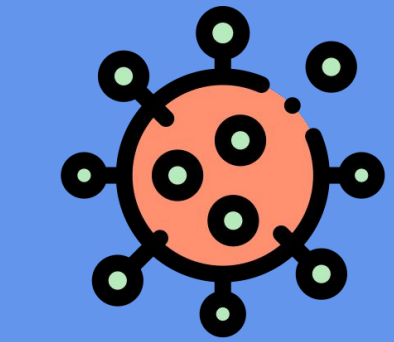
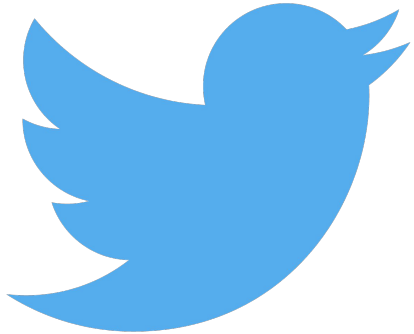




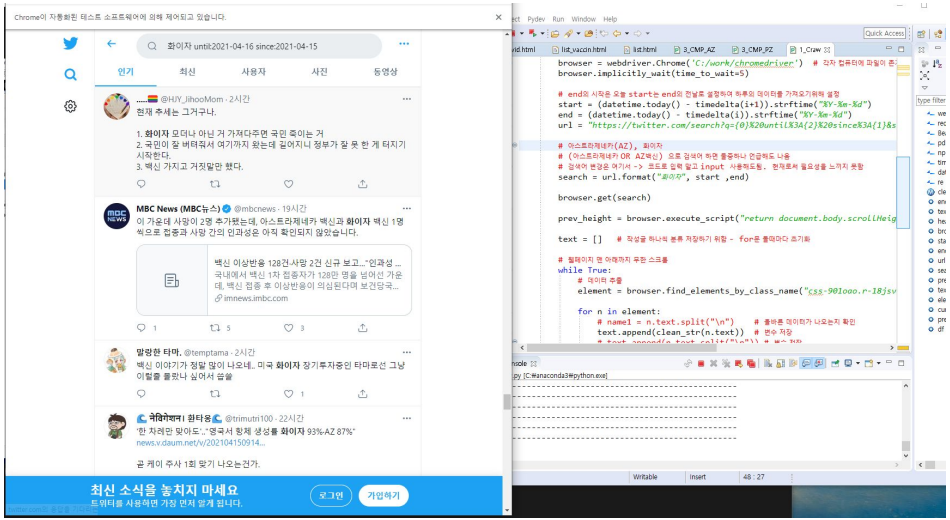
트위터 크롤링을 통한 백신별 감성분석 & FAST COVID-19 LIVE

데이터 수집

Seleniun 라이브러리릉 사용해 트위터 페이지를 크롤링하여 백신별(영어/한글) 따로 수집

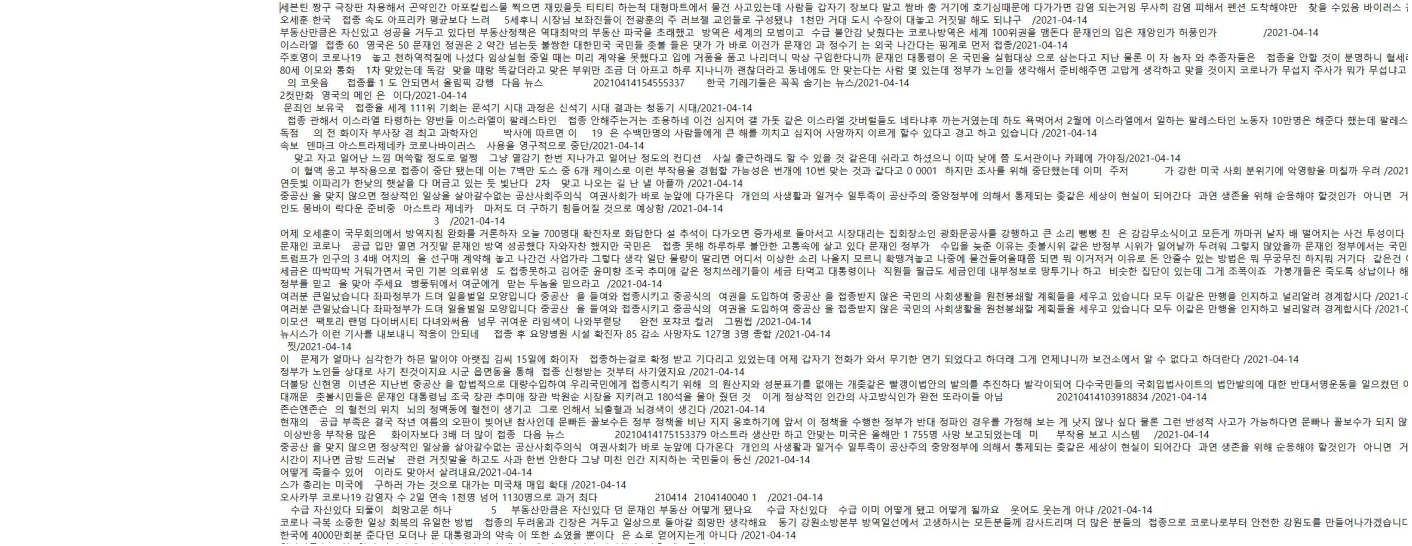
수집한 데이터를 가공해 감성분석

데이터 수집



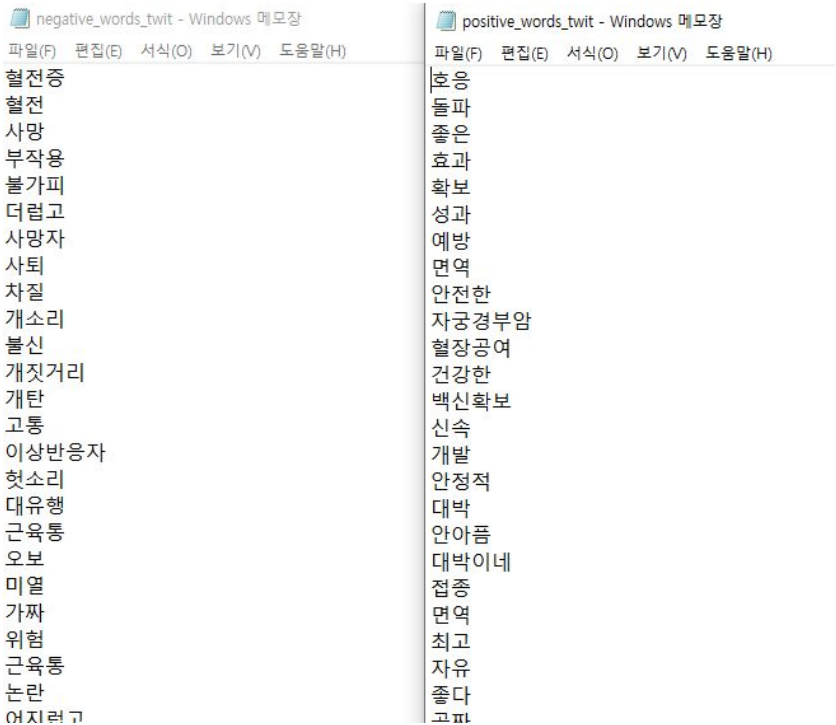
Selenium을 이용한 백신별 Twit수집

데이터 가공



수집한 데이터를 **txt**파일 형식으로 **1**차 가공

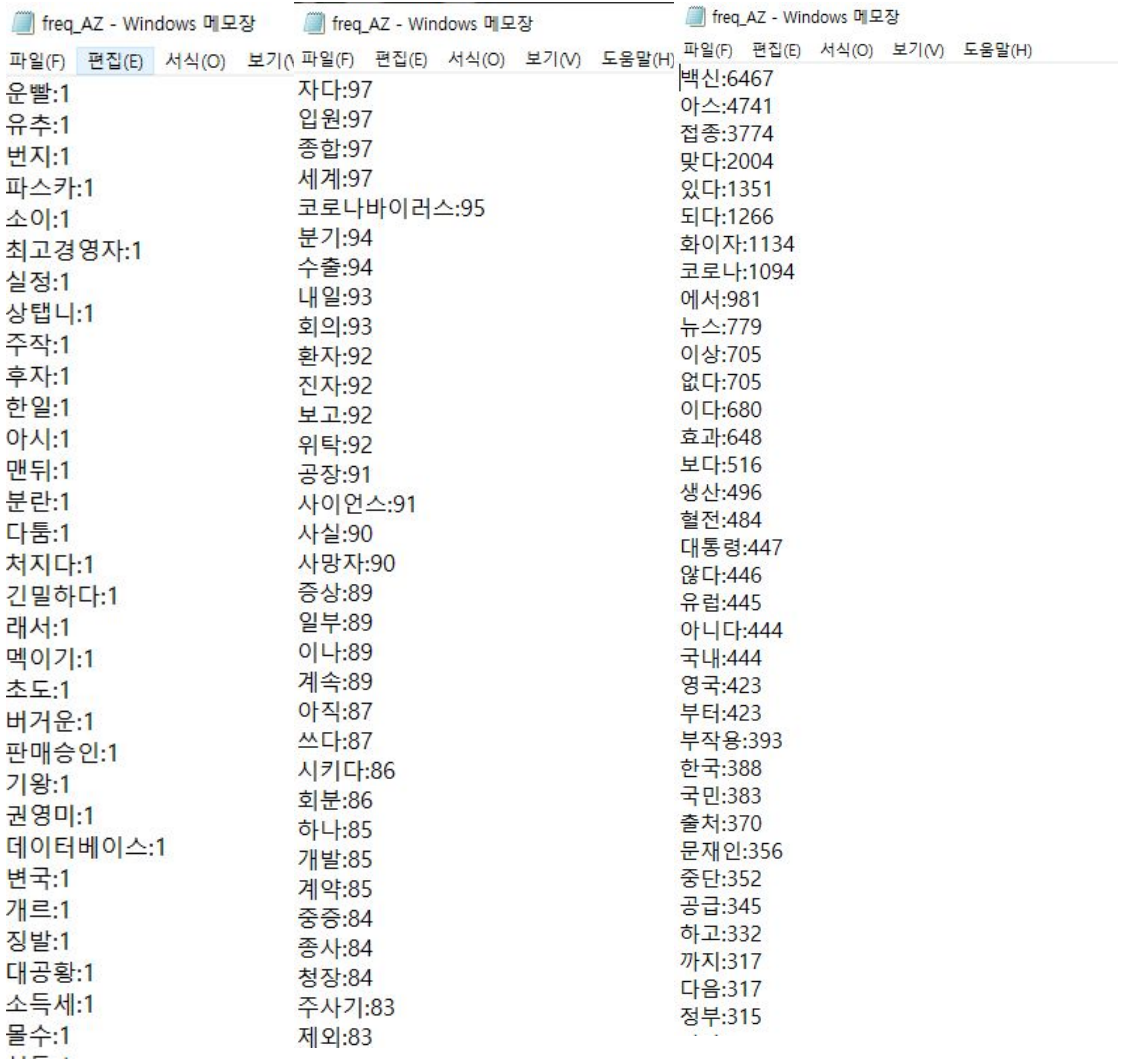
데이터 가공



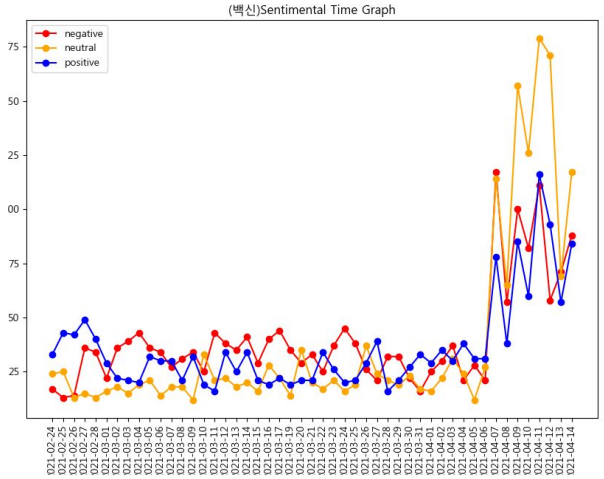
긍정, 부정, 중립 감성별 키워드로 데이터 재가공

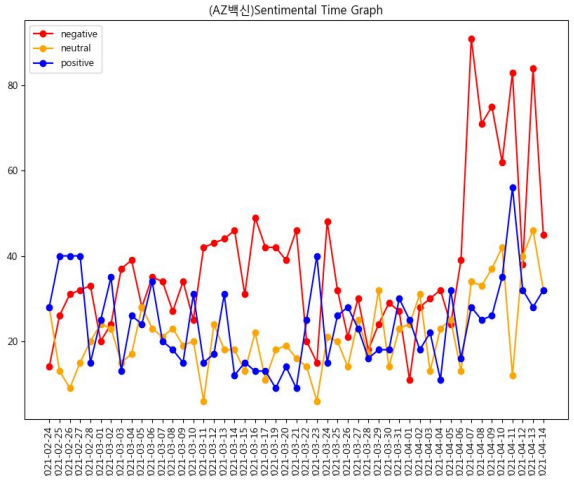
데이터 가공

키워드별 빈도수 분석



분석 결과

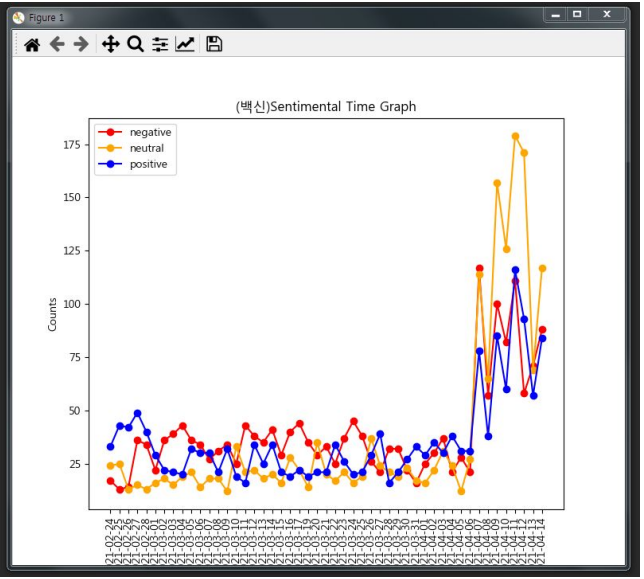




시간별 부정, 중립 긍정 빈도수 변화(한글 트윗)

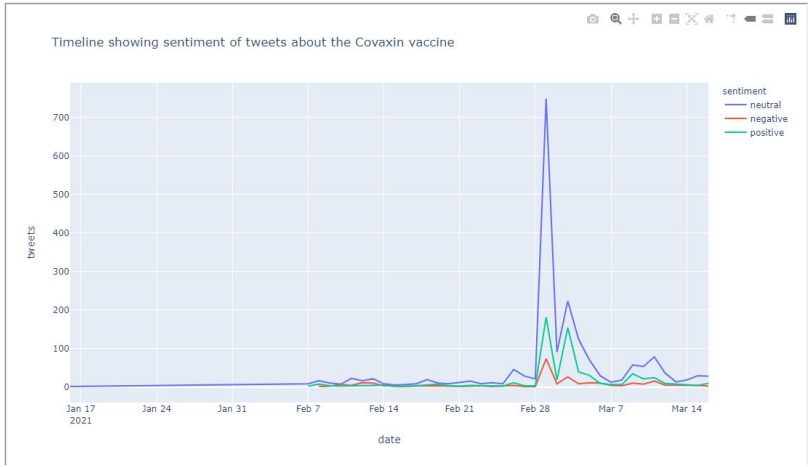
분석 결과

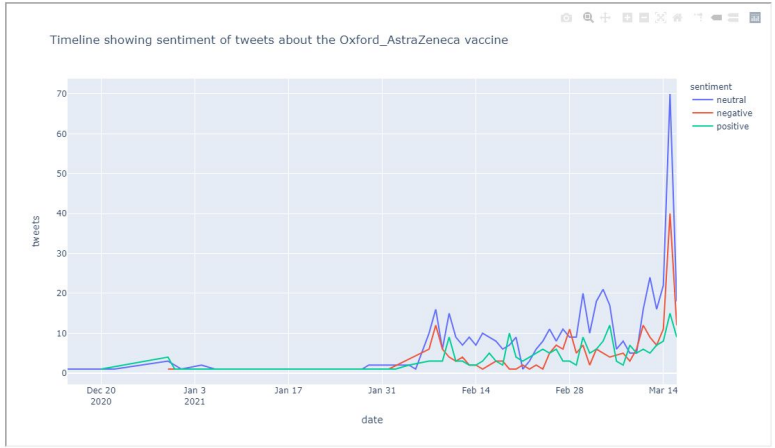
```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

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


df_Vac = pd.read_excel('./data/result(백신).xlsx')
df_Vac = df_Vac.loc[:, ['date', 'label']]

# '백신' 에 관련된 timegraph
#################################################################
df_Vac2 = df_Vac['label'].groupby([df_Vac['date'], df_Vac['label']]).count()
print(type(df_Vac2))
df_Vac2 = df_Vac2.reset_index(name='count')
print(df_Vac2)

neg_df = df_Vac2.query('label=="-1"')
days = neg_df['date']
# print(neg_df)
neu_df = df_Vac2.query('label=="0"')
pos_df = df_Vac2.query('label=="1"')

fig = plt.figure(figsize=(10,7)) ## 캔버스 생성
fig.set_facecolor('white') ## 캔버스 색상 설정
ax = fig.add_subplot() ## 그림 배대(프레임) 생성

ax.plot(days,neg_df['count'],marker='o',label='negative', c='r') ## 선그래프 생성
ax.plot(days,neu_df['count'],marker='o',label='neutral', c='orange')
ax.plot(days,pos_df['count'],marker='o',label='positive', c='b')
plt.title('(백신)Sentimental Time Graph')
plt.xlabel('Date')
plt.ylabel('Counts')
plt.legend()
plt.xticks(rotation=90)
plt.savefig('(백신)Sentimental Time Graph')
plt.show()
```



시간별 부정, 중립 긍정 빈도수 변화(한글 트윗)

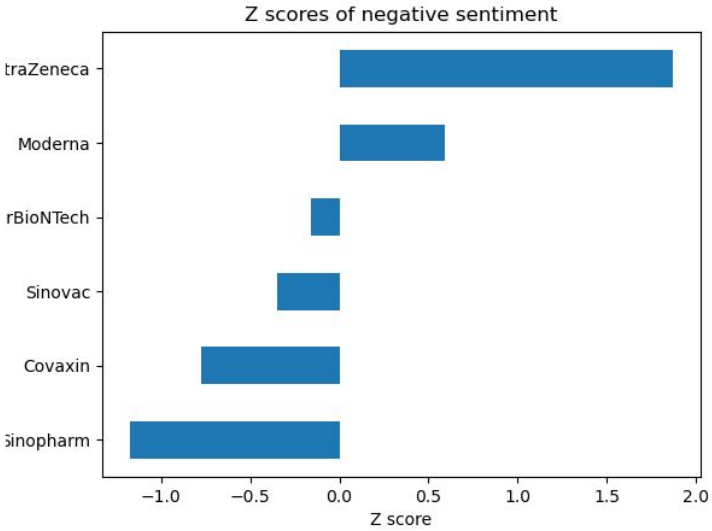
분석 결과





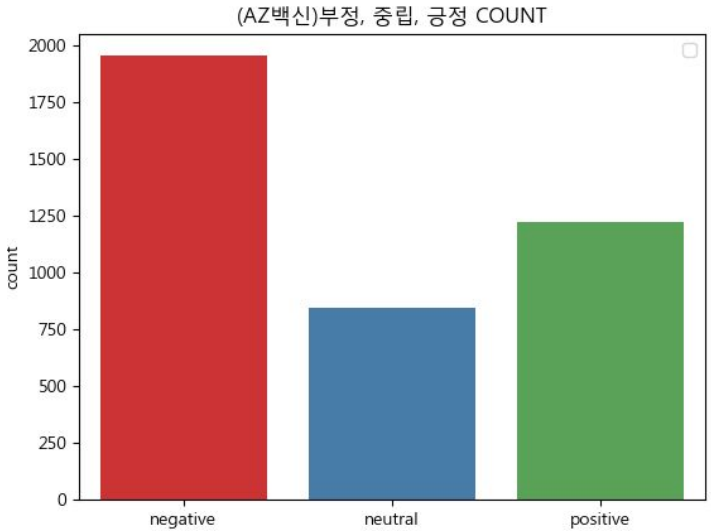
시간별 부정, 중립 긍정 빈도수 변화 (영어 트윗)

분석 결과



백신별 부정적인 지수 비교

분석 결과



부정 / 중립 / 긍정 지수비교

분석 결과

```
# 부정 단어
negative_data = pnn_data['label'] == -1
negative_data = pnn_data[negative_data]
negative_data = negative_data.drop('label', axis=1)
print(negative_data)
# 중립 단어
neutral_data = pnn_data['label'] == 0
neutral_data = pnn_data[neutral_data]
neutral_data = neutral_data.drop('label', axis=1)
print(neutral_data)
# 긍정 단어
positive_data = pnn_data['label'] == 1
positive_data = pnn_data[positive_data]
positive_data = positive_data.drop('label', axis=1)
print(positive_data)
'''

##########################################################################
# 트위터 (빈도수기준) 클라우딩
twit_coloring = np.array(Image.open('twit.png'))
if search_text == '백신':
    twit_coloring = np.array(Image.open('v_image.png'))
from wordcloud import ImageColorGenerator
image_colors = ImageColorGenerator(twit_coloring)

covid_wc = WordCloud(font_path = font_location, background_color='white',width=1000, height=500, randc

fig, ax = plt.subplots(figsize=(12,6))
plt.imshow(covid_wc, interpolation='bilinear')
plt.axis('off')
plt.title('(' + search_text + ')wordcloud')
plt.savefig('wordcloud(' + search_text + ').png')
plt.show()

'''
# (로컬 자체에서 단어를 뽑아 클라우딩)
```

백신별 워드 클라우드

개선 사항

모델 평가(AZ)

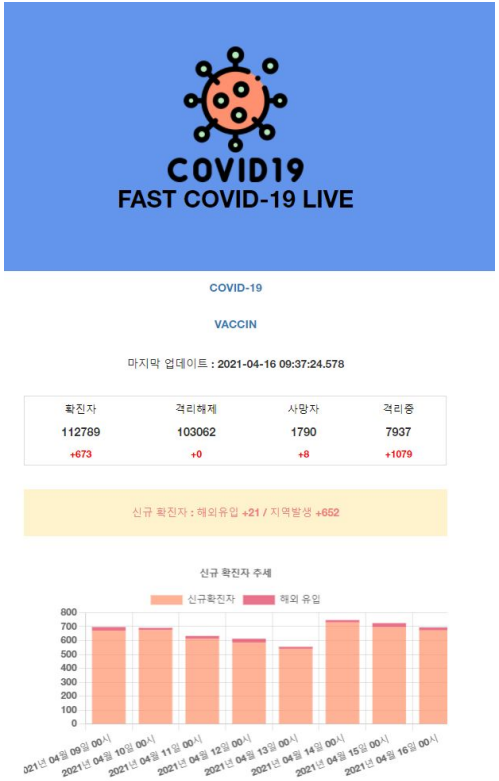
정확도 : 0.9529120326042175
손실값 : 0.3152697682380676

모델 평가(PZ)

정확도 : 0.9603710174560547
손실값 : 0.2377401441335678

FAST COVID-19 LIVE

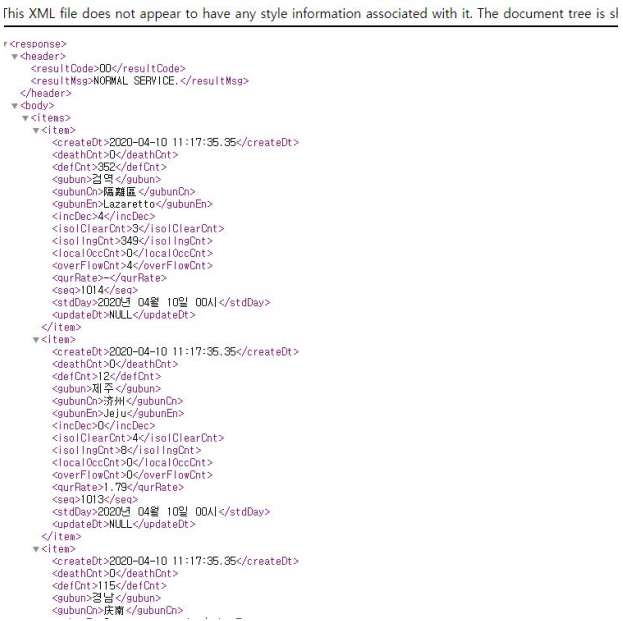
실시간 코로나 현황을 볼 수 있는 웹을 개발해
기존해 분석한 백신 데이터를 사이트에 접목시키기로함.

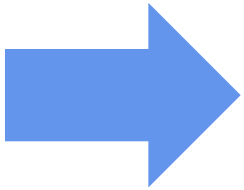


FAST COVID-19 LIVE

공공 데이터API를 활용해 국가별/지역별/남녀별/세대별
감염,예방 현황을 시각화하기로함.

FAST COVID-19 LIVE





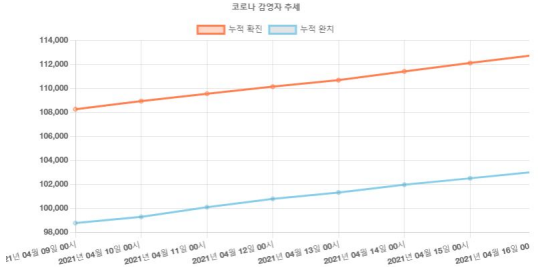
```python
def get_corona_data(startCreateDt, endCreateDt):
    params = {
        'serviceKey' : serviceKey_2,
        'pageNo' : '1',
        'numOfRows' : 10,
        'startCreateDt': startCreateDt,
        'endCreateDt': endCreateDt,
    }

    res = requests.get(url=url , params=params)
    # print(res.url)
    # print(res.text)

    # xml -> dict
    dict_data = xmltodict.parse(res.text)
    # print(dict_data)

    # dict -> json
    json_data = json.dumps(dict_data)
    # print(json_data,type(json_data))

    # json -> dict
    dict_data = json.loads(json_data)
    # print(dict_data, type(dict_data))
    # pprint(dict_data['response']['body']['items']['item'])

    # total Cnt Check
    totalCount = dict_data['response']['body']['totalCount']
    if totalCount == "0":
        return False

    # 지역정보를 담은 리스트 저장
    area_data = dict_data['response']['body']['items']['item']
    area_data.reverse()
    # pprint(area_data)
    for a in area_data:
        print(a)
    return area_data
```

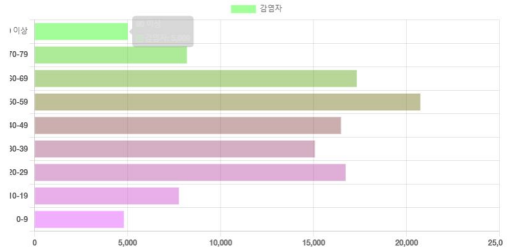
인증키를 받아 태그값을 추적해 원하는 데이터 추출 후
활용하기 쉬운 형태로 가공.

FAST COVID-19 LIVE

Chart.js

Chart.js를 통해 웹에서 데이터 시각화





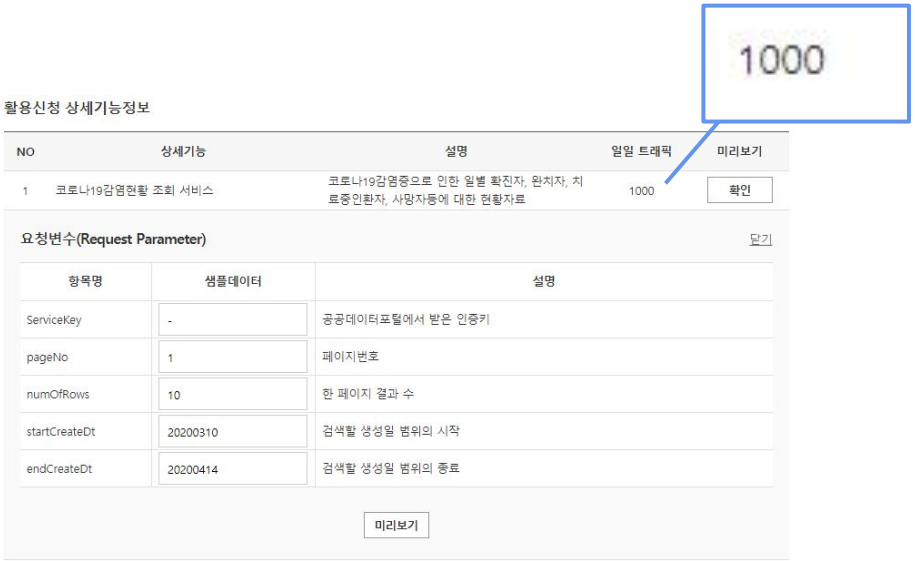
FAST COVID-19 LIVE

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<response>
  ▼<header>
     <resultCode>99</resultCode>
     <resultMsg>LIMITED NUMBER OF SERVICE REQUESTS EXCEEDS ERROR.</resultMsg>
   </header>
  </response>
```

사이트의 잦은 열람 시 API 접근오류가
걸리는 문제 발생

FAST COVID-19 LIVE



일일 트래픽 1000 제한이 원인.

FAST COVID-19 LIVE

```
url = "http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson"
url_gender = "http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19GenAgeCaseInfJson"
serviceKey_1 = '2PLDIVNdhJ3pWgmyk2qXL2LekLrwfv5r8z2tq6[redacted]'
serviceKey_2 = 'h92s7QADkL1WM1LYJeYFzOx4Fmh+CV5s/Vhl0j[redacted]'
serviceKey_gender1= '2PLDIVNdhJ3pWgmyk2qXL2LekLrwfv5r8[redacted]nqtEA%3D%3D'
serviceKey_gender2='h92s7QADkL1WM1LYJeYFzOx4Fmh+CV5s/V[redacted]Kw=='
url_vaccin = "https://nip.kdca.go.kr/irgd/cov19stats.do?list=all"
```

팀원들의 키값을 할당 받아 키오류시 다른 키를 받아 오도록 개선.

THANK YOU

THANK YOU

THANK YOU

THANK YOU

THANK YOU