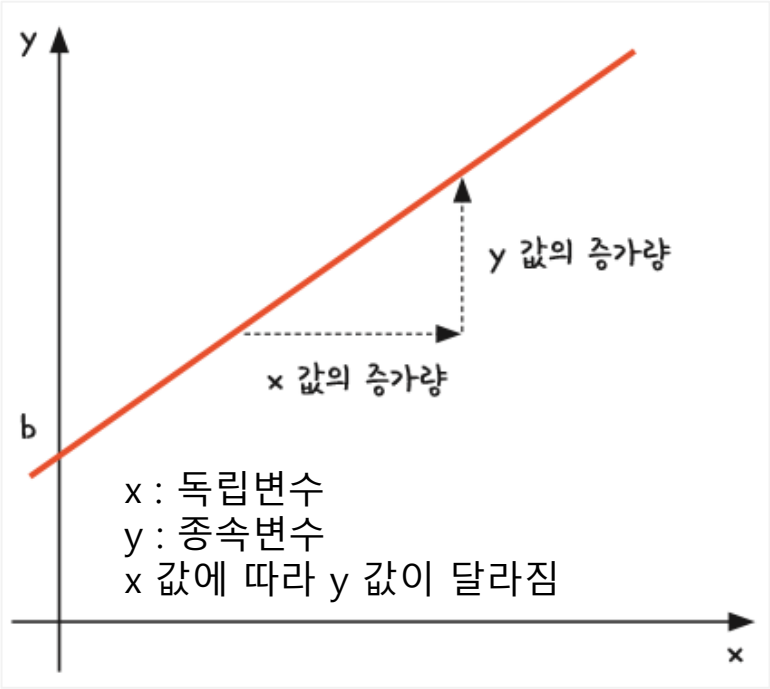
$$x = \{2,\ 4,\ 6,\ 8\}$$
$$y = \{81,\ 93,\ 91,\ 97\}$$

선형 회귀

- 공부한 시간과 성적을 좌표로 표현

$$x = \{2,\ 4,\ 6,\ 8\}$$
$$y = \{81,\ 93,\ 91,\ 97\}$$





선형 회귀

이 점들의 특징을 가장 잘 나타내는 선을 그리는 과정과 일치

선 모양 – 직선 즉 일차함수 $y = ax + b$
a – 기울기
b – y 축을 지나는 값인 y의 절편

$$\frac{y\text{값의 증가량}}{x\text{값의 증가량}}$$

선형 회귀

- 직선을 훌륭하게 그으려면
  - 직선의 기울기 a 값과 y 절편 b 값을 정확히 예측해 내야 함

- 선형 회귀는 곧 정확한 직선을 그려내는 과정
  - 최적의 a 값과 b 값을 찾아내는 작업

- 성적 예시 관점에서
  - 학생의 성적을 예측하고 싶을때, 정확한 직선을 그어 놓았다면 이 학생이 몇 시간을 공부했는지만 물어보면 됨
  - 정확한 a와 b의 값을 따라 움직이는 직선에 학생이 공부한 시간인 x 값을 대입하면 예측 성적인 y 값을 구할 수 있는 것

선형 회귀

- 정확한 기울기 a와 정확한 y 절편의 값 b를 알아내는 간단한 방법
  - 최소 제곱법 (method of least squares)

- 최소 제곱법
  - 회귀 분석에서 사용되는 표준 방식
  - 실험이나 관찰을 통해서 얻은 데이터를 분석하여 미지의 상수를 구할 때 사용하는 공식
  - 일차 함수의 기울기 a와 y 절편 b를 바로 구할 수 있음

선형 회귀

- 최소 제곱법
  - 지금 가진 정보가 x 값(입력 값, '공부한 시간')과 y 값(출력 값, '성적')일 때 최소 제곱법을 이용해 기울기 a를 구하는 방법
    - 각 x와 y의 편차를 곱해서 이를 합한 값을 구함
    - 그리고 이를 x 편차 제곱의 합으로 나눔

$$a = \frac{(x - x\text{ 평균})(y - y\text{ 평균})\text{의 합}}{(x - x\text{ 평균})\text{의 합의 제곱}}$$

- 식으로 표현시

$$a = \frac{\sum_{i=1}^{n}(x - mean(x))(y - mean(y))}{\sum_{i=1}^{n}(x - mean(x))^2}$$

선형 회귀

- 성적 예측과 최소 제곱법

공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

- 성적(y)과 공부한 시간(x)을 가지고 최소 제곱법을 이용해 기울기 a를 구하려면
  - x 값의 평균과 y 값의 평균을 각각 구함
    - 공부한 시간(x) 평균: (2 + 4 + 6 + 8) ÷ 4 = 5
    - 성적(y) 평균: (81+ 93 + 91 + 97) ÷ 4 = 90.5
  - 식에 대입

$$a = \frac{(2-5)(81-90.5)+(4-5)(93-90.5)+(6-5)(91-90.5)+(8-5)(97-90.5)}{(2-5)^2+(4-5)^2+(6-5)^2+(8-5)^2}$$

$$= \frac{46}{20}$$

$$= 2.3$$

기울기는 2.3!

선형 회귀

- 다음은 y 절편인 b를 구하는 공식

$$b = y\text{의 평균} - (x\text{의 평균} \times \text{기울기 } a)$$

- y의 평균에서 x의 평균과 기울기의 곱을 빼면 b의 값이 나온다는 의미
  - 식으로 표현하면 다음과 같음

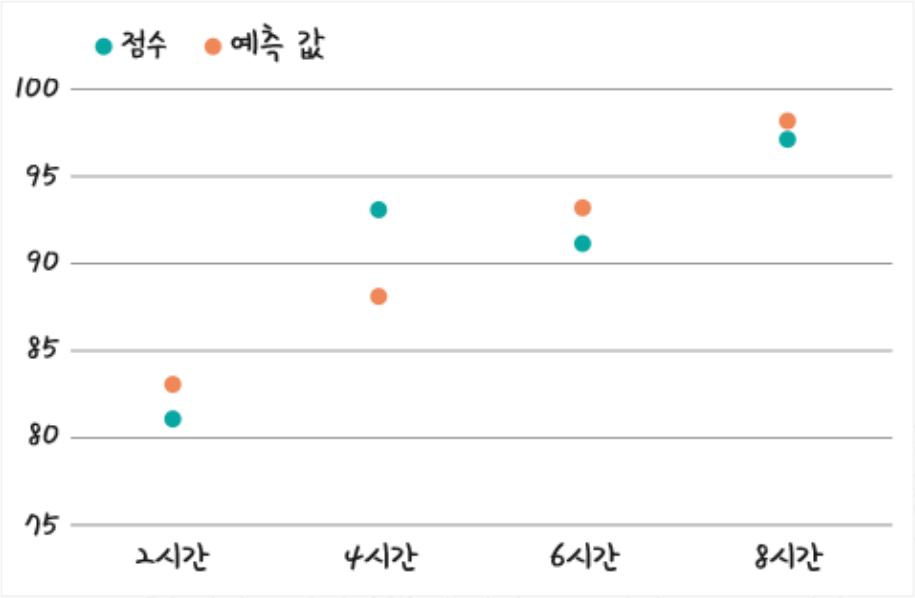
$$b = mean(y) - (mean(x) * a)$$

$$b = 90.5 - (2.3 \times 5)$$
$$= 79$$

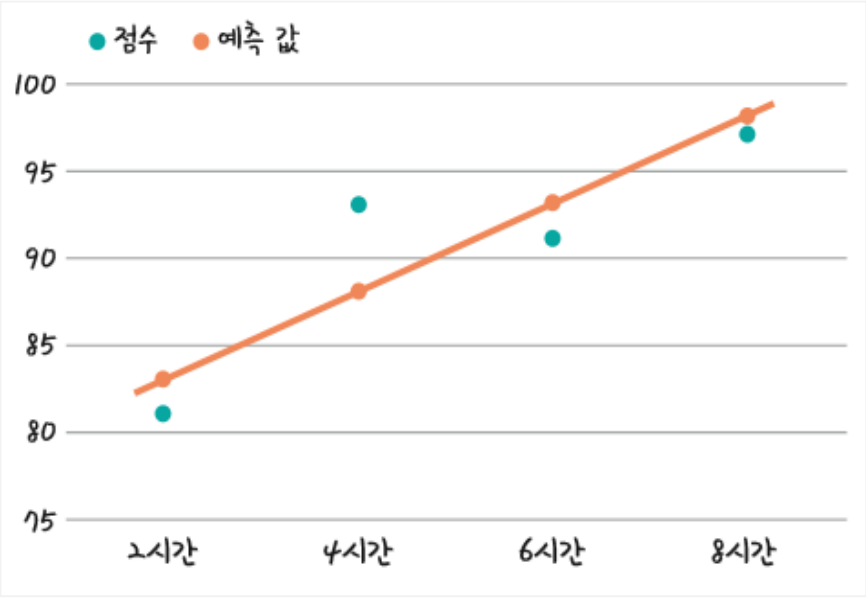
공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점
예측값	83.6	88.2	92.8	97.4

- $y = 2.3x + 79$ 위한 직선의 방정식이 완성

선형 회귀



공부한 시간, 성적, 예측 값을 좌표로 표현



오차가 최저가 되는 직선의 완성

- 오차가 가장 적은, 주어진 좌표의 특성을 가장 잘 나타내는 직선
  - 우리가 원하는 예측 직선
- 이 직선에 다른 x 값(공부한 시간)을 넣고 '공부량에 따른 성적을 예측'할 수 있음

K-평균 군집화

군집(Clustering)이란?

- 군집 (Clustering) :

비슷한 특징을 가지는 데이터 인스턴스들끼리 그룹화

레이블 없는 데이터에 대한 레이블 추론

K-평균 군집화

- K-mean 알고리즘
  - 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작
  - **가장 간단하고 널리 알려진 군집 알고리즘**
  - **클러스터의 중심을 찾는 알고리즘**
  - 데이터 포인트를 가장 가까운 클러스터 중심에 할당하고, 거리 평균으로 클러스터 중심을 다시 지정
    - 군집 중앙 : 해당 군집에 속하는 모든 점의 산술 평균
    - 각 점은 다른 군집의 중앙보다 자신이 속한 군집의 중앙에 더 가까움
  - 데이터가 커지고 복잡해 질 수록 유용한 정보 추출

```
pip  install  mglearn
```

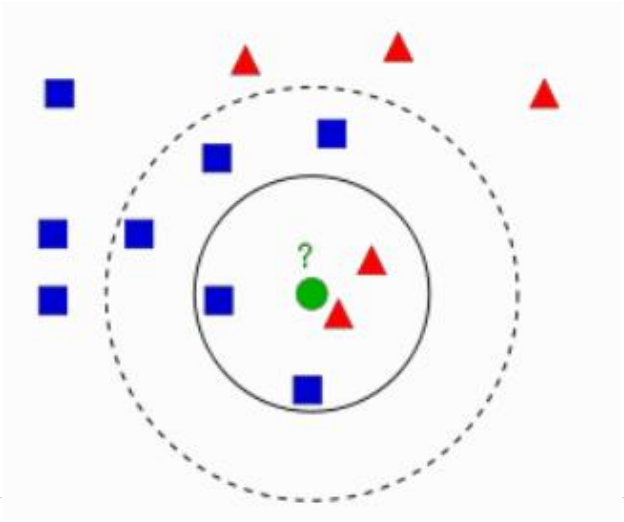
k-Nearest Neighbour(kNN)

- kNN은 지도학습 중 단순한 알고리즘을 이용한 방법
- https://opencv-python-tutroals 을 활용한 이해

문제

삼각형과 사각형이 있는 공간

이 공간에 가운데 초록색 원이 있는 경우, 이 원은 삼각형일까? 사각형일까?

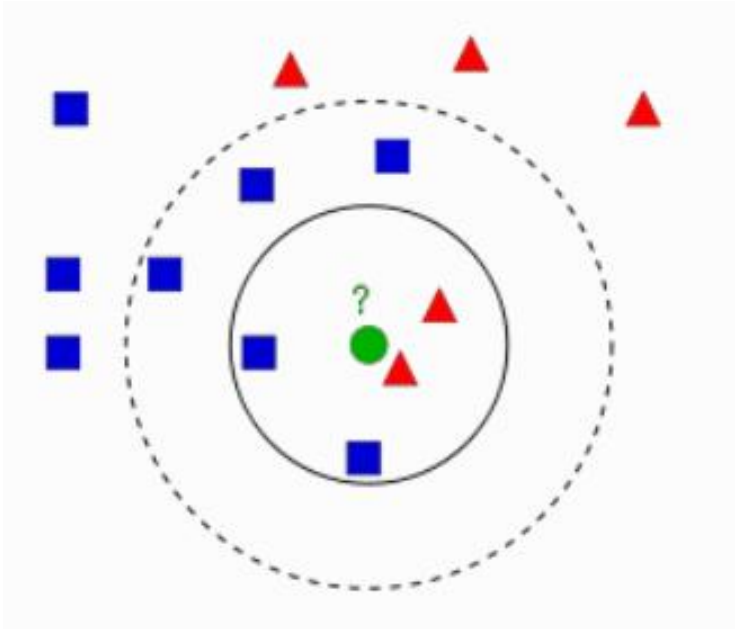


https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html#knn-understanding

k-Nearest Neighbour(kNN)

- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것



그렇다면 초록색 원은

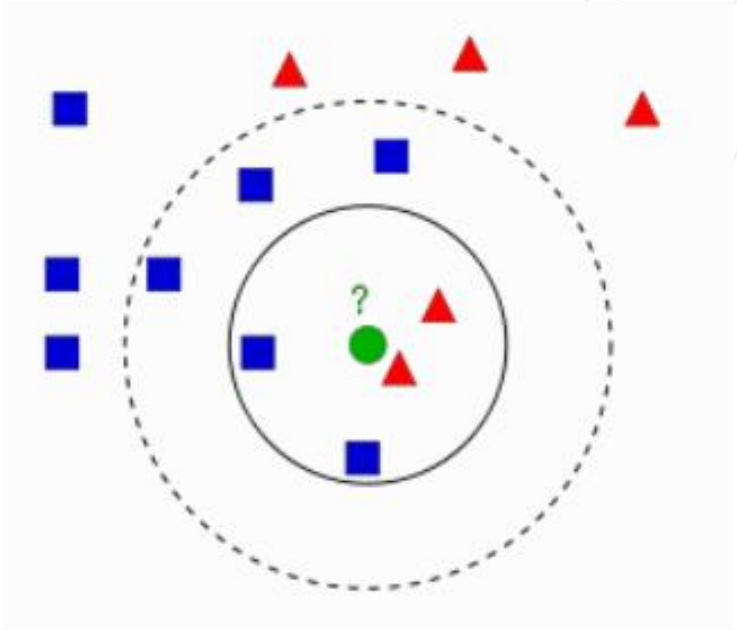
빨간 삼각형?
파란 사각형?

k-Nearest Neighbour(kNN)

- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음



그렇다면 초록색 원은

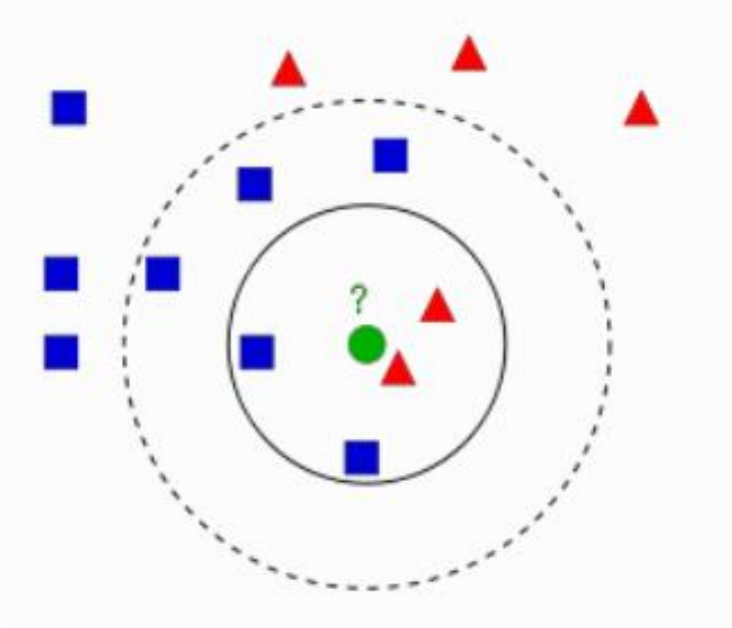
빨간 삼각형으로 추정

k-Nearest Neighbour(kNN)

- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음



그렇다면 초록색 원은

빨간 삼각형으로 추정

좀더 범위를 넓혀보면???

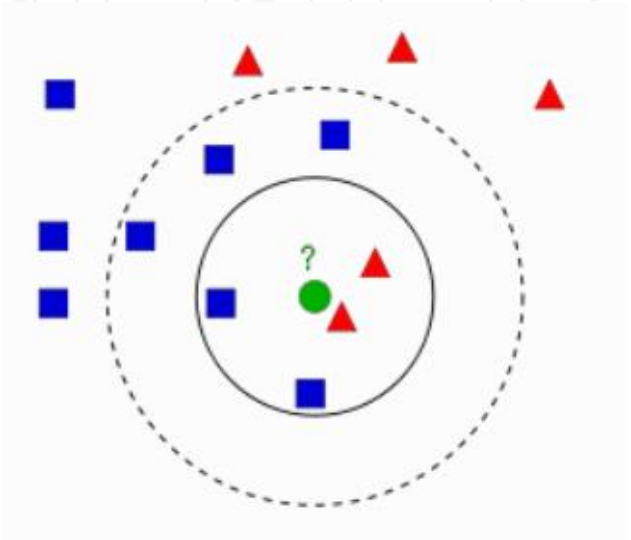
k-Nearest Neighbour(kNN)

- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음

  3단계 - 좀더 범위를 넓혀보면 오히려 파란색 점이 많이 있음



그렇다면 초록색 원은

빨간 삼각형으로 추정

좀더 범위를 넓혀보면???

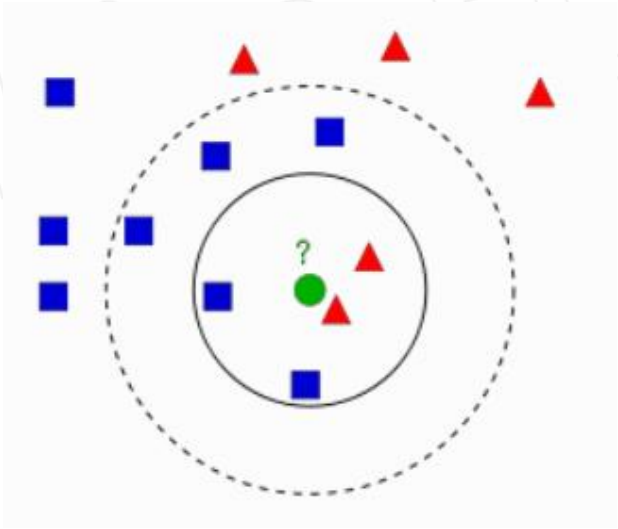
k-Nearest Neighbour(kNN)

- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음

  3단계 - 좀더 범위를 넓혀보면 오히려 파란색 점이 많이 있음



그렇다면 초록색 원은

빨간 삼각형으로 추정

좀더 범위를 넓혀보면 파란색 사각형으로 추정

범위를 몇 단계까지 넓혀 판단할 것인지?

k-Nearest Neighbour(kNN)

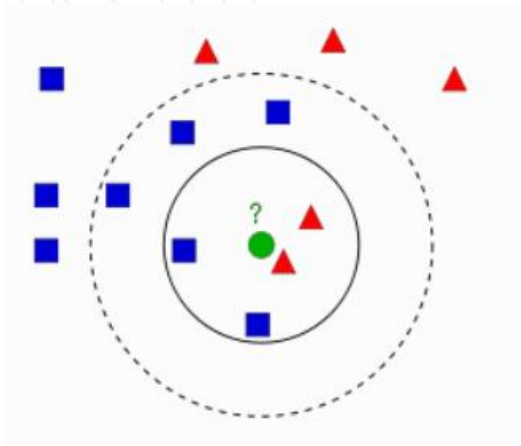
- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음

  3단계 - 좀더 범위를 넓혀보면 오히려 파란색 점이 많이 있음

  4단계 - 범위를 몇 단계까지 넓혀 판단할 것인지 결정하게 되는데 이때 넓히는 단계를 k값으로 결정



그렇다면 초록색 원은

빨간 삼각형으로 추정

좀더 범위를 넓혀보면 파란색 사각형으로 추정

범위를 몇 단계까지 넓혀 판단할 것인지?

k-Nearest Neighbour(kNN)

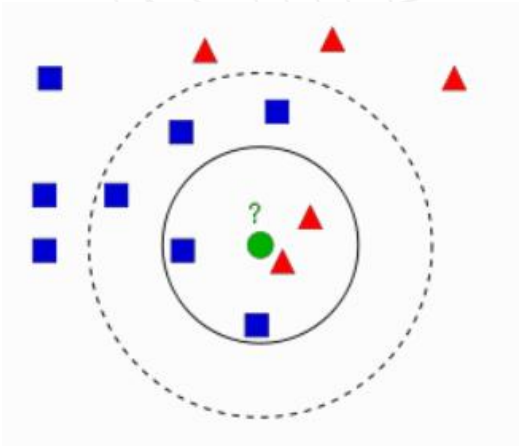
- 문제 풀이

  1단계 - 가장 가까운 점을 찾는 것

  2단계 - 이미지에서 보면 빨간색이 가까이에 있으니 초록색 원은 빨간색으로 판단할 수도 있음

  3단계 - 좀더 범위를 넓혀보면 오히려 파란색 점이 많이 있음

  4단계 - 범위를 몇 단계까지 넓혀 판단할 것인지 결정하게 되는데 이때 넓히는 단계를 k값으로 결정



k가 3인 경우 : 빨간색 2개와 파란색 1개 이기 때문에 초록색 원은 빨간색으로 판단할 수 있음

k값 7인 경우 : 빨간색 2개와 파란색 5개가 있기 때문에 파란색으로 판단할 수 있음

또한 k값에 가중치를 줄 수 있는데, 가까운 곳에 더 많은 가중치를 두어서 판단할 수도 있음

CNN

- 합성곱 신경망 (CNN)
- 이미지 인식 분야에서 강력한 성능을 발휘
- 음성인식이나, 자연어 처리 등에서 사용
- 활용성에서도 매우 뛰어난 성과
- 일반 신경망의 경우, 이미지 데이터를 그대로 처리
- 즉, 이미지 전체를 하나의 데이터로 생각해서 입력으로 받아들이기 때문에, 이미지의 특성을 찾지 못하고 위와 같이 이미지의 위치가 조금만 달라지거나 왜곡된 경우에 올바른 성능을 내지 않음
- 그러나 합성곱 신경망(CNN)은 이미지를 하나의 데이터가 아닌, 여러 개로 분할하여 처리
- 이렇게 하면 이미지가 왜곡되더라도 이미지의 부분적 특성을 추출할 수 있어 올바른 성능을 낼 수 있음

CNN

합성곱 신경망 (CNN)

이미지 인식 분야에서 강력한 성능을 발휘
음성 인식이나, 자연어 처리 등에서 사용
활용성에서도 매우 뛰어난 성과

합성곱 신경망(CNN)은 이미지를 하나의 데이터가 아닌, 여러 개로 분할하여 처리
이렇게 하면 이미지가 왜곡되더라도 이미지의 부분적 특성을 추출할 수 있어 올바른 성능을 낼 수 있음

일반 신경망

이미지 데이터를 그대로 처리

즉, 이미지 전체를 하나의 데이터로 생각해서 입력으로 받아들이기 때문에, 이미지의 특성을 찾지 못함

이미지의 위치가 조금만 달라지거나 왜곡된 경우에 올바른 성능을 내지 않음

실습 예시 – 붓꽃 품종 예측하기

실습을 위한 참고 사항

Iris-virginica

Iris-setosa

Iris-versicolor

아이리스 관찰해 보기

- 꽃봉오리가 마치 먹물을 머금은 붓과 같다 하여 우리나라에서 '붓꽃'이라 함
- 꽃잎의 모양과 길이에 따라 여러 가지 품종으로 나뉨
- 미국 국립 과학재단에서 제공하는 Machine Learning test 데이터

실습을 위한 참고 사항

- 붓꽃 품종 예측을 위한 학습 방식
  - 분류(Classification)
    - 다중 분류(multi classification)
      - 여러 개의 답 중 하나를 고르는 분류를 의미
      - 답안이 3개
    - 참(1)과 거짓(0)으로 해결하는 것이 아니라, 여러 개 중에 어떤 것이 답인지를 예측하는 문제
  - 붓꽃 데이터 세트로 붓꽃의 품종을 분류(Classification)
    - Feature = 꽃잎의 길이와 너비, 꽃받침의 길이와 너비

속성 / 클래스

	정보 1	정보 2	정보 3	정보 4	품종
1번째 아이리스	5.1	3.5	4.0	0.2	Iris-setosa
2번째 아이리스	4.9	3.0	1.4	0.2	Iris-setosa
3번째 아이리스	4.7	3.2	1.3	0.3	Iris-setosa
...	...	...	...	...	...
150번째 아이리스	5.9	3.0	5.1	1.8	Iris-virginica

샘플

- 샘플 수: 150
- 속성 수: 4
  - 정보 1: 꽃받침 길이 (sepal length, 단위: cm)
  - 정보 2: 꽃받침 넓이 (sepal width, 단위: cm)
  - 정보 3: 꽃잎 길이 (petal length, 단위: cm)
  - 정보 4: 꽃잎 넓이 (petal width, 단위: cm)
- 클래스: Iris-setosa, Iris-versicolor, Iris-virginica

실습을 위한 참고 사항

- 붓꽃 품종 예측하기
  - 지도학습(Supervised Learning)
  - 학습을 위한 다양한 피처와 분류 결정값이 레이블 데이터로 모델을 학습 한 뒤, 별도의 테스트 데이터세트에서미지의 레이블을 예측
  - 명확한 정답이 주어진 데이터를 먼저 학습 한 뒤 미지의 정답을 예측 하는 방식

학습 데이터 세트	테스트 데이터 세트
학습을 위해주어진 데이터 세트	Machine Learning 모델의 예측 성능을 평가하기 위해 별도로 주어진 데이터 세트

  - ML 알고리즘 중 의사결정트리(Decision Tree) 알고리즘 사용
    - DecisionTreeClassifier API 사용

붓꽃 데이터 기반의 ML 분류 예측 수행 프로세스

1단계 - 데이터 세트 분리 : 데이터를 학습 데이터와 테스트 데이터로 분리

2단계 - 모델 학습 : 학습 데이터를 기반으로 ML 알고리즘을 적용해 모델을 학습

3단계 - 예측 수행 : 학습된 ML 모델을 이용해 테스트 데이터의 분류(붓꽃 종류) 예측

4단계 - 평가 : 예측된 결과값과 테스트 데이터의 실제 결과값을 비교해서 ML 모델 성능 평가

```
X_train, X_test, y_train, y_test = train_test_split(iris_data, iris_label,
                                                    test_size=0.2, random_state=11)
```

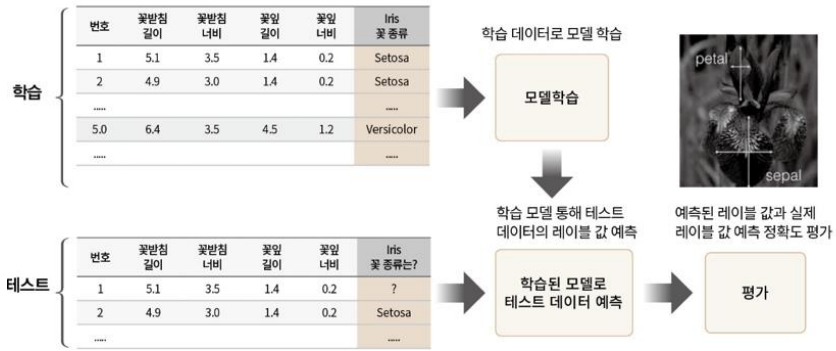
```
# DecisionTreeClassifier 객체 생성
dt_clf = DecisionTreeClassifier(random_state=11)

# 학습 수행
dt_clf.fit(X_train, y_train)
```

```
# 학습이 완료된 DecisionTreeClassifier 객체에서 테스트 데이터 세트로 예측 수행.
pred = dt_clf.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print('예측 정확도: {0:.4f}'.format(accuracy_score(y_test, pred)))
```

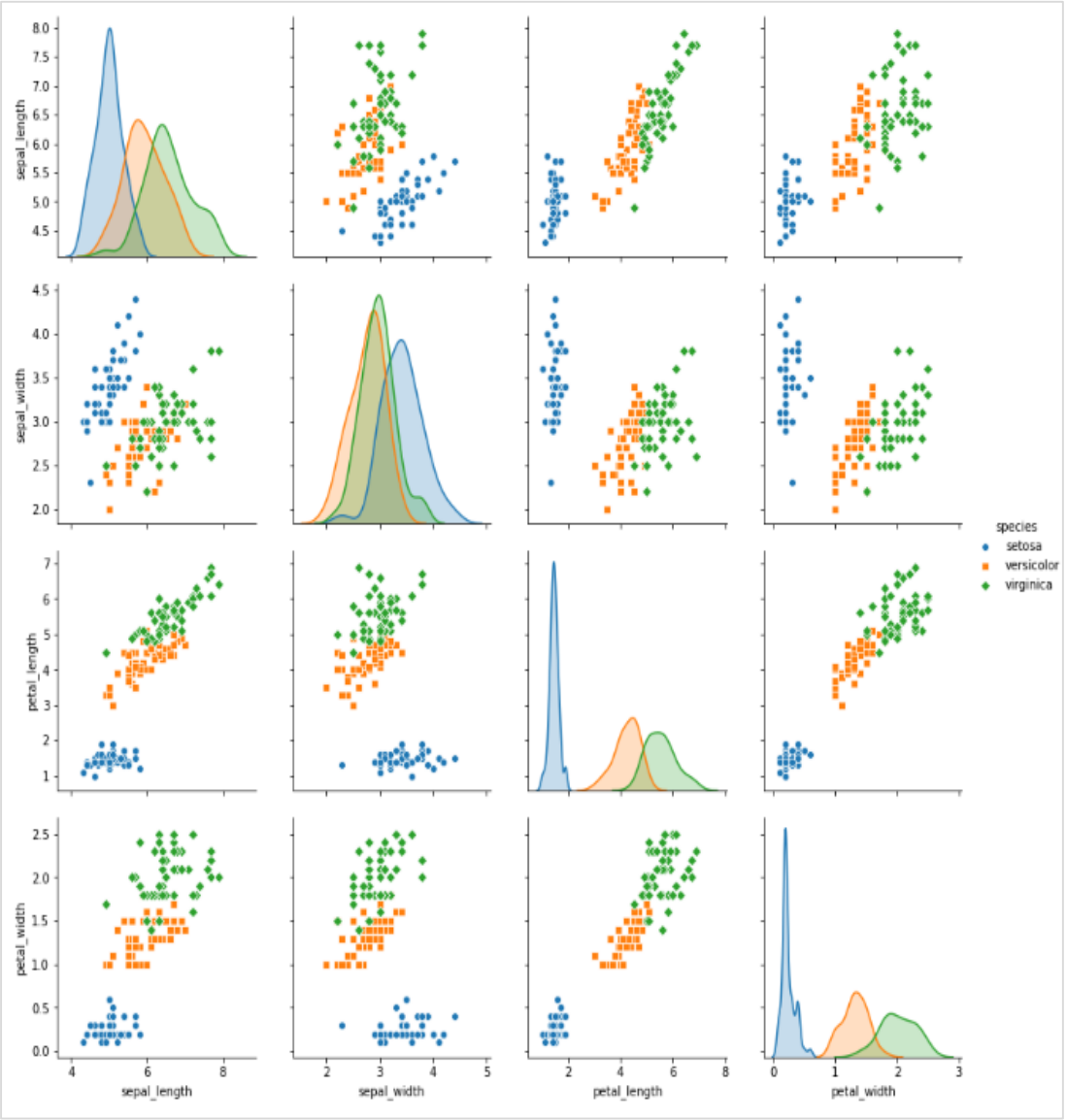
【Output】

```
예측 정확도: 0.9333
```



실습을 위한 참고 사항

- 속성
  - sepal_length - 꽃받침 길이
  - sepan_width - 꽃받침 넓이
  - petal_length - 꽃잎 길이
  - petal_width - 꽃잎 넓이



Machine Learning 용어

용 어	설 명
하이퍼 파라미터	Machine Learning 알고리즘별로 최적의 학습을 위해 직접 입력하는 파라미터들을 통칭 하이퍼 파라미터를 기반으로 알고리즘의 성능 튜닝을 할 수 있음
정확도	예측 결과가 실제 레이블 값과 얼마나 정확하게 맞는지 평가하는 지표
Feature	
기계 학습 (Machine Learning)	데이터의 규칙을 컴퓨터 스스로 찾아내는 것 의미

데이터 표현 단위

Rank	Type	Example
0	Scalar	1
1	Vector	[1, 1]
2	Matrix	[[1,1] , [1,1]]
3	3 tensor	[[[1,1], [1,1]], [[1,1], [1,1]]]
n	N tensor	...