파이썬 머신러닝분석

목차

Part 0. 개발환경 준비

1. 아나콘다(Anaconda) 배포판 설치
2. 스파이더(Spyder) 사용법

1. 아나콘다(Anaconda) 배포판 설치

- **아나콘다 배포판이란?**

1) 판다스, 넘파이, 맷플롯립 등 데이터 분석 라이브러리, Spyder 등 개발 도구(IDE)를 통합 지원.
2) 버전 관리와 패키지 업데이트가 편리.
3) 윈도우, 맥OS, 리눅스 모두 지원.

구 분	Anaconda	ActivePython	WinPython
개발자	Anaconda (미국)	ActiveState (캐나다)	WinPython 개발팀
비용	유료/무료	유료/무료	무료(오픈소스)
출시연도	2012년	2006년	2014년
운영체제	윈도우/맥/리눅스	윈도우/맥/리눅스/기타	윈도우
특징	Conda 패키지 관리 그래픽 환경(GUI)	Win32 API 지원 IDLE	WPPM 패키지 관리

[표 0-2] 파이썬 배포판의 종류

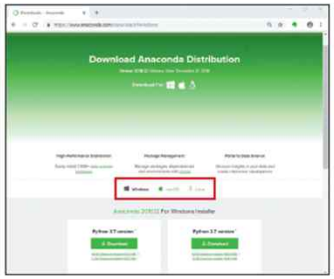
1. 아나콘다(Anaconda) 배포판 설치

1-1 아나콘다 공식 홈페이지 접속

웹브라우저 주소창에 다운로드 URL(https://www.anaconda.com/download/)을 입력한다. 또는 구글, 네이버 등 검색 엔진을 활용하여 "아나콘다 배포판 다운로드"를 검색하여 접속한다.

1-2 운영체제 선택

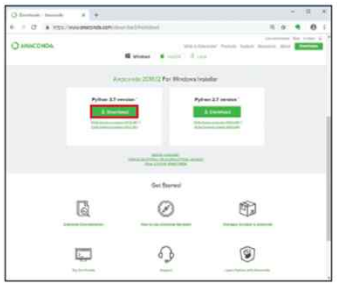
윈도우, 맥OS, 리눅스 중에서 사용 중인 PC(노트북)에 맞는 운영체제를 선택한다.



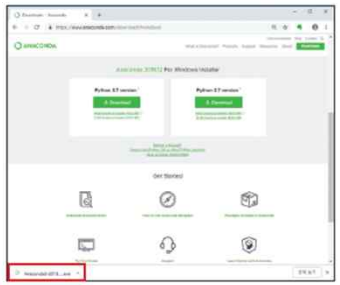
[그림 0-1] 아나콘다 다운로드 사이트
(https://www.anaconda.com/download/)

1-3 파이썬 버전 선택

화면을 아래로 스크롤하면 파이썬 버전을 선택할 수 있다. 이 책은 "Python 3" 버전을 기준으로 하기 때문에 "Python 3.* version" 문구가 있는 [Download] 버튼을 클릭한다. 사용 중인 운영체제에 맞춰 32비트/64비트 설치 파일을 구분하여 다운로드할 수 있다.



[그림 0-2] 파이썬 버전 선택



[그림 0-3] 설치 파일 다운로드

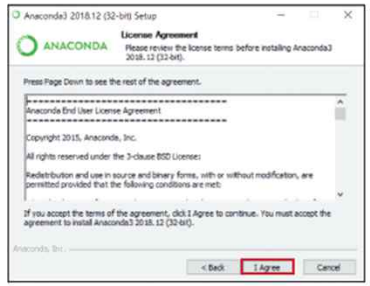
1-4 설치 파일 실행

설치 파일 다운로드가 완료되면 설치 파일을 더블클릭하여 실행한다.

❶ 설치 시작

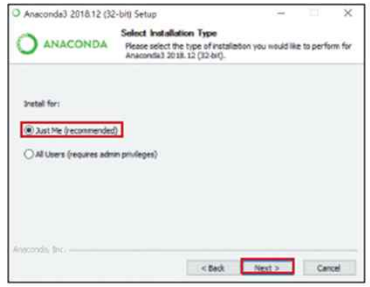
[그림 0-4] 설치 시작

❷ 이용약관 동의



[그림 0-5] 이용약관 동의

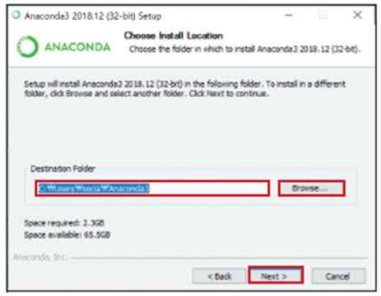
❸ 사용자 선택



[그림 0-6] 사용자 선택

1. 아나콘다(Anaconda) 배포판 설치

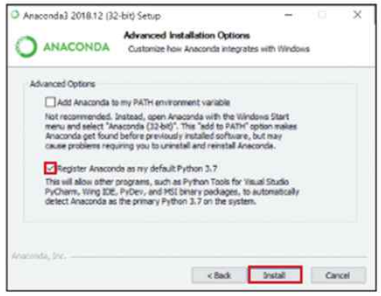
❹ 저장 경로 선택



[그림 0-7] 저장 경로 선택

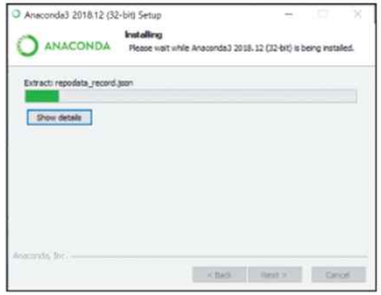
❺ 고급 옵션

설치 중에 나타나는 고급 옵션의 경우 디폴트 옵션을 그대로 사용하는 것을 권장한다.



[그림 0-8] 고급 옵션

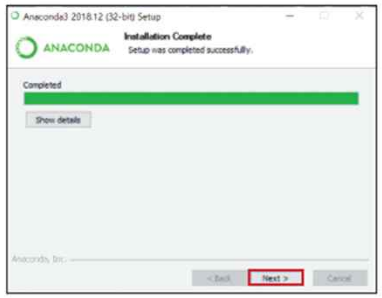
❻ 설치 진행(압축 파일 해제)



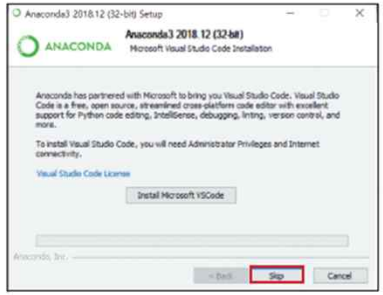
[그림 0-9] 설치 진행

❼ 설치 완료

Visual Studio 설치 여부를 묻는 화면이 나타났을 때 설치하지 않으려면 [Skip] 버튼을 누른다.



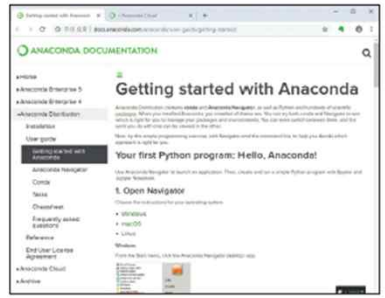
[그림 0-10] 설치 파일 압축 해제 완료



[그림 0-11] Visual Studio 설치 옵션

[그림 0-12] 아나콘다 설치 완료

아나콘다 문서(http://docs.anaconda.com/anaconda/user-guide/getting-started/) 사이트가 팝업 창에 열리는데, 필요한 내용을 찾아서 참고하기 유용하다.



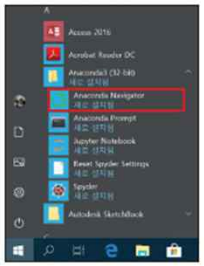
[그림 0-13] 아나콘다 문서(사용법 등)

1. 아나콘다(Anaconda) 배포판 설치

1-5 아나콘다 내비게이터 실행

❶ 윈도우 시작화면

윈도우 시작화면에 Anaconda3(32-bit/64-bit) 폴더가 설치되고, 폴더 안에 아나콘다 내비게이터(Anaconda Navigator)가 설치된 것을 확인한다.



[그림 0-14] 설치 확인

❷ 아나콘다 내비게이터 홈(Home) 화면

아나콘다 내비게이터를 실행하고, 다음의 화면이 나타나면 정상적으로 설치된 것이다.

[그림 0-15] 아나콘다 내비게이터 홈 화면

1-6 라이브러리 설치 확인

아나콘다(Anaconda) 화면 좌측의 "Environments" 메뉴를 클릭한다. 설치하면 "base(root)" 가상환경이 만들어진다. 모두 257개의 패키지(라이브러리)가 설치된 것이 확인된다.

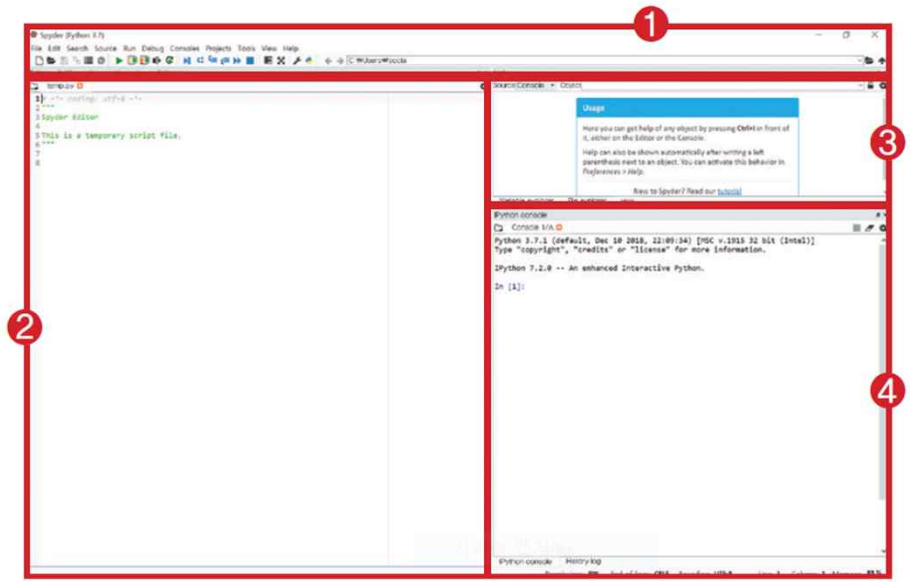
[그림 0-16] 아나콘다 가상환경 상세정보 화면

1-7 개발도구(IDE) 실행

좌측 메뉴 중에서 "Home"을 클릭하고, 실행하려는 개발도구를 찾아서 [Launch] 버튼을 클릭한다. [Launch] 버튼이 없고 [Install] 버튼만 보인다면 추가 설치가 필요한 경우이다. [Install] 버튼을 클릭하여 개발도구를 설치하고, [Launch] 버튼을 클릭하여 실행한다.

[그림 0-17] 개발도구 실행

2. 스파이더(Spyder) 사용법



[그림 0-18] 스파이더 초기 화면

❶ 주요 메뉴, 명령 버튼, 현재 파일 경로 등이 있는 부분이다.

❷ 파이썬 실행 코드를 입력하는 에디터 창을 나타낸다. 에디터 창 윗부분에 현재 편집 중인 파일명을 확인할 수 있다.

❸ 현재 파이썬 환경에서 사용 중인 변수(variable)와 저장하고 있는 데이터에 대한 정보를 제공한다. 파일 폴더를 검색하는 탭과 도움말을 검색할 수 있는 탭도 있다.

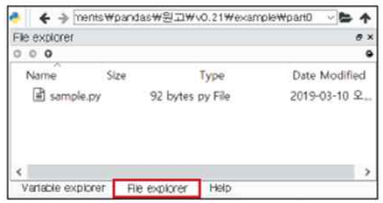
❹ IPython 콘솔(Console) 화면이다. 에디터 창에서 입력한 코드를 실행하면 이 부분에 결과가 출력된다. 콘솔에는 명령을 입력할 수 있는 프롬프트가 있어서, 파이썬 명령을 직접 입력하여 결과를 확인할 수도 있다. 이 책에서는 에디터 모드와 콘솔 모드를 필요에 따라 선택적으로 활용한다.

2. 스파이더(Spyder) 사용법

❷ 에디터 모드

[File]–[New File]을 선택하면 새로운 파일이 만들어진다. 좌측에 있는 에디터 창에 입력한 코드를 실행하려면, 코드 작성 전에 반드시 파일명을 지정하여 저장해야 한다. [File]–[Save As]를 선택하고, 폴더를 지정하여 'sample.py'라는 이름으로 저장한다. 에디터 창에 다음과 같이 코드[1]를 입력하고, 상단 메뉴에서 [File]–[Save]를 선택하거나 저장 버튼(💾)을 클릭하면 파일에 저장된다.

[그림 0-19] 폴더 검색



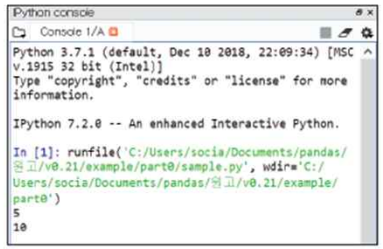
[그림 0-20] 파일 저장 위치 확인

로컬 PC에 저장되어 있는 파이썬 파일을 열어서 에디터 창에서 편집하려면, 상단 메뉴 [File]–[Open]을 선택하거나 열기 버튼(📂)을 클릭한다. 또는 주소창 옆의 폴더 검색 버튼(📂)을 눌러서 폴더를 지정하고, 'File Explorer' 탭을 클릭하면 'sample.py'라는 파일명을 찾을 수 있다. 파일명을 더블클릭하면 파일이 열리면서 에디터 창이 활성화된다.

[그림 0-21] 예제 코드 입력

예제 파일을 자료실에서 다운로드 받은 뒤 이런 방법으로 파일을 열면 에디터 창에서 편집하거나 실행해 볼 수 있다.

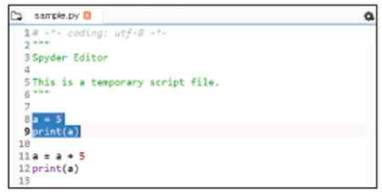
에디터 창에 입력하고 저장한 예제 코드('sample.py') 전체를 한번에 실행하려면, 상단 메뉴의 전부 실행 버튼(▶)을 클릭하거나 단축키 F5를 누른다. 우측 하단의 IPython 콘솔에 다음과 같이 실행 결과가 출력된다(변수 a에 저장된 값은 실행 순서에 따라 첫 줄에 5가 출력되고, 두 번째 줄에 다시 5가 더해진 10이 출력된다).



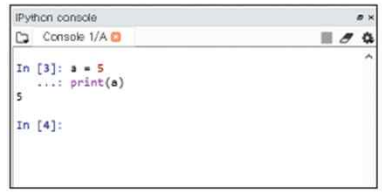
[그림 0-22] 전부 실행(콘솔 화면 실행 결과)

1) File: example/part0/sample.py

에디터 창에 입력한 코드를 일부분만 블록으로 선택하여 해당 부분만 별도로 실행할 수 있다. 이 책에서 "부분 실행"이라고 안내하는 방식이다. 실행하려는 코드 부분을 블록 영역으로 선택하고, 상단 메뉴의 부분 실행 버튼(⏯)을 클릭하거나 단축키 Ctrl+Enter를 누른다. 콘솔 화면에는 선택한 블록만 실행되어, a값으로 5만 출력된다.



[그림 0-23] 부분 실행(코드 블록으로 선택)



[그림 0-24] 부분 실행(콘솔 화면 실행 결과)

❸ (IPython) 콘솔 모드

IPython 콘솔은 인터랙티브 실행 환경이다. 따라서 콘솔에 표시된 명령 프롬프트에 파이썬 코드를 직접 입력하면 바로 실행 결과가 출력된다. 다음 그림에서 "`In [4]:`" 오른쪽에 커서를 놓고 원하는 코드를 입력하면 '`Out[4]:`' 우측에 실행 결과가 출력된다. 변수 a(현재값 5)에 숫자 5를 더하는 식을 입력한 결과, 숫자 10이 출력된 것을 확인할 수 있다. 다음 줄에는 다음의 명령 입력을 위해 '`In [5]:`'와 같이 명령 프롬프트가 다시 나타난다.

[그림 0-25] 콘솔 모드

3. 가상환경 생성

- Anaconda Prompt(anaconda3)를 관리자 권한으로 실행

- 가상환경 생성
  (base) c:/> conda create -n cpu_env python=3.7 openssl

```
(base) C:₩Windows₩system32>conda create -n cpu_env python=3.7 openssl
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:₩Users₩user₩anaconda3₩envs₩cpu_env

  added / updated specs:
    - openssl
    - python=3.7


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    certifi-2020.6.20          |           py37_0         156 KB
    pip-20.1.1                 |           py37_1         1.7 MB
    python-3.7.7               |       h81c818b_4        14.3 MB
    setuptools-49.2.1          |           py37_0         750 KB
    wheel-0.34.2               |           py37_0          66 KB
    wincertstore-0.2           |           py37_0          14 KB
    ------------------------------------------------------------
                                           Total:        17.0 MB

The following NEW packages will be INSTALLED:

  ca-certificates    pkgs/main/win-64::ca-certificates-2020.6.24-0
  certifi            pkgs/main/win-64::certifi-2020.6.20-py37_0
  openssl            pkgs/main/win-64::openssl-1.1.1g-he774522_0
  pip                pkgs/main/win-64::pip-20.1.1-py37_1
  python             pkgs/main/win-64::python-3.7.7-h81c818b_4
  setuptools         pkgs/main/win-64::setuptools-49.2.1-py37_0
  sqlite             pkgs/main/win-64::sqlite-3.32.3-h2a8f88b_0
  vc                 pkgs/main/win-64::vc-14.1-h0510ff6_4
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
  wheel              pkgs/main/win-64::wheel-0.34.2-py37_0
  wincertstore       pkgs/main/win-64::wincertstore-0.2-py37_0
  zlib               pkgs/main/win-64::zlib-1.2.11-h62dcd97_4


Proceed ([y]/n)? y


Downloading and Extracting Packages
```

가상환경 정보 출력
- (base) c:/> conda info --env

```
(base) C:₩Windows₩system32>conda info --env
# conda environments:
#
base                  *  C:₩Users₩user₩anaconda3
cpu_env                  C:₩Users₩user₩anaconda3₩envs₩cpu_env
```

생성된 가상환경으로 전환
- (base) c:/> activate cpu_env

```
(base) C:₩Windows₩system32>activate cpu_env

(cpu_env) C:₩Windows₩system32>python -V
Python 3.7.7
```

가상환경에서 주피터노트북 설치
- (base) c:/> conda install nb_conda

```
(cpu_env) C:₩Windows₩system32>conda install nb_conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.


(cpu_env) C:₩Windows₩system32>
```

3. 가상환경 생성

- 환경설정 파일 생성
  (base) c:/> jupyter notebook –generate-config

```
(cpu_env) C:₩Windows₩system32>jupyter notebook --generate-config
Writing default config to: C:₩Users₩user₩.jupyter₩jupyter_notebook_config.py

(cpu_env) C:₩Windows₩system32>
```

- 주피터 노트북 홈디렉터리 생성(소스디렉터리)
  C:₩Users₩user₩.jupyter₩jupyter_notebook_config.py
  파일을 메모장 또는 파이썬 프로그램으로 열어준다.

- CTRL + F => c.NotebookApp.notebook_dir 검색한다.

- 검색된 269 라인의 구문을 주석제거후
  소스디렉터리를 지정해준다.

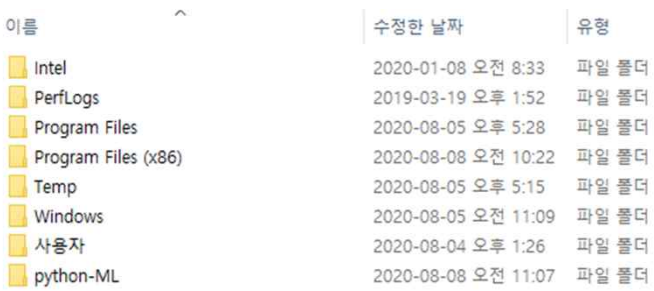
  c.NotebookApp.notebook_dir = "C:/python-ML"

```
## Dict of Python modules to load as notebook server extensions.Entry values can
#  be used to enable and disable the loading ofthe extensions. The extensions
#  will be loaded in alphabetical order.
#c.NotebookApp.nbserver_extensions = {}

## The directory to use for notebooks and kernels.
c.NotebookApp.notebook_dir = "C:/python-ML"

## Whether to open in a browser after starting. The specific browser used is
#  platform dependent and determined by the python standard library `webbrowser`
#  module, unless it is overridden using the --browser (NotebookApp.browser)
#  configuration option.
#c.NotebookApp.open_browser = True
```

C:/python-ML 폴더 생성



이름	수정한 날짜	유형
Intel	2020-01-08 오전 8:33	파일 폴더
PerfLogs	2019-03-19 오후 1:52	파일 폴더
Program Files	2020-08-05 오후 5:28	파일 폴더
Program Files (x86)	2020-08-08 오전 10:22	파일 폴더
Temp	2020-08-05 오후 5:15	파일 폴더
Windows	2020-08-05 오전 11:09	파일 폴더
사용자	2020-08-04 오후 1:26	파일 폴더
python-ML	2020-08-08 오전 11:07	파일 폴더

- 주피터노트북 실행
  (base) c:/> jupyter notebook

```
(cpu_env) C:₩Windows₩system32>jupyter notebook --generate-config
Writing default config to: C:₩Users₩user₩.jupyter₩jupyter_notebook_config.py

(cpu_env) C:₩Windows₩system32>jupyter notebook
[I 11:08:59.302 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 11:08:59.606 NotebookApp] [nb_conda] enabled
[I 11:08:59.606 NotebookApp] Serving notebooks from local directory: C:/python-ML
[I 11:08:59.606 NotebookApp] Jupyter Notebook 6.1.1 is running at:
[I 11:08:59.606 NotebookApp] http://localhost:8888/?token=5cb34f1ab08d57ff4b4e81d976a45b4c9d5eb56b2a84c2f6
[I 11:08:59.606 NotebookApp]  or http://127.0.0.1:8888/?token=5cb34f1ab08d57ff4b4e81d976a45b4c9d5eb56b2a84c2f6
[I 11:08:59.606 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 11:08:59.622 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/user/AppData/Roaming/jupyter/runtime/nbserver-3356-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=5cb34f1ab08d57ff4b4e81d976a45b4c9d5eb56b2a84c2f6
     or http://127.0.0.1:8888/?token=5cb34f1ab08d57ff4b4e81d976a45b4c9d5eb56b2a84c2f6
[I 11:09:05.926 NotebookApp] Creating new notebook in
[I 11:09:06.503 NotebookApp] Kernel started: bf690fa0-682a-4802-8471-1946fd484801, name: python3
[W 11:09:06.545 NotebookApp] 404 GET /nbextensions/widgets/notebook/js/extension.js?v=20200808110859 (::1) 5.98ms refe
rer=http://localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
```

Part 7. 머신러닝 데이터 분석

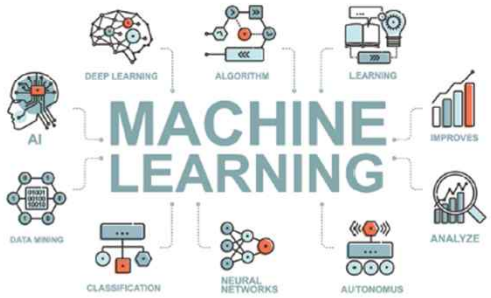
Part 7. 머신러닝 데이터 분석

1. 머신러닝 개요

1-1 머신러닝이란?

학문적 정의에서 벗어날 지는 모르겠지만 머신러닝(machine learning)이란 기계(컴퓨터 알고리즘) 스스로 데이터를 학습하여 서로 다른 변수 간의 관계를 찾아 나가는 과정이라고 정의할 수 있다. 해결하려는 문제에 따라 예측(prediction), 분류(classification), 군집(clustering) 알고리즘 등으로 분류된다. 예를 들면, 주가, 환율 등 경제지표 예측, 은행에서 고객을 분류하여 대출을 승인하거나 거절하는 문제, 비슷한 소비패턴을 가진 고객 유형을 군집으로 묶어내는 문제 등이 있다.

머신러닝이 워낙 다양한 영역에 걸쳐 있고, 사용하는 알고리즘과 방법론이 무수히 많기 때문에 체계적으로 이론을 정립해 나가려면 상당한 시간과 노력이 필요하다. 이 과정에서 많은 초심자들이 중간에 포기하거나 흥미를 잃게 된다. 따라서 복잡한 이론보다는 실제 데이터를 가지고 간단한 문제부터 예측해보는 실습을 통해 익혀 나가는 것이 바람직하다.



[그림 7-1] 머신러닝 개념

1-2 지도 학습 vs 비지도 학습

머신러닝은 크게 두 가지 유형으로 분류한다. 정답 데이터를 다른 데이터와 함께 컴퓨터 알고리즘에 입력하는 방식을 지도 학습(supervised learning)이라고 하고, 정답 데이터 없이 컴퓨터 알고리즘 스스로 데이터로부터 숨은 패턴을 찾아내는 방식을 비지도 학습(unsupervised learning)이라고 한다.

비유하자면 지도 학습은 정답지가 있어서 정답을 맞춰 보면서 문제를 풀어가는 학습 방법이고, 비지도 학습은 정답지 없이 스스로 답을 찾는 학습 방법이다. 지도 학습에는 회귀분석, 분류 모형이 있고, 비지도 학습 중에는 군집 분석이 대표적인 방법이다.

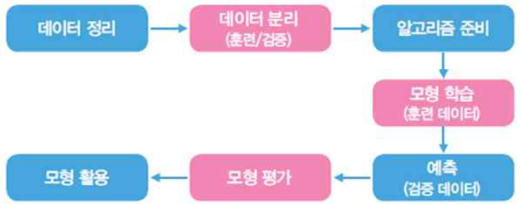
구 분	지도 학습 (supervised learning)	비지도 학습 (unsupervised learning)
알고리즘 (분석모형)	• 회귀분석 • 분류	• 군집 분석
특징	• 정답을 알고 있는 상태에서 학습 • 모형 평가 방법이 다양한 편	• 정답이 없는 상태에서 서로 비슷한 데이터를 찾아서 그룹화 • 모형 평가 방법이 제한적

[표 7-1] 지도 학습 vs 비지도 학습

1-3 머신러닝 프로세스

머신러닝 데이터 분석을 시작하기 전에 컴퓨터 알고리즘이 이해할 수 있는 형태로 데이터를 변환하는 작업이 선행되어야 한다. 분석 대상에 관해 수집한 관측값(observation)을 속성(feature 또는 variable)을 기준으로 정리한다. 따라서 판다스를 이용하여 데이터프레임으로 정리하는 과정이 필요하다. 데이터프레임의 열은 속성을 나타내는 변수들이 위치하고, 데이터프레임의 행은 하나의 관측값을 나타낸다. 분석 대상에 대한 관측값의 개수만큼 행을 늘리면 된다.

컴퓨터 알고리즘이 이해할 수 있도록 데이터프레임으로 변환한 다음에는 여러 속성(변수) 간의 관계를 분석하여 결과를 예측하는 모형을 학습을 통해 찾는다. 모형 학습에 사용하는 데이터를 훈련 데이터(train data)라고 부른다. 학습을 마친 모형의 예측 능력을 평가하기 위한 데이터를 검증 데이터(test data)라고 말한다. 검증 과정을 통해 학습을 마친 모형의 예측 능력을 평가하고, 평가 결과를 바탕으로 최종 모형을 확정하여 문제 해결에 적용한다.



[그림 7-2] 머신러닝 프로세스

Part 7. 머신러닝 데이터 분석

1. 머신러닝 개요

(1) 문제 정의
프로젝트의 목표를 정의하고 그에 따른 계획을 세우는 단계로, 대략적으로 어떤 데 이터를 수집해서 어떤 머신러닝 알고리즘으로 문제를 해결할지 결정한다.

(2) 데이터 획득
머신러닝 모델 학습에 필요한 데이터를 수집하는 단계이다.
데이터는 데이터베이스 테이블, 엑셀 파일, 로그 파일 등 다양한 포멧으로 존재할 수 있으며, 가능하면 머신러닝 학습에 용이하도록 하나의 포멧으로 통일해서 데이 터를 한곳에 수집합니다.
성공적인 머신러닝 프로젝트는 뛰어난 머신러닝 실력보다도 얼마나 충분히 데이터 를 획득했느냐에 의해 결정되므로 절대 소홀히 해서는 안 되는 단계이다.
획득된 데이터는 학습 데이터, 검증 데이터, 테스트 데이터로 구분되어 사용된다.

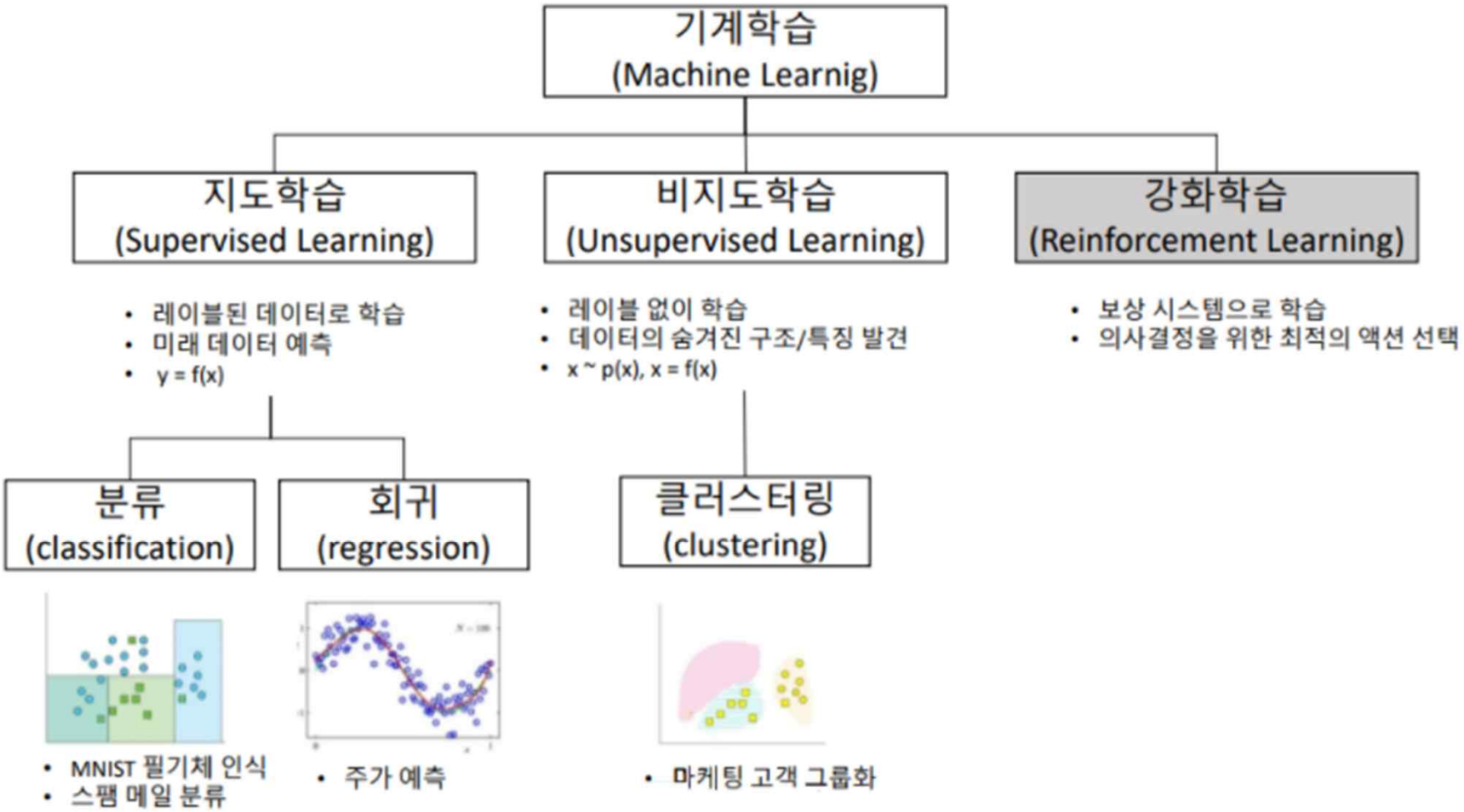
1. 머신러닝 개요

(3) 모델 구현
학습 데이터를 기반으로 한 개 이상의 머신러닝 모델을 구현한다.

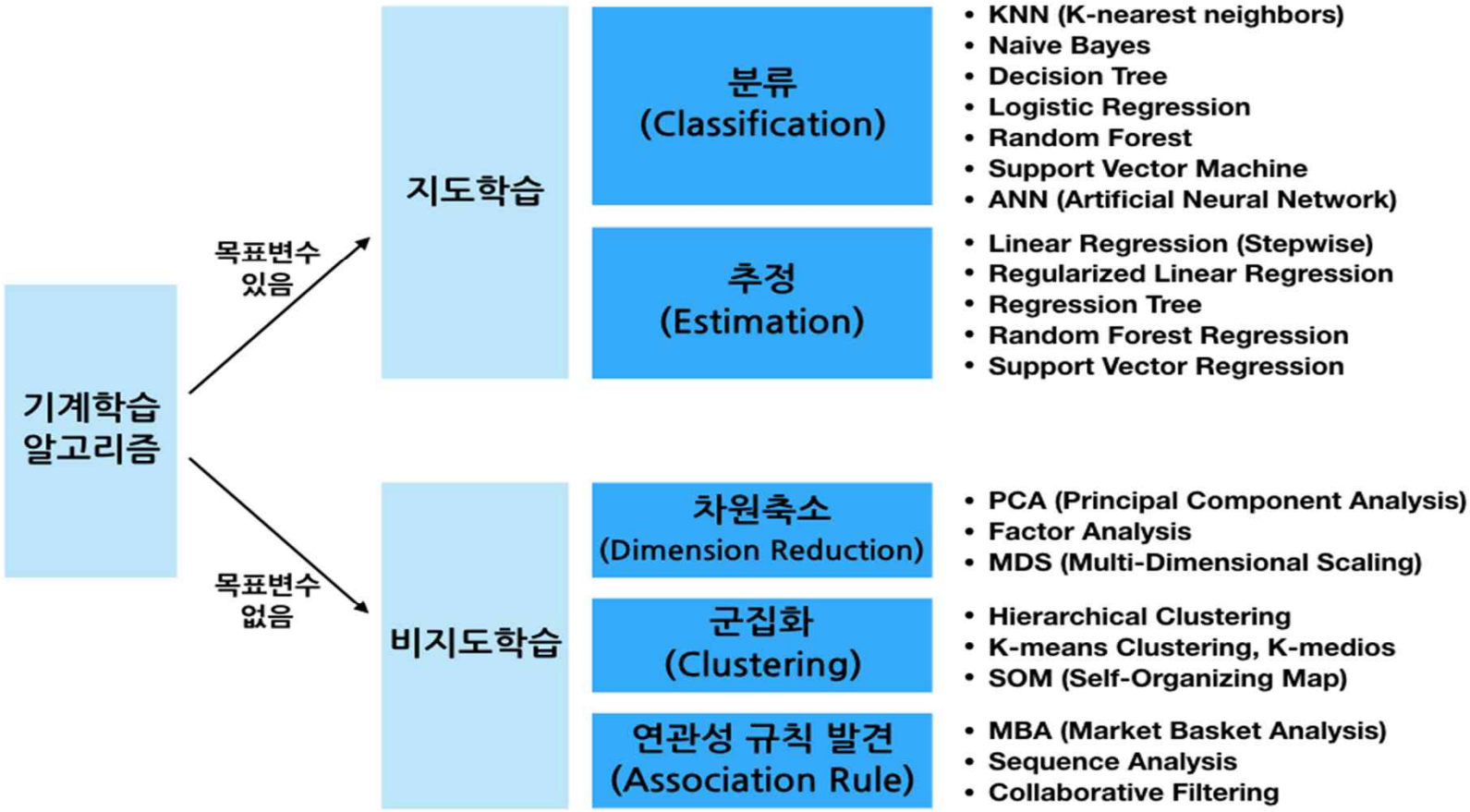
(4) 검증
검증 데이터를 사용해 구현된 머신러닝 모델들의 성능을 검증한다.
검증 결과를 바탕으로 머신러닝 모델을 튜닝해서 더욱 최적화된 머신러닝 모델을 만들 수 있다.
최종적으로 검증 결과가 가장 높은 모델 하나를 선택한다.

(5) 테스트 단계
검증 결과를 통해 선택된 모델의 성능을 테스트 데이터로 측정한다.
테스트 결과에 따라 프로젝트를 마무리할지, 아니면 이전 단계로 돌아가서 모델을 개선할지를 결정하게 된다.

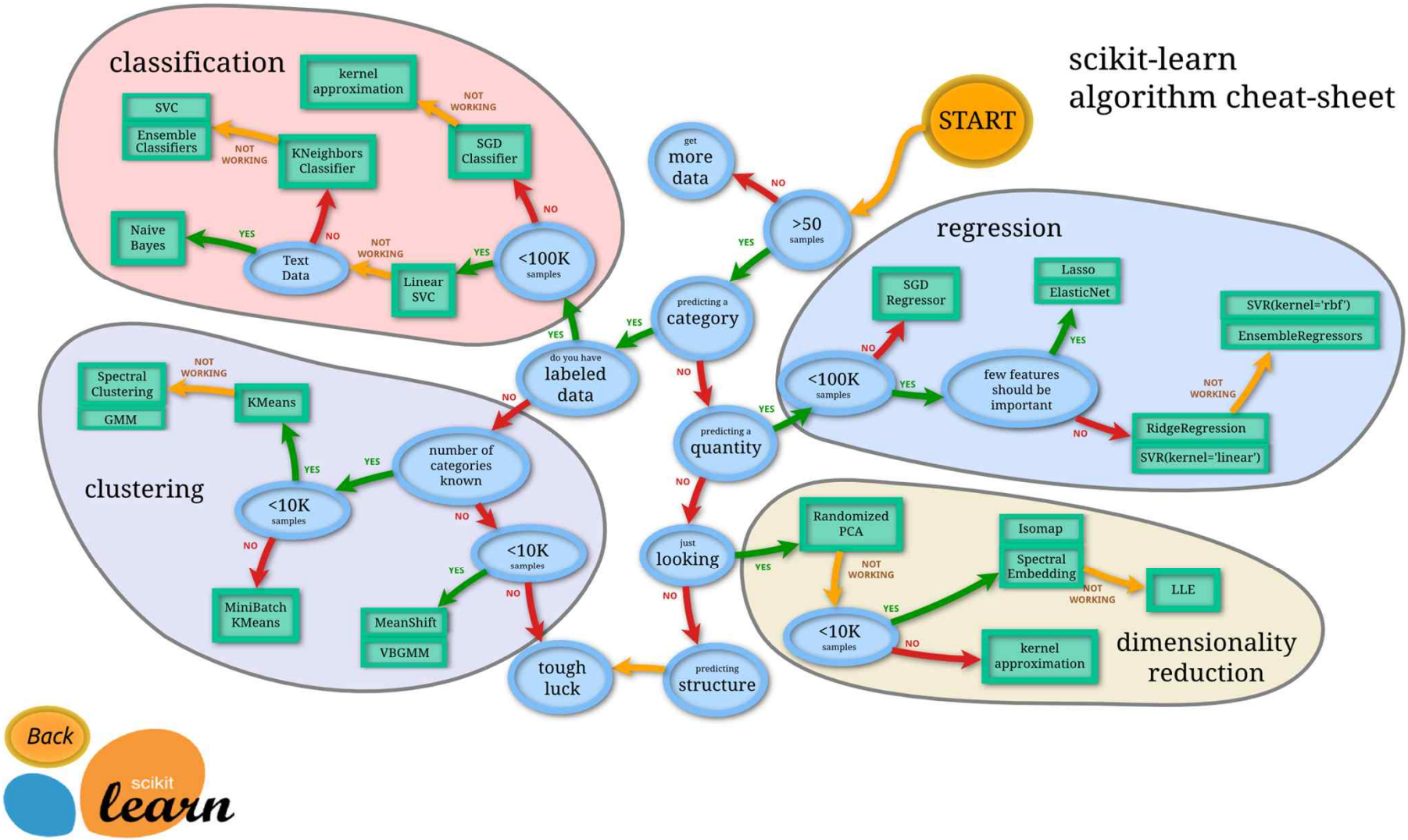
1. 머신러닝 개요


기계학습
(Machine Learnig)
지도학습
(Supervised Learning)
비지도학습
(Unsupervised Learning)
강화학습
(Reinforcement Learning)
• 레이블된 데이터로 학습
• 미래 데이터 예측
• y = f(x)
• 레이블 없이 학습
• 데이터의 숨겨진 구조/특징 발견
• x ~ p(x), x = f(x)
• 보상 시스템으로 학습
• 의사결정을 위한 최적의 액션 선택
분류
(classification)
회귀
(regression)
클러스터링
(clustering)
• MNIST 필기체 인식
• 스팸 메일 분류
• 주가 예측
• 마케팅 고객 그룹화

1. 머신러닝 개요


기계학습
알고리즘
목표변수
있음
목표변수
없음
지도학습
비지도학습
분류
(Classification)
추정
(Estimation)
차원축소
(Dimension Reduction)
군집화
(Clustering)
연관성 규칙 발견
(Association Rule)
• KNN (K-nearest neighbors)
• Naive Bayes
• Decision Tree
• Logistic Regression
• Random Forest
• Support Vector Machine
• ANN (Artificial Neural Network)
• Linear Regression (Stepwise)
• Regularized Linear Regression
• Regression Tree
• Random Forest Regression
• Support Vector Regression
• PCA (Principal Component Analysis)
• Factor Analysis
• MDS (Multi-Dimensional Scaling)
• Hierarchical Clustering
• K-means Clustering, K-medios
• SOM (Self-Organizing Map)
• MBA (Market Basket Analysis)
• Sequence Analysis
• Collaborative Filtering

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



Part 7. 머신러닝 데이터 분석

1. 머신러닝 알고리즘의 장단점 비교

K-최근접 이웃

장점 - 구현이 쉽다.
- 알고리즘을 이해하기 쉽다
- 하이퍼 파라미터가 적다

단점 - 예측 속도가 느리다.
- 메모리를 많이 쓴다
- 노이즈 데이터에 예민하다

서포트 벡터 머신

장점 - 적은 데이터로도 높은 정확도를 낸다.
- 예측속도가 빠르다
- 고차원 데이터 처리가 쉽다

단점 - 결정경계선이 겹치면 정확도가 낮아진다
- 수학적 이해도가 필요하다
- 커널트릭 오사용시 과대적합되기 쉽다

의사결정트리

장점 - 모델 추론 과정의 시각화가 쉽다
- 데이터의 특성파악이 용이하다
- 학습 및 예측 속도가 빠르다

단점 - 과대적합되기 쉽다
- 조정해야 할 하이퍼파라미터가 많다

랜덤 포레스트

장점 - 앙상블 효과로 의사결정트리의 과대적합 단점을 보완한다

단점 - 조정해야 할 하이퍼파라미터가 많다

Part 7. 머신러닝 데이터 분석

1. 머신러닝 알고리즘의 장단점 비교

나이브베이즈

장점 - 고차원 데이터 처리가 쉽다
- 구현이 쉽다
- 학습 및 추론 시간이 빠르다

단점 - 모든 변수가 독립변수라는 가설하에 작동함으로써 데이터가 가설과 다를 경우 정확도가 낮아진다.

로지스틱회귀

장점 - 데이터 분류시 확률을 제공한다.

단점 - 데이터의 특징이 많을 경우 학습이 어려워 과소적합되기 쉽다

선형회귀

장점 - 수집된 데이터를 통해 새롭게 관측된 데이터의 예측값(수치값)을 구할 수 있다.

단점 - 데이터 특징들이 선형 관계에 있다는 가설하에 작동함으로써 데이터 특징이 가설과 다를 경우 정확도가 낮아진다.

K 평균

장점 - 데이터 크기에 상관없이 군집화에 사용가능
- 구현하기 쉽다

단점 - 군집화 결과에 대한 확률 제공이 없다.
- 데이터의 분포가 균일하지 않을 경우 정확도가 떨어진다.

주성분 분석

장점 - 고차원 -> 저차원 데이터로 축소시 사용
- 구현이 쉽다

단점 - 차원 축소 시 정보의 손실 발생

Part 7. 머신러닝 데이터 분석

1-1. 지도학습 & 비지도 학습

학습이란 데이터를 특별한 알고리즘에 적용하여 머신러닝 모델을 정의된 문제에 최적화하는 과정을 말한다.

1-2. Supervised learning : 지도학습

지도학습이란 정답을 알려주면서 진행되는 학습으로, 학습시 데이터와 함께 레이블(정답)이 항상 제공되어야 한다. 지도학습을 하다보면 정답, 실제값, 레이블, 타깃, 클래스, y값 이라는 단어가 많이 혼용되지만 모두 다 같은 의미이다.
주로 주어진 데이터와 레이블을 이용해 새로운 데이터의 레이블을 예측해야 할 때 사용된다.
테스트할 때는 데이터와 함께 레이블을 제공해서 손쉽게 모델의 성능을 평가할 수 있다는 장점이 있지만, 데이터마다 레이블을 달기 위해 많은 시간을 투자해야 한다는 단점도 있다.
지도학습의 대표적인 예로는 분류와 회귀가 대표적이다.

1-3. Unsupervised learning : 비지도학습

비지도학습이란 레이블(정답)이 없이 진행되는 학습을 말한다. 따라서, 학습할 때 레이블 없이 데이터만 필요하다.
보통 데이터 자체에서 패턴을 찾아내야 할 때 사용되며, 레이블이 없기 때문에 모델 성능을 평가하는데 다소 어려움이 있다.
하지만 따로 레이블을 제공할 필요가 없다는 장점이 있다.
비지도학습의 대표적인 예로는 군집화와 차원 축소가 있다.

1-4. 분류(Classification)와 회귀(Regression)

분류(Classification)와 회귀(Regression)의 가장 큰 차이점은 데이터가 입력되었을 때 분류는 분리된 값으로 예측하고, 회귀는 연속된 값으로 예측한다는데 있다. 날씨로 예를 들면, 분류는 덥다/춥다와 같이 분리된 값으로 예측하는 반면에, 회귀는 30.5도, 3.5도와 같이 연속된 수치값으로 예측한다.

1-5. Classification : 분류

분류는 데이터가 입력되었을 때 지도학습을 통해 미리 학습된 레이블 중 하나 또는 여러 개의 레이블로 예측하는 것이다.

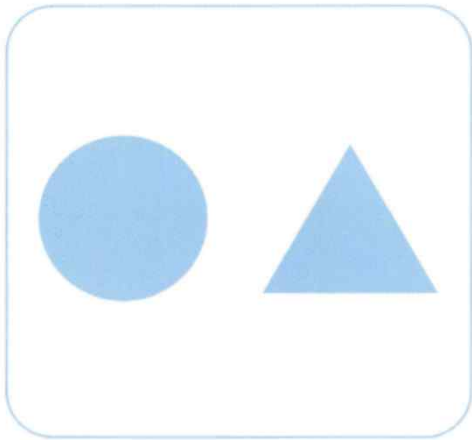
1) 이진분류
(예, 아니오), (남자, 여자)와 같이 둘 중 하나의 값으로 분류하는 경우 이진분류라고 부른다.

2) 다중분류
(빨강, 녹색, 파랑) 중 하나의 색으로 분류하거나, 0부터 9까지의 손글씨 숫자 중 하나의 숫자로 분류하기 처럼 여러 개의 분류값중에서 하나의 값으로 예측하는 문제를 다중분류라고 부른다.

1-6. 다중 레이블 분류

데이터가 입력됐을 때 두 개 이상의 레이블로 분류할 경우 다중 레이블 분류하고 한다.

예를들어,
분류값으로 세모, 네모 , 동그라미가 있을 경우 아래와 같은 그림이 입력값으로 들어오면 다중 레이블 분류 모델의 예측값은 (동그라미, 세모)가 되고, 다중 분류 모델일 경우 세모와 네모 중 더 높은 확률을 지닌 레이블로 예측하게 된다.



1-7. **Regression : 회귀**

회귀는 입력된 데이터에 대해 연속된 값으로 예측한다.

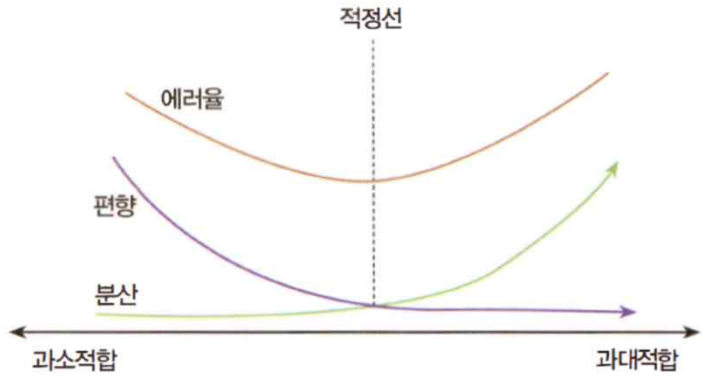
예를 들어,
날씨를 더움, 보통, 추움 이라는 3가지로만 예측하는 분류와 달리,

회귀는 35도, 34.5도, 34도와 같이 정해진 레이블이 아닌 연속성을 가진 수치로 예측한다.

1-8. **과대적합(overfitting) & 과소적(underfitting)**

머신러닝 모델 학습에 가장 큰 영향을 주는 것은 데이터이다.

데이터에서 충분히 특징을 찾아내지 못하고 머신러닝 모델을 학습할 경우 모델이 과소적합(underfitting) 되기 쉽고, 필요 이상의 특징으로 학습할 경우 모델이 과대 적합(overfitting) 되기 쉽다.



1-9. 과소적합(underfitting)

과소 적합은 모델 학습 시 충분한 데이터의 특징을 활용하지 못할 경우 발생한다.

사물	분류값	생김새
야구공	공	동그라미
농구공	공	동그라미
테니스공	공	동그라미
딸기	과일	세모
포도알	과일	동그라미

사물을 보고 공을 구분하는 머신러닝 모델을 만들 때 위 학습 데이터를 사용할 경우 어떤 문제가 발생 할까?

데이터의 특징으로는 생김새밖에 없으므로 생김새가 동그라미이면 공이라는 간단한 머신러닝 분류기를 만들 수 있다. 하지만 이 분류기는 학습 데이터에 대해서도 높은 정확도를 갖지 못한다.

1-10. 과대적합(overfitting)

학습 데이터에서 필요 이상으로 특징을 발견해서 학습 데이터에 대한 정확도는 상당히 높지만 테스트 데이터 또는 학습 데이터 외의 데이터에는 정확도가 낮게 나오는 모델을 과대 적합된 모델이라 부른다.

사물	분류값	생김새	크기	줄무늬
야구공	공	원형	중간	있음
농구공	공	원형	큼	있음
테니스공	공	원형	중간	있음
딸기	과일	세모	중간	없음
포도알	과일	원형	작음	없음

위 특징 모두를 사용해서 머신러닝 모델을 학습할 경우 "생김새가 원형이고 크기가 작지 않으며, 줄무늬가 있으면 공이다"라는 명제를 가진 머신러닝 모델이 만들어 질 수 있다.

1-11. Confusion matrix : 혼동 행렬

혼동 행렬(confusion matrix)은 모델의 성능을 평가할 때 사용되는 지표이다.
알파벳을 보여줬을 때 알파벳을 맞추는 머신러닝 모델의 혼동행렬을 보겠다.
종으로 나열된 A, B, C, D는 입력된 데이터의 실제값이고, 횡으로 나열된 A, B, C, D는 예측값이다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

A : A를 10번 보여줬을 경우, 9번은 A로 정확하게 맞췄으나, 1번은 B 라고 대답함

B : B를 20번 보여줬을 경우, 15번은 B로 맞췄으나 1번은 A, 3번은 C, 1번은 D 라고 대답함

c : C를 30번 보여줬을 경우, 24번은 C로 맞췄으나 5번은 A, 1번은 D 라고 대답함

D : D를 20번 보여줬을 경우, 15번은 D로 맞췄으나, 4번은 B, 1번은 C 라고 대답함

1-12. 머신러닝 모델의 성능 평가

TP(true positive) : 맞는 것을 올바르게 예측한 경우
데이터를 입력했을 때 데이터의 실제값을 올바르게 예측한 케이스를 TP라고 한다.
아래 혼동 행렬의 대각선 부분이 TP이다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

1-12. 머신러닝 모델의 성능 평가

TN(true negative) : 틀린 것을 올바르게 예측한 경우
틀린 것을 올바르게 예측한 것을 TN 이라고 한다.
A 클래스의 TN은 A가 아닌 클래스들을 A가 아니라고 예측한 모든 값이다.
아래 혼동행렬에 색깔로 표시한 셀들이 바로 A 클래스의 TN 이다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

1-12. 머신러닝 모델의 성능 평가

FP(false positive) : 틀린 것을 맞다고 잘못 예측한 경우
틀린 것을 맞다고 잘못 예측한 것을 FP라고 한다.
A 클래스로 예를 들면, A가 아닌 B, C, D가 실제값인 데이터들을 입력했을 때, A라고 잘못 예측한 값들이 A 클래스의 FP 이다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

1-12. 머신러닝 모델의 성능 평가

FN(false nagative) : 맞는 것을 틀렸다고 잘못 예측한 경우
맞는 것을 틀렸다고 잘못 예측한 것을 FN이라고 한다.
A 클래스로 예를 들면, A 라는 실제값인 데이터를 입력했을 때 A가 아니라고 예측한 모든 케이스를 FN 이라고 한다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

1-12. 머신러닝 모델의 성능 평가

Accuracy : 정확도
정확도(Accuracy)는 가장 일반적인 모델 성능 평가지표이다.
모델이 입력된 데이터에 대해 얼마나 정확하게 예측하는지를 나타낸다.
혼동 행렬상에서는 대각선(TP)을 전체 셀로 나눈 값에 해당한다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

정확도 = 9 + 15 + 24 + 15 / 80 = 0.78

위 모델의 정확도는 .78임을 확인할 수 있다.

1-12. 머신러닝 모델의 성능 평가

Precision : 정밀도
정밀도(precision) 는 모델의 예측값이 얼마나 정확하게 예측됐는가를 나타내는 지표이다.
정밀도는 언제 정확도보다 더 자주 시용될까?
가장 이해하기 쉬운 예로는 병원에서 암을 진찰하는 기계를 들 수 있다.

정밀도 = TP / (TP + FP)

A 모델의 경우
9 / (9 + 30) = 23%

B 모델의 경우
1 / (1 + 20) = 4.7%

표3.12 암 예측 모델 A의 혼동행렬

	암환자	일반환자
암환자	9	1
일반환자	30	60

표 3.13 암 예측 모델 B의 혼동행렬

	암환자	일반환자
암환자	1	9
일반환자	20	70

1-12. 머신러닝 모델의 성능 평가

Recall : 재현율
재현율(Recall) 은 실제값 중에서 모델이 검출한 실제값의 비율을 나타내는 지표이다.

재현율 = TP / (TP + FP)

A 모델의 경우
9 / (9 + 1) = 90%

B 모델의 경우
1 / (1 + 9) = 10%

표3.12 암 예측 모델 A의 혼동행렬

	암환자	일반환자
암환자	9	1
일반환자	30	60

표 3.13 암 예측 모델 B의 혼동행렬

	암환자	일반환자
암환자	1	9
일반환자	20	70

1-12. 머신러닝 모델의 성능 평가

F1 Score
정밀도도 중요하고 재현율도 중요한데 둘 중 무엇을 쓸지 고민될 수 있다.
이 두 값을 조화 평균내서 하나의 수치로 나타낸 지표를 F1 점수(F1 Score) 라고 한다.

조화평균 = 2 * a * b / (a + b)

F1 점수 = 2 * 재현율 * 정밀도 / (재현율 + 정밀도)