

인터넷 쇼핑몰 리뷰를 활용한
NLP 감성분석 데이터 파이프라인

2022.08.10

말하는 감자

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

개정이력

개정 번호	개정 내용 요약	추가/수정 항목	개정 일자	작성자
0.1	최초 제정 승인	목차 / 개요	2022.07.26	도효주
0.2	-	내용	2022.07.30	도효주
0.3	-	내용/부록	2022.08.03	도효주, 전중석
0.4	중간 제정 승인	내용	2022.08.05	도효주, 선우지훈
1.0	최종 제정 승인	결론	2022.08.07	도효주, 오승우

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

목 차

1. 서론	9
1.1. 프로젝트 개요	9
1.1.1. 주제 설정 동기	9
1.1.2. AS-IS, TO-BE	10
1.1.3. 기대 효과	12
1.1.4. 프로젝트 역할 분담	12
1.1.5. 프로젝트 일정	151
1.2. 프로젝트 환경	13
1.2.1. 환경 구성	13
1.2.2. 활용 도구	13
2. 본론	15
2.1. 수요 분석	15
2.2. 프로세스 분석	21
2.2.1. 서비스 플로우	21
2.2.2. 기능 플로우	23
2.2.3. 데이터 플로우	25
2.2.4. 인프라 요구사항 명세	26
2.3. 프로세스 설계	31
2.3.1. 프로세스 아키텍처	31
2.3.2. 시스템 구성	32
2.3.3. Usecase 시나리오	36

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.3.4. 데이터 연동 규격	45
2.4. 프로세스 구현	50
2.4.1. 인프라 구현	50
2.4.2. Usecase 기능 구현	110
2.4.3. 정상 동작 확인	142
3. 결론	145
3.1. 결과물 활용 방법	145
3.2. 문제점 및 개선 방안	145
3.3. 프로젝트 결과 및 향후 개선점	150

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

표 목 차

[표 1-1] 기대 효과	12
[표 1-2] 프로젝트 역할 분담	12
[표 1-3] 프로젝트 일정 : 부록 참고	12
[표 1-4] 환경 구성 및 OS 구성	13
[표 1-5] 활용 도구	14
[표 2-1] 인프라 요구사항 명세 개요	26
[표 2-2] 인프라 요구사항 - 기능 요구사항	27
[표 2-3] 인프라 요구사항 - 데이터 요구사항, 성능 요구사항	28
[표 2-4] 인프라 요구사항 - 시스템 장비 구성 요구사항, 테스트 요구 사항	29
[표 2-5] 인프라 요구사항 - 품질 요구사항, 프로젝트 관리 요구사항	30
[표 2-6] Python Crawler 소프트웨어 버전	32
[표 2-7] Kafka 소프트웨어 버전	33
[표 2-8] Jupyter Notebook 소프트웨어 버전	33
[표 2-9] Jupyter Notebook 프로그램 모듈 버전	34
[표 2-10] 기능 요구사항 - SFR-001	36
[표 2-11] 기능 요구사항 - SFR-002	37
[표 2-12] 기능 요구사항 - SFR-003	39
[표 2-13] 기능 요구사항 - SFR-004	40
[표 2-14] 데이터 요구사항 - DR-001	41
[표 2-15] 데이터 요구사항 - DR-002	42

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[표 2-16] 데이터 요구사항 - DR-003	43
[표 2-17] 데이터 요구사항 - DR-004	44
[표 2-18] Crawler - Kafka 연동 규격	45
[표 2-19] Kafka - OpenSearch 연동 규격 - Input	46
[표 2-20] Kafka - OpenSearch 연동 규격 - Outbound	48
[표 2-21] Crawler 파드 환경변수	57
[표 2-22] OpenSearch Master Node Configmap data Field Parameter	70
[표 2-23] OpenSearch Master Node Deployment Parameter	73
[표 2-24] Logstash Input Field Parameter	86
[표 2-25] 스마트 스토어 모자 Topic	92
[표 2-26] 스마트 스토어 티셔츠 Topic	92
[표 2-27] 파이썬 크롤러 사용 모듈	113
[표 2-28] Kafka 프로듀서 속성	116
[표 2-29] 파이썬 크롤러 사용 함수	118

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

그림 목 차

[그림 1-1] 예시 쇼핑몰	10
[그림 1-2] 별점-댓글 불일치 댓글	10
[그림 1-3] 대시보드	11
[그림 2-1] 이커머스 시장 규모	15
[그림 2-2] 이커머스 디지털 광고비 증감 추이	16
[그림 2-3] 이커머스 소비자들을 대상으로 한 조사	17
[그림 2-4] 온라인 쇼핑몰 관심 정보	18
[그림 2-5] 온라인 쇼핑몰 주 이용 채널	18
[그림 2-6] 온라인 쇼핑몰 선택 요인	19
[그림 2-7] 온라인 제품 정보 획득 경로	20
[그림 2-8] 서비스 플로우	21
[그림 2-9] 기능 플로우	23
[그림 2-10] 데이터 플로우	25
[그림 2-11] 프로세스 아키텍처	31
[그림 2-12] 카프카 클러스터	58
[그림 2-13] ELK 클러스터	64
[그림 2-14] Jupyter Notebook CUDA 설정 1	103
[그림 2-15] Jupyter Notebook CUDA 설정 2	104
[그림 2-16] Jupyter Notebook CUDA 설정 3	104
[그림 2-17] Jupyter Notebook cuDNN 설정 1	106
[그림 2-18] Jupyter Notebook cuDNN 설정 2	106

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[그림 2-19] Jupyter Notebook 접속 확인	109
[그림 2-20] 이중 키워드 댓글 예시	120
[그림 2-21] 이중 키워드 댓글 코드로 확인	120
[그림 2-22] Kibana Destination 설정	122
[그림 2-23] Kibana Monitor 설정 1	123
[그림 2-24] Kibana Monitor 설정 2	123
[그림 2-25] Kibana Configure Actions 설정	123
[그림 2-26] Kibana Monitor schedule 설정	125
[그림 2-27] Kibana Configure Actions 설정	126
[그림 2-28] Kibana Configure Actions Message 설정	126
[그림 2-29] Kibana 알림 확인	126
[그림 2-30] 모델 검증	134
[그림 2-31] 인덱스 패턴 생성	137
[그림 2-32] 인덱스 패턴 정의	138
[그림 2-33] 인덱스 패턴 Configure 세팅	138
[그림 2-34] 인덱스 패턴 생성 확인	139
[그림 2-35] Kibana 로그 확인	140
[그림 2-36] 크롤링 데이터 예시	142
[그림 2-37] 정제된 데이터 예시	142
[그림 2-38] Jupyter Notebook 분석 결과 예시	143
[그림 2-39] Kibana Slack 알림	143
[그림 2-40] 대시보드 구현	144
[그림 3-1] 크롤러 파드 재부팅 참고 자료	147

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

1. 서론

1.1. 프로젝트 개요

1.1.1. 주제 선정 동기

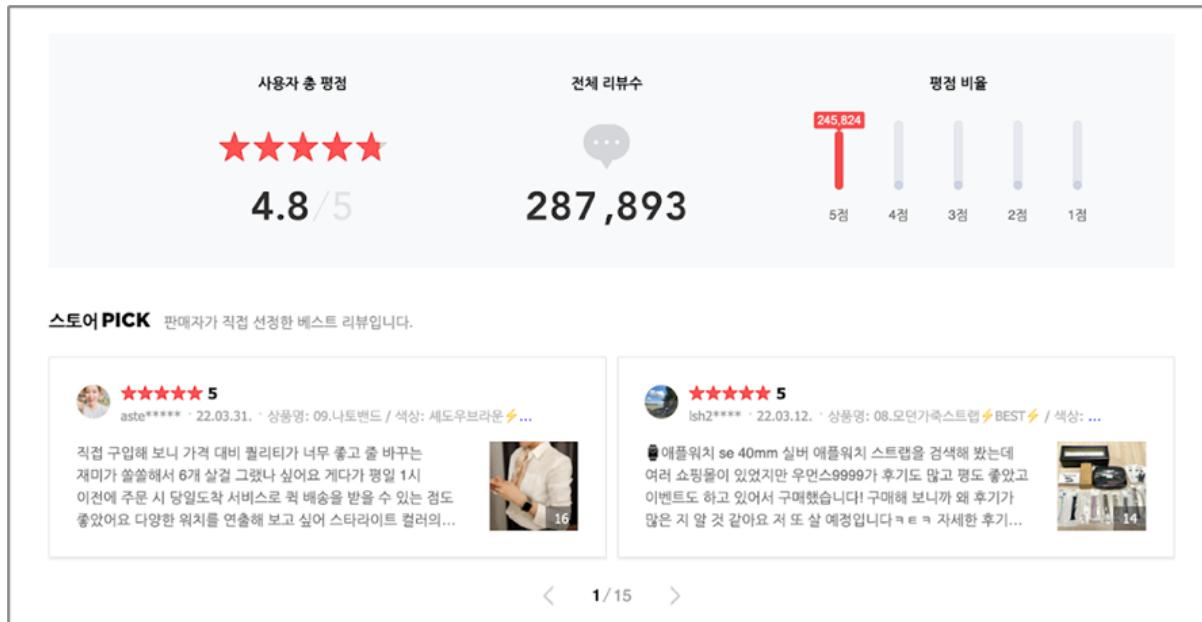
쇼핑몰의 수익성을 높이기 위해서 쇼핑몰 플랫폼 운영자는 소비자 니즈를 파악하여 고객의 제품 구매를 유도해야 한다. 동시에 수요가 낮은 제품은 어떤 부분(배송 서비스, 제품 자체 등)에서 부정적인 반응이 생성되었는지 파악하여 보완을 해야 한다. 하지만 쇼핑몰 플랫폼 운영자는 단시간에 많은 소비자의 반응을 일일이 확인하기 힘들다.

따라서 본 프로젝트에서는 머신 러닝을 통해 학습한 결과를 바탕으로 리뷰 감성 분석을 진행하여 타겟 키워드를 확인할 수 있는 대시보드를 생성함으로써 쇼핑몰 플랫폼 운영자가 직접 리뷰를 읽지 않아도 단시간에 상품에 대한 소비자 반응을 판단할 수 있도록 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

1.1.2. AS-IS, TO-BE

[AS-IS]



[그림 1-1] 예시 쇼핑몰

온라인 쇼핑몰 판매자가 쇼핑몰의 수익을 높이기 위해서는 고객의 리뷰를 읽고 분석할 필요가 있다. 하지만 판매 기간이 길어질수록 리뷰의 개수는 점점 증가할 것이고 그에 따라 쇼핑몰 판매자가 고객의 리뷰를 일일히 읽고 소비자 반응을 분석하는 것은 더욱 힘들어질 것이다.



[그림 1-2] 별점-댓글 불일치 댓글

판매자가 리뷰를 간략하게 파악하기에 용이한 방법으로 별점이 있지만 별점과 실제 리뷰간에 갭이 상당수 존재하며 이러한 갭은 실제로 판매자에게 혼동을 줄수 있는 요인이다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[TO-BE]



[그림 1-3] 대시보드

고객 쇼핑몰에서 판매하는 제품의 리뷰 중 소비자가 긍정적인 반응을 보이고 있는 리뷰의 키워드를 추출하고, 부정적인 반응을 보이고 있는 리뷰의 키워드를 추출하여 소비자가 어떠한 이유로 고객 쇼핑몰의 제품을 구입하는지 파악할 수 있도록 한다. 또한 동일 제품을 판매하고 있는 타사의 쇼핑몰 리뷰에서 자사 키워드를 기반으로 키워드 개수를 추출하여 고객 쇼핑몰과 타사 쇼핑몰이 어떤 점에서 소비자들의 반응 차이가 발생하고 있는지를 확인할 수 있도록 돋는다. 또한 별점과 리뷰 사이의 긍부정 반응 갭이 얼마나 존재하는지 파악하여 쇼핑몰 운영을 위한 지표로 제공하고자 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

1.1.3. 기대 효과

사용자 측면	- 상품에 대한 소비자 반응을 분석해서 키워드로 파악이 가능하게 한다. - 타겟 키워드를 설정하기 위한 리서치시간을 줄인다.
비즈니스 측면	상품에 대한 소비자 반응을 분석해서 키워드로 파악이 가능하게 한다. 타겟 키워드를 설정하기 위한 리서치시간을 줄인다.

[표 1-1] 기대 효과

1.1.4. 프로젝트 역할 분담

이름	직책	역할
도효주	PM	프로젝트 전체 일정 관리, 문서 작업
선우지훈	팀원	텍스트 데이터 정제 및 저장, 인프라 구축(OpenSearch, Logstash, Kibana, Kafka), 대시보드 구현
오승우	팀원	프로젝트 기획, 대시보드 구현, 데이터 분석 프로그래밍
전종석	팀원	데이터 크롤링 및 NLP 처리, 데이터 분석 프로그래밍, 인프라 구축, 대시보드 구현

[표 1-2] 프로젝트 역할 분담

1.1.5. 프로젝트 일정

[표 1-3] 프로젝트 일정 : 부록 참고

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

1.2. 프로젝트 환경

1.2.1. 환경 구성 및 OS 구성

환경 구성

소프트웨어	사용 이유
VSCode	코드 입력을 용이하게 해주는 소스 코드 에디터

OS 구성

소프트웨어	사용 이유
VirtualBox	Window 환경에서 Ubuntu 환경을 구축하기 위해 설치한 가상화 소프트웨어
Ubuntu	리눅스 시스템 자동화에 적합한 OS

[표 1-4] 환경 구성 및 OS 구성

1.2.2. 활용 도구

소프트웨어	사용 이유
Python	사용자 쇼핑몰 리뷰 추출 및 자연어 처리에 사용되는 프로그래밍 언어
AWS CLI	Amazon 서비스 통합 관리
Terraform	클라우드 프로바이더에 IaC 배포 자동화
Docker	크롤러 파드 이미지 생성
Amazon EKS	AWS 상에서 컨테이너화 된 애플리케이션 관리 자동화
Selenium(Chrome Driver)	쇼핑몰 리뷰 추출 시 랜더링 되는 웹 드라이버

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

소프트웨어	사용 이유
CUDA	GPU의 가상 명령어셋을 사용할 수 있도록 만들어주는 소프트웨어 레이어
cuDNN	심층 신경망을 위한 GPU 가속 프리미티브 라이브러리
Kafka	쇼핑몰 추출 리뷰 유실 방지
Anaconda	과학 연구 및 머신러닝 분야에 적합한 Python 패키지 의존성 관리 및 배포를 편리하게 해주는 패키지 관리자
JupyterNotebook	탐색적 데이터 분석, 데이터 정리 및 변환, 데이터 시각화, 통계적 모델링, 머신 러닝, 딥러닝 등의 각종 데이터 사이언스 문서 생성 애플리케이션
Tensorflow	딥러닝 라이브러리 중 하나이며 Python을 활용하여 연산처리 작성
mecab	일본어와 한국어의 유사점으로 한글 분석에도 동작하는 것을 확인하고 개발한 한국어 형태소 분석기
OpenSearch	일본어와 한국어의 유사점으로 한글 분석에도 동작하는 것을 확인하고 개발한 한국어 형태소 분석기
Logstash	- Kafka에 저장된 데이터 수집 (Consumer) - 쇼핑몰 리뷰 데이터 정제 및 인덱싱
Kibana	리뷰 감성 분석 결과 시각화

[표 1-5] 활용 도구

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

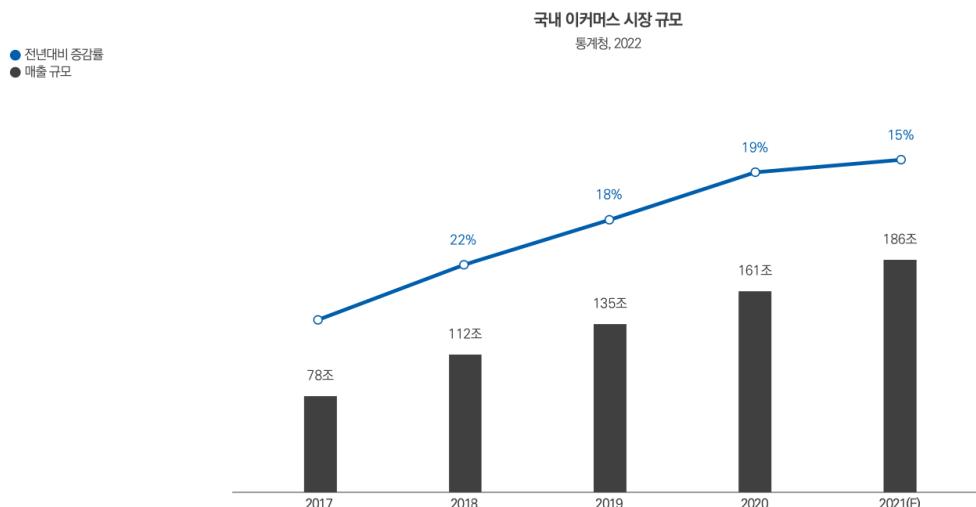
2. 본론

2.1. 수요 분석

해당 서비스는 특정 쇼핑몰에 입점해있는 사업자, 쇼핑몰 제작 솔루션 업체나 개인 쇼핑몰을 운영하는 사업자들을 소비자로 선정한다.

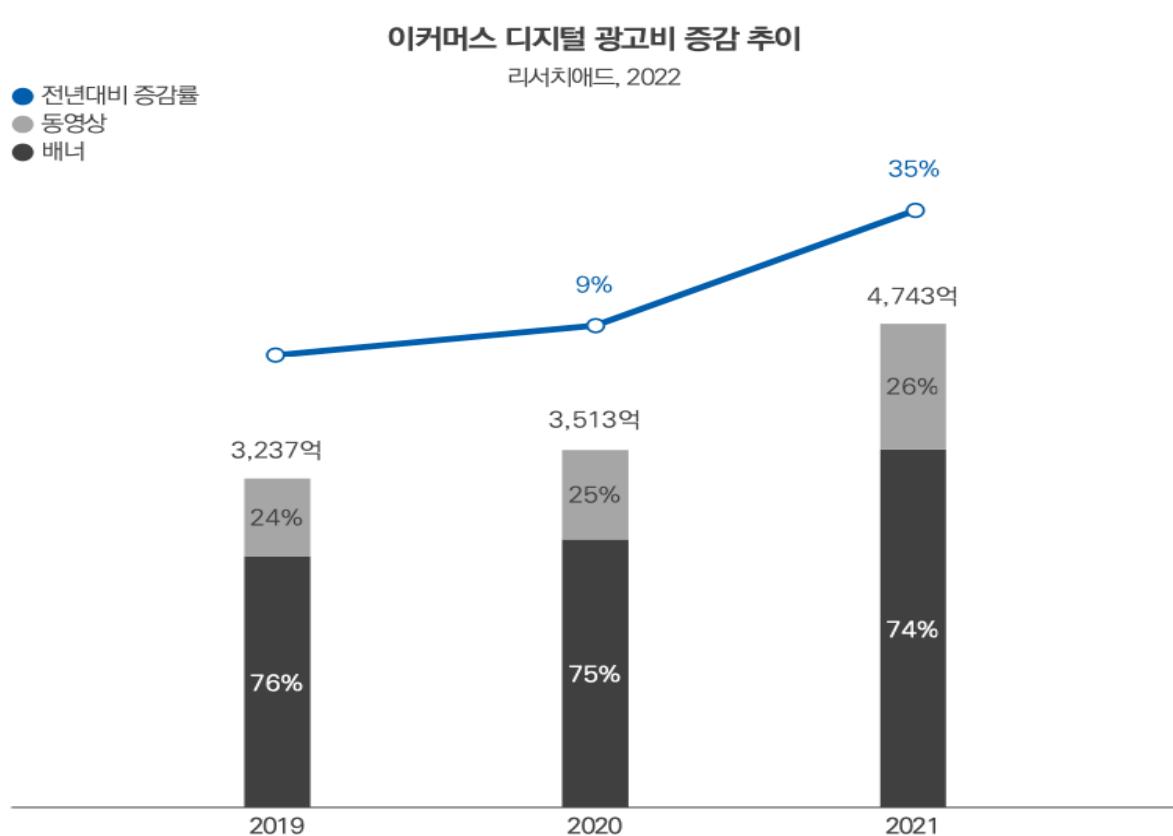
시장분석

2021년 이커머스 시장 규모는 약 186조 원으로 전년 대비 약 15% 증가했다. 비대면 소비의 확장, 이커머스 산업의 발전등으로 성장 추이는 지속될 것으로 전망된다.



[그림 2-1] 이커머스 시장 규모

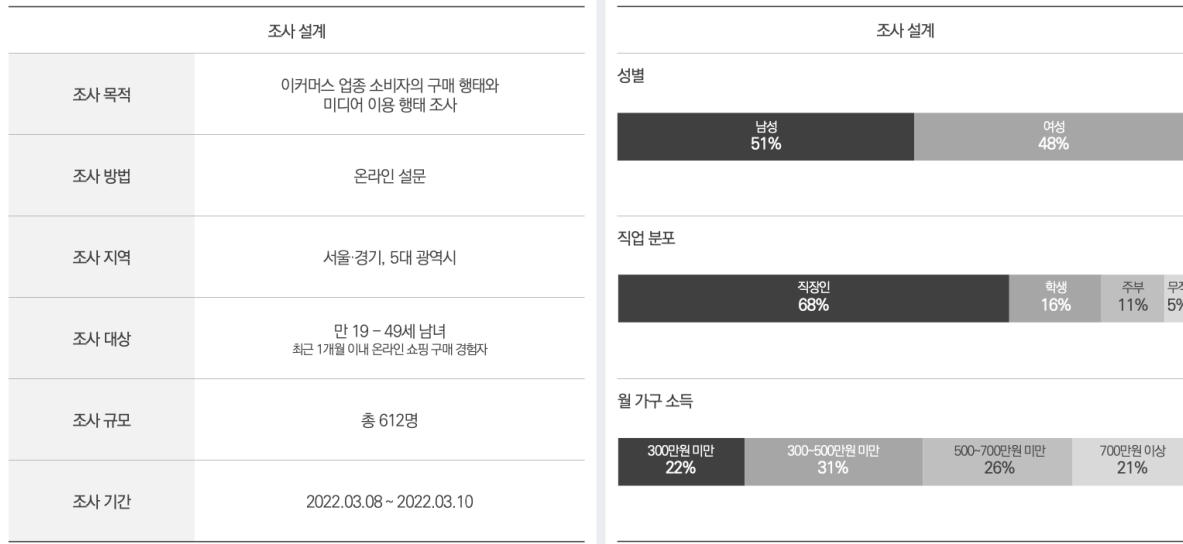
encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	



[그림 2-2] 이커머스 디지털 광고비 증감 추이

이커머스 업종 광고비는 전년 대비 1,230억이 증가했다. 무신사, 올리브영과 같은 안정적인 매출과 고객을 확보한 전문 쇼핑몰들이 카테고리를 확장해 나가면서 사업규모를 확대하고 있고 점점 경쟁이 치열해지는 중이다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	



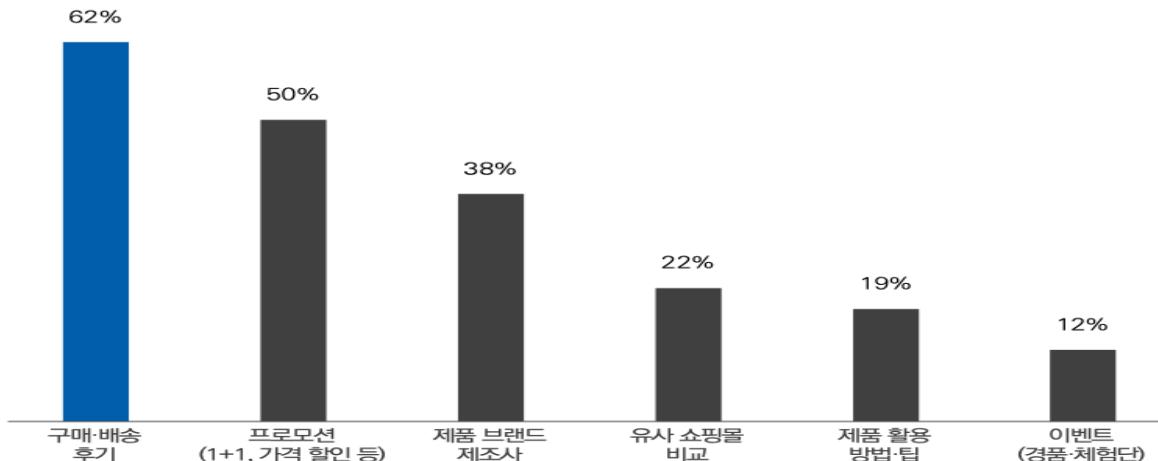
[그림 2-3] 이커머스 소비자들을 대상으로 한 조사

이커머스 소비자들을 대상으로 한 조사를 진행한 결과 이커머스 소비자들이 가장 크게 관심을 갖는 정보는 구입, 배송 후기였고 주 이용 온라인 쇼핑 채널은 네이버와 쿠팡을 가장 많이 이용했다.
(연령대별로 다소 차이가 존재)

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Q. 온라인 쇼핑 관심 정보

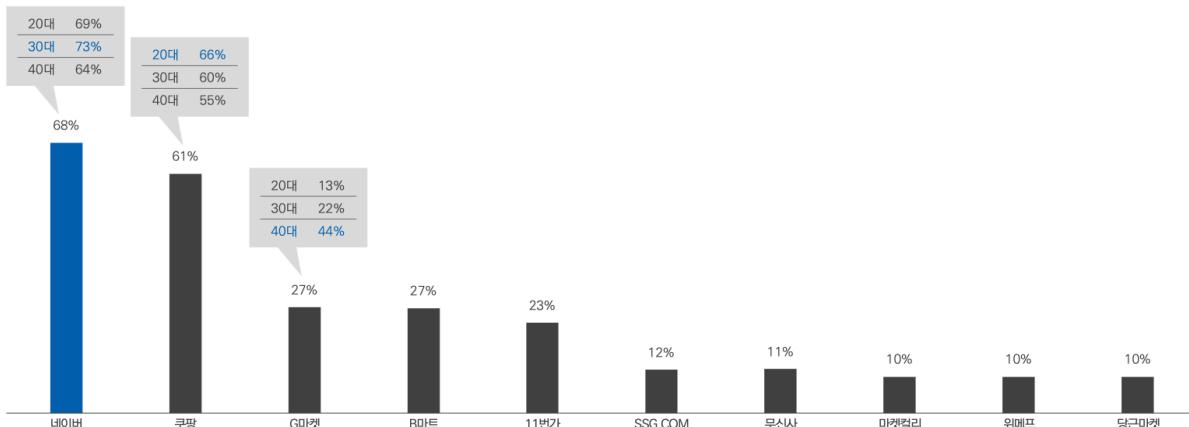
복수 응답



[그림 2-4] 온라인 쇼핑몰 관심 정보

Q. 온라인 쇼핑 주 이용 채널

복수 응답

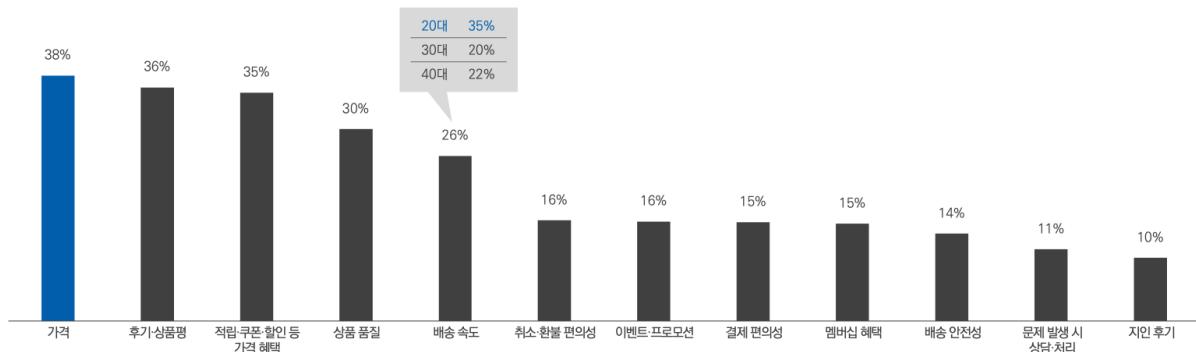


[그림 2-5] 온라인 쇼핑몰 주 이용 채널

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Q. 온라인 쇼핑 채널 선택 요인

복수 응답

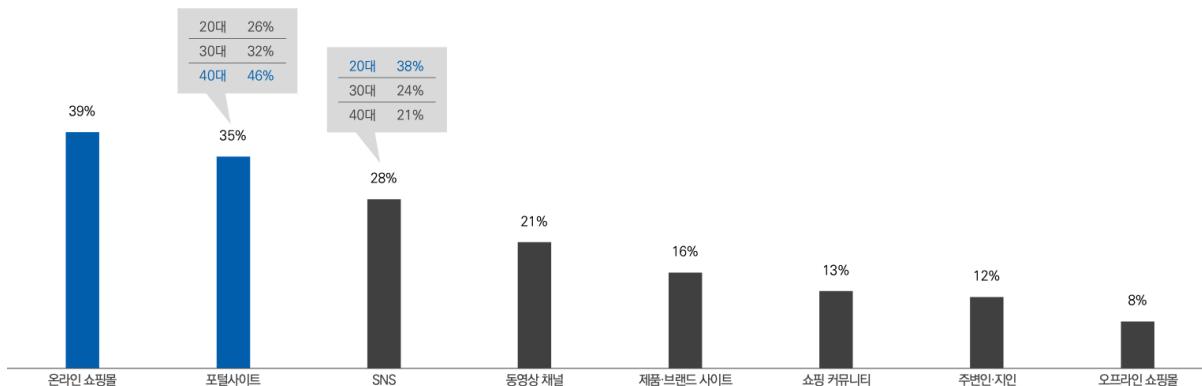


[그림 2-6] 온라인 쇼핑몰 선택 요인

쇼핑 채널 주 선택 요인은 가격과 사용후기로, 소비자들은 온라인 쇼핑 채널을 선택할 때 가격과 사용 후기를 주의 깊게 보는 것을 확인할 수 있었다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Q. 최근 온라인 구매 제품 정보 획득 경로
복수 응답



[그림 2-7] 온라인 제품 정보 획득 경로

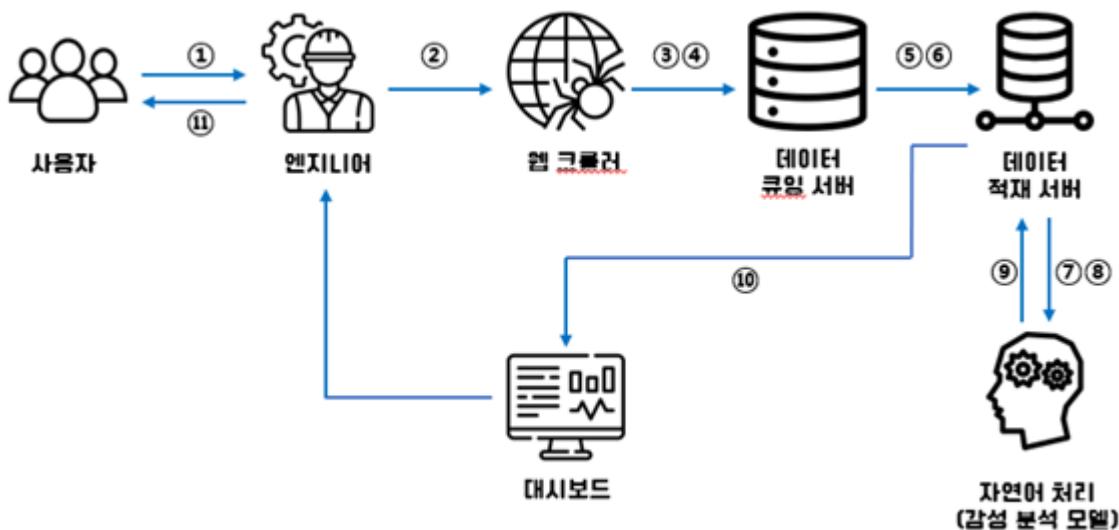
추가로 온라인에서 구매하는 제품의 정보는 온라인 쇼핑몰과 포털사이트에서 획득한다고 답했다. 온라인 구매 시, 쇼핑몰에서 즉시 탐색하는 경우가 가장 많았고 해당 정보 중 가장 신뢰하는 부분은 후기, 리뷰라고 답했다.

이처럼 이커머스 시장의 급속 성장과 동시에 해당 시장의 경쟁이 치열해지고 있다. 따라서 소비자들이 제품 구매 시 가장 크게 영향을 받는 리뷰를 관리하는 전략으로 경쟁에서의 우위를 점할 수 있다고 판단된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.2. 프로세스 분석

2.2.1. 서비스 플로우



[그림 2-8] 서비스 플로우

쇼핑몰 리뷰 분석 요청하기

- ① 사용자가 온라인 쇼핑몰 리뷰 분석을 요청한다.
- ② 엔지니어는 크롤러에게 사용자 쇼핑몰의 리뷰 추출을 요청한다.
- ③ 크롤러는 사용자의 쇼핑몰에 접근하여 리뷰, 별점, 작성일자를 추출한다.
- ④ 크롤러는 추출한 데이터를 카프카 브로커의 Topic에 저장한다.

쇼핑몰 리뷰 데이터 정제 및 저장하기

- ⑤ Topic에 저장된 데이터를 불러와 특수문자를 제거하고 JSON 포맷으로 변환한다.
- ⑥ 정제된 데이터를 인덱싱하여 데이터 적재 서버(OpenSearch)에 저장한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

쇼핑몰 리뷰 **NLP** 처리

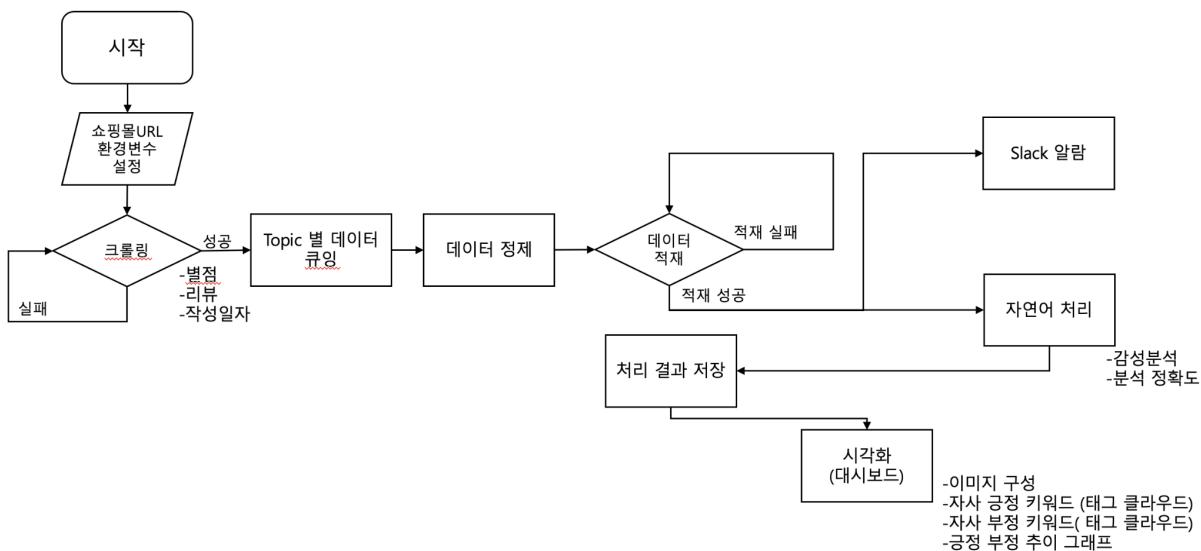
- ⑦ GRU를 사용하여 자사·타사·유사 쇼핑 리뷰 감성 분류를 진행한다.
- ⑧ 데이터 적재 서버에 저장 된 데이터를 불러와 자연어 처리를 한다.
- ⑨ 처리 결과를 다시 데이터 적재 서버에 저장한다.

분석 결과 시각화

- ⑩ 저장 된 데이터의 인덱스를 기반으로 대시보드 형태로 시각화한다.
- ⑪ 사용자에게 해당 대시보드의 주소를 제공한다

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.2.2. 기능 플로우



[그림 2-9] 기능 플로우

쇼핑몰 URL 환경 변수 설정

크롤러 파드 생성 시, 쇼핑몰 URL을 환경 변수로 설정하여 크롤링 할 쇼핑몰을 지정한다.

크롤링

크롤러 파드는 지정 된 쇼핑몰의 리뷰 데이터(별점, 리뷰, 작성일자)를 추출한다. 추출된 데이터는 큐잉 서버의 각 Topic으로 전송한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Topic 별 데이터 큐잉

데이터를 큐잉 서버에 저장하게 되면 데이터의 유실을 방지할 수 있다. 각 제품 별로 Topic이 생성되어 있기 때문에 크롤러 파드는 자신의 제품에 해당하는 Topic에 데이터를 전송한다.

데이터 정제

데이터 정제 서버는 Topic을 구독하고 있다가 Topic에 리뷰 데이터가 쓰이면 데이터를 끌어와 특수 문자를 제거하고 해당 프로젝트에서 사용하는 형태로 인덱싱한 다음 JSON 포맷으로 변환한다.

데이터 적재

정제를 마친 데이터는 데이터 적재 서버로 전송된다.

Slack 알림

데이터 적재 서버에 데이터가 안전하게 저장되면 Slack에 적재 완료 알림이 전송된다.

자연어 처리

데이터 적재 서버에서 데이터를 가져와 자연어 처리(감성 분석, 분석 정확도)를 진행한다.

처리 결과 저장

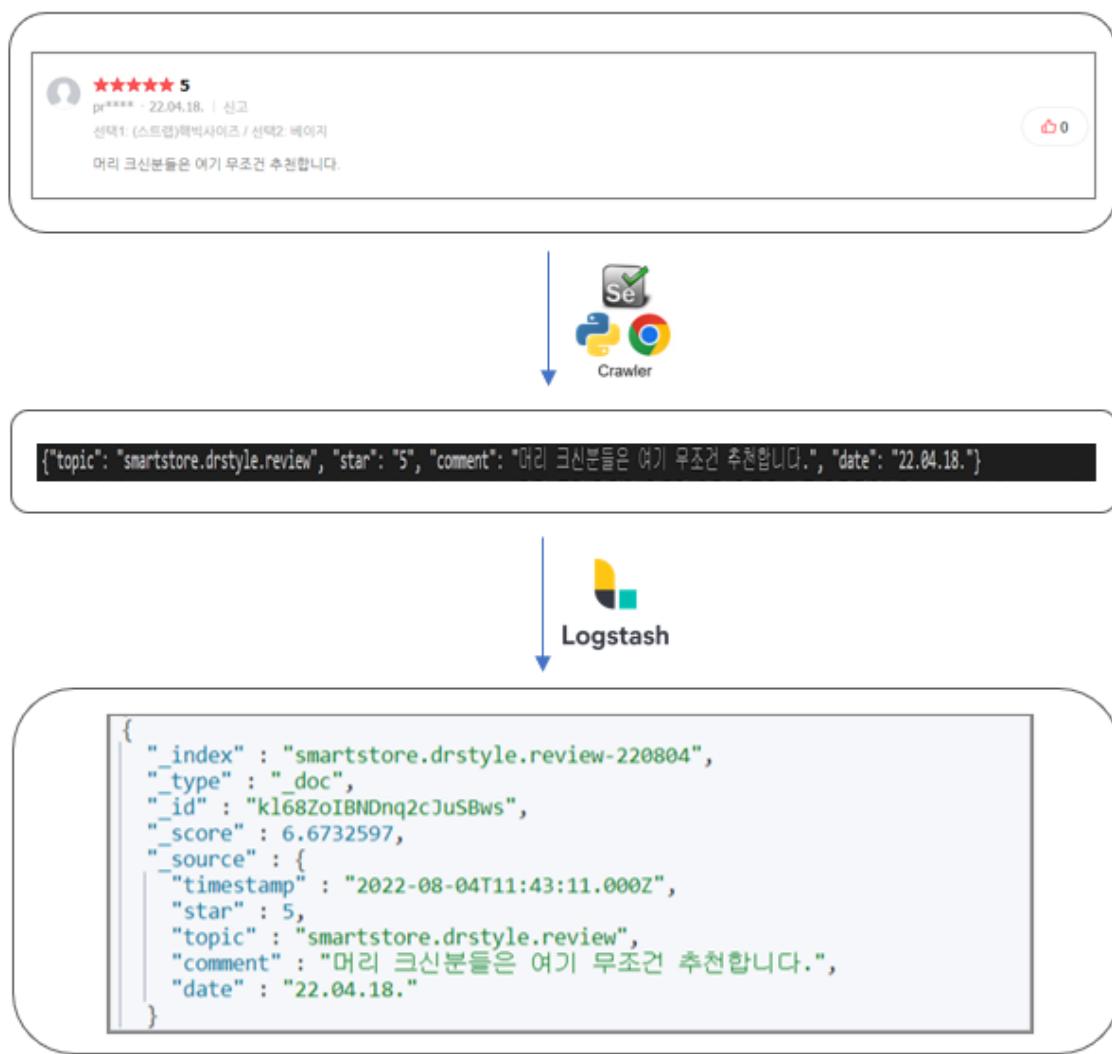
자연어 처리 결과를 데이터 적재 서버에 저장한다.

시각화 (대시보드)

데이터 적재 서버에서 감성 분석 결과를 전송 받아 타겟 키워드를 다양한 차트(Stackbar, Markdown, Data Table 등)를 통해 나타낸다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.2.3. 데이터 플로우



[그림 2-10] 데이터 플로우

HTML 포맷 쇼핑몰 리뷰는 파이썬 크롤러에 의해 추출 되어 일렬의 JSON 포맷으로 변환되고, 이 데이터가 Logstash를 거치면서 인덱싱 및 정제가 되어 multi-row 형태의 JSON 포맷으로 변환된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.2.4. 인프라 요구사항 명세

[개요]

구 분	설 명	개 수
기능 요구사항 (SFR : System Function Requirement)	목표시스템이 반드시 수행하여야 하거나 목표시스템을 이용하여 사용자가 반드시 할 수 있어야 하는 기능(동작)에 대해 기술한 것	4
데이터 요구사항 (DR : Data Requirement)	목표시스템의 서비스에 필요한 초기자료 구축 및 데이터 변환/이관을 위한 대상, 방법, 보안이 필요한 데이터 등 데이터를 구축하기 위해 필요한 사항을 기술한 것	4
성능 요구사항 (PER : Performance Requirement)	목표시스템의 처리속도 및 시간, 처리량, 동적·정적 용량, 가용성 등 성능에 대해 기술한 것	1
시스템 장비구성 요구사항 (ECR : Equipment Composition Requirement)	사업수행을 위해 필요한 H/W, S/W, N/W 등의 도입 장비 내역 및 구성요건(특정 설치시기 혹은 일정, 기존 장비와 호환 필요성 등)에 대해 기술한 것	1
테스트 요구사항 (TER : Test Requirement)	기능의 완성도를 확인하기 위한 단위시험, 통합시험 등의 요건을 기술한 것	1
품질 요구사항 (QR : Quality Requirement)	목표시스템이 가져야 하는 품질 항목, 품질 평가 대상 및 목표 값에 대한 요구사항을 기술한 것	1
프로젝트 관리 요구사항 (PMR : Project Management Requirement)	앞서 제시한 요건 외에 프로젝트의 원활한 수행을 위한 관리 방법 및 추진 단계별 수행방안에 대해 기술한 것	3
합 계		15

[표 2-1] 인프라 요구사항 명세 개요

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

【명세】

구분	ID	요구사항명	기능	세부사항
기능 요구사항	SFR-001	리뷰 크롤링을 통한 쇼핑몰 데이터 추출	데이터 추출	소비자의 쇼핑몰 리뷰에서 리뷰, 별점, 작성일자를 추출
				추출하고자 하는 URL에 따라 추출 진행
				추출이 완료된 후 성공/실패 여부와 추출에 걸린 시간을 Kafka에 전송
	SFR-002	리뷰 데이터 적재 알림 전송	알림 전송	Kafka에 크롤링 한 데이터가 적재 완료될 경우 Slack을 통해 알림 전송
	SFR-003	온라인 쇼핑몰 리뷰를 분석하기 위한 NLP	상위 키워드 추출	빈도 수에 따라 상위 키워드 추출하여 시각화
				20만개의 리뷰와 별점 데이터를 크롤링한 후 test데이터와 train 데이터로 나눔
				train 데이터로 리뷰에 대한 긍정/부정 분석 모델링
			분석 정확도 계산	test 데이터로 모델 검증
			단어 간 유사도 계산	키워드를 지정 후 키워드와 유사한 단어를 추출
	SFR-004	시각화를 통한 대시보드 생성	대시보드 생성	분석 결과를 바탕으로 데이터 프레임 생성
				데이터 프레임에 맞게 대시보드에서 시각화

[표 2-2] 인프라 요구사항 - 기능 요구사항

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

구분	ID	요구사항명	기능	세부사항
데이터 요구 사항	DR-001	크롤링 데이터 유실 방지를 위한 큐잉 서비스	데이터 유실 방지	크롤링 한 데이터를 받아오기 위해 상품 별 Topic 생성 Topic에 쓸인 데이터를 차례로 전달
	DR-002	온라인 쇼핑몰 리뷰 데이터 정제 및 형식 변환	수신 데이터 특수 문자 제거, 인덱싱 및 형식 변환	원하는 상품에 해당하는 Topic의 데이터 수신 수신 받은 데이터의 특수 문자 제거 및 인덱싱 정제된 데이터를 JSON 형태로 변환
	DR-003	온라인 쇼핑몰 리뷰 데이터 NLP를 위한 데이터 적재	크롤링 후 정제된 데이터 저장	자연어 처리를 하기 위한 데이터를 받아와 저장
	DR-004	온라인 쇼핑몰 리뷰 데이터를 NLP한 결과 데이터 적재	감성 분석 결과 데이터 저장	분석 결과를 데이터 프레임에 맞추어 저장
성능 요구 사항	PER-001	질의 · 응답 시간	질의응답 시간 및 오류메시지 응답시간	시스템의 모든 기능은 화면에 출력할 때 브라우저의 영향을 받지 않고 보여주지 않아야 함
				구축 시스템의 사용자 서비스 페이지는 평균 5초 이내에 처리되어야 함
				사용자 요청 작업 관련 평균 시간 초과 응답 시 성능향상 방안을 강구하여야 함

[표 2-3] 인프라 요구사항 - 데이터 요구사항, 성능 요구사항

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

구분	ID	요구사항명	기능	세부사항
시스템 장비 구성 요구 사항	ECR-001	시스템 장비구성 요구사항	시스템 장비구성 요구사항	Crawler: t3.medium * 2 EKS 내 2개의 노드에 지정 ElasticSearch-master: t3.large 1~3 ElasticSearch-data: t3.large 1~3 ElasticSearch-client: t3.large 1~3 Kibana: t3.medium 1~3 Logstash: t3.large 1~3 EKS 내 1~3개의 노드로 가용성 확보
				Jupyter Notebook: g4dn.xlarge 1 Kafka Cluster: t3.medium 3 EC2 내 1 혹은 3개의 노드로 구성
테스트 요구 사항	TER-001	테스트 요구사항	단위 테스트	프로그램 개발 일정 및 테스트 일정에 따라 개발된 프로그램에 대해 단위 테스트 실시 방안을 수립
			통합 테스트	통합 테스트는 최소 1회 이상 실시해야 하며, 테스트 일정에 따라 구체적인 실시방안을 수립하여 수행 후 결과 보고
			시험 운영	계획된 일정에 따라 시험운영 방안을 수립하여 제시

[표 2-4] 인프라 요구사항 - 시스템 장비 구성 요구사항, 테스트 요구 사항

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

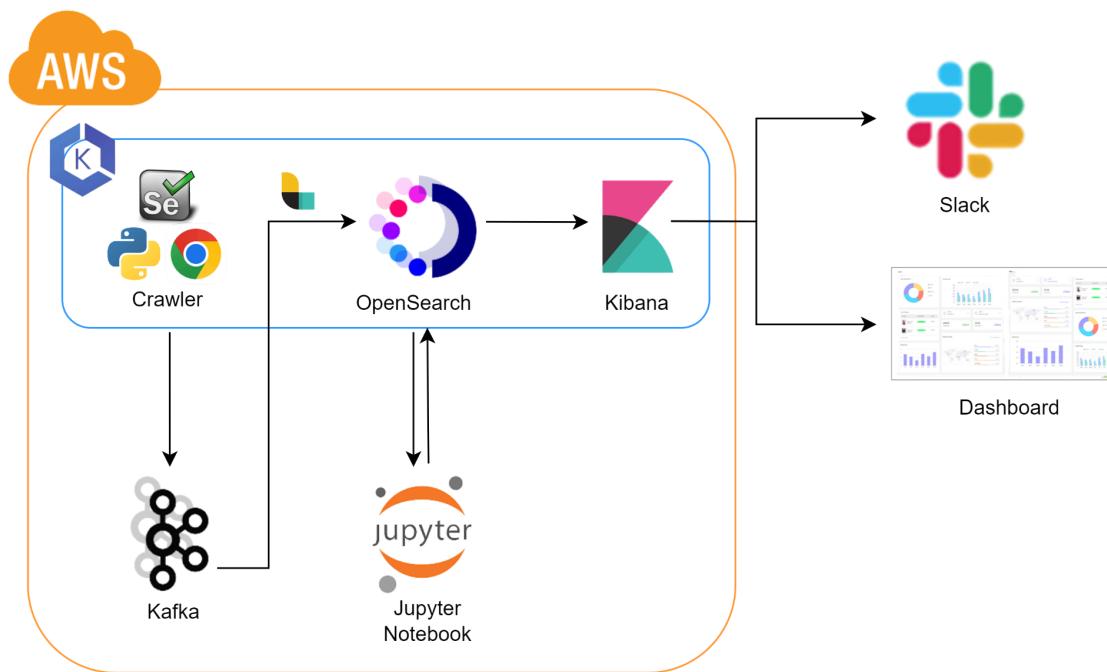
구분	ID	요구사항명	기능	세부사항
품질 요구사항	QR-001	품질 보증 활동	기능 구현의 정확성 향상 방안 수립 및 준수	개발 시스템은 제공되기로 한 기능 요구사항을 모두 제공해야하며, 초기 협의한 요구사항에서 변경이 필요한 경우 주관기관 담당자와 협의하여 요구사항을 변경 협의 및 과업 변경 가능
프로젝트 관리 요구사항	PMR-001	프로젝트 일정관리	프로젝트 일정관리에 관한 사항	일정 계획 제시 - 개발의 완성도를 높이기 위해서 프로젝트 착수에서 종료까지 체계적으로 프로젝트를 관리해야 함 단계별 일정관리 - 각 업무단위별 단계별 일정 계획을 수립하여 제시하고, 일정 지연이 발생하는 경우 별도 계획수립 등 만회 계획을 작성
	PMR-002	투입인력 및 역할 분담	투입인력에 관한 사항	투입인력 및 역할 분담 - 모든 투입인력은 프로젝트 인력으로 구성하여야 하며, 업무분담(역할)을 관리 및 작성
	PMR-003	형상 관리	형상 관리 방안 수립 및 준수	형상 관리 어플리케이션 개발부터 소멸까지의 소스 코드를 포함한 각종 산출물은 중앙에서 통합적으로 변경관리가 가능하도록 관리

[표 2-5] 인프라 요구사항 - 품질 요구사항, 프로젝트 관리 요구사항

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.3. 프로세스 설계

2.3.1. 프로세스 아키텍처



[그림 2-11] 프로세스 아키텍처

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.3.2. 시스템 구성

파이썬 크롤러

원하는 크롤러 규격에 맞는 Docker 이미지를 생성하여 배포에 사용한다.

【 이미지 】

이미지명 : ddung1203/kafkacrawler:10

(링크 : <https://hub.docker.com/repository/docker/ddung1203/kafkacrawler>)

사용 OS : Ubuntu 20.04

【 소프트웨어 버전 】

Software	Version
Python3	3.8
Python3-pip	20.0.2
Selenium	4.3.8
ChromeDriver	103.0.5060.134

[표 2-6] Python Crawler 소프트웨어 버전

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Kafka

[소프트웨어 버전]

Software	Version
Ubuntu	Amazon Linux 5.10
Open-JDK(자바)	1.8.0
Kafka	kafka_2.12-2.5.0

[표 2-7] Kafka 소프트웨어 버전

Jupyter Notebook

[소프트웨어 버전]

Software	Version
Ubuntu	20.04
CUDA	11.2
cuDNN	8.1
Anaconda	4.13.0
Python	3.9.12

[표 2-8] Jupyter Notebook 소프트웨어 버전

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[프로그램 모듈 버전]

Module	Description
re	일치하는 문자열 집합 지정
pandas	데이터 조작 및 분석을 위한 라이브러리
numpy	행렬이나 일반적으로 대규모 다차원 배열을 쉽게 처리를 위한 라이브러리
matplotlib.pyplot	MATLAB과 비슷하게 명령어 스타일로 동작하는 함수의 모음
boto3	Amazon EC2 및 Amazon S3와 같은 AWS 서비스를 생성, 구성 및 관리
smart_open	S3, HDFS, WebHDFS 또는 로컬 파일 간에 큰 파일을 효율적으로 스트리밍 하기 위한 라이브러리
Mecab	오픈 소스 형태소 분석 엔진인 MeCab을 사용하여 한국어 형태소 분석을 하기 위한 프로젝트
train_test_split	사이킷런(scikit-learn)의 model_selection 패키지 안에 train_test_split 모듈 손쉽게 train set과 test set을 분리
tensorflow	다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리
Elasticsearch	Elasticsearch의 공식 하위 클라이언트 파이썬의 모든 Elasticsearch 관련 코드를 위한 공통 기반을 제공
json_normalize	반구조화된 JSON 데이터를 일차원 표로 정규화
datetime	날짜와 시간을 조작하는 클래스

[표 2-9] Jupyter Notebook 프로그램 모듈 버전

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

ELK Stack

ELK Stack은 EKS 위에 DockerHub에 업로드 되어있는 이미지를 사용하여 배포한다.

[Logstash 이미지]

이미지명 : docker.elastic.co/logstash/logstash:7.10.2

[OpenSearch 이미지]

이미지명 : opendistro-for-elasticsearch:1.13.2

[Kibana 이미지]

이미지명 : opendistro-for-elasticsearch-kibana:1.13.2

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.3.3. Usecase 시나리오

【 기능 요구사항 】

요구사항 고유 번호	SFR-001				
요구사항 명칭	리뷰 크롤링을 통한 쇼핑몰 데이터 추출				
요구사랑 분류	기능 요구사항				
요구사항 항목	상세 설명				
개요	엔지니어는 크롤러를 통해 사용자 쇼핑몰의 리뷰를 추출할 수 있다.				
관련 엑터	주 엑터	엔지니어			
	보조 엑터	파이썬 크롤러, EKS 클러스터			
요구능력	클라이언트의 쇼핑몰 URL로 접속해 리뷰를 크롤링 해 올 수 있는 파이썬 파일 작성 개발 능력				
사전 조건	크롤링을 위한 모듈이 설치되어있는 파드(서버)와 EKS가 생성되어 있어야한다.				
시나리오	<ol style="list-style-type: none"> 사용자는 엔지니어에게 자사 쇼핑몰 데이터 키워드 분석을 요청한다. 엔지니어는 사용자로부터 쇼핑몰 URL을 전달받아 크롤링 파드를 생성하는 YAML 파일 내부에서 해당 URL을 환경변수로 설정한 후 EKS 위에 파드를 생성한다. 파드가 생성되면서 크롤러 파드는 내장 된 파이썬 코드를 실행한다. 크롤러는 사용자 쇼핑몰을 렌더링하여 별점, 리뷰, 작성일자 데이터를 추출한다. 크롤러는 추출한 데이터를 Kafka Broker 서버에 전송한다. 크롤러는 배치 작업을 통해 매일 정각 시간에 ④~⑤번 시나리오를 반복한다. 				
산출 정보	Data Collector 가이드 문서, 표준 연동 계획서, 인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드				

【표 2-10】 기능 요구사항 - SFR-001

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	SFR-002			
요구사항 명칭	리뷰 데이터 적재 알림 전송			
요구사랑 분류	기능 요구사항			
요구사항 항목	상세 설명			
개요	리뷰가 OpenSearch에 적재되면 Slack으로 적재 완료 알림을 받을 수 있다.			
관련 엑터	주 엑터	Kibana 서버		
	보조 엑터	EKS 클러스터		
요구능력	Kibana에서 OpenSearch의 데이터 적재 상황을 모니터링하고 알림 트리거 설정을 할 수 있는 개발 능력			
사전 조건	데이터 적재가 가능한 OpenSearch 파드(서버)와 EKS가 생성되어 있어야 한다.			
시나리오	1. 엔지니어는 Kibana가 OpenSearch에 데이터가 적재되는지 모니터링하도록 설정하여 OpenSearch에 데이터 적재 시 Slack으로 알림이 전송되게끔 Alerting Trigger를 설정한다. 2. Kafka는 크롤러로부터 전달 받은 데이터를 OpenSearch에 전송한다. 3. OpenSearch는 전송받은 데이터를 적재한다. 4. Kibana에서 감지하고 미리 지정되어있던 액션(Slack 알림)을 수행한다.			
산출 정보	인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드			

[표 2-11] 기능 요구사항 - SFR-002

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	SFR-003				
요구사항 명칭	온라인 쇼핑몰 리뷰를 분석하기 위한 NLP				
요구사랑 분류	기능 요구사항				
요구사항 항목	상세 설명				
개요	리뷰 데이터를 활용해 자연어 처리를 할 수 있다.				
관련 엑터	주 엑터	엔지니어			
	보조 엑터	Jupyter Notebook			
요구능력	자연어 처리를 할 수 있는 코드를 이해할 수 있는 능력, 머신러닝 파이썬 코드를 구현할 수 있는 개발 능력				
사전 조건	그래픽 드라이버가 설치되어 있는 서버와 Jupyter Notebook이 설치 된 서버가 존재해야 한다.				

→ 다음 페이지에 이어진다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

시나리오	1. 훈련 데이터와 테스트 데이터를 분리하는 작업을 진행한다.
	<p>1. 엔지니어는 정규 표현식을 사용하여 한글을 제외하고 모두 제거하는 데이터 정제를 진행한다.</p> <p>2. 형태소 분석기 Mecab을 사용하여 토큰화 작업을 수행한다.</p> <p>3. 불용어를 지정하여 필요없는 토큰들을 제거한다.</p> <p>4. 등장 빈도수가 1인 단어들의 수를 제외한 단어의 개수를 단어 집합의 최대 크기로 제한한다.</p> <p>5. 서로 다른 길이의 샘플들의 길이를 동일하게 맞춰주는 패딩 작업을 진행한다.</p> <p>6. 출력층에 로지스틱 회귀를 사용해야 하므로 활성화 함수로는 시그모이드 함수를 사용하고, 손실 함수로 크로스 엔트로피 함수를 사용한다. 하이퍼파라미터인 배치 크기는 64이며, 15 에포크를 수행한다.</p> <p>7. 검증 데이터 손실이 증가하면, 과적합 징후므로 검증 데이터 손실이 4회 증가하면 정해진 에포크가 도달하지 못하였더라도 학습을 조기 종료한다.</p> <p>8. Validation_split=0.2을 사용하여 훈련 데이터의 20%를 검증 데이터로 분리해서 사용하고, 검증 데이터를 통해서 훈련이 적절히 되고 있는지 확인한다.</p> <p>9. 엔지니어는 OpenSearch에 저장되어 있는 데이터를 Jupyter Notebook에서 Search로 데이터를 Get한다.</p> <p>10. 문장에 대한 예측을 위해서는 학습하기 전 전처리를 동일하게 적용한다.</p> <p>11. 엔지니어는 별점과 리뷰의 긍·부정 반응이 불일치하는 데이터를 조정한다.</p> <p>12. 추출 된 키워드 별로 "긍정·부정·중립" 라벨링을 진행한다.</p> <p>13. 기존의 ①번 항목의 데이터에 ⑨번 항목의 데이터를 추가하여 모델의 정확도를 향상시킨다.</p>
산출 정보	인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드

[표 2-12] 기능 요구사항 - SFR-003

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	SFR-004			
요구사항 명칭	시각화를 통한 대시보드 생성			
요구사랑 분류	기능 요구사항			
요구사항 항목	상세 설명			
개요	감성분석이 진행된 데이터를 Kibana 대시보드에 원하는 형태의 테이블 및 그레프를 표시할 수 있다.			
관련 엑터	주 엑터	엔지니어		
	보조 엑터	Kibana, OpenSearch, Jupyter Notebook, EKS 클러스터		
요구능력	Kibana의 인덱스를 이해할 수 있는 능력, Kibana의 다양한 시각화 도구를 활용할 수 있는 개발 능력			
사전 조건	Kibana가 설치 된 파드(서버)와 EKS가 존재해야 한다.			
시나리오	1. 엔지니어는 Kibana에서 인덱스 패턴을 생성한다. 2. 엔지니어는 Jupyter Notebook에서 감성분석이 진행된 데이터를 OpenSearch에 저장할 때 Kibana에서 생성한 인덱스를 대상으로 하여 전송한다. 3. 엔지니어는 OpenSearch에 저장된 데이터를 기반으로 하여 여러 차트(Stackbar, Markdown, Data Table 등)를 사용하여 데이터를 시각화한다. 4. 엔지니어는 해당 대시보드 주소를 사용자에게 전달한다. 5. 사용자는 대시보드에 접속하여 소비자 반응을 판단하고, 추출 키워드를 활용하여 판매 제품의 수요를 높일 수 있는 방법을 모색한다.			
산출 정보	인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드			

[표 2-13] 기능 요구사항 - SFR-004

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

【 데이터 요구사항 】

요구사항 고유 번호	DR-001				
요구사항 명칭	크롤링 데이터 유실 방지를 위한 큐잉 서비스				
요구사랑 분류	데이터 요구사항				
요구사항 항목	상세 설명				
개요	크롤링한 데이터를 유실 없이 저장소 서버로 전송할 수 있다.				
관련 엑터	주 엑터	Kafka Broker 서버			
	보조 엑터	Kafka Zookeeper 서버			
요구능력	Kafka Topic을 생성하고 해당 Topic으로 데이터를 전송할 수 있도록 설정하는 개발 능력				
사전 조건	Kafka Broker 및 Zookeeper 서버, 데이터를 가져오고 전송할 수 있는 서버가 존재해야 한다.				
시나리오	<ol style="list-style-type: none"> 엔지니어는 Kafka Broker 서버에 상품 별로 Topic을 생성한다. Topic의 복제수와 파티션은 3으로 설정한다. Kafka Broker 서버는 Producer(크롤러 서버)로부터 지정된 Topic으로 데이터를 차례로 전송 받는다. Kafka Broker 서버는 메시지 가져오기 요청이 들어온 Consumer(Logstash 서버)에게 데이터를 넘겨준다. 				
산출 정보	표준 연동 계획서, 인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드				

[표 2-14] 데이터 요구사항 - DR-001

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	DR-002			
요구사항 명칭	온라인 쇼핑몰 리뷰 데이터 정제 및 형식 변환			
요구사랑 분류	데이터 요구사항			
요구사항 항목	상세 설명			
개요	일렬의 데이터로 수신 된 데이터를 Multi-row JSON 데이터로 변환하여 전송한다.			
관련 엑터	주 엑터	Logstash 서버		
	보조 엑터	EKS 클러스터		
요구능력	Logstash Configmap 파일을 작성할 수 있는 개발 능력			
사전 조건	Logstash가 설치 된 파드(서버)가 존재해야 한다.			
시나리오	1. Kafka Consumer인 Logstash 서버는 Kafka Broker 서버에게 메시지 가져오기를 요청하여 원하는 Topic의 데이터를 가져온다. 2. Logstash 서버는 엔지니어가 설정한 기준에 맞추어 리뷰 데이터를 정제(특수문자 제거, 인덱싱 등)한다. 3. 엔지니어가 사용하는 OpenSearch 서버를 호스팅하여 정제한 데이터를 JSON 형식의 데이터로 변환하여 전송한다.			
산출 정보	표준 연동 계획서, 인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드			

[표 2-15] 데이터 요구사항 - DR-002

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	DR-003	
요구사항 명칭	온라인 쇼핑몰 리뷰 데이터 NLP를 위한 데이터 적재	
요구사항 분류	데이터 요구사항	
요구사항 항목	상세 설명	
개요	NLP를 수행하기 위한 데이터를 저장한다.	
관련 엑터	주 엑터	OpenSearch 서버
	보조 엑터	EKS 클러스터
요구능력	정제된 데이터를 받아와 저장 서버에 저장할 수 있는 개발 능력	
사전 조건	OpenSearch가 설치된 파드(서버)가 존재해야 한다.	
시나리오	Logstash 서버가 전송한 JSON 데이터를 받아와 저장한다.	
산출 정보	표준 연동 계획서, 인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드	

[표 2-16] 데이터 요구사항 - DR-003

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

요구사항 고유 번호	DR-004						
요구사항 명칭	온라인 쇼핑몰 리뷰 데이터를 NLP한 결과 데이터 적재						
요구사항 분류	데이터 요구사항						
요구사항 항목	상세 설명						
개요	NLP 처리가 끝난 데이터를 저장한다.						
관련 엑터	주 엑터	OpenSearch 서버					
	보조 엑터	EKS 클러스터					
요구능력	Jupyter Notebook에서 처리 된 데이터를 저장 서버로 전송할 수 있는 개발 능력						
사전 조건	OpenSearch가 설치 된 파드(서버)가 존재해야 한다.						
시나리오	Jupyter Notebook 서버가 전송한 데이터를 받아와 저장한다.						
산출 정보	인프라 요구사항 정의서, Git 레포지토리 가이드 문서, 프로젝트 최종 보고서, 구현 소스코드						

[표 2-17] 데이터 요구사항 - DR-004

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.3.4. 데이터 연동 규격

크롤러 - Kafka 연동 규격

[파이썬 크롤러]

인터넷 쇼핑몰 리뷰를 크롤링한 후 JSON(Dictionary) 형태로 저장하고 지정한 Kafka의 Topic으로 전송하는데 사용한다.

필드명	star	comment	date
Broker Address	3.38.10.106:9092	3.34.18.190:9092	13.209.146.71:9092
Topics	smartstore.goodnara.review		
	smartstore.180store.review		
	smartstore.theshopsw.review		
	smartstore.thecheaper.review		
	smartstore.cloony.review		
	smartstore.drstyle.review		
Input Data Format	HTML		
Output Data Format	JSON(List in a row)		
Parameter	M/O	Types	Description
url	M	string	크롤링할 주소
user_agent	M	string	사용자 애이전트
comment, star, date	O	list	크롤링 한 데이터 임시 저장

[표 2-18] Crawler - Kafka 연동 규격

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Kafka - OpenSearch 연동 규격

[Inbound]

Kafka에서 Logstash로 데이터를 전송할 때 사용한다.

Broker Address	3.38.10.106:9092	3.34.18.190:9092	13.209.146.71:9092
Topics	smartstore.goodnara.review		
	smartstore.180store.review		
	smartstore.theshopsw.review		
	smartstore.thecheaper.review		
	smartstore.cloony.review		
	smartstore.drstyle.review		
Input Data Format	JSON(List in a row)		
Output Data Format	JSON(List in a row)		
Parameter	M/O	Types	Description
Bootstrap_servers	M	string	클라이언트가 접근할 Kafka Broker
Topics	M	string	데이터를 받아올 Kafka의 Topic 이름
consumer_threads	M	int	컨슈머 쓰레드

[표 2-19] Kafka - OpenSearch 연동 규격 - Input

```

input {
  kafka {
    bootstrap_servers => "카프카 브로커 서비스 DNS 주소"
    topics => [Topic 이름 리스트]
    consumer_threads => 3
    isolation_level => "read_committed"
    value_deserializer_class => "org.apache.kafka.common.serialization.StringDeserializer"
    auto_offset_reset => "earliest"
    group_id => "smartstore"
  }
}

```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Filter]

```
mutate {
    gsub => ["message", "[\\\"/{}]", ""]
}
```

'gsub' Filter Plugin을 이용하여 특수문자 제거한다.

```
kv {
    field_split => ","
    value_split => ":"
}
```

Logstash는 하나의 message 필드에 모든 정보를 담아온다.

message 필드 안에 값을 "," 를 기준으로 데이터를 나누고, "."을 기준으로 key:value 형태로 변형하여 각각의 필드로 생성한다.

```
mutate {
    remove_field => [ "port", "@version", "host", "message", "@timestamp", "yy", "mm", "dd" ]
    rename => { " comment" => "comment" }
    rename => { " date" => "date" }
    rename => { " star" => "star" }
}
```

분석에 필요하지 않은 포트, version, host, message 필드를 제거한다. gsub로 특수문자 제거 시, 필드명 맨 앞에 공백이 붙으므로 필드 이름을 재설정해주어야한다.

```
mutate {
    convert => {
        "star" => "integer"
    }
}
```

String으로 설정되어있는 star 필드 값을 integer 형태로 변환한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Outbound]

Host Address	http://elasticsearch-client-http.ELK.svc.cluster.local:9200		
Input Data Format	JSON(List in a row)		
Output Data Format	JSON(Multi-row List)		
Parameter	M/O	Types	Description
codec	M	string	데이터 출력 정보 확인
			데이터 format 변환
hosts	M	string	데이터를 받아올 Kafka의 Topic 이름
index	M	string	OpenSearch로 보낼 데이터의 인덱스명
timeout	M	int	타임아웃까지 소요되는 시간

[표 2-20] Kafka - OpenSearch 연동 규격 - Outbound

```

output {
  stdout { codec => rubydebug }
  # 스마트 스토어 모자 Topic
  if [topic] =~ "smartstore.goodnara.review" {
    elasticsearch {
      hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
      index => "smartstore.goodnara.review-%{[@metadata][yymmdd]}"
      codec => "json"
      timeout => 120
    }
  } # if end
  else if [topic] =~ "smartstore.drstyle.review" {
    elasticsearch {
      hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
      index => "smartstore.drstyle.review-%{[@metadata][yymmdd]}"
      codec => "json"
      timeout => 120
    }
  } # if end
  else if [topic] =~ "smartstore.thecheaper.review" {
    elasticsearch {
      hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
      index => "smartstore.thecheaper.review-%{[@metadata][yymmdd]}"
      codec => "json"
      timeout => 120
    }
  }
}

```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

# if end
# 스마트 스토어 티셔츠 Topic
else if [topic] =~ "smartstore.180store.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
        index => "smartstore.180store.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
else if [topic] =~ "smartstore.cloony.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
        index => "smartstore.cloony.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
else if [topic] =~ "smartstore.theshopsw.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
        index => "smartstore.theshopsw.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
} # output end

```

Stdout를 사용하여 OpenSearch로 데이터를 전송한다. Index 이름은 “Topic 이름 + 날짜” 형식으로 형식으로 지정한다. 데이터는 최종적으로 JSON 형식으로 변환된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.4. 프로세스 구현

2.4.1. 인프라 구현

EKS

[EKS 사용 목적]

Amazon EKS는 자체적으로 여러 가용 영역에서 **Kubernetes** 컨트롤 플레인을 실행하고 확장시켜주기 때문에 고가용성과 확장성이 보장된다. 또한 부하에 따라 컨트롤 플레인의 인스턴스를 자동 확장하고 헬스 체크를 통해 인스턴스를 교체하며 버전 업데이트 및 패치 기능을 제공하기 때문에 AWS 상에서 컨테이너화된 애플리케이션 관리를 자동화하기 용이하다.

[Terraform을 통한 EKS 배포]

>> 배포 모듈 정보

모듈명 : Young-ook/terraform-aws-eks

모듈 Registry 주소 : <https://registry.terraform.io/modules/Young-ook/eks/aws/latest>

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> 배포 모듈 정보

EKS 코드 트리

작업 디렉토리 : [Datapipeline_Project/terraform/EKS](#)

```

└── default.auto.tfvars
└── fixture.tci.tfvars
└── main.tf
└── modules
    ├── addonalb-ingress
    ├── app-mesh
    ├── cluster-autoscaler
    ├── container-insights
    ├── ecr
    ├── iam-role-for-service-account
    ├── karpenter
    ├── lb-contoller
    ├── metrics-server
    ├── node-termination-handler
    └── prometheus
└── outputs.tf
└── variables.tf

```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Terraform 구성 파일

파일명 : `fixture.tc1.tfvars`

```

aws_region      = "ap-northeast-2"
azs            = ["ap-northeast-2a", "ap-northeast-2b", "ap-northeast-2c"]
cidr          = "10.1.0.0/16"
enable_igw    = true
enable_ngw    = true
single_ngw   = true
name          = "eks-autoscaling-tc1"
tags = {
  env = "dev"
  test = "tc1"
}
kubernetes_version = "1.21"
enable_ssm       = true
managed_node_groups = [
  {
    name      = "crawler"
    min_size  = 1
    max_size  = 6
    desired_size = 1
    instance_type = "t3.medium"
  },
  {
    name      = "ElasticSearch-master"
    min_size  = 1
    max_size  = 3
    desired_size = 1
    instance_type = "t3.large"
  },
  {
    name      = "ElasticSearch-data"
    min_size  = 1
    max_size  = 3
    desired_size = 1
    instance_type = "t3.large"
  },
],
  
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
{
  name      = "ElasticSearch-client"
  min_size  = 1
  max_size  = 3
  desired_size = 1
  instance_type = "t3.large"
},
{
  name      = "Kibana"
  min_size  = 1
  max_size  = 3
  desired_size = 1
  instance_type = "t3.medium"
},
{
  name      = "Logstash"
  min_size  = 1
  max_size  = 3
  desired_size = 1
  instance_type = "t3.large"
}
]
node_groups = []
fargate_profiles = []
```

EKS 모듈을 사용하였기 때문에 위의 구성파일을 사용하여 사용자 정의 매개변수에 대한 옵션을 설정하여 배포할 수 있다.

서울 리전(ap-northeast-2)에 1.21 버전의 EKS 클러스터를 생성한다. 3개의 가용 영역(ap-northeast-2a, ap-northeast-2b, ap-northeast-2c)에 소프트웨어 별로 관리형 노드 그룹을 배포한다. 각 소프트웨어가 사용하는 크기에 맞게 노드 그룹의 인스턴스 타입을 설정하고 최소 개수, 최대 개수, 원하는 개수를 지정한다. 현재 프로젝트에서는 인프라의 기능 별 분리를 위해 Crawler, OpenSearch(Master, Data, Ingest), Logstash, Kibana 총 6개의 노드 그룹을 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[EKS 배포]

테라폼으로 EKS를 배포하기 위해서는 우선 테라폼을 실행해야한다. 해당 디렉토리를 초기화 한다.

```
$ terraform init
```

앞서 설정한 사용자 정의 매개 변수에 대한 옵션이 적용된 EKS 클러스터를 배포한다.

매개 변수 옵션은 테라폼 명령어에 **var-file** 옵션을 추가하여 사용 가능하다.

apply 시 사용자의승인 절차 없이 배포되게 하기 위해 **--auto-approve** 옵션을 사용한다.

```
$ terraform plan -var-file fixture.tc1.tfvars
$ terraform apply -var-file fixture.tc1.tfvars --auto-approve
```

[kubeconfig 파일 생성]

```
$ aws eks update-kubeconfig --region ap-northeast-2 --name eks-autoscaling-tc1
```

기본적으로 **kubectl**은 **\$HOME/.kube** 디렉터리에서 **config**라는 이름의 파일을 찾는다. AWS EKS에서 쿠버네티스를 사용하기 위해 AWS CLI에서 **aws eks update-kubeconfig** 명령을 사용하여 자동으로 **kubeconfig** 파일을 생성한다. **--region** 옵션에 클러스터가 존재하는 리전을 지정하고 **--name** 옵션에 클러스터의 이름을 지정하여 앞서 생성한 **eks-autoscaling-tc1** 클러스터에서 **kubectl** 명령어를 사용할 수 있도록 설정한다.

📌 주의사항

- AWS CLI로 **kubeconfig**를 생성하려면 버전 **1.23.11** 또는 **2.6.3** 이상이 설치되어 있어야한다.
- 지정한 클러스터에서 **eks:DescribeCluster** API 작업을 사용할 수 있는 권한이 있어야한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

파이썬 크롤러

【 파이썬 크롤러 사용 목적 】

사용자의 쇼핑몰 리뷰에서 분석을 위해 필요한 항목만 추출하기 위해 파이썬 크롤러를 사용한다.

【 크롤러 파드 생성 】

>> Namespace 생성

크롤러 파드를 논리적으로 분리하여 확인할 수 있도록 **crawler Namespace**를 생성한다.

```
$ kubectl create ns crawler
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> 파드 생성

작업 디렉토리 : [Datapipeline_Project/Crawler](#)

파일 이름 : [kafka_producer_pod.yaml](#)

```
apiVersion: batch/v1
kind: CronJob
metadata:
  namespace: crawler
  name: crawler-1
  labels:
    app: crawler
spec:
  schedule: "00 00 * * *"
  jobTemplate:
    spec:
      template:
        metadata:
          labels:
            app: crawler
          annotations:
            "cluster-autoscaler.kubernetes.io/safe-to-evict": "false"
        spec:
          nodeSelector:
            Name: Crawler
          containers:
            - name: crawler-1
              image: ddung1203/kafkacrawler:10
              resources:
                requests:
                  memory: "3000Mi"
                  cpu: "1500m"
                limits:
                  memory: "3000Mi"
                  cpu: "1500m"
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

env:
- name: url
  value: 'https://smartstore.naver.com/goodnara/products/371623918?'
- name: topic
  value: smartstore.goodnara.review
- name: server
  value: "3.38.10.106:9092,3.34.18.190:9092,13.209.146.71:9092"
command: ["/bin/sh", "-c"]
args: ["cd /Datapipeline_Project/crawler; ./kafka_producer.py"]
restartPolicy: Never
  
```

Crawler 파드 생성을 위한 파일이다.

크롤링을 위한 소프트웨어와 파일 kafka_producer.py가 설치되어있는 Ubuntu 이미지 ddung1203/kafkacrawler:10 를 사용한다. 생성 된 파드 내에서 지정 된 환경변수를 기반으로 kafka_producer.py 를 실행한다.

환경변수

ENV	Description
url	리뷰 분석을 요청한 소비자의 쇼핑몰 url
topic	추출한 리뷰 데이터를 전송할 Kafka Broker의 Topic명
server	추출한 리뷰 데이터를 전송할 Kafka Broker 서버명

[표 2-21] Crawler 파드 환경 변수

앞서 생성한 EKS 클러스터의 **crawler** 노드 그룹에 크롤러 파드를 생성한다.

```
$ kubectl create -f kafka_producer_pod.yaml
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

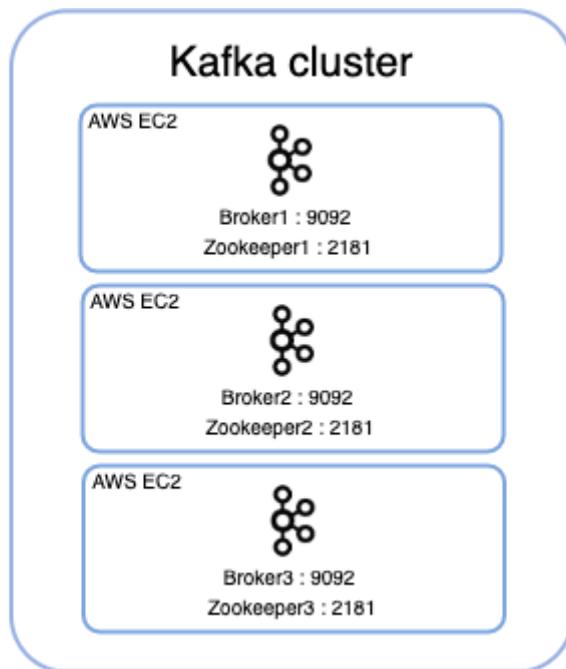
Kafka

[Kafka 사용 목적]

크롤러에서 적재 서버로 쇼핑몰의 리뷰 데이터 전송 시 발생할 수 있는 데이터 유실을 방지하기 위해 Kafka의 큐잉 기능을 사용한다.

[Kafka 인프라 생성]

>> 아키텍처 구성도



[그림 2-12] 카프카 클러스터

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Zookeeper 설정]

각 인스턴스에 설치된 Kafka의 config/zookeeper.properties 파일은 하나의 Zookeeper를 실행하는데 쓰이는 설정 파일이다. zookeeper1.properties, zookeeper2.properties, zookeeper3.properties와 같이 여러 개의 설정파일을 만들고 하나의 장비에서 다중으로 실행할 수 있다. 설정 파일을 다음과 같이 3대의 서버에 동일하게 추가한다.

새로 추가한 설정값은 클러스터를 구성하는데 필요한 설정 값들인데 여기서 주의할 점은 모든 Zookeeper 서버들은 동일한 변수 값을 가지고 있어야 한다.

파일명 : config/zookeeper.properties

```
# 주키퍼의 트랜잭션 로그와 스냅샷이 저장되는 데이터 저장 경로(중요)
dataDir=/tmp/zookeeper

# 주키퍼 사용 TCP Port
clientPort=2181

# 팔로워가 리더와 초기에 연결하는 시간에 대한 타임 아웃 tick의 수
initLimit=5

# 팔로워가 리더와 동기화 하는 시간에 대한 타임 아웃 tick의 수(주키퍼에 저장된 데이터가 크면 수를 늘려야함)
syncLimit=2

# server.1에 자기 자신은 0.0.0.0 로 입력 ex) 2번서버일 경우 server.2가 0.0.0.0이 된다
# server.{1} -> {1} 은 myid이다 즉, server.myid 형식으로 되어있다.
server.1=0.0.0.0:2888:3888

server.2=3.34.18.190:2888:3888

server.3=13.209.146.71:2888:3888
```

위 파일은 현재 server.1의 설정 파일이다. server 이름은 server.{myid} 형식이다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
mkdir /tmp/zookeeper $ echo 1 > /tmp/zookeeper/myid (서버 1)
mkdir /tmp/zookeeper $ echo 2 > /tmp/zookeeper/myid (서버 2)
mkdir /tmp/zookeeper $ echo 3 > /tmp/zookeeper/myid (서버 3)
```

/tmp/zookeeper 명령어로 디렉토리를 생성하고 해당 파일의 myid 부분에 각 서버의 myid를 넣는다.

- **dataDir**

server.1,2,3의 숫자는 인스턴스 ID이다. ID는 dataDir=/tmp/zookeeper 폴더에 myid파일에 명시가 되어야 한다. /tmp/zookeeper 디렉토리가 없다면 생성하고 myid 파일을 생성하여 각각 서버의 고유 ID값을 부여해야 한다.

- **initLimit**

팔로워가 리더와 초기에 연결하는 시간에 대한 타임아웃을 설정한다.

- **syncLimit**

팔로워가 리더와 동기화 하는데에 대한 타임아웃. 즉 이 틱 시간안에 팔로워가 리더와 동기화가 되지 않는다면 제거 된다.



initLimit 과 syncLimit 은 default 기본값이 없기 때문에 반드시 설정해야 한다.

- **server.1,2,3**

각 서버의 IP주소와 포트를 설정한다. 여기서 중요한 점은 만약 1번 서버의 설정파일을 변경 중이라면 1번 서버, 즉 자기 자신에 대한 IP주소는 자신의 Public IP 주소가 아니라 0.0.0.0으로 설정해야 한다. zookeeper 설정 시 해당하는 노드가 localhost에 위치해 있는 경우, 예외상황 발생을 막기 위해 0.0.0.0으로 지정하는 것을 권장한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Broker 설정]

Kafka의 `config/server.properties` 파일은 하나의 Kafka를 실행하는데 쓰이는 설정 파일이다.

Zookeeper와 마찬가지로 여러 개의 설정 파일을 만들고 다중 실행을 할 수 있다.

설정 파일 `config/server.properties`에 3대 서버 각 환경에 맞는 정보를 입력해 준다.

```
#####
# Server Basics #####
broker.id=1

#####
# Socket Server Settings #####
listeners=PLAINTEXT://:9092
advertised.listeners=PLAINTEXT://IP:9092
#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL
# 보안 설정시 프로토콜 매핑 설정

num.network.threads=3      # 네트워크를 통한 처리를 할때 사용할 네트워크 스레드 개수 설정
# The number of threads that the server uses for processing requests, which may include disk I/O
num.io.threads=8           # 브로커 내부에서 사용할 스레드 개수 지정

#####
# Log Basics #####
log.dirs=/tmp/kafka-logs      # 통신을 통해 가져온 데이터를 파일로 저장할 디렉토리 위치. 티택토리가
                                # 생성되어 있지 않으면 오류가 발생하므로 미리 생성해야함.

num.partitions=1              # 파티션의 개수를 명시하지 않고 토픽을 생성할 때 기본 설정되는 파티션의
                                # 개수. 파티션의 개수가 많을수록 병렬처리 데이터 양이 늘어남

log.retention.hours=-1

log.segment.bytes=1073741824    # 브로커가 저장할 파일의 최대 크기 지정 데이터 양이 많아 이 크기를
                                # 초과해도 새로운 파일 생성

log.retention.check.interval.ms=300000    # 브로커가 저장한 파일을 삭제하기 위해 체크하는 간격을 지정
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
#####
# Zookeeper #####
zookeeper.connect=server1-ip:2181,server2-ip:2181,server3-ip:2181
zookeeper.connection.timeout.ms=18000      # 주키퍼 세션 타임아웃 시간 설정
```

- **broker.id** : 실행할 브로커의 번호를 적는다. 클러스터를 구축할 때 브로커들을 구분하기 위해 단 하나 뿐인 번호로 설정해야 한다.
- **listeners** : 카프카 브로커가 통신을 위해 열어 둘 인터페이스 IP, 포트t, 프로토콜을 설정할 수 있다. 따로 설정하지 않으면 ANY로 설정된다.
- **advertised.listeners** : 카프카 클라이언트 또는 카프카 커맨드 라인 터미널에서 접속할 때 사용하는 IP와 포트 정보를 설정한다.
- **log.retention.hours** : 브로커가 저장한 파일이 삭제되기까지 걸리는 시간을 설정한다. -1로 설정하면 영구보존된다.
- **zookeeper.connect** : 카프카 브로커와 연동할 주키퍼의 IP와 포트를 설정한다.

[Zookeeper 및 Kafka 서버 구동]

Kafka를 구동하기 위해 먼저 Zookeeper를 구동한 다음 이후 Kafka를 구동해야 한다.

```
$ bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
$ bin/kafka-server-start.sh -daemon config/server.properties
```

[Topic 생성]

카프카에서 사용할 Topic을 생성한다. Topic은 Naming Convention에 따라 생성해야 하며, 이번 프로젝트에서는 쇼핑몰에 따라 Topic을 따로 생성하여 관리한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Convention]

카프카에서는 Topic을 많이 생성해서 사용하는데 Naming Convention 없이 사용하게 된다면 나중에 복잡해질 수도 있기 때문에 Topic 생성 시 주의해야 한다.

카프카에서 Topic을 생성할 때 유효한 문자는 “영문, 숫자, 마침표, 쉼표, 언더바, 하이픈” 만 사용할 수 있다. 유의할 점은 마침표와 언더바는 충돌할 수 있기 때문에 둘 중 하나만 사용해야 한다.

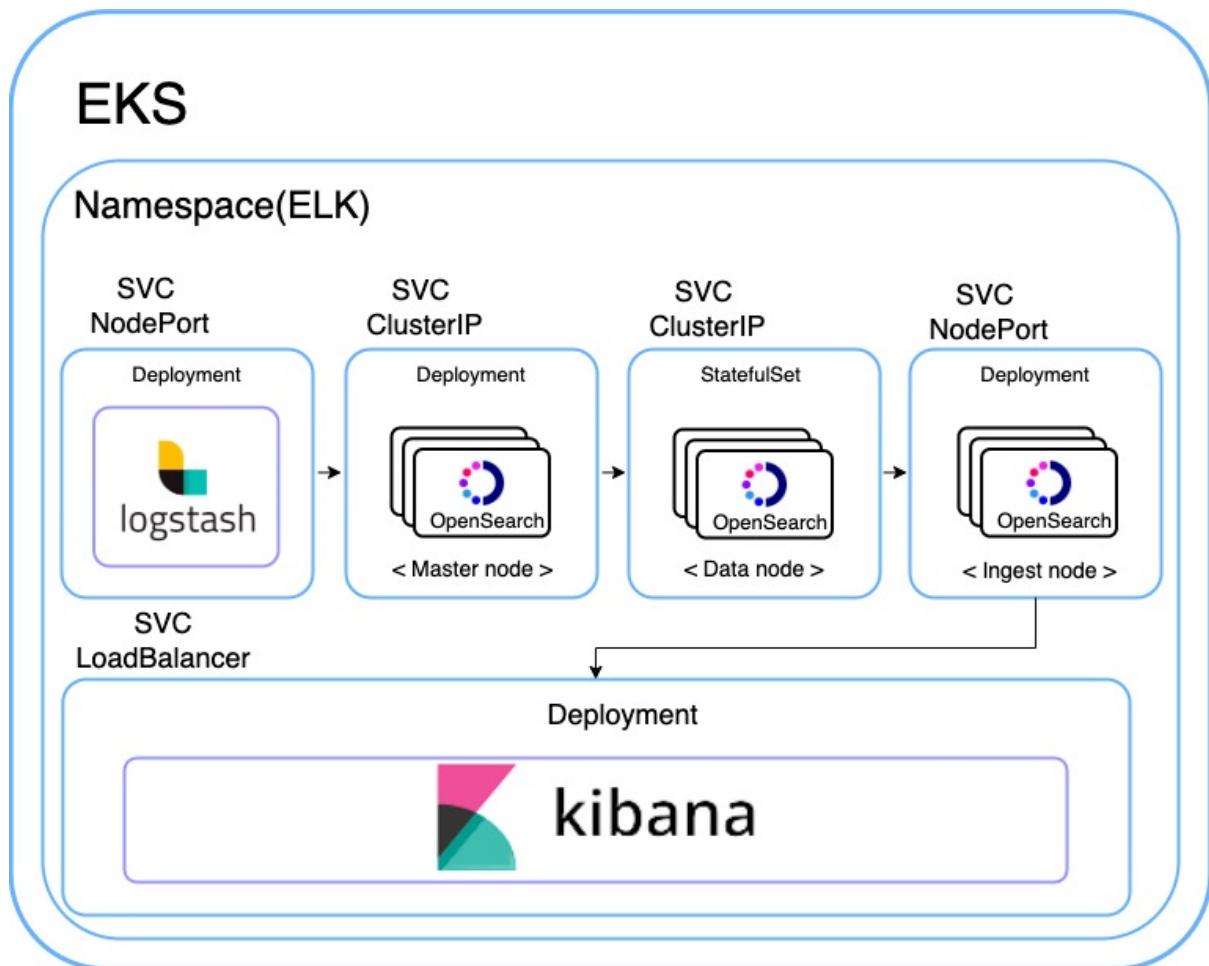
이번 프로젝트에서는 <department name>.<team name>.<dataset name> 규칙에 따라 Topic을 생성한다. Naming Convention에 따른 쇼핑몰 Topic 목록은 다음과 같다.

```
smartstore.goodnara.review (스마트 스토어 goodnara)
smartstore.drstyle.review (스마트 스토어 drstyle)
smartstore.thecheaper.review (스마트 스토어 thecheaper)
smartstore.180store.review (스마트 스토어 180store)
smartstore.cloony.review (스마트 스토어 cloony)
smartstore.theshopsw.review (스마트 스토어 theshopsw)
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

ELK Stack

[ELK Stack 아키텍처]



[그림 2-13] ELK 클러스터

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

ELK Stack Tree



OpenSearch

저장은 OpenSearch 가 담당한다. ELK Stack의 핵심은 OpenSearch 이며, 이 저장소를 기반으로 동작한다. OpenSearch 는 루씬 기반의 검색 엔진이다.

OpenSearch는 물리적으로 클러스터와 노드로 이루어져 있다. 노드는 크게 클러스터 상태 정보를 관리하는 마스터 노드, 색인된 데이터를 저장하고 있는 데이터 노드, 문서 변환 및 전처리를 담당하는 인제스트 노드로 구분된다.

현재 프로젝트의 OpenSearch 클러스터에는 마스터 노드 3개와 데이터 노드 3개, 인제스트 노드 3개로 총 9개의 노드로 구성되어 있다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Kibana

저장된 데이터를 분석하고 시각화한다. OpenSearch에 저장된 쇼핑몰 리뷰 데이터에 대한 Index 패턴을 생성하고 대시보드를 통해 시각화 한다. 로드밸런서 타입의 서비스를 생성하여 외부에서 접근이 가능했고, 로드밸런서 주소의 80번 포트로 접근하면 Kibana 클러스터에 접근할 수 있도록 구성했다.

Logstash

Deployment 방식으로 Logstash 파드를 배포하고 Nodeport 타입의 서비스로 배포하였다.

Kafka input plugin을 이용하여 브로커에 적재된 쇼핑몰 데이터를 OpenSearch의 데이터 노드에 적재한다.

Logstash 파이프 라인 중 Filter 플러그인을 사용하여 정제하도록 구성했다. Logstash를 통해 들어온 데이터는 Message라는 필드 안에 한 번에 들어오게 된다. “message” 필드만 저장해도 되지만 추후 원활한 시각화를 위해 로그를 구분하여 각각의 필드를 재생성하는 필터를 추가한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ OpenSearch

>> OpenSearch 사용 목적

OpenSearch는 데이터 수집, 검색, 시각화 및 분석을 쉽게 하는 커뮤니티 중심의 검색 및 분석 소프트웨어이다. 현재 프로젝트에서는 OpenSearch를 쇼핑몰 리뷰 데이터 및 리뷰 자연어 처리 결과 데이터를 저장하는 데이터 저장소 용도로 사용한다.

>> OpenSearch 인프라 생성

마스터 노드와 인제스트 노드는 Deployment 방식으로 배포하여 파드들을 관리하고, 데이터의 저장을 담당하는 데이터 노드는 StatefulSet 방식으로 생성하여 클러스터가 삭제되어도 데이터를 영구 보존할 수 있도록 설계했다.

마지막으로 Service는 마스터 노드와 데이터 노드는 ClusterIP 형태의 서비스로 구성하여 노드 간 통신은 가능하되 외부에서 접근할 수 없도록 설정하고, 인제스트 노드의 Service를 NodePort 형태로 배포하여 해당 노드를 통해서만 외부에서 접근 가능하도록 설정했다.

- Namespace 생성
- OpenSearch Master Node 생성
- OpenSearch Data Node 생성
- OpenSearch Ingest Node 생성

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>>> Namespace 생성

작업 디렉토리 : [Datapipeline_Project/Elasticsearch](#)

파일명 : namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: ELK
```

OpenSearch, Logstash, Kibana를 논리적으로 분리하여 관리하기 위해 ELK Namespace를 생성한다.

이후 생성되는 모든 리소스들은 Namespace 단위로 구분된다.

>>> Master Node 생성

Master Node는 인덱스의 메타 데이터, 샤프트의 위치와 같은 클러스터 상태 정보를 관리한다.

마스터 후보 노드를 하나만 놓게 되면 그 Master Node가 유실되었을 때 클러스터 전체가 작동을 정지 할 위험이 있으므로 최소 3개 이상의 툴수 개를 생성해야 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

작업 디렉토리 : [Datapipeline_Project/Elasticsearch/opensearch/master-node](#)

[Master Node Configmap 생성]

파일명 : [elasticsearch-master-configmap.yaml](#)

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: ELK
  name: elasticsearch-master-config
  labels:
    app: elasticsearch
    role: master
data:
  elasticsearch.yml: |-
    cluster.name: ${CLUSTER_NAME}
    node.name: ${NODE_NAME}
    discovery.seed_hosts: ${NODE_LIST}
    cluster.initial_master_nodes: ${MASTER_NODES}
    network.host: 0.0.0.0
    node:
      master: true
      data: false
      ingest: false
```

ELK Namespace 내에 OpenSearch 설정 값을 가진 Master Node의 [Configmap](#) 을 생성한다. 생성한다. 클러스터의 이름, 노드의 이름, 노드의 리스트를 통해 host를 찾도록 설정하고 `master: true` 옵션을 통해 마스터 노드로 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

data Field Parameter

Parameter	Description	Details
cluster.name	클러스터 이름	클러스터명이 동일해야 동일한 클러스터로 인식
node.name	노드 이름	-
discovery.seed_hosts	클러스터를 구성할 노드 리스트	-
cluster.initial_master_nodes	master 후보 리스트	-
network.host	노드의 IP	외부 접근 허용
node	노드의 기능	해당 노드는 master 노드로 사용

[표 2-22] OpenSearch Master Node Configmap data Field Parameter

[Master-Node Service 생성]

파일명 : `elasticsearch-master-service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  namespace: ELK
  name: elasticsearch-master
  labels:
    app: elasticsearch
    role: master
spec:
  ports:
  - port: 9300
    name: transport
  selector:
    app: elasticsearch
    role: master
```

ELK Namespace 내에 Master Node에 대한 서비스 오브젝트를 생성한다.

해당 서비스는 `app=elasticsearch, role=master` 레이블을 가진 파드가 TCP 9300 포트로 각 노드들과 통신할 수 있도록 설정했다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Master-Node Deployment 생성]

파일명 : `elasticsearch-master-deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: ELK
  name: elasticsearch-master
  labels:
    app: elasticsearch
    role: master
spec:
  replicas: 3
  selector:
    matchLabels:
      app: elasticsearch
      role: master
  template:
    metadata:
      labels:
        app: elasticsearch
        role: master
    spec:
      nodeSelector:
        Name: Master
      containers:
        - name: elasticsearch-master
          image: amazon/opendistro-for-elasticsearch:1.13.2
          imagePullPolicy: Always
          env:
            - name: CLUSTER_NAME
              value: elasticsearch
            - name: NODE_NAME
              value: elasticsearch-master
            - name: NODE_LIST
              value: elasticsearch-master,elasticsearch-data,elasticsearch-client
            - name: MASTER_NODES
              value: elasticsearch-master
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

- name: "ES_JAVA_OPTS"
  value: "-Xms1G -Xmx1G"
ports:
- containerPort: 9300
  name: transport
volumeMounts:
- name: config
  mountPath: /usr/share/elasticsearch/config/elasticsearch.yml
  readOnly: true
  subPath: elasticsearch.yml
- name: storage
  mountPath: /data
volumes:
- name: config
  configMap:
    name: elasticsearch-master-config
- name: "storage"
  emptyDir:
    medium: ""
initContainers:
- name: increase-vm-max-map
  image: busybox
  command: ["sysctl", "-w", "vm.max_map_count=262144"]
  securityContext:
    privileged: true

```

EKS의 Master Node 그룹에 배치하도록 **nodeSelector** 옵션을 추가하여 스케줄링 한다.

– OpenSearch Master Node container

OpenSearch 이미지를 사용해 **Master Node** 파드를 생성한다. Configmap에서 사용할 환경 변수를 지정해주고 미리 생성해 놓은 **Master Node** 서비스의 포트로 포트 포워딩한다. 앞서 생성한 Configmap이 적용될 수 있도록 Configmap 볼륨을 생성해 OpenSearch의 설정 파일에 마운트한다. 또한 데이터 저장을 위한 임시 볼륨을 생성하여 데이터를 저장할 디렉토리에 마운트한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Parameter	Description	Details
name	생성할 Deployment의 이름	elasticsearch-master
image	파드 생성에 사용하는 이미지	amazon/opendistro-for-elasticsearch: h:1.13.2
imagePullPolicy	파드를 생성할 때 업데이트 중에 지정된 이미지를 가져오는 방법	Always : 항상 저장소에서 이미지를 가져옴
env	설정 파일에 들어갈 환경 변수	ES_JAVA_OPTS=1G
ports	서비스 포트에 맵핑	앞서 생성했던 Master Node의 서비스 포트
volumeMounts	마운트할 볼륨 경로	elasticsearch.yml : OpenSearch 설정 파일

[표 2-23] OpenSearch Master Node Deployment Parameter

– OpenSearch Master Node initContainer

Linux 환경에서 OpenSearch가 정상적으로 작동하기 위해 설정해야 할 요소를 초기화한다. vm.max_map_count 변수 값을 조절하여 Linux kernel의 메모리 맵 영역의 최대 개수를 조작한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>>> Data Node 생성

CRUD, 검색 및 집계와 같은 데이터 관련 작업을 수행한다. 파드가 스케일링 되어도 데이터는 삭제되지 않고 유지되어야 하기 때문에 StatefulSet 오브젝트로 배포한다.

작업 디렉토리 : [Datapipeline_Project/Elasticsearch/opensearch/data-node](#)

[Data-Node Configmap 생성]

파일명 : [elasticsearch-data-configmap.yaml](#)

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: ELK
  name: elasticsearch-data-config
  labels:
    app: elasticsearch
    role: data
data:
  elasticsearch.yml: |-
    cluster.name: ${CLUSTER_NAME}
    node.name: ${NODE_NAME}
    discovery.seed_hosts: ${NODE_LIST}
    cluster.initial_master_nodes: ${MASTER_NODES}
    network.host: 0.0.0.0
    node:
      master: false
      data: true
      ingest: false
```

ELK Namespace 내에 Data Node의 Configmap 오브젝트를 생성한다. `data: true` 옵션을 통해 데이터를 저장하는 노드로 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Data-Node Service 생성]

파일명 : `elasticsearch-data-service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  namespace: ELK
  name: elasticsearch-data
  labels:
    app: elasticsearch
    role: data
spec:
  ports:
    - port: 9300
      name: transport
  selector:
    app: elasticsearch
    role: data
```

ELK Namespace 내에 Data Node에 대한 서비스 오브젝트를 생성한다.

해당 서비스는 `app=elasticsearch, role=data` 레이블을 가진 파드가 TCP 9300 포트로 다른 노드들과 통신할 수 있도록 하였다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Data-Node StatefulSet 생성]

파일명 : [elasticsearch-data-Statefulset.yaml](#)

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  namespace: ELK
  name: elasticsearch-data
  labels:
    app: elasticsearch
    role: data
spec:
  serviceName: "elasticsearch-data"
  selector:
    matchLabels:
      app: elasticsearch-data
      role: data
  replicas: 3
  template:
    metadata:
      labels:
        app: elasticsearch-data
        role: data
    spec:
      nodeSelector:
        Name: Data
      containers:
        - name: elasticsearch-data
          image: amazon/opendistro-for-elasticsearch:1.13.2
          env:
            - name: CLUSTER_NAME
              value: elasticsearch
            - name: NODE_NAME
              value: elasticsearch-data
            - name: NODE_LIST
              value: elasticsearch-master,elasticsearch-data,elasticsearch-client
            - name: MASTER_NODES
              value: elasticsearch-master

```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

- name: "ES_JAVA_OPTS"
  value: "-Xms1G -Xmx1G"
ports:
- containerPort: 9300
  name: transport
volumeMounts:
- name: config
  mountPath: /usr/share/elasticsearch/config/elasticsearch.yml
  readOnly: true
  subPath: elasticsearch.yml
- name: elasticsearch-data-persistent-storage
  mountPath: /data/db
  imagePullPolicy: Always
volumes:
- name: config
  configMap:
    name: elasticsearch-data-config
initContainers:
- name: increase-vm-max-map
  image: busybox
  command: ["sysctl", "-w", "vm.max_map_count=262144"]
  securityContext:
    privileged: true
volumeClaimTemplates:
- kind: PersistentVolumeClaim
  metadata:
    name: elasticsearch-data-persistent-storage
  annotations:
    volume.beta.kubernetes.io/storage-class: "gp2"
spec:
  accessModes: [ "ReadWriteOnce" ]
  storageClassName: standard
  resources:
    requests:
      storage: 10Gi

```

ELK Namespace 내에 Data Node의 StatefulSet 오브젝트를 생성하여 3개의 Data Node 파드를 배포한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

– OpenSearch Data Node container

OpenSearch 이미지를 사용해 Data Node 파드를 생성한다. Configmap에서 사용할 환경 변수를 지정해주고 미리 생성 해 놓은 Master Node 서비스의 포트로 포트 포워딩한다. 앞서 생성한 Config map이 적용될 수 있도록 Configmap 볼륨을 생성해 OpenSearch의 설정 파일에 마운트한다. 또한 데이터 저장 및 상태 유지를 위해 PVC 템플릿을 이용해서 `elasticsearch-data-persistent-storage`라는 이름의 PVC를 생성한다.

– OpenSearch Data Node initContainer

Linux 환경에서 OpenSearch가 정상적으로 작동하기 위해 설정해야 할 요소를 초기화한다. `vm.max_map_count` 변수 값을 조절하여 Linux kernel의 메모리 맵* 영역의 최대 개수를 조작한다.

– OpenSearch Data Node volumeClaimTemplates

데이터 저장 및 상태유지를 위해 PersistentVolumeClaim을 생성하여 10 기가바이트 용량의 `gp2` 타입 PersistentVolume을 생성한다. 읽기·쓰기 모드로 볼륨을 마운트할 수 있도록 설정하고 스토리지 클래스의 이름을 `standard`로 지정하여 PersistentVolumeClaim을 PersistentVolume에 바인딩한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>>> Ingest Node 생성

문서 변환 및 전처리를 담당하며 인덱싱 전에 문서를 변환하고 보강하기 위해 INGEST 파이프 라인을 문서에 적용한다.

작업 디렉토리 : [Datapipeline_Project/Elasticsearch/opensearch/client-node](#)

[Client-Node Configmap 생성]

파일명 : [elasticsearch-client-configmap.yaml](#)

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: ELK
  name: elasticsearch-client-config
  labels:
    app: elasticsearch
    role: client
data:
  elasticsearch.yml: |-
    cluster.name: ${CLUSTER_NAME}
    node.name: ${NODE_NAME}
    discovery.seed_hosts: ${NODE_LIST}
    cluster.initial_master_nodes: ${MASTER_NODES}
    network.host: 0.0.0.0
    node:
      master: false
      data: false
      ingest: true
```

ELK Namespace 내에 Ingest Node의 Configmap 오브젝트를 생성한다. `ingest: true` 옵션을 사용해 인제스트 노드로 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Client-Node Service 생성]

파일명 : `elasticsearch-client-configmap.yaml`

```
apiVersion: v1
kind: Service
metadata:
  namespace: ELK
  name: elasticsearch-client
  labels:
    app: elasticsearch
    role: client
spec:
  ports:
  - port: 9300
    name: transport
  selector:
    app: elasticsearch
    role: client
```

ELK Namespace 내에 Ingest Node에 대한 서비스 오브젝트를 생성한다. 해당 서비스는 `app=elasticsearch, role=client` 레이블을 가진 파드가 TCP 9300 포트를 통해 다른 노드와 통신할 수 있도록 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

파일명 : [elasticsearch-client-http.yaml](#)

```
apiVersion: v1
kind: Service
metadata:
  namespace: ELK
  name: elasticsearch-client-http
  labels:
    app: elasticsearch
    role: client
spec:
  type: NodePort
  ports:
  - port: 9200
    name: client
    targetPort: 9200
    nodePort: 30000
  selector:
    app: elasticsearch
    role: client
```

ELK Namespace 내에 Ingest Node에 대한 NodePort 서비스 오브젝트를 생성한다. `app=elasticsearch, role=client` 레이블을 가진 파드의 서비스로 동작한다. 해당 서비스의 노드 포트 30000은 TCP 9200 포트로 매피된다. 위의 서비스는 HTTP 전송을 관리하기 위해 사용된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Client-Node Deployment 생성]

파일명 : `elasticsearch-client-deployment.yaml`

```

apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: ELK
  name: elasticsearch-client
  labels:
    app: elasticsearch
    role: client
spec:
  replicas: 3
  selector:
    matchLabels:
      app: elasticsearch
      role: client
  template:
    metadata:
      labels:
        app: elasticsearch
        role: client
    spec:
      nodeSelector:
        Name: Client
      containers:
        - name: elasticsearch-client
          image: amazon/opendistro-for-elasticsearch:1.13.2
          env:
            - name: CLUSTER_NAME
              value: elasticsearch
            - name: NODE_NAME
              value: elasticsearch-client
            - name: NODE_LIST
              value: elasticsearch-master,elasticsearch-data,elasticsearch-client
            - name: MASTER_NODES
              value: elasticsearch-master
            - name: "ES_JAVA_OPTS"
              value: "-Xms4G -Xmx4G"

```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

ports:
  - containerPort: 9200
    name: client
  - containerPort: 9300
    name: transport
volumeMounts:
  - name: config
    mountPath: /usr/share/elasticsearch/config/elasticsearch.yml
    readOnly: true
    subPath: elasticsearch.yml
  - name: storage
    mountPath: /data
volumes:
  - name: config
    configMap:
      name: elasticsearch-client-config
  - name: "storage"
    emptyDir:
      medium: ""
initContainers:
  - name: increase-vm-max-map
    image: busybox
    command: ["sysctl", "-w", "vm.max_map_count=262144"]
    securityContext:
      privileged: true

```

ELK Namespace 내에 Ingest Node Deployment 오브젝트를 생성하여 3개의 Ingest Node 파드를 생성한다.

– OpenSearch Ingest Node container

OpenSearch 이미지를 사용해 파드를 생성한다. Configmap에서 사용할 환경 변수를 지정해주고 미리 생성해 놓은 Client Node 서비스의 포트로 포트 포워딩한다. 앞서 생성한 Configmap이 적용될 수 있도록 Configmap 볼륨을 생성해 OpenSearch의 설정 파일에 마운트한다. 또한 데이터 저장을 위한 임시 볼륨을 생성하여 데이터를 저장할 디렉토리에 마운트한다.

– OpenSearch Ingest Node initContainer

Linux 환경에서 OpenSearch가 정상적으로 작동하기 위해 설정해야 할 요소를 초기화한다. vm.max_map_count 변수 값을 조절하여 Linux kernel의 메모리 맵* 영역의 최대 개수를 조작한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ Logstash

>> Logstash 사용 목적

Logstash는 모든 형태, 크기, 소스의 데이터 수집하고 형식이나 복잡성에 관계없이 다음과 같이 데이터를 동적으로 변환하고 준비하여 원하는 곳으로 데이터를 라우팅해준다. 현재 프로젝트에서는 Logstash를 Kafka에서 쇼핑몰 리뷰 데이터를 수집하여 필드를 나눈 다음 OpenSearch로 전송하는 용도로 사용한다.

>> Logstash 인프라 생성

작업 디렉토리 : [Datapipeline_Project/Elasticsearch/logstash](#)

[Logstash Configmap 생성]

Configmap은 Logstash를 위한 설정 파일이다. Logstash에서는 이 파일을 사용하여 input, output, filter를 정의한다.

파일명 : [logstash-configmap.yaml](#)

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: logstash-config
  namespace: ELK
```

ELK Namespace 내에 Logstash 설정 값을 가진 ConfigMap 오브젝트를 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

data:
  # logstash conf
  logstash.yml: |
    http.host: "0.0.0.0"
    path.config: /usr/share/logstash/pipeline
    config.reload.automatic: true
  
```

`logstash.conf`는 로그 데이터의 파이프 라인 설정 파일이다. 외부 접속을 허용하고, Logstash 파이프라인 구성 파일의 경로를 지정한다. `config.reload.automatic` 옵션을 사용하여 설정 파일 변경 시 Logstash를 재시작 할 필요 없이 수정된 `conf` 파일을 `reload` 하여 적용한다.

[[Input]]

```

logstash.conf: |
  input {
    kafka {
      bootstrap_servers => ""
      topics =>
      ["smartstore.goodnara.review","smartstore.drstyle.review","smartstore.thecheaper.review","smartstore.180store.review","smartstore.cloony.review","smartstore.theshopsw.review"]
      consumer_threads => 3
      isolation_level => "read_committed"
      value_deserializer_class => "org.apache.kafka.common.serialization.StringDeserializer"
      auto_offset_reset => "earliest"
      group_id => "smartstore"
    }
  }
  
```

Logstash Input을 Kafka 브로커로 지정하여 Kafka에 큐잉되어있는 데이터를 수집해온다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Logstash Input Field

Field	Description	Detail
bootstrap_servers	브로커 주소 (IP:9092)	-
topics	브로커 Topic명 리스트	쇼핑몰 별 Topic 리스트를 표시한다
consumer_threads	컨슈머 쓰레드	파티션 개수와 동일하다
isolation_level	트랜잭션 격리수준	풀링 메시지는 커밋된 트랜잭션 메시지만 반환한다
value_deserializer_class	레코드값 역직렬화에 사용되는 JAVA Class (String)	-
group_id	컨슈머 그룹이름을 지정	-
auto_offset_reset	Consumer group으로 처음 브로커에 진입했을때, 데이터를 가지고오는 시작점을 지정	default 값은 latest, earliest는 초기에 consumer group이 설정되어 있지 않은 경우에만 적용되므로, 추후 컨슈머를 재시작 하더라도 데이터 중복 이슈를 해결할 수 있다.

[표 2-24] Logstash Input Field Parameter

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

✚ auto_offset_reset 추가설명

상황에 따라 데이터 누락이 발생할 수 있기 때문에 `earliest`로 설정한다. `earliest` 설정은 `broker`에 컨슈머 그룹이 설정되어 있지 않은 경우에만 적용되므로, 새로 컨슈머 그룹이 생성되었다면 데이터의 처음부터 읽어 오게 된다.

만약 기존 컨슈머 그룹이 지정된 상태에서 컨슈머를 재시작했다면, 컨슈머 그룹에 이미 오프셋 정보 및 메타 데이터가 남아있기 때문에 `earliest` 설정은 적용되지 않고, 오프셋 정보를 시작점으로 데이터를 읽어오기 때문에 데이터 중복 이슈를 예방할 수 있다.

[[Filter]]

`Logstash` 필터는 데이터가 소스에서 저장소로 이동하는 과정에서 각 이벤트를 구문 분석하고 명명된 필드를 식별하여 구조를 구축하며, 이를 공통 형식으로 변환 및 통합한다. 현재 프로젝트에서는 `Logstash filter`를 사용하여 `timestamp` 필드에 작성일자 데이터를 정제하여 인덱싱하고 리뷰 데이터의 특수문자를 제거한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Timestamp Field

```

filter {
    # ----- UTC(default) Timestamp -> KST Timestamp로 변환하기 -----
    mutate {
        add_field => {
            "timestamp" => ""    # timestamp 필드 생성(새로 생성된 필드의 기본 데이터 타입은
String이다.)
        }
    }
    # ruby 코드로 "@timestamp" 필드의 UTC 기준 현재 시간에 9시간을 더한 값을 timestamp 필드에
저장한다.
    ruby {
        code => "event.set('timestamp',
event.get('@timestamp').time.localtime('+09:00').strftime('%Y-%m-%d %H:%M:%S'))"
    }
    # timestamp 필드의 데이터는 String으로 날짜 형식으로 지정함(ISO8601)
    # ISO8601 = 2019-01-26T17:00:00Z
    date {
        match => ["timestamp", "ISO8601", "YYYY-MM-dd HH:mm:ss"]
        target => "timestamp"    # date 필터가 적용될 필드 지정
    }
    # timestamp를 파싱하여 yy mm dd만 추출해 yymmdd 필드로 저장한다.
    grok {
        match => {
            "timestamp" => "\d\d{INT:yy}-\{MONTHNUM:mm\}-\{MONTHDAY:dd\}\{GREEDYDATA\}"
        }
        # yymmdd를 메타필드로 저장한다.
        add_field => {
            "[@metadata][yymmdd]" => "%{yy}%{mm}%{dd}"
        }
    }
}

```

📌 주의

OpenSearch의 index를 일별로 생성하고 yyMMdd를 postfix로 설정하려고 한다. ex) index-220803

하지만 Logstash는 기본적으로 @timestamp를 UTC+0 표준시로 나타내기 때문에 한국 시간과는 약 9시간의 차이가 발생하게 된다. 따라서 UTC+0 를 KST(UTC+9)로 바꿔서 사용해야 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

- Filtering 과정
- `mutate filter plugin`으로 String 타입의 timestamp 필드 생성한다.
- `ruby filter plugin`으로 "@timestamp" 필드의 UTC 기준 현재 시간에 9시간을 더한 값을 timestamp 필드에 저장한다.
- `date filter plugin`으로 String 타입인 timestamp 필드의 데이터를 날짜 형식으로 변환한다.
(ISO8601 = 2019-01-26T17:00:00.000Z)
- `grok filter plugin`으로 date 필터가 적용될 필드 지정 timestamp를 파싱하여 yy mm dd만 추출해 yymmdd 필드로 저장하고 yymmdd를 메타필드로 저장한다.

message Field

```
# ----- message 필드 정제 -----
# 절규표현식으로 특수문자 1차 제거한다.
mutate {
  gsub => ["message", "[\\/{})", ""]
}
# comma를 기준으로 메세지 필드를 나눈 후, colon을 기준으로 Key, value 형식으로 필드를 생성한다.
kv {
  field_split => ","
  value_split => ":" 
}
# 사용하지 않을 필드들을 제거하고 필드의 이름을 재설정한다.
mutate {
  remove_field => [ "port", "@version", "host", "message", "@timestamp", "yy", "mm", "dd" ]
  rename => { "comment" => "comment" }
  rename => { "date" => "date" }
  rename => { "star" => "star" }
}
# star 필드를 String에서 Integer 타입으로 변환한다.
mutate {
  convert => {
    "star" => "integer"
  }
}
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Filtering 과정

- `mutate filter plugin`을 사용하여 정규 표현식으로 특수 문자를 1차로 제거한다.
- `kv filter plugin`을 통해 comma를 기준으로 메시지 필드를 나눈 후, colon을 기준으로 Key, value 형식으로 필드를 생성한다
- 분석에 필요하지 않은 필드는 제거하고 필드의 이름을 재설정한다.
- 분석에 필요한 star(평점) 필드의 데이터 타입을 Integer로 변환한다.

[[output]]

```
output {
    stdout { codec => rubydebug }
    # 스마트 스토어 모자 Topic
    if [topic] =~ "smartstore.goodnara.review" {
        elasticsearch {
            hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
            index => "smartstore.goodnara.review-%{[@metadata][yymmdd]}"
            codec => "json"
            timeout => 120
        }
    } # if end
    else if [topic] =~ "smartstore.drstyle.review" {
        elasticsearch {
            hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
            index => "smartstore.drstyle.review-%{[@metadata][yymmdd]}"
            codec => "json"
            timeout => 120
        }
    } # if end
    else if [topic] =~ "smartstore.thecheaper.review" {
        elasticsearch {
            hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
            index => "smartstore.thecheaper.review-%{[@metadata][yymmdd]}"
            codec => "json"
            timeout => 120
        }
    }
}
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

# if end
    # 스마트 스토어 티셔츠 Topic
else if [topic] =~ "smartstore.180store.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
        index => "smartstore.180store.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
else if [topic] =~ "smartstore.cloony.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
        index => "smartstore.cloony.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
else if [topic] =~ "smartstore.theshopsw.review" {
    elasticsearch {
        hosts => "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
        index => "smartstore.theshopsw.review-%{[@metadata][yymmdd]}"
        codec => "json"
        timeout => 120
    }
} # if end
} # output end

```

OpenSearch에 Topic에 따라 인덱스를 나누어 저장하도록 설정했다. 앞서 생성했던 메타데이터 필드의 KST yymmdd 를 활용해 Topic이름-yymmdd 형식으로 인덱스를 저장하여 각 Topic에 대한 인덱스를 일자 별로 나누어 관리할 수 있도록 설정했다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

스마트 스토어 모자 Topic

Topic	Hosts	Index
smartstore.goodnara.review	http://elasticsearch-client-http.E LK.svc.cluster.local:9200	smartstore.goodnara.review-%{ [@metadata][yymmdd]}
smartstore.drstyle.review		smartstore.drstyle.review-%{[@metadata][yymmdd]}
smartstore.thecheaper.review		smartstore.thecheaper.review- %{[@metadata][yymmdd]}

[표 2-25] 스마트 스토어 모자 Topic

스마트 스토어 티셔츠 Topic

Topic	Hosts	Index
smartstore.180store.review	http://elasticsearch-client-http.E LK.svc.cluster.local:9200	smartstore.180store.review-%{[@metadata][yymmdd]}
smartstore.cloony.review		smartstore.cloony.review-%{[@ metadata][yymmdd]}
smartstore.theshopsw.review		smartstore.theshopsw.review- %{[@metadata][yymmdd]}

[표 2-26] 스마트 스토어 티셔츠 Topic

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Logstash Service 생성]

파일명 : `logstash-svc-nodeport.yaml`

```
apiVersion: v1
kind: Service
metadata:
  name: logstash
  namespace: ELK
spec:
  type: NodePort
  ports:
    - port: 5000
      targetPort: 5000
  selector:
    app: logstash
```

ELK Namespace 내에 Logstash에 대한 NodePort 서비스 오브젝트를 생성한다. `app=elasticsearch` 레이블을 가진 파드의 서비스로 동작한다. Logstash 파드의 TCP 5000 포트로 요청이 전송되도록 설정했다.

[Logstash Service 생성]

Logstash 파드를 배포한다.

파일명 : `logstash-deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: logstash
  namespace: ELK
spec:
  replicas: 1
  selector:
    matchLabels:
      app: logstash
  template:
    metadata:
      labels:
        app: logstash
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

spec:
  nodeSelector:
    Name: Logstash
  volumes:
    - name: logstash-config-volume
      configMap:
        name: logstash-config
        items:
          - key: logstash.yml
            path: logstash.yml
    - name: logstash-pipeline-volume
      configMap:
        name: logstash-config
        items:
          - key: logstash.conf
            path: logstash.conf
  containers:
    - name: logstash
      image: docker.elastic.co/logstash/logstash:7.10.2
      resources:
        limits:
          cpu: 2000m
          memory: 2Gi
        requests:
          cpu: 1500m
          memory: 1.5G
      env:
        - name: LS_JAVA_OPTS
          value: '-Xmx1G -Xms1G'
      ports:
        - name: tcp
          containerPort: 5000
          protocol: TCP
      volumeMounts:
        - name: logstash-config-volume
          mountPath: /usr/share/logstash/config
        - name: logstash-pipeline-volume
          mountPath: /usr/share/logstash/pipeline

```

ELK Namespace 내에 Logstash Deployment 오브젝트를 생성하여 1개의 파드를 생성한다.

`nodeSelector` 옵션으로 `Logstash` 노드 그룹으로 스케줄링 되도록 설정한다. 이후 파드의 리소스를 설정하고 `Logstash` 컨테이너 포트를 5000번으로 변경한다. 마지막으로 앞서 설정했던 파이프라인 파일을 `volumeMounts` 옵션을 통해 마운트한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Logstash Volume

logstash.conf 파일은 volume에서 마운팅 시킨다.

Logstash Container

Docker repository에서 logstash 이미지를 다운로드하여 파드를 생성한다. 파드의 리소스를 cpu 1200~2000 메가바이트, memory 1.5~2기가바이트 부여한다.



주의

- ConfigMap에서 지정한 이름을 정확하게 매칭
- 정확한 볼륨 경로 지정

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ Kibana

>> Kibana 사용 목적

Kibana는 OpenSearch에서 색인 된 데이터를 검색하고 시각화한다. 현재 프로젝트에서는 OpenSearch에 저장되어 있는 데이터를 여러 차트로 시각화하여 사용자에게 감성 분석이 완료 된 키워드를 확인할 수 있도록 한다.

>> Kibana 인프라 생성

- Kibana 생성
- 인덱스 패턴 추가
- Slack 알람 설정
- 분석 데이터 인덱스 생성
- 대시보드 생성

작업 디렉토리 : [Datapipeline_Project/Elasticsearch/kibana](#)

[Kibana Configmap 생성]

파일명 : [kibana-configmap.yaml](#)

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: ELK
  name: kibana-config
  labels:
    app: kibana
data:
  kibana.yml: |-
    server.host: 0.0.0.0
    elasticsearch:
      hosts: ${ELASTICSEARCH_HOSTS}
```

ELK Namespace 내에 Kibana Configmap 오브젝트를 생성한다.

위 config 파일으로 Kibana가 외부와 소통할 수 있도록 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Kibana Service 생성]

파일명 : `kibana-service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  namespace: ELK
  name: kibana
  labels:
    app: kibana
spec:
  type: LoadBalancer
  ports:
  - port: 80
    name: webinterface
    targetPort: 5601
  selector:
    app: kibana
```

ELK Namespace 내에 `app=kibana` label을 가진 리소스를 대상으로 하는 로드밸런서 타입 서비스 오브젝트를 생성한다. 해당 서비스의 노드 포트 80은 TCP 5601 포트에 매피된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[Kibana Deployment 생성]

파일명 : `kibana-deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: ELK
  name: kibana
  labels:
    app: kibana
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kibana
  template:
    metadata:
      labels:
        app: kibana
    spec:
      nodeSelector:
        Name: Kibana
      containers:
        - name: kibana
          image: opendistro-for-elasticsearch-kibana:1.13.2
          command:
          ports:
            - containerPort: 5601
              name: webinterface
          env:
            - name: ELASTICSEARCH_HOSTS
              value: "http://elasticsearch-client-http.ELK.svc.cluster.local:9200"
          volumeMounts:
            - name: config
              mountPath: /usr/share/kibana/config/kibana.yml
              readOnly: true
              subPath: kibana.yml
      volumes:
        - name: config
      configMap:
        name: kibana-config
```

ELK Namespace 내에 Kibana Deployment 오브젝트를 생성하여 1개의 파드를 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

– Kibana Containers

Kibana 이미지를 사용해 파드를 생성한다. Configmap에서 사용할 환경 변수를 지정해주고 미리 생성해놓은 로드밸런서 서비스의 포트로 포트 포워딩한다. 앞서 생성한 Configmap이 적용될 수 있도록 Configmap 볼륨을 생성해 OpenSearch의 설정 파일에 마운트한다. config 파일에는 환경변수를 설정하여 OpenSearch를 호스팅하도록 설정한다.

Parameter

- name: Kibana 파드 이름
- image : Kibana 파드 이미지
- env : Kibana config 파일에 명시되어있는 환경 변수
- ports : 매팅 할 포트 및 이름
- volumeMounts : config 파일 마운트

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Jupyter Notebook

>> Jupyter Notebook 사용 목적

오픈소스 소프트웨어로 웹 실행 가능한 소스 코드 및 공식, 시각화 및 설명 텍스트를 포함하는 문서들을 생성하고 공유하는 기능을 제공한다. 데이터 정제, 변환, 산술 시뮬레이션, 수치모델링, 데이터 가시화, 기계학습 등에 사용될 수 있다. Jupyter Notebook은 Python, R, Julia 및 Scala를 포함하여 40 개가 넘는 프로그래밍 언어를 지원한다.

>> Jupyter Notebook 인프라 구현

[Ubuntu 그래픽 드라이버 설치]

텐서플로우에서 GPU를 사용하기 위한 CUDA 버전은 다음과 같다.

- NVIDIA GPU 드라이버 - CUDA 11.0에는 450.X 이상 필요
- CUDA Toolkit - Tensorflow는 CUDA 11을 지원
- CUPTI는 CUDA Toolkit과 함께 제공
- cuDNN SDK 8.0.4
- (선택사항) TensorRT 6.0 - 일부 모델에서 추론 처리량과 지연 시간을 향상

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[NVIDIA 그래픽 카드 드라이버 설치]

아래 명령으로 현재 사용중인 그래픽 카드에 설치할 수 있는 드라이버를 확인한다.

```
$ ubuntu-drivers devices

== /sys/devices/pci0000:00/0000:00:1e.0 ==
modalias : pci:v000010DEd00001EB8sv000010DEsd000012A2bc03sc02i00
vendor   : NVIDIA Corporation
model    : TU104GL [Tesla T4]
manual_install: True
driver   : nvidia-driver-460-server - distro non-free
driver   : nvidia-driver-470-server - distro non-free
...
```

확인 된 드라이버 중 하나를 설치한다.

```
$ sudo apt install nvidia-driver-460
```

재부팅 후 터미널에서 nvidia-smi 명령을 실행하여 설치한 드라이버 버전이 맞는지 확인한다.

```
$ nvidia-smi
```

```
Thu Aug  4 00:58:02 2022
+-----+
| NVIDIA-SMI 460.23.03   Driver Version: 460.32.03   CUDA Version: 11.2 |
+-----+
...
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[CUDA 11.2 설치]

– 기존 CUDA 삭제

데비안 패키지로 설치한게 아니라면 간단히 기존에 설치된 CUDA를 제거할 수 있다.

```
$ sudo rm -rf /usr/local/cuda*
```

~/.bashrc나 /etc/profile에 추가되어있는 CUDA 관련 또한 제거한다.

```
...
export PATH=$PATH:/usr/local/cuda-11.0/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-11.0/lib64
export CUDADIR=/usr/local/cuda-11.0
...
```

– CUDA 11.2 설치

아래 링크에 접속하여 자신의 조건에 해당하는 CUDA Toolkit 11.2.2 선택

링크 : <https://developer.nvidia.com/cuda-toolkit-archive>

선택 후 Base Installer 아래에 보이는 명령대로 설치 진행

```
$ wget
https://developer.download.nvidia.com/compute/cuda/11.2.2/local_installers/cuda_11.2.2_460.32.03_linux.run
$ sudo sh cuda_11.2.2_460.32.03_linux.run
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

continue를 선택한다.

```
Existing package manager installation of the driver found. It is strongly
recommended that you remove this before continuing.
Abort
Continue
```

Up/Down: Move | 'Enter': Select

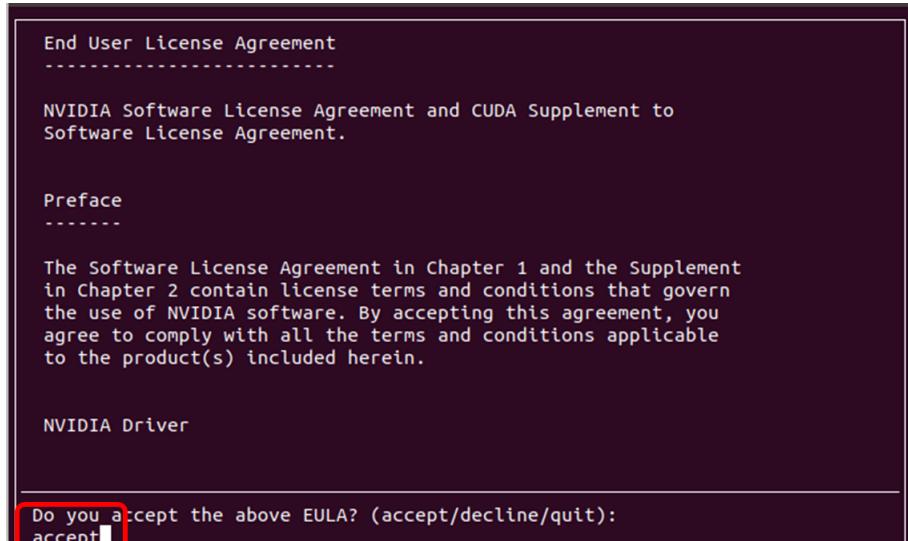
[그림 2-14] Jupyter Notebook CUDA 설정 1

만약 gcc version을 확인하라는 에러가 나온다면 개발을 위한 필수 프로그램을 설치한다.

```
$ sudo apt install build-essential
```

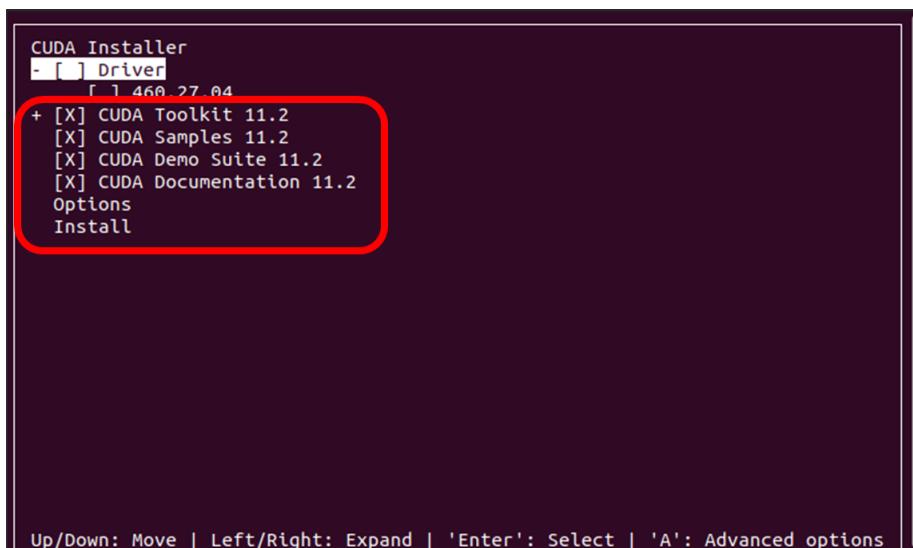
encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

“accept”를 입력 후 엔터 입력한다.



[그림 2-15] Jupyter Notebook CUDA 설정 2

Driver 항목 제외 후 설치한다.



[그림 2-16] Jupyter Notebook CUDA 설정 3

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

\$ source /etc/profile 명령을 사용하여 CUDA Toolkit 관련 설정을 환경 변수에 추가하고 적용한다.

```
$ sudo sh -c "echo 'export PATH=$PATH:/usr/local/cuda-11.2/bin' >> /etc/profile"
$ sudo sh -c "echo 'export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-11.2/lib64' >>
/etc/profile"
$ sudo sh -c "echo 'export CUDADIR=/usr/local/cuda-11.2' >> /etc/profile"
```

설치 완료 확인한다.

```
$ nvcc -V
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Mon_Nov_30_19:08:53_PST_2020
Cuda compilation tools, release 11.2, V11.2.67
Build cuda_11.2.r11.2/compiler.29373293_0
```

[cuDNN 8.1.0 설치]

아래 링크로 접속한다.

링크 : <https://developer.nvidia.com/cudnn>

☞ 주의

cuDNN을 다운받기 위해선 회원가입을 해야한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

회원가입과 로그인을 완료했다면 Archived cuDNN Releases를 클릭한다.

cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

I Agree To the Terms of the cuDNN Software License Agreement

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

[Download cuDNN v8.4.1 \[May 27th, 2022\], for CUDA 11.x](#)

[Download cuDNN v8.4.1 \[May 27th, 2022\], for CUDA 10.2](#)

[Archived cuDNN Releases](#)

[그림 2-17] Jupyter Notebook cuDNN 설정 1

목록에서 Download cuDNN v8.1.0 를 클릭한다.



[그림 2-18] Jupyter Notebook cuDNN 설정 2

cuDNN Library for Linux (x86_64) 링크를 복사하여 wget으로 다운 받는다.

```
$ wget https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.1.0.77/11.2_20210127/cudnn-11.2-linux-x64-v8.1.0.77.tgz
$ tar xvzf cudnn-11.2-linux-x64-v8.1.0.77.tgz
$ sudo cp cuda/include/cudnn* /usr/local/cuda/include
$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

링크 파일 생성

```
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_adv_train.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_adv_train.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_ops_infer.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_ops_infer.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_cnn_train.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_cnn_train.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_adv_infer.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_adv_infer.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_ops_train.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_ops_train.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_cnn_infer.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn_cnn_infer.so.8
$ sudo ln -sf /usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn.so.8.1.0
/usr/local/cuda-11.2/targets/x86_64-linux/lib/libcudnn.so.8
```

[Ubuntu 아나콘다]

패키지 관리와 디플로이를 단순화 할 목적으로 과학 계산을 위한 파이썬과 R 프로그래밍 언어의 오픈소스 배포판이다. 패키지 버전들은 패키지 관리 시스템 **conda**를 통해 관리된다.

링크 : <https://www.anaconda.com/>

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> 설치

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh
$ bash Anaconda3-2022.05-Linux-x86_64.sh
```

Anaconda 가상환경(base) 활성 / 비활성

- 활성 - conda activate base
- 비활성 - conda deactivate

pip 명령어를 사용하여 Jupyter Notebook 설치한다.

```
$ pip install notebook
```

Jupyter Notebook을 위한 Config 파일을 생성한다.

```
$ jupyter notebook --generate-config
```

서버 비밀번호 생성(Python)한다.

– 생성할 비밀번호 입력 후 생성 된 해시 복사

```
from notebook.auth import passwd
passwd()
```

생성 된 경로의 파일을 실행한다.

```
$ vi ~/.jupyter/jupyter_notebook_config.py
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

주요 설정

```
#비밀번호 설정
c.NotebookApp.password # 주석 해제 후 복사해둔 해쉬 붙여넣기

#기본 작업 경로 설정
c.NotebookApp.notebook_dir # 주석해제 후 실행하고자 하는 dir 지정

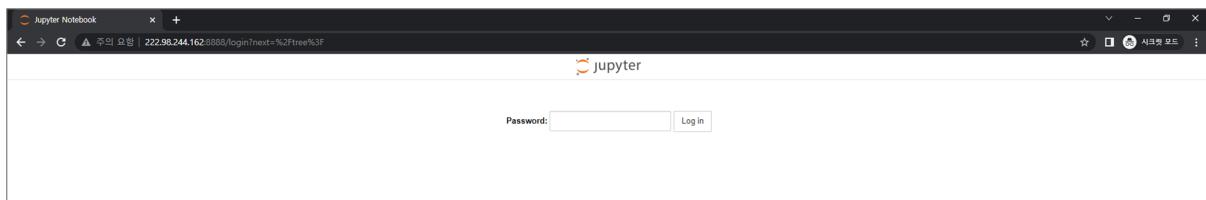
#외부 접속 허용
c.NotebookApp.allow_origin = '*'

#서버를 띄울 아이피 설정
c.NotebookApp.ip = '0.0.0.0'

#주피터 서버 실행 시 브라우저 실행 X
c.NotebookApp.open_browser = False
```

- 비밀번호 설정
- 기본 작업 경로 설정
- 외부 접속 허용 설정
- 서버를 띄울 IP 설정
- 주피터 서버 실행 시 브라우저 실행되지 않게 설정

Jupyter Notebook 접속 확인



[그림 2-19] Jupyter Notebook 접속 확인

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.4.2. UseCase 기능 구현

[SFR-001] 리뷰 크롤링을 통한 쇼핑몰 데이터 추출

I. 크롤러 배치 작업

매일 0시에 배치 작업 진행하며 비용 절감을 위해 클러스터를 오토스케일링한다.

OOMKILLED(Out Of Memory Killed) 오류 지속적 발생하여 t3.medium 클러스터를 6개로 오토 스케일링하도록 설정하였다.

(기존 1개의 클러스터에서, 배치 작업 진행시 6개의 클러스터로 구성)

📌 크롤러 배포 전 필수 수행 요소

Cluster Autoscaler 시 노드 감소 예외 옵션으로 노드 사용률이 저조하여도 해당 노드는 축소되지 않도록 한다.

EKS를 패치하여 cluster-autoscaler.kubernetes.io/safe-to-evict을 Cluster Autoscaler 파드에 추가

```
$ kubectl patch deployment cluster-autoscaler \
  -n kube-system \
  -p
'{"spec": {"template": {"metadata": {"annotations": {"cluster-autoscaler.kubernetes.io/safe-to-evict": "false" }}}}}'
```

Cluster Autoscaler 배포를 편집

```
$ kubectl -n kube-system edit deployment.apps/cluster-autoscaler
```

cluster-autoscaler 컨테이너 명령을 편집하여 다음 옵션을 추가

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
...
spec:
  containers:
    - command:
        - ./cluster-autoscaler
        - --v=4
        - --stderrthreshold=info
        - --cloud-provider=aws
        - --skip-nodes-with-local-storage=false
        - --expander=least-waste
        -
        --node-group-auto-discovery=asg:tag=k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/<YOUR CLUSTER NAME>
        - --balance-similar-node-groups
        - --skip-nodes-with-system-pods=false
    ...
...
```

Cluster Autoscaler 이미지 태그를 이전 단계에서 적어둔 버전으로 설정한다. [1.22.n](#) 을 사용자의 고유한 값으로 교체한다.

```
$ kubectl set image deployment cluster-autoscaler \
  -n kube-system \
  cluster-autoscaler=k8s.gcr.io/autoscaling/cluster-autoscaler:v<1.22.n>
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

II. 크롤러 배치 작업

▶ 모듈 Import

```
#!/usr/bin/python3
import os
import time
import json
from kafka import KafkaProducer
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import ElementNotInteractableException
```

사용 라이브러리 및 함수

Module	Function	Descriptions
os	-	OS에 의존하는 다양한 기능 제공
time	-	time : 시작 시간 측정 sleep : 명시적 대기
json	-	파이썬에서 JSON 형태의 데이터를 처리하기 위해 사용
kafka	KafkaProducer	파이썬에서 카프카 프로듀서 호출
selenium	webdriver	다양한 셀레니움 Browser 드라이버 설치

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Module	Function	Description
selenium.webdriver.common.by	By	해당 경로에서 Element를 찾기 위함
selenium.common.exceptions	NoElementException	웹 페이지 또는 애플리케이션에서 요소를 찾거나 액세스할 수 없을 때 발생하는 예외 처리
selenium.common.exceptions	ElementNotInteractableException	클릭할 성질의 Element가 존재할 때 발생하는 예외 처리

[표 2-27] 파이썬 크롤러 사용 모듈

▶ 셀레니움을 사용한 크롬 드라이버 불러오기

크롬 드라이버를 설치한 다음 아래와 같은 코드를 실행한다.

`user_agent`와 `options`를 생성하여 Ubuntu 크롤러가 쇼핑몰에서 사용자로 인식될 수 있도록 한다.

```
def chromeWebdriver():
    options = webdriver.ChromeOptions()
    driver = webdriver.Chrome('./chromedriver',options=options)
    # 1. User-Agent 설정
    user_agent = 'Mozilla/5.0 (X11; Linux X86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36'
    options.add_argument('user-agent={0}'.format(user_agent))
    # 2. Headless 모드 설정
    options.add_argument('--headless')
    options.add_argument('--no-sandbox')
    options.add_argument('--disable-dev-shm-usage')

    return driver
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

사용 변수

> User-Agent 설정

해당 설정을 통해 쇼핑몰 사이트가 크롤링 서버를 프로그램이 아닌 사용자로 인식할 수 있도록 설정한다.

> Headless 모드 설정

Ubuntu OS에서는 GUI를 제공하지 않기 때문에 Headless 모드로 셀레니움 코드를 실행하여 웹페이지를 렌더링해야한다.

▶ Kafka URL, Topic, Server 불러오기

```
url=os.environ.get("url")
topic=os.environ.get("topic")
server=os.environ.get("server")
print(url)
print(topic)
print(server)
```

환경변수로 지정한 `url`, `topic`, `server` 값을 불러 온 후 `print`를 사용하여 확인한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ 크롬 드라이버로 **URL**에 접속

```
driver = chromeWebdriver()
driver.get(url)
time.sleep(2)
```

`driver.get(url)` 를 통해 크롬 드라 지정 된 `url`의 웹 페이지에 접속한다.

`time.sleep(2)` 에 의해 접속 지연 시간이 2초로 설정되었기 때문에 크롤러는 2초 내에 접속을 완료해야한다.

▶ 변수 초기화

```
comment=[]
star=[]
date=[]

# 댓글 페이지 인덱스 (1-10:다음)
num = 2
```

`comment`(리뷰), `star`(별점), `date`(작성 일자) list와 리뷰 페이지 인덱스 변수 `num`을 초기화 한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ Kafka Producer 생성

```
# kafka producer
producer = KafkaProducer(acks=1, compression_type='gzip', bootstrap_servers=[server],
value_serializer=lambda x: json.dumps(x, ensure_ascii=False).encode('utf-8'))
```

프로젝트의 데이터 파이프라인에서 Python 크롤러는 Kafka의 Producer로 동작한다.

Kafka는 기본적으로 Java를 제공하지만 Python 등 ThirdParty에서 사용할 수 있도록 해준다.

프로듀서 속성

Option	Description	Details
acks	메시지 요청 후 요청 완료 전 승인 수	메시지 받은 사람이 메시지를 잘 받았는지 체크하는 옵션
compression_type	데이터 압축 포맷 (None, gzip, snappy, lz4 중 선택)	gzip 형식으로 압축하여 전달
bootstrap_servers	최초 연결을 위한 브로커 서버 목록	고정 IP가 부여 된 Kafka Broker 서버 주소 목록
value_serializer	메세지의 값을 직렬화 할 직렬처리기	- <code>json.dump()</code> 메소드 : json 딕셔너리를 유니코드로 표현 - <code>ensure_ascii=False</code> 옵션 : 데이터를 한글로 저장 - <code>encode('utf-8')</code> : 문자열(유니코드)을 byte 코드로 변환

[표 2-28] Kafka 프로듀서 속성

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

▶ 크롤링 시작

▶▶ 코드 상세 설명

>> 시간 측정

```
start=time.time()
```

크롤링을 시작하기 전, `time` 모듈의 `time()` 함수를 `start` 변수로 지정하여 크롤링 시간을 측정한다.

>> 페이지를 넘겨가며 계속적으로 크롤링 실행

```
while True:
    ...
    num += 1
    if num == 13:
        num = 2
```

크롤링이 계속적으로 진행될 수 있도록 무한 반복문 `while True` 를 사용한다.

리뷰 페이지는 총 10개의 인덱스로 이루어져 있기 때문에 `num` 변수를 설정하여 10개의 인덱스가 지나면 다음 리뷰페이지로 넘어갈 수 있도록 설정한다.

```
time.sleep(1)
tmp = driver.find_element(By.XPATH,
    '/html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/div/div/a[{}]').format(num)
tmp.send_keys("\n")
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

사용함수

Function	Description
time.sleep()	지연시간 측정 (실수 단위 지정 가능)
driver.find_element(By.<속성>, '<속성 값>')	By.XPATH : 태그의 경로에서 추출
tmp.send_keys("\n")	검색창에 엔터 입력

[표 2-29] 파이썬 크롤러 사용 함수

우선 `time.sleep()` 함수로 1초 지연시간을 설정한다. 이는 크롤러가 다음 리뷰 페이지를 크롤링하기 위해 넘어갈 때 발생하는 지연을 고려하여 설정하였다. (실험 결과 1초의 지연시간을 설정했을 때가 가장 문제 없이 데이터를 수집할 수 있었다.)

`driver.find_element(By.<속성>, '<속성 값>')` 함수를 사용하여 태그의 경로에 해당하는 곳(리뷰 페이지 인덱스)에서 엔터를 입력하여 다음 페이지로 이동한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> 리뷰, 별점, 작성일자 크롤링

```

for cmt in range(1,21):
    #comment
    time.sleep(1)
    list_comnt = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[2]/div/span'.format(cmt))
    if list_comnt.text == '한달 사용기':
        list_comnt = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[2]/div/span[2]'.format(cmt))
        if list_comnt.text == '재구매':
            list_comnt = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[2]/div/span[3]'.format(cmt))
        elif list_comnt.text == '재구매':
            list_comnt = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[2]/div/span[2]'.format(cmt))
            comment.append(list_comnt.text)

    #star
    list_star = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[1]/div[2]/div[1]/em'.format(cmt))
    star.append(list_star.text)

    #date
    list_date = driver.find_element(By.XPATH,
'//html/body/div/div/div[3]/div[2]/div[2]/div/div[3]/div[6]/div/div[3]/div/div[2]/ul/li[{}]/div/div/
div/div[1]/div/div[1]/div[1]/div[2]/div[2]/span'.format(cmt))
    date.append(list_date.text)

```

변수

- list_comnt : 리뷰를 크롤링해서 저장하는 list
- list_star : 별점을 크롤링해서 저장하는 list
- list_date : 작성일자를 크롤링해서 저장하는 list

한 페이지에 리뷰가 20개씩 존재하므로 반복문(for)을 설정하여 리뷰를 크롤링한다.

우선 `time.sleep()` 함수로 1초 지연시간을 설정한다. 이는 크롤러가 다음 리뷰를 크롤링하기 위해 이동할 때 발생하는 지연을 고려하여 설정하였다.

`driver.find_element` 함수를 이용해 경로를 통해 리뷰를 크롤링 해온다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	



【그림 2-20】이중 키워드 댓글 예시

```

▼<div class="_19SE1Dnqkf">
  ▼<div class="YEtwtZFLDz">
    <span class="_leidska71d">한달사용기</span> == $0
    <span class="_leidska71d">재구매</span>
    <span class="_3QOEeS6NLn">조아조아요 한두번 빠니까 목이 좀 늘어남 그래도 클래식 이즈 더 베스트 질이 좋아요</span>
  </div>
</div>

```

【그림 2-21】이중 키워드 댓글 코드로 확인

만약 위 사진과 같이 리뷰의 앞부분에 “한달사용기, 재구매+한달사용기, 재구매” 키워드가 앞에 존재하는 경우, if문을 활용하여 키워드가 아닌 리뷰를 추출할 수 있도록 크롤링 경로를 수정한다. 별점과 작성일자는 추가 키워드가 없으므로 추가 조건 없이 크롤링을 진행하여 list_star, list_date 리스트에 추출한 값을 넣는다.

>> Kafka에 추출 값 퍼블리싱 (전송)

```

tmp={'star':star.pop(), 'comment':comment.pop(), 'date':date.pop()}
producer.send(topic, value=tmp)

producer.flush()

```

tmp 딕셔너리의 key를 크롤링 한 요소들로 라벨링하고, pop() 함수를 사용하여 star, comment, date 리스트의 내부 요소를 딕셔너리의 value 값으로 넣어준다.

producer.send() 함수를 사용하여 프로듀서는 Kafka 브로커의 Topic에 tmp 딕셔너리를 전송한다.

producer.flush() 를 추가하여 클라이언트가 미해결 메시지가 브로커에 전달될 때까지 기다리도록 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> 크롤링 중지

```
# 크롤링 할 댓글이 없을 경우, 크롤링 성공
except NoSuchElementException:
    print("elapsed :", time.time() - start)
    tmp={'status':'Success', 'elapsed':time.time()-start}
    producer.send(topic, value=tmp)
    driver.quit()

# 다음 페이지가 없을 경우, 크롤링 성공
except ElementNotInteractableException:
    print("elapsed :", time.time() - start)
    tmp={'status':'Success', 'elapsed':time.time()-start}
    producer.send(topic,value=tmp)
    driver.quit()

# 기타 오류
except:
    tmp={'status':'Failed', 'elapsed':time.time()-start}
    producer.send(topic,value=tmp)
```

위의 코드로 크롤링이 성공한 경우를 예외처리 해준다.

크롤링에 성공한 경우 Success를 반환하고, 실패한 경우 Failed를 반환한다.

- 크롤링 할 리뷰이 없을 경우 (성공)
- 다음 페이지가 없을 경우 (성공)
- 기타 오류 (실패)

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[SFR-002] 리뷰 데이터 적재 알림 전송

Slack 알림 설정

스마트 스토어 쇼핑몰 별로 매일 정해진 시간에 수행되어야 하는 배치들이 있다. 이 배치들이 실행되어 브로커에 데이터가 적재되면 Logstash 컨슈머에서 데이터를 받아와 OpenSearch에 적재하게 된다.

데이터가 Logstash를 통해 OpenSearch에 정상적으로 적재되었는지 매일 확인하기 어렵기 때문에 알림들을 설정하였다. OpenSearch의 Alert 기능을 활성화하고, Crawler 파드가 네이버 스마트 스토어의 모든 리뷰를 성공적으로 가져왔다면 status 필드에 Success라는 메세지를 보내도록 설정해두어야 한다.

>> Destination 설정

The screenshot shows the 'Destination' configuration screen in Kibana. The 'Name' field is set to 'Send_slack_notification'. The 'Type' dropdown is set to 'Slack'. The 'Webhook URL:' field contains the value 'URL', which is highlighted with a red border.

[그림 2-22] Kibana Destination 설정

먼저 메시지를 보낼 Destination을 설정한다. Name, Type, webhook url을 입력하여 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

>> Monitor 설정

Configure monitor

Monitor name

Monitor state

Disabled monitors do not run.

 Disable monitor

[그림 2-23] Kibana Monitor 설정 1

모니터 생성 화면에 들어가서 Monitor의 이름을 입력한다.

Define monitor

Method of definition

Index

[그림 2-24] Kibana Monitor 설정 2

Define extraction query

```

1 | {
2 |   "size": 0,
3 |   "query": {
4 |     "bool": {
5 |       "filter": [
6 |         {
7 |           "range": {
8 |             "timestamp": {
9 |               "from": "{{{period_end}}}||+9h-2m",
10 |               "to": "{{{period_end}}}||+9h",
11 |               "include_lower": true,
12 |               "include_upper": true,
13 |               "format": "epoch_millis",
14 |               "boost": 1
15 |             }
16 |           }
17 |         },
18 |         {
19 |           "match": {
20 |             "status": {
21 |               "query": "Success",
22 |               "operator": "eq",
23 |               "prefix_length": 0,
24 |               "max_expansions": 50,
25 |               "fuzzy_transpositions": true,
26 |               "lenient": false,
27 |               "zero_terms_query": "NONE",
28 |               "auto_generate_synonyms_phrase_query": true,
29 |               "boost": 1
30 |             }
31 |           }
32 |         }
33 |       ],
34 |       "must": []
35 |     }
36 |   }
37 | }
```

Extraction query response

```

1 | {
2 |   "_shards": {
3 |     "total": 6,
4 |     "failed": 0,
5 |     "successful": 6,
6 |     "skipped": 0
7 |   },
8 |   "hits": {
9 |     "hits": [],
10 |     "total": {
11 |       "value": 0,
12 |       "relation": "eq"
13 |     },
14 |     "max_score": null
15 |   },
16 |   "took": 9,
17 |   "timed_out": false
18 | }
```

[그림 2-25] Kibana Monitor 설정 2

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Define Monitor 탭에서 Define using extraction query를 선택하고 조회할 index를 선택한 후, 어떤 쿼리로 조회할지 입력한다. 이번 프로젝트에서는 현재 시간부터 2분 이내에 들어온 status 필드 값 중 Success인 값을 조회하도록 설정했다. 조건을 설정한 쿼리는 다음과 같다.

[Opendisro-Alerting-Slack]

```
{
  "size": 0,
  "query": {
    "bool": {
      "filter": [
        {
          "range": {
            "timestamp": {
              "from": "{{period_end}}||+9h-2m",
              "to": "{{period_end}}||+9h",
              "include_lower": true,
              "include_upper": true,
              "format": "epoch_millis",
              "boost": 1
            }
          }
        },
        {
          "match": {
            "status": {
              "query": "Success",
              "operator": "OR",
              "prefix_length": 0,
              "max_expansions": 50,
              "fuzzy_transpositions": true,
              "lenient": false,
              "zero_terms_query": "NONE",
              "auto_generate_synonyms_phrase_query": true,
              "boost": 1
            }
          }
        }
      ],
      "adjust_pure_negative": true,
      "boost": 1
    }
  }
}
```

- **range** : 데이터가 조회될 범위를 설정할 수 있다. 현재 시간부터 2분 이내에 들어온 데이터를 조회하도록 설정했다. 여기서 중요한 점은 opensearch의 period_end는 UTC+0을 기준으로 설정되어 있다는 점이다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

앞서 설정했던 KST Timestamp와 약 9시간의 차이가 있기 때문에 period_end에 9시간을 더해 시간 값을 조정한 후 2분의 범위를 설정했다.

- match : status 필드에 Success라는 값을 조회하도록 설정했다.

Monitor schedule

When do you want this monitor to run?

Frequency

By interval

Every

1 Minutes

[그림 2-26] Kibana Monitor schedule 설정

마지막으로 Monitor schedule을 설정하고 Monitor 설정을 마무리한다. By interval로 1분마다 조회하도록 설정했다.

>> Trigger 설정

Monitor 설정을 완료한 후, Trigger를 설정한다. Trigger Name을 입력하고 Trigger Condition 항목을 설정한다. 정해진 시간 범위 내에 적재 성공 메시지가 1개 이상 발생하면 알림을 받기 위해 아래와 같이 설정한다.

```
ctx.results[0].hits.total.value > 0
```

마지막으로 trigger 생성 화면의 제일 하단에 있는 Configure Action 부분을 설정한다. Action Name을 입력하고 앞서 생성한 Destination을 선택한다.

Message 와 Message Preview 필드 중간에 있는 버튼을 눌러 Slack으로 메시지가 수신 되는지 확인한 후 Trigger를 생성한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Configure actions

Add action

Slack notification: Send slack message Delete

Action name Send slack message
Names can only contain letters, numbers, and special characters

Destination Send_slack_notification - (Slack)

Message subject Kibana Alerting

[그림 2-27] Kibana Configure Actions 설정

Message Info

Elasticsearch 적재 완료

- 적재 완료시간(UTC): {{ctx.periodEnd}}
- Link <[http://a81854a2eb5c24cb3ae4f7a2aec82a76-88758549.ap-northeast-2.elb.amazonaws.com/app/discover#/?_g=\(filters:!\(\),refreshInterval:\(pause:!t,value:0\),time:\(from:now-2m,to:now\)\)&_a=\(columns:!\(_source\),filters:!\(\),index:f587d9d0-1317-11ed-8f25-fd1033bc25e8,interval:auto,query:\(language:korean,query:""\),sort:\[!1\]\)](http://a81854a2eb5c24cb3ae4f7a2aec82a76-88758549.ap-northeast-2.elb.amazonaws.com/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-2m,to:now))&_a=(columns:!(_source),filters:!(),index:f587d9d0-1317-11ed-8f25-fd1033bc25e8,interval:auto,query:(language:korean,query:\)

Embed variables in your message using Mustache templates. [Learn more about Mustache](#) Send test message

[그림 2-28] Kibana Configure Actions Message 설정

Slack 메시지에 적재 완료 시간과 로그를 확인할 수 있는 URL을 메시지로 설정한다.

Team3 Slack Notification 웹 오후 9:46

Kibana Alerting

Elasticsearch 적재 완료

- 적재 완료시간(UTC): 2022-08-03T12:46:14.992Z
- Link [로그 보러가기](#)

[그림 2-29] Kibana 알림 확인

알림 메시지가 성공적으로 전송된 것을 확인할 수 있다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[SFR-003] 온라인 쇼핑몰 리뷰를 분석하기 위한 NLP

I. 코드 실행 방법

▶ 사용 라이브러리 및 함수

모듈을 설치해야 하는 목록은 아래와 같다.

Mecab 설치 방법을 제외하고는 pip install 으로 설치가 가능하다.

```
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import urllib.request
import tensorflow as tf
from hanspell import spell_checker
from collections import Counter
from konlpy.tag import Mecab
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

▶ Mecab 설치

Mecab은 일본어용 분석기를 한국어로 사용할 수 있도록 수정한 것으로 한국어 형태소 분석기 중 가장 뛰어난 성능을 가진다.

```
$ wget https://bitbucket.org/eunjeon/mecab-ko/downloads/mecab-0.996-ko-0.9.2.tar.gz
$ tar xvfz mecab-0.996-ko-0.9.2.tar.gz
$ cd mecab-0.996-ko-0.9.2
$ ./configure
$ sudo make
$ sudo make check
$ sudo make install
$ sudo ldconfig
```

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Mecab-ko-dic 설치

```
$ wget https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.1.1-20180720.tar.gz
$ tar xvfz mecab-ko-dic-2.1.1-20180720.tar.gz
$ cd mecab-ko-dic-2.1.1-20180720
$ ./configure
$ sudo make
$ sudo make install
```

II. 코드 상세 설명

파일명 : Naver_Shopping_Review_Sentiment_Analysis.ipynb

```
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import urllib.request
import boto3
from smart_open import smart_open
from collections import Counter
from konlpy.tag import Mecab
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.python import device_lib
```

감성분석에 필요한 모듈을 Import 한다.

```
device_lib.list_local_devices()
tf.test.is_gpu_available()
```

GPU가 tensorflow 연산에 활용 가능한지 체크한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

session = boto3.Session(profile_name='default')
s3 = session.resource('s3')
bucket = s3.Bucket('potatoes3')
with smart_open('s3://potatoes3/naver_shopping.txt', 'rt', encoding='UTF8') as f2:
    data=f2.read()

total_data = pd.read_table('ratings_total.txt', names=['ratings', 'reviews'])

```

S3에 저장된 TXT 파일을 불러온다.

```

# 평점이 4, 5인 리뷰에는 레이블 1을, 평점이 1, 2인 리뷰에는 레이블 0을 부여
total_data['label'] = np.select([total_data.ratings > 3], [1], default=0)

# 각 역에 대해서 중복인 샘플 데이터 삭제
total_data.drop_duplicates(subset=['reviews'], inplace=True)

# 훈련 데이터와 테스트 데이터를 3:1 비율로 분리
train_data, test_data = train_test_split(total_data, test_size = 0.25, random_state = 42)

# 정규 표현식을 사용하여 한글을 제외하고 모두 제거
train_data['reviews'] = train_data['reviews'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]","",)
train_data['reviews'].replace('', np.nan, inplace=True)

# 테스트 데이터에 대해서도 정규 표현식을 사용하여 한글을 제외하고 모두 제거
# 중복 제거
test_data.drop_duplicates(subset = ['reviews'], inplace=True)
# 정규 표현식 수행
test_data['reviews'] = test_data['reviews'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]","",)
# 공백은 Null 값으로 변경
test_data['reviews'].replace('', np.nan, inplace=True)
# Null 값 제거
test_data = test_data.dropna(how='any')

```

긍정적 리뷰이라고 판단되는 데이터(평점 4-5)에는 레이블 “1”을, 부정적 리뷰라고 판단되는 데이터(평점 1-2점)는 레이블 “0”을 부여한다. 같은 리뷰가 여러 개 업로드 된 경우를 대비해 `drop_duplicates`를 사용해 중복 데이터를 삭제한다.

학습 데이터를 3 : 1 (훈련 데이터 : 테스트 데이터) 비율로 분리한다. 정규 표현식을 사용하여 훈련 데이터와 테스트 데이터에서 한글을 제외하고 모두 제거한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
#형태소 분석기 Mecab
from eunjeon import Mecab
mecab = Mecab()

# 불용어를 지정하여 훈련 데이터와 테스트 데이터 내 필요없는 토큰들을 제거
stopwords = ['도', '는', '다', '의', '가', '이', '은', '한', '에', '하', '고', '을', '를', '인', '듯',
'과', '와', '네', '들', '듯', '지', '임', '계']

train_data['tokenized'] = train_data['reviews'].apply(mecab.morphs)
train_data['tokenized'] = train_data['tokenized'].apply(lambda x: [item for item in x if item not
in stopwords])
test_data['tokenized'] = test_data['reviews'].apply(mecab.morphs)
test_data['tokenized'] = test_data['tokenized'].apply(lambda x: [item for item in x if item not in
stopwords])

X_train = train_data['tokenized'].values
y_train = train_data['label'].values
X_test= test_data['tokenized'].values
y_test = test_data['label'].values
```

형태소 분석기 Mecab을 설치하고 불용어를 지정한 후 학습 데이터 내 필요 없는 토큰들을 정리한다.

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train)

threshold = 2
total_cnt = len(tokenizer.word_index) # 단어의 수
rare_cnt = 0 # 등장 빈도수가 threshold보다 작은 단어의 개수를 카운트
total_freq = 0 # 훈련 데이터의 전체 단어 빈도수 총 합
rare_freq = 0 # 등장 빈도수가 threshold보다 작은 단어의 등장 빈도수의 총 합

for key, value in tokenizer.word_counts.items():
    total_freq = total_freq + value

    # 단어의 등장 빈도수가 threshold보다 작으면
    if(value < threshold):
        rare_cnt = rare_cnt + 1
        rare_freq = rare_freq + value

print('단어 집합(vocabulary)의 크기 :',total_cnt)
print('등장 빈도가 %s번 이하인 희귀 단어의 수: %s'%(threshold - 1, rare_cnt))
print("단어 집합에서 희귀 단어의 비율:", (rare_cnt / total_cnt)*100)
print("전체 등장 빈도에서 희귀 단어 등장 빈도 비율:", (rare_freq / total_freq)*100)
```

기계가 텍스트를 숫자로 처리할 수 있도록 훈련 데이터와 텍스트 데이터에 정수 인코딩을 수행한다. 단어 집합이 생성되는 동시에 단어와 빈도수의 T를 key:value 형태로 받아 빈도 판단 후, 각 단어에 고유한 정수를 부여하고 등장 빈도가 적은 단어들(1회)은 자연어 처리에서 배제하도록 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```
vocab_size = total_cnt - rare_cnt + 2
print('단어 집합의 크기 :',vocab_size)

tokenizer = Tokenizer(vocab_size, oov_token = 'OOV')
tokenizer.fit_on_texts(X_train)
X_train = tokenizer.texts_to_sequences(X_train)
X_test = tokenizer.texts_to_sequences(X_test)
```

단어 집합의 크기를 토크나이저의 인자로 넘겨주고 텍스트 시퀀스를 정수 시퀀스로 변환한다.

정수 인코딩 과정에서 이보다 큰 숫자가 부여된 단어들은 OOV(Out of Vocabulary)로 변환한다.

전체 단어 개수 중 빈도수 2 이하인 단어는 제거하였고 0번 패딩 토큰과 1번 OOV(Out of Vocabulary) 토큰을 고려하여 +2를 해준다.

```
def below_threshold_len(max_len, nested_list):
    count = 0
    for sentence in nested_list:
        if(len(sentence) <= max_len):
            count = count + 1
    print('전체 샘플 중 길이가 %s 이하인 샘플의 비율: %s'%(max_len, (count / len(nested_list))*100))

max_len = 80
below_threshold_len(max_len, X_train)

# 훈련용 리뷰의 99.99가 80이하의 길이를 가지기 때문에, 훈련용 리뷰를 길이 80으로 패딩
X_train = pad_sequences(X_train, maxlen=max_len)
X_test = pad_sequences(X_test, maxlen=max_len)
```

서로 다른 길이의 샘플 데이터를 동일한 길이로 맞춰준다. 훈련용 데이터를 기준으로 패딩하였다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

from tensorflow.keras.layers import Embedding, Dense, GRU
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import load_model
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

embedding_dim = 100
hidden_units = 128

model = Sequential()
model.add(Embedding(vocab_size, embedding_dim))
model.add(GRU(hidden_units))
model.add(Dense(1, activation='sigmoid'))

es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=4)
mc = ModelCheckpoint('best_model.h5', monitor='val_acc', mode='max', verbose=1,
save_best_only=True)

model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
history = model.fit(X_train, y_train, epochs=15, callbacks=[es, mc], batch_size=64,
validation_split=0.2)

loaded_model = load_model('best_model.h5')
print("\n 테스트 정확도: %.4f" % (loaded_model.evaluate(X_test, y_test)[1]))

```

GRU로 네이버 쇼핑 리뷰 감성 분류를 진행한다.

하이퍼파라미터인 임베딩 백터의 차원은 100, 은닉 상태의 크기는 128이다. 모델은 다대일 구조의 LSTM을 사용한다.

해당 모델은 마지막 시점에서 두 개의 선택지 중 하나를 예측하는 이진 분류 문제를 수행하는 모델이다. 이진 분류 문제의 경우, 출력층에 로지스틱 회귀를 사용해야 하므로 활성화 함수로는 시그모이드 함수를 사용하고, 손실 함수로 크로스 엔트로피 함수를 사용한다. 하이퍼파라미터인 배치 크기는 64이며, 15 에포크를 수행한다.

EarlyStopping은 검증 데이터 손실이 증가하면, 과적합 징후이므로 검증 데이터 손실이 4회 증가하면 정해진 에포크가 도달하지 못하였더라도 학습을 조기 종료한다는 의미이다.

ModelCheckpoint를 사용하여 검증 데이터의 정확도가 이전보다 좋아질 경우에만 모델을 저장한다.

validation_split=0.2를 사용하여 훈련 데이터의 20%를 검증 데이터로 분리해서 사용하고, 검증 데이터를 통해서 훈련이 적절히 되고 있는지 확인한다.

검증 데이터는 기계가 훈련 데이터에 과적합되고 있는지 않은지 확인하기 위한 용도로 사용된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

```

def sentiment_predict(new_sentence):
    new_sentence = re.sub(r'[^ㄱ-ㅎㅏ-ㅣ가-힣 ]','', new_sentence)
    new_sentence = mecab.morphs(new_sentence)
    new_sentence = [word for word in new_sentence if not word in stopwords]
    encoded = tokenizer.texts_to_sequences([new_sentence])
    pad_new = pad_sequences(encoded, maxlen = max_len)

    score = float.loaded_model.predict(pad_new)
    if(score > 0.5):
        print("{:.2f}% 확률로 긍정 리뷰입니다.".format(score * 100))
        return 1
    else:
        print("{:.2f}% 확률로 부정 리뷰입니다.".format((1 - score) * 100))
        return 0

```

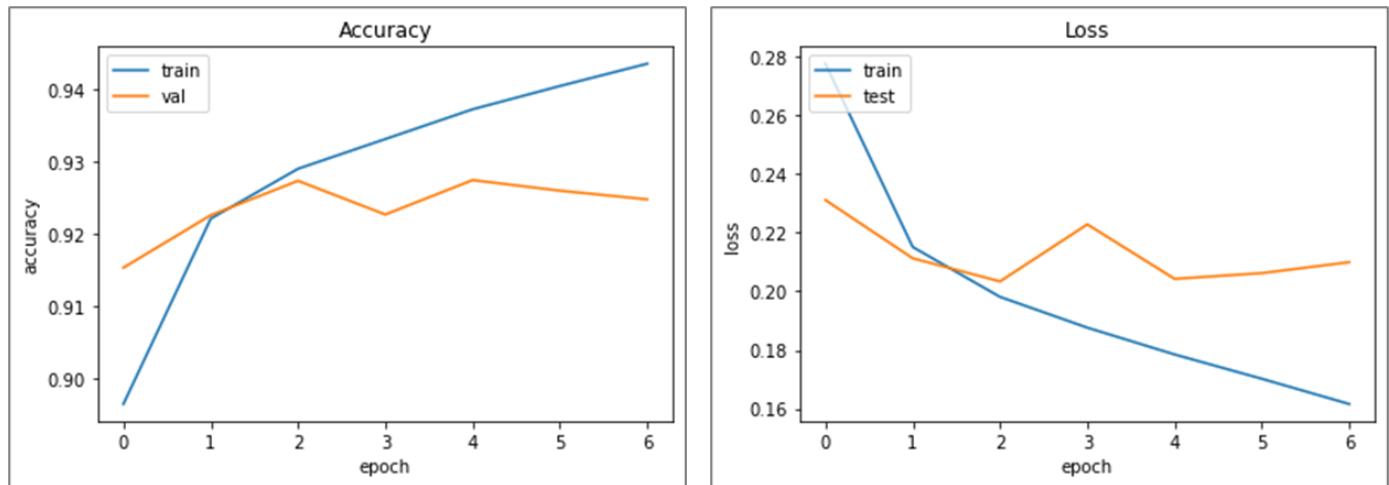
임의의 문장에 대한 예측을 위해 학습 전 전처리를 위와 동일하게 적용한다. 전처리 순서는 정규 표현식을 통한 한국어 외 문자 제거, 토큰화, 불용어 제거, 정수 인코딩, 패딩 순이다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

III. 모델 검증

하이퍼파라미터인 임베딩 백터의 차원은 100, 은닉 상태의 크기는 128이다. 모델은 다대일 구조의 LSTM을 사용한다. 해당 모델은 마지막 시점에서 두 개의 선택지 중 하나를 예측하는 이진 분류 문제를 수행하는 모델이다. 이진 분류 문제의 경우, 출력층에 로지스틱 회귀를 사용해야 하므로 활성화 함수로는 시그모이드 함수를 사용하고, 손실 함수로 크로스 엔트로피 함수를 사용한다. 하이퍼파라미터인 배치 크기는 64이며, 15 에포크를 수행한다.

딥러닝에서 에포크는 전체 트레이닝 셋이 신경망을 통과한 횟수이다. 신경망을 여러 번 통과하면서 정확도가 높아짐을 확인할 수 있다. 실제로 15 에포크로 설정되어 있지만, EarlyStopping 옵션으로 검증 데이터 손실이 증가하면, 과적합 징후이므로 검증 데이터 손실이 4회 증가하면 정해진 에포크가 도달하지 못하였더라도 학습을 조기 종료한다.



[그림 2-30] 모델 검증

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

IV. 모호한 리뷰의 감성 분석

파일명 : Customize Sentiment Result.ipynb

```
def sentiment_adjustment_manual(new_sentence, i):
    ns = new_sentence
    new_sentence = re.sub(r'[ㄱ-ㅎㅏ-ㅣㅏ-ㅣㄱ-ㅎ]', '', new_sentence)
    new_sentence = mecab.morphs(new_sentence)
    new_sentence = [word for word in new_sentence if not word in stopwords]
    encoded = tokenizer.texts_to_sequences([new_sentence])
    pad_new = pad_sequences(encoded, maxlen = max_len)

    score = float.loaded_model.predict(pad_new))
    if(score > 0.4) & (score < 0.6):
        print(ns)
        print('기준 별점: ' + str(data['star'][i]))
        new_label=input('Input Adjustment Data: ')
        return new_label

# 모호한 결과 사용자 조정

for i in range(len(data)):
    sp = sentiment_adjustment_manual(data['comment'][i], i)
    data['new_label'] = sp
print('Done')
```

긍·부정이 모호하다고 판단되는 리뷰은 자체 판단 후 긍·부정을 태깅한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

V. 기존 20만건 데이터에 크롤링한 데이터 추가

파일명 : [Data_Append.ipynb](#)

```
session = boto3.Session(profile_name='default')
s3 = session.resource('s3')
bucket = s3.Bucket('potatoes3')
with smart_open('s3://potatoes3/naver_shopping.txt', 'rt', encoding='UTF8') as f2:
    data=f2.read()

total_data = pd.read_table('ratings_total.txt', names=['ratings', 'reviews'])
```

S3에서 20만건의 데이터를 불러온다.

```
data1 = pd.read_csv('goodnara.csv')
data2 = pd.read_csv('drstyle.csv')
data3 = pd.read_csv('thecheaper.csv')
data4 = pd.read_csv('theshopsw.csv')
data5 = pd.read_csv('cloony.csv')
data6 = pd.read_csv('store180.csv')
data=pd.concat([data1, data2, data3, data4, data5, data6], ignore_index=True)
data=data.loc[:, ['comment', 'star']].dropna()

data=data.rename(columns={'comment':'reviews', 'star':'ratings'})
```

임시저장한 데이터를 불러온다.

```
total_data=pd.concat([total_data, data], ignore_index=True)

file = open("naver_shopping.txt", "w", encoding="UTF-8")
file.write(data)
file.close()

file_name = 'naver_shopping.txt'
bucket='potatoes3'
key='naver_shopping.txt'
s3 = boto3.client('s3')

res = s3.upload_file(file_name, bucket, key)
```

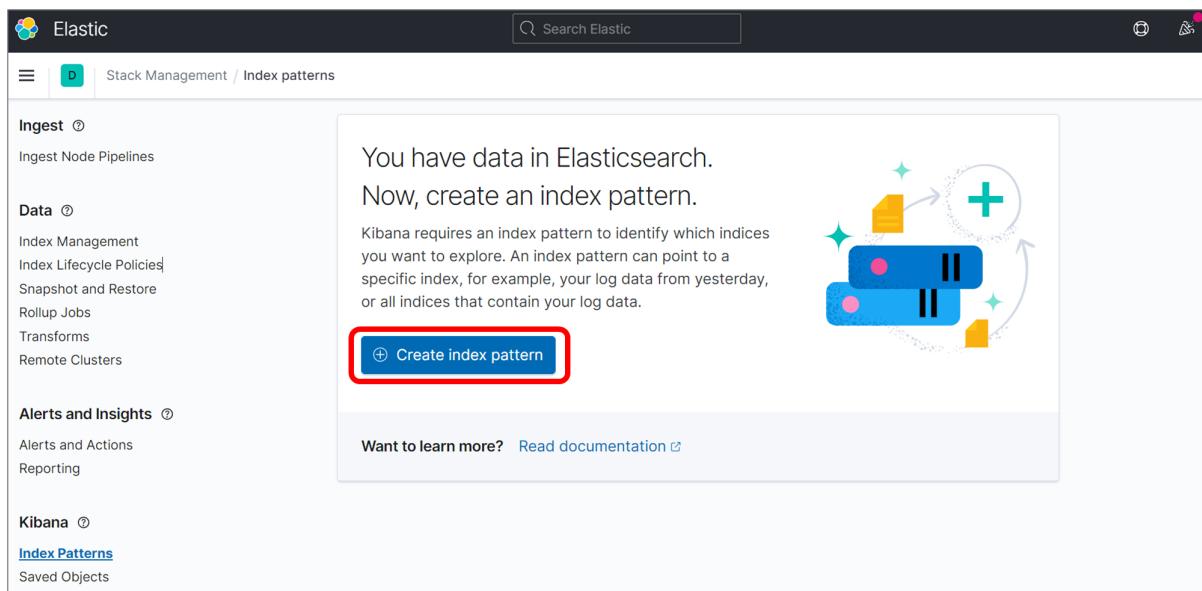
`pd.concat`을 사용해 동일한 형태의 DataFrame을 병합하고, `file`을 사용해 병합한 데이터를 `naver_shopping.txt` 파일로 임시저장한 후 S3에 업로드한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

[SFR-004] 시각화를 통한 대시보드 생성

I. 인덱스 패턴 추가

데이터 처리 및 분석을 위한 visualization을 만들기 전에 Kibana에 인덱스 패턴을 설정해야 한다. 인덱스 패턴은 검색 및 분석을 실행하는 OpenSearch Index를 식별하거나 필드를 설정하는데 사용한다. 인덱스 패턴은 여러 인덱스에 대응할 수 있는 선택적 와일드 카드를 포함한 문자열이다.



[그림 2-31] 인덱스 패턴 생성

Kibana에 접속해서 Discover 탭에 들어가면 새로운 인덱스 패턴을 추가할 수 있다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

Create index pattern

An index pattern can match a single source, for example, `filebeat-4-3-22`, or **multiple** data sources, `filebeat-*`.
[Read documentation](#)

Step 1 of 2: Define an index pattern

Index pattern name Next step >

Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, ", <, >, | are not allowed.

Include system and hidden indices

✓ Your index pattern matches 6 sources.

<code>smartstore.180store.review-220803</code>	Index
<code>smartstore.cloony.review-220803</code>	Index
<code>smartstore.drstyle.review-220803</code>	Index
<code>smartstore.goodnara.review-220803</code>	Index
<code>smartstore.thecheaper.review-220803</code>	Index
<code>smartstore.theshopsw.review-220803</code>	Index

Rows per page: 10 ▾

【그림 2-32】인덱스 패턴 정의

인덱스 패턴을 `smartstore*`로 설정한다.

Create index pattern

An index pattern can match a single source, for example, `filebeat-4-3-22`, or **multiple** data sources, `filebeat-*`.
[Read documentation](#)

Step 2 of 2: Configure settings

Specify settings for your `smartstore*` index pattern.

Select a primary time field for use with the global time filter.

Time field Refresh ▼

[Show advanced settings](#)

[Back](#) Create index pattern

【그림 2-33】인덱스 패턴 Configure 세팅

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

smartstore*

Time field: 'timestamp'

This page lists every field in the **smartstore*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API [Mapping API](#).

Fields (13) Scripted fields (0) Source filters (0)

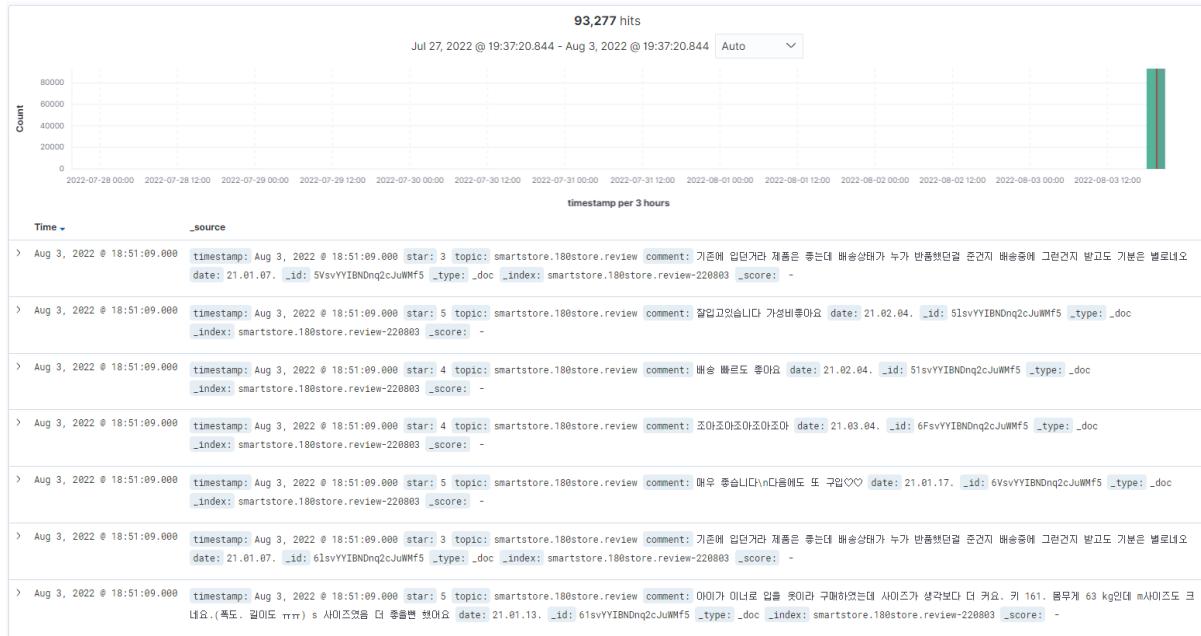
Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	✎
_index	string		●	●	✎
_score	number				✎
_source	_source				✎
_type	string		●	●	✎
comment	string		●		✎
comment.keyword	string		●	●	✎
date	string		●		✎
date.keyword	string		●	●	✎
star	number		●	●	✎

Rows per page: 10 < 1 2 >

[그림 2-34] 인덱스 패턴 생성 확인

time field를 timestamp로 설정하고 생성 버튼을 누르면 성공적으로 인덱스 패턴이 생성된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	



[그림 2-35] Kibana 로그 확인

인덱스 패턴이 생성되면 Kibana의 Discovery 탭에서 실시간으로 들어오는 로그들을 확인 할 수 있다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

II. 자연어 처리 데이터 인덱스 생성

Jupyter Notebook에서 자연어 처리 된 데이터로 시각화를 진행한다. 분석 데이터를 담을 인덱스를 생성한다.

```
PUT analyzed_data
{
  "settings": {
    "number_of_shards": 5,
    "number_of_replicas": 3
  },
  "mappings": {
    "properties": {
      "Name" : {
        "type": "keyword"
      },
      "Star" : {
        "type": "long"
      },
      "Date" : {
        "type" : "date"
      },
      "Word" : {
        "type": "keyword"
      },
      "Sentiment" : {
        "type" : "keyword"
      },
      "Adjustment-sentiment" : {
        "type" : "keyword"
      }
    }
  }
}
```

- number_of_shards : 해당 인덱스의 프라이머리 샤드의 수
- number_of_replicas : 해당 인덱스의 복제본 샤드의 수
- properties : 필드의 이름과 데이터 타입 (RDBMS의 schema에 해당)
 - Name : 쇼핑몰 이름
 - Star : 평점
 - Date : 리뷰 작성일
 - Word : 토큰화 된 리뷰 단어
 - Sentiment : 조정 전 감성 분석 결과 (긍정, 부정)
 - Adjustment-sentiment : 조정 후 감성 분석 결과 (긍정, 부정)

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

2.4.3. 정상 동작 확인

크롤링

```
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "머리 크신분들은 여기 무조건 추천합니다.", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여", "date": "22.04.18."}
```

[그림 2-36] 크롤링 데이터 예시

한 줄로 표현 된 JSON 데이터 리스트가 추출된다.

데이터 적재

```
{
  "_index" : "smartstore.drstyle.review-220804",
  "_type" : "_doc",
  "_id" : "G167ZoIBNDnq2cJu7hzC",
  "_score" : 6.6732597,
  "_source" : {
    "timestamp" : "2022-08-04T11:42:48.000Z",
    "star" : 5,
    "topic" : "smartstore.drstyle.review",
    "comment" : "머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여",
    "date" : "22.04.18."
  }
},
{
  "_index" : "smartstore.drstyle.review-220804",
  "_type" : "_doc",
  "_id" : "k168ZoIBNDnq2cJuSBws",
  "_score" : 6.6732597,
  "_source" : {
    "timestamp" : "2022-08-04T11:43:11.000Z",
    "star" : 5,
    "topic" : "smartstore.drstyle.review",
    "comment" : "머리 크신분들은 여기 무조건 추천합니다.",
    "date" : "22.04.18."
  }
}
```

[그림 2-37] 정제된 데이터 예시

데이터 정제단에서 OpenSearch로 쇼핑몰 리뷰 데이터가 적재되고

Jupyter Notebook에서 OpenSearch로 쇼핑몰 리뷰 감성 분석 데이터가 적재된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

리뷰 NLP

```
In [38]: sentiment_predict('무려 한사이즈 이상 커서 잠옷으로 입을 반품이 6000원인데 돈을 또 ¶
쓰게 되거나 잘못가게 되므로 학생들 구입시 꿈꼼하게 볼수있게 사이즈 안내 강조해서 표기바라')

1/1 [=====] - 0s 22ms/step
99.17% 확률로 부정 리뷰입니다.

In [32]: sentiment_predict('예쁘긴한데 티가 너무 길어요 .. 무슨 치마수준이라 바지 안입어도 될 길이에요 ¶
;; 수선해서 입어야할거같아여 .. 제질도 그닥 좋은 제질은 아니에요')

1/1 [=====] - 0s 476ms/step
98.03% 확률로 부정 리뷰입니다.
```

[그림 2-38] Jupyter Notebook 분석 결과 예시

Jupyter Notebook을 사용하여 데이터 학습 모델에 리뷰를 입력하여 감성 분석 결과를 확인할 수 있다. 해당 데이터 모델을 사용하여 사용자 쇼핑몰 리뷰의 감성 분석을 진행한다.

Slack 알림

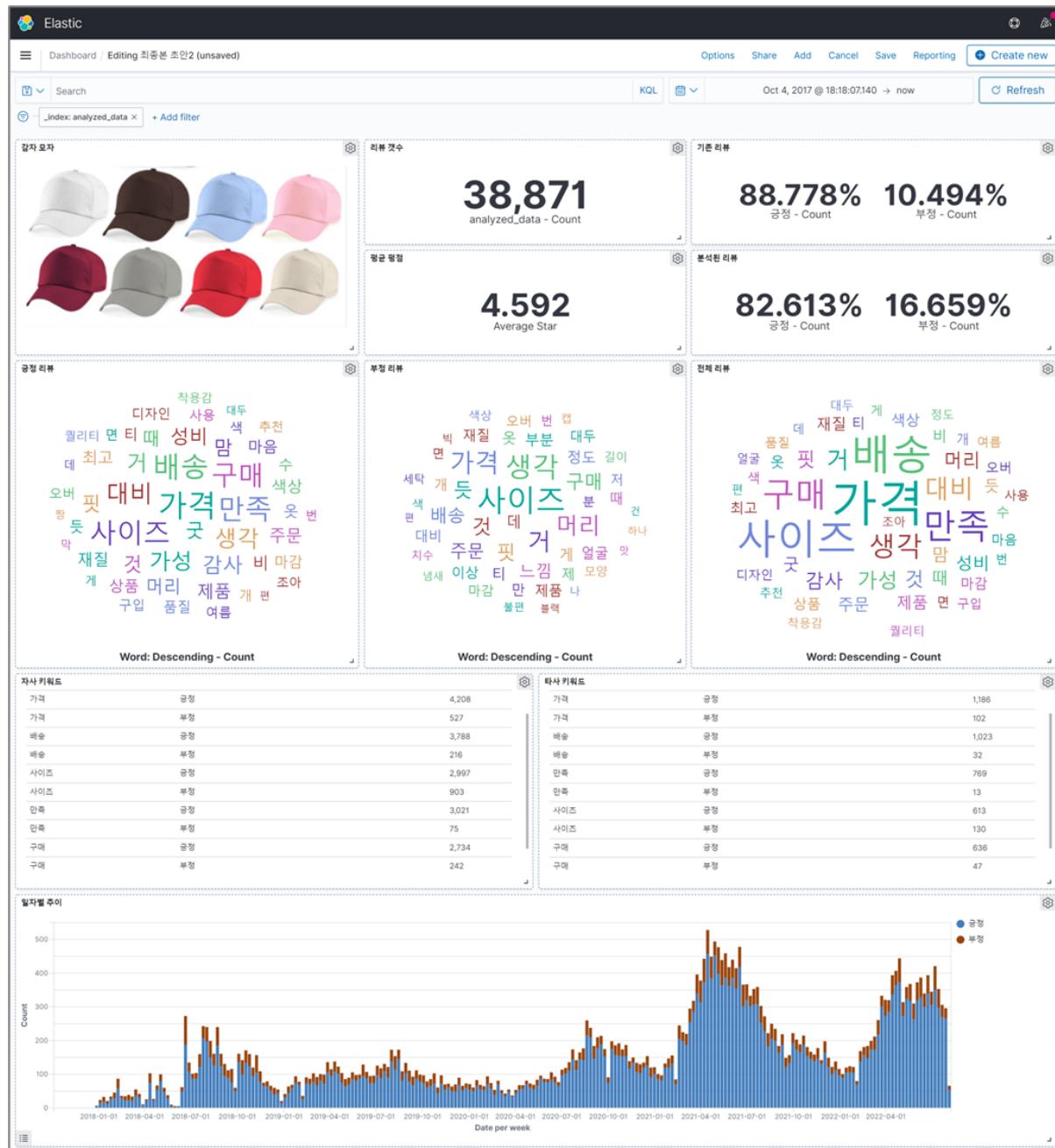


[그림 2-39] Kibana Slack 알림

OpenSearch에 데이터 적재 완료 시 Slack으로 알림 메시지가 전송된다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

대시보드



[그림 2-40] 대시보드 구현

쇼핑몰 제품의 이미지와 자사 긍정 키워드(태그 클라우드), 자사 부정 키워드(태그 클라우드), 타사 긍정 키워드(태그 클라우드), 자사 상위 키워드 카운트 데이터 테이블, 타사 상위 키워드 카운트 데이터 테이블, 긍·부정 추이 그래프를 대시보드에서 확인할 수 있다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

3. 결론

3.1. 결과물 활용 방안

한국 소비자 연맹에 따르면 인터넷 쇼핑몰 소비자들의 97.2%가 제품 구매 시 리뷰를 참고하여 제품을 구매한다. 그리고 제품이 아무리 좋아 보여도 리뷰가 좋지 않으면 96.7%가 제품을 구매하지 않는다. 이처럼 인터넷 쇼핑몰을 운영하는데 중요한 요소인 리뷰를 해당 서비스를 통해 쉽게 관리 할 수 있다. 감성 분석을 통해 부정적인 리뷰 및 쇼핑몰의 강점도 파악하며 경쟁사들의 리뷰까지 한눈에 파악하며 일자 별로 추적-관리할 수 있다.

3.2. 발생 문제(에러) 및 해결 방안

① 크롤링 코드 작동 오류

오류 및 원인

크롤링하는 데이터 양이 방대해질 경우 특정 사이트에서의 IP 차단한다.

이런 경우 해당 서버에 부하가 걸릴 수 있기 때문에 크롤링 활동을 차단하게 된다. 또한 한동안 해당 사이트에 접속이 불가능해진다.

해결 방법

파이썬의 `time.sleep()` 모듈을 사용하여 중간에 지연시간을 두게 한다.

크롤러가 다음 리뷰 페이지를 크롤링하기 위해 넘어갈 때 발생하는 지연을 고려하여 1초로 설정한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

② OOMKilled 오류로 인한 크롤러 파드 강제 종료

오류 및 원인

메모리가 부족하여 파드를 강제 종료 된다.

해결 방법

기본적으로 BestEffort → Burstable → Guranteed 순서로 종료된다. 가용 메모리는 파드의 사용중인 메모리 / 한계 메모리 * 100%로 계산된다.

Guranteed QoS 클래스가 할당되는 파드를 생성하였다.

[전제 조건]

- 파드 내 모든 컨테이너는 메모리 상한과 메모리 요청량을 가지고 있어야 한다.
- 파드 내 모든 컨테이너의 메모리 상한이 메모리 요청량과 일치해야 한다.
- 파드 내 모든 컨테이너는 CPU 상한과 CPU 요청량을 가지고 있어야 한다.
- 파드 내 모든 컨테이너의 CPU 상한이 CPU 요청량과 일치해야 한다.

해당 크롤러 컨테이너는 메모리 상한선과 메모리 요청량을 3000Mi, CPU 상한과 요청량은 1500m로 설정하였다. 또한 매일 0시에 배치 작업 진행하며 비용 절감을 위해 클러스터를 t3.medium 6개로 오토스케일링 하도록 설정하였다.

(기존 1개의 클러스터에서, 배치 작업 진행시 6개의 클러스터로 구성)

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

③ 크롤링 진행 중 파일 크롤러 파드 재부팅

오류

크롤링 진행 중인 파드가 소멸된 후 다시 시작되는 오류

원인

CA의 스케일 다운으로 인한 작업이 배정되어 있는 클러스터 스케일 다운

따라서 크롤링이 진행 중인 파드가 소멸된 후 다시 시작되는 오류 발생

해결 방법

아무 작업도 배정되지 않은 클러스터를 스케일 다운하도록 설정한다.

"cluster-autoscaler.kubernetes.io/safe-to-evict": "false" 을 annotations 내 삽입을 통해 CA가 파드가 올라와있는 클러스터 노드를 제거하지 못하도록 설정한다.

참고:

<https://github.com/kubernetes/autoscaler/blob/master/cluster-autoscaler/FAQ.md#what-types-of-pods-can-prevent-ca-from-removing-a-node>

☞ What types of pods can prevent CA from removing a node?

- Pods with restrictive PodDisruptionBudget.
- Kube-system pods that:
 - are not run on the node by default, *
 - don't have a [pod disruption budget](#) set or their PDB is too restrictive (since CA 0.6).
- Pods that are not backed by a controller object (so not created by deployment, replica set, job, stateful set etc). *
- Pods with local storage. *
- Pods that cannot be moved elsewhere due to various constraints (lack of resources, non-matching node selectors or affinity, matching anti-affinity, etc)
- Pods that have the following annotation set:

```
"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"
```

*Unless the pod has the following annotation (supported in CA 1.0.3 or later):

```
"cluster-autoscaler.kubernetes.io/safe-to-evict": "true"
```

Or you have overridden this behaviour with one of the relevant flags. See below for more information on these flags.

[그림 3-1] 크롤러 파드 재부팅 참고 자료

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

④ Replication Factor 에러

오류

Kafka 브로커와 통신하여 Topic 생성 시 출력 되는 오류

```
Error while executing topic command : Replication factor: 1 larger than available
brokers: 0.
[2022-07-22 03:19:23,022] ERROR
org.apache.kafka.common.errors.InvalidReplicationFactorException: Replication factor:
1 larger than available brokers: 0
```

원인

Kafka 설정파일(Kafka 설치 디렉토리 / config / server.properties)에 설정되어 있는 Zookeeper 정보와 Topic 생성 시 입력한 Zookeeper 정보가 불일치 해서 발생한다.

해결 방법

Kafka 설정 파일에 명시되어 있는 Zookeeper 정보를 Topic 생성 명령어에 동일하게 입력한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

⑤ Zookeeper 클러스터 실행 오류

오류

EC2 3대에 각각 Kafka와 Zookeeper를 설치하고 주키퍼를 실행할 때 출력 되는 오류

```
I won't be able to participate in leader election any longer Use
zookeeper.electionPortBindRetry property to increase retry count.
(org.apache.zookeeper.server.quorum.QuorumCnxManager)
```

원인

Zookeeper 설정 파일인 `zookeeper.properties` 파일에 해당 서버의 공인 IP를 설정하면 해당 포트를 인식하지 못해서 발생

해결 방법

서버 설정할 때, `localhost`가 아닌 `0.0.0.0:2181`로 설정

⑥ Logstash Timestamp UTC 설정

오류

Kafka 브로커와 통신하여 Topic 생성 시 출력 되는 오류

원인

Kafka 설정파일(Kafka 설치 디렉토리 / config / `server.properties`)에 설정되어 있는 Zookeeper 정보와 Topic 생성 시 입력한 Zookeeper 정보가 불일치 해서 발생한다.

해결 방법

Kafka 설정 파일에 명시되어 있는 Zookeeper 정보를 Topic 생성 명령어에 동일하게 입력한다.

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

3.3. 프로젝트 결과 및 향후 개선점

[프로젝트 결과]

- EKS 위에 노드 그룹 생성하여 리소스 배포 (파이썬 크롤러, ELK Stack)
- 웹 크롤링 데이터 큐잉
- 리뷰 데이터 정제 및 인덱싱
- 데이터 적재
- 데이터 감성 분석 모델을 이용한 쇼핑몰 리뷰 자연어 처리
- 리뷰 데이터를 활용한 시각화

[향후 개선점]

- 리뷰 데이터를 추가로 수집하여 HDFS를 사용하여 데이터 분산 처리
- EKS에서 Kafka 배포
- 데이터 전송 단의 로그 분석 대시보드 생성 (Grafana Dashboard)
- 효율적인 협업 툴 사용

encore	인터넷 쇼핑몰 리뷰를 활용한 NLP 감성분석 파이프라인			말하는 감자
	Category	첨부파일 버전	문서 최종 수정일	
	Manual	1.0	2022.08.07	

