주가와 검색어 간의 상관관계

수원대학교 데이터 과학부
19016002 고가연

목차

- 발표자 소개
- 프로젝트 주제 선정 이유
- 전체적인 흐름과 과정
- 결과 (시각화)
- 프로젝트 소감

발표자 소개

- 이름
  - 고가연
- 관심 분야
  - 돈
  - 삶의 질 향상
- 취미
  - 손으로 하는 것들 (공예, 베이킹 등)

주제 선정 이유

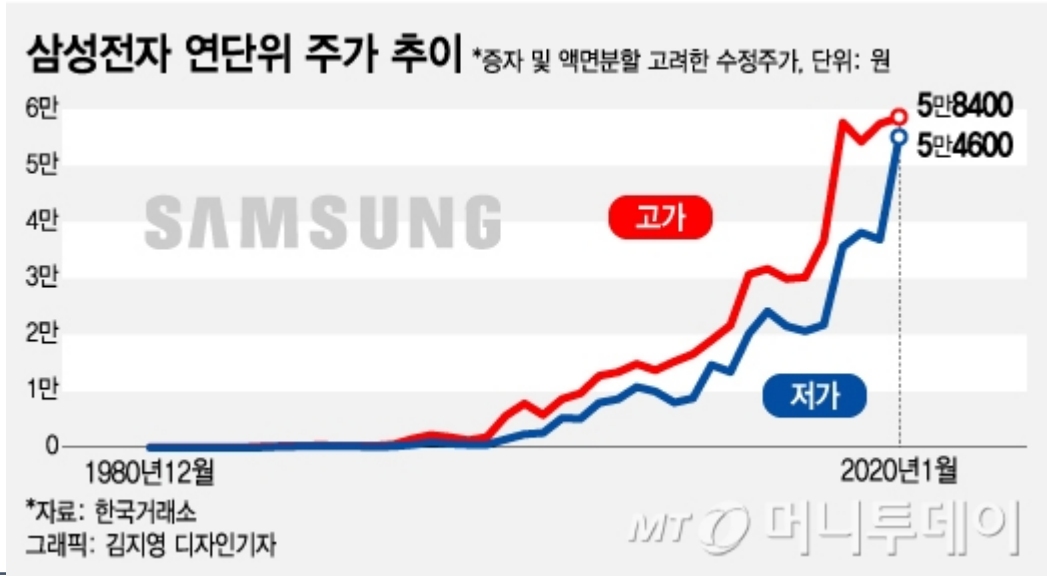
주가와 검색량의 상관관계 https://yobro.tistory.com/8?category=793224

전체적인 흐름.

- 기업 선정.
- Selenium 패키지로 기사 크롤링.
- 형태소 분석기 중 twitter(Okt)를 이용해 기사에서 키워드 추출.
- 키워드 중 가장 많이 나온 상위 3개 추출.
- 네이버 데이터 랩의 api를 통해 키워드 각각에 대한 검색량 추출.
- 야후 파이낸스 패키지로 주가 데이터 크롤링.
- 각 키워드 검색량과 주가 사이의 상관관계 분석.

과정 (0)
– 기업, 기간 선정

- 기업 – 주식하는 한국인이라면 누구나 한 주 쯤은 품고 있을 삼성전자.
- 기간 – 주가는 뉴스에 반응할 것이라는 가정을 잡고 시작했으므로 새소식의 키워드에 주가를 바로 매치할 수 있도록 짧게(일주일) 잡음.



과정 (1)
– 기사에서 키워드 추출

- 네이버에서 “삼성전자” 검색
- -> 최근 일주일 동안의 기사 100개 크롤링.
- 크롤링한 기사 엑셀파일로 저장.
- 엑셀 파일의 text 열을 하나의 문자열로 묶은 후,
- 형태소 분석기 Okt를 이용해 키워드 추출.

왜 Twitter(Okt)였는지?

- 속도를 기준으로 분석기 선택 시:
  Mecab >>>>>>>> Komoran >> Hannanum >> Okt >>>> Kkma
- 오타가 많은 문장을 분석 시:
  Komoran > Hannanum(큼직하게) ≒ Kkma(잘게) > Okt > Mecab
- 띄어쓰기가 제대로 되어있지 않은 문장 분석 시:
  Komoran = Kkma(매우 잘게) ≒ Mecab(잘게) ≒ Okt(큼직하게) > Hannanum

- Kkma(꼬꼬마): 속도면에서 뒤짐.
- Mecab: 분석 속도는 분석기 중에 가장 빠르나, 분석력이 딸림.

- Komoran(코모란): 오타가 있는 문장, 띄어쓰기가 제대로 되어있지 않은 문장에서 분석력이 드러남. (다른 분석기들에 비해 뛰어남.)
- Hannanum(한나눔): 분석력은 Kkma와 비슷했으나 속도면에서 Kkma보다 우수

과정 (1) 코드

페이지 별 상세 내용

- 1. 언론사별 본문 위치 태그 파싱 함수
- 2-1. 브라우저를 켜고 검색 키워드 입력
- 2-2. 기간, 언론사 선택 및 confirm
  (검색 옵션 설정)
- 3. 뉴스 크롤링
- 4-1. 데이터 전처리
- 4-2. 키워드 추출, 불용어 처리
- 5. 워드 클라우드 생성

```
######### 1. 언론사별 본문 위치 태그 파싱 함수 ########
def crawling_main_text(url):
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
    }
    req = requests.get(url, headers = headers)
    req.encoding = None
    soup = BeautifulSoup(req.text, 'html.parser')
    # 연합뉴스
    if ('yna' in url) | ('app.yonhapnews' in url):
        main_article = soup.find('div', {'class':'story-news'})# article'})
        if main_article == None:
            main_article = soup.find('div', {'class' : 'article-txt'})
        text = main_article.text
    # MBC
    elif '//imnews.imbc' in url:
        text = soup.find('div', {'itemprop' : 'articleBody'}).text
    # SBS
    elif 'news.sbs' in url:
        text = soup.find('div', {'itemprop' : 'articleBody'}).text
    # KBS
    elif 'news.kbs' in url:
        text = soup.find('div', {'id' : 'cont_newstext'}).text
    # JTBC
    elif 'news.jtbc' in url:
        text = soup.find('div', {'class' : 'article_content'}).text
    # 그 외
    else:
        text = ". "  #None으로 에러 해결이 안됨.
    return text.replace('₩n','').replace('₩r','').replace('<br>','').replace('₩t',' ')
press_list = ['연합뉴스','KBS','MBC','SBS','JTBC']
print('검색할 언론사 : {} | {}개 ₩n'.format(press_list, len(press_list)))
```

```
######### 2-1. 브라우저를 켜고 검색 키워드 입력 ##########

query = input('검색할 키워드  : ')
news_num = int(input('수집 뉴스의 수(숫자만 입력) : '))
browser = webdriver.Chrome('chromedriver')

news_url = 'https://search.naver.com/search.naver?where=news&query={}'.format(query)

browser.get(news_url)

time.sleep(sleep_sec)



######### 2-2. 검색 옵션 설정 ############################

search_opt_box = browser.find_element_by_xpath('//*[@id="search_option_button"]')

# 검색 옵션 버튼
search_opt_box.click()

time.sleep(0.02)

# 기간 설정 (최근 1주일)
browser.find_element_by_xpath('//*[@id="snb"]/div/ul/li[2]/a').click()

browser.find_element_by_xpath('//*[@id="snb"]/div/ul/li[2]/div/div[1]/ul[1]/li[3]/a').click()
```

```
# 언론사 버튼 class = "m _tab_option_"인 a 태그
press_box = browser.find_element_by_xpath('//*[@id="snb"]/div/ul/li[5]/a')

press_box.click()

# class = "press_category"인 div태그의 바로 부모 (id = 'order_cat' 인 div태그.)
press_category_box = browser.find_element_by_xpath('//*[@id="order_cat"]')

# browser.find_element_by_xpath('//*[@id="ca_p1"]').click() #첫판 일간지는 선택 안해줘도 됨.

browser.find_element_by_xpath('//*[@id="order_cat"]/div[1]/div/a[2]').click() #방송통신

browser.find_element_by_xpath('//*[@id="ca_1437"]').click() # jtbc
browser.find_element_by_xpath('//*[@id="ca_1056"]').click() # kbs
browser.find_element_by_xpath('//*[@id="ca_1055"]').click() # sbs
browser.find_element_by_xpath('//*[@id="ca_1214"]').click() # mbc
browser.find_element_by_xpath('//*[@id="ca_1001"]').click() # 연합뉴스

# 확인 버튼
press_button  = browser.find_element_by_xpath('//*[@id="snb"]/div/ul/li[5]/div/span/span[1]/button')

press_button.click()
```

```
######### 3. 뉴스 크롤링 ##################################################

news_dict = {}
idx = 1
cur_page = 1

while idx <= news_num:
    table = browser.find_element_by_xpath('//*[@id="main_pack"]/section[1]/div/div[3]/ul')

    table_li_list = table.find_elements_by_xpath('.//li[contains(@id, "sp_nws")]')

    li_a_list = [t_li.find_element_by_xpath('.//div/div/a[@class="news_tit"]') for t_li in table_li_list]

    for n in li_a_list[:min(len(li_a_list), news_num-idx+1)]:
        n_url = n.get_attribute('href')

        news_dict[idx] = {'title' : n.get_attribute('title'),

                          'url' : n_url,

                          'text' : crawling_main_text(n_url)}

        idx += 1

    if idx < news_num:
        cur_page +=1

        browser.find_element_by_xpath('//*[@id="main_pack"]/div[3]/div/a[2]').click()

        time.sleep(sleep_sec)

    else:
        print('₩n브라우저를 종료합니다.₩n' + '=' * 100)

        time.sleep(0.1)

        browser.close()

        break
```

```
######### 4-1. 데이터 전처리 ##########################

news_df = DataFrame(news_dict).T

folder_path = os.getcwd()

xlsx_file_name = '네이버뉴스_본문_{}개_{}_{}.xlsx'.format(news_num, query, date)

news_df.to_excel(xlsx_file_name)

os.startfile(folder_path)

news_df


######### 4-2. 키워드 추출, 불용어 처리 #################

file_dir = input("file 경로를 입력하세요.: ")

xlsx = pd.read_excel(file_dir)

txt = xlsx.loc[:,'text']

lst = []

for i in txt:

    lst.append(i)

string = " ".join(lst) #리스트를 string으로

okt = Okt() #객체 선언
```

```
# 명사만 추출
str_nouns_lst = okt.nouns(string)

str_nouns = " ".join(str_nouns_lst)

str_nouns


# 쓸데 없는 거 불용어 처리
stopwords = set(STOPWORDS)

stoplst = ['삼성전자', '삼성', '전자']

stopwords.update(stoplst)

stoplst3_str = "바로/훌쩍"

stoplst3 = stoplst3_str.split("/")

stopwords.update(stoplst3)


# 상위 n개 키워드 선택
str_lstt = str_nouns.split(" ")

# str_lstt에서 불용어 지우고 카운트!
for word in str_lstt:
    if word in stopwords:
        while word in str_lstt:
            str_lstt.remove(word)

count = Counter(str_lstt)

count_10 = count.most_common(10) # 상위 10개

count_10[0][0] # 제일 많이 언급된 키워드 확인
```

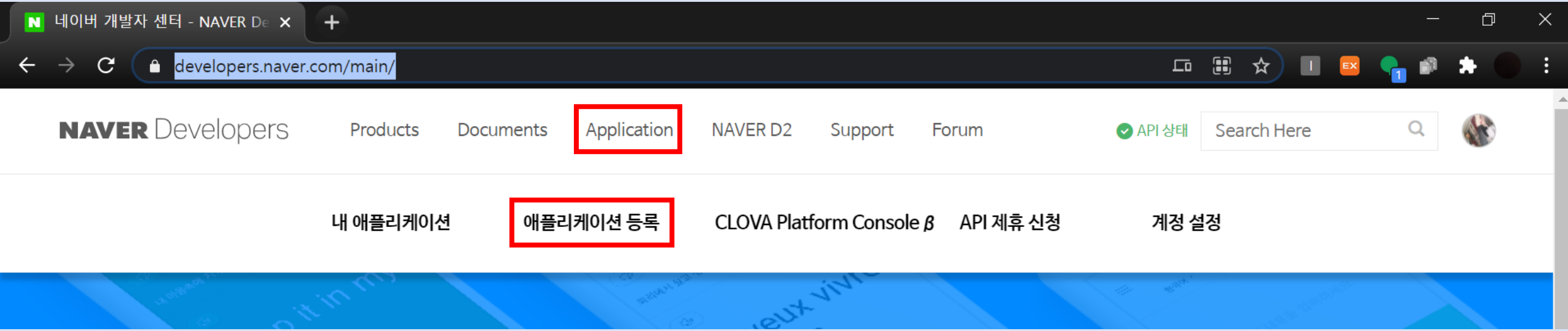
```
########## 5. 워드클라우드 생성 ##########################

date = str(datetime.now())
date = date[:date.rfind(':')].replace(' ', '_')
date = date.replace(':','시') + '분'

query = input('검색한 키워드 : ')

news_num = int(input('수집한 뉴스의 수: '))

# 위 5줄은 워드클라우드와 상관 x


mx_wrds = 1000

wc = WordCloud(font_path = 'C:/Windows/Fonts/malgun.ttf',
background_color = 'white',₩
                    width = 2000,₩
                    height = 2000,₩
                    max_words = mx_wrds,₩
                    max_font_size = 400,₩
                stopwords = stopwords)


wc.generate(str_nouns)

wc.to_file('네이버뉴스_본문_{}개_{}_{} basic,
max{}.png'.format(len(xlsx), query, date, mx_wrds))
```

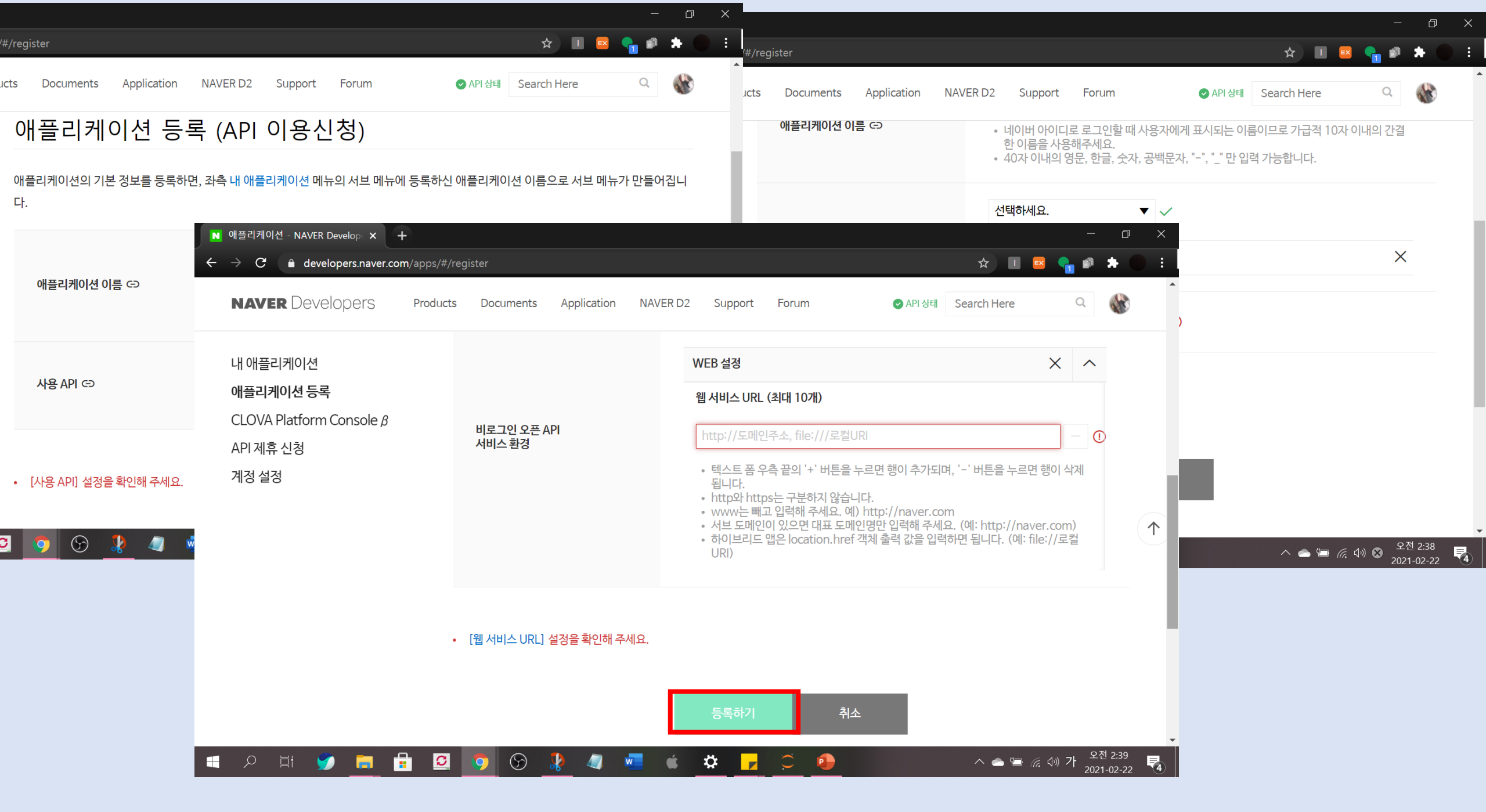
과정 (2)
– 키워드 검색량 추출

- 네이버 데이터 랩이 제공하는 open api 중 "검색어 트렌드"를 이용해 각 키워드의 검색량 추출.

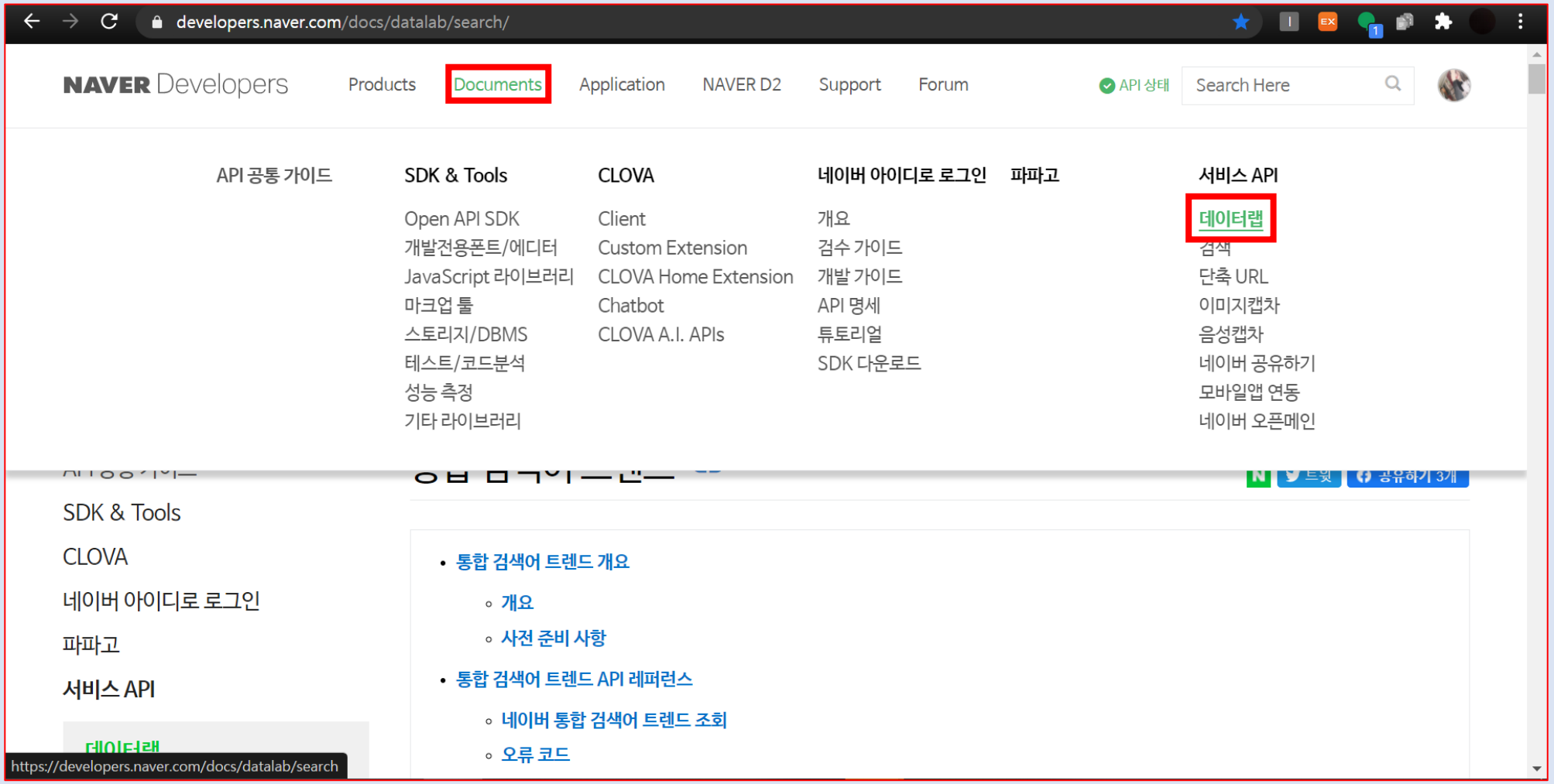
※ 네이버 개발자 센터 애플리케이션 등록 방법

- 네이버 개발자 센터 https://developers.naver.com/main/




애플리케이션 등록 (API 이용신청)
애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.
애플리케이션 이름
사용 API
[사용 API] 설정을 확인해 주세요.
NAVER Developers
Products
Documents
Application
NAVER D2
Support
Forum
API 상태
Search Here
내 애플리케이션
애플리케이션 등록
CLOVA Platform Console β
API 제휴 신청
계정 설정
비로그인 오픈 API 서비스 환경
WEB 설정
웹 서비스 URL (최대 10개)
http://도메인주소, file:///로컬URI
텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
http와 https는 구분하지 않습니다.
www는 빼고 입력해 주세요. 예) http://naver.com
서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file://로컬URI)
[웹 서비스 URL] 설정을 확인해 주세요.
등록하기
취소
선택하세요.
네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요.
40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_" 만 입력 가능합니다.
오전 2:38
2021-02-22
오전 2:39
2021-02-22

API에 대한 설명은 Documents 배너에서 볼 수 있음.



https://developers.naver.com/docs/datalab/search/#python

과정 (2) 코드

- 상세내용
- 네이버 개발자 센터에서 제공하는 open api로 각 키워드에 대한 검색량 추출

```
client_id = #"아이디"
client_secret = #"비밀번호"

url = "https://openapi.naver.com/v1/datalab/search"

body = '''{"startDate":"2021-02-18", "endDate":"2021-02-25",
"timeUnit":"date", "keywordGroups":[{"groupName": "이재용 취업
제한", "keywords":["이재용 취업 제한", "삼성전자 이재용", "삼성전
자 이재용부회장", "삼성전자 부회장"]}, {"groupName": "공장",
"keywords":["삼성전자 공장", "삼성전자 반도체", "삼성전자 오스틴",
"삼성전자 미국", "삼성전자 파운드리"]}, {"groupName": "갤럭시",
"keywords":["삼성전자 갤럭시", "갤럭시", "삼성전자 스마트폰", "삼
성전자 모바일 기기"]}]}'''


request = urllib.request.Request(url)

request.add_header("X-Naver-Client-Id", client_id)

request.add_header("X-Naver-Client-Secret", client_secret)

request.add_header("Content-Type", "application/json")

response = urllib.request.urlopen(request, data=body.encode("utf-
8"))

rescode = response.getcode()

if(rescode == 200):
    response_body = response.read()

    query = response_body.decode('utf-8')

    data_query = json.loads(query)

else:
    print("Error Code:" + rescode)
```

```
d1 = [] # 취
q1 = []
d2 = [] # 공
q2 = []
d3 = [] # 갤
q3 = []

for i in data_query['results']:
    for j in i['data']:
        if i['keywords'] == "이재용 취업 제한":
            d1.append(j['period'])
            q1.append(j['ratio'])

        elif i['keywords'] == "삼성전자 공장":
            d2.append(j['period'])
            q2.append(j['ratio'])

        elif i['keywords'] == "삼성전자 갤럭시":
            d3.append(j['period'])
            q3.append(j['ratio'])



q_df1 = pd.DataFrame()
q_df1['Date'] = d1
q_df1['queries'] = q1

q_df2 = pd.DataFrame()
q_df2['Date'] = d2
q_df2['queries'] = q2

q_df3 = pd.DataFrame()
q_df3['Date'] = d3
q_df3['queries'] = q3
```

과정 (3)
– 주가 차트 불러오기

- 야후 파이낸스 패키지를 이용해
  네이버 증권이나 인베스팅 닷컴 페이지에서 가져오는 것 보다
  편하게 주식 차트를 불러올 수 있었음.

(pip install yfinance)

```
from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override()
data = pdr.get_data_yahoo("005930.KS", start = "2020-02-19", end = "2021-02-19")
```

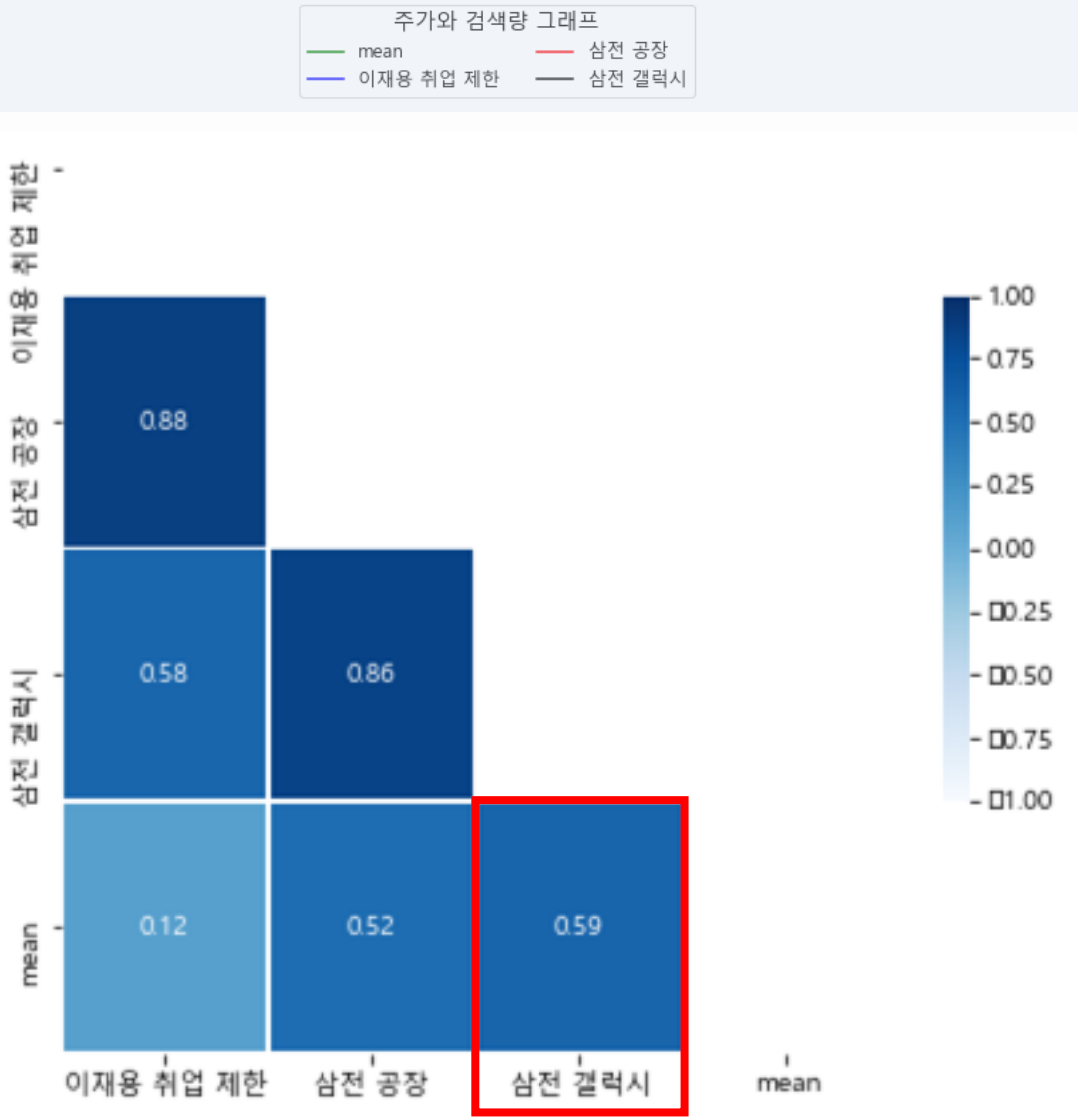
과정 (3.5) 코드

- 상세 내용
- 1-1. 야후 파이낸스 패키지를 이용해 주가 차트 불러온 후
- 1-2. 검색량 dataframe과 붙여줌.
- 1-3. 필요한 열만 남기기.

```
######### 1-1. 야후파이낸스로 주가 차트 크롤링 ###########

yf.pdr_override()

data = pdr.get_data_yahoo( " 005930.KS " , start =  " 2020-02-19 " , end =  " 2021-02-19 " )


data

# 하루 등락폭 평균치 계산해서 data 오른쪽에 붙여주기

m = pd.DataFrame()

m["mean"] = np.mean(data.iloc[:,1:3], axis=1)

data = pd.concat([data, m], axis = 1)


#인덱스 날짜 기준으로 조인

q_df1 = q_df1.set_index('Date')

q_df1.columns = ["이재용 취업 제한"]

q_df2 = q_df2.set_index('Date')

q_df2.columns = ["삼전 공장"]

q_df3 = q_df3.set_index('Date')

q_df3.columns = ["삼전 갤럭시"]
```

```
######### 1-2. 검색량 df와 붙여주기 ####################

df_left_join = pd.concat([q_df1, q_df2], axis = 1) #옆으로 이어붙이기

df_left_join = pd.concat([df_left_join,q_df3], axis = 1)

df_left_join = pd.merge(df_left_join,data, left_index = True, right_index = True, how = 'left')

df_left_join


columns = df_left_join.columns

columns


######### 1-3. 주가와 검색량 열만 남기기 ################
final_df = df_left_join[[columns[0], columns[1], columns[2], columns[9]]]


#주말 및 공휴일 등 휴장일에 주가가 NaN으로 찍혀 있는데, 이를 앞 날짜 주가로 대체
final_df = final_df.fillna(method = 'pad')


# 나머지 NaN은 0으로 채우기
final_df = final_df.fillna(0)
```

결과
– 상관 관계 시각화



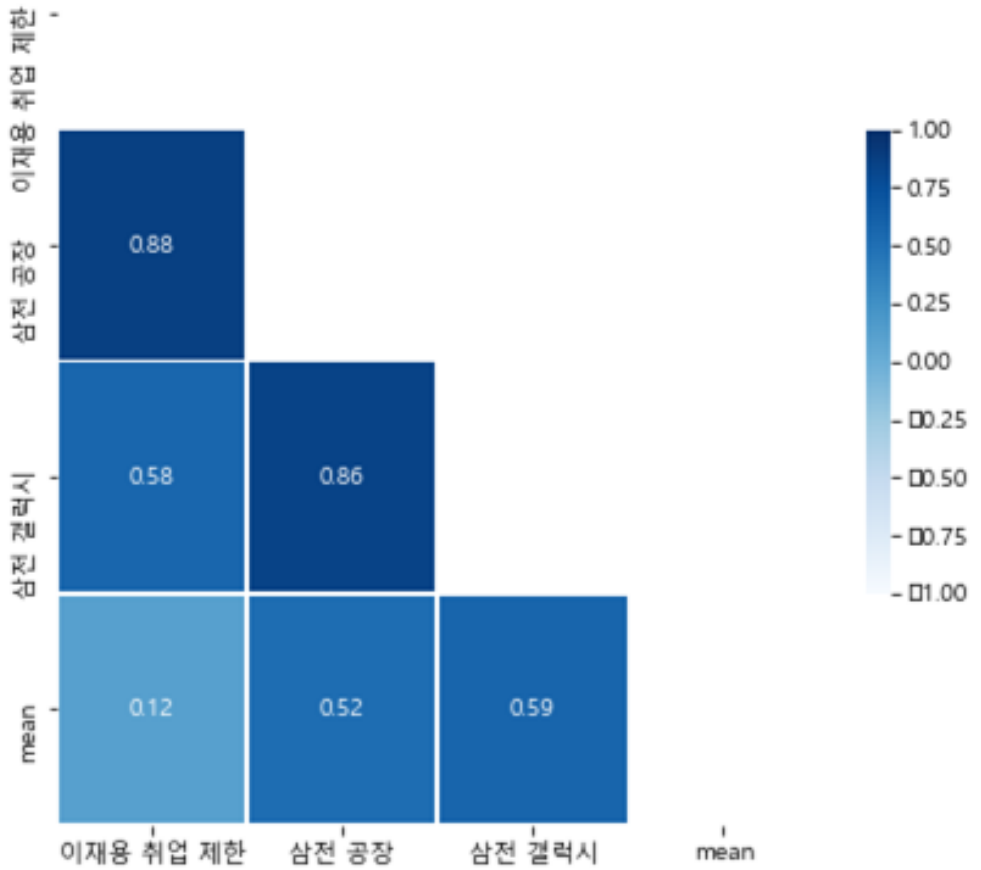
- 키워드들의 검색량과 주가 평균치 라인 차트를 하나의 axes 위에 그려 봄.
- 키워드 상위 3개의 일별 검색량과 주가의 평균와의 상관관계 출력.
- 키워드 “갤럭시”가 그나마 삼성전자 주가와 상관관계가 있다고 나옴.

결과 코드

- 상세 내용
- 1-1. 검색량과 주가 트렌드 line chart로 시각화
- 1-2. 상관관계를 시각화 하되, lower triangle 부분만 나타내기.

1-1. 검색량과 주가 트렌드 line chart로 시각화

```
plt.rcParams["figure.figsize"] = (16,12)
fig, ax1 = plt.subplots()
ax2 = ax1.twinx() # 반대 편에도 축 그려주는 코드
x = final_df.index
y1 = final_df['mean']
```



1-2. 상관관계 시각화

```
raw = final_df
df = raw.corr() # 상관관계 생성은 끝.

# 그림 사이즈 지정
fig, ax = plt.subplots( figsize=(7,7) )

# 삼각형 마스크 생성. (위 쪽 삼각형에 True, 아래 삼각형에 False)
mask = np.zeros_like(df, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# 히트맵
sns.heatmap(df,
            cmap = 'Blues',
            annot = True,   # 실제 값 표시.
            mask=mask       # 표시하지 않을 마스크 부분을 지정.
            linewidths=.5,  # 경계면 실선으로 구분.
            cbar_kws={"shrink": .5},# 컬러바 크기 절반으로.
            vmin = -1,vmax = 1   # 컬러바 범위 -1 ~ 1
           )
plt.show()
fig.savefig('주가와 검색량 간의 상관관계 표.png', dpi=fig.dpi)
```

소감

- 기억에 남는 에러:
  'Connection aborted.', RemoteDisconnected('Remote end closed connection without response')

https://butnotforme.tistory.com/entry/python%EC%9C%BC%EB%A1%9C-%EC%97%85%EB%AC%B4-%EC%9E%90%EB%8F%99%ED%99%94%EA%B9%8C%EC%A7%80-8-requests3?category=932590

참고

주가와 검색량 간의 상관관계 분석
https://yobro.tistory.com/8?category=793224

https://everyday-tech.tistory.com/entry/3%ED%83%84-%EC%89%BD%EA%B2%8C-%EB%94%B0%EB%9D%BC%ED%95%98%EB%8A%94-%EB%84%A4%EC%9D%B4%EB%B2%84-%EB%89%B4%EC%8A%A4-%ED%81%AC%EB%A1%A4%EB%A7%81-%EB%B3%B8%EB%AC%B8-%EA%B0%80%EC%A0%B8%EC%98%A4%EA%B8%B0

네이버 검색어 트렌드 API 적용 가이드
https://developers.naver.com/docs/datalab/search/#%EB%84%A4%EC%9D%B4%EB%B2%84-%ED%86%B5%ED%95%A9-%EA%B2%80%EC%83%89%EC%96%B4-%ED%8A%B8%EB%A0%8C%EB%93%9C-%EC%A1%B0%ED%9A%8C

.find_elements_by_xpath() 사용법, [@class=""] 힌트도 얻음.
https://www.fun-coding.org/crawl_advance5.html

워드클라우드
https://khann.tistory.com/60 (여기는 별 도움 안됨.)

https://liveyourit.tistory.com/58

KoNLPy
https://imworld.tistory.com/59 워클 Twitter 예제 빈도수, 형태소

https://liveyourit.tistory.com/57 워클 Okt 예제 형태소, 명사, 구 등, 빈도

에러 'Connection aborted.', RemoteDisconnected('Remote end closed connection without response') 해결
https://butnotforme.tistory.com/entry/python%EC%9C%BC%EB%A1%9C-%EC%97%85%EB%AC%B4-%EC%9E%90%EB%8F%99%ED%99%94%EA%B9%8C%EC%A7%80-8-requests3?category=932590

https://m.blog.naver.com/PostView.nhn?blogId=popqser2&logNo=221433758235&proxyReferer=https:%2F%2Fwww.google.com%2F

KoNLPy 설치 관련 문제
https://liveyourit.tistory.com/56

KoNLPy 공홈
https://konlpy-ko.readthedocs.io/ko/v0.4.3/

워클 불용어 처리
https://blog.naver.com/PostView.nhn?blogId=skfnsid123&logNo=221899440442&categoryNo=27&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=postView

워클 이미지 입히기 (이미지 마스킹)
https://blog.naver.com/PostView.nhn?blogId=skfnsid123&logNo=221899440442&categoryNo=27&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=postView

엑셀,csv읽기
https://woolbro.tistory.com/36

긍정어, 부정어 (감성어) 분류 (한 번 해 보려고 찾아만 봄.)
https://projectlog-eraser.tistory.com/19

워클서 키워드 상위 n개 뽑기
https://thinkwarelab.wordpress.com/2016/08/30/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%ED%98%95%ED%83%9C%EC%86%8C-%EB%B6%84%EC%84%9D%EC%9C%BC%EB%A1%9C-%EC%9B%8C%EB%93%9C%ED%81%B4%EB%9D%BC%EC%9A%B0%EB%93%9C-%EA%B7%B8%EB%A6%AC%EA%B8%B0/

워클, 키워드 뽑기, 불용어 처리, 사진이용
https://insightcampus.co.kr/insightcommunity/?mod=document&uid=12957

워클 불용어 제거해서 키워드 뽑기
https://mkjjo.github.io/python/2019/07/09/korean_preprocessing.html

상관관계 분석
http://blog.naver.com/PostView.nhn?blogId=kiddwannabe&logNo=221763497317&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView

https://blog.naver.com/kiddwannabe/221205309816

파이썬 merge, concat, join
https://yganalyst.github.io/data_handling/Pd_12/

파이썬 NaN 채우기
https://m.blog.naver.com/youji4ever/221791455668

twinx()한 거에 범례추가하기
https://www.javaer101.com/article/3592634.html

https://kongdols-room.tistory.com/87

matplotlib 그래프 저장
https://blog.naver.com/PostView.nhn?blogId=wideeyed&logNo=221682741587

https://codetorial.net/matplotlib/savefig.html

판다스 열단위 데이터 추출 (df에서 특정 행, 열 선택)
https://devpouch.tistory.com/46

https://blog.naver.com/PostView.nhn?blogId=rising_n_falling&logNo=221622971970&parentCategoryNo=12&categoryNo=15&viewDate=&isShowPopularPosts=false&from=postView

불용어 사전 update
https://blog.naver.com/PostView.nhn?blogId=skfnsid123&logNo=221899440442&categoryNo=27&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=postView

감사합니다.