KB증권 M-able mini가 품은

오해

KB Future Finance A.I. Challenge 제3회

KB-ALBERT를 활용한 금융 자연어 혁신 아이디어

중앙대학교 유승욱, 권예진, 김민주

진행 순서

01 상황 분석
아이디어 도출 배경

02 아이디어 소개
AdobeXD로 구현한 프로토타입

03 데이터 확보
데이터 크롤링 및 전처리
사용된 모든 코드는 Github*에 기록됨

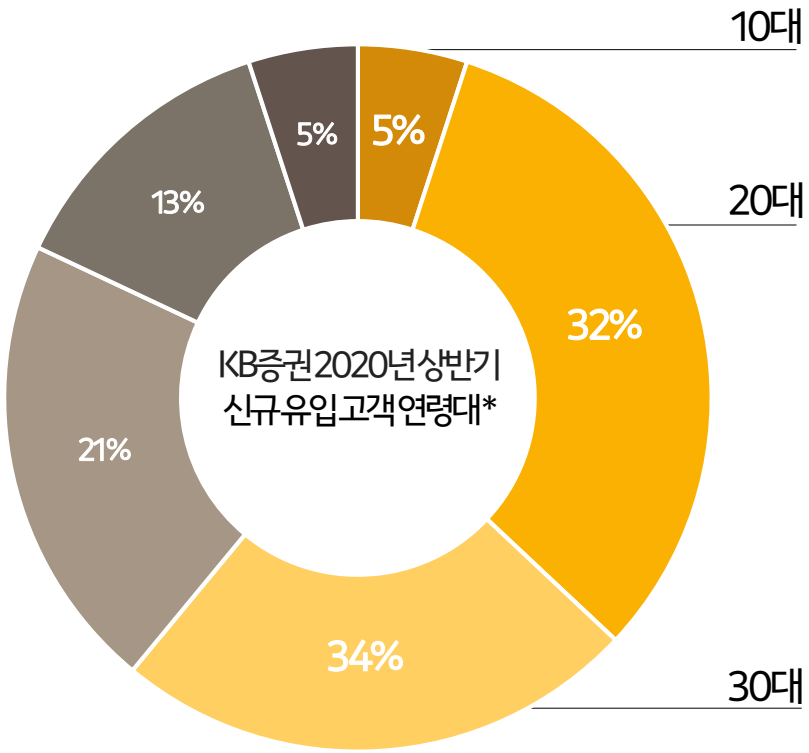
04 모델링
kb-albert 활용 감성분석 모델링
사용된 모든 코드는 Github*에 기록됨

05 확장
아이디어 강점 및 발전 방향

*Github: https://github.com/woog2ee/OHAE

1 상황 분석

MZ세대의 주식 관심 급증



- KB증권 2020년 상반기 신규 유입 고객을 살펴보면
  10대 - 20대의 비율이 전체의 37%를 차지하며,
  30대까지 포함할 경우 전체의 60%를 거뜬히 넘어감

- 삼성증권의 2020년 상반기 신규 유입 고객*역시
  30대 이하가 전체의 52.5%를 차지하며,
  카카오페이증권은 서비스 시작 9개월 만에 300만명이 개설*하여
  20대와 30대, 40대 모두 고른 분포를 보였음

- 이렇듯 주식시장에 대한 젊은 층의 높은 관심으로
  최근 온/오프라인 주식 스터디 및 각종 주식 커뮤니티가 생겨나기 시작함

출처: 각사,매일경제

MZ세대의 주식 관심 급증

“게임하듯이 주식·코인 투자”… 금감원 ‘MZ세대 보고서’*

▶ 2030 5명중 1명 주식 투자중

금융감독원이 최근 MZ세대(밀레니얼+Z세대)의 투자 성향을 분석하고 “MZ세대는 투자를 게임하듯이 한다”고 결론 지었다.
…

금감원의 ‘MZ세대의 특징과 금융산업에서의 시사점’ 내부 보고서는 MZ세대를 고위험 자산에 공격적인 투자 성향을 보이는 것으로 평가했다.
금감원은 MZ세대의 과도한 투기적 성향을 억제하는 대책을 마련해야 한다고 진단했다.
…

- 다양한 투자 프로그램의 등장 및 코로나로 인한 비대면 서비스의 확대로 젊은 층들의 주식시장 진입장벽이 낮아짐

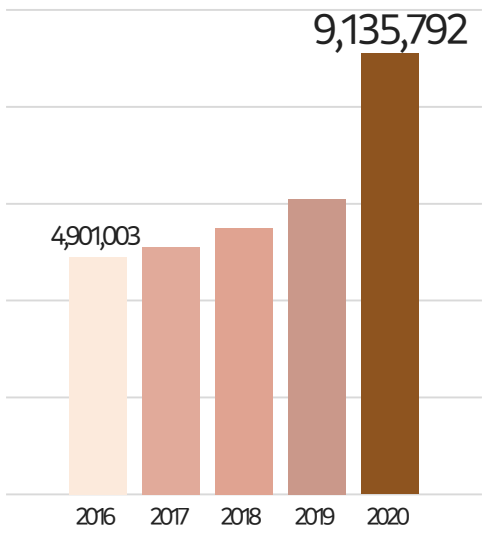
- 국내에서 주식을 투자하는 개인 3명 중 1명은 작년부터 투자를 시작한 것으로 나타나는 등* 최근 주식에 대한 관심이 급격히 증가한 양상

- 온라인 매체 및 커뮤니티 발달로 현재 인터넷 생태계는 그야말로 TMI(Too Much Information) 인 정보 포화 상태

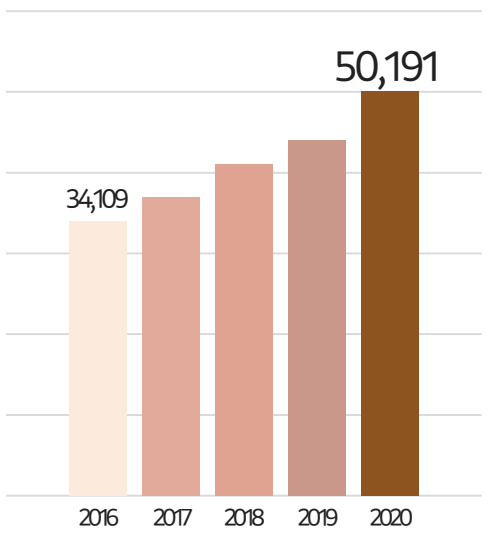
- 주식을 곧 시작한 젊은 층이 지나치게 매수/매도를 유도하는 자극적인 글을 편향적으로 받아들여 정보의 오해를 만들어 냄

출처: 매일경제, 연합뉴스

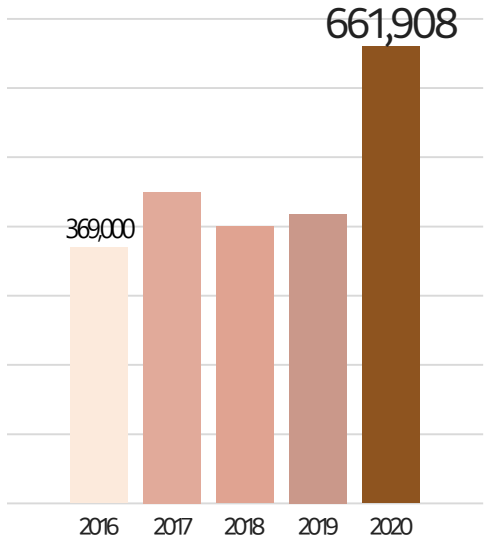
나날이 증가하는 투자시장 규모



개인소유자수 현황(단위: 명)*



개인보유주식수 현황(단위: 백만주)*

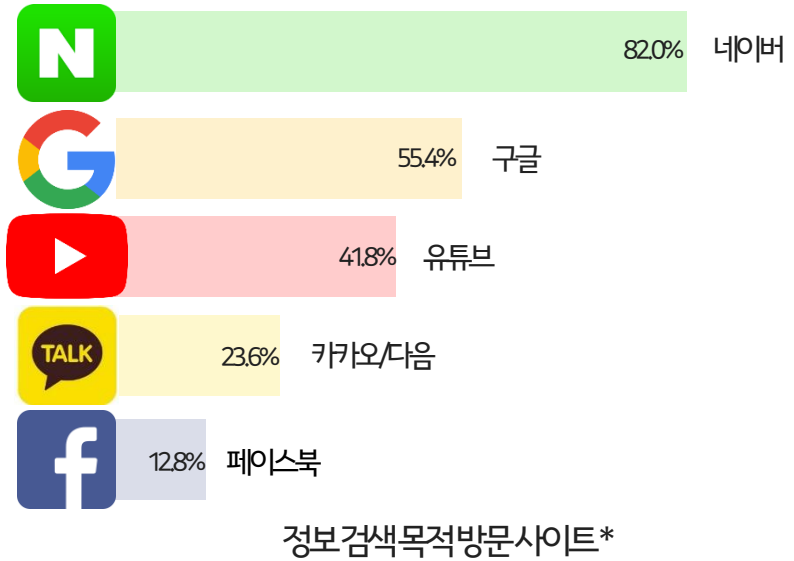


개인보유금액 현황(단위: 십억원)*

- 투자시장 규모를 살펴보면 주식 개인 소유자 수, 보유 주식 수, 보유 금액 모두 근 3년간 우상향했으며 앞으로도 꾸준히 추세가 증가할 것으로 전망됨
- 특히 코로나19 사태 이후 폭락했던 증시가 반등하는 과정에서 신규 개인 투자자들이 주식시장에 대거 진입함

출처: 한국예탁결제원

너무나도 방대한 정보의 원천



- SPRi 소프트웨어 정책 연구소에 따르면 *
  정보 검색을 목적으로 방문하는 사이트는
  네이버, 구글, 유튜브 순으로 카카오 / 다음, 페이스북이 그 뒤를 따름

- 밀레니얼 세대를 대상으로 선호하는 투자정보 채널 활용도 *를
  살펴본 결과 비대면 상황에서 '인터넷 전문 사이트 검색' 을 가장 선호하며
  '모바일 앱', '뉴미디어(유튜브)' 가 그 뒤를 따름

- 이렇듯 정보 검색과 투자정보 채널을 선택함에 있어
  온라인 포털 사이트의 순위가 높았으며
  주식 정보 습득에서도 유튜브 등의 뉴미디어가 새로 떠오름

출처: SPRi 소프트웨어정책연구소, 미래에셋은퇴연구소

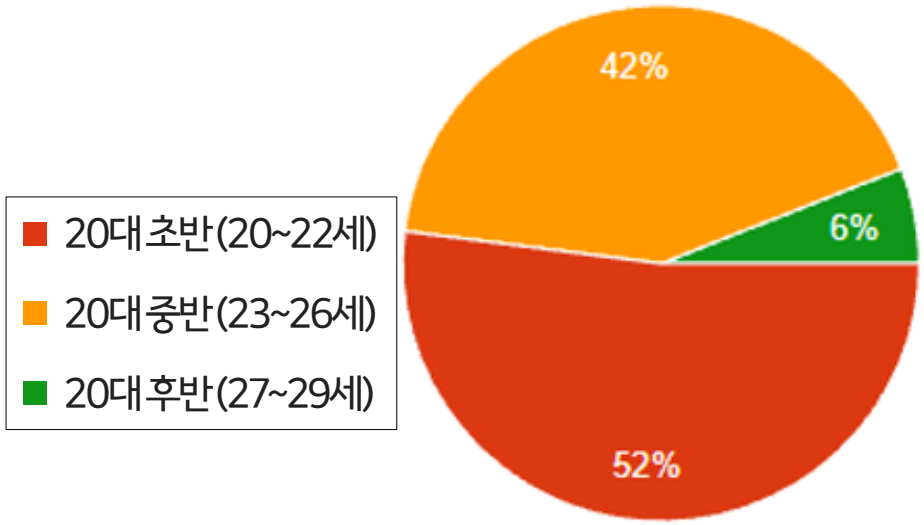
자체 설문 진행

<주식을 다룬 소셜 미디어 사용에 대한 설문>

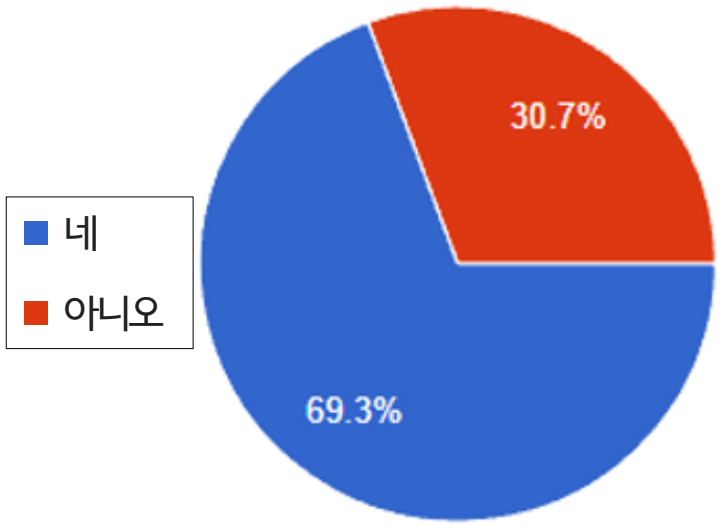
설문 기간: 2021. 10. 2.(토) - 2021. 10. 08.(금)
설문 대상: MZ세대 중 대학생을 포함한 20대 150명
설문 취지: 주식 거래 시 주로 활용하는 정보 및 증권앱이 무엇이며,
증권앱에서 바라는 점 및 불편한 점에 대한 조사

설문 정보

작성자의 나이를 알려주세요.



주식 거래 경험이 있으신가요?

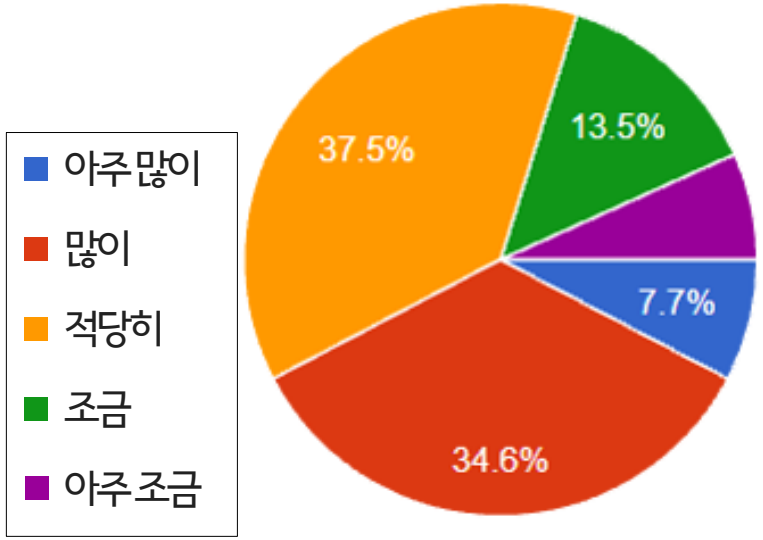


*주식 거래 경험의 유무에 따라 설문조사 질문을 다르게 구성함

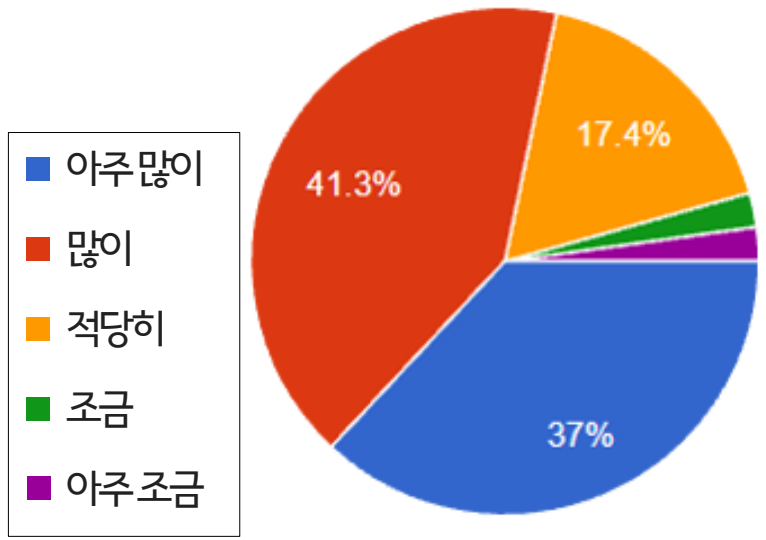
설문

주식 거래 시 온라인 상의 정보가 본인의 판단에 얼마나 영향을 미치나요/미칠까요?

[주식 거래 경험 有 응답]



[주식 거래 경험 無 응답]



- 주식 유경험자 (아주 많이+많이=42.3%) 에 비해 무경험자 (아주 많이+많이=78.3%) 의 응답에서 주식 거래 시 온라인 정보가 본인의 판단에 영향을 훨씬 크게 영향을 미칠 것이라는 답변을 확인
- 그러나 주식을 처음 접하는 MZ세대에게 온라인 상의 주식 정보를 제공하는 주식 정보 플랫폼은 부재함

2 아이디어 소개

M-able mini와 오해가 만나다

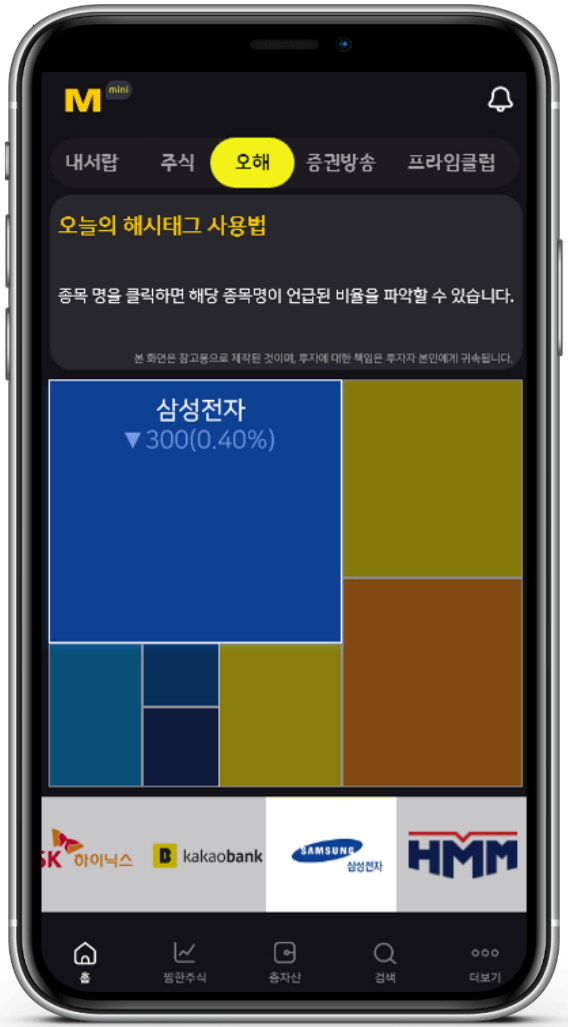
- MZ세대를 타겟으로 kb증권에서 런칭된
  MTS(모바일 트레이딩 시스템) M-able mini 앱에
  '오늘의 해시태그, 오해' 서비스 추가

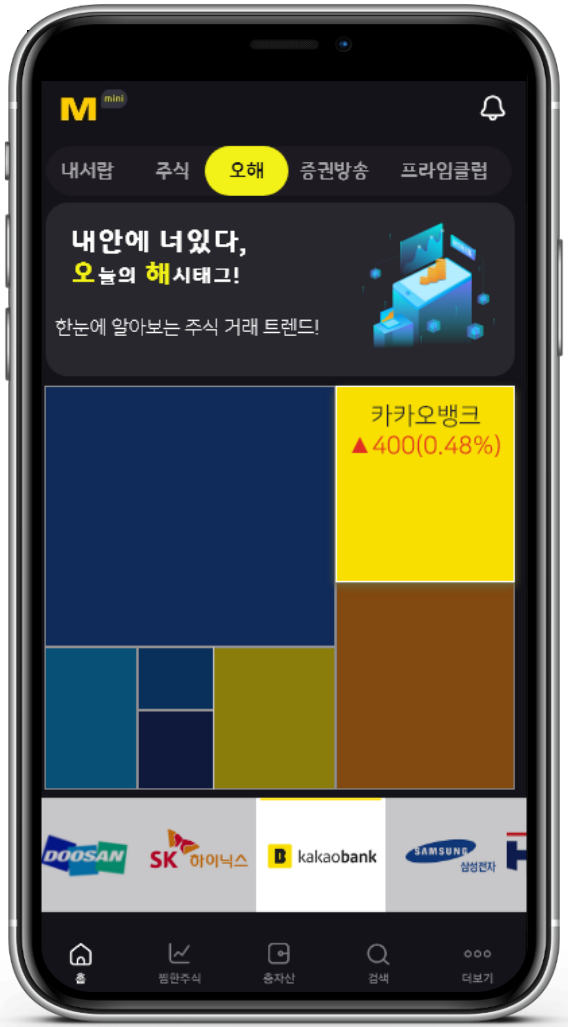
- 오늘의 해시태그, 오해는 5가지 데이터에서
  종목별로 같이 언급된 키워드를 빈도수로 카운트한 것
  오늘의 해시태그, 오해 = 해당 종목과 관련한 이슈 키워드

- 오해의 연관어까지 선정해서
  주식 종목에 대한 사람들의 오피니언을 분석하고
  주식 상승 / 주식 하락의 의미가 담긴 감성분석 서비스를 제공

일별 언급량으로 많이 언급된 종목이 무엇인지 한눈에!

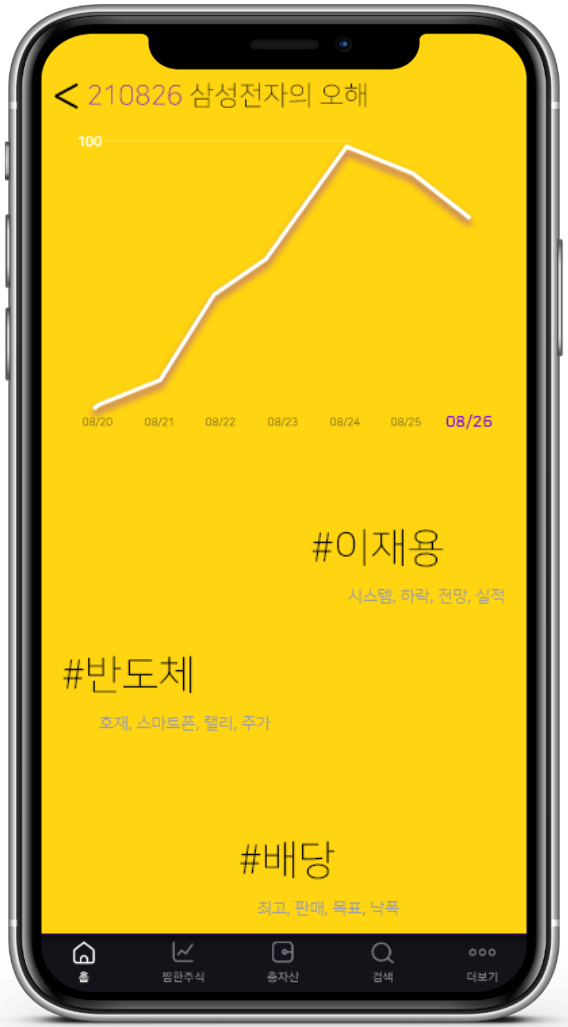
*데이터획득및언급량관련: 챕터3,4-1





종목에 대한 오해와, 오해에 따른 감성분석을 한눈에!

*오해선정및감성분석관련: 챕터4-2,4-3



3 데이터 확보

3-1. 선정 데이터

선정 데이터

오늘의 해시태그, 오해를 날마다 선택하기 위한 데이터 출처 선정 기준

1. 사람들이 주식 정보를 얻고자 대중적으로 이용하는가
2. 사람들이 주식 정보를 얻고자 의견을 나누는 '커뮤니티'가 형성되었는가
3. '하루 단위'로 데이터를 모았을 때 인사이트를 도출해낼 만큼의 양이 모여지는가

<크롤링 조건>

크롤링 일자 기준 지난주의 한국거래소 장마감 데이터 수집
예) 크롤링 일자가 8/16이라면 8/9(월)~8/13(금)의 장마감 데이터 수집해사용

시총 10조 이상 중 거래량 Top 10 종목을 선정, 해당 종목들을 언급한 게시글 수집
중복 종목 제외예) 삼성전자, 삼성전자우
인터넷에서 언급되는 다른 명칭까지 포함예) 삼성전자-삼전, SK하이닉스-하닉

크롤링날짜에따른종목반영날짜예시

선정 데이터

네이버뉴스	제목에 종목명이 언급된 네이버 뉴스 제목, 댓글 크롤링
다음뉴스	제목에 종목명이 언급된 다음 뉴스 제목, 댓글 크롤링
디시인사이드	제목에 종목명이 언급된 디시인사이드 주식갤러리 게시글 제목, 본문, 댓글 크롤링
종토방	제목에 종목명이 언급된 네이버 종토방 게시글 제목, 본문, 댓글 크롤링
유튜브	제목에 종목명이 언급된 유튜브 영상 제목, 댓글 크롤링

탈락 데이터

탈락된 데이터	대중적 이용도	커뮤니티 형성도	데이터 형성도	특이사항
페이스북, 인스타그램, 트위터 SNS	높음	낮음	높음	광고성 글 多
팍스넷 커뮤니티	낮음~보통	높음	낮음~보통	생각보다 글 업데이트가 느림
클리앙 커뮤니티	낮음~보통	높음	낮음~보통	생각보다 글 업데이트가 느림
에브리타임 커뮤니티	낮음	보통	낮음	대학생 전용 어플
블라인드 토픽 주식투자 커뮤니티	매우 낮음	보통	확인 불가	프라이빗 커뮤니티
가치투자연구소 네이버카페	보통	낮음~보통	낮음	생각보다 글 업데이트가 느림
함께하는 투자클럽 네이버카페	보통	낮음~보통	낮음	생각보다 글 업데이트가 느림

- 탈락된 데이터는 예상보다 글 발생 속도가 느리거나, 일부 조건을 충족해야만 열람할 수 있는 제한된 커뮤니티임
- 페이스북, 트위터와 같은 SNS에서 데이터 수집을 기대했으나 광고성 글이 너무 많으며, 주식 종목을 이야기하는 게시글은 소수에 불과

오해는 5가지 데이터로 구축 (1)

네이버뉴스

제목에 종목명이 언급된 네이버 뉴스 제목, 댓글 크롤링

```
def get_navernews_info(keyword):

    # 크롤링 정보(크롤링 날짜, 카테고리)
    yesterday = get_yesterday()
    categories = [259, 258, 261, 771, 310, 263]

    list_headline_concat    = []     # 모든 카테고리 제목
    list_address_concat     = []     # 모든 카테고리 주소 리스트
    list_comment_num_concat = []     # 모든 카테고리 댓글개수 리스트
    comment_headline_concat = []     # 모든 카테고리 댓글모을 제목 리스트
    comment_comment_concat  = []     # 모든 카테고리 제목별 댓글 리스트



    # 드라이버 실행
    driver = webdriver.Chrome('chromedriver.exe')
    driver.implicitly_wait(3)
    sleeptime = 0.5     # 프로세스 일시정지 시간



    # 카테고리를 돌며 크롤링
    for category in categories:
        page = 0
        list_headline = []     # 제목 리스트
        list_address  = []     # 주소 리스트
        top_headline = ''              # 페이지내 맨위 뉴스 제목
        top_headline_flag = False     # 다음 페이지 맨위 뉴스제목이 기존 페이지 맨위 뉴스제목과 다름
```

crawling_navernews.ipynb

- Selenium Webdriver 사용
- 크롤링 진행 날짜 기준 하루 전날 데이터 수집

- 네이버 경제 뉴스 중
  금융 - 증권 - 산업/재계 - 중기/벤처 - 생활경제 - 경제 일반
  카테고리에서 데이터 수집

오해는 5가지 데이터로 구축 (2)

다음뉴스

제목에 종목명이 언급된 다음 뉴스 제목, 댓글 크롤링

```
def get_daumnews_info(keyword):

    # 크롤링 정보(크롤링 날짜, 카테고리)
    yesterday = get_yesterday()
    categories = ['finance', 'industry', 'autos', 'stock', 'stock/market', 'stock/publicnotice', 'stock/

    list_headline_concat    = []    # 모든 카테고리 제목
    list_address_concat     = []    # 모든 카테고리 주소 리스트
    list_comment_num_concat = []    # 모든 카테고리 댓글개수 리스트
    comment_headline_concat = []    # 모든 카테고리 댓글모음 제목 리스트
    comment_comment_concat  = []    # 모든 카테고리 제목별 댓글 리스트


    # 드라이버 실행
    driver = webdriver.Chrome('chromedriver.exe')
    driver.implicitly_wait(3)
    sleeptime = 0.5     # 프로세스 일시정지 시간


    # 카테고리를 돌며 크롤링
    for category in categories:
        page = 0
        list_headline = []     # 제목 리스트
        list_address  = []     # 주소 리스트
        drop_address  = []     # 키워드 미포함 주소 리스트
        top_headline = ''              # 페이지내 맨위 뉴스 제목
        top_headline_flag = False     # 다음 페이지 맨위 뉴스제목이 기존 페이지 맨위 뉴스제목과 다를
```

crawling_daumnews.ipynb

- Selenium Webdriver, BeautifulSoup 사용
- 크롤링 진행 날짜 기준 하루 전날 데이터 수집

- 다음 경제 뉴스 중
  금융 - 기업산업 - 자동차 - 주식 -
  시황분석 - 공시 - 주식일반 - 생활경제
  카테고리에서 데이터 수집

오해는 5가지 데이터로 구축 (3)

디시인사이드

제목에 종목명이 언급된 디시인사이드 주식갤러리 제목,본문,댓글 크롤링

```
def get_dcinside_info(keyword):

    # 크롤링 정보(크롤링 날짜, 키워드 인코딩)
    sleeptime = 0.5     # 프로세스 일시정지 시간
    yesterday = get_yesterday()
    day_back = yesterday[5:]

    encoded_KEYWORD = urllib.parse.quote(keyword)
    num = 1



    title_ = []     # 제목 리스트
    url_   = []     # 주소 리스트
    time_  = []     # 게시날짜 리스트
    count_ = []     # 조회수 리스트

    # 최대 30페이지까지 크롤링
    while num <= 30:
        # 드라이버 실행
        driver = webdriver.Chrome('chromedriver.exe')
        time.sleep(sleeptime)



        # 디시인사이드 접속
        new_url = 'https://gall.dcinside.com/board/lists/?id=neostock&page='+str(num)+'&search_pos=&s_type=sea
        driver.get(new_url)
        time.sleep(sleeptime)
```

crawling_dcinside.ipynb

- Selenium Webdriver, BeautifulSoup 사용
- 크롤링 진행 날짜 기준 하루 전날 데이터 수집

- 디시인사이드 주식갤러리 내의 종목 게시판에서
  제목+본문 필터로 종목 이름을 검색한 결과 데이터 수집
  (너무 저급한 수준의 글을 사전에 방지하기 위함)

오해는 5가지 데이터로 구축 (4)

종토방

제목에 종목명이 언급된 네이버 종토방 게시글 제목, 댓글 크롤링

```
def get_jongto_info(keyword):

    # 크롤링 정보(크롤링 날짜, 종목코드)
    today, yesterday1 = get_yesterday()
    stockcode = pd.read_html('http://kind.krx.co.kr/corpgeneral/corpList.do?method=download', header=0)[0]
    stockcode = stockcode[['회사명', '종목코드']]
    stockcode = stockcode.rename(columns={'회사명': 'company', '종목코드': 'code'})
    stockcode.code = stockcode.code.map('{:06d}'.format)

    # 검색종목-검색어: 삼성전자-삼성전자, 삼성전자-삼전, 셀트리온-셀트
    code     = stockcode.loc[stockcode.company==keyword[0]]['code'].values.tolist()[0]     # 검색종목
    encoding = urllib.parse.quote(keyword[1], encoding='euc-kr')                           # 검색어



    # 드라이버 실행
    driver = webdriver.Chrome('chromedriver.exe')
    driver.implicitly_wait(3)
    sleeptime = 0.5     # 프로세스 일시정지 시간



    page    = 0
    page_y1 = 0         # 맨위 게시날짜가 어제인 페이지
    page_y2 = 0         # 맨위 게시날짜가 그저께인 페이지
    endpage = 0         # 더이상 보지않을 페이지
```

crawling_jongto.ipynb

- Selenium Webdriver 사용
- 크롤링 진행 날짜 기준 하루 전날 데이터 수집

- 한국거래소 홈페이지에서 6자리 종목코드를 받아와 종토방 페이지 url을 불러올 때 활용
- 네이버 종토방 내의 종목 게시판에서 제목으로 종목 이름을 검색한 결과 데이터 수집 (너무 저급한 수준의 글을 사전에 방지하기 위함)

오해는 5가지 데이터로 구축 (5)

유튜브	제목에 종목명이 언급된 유튜브 영상 제목, 댓글 크롤링

```
def get_youtube_info(keyword):

    # 드라이버 실행
    driver = webdriver.Chrome('chromedriver.exe')
    driver.implicitly_wait(3)
    sleeptime = 0.5     # 프로세스 일시정지 시간



    # 유튜브 접속
    keyword = keyword + ' 주가'
    driver.get(f'https://www.youtube.com/results?search_query={keyword}')
    time.sleep(sleeptime)



    # 영상 순서를 '업로드 날짜'로 변경
    driver.find_element_by_css_selector('#container > ytd-toggle-button-renderer').click()
    driver.find_element_by_xpath('/html/body/ytd-app/div/ytd-page-manager/ytd-search/div[1]/ytd-two-colu



    # 스크롤 최대한 내려서 파싱
    body = driver.find_element_by_tag_name('body')
    num  = 0
    while num < 10:
        # 현재 화면길이를 리턴받아 last_height에 넣음
        last_height = driver.execute_script('return document.documentElement.scrollHeight')
```

crawling_youtube.ipynb

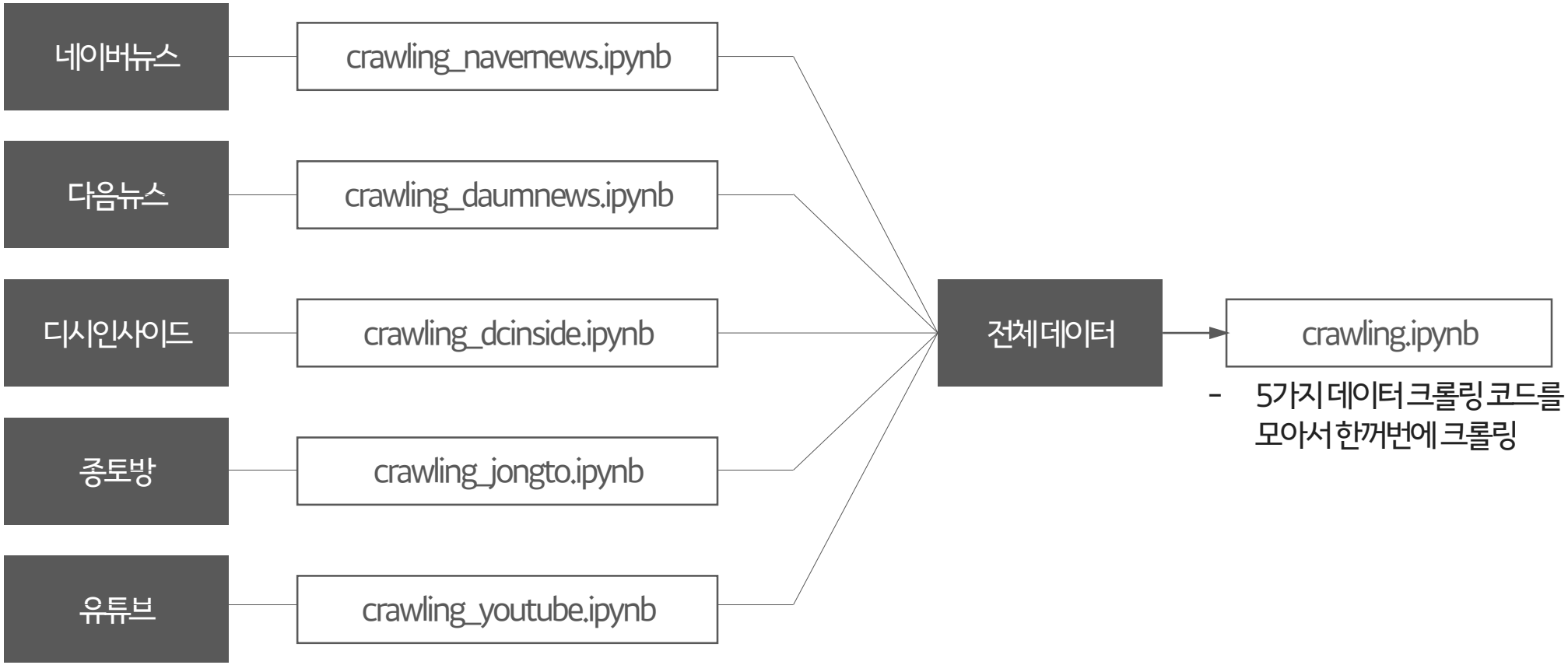
- Selenium Webdriver, BeautifulSoup 사용
- 크롤링 진행 날짜 기준 3일 전까지의 데이터 수집

- 유튜브 영상에서
  제목으로 '종목 이름+주가'를 검색한 결과 데이터 수집
  (주식과 관련 없는 영상을 사전에 방지하기 위함)

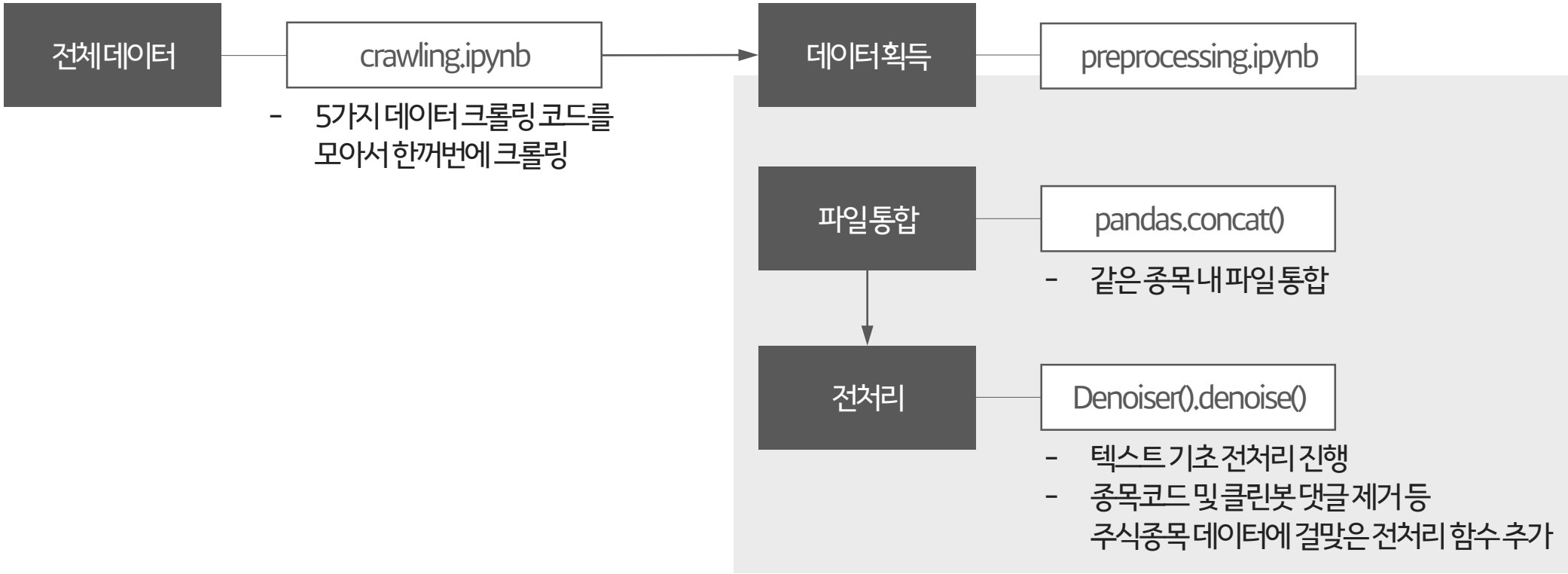
3 데이터 확보

3-2. 데이터 구축 프로세스

오해는 5가지 데이터로 구축



오해는 5가지 데이터로 구축



4 모델링

4-1. 언급량 측정

언급량 측정

<네이버 뉴스>

<디시인사이드 주식게시판>

<다음 뉴스>

<네이버 종토방>

<유튜브>

언급량 측정

- 같은 주식 종목을 언급하더라도 글의 신뢰도와 접근성을 이유로
  '뉴스의 제목'에 언급된 것과, '커뮤니티 댓글'에 언급된 것에는 같은 영향을 미치지 않는다고 판단
- 다음뉴스 / 디시인사이드 주식갤러리 / 네이버 종토방 / 네이버뉴스 / 유튜브 각각에서
  제목, 본문, 댓글에 종목명이 언급될 때, 언급량을 카운트하는 기준을 달리 세움으로써 디테일하게 언급량을 측정

*모든상황에서종목명언급없는경우각스코어의50%만반영

예) 다음뉴스제목에종목명이언급되면언급량+=0.5
종토방댓글에종목명이언급되면언급량+=0.1
유튜브제목에종목명이언급되지않으면언급량+=0.5*50%

언급량 측정

```python
# 분석할 날짜 입력
period = ['20210822', '20210823', '20210824', '20210825', '20210826']
```

1. 불러올 날짜 입력

```python
# 네이버뉴스 언급량 분석
datafrom = 'navernews'

for date in period:
    # 언급량 변수
    globals()[f'mentioned_{datafrom}_{date}'] = []

    for i in range(len(vocab)):
        # 종목 선택
        keyword = vocab[i][0]

        # 데이터프레임 불러오기
        globals()[f'{datafrom}_{date}_{keyword}_final_info'] = pd.read_csv(f'./{date}_data/{date}_{datafrom}/{datafrom}_{date}_{keyword}_
        globals()[f'{datafrom}_{date}_{keyword}_final_comment'] = pd.read_csv(f'./{date}_data/{date}_{datafrom}/{datafrom}_{date}_{keywor

        # 데이터프레임 빈 부분 채우기
        globals()[f'{datafrom}_{date}_{keyword}_final_info'].loc[globals()[f'{datafrom}_{date}_{keyword}_final_info']['제목'].isnull(), '
        globals()[f'{datafrom}_{date}_{keyword}_final_comment'].loc[globals()[f'{datafrom}_{date}_{keyword}_final_comment']['제목'].isnul
        globals()[f'{datafrom}_{date}_{keyword}_final_comment'].loc[globals()[f'{datafrom}_{date}_{keyword}_final_comment']['댓글'].isnul

        # 언급량 계산
        keywords = vocab[i]
        info = globals()[f'{datafrom}_{date}_{keyword}_final_info']
        comment = globals()[f'{datafrom}_{date}_{keyword}_final_comment']
        globals()[f'mentioned_{datafrom}_{date}_{keyword}'] = Analyzer().estimate_mentioned1(datafrom, info, comment, keywords)
        globals()[f'mentioned_{datafrom}_{date}'].append([keyword, globals()[f'mentioned_{datafrom}_{date}_{keyword}']])
        #print(f' {date} {datafrom} {keyword} 언급량 분석 완료')
```

2. 전처리된 파일을 불러와 언급량 계산

다음뉴스,디시인사이드,종토방,네이버뉴스,유튜브에서반복

```python
# 언급량 측정: 네이버뉴스, 다음뉴스, 유튜브
def estimate_mentioned1(self, datafrom, info, comment, keywords):
    mentioned_headline = 0
    mentioned_comment = 0

    # 데이터 출처에 따른 mention score 지정
    if datafrom == 'navernews':
        mention_score = self.mention_score_navernews
    elif datafrom == 'daumnews':
        mention_score = self.mention_score_daumnews
    elif datafrom == 'youtube':
        mention_score = self.mention_score_youtube

    # 언급량 카운트
    for keyword in keywords:
        # 제목에서 언급량
        keyword_count = 0
        for i in range(len(list(info['제목']))):
            if keyword in list(info['제목'])[i]:
                keyword_count += 1
            else:
                keyword_count += 0.5
        mentioned_headline += mention_score[0] * keyword_count

        # 댓글에서 언급량
        keyword_count = 0
        for i in range(len(list(comment['댓글']))):
            if keyword in list(comment['댓글'])[i]:
                keyword_count += 1
            else:
                keyword_count += 0.5
        mentioned_comment += mention_score[1] * keyword_count
    return mentioned_headline + mentioned_comment
```

3. Analyzer() 클래스 메서드 estimate_mentioned() 이용

언급량 측정

```
# 날짜별 언급량 통합
for date in period:
    globals()[f'mentioned_{date}'] = globals()[f'mentioned_daumnews_{date}']
    for i in range(len(vocab)):
        globals()[f'mentioned_{date}'][i][1] = float(globals()[f'mentioned_{date}'][i][1])

    for datafrom in ['dcinside', 'jongto', 'navernews', 'youtube']:
        for i in range(len(vocab)):
            globals()[f'mentioned_{date}'][i][1] += float(globals()[f'mentioned_{datafrom}_{date}'][i][1])
```

```
mentioned_20210822      # 20210822 5개 데이터에서 각 종목 언급량
```

```
[['삼성전자', 424.95000000000005],
 ['SK하이닉스', 75.25],
 ['카카오뱅크', 155.4],
 ['두산중공업', 72.35000000000001],
 ['HMM', 258.40000000000003],
 ['SK바이오사이언스', 95.30000000000001],
 ['한국전력공사', 61.95],
 ['카카오', 135.25],
 ['대한항공', 15.95]]
```

```
mentioned_20210823      # 20210823 5개 데이터에서 각 종목 언급량
```

```
[['삼성전자', 389.75],
 ['SK하이닉스', 125.80000000000001],
 ['카카오뱅크', 217.25],
 ['두산중공업', 90.3],
 ['HMM', 674.35],
 ['SK바이오사이언스', 160.85000000000002],
 ['한국전력공사', 60.85],
 ['카카오', 157.45],
 ['대한항공', 37.0]]
```

날짜별 언급량

```
# 종목별 언급량 통합
for date in period:
    for i in range(len(vocab)):
        keyword = vocab[i][0]

        if date == period[0]:
            globals()[f'mentioned_{keyword}_{period[0]}_to_{period[4]}'] = []
        globals()[f'mentioned_{keyword}_{period[0]}_to_{period[4]}'].append(globals()[f'mentioned_{date}'][i][1])
```

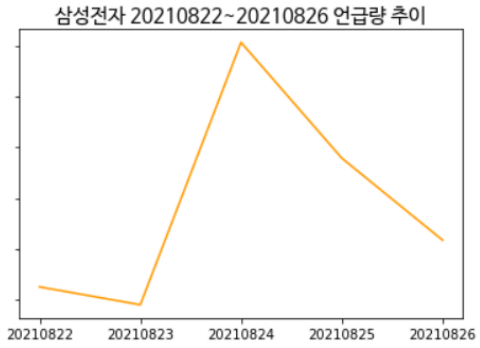
```
mentioned_삼성전자_20210822_to_20210826      # 20210822 부터 20210826 까지 5개 데이터에서 삼성전자 언급량
```

```
[424.95000000000005, 389.75, 907.25, 678.95, 517.3000000000001]
```

언급량 시각화

```
def visualize_mentioned_keyword_per_period(mentioned_keyword_period, keyword, period):
    font_path = './NanumBarunGothic.ttf'
    fontprop = fm.FontProperties(fname=font_path, size=10)

    plt.title(f'{keyword} {period[0]}~{period[4]} 언급량 추이', fontproperties=fontprop, fontsize=15)
    plt.plot(period, mentioned_keyword_period, color='#FF9B00')
    plt.yticks(color='w')
    plt.show()
```

```
visualize_mentioned_keyword_per_period(mentioned_삼성전자_20210822_to_20210826, '삼성전자', period)
```



종목별 언급량

언급량 측정

텍스트 분석 — analyzing.ipynb

언급량 측정

- Analyzer().estimate_mentioned1
  - (본문이 없는) 네이버뉴스, 다음뉴스, 유튜브에서 언급량 측정
- Analyzer().estimate_mentioned2
  - (본문이 있는) 디시인사이드, 종토방에서 언급량 측정

4 모델링

4-2. 오늘의 해시태그, 오해 선정

오늘의 해시태그, 오해 선정

<종목에 대해 오늘의 해시태그, 오해를 보여주는 모습(왼쪽)>

- 종목별 각 오해가 며칠 동안 지속되는지,
  어제 대비 오늘의 언급량이 어떤지 시각화
- 데이터 전체에서 빈도수 카운트로 오해를 선정함
- 현재 챕터 4-2에서 언급할 내용

<오늘의 해시태그, 오해의 연관어를 보여주는 모습(오른쪽)>

- 각 오해 별로 같이 언급된 연관어가 무엇인지 시각화
- 오해의 연관어가 언급된 텍스트에서
  감성분석한 결과 역시 다른 화면에서 보여줄 예정
- 다음 챕터 4-3에서 언급할 내용

오늘의 해시태그, 오해 선정

```
# 분석할 날짜 입력
period = ['20210822', '20210823', '20210824', '20210825', '20210826']
```

1. 불러올 날짜 입력

오늘의 해시태그, 오해 선정

```
for i in range(len(vocab)):
    # 종목 선택
    keyword = vocab[i][0]

    # 데이터 칼럼 추출
    globals()[f'datacolumns_{targetdate}_{keyword}'] = Analyzer().select_datacolumns(targetdate, keyword)

    # 오늘의 해시태그, 오해 선정
    globals()[f'OHAE_{targetdate}_{keyword}'] = Analyzer().select_OHAE(globals()[f'datacolumns_{targetdate}_{keyword}'])
```

```
# 날짜별 데이터 통합
def select_datacolumns(self, date, keyword):
    datafrom = self.datafrom
    datacolumns = []

    # 같은 날짜에서 5개 데이터 통합
    for i in range(len(datafrom)):
        if (datafrom[i] == 'youtube') and (keyword == '한국전력공사'):
            keyword = '한국전력'

        globals()[f'info{i}'] = globals()[f'{datafrom[i]}_{date}_{keyword}_final_info']
        globals()[f'comment{i}'] = globals()[f'{datafrom[i]}_{date}_{keyword}_final_comment']

        if (datafrom[i] == 'navernews') or (datafrom[i] == 'daumnews') or (datafrom[i] == 'youtube'):
            datacolumns.append(list(globals()[f'info{i}']['제목']))
            datacolumns.append(list(globals()[f'comment{i}']['댓글']))
        elif (datafrom[i] == 'dcinside') or (datafrom[i] == 'jongto'):
            datacolumns.append(list(globals()[f'info{i}']['제목']))
            datacolumns.append(list(globals()[f'info{i}']['본문']))
            datacolumns.append(list(globals()[f'comment{i}']['댓글']))
    return datacolumns
```

2. 전처리된 파일을 불러와 데이터 칼럼 통합

다음뉴스,디시인사이드,종토방,네이버뉴스,유튜브에서반복

```
# 오늘의 해시태그, 오해 선정
def select_OHAE(self, datacolumns):
    word2index = {}
    bow = []

    # 데이터프레임을 돌며 단어 수집
    for datacolumn in datacolumns:
        for headline in datacolumn:
            token = Okt().nouns(headline)
            for voca in token:
                if voca in OHAE_stopwords or voca.isdigit():
                    continue

                if voca not in word2index:
                    word2index[voca] = len(word2index)
                    bow.insert(len(word2index)-1, 1)
                else:
                    index = word2index[voca]
                    bow[index] = bow[index]+1

    # 단어 빈도수 리스트 형성
    word_count = []
    for i in range(len(bow)):
        word = list(word2index.keys())[i]
        count = bow[i]
        word_count.append([count, word])
    word_count.sort(reverse=True)
    return word_count[:5]
```

3. Analyzer() 클래스 메서드 select_OHAE() 이용

오늘의 해시태그, 오해 선정

```
OHAE_20210826_삼성전자
```

```
[[89, '반도체'], [57, '이재용'], [52, '배당'], [51, '화이팅'], [48, '기업']]
```

```
OHAE_20210826_삼성전자 = [OHAE_20210826_삼성전자[0][1], OHAE_20210826_삼성전자[1][1], OHAE_20210826_삼성전자[2][1]]
OHAE_20210826_삼성전자
```

```
['반도체', '이재용', '배당']
```

20210826일자삼성전자의오해

```
OHAE_20210826_SK하이닉스
```

```
[[29, '이사'], [23, '네이버'], [18, '반도체'], [16, '전망'], [13, '합병']]
```

```
OHAE_20210826_SK하이닉스 = [OHAE_20210826_SK하이닉스[0][1], OHAE_20210826_SK하이닉스[1][1], OHAE_20210826_SK하이닉스[2][1]]
OHAE_20210826_SK하이닉스
```

```
['이사', '네이버', '반도체']
```

20210826일자SK하이닉스의오해

```
OHAE_20210826_카카오뱅크
```

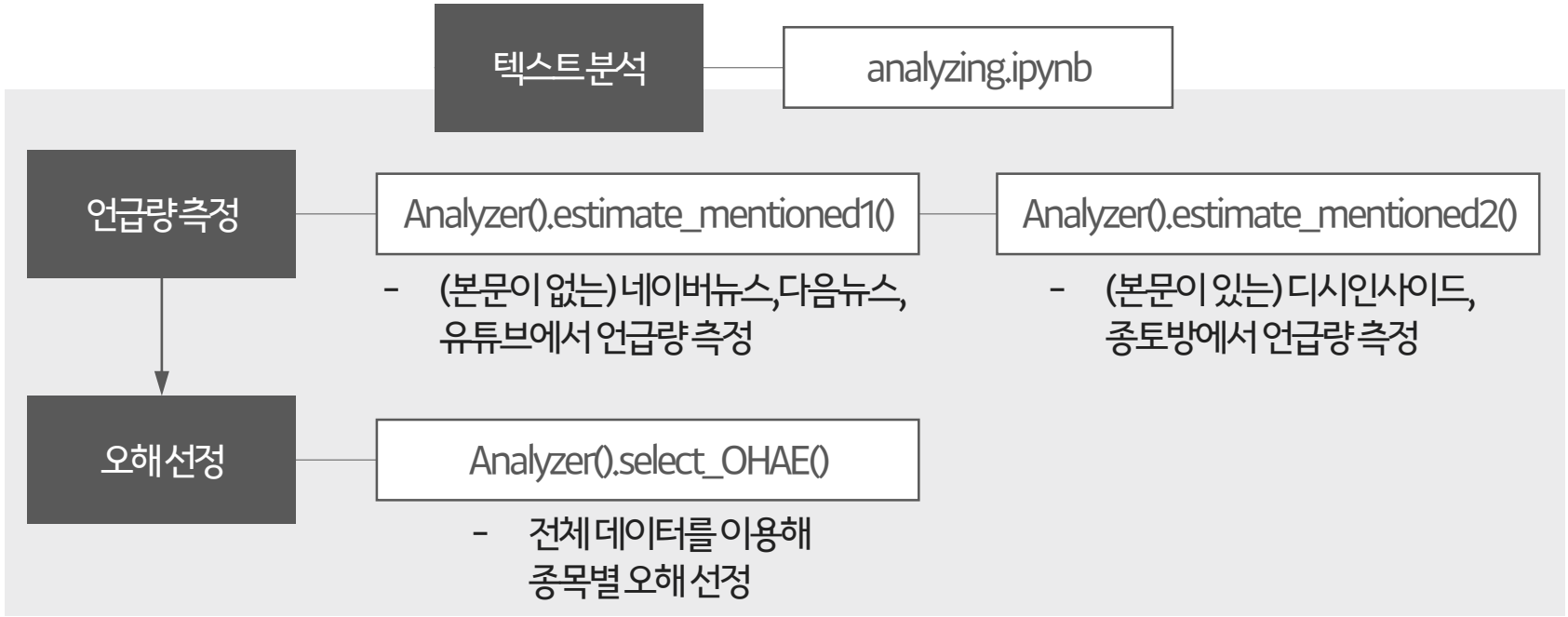
```
[[42, '대출'], [32, '금융'], [29, '은행'], [27, '전망'], [16, '기업']]
```

```
OHAE_20210826_카카오뱅크 = [OHAE_20210826_카카오뱅크[0][1], OHAE_20210826_카카오뱅크[1][1], OHAE_20210826_카카오뱅크[2][1]]
OHAE_20210826_카카오뱅크
```

```
['대출', '금융', '은행']
```

20210826일자카카오뱅크의오해

오늘의 해시태그, 오해 선정



4 모델링

4-3. kb-albert를 이용한 감성분석 모델링

kb-albert를 이용한 감성분석 모델링

- 20210816 ~ 20210825 총 10일 동안
  다음뉴스, 디시인사이드, 종토방, 네이버뉴스, 유튜브를 크롤링하며
  감성분석 모델링에 사용할 텍스트 데이터 수집

- 사용자의 의견이 직접적으로 반영되지 않은 뉴스 제목을 제외하고는
  모든 텍스트에 감성분석 라벨링을 진행하였으며, 해당 라벨은 아래와 같음
  <주가상승의미:1, 주가하락의미:0, 의미가없거나변별하기힘듦:2>

- 라벨링된 10일치 데이터(train)와 kb-albert를 이용하여
  주식 종목에 대한 사람들 의견을 대상으로 감성분석 모델 학습을 진행

- 발표자료 뒤에 언급될 모델 활용(validation)은 20210826 데이터를 이용

kb-albert를 이용한 감성분석 모델링

감성분석학습 — modeling.ipynb

- 라벨링된 10일치 데이터와 kb-albert를 이용하여 감성분석 모델 학습 진행

```
comment_classifier = pipeline('sentiment-analysis', model=model, tokenizer=tokenizer, framework='pt')

comments = ['떡락각이네여',
            '다들 왜 아직도 안나옴,, 무조건 손절',
            '진짜 삼전 언제 까지 떨어질지...',
            '하닉 이제 10만 가즈아',
            '수익률 보장인 주식이네여',
            '백퍼 떡상']
comments = comment_classifier(comments)
for comment in comments:
    print(f"label: {comment['label']}, with score: {round(comment['score'], 4)}")
```

```
label: negative, with score: 0.9466
label: negative, with score: 0.7685
label: negative, with score: 0.9294
label: positive, with score: 0.9719
label: positive, with score: 0.8608
label: positive, with score: 0.8968
```

떡락각이네요	Negative
다들 왜 아직도 안나옴,, 무조건 손절	Negative
진짜 삼전 언제 까지 떨어질지…	Negative
하닉 이제 10만 가즈아	Positive
수익률 보장인 주식이네여	Positive
백퍼 떡상	Positive

라벨링된 10일치 데이터와 kb-albert를 이용한 모델 출력 예시

kb-albert를 이용한 감성분석 모델링 활용

```
targetdate = '20210826'
```

1. 분석할 날짜 입력

```
# 오해 연관어를 추출할 네이버뉴스, 다음뉴스, 유튜브 제목/댓글 합산
for datafrom in ['navernews', 'daumnews', 'youtube']:
    for i in range(len(vocab)):
        cnt = 1
        keyword = vocab[i][0]
        if datafrom == 'youtube' and keyword == '한국전력공사':
            keyword = '한국전력'

        # 데이터프레임 칼럼 변경
        info = globals()[f'{datafrom}_{targetdate}_{keyword}_final_info']
        comment = globals()[f'{datafrom}_{targetdate}_{keyword}_final_comment']
        integrate = pd.DataFrame(index=range(len(info)+len(comment)))
        integrate['id'] = np.arange(len(integrate))
        integrate['document'] = list(info['제목']) + list(comment['댓글'])
        integrate['label'] = 0

        # 데이터프레임 통합
        if cnt == 1:
            globals()[f'전체2_{targetdate}_{keyword}'] = integrate
        else:
            globals()[f'전체2_{targetdate}_{keyword}'] = pd.concat([globals()[f'전체2_{targetdate}_{keyword}'], integrate], ignore_inde
        cnt += 1

# 오해 연관어를 추출할 디시인사이드, 종토방 제목/본문/댓글 합산
for datafrom in ['dcinside', 'jongto']:
    for i in range(len(vocab)):
        keyword = vocab[i][0]

        # 데이터프레임 칼럼 변경
        info = globals()[f'{datafrom}_{targetdate}_{keyword}_final_info']
        comment = globals()[f'{datafrom}_{targetdate}_{keyword}_final_comment']
        integrate = pd.DataFrame(index=range(len(info)*2+len(comment)))
        integrate['id'] = np.arange(len(integrate))
```

2. 감성분석 진행할 데이퍼프레임 통합

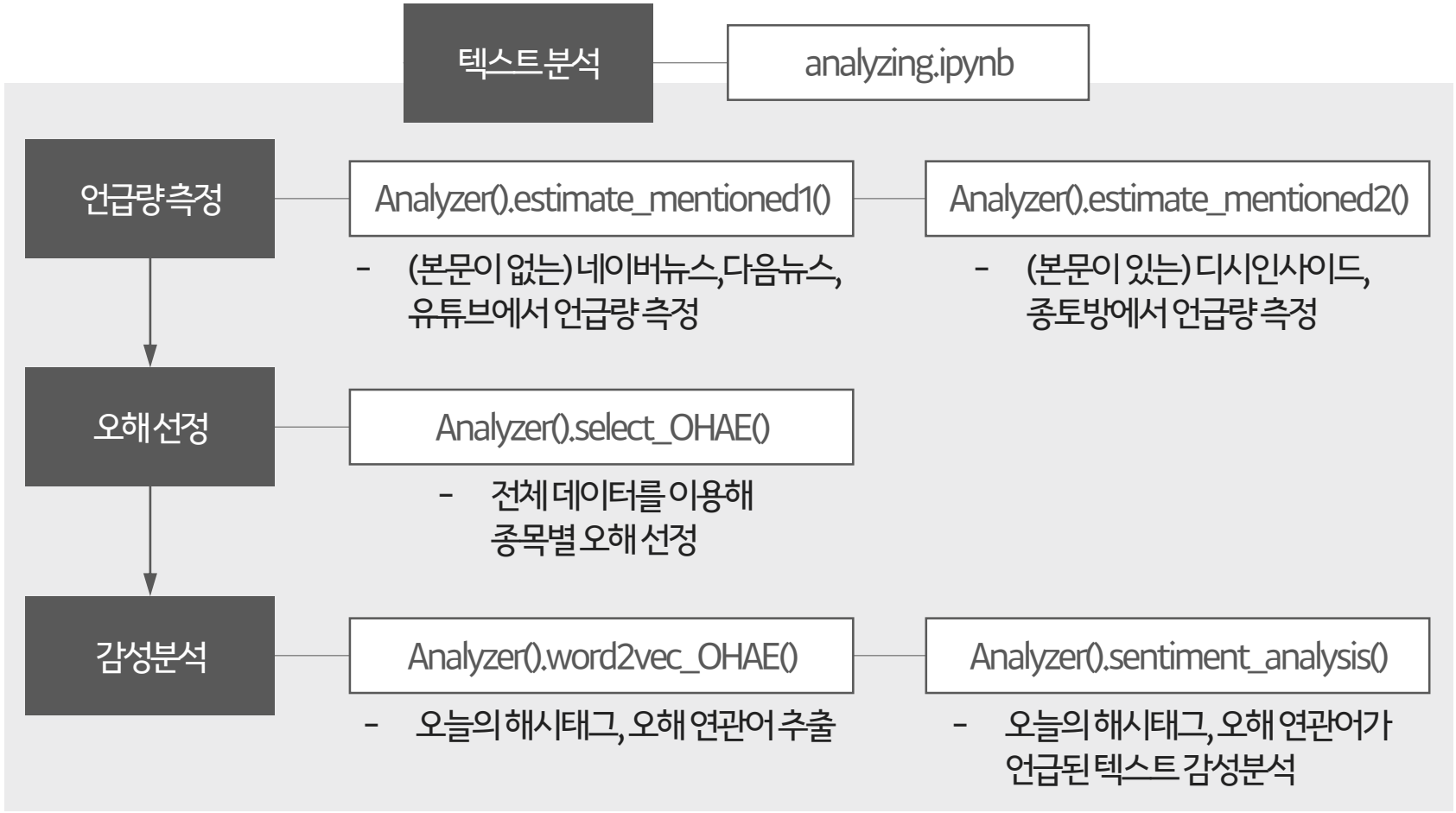
3. 오늘의 해시태그, 오해 연관어 추출

```
# 오늘의 해시태그, 오해 연관어 추출
def word2vec_OHAE(self, OHAE, dataframe):
    print('----------'*5)
    print(OHAE, '이(가) 포함된 텍스트에 벡터라이징 적용')
    text_with_OHAE = self.data_extraction(dataframe, OHAE)

    # 불용어 및 제외어 정의
    stopwords = ['삼','전','성','등','되다','전자','삼성','삼전','있다','에서','이다','을','기','의','가','이','은','들'
                 '는','좀','잘','걍','식','과','월','도','임','20만원','배','를','으로','자','에','와','한','하다',
                 '더','게','것','*','1','2','3','4','5','6','7','8','g','크다','아니다','지금','대비','때','율',
                 '위','올해','연말','분기','특별']
    removewords = ['향후', '종목', '및', '이유','저가','고가']

    # 토큰화
    tokenized_data = []
    for sentence in text_with_
        temp_X = Okt().nouns(s
        temp_X = [word for wor
        tokenized_data.append(

    # 토큰을 임베딩하여 벡터라
    model = Word2Vec(sentences

    # 코사인 유사도를 기준으로
    OHAE_word2vec_dict = model
    print('----------'*5)
    print('벡터라이징 결과(코사
    print(OHAE_word2vec_dict)
```

```
# 오늘의 해시태그, 오해 연관어 감성분석
def sentiment_analysis(self, OHAE_word2vec_list, dataframe):
    OHAE_top4_relationword = OHAE_word2vec_list[:4]
    print('----------'*5)
    print('연관어 4개: ', OHAE_top4_relationword)
    print('----------'*5)

    for 연관어 in OHAE_top4_relationword:
        # 오해 단어의 연관어가 포함된 텍스트만 추출
        OHAE_top4_Text = self.data_extraction_for_classifier(dataframe, 연관어)

        # 학습된 albert모델 불러와 사용
        kb_albert_model_path = './model'
        model_output_path = './OHAE_project/model_output'
        model = AutoModelForSequenceClassification.from_pretrained(model_output_path)
        # kbalberrCharTokenizer을 이용해 텍스트 토큰화
        tokenizer = KbAlbertCharTokenizer.from_pretrained(kb_albert_model_path)

        # 10일치의 텍스트가 학습된 감성분석 모델을 사용해 모든 텍스트에 대해서 positive/negative 예측
        comment_classifier = pipeline('sentiment-analysis', model=model, tokenizer=tokenizer, framework='pt')
        comments = OHAE_top4_Text

        # 감성분석 진행
        comments = comment_classifier(comments)
        positive = 0
        negative = 0
        total = len(comments)
        for comment in comments:
            if comment['label'] == 'positive':
                positive += 1
            if comment['label'] == 'negative':
                negative += 1
```

4. 오늘의 해시태그, 오해 연관어가 언급된 텍스트 감성분석

kb-albert를 이용한 감성분석 모델링 활용



5 확장

5-1. (주식을 하는) 현재 투자자의 관점

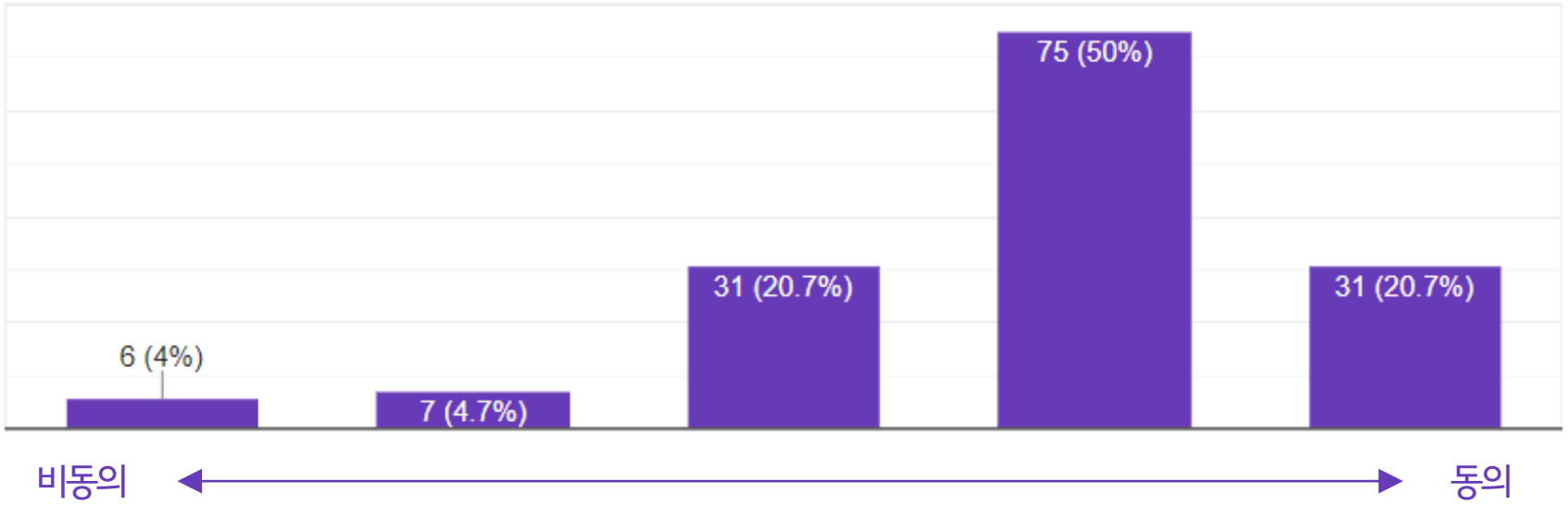
현재 투자자에게 오피니언 마이닝 제공

- 현재 kb증권 M-able mini에서 종목의 매도 / 매수 동향을 파악하는 방법은 두가지
  (1) 투자자 동향: 기간 중 누적 거래량 지표
  (2) 투자 의견: 증권사 별 투자 의견 및 목표가
- 오늘의 해시태그, 오해는 종목별 언급량 추이를 가시적으로 표현 가능
- 오늘의 해시태그, 오해 연관어 추출로 종목과 관련한 최근 이슈 파악 가능
- 오늘의 해시태그, 오해 연관어 감성분석으로 투자자들의 동향 파악 가능
- 자본시장에서 개인 투자자의 점유율이 갈수록 높아지는 만큼
  오피니언 마이닝으로 주식 정보를 활용하는 것이 앞으로도 중요 task가 될 것
- 오늘의 해시태그, 오해는 이러한 흐름의 선두주자로 출발!

현재 투자자에게 오피니언 마이닝 제공

OHAE 서비스가 올바른 주식 정보 판단에 도움이 될 것 같나요?

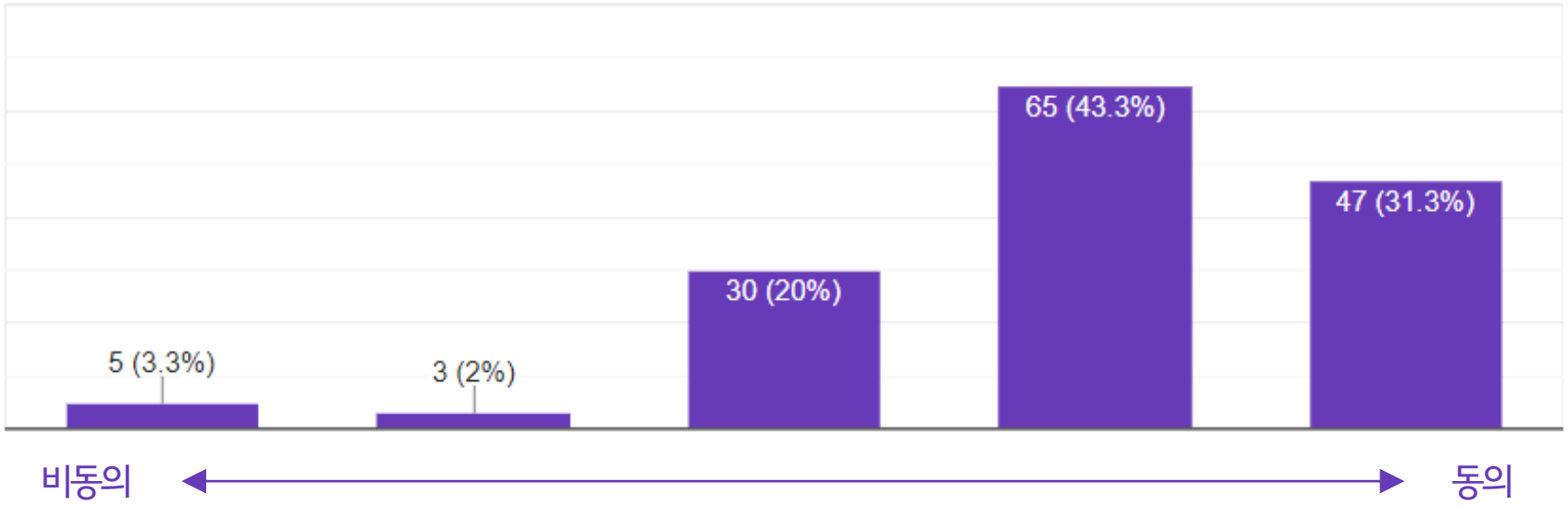
[주식 거래 경험 有/無 동시 응답]



현재 투자자에게 오피니언 마이닝 제공

OHAE 서비스가 기존 증권앱에서 보지 못했던 새로운 유형의 서비스인가요?

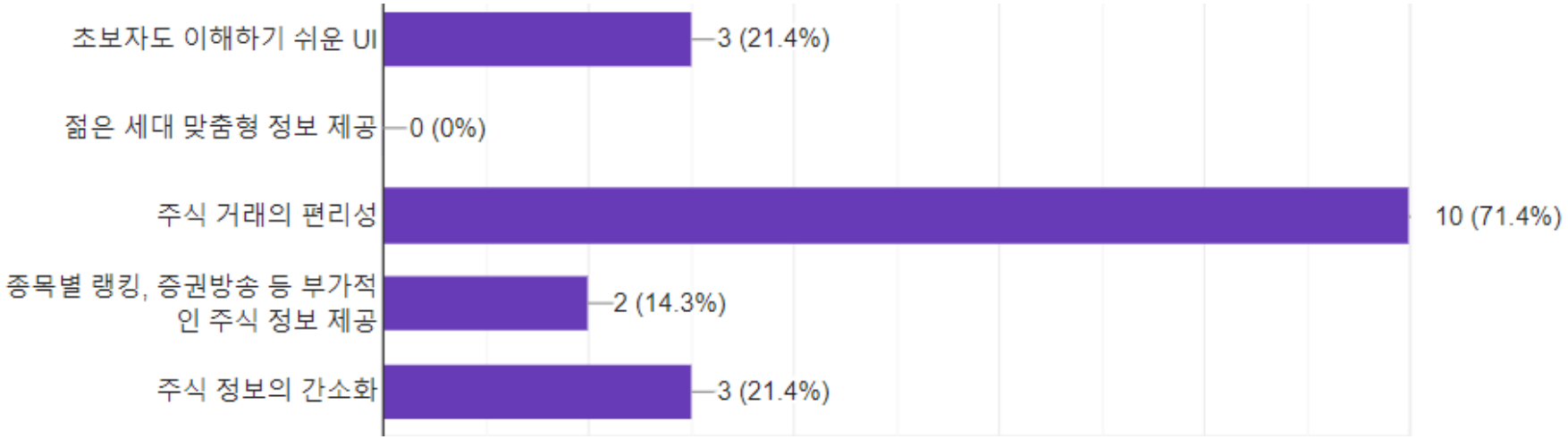
[주식 거래 경험 有/無 동시 응답]



현재 투자자에게 오피니언 마이닝 제공

KB증권 M-able mini를 사용해 보셨다면, 어떤 점으로 인해 사용하셨나요?

[주식 거래 경험 有 응답]



- 설문조사 결과, M-able mini를 사용해본 사람들은 '주식 거래의 편리성' 을 이유로 사용하였음
- 그 외의 이유중에서는 '부가적인 주식 정보 제공' 등이 적게 선택 되었으며 특히 '젊은 세대 맞춤형 정보 제공' 은 한번도 선택 받지 못함
  현재 M-able mini는 이같은 서비스를 투자자들에게 와닿게 제공하지 못하는 것으로 보임
- 이러한 관점에서 OHAE 서비스는 젊은 세대에게 시간의 흐름에 따라
  빠르게 변화하는 커뮤니티 정보를 오피니언 마이닝을 통해 한눈에 제공할 수 있을 것임

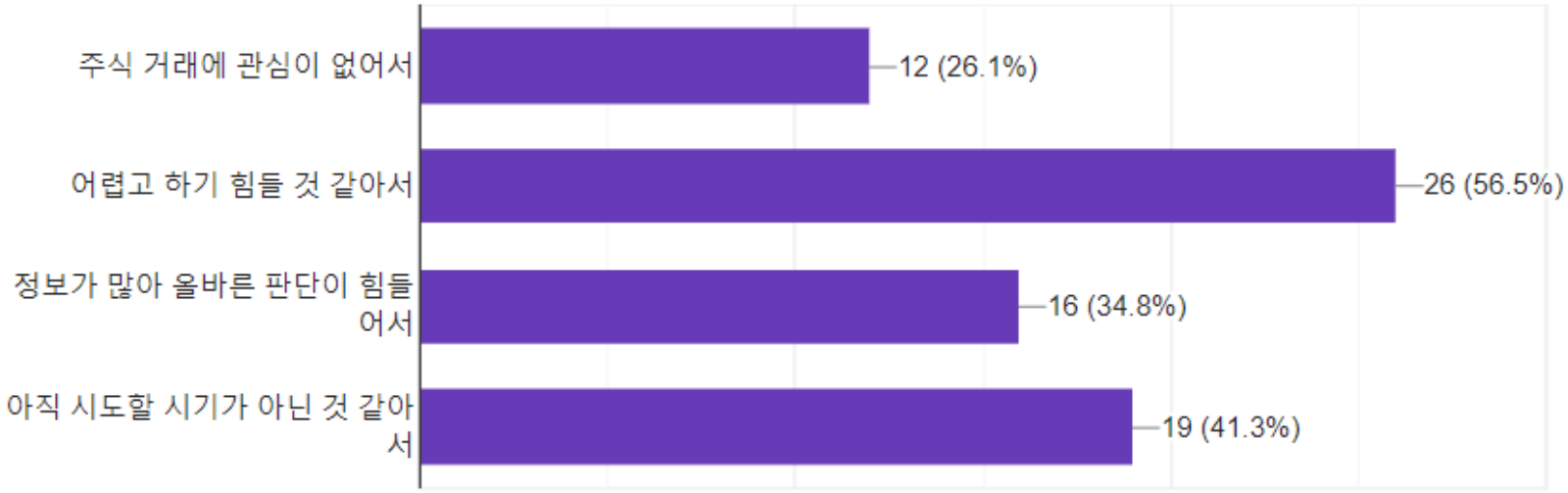
5 확장

5-2. (주식을 하지 않는) 미래 투자자의 관점

미래 투자자의 심리적 장벽 최소화

주식 거래를 시도하지 않은 이유는 무엇인가요?

[주식 거래 경험 無 응답]



- 설문조사 결과, 아직 주식 거래를 해보지 않은 사람들의 결정적인 이유는 '어렵고 하기 힘들 것 같아서' 로 확인되었음
- 주식 거래에 대한 심리적 장벽을 최소화한다면 주식 거래를 시도하는 사람들이 늘어날 것이며,
  그에 따라 증권앱으로의 유입이 증가할 것으로 예상됨

미래 투자자의 심리적 장벽 최소화

- MZ세대 및 2030세대 주식 투자자를 끌어들이고자
  M-able mini에 '오늘의 해시태그, 오해'를 추가 제공

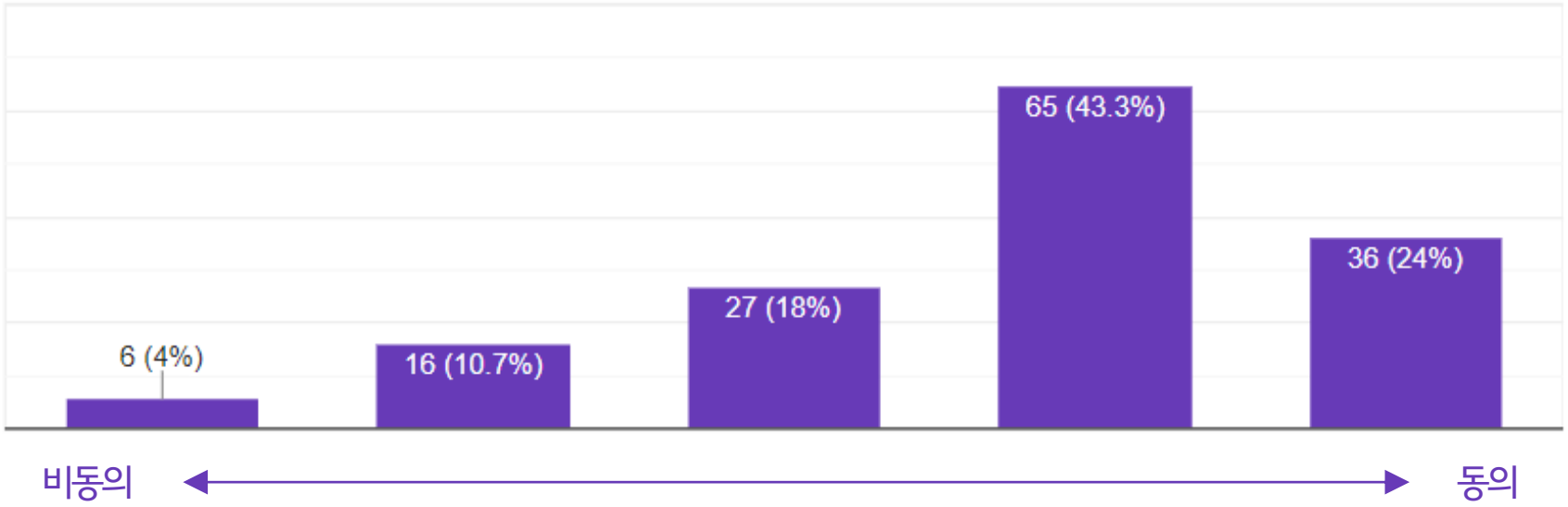
- 날마다 달라지는 종목별 언급량으로
  어느 종목이 얼마나 언급되었는지 한눈에 확인!
  오늘의 해시태그, 오해로 종목별 이슈 키워드를 한눈에 확인!
  오늘의 해시태그, 오해 연관어 감성분석 결과로 사람들 반응 한눈에 확인!
- 이처럼 오해를 반영한 오피니언 마이닝으로 보기 쉽고 간단한 증권 정보 제공

- 현재 젊은 층의 많은 관심을 받는 토스증권, 카카오페이증권에 몰린
  MZ세대 및 2030세대 미래 투자자의 이목을 충분히 이끌 것으로 예상

미래 투자자의 심리적 장벽 최소화

OHAE 서비스가 기존 증권앱에 런칭된다면 사용할 의사가 있으신가요?

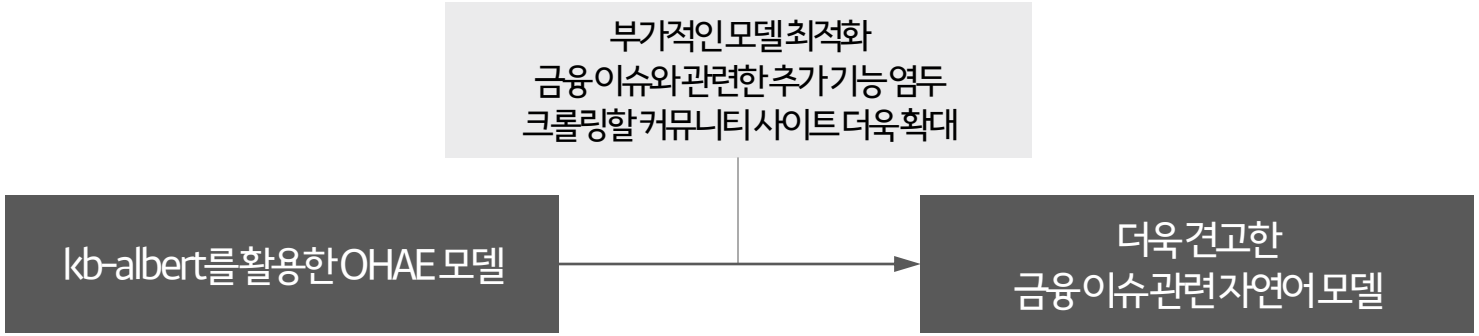
[주식 거래 경험 有/無 동시 응답]



5 확장

5-3. 자연어 모델의 관점

금융 이슈 관련 자연어 모델의 지속적 구축



- 현재 kb-albert를 활용한 OHAE 모델은 주식 종목에 대한 반응이 즉각적으로 드러나는 텍스트 데이터를 수집하고,
  그 속에서 주가 상승 / 하락에 대한 감성분석을 학습시킴으로써 추후 얻게 될 새로운 데이터에도 유연하게 적용 가능함
- 현재는 10일치 데이터로 라벨링해 학습을 진행했으나,
  매번 크롤링으로 추가 데이터를 쌓아 학습을 진행한다면 그에 따라 모델 정확도가 더욱 높아질 것으로 예상
- 추후 상황에 따라 크롤링할 커뮤니티를 추가 확정해 데이터의 양을 키우고,
  지속적인 모델 보완 및 최적화로 OHAE 모델의 정확도를 높여나간다면 더욱 견고한 금융 이슈 관련 자연어 모델로서 입지를 다지게 될 것

감사
합니다

KB Future Finance A.I. Challenge 제3회
KB-ALBERT를 활용한 금융 자연어 혁신 아이디어
중앙대학교 유승욱, 권예진, 김민주