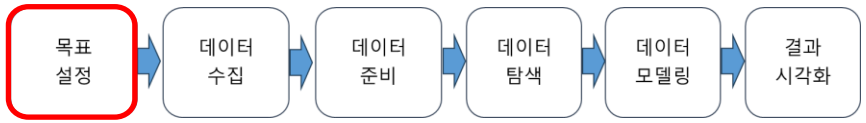
빅데이터시스템

텍스트마이닝-감성분석

education.drone@gmail.com
by sEOK-iLL, jEONG

목 차



*. 전수업리뷰

⏳ 회귀분석 - 목표설정 **- 자동차 연비 예측**

- 목표

  - 자동차 연비 데이터에 머신러닝 기반의 회귀 분석을 수행
  - 연비에 영향을 미치는 항목을 확인하고, 그에 따른 자동차 연비를 예측

- 핵심 개념 이해

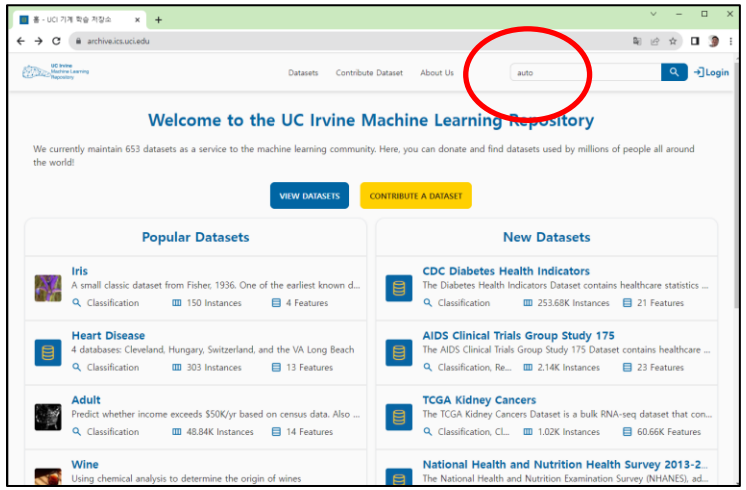
  - 학습을 하기 위한 훈련 데이터에 입력과 출력을 같이 제공
  - 문제(입력)에 대한 답(출력, 결과값)을 아는 상태에서 학습하는 방식 이해
  - 입력 : 예측 변수, 속성, 특징
  - 출력 : 반응 변수, 목표 변수, 클래스, 레이블

*. 전수업리뷰

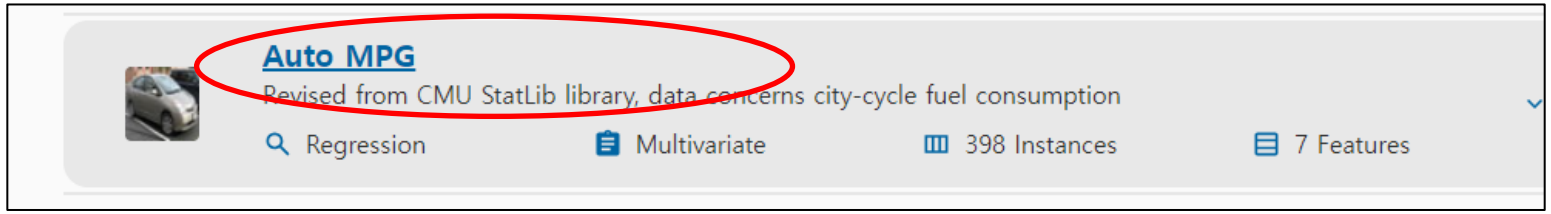
⏳ 데이터 수집

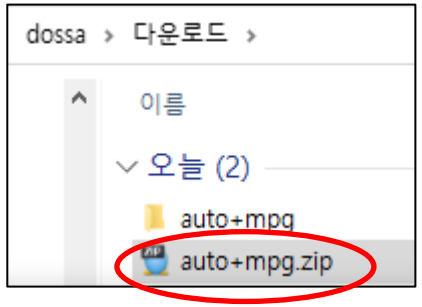
- UCI 머신러닝 레퍼지터리

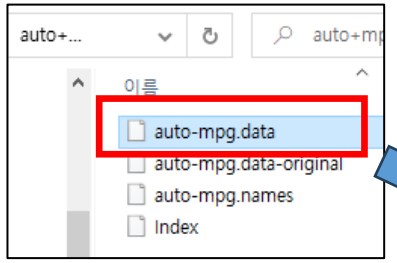
https://archive.ics.uci.edu/

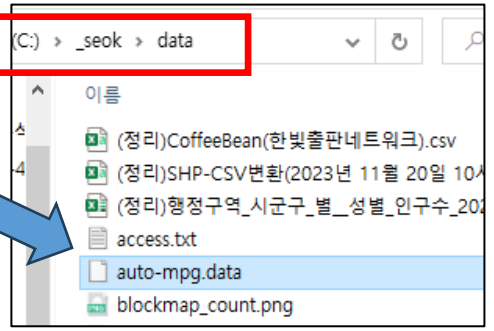


auto 로 검색







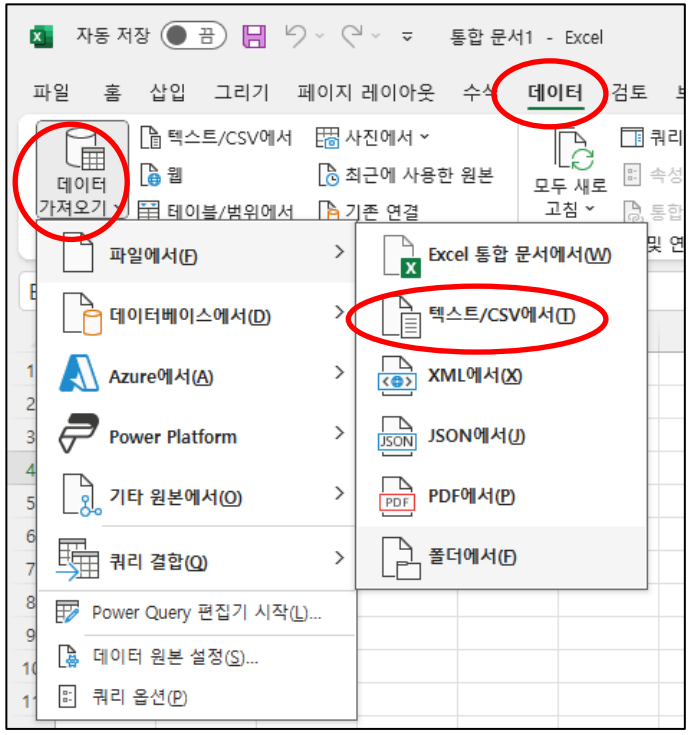


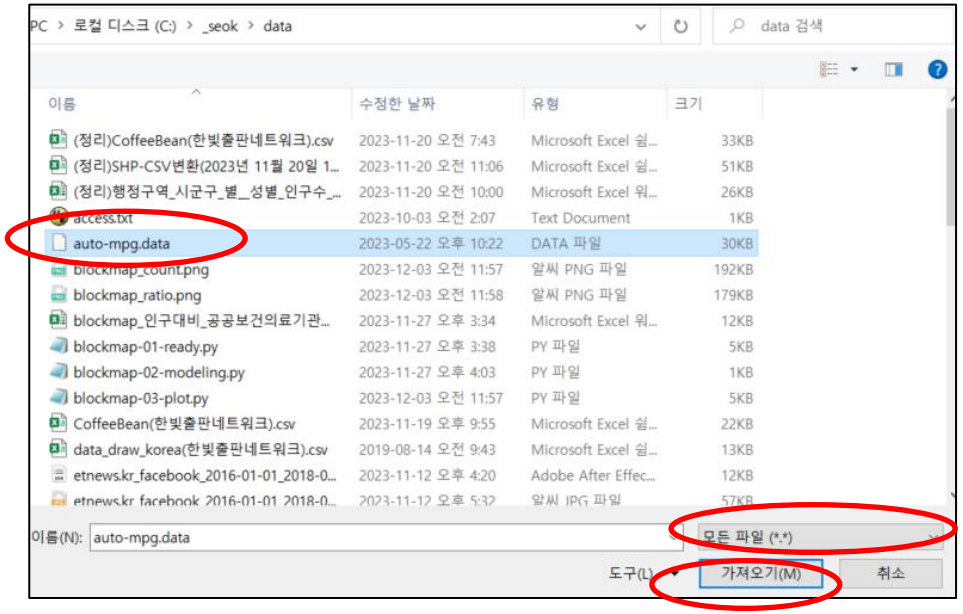
목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → 데이터 모델링 → 결과 시각화

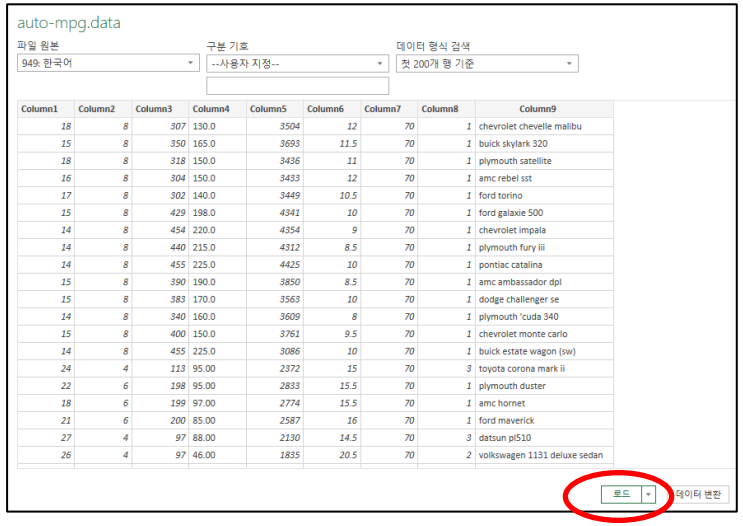
*. 전수업리뷰

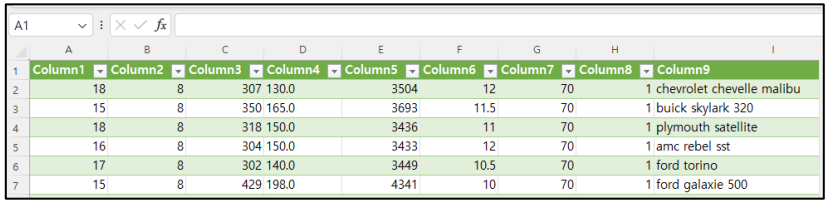
⏳ 데이터 수집

- 데이터 변환









- Excel 을 실행
- 데이터 – 데이터가져오기 – 파일에서 – 텍스트/csv에서

*. 전수업리뷰

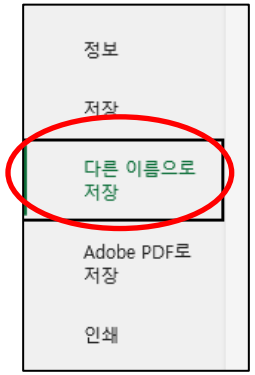
⏳ 데이터 수집

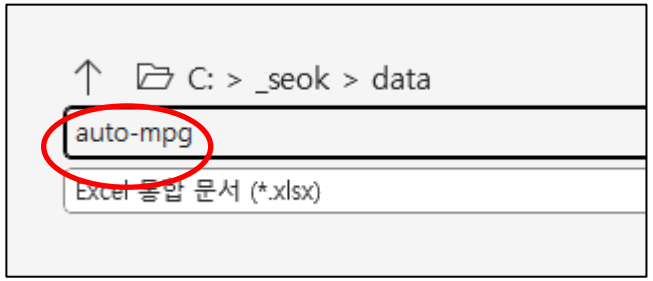
- 데이터 변환 – 컬럼이름 변경

연비 실린더수 배기량 마력 무게 가속도 모델년도 원본 차이름

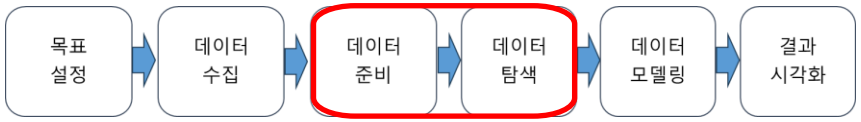
	A	B	C	D	E	F	G	H	I
1	mpg	cylinder	displ	hp	weight	accel	modely	origin	carname
2	18	8	307	130.0	3504	12	70	1	chevrolet chevelle malibu
3	15	8	350	165.0	3693	11.5	70	1	buick skylark 320
4	18	8	318	150.0	3436	11	70	1	plymouth satellite
5	16	8	304	150.0	3433	12	70	1	amc rebel sst
6	17	8	302	140.0	3449	10.5	70	1	ford torino

- mpg, cylinder, displ, hp, weight, accel, modely, origin, carname





c:_seok\data\auto-mpg.xlsx 로 저장



*. 전수업리뷰

⏳ 데이터 준비 및 탐색

- 파일을 읽어서 내용 확인 및 컬럼 삭제

4칸

```
#=====================
# 회귀분석- 자동차연비 예측
#=====================
import numpy as np
import pandas as pd

df = pd.read_excel('./auto-mpg.xlsx', index_col = 0)

print('데이터셋 크기: ', df.shape)
print(df.head())
print('01', '=' * 40)

# 사용하지 않는 컬럼 삭제
df = df.drop(['carname', 'origin', 'hp'], axis = 1, inplace = False)
print(df.head())
print('02', '=' * 40)
```

_seok\data\auto-mpg-01-ready.py 로 저장

```
명령 프롬프트
None
03 ========================================

C:₩_seok₩data>python mpg-01-ready.py
데이터셋 크기:  (398, 9)
    mpg  cylinder  displ     hp  weight  accel  modely  origin                   carname
0  18.0         8  307.0  130.0    3504   12.0      70       1  chevrolet chevelle malibu
1  15.0         8  350.0  165.0    3693   11.5      70       1          buick skylark 320
2  18.0         8  318.0  150.0    3436   11.0      70       1         plymouth satellite
3  16.0         8  304.0  150.0    3433   12.0      70       1              amc rebel sst
4  17.0         8  302.0  140.0    3449   10.5      70       1                ford torino
01 ========================================
    mpg  cylinder  displ  weight  accel  modely
0  18.0         8  307.0    3504   12.0      70
1  15.0         8  350.0    3693   11.5      70
2  18.0         8  318.0    3436   11.0      70
3  16.0         8  304.0    3433   12.0      70
4  17.0         8  302.0    3449   10.5      70
02 ========================================
```

cd _seok\data

python auto-mpg-01-ready.py

목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → 데이터 모델링 → 결과 시각화

*. 전수업리뷰

⌛ 데이터 준비 및 탐색

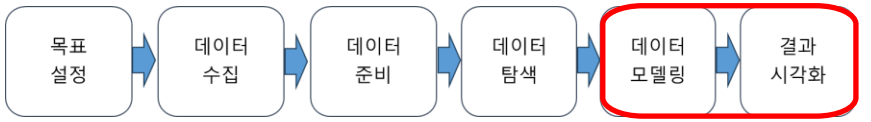
- 정보 확인 및 저장

(계속) 4칸

```
# 데이터셋의 정보 확인
print(df.info())
print('03', '=' * 40)

# 저장
df.to_excel('./auto-mpg-result.xlsx')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   mpg       398 non-null    float64
 1   cylinder  398 non-null    int64
 2   displ     398 non-null    float64
 3   weight    398 non-null    int64
 4   accel     398 non-null    float64
 5   modely    398 non-null    int64
dtypes: float64(3), int64(3)
memory usage: 18.8 KB
None
03 ========================================

C:₩_seok₩data>
```

```
명령 프롬프트

2023-11-20  오전 10:00        26,452 (정리)행정구역_시군구_별
2023-10-03  오전 02:07            10 access.txt
2023-12-03  오후 02:41           534 auto-mpg-01-ready.py
2023-12-03  오후 02:40        18,605 auto-mpg-result.xlsx
2023-05-22  오후 10:22        30,288 auto-mpg.data
2023-12-03  오후 02:17        39,678 auto-mpg.xlsx
```



*. 전수업리뷰

⌛ 데이터모델링 및 시각화

- 선형회귀분석 모델 구축

4칸

```
#=====================
# python –m pip install scikit-learn
# 회귀분석- 자동차연비 예측
# 사이킷런 패키지 활용
#=====================
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
```

_seok\data\auto-mpg-02-sklearn.py 로 저장

사이킷런

- ✓ 파이썬으로 머신러닝을 수행하기 위한 쉽고 효율적인 개발 라이브러리
- ✓ 보스톤 주택 가격 데이터 등과 같은 머신러닝 분석용 데이터셋 을 기본적으로 제공
- ✓ 전체 n개의 컬럼 중 앞에서 (n-1)개의 컬럼은 독립 변수 X를 의미
- ✓ 마지막 컬럼은 종속 변수 Y

목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → 데이터 모델링 → 결과 시각화

*. 전수업리뷰

⌛ 데이터모델링 및 시각화

- 선형회귀분석 모델 구축

(계속) 4칸

```
df = pd.read_excel('./auto-mpg-result.xlsx', index_col=0)

print(df)
print('01', '=' * 40)

# X, Y 분할하기
Y = df['mpg']
X = df.drop(['mpg'], axis = 1, inplace = False)

print(Y)
print('02', '=' * 40)

print(X)
print('03', '=' * 40)
```

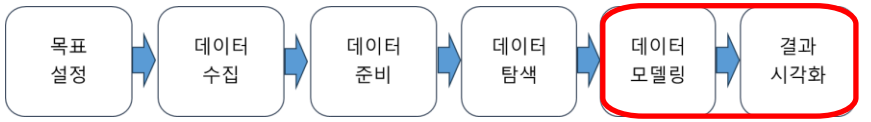
```
C:₩_seok₩data>python auto-mpg-02-sklearn.py
      mpg  cylinder  displ  weight  accel  modely
0    18.0         8  307.0    3504   12.0      70
1    15.0         8  350.0    3693   11.5      70
2    18.0         8  318.0    3436   11.0      70
3    16.0         8  304.0    3433   12.0      70
4    17.0         8  302.0    3449   10.5      70
..    ...       ...    ...     ...    ...     ...
393  27.0         4  140.0    2790   15.6      82
394  44.0         4   97.0    2130   24.6      82
395  32.0         4  135.0    2295   11.6      82
396  28.0         4  120.0    2625   18.6      82
397  31.0         4  119.0    2720   19.4      82

[398 rows x 6 columns]
01 ========================================
```

```
0      18.0
1      15.0
2      18.0
3      16.0
4      17.0
      ...
393    27.0
394    44.0
395    32.0
396    28.0
397    31.0
Name: mpg, Length: 398, dtype: float64
02 ========================================
```

Y

```
     cylinder  displ  weight  accel  modely
0           8  307.0    3504   12.0      70
1           8  350.0    3693   11.5      70
2           8  318.0    3436   11.0      70
3           8  304.0    3433   12.0      70
4           8  302.0    3449   10.5      70
..        ...    ...     ...    ...     ...
393         4  140.0    2790   15.6      82
394         4   97.0    2130   24.6      82
395         4  135.0    2295   11.6      82
396         4  120.0    2625   18.6      82
397         4  119.0    2720   19.4      82

[398 rows x 5 columns]
03 ========================================
```

X



*. 전수업리뷰

⌛ 데이터모델링 및 시각화

- 선형회귀분석 모델 구축

(계속) 4칸

```
# 훈련용 데이터와 평가용 데이터 분할하기
# 데이터를 7:3 비율로 분할
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state = 0)

print('X_train = ', X_train)
print('04', '=' * 40)

# 선형 회귀 분석 : 모델 생성
reg = LinearRegression()

# 선형 회귀 분석 : 모델 훈련
reg.fit(X_train, Y_train)
```

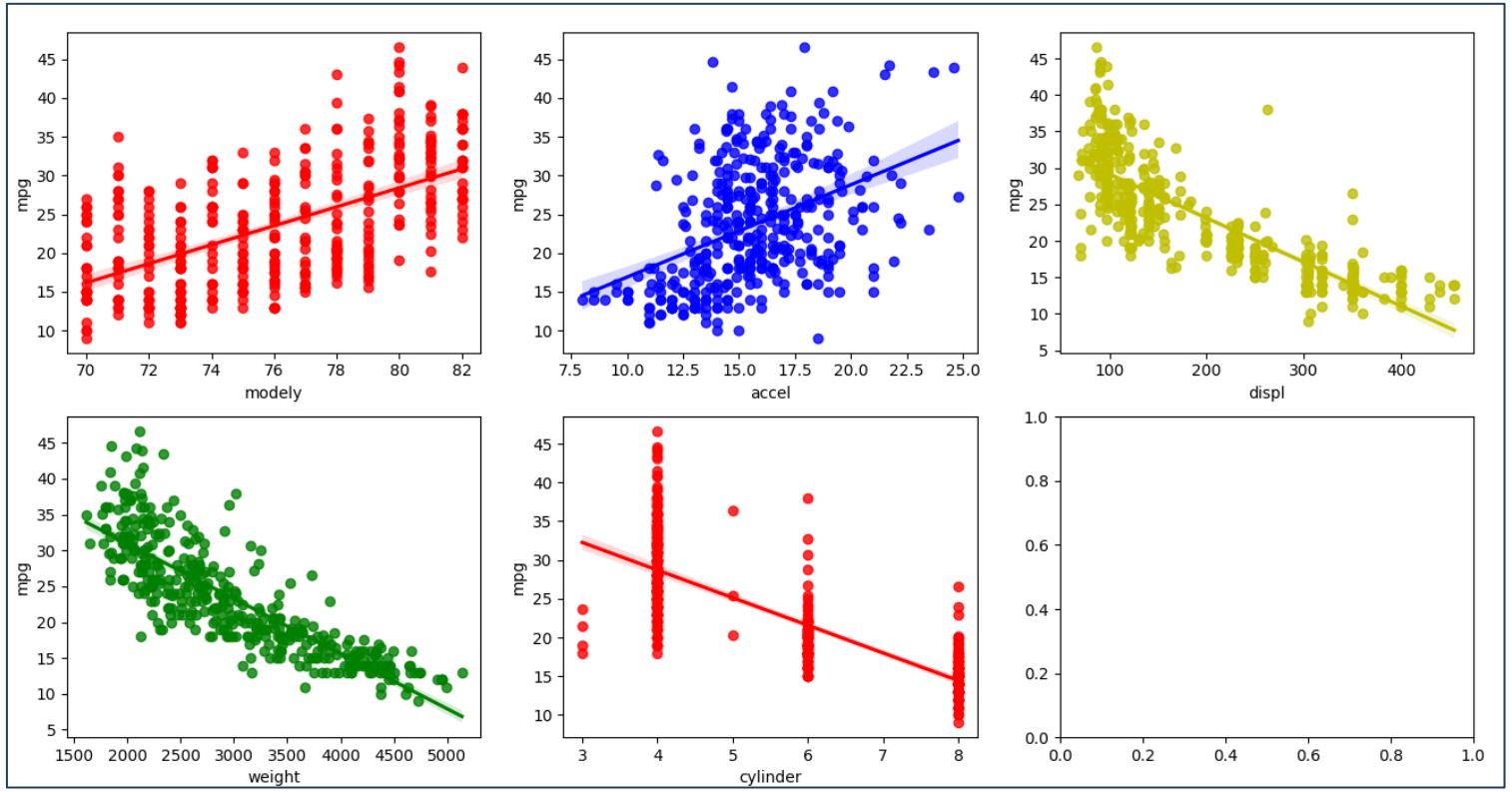
```
X_train =       cylinder  displ  weight  accel  modely
230                8  350.0    4165   11.4      77
357                4  119.0    2615   14.8      81
140                8  304.0    4257   15.5      74
22                 4  104.0    2375   17.5      70
250                8  318.0    3735   13.2      78
..               ...    ...     ...    ...     ...
323                4  156.0    2800   14.4      80
192                6  250.0    3353   14.5      76
117                4   68.0    1867   19.5      73
47                 6  250.0    3282   15.0      71
172                4   90.0    2223   16.5      75

[278 rows x 5 columns]
04 ========================================
```

목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → **데이터 모델링** → 결과 시각화

*. 전수업리뷰

⌛ 데이터모델링 및 시각화

- 시각화



✓ 결론적으로는,

✓ 모델연식이 클수록(최신모델)

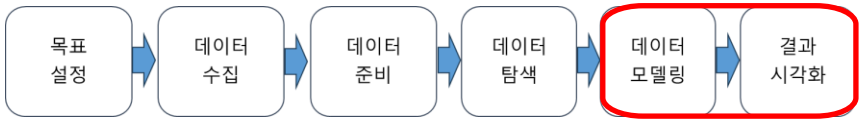
✓ 가속이 높을 수록

✓ 연비가 높다

✓ 배기량, 무게, 실린더수가 낮을 수록

✓ 연비가 높다

*. 전수업리뷰



⌛ 데이터모델링 및 시각화

- 예측

cd _seok\data

python auto-mpg-03-predict.py

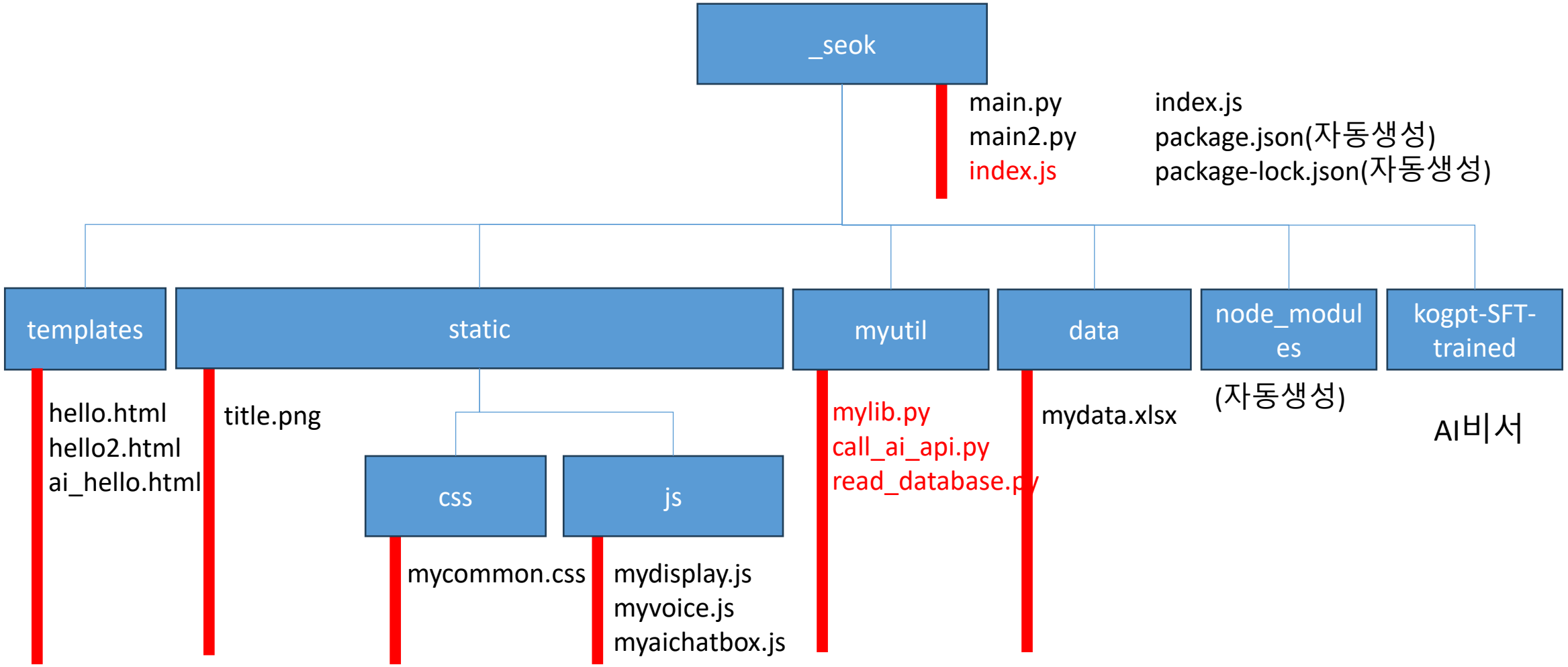
(계속) 4칸

```
# 질문과 답
print("연비를 예측하고 싶은 차의 정보를 입력해주세
cylinders_1 = int(input("cylinders(8) : "))
displacement_1 = int(input("displacement(350) : "))
weight_1 = int(input("weight(3200) : "))
acceleration_1 = int(input("acceleration(22) : "))
model_year_1 = int(input("model_year(99) : "))

mpg_predict = reg.predict([[cylinders_1, displacement_

print("이 자동차의 예상 연비(MPG)는 %.2f입니다." %
```

```
C:₩_seok₩data>python auto-mpg-03-predict.py
연비를 예측하고 싶은 차의 정보를 입력해주세요.
cylinders(8) : 8
displacement(350) : 350
weight(3200) : 3200
acceleration(22) : 22
model_year(99) : 99
C:₩Application₩Python310₩lib₩site-packages₩sklearn₩base.py
nearRegression was fitted with feature names
  warnings.warn(
이 자동차의 예상 연비(MPG)는 41.32입니다.

C:₩_seok₩data>
```

*. 전수업리뷰

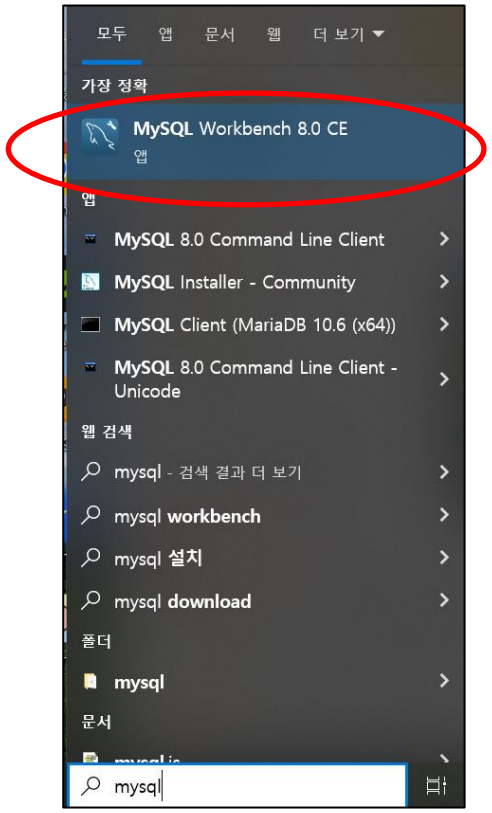
⏳ 디렉토리 구성 : AI 비서 – 대화내용로깅



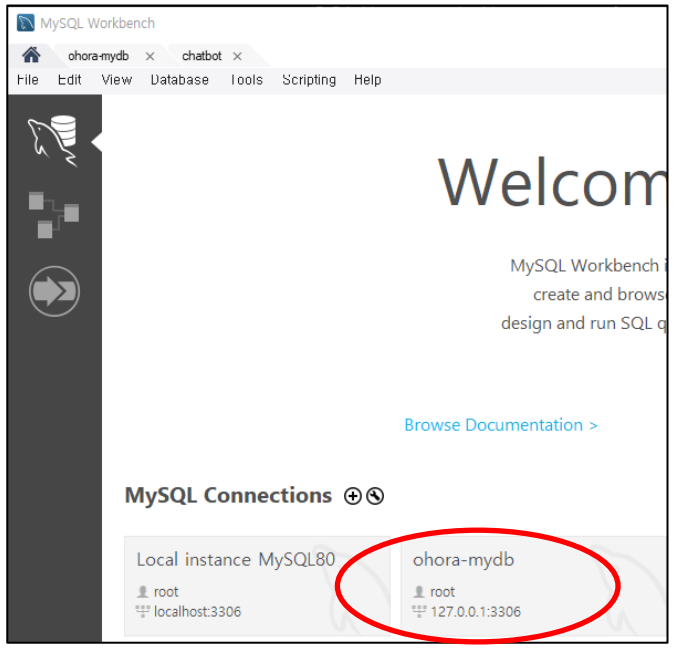
*. 전수업리뷰

※ MySQL 이 설치가 안되어 있으면,
[기타 프로그램 설치 참조]

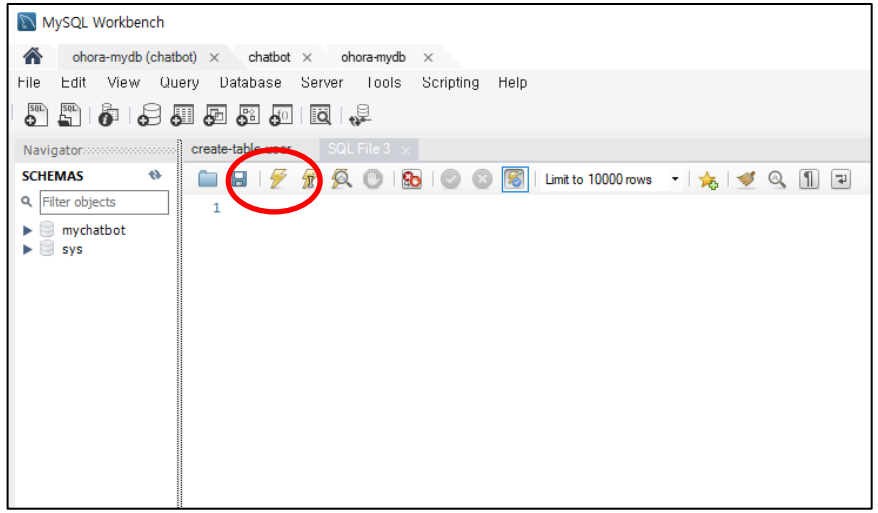
⌛ MySQL 접속 – Workbench 실행



- 윈도우 검색창에서 mysql 입력



- root로 로그인



- SQL 명령어 실행창

*. 전수업리뷰

⌛ SQL 명령 실행

4칸

```
create user 'aiya'@'%' identified by '10041004';

GRANT ALL PRIVILEGES ON *.* to 'aiya'@'%' with grant option;
```

- 사용자 생성과 권한 부여

*. 전수업리뷰

⌛ SQL 명령 실행

4칸

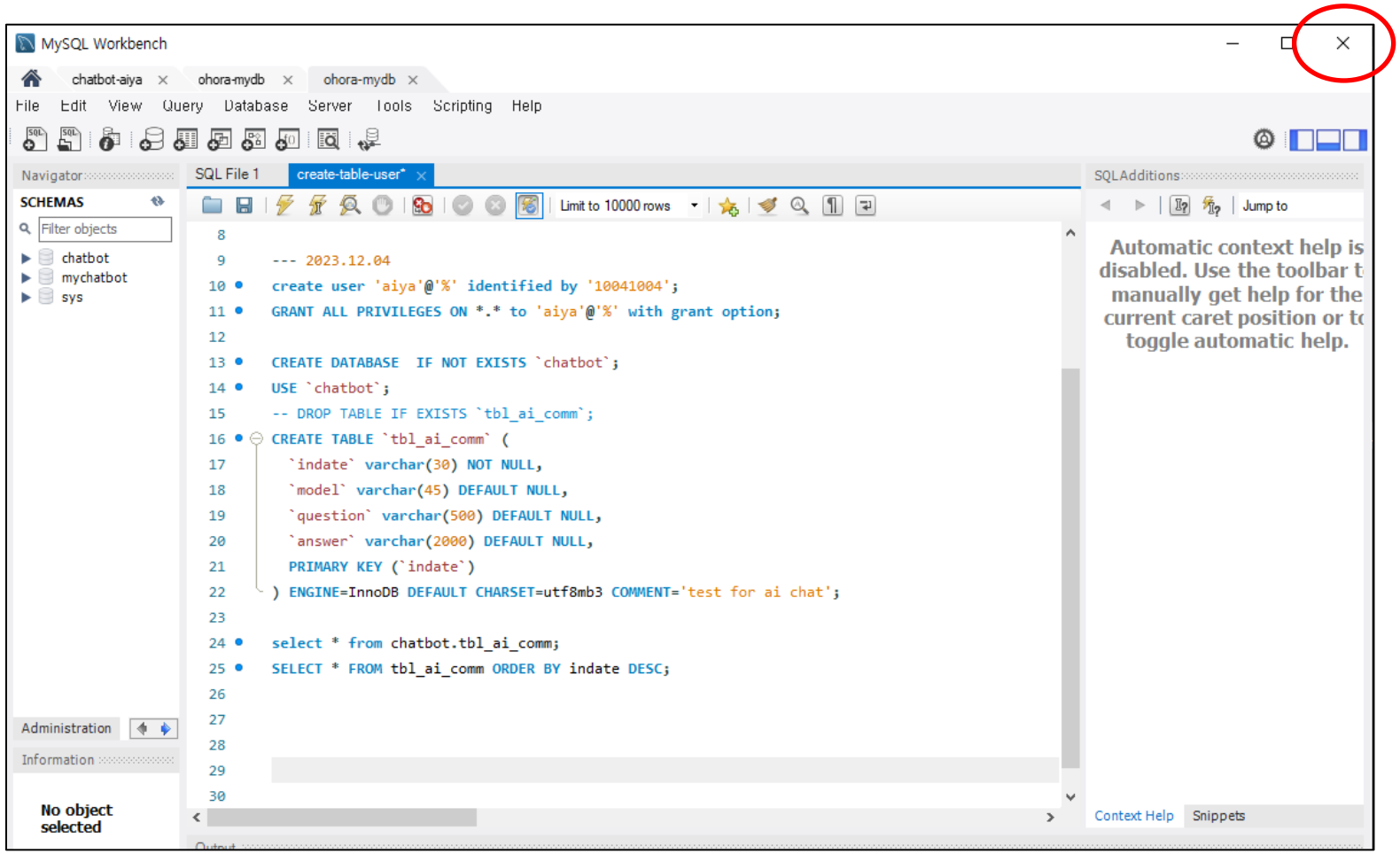
```
CREATE DATABASE  IF NOT EXISTS `chatbot`;

USE `chatbot`;

-- DROP TABLE IF EXISTS `tbl_ai_comm`;
CREATE TABLE `tbl_ai_comm` (
    `indate` varchar(30) NOT NULL,
    `model` varchar(45) DEFAULT NULL,
    `question` varchar(500) DEFAULT NULL,
    `answer` varchar(2000) DEFAULT NULL,
    PRIMARY KEY (`indate`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COMMENT='test for ai chat';


select * from chatbot.tbl_ai_comm;
```

- 스키마 생성

- 테이블 생성

*. 전수업리뷰

⌛ MySQL Workbench 종료



*. 전수업리뷰

⌛ myutil/read_database.py 프로그램 생성

4칸

```
#================================
# Read DataBase MySQL
#================================
import sys, io
import requests
from mylib import mylib_Read_DB

sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding='utf-8')

if __name__ == '__main__':

    sResult = "<html><head>"
    sResult += "<meta charset='utf-8'>"
    sResult += "<meta name='viewport' content='width=device-width, initial-scale=1'>"
    sResult += "</head><body>"
```

- 신규생성

- 메모장이나 vsc 사용

*. 전수업리뷰

⌛ myutil/read_database.py 프로그램 생성

4칸

```
sql =   "SELECT * FROM tbl_ai_comm "
sql += " ORDER BY indate DESC "

sResult += "<table border=1><tbody>"
sResult += mylib_Read_DB(sql)
sResult += "</tbody></table>"
sResult += "</body></html>"

print(sResult)
```

- 신규생성

- 메모장이나 vsc 사용

*. 전수업리뷰

⌛ myutil/mylib.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸 16칸

```
    for index, row in df.iterrows():
        sResult += "<tr role='row'>"
        for _f in fields:
            if str(row[_f]) == '':
                sResult += "<td role='cell'>-</td>"
            else:
                sResult += "<td role='cell'>" + str(row[_f]) + "</td>"

        sResult += "</tr>"
    sResult += "</tbody></table>"
    sResult += "</body></html>"

    return(sResult)

#============================
# 공통 라이브러리
# MySQL DB 에 저장
#============================
```

*. 전수업리뷰

⌛ myutil/mylib.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸 16칸

```
import pymysql
def mylib_Write_DB(_datetime, _model, _question, _answer):
    conn = pymysql.connect(host="localhost",
                           port=int("3306"),
                           user="aiya",
                           password="10041004",
                           db="chatbot",
                           charset="utf8")
    sql =   "INSERT INTO tbl_ai_comm (indate, model, question, answer) "
    sql += " VALUES (%s, %s, %s, %s)"

    try:
        with conn:
            with conn.cursor() as cur:
                cur.execute(sql, (_datetime, _model, _question, _answer))
                conn.commit()
    except Exception as ee:
        print('DB write error sql=', sql, '  || ee=', ee)
```

*. 전수업리뷰

⌛ myutil/mylib.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸 16칸

```
#============================
# 공통 라이브러리
# MySQL DB 에서 데이터 읽기
#============================
import pymysql

def mylib_Read_DB(_sql):
    sResult = "<thead><tr><th>시간</th><th>모델</th>"
    sResult += "<th>질문</th><th>응답</th></tr></thead>"

    conn = pymysql.connect(host="localhost",
                           port=int("3306"),
                           user="aiya",
                           password="10041004",
                           db="chatbot",
                           charset="utf8")
```

*. 전수업리뷰

⌛ myutil/mylib.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸 16칸

```
    try:
        with conn:
            with conn.cursor() as cur:
                cur.execute(_sql)
                result = cur.fetchall()
                for data in result:
                    sResult += "<tr>"
                    sResult += "<td>" + str(data[0]) + "</td>"
                    sResult += "<td>" + str(data[1]) + "</td>"
                    sResult += "<td>" + str(data[2]) + "</td>"
                    sResult += "<td>" + str(data[3]) + "</td>"
                    sResult += "</tr>"

    except Exception as ee:
        print('DB read error sql=', _sql, '  || ee=', ee)

    return sResult
```

*. 전수업리뷰

⌛ myutil/call_ai_api.py 프로그램 수정

- 붉은 색만 추가

4칸

```
#================================
#AI API
#================================
import sys, io
import requests
import json
import datetime as dt
from mylib import mylib_Write_DB

sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding='utf-8')

ai_band = "myai"
ai_model = "aiya"
AI_URL = "http://172.16.11.220:9999/api/get_data"
SECRET_KEY = "AAAAAAAAAAAABBBCCC111"
#-------------------------------------------------
class ChatbotMessageSender:
    ep_path = AI_URL
```

*. 전수업리뷰

⌛ myutil/call_ai_api.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
#-----------------------------------------------------
if __name__ == '__main__':
    myquery = ''
    quest1 = sys.argv[1]

    myai = ChatbotMessageSender()
    myquery = quest1
    res1 = myai.req_message_send()

    json_dict = json.loads(res1.text)

    sResult = json_dict['answer']

    itstime = dt.datetime.now()
    THISTIME = itstime.strftime("%Y%m%d%H:%M:%S:") + str(itstime.microsecond)[:3]
```

*. 전수업리뷰

⌛ myutil/call_ai_api.py 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
        try:
            mylib_Write_DB(THISTIME, ai_model, quest1, sResult)
        except Exception as ee:
            print('DB write error : ', ee, '\n')

        print(sResult)
```

*. 전수업리뷰

⌛ index.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
//================================
app.get('/read', function (req, res) {
    var htmlStr = '';

    const { spawn } = require("child_process");
    const Python = spawn("python", ["myutil/read_database.py"]);

    Python.stdout.on("data", (data) => {
      htmlStr += data;
    });

    Python.stderr.on("data", (data) => {
      htmlStr += data;
    });
```

*. 전수업리뷰

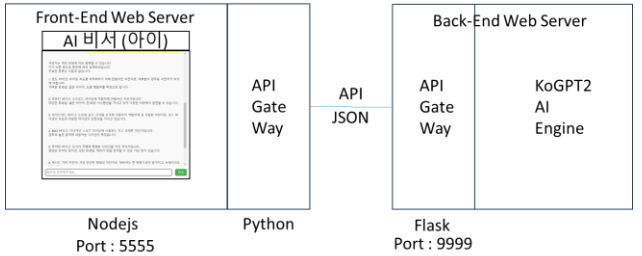
⌛ index.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

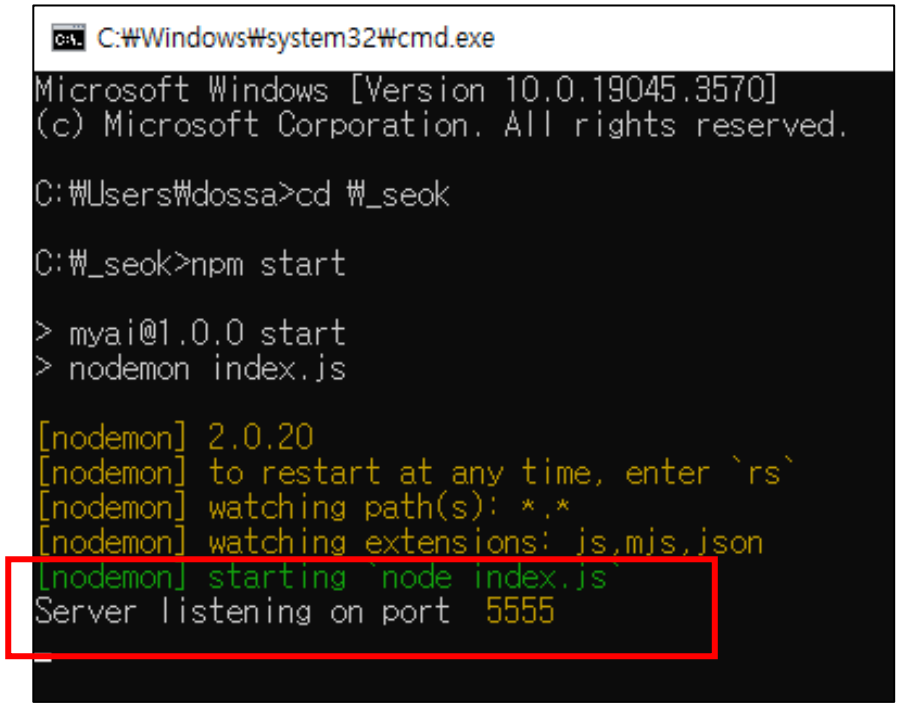
```
    Python.on("close", (code) => {
        res.writeHead(200, { 'Content-Type': 'text/html;charset=UTF-8' });

        res.write(htmlStr);
        res.end();
    });
});

//================================
const port = 5555;
app.listen(port, ()=>{
    console.log('Server listening on port ', port);
});
```

*. 전수업리뷰



⌛ 실행

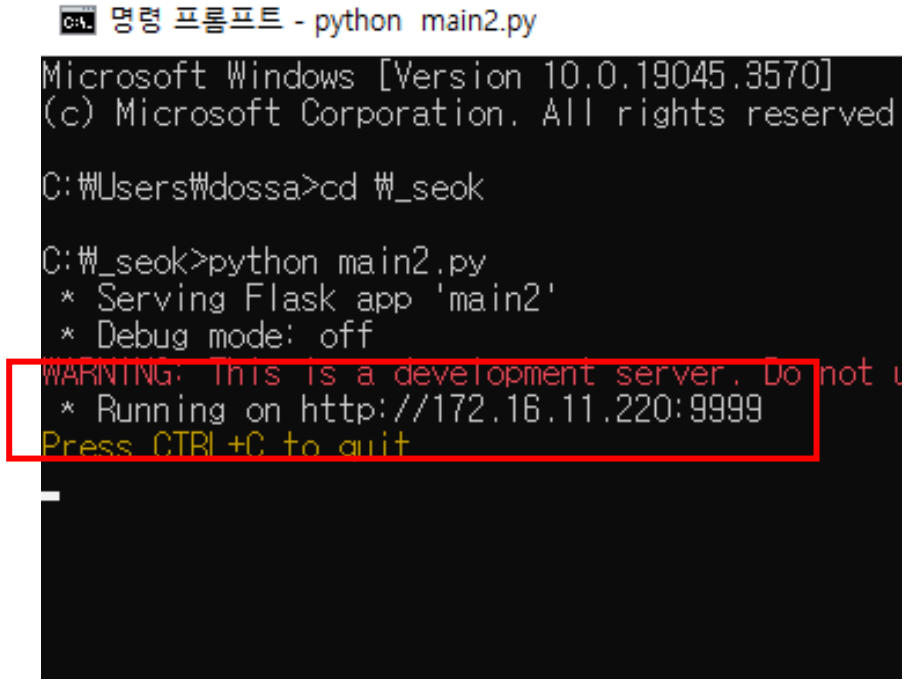
- nodejs 실행



cd _seok

npm start

- Flask 실행

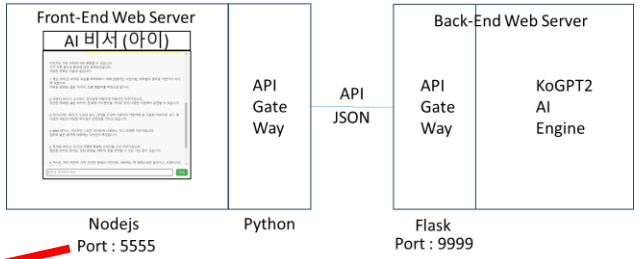


cd _seok

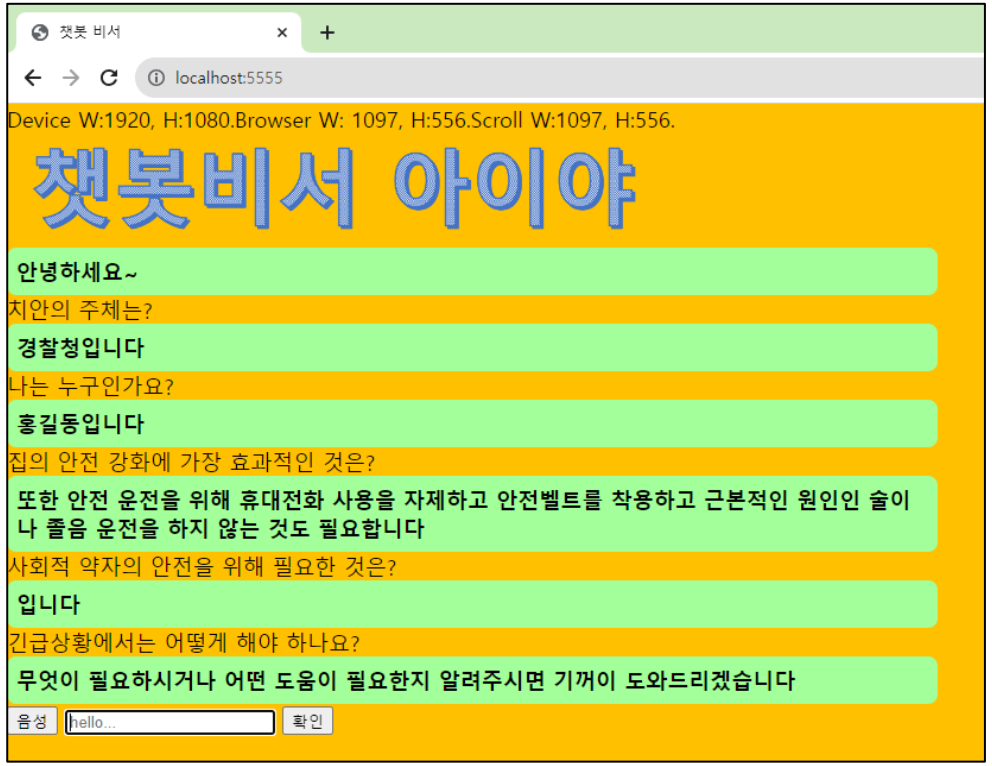
python main2.py

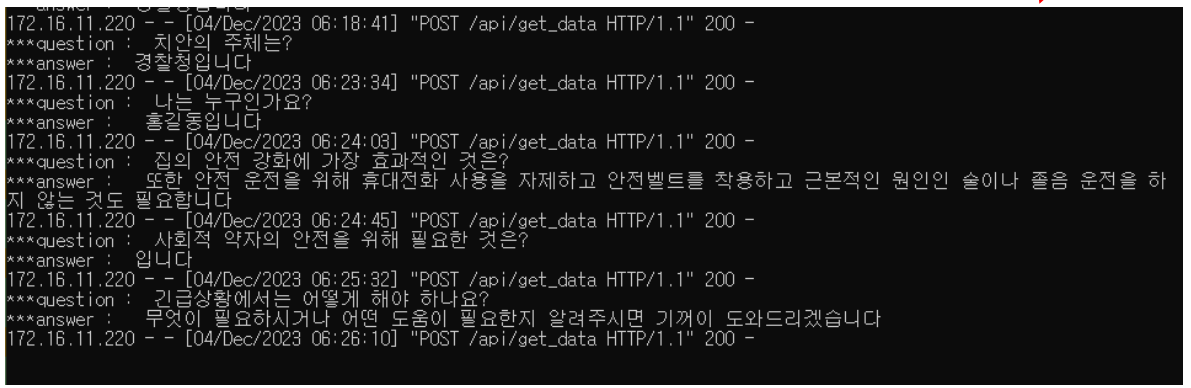
*. 전수업리뷰

⌛ 접속

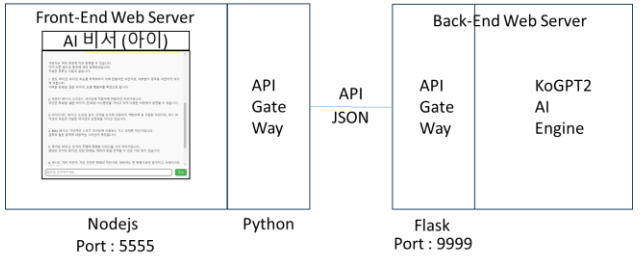


http://localhost:5555/





*. 전수업리뷰



⌛ DB 확인

http://localhost:5555/read

localhost:5555/read

localhost:5555/read

시간	모델	질문	응답
2023120406:26:10:699	aiya	긴급상황에서는 어떻게 해야 하나요?	무엇이 필요하시거나 어떤 도움이 필요한지 알려주시면 기꺼이 도와드리겠습니다
2023120406:25:32:858	aiya	사회적 약자의 안전을 위해 필요한 것은?	입니다
2023120406:24:45:773	aiya	집의 안전 강화에 가장 효과적인 것은?	또한 안전 운전을 위해 휴대전화 사용을 자제하고 안전벨트를 착용하고 근본적인 원인인 술이나 졸음 운전을 하지 않는 것도 필요합니다
2023120406:24:03:176	aiya	나는 누구인가요?	홍길동입니다
2023120406:23:34:433	aiya	치안의 주체는?	경찰청입니다

*. 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

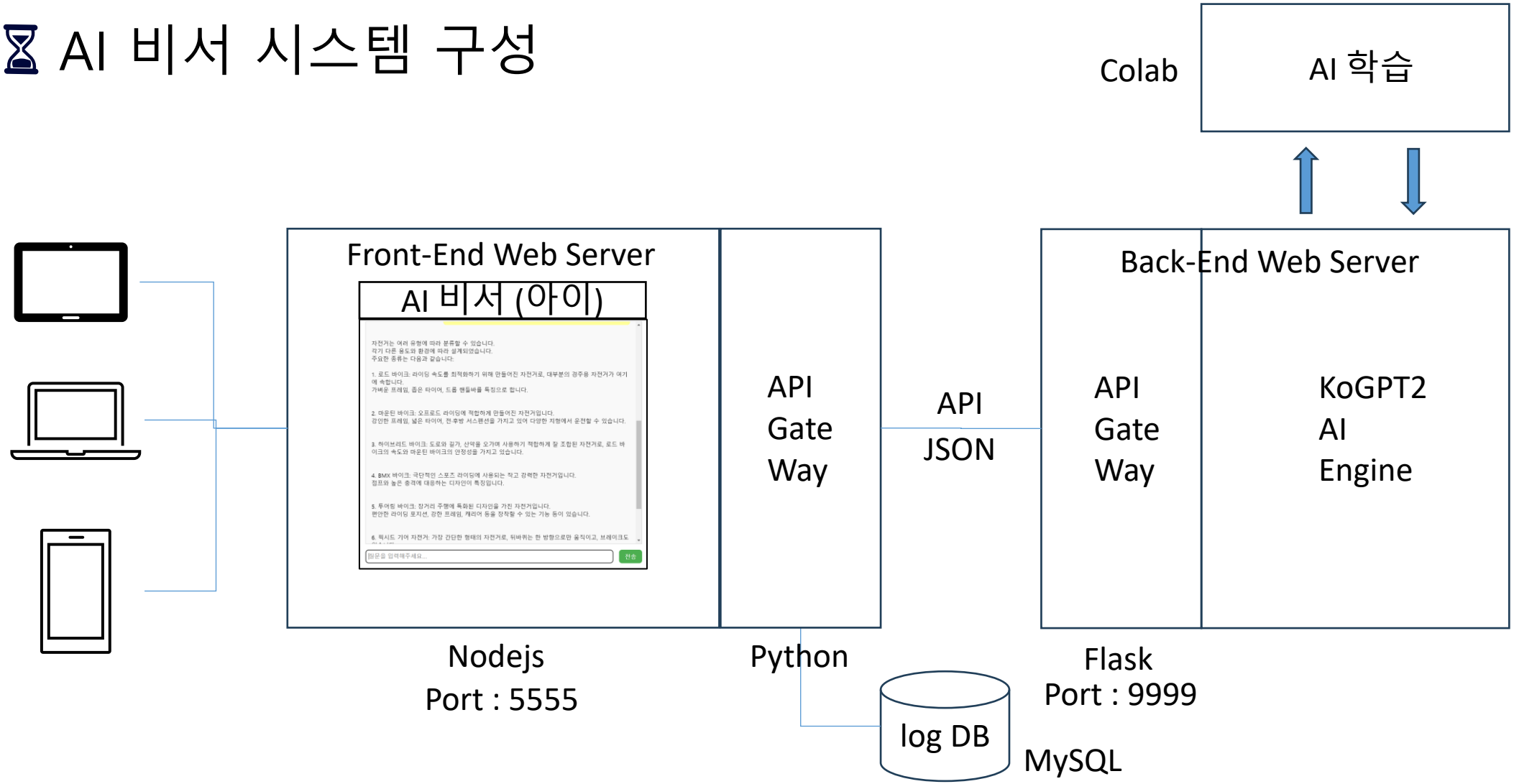
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

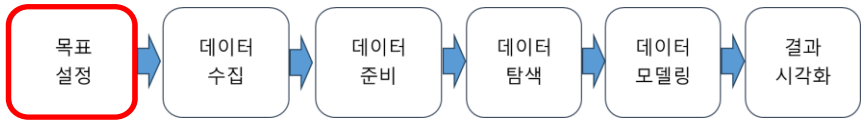
AI 비서학습

- ✓ 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. 텍스트마이닝 - 감성분석

⏳ 목표설정

- 목표

  - 감성 분류 모델을 이용하여 네이버 뉴스에서 크롤링한 <코로나> 관련 텍스트에 대해 감성을 분석

- 핵심 개념 이해

  - 텍스트에서 사용자의 주관적인 의견이나 감성, 태도를 분석하는 텍스트 마이닝 분석 기법
  - 텍스트에서 감성을 나타내는 단어를 기반으로 긍정 또는 부정의 감성을 결정
  - 감성 단어에 대한 사전을 가진 상태에서 단어를 검색하여 점수를 계산하며, 최근에는 머신러닝 기반의 감성 분석이 늘어나고 있음

1. 텍스트마이닝

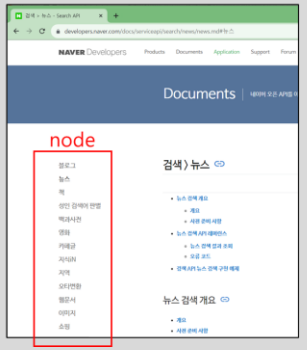
⏳ 데이터 수집

- 네이버 API 크롤링
  - 네이버 뉴스에서
  - 코로나로 검색

2. 네이버 API

⏳ 요청과 응답

https://developers.naver.com/docs/serviceapi/search/news/news.md#뉴스



구분	내용 및 설명	
URL	뉴스	https://openapi.naver.com/v1/search/**news**.json
	블로그	https://openapi.naver.com/v1/search/**blog**.json
	카페	https://openapi.naver.com/v1/search/**cafearticle**.json
	영화	https://openapi.naver.com/v1/search/**movie**.json
	쇼핑	https://openapi.naver.com/v1/search/**shop**.json

데이터 요청 주소

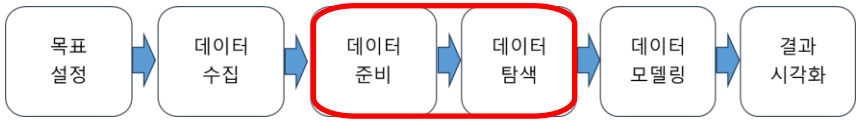
- gitgub 에 있는 파일 활용

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

다운로드 받아서 자신의 data 폴더로 이동

c:₩_seok₩data₩코로나_naver_news(한빛출판네트워크).json
c:₩_seok₩data₩ratings_train(한빛출판네트워크).txt

1. 텍스트마이닝



⏳ 데이터 준비 및 탐색

- 선언부

4칸

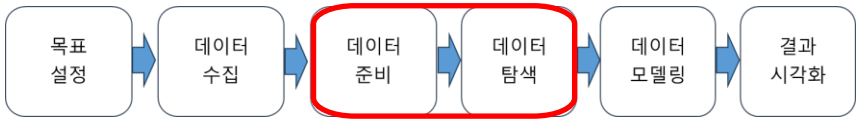
```
#-------------------------------------------------------
def okt_tokenizer(text):
    tokens = okt.morphs(text)
    return tokens
#======================
# python -m pip install sikit-learn
# 텍스트마이닝 - 감성분석
#======================
import json
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from konlpy.tag import Okt
import re
```

_seok\data\textm-01-ready.py 로 저장

✓ KoNLPy에서 사용 가능한 품사 태깅 패키지 Okt 활용 : 명사 분리

✓ 사이킷런 활용 : 파이썬으로 머신러닝을 수행하기 위한 쉽고 효율적인 개발 라이브러리

```
cd \_seok\data

python textm-01-ready.py
```



1. 텍스트마이닝

⏳ 데이터 준비 및 탐색

- 선언부

(계속) 4칸

```
okt = Okt()

#warning 메시지 표시 안함
import warnings
warnings.filterwarnings(action = 'ignore')
```

✓ 불필요한 Warning Error 제거

1. 텍스트마이닝

⏳ 데이터 준비 및 탐색

- 파일을 읽어서 내용 확인

(계속) 4칸

```
#-----------------------------------------------------------
train_df = pd.read_csv('./ratings_train(한빛출판네트워크).txt',
    encoding = 'utf8', sep = '\t')

# 결측치 제거 : null 제거
train_df = train_df[train_df['document'].notnull()]

print(train_df['label'].value_counts())
print('01', '=' * 40)

# 한글이 아닌 문자 제거
train_df['document'] = train_df['document'].apply(lambda x : re.sub(r'[^ ㄱ-ㅣ 가-힣]+', " ", x))

print(train_df.head())
print('02', '=' * 40)
```

```
C:₩_seok₩data>python textm-01-ready.py
label
0    75170
1    74825
Name: count, dtype: int64
01 ========================================
         id                                           document  label
0   9976970                                아 더빙  진짜 짜증나네요 목소리      0
1   3819312                       흠 포스터보고 초딩영화줄 오버연기조차 가볍지 않구나      1
2  10265843                                    너무재밓었다그래서보는것을추천한다      0
3   9045019                        교도소 이야기구먼  솔직히 재미는 없다 평점 조정      0
4   6483659  사이몬페그의 익살스런 연기가 돋보였던 영화 스파이더맨에서 늙어보이기만 했던 커스틴
02 ========================================
```

1. 텍스트마이닝

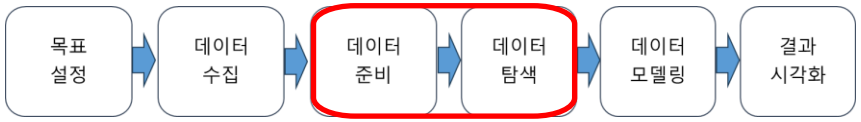
⏳ 데이터 준비 및 탐색

- 파일을 읽어서 내용 확인

(계속) 4칸

```
#-----------------------------------------------------------
file_name = './코로나_naver_news(한빛출판네트워크).json'
with open(file_name, encoding = 'utf8') as _file:
    data = json.load(_file)

print(data)
print('03', '=' * 40)
```

[{'cnt': 1, 'description' '코로나발 경제 위기 대응을 위해 돈 쓸 곳은 늘어났지만, 국세 수입은 줄어들면서 정부의 재정 마련에 대한 우려가 컸다. 이 때문에 한국개발원(KDI) 등 국책연구기관들은 증세를 화두로 꺼내들었지만, 정부 여당은 증세에... ', 'pDate': '2020-06-04 14:12:00', 'title': "결국 '증세론' 먼저 꺼내든 與..."증세없는 '기본소득' 불가능""}, {'cnt': 2, 'description': '▲ 지난 2일 창녕군보건소 앞에 설치한 선별진료소에서 검사자가 체온을 측정하고 있다.©(사진제공=창녕군청)



1. 텍스트마이닝

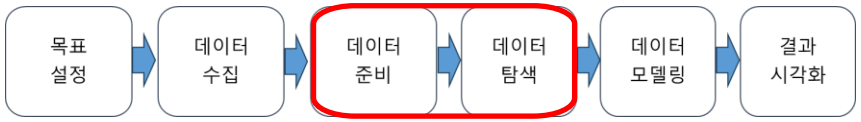
⏳ 데이터 준비 및 탐색

- 데이터프레임으로 구성

(계속) 4칸

```
data_title = []
data_description = []

for item in data:
    data_title.append(item['title'])
    data_description.append(item['description'])

print(data_title)
print('04', '=' * 40)

print(data_description)
print('05', '=' * 40)

data_df = pd.DataFrame({'title':data_title, 'description':data_description})

print(data_df)
print('06', '=' * 40)
```

```
들이...', "제56회 대종상영화제 참석한 문가영제56회 대종상영
종'이 최우수 작품상을 거머쥤다. 남녀주연상은... "]
05 ========================================
                                                  title
0          결국 '증세론' 먼저 꺼내든 與…&amp;quot;증세없는 '기본
1                                  창녕군, '창녕형' 비대면 선
2                              DK모바일, 메인 홍보 모델로 AOA '
3           김병민 &amp;quot;기본소득도 필요하면 논의 테이블에 올
4                                  이재갑 장관, 고용안정지원금
...                                                     ...
795  [세계의 눈] &amp;quot;<b>코로나</b>19 영향, 현대차 매출 큰
796                            중부발전, 소규모 태양광사업자 REC 판
797  &amp;quot;<b>코로나</b> 함께 극복&amp;quot;…박종환 자유총연맹
798                  치과의사 7000명 모이는 행사 앞두고 치협 회
799                              [IS포토]대종상영화제] 문가영, '

[800 rows x 2 columns]
06 ========================================
```



1. 텍스트마이닝

⏳ 데이터 준비 및 탐색

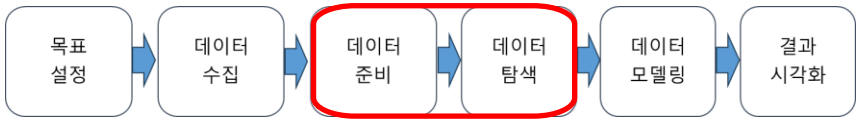
- 데이터의 벡터화

※ TF-IDF 기반의 벡터화 :
- 특정 문서에 많이 나타나는 단어는 해당 문서의 단어 벡터에 가중치를 높임
- 모든 문서에 많이 나타나는 단어는 범용적으로 사용하는 단어로 취급하여 가중치를 낮추는 방식

(계속) 4칸

```
#-----------------------------------------------------
print('잠시만 기다리세요(약 5분)====================')

tfidf = TfidfVectorizer(tokenizer = okt_tokenizer,
    ngram_range = (1, 2), min_df = 3, max_df = 0.9)
tfidf.fit(train_df['document'])
train_tfidf = tfidf.transform(train_df['document'])

print('조금만 더 기다리세요(약 2분)==================')

SA_lr = LogisticRegression(random_state = 0)
SA_lr.fit(train_tfidf, train_df['label'])
LogisticRegression(random_state = 0)
```

- ✓ 사이킷런의 TfidfVectorizer를 이용하여 TF-IDF 벡터화에 사용할 tfidf 객체를 생성
- ✓ 토큰 생성기tokenizer는 okt_tokenizer() 함수로 1~2개 단어로 정의함
- ✓ 토큰은 출현 빈도가 최소min_df 3번 이상이고 최대 max_df 90% 이하인 것만 사용
- ✓ 벡터화할 데이터 train_df['document']에 대해 벡터 모델 tfidf의 내부 설정값을 조정fit()하고 벡터로 변환을 수행transform()



1. 텍스트마이닝

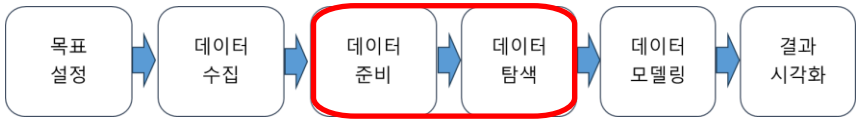
⏳ 데이터 준비 및 탐색

- 데이터의 벡터화

(계속) 4칸

```
params = {'C': [1, 3, 3.5, 4, 4.5, 5]}
SA_lr_grid_cv = GridSearchCV(SA_lr, param_grid = params,
    cv = 3, scoring = 'accuracy', verbose = 1)

SA_lr_grid_cv.fit(train_tfidf, train_df['label'])
print(SA_lr_grid_cv.best_params_, round(SA_lr_grid_cv.best_score_, 4))
print('07', '=' * 40)

#최적 매개변수의 best 모델 저장
SA_lr_best = SA_lr_grid_cv.best_estimator_
```

- ✓ 하이퍼 매개변수 C에 대해 비교 검사를 할 6개 값[1, 3, 3.5, 4, 4.5, 5]을 params로 하고,
- ✓ 교차 검증cv을 3,
- ✓ 모형 비교 기준은 정확도로 설정scoring='accuracy'
- ✓ GridSearchCV 객체를 생성
- ✓ train_tfidf와 label 컬럼에 대해 설정값을 조정fit()
- ✓ GridSearchCV에 의해 찾은 최적의 C 매개변수best_params와 최고 점수best_score를 출력하여 확인



1. 텍스트마이닝

⏳ 데이터 준비 및 탐색

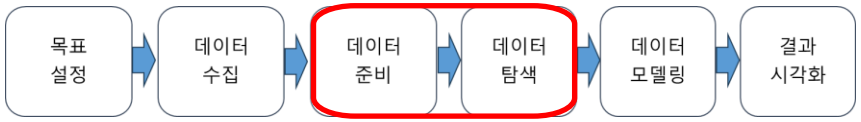
- 감성분석

(계속) 4칸

```
#-----------------------------------------------------
#1) 분석할 데이터의 피처 벡터화 ---<< title >> 분석
data_title_tfidf = tfidf.transform(data_df['title'])

#2) 최적 매개변수 학습 모델에 적용하여 감성 분석
data_title_predict = SA_lr_best.predict(data_title_tfidf)

#3) 감성 분석 결과값을 데이터프레임에 저장
data_df['title_label'] = data_title_predict

print(data_df['title_label'])
print('08', '=' * 40)
```

0 : 부정 감성
1 : 긍정 감성

```
0      0
1      0
2      0
3      0
4      0
      ..
795    0
796    1
797    1
798    0
799    0
Name: title_label, Length: 800, dtype: int64
08 ========================================
```



1. 텍스트마이닝

⏳ 데이터 준비 및 탐색

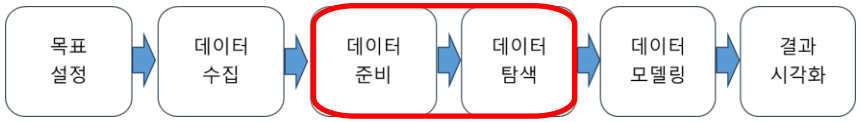
- 감성분석

(계속) 4칸

```
#-----------------------------------------------------
#1) 분석할 데이터의 피처 벡터화 ---<< description >> 분석
data_description_tfidf = tfidf.transform(data_df['description'])

#2) 최적 매개변수 학습 모델에 적용하여 감성 분석
data_description_predict = SA_lr_best.predict(data_description_tfidf)

#3) 감성 분석 결과값을 데이터프레임에 저장
data_df['description_label'] = data_description_predict

print(data_df['description_label'])
print('09', '=' * 40)
```

0 : 부정 감성
1 : 긍정 감성

```
0      0
1      0
2      0
3      1
4      0
      ..
795    0
796    1
797    1
798    1
799    0
Name: description_label, Length: 800, dtype: int64
09 ========================================
```



1. 텍스트마이닝

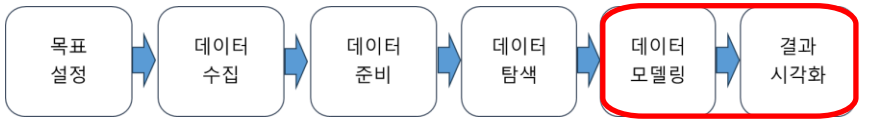
⏳ 데이터 준비 및 탐색

- 데이터 저장

(계속) 4칸

```
#-----------------------------------------------------
data_df.to_csv('./(정리)코로나_naver_news.csv', encoding = 'utf8')

print('10', '=== 파일이 저장 되었습니다.')
```

```
10 === 파일이 저장 되었습니다.
```



1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

- 선언부

4칸

```
#-------------------------------------------------------
def okt_tokenizer(text):
    tokens = okt.morphs(text)
    return tokens
#-------------------------------------------------------
#=====================
# python -m pip install sikit-learn
# 텍스트마이닝 - 감성분석 - 시각화
#=====================
import pandas as pd
from konlpy.tag import Okt
from sklearn.feature_extraction.text import TfidfVectorizer

okt = Okt()

#warning 메시지 표시 안함
import warnings
warnings.filterwarnings(action = 'ignore')
```

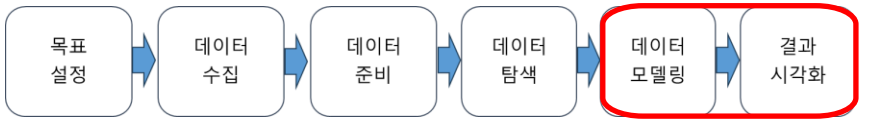
_seok\data\textm-02-plot.py 로 저장

✓ KoNLPy에서 사용 가능한 품사 태깅 패키지
Okt 활용 : 명사 분리

✓ 사이킷런 활용 : 파이썬으로 머신러닝을
수행하기 위한 쉽고 효율적인 개발 라이브러리

cd _seok\data

python textm-02-plot.py



1. 텍스트마이닝

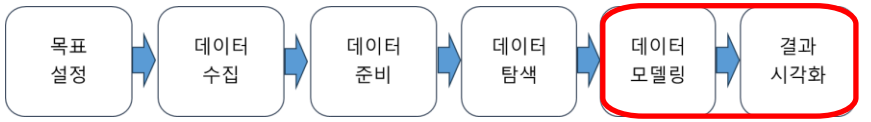
⏳ 데이터모델링 및 시각화

- 파일 읽어서 데이터 확인

(계속) 4칸

```
#-----------------------------------------------------------
data_df = pd.read_csv('./(정리)코로나_naver_news.csv',
    index_col=0, encoding = 'utf8')

print(data_df.head())
print('01', '=' * 40)

print(data_df['title_label'].value_counts())
print(data_df['description_label'].value_counts())
print('02', '=' * 40)
```

```
C:₩_seok₩data>python textm-02-plot.py
                                                  title  ... description_label
0  결국 '증세론' 먼저 꺼내든 與…&amp;quot;증세없는 '기본소득' 불가능&amp;quot;  ...
1                          창녕군, '창녕형' 비대면 선별진료소 운영  ...
2                  DK모바일, 메인 홍보 모델로 AOA '설현' 선정  ...
3   김병민 &amp;quot;기본소득도 필요하면 논의 테이블에 올려야&amp;quot; [인터뷰]  ...
4                        이재갑 장관, 고용안정지원금 서울센터 방문  ...

[5 rows x 4 columns]
01 ========================================
title_label
0    484
1    316
Name: count, dtype: int64
description_label
0    424
1    376
Name: count, dtype: int64
02 ========================================
```



1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

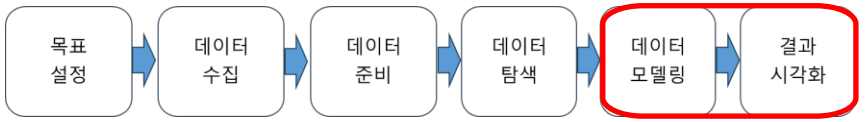
- 긍정, 부정 감성 데이터 분리

(계속) 4칸

```
#----------------------------------------------------------
columns_name = ['title', 'title_label', 'description', 'description_label']
NEG_data_df = pd.DataFrame(columns = columns_name)
POS_data_df = pd.DataFrame(columns = columns_name)
```

✓ NEG_data_df : 부정(Negative)

✓ POS_data_df : 긍정(Positive)



1. 텍스트마이닝

⌛ 데이터모델링 및 시각화

- 긍정, 부정 감성 데이터 분리

(계속) 4칸 8칸 12칸

```
for i, data in data_df.iterrows():
    title = data["title"]
    description = data["description"]
    t_label = data["title_label"]
    d_label = data["description_label"]

    if d_label == 0: #부정 감성 샘플만 추출
        NEG_data_df = NEG_data_df._append(pd.DataFrame([[title, t_label, description,
            d_label]],columns = columns_name), ignore_index = True)
    else : #긍정 감성 샘플만 추출
        POS_data_df = POS_data_df._append(pd.DataFrame([[title, t_label, description,
            d_label]], columns = columns_name), ignore_index = True)
```

✓ Description 에 대해

✓ NEG(부정), POS(긍정) 추출

목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → 데이터 모델링 → 결과 시각화

1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

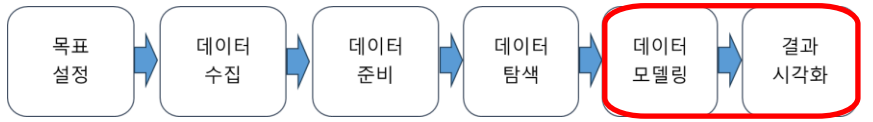
- 긍정, 부정 감성 데이터 분리

(계속) 4칸

```
#-----------------------------------------------------
print(NEG_data_df)
print('03', '=' * 40)

print(POS_data_df)
print('04', '=' * 40)

print(len(NEG_data_df), len(POS_data_df))
print('05', '=' * 40)
```

```
                                                  title  ... description_label
0       결국 '증세론' 먼저 꺼내든 與…&amp;quot;증세없는 '기본소득' 불가능&amp;quot;  ...
1                                  창녕군, '창녕형' 비대면 선별진료소 운영  ...
2                              DK모바일, 메인 홍보 모델로 AOA '설현' 선정  ...
3                                  이재갑 장관, 고용안정지원금 서울센터 방문  ...
4                  '부산행'→'반도'로 이어지는 한국형 좀비 세계관, 칸의 선택 받았다  ...
..                                                  ...  ...  ...
419  BNP파리바 &amp;quot;<b>코로나</b>19 채권, 국제금융시장서 184조4천억원...  ...
420                          [IS포토]대종상영화제] 강기영, '기분좋은 손인사'  ...
421                                  괴산보건소 '직장인 야간운동교실' 재개  ...
422  [세계의 눈] &amp;quot;<b>코로나</b>19 영향, 현대차 매출 큰 타격&amp;quo...  ...
423                              [IS포토]대종상영화제] 문가영, '발칼 듬~뽁'  ...

[424 rows x 4 columns]
03 ========================================
                                                  title  ... description_label
0       김병민 &amp;quot;기본소득도 필요하면 논의 테이블에 올려야&amp;quot; [인터뷰]  ...
1         경기도의회 농정해양위, 농민기본소득 도입 이재명 지사와 집행부에 강력 촉구
2          당신이 아는 미술, 시장이 아는 미술 ④ 한국 현대 미술 시장을 이끄는 작가
3                                BBQ, 상반기 신입 및 경력직 공채 시행  ...
4                                          미소짓는 의료진  ...                1
..                                                  ...  ...  ...
371  원주시 캠프롱 개방행사 'CAMP 2020' 추진... 7일간 공연, 기획전시, 포...  ...
372              여주시의회 박시선 의원, 체류형 관광도시 로드맵과 전략수립 촉구  ...
373                중부발전, 소규모 태양광사업자 REC 판매대금 '선 지급'  ...
374  &amp;quot;<b>코로나</b> 함께 극복&amp;quot;…박종환 자유총연맹 총재, 캠페인 참여  ...
375            치과의사 7000명 모이는 행사 앞두고 치협 회장 "행사 취소해달라"  ...

[376 rows x 4 columns]
04 ========================================
424 376
05 ========================================
```

1. 텍스트마이닝



⏳ 데이터모델링 및 시각화

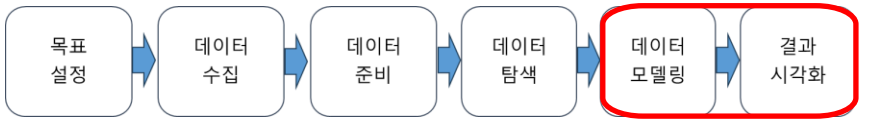
- 형태소 분석을 통해 명사 추출(긍정)

(계속) 4칸

```
#-----------------------------------------------------------
POS_description = POS_data_df['description']
POS_description_noun_tk = []
POS_description_noun_join = []

for d in POS_description:
    POS_description_noun_tk.append(okt.nouns(d))   #명사 형태소만 추출

print(POS_description_noun_tk)   #작업 확인용 출력
print('06', '=' * 40)

for d in POS_description_noun_tk:
    d2 = [w for w in d if len(w) > 1] #길이가 1보다 큰 토큰만 추출
    POS_description_noun_join.append(" ".join(d2))  #토큰 연결하여 리스트 구성

print(POS_description_noun_join) #작업 확인용 출력
print('07', '=' * 40)
```

[['변화', '핵심', '중', '우리', '사회', '신종', '코로나', '바이러스', '감염증', '코로나', '의', '위기', '마주', '언제', '끝', '날', '또', '앞', '미래', '국가', '국민', '어려움', '해결', '저희'], ['한편', '코로나', '로', '해외', '식', '재료', '사재기', '국민', '먹거리', '안정', '생산', '것', '포스트', '코로나', '의', '과제', '부각', '농민', '기본소득', '도입', '통해', '안정', '생산', '기반', '확충', '것'],

['변화 핵심 우리 사회 신종 코로나 바이러스 감염증 코로나 위기 마주 언제 미래 국가 국민 어려움 해결 저희', '한편 코로나 해외 재료 사재기 국민 먹거리 안정 생산 포스트 코로나 과제 부각 농민 기본소득 도입 통해 안정 생산 기반 확충',



1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

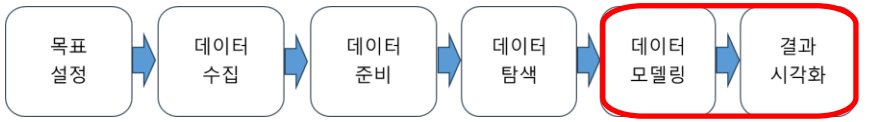
- 형태소 분석을 통해 명사 추출(부정)

(계속) 4칸

```
#-----------------------------------------------------------
NEG_description = NEG_data_df['description']
NEG_description_noun_tk = []
NEG_description_noun_join = []

for d in NEG_description:
    NEG_description_noun_tk.append(okt.nouns(d)) #명사 형태소만 추출

print(NEG_description_noun_tk)   #작업 확인용 출력
print('08', '=' * 40)

for d in NEG_description_noun_tk:
    d2 = [w for w in d if len(w) > 1] #길이가 1보다 큰 토큰만 추출
    NEG_description_noun_join.append(" ".join(d2))  # 토큰 연결하여 리스트 구성

print(NEG_description_noun_join) #작업 확인용 출력
print('09', '=' * 40)
```

[['코로나', '발', '경제', '위기', '대응', '위해', '돈', '곳', '국세', '수입', '정부', '재정', '마련', '대한', '우려', '이', '때문', '한국', '개발', '등', '국책', '연구기관', '증세', '화두', '정부', '여당', '증세'], ['지난', '창녕군', '보건소', '앞', '설치', '선', '진료', '소', '검사', '체온', '측정', '사진',

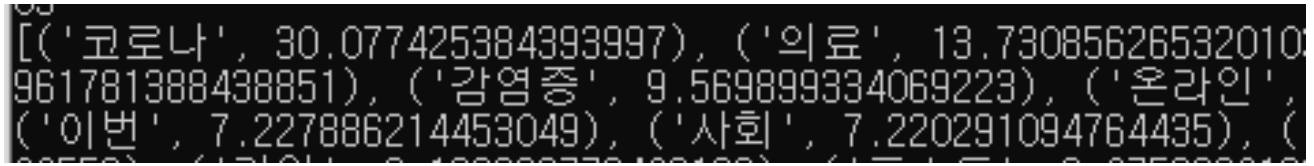
['코로나 경제 위기 대응 위해 국세 수입 정부 재정 마련 대한 우려 때문 한국 개발 국책 연구기관 증세 화두 정부 여당 증세', '지난 창녕군 보건 소 설치 진료 검사 체온 측정 사진 제공 창녕군 코로나 장기 대비 비대 진료 도입 경남 창녕군 지난 도내



1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

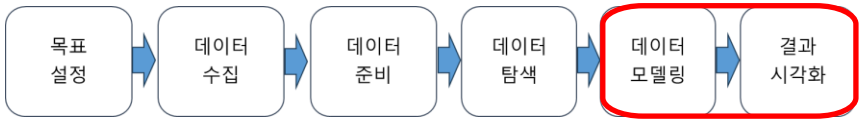
- TFIDF 값 구한 후 정렬(긍정)



(계속) 4칸

```
#-----------------------------------------------------------
POS_tfidf = TfidfVectorizer(tokenizer = okt_tokenizer, min_df = 2)
POS_dtm = POS_tfidf.fit_transform(POS_description_noun_join)

POS_vocab = dict()

for idx, word in enumerate(POS_tfidf.get_feature_names_out()):
    POS_vocab[word] = POS_dtm.getcol(idx).sum()

POS_words = sorted(POS_vocab.items(), key = lambda x: x[1], reverse = True)

print(POS_words) #작업 확인용 출력
print('10', '=' * 40)
```

- ✓ 긍정 감성 뉴스에 대한 TF-IDF 기반 문서단어행렬(DTM) 구성
- ✓ 문서에 나타난 단어의 TF-IDF를 구하는 작업은 문서 단위로 토큰이 연결되어 있는 POS_description_ noun_join사용
- ✓ TfidfVectorizer 객체를 생성하고 POS_description_noun_join에 대해 TF-IDF 값을 구하여 DTM을 구성
- ✓ TFIDF 값의 합을 구하고 내림차순정렬



1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

- TFIDF 값 구한 후 정렬(부정)

```
[('코로나', 34.33164055974099), ('바이러스', 15.321751669293597)
1.793650974243455), ('확산', 9.695809335612335), ('대종상영화제'
386), ('지역', 7.884816390865208), ('방역', 7.677387404057251),
```

(계속) 4칸

```
#-----------------------------------------------------------
NEG_tfidf = TfidfVectorizer(tokenizer = okt_tokenizer, min_df = 2 )
NEG_dtm = NEG_tfidf.fit_transform(NEG_description_noun_join)

NEG_vocab = dict()

for idx, word in enumerate(NEG_tfidf.get_feature_names_out()):
    NEG_vocab[word] = NEG_dtm.getcol(idx).sum()

NEG_words = sorted( NEG_vocab.items(), key = lambda x: x[1], reverse = True)

print(NEG_words) #작업 확인용 출력
print('11', '=' * 40)
```

- ✓ 부정 감성 뉴스에 대한 TF-IDF 기반 문서단어행렬(DTM) 구성
- ✓ 문서에 나타난 단어의 TF-IDF를 구하는 작업은 문서 단위로 토큰이 연결되어 있는 NEG_description_ noun_join사용
- ✓ TfidfVectorizer 객체를 생성하고 NEG_description_noun_join에 대해 TF-IDF 값을 구하여 DTM을 구성
- ✓ TFIDF 값의 합을 구하고 내림차순정렬

목표 설정 → 데이터 수집 → 데이터 준비 → 데이터 탐색 → 데이터 모델링 → 결과 시각화

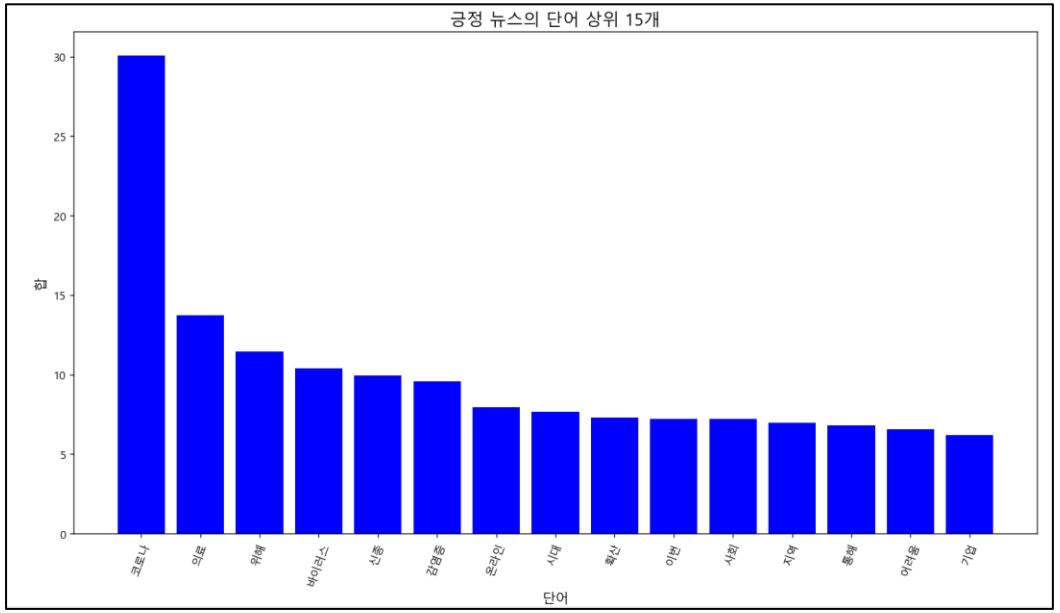
1. 텍스트마이닝

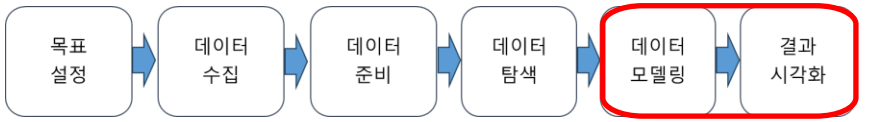
⏳ 데이터모델링 및 시각화

- 바 차트 시각화

(계속) 4칸

```
#-----------------------------------------------------------
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

fm._get_fontconfig_fonts()
font_location = 'C:/Windows/Fonts/malgun.ttf'
font_name = fm.FontProperties(fname = font_location).get_name()
matplotlib.rc('font', family = font_name)

max = 15  #바 차트에 나타낼 단어의 수

plt.bar(range(max), [i[1] for i in POS_words[:max]], color = "blue")
plt.title("긍정 뉴스의 단어 상위 %d개" %max, fontsize = 15)
plt.xlabel("단어", fontsize = 12)
```





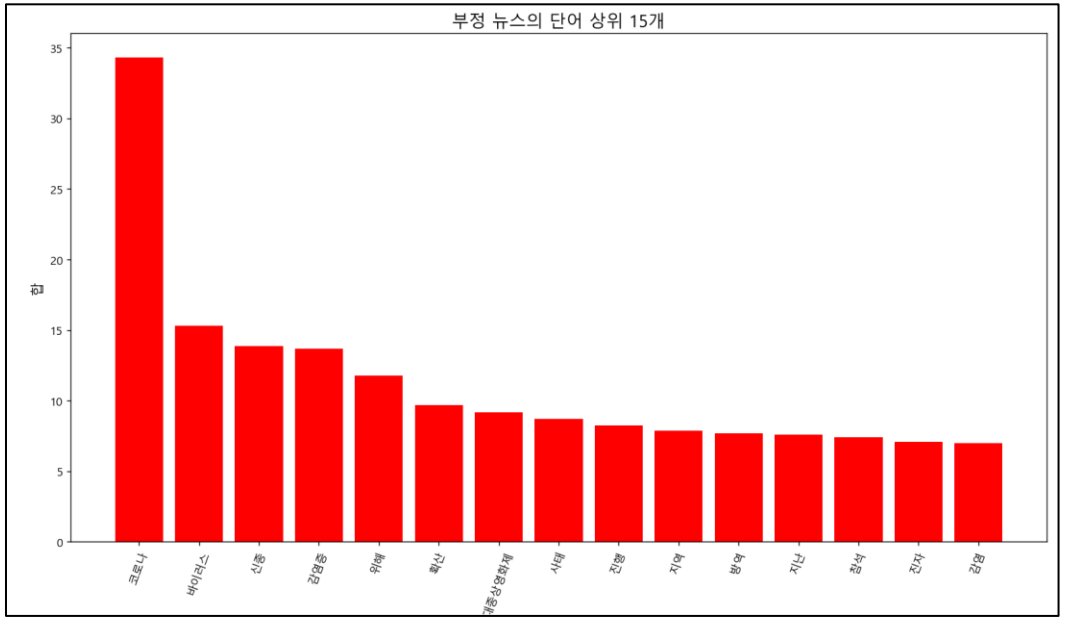
1. 텍스트마이닝

⏳ 데이터모델링 및 시각화

- 바 차트 시각화

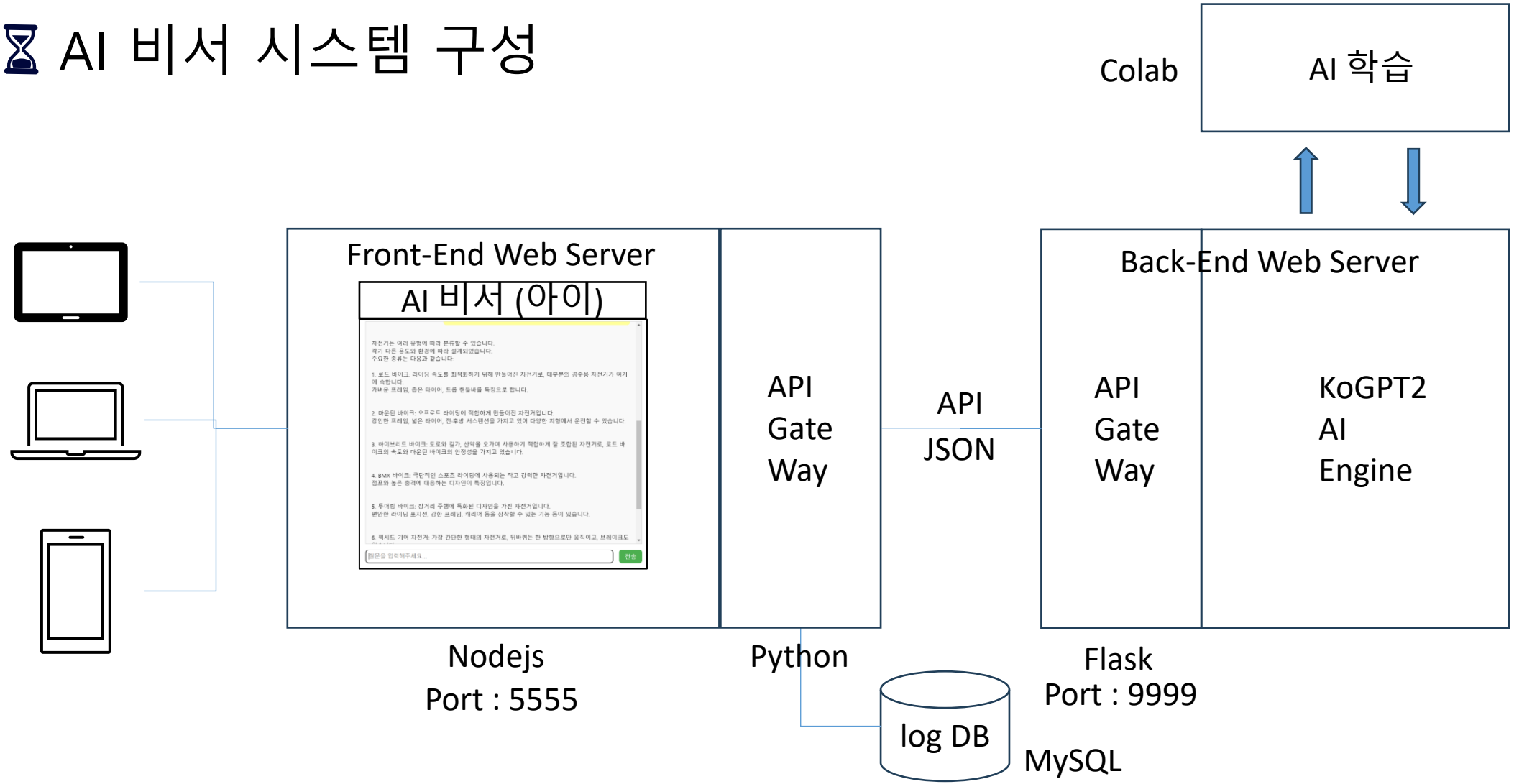
(계속) 4칸

```
plt.ylabel("합", fontsize = 12)
plt.xticks(range(max), [i[0] for i in POS_words[:max]], rotation = 70)

plt.show()

plt.bar(range(max), [i[1] for i in NEG_words[:max]], color = "red")
plt.title("부정 뉴스의 단어 상위 %d개" %max, fontsize = 15)
plt.xlabel("단어", fontsize = 12)
plt.ylabel("합", fontsize = 12)
plt.xticks(range(max), [i[0] for i in NEG_words[:max]], rotation = 70)

plt.show()
```



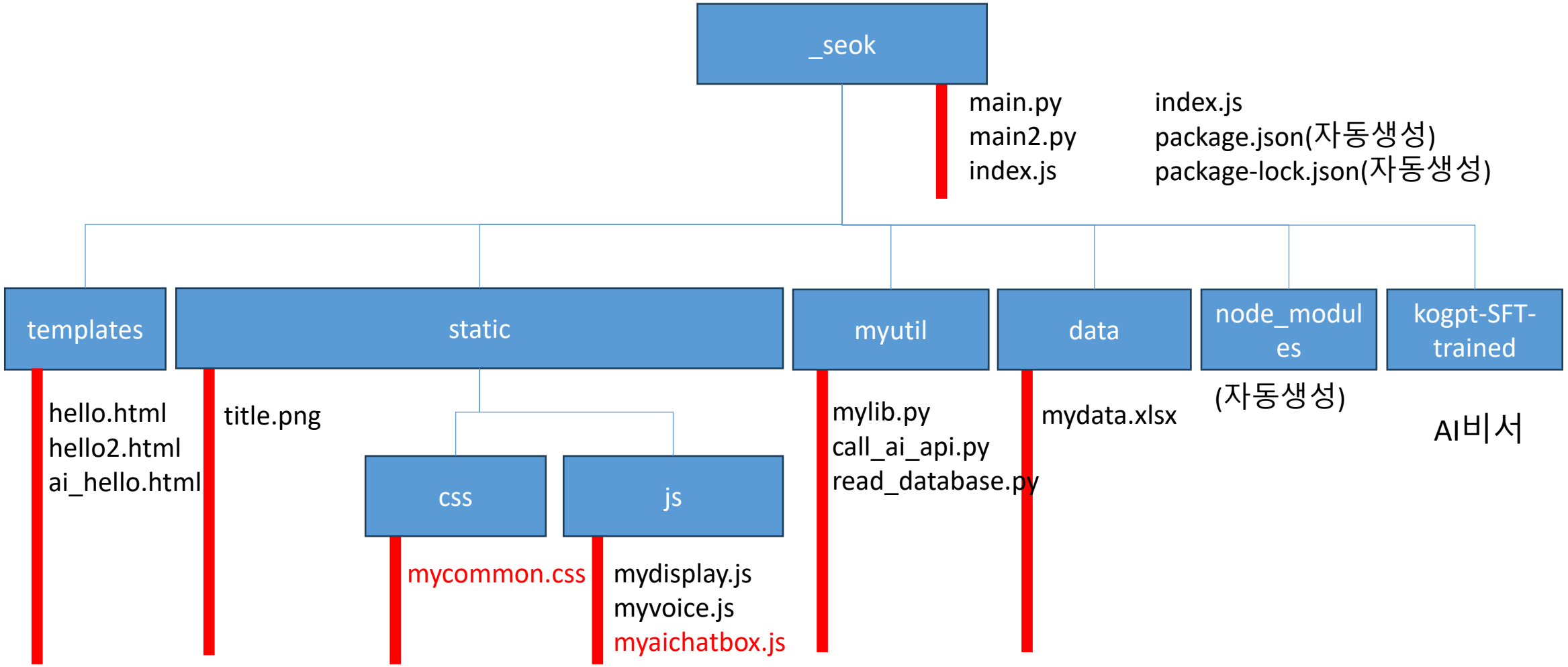
2. AI 비서

AI 비서 시스템 구성



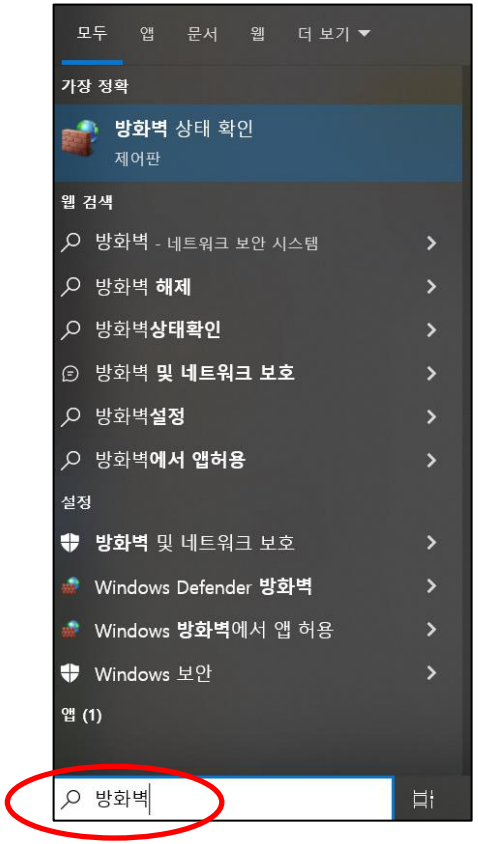
2. AI 비서

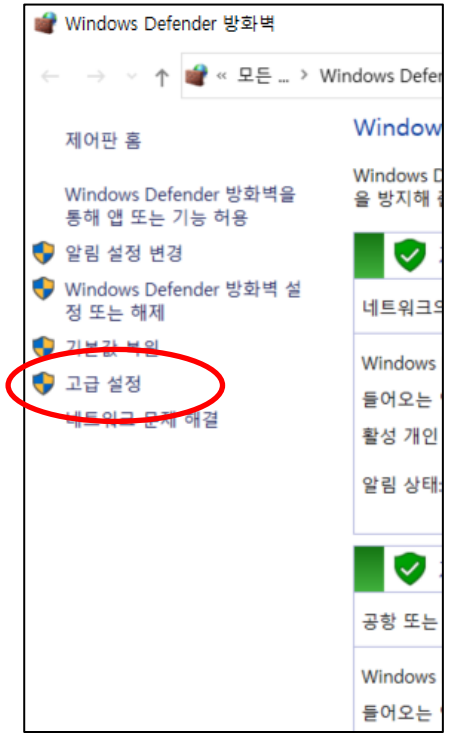
⧖ 디렉토리 구성

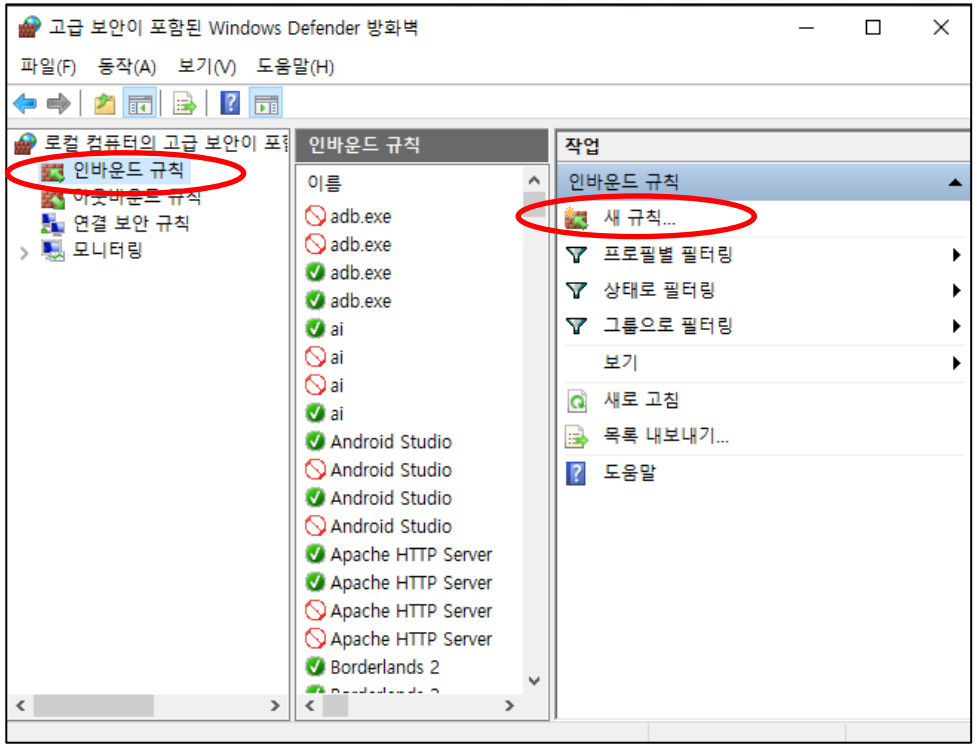


2. AI 비서

⏳ 방화벽열기

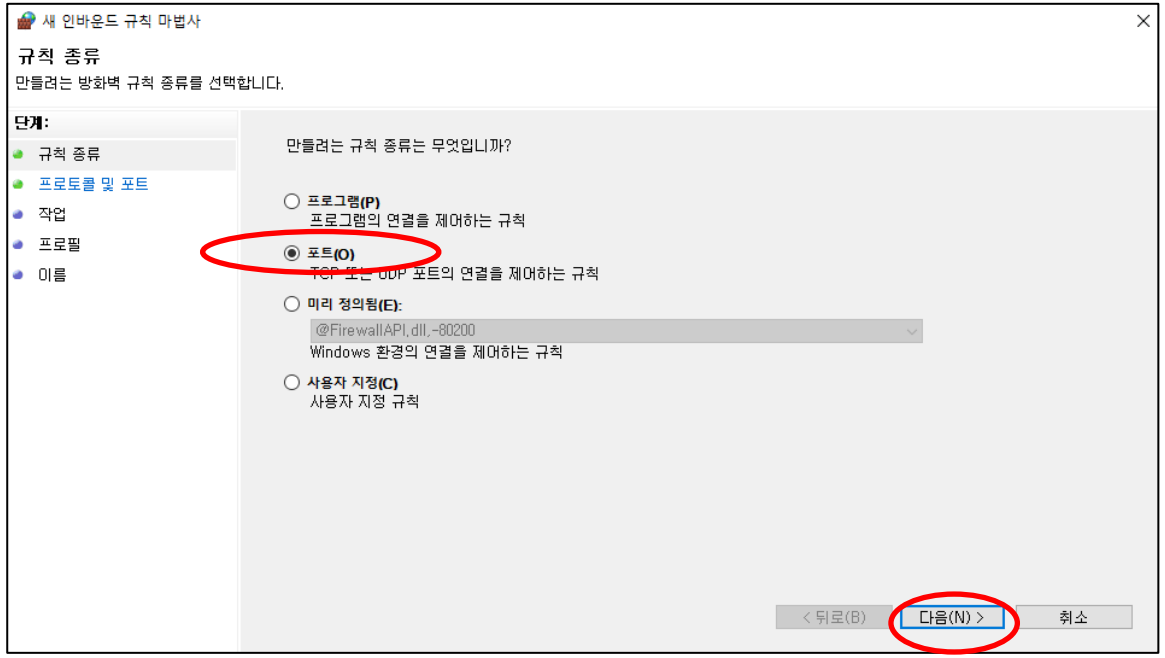

모두
앱
문서
웹
더 보기
가장 정확
방화벽 상태 확인
제어판
웹 검색
방화벽 - 네트워크 보안 시스템
방화벽 해제
방화벽상태확인
방화벽 및 네트워크 보호
방화벽설정
방화벽에서 앱허용
설정
방화벽 및 네트워크 보호
Windows Defender 방화벽
Windows 방화벽에서 앱 허용
Windows 보안
앱 (1)
방화벽

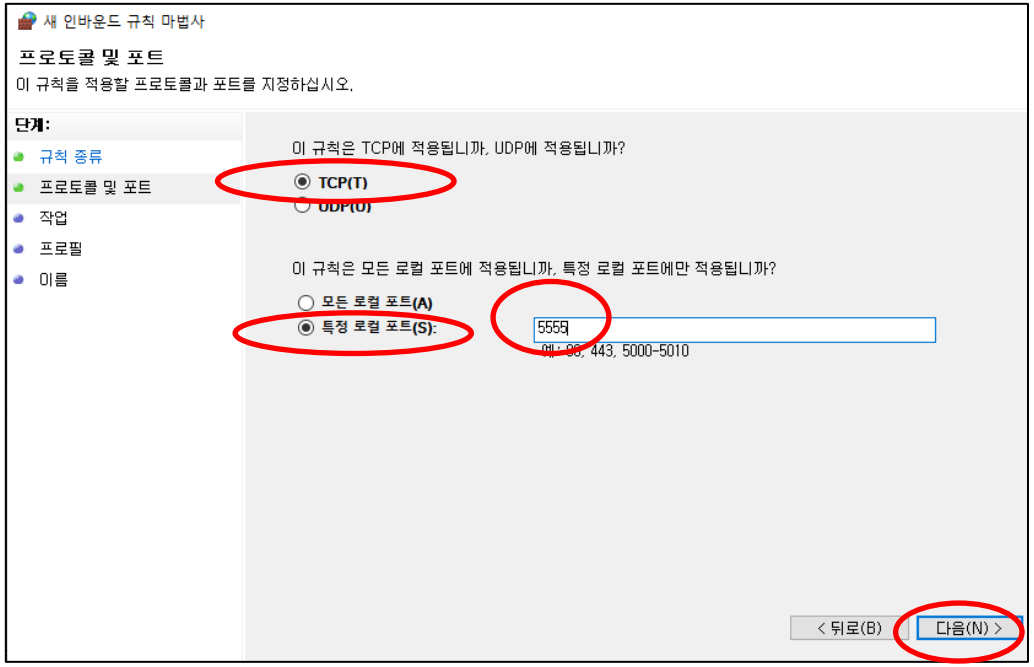

Windows Defender 방화벽
제어판 홈
Windows Defender 방화벽을 통해 앱 또는 기능 허용
알림 설정 변경
Windows Defender 방화벽 설정 또는 해제
고급 설정


고급 보안이 포함된 Windows Defender 방화벽
파일(F)
동작(A)
보기(V)
도움말(H)
로컬 컴퓨터의 고급 보안이 포
인바운드 규칙
연결 보안 규칙
모니터링
이름
adb.exe
adb.exe
adb.exe
adb.exe
ai
ai
ai
ai
Android Studio
Android Studio
Android Studio
Android Studio
Apache HTTP Server
Apache HTTP Server
Apache HTTP Server
Apache HTTP Server
Borderlands 2
작업
인바운드 규칙
새 규칙...
프로필별 필터링
상태로 필터링
그룹으로 필터링
보기
새로 고침
목록 내보내기...
도움말

2. AI 비서

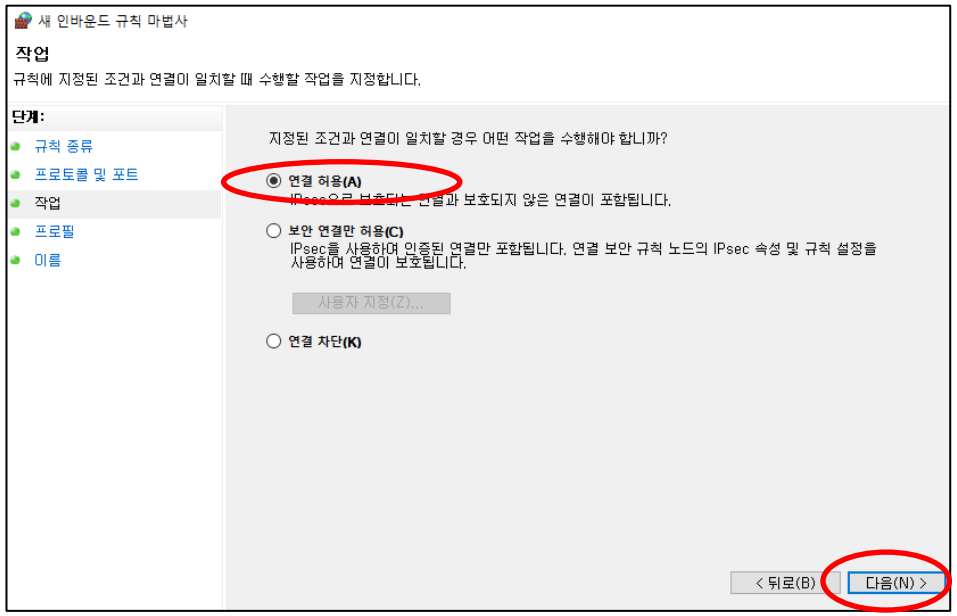
⏳ 방화벽열기

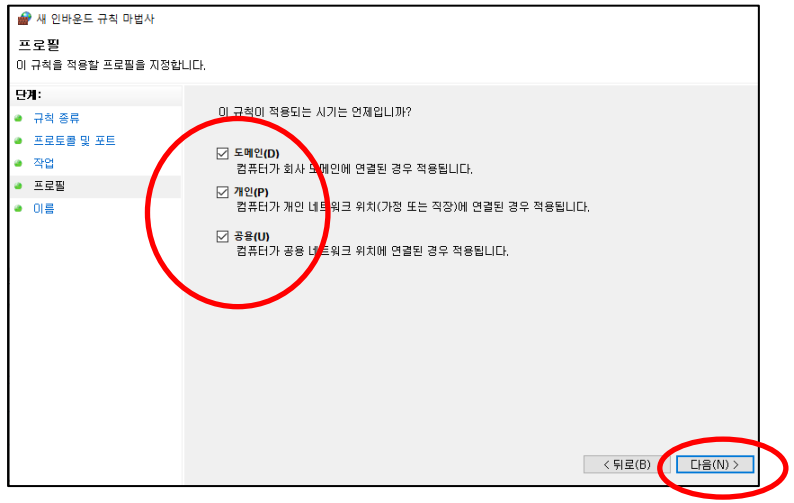

새 인바운드 규칙 마법사
규칙 종류
만들려는 방화벽 규칙 종류를 선택합니다.
단계:
규칙 종류
프로토콜 및 포트
작업
프로필
이름
만들려는 규칙 종류는 무엇입니까?
프로그램(P)
프로그램의 연결을 제어하는 규칙
포트(O)
TCP 또는 UDP 포트의 연결을 제어하는 규칙
미리 정의됨(E):
@FirewallAPI.dll,-80200
Windows 환경의 연결을 제어하는 규칙
사용자 지정(C)
사용자 지정 규칙
< 뒤로(B)
다음(N) >
취소

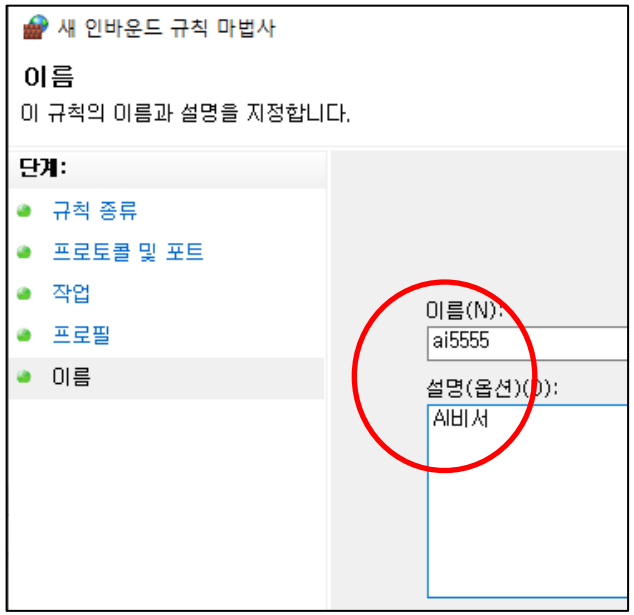

새 인바운드 규칙 마법사
프로토콜 및 포트
이 규칙을 적용할 프로토콜과 포트를 지정하십시오.
단계:
규칙 종류
프로토콜 및 포트
작업
프로필
이름
이 규칙은 TCP에 적용됩니까, UDP에 적용됩니까?
TCP(T)
UDP(U)
이 규칙은 모든 로컬 포트에 적용됩니까, 특정 로컬 포트에만 적용됩니까?
모든 로컬 포트(A)
특정 로컬 포트(S):
5555
예: 80, 443, 5000-5010
< 뒤로(B)
다음(N) >

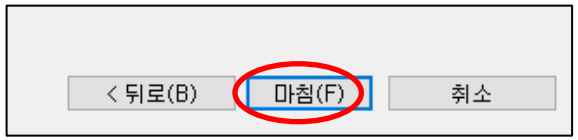
2. AI 비서

⏳ 방화벽열기


새 인바운드 규칙 마법사
작업
규칙에 지정된 조건과 연결이 일치할 때 수행할 작업을 지정합니다.
단계:
규칙 종류
프로토콜 및 포트
작업
프로필
이름
지정된 조건과 연결이 일치할 경우 어떤 작업을 수행해야 합니까?
연결 허용(A)
보안 연결만 허용(C)
IPsec을 사용하여 인증된 연결만 포함됩니다. 연결 보안 규칙 노드의 IPsec 속성 및 규칙 설정을 사용하여 연결이 보호됩니다.
사용자 지정(Z)...
연결 차단(K)
< 뒤로(B)
다음(N) >


새 인바운드 규칙 마법사
프로필
이 규칙을 적용할 프로필을 지정합니다.
단계:
규칙 종류
프로토콜 및 포트
작업
프로필
이름
이 규칙이 적용되는 시기는 언제입니까?
도메인(D)
컴퓨터가 회사 도메인에 연결된 경우 적용됩니다.
개인(P)
컴퓨터가 개인 네트워크 위치(가정 또는 직장)에 연결된 경우 적용됩니다.
공용(U)
컴퓨터가 공용 네트워크 위치에 연결된 경우 적용됩니다.
< 뒤로(B)
다음(N) >


새 인바운드 규칙 마법사
이름
이 규칙의 이름과 설명을 지정합니다.
단계:
규칙 종류
프로토콜 및 포트
작업
프로필
이름
이름(N):
ai5555
설명(옵션)(D):
AI비서


< 뒤로(B)
마침(F)
취소

2. AI 비서

static/css/mycommon.css 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
/*============================*/
/* 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;
}
```

2. AI 비서

static/css/mycommon.css 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
.user-message {
    align-self: flex-start;
    font-weight: bold;
    background-color: #ffff99;
    padding: 0.5rem;
    border-radius: 0.5rem;
    max-width: 70%;
    word-wrap: break-word;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 1rem;
    box-sizing: border-box;
    background-color: #fff;
    box-shadow: 0 0 5px rgba(0, 255, 0, 0.1);
    position: absolute;
}
```

2. AI 비서

static/css/mycommon.css 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
.chatbox {
    display: flex;
    flex-direction: column;
    max-height: 700px;
    overflow-y: auto;
    border: 1px solid #ccc;
    padding: 1rem;
    margin-bottom: 1rem;
    box-sizing: border-box;
    background-color: #fff;
}
```

2. AI 비서

static/js/myaichatbox.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
//=============================================
// AI 채팅 박스
//=============================================
function typeWriter(element, text, i, fnCallback) {
    if (i < text.length) {
        if (text.charAt(i) === '<') {
            var tag = '';
            while (text.charAt(i) !== '>') {
                tag += text.charAt(i);
                i++;
            }
            tag += '>';
            element.innerHTML += tag;
            i++;
        } else {
            element.innerHTML += text.charAt(i);
            i++;
        }
```

2. AI 비서

static/js/myaichatbox.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
        setTimeout(function() { typeWriter(element, text, i, fnCallback);}, 10);
    } else if (typeof fnCallback == 'function') {
        setTimeout(fnCallback, 700);
    }
}

function scrollToBottom() {
    $('#chatbox').animate({scrollTop: $('#chatbox').prop('scrollHeight')}, 700);
}

function animateBotResponse(response) {
    var botMessage = $('<div class="bot-message"></div>');
    $('#chatbox').append(botMessage);
    typeWriter(botMessage[0], response, 0, function() {
        scrollToBottom();
    });
}
```

2. AI 비서

static/js/myaichatbox.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

```
$(function() {
    $('#input-form').submit(function(event) {
        event.preventDefault();
        var user_input = $('#input_data').val();
        if (!user_input) return;
        $('#chatbox').append('<div class="user-message">' + user_input + '</div>');
        $('#input_data').val('');
        scrollToBottom();
        $.ajax({
            url: '/get_answer',
            type: 'POST',
            data: JSON.stringify({question: user_input}),
            contentType: 'application/json',
            success: function(data) {
                var bot_response = data.answer;
                bot_response = bot_response.replace(/\n/g, '<br>');
                animateBotResponse(bot_response);
            }
```

2. AI 비서

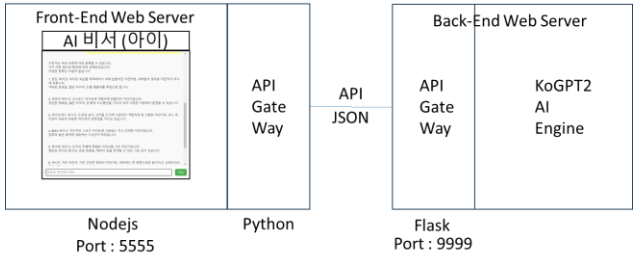
static/js/myaichatbox.js 프로그램 수정

- 붉은 색만 추가

4칸 8칸 12칸

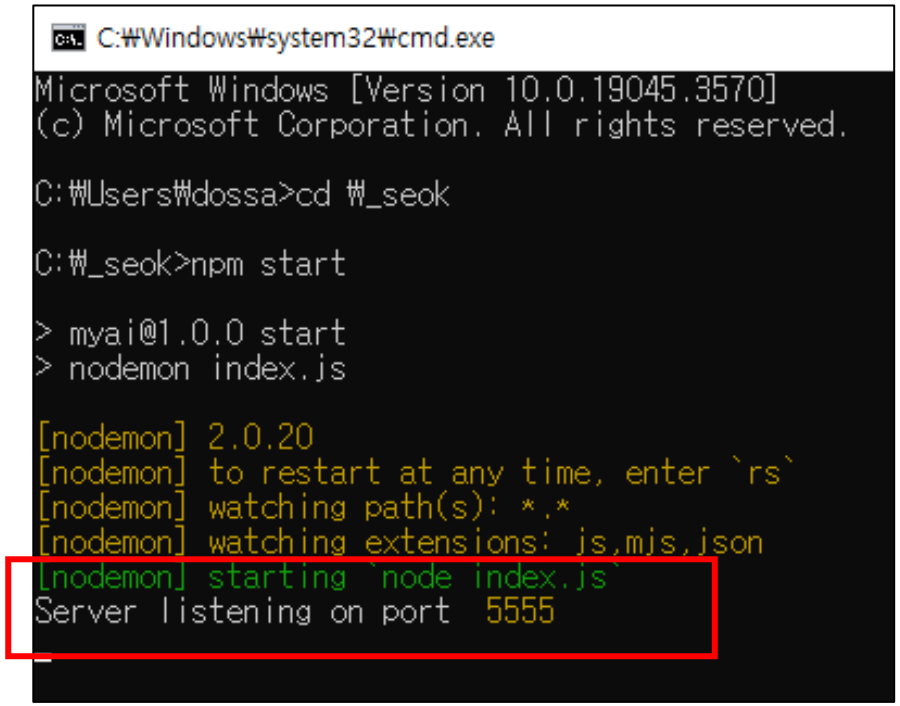
```
        });
    });
});
```

2. AI 비서



실행

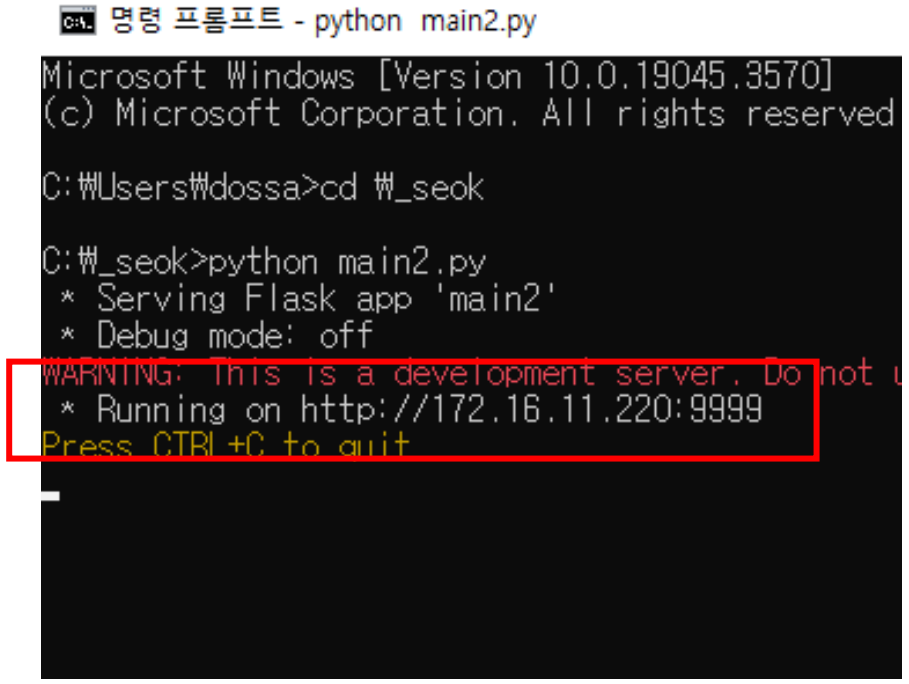
- nodejs 실행



cd _seok

npm start

- Flask 실행



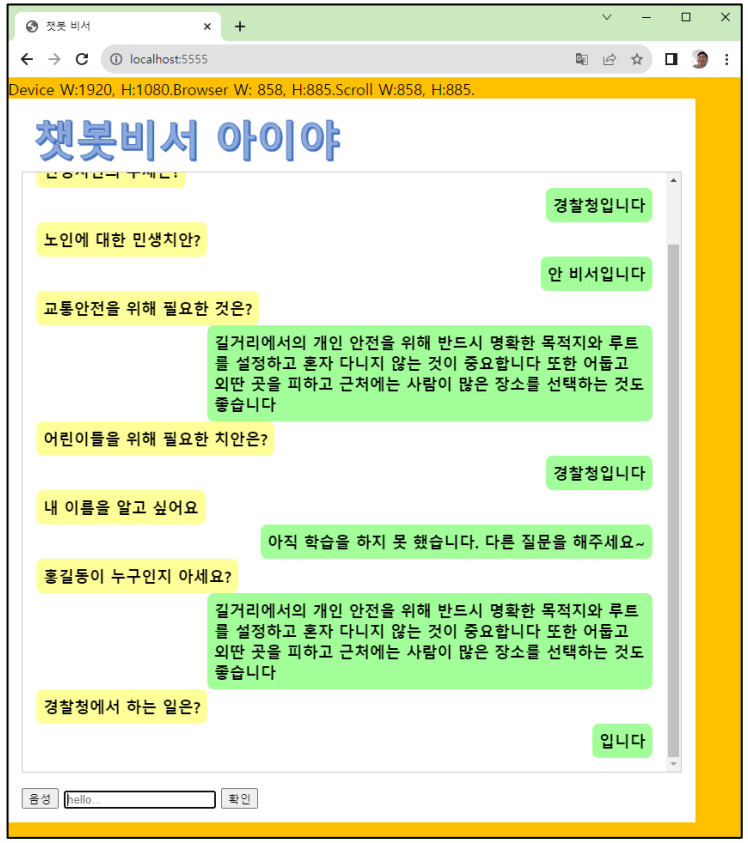
cd _seok

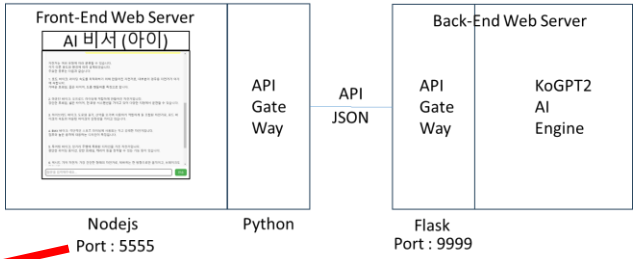
python main2.py

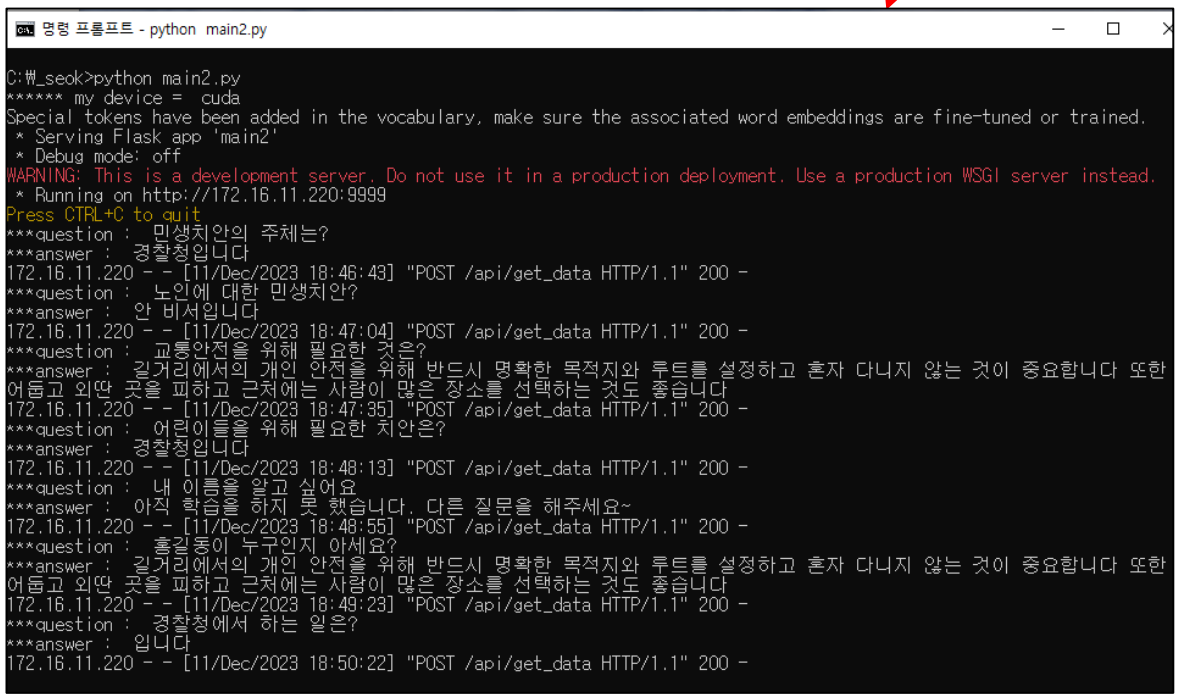
2. AI 비서

⌛ 접속

http://localhost:5555/





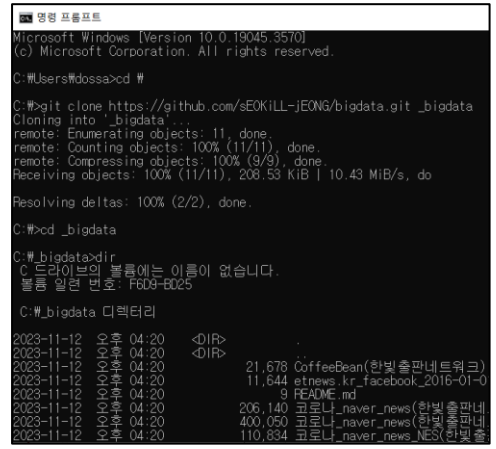


기타 프로그램 설치

- Git 다운로드 : https://git-scm.com
- 데이터셋 다운로드

CMD 창을 열고

```
cd ₩
git clone https://github.com/sEOKiLL-jEONG/bigdata.git _bigdata
cd _bigdata
dir
```



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

유투버 : 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

QnA