데이터과학과 머신러닝

2021 학생과목선택권 확대를 위한 교사 역량강화 연수

경덕여자고등학교 교사 이준구
(heback@gmail.com)

목차

왜 데이터과학인가?

일반적인 데이터의 의미

「명사」

「1」 이론을 세우는 데 기초가 되는 사실. 또는 바탕이 되는 자료.

- 고용 실태에 관한 데이터.
- 조직체의 활동에 필요한 데이터를 수집하여 체계적으로 정리하다.

「2」 관찰이나 실험, 조사로 얻은 사실이나 정보.

- 연구 데이터를 내다.

「3」『정보·통신』 컴퓨터가 처리할 수 있는 문자, 숫자, 소리, 그림 따위의 형태로 된 정보.

- 데이터 용량.
- 데이터를 입력하다.
- 데이터를 저장하다.

<표준국어대사전>

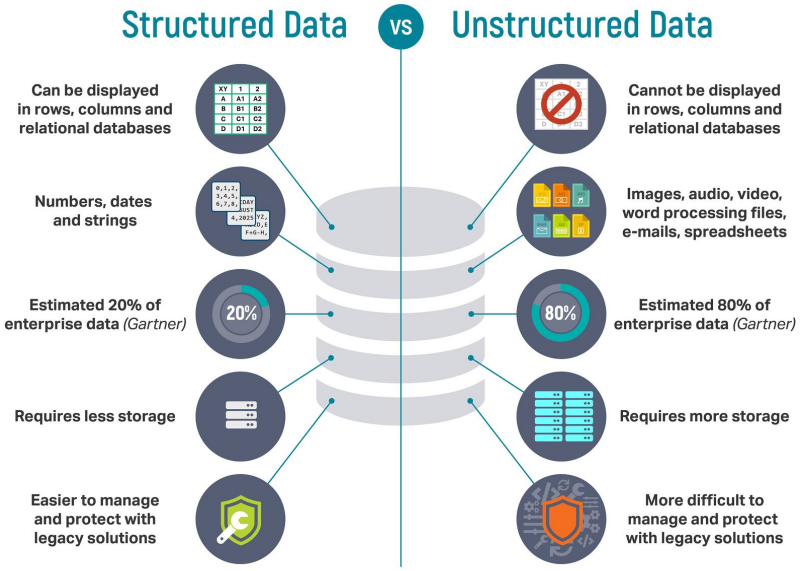
데이터과학에서 데이터란 무엇인가?

컴퓨터가 처리 할 수 있는 형태로 번역 된 사실(숫자, 단어, 측정, 관찰 등)의 모음

컴퓨터가 이해하는 세상의 속성

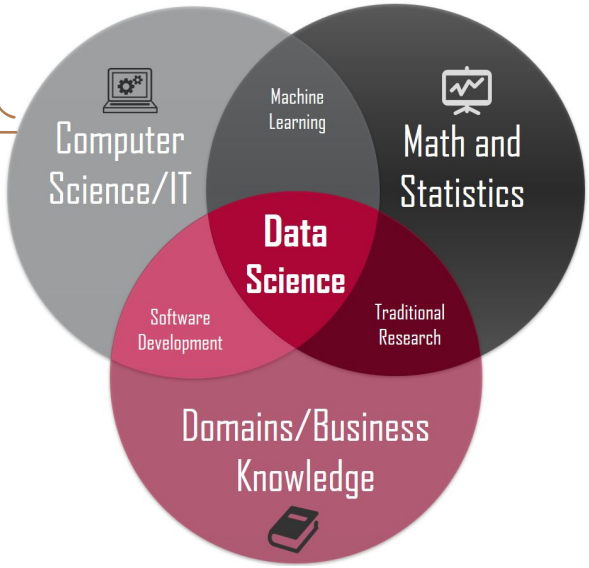
데이터 유형

정형 데이터	비정형 데이터
구조화된 데이터	구조화되지 않은 데이터
컴퓨터가 처리할 수 있는 데이터	사람이 읽을 수 있는 데이터
저장된 항목 간의 관계를 인식할 수 있음 행과 열 즉, 테이블 형태, 특히 관계형 데이터베이스	정형 데이터를 제외한 모든 데이터 텍스트 파일, 이미지, 웹문서, 일반 문서 인간 혹은 기계가 생성
모든 데이터의 20%	모든 데이터의 80%

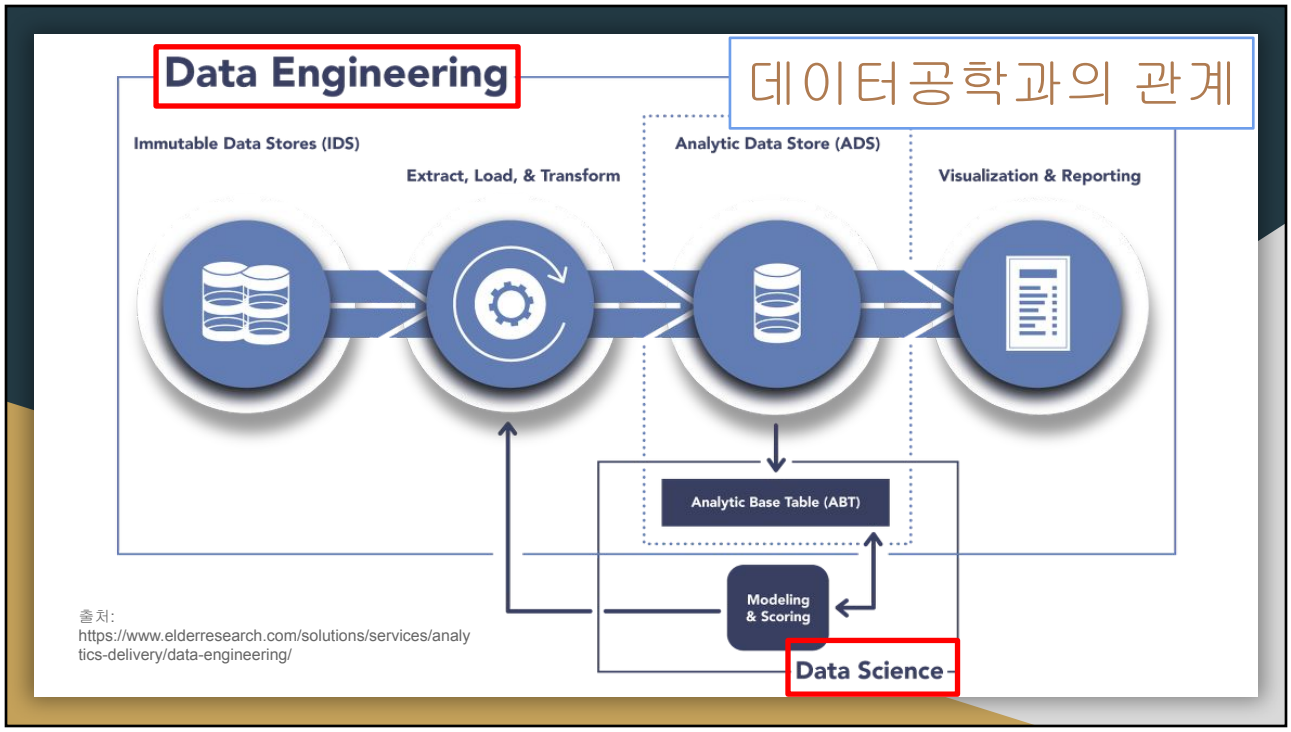


출처: https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/

데이터과학 구성요소



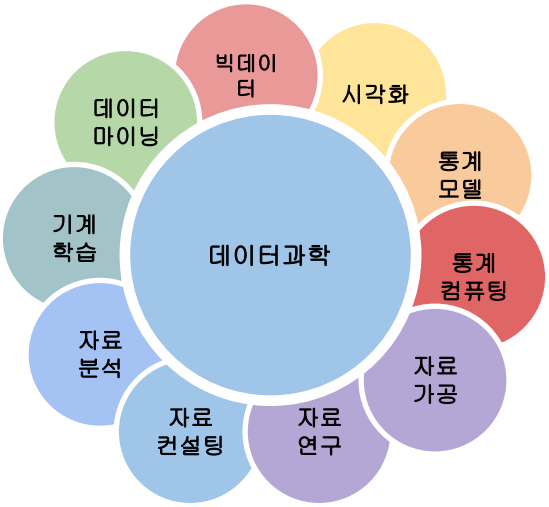
출처: https://towardsdatascience.com/introduction-to-statistics-e9d72d818745



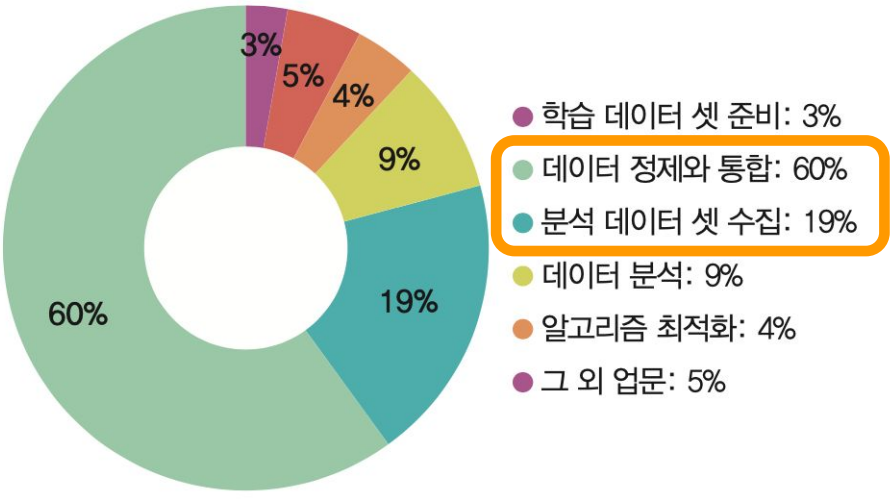
데이터 공학과의 관계

공통점	데이터과학	데이터공학
소프트웨어 기술 빅데이터	통계학 머신러닝 도메인지식	빅데이터 Infrastructure 분산처리 기술 (Hadoop, Spark) 분산시스템 데이터처리 Pipeline 실시간 처리 시스템

데이터과학의 연구 분야



데이터 과학자들이 가장 많이 소비하는 시간



데이터과학
교육과정 구성의 이해

성격 #1

데이터과학은 데이터로부터 통찰(insight)과 가치를 발굴하는 방법을 연구하는 실용 학문이며, **데이터과학 과목**은 데이터 기반의 합리적 의사결정을 위한 과학적 방법을 탐구하고 실생활의 문제해결에 필요한 기술을 습득하여 통합적으로 적용하는 능력과 태도를 함양하는 과목이다.

성격 #2

빅데이터로 인한 사회적 패러다임 변화를 올바르게 인식하고, 데이터 기반 의사결정 과정이 갖는 특장점과 지능정보기술이 가져올 파급효과와 잠재력을 이해하며, **데이터 분석 및 활용을 위한 기본 지식과 기술**을 익혀 실생활 및 다양한 학문 분야의 문제를 창의적으로 해결하는 역량을 기르기 위한 과목

데이터과학 과목의 내용

'데이터 프로그래밍', **'통계분석'**, **'머신러닝'**, **'딥러닝'** 영역으로 구분되며, '데이터 프로그래밍'과 '통계분석' 영역은 지능정보기술사회의 인재로서 갖추어야 할 데이터과학 소양을 증진하는 데 중점을 둔다. '머신러닝'과 '딥러닝'은 데이터 활용 기술인 기계학습의 개념과 원리를 이해하고 실생활 및 다양한 학문 분야의 문제해결 능력 신장에 중점을 둔다.

데이터과학 과목의 목표

- **데이터과학의 기본개념과 원리를 이해**하고 **컴퓨팅시스템을 활용하여 데이터를 수 집하고 처리하는 능력과 태도를 함양**한다.
- **데이터 분석에 필요한 통계적 개념을 이해**하고 **컴퓨팅시스템을 활용하여 데이터를 분석하고 예측하는 능력을 함양**한다.
- **머신러닝**의 **주요 알고리즘을 이해**하고 컴퓨팅시스템을 활용하여 데이터 분석 모델을 문제 해결에 적용하는 능력과 태도를 함양한다.
- **딥러닝**의 **활용 사례 탐구**를 통해 개념과 원리를 이해하고, 딥러닝 기법으로 실생활 문제를 해결할 수 있는 능력과 태도를 함양한다.

데이터 프로그래밍 성취기준

- [12데과01-01] 데이터과학에서 데이터분석을 위해 활용되는 기술과 분야를 파악하고, 관련 분야의 직업과 진로를 탐색한다.
- [12데과01-02] 데이터의 특색에 따라 데이터를 구분하고 비교한다.
- [12데과01-03] 데이터베이스의 특징을 알고 SQL을 이해한다.
- [12데과01-04] 데이터 보안의 필요성을 알고 개인정보보호와 정보보안을 실천한다.
- [12데과01-05] 데이터분석 프로그래밍의 통합 개발 환경 및 특성을 이해한다.
- [12데과01-06] 자료형과 연산자를 활용한 프로그램을 작성한다.
- [12데과01-07] 표준입출력과 파일입출력을 활용한 프로그램을 작성한다.
- [12데과01-08] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.
- [12데과01-09] 데이터의 표현 형식을 이해하고 데이터를 수집한다.
- [12데과01-10] 배열 생성 및 데이터 접근 프로그램을 작성한다.
- [12데과01-11] 데이터 조회, 수정 등 데이터 처리 프로그램을 작성한다.
- [12데과01-12] 데이터 처리 결과를 시각화 정보로 표현한다.

통계분석 성취기준

- [12데과02-01] 데이터과학과 통계의 관계를 설명한다.
- [12데과02-02] 평균, 중앙값, 최빈값의 개념을 설명하고, 이를 계산하는 프로그램을 작성한다.
- [12데과02-03] 산포도의 개념을 설명하고, 이를 계산하는 프로그램을 작성할 수 있다.
- [12데과02-04] 확률과 확률분포의 개념을 예시를 통해 설명한다.
- [12데과02-05] 대푯값과 산포도를 통해 통계적 추정을 작성한다.
- [12데과02-06] 두 변수간의 관계를 시각화하여 표현하고, 상관관계와 상관계수의 개념과 값의 의미를 설명한다.
- [12데과02-07] 두 변수간에 존재하는 선형적 관계를 파악하고, 최소제곱법의 원리 를 설명한다.
- [12데과02-08] 최소제곱법을 적용한 회귀분석 프로그램을 작성하고, 그 결과를 예 측의 관점에서 설명한다.
- [12데과02-09] 주어진 모델의 성능지표(평균제곱오차, 결정계수)를 계산하는 프로 그램을 작성하고, 그 결과에 대하여 설명한다.
- [12데과02-10] 모델 평가를 위한 방법(홀드아웃 교차 검증법, k-분할 교차 검증 법)에 대하여 설명한다.
- [12데과02-11] 주어진 모델을 테스트할 수 있는 프로그램을 작성하여, 테스트용 데이터를 적용하고 그 결과를 설명한다.

머신러닝 성취기준

- [12데과03-01] 머신러닝 기술의 발전 과정과 활용 분야를 탐색한다.
- [12데과03-02] 머신러닝에 대한 다양한 정의를 탐색하고 실생활 문제에 적용한다.
- [12데과03-03] 학습방법에 따라 머신러닝 모델을 지도, 비지도, 강화 학습으로 분류 한다.
- [12데과03-04] 머신러닝 모델링에서 사용하는 용어를 정의하고 설명한다.
- [12데과03-05] 경사하강법의 개념과 원리를 이해하고, 단계별로 설명한다.
- [12데과03-06] 지도학습의 회귀모델을 적용하여 문제를 해결한다.
- [12데과03-07] 지도학습의 분류모델을 적용하여 문제를 해결한다.
- [12데과03-08] 머신러닝 모델의 성능향상을 위한 방법을 탐색한다.

딥러닝 성취기준

- [12데과04-01] 딥러닝과 머신러닝과의 차이점, 딥러닝 기법의 종류 및 특징, 딥러 닝의 활용분야를 이해하고 목적에 따라 딥러닝의 기법을 선택한다.
- [12데과04-02] 인간의 뉴런과 퍼셉트론에서의 자극 전달 방법을 비교하여 이해하 고 퍼셉트론의 XOR문제를 다층퍼셉트론을 활용하여 해결한다.
- [12데과04-03] 오차역전파법의 개념과 필요성, 동작원리를 이해하고 오차 함수의 종류를 설명한다.
- [12데과04-04] 활성화 함수의 개념과 종류, 특징을 설명한다.
- [12데과04-05] 딥러닝의 속도와 정확도를 해결하기 위한 방법과 그 특징을 설명한다.
- [12데과04-06] 딥러닝의 입력층, 은닉층, 출력층을 설계하고 딥러닝의 모델 컴파 일, 모델 최적화, 모델 실행 설정을 실제 코드로 작성한다.
- [12데과04-07] MNIST에서 학습셋과 테스트셋을 구분하고 원하는 데이터의 형태로 가공한다.
- [12데과04-08] 가공된 데이터셋이 주어질 때 DNN을 활용한 딥러닝 코드를 작성한다.

데이터과학 실습환경 준비하기

데이터과학 분야에서 사용되는 도구들

[프로그램 / 언어]

1. Excel : 데이터 사이언스 입문용으로 소규모 데이터를 다룰 때 적합.
2. R : 통계 전용 프로그래밍 언어. 패키지를 활용한 데이터 시각화 등의 부분에 장점을 가지고 있음.
3. Python (파이썬) : 프로그래밍 언어. R에 비해 배우기 쉽고 대용량의 데이터 처리가 원활함.
   * IDLE (아이들) : Python의 기본 제공 에디터.
   * Jupyter (iPython) : Python 에디터. Notebook 형태로 깔끔하게 코드를 보여줌. 마크업 언어 지원.
   * Pycharm (파이참) : Python 에디터. 가상환경 / Django 등의 기능을 제공하지만, 일부 기능이 유료임.
   * Sublime Text 3 : Python 에디터. Linux / OS X / Windows에서 모두 사용 가능.
   * Anaconda : Python 패키지 관리자의 일반인 배포판. 라이브러리 추가를 손쉽게 해준다.
4. SQL : 데이터베이스의 자료 처리용 프로그래밍 언어. (Ex. Oracle, MySQL)

데이터과학 분야에서 사용되는 도구들

[Python Library]

1. Pandas : 데이터 처리/분석. R과 유사하게 테이블 형태 데이터를 위한 데이터프레임 자료형을 제공.
2. Matplotlib (맷플라립) : 과학 계산용 '그래프/시각화'. MATLAB의 그래프 기능과 유사함.
3. Numpy (넘파이) : 수학/과학 계산. 통계/선형대수/벡터/금융 관련 등.
4. Scipy (싸이파이) : 고급 과학 계산. (고성능 선형대수, 함수 최적화, 신호 처리 등)
5. TensorFlow : 머신러닝(딥러닝). 구글에서 개발한 오픈 소스 코드.
6. Pytorch (파이토치) : 머신러닝(딥러닝). 오픈 소스 + 간결함 + 빠른 구현 + 배우기 쉬움 = 최근 급 부상 중.
7. Scikit-learn (싸이킷런) : 머신러닝.
8. Keras : 신경망 관련 머신러닝. 오픈 소스이며, TensorFlow 등의 딥 러닝 라이브러리를 포함하고 있음.
   * Django (장고) : Python의 풀 스택/오픈 소스 웹 프레임워크.
   * Flask : Python의 마이크로 프레임워크. Django에 비해 가볍고 빠르며, 핵심 모듈 교체가 쉬움.

데이터과학 분야에서 사용되는 도구들

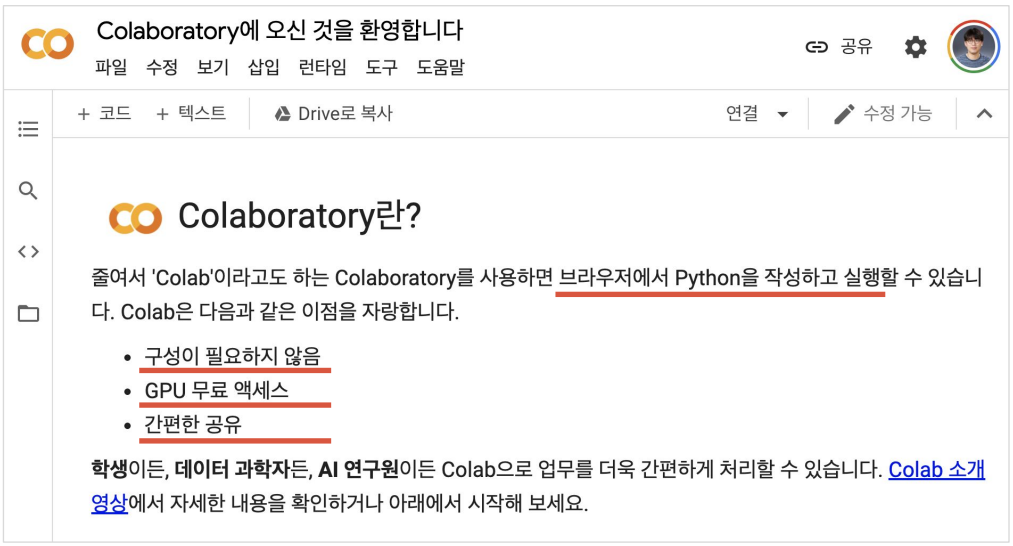
우리의 선택은...

Python ✚ **Jupyter** ✚ **Python Libraries**

Google Colab

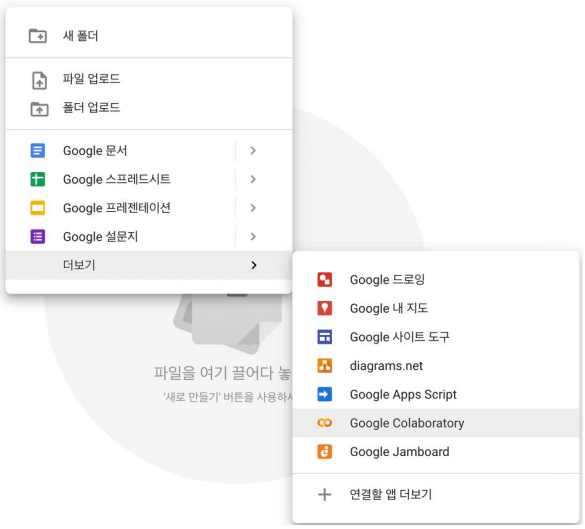
https://colab.research.google.com/

구글 코랩

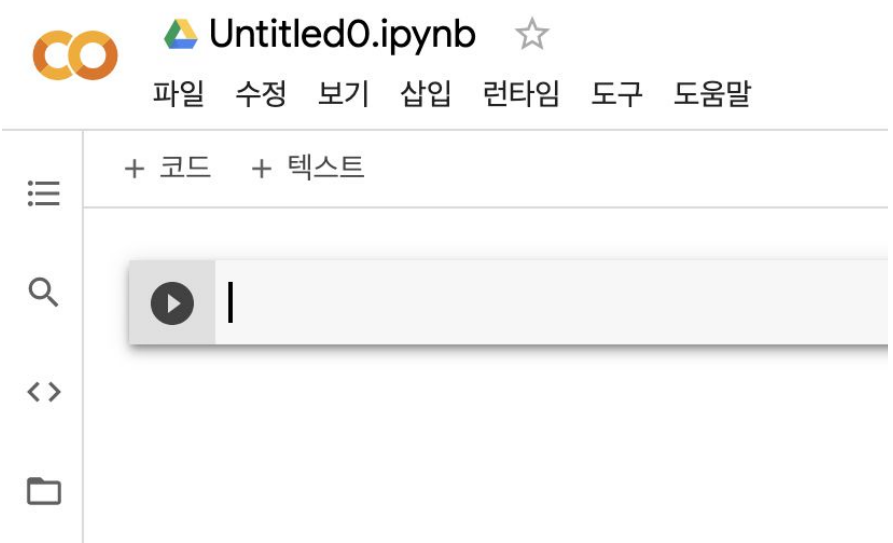


구글 코랩

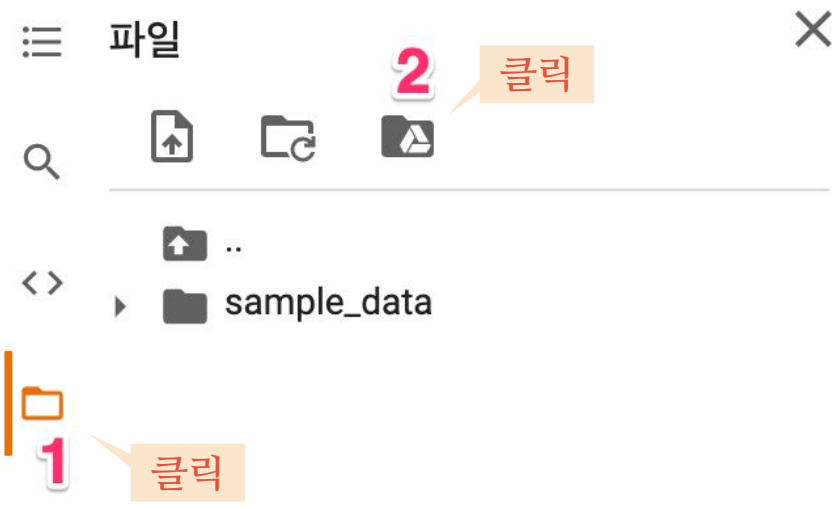
구글 드라이브와 연동



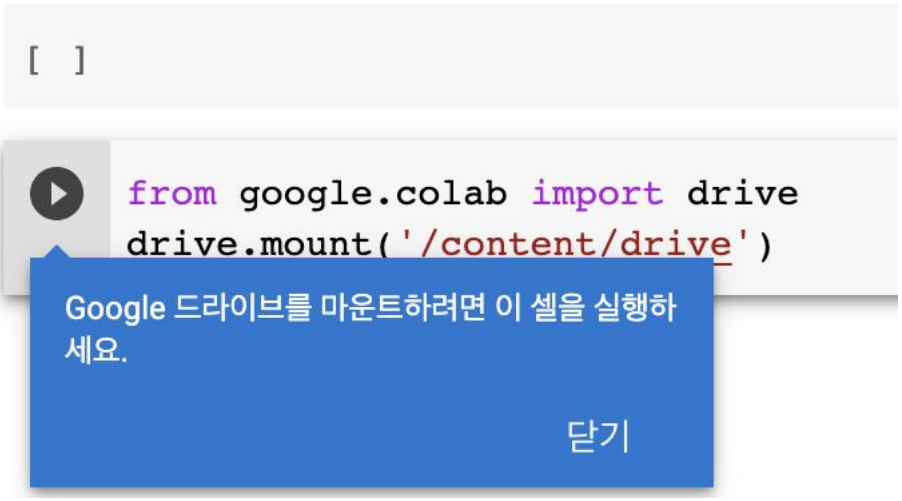
구글 코랩



구글 코랩


파일
2
클릭
..
sample_data
1
클릭

구글 코랩


[ ]
from google.colab import drive
drive.mount('/content/drive')
Google 드라이브를 마운트하려면 이 셀을 실행하세요.
닫기

구글 코랩

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-

Enter your authorization code:
```

클릭

구글 코랩

Google

로그인

이 코드를 복사하여 애플리케이션으로 전환한 다음 붙여넣으세요.

복사

구글 코랩

```
ls
```

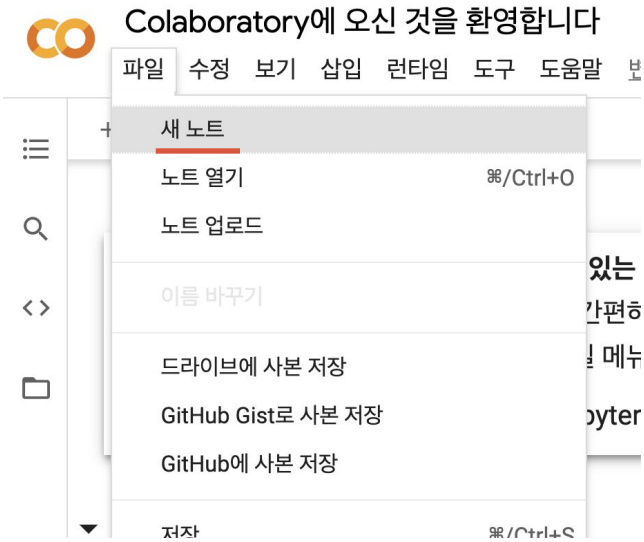
```
Untitled0.ipynb
```

```
[13] cd "study_data"
```

```
/content/drive/My Drive/study_data
```

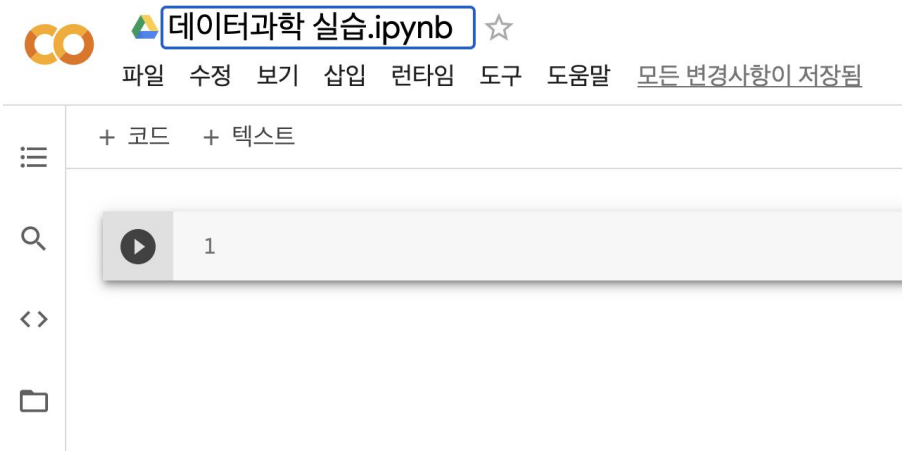
구글 드라이브 사용시 항상 절대경로 사용

구글 코랩 새 노트 만들기



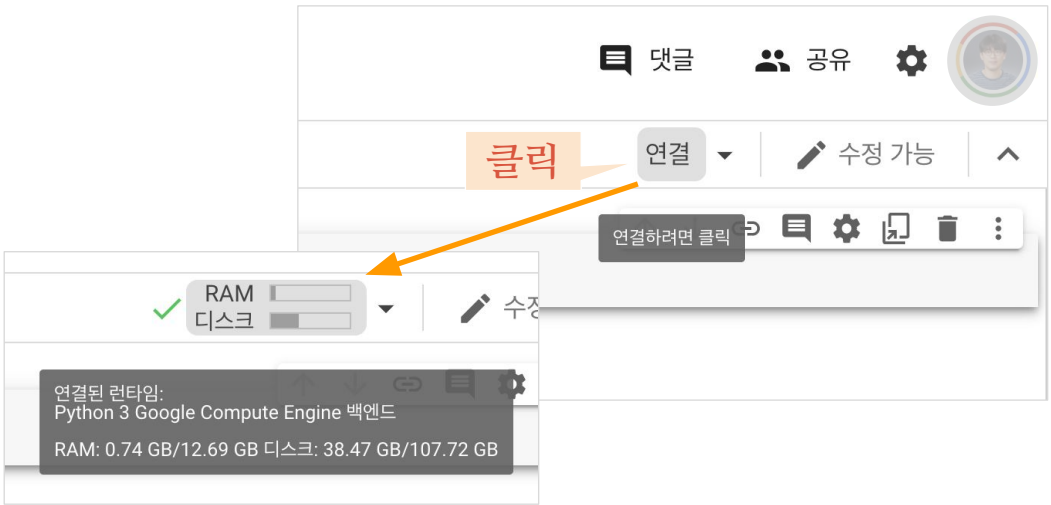
구글 코랩

노트 파일 이름 변경



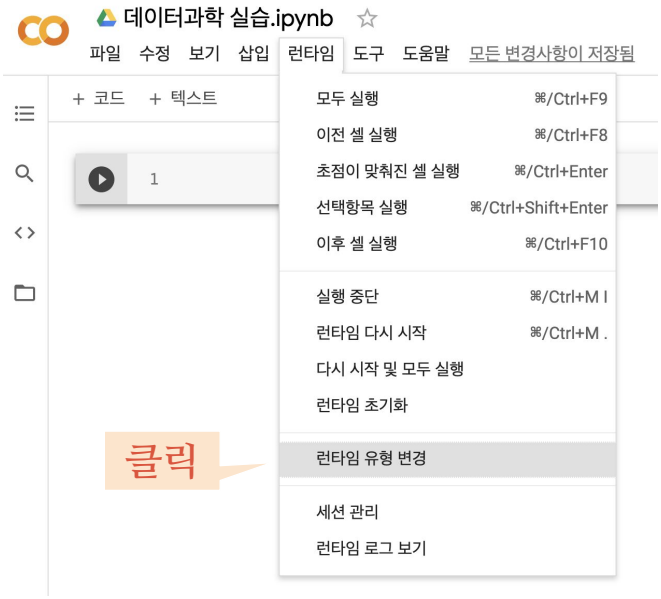
구글 코랩

코랩 런타임 연결



구글 코랩

런타임 유형 변경

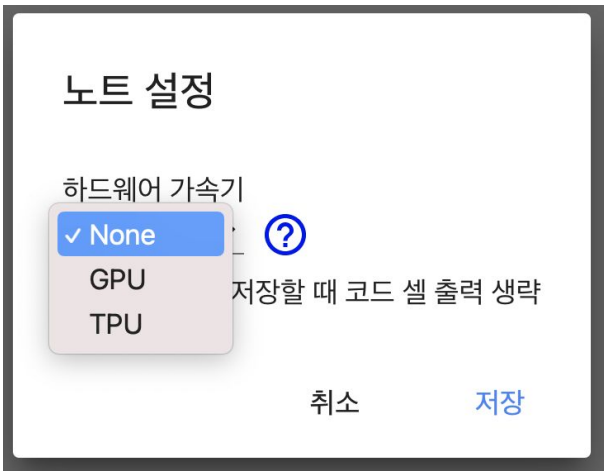


구글 코랩

하드웨어 가속기 선택

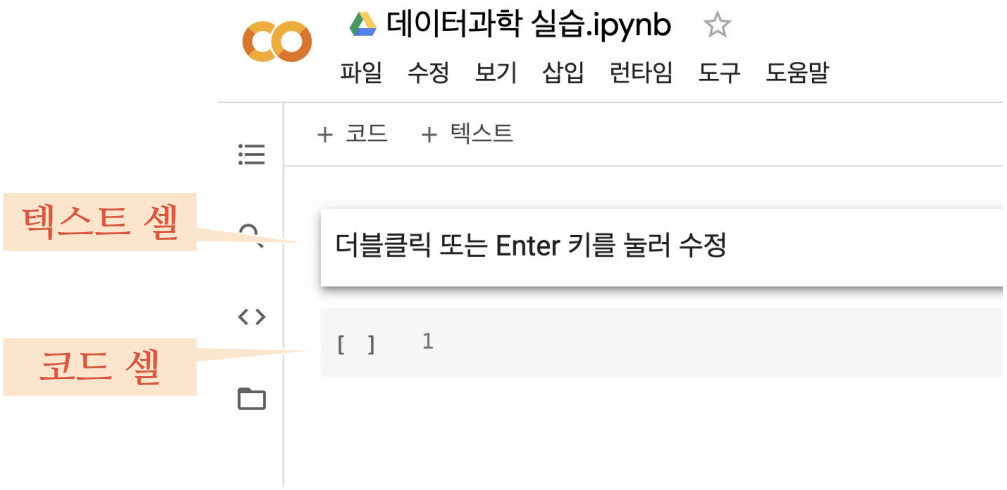
머신러닝(딥러닝) 코드를 실행할 때
GPU 혹은 TPU 중 하나를 선택

※ 일반 파이선 코드 실행할 때는 반드시
선택해야하는 것은 아님



구글 코랩

노트의 구성



구글 코랩

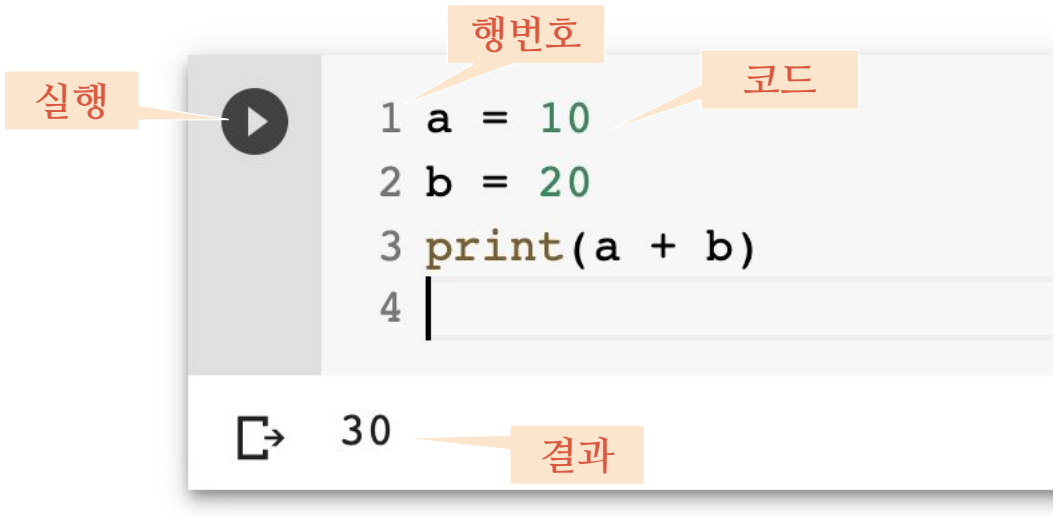
텍스트 셀

https://colab.research.google.com/notebooks/markdown_guide.ipynb

Markdown	Preview
`**bold text**`	**bold text**
`*italicized text*` or `_italicized text_`	*italicized text*
`` `Monospace` ``	`Monospace`
`~~strikethrough~~`	~~strikethrough~~
`[A link](https://www.google.com)`	A link
`![An image](https://www.google.com/images/rss.png)`	

구글 코랩

코드 셀



코딩없이 인공지능 데이터 분석

orange

오렌지 https://orangedatamining.com/

orange

Data Mining
Fruitful and Fun

Open source machine learning and data visualization.

Build data analysis workflows visually, with a large, diverse toolbox.

Download Orange

클릭

오렌지 다운로드 후 설치

Windows

macOS

Download the latest version for Windows

Download Orange 3.29.3

클릭

Standalone installer (default)

Orange3-3.29.3-Miniconda-x86_64.exe (64 bit)
Can be used without administrative priviledges.

오렌지

오렌지 활용 예제(분류)

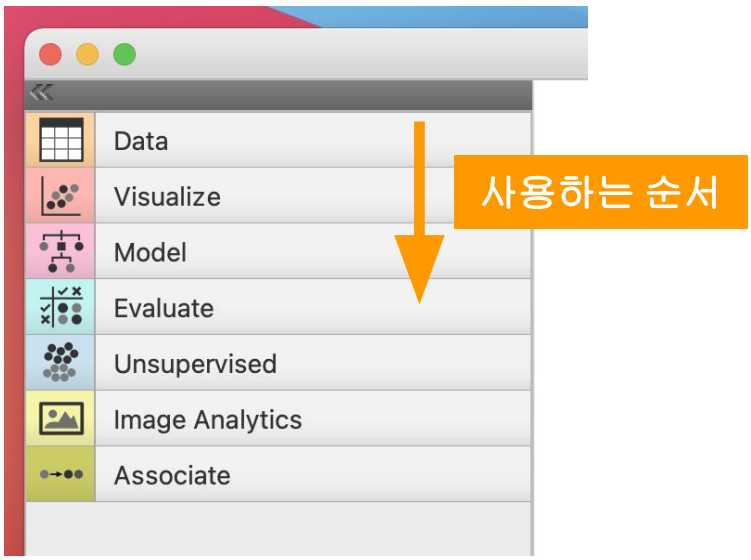
데이터 파일 준비

https://bit.ly/이직예측

다운로드 후 압축 해제!

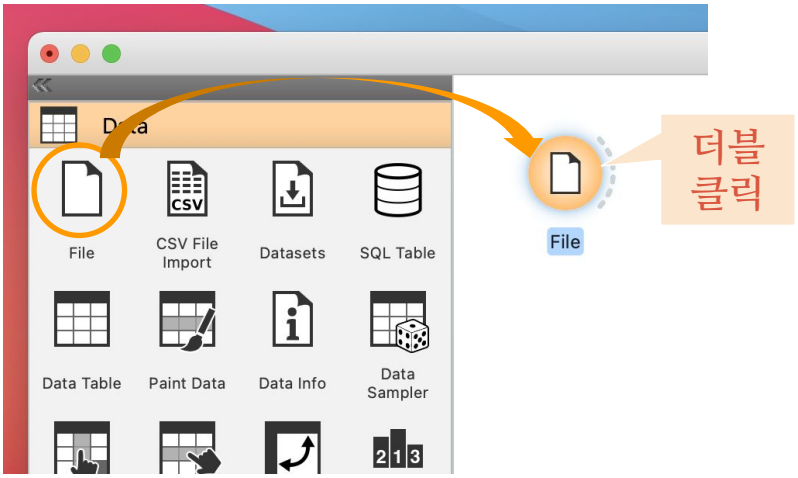
오렌지

오렌지 활용 예제(분류)

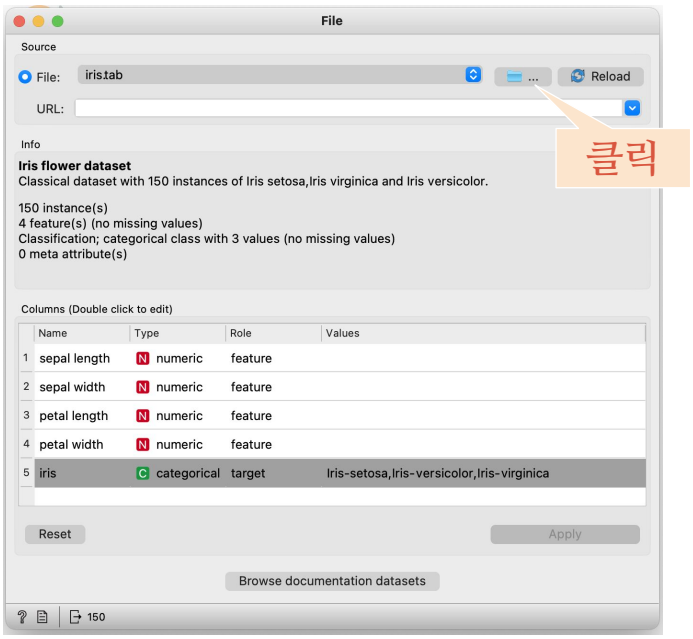


오렌지

오렌지 활용 예제(분류)

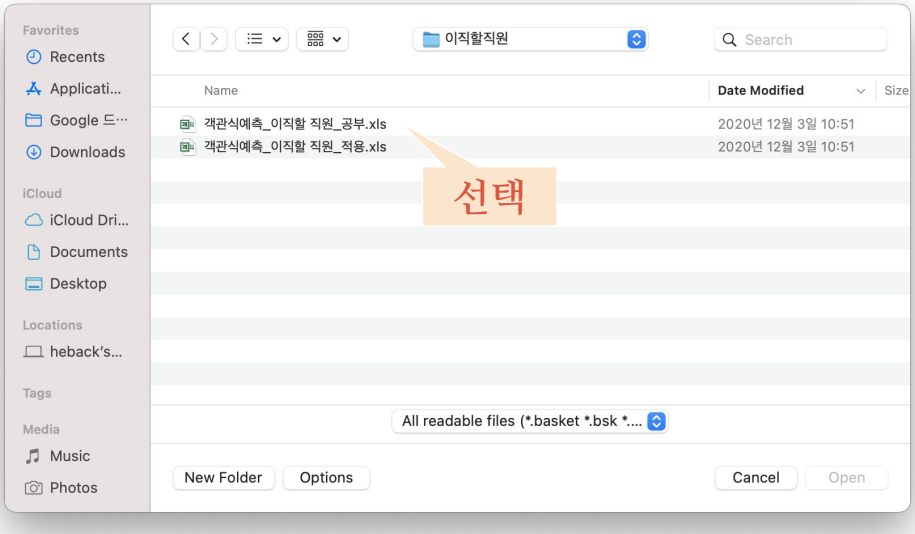


오렌지

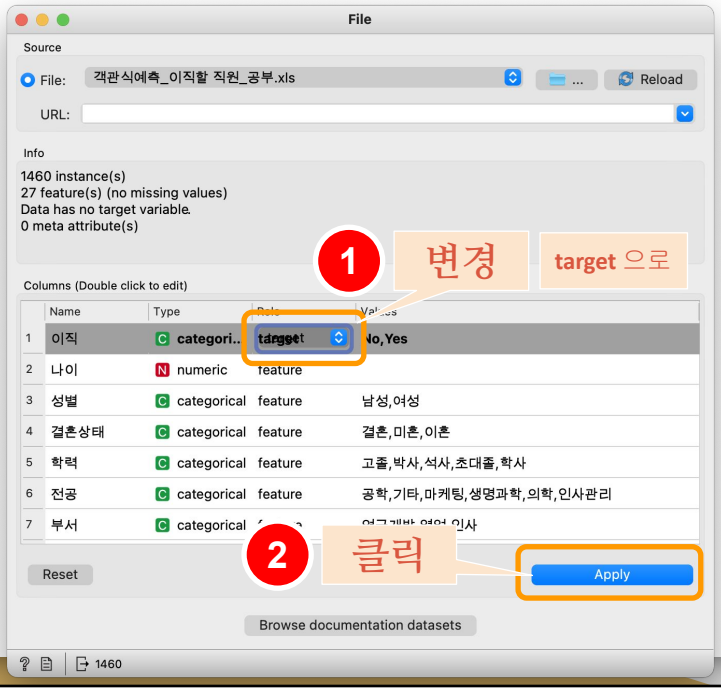


오렌지

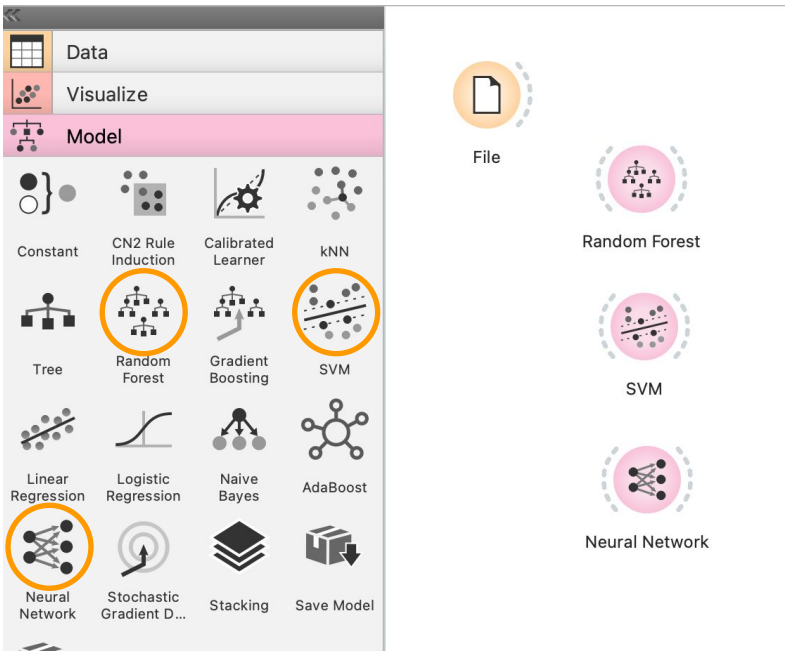
오렌지 활용 예제(분류)



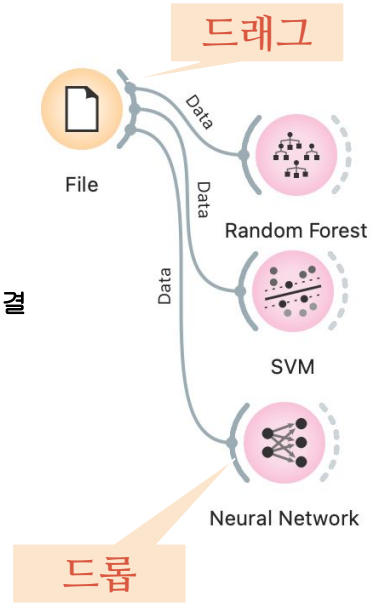
오렌지



오렌지

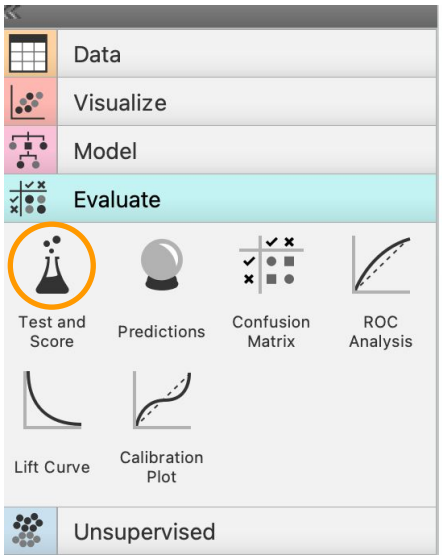


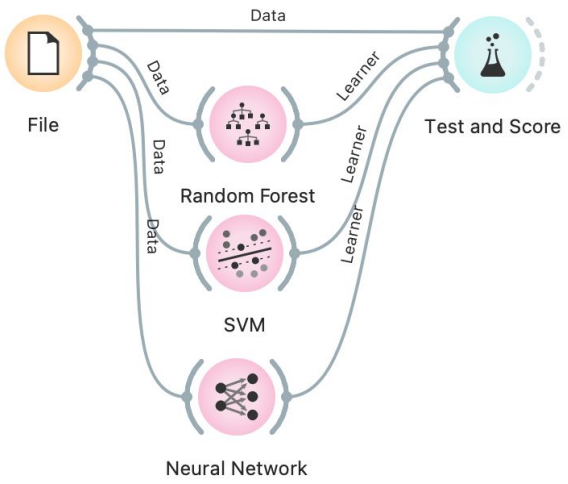
오렌지



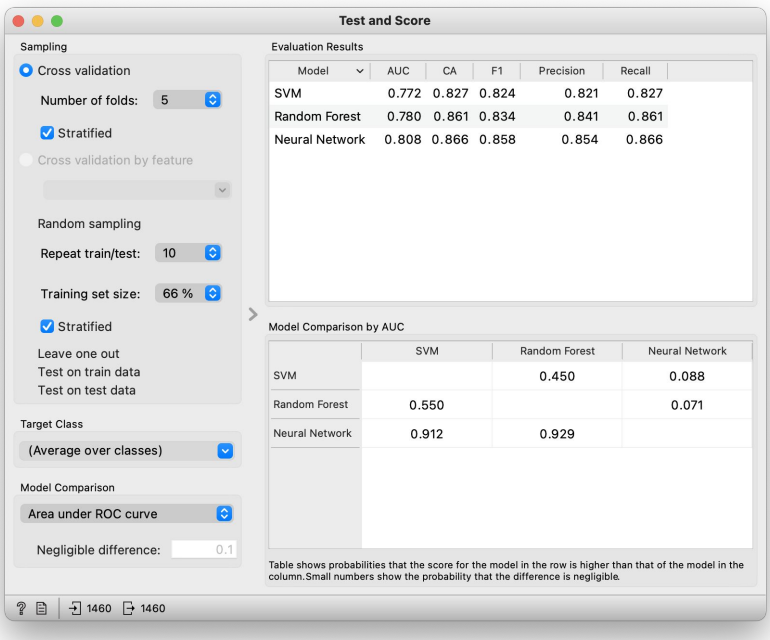
데이터와 모델을 연결

오렌지


Data
Visualize
Model
Evaluate
Test and Score
Predictions
Confusion Matrix
ROC Analysis
Lift Curve
Calibration Plot
Unsupervised


Data
File
Data
Learner
Test and Score
Data
Learner
Random Forest
Data
Learner
SVM
Neural Network

오렌지


Test and Score
Sampling
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
Stratified
Leave one out
Test on train data
Test on test data
Target Class
(Average over classes)
Model Comparison
Area under ROC curve
Negligible difference: 0.1
Evaluation Results
Model AUC CA F1 Precision Recall
SVM 0.772 0.827 0.824 0.821 0.827
Random Forest 0.780 0.861 0.834 0.841 0.861
Neural Network 0.808 0.866 0.858 0.854 0.866
Model Comparison by AUC
SVM Random Forest Neural Network
SVM 0.450 0.088
Random Forest 0.550 0.071
Neural Network 0.912 0.929
Table shows probabilities that the score for the model in the row is higher than that of the model in the column.Small numbers show the probability that the difference is negligible.
1460
1460

데이터과학 교과서 분석 및 실습

데이터 프로그래밍

1. 데이터 과학의 이해
2. 데이터 분석 프로그래밍

1 데이터 과학의 이해

01 데이터 과학과 진로

데이터 과학은 데이터를 탐구하여 이로부터 의미 있는 정보를 추출해 내는 학문이라고 할 수 있어.

학습 목표

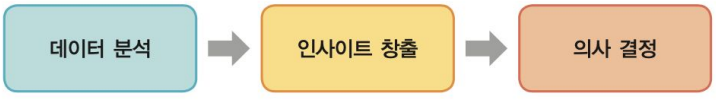
- 데이터 과학의 의미를 알고, 활용되는 기술과 분야를 설명할 수 있다.
- 데이터 과학 관련 분야의 직업과 진로를 탐색할 수 있다.

1. 데이터 과학의 개념

1 데이터 과학이란

데이터 과학(Data Science)은 데이터를 탐구하고 분석하여 그 안에서 체계, 즉 패턴을 발견하고 이를 활용하여 현실의 문제를 해결해 나가는 학문이다. 이에 데이터 과학은 데이터를 다루기 위해 프로그래밍 언어를 활용하고, 또 알고리즘을 적용하기 위해 컴퓨팅 기술을 사용한다. 그리고 데이터를 분석하기 위해서는 수학과 통계 기법 등 과학적 원리를 적용한다.

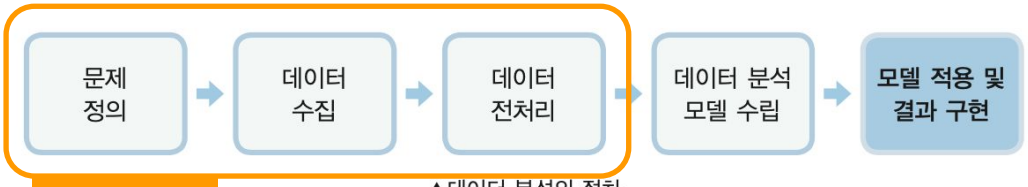
바야흐로 데이터 시대로 접어들면서 데이터 과학에 대한 관심이 높아지고 있다. 데이터 과학의 목적은 데이터 분석을 통해 지식과 *인사이트를 창출하고, 이를 통해 올바른 의사 결정을 할 수 있도록 도움을 주는 것이다.



▲데이터 과학 프로세스

❷ 데이터 분석의 절차

데이터 분석은 다음과 같은 단계로 이루어진다. 데이터 분석 과정을 통해 데이터 과학이 어떻게 적용되는지 알아보자.

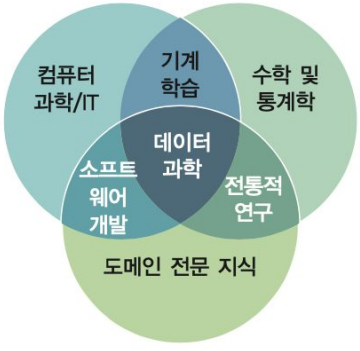


▲데이터 분석의 절차

키 포인트

2. 데이터 과학의 기술 분야

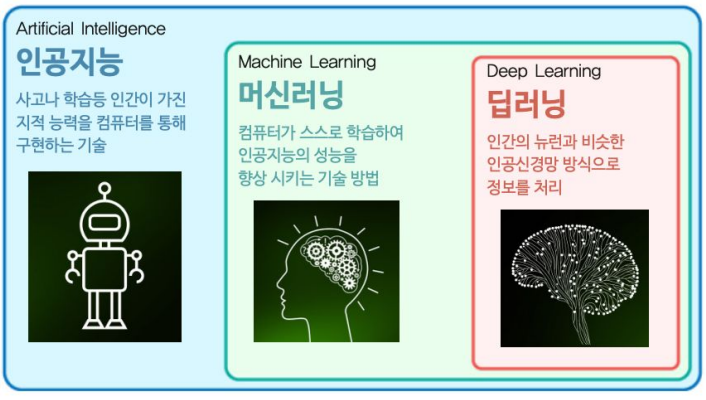
데이터 과학은 융합 영역을 다루므로 필요한 기술과 분야는 다양하다. 컴퓨팅 기술과 수학 및 통계학적 지식도 필요하며, 다루려고 하는 해당 내용에 대한 전문적인 지식도 갖춰야 한다.



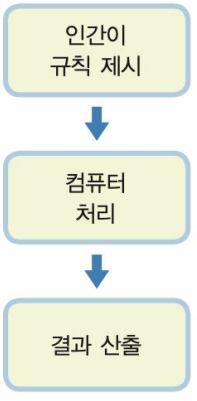
▲ 데이터 과학 벤다이어그램

문제 정의	데이터 수집	데이터 전처리	데이터 분석 모델 수립	모델 적용 및 결과 구현	인사이트 창출	평가
의료, 금융, 제조, 교통 등 해당 분야의 전문 지식	해당 분야의 전문 지식, 컴퓨팅 기술	컴퓨팅 기술(R, 파이선, SQL 등)	통계 도구 및 분석 알고리즘 활용 기술(수학, 통계, 머신러닝, 딥러닝 등)	컴퓨팅 기술, 데이터 시각화 기술	해당 분야의 전문 지식	의사소통 기술, 프레젠테이션 기술

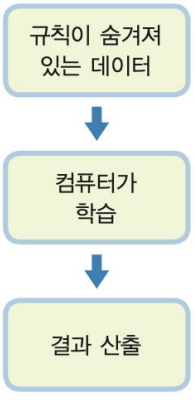
▲데이터 과학 절차에 따라 필요한 기술



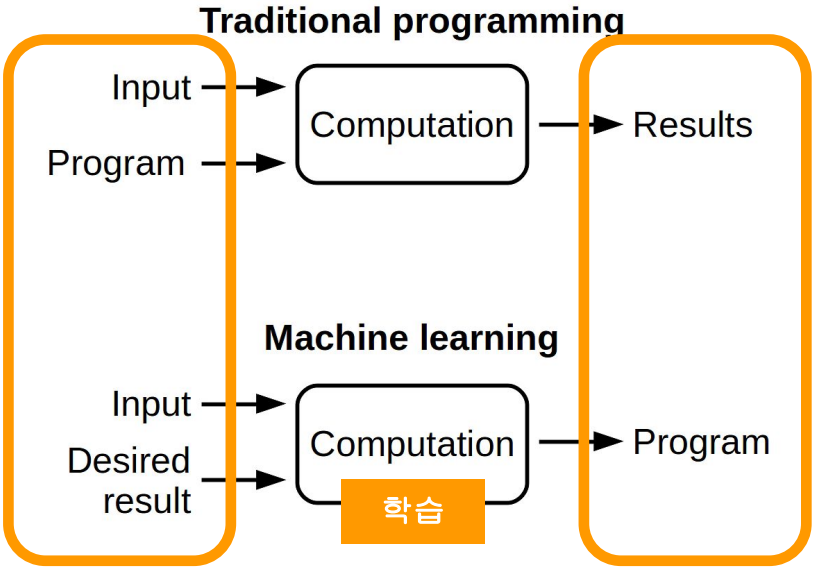
▲인공지능, 머신러닝, 딥러닝 관계



▲~~컴퓨터 과학 프로세스~~
전통적 프로그래밍



▲머신러닝 프로세스



3. 데이터 과학자

최근 데이터 분석에 대한 관심이 높아지고, 분석 기술에 대한 수요가 늘어나면서 *데이터 과학자가 새롭게 주목받고 있다. 다수의 경제 전문가들은 21세기 가장 유망한 직업으로 데이터 과학자를 꼽고 있다. 데이터 과학자는 어떤 일을 하며, 어떤 능력을 갖추어야 할까?

데이터사이언티스트, 10년 후에는 사라진다는데..	데이터 사언티스트의 현실
수많은 데이터 사이언티스트들이 직장을 떠나는 이유는 무엇인가?	

02 데이터의 이해

학습 목표

- 데이터의 의미를 이해한다.
- 정형 데이터, 반정형 데이터, 비정형 데이터를 비교하여 설명할 수 있다.
- 관계형 데이터베이스 관리 시스템의 특징을 알고 SQL 언어를 이해한다.

03 데이터 보안

학습 목표

- 개인 정보 침해 사례를 탐색하고, 데이터 보안의 필요성을 설명할 수 있다.
- 사용하는 컴퓨팅 기기에서 데이터 보안을 계획하고 실천할 수 있다.

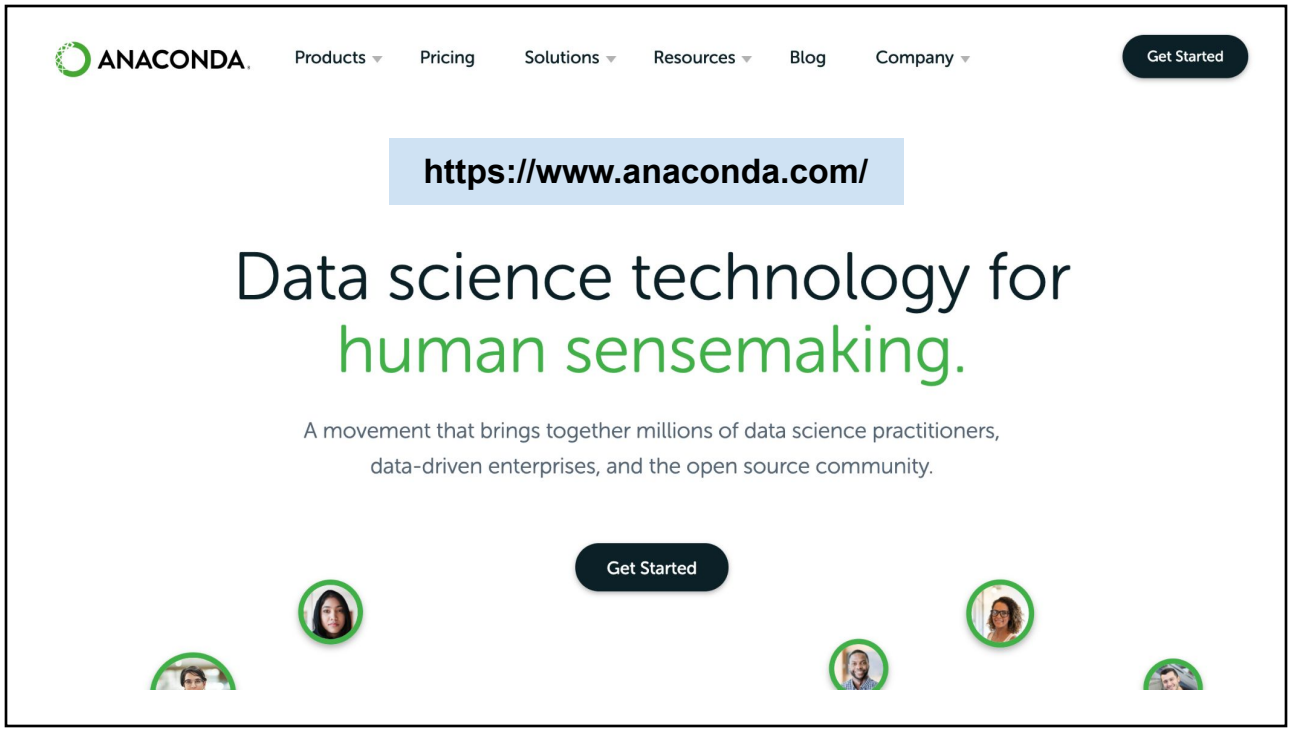
인공지능 윤리가 더 심각함 https://youtu.be/i9YcpNxXJpQ

2 데이터 분석 프로그램

01 통합 개발 환경

학습 목표

- 파이선 언어의 통합 개발 환경을 이해할 수 있다.
- 데이터 분석을 위한 필수 라이브러리를 설치할 수 있다.


ANACONDA.
Products
Pricing
Solutions
Resources
Blog
Company
Get Started
https://www.anaconda.com/
Data science technology for
human sensemaking.
A movement that brings together millions of data science practitioners,
data-driven enterprises, and the open source community.
Get Started


JET BRAINS
개발자 도구
팀 도구
학습 도구
솔루션
스토어
PyCharm
2021.2 출시 예정
새로운 기능
기능
자세히 알아보기
구매
다운로드
https://www.jetbrains.com/ko-kr/pycharm/
PC
PyCharm
전문 개발자용
Python IDE
다운로드
완벽한 기능을 갖춘 Professional 또는 무료 Community

02 데이터 분석을 위한 파이선의 기초

학습 목표

- 데이터 분석을 위한 파이선의 기초적인 문법을 이해할 수 있다.
- 공공 데이터를 수집하고 원하는 데이터를 출력할 수 있다.

Ⅱ 데이터 과학을 위한 통계

1. 데이터 탐색
2. 데이터 분석과 예측
3. 모델 평가

1 데이터 탐색

01 데이터를 대표하는 값

학습 목표

- 데이터 과학과 통계의 관계를 설명할 수 있다.
- 평균, 중앙값, 최빈값의 개념을 설명할 수 있다.
- 주어진 데이터에 대하여 파이선으로 평균, 중앙값, 최빈값을 계산하는 프로그램을 만들 수 있다.

02 데이터의 흩어짐 측정

학습 목표

- 산포도의 개념을 설명할 수 있다.
- 파이선으로 범위, 사분위 범위, 분산, 표준편차를 계산하는 프로그램을 만들 수 있다.

2 데이터 분석과 예측

01 통계적 추정

대푯값과 산포도만으로 데이터를 예측할 수 있을까?

학습 목표

- 확률과 확률분포의 개념을 예시를 통해 설명할 수 있다.
- 대푯값과 산포도를 통해 통계적 추정을 할 수 있다.

02 상관관계

두 변수 간의 관계를 어떻게 나타낼 수 있을까?

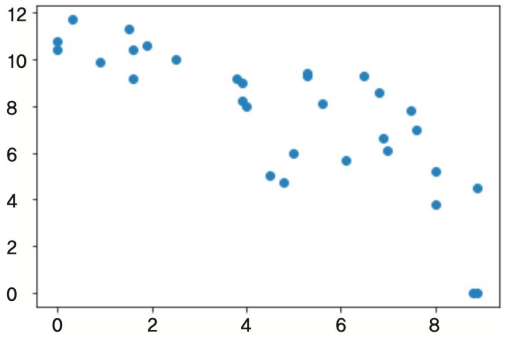
학습 목표

- 두 변수 간의 관계를 시각화하여 표현할 수 있다.
- 상관관계와 상관계수의 개념을 이해하고, 상관계수가 나타내는 의미를 설명할 수 있다.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding =
"cp949")
plt.scatter(df['평균 전운량(1/10)'], df['합계 일조 시간(hr)'])
```

실행 결과



[결과 해석]
데이터의 산점도를 보면 평균 전운량이 많은 날일수록 일조 시간은 짧아지는 경향을 확인할 수 있다. 사실 구름의 양과 해가 비추는 시간이 밀접한 관련이 있음은, 굳이 데이터를 확인하지 않더라도 어느 정도 짐작할 수 있을 것이다.

```
import pandas as pd

df=pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding="cp949")
print("피어슨 상관계수:", df['평균 전운량(1/10)'].corr(df['합계 일조 시간(hr)'],
method="pearson"))
print("스피어만 상관계수:", df['평균 전운량(1/10)'].corr(df['합계 일조
시간(hr)'], method="spearman"))
print("켄달 상관계수 :", df['평균 전운량(1/10)'].corr(df['합계 일조 시간(hr)'],
method="kendall"))
```

실행 결과

- 피어슨 상관계수: -0.7876136446325478
- 스피어만 상관계수: -0.8190716614835573
- 켄달 상관계수: -0.6543493722829523

03 회귀분석

학습 목표

- 두 변수 간의 관계를 직선을 이용해서 나타낼 수 있다.
- 회귀분석을 위한 최소제곱법의 원리를 설명할 수 있다.
- 파이선 프로그래밍을 통해 주어진 데이터에 대한 회귀분석을 할 수 있다.
- 회귀분석 결과를 바탕으로 주어진 독립변수에 대한 종속변수를 예측할 수 있다.

2019년 3월 대구광역시의 평균 전운량과 일조 시간 데이터에 최소제곱법이 적용된 회귀분석을 통하여 계수(b_0, b_1)를 구해 보자.

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

df = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")

x = np.array(df['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df['합계 일조 시간(hr)']).reshape(-1, 1)

model = LinearRegression().fit(x, y)
print("기울기 :", model.coef_, "절편 :", model.intercept_)
```

실행 결과

- 기울기: [[-0.83737793]]
- 절편: [11.58804862]

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
df = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")

x = np.array(df['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df['합계 일조 시간(hr)']).reshape(-1, 1)

model = LinearRegression().fit(x, y)
cloudy = 3
b0 = model.intercept_[0]
b1 = model.coef_[0][0]
print('평균 전운량 :', cloudy, '일조 시간 :', b0 + b1 * cloudy)
print('평균 전운량 :', cloudy, '일조 시간 :', model.predict([[cloudy]]))
```

01 성능 나타내기

학습 목표

- 주어진 모델의 잔차를 파이선을 통해 시각화할 수 있다.
- 주어진 모델에 대한 성능지표(평균제곱오차, 결정계수)를 구할 수 있다.

2 평균제곱오차와 결정계수

모델의 성능을 평가할 수 있는 지표로 평균제곱오차(Mean Squared Error)와 결정계수가 있다. 평균제곱오차는 각 데이터에 대한 잔차 제곱의 평균을 구한 것으로 일반적으로 값이 작을수록 모델의 성능이 뛰어남을 의미하는데, 이는 최소제곱법의 관점이 평가에 적용된 것이다.

결정계수(R^2)는 모델을 통해 설명할 수 있는 데이터의 비율을 나타내는 지표로 모델의 평균제곱오차 값과 종속변수의 분산 값을 이용하여 계산한다. 0 이상 1 이하의 값으로 표현되며, 1에 가까울수록 모델의 성능이 뛰어남을 의미한다. 파이선에서 최소제곱법을 통해 구한 선형회귀 모델의 성능을 알아보자.

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

df = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding =
"cp949")
x = np.array(df['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df['합계 일조 시간(hr)']).reshape(-1, 1)

model = LinearRegression().fit(x, y)

print("평균제곱오차:", mean_squared_error(y, model.predict(x)))
print("결정계수:", r2_score(y, model.predict(x)))
```

02 평가 방법 이해·적용하기

주어진 데이터를 효과적으로 이용하려면 어떻게 해야 할까?

학습 목표

- 홀드아웃 교차 검증법에 대하여 설명할 수 있다.
- k-분할 교차 검증법에 대하여 설명할 수 있다.
- 파이선에서 주어진 모델에 테스트용 데이터를 적용하여 그 결과를 설명할 수 있다.

1. 홀드아웃 교차 검증법

우리는 앞에서 최소제곱법을 적용한 회귀분석으로 두 변수 간의 관계를 나타내는 선형 모델을 얻었으며, 그 모델의 성능을 평가할 수 있는 지표도 계산해 보았다. 더 나아가 머신러닝이나 딥러닝에서는 학습용 데이터와 시험용 데이터를 구분함으로써 학습용 데이터로 학습된 모델을 시험용 데이터로 검증하는 과정을 통해 모델의 성능을 점차적으로 개선해 나간다.

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

#회귀분석 모델을 얻기 위한 데이터
df1 = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding = 'cp949')
x = np.array(df1['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df1['합계 일조 시간(hr)']).reshape(-1, 1)

#테스트용 데이터
df2 = pd.read_csv('2019_05_Daegu_Sunshine_Cloudiness.csv', encoding = 'cp949')
x_test = np.array(df2['평균 전운량(1/10)']).reshape(-1, 1)
y_test = np.array(df2['합계 일조 시간(hr)']).reshape(-1, 1)
model = LinearRegression().fit(x, y) #회귀분석 모델
plt.scatter(model.predict(x), model.predict(x)-y)
plt.scatter(model.predict(x_test), model.predict(x_test)-y_test,
            c='orange', marker='s')
#테스트용 데이터에 기존의 모델 적용
plt.hlines(y=0, xmin=2, xmax=13, color='red')
```

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

df1 = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")
x = np.array(df1['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df1['합계 일조 시간(hr)']).reshape(-1, 1)

df2 = pd.read_csv('2019_05_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")
x_test = np.array(df2['평균 전운량(1/10)']).reshape(-1, 1)
y_test = np.array(df2['합계 일조 시간(hr)']).reshape(-1, 1)

model = LinearRegression().fit(x, y)

plt.scatter(model.predict(x), model.predict(x)-y)
plt.scatter(model.predict(x_test), model.predict(x_test)-y_test,
            c='orange', marker='s')
plt.hlines(y=0, xmin=2, xmax=13, color='red')

print("평균제곱오차 :", mean_squared_error(y, model.predict(x)))
print("결정계수 :", r2_score(y, model.predict(x)))

print("테스트용 데이터에 대한 평균제곱오차 :", mean_squared_error(y_test,
                                                  model.predict(x_test)))
print("테스트용 데이터에 대한 결정계수 :", r2_score(y_test, model.predict(x_test)))

```

2. K-분할 교차 검증법

홀드아웃 교차 검증법을 사용한다고 하더라도, 학습용 데이터와 시험용 데이터가 어떻게 분할되느냐에 따라 모델의 성능이 달라질 수 있다. 이는 만약 [실습 18]의 상황에서 기존 3월 데이터에 5월의 보름 정도 데이터를 추가하여 회귀모델을 얻었다면, 나머지 5월의 데이터를 더 잘 설명할 수도 있었음을 의미한다.

이와 같은 불확실성을 바탕으로, 주어진 데이터를 최대한 활용하여 더 좋은 모델을 얻고자 홀드아웃 교차 검증법의 개념을 확장시킨 것이 K-분할 교차 검증법(K-fold cross-validation)이다.

K-분할 교차 검증법은 주어진 데이터 셋을 K개로 분할한 다음, K번에 걸쳐 학습용 데이터와 시험용 데이터의 조합을 바꿔가며 모델을 검증한다. 각각의 데이터가 어떤 한 번의 조합에서 시험용으로 사용되며, 모델의 평가 지표들을 평균과 같은 방식으로 결합함으로써 모델의 일반화 가능성을 높이는 방법이다.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

df1 = pd.read_csv('2019_03_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")
df2 = pd.read_csv('2019_05_Daegu_Sunshine_Cloudiness.csv', encoding = "cp949")
df = pd.concat([df1, df2])
#print(df)

model = LinearRegression()
x = np.array(df['평균 전운량(1/10)']).reshape(-1, 1)
y = np.array(df['합계 일조 시간(hr)']).reshape(-1, 1)
score = cross_val_score(model, x, y, scoring="r2", cv=2)
cv1 = KFold(2, shuffle=True)
score2 = cross_val_score(model, x, y, scoring="r2", cv=cv1)
print("순서 고정 분할에 따른 결정계수 :", score, " 결정계수 평균 :", score.mean())
print("순서 랜덤 분할에 따른 결정계수 :", score2, " 결정계수 평균 :", score2.mean())
```

Ⅲ 머신러닝

1. 머신러닝 이해
2. 머신러닝 모델
3. 머신러닝 핵심 알고리즘
4. 머신러닝 모델 적용

1 머신러닝 이해

01 머신러닝의 발달

머신러닝 기계의 연료는 바로 데이터야.

학습 목표

- 머신러닝 기술의 탄생과 발전 과정을 탐색한다.
- 인공지능과 튜링 테스트의 관계를 설명할 수 있다.

02 머신러닝의 활용 분야

학습 목표

- 문제 해결에 머신러닝이 필요한 이유를 설명할 수 있다.
- 실생활에서 머신러닝의 활용 사례를 탐색한다.

03 머신러닝의 정의

학습 목표

- 머신러닝의 정의를 이해한다.
- 머신러닝의 정의를 예시에 적용하여 설명할 수 있다.

2 머신러닝 모델

01 지도 학습

메일이
스팸인지 아닌지를
어떻게 알까?

학습 목표

- 머신러닝의 모델 분류 방법에 대해 이해한다.
- 머신러닝의 지도 학습 모델을 이해하고, 활용 분야를 탐색한다.

02 비지도 학습

정답이 없는데
어떻게 학습하지?

학습 목표

- 머신러닝의 비지도 학습 모델을 이해한다.
- 머신러닝 비지도 학습 모델의 활용 분야를 탐색한다.

03 강화 학습

우리가 즐겨하는 게임은 어떤 모델로 만드는지 아니?

학습 목표

- 머신러닝의 강화 학습에 대해 이해한다.
- 머신러닝 강화 학습의 활용 분야를 탐색한다.

3 머신러닝 핵심 알고리즘

01 머신러닝의 과정

머신러닝 모델을 만드는 절차가 궁금한데.

학습 목표

- 머신러닝을 통한 문제 해결 방법의 특징을 이해한다.
- 머신러닝을 구현하는 전체 과정을 이해한다.

02 경사하강법

수포자(수학 포기자)도 머신러닝을 배울 수 있을까?

학습 목표

- 경사하강법의 핵심 원리를 단계별로 적용할 수 있다.
- 경사하강법을 실생활에 적용할 수 있다.

4 머신러닝 모델 적용

01 회귀 모델

주식 가격 예측이 대표적인 회귀 모델이지

학습 목표

- 단순 선형회귀 모델을 이용한 문제 해결 절차를 이해할 수 있다.
- 다중 선형회귀 모델을 이용한 문제 해결 절차를 이해할 수 있다.

02 분류 모델

스팸 필터가
분류 모델이라구?

학습 목표

- 이진 분류와 다중 분류를 구분하여 설명할 수 있다.
- 분류 모델을 이용한 문제 해결 절차를 이해할 수 있다.

Ⅳ 딥러닝

1. 딥러닝의 이해
2. 딥러닝 모델링

1 딥러닝의 이해

01 딥러닝의 개요

딥러닝은 인간의 뉴런처럼 퍼셉트론이라는 신경망 기본 구조를 이용해 인간의 두뇌를 흉내 낸 인공지능이야!

학습 목표

- 딥러닝의 개념과 딥러닝 알고리즘의 종류와 특징, 활용 분야를 이해할 수 있다.
- 딥러닝의 코드를 실행하고 딥러닝의 전체적인 코드 구조를 이해한다.

02 퍼셉트론

인공 신경망의 퍼셉트론은 어떻게 정보를 전달할까?

학습 목표

- 인공 신경망의 기본 단위인 퍼셉트론의 개념을 알 수 있다.
- 가중치, 바이어스, 활성화 함수의 개념을 알 수 있다.
- 퍼셉트론의 한계에 대해 이해할 수 있다.

03 다층 퍼셉트론

학습 목표

- 다층 퍼셉트론을 이용한 XOR 문제 해결 방법을 설명할 수 있다.
- 다층 퍼셉트론의 구조를 이해하고, 코드로 표현할 수 있다.

04 오차역전파

학습 목표

- 오차역전파의 개념에 대해 설명할 수 있다.
- 활성화 함수의 종류를 이해하고 문제 상황에 맞는 활성화 함수를 코드로 표현 할 수 있다.
- 손실 함수와 최적화 기법의 개념을 이해하고 문제 상황에 맞는 손실 함수와 최적화 기법을 코드로 표현할 수 있다.

2 딥러닝 모델링

01 딥러닝 모델링 개념

딥러닝의 성능을 향상시키기 위해 필요한 것은 무엇일까?

학습 목표

- 딥러닝과 데이터의 관계를 이해하고 데이터 가공이 필요한 이유를 설명할 수 있다.
- 딥러닝 모델 만들기의 전체 흐름을 알 수 있다.

02 DNN 모델 만들기

딥러닝에서 학습을 한다는 것은 무엇을 의미할까?

학습 목표

- 실생활 문제를 해결하기 위한 딥러닝 모델 설계 과정을 알 수 있다.
- 이진 분류 문제를 해결하기 위한 심층신경망 모델을 설계할 수 있다.

03 다중 분류 문제 해결 모델 만들기

학습 목표

- 다중 분류 문제를 해결하기 위한 심층신경망 모델을 설계할 수 있다.
- 다중 분류 문제 해결에 적절한 활성화 함수와 문자열 변환 기법을 이해할 수 있다.

04 과적합 방지하기

학습 목표

- 과적합의 개념을 이해하고 과적합을 해결하기 위한 방법을 적용한 딥러닝 모델을 설계할 수 있다.
- 학습 데이터를 학습셋과 테스트 셋으로 구분하는 장점을 이해하고 분할할 수 있다.

05 학습 자동 중단하기

학습 목표

- 학습 자동 중단 기능을 적용한 딥러닝 모델을 설계할 수 있다.
- 학습 자동 중단 기능을 이용하여 베스트 모델을 만들 수 있다.

06 이미지 분류하기

학습 목표

- MNIST 데이터셋을 불러와 학습셋과 테스트셋으로 분류한다.
- 이미지 분류를 위해 완전 연결 신경망을 사용하여 딥러닝 모델링을 수행한다.