- 텍스트 분석 과제

기간	2024.04.19 ~ 2023.04.26 (8일)
과제내용	

- 프로젝트 프로세스

PHASE 01

Json 구성, CSV파일 관리

1. JSON파일 구성 확인
2. 기사의 긍정,부정 항목에 대해 계산
3. JSON 파일을 합쳐서 CSV파일로 관리

PHASE 02

EDA, 정제

1. 결측치 데이터 본문확인
2. 데이터의 본문길이, 분포 체크(박스플롯, 히스토그램)
3. 데이터 정제
4. 워드클라우드 시각화
5. 이상치 제거

PHASE 03

데이터 전처리

1. 정규분포를 이용한 전처리
2. 형태소 분석 및 토큰화
3. 단어 빈도수를 이용해 불용어 사전 제작.
4. 시각화

PHASE 04

핵심 키워드 추출

1. 전처리, 불용어 사전을 통한 시각화
2. TF-IDF
3. TextRank

PHASE 05

인프라 스트럭처 다이어그램

1. 인프라 스트럭처 다이어그램

01. JSON 구성 확인

```
#데이터 뽑아오기.
for item in data["data_investing"]:
    title = item["title"]
    url = item["url"]
    host = item["host"]
    imgurl = item["imgurl"]
    docsent = item["docsent"]
    sentscore = item["sentscore"]
    text = item["text"]

    # 각 항목에 대한 분석 작업 수행

    # 예시로 각 기사의 제목과 호스트를 출력
    print("기사 제목:", title)
    print("호스트:", host)
    print("")
```

뉴스 기사 데이터: 26454개

뉴스 제목

뉴스 URL

뉴스 포털

기사 본문 이미지

감성분석(긍정,부정,중립)

스코어링 점수 (-1 <=0 <= 1)

뉴스 본문

데이터의 구성 확인.

```
#감정 분석 결과 통계량 계산
sentiments = [item['docsent'] for item in data['data_investing']]

positive_count = sentiments.count('positive')
neutural_count = sentiments.count('neutral')
negative_count = sentiments.count('negative')

print('감성 분석 결과 통계')
print('Positive 기사 수:', positive_count)
print('neutral 기사 수:', neutural_count)
print('negative_count', negative_count)
#총 기사수 개수와 똑같은 것을 확인.
```

긍정 기사: 862개

중립 기사:4559개

부정 기사:415게

- 긍정적인 기사와 부정적인 기사 개수를 통해 어느 정도 분포를 가지고 있는지 1차적으로 확인.

01. CSV 파일 관리

```python
import pandas as pd
import json

# 데이터를 저장할 리스트
data_list = []

# 읽어올 JSON 파일들의 이름 리스트 추가.
json_files = ["20220204.json", "20220304.json", "20220404.json","20220504.json","20220604.json","20220704.json","20220804.json"]

# 각 JSON 파일을 읽어와서 데이터를 추출하여 리스트에 저장.
for file in json_files:
    with open(file, "r", encoding="utf-8") as f:
        data = json.load(f)
        data_id = data["data_id"]  # JSON 파일의 data_id를 날짜로 사용.
        for item in data["data_investing"]:
            title = item["title"]
            url = item["url"]
            host = item["host"]
            imgurl = item["imgurl"]
            docsent = item["docsent"]
            sentscore = item["sentscore"]
            text = item["text"]

            # 각 항목을 딕셔너리로 저장하여 리스트에 추가.
            data_list.append({
                "data_id": data_id,
                "title": title,
                "url": url,
                "host": host,
                "imgurl": imgurl,
                "docsent": docsent,
                "sentscore": sentscore,
                "text": text
            })

# 리스트를 데이터프레임으로 변환.
df = pd.DataFrame(data_list)

# 데이터프레임을 CSV 파일로 내보내기.
df.to_csv("combined_news.csv", index=False, encoding="utf-8")

print("CSV 파일이 성공적으로 생성되었습니다.")
```

- 파일을 합쳐 csv 파일로 관리.
- test해보기 좋은 환경으로 제작.

02. EDA(데이터 확인)


```python
df.info()
# 데이터의 text 결측값 존재, 결측값 해결 방법( 어떻게 할까 ? )
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26454 entries, 0 to 26453
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   data_id    26454 non-null  int64
 1   title      26454 non-null  object
 2   url        26454 non-null  object
 3   host       26454 non-null  object
 4   imgurl     26454 non-null  object
 5   docsent    26454 non-null  object
 6   sentscore  26454 non-null  float64
 7   text       25023 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 1.6+ MB
```

- 파일을 확인해보니 전체적인 데이터를 확인.
- text 본문 데이터에 결측치가 있기 때문에 결측치를 확인, 결측치를 어떻게 처리할지 뉴스 제목을 보고 정해보기로 결정.
- 데이터의 크기를 확인하고 어떤 컬럼을 쓸지 결정.

```python
# 데이터의 크기 확인.
df.shape
```

```
(26454, 8)
```

02. EDA(결측값 확인)


```
# sentscore이 positive이고 text가 결측값인 경우의 기사 제목(title) 출력
missing_text_positive = df[(df['docsent'] == 'positive') & df['text'].isnull()]
print("sentscore이 positive이고 text가 결측값인 경우의 기사 제목:")
print(missing_text_positive['title'])

# sentscore이 neutral이고 text가 결측값인 경우의 기사 제목(title) 출력
missing_text_neutral = df[(df['docsent'] == 'neutral') & df['text'].isnull()]
print("\nsentscore이 negative이고 text가 결측값인 경우의 기사 제목:")
print(missing_text_neutral['title'])

# sentscore이 negative이고 text가 결측값인 경우의 기사 제목(title) 출력
missing_text_negative = df[(df['docsent'] == 'negative') & df['text'].isnull()]
print("\nsentscore이 negative이고 text가 결측값인 경우의 기사 제목:")
print(missing_text_negative['title'])
```

```
sentscore이 positive이고 text가 결측값인 경우의 기사 제목:
8                        [인사] 한국증권금융
23                         [인사] 키움증권
37           2022년 광진구, 이렇게 달라집니다!
39        광산구, 상생경제 '전국 최고' 지자체 선정
89             휴온스글로벌, 송수영 총괄사장 영입
                      ...
23362          [부고] 박성규 한화생명 상무 부친상
23469               [인사] 한국농업기술진흥원
23471                      [인사]교육부
25448                       울주군 인사
25777                        중구 인사
Name: title, Length: 326, dtype: object

sentscore이 negative이고 text가 결측값인 경우의 기사 제목:
10                          충북 동부축 고속도로망 국가계획 반영 추진
43                                      당좌거래정지
50          홍 부총리 "도심복합 등 12만3000호 이상 후보지 올해 추가 선정"
54                  [속보] 1월 소비자물가 3.6% 상승…4개월 연속 3%대
68        서울시, 외식업 등 식품자영업자에 총 200억 1% 저리대출… 작년의 10배
                                ...
26062                  코스닥 017p002 내린 80417개장 | 한경닷컴
26069                  코스피 106p004 내린 243856개장 | 한경닷컴
26233                     풀무원 '전골 밀키트' 출시…밀키트 사업 진출
26330               한국투자신탁운용, 삼성그룹 16개종목 투자펀드 운용전략 재편
...                                                    ...
21601                                      아이디 6649화
21721                    음주 사고 낸 뒤 도주한 운전자…항소심서 '무죄'
21786                         잘나가는 리츠株의 굴욕…공모가 대거 깨져
Name: title, Length: 102, dtype: object
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

- 1차적으로 결측값이 있는 데이터의 제목을 추출해서 확인.
- 마이너스 요소인 제목과, 플러스 요소인 제목이 모두 모여있음을 확인.
- 직접 url을 들어가서 본 결과 감성분석 스코어링은 제목을 통해 산출한 것을 확인.

긍정,부정 결측값 확인

```
positive_sentiments.info()
#긍정부분에서 결측값 326개    1,431
```

```
neutral_sentiments.info()
#증립 부분에서 결측값 1,003개 확인.
```

```
negative_sentiments.info()
#부정부분에서 결측값 102개 확인.
```

- 파일을 확인해보니 전체적인 데이터를 확인.
- text 본문 데이터에 결측치가 있기 때문에 결측치를 확인, 결측치를 어떻게 처리할지 뉴스 제목을 보고 정해보기로 결정.

02. EDA(뉴스데이터 본문 길이 확인)


```python
# 데이터의 본문 길이를 확인.
text_length = df['text'].astype(str).apply(len)
text_length
```

```
0         769
1        1193
2        1031
3         242
4         219
        ...
26449     135
26450    1275
26451     595
26452     944
26453    1103
Name: text, Length: 26454, dtype: int64
```

```python
import numpy as np
#데이터의 분포 확인.
print('뉴스 길이 최댓값: {}'.format(np.max(text_length)))
print('뉴스 길이 최솟값: {}'.format(np.min(text_length)))
print('뉴스 길이 평균값: {:.2f}'.format(np.mean(text_length)))
print('뉴스 길이 표준편차: {:.2f}'.format(np.std(text_length)))
print('뉴스 길이 중간값: {}'.format(np.median(text_length)))
# 데이터의 분포가 너무 고르지 않음. 모델에 넣기 전 중간값으로 할지 고민.
```

✓ 0.0s

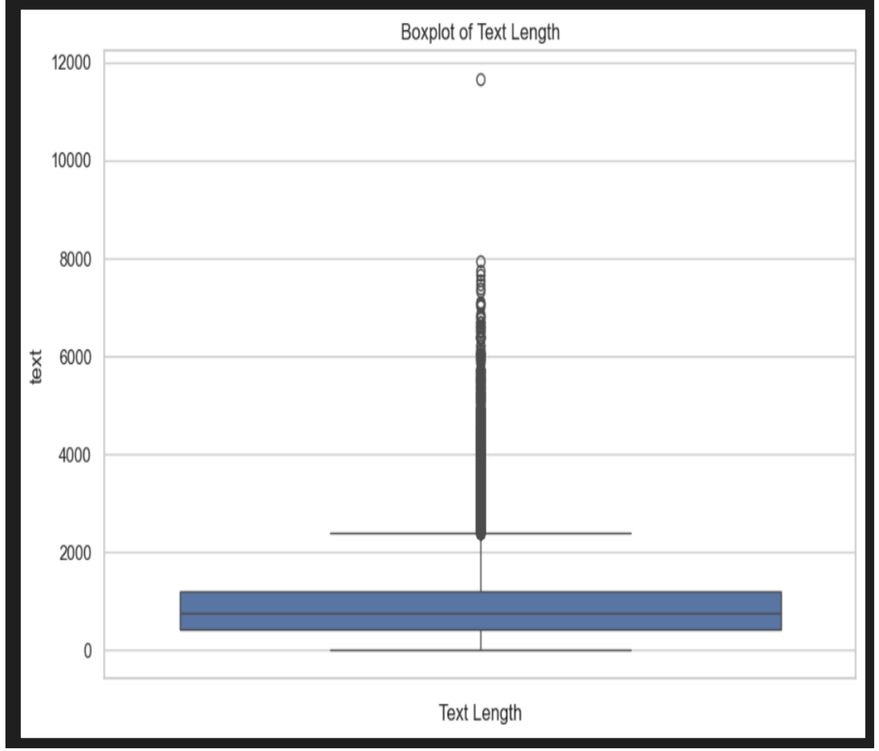
```
뉴스 길이 최댓값: 11651
뉴스 길이 최솟값: 3
뉴스 길이 평균값: 924.34
뉴스 길이 표준편차: 787.39
뉴스 길이 중간값: 755.0
```

- 본문 길이의 비율을 보기 위해 분포와 길이를 확인.
- 본문 길이가 최댓값과 평균값, 중간값과의 차이가 많이 나는 것을 확인.
- 만약 요약을 하거나 모델에 넣고 돌린다면 패딩처리가 필수적이기 때문에 모델을 무겁지 않게 줄이는 것이 필수적으로 보임.

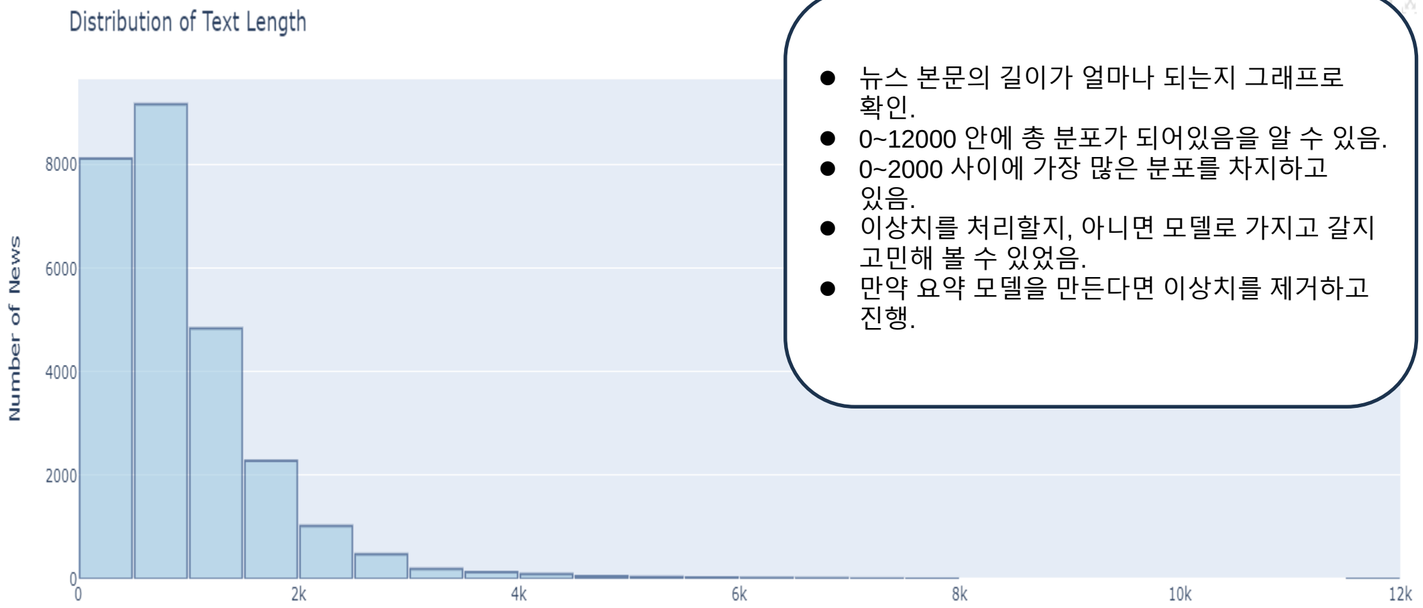
02. EDA(본문 길이 분포 시각화)


```
import seaborn as sns
import matplotlib.pyplot as plt


sns.set(style="whitegrid")

# 박스플롯 그리기
plt.figure(figsize=(10, 6))
sns.boxplot(y=text_length)
plt.title('Boxplot of Text Length')
plt.xlabel('Text Length')
plt.show()

#일부 길이가 긴 데이터가 있다는 것을 확인.
#앞에 분포를 확인했다는 것과 마찬가지로 중간값과 평균값이 전체 데이터로 봤을 때 아래쪽에 위치함.
```

- 일부 길이가 긴 데이터가 있다는 것을 확인.
- 앞의 분포와 마찬가지로 중간값과 평균값이 전체 데이터를 봤을 때 아래쪽에 위치하였고, 모델에 넣기 전 처리가 필요.




```
import plotly.express as px
# Plotly를 사용하여 인터랙티브한 히스토그램 사용
fig = px.histogram(x=text_length, nbins=50, labels={'x': 'Length of News', 'y': 'Number of News'})

# 그래프에 추가적인 스타일링 적용
fig.update_traces(marker_color='rgb(158,202,225)', marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5, opacity=0.6)
fig.update_layout(title='Distribution of Text Length', xaxis_title='Length of News', yaxis_title='Number of News',
                  bargap=0.05)
fig.show()
#뉴스 본문 길이가 500-999 자가 제일 많은 것을 확인.
#분포가 고르지 않기 때문에 모델에 돌리기 전 전처리 필수로 보임.
```



- 뉴스 본문의 길이가 얼마나 되는지 그래프로 확인.
- 0~12000 안에 총 분포가 되어있음을 알 수 있음.
- 0~2000 사이에 가장 많은 분포를 차지하고 있음.
- 이상치를 처리할지, 아니면 모델로 가지고 갈지 고민해 볼 수 있었음.
- 만약 요약 모델을 만든다면 이상치를 제거하고 진행.

02. EDA(이상값)


```
#글자수가 2,500개 이상인 데이터가 적은 것을 확인 -> 이 데이터들은 어떤 데이터들인지 확인. -> 향후 데이터를 분석할 때 어떻게 처리할지 고민.
#
long_texts = df[df['text'].str.len() >= 2500]

# 확인
long_texts[['data_id', 'title', 'url', 'host', 'imgurl', 'docsent', 'sentscore', 'text']]
```

	data_id	title	url	host	imgurl	docsent	sentscore	text
15	20220204	[이슈 리포트] 109년 된 '비밀의 사원'...최악 인플레 공포 지울까	https://www.sedaily.com/NewsView/2620DC2KH1	sedaily.com	https://newsimg.sedaily.com/2022/02/04/2620DC2...	negative	-0.897685	100년이 조금 넘는 연준의 역사에서 큰 족적을 남긴 중앙은행의 수장들이 있었는데,...
67	20220204	서울 강남 사무실 '귀하신 몸'... 공실률 0% 대 : 비즈N	https://bizn.donga.com/realestate/3/0102/20220...	bizn.donga.com	https://dimg.donga.com/a/560/0/90/5/wps/ECONOM...	neutral	0.000000	2년 전 직원 4명으로 출발한 회사가 급성장하면서 지난해 초부터 대형 사무실 물색에...
232	20220204	2022년 2월 3일(목) 미국 증시 시황... 하락세로 마감 - 조선일보	https://www.chosun.com/economy/global-stock/20...	chosun.com	https://images.chosun.com/resizer/IRkD6sBT7Ck8...	neutral	0.000000	3일(현지 시각) 미국 증시가 하락세를 보였다.3대 주가지수인 S&P500이 전 거...
250	20220204	SK텔레콤, 스팸·스미싱 보이스피싱 피해 예방법 공개	http://www.csbn.co.kr/news/article.html?no=233807	csbn.co.kr	http://www.csbn.co.kr/data/photos/adexpo/20220...	neutral	0.000000	(한국안전방송) SK텔레콤이 스팸 및 스미싱, 보이스피싱 사기 피해에 대한 고객의 ...
259	20220204	[대선후보 첫 TV토론] 지상중계-10(일자리·성장)	https://www.yna.co.kr/view/AKR20220203168400000...	yna.co.kr	https://img6.yna.co.kr/photo/yna/YH/2022/02/03...	neutral	0.000000	2022.2.3 [국회사진기자단] srbaek@yna.co.kr ⟲ 주도권 토론 ▲...
...	...	...	...	...	...	...	...	...
26359	20220804	한은 "빅스텝 금리인상, 2년 뒤 집값 최대 1.4% 하락 효과"	https://vnexplorer.net/%ed%95%9c%ec%9d%80-%eb%...	vnexplorer.net	https://img-s-msn-com.akamaized.net/tenant/amp...	negative	-0.648498	1.75%포인트 인상이 점진적으로 이뤄졌기 때문이라는 게 한은 조사국의 설명이다.\...
26373	20220804	한지민, 오늘은 청순 말고 고혹	https://vnexplorer.net/%ed%95%9c%ec%a7%80%eb%a...	vnexplorer.net	https://img-s-msn-com.akamaized.net/tenant/amp...	positive	0.847081	(엑스포츠뉴스 오승현 인턴기자) 배우 한지민이 화려한 메이크업과 블랙 드레스로 고혹...
26382	20220804	한투증권, 美 부동산 플랫폼 코리니와 업무 협약 체결	https://vnexplorer.net/%ed%95%9c%ed%88%ac%ec%a...	vnexplorer.net	https://img-s-msn-com.akamaized.net/tenant/amp...	neutral	0.000000	... See Details: '나는 솔로' 6기 정숙, 결혼 코앞에 두고 건강 이...
26385	20220804	한투증권-美부동산 플랫폼 코리니 업무협약	https://vnexplorer.net/%ed%95%9c%ed%88%ac%ec%a...	vnexplorer.net	https://img-s-msn-com.akamaized.net/tenant/amp...	neutral	0.000000	3분기 전기요금이 인상과 더불어 오는 10월에도 kWh당 4.9원이 오를 예정인 가...
26397	20220804	해수부, 전남 흑산면 해역 '고수온 주의보' 발령...수온 28℃ 도달	https://vnexplorer.net/%ed%95%b4%ec%88%98%eb%b...	vnexplorer.net	https://img-s-msn-com.akamaized.net/tenant/amp...	neutral	0.000000	(해양수산부 제공)© 뉴스1 해양수산부(장관 조승환)가 최근 이어지는 폭염으로 수온...

1054 rows × 8 columns

- 글자수가 2,500 이상인 데이터가 적었기 때문에, 이 데이터들을 확인해보고 모델의 크기와 시간에 따라 살릴 데이터와 드랍할 데이터를 구분.

```
# 길이가 2500 이상인 데이터 중 docsent 컬럼 값별 개수 확인
docsent_counts = long_texts['docsent'].value_counts()

# 확인
print(docsent_counts)

#긍정, 부정 값이 생각보다 적음을 알 수 있었다. 중립부분의 데이터가 유의미하지 않다면 모델이 무거워 지지 않게 날리는 방법도 고민.(카테고리에 따라 생각할 수 있음.)
```

```
docsent
neutral     728
negative    168
positive    158
Name: count, dtype: int64
```

- 글자수가 2500개 이상인 데이터 중 긍정 부정 값이 생각보다 적기 때문에 모델이 무겁다면 데이터를 드랍할 수도 있음.

```
max_length_row = df[df['text'].str.len() == text_length.max()]

# 해당 행 출력
max_length_row[['data_id', 'title', 'url', 'host', 'imgurl', 'docsent', 'sentscore', 'text']]

# 중립 기사인 것을 확인. 본문의 텍스트 양이 너무 많지만 어떻게 처리할지 아직 고민.
```

Python

	data_id	title	url	host	imgurl	docsent	sentscore	text
19060	20220604	상하이 조업재개→산업,민생경제 회복을 위한 지원정책 가속화	http://m.dailychina.co.kr/3975	m.dailychina.co.kr	https://f.xza.co.kr/http://www.dailychina.co.k...	neutral	0.0	(1) 외자기업 서비스 전담팀을 구성하여 상하이시와 각 구 정부의 조업복귀 서비스를...

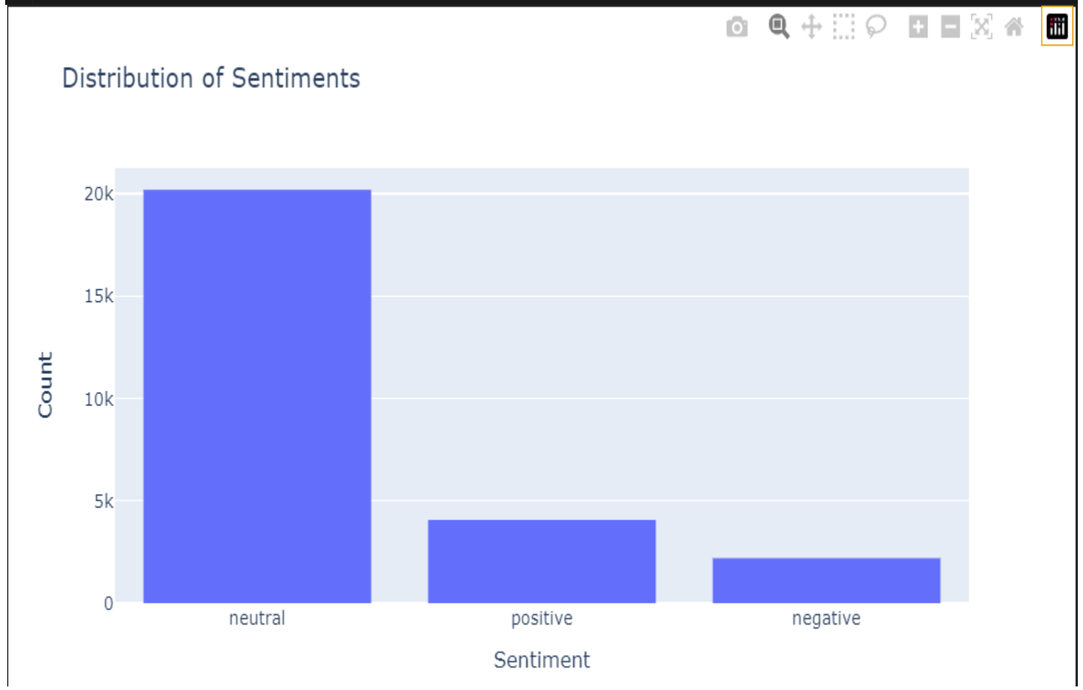
- 글자수가 제일 많은 뉴스기사 데이터는 감성분석 상 중립데이터, 그렇기 때문에 클렌징 과정에서 드랍할 수도 있음.
- 해당 url을 들어가봤지만 들어가지지 않음.

02. EDA(시각화)

```
word_cloud = [news for news in df['text'] if type(news) is str]
#wordcloud로 데이터 확인.
from wordcloud import WordCloud
#
font_path = r'C:\Users\rkh03\Desktop\class\watson_ko\NanumGothic.ttf'

wordcloud = WordCloud(font_path=font_path).generate(''.join(word_cloud))

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
#기사 데이터이다 보니 동사형으로 많이 나온 것을 확인.
#이 데이터만으로는 유의미한 도출을 할 수 없을 것 같음을 확인.
```

- 우선 본문(text) 안에 들어있는 뉴스 본문 데이터를 가지고 wordcloud를 통해 많이 사용된 단어를 확인.
- 기사데이터이다 보니 동사형으로 많이 나왔으며 이 데이터만으로는 유의미한 핵심 데이터를 가져올 수 없음.

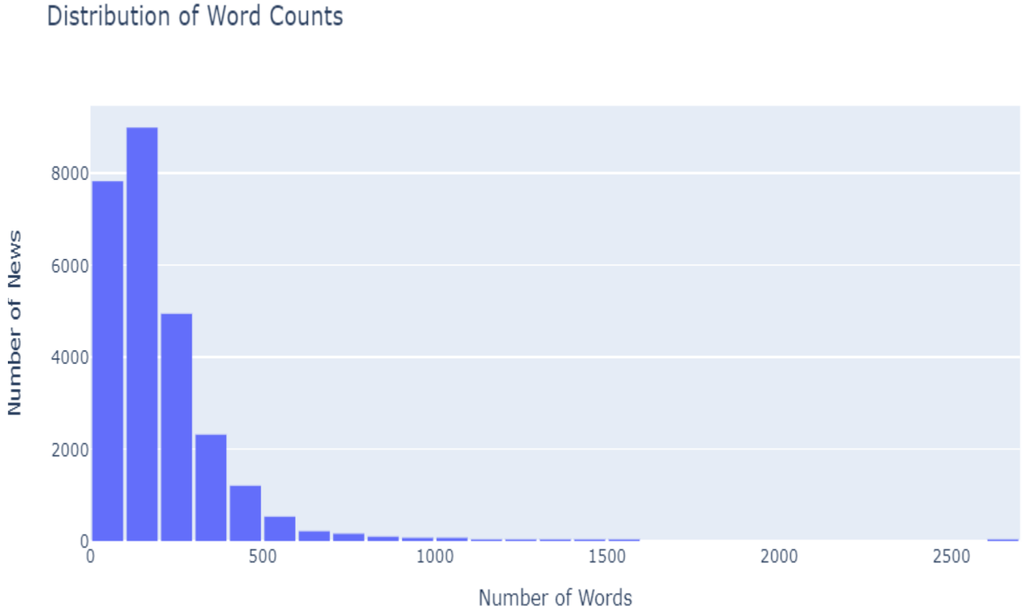
```
#긍정, 중립, 부정 비율 확인.
# decsent 컬럼의 분포 시각화
fig = px.histogram(df, x='docsent', title='Distribution of Sentiments', labels={'docsent': 'Sentiment'})
fig.update_layout(xaxis={'categoryorder': 'total descending'}, yaxis_title='Count',width=800)
fig.show()
```



- 중립데이터가 압도적으로 많은 것을 확인. 부정적인 데이터가 아주 적음(데이터의 클래스 불균형 발생)
- 모델링을 하기 전 클래스 불균형 문제를 해결하기 위해 오버샘플링, 언더샘플링, 또는 샘플 가중치 부여 등의 기술을 고민해볼 수 있었음.
- 모델을 평가할 때 정확도가 아닌 정밀도, 재현율, F1 점수 등의 다양한 성능 지표를 고려.
- 데이터를 추가 수집할 수 있는 경우 추가 수집 또한 고려.

02. EDA(본문의 단어 수 체크)

```python
#뉴스의 단어 수 체크
import plotly.graph_objects as go

word_counts = df['text'].astype(str).apply(lambda x:len(x.split(' ')))

# 단어 수 분포를 히스토그램으로 표시
fig = go.Figure(data=[go.Histogram(x=word_counts, nbinsx=50)])

# 그래프 레이아웃 설정
fig.update_layout(
    title='Distribution of Word Counts',
    xaxis_title='Number of Words',
    yaxis_title='Number of News',
    bargap=0.1,  # 막대 사이의 간격 설정
)

# 그래프 표시
fig.show()
```



- 뉴스 본문의 단어 수를 체크함으로써 분포를 확인.
- 단어 길이의 경우 대부분 100~200 사이에 분포돼 있음.
- 분포가 고르지 않기 때문에 처리가 필요해보임.
- 만약 요약 모델을 돌린다면 패딩 or 잘라내기를 사용.
- 또는 Transformer 모델을 사용(각 요소에 대한 가중치 할당)

```python
#데이터의 분포 확인.
print('뉴스 길이 최댓값: {}'.format(np.max(word_counts)))
print('뉴스 길이 최솟값: {}'.format(np.min(word_counts)))
print('뉴스 길이 평균값: {:.2f}'.format(np.mean(word_counts)))
print('뉴스 길이 표준편차: {:.2f}'.format(np.std(word_counts)))
print('뉴스 길이 중간값: {}'.format(np.median(word_counts)))
# 데이터의 분포가 너무 고르지 않음. 모델에 넣기 전 중간값으로 할지 고민.

#평균 190, 중간값 165지만 글자수 제한을 1600정도로 잡아두고 모델을 돌릴까 고민.
```

✓ 0.0s

```
뉴스 길이 최댓값: 2642
뉴스 길이 최솟값: 1
뉴스 길이 평균값: 190.98
뉴스 길이 표준편차: 165.70
뉴스 길이 중간값: 156.0
```

- 단어 길이에 대한 통계값을 확인.
- 최대값은 2642, 최솟값은 1. 이를 통해 너무 큰 분포를 가지고 있는 데이터임을 다시 한 번 확인.

03. 전처리(결측값 제거, 정규표현식, 토큰화)


```
df_cleaned = df_cleaned.dropna(subset=['text'])

print('결측값 제거 후 데이터 프레임 크기:', df_cleaned.shape)
```

```
결측값 제거 후 데이터 프레임 크기: (25022, 8)
```

- 본문의 내용 중 핵심 키워드를 잡아낼 것이기 때문에 본문 결측값 드랍.
- 원데이터는 유지 하기 위해 새로운 변수값에 저장.

```
#csv파일로 내보내기
df_cleaned.to_csv('df_cleaned.csv', index=False, encoding='utf-8')
```

```
import pandas as pd
df_cleaned = pd.read_csv('df_cleaned.csv', encoding='utf-8')
df_cleaned.head()
```

- 원 데이터 내용은 그대로 두기 위해 새로운 CSV파일로 저장.

```
import re
from konlpy.tag import Okt
from collections import Counter

# Konlpy의 Okt 형태소 분석기를 사용
okt = Okt()

# 특수문자 제거 함수
def remove_special_characters(text):
    # 특수문자 제거를 위한 정규표현식
    text = re.sub(r"[^\w\s'|]", "", text)
    return text

def rsc(text):
    # 특수문자 제거를 위한 정규표현식]
    text = re.sub(r"[^a-zA-Z0-9ㄱ-힣\s]", "", text)
    return text

def remove_newline(text_list):
    # 리스트의 각 요소를 문자열로 결합
    text = ''.join(text_list)
    # 개행 문자 제거
    text = text.replace("\n", "")
    return text

# 형태소 분석 및 토큰화 함수
def tokenize(text):
    # 형태소 분석 및 토큰화
    tokens = okt.morphs(text)
    return tokens


df_cleaned['text'] = df_cleaned['text'].apply(remove_special_characters)
df_cleaned['text'] = df_cleaned['text'].apply(rsc)

df_cleaned['text'] = df_cleaned['text'].apply(remove_newline)
df_cleaned['text'] = df_cleaned['text'].apply(tokenize)


print(df_cleaned['text'])
```

- 한국어 기사이기 때문에 Konlpy(Open Korea Text)를 사용.
- 주로 코모란을 뉴스 데이터에서 사용하지만 태깅 없이 핵심어만 추출하기 때문에 Okt로 진행
- 전처리 부분에서는 특수문자 제거를 위해 2개의 정규표현식 사용, 개행문자 /n(줄바꿈) 제거.
- 형태소 분석 및 토큰화.

```
0        [NBDC, 개요, NBDC, 누, 비드, 치는, 메타, 버스, 매거진으로, 다양한...
1        [다만, 매출, 에, 비해, 적자, 폭도, 크게, 상승, 했다, 리얼리티, 랩스, ...
2        [경기, 핫, 타임, 뉴스, 김삼, 영, 기자, 수천만, 원, 규모, 의, 법원, ...
3        [경기도, 시흥시, 배, 미, 골길, 23, 목감동, 미디어, 타임즈, 의, 모든,...
4        [경기도, 시흥시, 배, 미, 골길, 23, 목감동, 미디어, 타임즈, 의, 모든,...
                                ...
25017    [이, 회사, 는, 29일, 을, 제외, 하고, 26일, 부터, 1일, 까지, 3,...
25018    [8월, 2일, 온라인, 코딩, 교육, 플랫폼, 스파르타, 코딩, 클럽, 을, 통해...
25019    [2만, 3035달러, 약, 3020만원, ChartsBTC, ChartsBtc, ...
25020    [2020년, 애드엑스, 에, 인수, 합병, 된, 애드, 파이는, 2016년, 출시...
25021    [계속, 해서, 그, 는, 국내, 상황, 도, 긍정, 적, 으로, 코로나, 19, ...
Name: text, Length: 25022, dtype: object
```

- 결과물 확인.
- 토큰화까지 처리된 것을 확인.

03. 전처리(불용어 처리)


```python
from collections import Counter
# 모든 단어를 하나의 리스트로 펼치기
all_words = [word for sublist in df_cleaned['text'] for word in sublist]

# 단어 빈도수 계산
word_counts = Counter(all_words)

# 가장 빈도가 높은 단어 20개 출력
print(word_counts.most_common(100))
```

```
[('을', 240155), ('이', 171497), ('에', 154353), ('의', 151705), ('를', 148700),
('은', 116135), ('는', 105670), ('으로', 101113), ('가', 89784), ('한', 88910),
('로', 58944), ('등', 56230), ('에서', 52544), ('과', 51195), ('것', 48031)
```

- 핵심 키워드를 뽑기 전 단어의 빈도수를 계산 후 결과 내용을 통해 불용어 사전 직접 제작.
- 해당 코드로 4번을 반복.

```python
# 불용어 사전 직접 정의
stopwords = ["을", "를", "이", "가", "은", "는", "이런", "저런",'가',',',',','이','다','는','에','기','지','을','가','로','0','고','의','한','하','2','대','1','은','인','자','사','시'
             ,'를','해','서','원','수','도','상','정','업','전','으','장','보','제','3','스','했','부','리','""''',')','()','있','금','주','일','비','""','과','국','적',
             '5','만','성','경','공','%','어','나','위','라','소','4','등','계','회','조','년','중','-','면','구','아','세','신','화','개','산','용','관','트','동','행',
             '재','연','들','출','유','할','했다','에서','과','것','적','수','하는','하고','할','들','인','도','와','이다','해','있다','및','전','고','다','된','원','있는',
             '말','까지','위','통해','기','3일','위해','대','중','1','제','부터','지','될','2','|','명','개','연','관련','이라고','더','된다','보다','주','이번','이상',
             '에는','세','시','내','지난','으로','ㅣ','밝혔다','한다','됐다','에게','경우','따르면','이후','에도','액','대해','19','함께','예정','되는','그','점','하며',
             '같은','최대','또한','따라','하면','대한','약','달','현재','주요','다양한','수준','했다고','때문','되고','에서는','지난해','대비','올해','간','총','날','전체',
             '후','건','특히','가장','제공','기자','하기','지난달','하지','있도록','최근','분야','예상']

def remove_stopwords(tokens, stopwords):
    tokens = [token for token in tokens if token not in stopwords]
    return tokens


# 불용어 제거 적용
df_cleaned['text'] = df_cleaned['text'].apply(lambda x: remove_stopwords(x, stopwords))
```

- 불용어 사전 제작한 것을 기반으로 제거.
- 그럼으로써 명사형 데이터와, 중요 핵심 키워드가 되는 데이터만 남음.

04. 핵심 키워드 추출(전처리, 불용어 사전을 통한 시각화)


```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import random

# 워드 클라우드 객체 생성
wordcloud = WordCloud(
    font_path="NanumGothic.ttf",
    background_color="white",
    width=800,
    height=400,
    colormap="viridis",
    max_words=100,
    prefer_horizontal=0.7,
    min_font_size=10,
    max_font_size=200,
    random_state=42
)

# 빈도수 데이터를 이용하여 워드 클라우드 생성
wordcloud.generate_from_frequencies(word_counts)

# 색상 함수 정의
def random_color_func(word=None, font_size=None, position=None, orientation=None, font_path=None, random_state=None):
    return "rgb({}, {}, {})".format(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))

# 워드 클라우드 시각화
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud.recolor(color_func=random_color_func, random_state=3), interpolation="bilinear")
plt.axis("off")
plt.show()
```

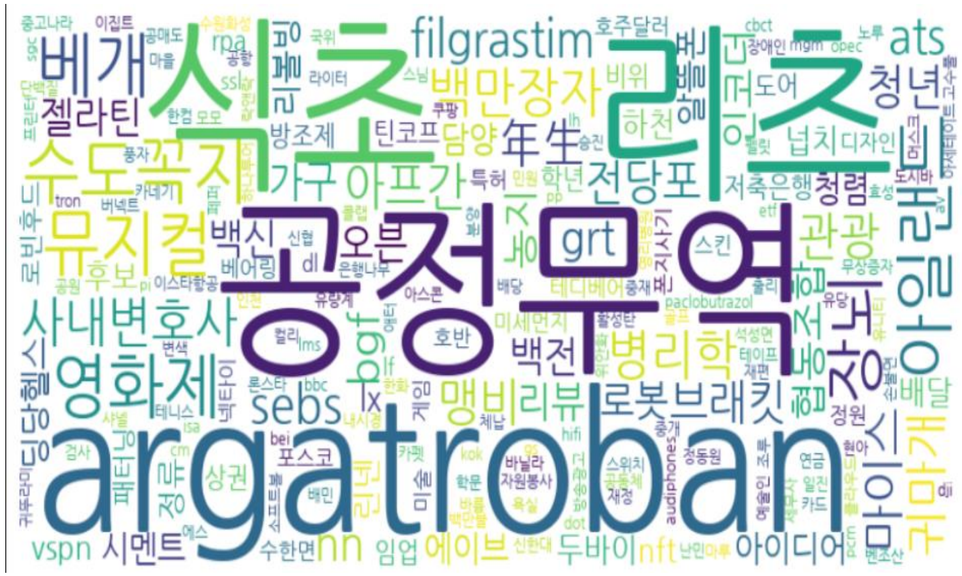
- 전처리, 불용어 사전을 통해 전처리 후 워드클라우드로 시각화.
- 이 시각화를 통해 핵심 키워드를 1차로 추출해서 볼 수 있었음.
- 주로 경제에 관련된 데이터가 많이 나온 것을 확인할 수 있음. = 경제 카테고리

04. 핵심 키워드 추출(TF-IDF)


```python
from sklearn.feature_extraction.text import TfidfVectorizer

# 토큰화된 텍스트를 다시 문자열로 변환
def join_tokens(tokens):
    return ' '.join(tokens)

# 토큰화된 텍스트를 문자열로 변환
df_cleaned['text_str'] = df_cleaned['text'].apply(join_tokens)

tfidf = TfidfVectorizer()

# TF-IDF 행렬 생성
tfidf_matrix = tfidf.fit_transform(df_cleaned['text_str'])

# 단어 목록 기입.
words = tfidf.get_feature_names_out()

# TF-IDF 점수를 기준으로 단어 정렬
tfidf_scores = tfidf_matrix.max(axis=0).toarray().flatten()
word_tfidf_scores = list(zip(words, tfidf_scores))
word_tfidf_scores.sort(key=lambda x: x[1], reverse=True)

# 상위 30개의 핵심 키워드 추출
top_n = 30
top_keywords = [word for word, score in word_tfidf_scores[:top_n]]

print("상위 {}개의 핵심 키워드:".format(top_n))
for keyword in top_keywords:
    print(keyword)
```

```
상위 30개의 핵심 키워드:
리츠
공정무역
식초
argatroban
수도꼭지
아일랜드
장뇌
영화제
베개
뮤지컬
사내변호사
백만장자
병리학
年生
sebs
filgrastim
맹비
관광
아프간
마이스
ats
귀마개
브래킷
nn
...
bgf
grt
젤라틴
백전
```

- 많이 쓰이는 TF-IDF를 통해 핵심 키워드 30개 추출.
- 텍스트 데이터에서의 단어의 상대적인 중요성을 평가하려 했지만 기존에 뽑아보았던 빈도수가 높은 데이터가 뽑히지 않음.
- 파라미터 (빈도수 조절) 를 수정해보았지만 생각보다 유익한 데이터를 뽑지 못함.
- 모델을 돌리는데에는 시간이 오래 걸리지 않음.(2.3초)



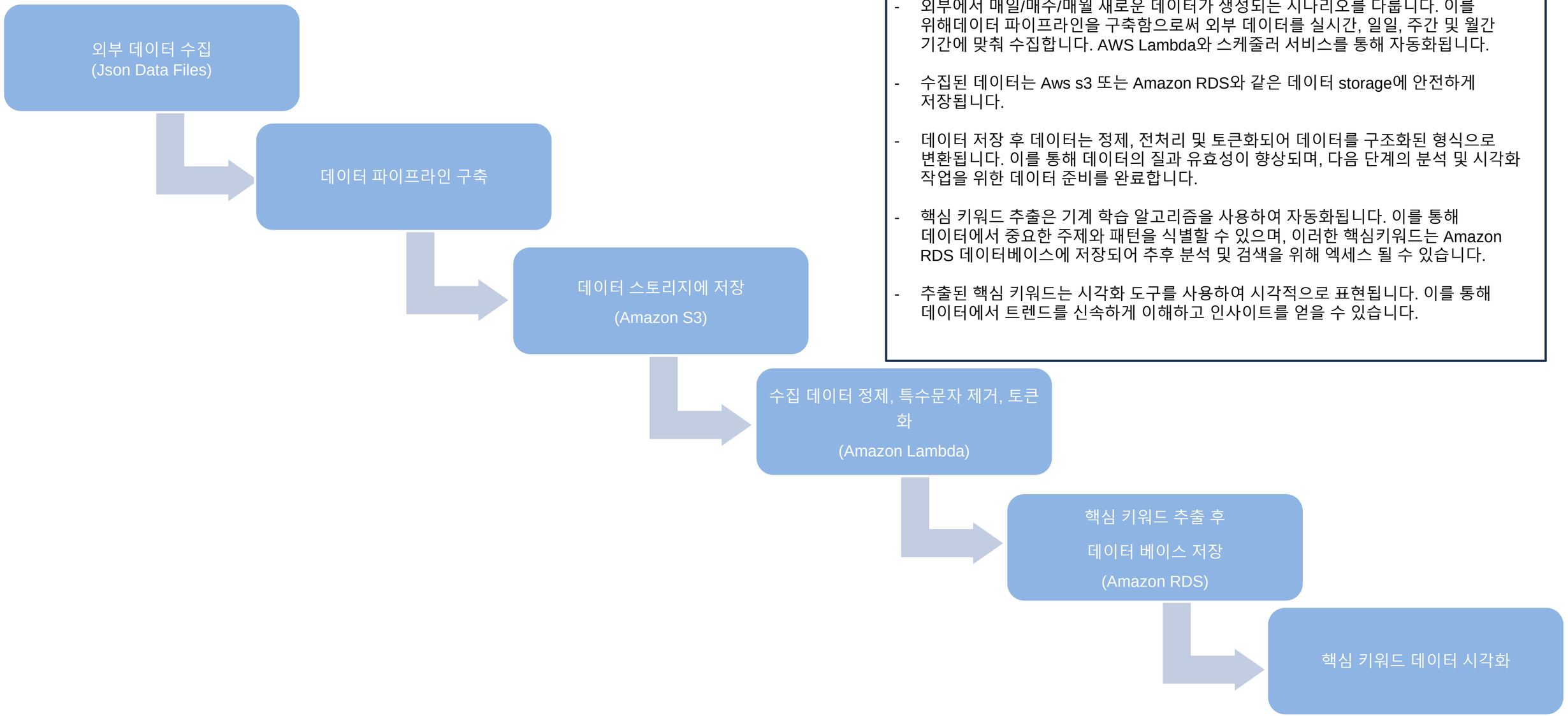
04. 핵심 키워드 추출(TextRank)

```python
from textrank import KeywordSummarizer

# 텍스트 데이터 로드
text_data = df_cleaned['text'].apply(lambda x: ' '.join(x))

# 형태소 분석기 초기화
okt = Okt()

# 명사 추출 함수 정의
def get_nouns(text):
    nouns = okt.nouns(text)
    return [noun for noun in nouns if len(noun) > 1]  # 한 글자 명사는 제외

# 텍스트에서 명사 추출
nouns_list = text_data.apply(get_nouns)

# 단어 빈도수 계산
all_nouns = [noun for sublist in nouns_list for noun in sublist]
word_counts = Counter(all_nouns)

# TextRank를 이용한 키워드 추출
summarizer = KeywordSummarizer(tokenize = okt.nouns, min_count=2, window=-1)
keywords = summarizer.summarize(text_data.tolist(), topk=30)

# 추출된 키워드 출력
for word, rank in keywords:
    print(word, rank)
```

```
시장 180.5553182509536
기업 177.23728017707953
사업 162.36379286778646
투자 133.61888142591053
금융 117.676989988995
지원 103.80757044285716
지역 96.23406792563061
서비스 80.742729484037
금리 78.43118939702352
거래 76.82542446496224
정부 74.81478909834723
상승 74.69815801652688
미국 73.36407236301389
산업 70.2392839238352
증가 69.6138612248117
경제 67.3722766029702
대출 67.2911182240476
계획 66.3382983401796
한국 66.264500505389
가격 63.77852480429917
은행 63.15328042108045
확대 60.224647644170346
성장 59.37543103548086
글로벌 59.01671981546816
대표 58.186797163750015
...
매출 56.09594343092034
코로나 55.49873077339744
기술 55.15572641993985
기준 54.9854185191459
```

- textrank를 통해 핵심 키워드 30개 추출.
- 기존의 전처리, 불용어사전을 통해 제작했던 모델과 유사함. 하지만 '시장' 단어가 제일 높은 데이터로 나타남.
- 모델을 돌리는데에는 시간이 꽤 오래 걸림.(46분 4.9초)

05. 인프라 스트럭처 다이어그램



- 외부에서 매일/매주/매월 새로운 데이터가 생성되는 시나리오를 다룹니다. 이를 위해데이터 파이프라인을 구축함으로써 외부 데이터를 실시간, 일일, 주간 및 월간 기간에 맞춰 수집합니다. AWS Lambda와 스케줄러 서비스를 통해 자동화됩니다.
- 수집된 데이터는 Aws s3 또는 Amazon RDS와 같은 데이터 storage에 안전하게 저장됩니다.
- 데이터 저장 후 데이터는 정제, 전처리 및 토큰화되어 데이터를 구조화된 형식으로 변환됩니다. 이를 통해 데이터의 질과 유효성이 향상되며, 다음 단계의 분석 및 시각화 작업을 위한 데이터 준비를 완료합니다.
- 핵심 키워드 추출은 기계 학습 알고리즘을 사용하여 자동화됩니다. 이를 통해 데이터에서 중요한 주제와 패턴을 식별할 수 있으며, 이러한 핵심키워드는 Amazon RDS 데이터베이스에 저장되어 추후 분석 및 검색을 위해 엑세스 될 수 있습니다.
- 추출된 핵심 키워드는 시각화 도구를 사용하여 시각적으로 표현됩니다. 이를 통해 데이터에서 트렌드를 신속하게 이해하고 인사이트를 얻을 수 있습니다.