

CONSUMER **STORAGE**

COMPUTERS **MARKETING** **SAMPLE**

BYTES **BIG DATA** **RESEARCH**

BEHAVIOR **ANALYTICS** **TECHNOLOGY**

INFORMATION **SIZE** **INTERNET**



박성용 SeongYong Park

☎ 010-9522-0172

✉ pty04290@gmail.com

🐙 <https://github.com/pty902/Resume>

Introduce 팀 프로젝트에서 **PM**을 맡아 Ideation을 담당하였으며 전체적인 일정과 계획을 짜서 팀원들에게 업무 분담을 하였습니다.

또한 OpenAPI, WebCrawling 등의 기술을 사용해서 팀 프로젝트의 **자료수집 및 전처리** 파트를 담당하였습니다.

SKILL (1~5)

Python 3

Linux 3

SQL 3

R 3

HTML5 3

CSS3 3

JavaScript 3



MariaDB



ORACLE
DATABASE



Markup Language
Content

Style sheet Language
Presentation

Programming Language
Behavior

Interests

TensorFlow
Machine Learning
Deep Learning



- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획

- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획

프로젝트 명 : Fine Dust

개발기간 : 2018.12.06 ~ 2019.01.23

참여인원 : 3명

담당업무 : PM (자료수집 및 전처리)

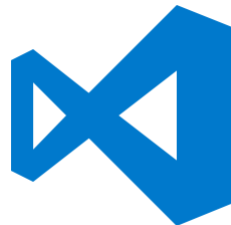
개발환경 : Windows7 64bit

Python 3.6.4 R 3.5.1

MariaDB 10.3 MYSQL 5.7

사용도구 : Jupyter 5.0.0, VSCode , Github Desktop
R Studio 1.1.4, 공공데이터 OpenAPI, Sublime text3
AWS EC2, Xshell 6, Ubuntu 17.10 64 bit

사용기술 : Python R HTML5 CSS3 JavaScript



서울경제

미세먼지에도 마스크 안쓰는 노인

미세먼지에도 마스크 안쓰는 노인

면역력 약해 질병발생 높는데
"귀찮아서" 악영향 인식 안해

기획 보기

오지현 기자 | 2018-01-21 17:19:13 | 기획·연재 31면

역량

기획사육률



초미세먼지 주의보가 발령된 지난 20일 종로구 탑골공원을 찾은 노인들 중 대다수가 마스크를 쓰지 않은 채 야외에서 담소를 나누고 있다. /오지현기자

미세먼지에대한 인식미비

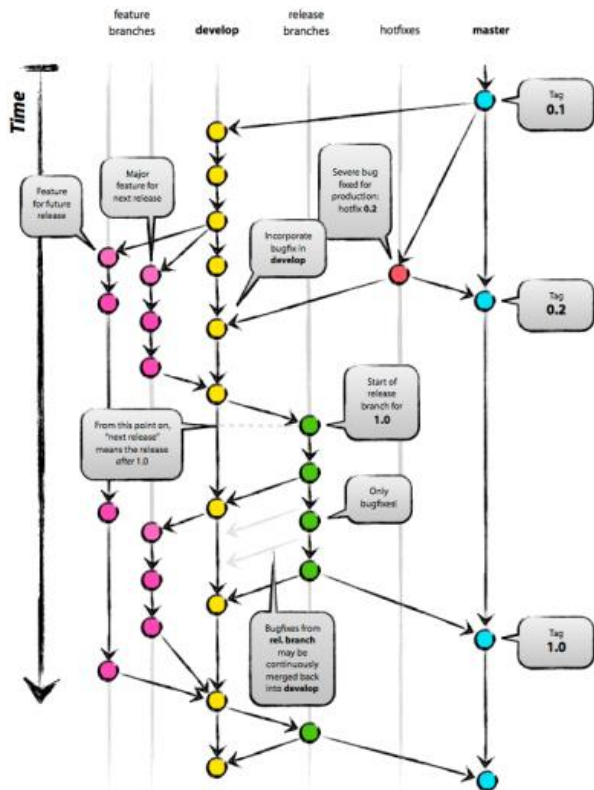
미세먼지 개의치 않는 취약계층인 노인들
삼겹살 먹으면 미세먼지가 배출된다는 잘못된 상식
식약처 인증을 받은 마스크 같은 관련 제품 필요성 인식 부족

프로젝트 프로세스

1. 미세먼지 관련 제품 업체에서
매출 상승을 위한 분석 의뢰를 받았다고 가정
2. 미세먼지와 관련 질병과의 상관관계를 분석하여
소비자들에게 관련 제품의 필요성을 인식시키고
객관적인 데이터에 기반한 관련 제품 판매전략 수립
3. 현업이 정말 필요로 하고 가치 있어 할 만한
분석을 하기 위해 업체와 Co-Work를 하고
분석 결과를 이해서키며 피드백을 받는 프로세스 추가

출처 : 서울경제

<https://www.sedaily.com/NewsView/1RUIJ0XL33>



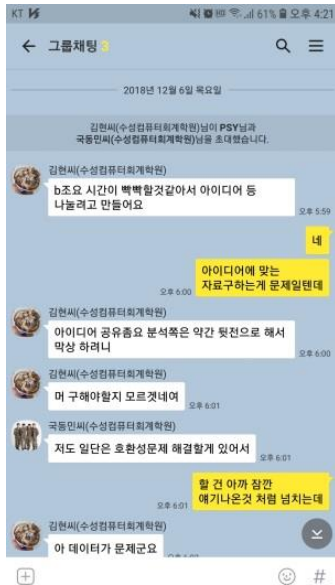
Branch를 5개로 나누어 버전 관리하였습니다.

작업물 Merge시 Conflict를 최대한 방지할 수 있었고

작업과정을 일목요연하게 볼 수 있었습니다.

출처 : nvie.com

<https://nvie.com/posts/a-successful-git-branching-model/>



Initial commit	Verified	a41bc7d	<>
ks5721 committed on 12 Dec 2018			
Add files via upload	Verified	bd34227	<>
pty902 committed on 17 Dec 2018			
유아교육 업체관련 정보	Verified	29b78b8	<>
kdm0904 committed on 19 Dec 2018			
maps	Verified	a54dc79	<>
kdm0904 committed on 7 Jan			
Fine Dust PPT For Presentation ver 1.0	Verified	56f3447	<>
pty902 committed on 20 Jan			
Commits on Jan 23, 2019			
html		d8727dd	<>
ks5721 committed 27 days ago			
Fine Dust PPT For Print	Verified	5739716	<>
pty902 committed 27 days ago			
Fine Dust PPT For Presentation	Verified	7fc1b26	<>
pty902 committed 27 days ago			

2018.12.06 프로젝트 팀 결성

2018.12.12 그룹 Github 생성

2018.12.17 Feature Branch 자료저장

인터넷검색 OpenAPI WebCrawling 등으로 수집한 자료들과 초기 분석 결과를 홈페이지 저장

2018.12.19 Develop Branch 자료저장

전처리된 자료들과 CSS3 JavaScript 처리된 홈페이지, 완성된 분석 결과물 (어린이집 경로당 관련 질병 등) 저장

2019.01.07 Release Branch 자료저장

초기 PPT, Develop 자료 중 회의를 거쳐 선별된 어린이집 경로당 관련 질병 관련 제품 분석 결과물을 코드와 함께 저장

2019.01.20 Hotfix Branch 자료저장

PPT, 분석 결과물 수정 및 아이디어 추가, 코드 정리

2019.01.23 Master Branch 자료저장

PPT 홈페이지 마무리 후 프로젝트 종료

- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획

```
<?xml version="1.0" encoding="UTF-8"?><response><header><resultCode>00</resultCode><resultMsg>NORMAL SERVICE.</resultMsg></header><body><items><stationName>창전동</stationName><totalCount>3, '0': {'stationName': '창전동', 'addr': '경기 이천시 영창로 163번길 28평생학습센터 서희관내 여론선설터', 'tm': 8.6}, '1': {'stationName': '설성면', 'addr': '경기 이천시 설성면 신필리산 88-5(전파연구소 입구)', 'tm': 9.3}, '2': {'stationName': '여주', 'addr': '경기 여주시 여흥로11번길 26중앙동주민센터 옥상', 'tm': 14.9}</items><numOfRows>10</numOfRows></body></response>
```

```
pm10 = a.station("수창동", "DAILY")['0']['pm10Value']
# 첫번째 pm10Value값만 표시
```

```
print(pm10)
```

```
95
```

```
print(c.nearby("244148.546388", "412423.75772"))
# TM측량 방식 x,y좌표의 근접측점소 표시

{'totalCount': 3, '0': {'stationName': '창전동', 'addr': '경기 이천시 영창로 163번길 28평생학습센터 서희관내 여론선설터', 'tm': 8.6}, '1': {'stationName': '설성면', 'addr': '경기 이천시 설성면 신필리산 88-5(전파연구소 입구)', 'tm': 9.3}, '2': {'stationName': '여주', 'addr': '경기 여주시 여흥로11번길 26중앙동주민센터 옥상', 'tm': 14.9}}
```

```
print(c.detail("대구", "진천동"))
# 주소와 측정소명으로 검색한 데이터

{'totalCount': 1, '0': {'addr': '대구 달서구 월배로 131월배초등학교', 'dmX': '35.816725', 'dmY': '128.528732', 'item': 'S02, CO, 03, NO2, PM10, PM2.5', 'mangName': '도시대기', 'map': '', 'oper': '대구광역시보건환경연구원', 'photo': 'http://www.airkorea.or.kr/airkorea/station_photo/NA/MS/station_images/422204/INS1DE_0THER1.jpg', 'stationName': '진천동', 'vrm1': '', 'year': '2017'}}
```

```
print(c.tmcde("수창동"))
# 읍면동명으로 검색한 데이터

{'totalCount': 1, '0': {'sidName': '대구광역시', 'sggName': '중구', 'undName': '수창동', 'tmX': '343144.187069', 'tmY': '265041.237971'}}
```

```
print(a.strangelist(page_no='1', num_of_rows='10'))
# 통합대기환경지수 나쁨 등급 이상인 주소와 측정소명

{'totalCount': 1, '0': {'addr': '인천 남동구 남동대로 217(교잔동)한국산업단지관리공단 인천지역본부 옥상', 'stationName': '교잔'}}
```

```
print(a.sido("대구"))
# 시도명으로 검색한 데이터

{'totalCount': 15, '0': {'dateTime': '2018-12-21 08:00', 'mangName': '도시대기', 'stationName': '수창동', 'khaiGrade': '3', 'khaiValue': '231', 'coGrade': '1', 'coValue': '1.2', 'no2Grade': '2', 'no2Value': '0.043', 'o3Grade': '1', 'o3Value': '0.003', 'pm10Grade': '3', 'pm10Value': '95', 'pm10Value24': '90', 'pm25Grade': '3', 'pm25Value': '62',
```

공공데이터포털 OpenAPI를 **urllib** 파이썬 모듈을 사용하여 **request** 후 **parsing** 하여 자료수집

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20ust\(Open%20API\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20ust(Open%20API).ipynb) In [1] 참고

핵심코드

```
import json
import urllib.parse
import urllib.request
```

```
class PollutionData(object):
    def __init__(self, imported_key):
        self._service_key = str(imported_key)
        self._mainurl = 'http://openapi.airkorea.or.kr/openapi/services/rest/ArprtnInforInquireSvc/'

    def station(self, station_name, data_term, page_no='1', num_of_rows='10', ver='1.2'):
        verurl = '&ver=' + ver
        # It's for changing version, it seems like version system doesn't work on OpenAPI, so I comment them.
        # verurl = '&ver='
        # if ver == '0': verurl = ''
        # else: verurl += ver

        urllist = [
            self._mainurl + 'getMsrstnAcctoRltMasureDnsty?', 'stationName=', urllib.parse.quote(station_name),
            '&dataTerm=', data_term, '&pageNo=', page_no, '&numOfRows=', num_of_rows,
            '&ServiceKey=', self._service_key, '&returnType=json', verurl
        ]
        url = ''.join(urllist)

        response = urllib.request.urlopen(url).read().decode("utf-8")
        jsondata = json.loads(response)

        returndata = dict()
        returndata['totalCount'] = jsondata['totalCount']

        for i in range(0, returndata['totalCount']):
            returndata[str(i)] = dict()
            list = [
                'dateTime', 'mangName', 'khaiGrade', 'khaiValue', 'coGrade',
                'coValue', 'no2Grade', 'no2Value', 'o3Grade', 'o3Value',
                'pm10Grade', 'pm10Value', 'pm10Value24', 'pm25Grade', 'pm25Value',
                'pm25Value24', 'so2Grade', 'so2Value'
            ]
            for j in list:
                returndata[str(i)][j] = jsondata['list'][i][j]

        return returndata
```

```
{'sidoName': '대구', 'dataTime': '2014-03', 'so2Avg': '0.006', 'coAvg': '0.5', 'o3Avg': '0.033', 'no2Avg': '0.036', 'pm10Avg': '59', 'so2Max': '0.033', 'coMax': '1.7', 'o3Max': '0.094', 'no2Max': '0.103', 'pm10Max': '416', 'so2Min': '0.002', 'coMin': '0.3', 'o3Min': '0.003', 'no2Min': '0.008', 'pm10Min': '9'}
```

핵심코드

지 역	측정 시간	so2Avg	coAvg	o3Avg	no2Avg	pm10Avg	so2Max	coMax	o3Max	no2Max	pm10Max	so2Min	coMin	o3Mi
0	대구 2014-03	0.006	0.5	0.033	0.036	59	0.033	1.7	0.094	0.103	416	0.002	0.3	0.003

지 역	측정 시간	so2Avg	coAvg	o3Avg	no2Avg	pm10Avg	so2Max	coMax	o3Max	no2Max	pm10Max	so2Min	coMin	o3Mi
0	대구 2014-03	0.006	0.5	0.033	0.036	59	0.033	1.7	0.094	0.103	416	0.002	0.3	0.003
1	대구 2015-03	0.007	0.6	0.032	0.036	79	0.043	2.0	0.086	0.101	401	0.003	0.2	0.002
2	대구 2016-03	0.006	0.5	0.031	0.035	68	0.037	2.5	0.083	0.100	215	0.002	0.2	0.002

```
sN = dic2['sidoName']
dT = dic2['dataTime']
so2A = dic2['so2Avg']
coA = dic2['coAvg']
o3A = dic2['o3Avg']
no2A = dic2['no2Avg']
pm10A = dic2['pm10Avg']
so2Ma = dic2['so2Max']
coMa = dic2['coMax']
o3Ma = dic2['o3Max']
no2Ma = dic2['no2Max']
pm10Ma = dic2['pm10Max']
so2Mi = dic2['so2Min']
coMi = dic2['coMin']
o3Mi = dic2['o3Min']
no2Mi = dic2['no2Min']
pm10Mi = dic2['pm10Min']
```

import csv

```
f = open('Daegu_period.csv', 'w', encoding='utf-8', newline='')
wr = csv.writer(f)
wr.writerow(['지역', '측정시간', 'so2Avg', 'coAvg', 'o3Avg', 'no2Avg', 'pm10Avg', 'so2Max', 'coMax', 'o3Max', 'no2Max', 'pm10Max', 'so2Min', 'coMin', 'o3Min', 'no2Min', 'pm10Min'])
wr.writerow([sN, dT, so2A, coA, o3A, no2A, pm10A, so2Ma, coMa, o3Ma, no2Ma, pm10Ma, so2Mi, coMi, o3Mi, no2Mi, pm10Mi])
f.close()
# Daegu_period.csv 파일에 OpenAPI로 얻은 데이터 저장
```

```
ob2.loc[2] = [sN, dT, so2A, coA, o3A, no2A, pm10A, so2Ma, coMa, o3Ma, no2Ma, pm10Ma, so2Mi, coMi, o3Mi, no2Mi, pm10Mi]
```

```
# 세번째 위치에서 sN, dT, so2A, coA, o3A, no2A, pm10A, so2Ma, coMa, o3Ma, no2Ma, pm10Ma, so2Mi, coMi, o3Mi, no2Mi, pm10Mi 값 읽기
```

수집한 데이터를 csv 파이썬 모듈을 사용하여 Daegu_period.csv 파일로 저장

<https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust%20API.ipynb> [90] 참고

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><response><header><resultCode>00</resultCode><resultMsg>NORM
AL SERVICE.</resultMsg></header><body><items><item><age>전체</age><diagAmt01>127,469,000[2013]</diagAmt01><
diagAmt02>140,291,223[2012]</diagAmt02><diagAmt03>159,820,410[2011]</diagAmt03><pintCnt01>
1,853,656[2013]</pintCnt01><pintCnt02>2,074,482[2012]</pintCnt02><pintCnt03>2,199,501[2011]</p
intCnt03><num>1</num><sickCd>J45</sickCd><sickNm>천식</sickNm><vstDdCnt01>6,103,693[2013]</vstDdCnt01><
vstDdCnt02>7,208,364[2012]</vstDdCnt02><vstDdCnt03>8,220,046[2011]</vstDdCnt03></item><item><a
```

전체	127,469,000[2013]	140,291,223[2012]	159,820,410[2011]	1,853,656[2013]	2,074,482[2012]	2,199,501[2011]	J45천식	6,103,693[2013]	7,208,364[2012]	8,220,046[2011]
전체	56,915,881[2013]	63,954,906[2012]	75,607,221[2011]	860,128[2013]	955,841[2012]	1,024,092[2011]	2maleJ45천식	2,894,335[2013]	3,453,146[2012]	4,046,128[2011]
10세미만	23,894,698[2013]	28,745,646[2012]	35,970,014[2011]	388,479[2013]	423,939[2012]	483,679[2011]	3maleJ45천식	1,429,239[2013]	1,809,937[2012]	2,316,639[2011]
10-19세	3,546,344[2013]	4,111,260[2012]	5,079,224[2011]	74,515[2013]	91,284[2012]	106,054[2011]	4maleJ45천식	173,604[2013]	216,550[2012]	266,150[2011]
20-29세	1,798,268[2013]	1,822,439[2012]	2,116,850[2011]	36,875[2013]	41,622[2012]	41,488[2011]	5maleJ45천식	71,915[2013]	79,645[2012]	85,340[2011]
30-39세	2,801,029[2013]	2,926,471[2012]	3,211,219[2011]	58,048[2013]	66,456[2012]	66,591[2011]	6maleJ45천식	125,134[2013]	140,910[2012]	149,086[2011]
40-49세	3,281,505[2013]	3,439,008[2012]	3,944,220[2011]	63,590[2013]	72,227[2012]	72,460[2011]	7maleJ45천식	155,044[2013]	172,902[2012]	185,998[2011]
50-59세	4,787,323[2013]	4,937,599[2012]	5,443,246[2011]	77,350[2013]	85,721[2012]	84,195[2011]	8maleJ45천식	222,253[2013]	245,017[2012]	277,651[2011]
60-69세	5,751,422[2013]	6,475,996[2012]	7,632,933[2011]	76,768[2013]	86,190[2012]	86,992[2011]	9maleJ45천식	271,911[2013]	312,202[2012]	370,456[2011]
70-79세	7,035,229[2013]	7,551,665[2012]	8,411,059[2011]	72,351[2013]	77,651[2012]	73,714[2011]	10maleJ45천식	326,821[2013]	354,959[2012]	370,456[2011]
80세이상	4,020,063[2013]	3,944,822[2012]	3,798,447[2011]	21,058[2013]	21,555[2012]	20,473[2011]	11maleJ45천식	118,414[2013]	121,024[2012]	119,944[2011]

핵심코드

```
from bs4 import BeautifulSoup
import urllib.request

req = urllib.request.urlopen(url)
xml = BeautifulSoup(req, 'xml')

for item in xml.findAll('item'):
    print(item.text)

# 질병성별연령별통계 찾는 코드
# item 클래스의 text만 추출함

age = []
diagAmt01 = []
diagAmt02 = []
diagAmt03 = []
pintCnt01 = []
pintCnt02 = []
pintCnt03 = []
sickCd = []
sickNm = []
vstDdCnt01 = []
vstDdCnt02 = []
vstDdCnt03 = []

list_soup = xml.find_all('item')

for item in list_soup:
    age.append(item.find('age').get_text())
    diagAmt01.append(item.find('diagAmt01').get_text())
    diagAmt02.append(item.find('diagAmt02').get_text())
    diagAmt03.append(item.find('diagAmt03').get_text())
    pintCnt01.append(item.find('pintCnt01').get_text())
    pintCnt02.append(item.find('pintCnt02').get_text())
    pintCnt03.append(item.find('pintCnt03').get_text())
    sickCd.append(item.find('sickCd').get_text())
    sickNm.append(item.find('sickNm').get_text())
    vstDdCnt01.append(item.find('vstDdCnt01').get_text())
    vstDdCnt02.append(item.find('vstDdCnt02').get_text())
    vstDdCnt03.append(item.find('vstDdCnt03').get_text())

# 클래스별로 값을 리스트에 추가함
```

공공데이터포털 OpenAPI를 urllib 파이썬 모듈을 사용하여 request 후 Parsing 하여 자료수집

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20ust%20Disease\(Open%20API\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20ust%20Disease(Open%20API).ipynb) In [12] 참고

	Age	DiagAmt01	DiagAmt02	DiagAmt03	PintCnt01	PintCnt02	PintCnt03	SiokC
0	전체	127,469,000[2013]	140,291,223[2012]	159,820,410[2011]	1,853,656[2013]	2,074,482[2012]	2,199,501[2011]	J45
1	전체	56,915,881[2013]	63,954,906[2012]	75,607,221[2011]	860,128[2013]	955,841[2012]	1,024,092[2011]	J45
2	10세 미만	23,894,698[2013]	28,745,646[2012]	35,970,014[2011]	388,479[2013]	423,939[2012]	483,679[2011]	J45
3	10~19세	3,546,344[2013]	4,111,260[2012]	5,079,224[2011]	74,515[2013]	91,284[2012]	106,054[2011]	J45
4	20~29세	1,798,268[2013]	1,822,439[2012]	2,116,850[2011]	36,875[2013]	41,622[2012]	41,488[2011]	J45

	SickCd	SickNm	Age	DiagAmt01	DiagAmt02	DiagAmt03	PintCnt01	PintCnt02	PintC
0	J45	천식	전체	127,469,000[2013]	140,291,223[2012]	159,820,410[2011]	1,853,656[2013]	2,074,482[2012]	2,199
1	J45	천식	전체	56,915,881[2013]	63,954,906[2012]	75,607,221[2011]	860,128[2013]	955,841[2012]	1,024
2	J45	천식	10세 미만	23,894,698[2013]	28,745,646[2012]	35,970,014[2011]	388,479[2013]	423,939[2012]	483,6
3	J45	천식	10~19세	3,546,344[2013]	4,111,260[2012]	5,079,224[2011]	74,515[2013]	91,284[2012]	106,0
4	J45	천식	20~29세	1,798,268[2013]	1,822,439[2012]	2,116,850[2011]	36,875[2013]	41,622[2012]	41,48

핵심코드

```
import pandas as pd
data = {'Age':age, 'DiagAmt01':diagAmt01, 'DiagAmt02':diagAmt02, 'DiagAmt03':diagAmt03, 'PintCnt01':pintCnt01, #
        'PintCnt02':pintCnt02, 'PintCnt03':pintCnt03, 'SickCd':sickCd, 'SickNm':sickNm, 'VstDdCnt01':vstDdCnt01, #
        'VstDdCnt02':vstDdCnt02, 'VstDdCnt03':vstDdCnt03}
df = pd.DataFrame(data)
df.head()
# Pandas모듈을 사용하여 '데이터프레임'에 : 값을 입력하여 데이터화합
```

```
df = pd.DataFrame(data, columns=['SickCd', 'SickNm', 'Age', 'DiagAmt01', 'DiagAmt02', 'DiagAmt03', #
                                'PintCnt01', 'PintCnt02', 'PintCnt03', 'VstDdCnt01', 'VstDdCnt02', 'VstDdCnt03'])
df.head(5)
# 칼럼의 순서를 정리함
```

```
df.to_csv('../data/disease_01.csv', sep=',', encoding='UTF-8')
# 전처리한 데이터를 disease_01.csv 파일로 저장
```

수집한 데이터를 pandas 파이썬 모듈을 사용하여 disease_01.csv 파일로 저장

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust%20Disease\(Open%20API\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust%20Disease(Open%20API).ipynb) In [43] 참고

[' 기본검색' 입력한 단어가 하나이상 포함된 문서 검색 ' ,
 ' 상세검색' 정확히 일치하는 단어/문장(' ') 반드시 포함하는 단어(+) 제외하는 단어(-) 기본검색의 결과 범위를 줄이고자 할 때 사용합니다. 여러개의 단어를 입력하실 때는콤표(,)로 구분해서 입력하세요.
 ' 연관검색어도들람' 오늘 미세먼지 농도, 오늘미세먼지, 미세먼지 주의보, 초미세먼지 주의보, 오늘 서울 미세먼지, 황사, 날씨, 내일 날씨, 현재미세먼지, 전국날씨, 일기 후 1주일간 유지됩니다. 연관검색어를 다시 보시겠습니까? 열기 ' ,
 ' 질문' 미세먼지 저감 대책(저감장치) 관련, 2019.02.17, 19년도 2월15일 부터 시행 되는 미세먼지 점검대책 관련하여 5등급 차량 중 저감장치 개보되지 않은 차량도 단속 대상이 되나요? 답변 ○ 「미세먼지 저감 및 관리에 관한 특별법」 시행(19.2.15.)에 따라 미세먼지 비상저감조치 발령시 수도권(서울, 인천, 경기)지역에서 시·도 조례로 배출가스 5등급차량의... 지식파트너 답변 님네임국민신문고 ' ,
 ' 질문' 서울 52세/여 상열감, 얼굴이 알아올라요, 미세먼지 때문인가요? 6월 전, 뉴스에서 미세먼지가 많으면 뇌에도 안좋고 암도 생긴다고 하니가 그런 뉴스들은날은 기분이 그래서 그런지 더 심하게 불안하고 우울도 심하고 가슴도 두근거리고... , 답변 초미세먼지는 호흡기를 통해 인체내에 들어온 후에, 혈액 속의 황산화산을 증가시켜 면역력을 떨어뜨리고 중추신경계와 뇌 신경세포를 손상시켜 뇌 질환을 유발하기도.... 한의사 답변 님네임정아안 ' ,
 ' 질문' 우리나라에서 황사나 미세먼지가 가장 2018.11.27, 우리나라에서 황사나 미세먼지가 가장 심했던 때는 언제인가요? , 답변 황사일수와 미세먼지 농도에 관한 데이터는 기상자료개방포털에서 확인하실 수 있습니다. https://data.kma.go.kr/climate/yellowDust/selectYellowDustChart.do?pgmNo=112... 지식파트너 답변 님네임기상철 ' ,
 ' 질문' 알리바바에서 미세먼지 마스크를 사다가 2019.01.21, 1.알리바바에서 미세먼지 마스크를 사다가 한국시장에서 쿠팔이나 지마켓 네이브스토어 등에서 팔고 있습니다 무엇보다 시작해야할까요?? 2.수입하는 품목도 여러가지... , 답변 미세먼지 마스크는 의약품에 해당하므로 수입을 위해서는 식품의약품안전처장에게 수입업신고를 해야 하고, 수입시마다 한국의약품수출입화환장에게... , 관세사 답변 님네임임형철 ' ,
 ' 질문' 미세먼지 많은 날 종일 밖에서 뛰었다 2019.02.07, 피로누적인가요, 미세먼지 때문인가요? 미세먼지 많은 날 종일 밖에서 뛰었다

urllib 파이썬 모듈을 사용하여 네이버 지식IN에 request 하여 자료수집

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust\(Nltk\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust(Nltk).ipynb) In [4] 참고

핵심코드

```
from tqdm import tqdm_notebook
present_candi_text = []
for n in tqdm_notebook(range(1, 10000, 10)):
    tmp1 = 'https://search.naver.com/search.naver?where=kin&sm=tab_jum&query='
    quote = rep.quote_plus("미세먼지")
    url = tmp1 + quote
    html = urllib.request.Request(url, headers={'User-Agent': 'Mozilla/5.0'})
    response = urlopen(html)
    soup = BeautifulSoup(response, "html.parser")
    tmp = soup.find_all('dl')
    for line in tmp:
        present_candi_text.append(line.text)
    time.sleep(0.5)
# 네이버 지식IN에 미세먼지 검색어로 답변 1만개 추출함
```



```
[('미세먼지', 29996),
 ('오늘', 4617),
 ('감', 4614),
 ('저', 3845),
 ('국가', 3845),
 ('날씨', 3079),
 ('황사', 3077),
 ('때문', 3076),
 ('인가요', 3076),
 ('날', 3076),
 ('계속', 3076),
 ('훈련', 3076),
 ('롯데', 3076),
 ('입력', 2310),
 ('서울', 2308),
 ('일', 2308),
 ('부터', 2307),
 ('배출', 2307),
 ('뇌', 2307),
 ('시켜', 2307),
 ('마스크', 2307),
 ('수입', 2307),
 ('의사', 2307),
 ('좋은', 2307),
 ('아외', 2307),
 ('요즘', 2307),
 ('환기', 2307),
 ('포함', 1540),
 ('주의보', 1540),
 ('/', 1539),
 ('농도', 1539),
 ('초미세먼지', 1539),
 ('내', 1539),
 ('현재', 1539),
 ('대책', 1538),
```

핵심코드

```
import nltk
from konlpy.tag import Okt; t = Okt()

present_text = ''
for each_line in present_candi_text[:10000]:
    present_text = present_text + each_line + '\n'
# 하나의 글로 present_text에 저장

tokens_ko = t.morphs(present_text)
tokens_ko
# 1만개 문장에서 형태소 분석을 마친 단어를 저장
```

수집한 데이터를 Nltk 파이썬 모듈로 자연어 처리

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust\(Nltk\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust(Nltk).ipynb) In [6] 참고

```
[('폐암', 0.7378767728805542),
 ('위해', 0.7320250272750854),
 ('발생', 0.7289312481880188),
 ('소송', 0.7218227386474609),
 ('다음', 0.721001386642456),
 ('상대로', 0.7118032574653625),
 ('승소', 0.7105224132537842),
 ('성리', 0.7090495824813843),
 ('책임', 0.7032604217529297),
 ('노력', 0.7027101516723633)]

[(('다가', 0.5108148455619812),
 ('사', 0.5060806274414062),
 ('지마켓', 0.5006664991378784),
 ('스토어', 0.4950808584690094),
 ('쿠팡', 0.4936814308166504),
 ('21', 0.47552651166915894),
 ('팔고', 0.4746638834476471),
 ('알리바바', 0.4734190106391907),
 ('시장', 0.45989158749580383),
 ('무엇', 0.45259803533554077)]
```

Gensim 파이썬 모듈을 사용하여 [연관자료](#) 수집

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust\(Nltk\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust(Nltk).ipynb) [20] 참고

핵심코드

```
import gensim
from gensim.models import word2vec
```

```
twitter = Okt()
results = []
lines = present_candi_text
for line in lines:
    malist = twitter.pos(line, norm=True, stem=True)
    r = []
    for word in malist:
        if not word[1] in [".Josa", "Eomi", "Punctuation"]:
            r.append(word[0])
    r1 = (" ".join(r)).strip()
    results.append(r1)
    print(r1)
# 조사 어미 구두를 제거함
```

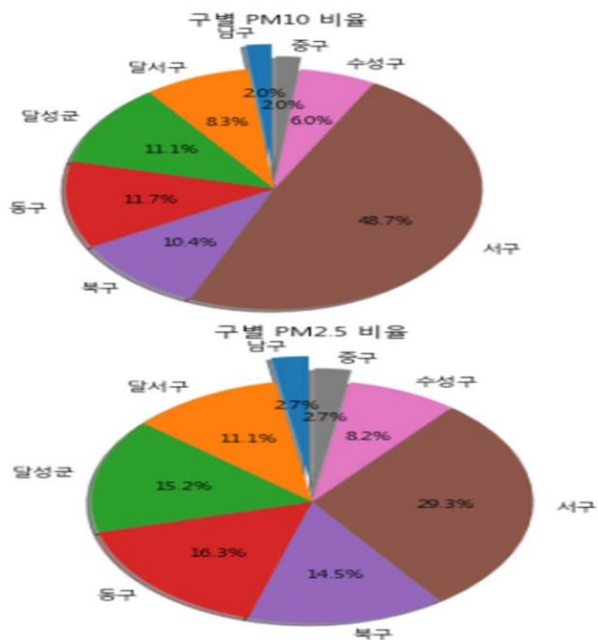
```
data_file = 'pres_girl.data'
with open(data_file, 'w', encoding='utf-8') as fp:
    fp.write("\n".join(results))
# DATA파일로 저장
```

```
data = word2vec.LineSentence(data_file)
model = word2vec.Word2Vec(data, size=200, window=10, hs=1, min_count=2, sg=1)
model.save('pres_girl.model')
# MODEL파일로 저장
```

```
model = word2vec.Word2Vec.load("pres_girl.model")
model.most_similar(positive=['국가'])
# 국가와 유사한 단어 찾을
```

```
model.most_similar(positive=['미세먼지', '마스크'])
# 미세먼지 마스크와 유사한 단어를 찾을
```


구별 비율



미세먼지를 가장 많이 배출하는 곳은 서구이고 가장 적게 배출하는 곳은 남구 중구임을 알 수 있다.

2015년 기준

출처 : 통계청

matplotlib.pyplot 파이썬 모듈을 사용하여 파이 그래프로 분석 결과 시각화

https://github.com/Bgroupbb/b-store/blob/Hotfix/findustemssion/finedust_emssion_rate_daegu_gu.ipynb In [17] 참고

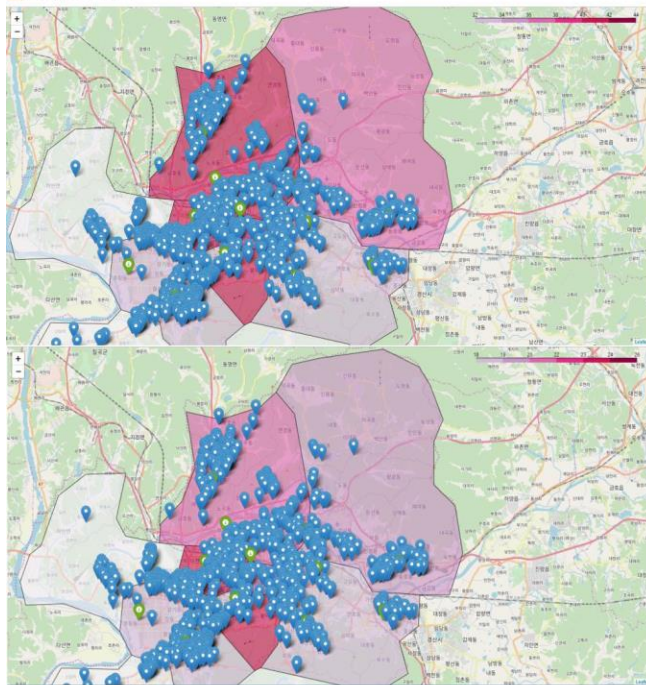
핵심코드

```
sizes2 = [26.238, 108.183, 147.926, 158.099, 140.682, 284.892, 79.259, 26.249]
explode = (0.2, 0, 0, 0, 0, 0, 0, 0.1)

plt.figure(figsize=(20, 15))
fig2, ax2 = plt.subplots()
ax2.pie(sizes2, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax2.axis('equal')

plt.title('구별 PM2.5 배출 비율')
plt.tight_layout()
plt.legend(labels, loc='center left',
           bbox_to_anchor=(1, 0.5))
plt.show()
```

어린이집 분포



출처 : 통계청

Folium 파이썬 모듈을 사용하여 Google Maps에
Json 파일을 이용한 대구 미세먼지 구별 분포를 표시하고 어린이집 위치를 Marker로 표시

<https://github.com/Bgroupbb/b-store/blob/Hotfix/kid%26air/Daegukid%26air.ipynb> In [109] 참고

핵심코드

```
mapping = folium.Map(location=[df['lat'].mean(),
df['lng'].mean()], zoom_start=11)

mapping.choropleth(geo_data=geo_data,
data = gu_data['연_평균값'],
columns = [gu_data.index,
'연_평균값'],
fill_color = 'PuRd',
key_on = 'feature.id')

for n in df.index:
    if df['도로명주소'][n] != 'Multiple':
        folium.Marker([df['lat'][n], df['lng'][n]],
popup=df['상호명'][n]).add_to(mapping)

for x in air.index:
    if air['측정소'][x] != 'Multiple':
        folium.Marker([air['lat'][x],
air['lng'][x]], popup=air['측정소'][x],
icon=folium.Icon(color= 'green', icon='info-
sign')).add_to(mapping)

mapping
```

■ 미세먼지농도와 관련질병환자수의 상관관계

	PM10	PM2.5	C34	I20	I50	I64	J18
PM10	1.0000000	0.8315156	-0.387220458	0.20192382	0.3975649	0.29913714	0.82596620
PM2.5	0.8315156	1.0000000	-0.302565221	0.27783479	0.2957580	0.31671769	0.71479554
C34	-0.3872205	-0.3025652	1.000000000	0.53847864	0.5239977	0.34968087	-0.08172730
I20	0.2019238	0.2778348	0.538478641	1.000000000	0.9060248	0.84382789	0.08399057
I50	0.3975649	0.2957580	0.523997696	0.90602484	1.000000000	0.83164942	0.32191412
I64	0.2991371	0.3167177	0.349680875	0.84382789	0.8316494	1.000000000	0.05281766
J18	0.8259662	0.7147955	-0.081727302	0.08399057	0.3219141	0.05281766	1.000000000
J44	0.7916848	0.7347171	-0.007118199	0.61951078	0.7522035	0.65015171	0.65359746
J45	0.8240417	0.7834161	-0.128731840	0.12967552	0.2927814	0.07010298	0.96087105
	J44	J45					
PM10	0.791684843	0.82404169					
PM2.5	0.734717149	0.78341608					
C34	-0.007118199	-0.12873184					
I20	0.619510783	0.12967552					
I50	0.752203480	0.29278144					
I64	0.650151714	0.07010298					
J18	0.653597458	0.96087105					
J44	1.000000000	0.66214793					
J45	0.662147935	1.000000000					

출처 : 통계청 중앙응급의료센터

cor R 함수를 사용하여 미세먼지 농도와 관련질병환자수의 상관관계를 시각화

https://github.com/Bgroupbb/b-store/blob/Hotfix/ill%26dust_cor/illdust.R Line 1 ~ 10 참고

핵심코드

```

dat <-
read.csv('C:/Users/USER/Desktop/프로젝트/csv/R 그래프용/ill.csv')
cor(dat)
library(ggplot2)
ggplot(dat, aes(J18, PM10))+ geom_point() + geom_smooth(method =
"lm")
ggplot(dat, aes(J18, PM2.5))+ geom_point() + geom_smooth(method
= "lm")
ggplot(dat, aes(J44, PM10))+ geom_point() + geom_smooth(method =
"lm")
ggplot(dat, aes(J44, PM2.5))+ geom_point() + geom_smooth(method
= "lm")
ggplot(dat, aes(J45, PM10))+ geom_point() + geom_smooth(method =
"lm")
ggplot(dat, aes(J45, PM2.5))+ geom_point() + geom_smooth(method
= "lm")
corrplot(dat)

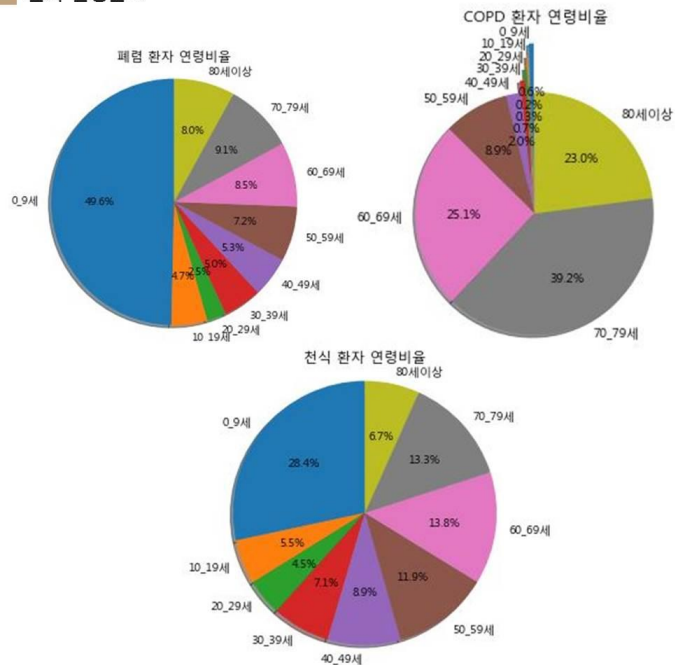
```

J18 폐렴
J44 만성 폐색성 폐질환
J45 천식

0.1이상 무시 선형관계
0.3이상 약한 선형관계
0.7이상 뚜렷한 선형관계
1.0이상 강한 선형관계

2017년 기준

환자 연령분포



미세먼지 관련 질병별
취약계층 분석 결과
폐렴은 어린이
COPD는 노인
천식은 노약자 모두
발병률이 높음을 확인

2017년 기준

출처 : 보건 의료 빅데이터 개방 시스템

matplotlib.pyplot 파이썬 모듈을 사용하여 파이 그래프로 분석 결과 시각화

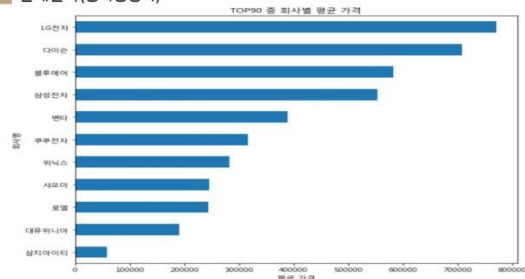
https://github.com/Bgroupbb/b-store/blob/Hotfix/all_ills.ipynb In [8] 참고

핵심코드

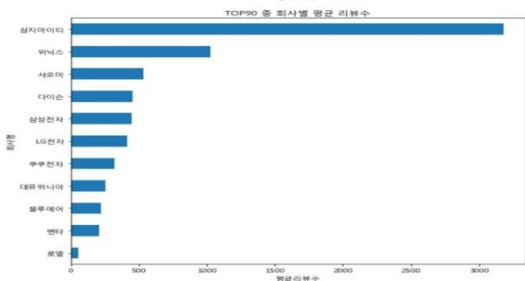
```
labels = ['0_9 세', '10_19 세',
          '20_29 세', '30_39 세', '40_49 세',
          '50_59 세', '60_69 세', '70_79 세',
          '80 세이상']
sizes = [3863, 1169, 1855, 4048, 12084,
         53292, 149978, 234384, 137240]
explode = (0.4, 0.3, 0.2, 0.1, 0, 0, 0,
           0, 0)
fig, ax = plt.subplots()
ax.pie(sizes, explode=explode,
       labels=labels, autopct='%1.1f%%',
       shadow=True, startangle=90)
ax.axis('equal')

plt.title ('COPD 환자 연령비율')
plt.tight_layout()
plt.show()
```

판매전락(공기청정기)



삼지아이티가 가격도 싸고
리뷰수(구매수)도
압도적으로 많다.



출처 : 다나와

핵심코드

```
pv2 = pd.pivot_table(df, columns=["company"],
values=['reviews'], aggfunc=[np.mean], fill_value=0)+
pv2+
pv2.to_excel(r'C:\Users\USER\Desktop\프로젝트\csv\크롤
링파일\공기청정기\companyreviews.xlsx', encoding='UTF-
8')+
cr
=
pd.read_excel(r'C:\Users\USER\Desktop\프로젝트\csv\크롤
링파일\공기청정기\companyreviews2.xlsx', encoding='UTF-
8')+
cr.set_index('company', inplace=True)+
cr.head()+
cr['reviews'].sort_values().plot(kind='barh',
figsize=(9, 8))+
plt.title('TOP90 중 회사별 평균 리뷰수')+
plt.ylabel('회사명')+
plt.xlabel('평균리뷰수')+
plt.show()+
```

matplotlib.pyplot 파이썬 모듈을 사용하여 **바 그래프**로 분석 결과 시각화

https://github.com/Bgroupbb/b-store/blob/Hotfix/shoppingmall_goods.ipynb In [19] 참고

■ 결론



홈페이지 <https://bgroub.net>

실시간정보를 홈페이지를 통해서 얻을 수 있다.
다양한 정보 및 인터넷 쇼핑 구매가이드 미세먼지관련 어플들을 소개를 하고있다.

Ubuntu 17.10, Apache 2.9.1, AWS EC2를 사용하여 서버를 만들고
Sublime Text3로 HTML5 CSS3 JavaScript를 코딩하여 만든 홈페이지

<https://github.com/Bgroupbb/b-store/blob/master/html/index.html> Line 2 ~ 25 참고

핵심코드

```
<head>+
  <meta charset="UTF-8">+
  <meta name="description" content="">+
  <meta http-equiv="X-UA-Compatible" content="IE=edge">+
  <meta name="viewport" content="width=device-width, initial-scale=1,
  shrink-to-fit=no">+
  <!-- Title -->+
  <title>B 조 단체구매</title>+
+
  <link rel="icon" type="image/png" sizes="32x32" href="/img/core-
  img/favicon.ico" />+
  <!-- Style CSS -->+
  <link rel="stylesheet" href="style.css">+
  </link>
href="https://fonts.googleapis.com/css?family=Nanum+Pen+Script&subset=
Korean" rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Ubuntu"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Black+Han+Sans"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Oswald"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Oute+Font"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Nanum+Pen+Script"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Roboto+Slab+KR"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Leon+Bung"
  rel="stylesheet">+
  <link href="https://fonts.googleapis.com/css?family=Jua"
  rel="stylesheet">+
+
</head>+
```

- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획

미세먼지관련상품판매 기획안 결론

1. 판매지역 : 대구광역시 서구, 북구, 남구, 중구

2. 타겟 연령층 : 30~50대

3. 홍보 및 판매전략 집중이 필요한 주요 회사상품 :

마스크 : 3M, 유한킴벌리, 파인텍, 웰킵스, 장정산업

가성비 - 장정산업 애니가드 방진마스크 VC201G

최고 인기상품 - 파인텍 네퓨어 황사방역용 마스크 KF94 대형

공기청정기 : 삼성, LG, 위닉스, 삼지아이티

가성비 - 삼지아이티 PISnet 피스넷 에어, 퓨어 360

최고 인기상품 - 삼성전자 블루스카이 AX60N5580WDD

홈페이지를 통해 '다나와' 사이트를 연결해서

자세한 인기상품을 볼 수 있음

구매 가이드라인 세일정보등 도움되는 정보들을 제공



<http://bgroup.ml/>



<https://github.com/Bgroupbb/b-store/blob/master/Fine%20Dust.pdf>

이 프로젝트로 미세먼지와 관련질병의 상관관계를 분석하려했으나 R을 잘 다루지 못 해 통계적인 내용을 구하기 힘들었습니다.

그리고 머신러닝을 하지 못 하여 미세먼지 향후 예측을 못 했습니다.

그래서 R 과 머신러닝을 개인적으로 공부하여 수학과 통계적인 부분에서 설득력을 가질 수 있도록 하겠습니다.

1. 판매지역 우선순위 설정 할 수 있다.

- 미세먼지 분포도가 심한 지역
- 취약계층 분포가 많은 지역

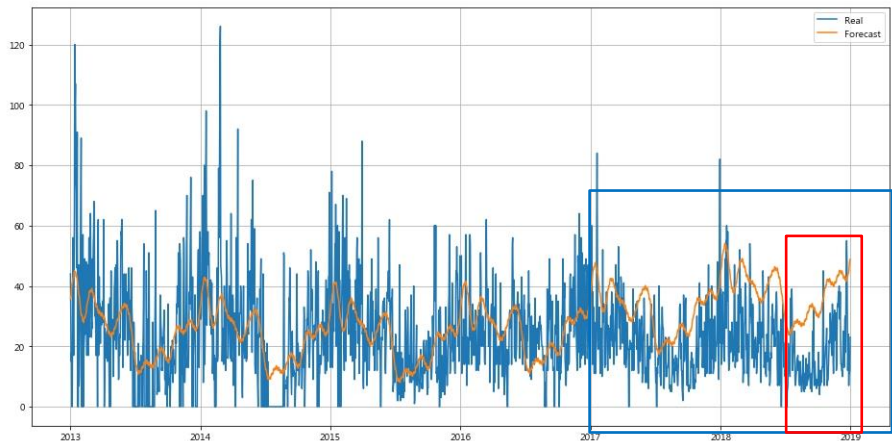
2. 판매대상 우선순위 설정 할 수 있다.

- 취약계층 분포가 많은 지역

3. 판매상품 우선순위 설정 할 수 있다.

- 인기상품
- 가성비상품

- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획



Prophet 파이썬 모듈을 사용하여 730일(2년) 치의 자료를 예측

[https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust\(Prophet\).ipynb](https://github.com/pty902/Resume/blob/master/Fine%20Dust%20Project/Fine%20Dust(Prophet).ipynb) In [30~34] 참고

핵심코드

```
m = Prophet(daily_seasonality=True)
m.fit(df2)

future = m.make_future_dataframe(periods=730)
future.tail()

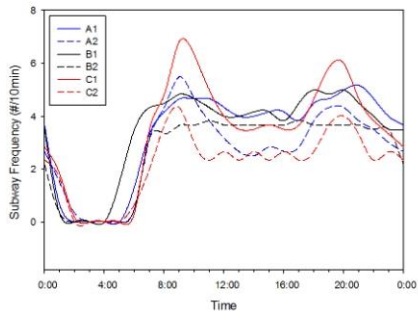
forecast = m.predict(future) forecast[['ds', 'yhat', 'yhat_lower',
'yhat_upper']].tail()

plt.figure(figsize=(16,8))
plt.plot(df3['ds'], df3['y'], label='Real')
plt.plot(forecast['ds'], forecast['yhat'], label='Forecast')
plt.grid()
plt.legend()
plt.show()
```

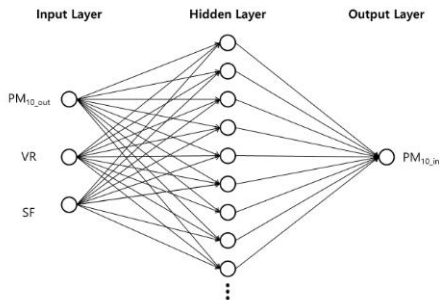
2017~2018년 실제값과
2017~2019년 예측값(파란박스) 을 같이 두고 그래프화

2018년 7~10월, 12월값(빨간박스)을
제외하곤 대체적으로 비슷한 모습
2018년 7월 고기압 2018년 12월 러시아발 고기압때문에
중국으로부터의 미세먼지 유입이 없었기 때문이었음

- 프로젝트 서론
- 프로젝트 본론
- 프로젝트 결론
- 현재 연구 진행상황
- 추후 연구 계획

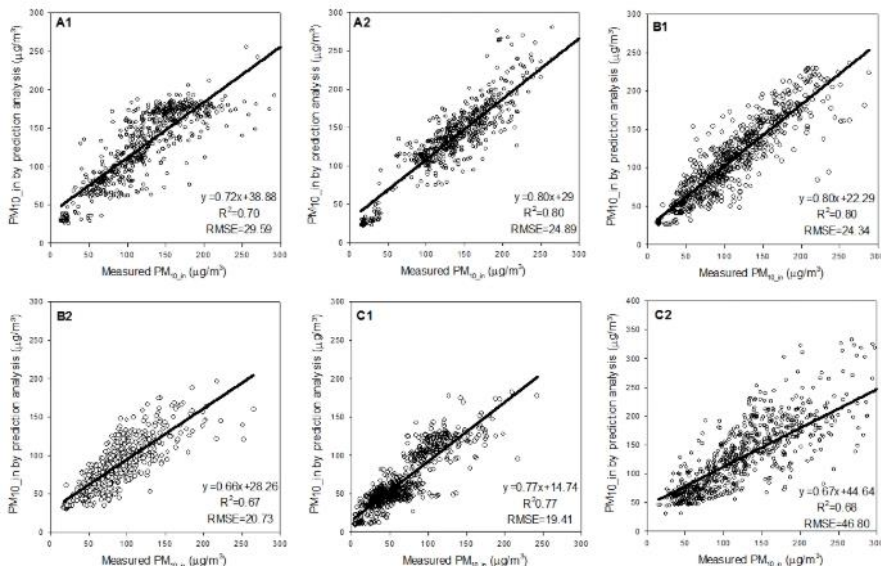


[그림 3] 지하철 미세먼지 예측 ANN모델 구조와 입력변수 데이터 패턴*7



머신러닝을 공부를 하여 모델을 훈련시켜보겠습니다.
그래서 향후 미세먼지 농도 예측을 다양하게 해보고
최적의 모델을 찾아서 실용성을 높여보겠습니다.

관련자료 출처 : 카카오시리포트 권순박
<https://brunch.co.kr/@kakao-it/219>



[그림 4] 서울시 6개 지하철 역사 내 ANN모델을 활용한 미세먼지 예측 정확도*8

Thank You!

