2022년 데이터 분석 청년인재 양성사업 일경험수련 결과보고서

네이버 검색량 데이터 기반 식품 검색어 이상탐지모델(Anomaly Detection) 구축

2022.02.28

Y220793	이건하

식품안전정보원 정보서비스부(식품안전빅데이터분석)

목 차

Ⅰ. 분석 개요 ····· 1
1. 배경 및 필요성 ····· 1
2. 수행 기관 ····· 3
3. 분석 목표 ····· 4

Ⅱ. 분석 방법 ····· 4
1. 분석 프로세스 ····· 4
2. 분석 도구/환경 ····· 4
3. 데이터 수집 ····· 5
4. 분석 방법 ····· 21

Ⅲ. 분석 결과 ····· 31
1. 결과 제공 형태 ····· 31
2. 수행 결과 ····· 34

Ⅳ. 결론 ····· 40
1. 기대효과 및 활용방안 ····· 41
2. 문제점 및 개선방안 ····· 41

Ⅴ. 참고자료/부록 ····· 42

Ⅰ. 분석 개요

1. 배경 및 필요성

○ 식품위생법 제68조(식품안전정보원의 사업) 제①항 제1호[1]에 명시된 '식품안전정보원'의 설립목적에 따른 '식품안전정보수집·분석·정보제공 등'에 기여하기 위함

○ 국민에게 다양하고 편리한 식품안전정보를 제공하기 위한 식품안전나라 누리집의 '외부환경·사용자 요구변화·정보이용 패턴 등을 파악'의 일환으로 새로운 정보통신기술을 이용한 모니터링 시스템 구축에 기여하기 위함

○ 우리나라 국민의 관심사 파악을 위한 도구로써, 2022년 검색엔진 점유율 60%이상을 차지하는 포털사이트 '네이버'의 검색량이 적합한 수단이라고 판단

그림 1 데이터 솔루션 기업 '인터넷트렌드'의 2022년 검색엔진 점유율 통계 결과

2. 수행 기관

○ 수행 기관

: 한국지능정보사회진흥원(NIA) 주관사업 '2022년 데이터 분석 청년인재 양성사업' 참여기관 '식품안전정보원(정보서비스부)'

○ 수행 기간

: 한국지능정보사회진흥원(NIA) 주관사업 '2022년 데이터 분석 청년인재 양성사업' 일경험수련 협약시작일 2022년 9월5일부터 협약종료일 2023년 2월28일까지(약 6개월).

1) 1. 국내외 식품안전정보의 수집·분석·정보제공 등 2. 식품이력추적관리 등을 위한 정보시스템의 구축·운영 등 3. 식품이력추적관리의 등록·관리 등 4. 식품이력추적관리에 관한 교육 및 홍보 5. 식품사고가 발생한 때 사고의 신속한 원인규명과 해당 식품의 회수·폐기 등을 위한 정보제공 6. 식품위해정보의 공동활용 및 대응을 위한 기관·단체·소비자단체 등과의 협력 네트워크 구축·운영 7. 그 밖에 식품안전정보 및 식품이력추적관리에 관한 사항으로서 식품의약품안전처장이 정하는 사업

3. 분석 목표

○ 대한민국 검색엔진 포털사이트 ‘네이버’의 식품 및 식품안전 용어의 검색량을 파악하여 국민의 식품 및 식품안전에 관한 관심사 모니터링

○ 시계열 예측모델을 통해 식품 및 식품안전 용어의 일반적인 검색량을 계산하고 이를 실제 검색량 비교하여, 일반적인 검색량에 비해 실제검색량이 비정상적으로 높은 수치를 기록하는 검색어를 탐지하여 매일 식품 및 식품안전 관련 이슈를 빠르게 파악할 수 있는 자동화 모델 구축

Ⅱ. 분석 방법

1. 분석 프로세스

○ 네이버 데이터랩 ‘통합검색어 트렌드 API[2)]’를 이용한 식품 및 식품안전 용어의 상대적 검색량 추출

○ ‘NAVER Search Ad API’의 최근 한달 검색량 절대값을 이용한 전체 기간 대상 검색어 상대값→절대값 변환

○ 네이버 검색량 데이터와 시계열 이상탐지 모델 Prophet을 이용한 식품 및 식품안전 용어의 일반적인 검색량 산출

○ 일반적인 검색량 대비 비일반적으로 높은 검색량을 보이는 검색어와 기간 탐지 (Anomaly Detection)

○ 비일반적인 검색량을 보인 기간에 대한 뉴스 크롤링

○ 한국어 형태소 분석기를 이용한 뉴스 내 주요 키워드 추출

○ 식품 및 식품관련 검색어의 비일반적 검색량이 나타났던 기간에 대한 주요 키워드 분석 및 시각화

2. 분석 도구/환경

○ 한국지능정보사회진흥원(NIA) 주관사업 ‘2022년 데이터 분석 청년인재 양성사업’ 일경험 수련 기간 중 제공된 노트북(LAPTOP-A96P6D2C)을 이용하여 분석 진행

○ GPU/서버 등 추가적인 환경 없이 CPU 환경(Gen Intel(R) Core(TM) i5-1135G7)에서 분석 진행

○ 프로그래밍 언어로는 파이썬(Python3)을 사용

2) 통합검색어 트렌드는 네이버 통합검색에서 발생하는 검색어를 연령별, 성별, 기기별(PC, 모바일)로 세분화해서 조회할 수 있는 API

3. 데이터 수집

3.1. 모니터링 목표 검색어 (1755개 중 중복검색어를 제외한 1629개)

: ⑥항목과 ⑦항목의 자료는 협약기관인 '식품안전정보원'으로부터 내부적 활용을 위해서만 사용하기로 한 자료로 담당기관에 확인받았기에(외부 유출 불가), 해당 자료를 포함한 모니터링 목표 검색어의 세부항목에 대해서는 공개하지 않는다.

① 자체검색어 리스트 추가 (18개)

② 맛집 정보 서비스 식신[3]의 카테고리와 메뉴[4] (105개)

카테고리	메뉴
양식/레스토랑	햄버거, 피자, 패밀리레스토랑, 패스트푸드, 스테이크하우스, 씨푸드, 뷔페, 돈가스, 유로피언레스토랑, 퓨전레스토랑, 이태리레스토랑, 프렌치레스토랑
카페/디저트	베이커리/제과점, 카페/커피숍, 컵케익, 도넛, 브런치, 아이스크림, 카페테리아/식당, 애견카페, 북카페
한식	한정식, 해물탕/해물요리/꽃게, 설렁탕/곰탕/도가니탕, 라면/칼국수/국수/수제비, 찌개/전골/국/탕, 파전/모듬전/빈대떡, 비빔밥/돌솥밥/쌈밥, 해장국/국밥, 순대국, 감자탕, 철판구이/볶음밥, 장어구이/꼼장어, 낙지, 회, 전복, 홍어, 냉면, 순두부, 떡, 죽, 떡볶이/순대/튀김/만두, 전라도음식, 그밖에또다른것
고기/구이류	삼겹살/목살, 돼지갈비/갈매기살, 불고기/갈비살/차돌박이, 꽃등심/등심/육회, 오리훈제/구이/로스/탕, 닭볶음탕/닭갈비/닭발, 삼계탕/백숙/찜닭, 치킨/훈제, 전골/수육, 곱창/양/대창/막창, 족발/보쌈
일식/중식/세계음식	일본음식/초밥, 중국음식, 카레, 샤브샤브, 딤섬, 쌀국수, 이자까야, 이슬람음식, 지중해음식, 남미음식, 태국음식, 인도음식, 동남아음식, 아프리카음식, 그밖에또다른것
나이트라이프	소주, 막걸리/동동주, 포장마차, 실내포장마차, 사케, 맥주/호프, 와인, 바, 칵테일, 호텔바, 가라오케, 나이트클럽, 클럽

그림 2 맛집 정보 서비스 '식신'의 카테고리와 메뉴

③ 식품안전나라 공공데이터 '용어사전(식품첨가물용어집)' (653개)

그림 3 식품안전나라 공공데이터 누리집

3) 푸드테크 기업 '식신(SikSin)'이 운영하는 맛집검색 웹/앱 서비스.
4) 이미지 출처. 논문("An Empirical Study on the Influence of Weather and Daytime on Restaurant Menu search System". == "날씨 및 요일 특성이 음식점 메뉴 검색시스템 이용에 미치는 영향에 관한 실증 연구")

④ 네이버 데이터랩 식품 카테고리 인기검색어[5] top500 (500개)

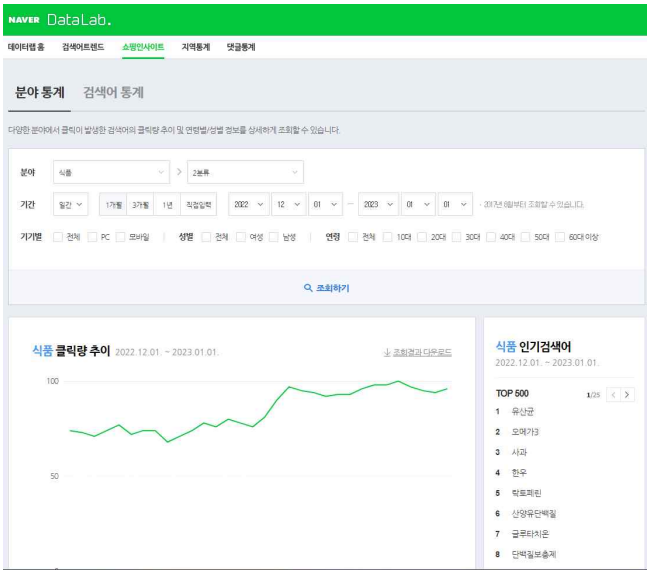


그림 4 네이버 데이터랩

⑤ 요식업브랜드명[6] (100개)

1	역전할머니맥주1982	외식	522.5	98.0	103.5	91.7	46.5	92.6	90.2
2	군자대한곱창	외식	519.1	99.8	115.5	82.5	42.0	89.1	90.2
3	푸라닭	외식	517.6	103.2	111.0	75.2	46.5	86.7	95.0
4	지코바양념치킨	외식	512.2	99.8	99.0	97.2	46.5	89.1	80.8
5	곱분이곱창	외식	508.1	92.8	108.0	78.8	48.0	90.2	90.2
6	비에이치씨(BHC)	외식	502.4	103.2	93.0	64.2	61.5	90.2	90.2
7	피자나라 치킨공주	외식	497.6	101.5	76.5	78.8	61.5	89.1	90.2

그림 5 멕세스 컨설팅. 가맹하고 싶은 외식업 top100

⑥ 글로벌정보부의 '식품안전이슈 관련 키워드(식품안전 지정어)' (74개)

⑦ '식품안전나라 검색어 기록[7]'으로부터 텍스트 추출 (305개)

5) 기간은 20221015~20221115
6) 2022년 가맹하고 싶은 외식업 프랜차이즈(멕세스 컨설팅)
7) 2022/09/13 기준 식품안전나라 홈페이지 내 2022년 검색량 합계 4000건 이상인 검색어 중 선별

3.2. 네이버 데이터랩 '통합검색어 트렌드 API'를 이용한 식품 및 식품 안전 용어의 '상대적 검색량' 추출

① 네이버 데이터랩 '통합검색어 트렌드 API'

: 특정 검색어의 검색량이 가장 높은 날의 수치를 100으로 설정한 뒤, 이 날의 검색량을 기준으로 다른 날짜/다른 검색어(최대5개)의 상대적 검색량을 제공해주는 API서비스. 검색기간[8], 기기(모바일/PC), 나이대[9], 성별(남/녀)에 따른 분류 가능. 본 프로젝트에서는 인적특성에 따른 구분을 하지 않았지만, 필요시 파라미터를 변경하여 개별 특성에 따른 검색도 가능하다.

	날짜	강아지	고양이	토끼	너구리	다람쥐
0	2016-01-01	18.38176	18.04358	5.75150	3.59468	2.37224
1	2016-01-02	18.38927	18.66482	5.60120	3.80260	2.57014
2	2016-01-03	17.20941	17.82064	5.42334	3.50200	2.54008
3	2016-01-04	16.17484	17.26703	4.73947	3.77755	2.75050
4	2016-01-05	16.20991	17.18937	4.87474	3.93787	2.51503
...	...	...	...	...	...	...
2553	2022-12-28	10.31312	13.67985	12.48747	3.49198	3.09368
2554	2022-12-29	10.19539	13.81262	14.02054	3.49448	2.94589
2555	2022-12-30	10.85170	13.42935	18.82765	3.27655	2.95841
2556	2022-12-31	11.07214	14.35871	29.37625	3.53206	2.65781
2557	2023-01-01	11.80861	15.29559	63.51452	3.44689	2.70791

2558 rows × 6 columns

그림 6 '통합 검색어 트렌드 API' 호출 결과. 100 기준 검색량 상대값 제공

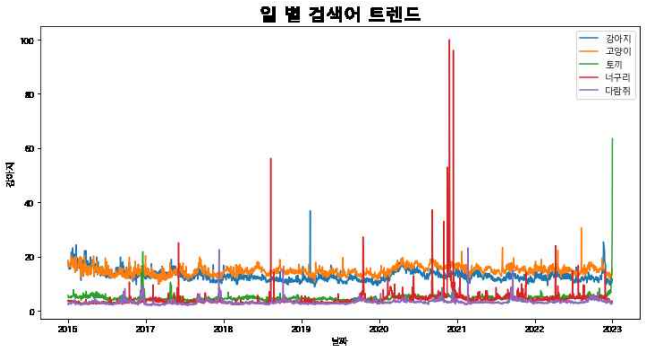


그림 7 상대적 검색량 결과를 선그래프를 이용해 시각화한 예시

② '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)[10]' 탐색

: 현재 네이버 데이터랩 '통합 검색어 트렌드 API'는 검색어의 절대값을 제공하지 않기 때문에 기준점인 최대값(=100. max_key)이 바뀔 경우 검색어들의 상대값이 달라지게 된다. 다만 '통합 검색어 트렌드 API'는 비교 가능한(=한번에 검색 가능한) 검색어의 수를 최대 5개로 제한하고 있으므로, 6개 이상의 검색어를 비교하기 위해서는 상대값의 기준(=100. max_key)을 통일시켜야 할 필요가 있다.

→ 이를 해결하기 위해 5개의 검색어 중 '가장 높은 검색량(100)을 포함하는 단어(max_key)'에 해당하는 검색어를 찾은 후, 해당 검색어를 다음 4개(최대 비교가능 검색어의 수가 5개이므로, 최대값에 해당하는 검색어+4개) 검색어 리스트의 검색단계에 포함시키는 작업을 반복실행[11]하여 전체 검색어 리스트 중 최대값을 갖는 하나의 검색어(total_max_key)를 찾도록 하였다.

8) 기간은 선택가능하며, 본 프로젝트에서는 최초제공일인 2016년 1월 1일부터 최근 제공일(검색일 기준 하루 전)까지로 설정

9) '12세미만', '13세~18세', '19세~24세', '25세~29세', '30세~34세', '35세~39세', '40세~44세', '45세~49세', '50세~54세', '55세~59세', '60세이상' 총 11단계로 구성

10) 모든 단어의 검색량 상대값을 비교했을 때, 특정일의 검색량이 '100'인 검색어가 최대 검색량을 포함하는 검색어이다.

11) 만약 최대값을 갖는 검색어가 갱신될 경우 이전 검색단계의 최대값으로 넘어온 검색어가 아닌, 갱신된 새 검색어가 다음 검색단계로 넘어간다.

(예시)

- '강아지', '고양이', '토끼', '너구리', '다람쥐' 5개 검색어의 비교에서, 가장 높은 검색량(100)을 포함하는 단어(max_key)는 '너구리'이므로, '너구리'를 다음 4개의 검색어와의 검색단계에 포함시킨다.

	날짜	강아지	고양이	토끼	너구리	다람쥐
1792	2020-11-27	12.57765	16.62825	5.33567	100.00000	3.02104

그림 8 전체 기간 중 '강아지', '고양이', '토끼', '너구리', '다람쥐' 5개 검색어에 대한 가장 높은 검색량(100)은 '2020년 11월 27일 너구리'에 대한 검색량

- 가장 높은 검색량을 포함하는 '너구리'가 다음 검색단계에 포함되었다.

	날짜	너구리	얼룩말	고라니	원숭이	호랑이
0	2016-01-01	3.55699	0.60729	1.88136	33.15321	6.19934
1	2016-01-02	3.76273	0.60977	1.93837	14.18585	7.13630
2	2016-01-03	3.46528	0.64695	2.12924	9.65471	6.05557
3	2016-01-04	3.73794	0.70148	1.80452	11.95002	4.83107
4	2016-01-05	3.89658	0.73122	2.81089	8.74005	4.64516
...	...	...	...	...	...	...
2553	2022-12-28	3.45537	0.58498	2.47130	1.83674	3.94864
2554	2022-12-29	3.45784	0.59985	2.38950	2.07966	4.51379
2555	2022-12-30	3.24219	0.57259	2.71918	2.12428	4.43943
2556	2022-12-31	3.49503	0.51805	2.82329	1.99538	5.38631
2557	2023-01-01	3.41075	0.56267	3.04885	2.22095	5.21527

2558 rows × 6 columns

그림 9 '너구리'를 포함하여 진행한 다음 검색단계

- 이 작업을 전체 검색어 리스트가 끝날 때까지 반복하여 최종적으로 '최대 검색량을 포함하는 검색어(total_max_key)'를 탐색한다.

```
total_max_key
```

```
'사자'
```

그림 10 모든 검색어에 대한 비교 결과 최대 검색량을 포함하는 검색어를 'total_max_key'로 지정

③ 중복 검색어 제거(오류 처리1)

: 만약 검색어 리스트를 여러 곳에서 가져오게 된다면, 전체 리스트에서 중복되는 검색어가 있을 수 있다. 이 경우 결과 출력 시 검색어의 이름이 _x, _y 형태로 바뀌게 되는데, 이를 후처리 과정에서 제거할 경우 탐색시간 및 API호출 수 낭비로 이어질 수 있으므로, API호출 작업이 이루어지기 전에 검색어의 중복여부를 확인하도록 설정하였다.

	날짜	사과_x	사과_y
0	2016-01-01	17.62633	17.62633
1	2016-01-02	18.08640	18.08640
2	2016-01-03	20.16389	20.16389
3	2016-01-04	27.24462	27.24462
4	2016-01-05	26.13040	26.13040
...	...	...	...
2553	2022-12-28	30.89641	30.89641
2554	2022-12-29	28.86205	28.86205
2555	2022-12-30	25.62001	25.62001
2556	2022-12-31	23.42750	23.42750
2557	2023-01-01	27.37402	27.37402

그림 11 중복되는 검색어가 있을 경우 _@ 형식으로 검색어가 변경

④ 리스트가 4의 배수가 아닐 때(오류처리2)

: 검색어의 마지막 비교단계에서는 남은 검색어의 수가 4개미만이 될 수도 있다. API 호출 시 5개[12]의 검색어를 비교하도록 설정하였기 때문에, 이 경우에도 오류를 반환한다. 이러한 오류를 방지하기 위해, 중복값을 제거한 전체 검색어 리스트[13] 수가 '4의 배수+1[14]'이 아닐 경우 '임시'라는 검색어를 반복하여 추가[15]하여 오류를 방지하였다.

⑤ 5개의 검색어 비교 시, 특정 검색어가 '모든 날짜'에 검색량이 현저히 적거나 없는 경우(오류 처리3)-error_list

: 5개의 검색어 비교 시, 특정 검색어가 '모든 날짜'에 검색량이 현저히 적거나 없는 경우 다른 4개의 검색어의 검색량과 관계없이 오류를 반환하고 있다(다른 4개의 검색어의 상대적 검색량도 출력하지 않음).

→ 이 때 어떤 검색어의 값이 존재하지 않는지, 검색량이 존재하지 않는 검색어가 몇 개인지를 알려주지 않기 때문에 4개[16]의 검색어에 대해서 '개별로 검색량을 호출[17]'하여 오류가 있는 검색어를 전체 검색어 목록에서 제외하도록 설정하였음.

---

12) 4개의 비교대상 검색어 + 이전 단계의 최대값을 갖는 검색어(max_key)
13) 원본 리스트에 대해 중복값 제거와 '각주 13번'의 과정을 거친 이후의 리스트를 의미
14) 맨 처음 비교시에는 4개가 아닌 5개의 비교가 실행되기 때문에, 검색어의 개수는 4의 배수가 아닌 1을 더한 값이 되어야 한다.
15) 남은 검색어 수에 따라 최대 3개(남은 검색어가 1개일 경우)까지 추가될 수 있다. 결과 반환 시에는 '임시'라는 검색어를 삭제하고 출력하도록 설정하였다.
16) 이전 검색단계에서 최대값(max_key)으로 들어온 1개의 검색어는 오류가 없다는 전제하에 넘어오기 때문에 오류점검 대상이 아니기 때문에 5개가 아닌 4개
17) 하나의 검색어를 다른 단어와 비교하지않고, 다른 날짜 간 검색량만 비교(상대값 비교에 사용되지 않고 출력이 정상적으로 되는지만 확인하는 절차)

(예시)

- ɣ-글루타밀트랜스펩티다아제', '계란', '수박', '크릴오일' 4개 단어 검색 중 검색량 값을 갖지 않는 검색어가 있는 것이 확인되어 개별 호출을 통해 오류 검색어 확인.

```
error_list
```

```
[['ɣ-글루타밀트랜스펩티다아제', '계란', '수박', '크릴오일']]
```

그림 12 검색량이 존재하지 않는 검색어가 1개 이상 포함되어 검색값을 반환하지 못한 검색어 목록

- 오류가 발생했던 검색단계 내 4개 검색어(error_list)를 하나씩 실행하여 오류를 반환했던 검색어(final_error)를 확인

```
final_error #검색량 검색결과 하루의 결과도 반환하지 않는 경우
```

```
['ɣ-글루타밀트랜스펩티다아제']
```

그림 13 개별검색 결과 검색량이 존재하지 않는 검색어는 'ɣ-글루타밀트랜스펩티다아제'

⑥ 5개의 검색어 비교 시, 특정 검색어가 '특정 날짜(모든 날짜)'에 검색량이 현저히 적거나 없는 경우(오류 처리4)-single_list

: 5개의 검색어 비교 시, 특정 검색어가 '특정 날짜(⑤의 '모든 날짜'와 구별)'에 검색량이 현저히 적거나 없는 경우 그 날의 값이 0이 아닌 NaN(결측값)으로 표시되는데(오류로 인식X), 이 때 NaN값을 포함하는 검색어가 5개의 리스트 중 중간에 오는 경우에는 문제가 없지만(그림14. 해당 날짜의 자신의 검색량만 NaN으로 표시), 5개의 검색어 리스트 중 첫 번째로 오는 경우, 해당 날짜의 다른 4개의 검색어의 상대적 검색량까지 표시하지 않고 있다(그림15. 결과자체를 오류로 반환하지는 않지만, 출력결과에는 첫 검색어의 결과가 없는 날짜에 대해서 5개 검색어의 값이 전부 제외[18]).

→ 리스트의 첫 번째로 오는 검색어를 항상 직전 검색단계[19]에서 '최대값(max_key)'을 가지고 있는 검색어가 되도록 설정하였다(total_max_key를 찾기 위해 넘어온 max_key). 만약 최대값(max_key)이 갱신됐을 때, 새 최대값을 갖는 검색어(maxkey)의 검색값에 하루라도 NaN값이 포함됐을 경우(이전 단계에서 검증[20]), 이 검색어를 제외시키고[21] 남은 검색어에 대해서 최대값을 가지는 검색어를 다시 찾는 작업을 반복하여, 5개의 검색어 비교 시 항상 리스트의 첫 번째로 오는 검색어는 '모든 날짜에 대한 검색값을 가지는 검색어'가 되도록 하였다.

---

18) 모든 날짜에 대한 검색량을 비교한 것이 아니게 되므로, 출력 결과(특정일이 제외된 비교 결과)에서 100의 값을 검색어와, 실제 최대값(모든 날짜에 대한 비교 결과)을 갖는 검색어가 달라질 수 있다.

19) 맨 처음 검색단계에서 문제가 발생하지않도록 전체 리스트에서 가장 처음 오는 검색어에 대해서는 반드시 '모든 날짜에 대한 검색값을 가지는 검색어'가 되도록 별도의 실행을 거치도록 하였다.

20) 이전 단계에서는 2단계 이전 max_key가 첫 검색어로 왔을 것이기 때문에, 이전 단계에서는 갱신된 새 최대값을 갖는 검색어(지금 max_key)에 결측값이 있더라도 NaN이 표시된다.

21) 전체 검색어 리스트에서 제외가 아닌, 별도의 리스트(single_list)로 저장해놓은 뒤, 전체 리스트를 한바퀴 돈 시점에서 최대값을 갖는 검색어(total_max_key)와 다시 비교 결과, 최대값이 갱신되지 않을 경우을 갖는 검색어보다 최대값이 낮은 경우(최대값 갱신X) NaN값이 있는 형태로 포함시키고, 최대값을 갖는 검색어보다 최대값이 높은 경우 상대값에는 포함하지 않고 후에 절대값으로 변환하는 단계에서 포함하도록 설정하였다.

	날짜	사과	식품안전나라	식약처	포도	딸기
0	2016-01-01	0.49054	NaN	0.05961	0.08402	0.40311
1	2016-01-02	0.50334	NaN	0.07262	0.08262	0.41292
2	2016-01-03	0.56116	NaN	0.08002	0.09122	0.70360
3	2016-01-04	0.75822	NaN	0.62398	0.19825	0.60037
4	2016-01-05	0.72721	NaN	0.62098	0.10943	0.56536
...	...	...	...	...	...	...
2553	2022-12-28	0.85985	0.70380	0.40131	0.16424	1.15754
2554	2022-12-29	0.80323	0.39951	0.37691	0.16224	1.84234
2555	2022-12-30	0.71301	0.30228	0.26907	0.17525	2.82863
2556	2022-12-31	0.65199	0.06561	0.07002	0.18665	2.12582
2557	2023-01-01	0.76182	0.08102	0.08042	0.15084	2.17984

2558 rows × 6 columns

그림 14 2016년 1월 검색량이 존재하지않는 '식품안전나라'의 상대적 검색량 값이 '0'이 아닌 'NaN'으로 표시

	날짜	식품안전나라	식약처	사과	포도	딸기
0	2016-10-25	0.00100	0.61538	0.97628	0.15304	0.31129
1	2016-12-09	0.00140	0.50795	0.48414	0.09042	0.43092
2	2017-01-23	0.04181	0.65519	0.77322	0.12063	0.52615
3	2017-01-24	0.03240	0.67179	0.70540	0.11923	0.50354
4	2017-01-25	0.03881	0.60898	0.65159	0.10343	0.42332
...	...	...	...	...	...	...
2167	2022-12-28	0.70380	0.40131	0.85985	0.16424	1.15754
2168	2022-12-29	0.39951	0.37691	0.80323	0.16224	1.84234
2169	2022-12-30	0.30228	0.26907	0.71301	0.17525	2.82863
2170	2022-12-31	0.06561	0.07002	0.65199	0.18665	2.12582
2171	2023-01-01	0.08102	0.08042	0.76182	0.15084	2.17984

2172 rows × 6 columns

그림 15 결측값이 있는 검색어가 5개의 검색어 리스트 중 첫 번째로 오는 경우, 해당 날짜의 다른 4개의 검색어의 값을 누락시킨 채로 결과 반환 (2016-01-01부터가 아닌 없는 날짜 누락)

⑦ 최종단계. '과정①'에서 찾은 '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)'를 기준으로 모든 검색어에 대한 상대값 출력

: '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)'는 모든 날짜에 검색량값이 존재하면서, 리스트 내 어떤 단어와 비교해도 상대값이 변하지 않는[22] 검색어이다. 따라서 '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)'를 모든 검색단계[23]에서 항상 첫 번째로 오도록 설정한다면, 전체 리스트는 '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)'의 최대값(=100)이라는 동일한 기준 하에 상대값이 도출될 것이다[24].

	날짜	계란	포도	배	사과	망고	옥수수	벼	쌀
0	2016-01-01	0.02987	0.01362	0.04742	0.07953	0.06428	0.02293	0.00502	0.02351
1	2016-01-02	0.02523	0.01339	0.04541	0.08160	0.07272	0.02406	0.00476	0.02244
2	2016-01-03	0.02841	0.01479	0.04852	0.09098	0.07596	0.02799	0.00509	0.02951
3	2016-01-04	0.03467	0.03214	0.05870	0.12293	0.07713	0.03078	0.00846	0.03451
4	2016-01-05	0.03360	0.01774	0.06315	0.11790	0.07223	0.03344	0.00755	0.03347

그림 16 동일한 기준에 의해 호출한 상대적 검색량(최종 결과)

22) 항상 100의 값이 자신에게서 나오기 때문
23) '전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)' + 검색어 4개씩 비교
24) 실제로는 API호출 1건당 한번에 4개 단어의 상대적 검색량이 출력되며, 각각의 결과를 하나로 통합하는 과정은 별도의 코드로 작성하였다.

3.3. ‘NAVER Search Ad API(네이버광고API)’를 이용한 검색량 상대값→절대값 변환

① 검색어의 상대값을 실제값으로 변환

: ‘NAVER Search Ad API(네이버광고API)’에는 특정 검색어의 최근 30일치 검색 절대값 합계를 제공하는 API가 존재한다. 이를 앞서 구한 상대적 검색량값과 연계하여, ‘**최근 30일 검색량 절대값 * 100/최근 30일 상대값의 합계**’를 계산(예시: **1.6772900000000002 : 52400 = 100 : x** → 상대값의 100의 검색량 절대값 추정치는 **3,124,087**)하여, ‘상대값 수치 100(total_max_key의 최대값)’의 검색량 절대값을 구한 뒤, 다른 모든 상대값에 대하여 곱셈을 진행(100이라는 동일한 기준으로 산정된 상대값이므로)하여 상대값을 절대값으로 변환하는 작업을 진행하였다[25].

```
search_keyword('식품안전나라')
```

```
52400
```

그림 17 식품안전나라 최근 30일 검색량

```
# 식품안전나라 최근 1달 상대값
df['식품안전나라'].tail(30).sum()
```

```
1.6772900000000002
```

그림 18 식품안전나라 최근 30일 상대값 합계(절대값 100의 값에 따라 달라짐)

날짜	계란	포도	배	사과	망고	옥수수	벼
2016-01-01	0.02987	0.01362	0.04742	0.07953	0.06428	0.02293	0.00502
2016-01-02	0.02523	0.01339	0.04541	0.08160	0.07272	0.02406	0.00476
2016-01-03	0.02841	0.01479	0.04852	0.09098	0.07596	0.02799	0.00509
2016-01-04	0.03467	0.03214	0.05870	0.12293	0.07713	0.03078	0.00846
2016-01-05	0.03360	0.01774	0.06315	0.11790	0.07223	0.03344	0.00755

그림 19 검색어에 대한 상대적 검색량 (변환 전)

→

날짜	계란	포도	배	사과	망고	옥수수	벼
2016-01-01	940.08641	428.65674	1492.43045	2503.01547	2023.05840	721.66660	157.99243
2016-01-02	794.05357	421.41805	1429.17053	2568.16374	2288.68710	757.23063	149.80955
2016-01-03	894.13642	465.47968	1527.05030	2863.37668	2390.65831	880.91793	160.19551
2016-01-04	1091.15486	1011.52920	1847.44132	3868.92609	2427.48124	968.72647	266.25815
2016-01-05	1057.47919	558.32383	1987.49437	3710.61894	2273.26553	1052.44357	237.61809

그림 20 상대적 검색량을 통해 구한 검색량 절대값 (변환 후)

25) 실제값으로 변환 시, 앞서 ‘전체 검색어 리스트 중 최대 검색량을 포함하는 검색어(total_max_key)’보다 높은 최대값을 가지고 있지만 결측값이 존재했던 검색어 목록(single_list)에 대해서는 개별로 절대검색량을 계산하여 전체 목록에 다시 포함시켰다(‘각주15번’내용).

② 실제값의 오차 계산

: 상대값을 실제값으로 변환하는 과정에서, 'NAVER Search Ad API의 최근 한달치 검색량이 1의자리 혹은 10의자리 값[26]을 반환하지 않는 점', '네이버 트렌드 API의 출력 결과는 상대값 소수점 5자리까지만 제공하는 점', '모든 검색어에 대해 개별로 절대값을 구해 계산한 것 아닌 샘플 5개의 평균값으로 계산[27]하는 점' 3가지 이유로 일반적으로 0.5~2%정도의 오차[28]가 발생한다. 다만 절대값이 실제값과 조금의 오차가 있더라도, 같은 기준에 따라 소수점 5자리까지의 상대값을 곱했기 때문에 다른 검색어와의 추세 차이는 영향이 없기 때문에[29] 본 분석에 목적 달성에 있어 영향을 끼치지 않는다(해당날짜의 자신의 절대값이 실제 검색량과 오차가 있더라도, 다른날짜/다른 검색어와의 검색량 순위가 달라지지는 않는다는 의미).

- 5개의 샘플 검색어에 대해서 오차율(검색어 개별로 한달검색량에 대해서 자신의 100%를 계산하여 곱했을 때와의 차이)[30]과 MSE(평균제곱오차)[31]을 계산[32]하여 평균을 구하면 다음과 같이 나온다.

```
final_df, error_per_average, MSE_average = real_amount(Food_list)
```

```
5개의 샘플을 추출하여 오차율과 MSE를 계산 중 입니다...
검색어 '보리'에 대한 오차율: 1.3644929970598807, MSE: 0.020061374665515677
검색어 '복숭아'에 대한 오차율: 1.2997757067689488, MSE: 0.04076739362568196
검색어 '쌀'에 대한 오차율: 0.07723488729245402, MSE: 0.0005544919448980186
검색어 '참외'에 대한 오차율: 0.6925850372051959, MSE: 0.008193106274616162
검색어 '포도'에 대한 오차율: 0.5945207503453135, MSE: 0.012911805659767218
```

```
# 오차율(5개 샘플 평균)
error_per_average
```

```
0.8057218757343586
```

```
# 평균제곱오차(5개 샘플 평균)
MSE_average
```

```
0.016497634434095805
```

그림 21 샘플을 통해 계산한 평균 오차율, 평균제곱오차 예시

③ 다른 유료사이트(블랙키위[33])와 비교

: 개별로 절대값을 곱한 것으로 추정되는 유료사이트 '블랙키위'와 비교했을 때, 샘플에 포함되지 않은 단어 '망고'에 대한 오차가 약 1.13%로, 샘플을 통해 계산한 0.8%와 큰 차이가 없다. 이는 1의 자리를 제외하고 반환하는 '블랙키위'의 데이터 제공방식, 실행 즉시 데이터를 가져오지 않고 시간적 제약 없이 서버를 통해 미리 검색어 개별로 한달치 절대값을 곱할 수 있는(세번째문제) 여건에 따른 차이로 발생하는 한계이다(또한 '④항목'에서 후술할 이유로 본 분석에서 사용한 동일한 기준에 의한 일괄실행이 더 합리적이라고 판단된다.).

---

26) 검색량값이 높을수록 제공하는 자리수의 최소단위가 높아진다.
27) 'NAVER Search Ad API'를 검색어 각각의 한달치 검색량을 이용하여 절대값으로 변환할 경우, 정확도가 조금 올라갈 수는 있지만, 모든 검색어를 하나씩 계산하는 것은 시간문제+api호출문제('NAVER Search Ad API'는 API의 최대 호출 가능 수치를 명시해놓지 않았으며, 일정 횟수 이상 반복 시 자동으로 차단하고 있음)로 인해 개별 실행이 아닌 일괄 실행하였다.
28) 첫 번째와 두 번째 이유는 API 제공형태에 따른 문제이므로 더 이상의 정확도 개선이 불가능
29) '⑤ 항목'에서 후술할 이유로, 개별로 절대값을 구해 계산하는 방법을 사용하지 않는 가장 중요한 이유
30) (실제값-예측값)의 절대값 / 실제값 * 100
31) (실제값-예측값)의 합계의 제곱 / 기간
32) 여기서 실제값은 검색어 개별로 한달치 검색량을 통해 계산한 값을 사용
33) 결제가 필요한 구독형 상품이다. 무료플랜의 경우 최근 1주일치 검색량만 제공하며, 검색시간당 검색량에 제한을 두고 있다.

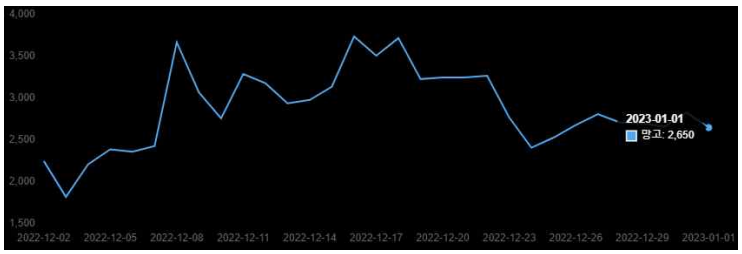


그림 22 유료 사이트 '블랙키위'에서 제공하는 최근 1주일 검색어의 값과 비교

	날짜	망고
2551	2022-12-26	2717.34384
2552	2022-12-27	2851.10237
2553	2022-12-28	2748.81644
2554	2022-12-29	2738.74521
2555	2022-12-30	2697.83083
2556	2022-12-31	2866.20922
2557	2023-01-01	2680.52091

그림 23 상대값→절대값 변환을 통해 계산한 '망고'의 검색량

④ 특정 검색어에 대해 오차율이 유독 높게 나오는 문제 ('NAVER Search Ad API'의 정보제공 오류로 추정[34])

- 'NAVER Search Ad API(네이버광고API)'의 최근 30일치 검색량의 정보제공 오류(추정)

: 샘플을 통한 변환값과의 오차율 계산 시, '그림21'의 '찌개'와 같이 오차율이 유독 높게 측정되는 검색어가 등장하는 경우가 있다[35]. 다만 이는 변환상의 오류가 아닌 'NAVER Search Ad API'의 정보제공 오류로 추정되므로, 상대값→절대값 변환방식을 수정하지 않고 오차율이 10%인 검색어가 샘플로 포함 됐을 경우 샘플을 재추출하도록 했다.

```
final_df, error_per_average, MSE_average = real_amount(Food_list)
```

```
5개의 샘플을 추출하여 오차율과 MSE를 계산 중 입니다...
검색어 '감자탕'에 대한 오차율: 4.4502991328732335, MSE: 35.013856147211555
검색어 '칼국수'에 대한 오차율: 3.7996992197729575, MSE: 20.972999552951645
검색어 '찌개'에 대한 오차율: 13.972027058336, MSE: 10.125649103153494
검색어 '빈대떡'에 대한 오차율: 4.289601885492152, MSE: 0.10776909678567274
검색어 '차돌박이'에 대한 오차율: 4.39154756862385, MSE: 1.268877483330225
```

그림 24 오차율이 10%인 검색어가 샘플로 포함된 경우. 기준이 되는 '샘플 5개의 평균값'이 올라가면서 모든 검색어에 대한 오차범위가 커짐

```
# NAVER Search Ad API의 호출결과
search_keyword('찌개')
```

```
34610
```

```
# 상대값을 이용해 변환한 절대값
final_df['찌개'].tail(30).sum()
```

```
29774.28143510992
```

그림 25 '찌개'에 대한 최근 1달 검색량 차이. 'NAVER Search Ad API'의 호출 결과와 '상대값→절대값 변환'을 통해 계산한 변환 결과 사이에 13.972% 수준의 차이가 존재

34) '⑤ 항목'에서 자세하게 후술

35) 5개 샘플 각각의 계산값을 평균 낸 값과 자신의 원래값을 비교하는 것이므로, 하나 이상의 이상치가 샘플에 포함되면 평균값 자체가 높아져 전체적인 오차율이 높아질 수 밖에 없다. '그림22'를 보면 실제로 'NAVER Search Ad API'의 호출결과와 계산한 변환결과에 차이가 존재한다.

- 검색량 절대값뿐만 아니라 상대적 순위에서도 차이 발생

: '회','찌개','비빔밥' 3개의 검색어에 대해서 각각 개별 월별 검색량으로 절대값을 구한 결과[36], 01/09 날짜에 대해서 '비빔밥>찌개>회' 순서로 검색량이 높았다(그림26). 반면 상대값→절대값 변환을 통해 산출한 결과에 의하면, 01/09 날짜의 검색량 수치가 '비빔밤>회>찌개' 순서로 높아 '회'와 '찌개'의 순서가 바뀐 것을 확인 할 수 있다(그림27). 이는 개별로 절대값을 계산하는 방법(첫 번째 방식)을 사용한 것으로 추정[37]되는 '블랙키위'의 순위와도 다른 결과이다(그림28, 그림29, 그림30. '비빔밥>찌개>회' 순서).

	날짜	회	찌개	비빔밥
2559	2023-01-03	1239.21077	1266.81259	1291.24401
2560	2023-01-04	1268.22995	1251.97558	1217.67898
2561	2023-01-05	1381.29601	1184.17870	1225.99728
2562	2023-01-06	1526.30826	1068.36761	1148.27309
2563	2023-01-07	1928.39528	1104.12068	1199.04943
2564	2023-01-08	1662.37223	1170.68114	1412.55267
2565	2023-01-09	1379.28892	1430.74092	1474.68003

그림 26 항목 각각에 대해 개별 월별 검색량으로 절대값을 구한 결과 (오차율 계산 시 사용했던 샘플 개별값을 구할 때 사용했던 방식)

	날짜	회	찌개	비빔밥
2559	2023-01-03	1313.44871	1089.81319	1320.89436
2560	2023-01-04	1344.20635	1077.04922	1245.64008
2561	2023-01-05	1464.04591	1018.72493	1254.14940
2562	2023-01-06	1617.74547	919.09500	1174.64046
2563	2023-01-07	2043.92049	949.85264	1226.58275
2564	2023-01-08	1761.96069	1007.11326	1444.98859
2565	2023-01-09	1461.91858	1230.83741	1508.54256

그림 27 상대값을 변환하여 절대값을 구한 결과(샘플의 평균값을 통해 변환한 절대값). **오차율10% 이상인 검색어는 제외될 것이므로 실제로 그림과 같이 큰 오차는 발생하지 않는다**.

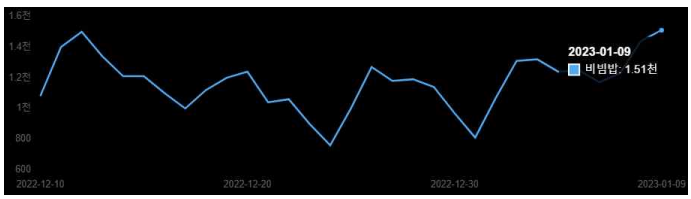


그림 28 블랙키위 '비빔밥' 검색결과(1위)

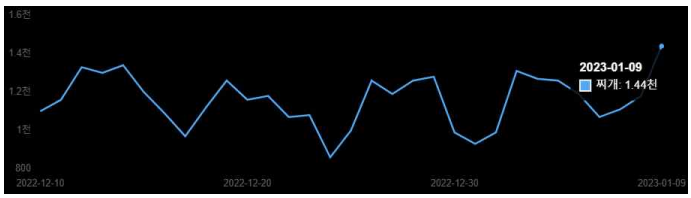


그림 29 블랙키위 '찌개' 검색결과(2위)

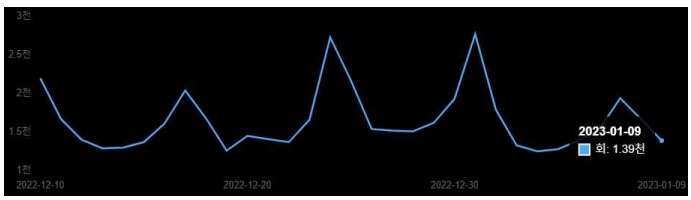


그림 30 블랙키위 '회' 검색결과(3위)

36) 오차율 계산 시 사용했던 샘플 개별값을 구할 때 사용했던 방식

37) 검색량의 절대값을 더 정확하게 산출하기 위한 방법이지만, '③ 항목'에서 설정한 이유로 본 프로젝트에서는 샘플 5개의 평균값으로 대체했었다.

⑤ 절대적 검색량의 정확성 vs 상대적 순위의 일관성

: '④항목'에서 월별 검색량 합계량을 제공하는 'NAVER Search Ad API'의 일부검색어[38]에 대해서 정보제공 오류가 추정된다고 언급했었다. 상대적 순위를 정확하게 비교하기 위해 '회','찌개','비빔밥' 3개의 검색어에 대해서만 상대적 검색량을 비교해본 결과(그림31), 전체 검색어에 대해서 구했던 상대적 검색량(그림32)과 비율이 정확하게 일치[39]했는데, 2023/01/09 날짜에 대한 3개 검색어의 검색량 순위는 '비빔밥>찌개>회' 순서로 나타나는 것을 확인할 수 있다. 'NAVER Search Ad API'를 이용해 변환을 진행한뒤 결과가 달라졌다면, '④항목'에서 나타났던 오차[40]는 'NAVER Search Ad API'의 일부 검색어에 대한 정보제공 오류[41]에 의해 발생하는 것으로 추정할 수 있다.

→ 'NAVER Search Ad API'를 신뢰하여, 개별 검색어에 대한 정확한[42] 절대적 검색량을 구하는 것이 목적이라면 개별 월별 검색량으로 절대값을 구한 결과를 신뢰하면 되겠으나('비빔밥>찌개>회'), 본 프로젝트에서는 절대적 검색량의 정확도가 일부 떨어지더라도 다른날짜/다른검색어 간의 검색량 순위 일관성을 유지('비빔밥>회>찌개')하는 것이 목적이기 때문에 '④항목'의 오차를 'NAVER Search Ad API'의 일부 검색어에 대한 정보제공 오류로 판단[43]하였다. 결과적으로 '④ 항목'에서 언급한 일부 검색어에 대한 오차에 대해서 '③항목'에서 언급한 시간적 제약에 따른 한계뿐만 아니라 '검색량 순위 일관성 유지'를 위해서, 개별로 월별 검색량을 구해 절대 검색량을 구하는 방법이 아닌 '추출한 샘플의 평균값'이라는 동일한 기준에 의해 절대검색량을 구하는 것이 전체적으로 더 정확한 결과라고 판단하였다.

	날짜	회	찌개	비빔밥
2559	2023-01-03	28.83407	23.92366	28.99697
2560	2023-01-04	29.50895	23.64440	27.34465
2561	2023-01-05	32.13870	22.36444	27.53083
2562	2023-01-06	35.51314	20.17686	25.78543
2563	2023-01-07	44.86851	20.85175	26.92576
2564	2023-01-08	38.67814	22.10844	31.71980
2565	2023-01-09	32.09215	27.01885	33.11612

그림 31 3개의 검색어에 대해서만 상대적 검색량을 비교해본 결과

	날짜	회	찌개	비빔밥
2559	2023-01-03	0.14818	0.12295	0.14902
2560	2023-01-04	0.15165	0.12151	0.14053
2561	2023-01-05	0.16517	0.11493	0.14149
2562	2023-01-06	0.18251	0.10369	0.13252
2563	2023-01-07	0.23059	0.10716	0.13838
2564	2023-01-08	0.19878	0.11362	0.16302
2565	2023-01-09	0.16493	0.13886	0.17019

그림 32 전체 검색어에 대해서 구했던 3개 검색어의 상대적 검색량

⑥ 연령대/성별 특성에 따른 비교는 제공하지 않음

: '통합 검색어 트렌드 API'는 특정 집단의 다른날짜/다른검색어에 대한 검색량 비교는 제공하고 있으나 같은 단어에 대한 집단별(연령대/성별) 비교값을 제공하고 있지 않다. 'NAVER Search Ad API'에서도 '기기'에 따른 분류만 제공하고 있을 뿐 집단별(연령대/성별) 분류를 제공하고 있지 않기 때문에, '타 집단 간 상대값 비교' 혹은 '상대값→절대값변환' 작업은 진행할 수 없다[44].

---

38) 오차율이 높게 측정되는 검색어
39) 194.58 : 1 비율로 소수점 2번째 자리까지 동일
40) 특정 검색어에 대해 오차율이 유독 높게 나오는 문제
41) 한달치 검색어가 1의자리/10의자리 값을 반환하지 않음', '네이버 트렌드 API의 출력 결과는 상대값 소수점5자리까지만 제공' 2가지 이유로 발생하는 오차의 범위를 넘어서는 오차
42) 다만 이 역시 추정값이며, '각주 34)'의 이유로 정확성을 신뢰하는 것은 모순적이다.
43) 다만 'NAVER Search Ad API'가 아닌 '통합 검색어 트렌드 API'의 오차일 가능성도 존재하는데, 애초에 'NAVER Search Ad API'를 이용한 절대값 계산방법은 '통합 검색어 트렌드 API'가 정확하다는 전제('절대값'x'비율'에서 '비율'의 출처)하에 이루어지는 작업이므로, '통합 검색어 트렌드 API'만의 오류이더라도 결과적으로 'NAVER Search Ad API'를 통한 절대값 계산의 오류로 이어지게 된다.
44) 검색어에 대한 '동일 집단 내' 검색량 상대값을 구하는 것만 가능

3.4. 뉴스 크롤러를 이용해 수집한 뉴스제목/네이버내용

① 뷰티플수프(BeautifulSoup)

: 네이버 뉴스의 특성상 url을 통한 페이지 이동(파싱)이 자유롭기 때문에, 속도가 빠른 정적 웹크롤링 라이브러리인 BeautifulSoup을 이용해 뉴스 크롤러를 작성하였다. 이 크롤러는 네이버 검색창에 특정 검색어를 입력했을 때 나오는 뉴스를 가져오며, '기간'을 설정한 크롤링이 가능하다. 관련성 없는 뉴스의 수를 줄이기 위해 검색어 양 옆에 쌍따움표("")를 추가[45]하여 검색하도록 설정하였다.

그림 33 검색어 그대로 검색한 결과

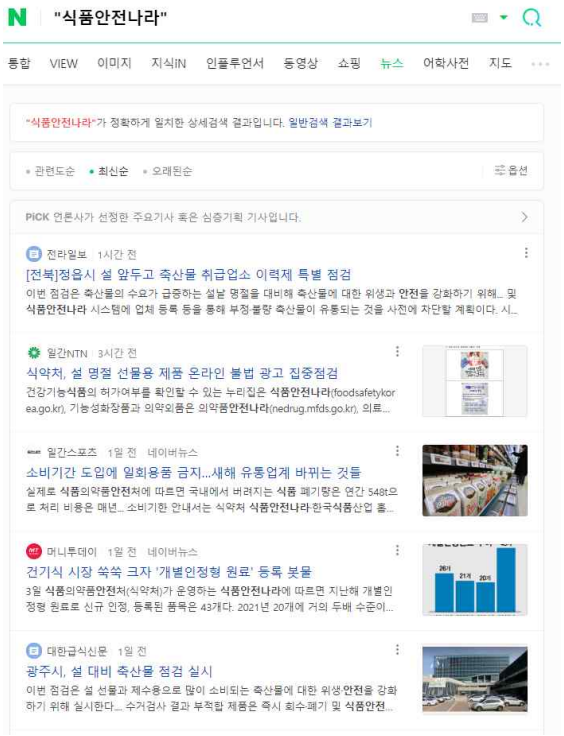


그림 34 검색어 양 옆에 쌍따움표("")를 추가하여 검색한 결과

② 뉴스 크롤링 과정 (페이지단위 크롤링을 날짜단위 크롤링으로)

: 검색어, 시작날짜, 종료날짜를 입력하면 크롤링 실행(본 프로젝트에서는 '입력'방식으로 사용자의 직접입력을 통한 방법이 아닌, 검출된 이상검색어와 이상일자를 이용하여 자동으로 크롤러를 작동하도록 설정)

```
# main 안에 인자는 '검색어', '시작일', '종료일'
if __name__ == "__main__":
    search_keyword = input(f'검색어를 입력하세요: ')
    start_date = input(f'시작일자를 입력하세요(YYMMDD): ')
    end_date = input(f'종료일자를 입력하세요(YYMMDD): ')
    all_news_df, naver_news_df = main(search_keyword, start_date, end_date)
```

그림 35 뉴스 크롤러 실행 코드(실제 입력값은 직접입력이 아닌 자동입력)

45) 네이버 검색엔진의 상세검색 기능 중 하나로, 검색어 양 옆에 쌍따움표("") 추가 시 '정확하게 일치한 결과'만 반환한다.

- 모든 뉴스 탐색 완료 시 종료 (종료조건)

: 네이버 뉴스 크롤러는 '페이지단위(url의 'start' 파라미터를 10단위로 증가)'로 작성되어 있다. 다만 네이버 검색결과는 실제 페이지가 없더라도, 에러를 반환하지않고 "'00'에 대한 검색결과가 없습니다." 문장을 반복적으로 보여주기 때문에 별도로 종료조건을 지정해주어야 한다. 따라서 "검색결과 화면 내 '뉴스'가 하나도 없을 경우"를 종료조건으로, 현재까지의 상황을 저장하고 크롤러를 종료하도록 설정하였다.

그림 36 페이지에 더 이상 뉴스가 없는 경우 (종료조건)

- 400페이지[46] 제한 문제

: 네이버 뉴스의 경우 뉴스 검색결과를 한번에 400페이지까지만 제공하고 있다. 따라서 뉴스량이 많은 검색어에 대해 긴 기간동안의 뉴스를 크롤링할 경우, 400페이지 이후의 뉴스를 가져오지 못한다[47]. 이를 해결하기 위해 검색도중 400번째 페이지를 넘어갈 경우, 다시 현재 크롤링 중인 날짜를 시작일[48]로하는 재귀함수 구조를 사용하여 400페이지가 넘는 검색결과에 대해서도 모든 뉴스를 정상적으로 가져 온 뒤 크롤러가 종료하되도록 설정하였다. 재귀과정에서 발생하는 400페이지(재귀함수 시작일자) 뉴스의 중복문제에 대해서도 'drop_duplicates' 함수를 통해 처리하였다.

그림 37 네이버 검색엔진은 검색 일회당 400페이지의 뉴스만 제공

46) 10건씩 400페이지. 총 4000건의 뉴스

47) 400페이지가 넘어갈 경우, 오류가 발생하지는 않으며 400번째 페이지를 반복 출력한다.

48) 종료일자는 동일

③ ‘모든 뉴스’의 ‘본문 내용 제외(제목만)[49]’ 크롤링

: 지정한 날짜의 특정 키워드 검색결과 출력되는 뉴스 상세검색 결과 시 출력되는 모든 언론사의 ‘날짜’, ‘제목’, ‘링크’, ‘언론사명’ 태그(HTML)를 추출하여 저장하였다. 이때 최근 1주일 이내 게시된 기사의 경우 ‘날짜’가 ‘yyyy-mm-dd’ 형태가 아닌 ‘00시간/00일 전’과 같은 형태로 출력되는데, 작업 실행시점을 기준으로 ‘00시간/00일 전’을 계산하도록 하여 ‘yyyy-mm-dd’형태로 변환하였다 (현재시간-'00시간 전'이 양수면 오늘날짜, 현재시간-'00시간 전'이 음수면 어제날짜, '00일 전'이면 오늘날짜-00일).

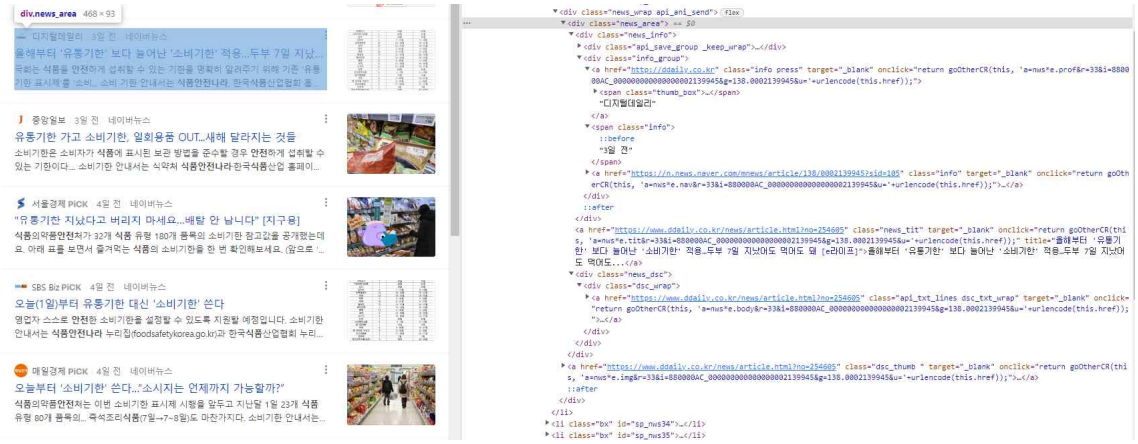


그림 38 네이버 뉴스 검색창 HTML태그

	A	B	C	D	E
1	date	title	link	press	
2	2017-01-23	식품안전정보	http://www.k	정책브리핑	
3	2017-01-23	식품안전정보	http://app.yo	연합뉴스	
4	2017-01-23	식약처, '식품	http://www.k	머니투데이	
5	2017-01-23	식약처, 식품	http://www.k	파이낸셜뉴스	
6	2017-01-23	식품안전정보	http://www.k	뉴시스	
7	2017-01-23	식약처, '식품	http://www.y	약업신문	
8	2017-01-23	식품안전정보	http://www.k	쿠키뉴스	
9	2017-01-23	식약처, 23일	http://www.g	글로벌이코노믹	
10	2017-01-23	식품안전정보	http://www.v	한국경제TV	

그림 39 네이버 뉴스 검색창 HTML 구조로부터 ‘날짜’, ‘제목’, ‘링크’, ‘언론사명’을 추출하여 csv파일로 저장한 결과

49) 뉴스 링크 클릭 시 언론사 자체 페이지로 넘어가는 경우 언론사 개별로 HTML태그에 맞춰 크롤러를 작성해줘야 한다. 시간관계 해당 작업은 진행하지 않음.

④ ‘네이버 뉴스’의 ‘본문 내용 포함(제목 포함)’ 크롤링

: 지정한 날짜의 특정 키워드 검색결과 출력되는 뉴스 상세검색 결과 시 출력되는 뉴스 중 ‘그림40’처럼 자체 언론사뿐만 아니라 ‘네이버뉴스’에도 동시에 게재한 경우, ‘제목’뿐만 아니라 ‘본문 내용’도 추출할 수 있다(‘날짜’,‘제목’,‘링크’,‘본문내용’,‘언론사명’)[50]. 날짜가 ‘yyyy-mm-dd’형식이 아닌 ‘yyyy.mm.dd’와 같은 형식로 출력되는 일부 결과에 대해서는, ‘yyyy-mm-dd’ 형식으로 변환해 저장하도록 설정했다.

그림 40 ‘네이버뉴스’에도 동시 게재된 뉴스

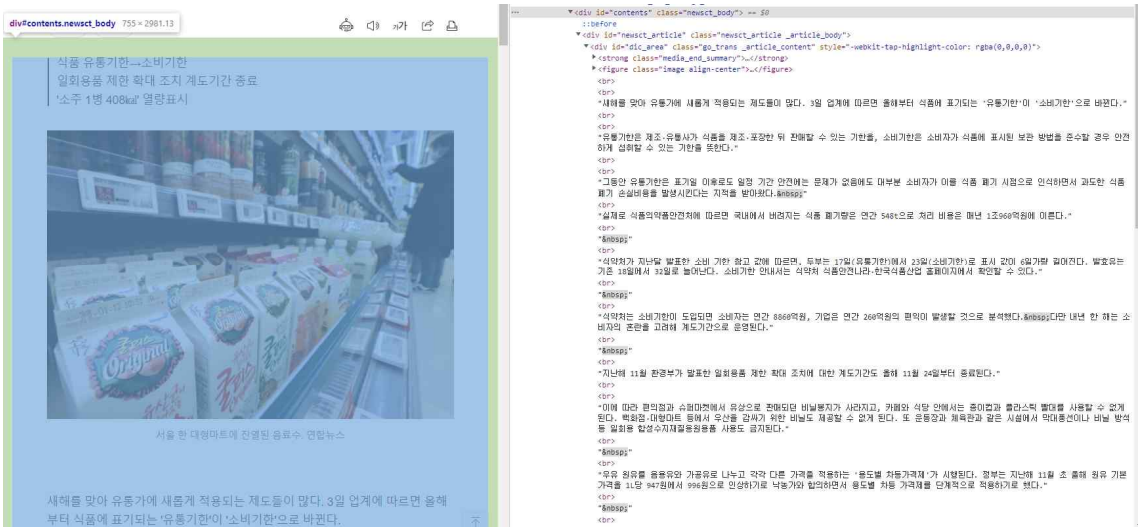


그림 41 네이버 뉴스 HTML태그

	A	B	C	D	E
1	date	title	link	content	press
2	2017-01-23	식품안전정	https://n.n	[식품안전	정책브리핑
3	2017-01-23	식품안전정	https://n.n	[(서울=연합	연합뉴스
4	2017-01-23	식약처, '식	https://n.n	[머니투데이	머니투데이
5	2017-01-23	식약처, 식	https://n.n	[국민에게	파이낸셜뉴스
6	2017-01-23	식품안전정	https://n.n	[서울=뉴시	뉴시스
7	2017-01-23	식품안전정	https://n.n	[한국경제	한국경제TV
8	2017-01-23	찌개-국 끓	https://n.n		코메디닷컴
9	2017-01-23	식품안전 ?	https://n.n	[국민에게	파이낸셜뉴스
10	2017-01-23	식품안전정	https://n.n	[	부산일보

그림 42 네이버 뉴스 HTML 구조로부터 ‘날짜’, ‘제목’, ‘링크’, ‘본문내용’, ‘언론사명’을 추출하여 csv파일로 저장한 결과

50) 본문 내용을 가져오기 위해서는 언론사 개별로 HTML태그에 맞춰 크롤러를 작성해줘야한다. 다만 자체 언론사뿐만 아니라 네이버뉴스에도 동시에 게재한 경우 언론사별 HTML이 아닌 ‘네이버뉴스’의 HTML구조에 따른 크롤링이 가능해 ‘본문내용’을 가져올 수 있다.

4. 분석 방법

4.1 네이버 검색량 데이터 기반 시계열 예측모델(Prophet)

: '3.과정'을 통해 수집한 2016년 이후 검색량 절대값 데이터에 대해서, 페이스북에서 만든 시계열 예측 라이브러리 Prophet(과거 fprophet)을 이용하여 시계열 예측 작업을 진행하였다. 이는 이후 '4.2 과정'에서 구축할 이상탐지모델(Anomaly Detection)의 전작업으로, 실제값이 '이상값[51]'인지를 판별하기 위한 비교 데이터인 '일반적인 검색량'을 구하기 위한 작업이다.

① Prophet 라이브러리

: 페이스북에서 공개한 시계열 예측모델인 Prophet 모형은 트렌드(growth)[52], 계절성(seasonality)[53], 휴일(holidays)[54] 3가지의 main components로 이루어진 가법 회귀모델[55]이다. g(t)는 비주기적 변화를 반영하는 추세 함수(=트렌드), s(t)는 계절성(주기적인 변화)을, h(t)는 휴일(불규칙 이벤트)[56]의 영향력을 나타낸다. Et은 필수 구성요소가 아니 사용자의 선택적 지정 사항이며, 모델링이 되지 않는 특이한 변화를 모델링한다.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

그림 43 Prophet 모델의 예측 원리

② 시계열예측모델의 정확도 평가(Train Score를 쓰게되는 이유)-이 모델의 신뢰성

: 이상탐지 모델의 신뢰성에 있어서 가장 중요한 것은 '시계열 예측 모델의 정확성'이다. '4.2 항목'에서 후술하겠지만 이상탐지모델은 '당일의 일반적인 검색량'보다 현저히 높은 검색량이 측정됐을 때, 이를 검출해내는 원리이다. 이상 검색량 검출의 전제인 '당일의 일반적인 검색량'을 제대로 산정하지 못한다면 해당 모델의 결과는 신뢰성을 보장할 수 없다. 성능 측정 지표로는 MAE[57]와 MAPE[58]를 사용했다.

51) 데이터의 품질과 관련된 단어인 '이상치'와 구별되는 단어로, 실제값으로서 의미있는 수치이다.
52) 변경점을 기반으로 추세의 변화를 자동으로 감지
53) 푸리에 급수를 사용하여 모델링된 연간 계절 성분(yearly), 더미변수를 사용하는 주간 계절 성분(weekly)
54) 사용자가 제공한 중요한 공휴일 목록(holidays). 프로젝트에서는 대한민국 휴일데이터 사용
55) 가법모형(Additive Model)은 종속변수를 각 개별 설 명변수들만의 비모수적 함수의 합으로 표현하는 방법
56) 사용자 지정사항이며, 본 프로젝트에서는 대한민국의 공휴일을 추가
57) 실제 값과 예측 값의 차이(Error)를 절대값으로 변환해 평균화
58) MAE를 퍼센트로 변환한 값

- **Test Score(2022년 이전 데이터로 훈련하여 2022년 데이터 테스트)**

: 일반적인 시계열모델에서 사용하는 성능평가 방법. 성능은 (MAE/MAPE(Test Score 약 24%)) 정도 수준이며, '검색량'라는 데이터의 특성상 '추세(트렌드)'에 민감하기 때문에 최근 1년 데이터가 훈련모델에서 빠진 상황에서의 성능은 떨어질 수 밖에 없다[59].

- **Train Score[60] (예측일의 검색량값도 훈련에 포함)**

: 일반적인 시계열 모델과 달리, 본 프로젝트에서 구축한 시계열모델은 장기간이 아닌 "당일의 일반적인 예상 검색량보다 현저히 높은 검색량"을 찾아내는 것에 목적[61]이 있으므로, 훈련 데이터에 예측일의 데이터를 포함시킨 방법[62]으로 모델을 구축하였다. 성능측정 결과 Test Score와 비교했을 때 예측모델의 성능이 올라간 것을 볼 수 있다(그림45).

→ 주가예측과 같이 예외적인 상황에 대해서도 정확한 예측을 해야하는 '시계열 목적 모델'과 달리, 일반적인 상황이라고 예측을 하는 것이 '이상탐지 모델'에 목적이므로('비주기적 이상 검색량'에 대한 '부정확도'가 곧 '이상정도'), '그림 45'과 같이 '비주기적 이상 검색량'에 대해서 정확도가 낮아더라도 문제되지 않는다[63].

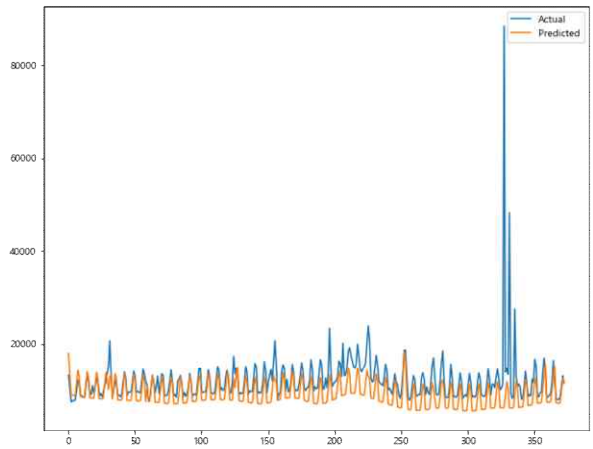


그림 44 Test Score 측정 결과(치킨). MAE: 2869.769 / MAPE: 21.346

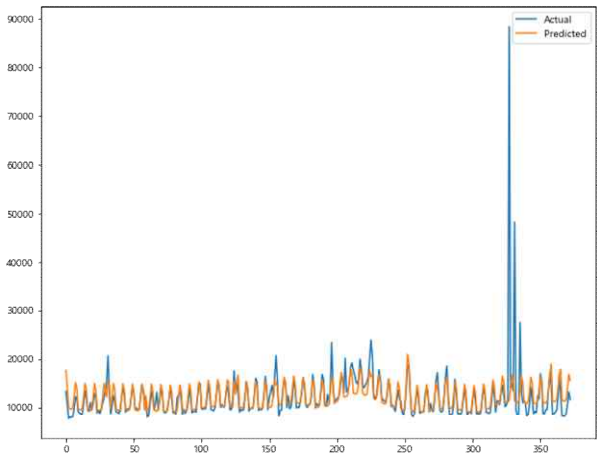


그림 45 Train Score 측정 결과(치킨). MAE: 1504.908 / MAPE: 10.423 (10.9%↓)

59) 실제로 일반적인 계절성만 가지는 검색어에 대해서는 Test Score가 좋은 편이였지만, 2022년에 전체적인 검색량 상승이 있는 경우에 대해서는 Train Score에 비해 예측 정확도가 떨어지는 편이다('양식'의 Test Score: MAE 69.915 / MAPE 7.544, Train Score: MAE 63.129 / MAPE 6.876).

60) 측정목표 날짜의 데이터까지 훈련모델에 포함한다는 의미이며, 실제로 본 프로젝트의 모델은 당일의 검색량까지 훈련모델에 포함 시킨다.

61) 예시: '어제의 예상 검색량'과, '어제의 실제 검색량'을 비교

62) Train Score를 측정할 때와 동일한 방식

63) 결과적으로 본 시계열모델의 목적은 '일반적인 예상 검색량'을 찾아내는 것에 있으므로 다음과 같이 평가한 것이며, 모델의 목적이 '비주기적 이상 검색량'까지 정확하게 예측하기 위한 것이라면 적절한 평가가 아니다.

③ 과거 비주기적 이상치 이력(해결과제)-이상치가 Train데이터에 포함된 경우

: 본 프로젝트는 소수 항목에 대한 정확한 시계열예측을 하는 것을 주 목적으로 두고있지 않다(비 주기적 오차까지 정확하게 예측하지 않겠다는 의미). Prophet 모형에 대한 파라미터수정/이벤트발생처리와 같은 방법을 통해 정확도를 높이는 작업은 가능하긴 하지만(그림46, 그림47), 코로나 확진자 수, 주가예측과 같이 하나의 항목에 대한 정확한 예측을 하는 것이 목적이 아니며 1000개가 넘는 검색어를 대상으로 모니터링을 진행하는 것이 목적이기 때문에, 검색어 하나하나에 대해서 정확도를 높이는 작업을 진행하지는 않았다[64]. 이로인해 '비주기적 과거 이상치'가 있는 경우 모델의 정확도가 떨어져 보인다는 문제가 있지만, 우연히 과거 이상치와 같은 날짜에 사건이 발생하지 않는 한[65] '이상탐지' 자체는 정상적으로 이루어질 것이다('4.2의 ③항목'에서 자세하게 후술).

→ (검색어 개별 예측 성능 향상 예시) 다른 이상 검색량과 연관성 없는 이상 검색량(=과거 비주기적 이상 검색량)을 예외처리하는 방법(그림 46), 이상 날짜의 뉴스를 크롤링하여 텍스트분석을 진행하여 유사한 키워드가 뉴스에 출현할 경우 holiday 파라미터로 사용하는 방법 등.

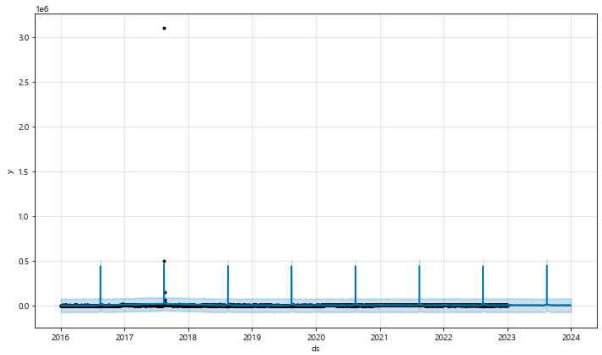


그림 46 검색어 '계란'에 대한 예측모델. '살충제 계란'으로 이슈가 있던 '2017년 8월 15~16일'이라는 이상치로 인해 매년 8월15일 경 예측값을 높게 예측

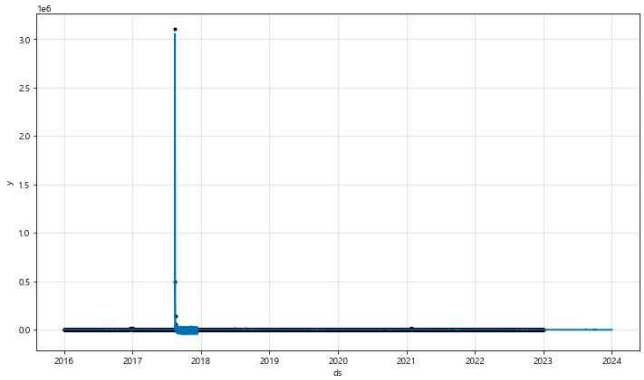


그림 47 '그림 46'의 2017년 8월15~16일을 예외처리한 결과(=holiday에 추가). 단발성 이상치(과거 비주기적 이상 검색량)를 예외처리할 경우 이를 가중치에 포함시키지않기 때문에 예측 성능이 향상 될 수 있음

64) 모든 검색에 대해 일괄적으로 정확도를 높이는 작업은 최대한 진행하였고, 검색어 하나하나에 대해 개별적으로 정확도를 높이는 작업도 가능하나 이는 프로젝트의 진행 범위를 벗어났다고 판단하여 진행하지 않음

65) 주기성이 있는 이상 검색량는 문제되지 않으며 모델의 학습과정에 이미 포함되어 있다. -예: 매년 빼빼로데이가 때 '빼빼로'에 대한 검색량이 높을 것으로 예측하고 있으며, 이는 문제되지 않음.

4.2 시계열 데이터 이상탐지모델(Anomaly Detection)

① 이상탐지 원리

: 이상탐지모델은 '당일의 일반적인 예상 검색량(yhat)'보다 현저히 높은 검색량이 측정됐을 때, 이를 검출해내는 모델이다. 측정 오차가(error)[66]이 예측범위(uncertainty)[67]보다 일정 수준[68] 이상 클 경우, 해당 날짜의 검색량을 이상값으로 검출[69]한다. '그림48'과 '그림49'는 '라면'의 이상검색량 검출결과인데, 코로나19의 초기 유행으로 생필품 수요가 증가한 2020년 2월말~3월초 기간동안 검색량이 급증했던 것을 확인할 수 있다.

ds	y	yhat	yhat_lower	yhat_upper	error	uncertainty	anomaly
2016-12-19	9200.78582	2718.91241	1411.14403	3975.81837	6481.87341	2564.67434	Yes
2017-06-14	7429.96969	2664.79770	1497.75062	3967.45331	4765.17199	2469.70269	Yes
2020-02-21	12993.29507	3671.25674	2319.05000	5113.09050	9322.03833	2794.04051	Yes
2020-02-22	18371.96133	3664.45628	2179.02964	4962.39402	14707.50505	2783.36438	Yes
2020-02-23	22351.26301	4037.35605	2658.17197	5305.59377	18313.90696	2647.42180	Yes
2020-02-24	27699.29497	4042.87353	2820.71273	5258.60480	23656.42143	2437.89207	Yes
2020-02-25	24426.20349	4039.89781	2623.41539	5371.31033	20386.30568	2747.89495	Yes
2020-02-26	18376.05728	4006.93497	2655.65356	5277.54877	14369.12231	2621.89521	Yes
2020-02-27	15721.45440	3987.19127	2745.95221	5289.89682	11734.26313	2543.94462	Yes
2020-02-28	10988.83907	3799.10136	2462.62173	5163.93714	7189.73772	2701.31542	Yes
2020-02-29	9305.82990	3707.78491	2466.52762	5008.04188	5598.04498	2541.51426	Yes
2020-03-01	7824.97296	3757.03545	2458.66419	5038.81843	4067.93751	2580.15424	Yes

그림 48 '라면'의 이상검색량이 검출된 날짜. anomaly가 Yes인 열만 추출

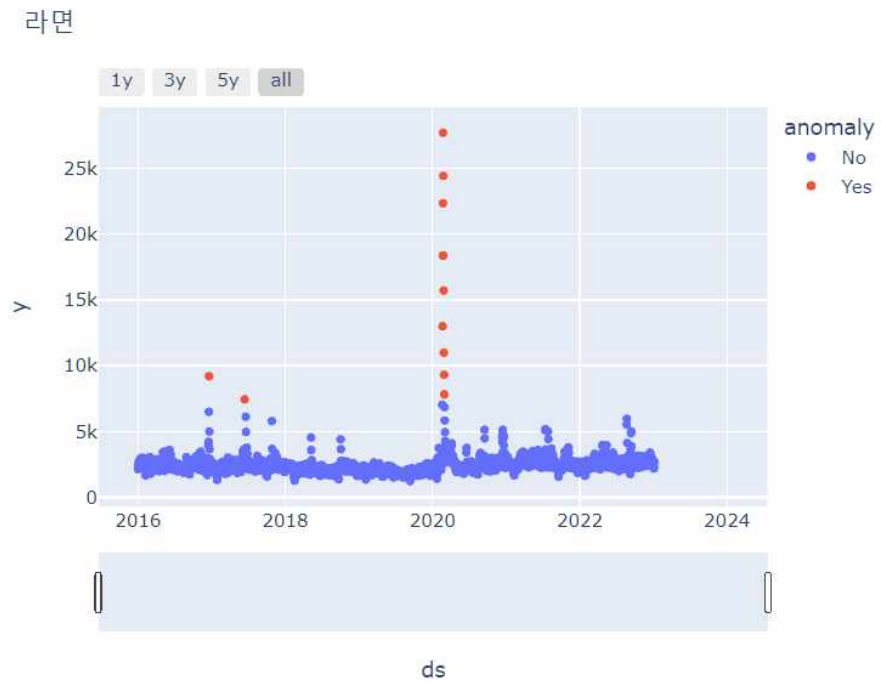


그림 49 검색어 '라면'에 대한 이상검색량 검출결과. anomaly가 Yes인 날짜를 빨간색으로 표시

66) (실제값(y) - '당일의 일반적인 예상 검색량(yhat)')
67) ('당일의 일반적인 예상 검색량 상한(yhat_upper)' - '당일의 일반적인 예상 검색량 하한(yhat_lower)')
68) 본 모델에서는 1.5배를 적용. ②에서 후술.
69) anomaly 열에 Yes를 할당

② 기준설정(한계점1)

: "'당일의 일반적인 예상 검색량(yhat)'보다 현저히 높은 검색량"을 판단함에 있어 '현저히 높은'에 대한 기준[70]을 설정해야 한다. 현재는 정성적인 판단에 의해 실제값이 예측범위(예측상한-예측하한)보다 '1.5배(가중치)' 이상 '높은 경우[71]'로 설정하였지만, 어느정도의 수치부터 이상치로 봐야할지에 대한 정량적인 기준이 부족하다. 다만, 무분별한 이상검출을 제한하기 위한 방법으로, 실제값이 예측범위(예측상한-예측하한)보다 '1.5배' 이상 '높더라도, 검색량 절대값 자체가 높지않은 경우[72]는 이상검출여부 컬럼값을 "Yes"에서 "Low"로 변환하여 이상검색량으로 검출되지 않도록 설정하였다(그림50).

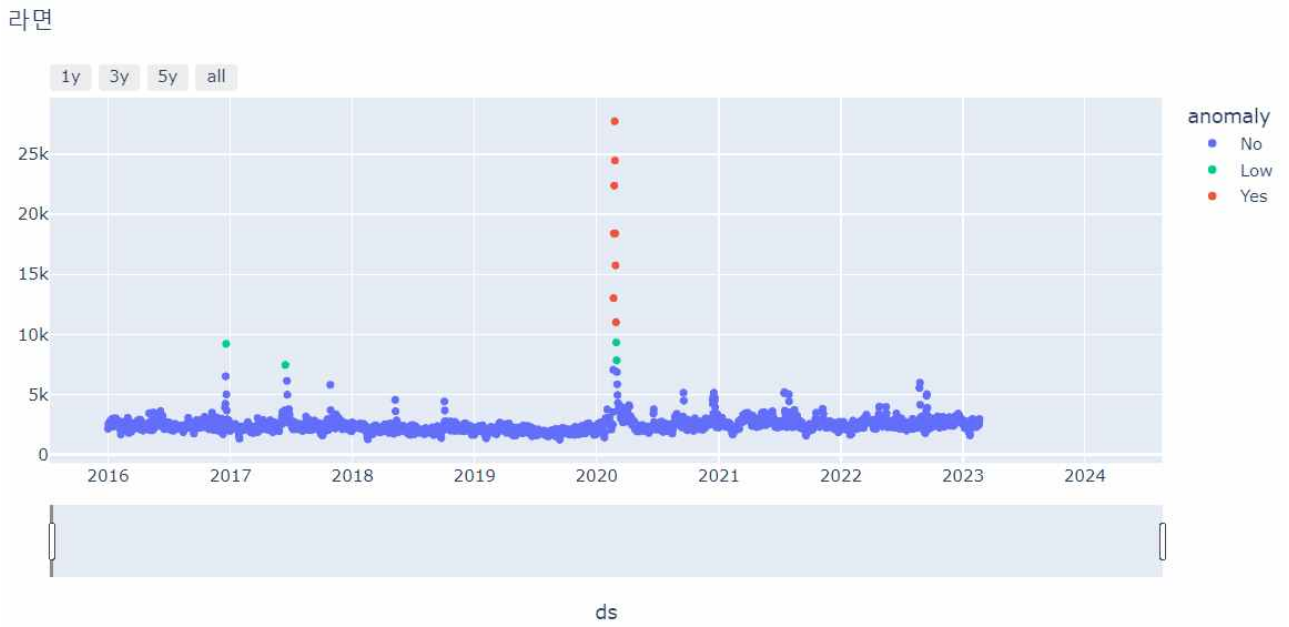


그림 50 검색어 '라면'에 대한 이상검색량 검출 결과. 검색량 절대값이 10000건 이하인 Anomaly값 Yes에 대해서 초록색으로 변환

③ '이상치(비주기적 이상검색량)'와 '관심사(주기적인 높은 검색량)'의 구분문제(한계점2)

: 실제 '비주기적 이상검색량'과 해당 시기에 '관심사(주기적으로 검출되는 높은 검색량 값)'에 대한 구분이 어렵다는 문제가 있다. 즉 본 프로젝트의 모델은 비주기적 이상 검색량뿐만 아니라, 사람들이 해당 시기에 관심이 있고 그 시점에 검색량이 급증하는 항목(주기인 높은 검색량) 중 일부를 결과로 포함하여 반환하기도 한다. 이는 Prophet 모형이 3개의 main component[73]로 가중치가 분산되는 구조적인 문제로 인해 발생하는 문제이며, 특히 주기적 이벤트 중 기간이 짧고 상승폭이 가파른 큰 항목에 대해서 더 자주 발생하는 문제이다.

---

70) 측정 오차가(error)이 예측범위(uncertainty)보다 일정 수준 이상 클 경우의 '일정 수준'을 산정하는 가중치
71) '방문자 수'와 같은 케이스는 낮은 경우도 고려해야하지만, 검색어의 경우 낮은 경우는 중요하지 않다고 판단
72) 검색량 절대값 자체가 낮은 단어의 경우 실제 이슈가 없음에도, 다른 사용자의 크롤러로 작동과 같은 이유만으로도 검색량의 증가폭이 높게 측정될 수 있으며, 뉴스로 기사화 될 정도의 이슈가 아닌 항목은 제외하는게 맞다고 판단하여 '검색량 절대값 10000만건 이하'인 항목에 대해서는 Anomaly값을 Yes에서 Low로 변환하였음
73) 트렌드(growth), 계절성(seasonality), 휴일(holidays)

(예시)

- (그림 52) 빼빼로데이가 공휴일이라면 ‘holiday’, 늘 같은 요일이라면 ‘weekly’에 대해서도 가중치를 받겠지만, 요일이 늘 달라지기 때문에 yearly(매년 11월11일 인근)에서 높은 가중치를 받더라도, weekly를 비롯한 다른 가중치는 실제보다 낮게 산정될 수 밖에 없다.

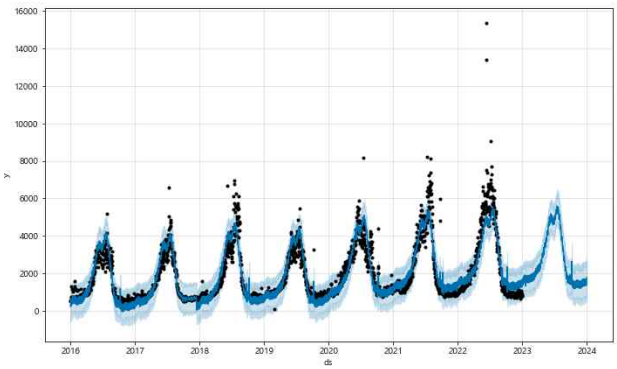


그림 51 ‘수박’에 대한 예측모델. 계절성을 어느정도 반영하기는 하나, 민감도를 아주 높게 설정하지 않는 이상(민감도를 높게 설정할 경우 예측 폭이 너무 커져 부적합) 전반적인 추세를 기반으로 예측하기 때문에, 검색량 값이 최고점에 가까운 1~2일 대해서 이상치로 반환하고 있음

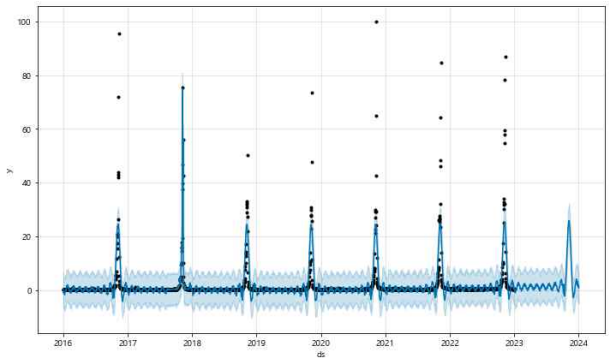


그림 52 ‘빼빼로’에 대한 예측모델. ‘yearly’에 대한 가중치로 11월 11일 인근 검색량을 높게 예측하지만, ‘수박’과 다르게 서서히 올라가는 추세(trend)도 없으며 공휴일(holiday)도 아니고 요일(weekly)이 매번 달라지기 때문에, 값을 어느정도 높게 예측하더라도 실제값만큼 높게는 예측하지 못하기 때문에 이상 검색량으로 검출된다. 2017년 11월11은 주말(holiday)이였기 때문에 holiday 가중치도 함께 적용받아 상대적으로 정확하게 예측.

4.3. 한국어 형태소 분석기를 이용한 뉴스 내 주요 키워드 추출

(‘4.2 항목’을 통해 검출한 ‘특정 검색어’의 ‘이상일자’에 대해, ‘3.4 항목’의 크롤러를 이용하여 수집한 뉴스 데이터를 대상으로 텍스트 분석을 진행하였다.)

① 형태소분석기

: 한글 자연어 처리를 위한 형태소 분석기로 ‘KoNLPy’를 사용했다. KoNLPy 라이브러리의 내의 한나눔(hannanum), 꼬꼬마(kkma), 코모란(komoran), 은전한닢(mecab), Okt(구 Twitter) 5개의 형태소분석기를 비교해보았으며, 최종적으로 대량데이터에 대한 처리속도가 빠르고(그림52), 사용자사전을 추가할 수 있는 은전한닢(mecab)을 사용하는 것으로 결정했다.

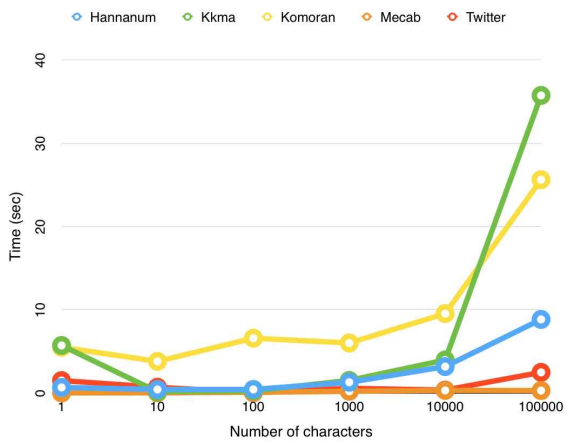


그림 53 형태소분석기 속도 비교(KoNLPy 공식)

keyword_hannanum	keyword_kkma	keyword_komoran	keyword_mecab	keyword_okt
[복면가왕, 모아니면, 조관우깨진, 소주병, 사연10여, 바늘]	[복면, 조관, 조관우, 소주병, 사연, 사연10여, 10, 바늘]	[복면, 아니, 면도, 조관우, 소주병, 사연, 10, 바늘]	[복면, 가왕, 조관우, 소주병, 사연, 바늘]	[복면, 조관우, 주병, 사연, 바늘]
[경기도의회, 더민주, 조기임시회, 개최여부, 15일, 결정]	[경기, 경기도, 경기도의회, 의회, 민주, 조기, 조기임시회, 임시회, 개최, 개...	[경기도의회, 민주, 조기, 임시회, 개최, 여부, 결정]	[경기도, 의회, 민주, 조기, 임시회, 개최, 여부, 결정]	[경기도, 의회, 민주, 조기, 임시회, 개최, 여부, 결정]

그림 54 5개 형태소분석기의 성능 직접 비교

② 텍스트 전처리 (지정패턴 제거)

: 이메일(기자 이메일), URL, 한글오타(자음만/모음만), 괄호 및 괄호 안에 있는 문자('(속보)', '(종합)'), 특수문자, 공백(양 끝 공백은 모두제거, 문자 사이 공백은 1개만 오도록)에 대한 텍스트 전처리 작업을 진행하였다.

예시: (복면가왕 모아니면도 조관우,깨진 소주병에 목 찔린 사연.."10여 바늘 꿰매")
→ (복면가왕 모아니면도 조관우깨진 소주병에 목 찔린 사연10여 바늘 꿰매)

③ 사용자사전(단어뭉치)

: 형태소분석기는 기본적으로 최소단위로 문장을 분리하기 때문에, 분리를 원치않는 단어들까지 분리되는 문제가 있다(특히 복합명사)[74]. 이러한 문제를 보정하기 위해 은전한닢(mecab)의 경우 사용자가 직접 용어를 등록할 수 있도록 '사용자사전' 기능을 제공하고 있다. 사전의 구조는 '[표층형/0/0/0/품사태그/의미분류/종성유무/읽기/타입/첫번째품사/마지막품사/표현]'의 형태로 구성되어 있으며, 본 프로젝트는 '명사'를 추출 한 뒤 시각화하는 작업을 진행하고 있으므로, '명사'에 해당하는 품사를 추가하는 코드를 작성하여 사용자사전을 추가하였다. 이 과정에서 파이썬이 아닌 'PowerShell'을 이용한 작업이 요구됐지만, 파이썬 내에서 'PowerShell'을 작동[75]하도록 설정하여 사용자의 작업을 최소화했다.

```
#사용자사전에 작성한 내용 적용

#reset 했으므로 다시 import
import subprocess

p = subprocess.Popen(
  [
    "powershell.exe",
    "-noprofile", "-c",
    r"""
    Start-Process -Verb RunAs -Wait powershell.exe -Args "
     -noprofile -c Set-ExecutionPolicy Unrestricted; cd c:¥mecab; .¥tools¥add-userdic-win.ps1;
      "
    """
  ]
)
p.communicate()
```

그림 55 파이썬 내 PowerShell 명령어 실행(사용자 사전 업데이트)

```
# 우선순위 조정

#reset 했으므로 다시 import
import pandas as pd
df_ = pd.read_csv('C:/mecab/mecab-ko-dic/user-nnp_new.csv', header=None)
df_[3] = list(map(lambda x: 0, df_[3]))
df_.to_csv('C:/mecab/mecab-ko-dic/user-nnp_new.csv', header=None, index=False)

p = subprocess.Popen(
  [
    "powershell.exe",
    "-noprofile", "-c",
    r"""
    Start-Process -Verb RunAs -Wait powershell.exe -Args "
     -noprofile -c Set-ExecutionPolicy Unrestricted; cd c:¥mecab; .¥tools¥compile-win.ps1;
      "
    """
  ]#,shell=True #관리자권한
)
p.communicate()
```

그림 56 파이썬 내 PowerShell 명령어 실행(우선순위 업데이트)

④ 불용어

: 추출한 명사에 대한 불용어처리 작업 진행했다. 불용어사전의 경우 'RANKS NL'의 한국어 불용어사전(791개)을 이용했으며, 추가적으로 1글자 단어도 추출 대상에서 제외하였다.

---

74) '띄어쓰기'같은 명확한 기준이 아닌, 최소단위의 명사로 나누기 때문에 발생하는 필연적 문제. '파인애플피자'를 최소단위 명사인 파인애플/피자 형태로 분리하는 원리

75) '파이썬명령(사전추가)-파이썬종료-powershell실행-파이썬명령(우선순위조정)-파이썬종료-powershell실행'의 과정을 파이썬 내에서 메모리를 초기화한 뒤, 코드를 재실행하도록 작성하여 사용자의 별도 명령없이 실행되도록 구현

⑤ 동의어/유의어 지정

: 가중치 및 순위산정 시 같은 의미를 가진 단어들의 점수가 분산되지 않도록 동의어/유의어 사전을 지정해줄 필요가 있다. 동의어/유의어 사전을 생성한 후, 하나의 묶음에 속하는 단어들을 모두 동일한 단어로 통일시키는 작업을 통해 이후의 분석단계에서 가중치가 분산되지 않도록 설정하였다.

```
# 동의어/유의어
'SYNONYM_DICT': { # 오른쪽에 있는 리스트 값을 모두 왼쪽값으로
                  '식약처' : ['식품의약품안전처'],
                  '코로나' : ['코로나19', 'covid'],
                  '한국' : ['대한민국'],
                  '계란' : ['달걀']
                  },
```

그림 57 동의어/유의어 사전 예시. (식약처/식품의약품안전처, 코로나/코로나19/covid, 한국/대한민국, 계란/달걀)

4.4. 이상검색량에 대한 텍스트 분석 시각화

(특정 검색어의 검색량이 특정 일자에 비일반적으로 높게 측정된 경우, 해당 기간에 어떤 사건이 있었는지를 유추하기 위한 작업)

① 워드클라우드(빈도기반)

: 워드클라우드는 비정형 텍스트 데이터를 분석하는 대표적인 기법 중 하나로, 데이터 내 주요 키워드를 나타낼 수 있는 시각화 기법이다. 본 프로젝트에서는 이상탐지모델을 통해 검출한 특정기간에 비일반적인 검색량을 보이는 키워드를 이용해 크롤링한 뉴스데이터에 대해서, 해당 기간 동안 많이 출현(빈도)한 키워드를 나타내기 위해 워드클라우드를 이용한 시각화를 진행해봤다.

그림 58 20170815~20170816 기간 '계란' 관련 뉴스 시각화 결과. '계란'의 이상검색량의 이유가 '살충제'와 관련되어 있다는 것을 한 눈에 알아볼 수 있음

그림 59 20200609 '식약처' 관련 뉴스 시각화 결과. 이상검색량의 이유가, '크릴오일'과 관련되어 있음을 한 눈에 알아볼 수 있음

② 의미연결망 SNA(semantic network analysis)

: 의미연결망은 단어 간에 공유된 의미를 바탕으로 데이터의 구조 체계화를 분석하는 소셜 네트워크 분석 기법으로, 본 프로젝트에서는 이상탐지모델을 통해 검출한 특정기간에 비일반적인 검색량을 보이는 키워드를 이용해 크롤링한 뉴스데이터에 대해서, 제목 내 동시출현 빈도를 기반으로 의미연결망을 생성하였다. 연결중심성[76], 매개중심성[77], 근접중심성[78], 위세중심성(고유벡터중심성)[79] 4가지 방법을 통해 중심성을 계산해 시각화해보았으며, 단순한 빈도 수 기반 분석(워드클라우드로 대체 가능)과 차별점을 가지면서 주요 키워드만 한눈에 알아보기 좋은 위세중심성 방법을 사용하여 시각화를 진행해보았다. 결과적으로 워드클라우드와 다르게 주제가 여러개일 경우에도 주제별 주요 키워드를 한 눈에 파악할 수 있었다(그림61).

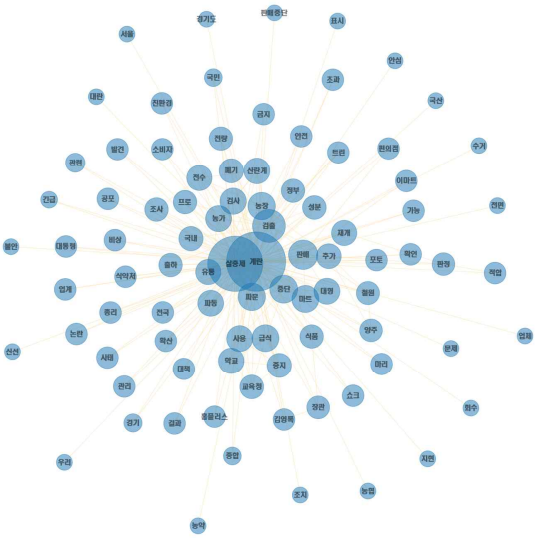


그림 60 20170815~20170816 기간 '계란' 관련 뉴스 시각화 결과. 이상검색량의 이유가 '살충제'와 관련되어 있다는 것을 한 눈에 알아볼 수 있음

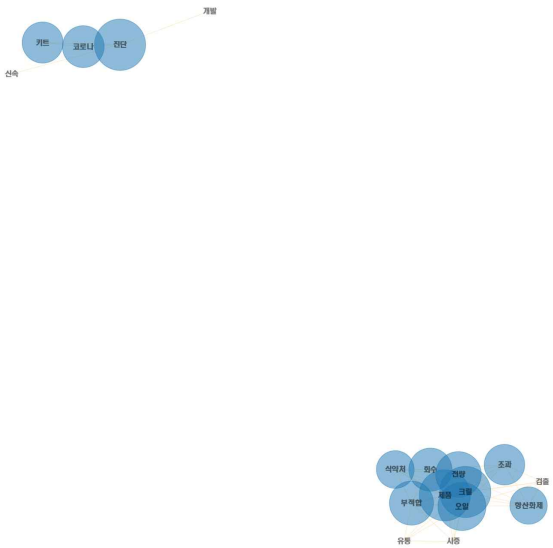


그림 61 20200609 '식약처'의 관련 뉴스 시각화 결과. 이상검색량의 이유가 '코로나진단키드', '크릴오일 회수조치' 2가지 이슈와 관련되어 있음을 유추가능

76) 네트워크의 노드들이 얼마나 많은 연결을 가지고 있는지를 나타내는 지표
77) 한 노드가 네트워크 내의 다른 노드들 사이에 위치하고 있는지를 나타내는 지표
78) 네트워크의 다른 모든 노드들과 얼마나 근접하게 연결되어 있는지를 나타내는 지표
79) 개별 노드의 중심성과 연결된 이웃노드들의 중심성 지표를 함께 고려하여 한 노드의 영향력을 측정하는 지표

③ TF[80]-IDF[81](Term Frequency - Inverse Document Frequency)

: TF-IDF[82]는 정보 검색과 텍스트 마이닝에서 이용하는 가중치로, 여러 문서(뉴스하나)로 이루어진 문서군(기간 내 전체뉴스)이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치이다. 본 프로젝트에서는 이상탐지모델을 통해 검출한 특정기간에 비일반적인 검색량을 보이는 키워드를 이용해 크롤링한 뉴스데이터에 대해서, TF-IDF 방법을 통해 키워드의 문서 내 점수(문서 내 중요도)를 먼저 계산한 뒤, 키워드별로 문서 내 점수를 합산 해 총 합산 점수 상위 키워드를 추출해보았다. 전반적으로 단순 빈도값 순으로 정렬한 값과 크게 다르지 않게 나왔으나, 상대적인 중요도 수치까지 한 눈에 파악하기에는 더 용이했으며, 뉴스량이 많은 주제(여러개의 주제에 대한 결과를 한 눈에 확인하기는 어렵지만, 기간 내 가장 핵심 주제에 대한 주요 키워드들을 확인하기에 적합)에 대해서 더 중점적으로 표시해주는 것을 알 수 있었다('그림63'를 보면 '크릴오일'에 관련된 내용이 대부분이며, 워드클라우드와 sna에 출현했던 '코로나진단키드'관련 내용이 적음).

계란	610.5086
살충제	450.6378
검출	183.5646
판매	181.7337
파문	141.1147
중단	140.4598
검사	134.8073
농장	132.382
파동	122.1388
농가	116.9689

그림 62 20170815~20170816 기간 '계란' 관련 뉴스 시각화 결과. 이상검색량의 이유가 '살충제'와 관련되어 있다는 것을 한 눈에 알아볼 수 있음

크릴	37.59781
제품	35.33143
부적합	30.00216
오일	28.49803
회수	24.52142
전량	23.98503
식약처	23.85151
초과	16.47642
항산화제	14.10979
검출	14.05425

그림 63 20200609 '식약처' 관련 뉴스 시각화 결과. 이상검색량의 이유가, '크릴오일'과 관련되어 있음을 한 눈에 알아볼 수 있음

80) 문서에서 특정 단어(term)가 얼마나 자주 출현하는지를 측정하는 지표(문서 d에서 단어 t의 출현 빈도 / 문서 d에서 총 단어의 수)

81) 말뭉치(corpus)에서 특정 단어의 중요도를 측정하는 지표. log(말뭉치에서 총 문서의 개수 / 단어 t를 포함하는 문서의 개수)

82) TF점수와 IDF점수를 곱한 수치 (TF(t, d) x IDF(t, D))

Ⅲ. 분석 결과

1. 결과 제공 형태

① 과거 내역 보기(최초 실행)

: 1000개가 넘는 검색어에 대해 위 분석과정(이상탐지 모델)을 일괄실행하는 경우, 데이터 수집과정에서 드는 시간적 비용이 많이 발생한다(특히 뉴스크롤러). 이를 해결하기 위해 '이어하기'옵션을 생성하여 수집완료한 과거 이력 내용을 'resume_list.txt' 파일에 한 줄씩 추가하도록했고, 최초 실행시에 소요되는 시간에서 이미 수집완료한 데이터에 대한 실행시간을 단축시켰다. 또한 과거 데이터를 축적시켜놓음으로써, 매일 이상탐지 프로그램이 실행되기 전이라도, 과거이력에 대한 이상검색량 조회를 가능하도록 했다.

② 스케줄러(매일 탐지)

: 이상탐지 모델의 주 목적은 모니터링 대상 검색어에 대해서 이슈가 발생할 경우, 해당 이슈내용을 빠르게 파악하는 것에 있다. 결국 이상탐지 모델은 작동시점 기준 '어제'의 이슈가 발생했을 경우 빠르게 이를 검출하여 출근 시점에 사용자들(식품안전정보원 관계자)이 이슈내용을 빠르게 파악할 수 있어야한다. 이를 위해 전체 코드를 '.bat'확장자를 통해 일괄실행되도록 작성하였고, '.bat' 파일을 윈도우스케줄러와 연결해 매일 아침에 프로그램이 자동으로 작동하여 출근시점에 사용자들이 어제의 이슈가 발생했는지를 확인할 수 있도록 했다.

③ 카카오API '나에게 보내기' 알림 (어제 이상검색량 발생시 알림)

: 카카오의 REST API 중 '나에게 기본 템플릿으로 메시지 보내기[83]'는 무료로 '텍스트' 내용을 실시간으로 사용자의 카카오톡 계정으로 알림을 보내는 기능을 제공한다. 이를 '②항목'의 실행과 연동하여, 만약 '어제'의 이상검색량이 감지된 경우, 이상검색량을 가지는 키워드의 내용을 '나와의 채팅'으로 전송하도록 하였다.

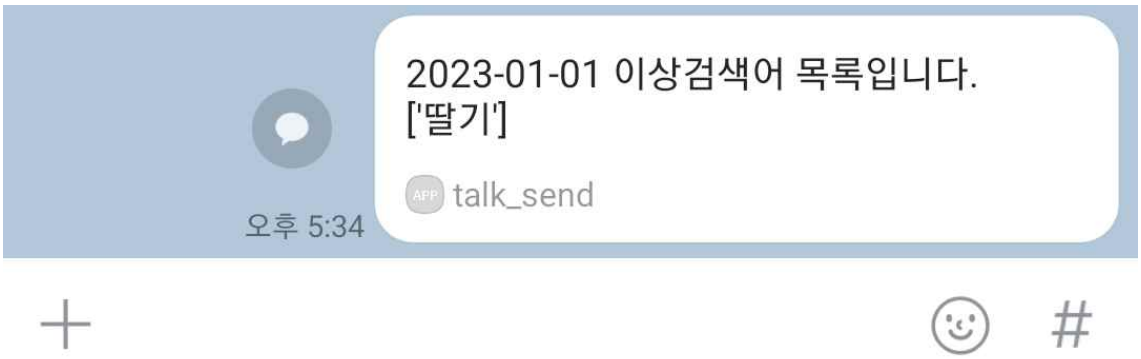


그림 64 지정기간에 이상검색량이 발견된 경우 카카오톡 '나에게 보내기' 알림 제공(실제 전송결과). 실제 적용될 경우 매일 오전에 결과가 전송

83) https://developers.kakao.com/tool/rest-api/open/post/v2-api-talk-memo-default-send

④ 이상검색량이 검출된 검색어의 이상검출기간 주요 키워드 분석 내용 시각화

- SNA(의미연결망) 시각화

: 이슈가 발생했을 경우 이상검색량의 원인을 한눈에 파악이 가능하면서도, 주제가 여러개일 경우에도 주제별로 각각 어떤 주요 내용이 있었는지를 파악하기 적합한 방법으로 'SNA(의미연결망)'이 가장 적합하다고 판단했다. 그림65,66.67은 '라면'에 대한 20200221~20200228 기간(이상탐지모델을 통해 검출된 기간)의 주요 키워드를 시각화 결과이다. 그림65(SNA)의 경우 '코로나 생필품', '오뚜기 컵누들'과 같이 여러개의 주제에 대해서 한 눈에 파악이 가능하지만, 그림66(워드클라우드)와 그림67(tf-idf)의 경우 '오뚜기 컵누들'에 관한 내용을 구분해서 파악하기 어렵다(심지어 그림67의 경우 핵심주제가 아니기 때문에 아예 나타나지않음).

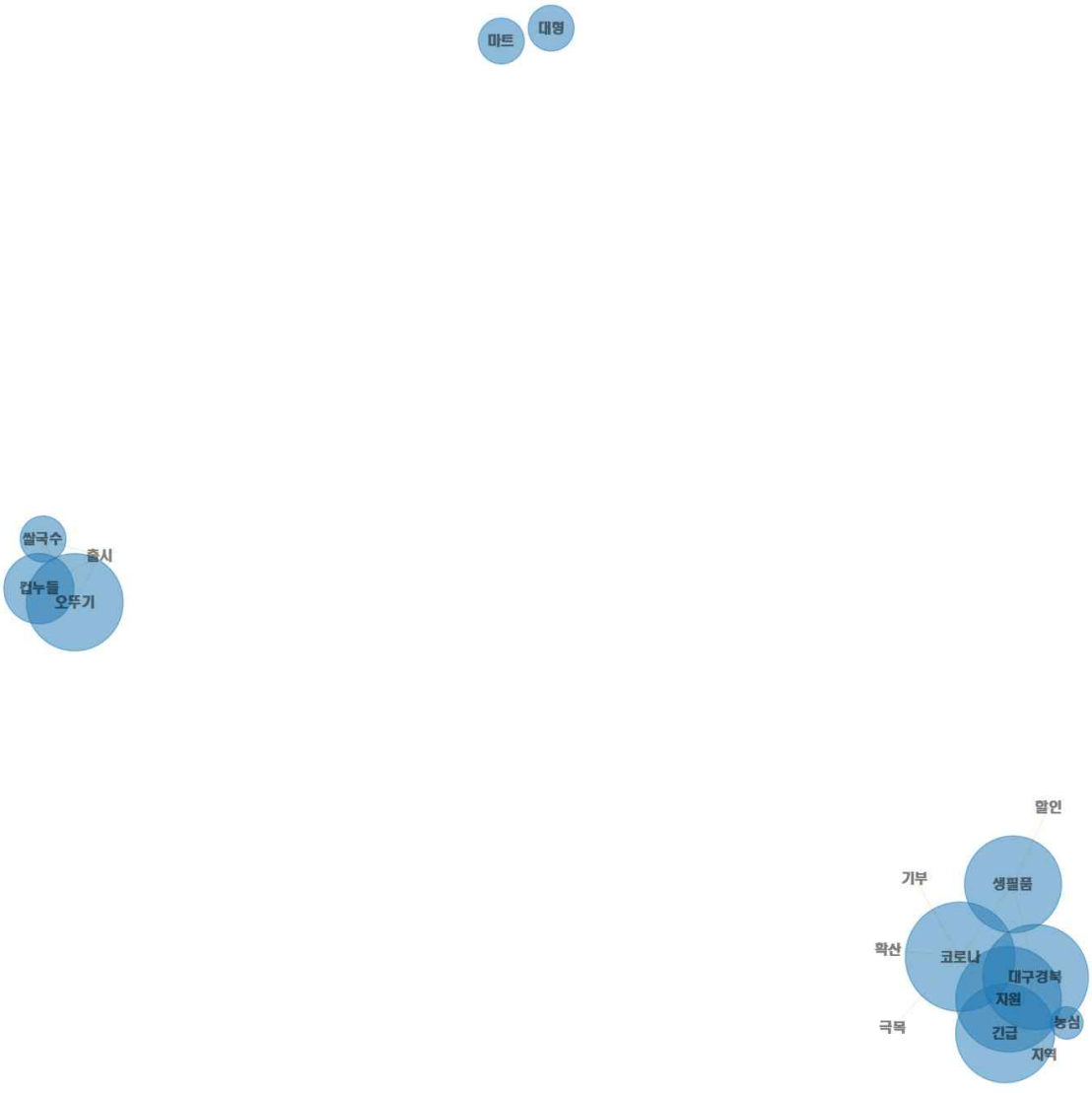


그림 65 의미연결망을 통해 시각화한 '라면'에 대한 20200221~20200228 기간 주요 키워드

- **워드클라우드/TF-IDF (의미연결망의 한계)**

: 다만 의미연결망 분석의 경우, 데이터가 일정 규모 이상인 경우라면 '④항목'에서 설명한 장점으로 인해 다른 시각화기법보다 효과적이지만, 데이터의 규모가 너무 작은 경우 단어 간 연관성을 생성하지 못한다는 문제가 있다. 검색어 대상 이상탐지모델의 특성상 1일단위의 뉴스를 이용한 결과도출이 많기 때문에 일부 결과(뉴스가 없거나 적은 경우)에 대해서 빈 결과[84]를 반환하기도 한다[85]. 이처럼 의미연결망의 결과가 빈 이미지를 반환하는 경우를 대비하기 위해, 시각화 결과 제공시 '워드클라우드'와 'TF-IDF' 결과를 함께 제공하는 형태로 코드를 작성하였다.

그림 66 워드클라우드를 통해 시각화한 '라면'에 대한 20200221~20200228 기간 주요 키워드

코로나	148.6929
생필품	85.26719
지원	81.55823
대구경북	69.8998
농심	61.50717
긴급	54.54544
대구	51.37591
지역	49.03974
라면	43.02436
할인	42.95838

그림 67 tf-idf를 통해 시각화한 '라면'에 대한 20200221~20200228 기간 주요 키워드

84) 단어와 단어 사이에 연결성이 생길만큼의 뉴스데이터가 존재하지 않음

85) 특히 'SNS'와 '커뮤니티' 등에서 이슈가 되었더라도 언론사에서 다루지 않는 경우에도 뉴스 데이터가 많지 않을 수 있다.

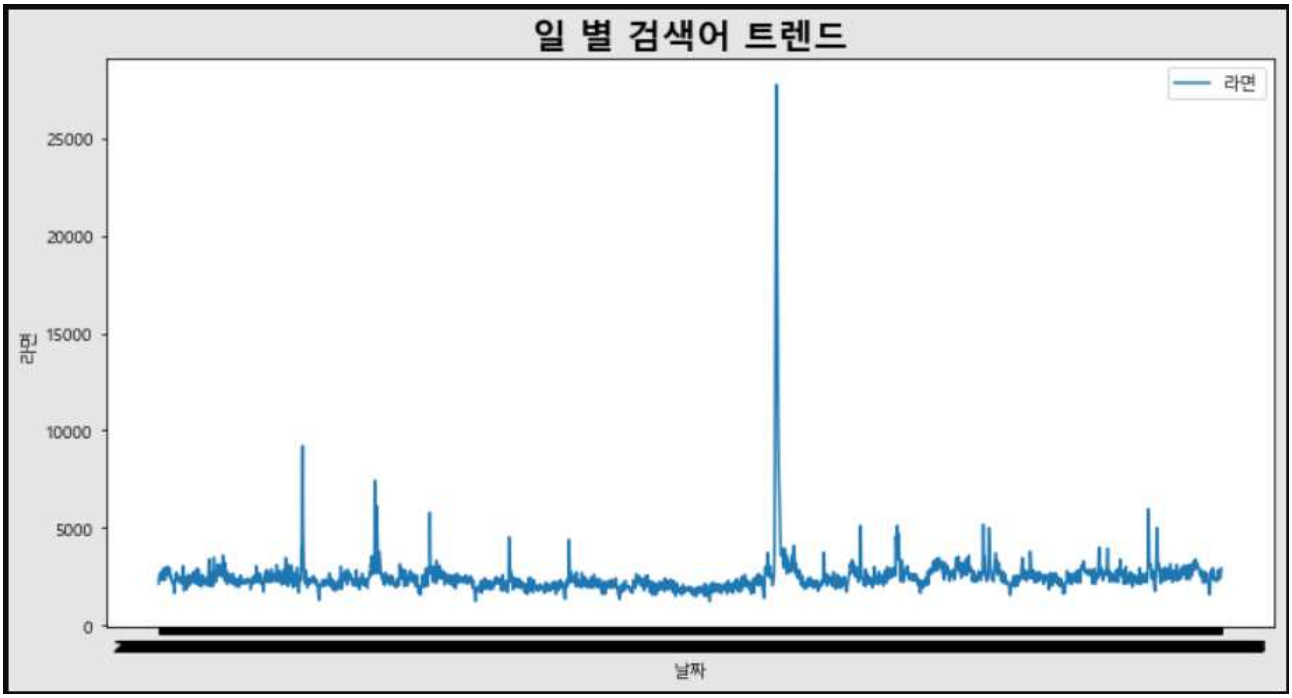
2. 수행 결과

① 네이버 데이터랩 '통합검색어 트렌드 API'를 이용한 '상대적 검색량' 추출

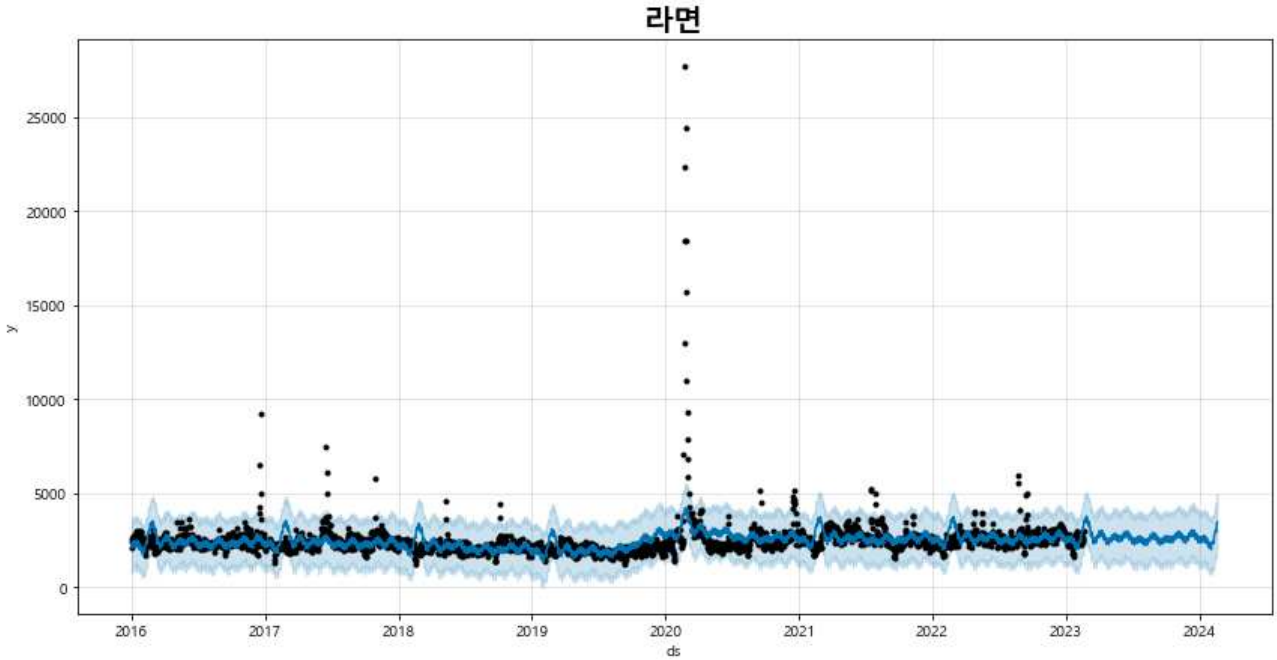
날짜	계란	포도	배	사과	망고	옥수수	벼	쌀	보리	식약처
2016-01-01	0.01877	0.00856	0.02979	0.04997	0.04039	0.01441	0.00315	0.01477	0.02133	0.00607
2016-01-02	0.01585	0.00841	0.02853	0.05128	0.04569	0.01512	0.00299	0.0141	0.03118	0.00739
2016-01-03	0.01785	0.00929	0.03049	0.05717	0.04773	0.01758	0.00319	0.01854	0.03126	0.00815
2016-01-04	0.02178	0.02019	0.03689	0.07724	0.04846	0.01934	0.00531	0.02168	0.03149	0.06357
2016-01-05	0.02111	0.01114	0.03968	0.07408	0.04539	0.02101	0.00474	0.02103	0.02951	0.06326
2016-01-06	0.02268	0.01131	0.03791	0.07304	0.04716	0.02207	0.00448	0.02068	0.02814	0.06265
2016-01-07	0.02217	0.01169	0.03597	0.07508	0.04526	0.02131	0.00383	0.02331	0.02612	0.06314
2016-01-08	0.02034	0.01	0.03456	0.0642	0.04963	0.01879	0.00413	0.02213	0.02386	0.0559
2016-01-09	0.01997	0.00851	0.02955	0.05386	0.04153	0.0172	0.00313	0.01722	0.0227	0.00919
2016-01-10	0.01873	0.00943	0.03014	0.05892	0.04657	0.02076	0.00309	0.02083	0.02272	0.00951
2016-01-11	0.02184	0.01936	0.03899	0.07306	0.05172	0.02199	0.00442	0.02708	0.02443	0.06293

② 'NAVER Search Ad API(네이버광고API)'를 이용한 검색량 상대값→절대값 변환

날짜	계란	포도	배	사과	망고	옥수수	벼	쌀	보리	식약처
2016-01-01	941.3626	429.3055	1494.043	2506.121	2025.66	722.6977	157.9804	740.7526	1069.753	304.4257
2016-01-02	794.9173	421.7826	1430.851	2571.821	2291.468	758.3059	149.956	707.1504	1563.755	370.627
2016-01-03	895.2223	465.9168	1529.15	2867.219	2393.779	881.6811	159.9865	929.8275	1567.767	408.743
2016-01-04	1092.322	1012.579	1850.126	3873.78	2430.391	969.9495	266.3098	1087.306	1579.303	3188.195
2016-01-05	1058.719	558.699	1990.052	3715.298	2276.422	1053.704	237.7229	1054.707	1480.001	3172.648
2016-01-06	1137.459	567.2249	1901.282	3663.139	2365.192	1106.866	224.6832	1037.154	1411.292	3142.055
2016-01-07	1111.881	586.2828	1803.986	3765.45	2269.903	1068.75	192.0841	1169.055	1309.984	3166.629
2016-01-08	1020.102	501.5251	1733.271	3219.791	2489.069	942.3657	207.1299	1109.875	1196.639	2803.525
2016-01-09	1001.546	426.7979	1482.007	2701.214	2082.834	862.6232	156.9774	863.6262	1138.462	460.9016
2016-01-10	939.3565	472.9382	1511.597	2954.986	2335.602	1041.166	154.9713	1044.677	1139.465	476.9504
2016-01-11	1095.331	970.9526	1955.446	3664.142	2593.888	1102.854	221.6741	1358.13	1225.226	3156.097



③ 시계열 예측 모델(prophet)을 이용한 검색어의 '일반적인 검색량' 예측



④ 일반적인 검색량 대비 실제 검색량이 높은 이상 항목 탐지(Anomaly Detection)

⑤ 이상 검출 항목 라벨링 및 전처리('[검색어, 시작일자, 종료일자]' 형태로 변환)

날짜	계란	포도	배
2016-01-01	[941.362601978884, 'No']	[429.3054807106685, 'No']	[1494.0432558844411, 'No']
2016-01-02	[794.9172744467402, 'No']	[421.7826042963461, 'No']	[1430.8510940041324, 'No']
2016-01-03	[895.2222933043731, 'No']	[465.9168125937045, 'No']	[1529.1500124846125, 'No']
2016-01-04	[1092.3216553596217, 'No']	[1012.5791653678036, 'No']	[1850.1260728290376, 'No']
2016-01-05	[1058.7194740423147, 'No']	[558.6989550370149, 'No']	[1990.0515741354357, 'No']
2016-01-06	[1137.4589138455565, 'No']	[567.2248816399137, 'No']	[1901.2816324464304, 'No']
2016-01-07	[1111.88113403686, 'No']	[586.2828352228639, 'No']	[1803.9857641545264, 'No']
2016-01-08	[1020.102041782126, 'No']	[501.5250942881642, 'No']	[1733.2707258598955, 'No']
2016-01-09	[1001.545613293464, 'No']	[426.7978552392277, 'No']	[1482.0066536215252, 'No']
2016-01-10	[939.3565016017316, 'No']	[472.9381639137388, 'No']	[1511.596634184527, 'No']
2016-01-11	[1095.3308059253504, 'No']	[970.9525825418858, 'No']	[1955.446342629552, 'No']

날짜	검색어
2016-01-09	['족발']
2016-01-15	['콜마비앤에이치']
2016-01-16	['갱년기']
2016-01-22	['네네치킨']
2016-01-23	['네네치킨']
2016-01-24	['피자', '네네치킨']
2016-01-25	['신전떡볶이']
2016-01-26	['코코넛오일']
2016-01-30	['코코넛오일', '네네치킨']
2016-01-31	['과메기', '코코넛오일']

검색어	날짜
계란	['2017-08-15', '2017-08-16']
포도	['2022-09-18', '2022-09-19']
배	['2023-01-30']
사과	['2016-02-15', '2018-04-22', '2018-04-23', '2

검색어	시작일자	종료일자
계란	[20170815]	[20170816]
포도	[20220918]	[20220919]
배	[20230130]	[20230130]
사과	[20160215, 20180422, 202	[20160215, 20180423, 20

⑥ 검출된 검색어와 기간에 해당하는 뉴스 크롤링

date	title	link	press
2020-02-21	[인터뷰] 눈	http://www	아주경제
2020-02-21	[사자성어는	http://www	중도일보
2020-02-21	'기생충 오	http://www	농민신문
2020-02-21	강원양돈농	http://www	농민신문
2020-02-21	[월간중앙]	https://nev	중앙일보
2020-02-21	전국 우수기	http://www	강원도민일보
2020-02-21	[일과 신앙	http://new	국민일보
2020-02-21	무더기 확기	http://www	경북일보
2020-02-21	노인병원·정	http://new	매일경제언론사 선정
2020-02-21	[포토뉴스]	http://www	강원일보
2020-02-21	'고지혈증	http://www	인사이트코리아

⑦ 자연어처리를 위한 사전 구축 및 텍스트 전처리

```
#사용자사전에 작성한 내용 적용

#reset 했으므로 다시 import
import subprocess

p = subprocess.Popen(
  [
    "powershell.exe",
    "-noprofile", "-c",
    r"""
    Start-Process -Verb RunAs -Wait powershell.exe -Args "
     -noprofile -c Set-ExecutionPolicy Unrestricted; cd c:¥mecab; .¥tools¥add-userdic-win.ps1;
     "
    """
  ]
)
p.communicate()
```

```
# 우선순위 조정

#reset 했으므로 다시 import
import pandas as pd
df_ = pd.read_csv('C:/mecab/mecab-ko-dic/user-nnp_new.csv', header=None)
df_[3] = list(map(lambda x: 0, df_[3]))
df_.to_csv('C:/mecab/mecab-ko-dic/user-nnp_new.csv', header=None, index=False)

p = subprocess.Popen(
  [
    "powershell.exe",
    "-noprofile", "-c",
    r"""
    Start-Process -Verb RunAs -Wait powershell.exe -Args "
     -noprofile -c Set-ExecutionPolicy Unrestricted; cd c:¥mecab; .¥tools¥compile-win.ps1;
     "
    """
  ],shell=True #관리자권한
)
p.communicate()
```

```
# 텍스트 전처리
df['title_c'] = df.apply(clean_text, axis=1)
# 형태소 분석기 가동
df = morphological_analyzer(df)
# 불용어 적용
stopword = _cfg['STOP_WORD'] #stopword = [] #(직접 입력도 가능)
df = apply_stopword(df,stopword)
# 동의어/유의어 사전
synonym_dict = _cfg['SYNONYM_DICT']
df = apply_synonym_dict(df,synonym_dict) #synonym_dict = [] #(직접 입력도 가능)
df.to_csv('data/result/news_noun.csv', encoding='utf-8-sig', index=False)
file_name_ = f'{recent_file[27:-19]}{recent_file[-19:-12]}{recent_file[-12:-4]}'
```

⑧ 형태소 분석기를 이용한 뉴스 제목 내 명사 추출

title_c	keyword_hannanum	keyword_kkma	keyword_komoran	keyword_mecab	token_mecab	keyword_okt
인터뷰 눈빛요정 공유림 드라마 터치에서 빛나는 존재감 각인	[인터뷰, 눈빛요정, 공유, 드라마, 터치, 존재감, 각인]	[인터뷰, 눈빛, 눈빛요정, 요정, 공유림, 드라마, 터치, 존재, 존재감, 각인]	[인터뷰, 눈빛, 요정, 유림, 드라마, 터치, 존재, 각인]	[인터뷰, 눈빛, 요정, 공유림, 드라마, 터치, 존재감, 각인]	[인터뷰, 눈빛, 요정, 공유림, 드라마, 터치, 에서, 빛나, 존재감, 각인]	[인터뷰, 눈빛, 요정, 유림, 드라마, 터치, 존재, 각인]

date	title	link	press	title_c	keyword_mecab
2020-02-21	[인터뷰] 눈	http://www	아주경제	인터뷰 눈빛	['인터뷰', '눈빛', '요정', '공유림', '드라마', '터치', '존재감', '각인']
2020-02-21	[사자성어는	http://www	중도일보	사자성어는	['사자성어', '이음매', '산수경석']
2020-02-21	'기생충 오	http://www	농민신문	기생충 오스	['기생충', '오스카상', '덕분', '식품', '수출', '훈풍', '기대']
2020-02-21	강원양돈농	http://www	농민신문	강원양돈농	['강원', '농협', '조합원', '구서', '지원']
2020-02-21	[월간중앙]	https://nev	중앙일보	월간중앙 ㅋ	['월간중앙', '국민', '시작', '안철수', '직설']
2020-02-21	전국 우수ㅈ	http://www	강원도민일	전국 우수ㅈ	['전국', '우수', '점포', '비결', '상권', '분석', '철저', '매장', '관리']
2020-02-21	[일과 신앙	http://new	국민일보	일과 신앙	['신앙', '불시착', '돌보', '대기', '행복', '전도자']
2020-02-21	무더기 확ㅈ	http://www	경북일보	무더기 확ㅈ	['무더기', '확진', '소식', '위험', '식당가', '도심', '한산']
2020-02-21	노인병원·ㅈ	http://new	매일경제인	노인병원정	['노인', '병원', '정신', '병동', '장례식장', '직원', '환자', '격리']
2020-02-21	[포토뉴스]	http://www	강원일보	포토뉴스춘	['포토', '뉴스', '춘천', '소양제', '교회', '상자', '기탁']

⑨ 검출된 검색어와 기간에 대한 뉴스 내 주요 키워드 분석 (SNA,워드클라우드,TF-IDF)

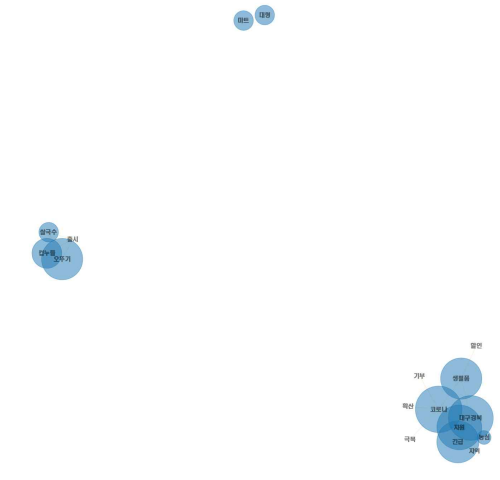


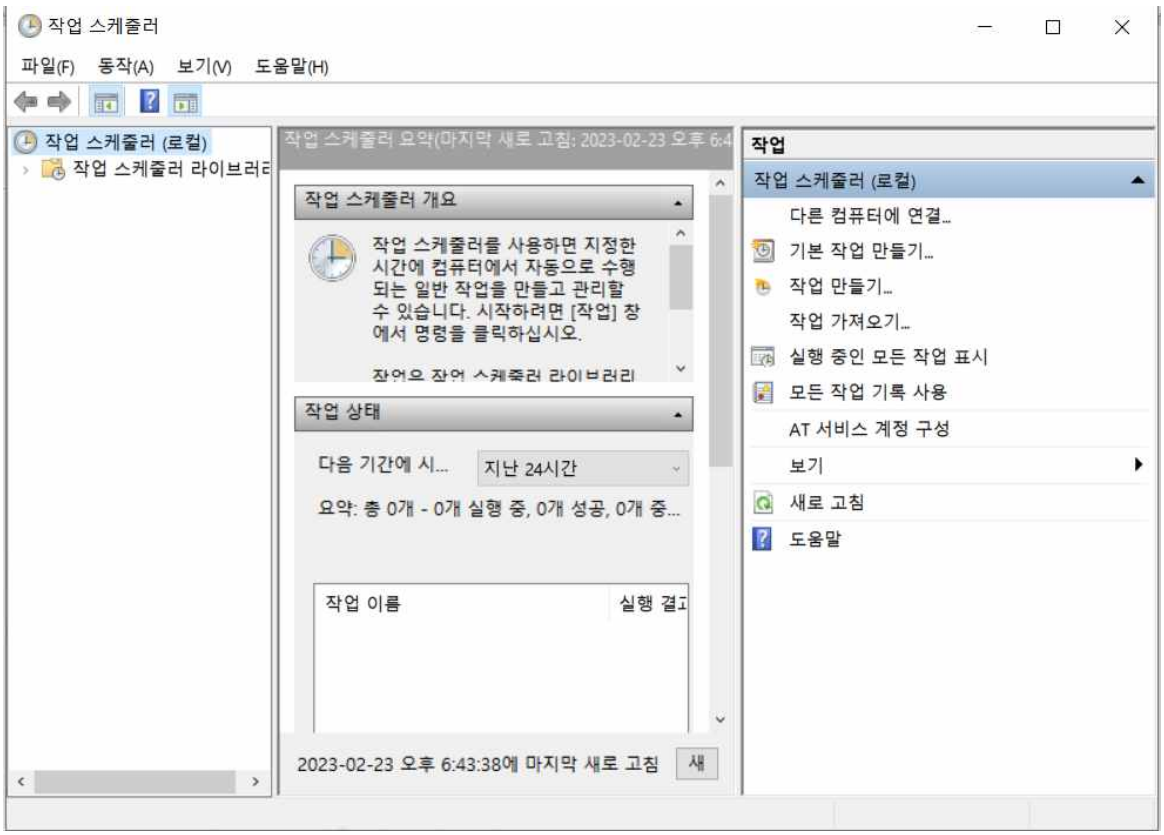
그림 83 SNA

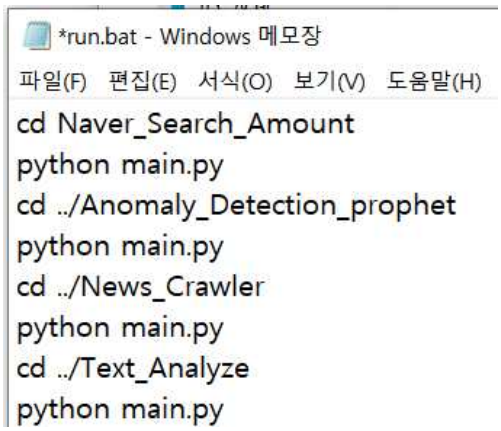
그림 84 워드클라우드

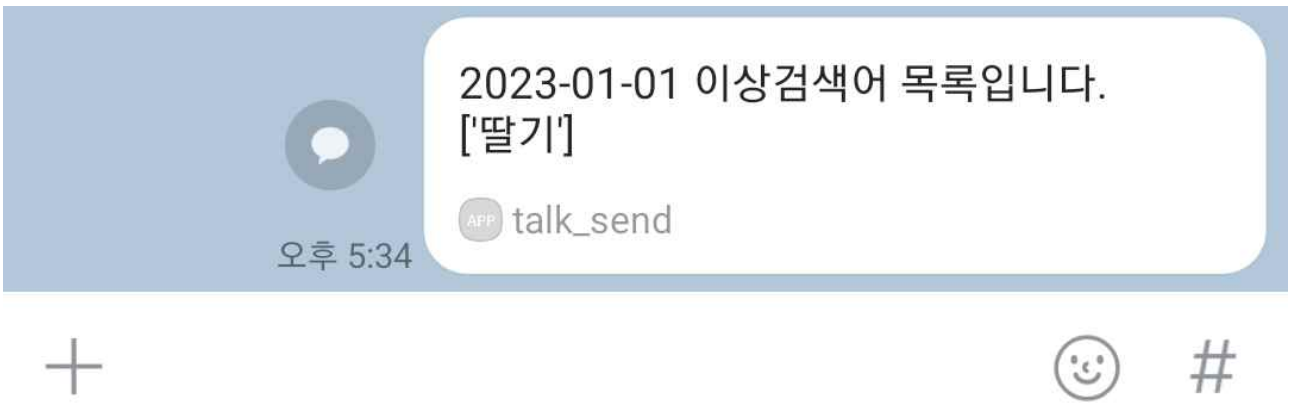
코로나	148.6929
생필품	85.26719
지원	81.55823
대구경북	69.8998
농심	61.50717
긴급	54.54544
대구	51.37591
지역	49.03974
라면	43.02436
할인	42.95838

그림 85 TF-IDF

⑩ 스케줄러를 이용한 이상검색량 자동점검 및 업데이트(어제 이상검색량이 발생할 경우 카카오톡 나에게 보내기를 통해 알림 제공)







Ⅳ. 결론

1. 기대효과 및 활용방안

○ 기대효과

- 본 프로젝트의 결과물은 식품 및 식품안전 관련한 이슈(용어) 모니터링을 위한 수단으로, 대국민 식품안전정보 포털인 식품안전나라에서 사용 가능. 따라서, 해당 시스템에 적용하여 관계자들이 빠르게 정보를 입수하기 위한 수단활용 여부 및 실제 시스템 반영에 대해 긍정적으로 검토 중임
- 식품 및 식품안전 이슈에 관한 관계자들의 모니터링 능률이 향상을 통해 국민 관심사를 더 빠르게 파악할 수 있게 하며, 국민 관심사와 일치하는 내용에 대한 더 빠르고 정확한 정보전달을 가능하게 함
- 국민들에게 빠르고 정확한 식품안전 정보를 전달함으로써, 결과적으로 '국민건강 및 식품안전'이라는 궁극의 목적 달성에 기여

○ 활용방안

- 식품 및 식품안전 용어 모니터링을 위한 수단으로써 업무시스템에 실제적으로 적용되어 활용하는 것에 대한 검토 단계
- 모니터링 대상 검색어의 확대를 통해 더 많은 주제에 대한 국민들의 관심사 파악 가능
- 식품 및 식품안전과 관련된 핵심 이슈에 가려진 다른 이슈들에 대해서도 정보습득이 가능
- '네이버 검색량'뿐만 아니라 '방문자 수'와 같은 다른 시계열 수치에 대한 이상탐지 모델 적용

2. 문제점 및 개선방안

○ 문제점

- '이상 검색량(비주기적 이상검색량)'와 '관심사(주기적으로 발생하는 높은 검색량)'의 구분 문제
- '뉴스 데이터'를 기반으로 텍스트분석을 진행하기 때문에, 'SNS'와 '커뮤니티' 등에서 이슈가 되었지만 뉴스데이터가 존재하지 않는 이슈에 대해서는 파악이 어려움[86)]

○ 개선방안

- 모니터링 대상 검색어의 확대를 통해 더 많은 주제에 대한 국민들의 관심사 파악
- 커뮤니티/SNS 크롤러 추가 구축을 통한 국민 특성(연령대/성별 등)에 따른 세부적인 관심사 파악
- 사용자사전의 확장 구축을 통한 더 세밀한 텍스트 분석 결과 생성
- 불용어사전의 확장 구축을 통한 더 정확한 텍스트 분석 결과 생성

86) 이슈가 되었다는 사실을 파악하더라도, 원인을 파악하기 어려움

Ⅴ. 참고자료/부록

❍ https://developers.naver.com/docs/serviceapi/datalab/search/search.md
❍ https://naver.github.io/searchad-apidoc/#/guides
❍ https://github.com/facebook/prophet
❍ https://www.analyticsvidhya.com/blog/2021/12/anomaly-detection-model-using-facebook-prophet/
❍ https://wonhwa.tistory.com/52
❍ https://somjang.tistory.com/entry/Google-Colab%EC%97%90%EC%84%9C-mecab-ko-dic-%EC%82%AC%EC%9A%A9%EC%9E%90-%EC%82%AC%EC%A0%84-%EB6%94%EA%B0%80%ED%95%98%EA%B8%B0
❍ https://stackoverflow.com/questions/73153170/run-powershell-as-administrator-from-python
❍ https://konlpy.org/ko/v0.4.3/morph/
❍ https://www.ranks.nl/stopwords/korean
❍ https://mons1220.tistory.com/241
❍ https://wikidocs.net/31698
❍ https://foreverhappiness.tistory.com/38