컴퓨팅 사고와 인공지능
실습2

튜터 신연우
튜터 송진하
튜터 이채연

Lecture program

- 1st Lecture: Introduction
- 2nd Lecture: Matrix and Vector
- 3rd Lecture: Perceptron
- 4th Lecture: Understand Learning
- 5th Lecture: Machine Learning
- 6th Lecture: Learning Techniques
- 7th Lecture: Deep Learning

8th Practice: Deep Learning Library

9th Practice: Python Programming

10th Practice: Deep Learning Programming 1

11th Practice: Deep Learning Programming 2

12th Lecture: Generative AI

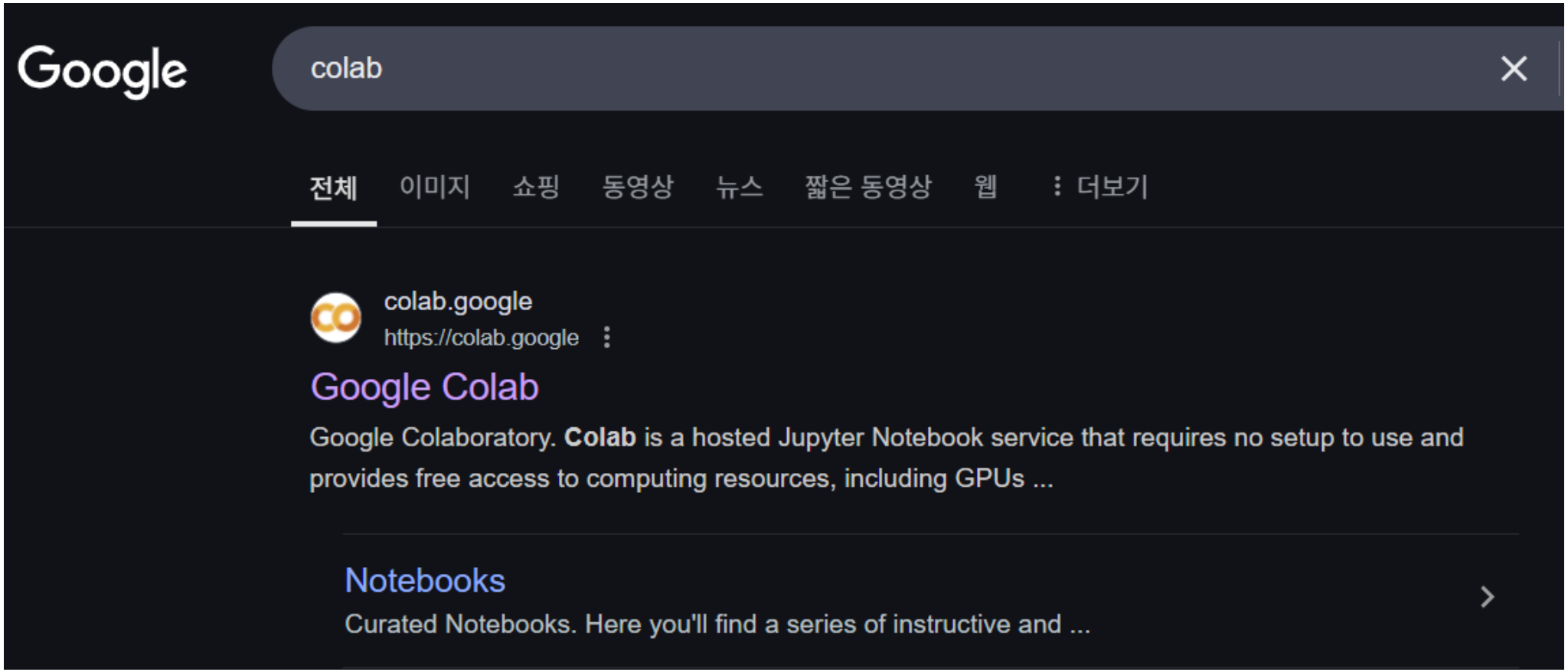
13th Practice: AI and Ethics

14th Lecture: Final Term Exam

15th Lecture: Make-up Lecture

Google Colab Usage

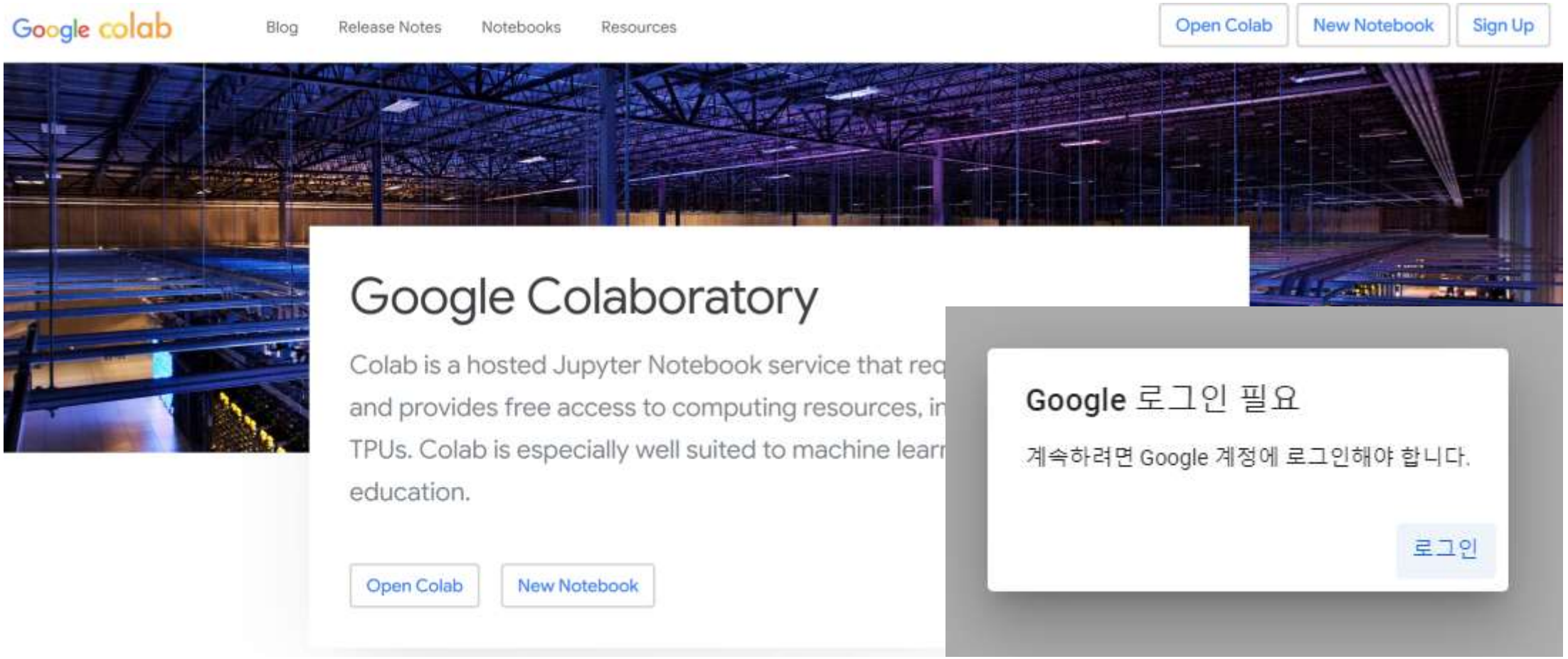
Google Colab Usage



코랩을 쓰는 가장 큰 이유는 프로그램을 따로 설치할 필요 없고,
구글 계정만 있으면 GPU까지 무료로 쓸 수 있다는 장점이 있습니다.

코랩을 실행하는 방법은
(1) 구글에 colab 혹은 코랩을 검색합니다.

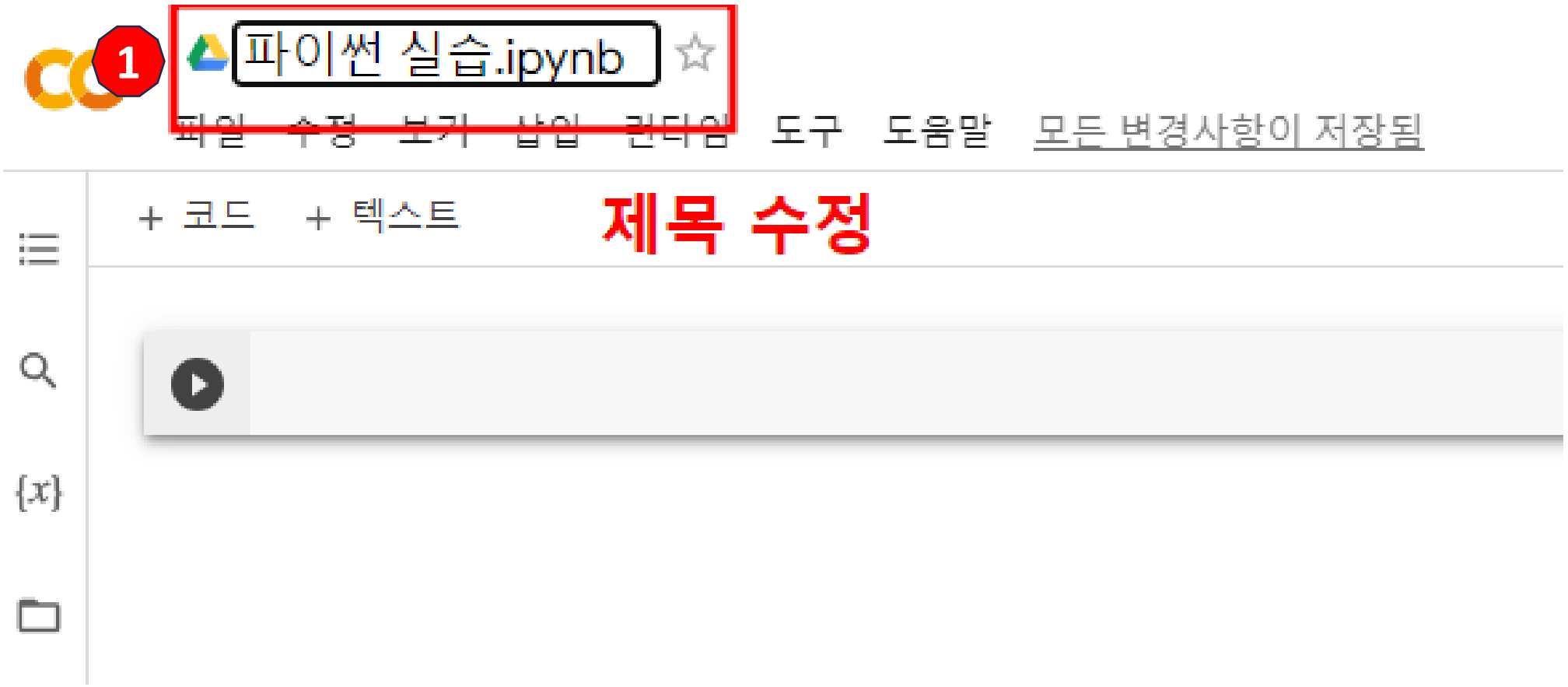
Google Colab Usage



(2) 들어가면 이런 시작 화면이 뜹니다.

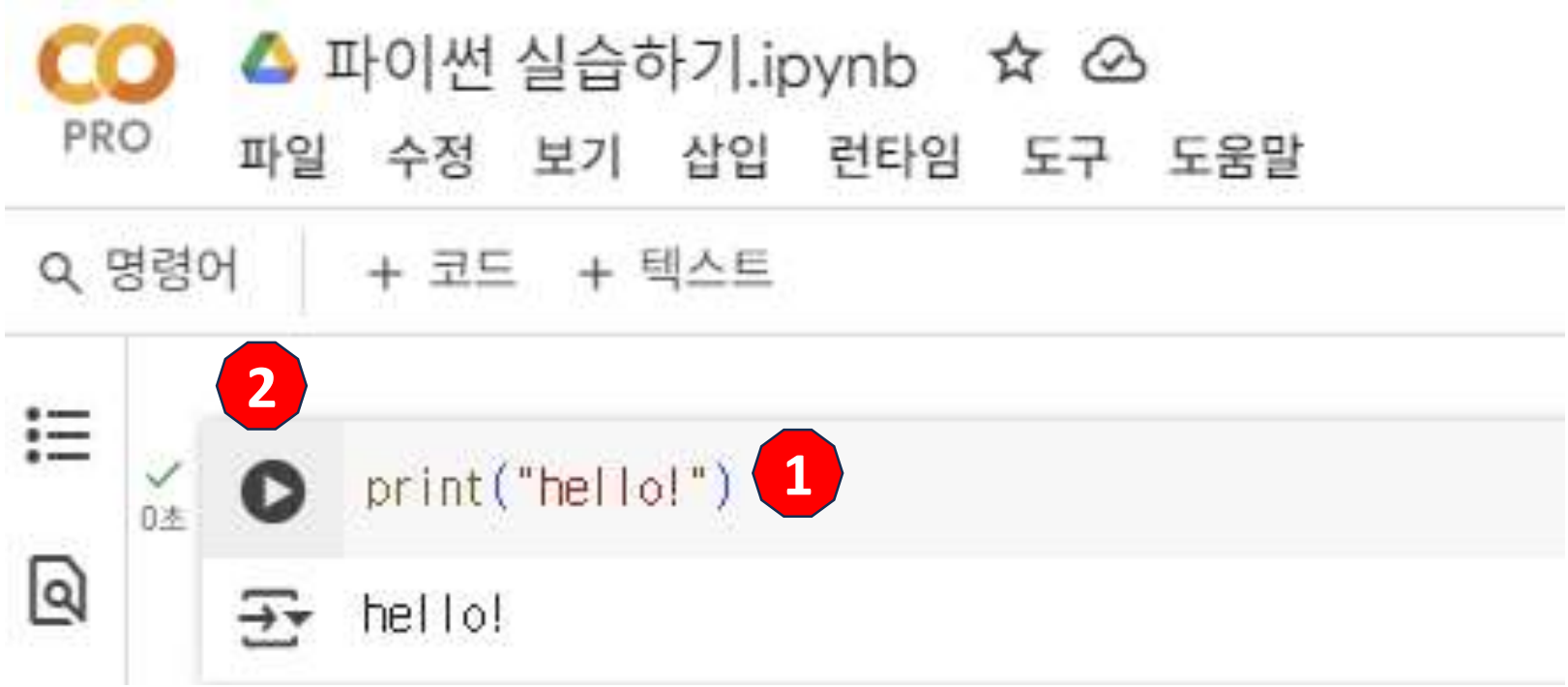
-> New Notebook -> 본인 구글 계정으로 로그인

Google Colab Usage



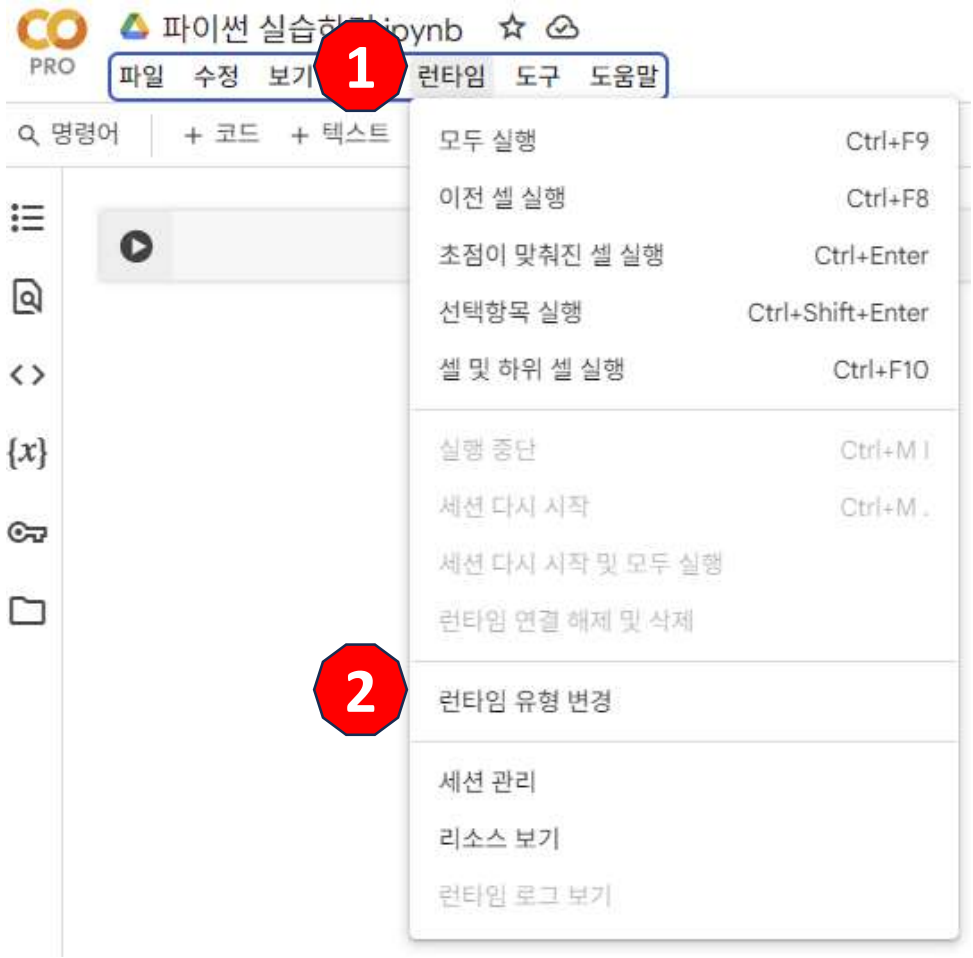
(3) 로그인 후 다시 새 노트를 만들면, 위에 사진처럼 뜨면 된 겁니다.
이때 다시 파일을 구분하기 위해 제목을 수정합니다.

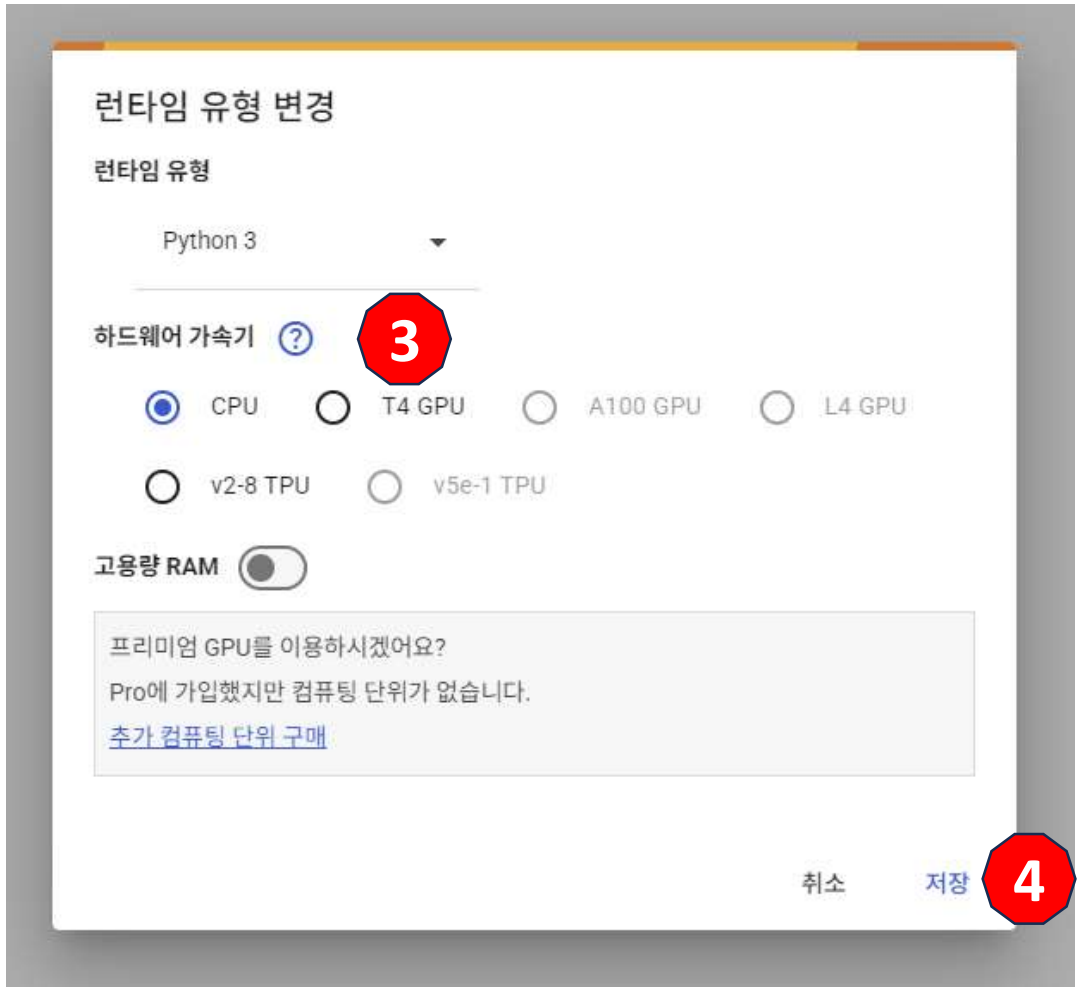
Google Colab Usage



(3) 코랩은 Jupyter notebook과 거의 유사합니다. (1) 그래서 셀 창에 print 구문을 입력하고,
(2) 재생 표시를 누르거나 (shift + enter)를 누르게 되면 실행됩니다.

Google Colab Usage



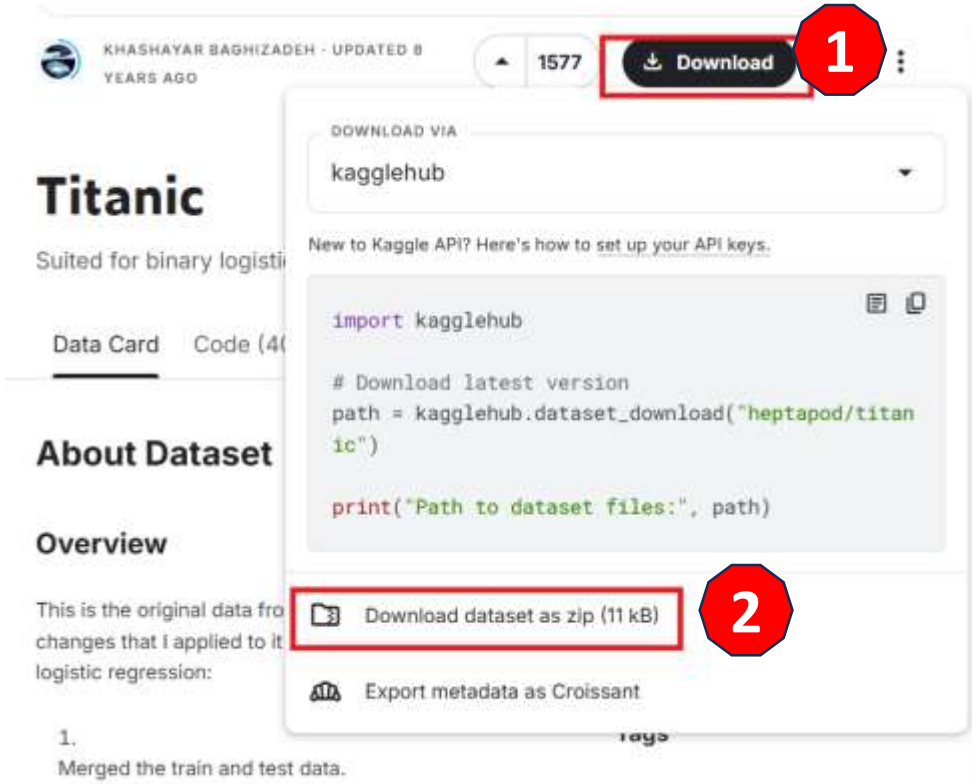


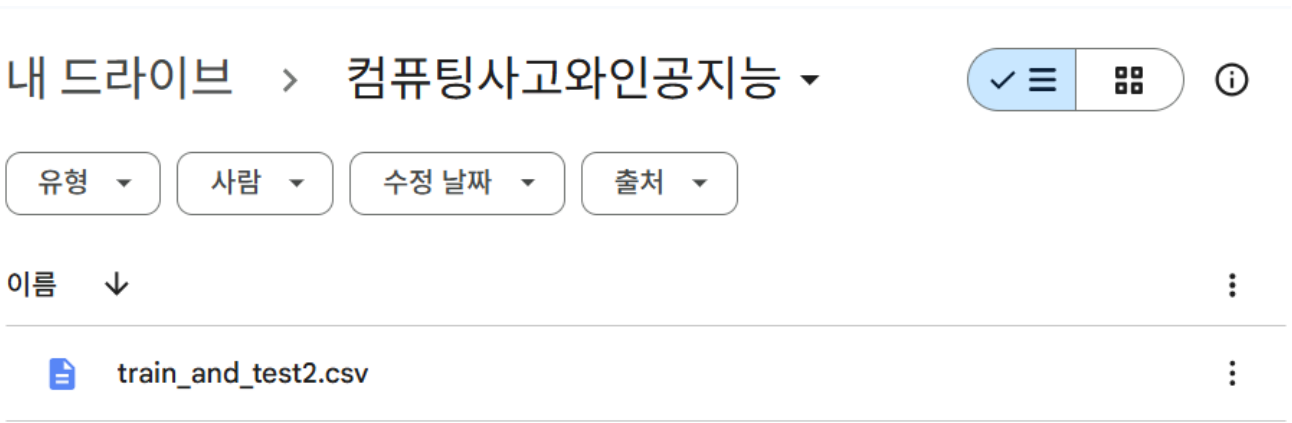
(3) GPU는 딥러닝을 돌릴 때 필요한 자원으로 코랩에서는 무료로 지원해주고 있습니다.
설정은 위와 같이 (1) 런타임을 누르고 (2) 런타임 유형 변경을 누릅니다. 그러면 오른쪽처럼 뜨게 되는데,
(3) 이때 T4 GPU나 v2-8 GPU를 클릭하고 (4) 저장하면 GPU를 간단하게 설정할 수 있습니다.
혹은 코드로도 지정할 수 있는데 다음주 실습 때 자세히 설명 예정입니다.

Google Colab Usage

타이타닉 데이터셋 다운로드

https://www.kaggle.com/datasets/heptapod/titanic

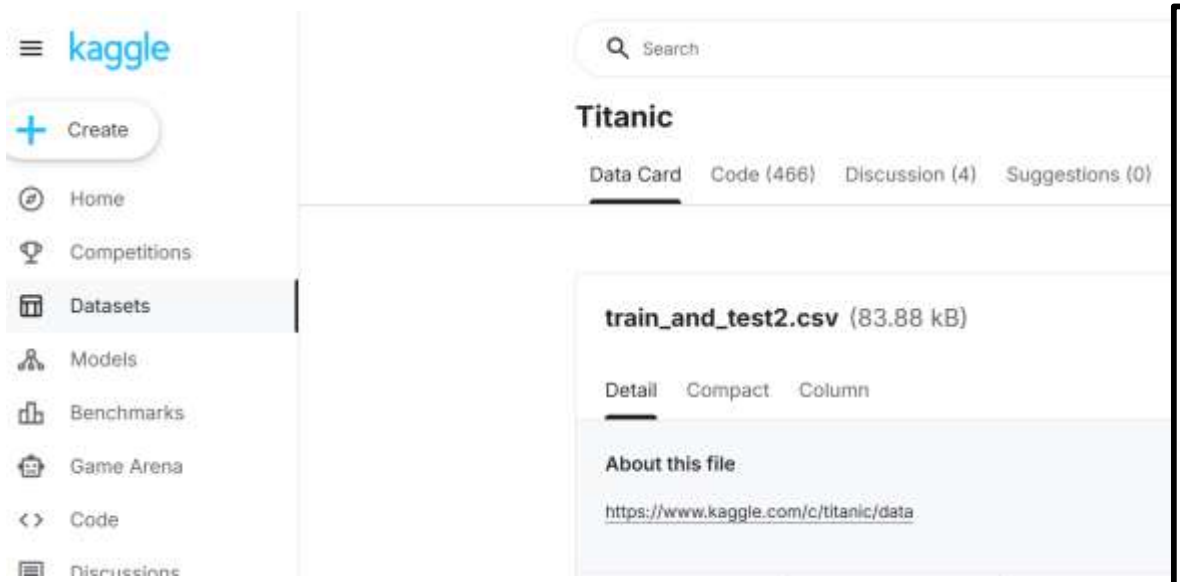




먼저 데이터를 다운로드합니다. 이 예시에서는 캐글의 타이타닉 데이터셋을 활용하겠습니다.
(1), (2)를 눌러 압축 파일을 다운로드합니다.
압축 해제 후 train_and_test2.csv파일을 구글드라이브 작업 폴더에 업로드합니다(이 예시에서는 '컴퓨팅사고와 인공지능' 폴더를 만들었습니다)

Titanic Dataset

탑승자의 신원과 관련된 정보와 생존 여부가 포함되어 있어,
타이타닉호 침몰 사고의 승객 생존 여부를 예측하는 데 사용되는 데이터셋

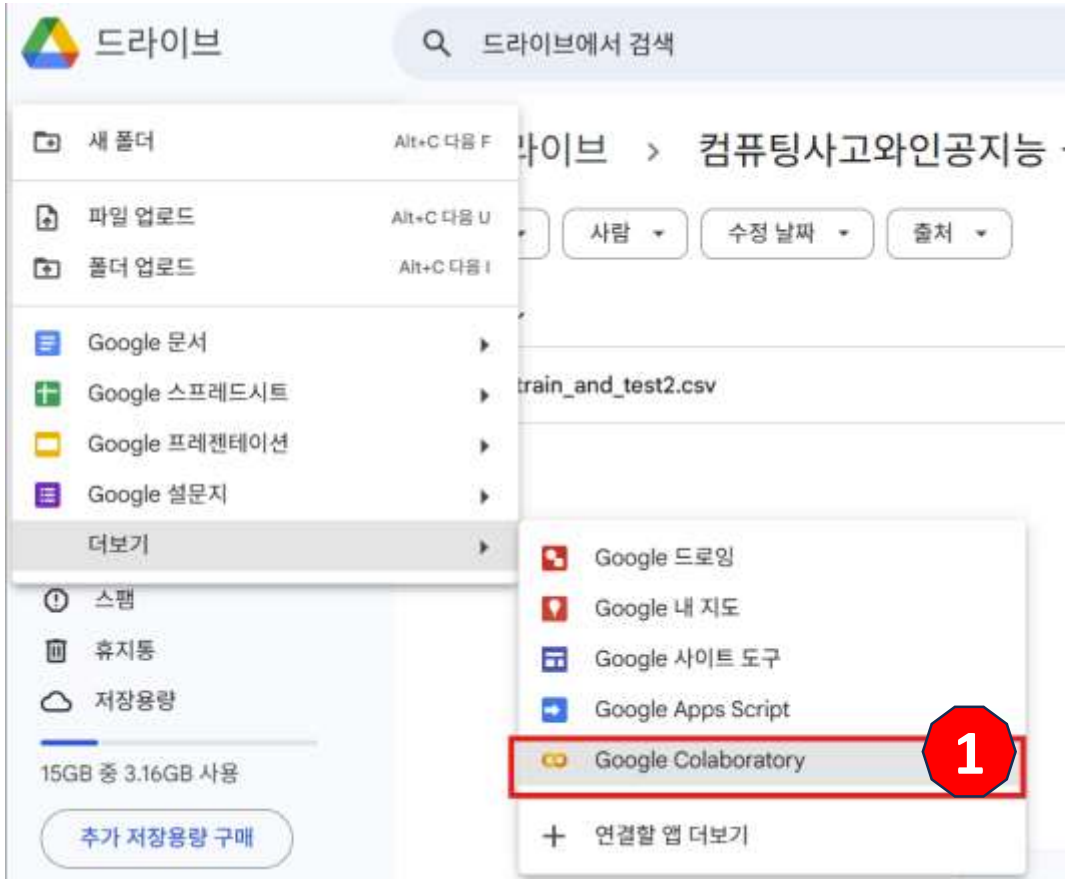


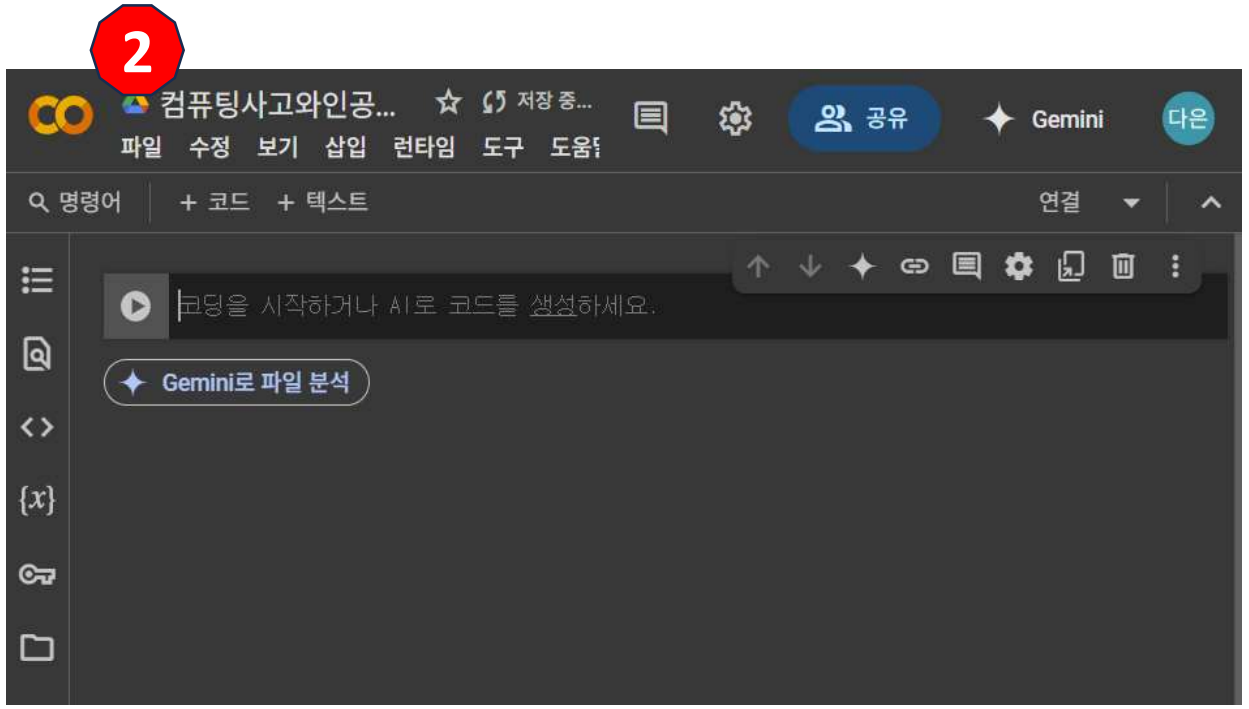
Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Cf) Titanic 데이터셋 정보 확인 가능

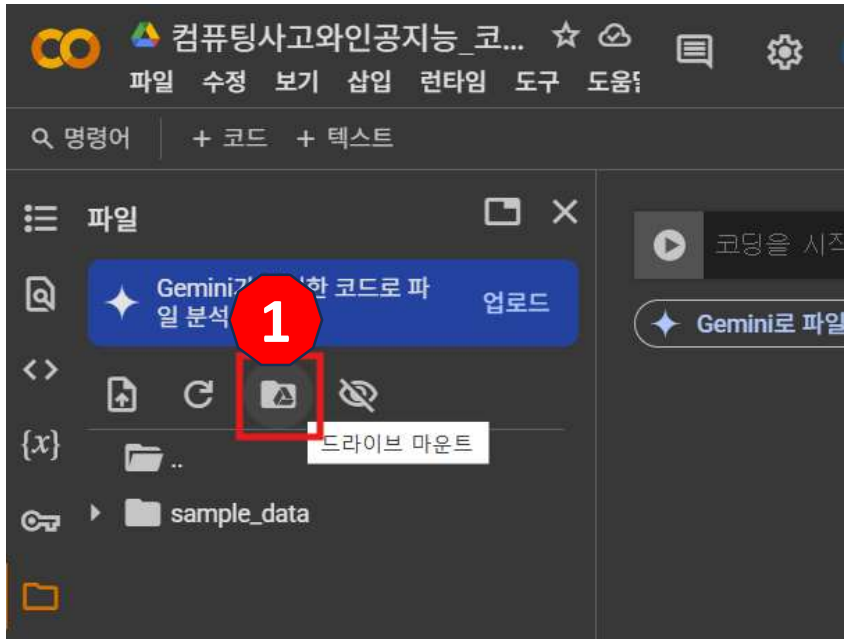
Google Colab Usage

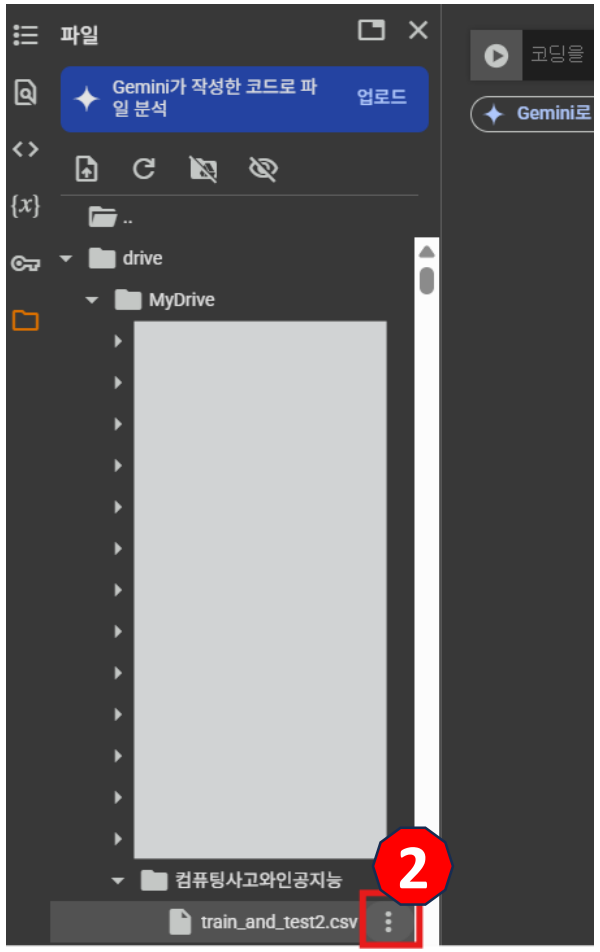


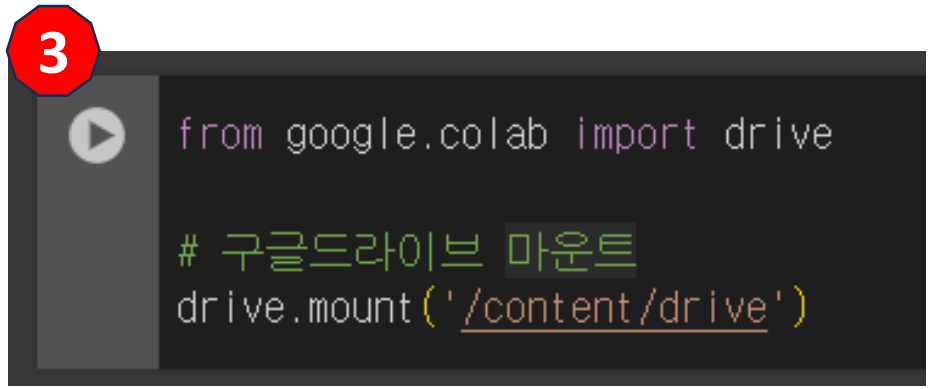


현재 위치에서 코랩 파일을 새로 열겠습니다.
코랩에서 구글드라이브에 저장된 데이터 파일을 불러오려면, 먼저 구글 드라이브를 코랩에 마운트(연결)해야 합니다.
먼저 (1) 코랩 노트북 파일을 새로 만들고 (2) 파일 이름을 변경해주세요.

Google Colab Usage







먼저 구글 드라이브를 코랩에 마운트(연결)하는 방법은 2가지가 있습니다.
(1) 폴더 아이콘을 누르고 드라이브 마운트 아이콘을 클릭합니다.
데이터를 불러오기 위해 데이터파일이 저장되어 있는 경로를 찾아가 (2)를 누르고 경로 복사를 해줍니다.
(마운트 버튼을 클릭하고 기다리면 MyDrive가 나타납니다)
(3)의 코드를 실행하는 방법으로도 마운트가 가능합니다. *(1번, 3번 방법 중 택 1)*

Google Colab Usage

구글드라이브와 코랩 노트북이 연결되었으니 데이터를 불러오겠습니다.
(1) Import pandas as pd로 판다스 라이브러리를 불러온 후
(2) pd.read_csv("데이터 파일 경로")로 데이터를 불러오고, 이를 변수 data에 저장합니다.
(3) data.head(5) -> 위에서부터 5행만 조회 로 확인하면 데이터가 잘 로드된 것을 확인할 수 있습니다.

Scikit-learn

파이썬에서 머신러닝 모델을 구현할 수 있도록 개발된 오픈소스 라이브러리

모듈	역할
sklearn.datasets	미리 준비된 데이터셋 불러오기
sklearn.svm	서포트 벡터 머신 계열 모델
sklearn.linear_model	선형 회귀, 로지스틱 회귀 등 선형 기반 모델
sklearn.model_selection	훈련/테스트 분리, 하이퍼파라미터 탐색 등
⋮	⋮

Python Programming with tensor

Modeling - Load Dataset

- 사이킷런(scikit-learn) 라이브러리 설치
  - pip install scikit-learn 명령어로 라이브러리 설치

TIP https://scikit-learn.org/dev/_downloads/scikit-learn-docs.pdf에 접속하면 가장 최신 버전의 사이킷 런 사용 설명서를 무료로 다운로드할 수 있다. 무려 2,500여 쪽에 달하는 방대한 문서다. 그렇다고 겁먹을 필요는 없다. 필요한 부분을 선택적으로 참조하면 된다.

Google Colab Usage

2교시 수업 내용을 진행하기 위해 scikit-learn 라이브러리가 필요합니다.
(1) !pip install scikit-learn을 입력 후 실행하여 라이브러리를 설치해주세요.

Modeling - Load Dataset

- iris 데이터셋 읽기

```
01  from sklearn import datasets
02
03  d=datasets.load_iris()    # iris 데이터셋을 읽고
04  print(d.DESCR)            # 내용을 출력
```

  - 01행: sklearn 모듈의 datasets 클래스를 불러옴
  - 03행: load_iris 함수를 호출해 iris 데이터셋을 읽어 객체 d에 저장
  - 04행: 객체 d의 DESCR 변수를 출력

- 기계 학습의 용어
  - 샘플로 구성되는 데이터셋
  - 특징으로 구성되는 특징 벡터(feature vector)
  - 부류(class)

TIP 기계 학습이 사용하는 데이터는 여러 개의 샘플을 담고 있어서 데이터셋(data set)이라 부르기도 한다. 이 책에서는 데이터와 데이터셋을 엄밀히 구분하지 않고 함께 사용하는데, 데이터셋은 iris처럼 특정한 데이터를 가리킬 때 주로 사용한다.

Modeling - Dataset

```
Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

:Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
...
```

150개의 샘플

네 개의 특징(feature)

세 개의 부류 (품종)

Modeling - Dataset

04행까지는 앞의 슬라이드 참고

```
05  for i in range(0,len(d.data)):          # 샘플을 순서대로 출력
06      print(i+1,d.data[i],d.target[i])
```

```
1 [5.1 3.5 1.4 0.2] 0
2 [4.9 3.  1.4 0.2] 0
3 [4.7 3.2 1.3 0.2] 0
4 [4.6 3.1 1.5 0.2] 0
...
51 [7.  3.2 4.7 1.4] 1
52 [6.4 3.2 4.5 1.5] 1
53 [6.9 3.1 4.9 1.5] 1
54 [5.5 2.3 4.  1.3] 1
...
101 [6.3 3.3 6.  2.5] 2
102 [5.8 2.7 5.1 1.9] 2
103 [7.1 3.  5.9 2.1] 2
104 [6.3 2.9 5.6 1.8] 2
...
```

d.target(레이블)

d.data(특징 벡터)

Modeling - Dataset

- 샘플을 특징 벡터와 레이블로 표현
  - 특징 벡터는 **x**로 표기(d는 특징의 개수로서 특징 벡터의 차원이라 부름)

$$\text{특징 벡터: } \mathbf{x}=(x_1,x_2,\cdots,x_d)$$

  - 레이블은 0,1,2,…,c-1의 값 또는 1,2,…,c-1,c의 값 또는 원핫 코드
    - 원핫 코드는 한 요소만 1인 이진열
      - 예) Setosa는 (1,0,0), Versicolor는 (0,1,0), Virginica는 (0,0,1)로 표현

	특징 벡터 $\mathbf{x}=(x_1, x_2, \cdots, x_d)$	레이블(참값) y
샘플 1:	(5.1, 3.5, 1.4, 0.2)	0
샘플 2:	(4.9, 3.0, 1.4, 0.2)	0
…	…	…
샘플 51:	(7.0, 3.2, 4.7, 1.4)	1
샘플 52:	(6.4, 3.2, 4.5, 1.5)	1
…	…	…
샘플 101:	(6.3, 3.3, 6.0, 2.5)	2
샘플 102:	(5.8, 2.7, 5.1, 1.9)	2
…	…	…
샘플 n:	(5.9, 3.0, 5.1, 1.8)	2

iris 데이터셋 (n=150, d=4)

Modeling - Dataset

Ex)

x=(꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비)

x= (5.1, 3.5, 1.4, 0.2)→ (1,0,0)

즉, Setosa

⇒ 이러한 특징 벡터를 학습시켜 분류 모델을 구현한다

Modeling – Machine Learning

- Machine Learning
  - SVM(support vector machine)이라는 기계 학습 모델을 사용

프로그램 3-1(c) iris에 기계 학습 적용: 모델링과 예측

```
07  from sklearn import svm
08
09  s=svm.SVC(gamma=0.1,C=10)                    # svm 분류 모델 SVC 객체 생성하고
10  s.fit(d.data,d.target)                       # iris 데이터로 학습
11
12  new_d=[[6.4,3.2,6.0,2.5],[7.1,3.1,4.7,1.35]]  # 101번째와 51번째 샘플을 변형하여
                                                   새로운 데이터 생성
13  res=s.predict(new_d)
14  print("새로운 2개 샘플의 부류는", res)
```

하이퍼 매개변수

훈련 집합

테스트 집합

```
새로운 2개 샘플의 부류는 [2 1]
```

  - 09행: SVM의 분류기 모델 SVC 클래스의 객체를 생성하여 s에 저장
  - 10행: 객체 s의 fit 함수는 훈련 집합을 가지고 학습을 수행(매개변수로 특징 벡터 iris.data와 레이블 iris,target을 설정)
  - 13행: 객체 s의 predict 함수는 테스트 집합을 가지고 예측 수행

Modeling – Machine Learning

- Confusion matrix(혼동행렬)
  - 성능 측정 지표

[10]
✓ 0초

```
import numpy as np

pred=s.predict(d.data)       # 전체 데이터에 대해 모델 s로 예측

conf=np.zeros((3,3))       #혼동행렬을 저장할 3x3 행렬 생성(iris=3개의 클래스)
for i in range(len(pred)):
  conf[pred[i]][d.target[i]] +=1        #실제 클래스와 예측 클래스 비교
print(conf)
```

```
[[50.  0.  0.]
 [ 0. 48.  0.]
 [ 0.  2. 50.]]
```

  - 1번을 2번으로 2개 잘못 분류
  - versicolor(1) vs virginica(2)
  - → 데이터 보강

Computational Thinking & Artificial Intelligence

Thanks for your attention.

Eunsom Jeon
Dept. of Computer Science & Engineering
ejeon6@seoultech.ac.kr