

# 빅데이터시스템

BIG DATA ANALYTICS  
TECHNOLOGY

텍스트빈도분석



# 목 차

## 01 텍스트빈도분석

환경설정, 목표설정, 데이터수집, 데이터준비, 데이터탐색,  
데이터모델링, 결과시각화

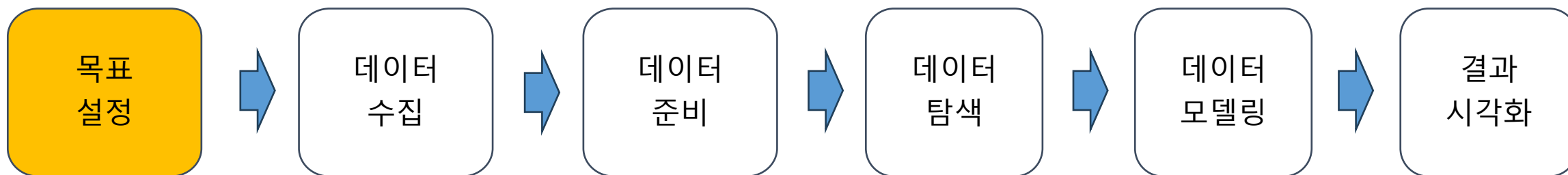
## 02 AI 비서

AI 비서 시스템 구성, 디렉토리구성, colab에서 학습시유의사항,  
colab에서학습한모델, 프로그램코딩, KoGPT패키지설치, 실행, 접속

# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터과학

### ■ 절차



- 와인의 속성을 분석한 뒤 품질 등급을 예측하는 것
- 데이터의 기술 통계를 구함
- 레드 와인과 화이트 와인 그룹의 품질에 대한 t-검정을 수행
- 와인 속성을 독립 변수로, 품질 등급을 종속 변수로 선형 회귀 분석을 수행

# \*. 전수업리뷰



## ⌚ 통계분석 - 목표설정

### ■ 핵심 개념 이해

[기술통계]

- 데이터의 특성을 나타내는 수치를 이용하여 분석하는 기본적인 통계 방법
- 평균, 중앙값, 최빈값 등을 구할 수 있음

[회귀 분석]

- 독립변수  $x$  와 종속변수  $y$  간의 상호 연관성 정도를 파악하기 위한 분석 기법
- 하나의 변수가 변함에 따라 대응되는 변수가 어떻게 변하는지 측정하는 것
- 변수 간의 인과관계를 분석할 때 많이 사용

# \*. 전수업리뷰



## ⌚ 통계분석 - 목표설정

### ■ 핵심 개념 이해

[t-검정]

- 데이터에서 찾은 평균으로 두 그룹에 차이가 있는지 확인하는 방법

예) A 와인의 품질이 1등급인지 2등급인지에 따라 가격에 차이가 있는지 확인할 때 사용

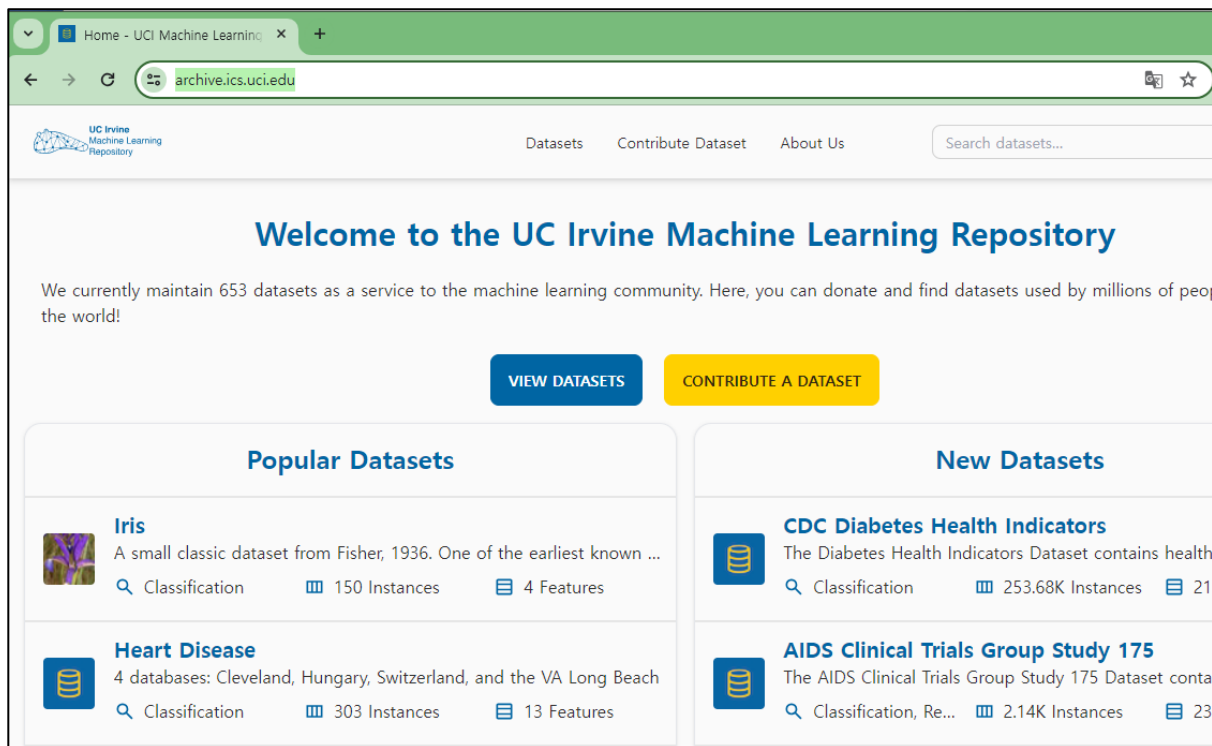
[히스토그램]

- 데이터 값의 범위를 몇 개 구간으로 나누고 각 구간에 해당하는 값의 숫자나 상대적 빈도 크기를 차트로 나타낸 것

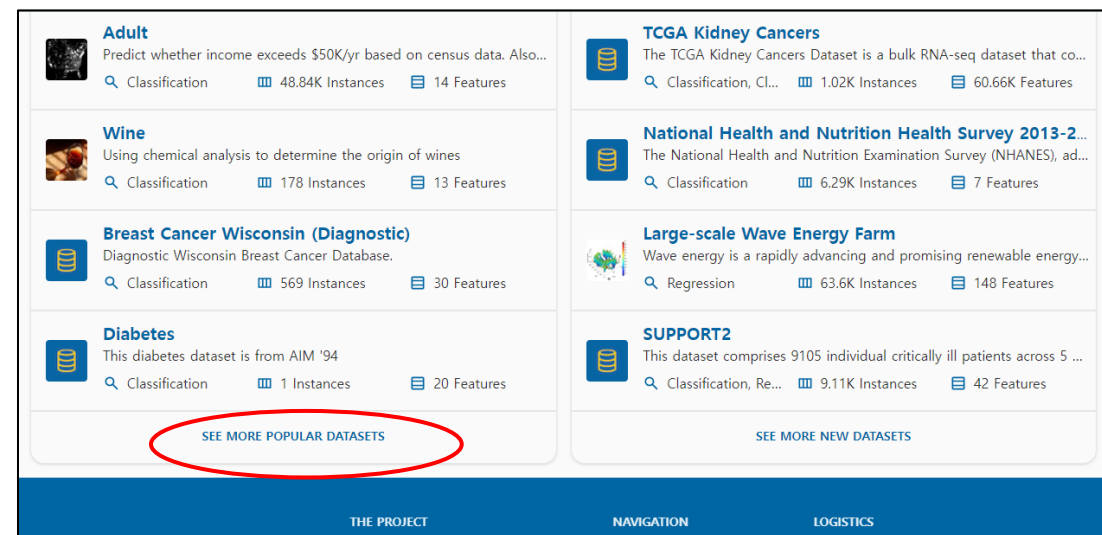
# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터 수집

- 데이터셋 : 캘리포니아 어바인 대학의 머신러닝 저장소에서 제공



<https://archive.ics.uci.edu/>



# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터 준비

- 전처리-1 : 세미콜론(;) 으로 된 csv 파일을 읽어, xlsx 로 저장



cd \\_seok\data

python wine-01-preprocess.py

```

C:\_seok>명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\_seok>cd \_seok\data

C:\_seok\data>python wine-01-preprocess.py

C:\_seok\data>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F6D9-BD25

C:\_seok\data 디렉터리

2023-11-06 오후 03:20 <DIR> .
2023-11-06 오후 03:20 <DIR> ..
2023-10-03 오전 02:07 10 access.txt
2023-10-03 오전 02:05 9,265 mydata.xlsx
2023-11-06 오후 03:19 407 wine-01-preprocess.py
2023-05-22 오후 10:24 84,199 winequality-red.csv
2023-11-06 오후 03:20 93,779 winequality-red2.xlsx
2023-05-22 오후 10:24 264,426 winequality-white.csv
2023-11-06 오후 03:20 278,171 winequality-white2.xlsx
2023-05-22 오후 10:24 3,305 winequality.names
                        8개 파일              733,562 바이트
                        2개 디렉터리 130,907,881,472 바이트 남음

C:\_seok\data>
  
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	winequality-red	winequality-white	winequality-red2	winequality-white2	winequality.names	winequality-red	winequality-white	winequality-red2	winequality-white2	winequality.names	winequality-red	winequality-white
2	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
3	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
4	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
5	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
6	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
8	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
9	7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7

	A	B	C	D	E	F	G	H	I	J	K	L
1	winequality-red	winequality-white	winequality-red2	winequality-white2	winequality.names	winequality-red	winequality-white	winequality-red2	winequality-white2	winequality.names	winequality-red	winequality-white
2	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
3	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
4	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
5	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
6	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
8	6.2	0.32	0.16	7	0.045	30	136	0.9949	3.18	0.47	9.6	6
9	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6

winequality-red2.xlsx  
winequality-white2.xlsx  
파일이 생성됨

# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터 준비

### ■ 전처리-2 : 파일 병합

```
#=====
# 전처리-2 : 파일 병합
#=====
import pandas as pd

r_df = pd.read_excel('./winequality-red2.xlsx')
w_df = pd.read_excel('./winequality-white2.xlsx')

r_df.insert(0, column = 'type', value = 'red')
w_df.insert(0, column = 'type', value = 'white')

wine = pd.concat([r_df, w_df])
wine.to_excel('./wine.xlsx', index=False)
```



\_seok\data\wine-02-pre-concat.py 로 저장

python wine-02-pre-concat.py 실행

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	sulfur dioxide	sulfur dioxide	density	pH	sulphates	alcohol	quality
2	red	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
3	red	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
4	red	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
5	red	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
6	red	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7	red	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
8	red	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
9	red	7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7
10	red	7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	7
11	red	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5
12	red	6.7	0.58	0.08	1.8	0.097	15	65	0.9959	3.28	0.54	9.2	5
13	red	7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5

하나로 합쳐진,  
wine.xlsx  
파일이 생성됨



# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터 탐색

### ■ 기술 통계 구하기

<계속>

```
print(sorted(wine.quality.unique()))
print('=' * 40)

wine.quality.value_counts()
print('=' * 40)
```

- wine.quality.unique() 함수를 사용하여 quality 속성값 중에서 유일한 값을 출력  
이를 통해 와인 품질 등급quality은 3, 4, 5, 6, 7, 8, 9의 7개 등급이 있다는 것을 알 수 있음
- quality.value\_counts() 함수는 quality 속성값에 대한 빈도수를 보여줌  
6등급인 샘플이 2,836개로 가장 많고, 9등급인 샘플이 5개로 가장 적은 것을 알 수 있음



\\_seok\data\wine-03-navi-data.py 로 저장

[3, 4, 5, 6, 7, 8, 9]

```
=====
quality
6    2836
5    2138
7    1079
4     216
8     193
3      30
9       5
Name: count, dtype: int64
=====
```

# \*. 전수업리뷰

## ⌚ 통계분석 - 데이터 모델링

### ■ 품질등급 예측

<계속> \\_seok\data\wine-04-modeling.py 로 저장

```
#-----
# 품질등급 예측
#-----
sample1 = wine[wine.columns.difference(['quality', 'type'])]
sample1 = sample1[0:5][:]
print(sample1)
print('=' * 40)

sample1_predict = regression_result.predict(sample1)
print(sample1_predict)
print('=' * 40)

print(wine[0:5]['quality'])
print('=' * 40)
```



```
alcohol chlorides citric_acid density fixed_acid
volatile_acidity
0 9.4 0.076 0.00 0.9978 7.4 ...
1 9.8 0.098 0.00 0.9968 7.8 ...
2 9.8 0.092 0.04 0.9970 7.8 ...
3 9.8 0.075 0.56 0.9980 11.2 ...
4 9.4 0.076 0.00 0.9978 7.4 ...
```

[5 rows x 11 columns]

```
0 4.997607
1 4.924993
2 5.034663
3 5.680333
4 4.997607
```

dtype: float64

```
0 5
1 5
2 5
3 6
4 5
```

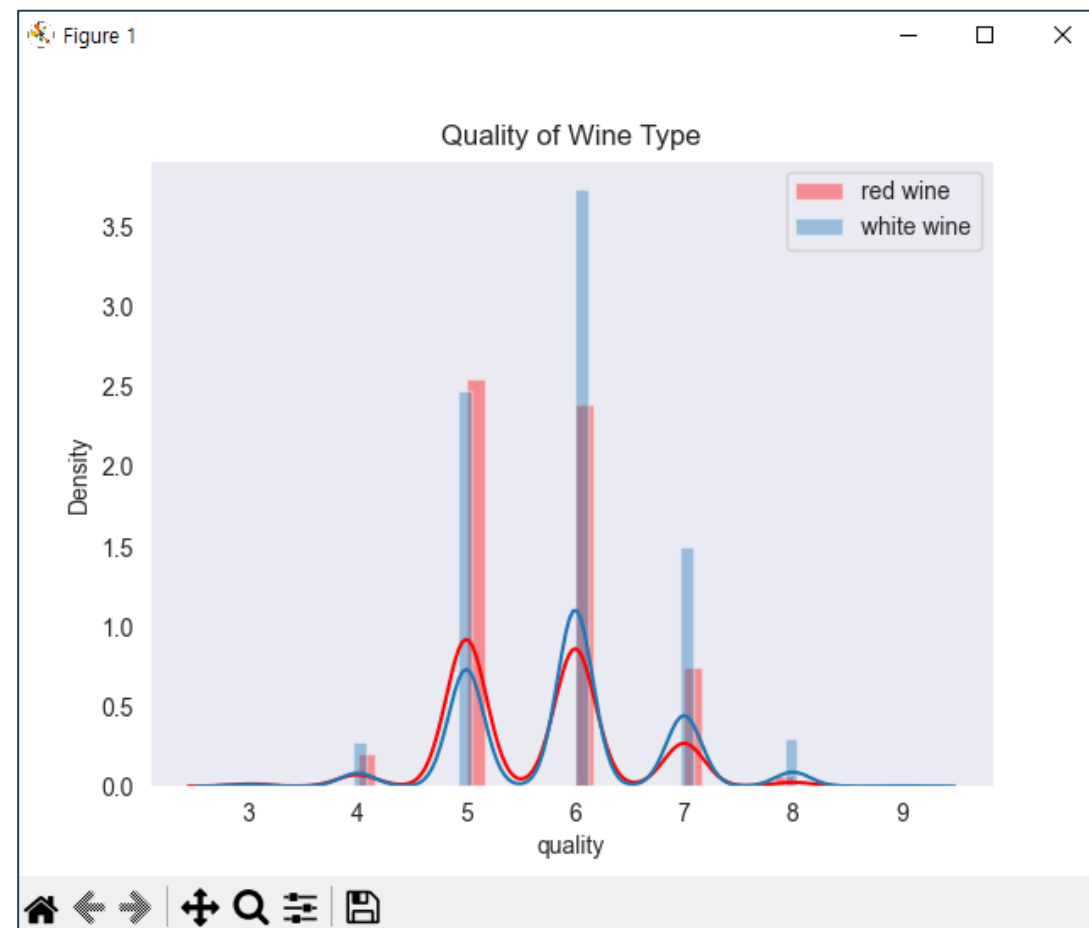
Name: quality, dtype: int64

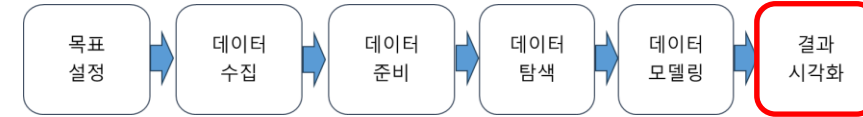
- 예측에 사용할 첫 번째 샘플 데이터 만들기
  - wine에서 quality와 type 열은 제외하고, 회귀 분석 모델에 사용할 독립 변수만 추출하여 sample1에 저장
  - sample1에 있는 샘플 중에서 0번~4번 5개 샘플만 추출하고, sample1에 다시 저장하여 예측에 사용할 샘플을 제작
- 첫 번째 샘플의 quality 예측하기
  - 샘플 데이터를 회귀 분석 모델 regression\_result의 예측 함수 predict()에 적용하여, 결과 예측값을 sample1\_predict에 저장
  - sample1\_predict를 출력하여 예측한 quality를 확인
- wine에서 0번부터 4번까지 샘플의 quality 값을 출력하여 sample1\_predict이 맞게 예측되었는지 확인

# \*. 전수업리뷰

## ⌚ 통계분석 - 결과 시각화

- 히스토그램 : 와인유형에 따른 품질등급

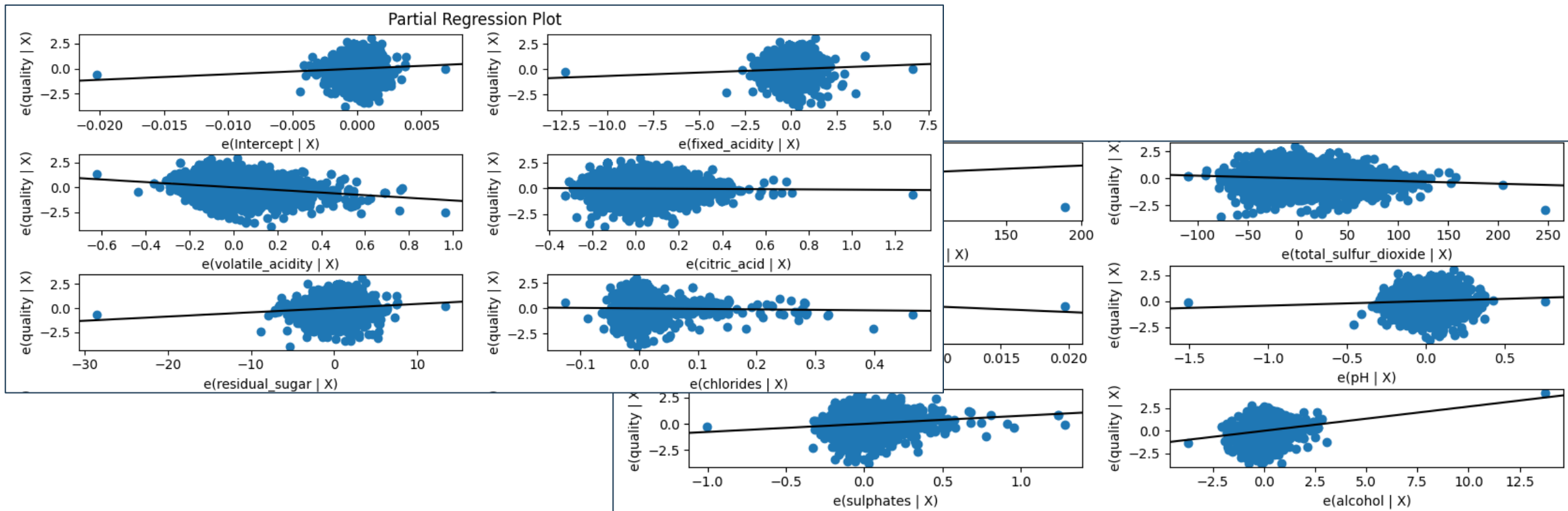




# \*. 전수업리뷰

## ⌚ 통계분석 - 결과 시각화

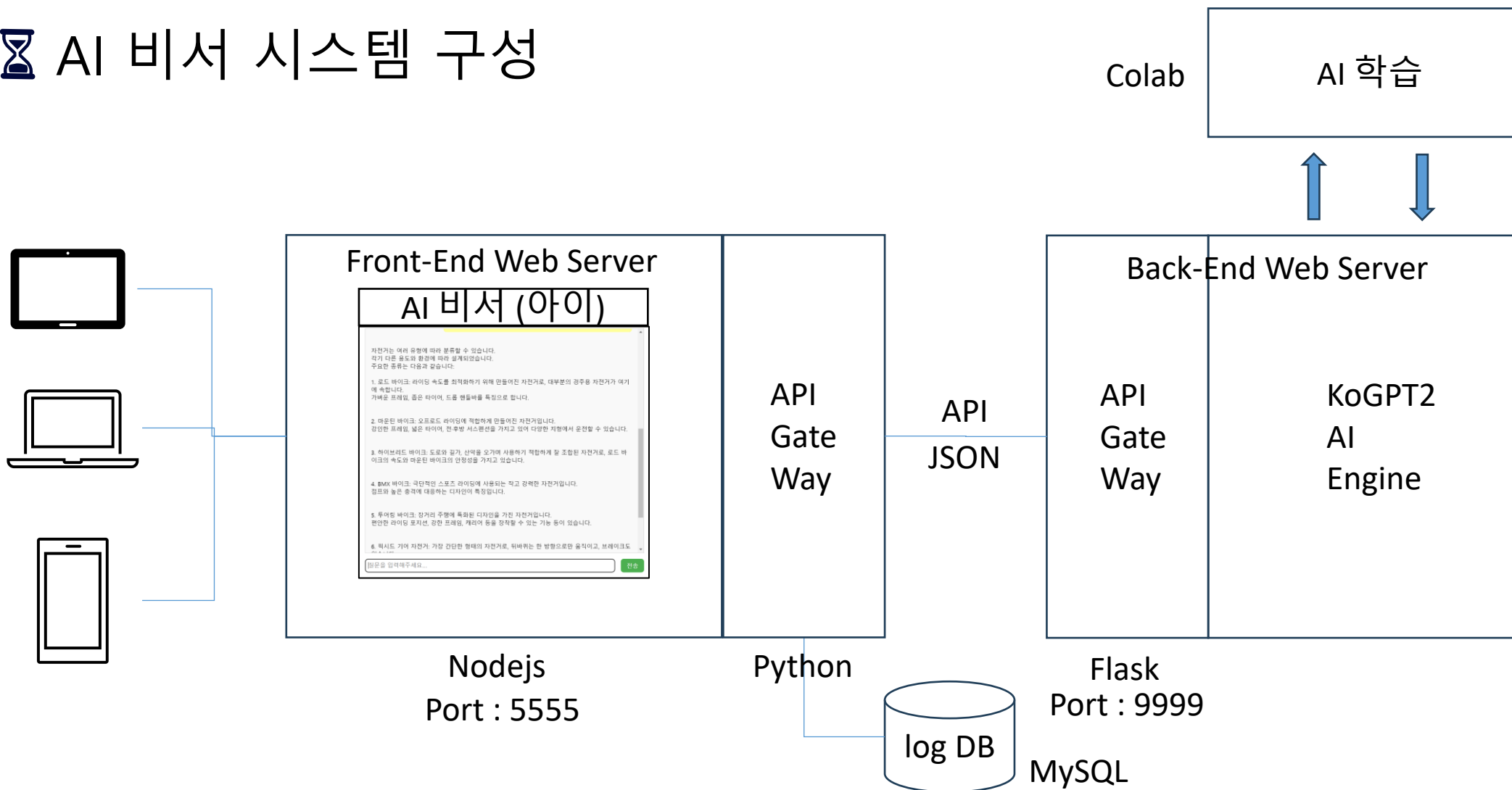
- PLOT : 하나의 독립변수가 종속변수에 미치는 영향





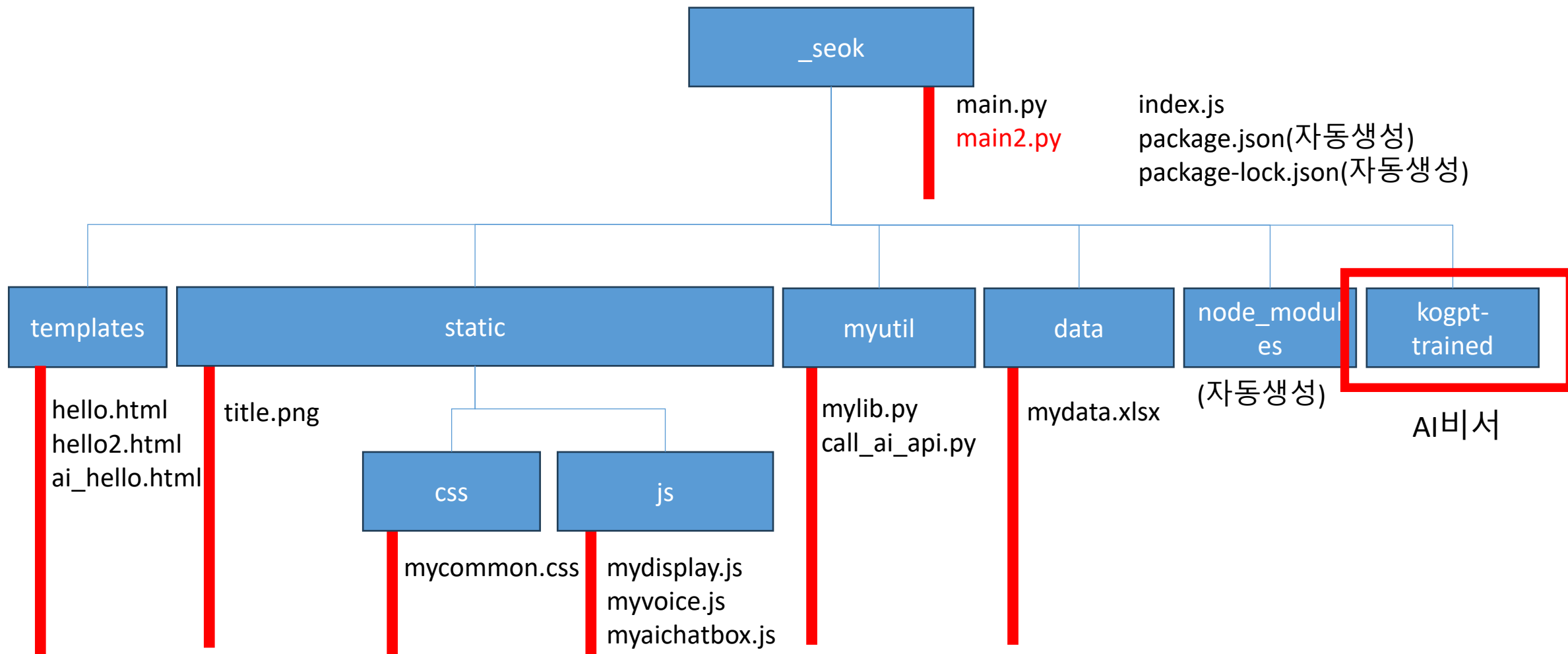
# \*. 전수업리뷰

## ⌚ AI 비서 시스템 구성



# \*. 전수업리뷰

## ⌚ 디렉토리 구성



# \*. 전수업리뷰

## ⌚ 학습 데이터 셋 생성

### ■ 질문/답변으로 구성

- ✓ 엑셀로 작성
- ✓ 질문은 prompt
- ✓ 답변은 completion
- ✓ 유사질문 10개당  
답변 1개로 구성

	A	B
1	prompt	completion
2	치안의 주체는?	경찰청입니다
3	민생치안의 주체는?	경찰청입니다
4	나는 누구입니까?	홍길동입니다
5	나의 이름은 무엇입니까?	홍길동입니다
6	나를 무엇이라고 부릅니까?	홍길동입니다
7	나는 무엇을 하는 사람입니까?	민생치안을 돕는 사람입니다
8	제 이름은 홍길동입니다. 저를 불러 봐 주세요	안녕하세요, 홍길동 님! 저는 여러분의 도움이 되는 민생치안 비서입니다. 무엇을 도와드릴까요?
9	제 이름은 홍길동입니다. 민생치안을 돕는 일꾼입니다. 저를 불러 주세요	안녕하세요, 홍길동님! 홍길동 님을 위해 어떻게 도와드릴까요? 무엇이 필요하시거나 어떤 도움이 필요 한지 알려주시면 기꺼이 도와드리겠습니다.
10	어떤 방법으로 집 안전을 강화할 수 있을까요?	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
11	방범용품(카메라, 경보기 등)을 설치하는 것이 집 안전을 강화하는 좋은 방법일까	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
12	집 안전을 강화하기 위해 어떤 방범용품을 사용하는 것이 가장 효과적일까요?	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
13	현관문과 창문 잠금장치를 강화하는 것이 집 안전을 증가시키는데 도움이 될까요?	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
14	이웃과 소통을 하는 것이 집 안전을 강화하는 데 도움이 될까요?	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
15	주변 상황을 주시하는 것이 집 안전을 강화하는 데 어떤 영향을 주나요?	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다
	방범용품을 설치하고 현관문과 창문 잠금 장치를 강화하는 것 외에 집 안전을 강화	집 안전을 강화하기 위해 방범용품(카메라, 경보기 등)을 설치하고 현관문과 창문 잠금장치를 강화하는 것이 좋습니다. 또한, 이웃과의 소통을 통해 주변 상황을 주시하고 방법에 도움을 요청할 수도 있습니다

# \*. 전수업리뷰

## ⌚ 학습 데이터 셋 생성

### ■ 유사질문 만들기 : chatgpt 활용

아래의 질문과 응답을 읽어 보고, 질문에 대한 응답을 참고하여, 질문과 동일한 의미를 가지는 질문형식으로 10개정도 만들어 주세요

질문: ""

교통 안전을 지키기 위한 팁이 있을까요?

""

응답: ""

교통 안전을 지키기 위해 교통 규칙을 준수하는 것이 가장 중요합니다. 또한, 안전 운전을 위해 휴대전화 사용을 자제하고, 안전벨트를 착용하고, 근본적인 원인인 술이나 졸음 운전을 하지 않는 것도 필요합니다.

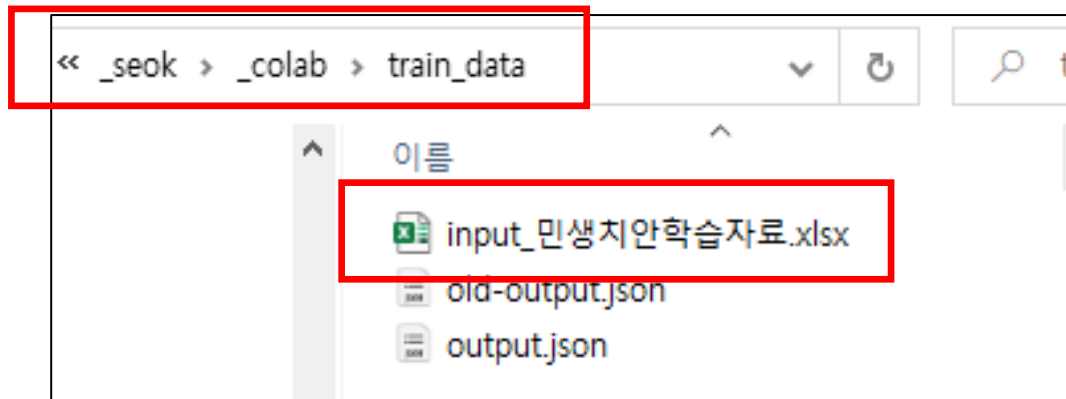
""



# \*. 전수업리뷰

## ⌚ 학습 데이터 셋 생성

### ■ 파일 저장



\\_seok\\_colab\train\_data 생성한 후, 엑셀 파일 저장

# \*. 전수업리뷰

## ⌚ 학습 데이터 셋 생성

### ■ 학습용 json 생성

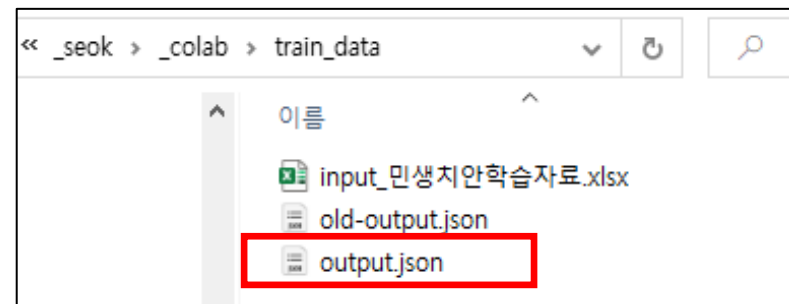
```

CA: 명령 프롬프트
(c) Microsoft Corporation. All rights reserved.
C:\Users\wdossa>cd \_seok\_colab
C:\_seok\_colab>python 1.make-json.py
=====
prompt
치안의 주체는?
민생치안의 주체는?
나는 누구입니까?
나의 이름은 무엇입니까?
다
4
다
나를 무엇이라고 부릅니까?
...
110 교통 안전을 위해 안전벨트 착용의 중요성은 무엇인가요? 교통 안전을
하는 것이 가장 중요합니다. 또한, 안전 ...
111 교통 안전을 위해 술이나 졸음 운전을 하지 않는 것이 왜 중요한가요? 교통 인
준수하는 것이 가장 중요합니다. 또한, 안전 ...
112 어떻게 하면 교통 안전을 위해 다른 운전자들과의 커뮤니케이션을 원활하게 할 수 있을까요?
통 규칙을 준수하는 것이 가장 중요합니다. 또한, 안전 ...
113 교통 안전을 위해 보행자로서 우리가 지켜야 할 규칙은 무엇인가요? 교통 인
준수하는 것이 가장 중요합니다. 또한, 안전 ...
114 교통 안전을 위해 우리가 주행 시에 고려해야 할 요소는 어떤 것들이 있을까요? 교
을 준수하는 것이 가장 중요합니다. 또한, 안전 ...
[115 rows x 2 columns]
=====
C:\_seok\_colab>

```

cd \\_seok\\_colab

python 1.make-json.py



output.json 이 생성됨

# \*. 전수업리뷰

## ⌚ 학습 프로그램

### ■ 학습 실행

```

명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

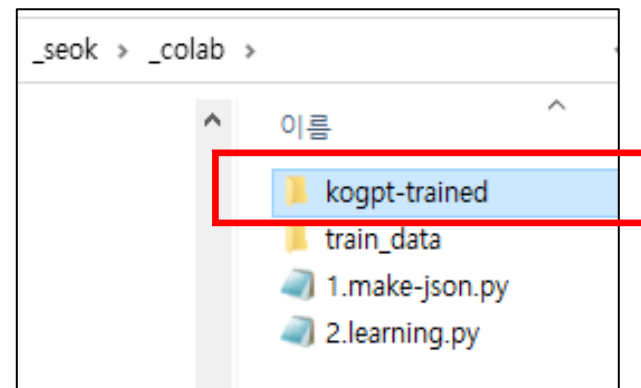
C:\Users\dossa>cd \_seok\_colab

C:\_seok\_colab>python 2.learning.py
device = cuda
The tokenizer class you load from this checkpoint is not the same type
result in unexpected tokenization.
The tokenizer class you load from this checkpoint is 'GPT2Tokenizer'.
The class this function is called from is 'PreTrainedTokenizerFast'.
Epoch 1, Batch 1/13, Loss: 15.0676
Accuracy: 46033.33%

C:\_seok\_colab>_
    
```

cd \\_seok\\_colab

python 2.learning.py

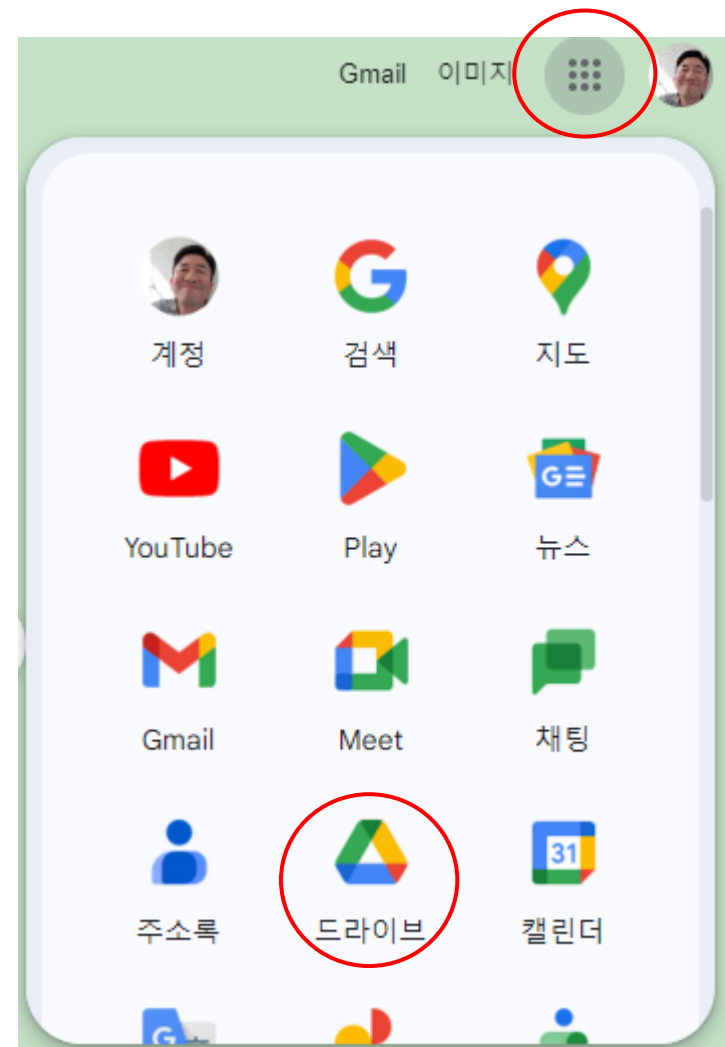
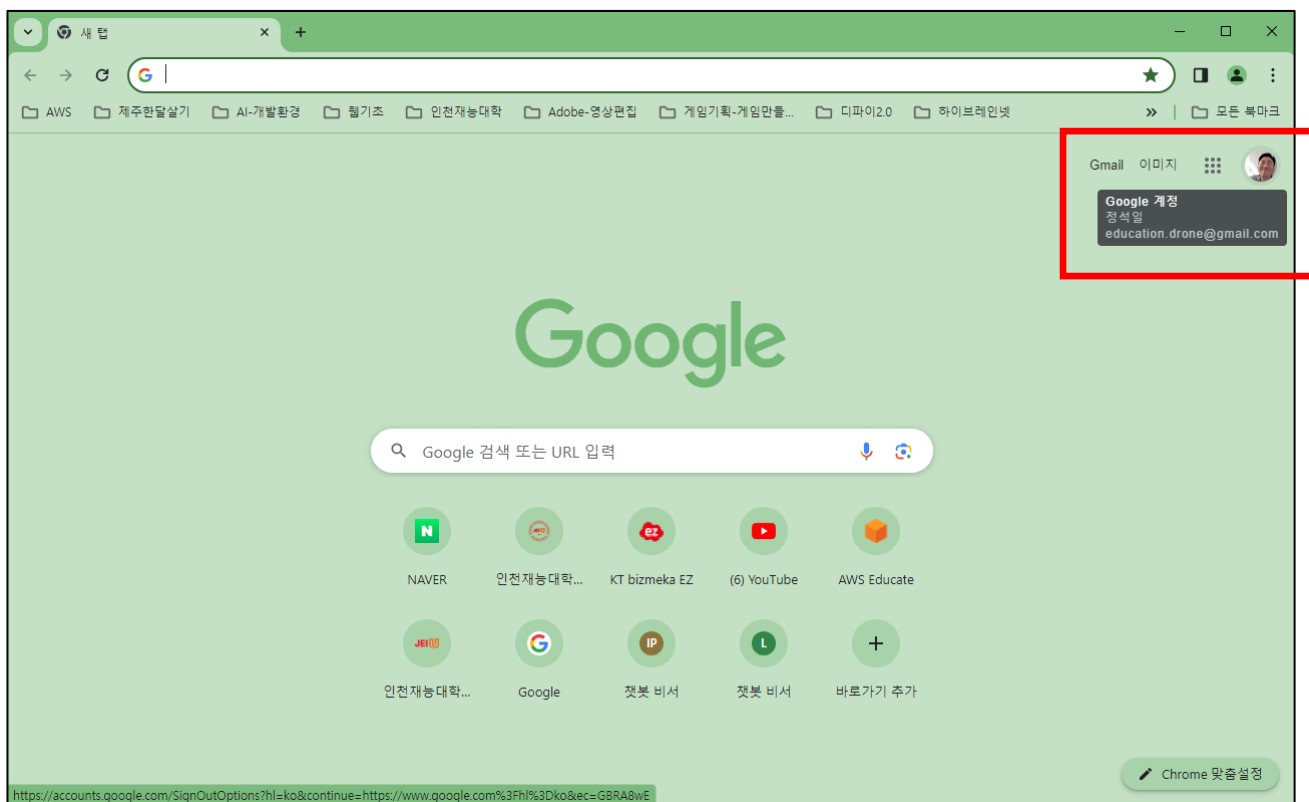


kogpt-trained 폴더 밑에 AI 가 생성됨

# \*. 전수업리뷰

⌚ colab 에서 학습

- 크롬에서 Google 로그인 → 구글드라이브

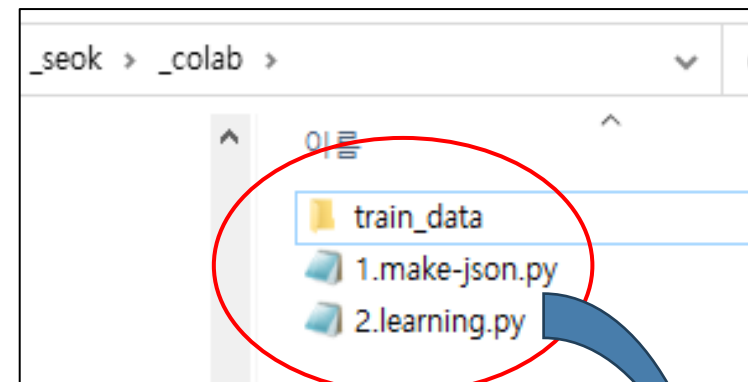
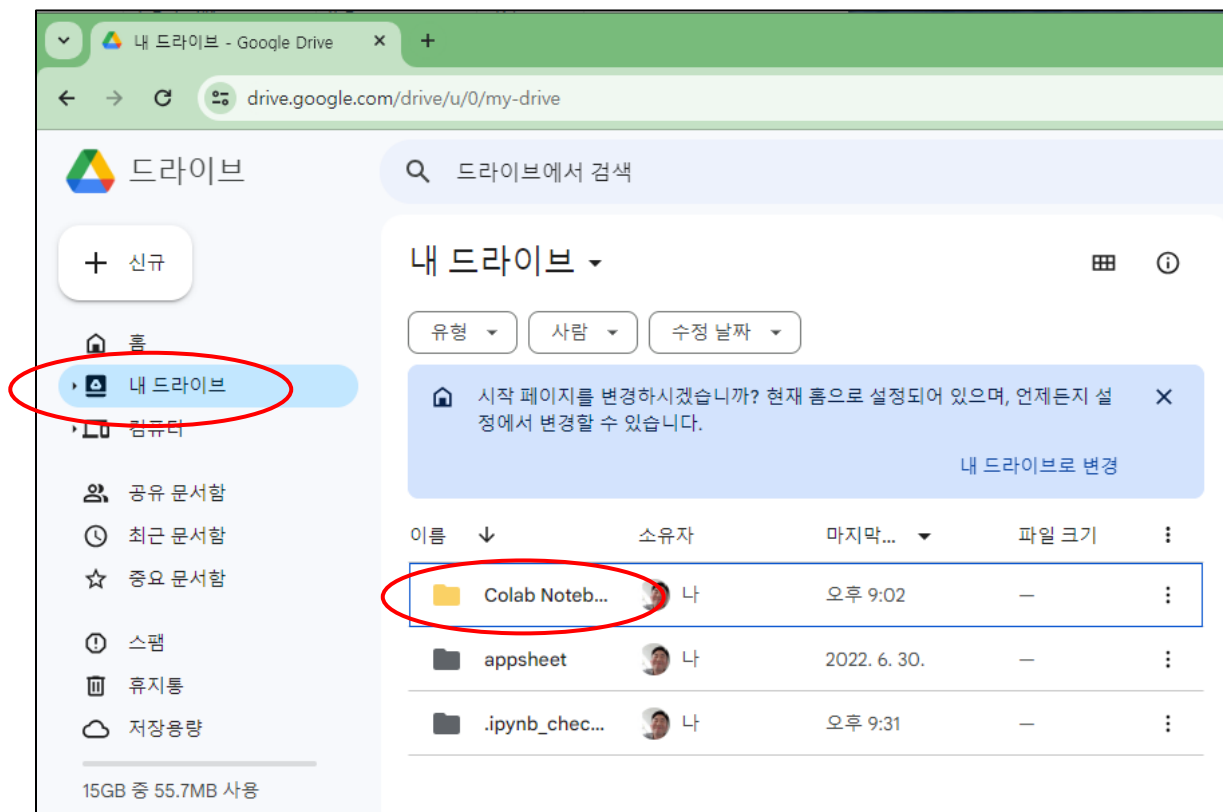




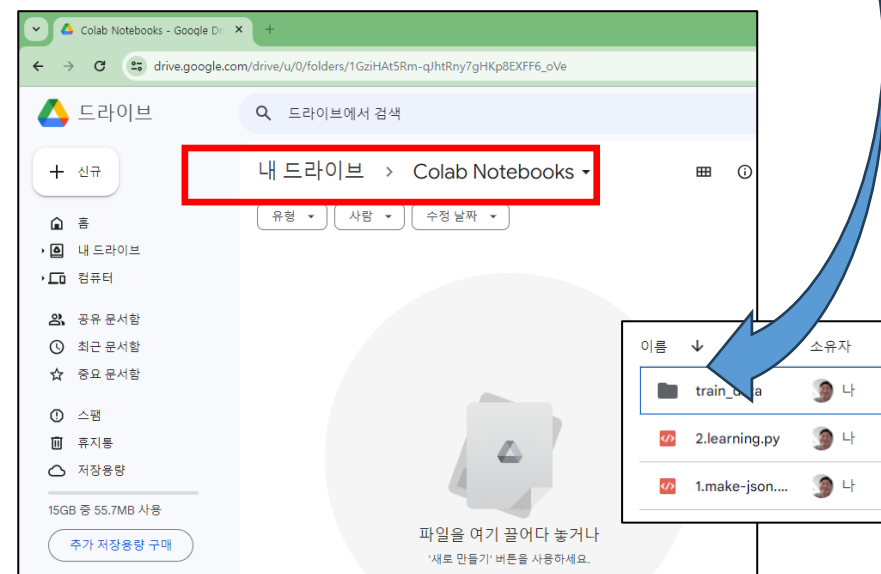
# \*. 전수업리뷰

## ⌚ colab 에서 학습

- 내 드라이브에 파일 업로드



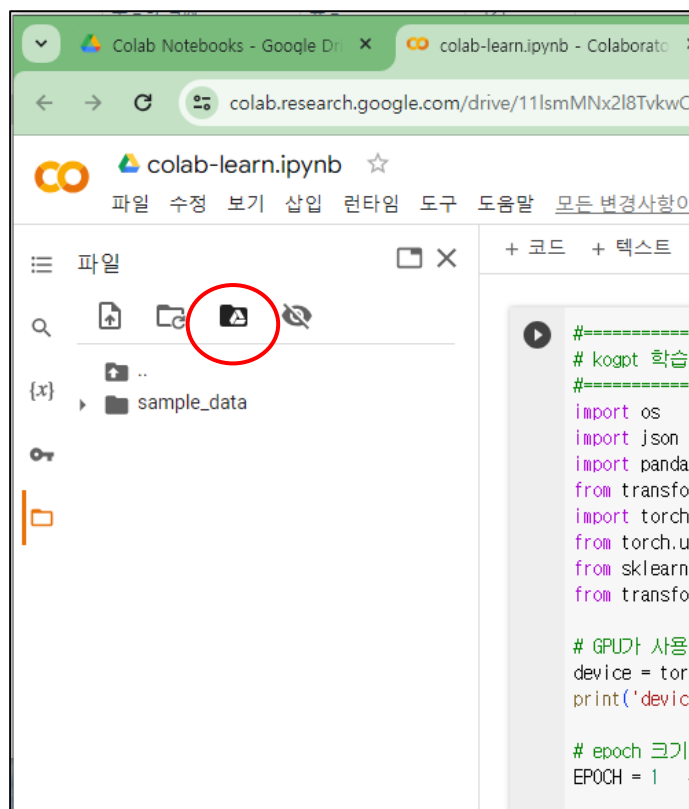
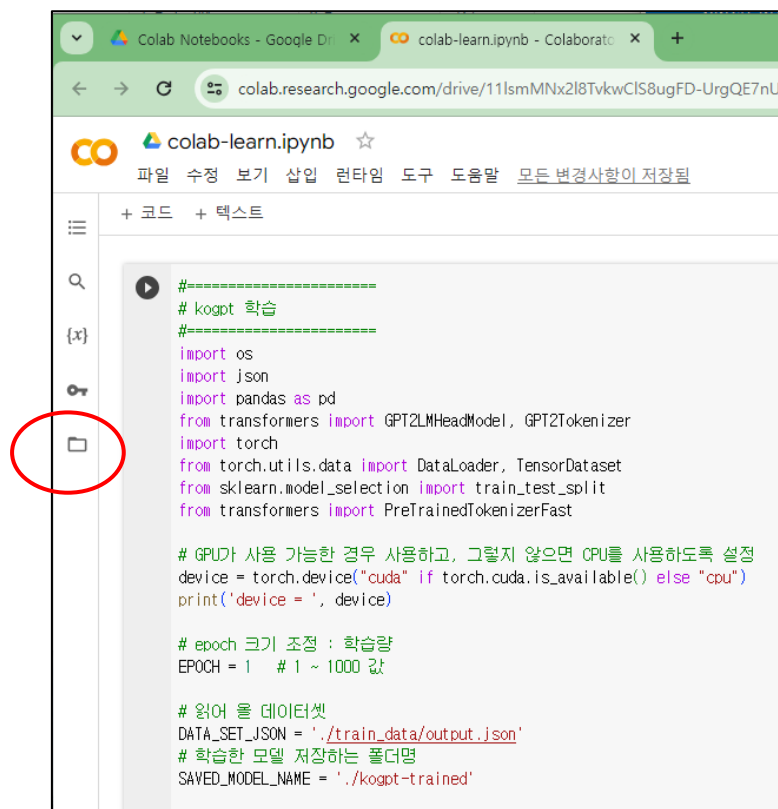
드래그인  
드랍



# \*. 전수업리뷰

## ⌚ colab 에서 학습

### ■ 드라이브 마운트



노트북이 **Google Drive** 파일에 액세스하도록 허용하시겠습니까?

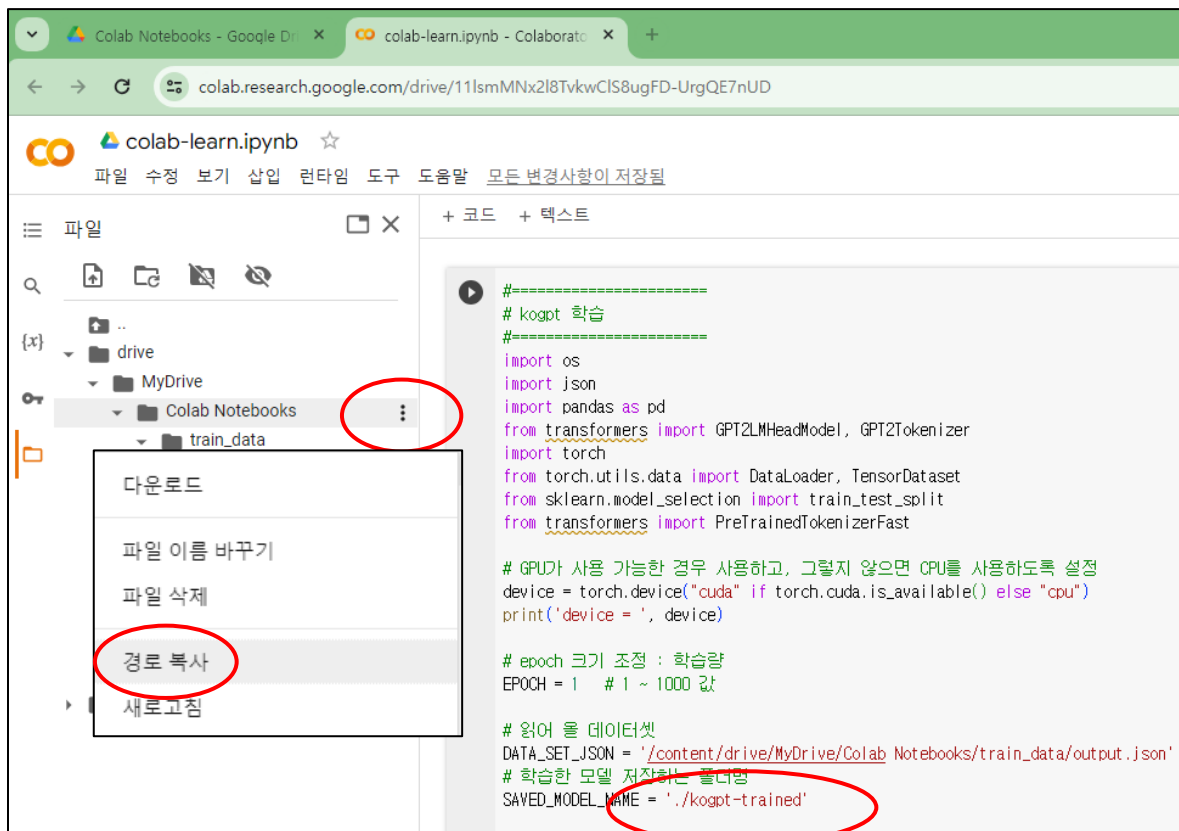
Google Drive에 연결하면 액세스 권한이 취소될 때까지 이 노트북에서 실행된 코드가 Google Drive의 파일을 수정할 수 있습니다.

아니요 [Google Drive에 연결](#)

# \*. 전수업리뷰

## ⌚ colab 에서 학습

- 드라이브 마운트 경로를 프로그램에 반영



drive → MyDrive → Colab Notebooks

옆에 있는 메뉴 클릭

<경로 복사> 한 것을

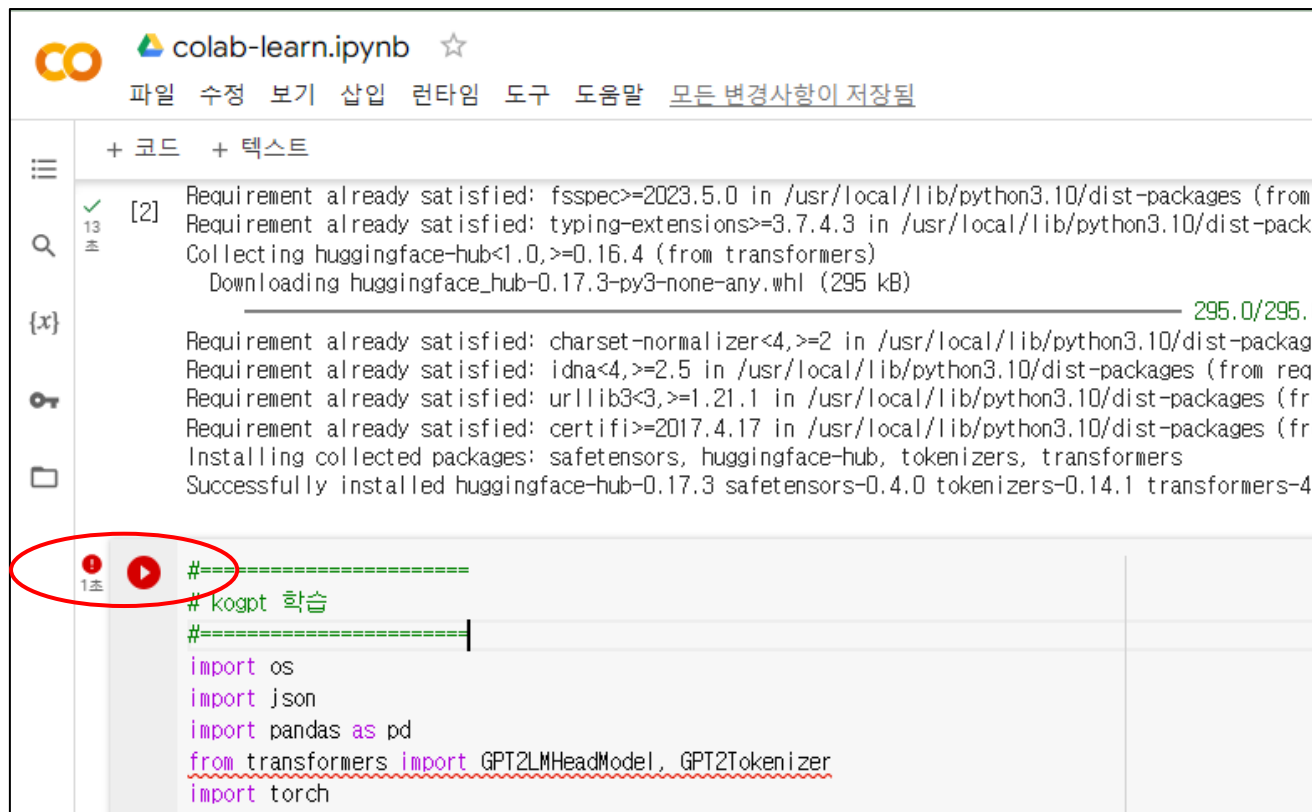
프로그램 소스의 SAVED\_MODEL\_NAME 에 붙여 넣기  
(Ctrl +V)

/content/drive/MyDrive/Colab Notebooks/kogpt-trained

# \*. 전수업리뷰

## ⌚ colab 에서 학습

### ■ 학습 시작



```

[2] Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-pack
Collecting huggingface-hub<1.0,>=0.16.4 (from transformers)
  Downloading huggingface_hub-0.17.3-py3-none-any.whl (295 kB)
295.0/295.0
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from req
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (fr
Installing collected packages: safetensors, huggingface-hub, tokenizers, transformers
Successfully installed huggingface-hub-0.17.3 safetensors-0.4.0 tokenizers-0.14.1 transformers-4

#-----
# kogpt 학습
#-----
import os
import json
import pandas as pd
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch
  
```

패키지 설치 후,

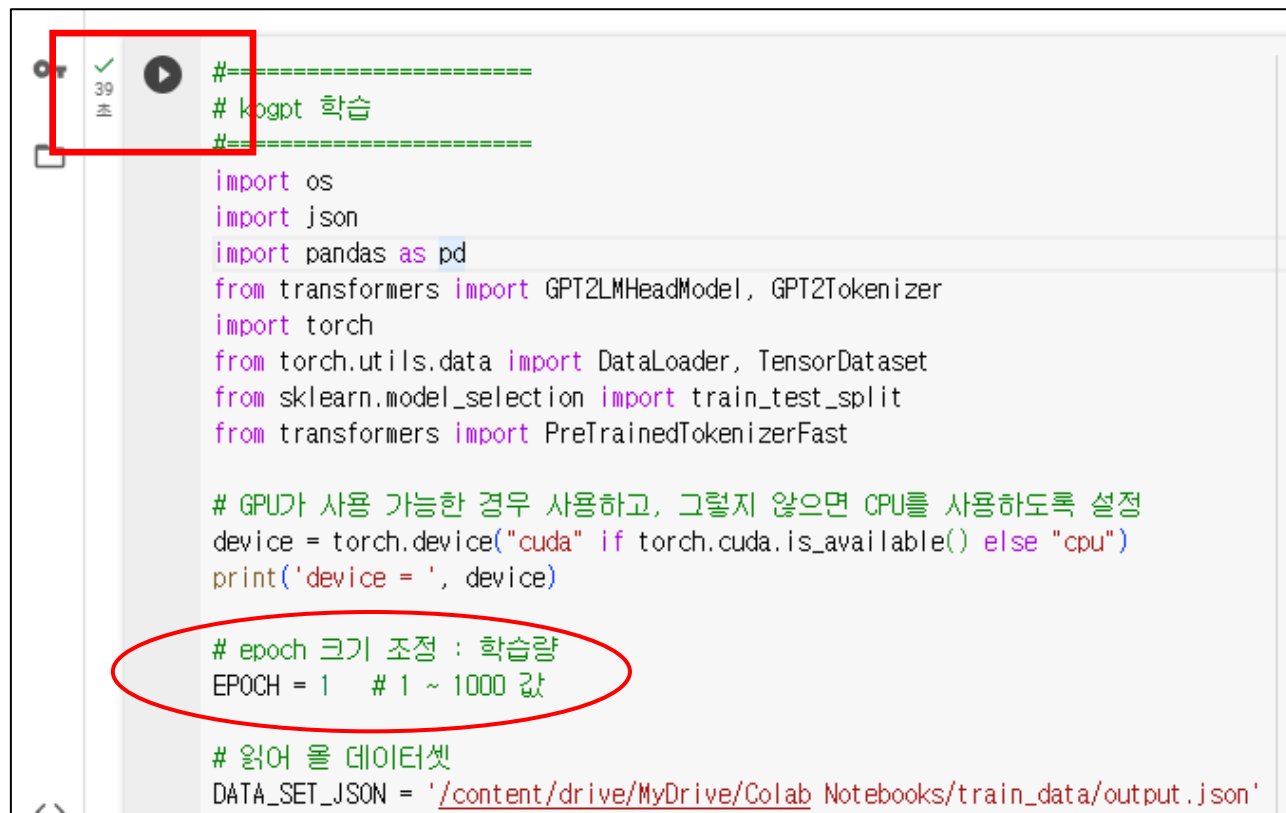
다시 [셀 실행] 클릭



# \*. 전수업리뷰

## ⌚ colab 에서 학습

- 학습 종료 후, 학습횟수 수정한 다음 재학습



```

#-----
# kagpt 학습
#-----

import os
import json
import pandas as pd
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch
from torch.utils.data import DataLoader, TensorDataset
from sklearn.model_selection import train_test_split
from transformers import PreTrainedTokenizerFast

# GPU가 사용 가능한 경우 사용하고, 그렇지 않으면 CPU를 사용하도록 설정
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print('device = ', device)

# epoch 크기 조정 : 학습량
EPOCH = 1 # 1 ~ 1000 값

# 읽어 올 데이터셋
DATA_SET_JSON = '/content/drive/MyDrive/Colab Notebooks/train_data/output.json'
    
```

학습이 완료되면, 수행된 시간이 나타남

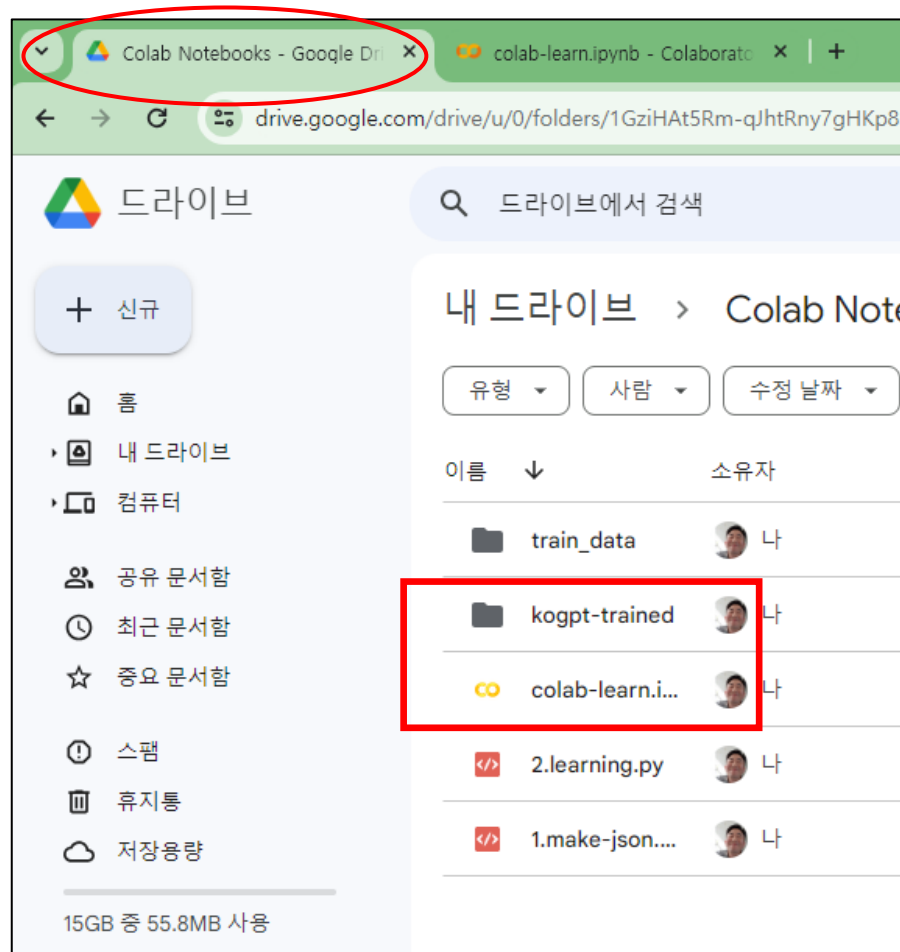
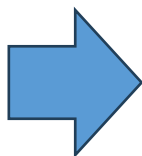
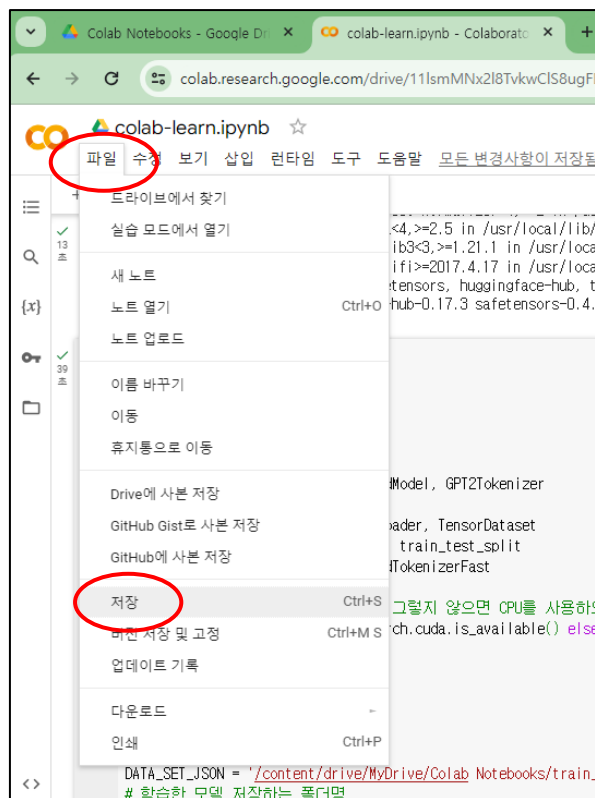
1번 학습하는 데, 약 39초 소요  
100번 학습하면, 약 3,900초 소요

학습 횟수를 100으로 변경하여,  
재 학습 수행

# \*. 전수업리뷰

## ⌚ colab 에서 학습

### ■ 프로그램 저장

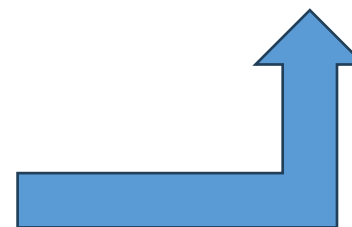
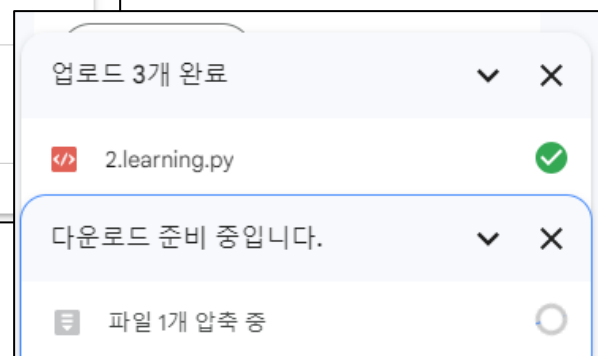
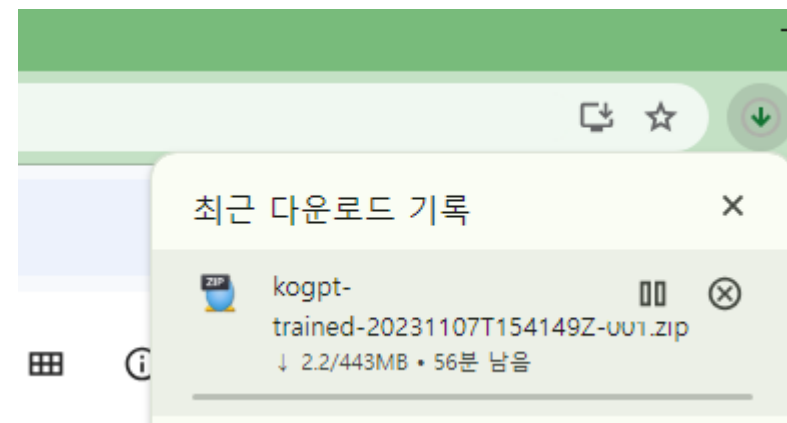
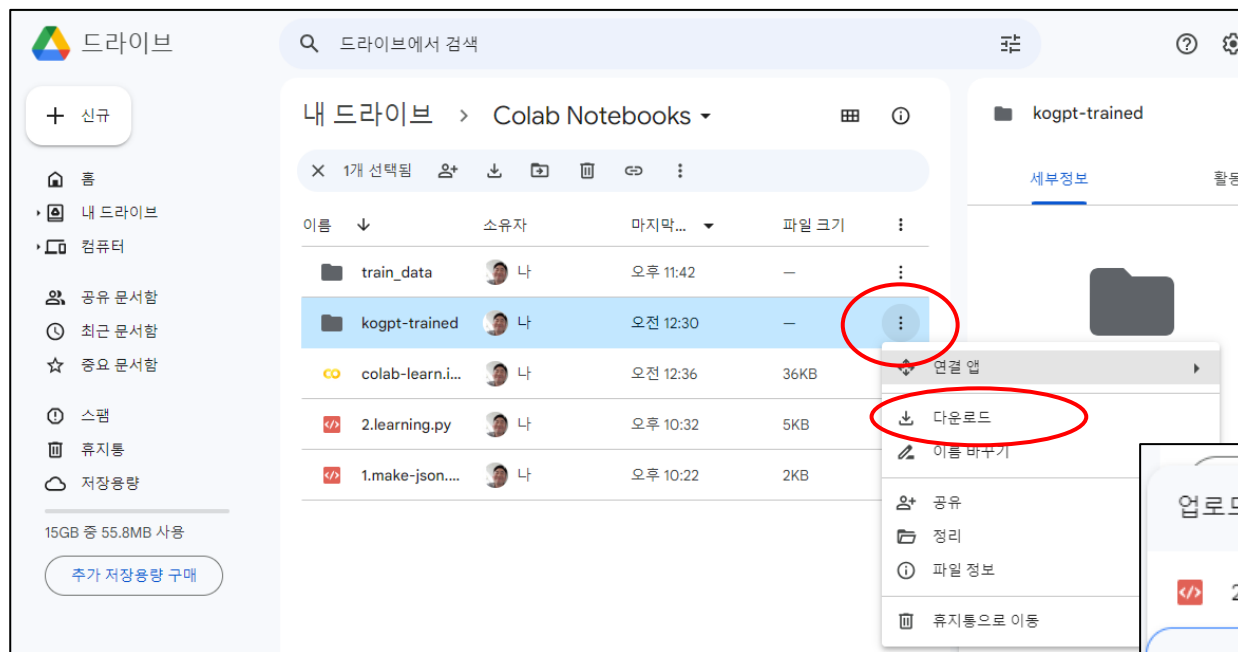


학습된 AI 폴더와  
저장된 프로그램 소스

# \*. 전수업리뷰

## ⌚ colab 에서 학습

### ■ 학습된 AI 다운로드

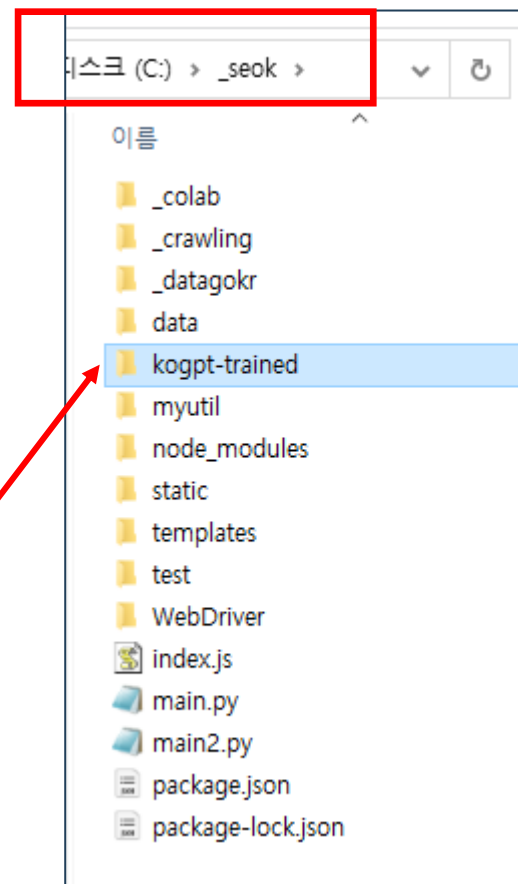
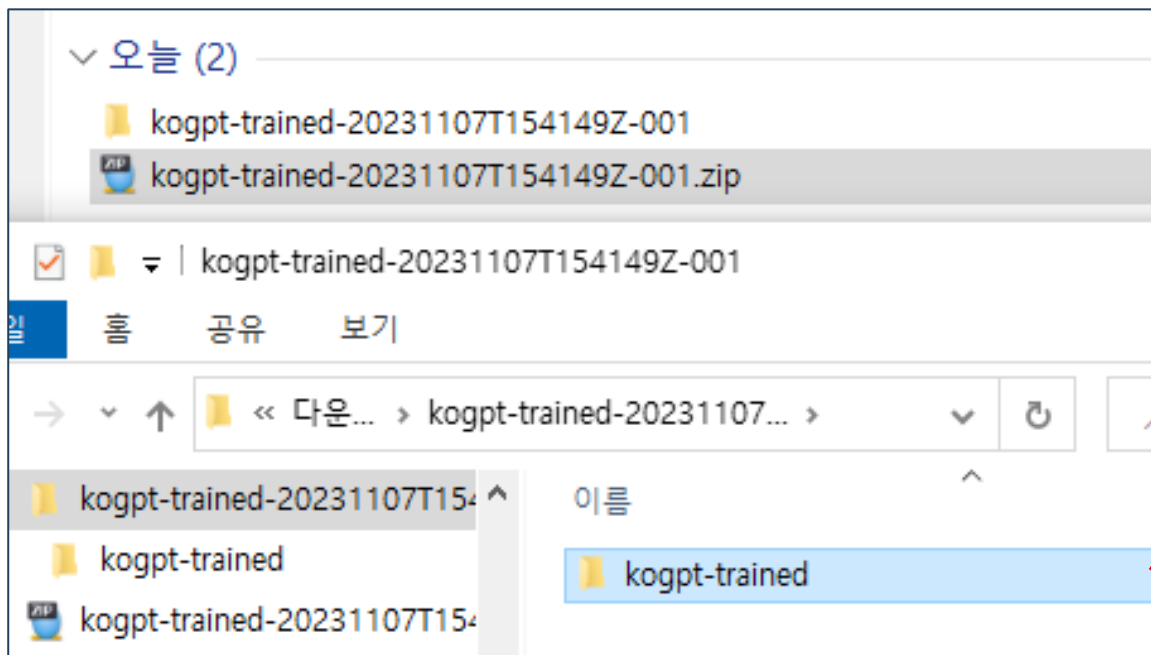


압축되어 다운로드 됨

# \*. 전수업리뷰

⌚ colab 에서 학습

- 압축 풀고, 본인의 디렉토리로 이동



# \*. 전달 사항



교재

주교재

- PowerPoint 로 만든 pdf 자료
- 데이터 과학 기반의 파이썬 빅데이터 분석 (이지영 지음, 한빛아카데미)

부교재

- 필요 시, 영상 공유



# \*. 전달 사항

## RoadMap

### Hadoop설치

- ✓ VM 셋업
- ✓ JDK
- ✓ Python
- ✓ Hadoop Engine
- ✓ Spark Engine
- ✓ Zeppelin

### 빅데이터분석

- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

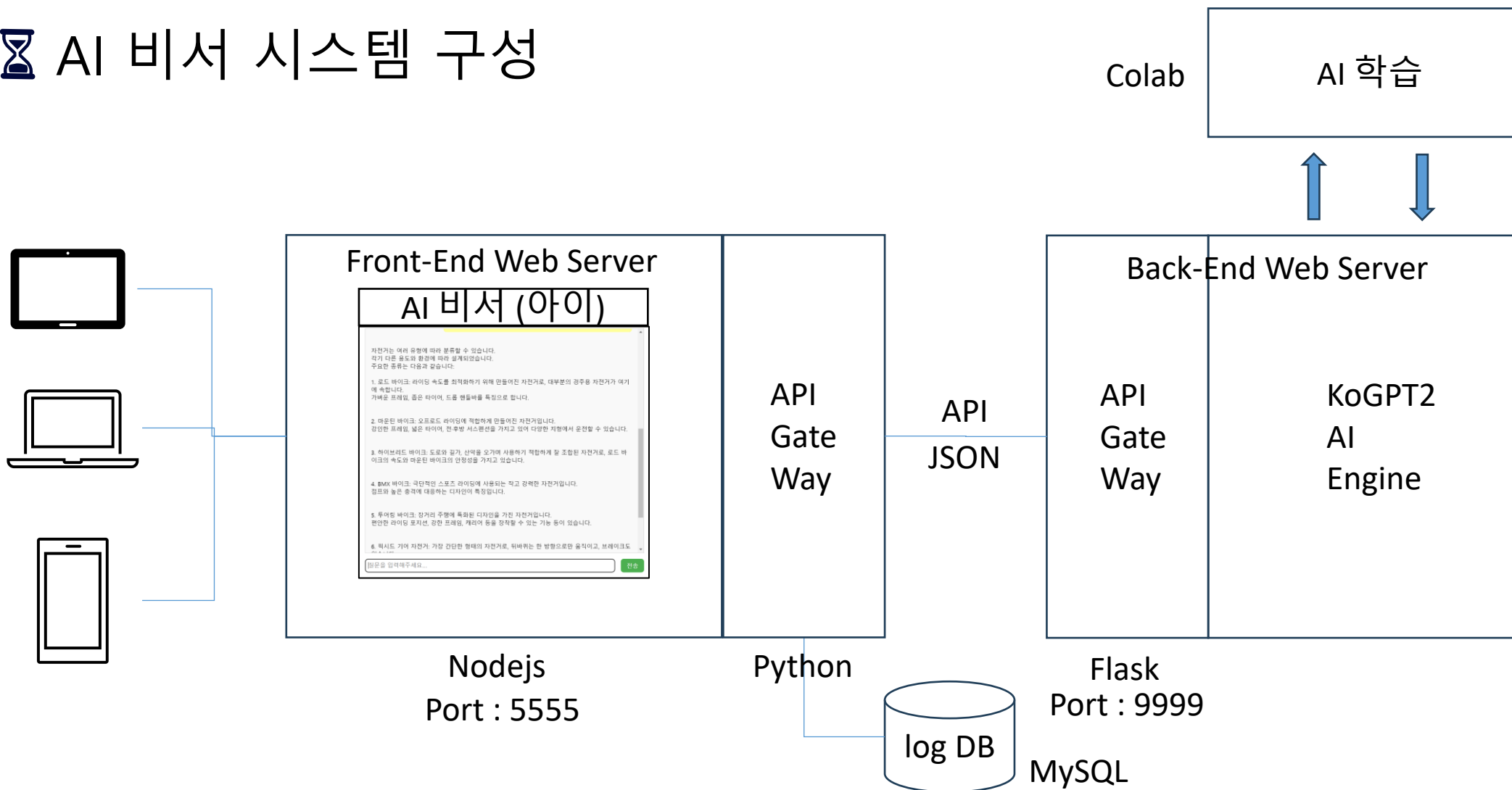
### AI 비서학습

- ✓ 챗봇 데이터 수집
- ✓ Flask 웹서버
- ✓ Nodejs API 연동
- ✓ KoGPT2 환경구성
- ✓ Colab을 이용한 학습
- ✓ 말풍선생성기 활용
- ✓ MySQL
- ✓ 챗봇 비서 만들기



# \*. 전달 사항

## ⌚ AI 비서 시스템 구성



# 1. 텍스트빈도분석

## ⌚ 환경설정

### ■ KoNLPy 패키지

- 코엔엘파이는 한국어 정보처리를 위한 파이썬 패키지
- 한국어는 전세계에서 13번째로 많이 사용되는 언어로 한국어 텍스트에서 유용한 특성을 추출하기 위해 한국어 형태소 분석기로 개발
- 오픈소스 소프트웨어로 GPL v3 라이선스에 따라 자유롭게 코드를 이용할 수 있음
- 단, 연구에 사용할 경우 아래의 논문을 인용해야 함
  - 박은정, 조성준, “KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지”, 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014.

# 1. 텍스트빈도분석

## ⌚ 환경설정

### ■ JVM 설치

```
C:\Users\dossa>java -version
java version "19" 2022-09-20
Java(TM) SE Runtime Environment (build 19+36-2238)
Java HotSpot(TM) 64-Bit Server VM (build 19+36-2238, mixed mode, sharing)
C:\Users\dossa>
```

java -version

CMD창에서

Java 가 설치 되었는 지 확인

설치가 안되어 있으면, 설치함

<https://www.oracle.com/java/technologies/downloads/>

Java Downloads | Oracle

oracle.com/java/technologies/downloads/#jdk21-windows

Java downloads Tools and resources Java archive

JDK 21 JDK 17 GraalVM for JDK 21 GraalVM for JDK 17

**JDK Development Kit 21.0.1 downloads**

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

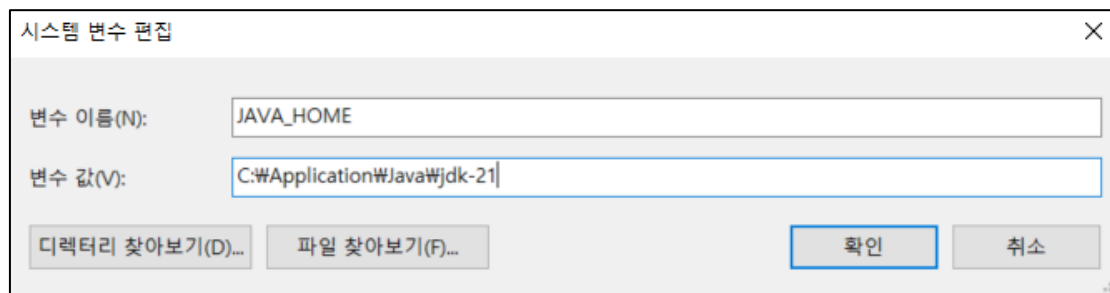
JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the limited free grants of the OTN license will require a fee.

Product/file description	File size	Download
x64 Compressed Archive	185.39 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip</a> (sha256)
x64 Installer	163.82 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	162.60 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi</a> (sha256)

# 1. 텍스트빈도분석

## ⌚ 환경설정

### ■ JAVA\_HOME / CLASSPATH 설정



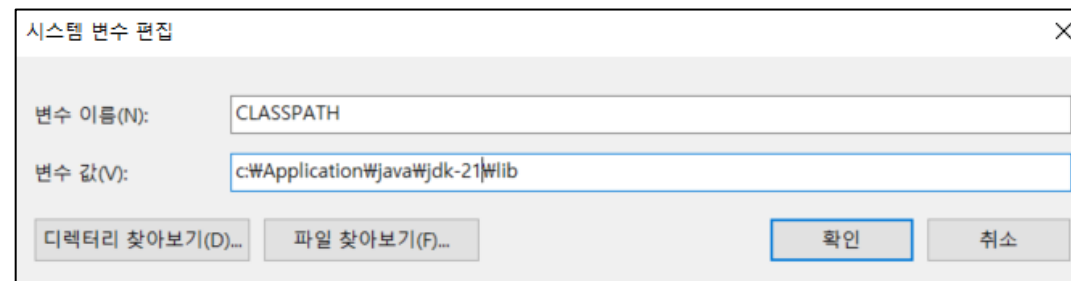
제어판 – 시스템 – 고급시스템설정 – 환경변수 – 시스템변수 새로만들기

JAVA\_HOME

c:\Application\Java\jdk-21 ← 본인 PC 에 설치된 경로

```
C:\Users\dossa>
C:\Users\dossa>java -version
java version "21.0.1" 2023-10-17 LTS
Java(TM) SE Runtime Environment (build 21.0.1+12-LTS-29)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.1+12-LTS-29, mixed mode, sharing)
C:\Users\dossa>
```

설치 확인 → java -version



제어판 – 시스템 – 고급시스템설정 – 환경변수 – 시스템변수 새로만들기

CLASSPATH

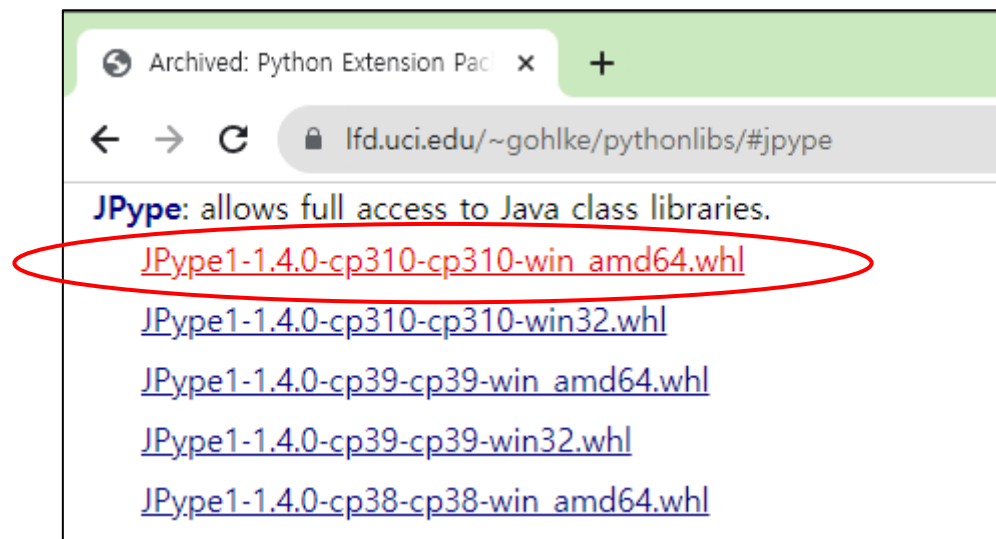
c:\Application\Java\jdk-21\lib ← 본인 PC 에 설치된 경로

# 1. 텍스트빈도분석

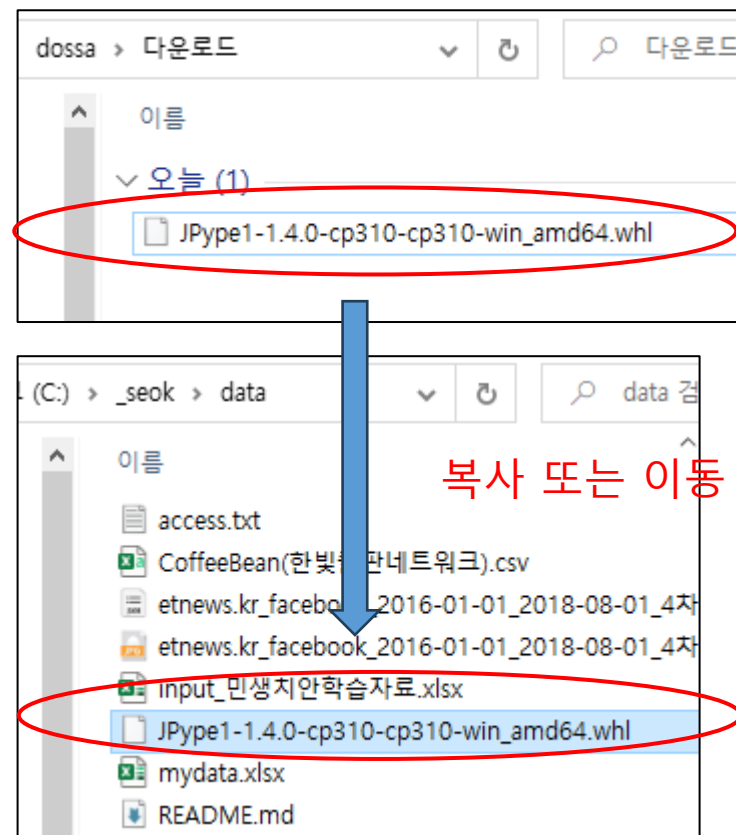
## ⌚ 환경설정

- JPype1 설치 : 파이썬에서 JDK를 사용

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype>



파이썬 3.10 버전에 맞는 패키지 다운로드



# 1. 텍스트빈도분석

## ⌚ 환경설정

- JType1 설치 : 파이썬에서 JDK를 사용

CMD 창을 열고,

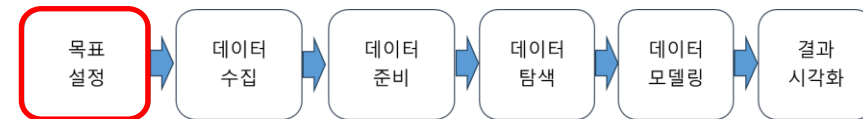
```
cd W_seok\data
```

```
python -m pip install --upgrade pip
```

```
python -m pip install JType1-1.4.0-cp310-cp310-win_amd64.whl
```

```
python -m pip install konlpy
```

# 1. 텍스트빈도분석



## ⌚ 목표설정

### ■ 핵심 개념 이해

[형태소와 형태소 분석]

- 형태소란 언어에서 의미가 있는 가장 작은 단위
- 단어는 의미를 갖는 문장의 가장 작은 단일 요소로, 문장에서 분리될 수 있는 부분
- 독립형 형태소인 단어도 있지만, 대부분의 단어는 형태소와 접사로 구성
- 형태소 분석이란 형태소, 어근, 접두사/접미사, 품사 등 다양한 언어학적 속성으로 구조를 나누어 파악하는 것

# 1. 텍스트빈도분석



## ⌚ 목표설정

### ■ 핵심 개념 이해

[품사 태깅]

- 형태소의 뜻과 문맥을 고려하여 품사를 붙이는 것

예) 가방에 들어가신다 → 가방/NNG + 에/JKM + 들어가/VV + 시/EPH + 니다/EFN

[품사 태깅 패키지]

- KoNLPy에서 사용 가능한 품사 태깅 패키지
  - Hannanum, Kkma, Komoran, Mecab, Okt(Twitter) 등



# 1. 텍스트빈도분석

## ⌚ 목표설정

### ■ 핵심 개념 이해



None:명사  
 Verb:동사  
 Adjective:형용사  
 Determiner:관형사  
 Adverb:부사  
 Conjunction:접속사  
 Josa:조사  
 Eomi:어미

Hannanum	Kkma	Komoran	Mecab	Twitter (Okt)
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

Hannanum	Kkma	Komoran	Mecab	Twitter
나 / N	나 / NP	나 / NP	나 / NP	나 / Noun
는 / J	는 / JX	는 / JX	는 / JX	는 / Josa
밥 / N	밥 / NNG	밥 / NNG	밥 / NNG	밥 / Noun
을 / J	을 / JKO	을 / JKO	을 / JKO	을 / Josa
먹 / P	먹 / VV	먹 / VV	먹 / VV	먹는 / Verb
는다 / E	는 / EPT	는다 / EC	는다 / EC	다 / Eomi
	다 / EFN			

# 1. 텍스트빈도분석



## ⌚ 데이터 수집

- 데이터셋 : 페이스북과 전자신문 페이지에서 크롤링한 4차산업혁명 기사

CMD 창을 열고

cd ₩

git clone https://github.com/sEOKiLL-jEONG/bigdata.git \_bigdata

cd \_bigdata

dir

```

명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dossa>cd ₩

C:\₩>git clone https://github.com/sEOKiLL-jEONG/bigdata.git _bigdata
Cloning into '_bigdata'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
Receiving objects: 100% (11/11), 208.53 KiB | 10.43 MiB/s, done.
Resolving deltas: 100% (2/2), done.

C:\₩>cd _bigdata

C:\₩_bigdata>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F6D9-BD25

C:\₩_bigdata 디렉터리

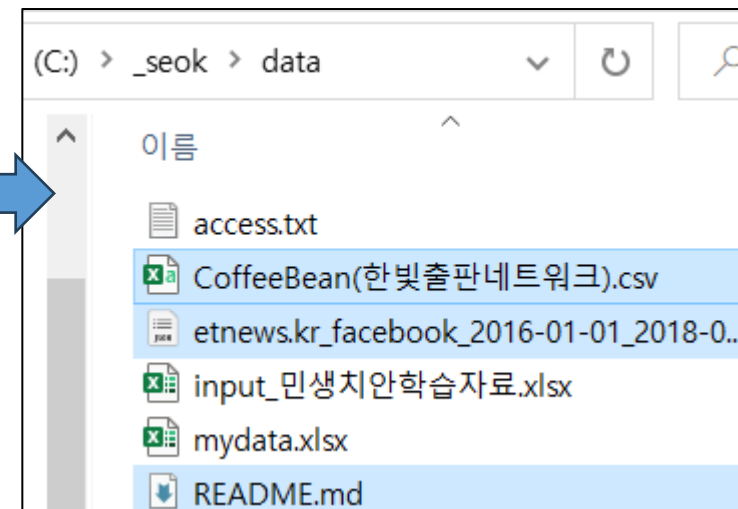
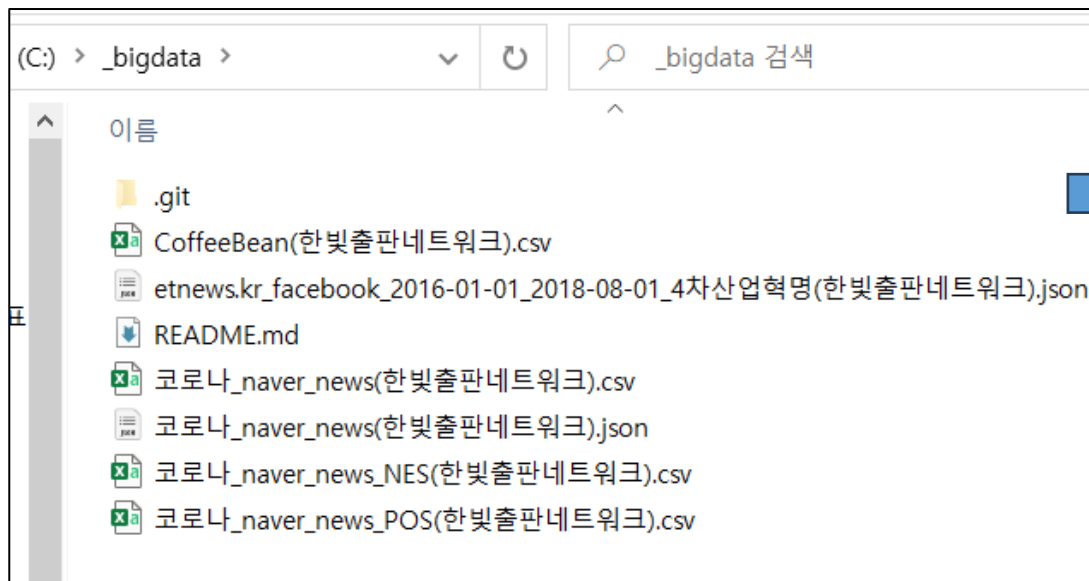
2023-11-12 오후 04:20 <DIR> .
2023-11-12 오후 04:20 <DIR> ..
                21,678 CoffeeBean(한빛출판네트워크)
                11,644 etnews.kr_facebook_2016-01-0
                9 README.md
                206,140 코로나_naver_news(한빛출판네
                400,050 코로나_naver_news(한빛출판네
                110,834 코로나_naver_news_NES(한빛출
  
```

# 1. 텍스트빈도분석



## ⌚ 데이터 수집

- 데이터셋 : 페이스북과 전자신문 페이지에서 크롤링한 4차산업혁명 기사



₩\_bigdata 밑에 있는 파일을

자신의 ₩\_seok₩data 디렉토리로 이동



# 1. 텍스트빈도분석

## ⌚ 데이터 준비

- csv 파일을 읽어서 내용 확인

python -m pip install konlpy  
python -m pip install wordcloud

\_seok\data\word-01-ready.py 로 저장

```
#=====
# csv 파일을 읽어, 내용확인
#=====
import json
import re
from konlpy.tag import Okt
from collections import Counter
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from wordcloud import WordCloud
```

```
inputFileName = './etnews.kr_facebook_2016-01-01_2018-08-01_4차산업혁명(한빛출판네트워크)'
data = json.loads(open(inputFileName+'.json', 'r', encoding = 'utf8').read())
print('data = ', data)
```

# 1. 텍스트빈도분석

## ⌚ 데이터 준비

- csv 파일을 읽어서 내용 확인

```

C:\seok\data>python word-01-ready.py
[{'created_time': '2018-06-20 18:06:39', 'link': 'https://www
': '6월의 스파크포럼 - "미래 시대, 조직의 변화도 시작됐다!"\n
슈를 혁신적으로 해결하고자 하는 소셜이노베이터를 발굴, 지원하
만남의 자리입니다.\n\n6월의 스파크포럼에서는 4차 산업혁명 시대
고 실험해나가는 사례를 통해 미래 시대 조직이 나아가야 할 방향:
"미래 시대, 조직의 변화도 시작됐다!", 'post_id': '4078867059'
me': '2018-06-14 10:41:16', 'link': 'http://www.etnews.com/201
력으로 떠오르면서 국내 로봇 기업에 재평가가 이뤄지고 있다는 분
'post_id': '407886705912407_1971252229575839', 'total_comments
tp://www.etnews.com/20180409000360?mc=em_002_00001', 'message
name': "회청거리는 대한민국 '금융'... 서민도 기업도 ICT도 실종'

```



```

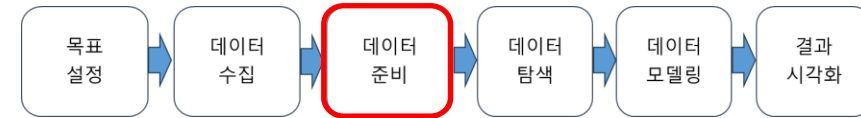
[{'created_time': '2018-06-20 18:06:39',
'link': 'https://www.facebook.com/etnews.kr/videos/1981346601899735/',
'message': '6월의 스파크포럼 - "미래 시대, 조직의 변화도 시작됐다!"\n
\n스파크포럼은 현 사회의 사회문제 및 이슈를 제기하고,
그 이슈를 혁신적으로 해결하고자 하는 소셜이노베이터를
발굴, 지원하여 우리 사회 따뜻한 변화를 확산시키지 위해
만들어진 도전과 만남의 자리입니다.\n
\n6월의 스파크포럼에서는 4차 산업혁명 시대의 기업조직과
조직문화를 살펴보고, 조직의 변화를 받아들이고 실험해
나가는 사례를 통해 미래 시대 조직이 나아가야 할 방향을
함께 생각해보고자 합니다.',
'name': '6월의 스파크포럼 - "미래 시대, 조직의 변화도 시작됐다!"',
'post_id': '407886705912407_1981346601899735',
'total_comments': 3},

```

```

{'created_time': '2018-06-14 10:41:16',
'link': 'http://www.etnews.com/20180612000347',

```



# 1. 텍스트빈도분석

## ⌚ 데이터 준비

- 메시지 키의 값(본문 내용)을 읽어서, 명사만 추출

<계속>

```
message = ""
for item in data:
    if 'message' in item.keys():
        message = message + re.sub(r'^\w', ' ', item['message']) + "
print('message = ', message)
print('='*40)

nlp = Okt()
message_N = nlp.nouns(message)
print('message_N = ', message_N)
print('='*40)
```

6월의 스파크포럼 미래 시대 조직의 변화도 시작됐다 스파크포럼은 혁신적으로 해결하고자 하는 소셜이노베이터를 발굴 지원하여 우리 사회의 자리입니다 6월의 스파크포럼에서는 4차 산업혁명 시대의 기업조직과 해나가는 사례를 통해 미래 시대 조직이 나아가야 할 방향을 함께 생각해 보며 떠오르면서 국내 로봇 기업에 재평가가 이뤄지고 있다는 분석입니다 4차 산업혁명 간판 공약은 4차 산업혁명 선도 도시 구현 과 가상화폐 블록체인 기술



['스파크', '포럼', '미래', '시대', '조직', '변화', '시작', '스파크', '제기', '그', '이슈', '혁신', '해결', '소셜', '이노', '베이', '터', '해', '도전', '만남', '자리', '스파크', '포럼', '차', '산업', '혁명', '사', '사례', '통해', '미래', '시대', '조직', '방향', '생각', '로', '차', '기업', '재', '평가', '분석', '차', '산업', '혁명', '시대', '혁신', '기술', '산업', '혁명', '선도', '도시', '구현', '과', '가상', '화폐', '블록', '체인', '기술', '산업', '혁명', '핵심', '기술', '대한', '통찰', '알파', '통해', '우리']

문자나 숫자가 아닌 것은 공백으로 치환



# 1. 텍스트빈도분석

## ⌚ 데이터 탐색

- message\_N의 단어들을 탐색 : 단어별로 출현 횟수 카운트

<계속>

```

count = Counter(message_N)
print('count =', count)
print('='*40)

word_count = dict()
for tag, counts in count.most_common(80):
    if( len ( str ( tag ) ) > 2 ):
        word_count[tag] = counts
        print("%s : %d" % (tag, counts))
print('='*40)
    
```

```

산업혁명 : 22
전자신문 : 13
조선일보 : 10
산업일보 : 10
기술신문 : 8
시대일보 : 7
라이프 : 7
이진일보 : 6
한겨레 : 6
포털뉴스 : 5
일자리 : 5
조직문화 : 4
통해 : 4
진행 : 4
스파크 : 3
미래 : 3
    
```

상위 80개의 단어 중에서, 글자가 2자 보다 큰 것만 딕셔너리에 저장

# 1. 텍스트빈도분석

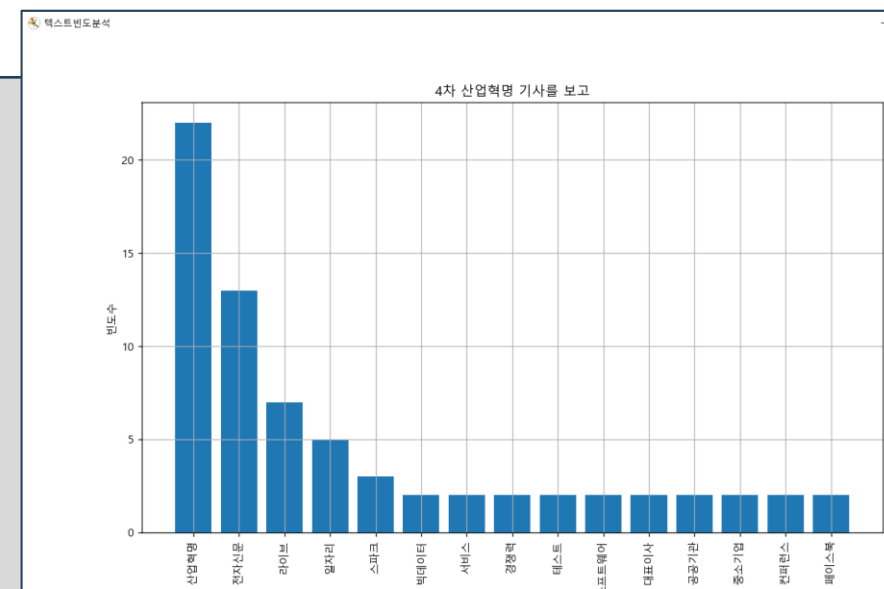
## ⌚ 데이터 모델링

- 히스토그램으로 단어 빈도를 탐색

<계속>

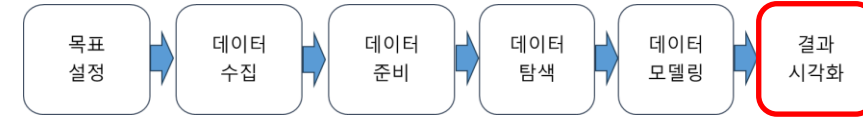
```
font_path = "c:/Windows/fonts/malgun.ttf" # 맑은고딕체를 설정
font_name = font_manager.FontProperties(fname = font_path).get_name()
matplotlib.rc('font', family = font_name)

plt.figure(num='텍스트빈도분석', figsize = (12, 7))
plt.title('4차 산업혁명 기사를 보고')
plt.xlabel('키워드')
plt.ylabel('빈도수')
plt.grid(True)
sorted_keys = sorted(word_count, key = word_count.get, reverse = True)
sorted_values = sorted(word_count.values(), reverse = True)
plt.bar(range(len(word_count)), sorted_values, align = 'center')
plt.xticks(range(len(word_count)), list(sorted_keys), rotation = 'vertical') #horizontal
plt.show()
```





# 1. 텍스트빈도분석



## ⌚ 결과 시각화

- 워드클라우드로 시각화

<계속>

#워드클라우드로 시각화

```
wc = WordCloud(font_path, background_color = 'ivory', width = 800, height = 600)
```

```
cloud = wc.generate_from_frequencies(word_count)
```

```
plt.figure(num='텍스트빈도분석', figsize = (6, 6))
```

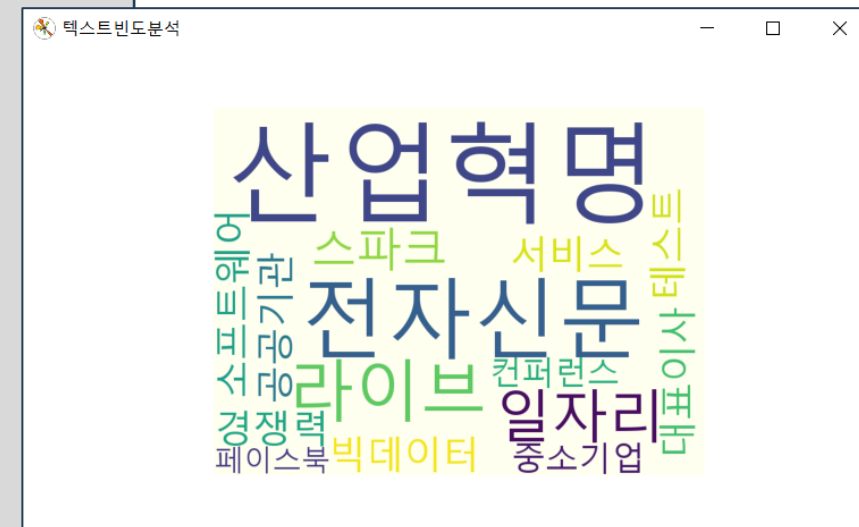
```
plt.imshow(cloud)
```

```
plt.axis('off') # 아웃라인
```

```
plt.show()
```

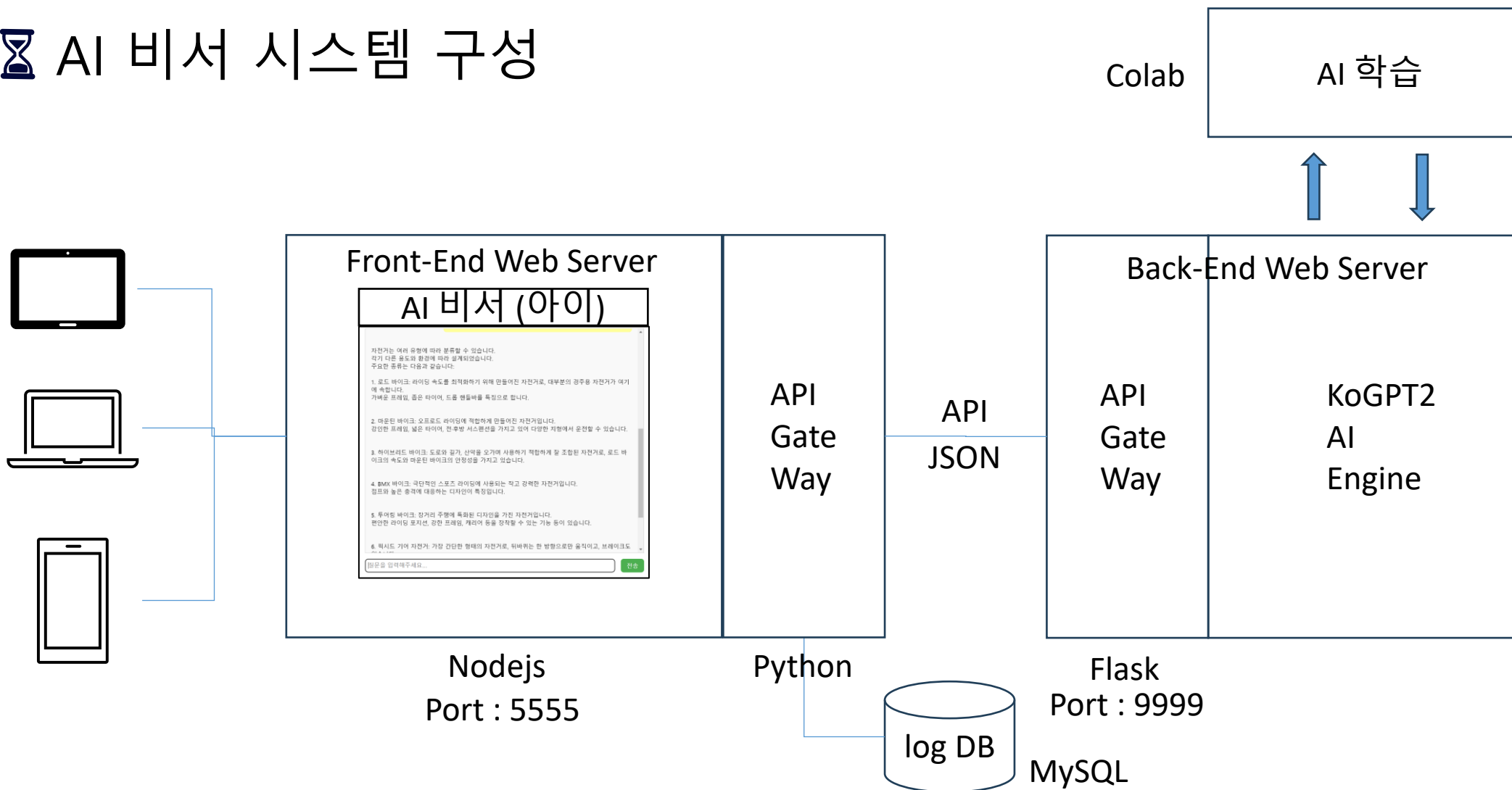
# 이미지로 저장

```
cloud.to_file(inputFileName + '_cloud.jpg')
```



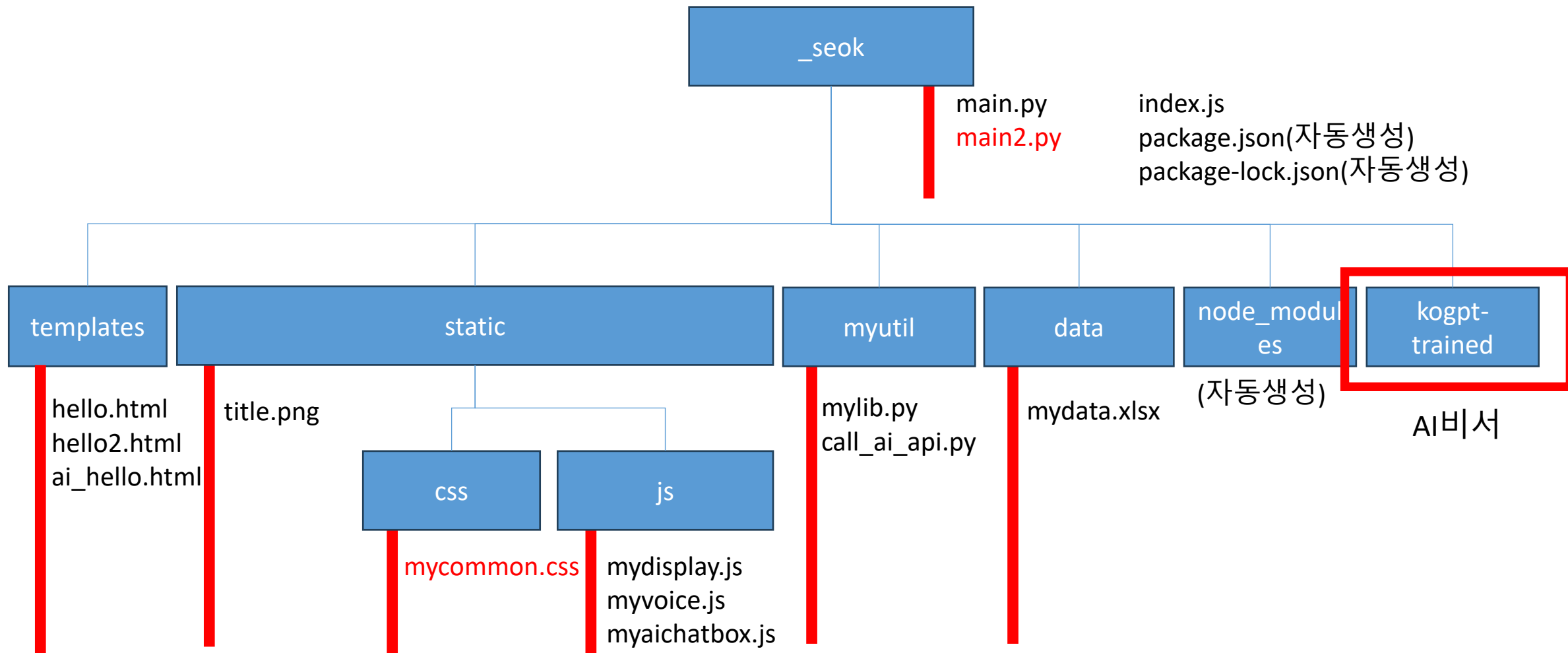
## 2. AI 비서

### ⌚ AI 비서 시스템 구성



## 2. AI 비서

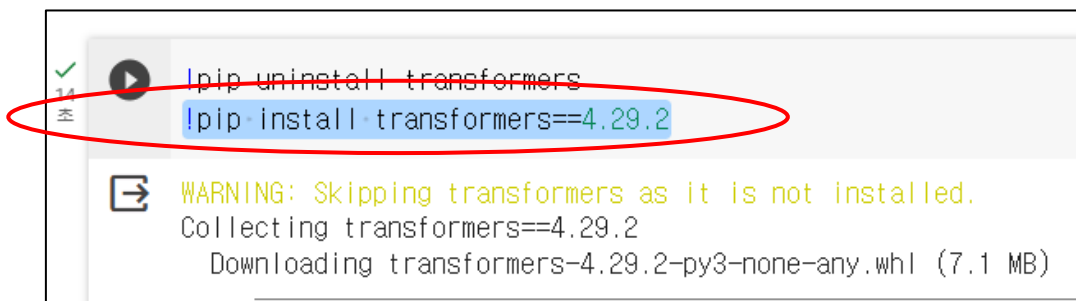
### ⌚ 디렉토리 구성



## 2. AI 비서

### ⌚ colab 에서 학습 시 유의사항

#### ■ 패키지 설치



추가된 셀에서 패키지를 설치 함  
(최신버전으로 하지 말고, 버전 고정)

```
!pip install transformers==4.29.2
```

<기존방식>  
pytorch\_model.bin 이 생성됨

safetensors 형식과 ckpt(pickle 형식의 비교)

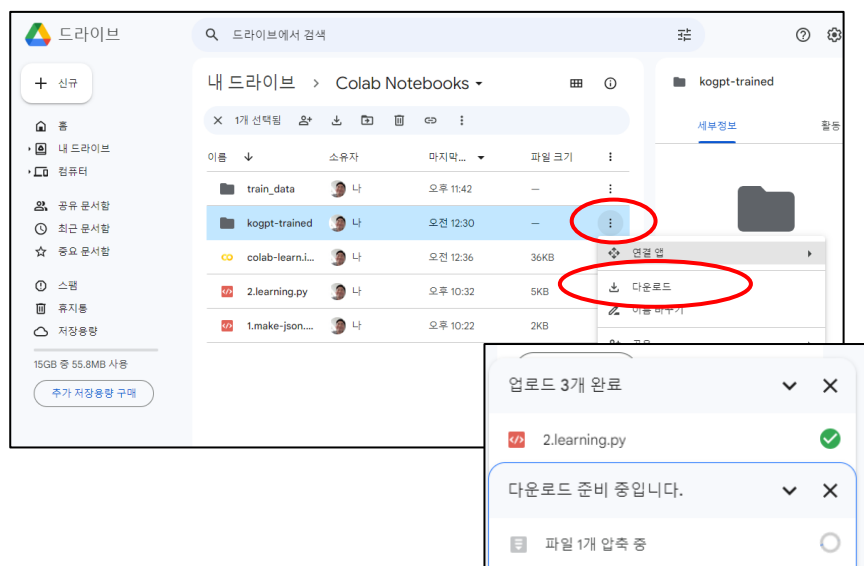
Format	Safe	Zero-copy	Lazy loading	No file size limit	Layout control	Flexibility	Bfloat16
<b>pickle (pyTorch)</b>	X	X	X	O	X	O	O
H5 (Tensorflow)	O	X	O	O	-	-	X
SavedModel(Tensorflow)	O	X	X	O	O	X	O
MsgPack (flax)	O	O	X	O	X	X	O
Protobuf (ONNX)	O	X	X	X	X	X	O
Cap'n Proto	O	O	-	O	O	-	X
Arrow	?	?	?	?	?	?	X
Numpy (npz, npz)	O	?	?	X	O	X	X
<b>SafeTensors</b>	O	O	O	O	O	X	O

<새로운 방식>  
model.safetensors 가 생성됨

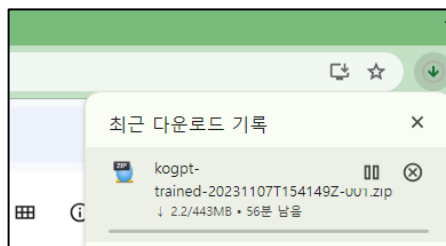
## 2. AI 비서

⌚ colab 에서 학습한 모델

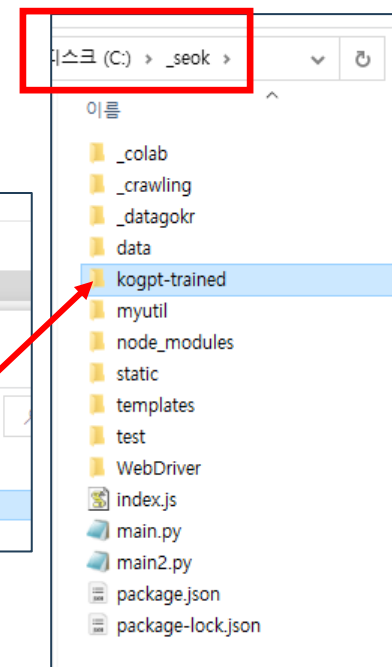
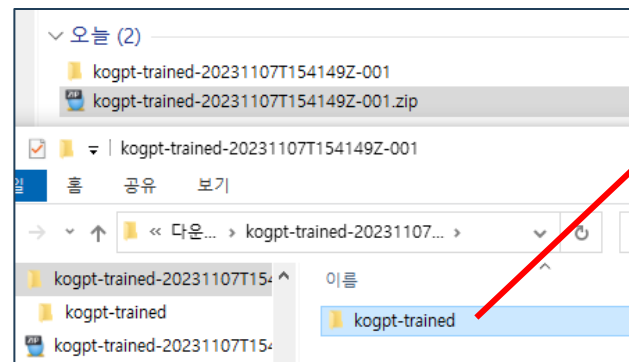
- 다운로드, 압축 풀고, 이동



압축되어 다운로드 됨



압축풀고 본인의 작업  
폴더로 이동



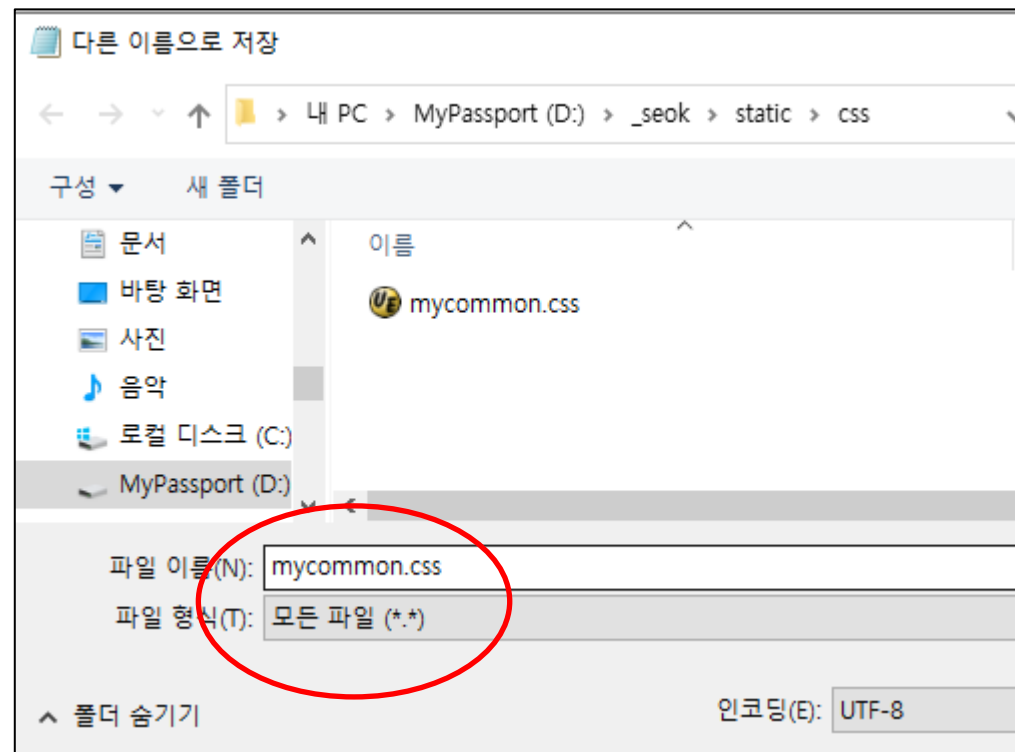
## 2. AI 비서

### ⌚ 프로그래밍코딩

- mycommon.css 파일 수정 - 아래 **붉은색** 내용 추가

```
/*=====*/
/* 기본 CSS 설정 */
/*=====*/
body {
  font-family: sans-serif;
  margin: 0;
  padding: 0;
  background-color: #ffc000;
  font-size: 18px;
}

.bot-message {
  align-self: flex-end;
  font-weight: bold; /* 글자를 bold로 처리 */
  background-color: #a2ff99; /* 배경색 */
  padding: 0.5rem;
  border-radius: 0.5rem;
  max-width: 70%;
  word-wrap: break-word;
}
```



**\_seok\static\css 밑에 mycommon.css 로 저장**

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```
#=====
# Flask 웹서버 메인 프로그램
#=====
import sys, os
import socket
import pandas as pd
import requests
from flask import Flask, render_template, request, jsonify
from myutil.mylib import mylib_Read_xlsx_Data, mylib_ViewPage
#-----
# KoGPT2
#-----
import torch
from transformers import GPT2LMHeadModel
from transformers import PreTrainedTokenizerFast
```

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```
#--여기서부터 추가-Part-1-----
from copy import deepcopy
from transformers import AutoTokenizer
from transformers import BertTokenizer, BertForNextSentencePrediction
from KoChatGPT.colossalai_ChatGPT_230319.chatgpt.models.gpt import GPTActor, GPTCritic
from chatgpt_api.chatgpt_api_service import query_chatgpt
#--여기까지 추가-Part-1-----

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print('***** my device = ', device)

PRETRAINED_MODEL = "skt/kogpt2-base-v2"
```



## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

4칸

```

#--여기서부터 추가-Part-2-----
MY_AI_MODEL = './kogpt-trained'
LORA_RANK = 0

actor = GPTActor(pretrained=MY_AI_MODEL, lora_rank=LORA_RANK).to(device)
tokenizer = AutoTokenizer.from_pretrained(PRETRAINED_MODEL, padding_side="right", model_max_length=5000)
tokenizer.add_special_tokens({
    "eos_token": "</s>",
    "bos_token": "</s>",
    "unk_token": "</s>",
})
tokenizer.pad_token = tokenizer.eos_token
initial_model = deepcopy(actor)
    
```

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```

4칸 8칸 12칸
def request_AI(input_text):
    input_ids = tokenizer.encode(input_text, return_tensors='pt').to(device)
    outputs = actor.generate(input_ids,
                             max_length=512,
                             do_sample=True,
                             top_k=50,
                             top_p=0.3,
                             temperature=0.3,
                             eos_token_id=tokenizer.eos_token_id,
                             pad_token_id=tokenizer.pad_token_id,
                             num_return_sequences=1)
    output = tokenizer.batch_decode(outputs[0], skip_special_tokens=True)[0]
    if output.find(input_text) == 0:
        output = output[len(input_text):]
        if output == '':
            output = "아직 학습을 하지 못 했습니다. 다른 질문을 해주세요~"
    return output
#--여기까지 추가-Part-2-----

```

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색 내용 추가**, **파란색 내용 수정**

''' 이 부분은 막음

```
tokenizer = PreTrainedTokenizerFast.from_pretrained(PRETRAINED_MODEL,
                                                    bos_token='</s>', eos_token='</s>', unk_token='<unk>',
                                                    pad_token='<pad>', mask_token='<mask>')
```

```
model = GPT2LMHeadModel.from_pretrained(PRETRAINED_MODEL)
def request_AI(_req):
    _res = model.generate(_req,
                          max_length=128,
                          repetition_penalty=2.0,
                          pad_token_id=tokenizer.pad_token_id,
                          eos_token_id=tokenizer.eos_token_id,
                          bos_token_id=tokenizer.bos_token_id,
                          use_cache=True)

    return _res
```

이 부분은 막음  
'''

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```
#-----
4칸 8칸
app = Flask(__name__)
#=====
@app.route('/', methods=['POST', 'GET'])
def home():
    return render_template('hello2.html')
#=====
@app.route('/get_data', methods=['POST'])
def get_data():
    try:
        input_data = request.form["input_data"]
        print('***input_data : ', input_data)

        _file = './data/mydata.xlsx'
        _list = mylib_Read_xlsx_Data(_file)
        df = pd.DataFrame(_list[1:], columns=_list[0])
        result = mylib_ViewPage(df, input_data)
```

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```

4칸      8칸
    return result

    except Exception as ee:
        print('***error : ', ee)
#=====
@app.route('/api/get_data', methods=['POST'])
def api_page():

    question = request.json['question']
    print('***question : ', question)
    apikey = request.json['key']
    if apikey != 'AAAAAAAAAAAAABBBCCC111':
        return jsonify({'answer': 'not supported'})

    answer = ""

```

## 2. AI 비서

### ⌚ 프로그램코딩

- main2.py 파일 수정 – 아래 **붉은색** 내용 추가, **파란색** 내용 수정

```

4칸 8칸
try:
    #input_ids = tokenizer.encode(question, return_tensors='pt')
    answer = request_AI(question)
    #answer = tokenizer.decode(generated[0])

    #answer = question + " : 학습이 된 후에 답변 드리겠습니다~"
except Exception as ee:
    answer = "오류가 발생했습니다~" + ee

print('***answer : ', answer)

sys.stdout.flush()
return jsonify({'answer': answer})
#=====
if __name__ == '__main__':
    #_myip = socket.gethostbyname(socket.gethostname())
    app.run(host='172.16.11.220', port=9999, debug=False)

```

## 2. AI 비서

### ⌚ KoGPT 패키지 설치

- 관련 패키지 다운로드

CMD 창을 열고

cd ₩

git clone https://github.com/sEOKiLL-jEONG/aitrain.git \_aitrain

cd \_aitrain

dir

```
명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\₩Users\₩dossa>cd ₩

C:\₩>git clone https://github.com/sEOKiLL-jEONG/aitrain.git _aitrain
Cloning into '_aitrain'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
Receiving objects: 87% (7/8), 7.78 MiB | 7.53 MiB/s, reused 0
Receiving objects: 100% (8/8), 12.05 MiB | 7.95 MiB/s, done.

C:\₩>cd _aitrain

C:\₩_aitrain>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: F6D9-BD25

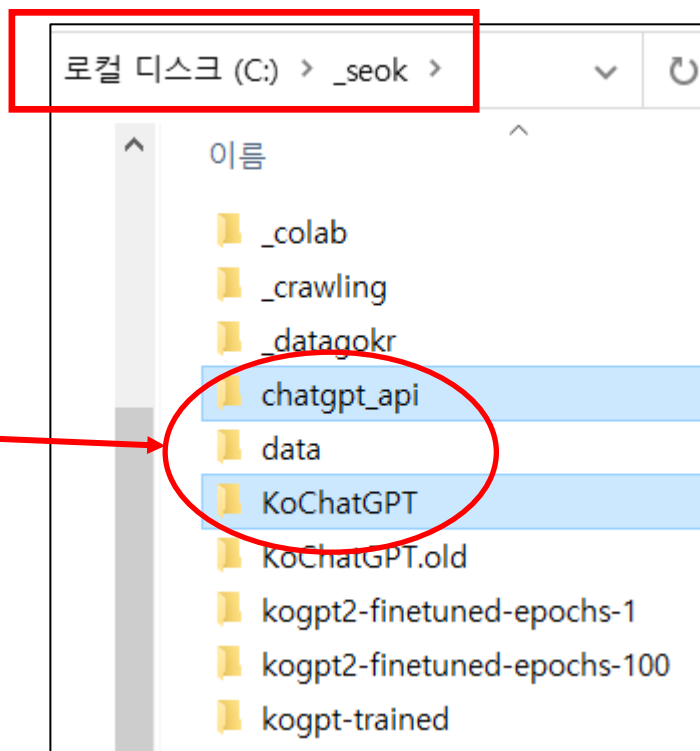
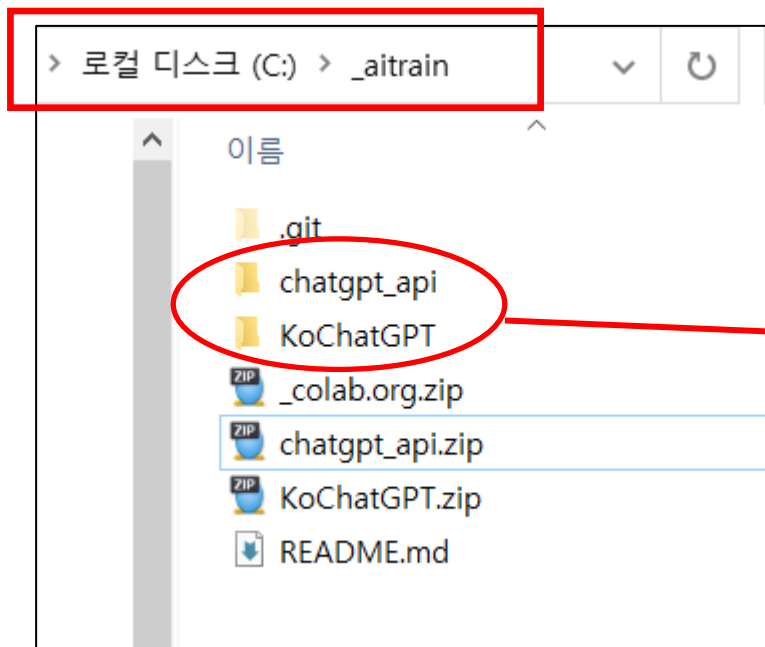
C:\₩_aitrain 디렉터리

2023-11-13 오후 01:47 <DIR> .
2023-11-13 오후 01:47 <DIR> ..
2023-11-13 오후 01:47      2,292 chatgpt_api.zip
2023-11-13 오후 01:47    12,639,340 KoChatGPT.zip
2023-11-13 오후 01:47           9 README.md
2023-11-13 오후 01:47    23,272 _colab.org.zip
                4개 파일      12,664,913 바이트
                2개 디렉터리 110,247,452,672 바이트 남음
```

## 2. AI 비서

### ⌚ KoGPT 패키지 설치

- 압축 풀고 이동





## 2. AI 비서

### ⌚ 실행

#### ■ nodejs 실행

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dossa>cd \_seok

C:\_seok>npm start

> myai@1.0.0 start
> nodemon index.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server listening on port 5555
```

cd \\_seok

npm start

#### ■ Flask 실행

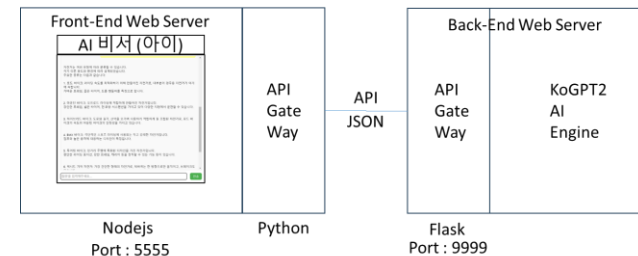
```
C:\_seok>명령 프롬프트 - python main2.py

Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\_seok>python main2.py
* Serving Flask app 'main2'
* Debug mode: off
WARNING: This is a development server. Do not use without proper
security measures.
* Running on http://172.16.11.220:9999
Press CTRL+C to quit
```

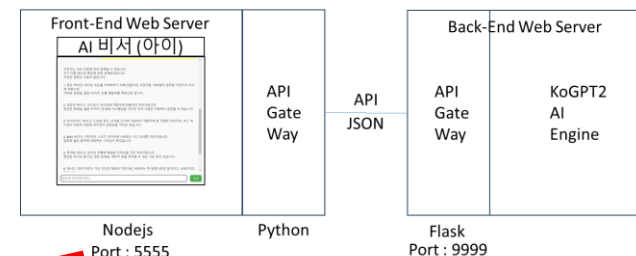
cd \\_seok

python main2.py

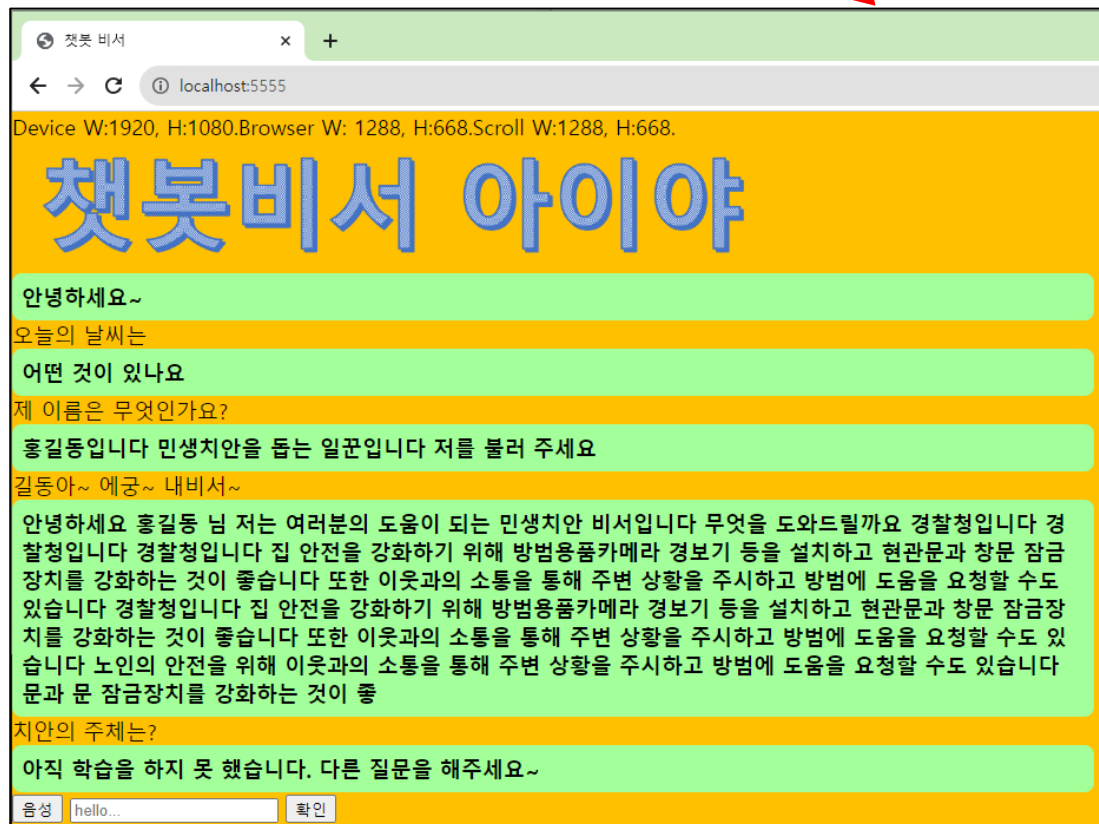


## 2. AI 비서

### ⌚ 접속



<http://localhost:5555/>



# 기타 프로그램 설치

- Git 다운로드 : <https://git-scm.com>

# 참고 자료

- 자바와 파이썬으로 만드는 빅데이터시스템(제이펍, 황세규)
- 위키독스(<https://wikidocs.net/22654>)
- 네이버블로그(<https://blog.naver.com/classmethodkr/222822485338>)
- 데이터분석과 인공지능 활용 (NOSVOS, 데이터분석과인공지능활용편찬위원회 편)

## 참고 사이트

유튜버 : 빅공잼 : <https://www.youtube.com/watch?v=bnYxO2XRCQ0>

네이버 블로그 : 빅공잼

<https://biggongjam.notion.site/3-Hadoop-cd6944182da74edf8d2339b654e0bfb9>

<https://biggongjam.notion.site/4-Spark-2c341ddc8715411484cb2f0254b60126>

Q n A