

# Let's GRIT!

전진영, 이준희, 이민재, 전수빈

<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>
<b>개요</b>	<b>Crawling NLP</b>	<b>시행과정</b>	<b>총평</b>	<b>Q&amp;A</b>

## SNS 데이터를 활용한 주가 방향성 예측

- 네이버 종목토론방 본문 텍스트 데이터를 활용한 1 시간 후 주가 등락 예측



# 데이터 수집 준비

Let's GRIT!

[ 추가 ] 한국투자증권 HTS 설치

[ SNS ] 네이버 종목토론방

- 트위터 등 일반적인 SNS 에 비해
- 실제 주주의 비율이 높을 것으로 판단

[ 종목 ] 종목 : 카카오

- 선행연구자료에서 높은 예측 정확도를 보였던 회사 중 하나

[ 데이터 수집 기간 ]

변경 전 : 2021.09.06 ~ 2021.09.17 → 최근 기간으로 수정

변경 후 : 2022.06.01 ~ 2022.06.30

[ 아쉬운 점 ] SNS 선택 과정

→ “왜” 네이버 종목토론방을 선택했는가?

- 주주 비율이 높을 것으로 예상되는 토론방은 네이버에만 존재하는 것이 아님
- 다음(daum) 주식토론방, 팍스넷, 씽크풀, 주식 관련 유튜브 댓글 등
- 위와 같이 여러가지 경우의 수가 존재

✓ 결과적으로는 가장 활성도가 높을 것으로 여겨지는 ‘네이버’ 를 선택한 것은 변함이 없었지만,

✓ 금번 의사결정 과정에서는 네이버의 “친숙함” 에 기댄 것이라고 생각됨

→ 어떻게 하는 것이 좋았을까?

방법 1. 네이버, 다음, 팍스넷, 씽크풀 등 모든 SNS 댓글 수집

방법 2. 부득이하게 1개만 선택할 경우

- 객관적인 판단을 할 수 있도록 Data 를 수집, 검토하여 최종 선택에 활용
- 현재의 경우 효과적인 data = 댓글 등록 건수  
→ 매체별 신규 등록 건수 비교하여 1일 평균 등록건수가 가장 많은 SNS 선택

# 데이터 수집 1. 주가

Let's GRIT!

시간	시가	고가	저가	종가	5	10	20	60	120	거래량	5	10	20	60	120
06/30,15:00	70,000	70,300	69,900	69,900	0,140.00	70,430.00	70,665.00	70,253.33	73,367.50	217,743	146,331.60	163,305.80	164,617.30	227,699.18	266,641.03
06/30,14:00	70,100	70,300	70,000	70,100	0,200.00	70,550.00	70,740.00	70,241.67	73,466.67	132,410	150,170.60	150,086.30	168,066.80	222,777.77	269,034.72
06/30,13:00	70,100	70,200	70,000	70,100	0,260.00	70,620.00	70,825.00	70,320.00	73,568.33	130,518	149,570.20	148,712.10	167,410.30	222,777.77	268,175.68
06/30,12:00	70,400	70,500	70,100	70,200	0,400.00	70,700.00	70,900.00	70,400.00	73,670.00	120,000	148,680.00	147,832.00	166,518.00	222,777.77	267,266.67
06/30,11:00	70,200	70,900	70,100	70,400	0,580.00	70,730.00	70,930.00	70,430.00	73,780.00	110,000	147,770.00	146,922.00	165,610.00	222,777.77	266,357.78
시간	시가	고가	저가	종가	5	10	20	60	120	거래량	5	10	20	60	120
2022-06-30	70,400	71,100	69,900	69,900	0,120.00	70,350.00	70,550.00	70,050.00	73,160.00	210,000	145,420.00	162,390.00	163,500.00	227,000.00	266,000.00
2022-06-29	70,100	71,300	69,700	70,800	0,560.00	70,550.00	70,750.00	70,250.00	73,260.00	130,000	149,570.00	148,710.00	167,410.00	222,777.77	269,034.72
2022-06-28	71,600	72,300	70,600	71,600	0,100.00	70,780.00	76,200.00	85,413.33	90,895.00	1,082,711	1,759,640.00	2,070,251.80	1,992,835.60	1,713,317.13	2,245,990.52
2022-06-27	72,400	72,900	70,600	71,800	0,880.00	71,280.00	76,705.00	85,995.00	91,248.33	1,470,633	1,848,180.80	2,196,242.40	1,990,637.80	1,713,317.13	2,251,897.12
2022-06-24	67,400	71,500	67,400	71,500	0,440.00	71,750.00	77,190.00	86,573.33	91,595.83	2,331,632	2,041,348.40	2,266,784.40	1,984,021.40	1,707,396.80	2,259,398.98

원본데이터 수정

60분 단위 종가	
일시	종가
2022-06-30 15:00	69,900
2022-06-30 14:00	70,100
2022-06-30 13:00	70,100
2022-06-30 12:00	70,200
2022-06-30 11:00	70,400

날짜별 종가	
일시	종가
2022-06-30 16:00	69,900
2022-06-29 16:00	70,800
2022-06-28 16:00	71,600
2022-06-27 16:00	71,800
2022-06-24 16:00	71,500

## [ 60분 단위 종가 ] 의 날짜 데이터 형태 수정

원본		수기 입력		LEFT	MID	RIGHT	H3&"-"&I3&"-"&J3	L3&" "K3
시간_원본	종가_원본	년도	월	일	시간	년월일	년월일	일시
06/30,15:00	69,900	2022	06	30	15:00	2022-06-30	2022-06-30	2022-06-30 15:00
06/30,14:00	70,100	2022	06	30	14:00	2022-06-30	2022-06-30	2022-06-30 14:00
06/30,13:00	70,100	2022	06	30	13:00	2022-06-30	2022-06-30	2022-06-30 13:00
06/30,12:00	70,200	2022	06	30	12:00	2022-06-30	2022-06-30	2022-06-30 12:00
06/30,11:00	70,400	2022	06	30	11:00	2022-06-30	2022-06-30	2022-06-30 11:00
06/30,10:00	70,200	2022	06	30	10:00	2022-06-30	2022-06-30	2022-06-30 10:00
06/30,09:00	70,400	2022	06	30	09:00	2022-06-30	2022-06-30	2022-06-30 09:00
06/29,15:00	70,800	2022	06	29	15:00	2022-06-29	2022-06-29	2022-06-29 15:00
06/29,14:00	71,100	2022	06	29	14:00	2022-06-29	2022-06-29	2022-06-29 14:00
06/29,13:00	71,100	2022	06	29	13:00	2022-06-29	2022-06-29	2022-06-29 13:00
06/29,12:00	71,100	2022	06	29	12:00	2022-06-29	2022-06-29	2022-06-29 12:00
06/29,11:00	70,800	2022	06	29	11:00	2022-06-29	2022-06-29	2022-06-29 11:00
06/29,10:00	70,900	2022	06	29	10:00	2022-06-29	2022-06-29	2022-06-29 10:00
06/29,09:00	70,500	2022	06	29	09:00	2022-06-29	2022-06-29	2022-06-29 09:00
06/28,15:00	71,600	2022	06	28	15:00	2022-06-28	2022-06-28	2022-06-28 15:00

## [ 날짜별 종가 ] 의 날짜 데이터 형태 수정

셀 서식

표시 형식

맞춤

글꼴

테두리

채우기

보호

범주(C):

일반

숫자

통화

회계

날짜

시간

백분율

분수

지수

텍스트

기타

사용자 지정

보기

2021-09-01 16:00

형식(D):

yyyy"-mm"-dd hh:mm;@

h:mm:ss AM/PM

h:mm

h:mm:ss

h"시" mm"분"

h"시" mm"분" ss"초"

yyyy-mm-dd h:mm

mm:ss

mm:ss.0

@

[h]:mm:ss

yyy"-mm"-dd hh:mm;@

삭제(D)

찾기 및 바꾸기

찾기(D)

바꾸기(P)

찾을 내용(N):

00:00:00 AM

바꿀 내용(E):

4:00:00 PM

모두 바꾸기(A)

바꾸기(R)

모두 찾기(I)

다음 찾기(F)

# 데이터 수집 2. Crawling [ 1 / 2 ]

Let's GRIT!

## Crawling? ( = Scraping )

- 웹 페이지에서 필요로 하는 데이터를 추출해내는 기법
- Crawler : crawling 을 목적으로 작성된 프로그램

Selenium 으로 전체 html만 받아오게 하고,  
실제 파싱은 bs4 가 맡게 하면 속도 향상이 있다고  
한다.

### ※ 대표적인 소프트웨어

	장점	한계점
Beautifulsoup4	Html 을 분석 가능한 형태로 가공할 수 있는 라이브러리 중 가장 유명하고 많이 사용되는 외부 라이브러리 속도가 빠름	정적 페이지만 가능
Selenium	Chrome-driver 를 이용해 Chrome 을 제어하기 위해 사용 동적 페이지를 불러올 때 유용	속도 느림

✓ 파싱(Parsing) : html 의 어떤 문자열 데이터 속에서 필요한 데이터를 분석, 추출하는 방식을 일컫는 용어

# 데이터 수집 2. Crawling ( 2 / 2 )

Let's GRIT!

카카오 : 네이버 금융

finance.naver.com/item/board.naver?code=035720&page=4165

Chrome이 자동화된 테스트 소프트웨어에 의해 제어되고 있습니다.

나는 연애하기 좋은 사람? 테스트 > 더 알아보기 >

카카오 035720 2022.07.27 13:30 기준(장중) 실시간 기입개요

71,500 전일 대비 ▼900 -1.24%

전일 72,400 고가 72,600 (상한가 94,100) 거래량 627,056

시가 72,500 저가 71,200 (하한가 50,700) 거래대금 44,899 백만

종목정보 | 시세 | 차트 | 투자자별 매매동향 | 뉴스공시 | 종목분석 | **종목토론실** | 전자공시 | 공매도현황

종목토론실 토론타입 활용 TIP 과 운영원칙 안내 글쓰기

날짜	제목	글쓴이	조회	공감	비공감
2021.09.10 13:51	외인형들	ddas****	172	0	1
2021.09.10 13:50	조금 더 떨어질수 있다고 합니다.ㅏㅏ [1]	conv****	244	1	0
2021.09.10 13:50	사려고했더니 공정거래위원회서도 규제한덴다 [2]	cky1****	252	2	0
2021.09.10 13:49	결국 처벌어지네 ㅋㅋ [1]	redp****	157	2	0
2021.09.10 13:49	큰손들 오전에 끌어올리고	mooc****	159	2	0
2021.09.10 13:49	나간돈 20조가 다시 들어오겠냐	only****	139	0	0
2021.09.10 13:48	튀어라 들어가 음봉이다	486g****	131	1	0
2021.09.10 13:48	이제말이 한다면 진짜한다..들고있기 너무...	cara****	177	1	0
2021.09.10 13:48	위꼬리 실하게 달리는것 보니까	vet****	158	1	0
2021.09.10 13:47	내리면	tdc****	107	0	0
2021.09.10 13:47	157000원에 샀음 [1]	qaz****	333	2	1
2021.09.10 13:47	시총이 1조 빠리였음	qorg****	189	1	0
2021.09.10 13:46	플랫폼 없어지면 [3]	gak****	245	7	2
2021.09.10 13:45	130,000원 깨지면	578****	219	1	1

투자정보 호가10단계

시가총액 31조 8,138억원  
시가총액순위 코스피 11위  
상장주식수 444,948,291  
액면가 | 매매단위 100원 | 1주

외국인한도주식수(A) 444,948,291  
외국인보유주식수(B) 128,176,395  
외국인소진율(B/A) 28.81%

투자의견 | 목표주가 4.00배수 108,412  
52주최고 | 최저 157,500 | 66,200

PER | EPS(2022.03) 12.88배 | 5,551원  
추정PER | EPS 17.61배 | 4,061원  
PBR | EPS(2022.03) 2.75배 | 25,977원  
배당수익률 | 2021.12 0.07%

동일업종 PER 19.34배  
동일업종 등락률 -0.59%

최근조회 MY STOCK

카카오 71,500 ▼900

# data frame 에 담기

# csv 파일로 저장

```
import pandas as pd
a = pd.DataFrame({'종목명' : com_1,
                  '시간' : time_1,
                  '제목' : title_1,
                  '본문' : text_1})
```

Python


	종목명	시간	제목	본문
0	카카오	2022.07.25 15:58	20억튀..유영두 어디갔노?	.n.n.n.n.n.n코스피 5천간다?n.n 카카오 100만일 간다?n...
1	카카오	2022.07.25 15:46	최소 6자리에 가서 놀아라. 자회사 쪼개기 상장 그만두지 못할까?	카캠을 또 쪼개서 오딘을 상장해? 주주들도 너희들 쪼개 버릴 것이다. 양아치 짓 고...
2	카카오	2022.07.25 15:45	개돼지님들 오늘도 희망회로 잘돌리셨습니까?	x청한 여러분을 보면서 오늘도 살아갈 희망을 얻습니다. 맹큐 dog pig
3	카카오	2022.07.25 15:12	73000원 돌파	돌파하면 go go다
4	카카오	2022.07.25 15:09	☆<지금시간>외국인+기관<속보>☆	<지금시간->n-><외국인+기관들>3면<사자>+함n....-><투신+연기금+...

# csv 파일로 저장


```
a.to_csv("./data/kakao_2206.csv")
```

Python


## 자연어는 너무 어렵다.....!!!




개인적인 의견을 말하자면  
이 요리엔 아주 큰 문제가 있어




바로 내가 이걸 이제서야  
비로소 맛볼 수 있었다는  
점이 그러하네




왜냐면 일찍 먹어봤다면  
네 요리 실력이 이딴 식으로  
소름돋는지 미리 알았을테니까



감사합니다 셰프



죄송합니다 셰프





- 자연어? 우리가 일상생활에서 사용하는 언어
- 자연어 처리? 자연어의 의미를 분석하여 컴퓨터가 인식할 수 있도록 하는 일
- 감성 분석? 텍스트의 의견이나 감성, 평가, 태도 등의 정보를 분석하는 과정

➔ 활용 분야

음성 인식, 내용 요약, 번역, 사용자의 감성 분석,  
텍스트 분류 작업, 질의 응답 시스템, 챗봇 등

- 1단계) Preprocessing
- 2단계) Tokenization
- 3단계) Embedding
- 4단계) Learning
  - Lable이 있는 dataset 필요 (긍/부정 평가가 완료된 문장)
  - dataset 이 없다면 직접 만들어야 함

※ 자연어처리 관련 용어 정리

Corpus (말뭉치)	자연어 처리 연구 등을 염두에 두고 수집한 텍스트 데이터 (사전적 의미 = 언어 정보 처리를 위한 필수적 기본 자료)	전 처리 필요
Stop-word (불용어)	문장에 자주 등장하지만 분석에 큰 도움이 되지 않는 단어들	주로 조사, 접미사 (패키지 존재) 개발자가 직접 정의한 단어 등
Tokenization	생성한 말뭉치를 token 이라고 불리는 단위로 나누는 작업 (문장을 쪼개는 작업)	단어 단위, 형태소 단위.. 등
Embedding	각각의 token 을 벡터로 변환하는 과정	

# 준비 과정

Let's GRIT!

## ※ 한글 형태소 분석기

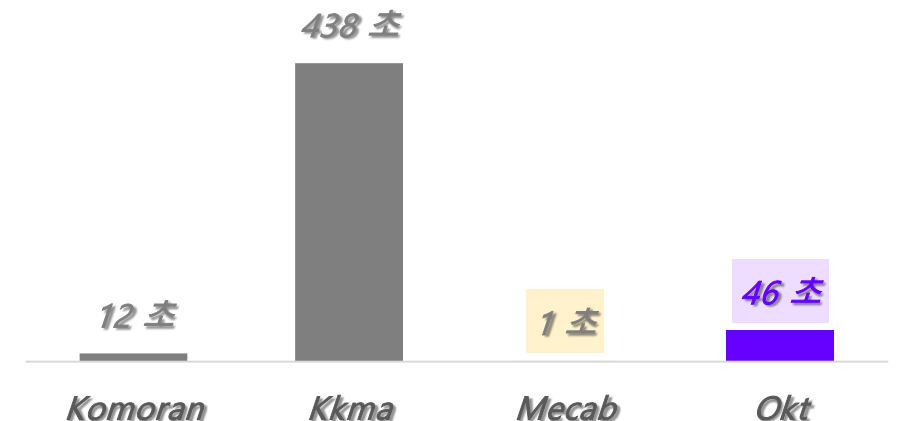
	장점	한계점
Komoran	자소 분리 가능 오탈자 분석 가능 고유명사 분석 가능	긴 로딩 속도 띄어쓰기 없는 문장분석에 취약
KKma (꼬꼬마)	띄어쓰기 오류에 덜 민감	분석 시간 미 정제된 문서 정확도 낮음
Mecab	새로운 사전 추가 가능 띄어쓰기에서 가장 좋은 성능((twitter 전) 속도, 정확도	미등록어 처리 문제 동음이의어 처리 문제
Okt (Twitter)	띄어쓰기 성능이 가장 좋음 Stemming 가능 (어간추출 = 형태소분석) 인터넷 텍스트에 강함(이모티콘, 해쉬태그 등) 비속어, 비표준어도 분석 가능	미등록어 처리 문제 동음이의어 처리문제 분석 범주 적은 편

※ 한글 형태소 분석기로 주로 Mecab, Okt 활용됨

처음에 Mecab 으로 코드 작성  
→ 윈도우에서 설치가 안 되어서 몇 번 시도하다 바로 폐기

Mecab 외에 한글 형태소 분석기에서 많이 사용되는 Okt  
선택

### Learning time (10,000 reviews)



## 다빈도 키워드 추출

➔ 다빈도 키워드를 추출하여 주가 방향성과 비교해보자 (상관관계 파악)

- SNS data
- 시간별 증가
- 등락률

	전체날짜	날짜	시간	종목명	제목	본문	제목+본문	시간변환	종가	동학률
4496	2022.06.30 09:02	2022-06-30	902	카카오	한국	공매 또 출동인가?	한국공매 또 출동인가?	10:00:00	70200	-0.002849
4497	2022.06.30 09:06	2022-06-30	906	카카오	딱 하루오르고	계속떨어지네	딱 하루오르고계속떨어지네	10:00:00	70200	-0.002849
4498	2022.06.30 09:08	2022-06-30	908	카카오	어차피	6만원대로 내릴꺼임~~지금 간보는거임ㄱ	어차피6만원대로 내릴꺼임~~지금 간보는거임ㄱ	10:00:00	70200	-0.002849
4499	2022.06.30 09:11	2022-06-30	911	카카오	영두야	밥은 먹고 실례발치나	영두야밥은 먹고 실례발치나	10:00:00	70200	-0.002849
4500	2022.06.30 09:11	2022-06-30	911	카카오	네이버,카카오 다 저점같은데,,	어디 들어가는게 낫나요,, 카카오가 더 땡기는데,,,	네이버,카카오 다 저점같은데,, 어디 들어가는게 낫나요,, 카카오가 더 땡기는데,,,	10:00:00	70200	-0.002849
4501	2022.06.30 09:11	2022-06-30	911	카카오	채팅회사 잘 지냈니!!	너 다 분할해서\n1만원이면 맞아	채팅회사 잘 지냈니!!너 다 분할해서\n1만원이면 맞아	10:00:00	70200	-0.002849
4502	2022.06.30 09:13	2022-06-30	913	카카오	오	지저스 칠만도 깨지나!!!	오지저스 칠만도 깨지나!!!	10:00:00	70200	-0.002849
4503	2022.06.30 09:16	2022-06-30	916	카카오	카카오는 N자형의 하락	N자 모양을 차트에 막 올려봐라\n\n\n겹치는 상황이 포착될 것이다\n\n\n쉽게...	카카오는 N자형의 하락N자 모양을 차트에 막 올려봐라\n\n\n겹치는 상황이 포착될...	10:00:00	70200	-0.002849
4504	2022.06.30 09:22	2022-06-30	923	카카오	카카오 목표주가	카카오의 전일가는 70,800원 으로 마감했으 나 현재 70773내 6월 28일 0시	카카오 목표주가카카오의 전일가는 70,800원 으로 마감됐으며 현재 70773내 6월	10:00:00	70200	-0.002849

# 1차 시도

## 다빈도 키워드 추출

Let's GRIT!

### # Tokenization

```
from konlpy.tag import Mecab
import re

mecab = Mecab()
text = mecab.nouns(a['본문'].to_string()) # 문자열에서 명사만 추출하기
text
```

['코스피',  
'천',  
'카카오',  
'만',  
'원',  
'카',  
'gem',  
'오딘',

본문

.\n.\n.\n.\n.\n코스피 5천간다?\n\n카카오 100만원 간다?\n...  
카gem을 또 쪼개서 오딘을 상장해? 주주들도 너희들 쪼개 버릴 것이다. 양아치  
짓 고...  
x청한 여러분을 보면서 오늘도 살아갈 희망을 얻습니다. 땡큐 dog pig

### # 다빈도 키워드 추출

```
from collections import Counter

count = Counter(text)
tag_count = []
tags = []

for n, c in count.most_common(100):
    dics = {'tag': n, 'count': c}

    if len(dics['tag']) >= 2 and len(tags) <= 49:
        tag_count.append(dics)
        tags.append(dics['tag'])

for tag in tag_count:
    print("{:<14}".format(tag['tag']), end='\t')
    print("{}".format(tag['count']))
```

카카오	4247
기업	737
매수	729
개미	723
오늘	663
주식	604
지금	594
규제	423
김범수	414
반등	391
정부	390
하락	375
추가	364
이제	346
공매도	340
네이버	328
내일	314
주주	312
민주당	297

다빈도 키워드 중  
“어떤 키워드가 중요하고,  
어떤 키워드는 중요하지 않은 지..”  
객관적으로 어떻게 판단하지???  
→ [결론] 유사도 분석 추가

→ `중요한 키워드` 를 선택하기 위해 유사도 분석 과정 추가

- ① read.csv : "kakao\_2206.csv"
- ② df 생성 / [전체] column 생성 (제목+본문)
- ③ 중복데이터 삭제
- ④ 결측치 확인 및 제거
- ⑤ 한글 정제 및 불용어 제거 : 한국 불용어 사전 이용
- ⑥ 토큰화

# word 2 vec

```
from gensim.models import Word2Vec

model = Word2Vec(sentences = tokenized_data,
                 vector_size = 100,      # 벡터의 차원 수. 즉, 단어 당 만들어질 벡터의 크기
                 window = 5,            # 컨텍스트 윈도우 크기 (기본값 5 = 좌우 5개)
                 min_count = 5,         # 단어 최소 빈도 수 제한 = 빈도가 적은 단어들은 학습하지 않음
                 workers = 4,           # 학습을 위한 프로세스 수
                 sg = 0)                # 훈련 알고리즘 적용 => 0은 CBOW, 1은 Skip-gram
```

```
similar_word = model.wv.most_similar("카카오",topn=100)    # topon : 출력결과 개수 지정 (기본값 = 10)
similar_word
```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
[('뱅크', 0.9948774576187134),
 ('페이', 0.9932606220245361),
 ('모빌리티', 0.9841142892837524),
 ('적정', 0.9816797971725464),
 ('삼성', 0.9800845384597778),
 ('추가', 0.9786707162857056),
 ('대견하다', 0.9755174517631531),
 ('치고는', 0.9749945998191833),
 ('엘지', 0.9747391939163208),
 ('매각', 0.9744933247566223),
```

## Word 2 Vec 설명

## Word 2 Vec = Word to Vector

```
from gensim.models import Word2Vec
```

```
model = Word2Vec(sentences = tokenized_data,
                 vector_size = 100,      # 벡터의 차원 수. 즉, 단어 당 만들어질 벡터의 크기
                 window = 5,            # 컨텍스트 윈도우 크기(기본값 5 = 좌우 5개)
                 min_count = 5,         # 단어 최소 빈도 수 제한 = 빈도가 적은 단어들은 학습하지 않음
                 workers = 4,           # 학습을 위한 프로세스 수
                 sg = 0)                # 훈련 알고리즘 적용 => 0은 CBOW, 1은 Skip-gram
```

```
similar_word = model.wv.most_similar("카카오",topn=100)      # topen : 출력결과 개수 지정 (기본값 = 10)
similar_word
```

Output exceeds the size limit. Open the full output data in a text editor

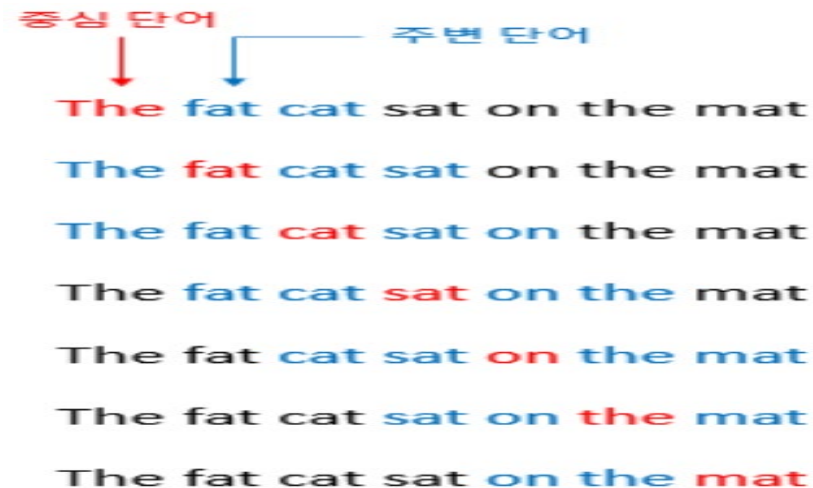
```
[('뱅크', 0.9948774576187134),
 ('페이', 0.9932606220245361),
 ('모빌리티', 0.9841142892837524),
 ('적정', 0.9816797971725464),
 ('삼성', 0.9800845384597778),
 ('추가', 0.9786707162857056),
 ('대견하다', 0.9755174517631531),
 ('치고는', 0.9749945998191833),
 ('엘지', 0.9747391939163208),
 ('매각', 0.9744933247566223),
```

### CBOW : 주변 단어를 통해 단어 예측

→ 나는 (대한민국) 국민이다

## Skip-Gram : 단어를 통해 주변 단어 예측

→ (나는) 대한민국 (국민이다)



## 2차 시도

## 키워드 유사도 분석

Let's GRIT!

### ※ 1~2차 시도 후 결과

#### ① 다빈도 키워드 & 유사도

카카오	4247
기업	737
매수	729
개미	723

```
[('뱅크', 0.9948774576187134),  
 ('페이', 0.9932606220245361),  
 ('모빌리티', 0.9841142892837524),
```

#### ② 주가

일시	종가	주가 변동
2022-06-30 15:00	69,900	하락
2022-06-30 14:00	70,100	-
2022-06-30 13:00	70,100	하락
2022-06-30 12:00	70,200	하락
2022-06-30 11:00	70,400	NaN

1. 유사도 수치를 이끌어내는 “중심단어”를  
단순히 `빈도` 로만 정해도 되는 것이 맞나?

2. 빈도-유사도-주가를 대체 어떻게 연결해야 하는가??

### ※ 우리가 원했던 형태

Text e.g. : 이번 달 카카오 실적 역대 최대래!

text 평가  
긍정/부정

주가 비교  
상승/하락

상관관계  
있음/없음

### [ 원인 ]

상당수의 선행 연구자료에서 키워드 간 유사도를 산출하고 있음  
= 이것이 text 의 극성을 분류해주는 것으로 오판함

NLP 기초 실습은 주로 ‘네이버 영화 리뷰’ dataset 을 활용  
= 이 자료에는 이미 해당 문장에 대한 긍정/부정 lable 이 마련되어 있음

### [ 결론 ]

문장에 대한 극성(긍정/부정) 평가(lable) 필요

→ 감성사전 필요

→ 감성사전을 구축할 시간적 여유 없음

→ 현재 구축된 사전 사용 결정

→ \*KOSAC 사용 시도했으나 실패 (메일 회신 無)

→ \*VADER(영어 분장분석 외부 라이브러리) 사용해 보기로 결정

→ VADER 감정분석 라이브러리를 활용한 감정분석 시도

### ※ Vader

비지도학습에 기반한 감정 분석 라이브러리 (영어)

- 주로 SNS 텍스트에 대한 감성 분석을 제공하기 위한 패키지
- 뛰어난 감성 분석 결과를 제공하고, 수행시간이 빠름
- 대용량 텍스트 데이터에 잘 사용되는 패키지

→ SentimentIntensity Analyzer 클래스 이용

→ `from nltk.sentiment.vader import SentimentIntensityAnalyzer`

**‘영어’를 위한 라이브러이므로  
한글에 적용하는 것은 부적절하지만  
시도라도 한 번 해보자!**

긍정/중립/부정 score 를 적절히 조합  
→ ‘Compound score’ 로 최종 평가

+	0	-
긍정 (보통 0.1 이상)	중립	부정



→ SentimentIntensity Analyzer 라이브러리를 사용하기 위해 한글을 영어로 번역 → Papago 사용

## 파파고 번역1. 네이버 open API 사용

```
url = 'https://openapi.naver.com/v1/papago/n2mt'
client_id = 'qRRDuul_fIUg_jytEypx'
client_secret = 'eMUZEK5G_1'

text = a
afterTranslate = []
source = 'ko'
target = 'en'

encText = urllib.parse.quote(str(a))
data = f'source={source}&target={target}&text=' + encText
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id", client_id)
request.add_header("X-Naver-Client-Secret", client_secret)
response = urllib.request.urlopen(request, data=data.encode("utf-8"))
rescode = response.getcode()

if (rescode == 200):
    response_body = response.read()
    decode = json.loads(response_body.decode('utf-8'))
    # print(decode)
    result = decode['message']['result']['translatedText']
    print(result)
else:
    print("Error Code:" + rescode)
```

### 문제점) 번역한 글들이 하나로 합쳐진 채로 추출됨

At this time, KAKAO's main dish is being swept away by Joseon.Out of the 7 main news, 5 of them are Chosun.  
I get paid a lot, but I don't like to keep the audio line open during work hours.  
2. I'm so disappointed^^ Comments about grandfather Yoo Young-doo left in Thor's room...  
You may not know if I'm shaved, but I'll sit comfortably in the seat that Kim Bum Soo created and use it as an excuse for metaverse.  
Four hundred-year-old Yankee bullies!We were originally one, but forced by Mi and So...  
...  
6615 If you think Yoon Seokyeol is an outcast, <https://youtu.be/C2-HTUyE6Y...>  
6616, I can't sleepI don't know how much I'll be hooked on the Inverse tomorrow.money copy  
6617, how much was the average before the liquid level?I'm curious.  
6618 NASDAQ: Seven thousand.\n\nKAO goes to 30,000.\n\nLOL  
Why don't the 6619 KAKAO affiliates pay dividends? The more I think about it, the more disgusting it is  
Name: overall, Length: 6620, dtype: object

```
df1 = pd.DataFrame(data = [result], columns=['1'], index=[0])
df1
```

1

0 At this time, KAKAO's main dish is being swept...

# 3차 시도

## Vader 활용 감정분석 ; papago 활용

Let's GRIT!

해결 방안 1 : 어떤 값(예: " or ' or \n)을 기준으로 분할 → but, 명확한 기준이 되는 값이 없음

### 1-1. enter 기준(\n)으로 분할

```
1 aa = result.split("\n")
2 aa
```

["At this time, KAKAO's main dish is being swept away by Joseon.Out of the 7 main news, 5 of them are Chosun.",  
"I get paid a lot, but I don't like to keep the audio line open during work hours.",  
"2. I'm so disappointed^^ Comments about grandfather Yoo Young-doo left in Thor's room...",  
"You may not know if I'm shaved, but I'll sit comfortably in the seat that Kim Bum Soo created and use it as an excuse for metaverse.",  
'Four hundred-year-old Yankee bullies!We were originally one, but forced by Mi and So...']

**문제점**

### 1-2. " 기준으로 분할

```
1 bb = result.split('"')
2 bb
```

["At this time, KAKAO's main dish is being swept away by Joseon.Out of the 7 main news, 5 of them are Chosun.\nI get paid a lot, but I don't like to keep the audio line open during work hours.\n2. I'm so disappointed^^ Comments about grandfather Yoo Young-doo left in Thor's room...\nYou may not know if I'm shaved, but I'll sit comfortably in the seat that Kim Bum Soo created and use it as an excuse for metaverse.\nFour hundred-year-old Yankee bullies!We were originally one, but forced

### 1-3. ' 기준으로 분할

```
1 cc = result.split("'")
2 cc
```

['At this time, KAKAO',  
's main dish is being swept away by Joseon.Out of the 7 main news, 5 of them are Chosun.\nI get paid a lot, but I don',  
't like to keep the audio line open during work hours.\n2. I',  
'm so disappointed^^ Comments about grandfather Yoo Young-doo left in Thor',  
's room...\nYou may not know if I',  
'm shaved, but I',  
'll sit comfortably in the seat that Kim Bum Soo created and use it as an excuse for metaverse.\nFour hundred-year-old Yankee bullies!We were originally one, but

**기준을 잡고 데이터를 분할해도 6620개로 분할 안됨**

**→ 번역한 글들이 하나로 합쳐진 채로 추출됨**

```
1 type(result)
2 result
3 result.replace('.', '').replace(',', '').replace(' ', '').strip()
4 result
5 if(len(s) != 0):
6     _result.append(s)
7 _result
```

["At this time KAKAO's main dish is being swept away by Joseon",  
'Out of the 7 main news 5 of them are Chosun',  
"I get paid a lot but I don't like to keep the audio line open during work hours",  
"2. I'm so disappointed^^ Comments about grandfather Yoo Young-doo left in Thor's room",  
"You may not know if I'm shaved, but I'll sit comfortably in the seat that Kim Bum Soo created and use it as an excuse for metaverse",  
"Four hundred-year-old Yankee bullies!We were originally one, but forced by Mi and So..."]

# 3차 시도

## Vader 활용 감정분석 ; papago 활용

Let's GRIT!

해결 방안 1 : 파파고 해석에서 한 줄로 출력되는 결과값을 txt 파일로 저장하도록 수정

### 파파고 번역 openAPI사용

→ txt파일 로 저장

```
1 import os
2 import sys
3 import urllib.request
4 import json
5 client_id = "qRRDuul_fIUg_ivtEvpX"
6 client_secret = "eMUZEK1m3u3cc"
7 fileName = 'C:/PYTHON/GRIT/결과값.txt'
8
9 #번역할 메모장 불러오기
10 with open(fileName, 'r', encoding='utf-8') as f:
11     srcText = f.read()
12
13 encText = urllib.parse.quote(srcText)
14 # data = "source=ko&target=en&text=" + encText
15 data = "source=ko&target=en&text=" + "%urlib.parse.quote(srcText)"
16 url = "https://openapi.naver.com/v1/papago/n2mt/translate"
17 request = urllib.request.Request(url)
18 request.add_header("X-Naver-Client-Id", client_id)
19 request.add_header("X-Naver-Client-Secret", client_secret)
20 response = urllib.request.urlopen(request, data=data.encode("utf-8"))
21 rescode = response.getcode()
22 if(rescode==200):
23     response_body = response.read()
24     # print(response_body.decode('utf-8'))
25
26 #json 형 변환
27 res = json.loads(response_body.decode('utf-8'))
28 from pprint import pprint
29 pprint(res)
```

**문제점)**

**HTTP Error는 서버가 차단됐을 때 생기는 오류라고 함**

**→ 포기**

**→ Google 번역기 사용 모색**

```
30
31 #파일 생성
32 with open('translate.txt', 'w', encoding='utf8') as f:
33     f.write(res['message']['result']['translatedText'])
34
35 else:
36     print("Error Code:" + rescode)
```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

-----

HTTPError: HTTP Error 400: Bad Request

Traceback (most recent call last):

<ipython-input-113-0ad4d3cb6953> in <module>

18 request.add\_header("X-Naver-Client-Id", client\_id)

19 request.add\_header("X-Naver-Client-Secret", client\_secret)

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

21 rescode = response.getcode()

22 if(rescode==200):

c:\Users\subin\Anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, cafile, capath, cadefault, context)

220 else:

221 opener = \_opener

--> 222 return opener.open(url, data, timeout)

223

224 def install\_opener(opener):

c:\Users\subin\Anaconda3\lib\urllib\request.py in open(self, fullurl, data, timeout)

529 for processor in self.process\_response.get(protocol, []):

530 meth = getattr(processor, meth\_name)

--> 531 response = meth(req, response)

532

533 return response

c:\Users\subin\Anaconda3\lib\urllib\request.py in http\_response(self, request, response)

638 # request was successfully received, understood, and accepted.

...

--> 649 raise HTTPError(req.full\_url, code, msg, hdrs, fp)

650

651 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 400: Bad Request

# 4차 시도

## Vader 활용 감성분석 ; google 활용

Let's GRIT!

활용방안 : 구글 파일 번역 기능을 통해 제목+본문 데이터를 영어로 번역 후 Vader 사용

```
1 df = pd.read_csv('./data/카카오(eng).csv')
2 #제목+본문 데이터를 구글번역기로 번역한 파일 불러오기
3 df
```

✓ 0.2s

Python

	full date	Item name	all
0	2022.06.01 05:23	cacao	At this time, the main KAKAO is swept away by ...
1	2022.06.01 08:44	cacao	get a lot of money and it's dissatisfying that...
2	2022.06.01 08:59	cacao	Grandpa Yoo Young-doo is very disappointing~~^...
3	2022.06.01 09:29	cacao	I don't know what he's doing, but he's trying ...
4	2022.06.01 09:37	cacao	An unwritten Yankee puppet who wants to fight ...
...	...	...	...
6615	2022.06.30 23:31	cacao	If you see this and think that Seok-Yeol Yoon ...
6616	2022.06.30 23:34	cacao	I can't sleep. How much will I be stuck in the...
6617	2022.06.30 23:40	cacao	I wonder what the average price was before thi...
6618	2022.06.30 23:52	cacao	7,000 on Nasdaq. KAKAO goes 30,000. haha
6619	2022.06.30 23:54	cacao	Why don't KAKAO affiliates pay dividends? The ...

6620 rows × 3 columns

```
1 df.isna().sum() # 결측치 확인
```

✓ 0.1s

Python

```
full date    0
Item name    0
all          0
dtype: int64
```

### # Vader 사용 → 감성분석 수치 도출

```
1 sia = SentimentIntensityAnalyzer()
2
3 list = [] # 빈 리스트 생성
4 for index, row in df.iterrows():
5     list.insert(0, sia.polarity_scores(row['all']))
6
7
8 df2 = pd.DataFrame(data = list, columns=['neg', 'neu', 'pos', 'compound'],
9                    index=df.index)
10 df2
```

✓ 5.9s

Python

	neg	neu	pos	compound
0	0.000	0.903	0.097	0.0762
1	0.000	0.667	0.333	0.4588
2	0.000	1.000	0.000	0.0000
3	0.143	0.857	0.000	-0.2500
4	0.344	0.656	0.000	-0.9001
...	...	...	...	...
6615	0.206	0.754	0.040	-0.9873
6616	0.000	0.750	0.250	0.7684
6617	0.178	0.703	0.119	-0.4810
6618	0.169	0.753	0.078	-0.7216
6619	0.000	1.000	0.000	0.0000

6620 rows × 4 columns

## Vader 활용 감정분석 ; google 활용

**활용 방안 :** 구글 파일 번역 기능을 통해 제목+본문 데이터를 영어로 번역 후 Vader 사용

## # 상관관계 도출

 $2 \, dv$ 

✓ 0.2s

		compound	종가	전시간종가	전시간대비변동가격	전시간기준등락율
날짜	시간변환					
2022-06-02	10:00:00	0.112439	83600	83300.0	300.0	0.360144
	11:00:00	0.159000	83500	83600.0	-100.0	-0.119617
	12:00:00	-0.492033	83400	83500.0	-100.0	-0.119760
	13:00:00	-0.047057	83500	83400.0	100.0	0.119904
	14:00:00	0.205371	83600	83500.0	100.0	0.119760
...	...	...	...	...	...	...
2022-06-30	12:00:00	0.400167	70200	70400.0	-200.0	-0.284091
	13:00:00	-0.289233	70100	70200.0	-100.0	-0.142450
	14:00:00	0.359688	70100	70100.0	0.0	0.000000
	15:00:00	0.251736	69900	70100.0	-200.0	-0.285307
	16:00:00	0.109861	69900	69900.0	0.0	0.000000

159 rows x 5 columns

```
1 # compound와 전시간기준등락율의 상관관계
2 dv['compound'].corr(dv['전시간기준등락율'])
```

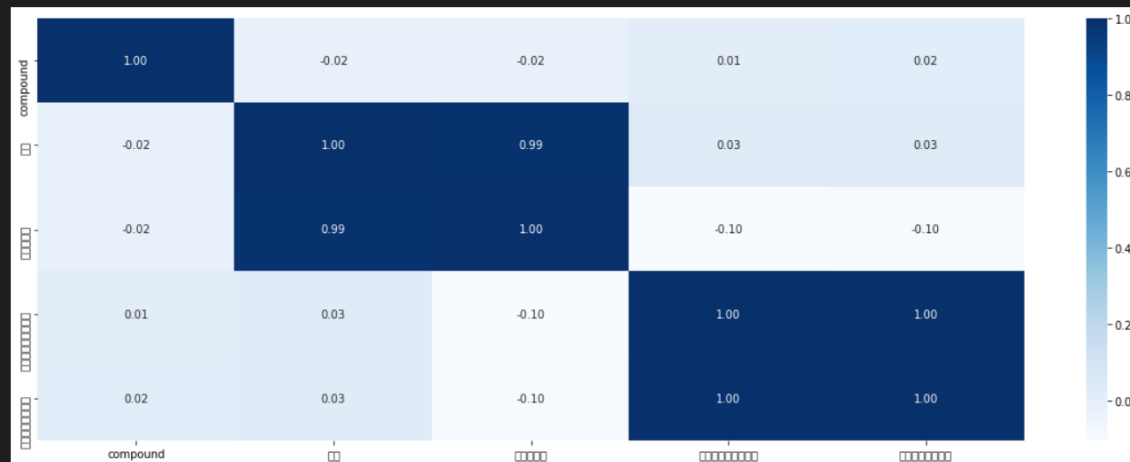
✓ 0.1s

0.019613025798778962

```
1 # compound와 전시간기준등락율의 공분산
2 dv['compound'].cov(dv['전시간기준등락율'])
```

✓ 0.1s

0.002449114244804781



[ 한 줄 평 ] 우린 NLP 에 대한 지식/이해가 너무 부족한 상태로 텍스트 마이닝에 도전했다..

## [ 아쉬운 점 ]

### 1. 선행연구자료 선택이 너무 편중되었던 것이 아닌가?

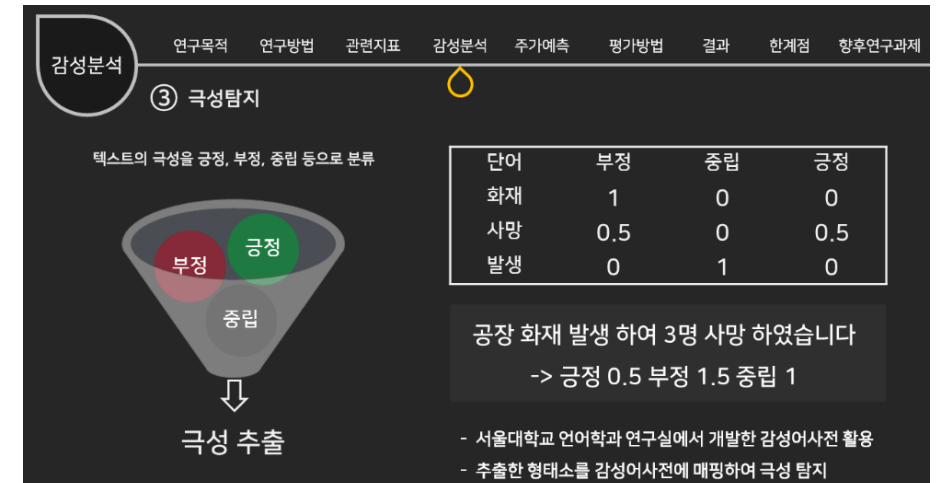
- : 해커톤 진행 방향성 논의 단계에서 확인했던 자료들 대부분이 [예측] 을 위한 연구였음
- 문장의 감정분석 과정이 생략된 논문이 많음 (분석 결과만 기재)
- 문장에 Lable(긍정/부정) 을 부여하는 과정에 대한 사전 이해 부족

### 2. NLP 에 대한 지식 부족

- `자연어 처리`에 대해서 팀원들이 함께 모여 study 하는 시간이 필요했음

### 3. 구글링을 통한 정보 검색의 한계

- NLP 방법에 대한 정보는 대부분 네이버 영화 리뷰로 실습해 보았다는 내용
- 실질적인 문제 해결 방법을 찾기가 어려웠음



## Stock\_Prediction

대학교 캡스톤 디자인이라는 과목에서 5명의 팀원들과 진행했던 프로젝트입니다.

과거의 데이터들 중 보조지표와 뉴스 기사 데이터를 활용해 다음의 주가의 등락을 예측하는 것입니다.

테스트는 여러 종목에서 했으며 대표적으로 한진에서 테스트한 코드를 올리며, 네이버에서 주식데이터를 어떻게 가져왔는지 보여드릴 수 있는 코드도 포함합니다.

감성분석을 한 코드는 다른 팀원의 것으로 따로 가져오지는 못하였습니다. re\_score.csv가 감성분석을 이용하여 csv파일로 만든 것입니다.

## 감성분석 Model은 영어만 있는 것이 아니었다!

### 1. KoBERT

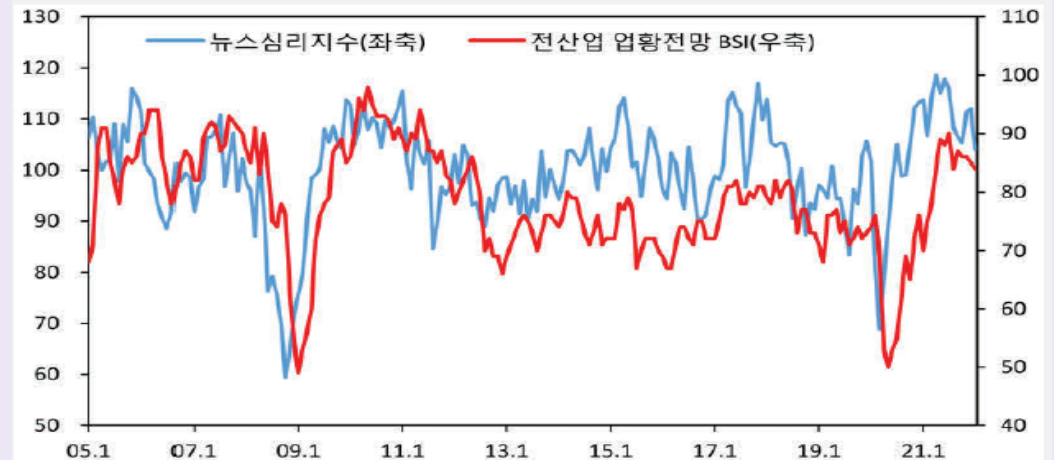
- BERT model 의 한국어 버전
- BERT : google이 만든 AI 언어모델, 2018년 소개

### 2. KR – FinBERT

- 금융 관련 뉴스데이터 훈련
- ✓ Etc. KcBERT : 한국어 뉴스 댓글 훈련

## 한국은행 NSI 지수 2022년 제 1호 국민계정리뷰

(월별 뉴스심리지수와 전산업 업황전망BSI)



- ✓ NSI (뉴스심리지수) : 뉴스기사의 텍스트를 분석함

**Q & A**