INCIZOR

W2V 기반
뉴스트렌드 분석

안녕하세요. 지원자 이민성입니다.
Word2Vec 기반의 뉴스 데이터 트렌드 및 키워드 분석을 진행
하였습니다.

INCIZOR

PART 01

ABOUT **TASK**

분석 소개 및 목표 설명

PART 02

HOW I **WORK**

분석과정 설명

PART 03

WHAT I **DID**

분석 리포트

INCIZOR

PART
01

ABOUT Task

분석 소개 및 목표 설명

W2V 단어 임베딩 후
각 벡터 값의 평균을 구해 문서 임베딩을 진행함.

임베딩된 값을 K-MEANS 기법을 활용하여
군집분석을 수행함.

각 군집 별 TR를 수행,
군집별 키워드를 추출하여 기사 주제를 라벨링함.

INCIZOR

PART
02

HOW
I WORK

분석 과정 소개

```
def sub_special(s): ## 특수문자, 숫자, 영어, 중복공백 제거
  rs = re.sub(r'[^가-힣]',' ',s)
  rr = re.sub(' +', '', rs)
  return rr
```

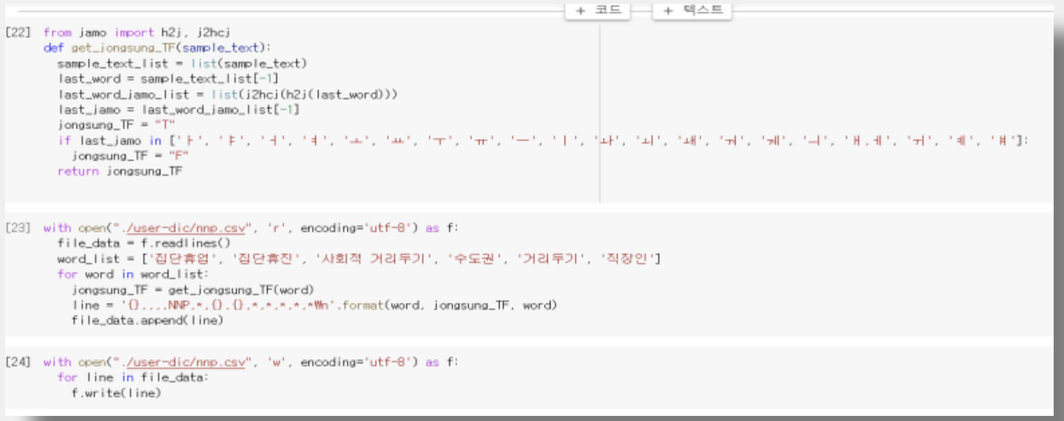
1차 전처리

정규표현식을 활용하여,
특수문자, 숫자, 영어, 중복공백 등을 제거하였습니다.

```
sample = data.copy()
sample['cleansed'] = np.NaN
sample.cleansed = sample.text.apply(sub_special)
sample['mecab'] = sample.cleansed.apply(mecab.nouns)
```

형태소 분석

mecab의 빠른 속도를 활용하여,
2만9천건의 문서를 적절한 시간내에 형태소 분석을 하였습니다.



```
[22] from jamo import h2j, j2hcj
     def get_jongsung_TF(sample_text):
       sample_text_list = list(sample_text)
       last_word = sample_text_list[-1]
       last_word_jamo_list = list(j2hcj(h2j(last_word)))
       last_jamo = last_word_jamo_list[-1]
       jongsung_TF = "T"
       if last_jamo in ['ㅏ', 'ㅑ', 'ㅓ', 'ㅕ', 'ㅗ', 'ㅛ', 'ㅜ', 'ㅠ', 'ㅡ', 'ㅣ', 'ㅘ', 'ㅙ', 'ㅚ', 'ㅝ', 'ㅞ', 'ㅟ', 'ㅢ', 'ㅐ', 'ㅒ', 'ㅖ', 'ㅔ']:
         jongsung_TF = "F"
       return jongsung_TF

[23] with open("./user-dic/nnp.csv", 'r', encoding='utf-8') as f:
       file_data = f.readlines()
       word_list = ['집단휴업', '집단휴진', '사회적 거리두기', '수도권', '거리두기', '직장인']
       for word in word_list:
         jongsung_TF = get_jongsung_TF(word)
         line = '{},,,,NNP,*,{},{},*,*,*,*,*\n'.format(word, jongsung_TF, word)
         file_data.append(line)

[24] with open("./user-dic/nnp.csv", 'w', encoding='utf-8') as f:
       for line in file_data:
         f.write(line)
```

사용자 사전 정의

Mecab의 사용자 사전을 활용하여,
수도권, 집단휴진, 거리두기와 같은 단어들을 분석하였습니다.

```
def word_cleansing(data):
  for i in range(len(data)): #불용어 제거
    result = []
    for w in data.mecab[i]:
      if w not in stopwords:
        result.append(w)
    data.mecab[i] = result
```

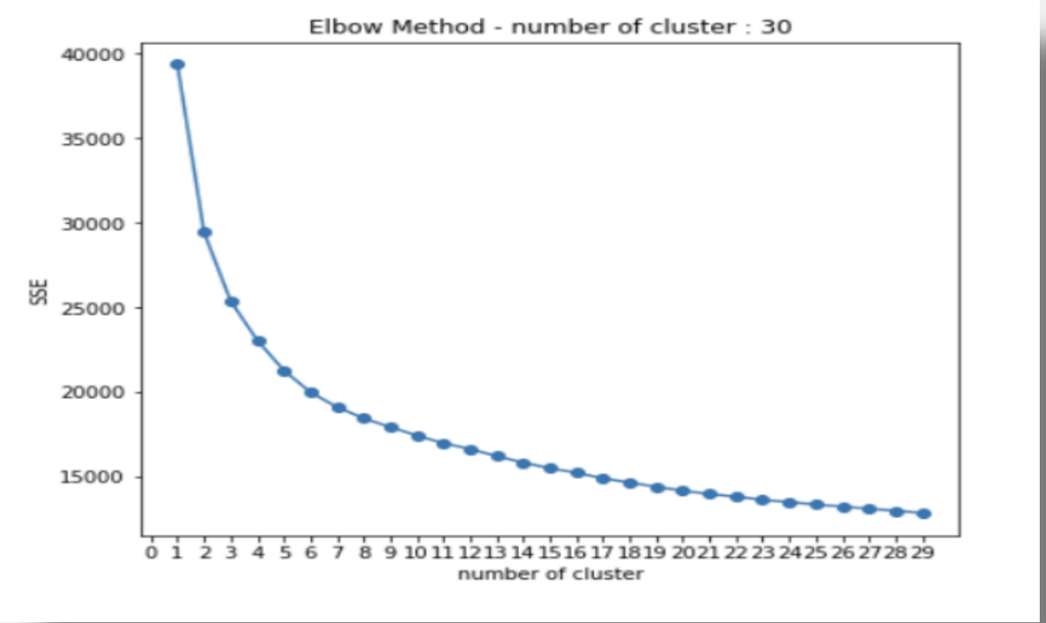
2차 전처리

불용어 사전을 별도로 정의하였고,
정의된 사전에 따라 불필요한 단어들을 추가적으로 제거하였습니다.

```
[101] embedding_model = Word2Vec(sample.mecab, size=50, window = 2, min_count=3, workers=4, iter=10, sg=1, seed=0)

[102] sample['wv'] = sample['mecab'].map(get_sentence_mean_vector)

[105] create_model(sample)
```



기사 임베딩

W2V 차원수 50으로 설정하여 진행하였고,
기사별 단어들의 평균 벡터값을 구하여,
기사의 벡터값을 도출하였음.

Elbow기법을 통한 클러스터 수

30개의 클러스터까지 설정하여,
SSE 값을 도출, 최적 군집수를 설정해보려 하였으나,
큰 변곡점이 없어 임의의 군집 15를 설정하였음.

```
keyword_extractor = KeywordSummarizer(tokenize = token,
    min_count=3,
    window=8,
    min_cooccurrence=2,
    vocab_to_idx=None,
    df=0.85,
    max_iter=30,
    verbose=False)

keywords1 = keyword_extractor.summarize(aaa, topk=20)
```

TextRank

최소 동시발생수는 2,
최소 빈도수는 3으로 설정하여
TextRank를 적용하였음.

분석 결과

```
[('한겨레', 4.194989008611455),
 ('무단', 3.580114990374057),
 ('전재', 3.523437322177987),
 ('배포', 3.352361766532243),
 ('금지', 3.024179790544881),
 ('신문', 2.2936717701707083),
 ('그림판', 2.17600890529961
 ('철', 2.050795100271635),
 ('권', 1.7988542731265786),
 ('특파원', 1.74019691361120
 ('경향신문', 1.027940291771
 ('부임', 0.6454825919641209
 ('서울', 0.6281624327729438
 ('현지', 0.5849024163392887
 ('채널', 0.579933754536589)
 ('사진', 0.5676890857542334
 ('동아일보', 0.481168315808
 ('팀', 0.4507021481754896),
 ('신임', 0.4494391327963157
 ('조선일보', 0.42261635849148
```

```
[('금지', 1.4800842152435312),
 ('배포', 1.4800842152435312),
 ('전재', 1.4800842152435312),
 ('무단', 1.4800842152435312),
 ('경향신문', 0.9722775960045815),
 ('서민호', 0.6627607235566094),
 ('조선일보', 0.2847640363193653),
 ('동아일보', 0.15986078314531943)]
```

TroubleShooting

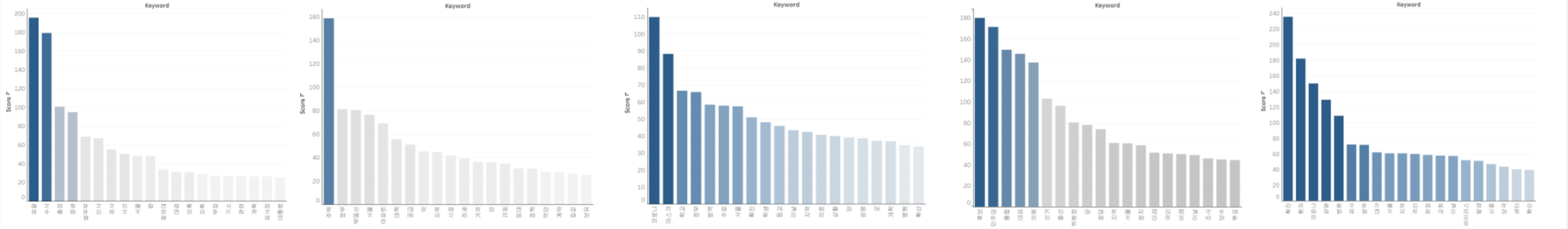
군집별 분석 결과, 2개의 카테고리(총 500건)의
이상 기사가 발견되어, 해당 카테고리를 제외하여
총 13개의 카테고리가 형성되었음.

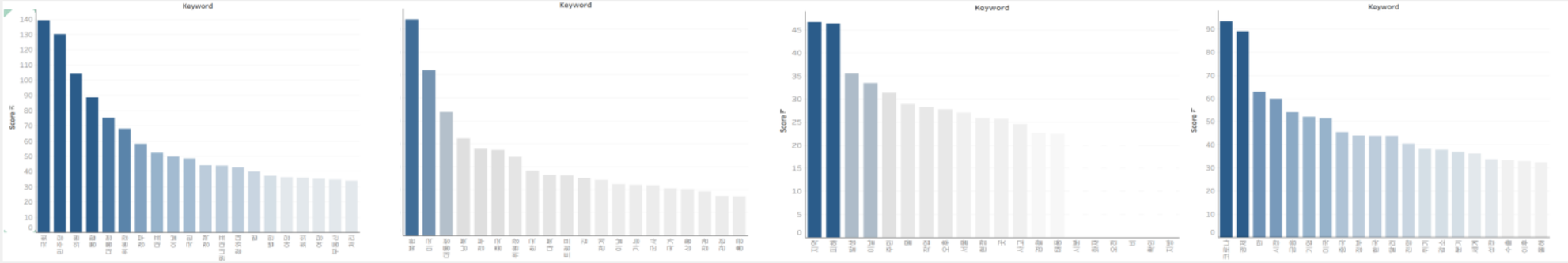
INCIZOR

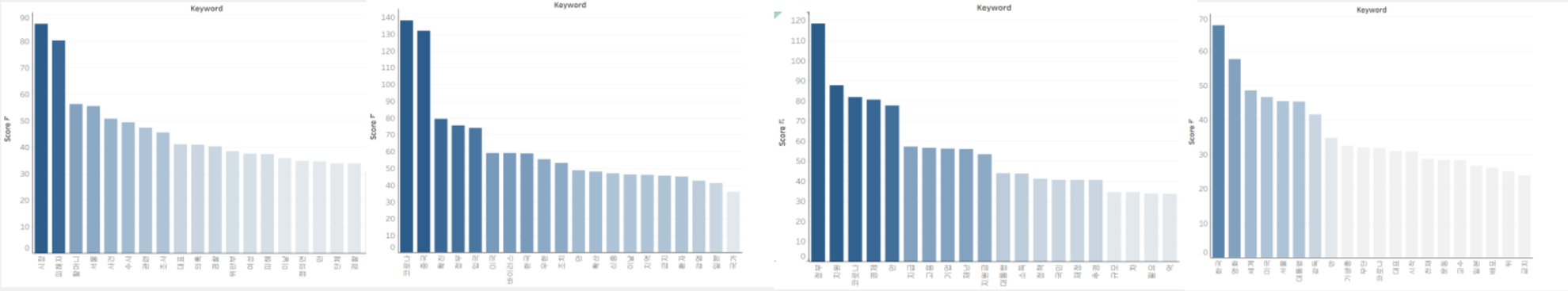
PART
03

WHAT I DID

분석 리포트


Keyword
Score F


Keyword
Score F


Keyword
Score F

INCIZOR

THANK YOU

이민성

감사합니다.