

OliveYoung 데이터 기반 이용한 추천시스템

경북대학교 데이터사이언스 대학원

진민준

Table of Contents

1. 연구목표	3P
2. Data Crawling	4P
3. 데이터 라벨링 & 감성분석	6P
- 긍정부정 gpt 라벨링	
- Electra 모델 긍정,부정 판별	
- 형태소 분석 및 단어 추출	
- 모형	
4. 평점별 키워드 추출	13P
5. gpt 추천시스템	15P

연구 목표

연구 목표

1. 스킨케어 제품들에 대한 긍정,부정(**Xlm-Roberta**이용)분석
2. 스킨케어(스킨/토너, 에센스/세럼/앰플, 크림,로션,미스트/오일, 스킨케어세트)핵심 키워드 분석
3. **chat-gpt**를 이용한 **fine-tuning** 추천시스템

스킨케어

스킨/토너

에센스/세럼/앰플

크림

로션

미스트/오일

스킨케어세트

Data Crawling



구달 >

[한정기획] 구달 청쿨 비타C 잡티케어 세럼
30ml+31ml 리필 기획

28,000원 **20,160원** 혜택 정보

세일 쿠폰 증정 오늘드림

194명이 보고있어요

배송정보

일반배송 | 2,500원 (20,000 원 이상 무료배송)
올리브영 배송 | 평균 4일 이내 배송

오늘드림 | 2,500원 또는 5,000원

픽업 | 배송비 조건 없음

결제혜택

THE CJ 카드 추가10%할인

CJ ONE 포인트 최대1% 적립 예상

상품을 선택해주세요

상품금액 합계

0원



마시는복숭아

TOP 702

민감성 · 여름쿨톤 · 각질 · 다크서클

한달이상사용

★★★★☆

24.06.02

피부타입 건성에 좋아요 | 피부고민 보습에 좋아요 | 자극도 자극없이 순해요

보습감 있긴한데 애매한 제품인거 같습니다..

뭔가 코팅막 씌워지듯 마무리가 되는데 그러다보니 엄청 산뜻하다라는 느낌은 들지 않았어요.
약간 걸 수분을 안날라가게 하려는거 같은데 막 흡수되는 느낌은 아녔어요.

겨울철 기준으로는 속당김이 기존 제품에서 토너 하나 바꿨다고 없어지진 않았습니다. 그냥 무난하긴한데 굳이 장점은 못느끼겠어요.

도움이 되었다면 도움 눌러주세요 감사합니다.



이 리뷰가 도움이 돼요!

36

신고하기

수집한 데이터 목록

1. 브랜드
2. 제품명
3. 정가
4. 할인가격
5. id
6. 평점
7. 피부타입
8. 피부고민
9. 자극도
10. 리뷰를 작성한 날짜
11. 리뷰

Data Crawling

category	ranking	name	brand	price	sale_price	date	rate	id	skin_type	...	select_3_content	txt
로션	1	[5년연속1위]아이디얼포맨 퍼펙트 올인원 어워즈 한정 기획	아이디얼포맨	29,000	20,000	2023.09.09	1	요요2	복합성 원톤 모공 민감성	...	보통이에요	택배 상자 열어서 보니 Wn제 품 상자 아랫부분이 개봉되어 있길래 Wn 찢찢한 기분이 들어 ...
로션	1	[5년연속1위]아이디얼포맨 퍼펙트 올인원 어워즈 한정 기획	아이디얼포맨	29,000	20,000	2023.10.20	1	멀대	복합성 원톤 탄력	...	자극없이 순해요	자주 구매하는 상품입니다. 가격대비가 성비가 너무 좋아 구매합니다.
로션	1	[5년연속1위]아이디얼포맨 퍼펙트 올인원 어워즈 한정 기획	아이디얼포맨	29,000	20,000	2024.03.21	2	wltn***	건성 여름 쿨톤 각질 민감성	...	보통이에요	향도 좋고 바르는 누낌도 좋고 올인원이어서 하나만 바르면 도요
30380 rows x 25 columns												

- 1. 데이터 개수: 30380개
- 2. 변수 개수: 25개

Chat-gpt 긍정, 부정 라벨링

```
openai.api_key='sk-proj-Kh1Lgbwt7A0BB4NEXJ4dT3B1bkFJts6PqHIXe0vHQTpSFT0V'
```

```
def sentiment_analyze(text):  
    prompt = f'''You're an assistant, labeling data on a consistent basis.  
    Label the given text with one of the following sentiments: Positive, Negative.  
    Do not enclose your output in double or single quotes, just the label.  
    Follow the example to analyze the sentiment.
```

```
    given text: 아쉽고 트러블이 났어요.  
    sentiment: Negative  
  
    given text: 무난해요.  
    sentiment: Positive  
  
    given text: {text}  
    sentiment: '''
```

```
try:  
    response = openai.ChatCompletion.create(  
        model='gpt-3.5-turbo',  
        temperature=0,  
        messages=[  
            {"role": "user", "content": prompt}  
        ]  
    )  
  
    sentiment = response.choices[0].message['content']  
    print(sentiment)  
    return sentiment  
except:  
    sentiment = 'error'  
    return sentiment
```

```
if __name__=="__main__":
```

```
    #df=pd.read_csv('oliveyoung_full_plus_everything.csv',encoding='cp949')  
    df_more['sentiment']=df_more['txt'].apply(sentiment_analyze)  
    #df.to_csv('oliveyoung_full_plus_everything_sent.csv',index=False,encoding='cp949')
```

```
Positive  
Negative  
Negative  
Positive  
Positive  
Negative  
Negative  
Positive  
Negative  
Negative  
Positive  
Positive  
Positive  
Positive  
Positive  
Positive
```

Open ai gpt 라벨링

1. 4000개 긍정 라벨링
데이터 획득
2. 4000개 부정 라벨링
데이터 획득

Electra 모델을 통한 긍정, 부정 판별

< epoch 30 학습과정 >

```
Epoch 0, Loss: 0.1587659865617752
Epoch 1, Loss: 0.06835024058818817
Epoch 2, Loss: 0.06149432435631752
Epoch 3, Loss: 0.05317007005214691
Epoch 4, Loss: 0.09770093858242035
Epoch 5, Loss: 0.0047541214153170586
Epoch 6, Loss: 0.045310065150260925
Epoch 7, Loss: 0.0844046026468277
Epoch 8, Loss: 0.032034531235694885
Epoch 9, Loss: 0.025134166702628136
Epoch 10, Loss: 0.03402005136013031
Epoch 11, Loss: 0.005221381317824125
Validation Accuracy: 0.9430000009536743
Epoch 12, Loss: 0.022837327793240547
Validation Accuracy: 0.9505000009536743
Epoch 13, Loss: 0.016450298950076103
Validation Accuracy: 0.9465000004768371
Epoch 14, Loss: 0.007642127573490143
Validation Accuracy: 0.9550000014305114
Epoch 15, Loss: 0.011657018214464188
Validation Accuracy: 0.9510000009536743
Epoch 16, Loss: 0.0015434089582413435
Validation Accuracy: 0.9430000009536743
Epoch 17, Loss: 0.003278261050581932
Validation Accuracy: 0.9480000014305114
Epoch 18, Loss: 0.0009160549961961806
Validation Accuracy: 0.9540000004768372
Epoch 19, Loss: 0.00033718213671818376
Validation Accuracy: 0.9505000009536743
Epoch 20, Loss: 0.00039192946860566735
Validation Accuracy: 0.9550000004768372
Epoch 21, Loss: 0.0032116221264004707
Validation Accuracy: 0.9510000009536743
Epoch 22, Loss: 0.002511967672035098
Validation Accuracy: 0.9530000009536743
Epoch 23, Loss: 0.0073660798370838165
Validation Accuracy: 0.9410000004768372
Epoch 24, Loss: 0.01820635423064232
Validation Accuracy: 0.9500000004768372
Epoch 25, Loss: 0.012954250909388065
Validation Accuracy: 0.9510000009536743
Epoch 26, Loss: 0.000596801342908293
Validation Accuracy: 0.9505000009536743
Epoch 27, Loss: 0.00026358477771282196
Validation Accuracy: 0.9535000009536743
Epoch 28, Loss: 0.0030767328571528196
Validation Accuracy: 0.9425000014305115
Epoch 29, Loss: 0.0025309009943157434
Validation Accuracy: 0.9470000009536743
Best Test Accuracy: 0.9550 at Epoch 14
```

< test 성능 측정 >

```
model.eval()
total_accuracy = 0
total_samples = 0

for batch in test_loader:
    with torch.no_grad():
        batch_input_ids = batch['input_ids'].to(device)
        batch_attention_mask = batch['attention_mask'].to(device)
        batch_labels = batch['labels'].to(device)

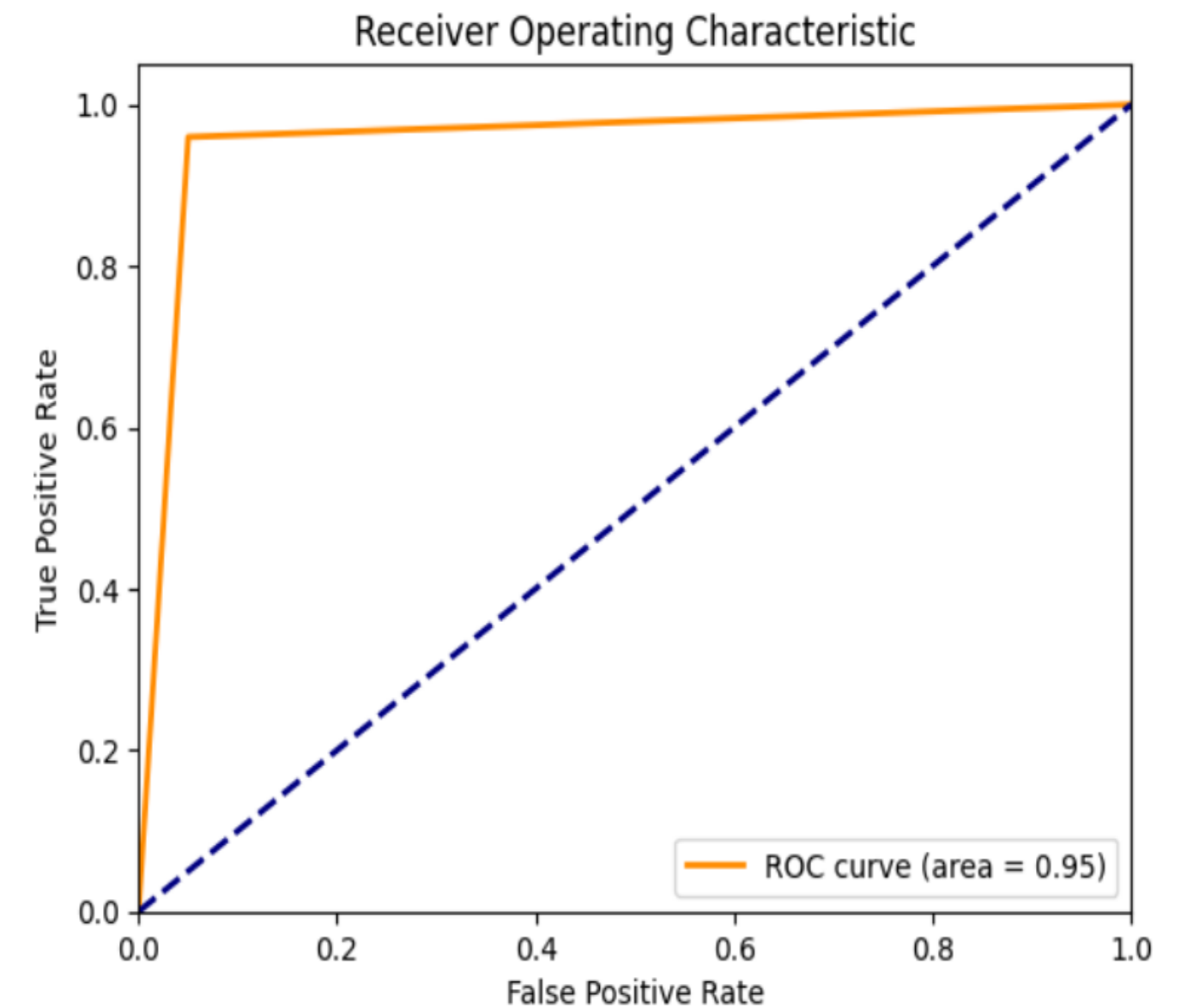
        outputs = model(input_ids=batch_input_ids, attention_mask=batch_attention_mask)
        logits = outputs.logits
        predictions = torch.argmax(logits, dim=-1)
        correct_predictions = (predictions == batch_labels).float()
        accuracy = correct_predictions.mean()

        total_accuracy += accuracy.item() * batch['input_ids'].size(0)
        total_samples += batch['input_ids'].size(0)

latest_model_test_accuracy = total_accuracy / total_samples
print(f'Latest model Test Accuracy: {latest_model_test_accuracy}')
```

Latest model Test Accuracy: 0.9470000009536743

< Roc Curve >



Electra 모델

1. 학습시 Epoch30을 했을때 validation accuracy:94~95%
2. Test accuracy:94.70%

Electra 모델을 통한 긍정, 부정 판별 예시

Electra 모델을 통한 긍정, 부정 판별 예시

잘은 면도로 인해 생기는 트러블.. 예전에는 없었던 거뭇거뭇한 자국들 때문에 고민이 많다는 남자친구!

맨날 깎다가 실패했다며 흥터져서 나타날때마다 안쓰럽더라고요

특히 마스크 같은 외부 자극들로 인해 젊고 탄력있던 피부가 축축 처지고 힘없어짐을 느끼면서 아 남자도 관리가 필요하구나, 절실하게 느끼고 있다는 보이뿌렌

올영 매장에서 직원분께 물어보니 아이디얼포맨 제품이 무난하게 쓰기 좋다고 하더라고요! 특히 여러 라인으로 세분화되어있는게 맘에 들어요 제품으로 데려오게 되었답니다

시카 플루이드는 여드름과 민감성 피부 사용 적합 테스트를 완료한 제품으로 면도를 하고 나서 따끔따끔하시거나 외부 자극으로 인한 트러블이 많이 나시는 분들을 위한 제품입니다!

안 그래도 실내와 외부의 온도차와 미세먼지, 다양한 자극들로 인해 민감해져 있는 남자들 피부를 위해 진정에 좋다는 시카 complex와 어성초, 티트리 성분이 들어있다고 하니 당장 주변 남자들에게 이 제품 선물해보시길...

1. 긴 문장이라도 문맥을 파악하여 긍정으로 인식

여름과 가을까지도 잘 몰랐는데 겨울이 되니까 확실히 플루이드 제형은 너무 가벼운거 같아요 동생 얼굴을 보니까 되게 건조하더라고요

다른 보습제랑 같이 바르시는걸 추천드리지만 이게 이름이 올인원이라 약간 모순이긴 하지만 겨울만큼은 귀찮으시더라도 보습에 더 신경 쓰시면 좋을거 같아요!

2. 두루뭉실하게 말한 문장도 부정으로 인식

Electra 모델을 통한 평점 재산출

<평점 재산출>

```
# 저장할 column 정의
predicted = []
review_sentiment_score=[]
negative_probs=[]
positive_probs=[]

# 예측 함수
for data in loader:
    with torch.no_grad():
        input_ids = data['input_ids'].to(device)
        attention_mask = data['attention_mask'].to(device)

        outputs = model(input_ids=input_ids, attention_mask=attention_mask)
        logits = outputs.logits

        # 확률값으로 변환
        probs=torch.softmax(logits, dim=-1)
        negative_prob = probs[0][0].item() # 부정 확률
        positive_prob = probs[0][1].item() # 긍정 확률
        negative_probs.append(negative_prob)
        positive_probs.append(positive_prob)

        # 가중 평균하여 감성 점수 생성
        # 가중치 예시: 부정 확률의 가중치를 0.7, 긍정 확률의 가중치를 0.3으로 설정
        weight_negative = 0.7
        weight_positive = 0.3
        sentiment_score =
        (weight_positive * positive_prob - weight_negative * negative_prob) / (weight_positive + weight_negative)
        review_sentiment_score.append(sentiment_score)

        predictions = torch.argmax(logits, dim=-1)
        predicted.append(predictions.item())

df['predict_sentiment']=predicted
df['negative_prob']=negative_probs
df['positive_prob']=positive_probs
df['score']=review_sentiment_score
df['final_score']=df['score']+df['rate']
```

1. 입력 데이터를 모델에 전달하고 모델의 출력인 로짓을 계산
2. 로짓을 확률로 변환하기 위해 소프트맥스 함수를 사용(클래스에 대한 예측확률)
3. 감성 점수 계산 – 부정과 긍정 확률을 이용해 감성 점수를 계산
 - 부정 확률에는 더 높은 가중치(0.7)
 - 긍정 확률에는 낮은 가중치(0.3)를 적용
 - 결국에는 가중치에 따라 부정적인 감성을 더 크게 반영

감성점수재산출공식: (긍정확률*부정확률 가중치 - 부정 가중치*부정확률)/ (긍정 가중치 + 부정가중치)

Electra 모델을 통한 평점 재산출 결과

<평점 재산출 결과물>

rate	id	skin_type	select_1_t	select_1_c	select_2_t	select_2_c	select_3_t	select_3_c	txt	sentiment	predict_se	negative_p	positive_p	score	final_score
									열어보니 제품 상자 아랫부분 이 개봉 되어 있길 래 찝찝한 기 분이 들어 별로였어 요. 그런데 제 요요2 복합성 원 피부타입 복합성에 피부고민 진정에 좋 자극도 보통이에 제품을 보니 화장품 입 구 쪽에 보통 썰링 테이프? 붙어있지 않나요? 이 제품은 안붙어있 는데요. 원래 이런 건지 확인 부탁						
									그런데 제 요요2 복합성 원 피부타입 복합성에 피부고민 진정에 좋 자극도 보통이에 제품을 보니 화장품 입 구 쪽에 보통 썰링 테이프? 붙어있지 않나요? 이 제품은 안붙어있 는데요. 원래 이런 건지 확인 부탁	0	0	0.999897	0.000103	-0.699897	0.300103
	1	멀대	복합성 원	피부타입	건성에 좋	피부고민	보습에 좋	자극도	자극없이 자주 구매	1	1	0.000172	0.999828	0.299828	1.299828
	2	wltn****	건성 여름	피부타입	복합성에	피부고민	진정에 좋	자극도	보통이에 도향도 좋고	1	1	0.000208	0.999792	0.299792	2.299792
									남편과 아 들이 순하 고 자극없 이 쓰는						
	3	toran5	약건성 여	피부타입	복합성에	피부고민	진정에 좋	자극도	자극없이 애용품이	1	1	0.000222	0.999778	0.299778	3.299778

```
merge['final_score'].max(),merge['final_score'].min()
```

```
(5.299875053649885, 0.3000631662682281)
```

평점 재산출 값

1. 평점 재산출 최대값: 5.298
2. 평점 재산출 최소값: 0.300

Etri 에서 개발한 형태소 분석 활용

<한국 전자 통신 연구원에서 제공하는 형태소분석 api 활용을 통한 형태소 분석 >

언어 분석 기술

언어 분석 기술

API 호출 1일 허용량

언어 분석을 위한 API
사용하기

구현 예제

활용 예제

언어 분석 API 레퍼런스

참고문헌

언어 분석 기술

언어 분석 기술은 자연어 문장의 의미를 이해하기 위한 기술로, 단어의 형태와 의미 및 문장의 구조와 의미를 분석하여 지식처리를 위한 기반 기술로 활용됩니다.

언어 분석을 위한 6종의 API는 HTTP 기반의 REST API 인터페이스로 JSON 포맷 기반의 입력 및 출력을 지원하며 ETRI에서 제공하는 API Key 인증을 통해 사용할 수 있는 Open API입니다. 사용자가 요청하는 분석 코드에 따라 형태소 분석, 어휘의미 분석(동음이의어 분석, 다의어 분석), 개체명 인식, 의존 구문분석, 의미역 인식 결과를 제공합니다.

언어분석 기술은 문어체 대상 기술 및 구어체 대상 기술로 구분되며, 문어체의 경우 형태소분석, 개체명인식, 어휘의미분석(동형이의어분석, 다의어분석), 의존구문분석, 의미역인식 6종 기술을, 구어체의 경우 형태소분석, 개체명인식 2종 기술을 제공합니다.

형태소 분석 API
(문어/구어)

형태소 분석 API는 자연어 문장에서 의미를 가진 최소 단위인 형태소(명사, 동사, 형용사, 부사, 조사, 어미 등)를 분석하는 기술로, 태그셋은 세종 태그셋을 기반으로 한 [TTA 표준 형태소 태그셋 \(TTAK.KO-11.0010/R1\)](#)에 기반합니다.

```
def extract_tags(json_string, tags):  
    # Convert JSON string to dictionary  
    parsed_json = json.loads(json_string)  
  
    # Initialize a list to collect the extracted words  
    extracted_words = []  
  
    # Extract the sentence objects that might contain the tags  
    if 'sentence' in parsed_json['return_object']:  
        sentences = parsed_json['return_object']['sentence']  
  
    # Iterate through each sentence and then through each morpheme in the sentence  
    for sentence in sentences:  
        if 'morp' in sentence:  
            for morpheme in sentence['morp']:  
                # If the morpheme's type is in the specified tags, append the word to the list  
                if morpheme['type'] in tags:  
                    extracted_words.append(morpheme['lemma'])  
  
    return extracted_words  
  
tags_of_interest = ['NNG', 'VA', 'MAG', 'VV'] #일반명사, 일반동사, 형용사, 부사  
  
# Ensure the dataframe is named appropriately; assuming your dataframe is named 'output'  
output['extracted'] = output['output'].apply(lambda x: extract_tags(x, tags_of_interest))
```

문장에서 명사, 동사, 형용사, 부사 추출

형태소 분석 및 단어 추출 결과

<형태소 분석 및 단어 추출 결과>

ct_1_content	select_2_title	select_2_content	select_3_title	select_3_content	txt	sentiment	output	extracted
성에 좋아요	피부고민	보습에 좋아요	자극도	자극없이 순해요	세안 후 자기전 바르는 로션이 에요.wn 매번 다 떨어지 면 구매 하곤 하 는 로션 이라 자 주...	Positive	{"result":0,"return_object": {"doc_id":"","DCT"...	[세안, 후, 전, 로션, 매번, 다, 구매, 로션, 자주, 구매]
에 좋아요	피부고민	보습에 좋아요	자극도	자극없이 순해요	신랑 로 션으로 올인원 을 바르 는게 좋 아해서 wn늘 행 사할때 가성비 템으로 구매하 곤 해요!	Positive	{"result":0,"return_object": {"doc_id":"","DCT"...	[신랑, 로 션, 올, 인 원, 늘, 행 사, 때, 가 성비템, 구 매]
에 좋아요	피부고민	보습에 좋아요	자극도	자극없이 순해요	친구가 올인원 제품 필 요하다 고 해서 구매해 준 제 품!wnwn 수부지 (추정) 타 입이라 ...	Positive	{"result":0,"return_object": {"doc_id":"","DCT"...	[친구, 올, 인원, 제품, 필요, 구매, 제품, 수부 지, 추정, 타입, 산뜻 하, ...]

1. txt: 구매자가 남긴 원본 리뷰

2. output: 형태소 추출한 결과

3. extracted: 동사,명사,형용사, 부사 추출

TF-IDF 벡터화 후 평점 Bi-gram 키워드 추출

<5점 이상평점 제품 핵심 키워드>

자극 없이 순하
피부 진정 효과 꾸준히 사용
스킨 케어 추천
크림 바르
수분 크림
건성 피부
피부 타입
제품 사용
진정 효과
꾸준히 사용
구매 의사

<4점 이상 5점 이하평점 5점 이상 제품 핵심 키워드>

자극 없이 순하
피부 진정 효과 꾸준히 사용
스킨 케어 추천
크림 바르
수분 크림
건성 피부
피부 타입
제품 사용
진정 효과
꾸준히 사용
구매 의사

워드
클라우드

<1점 이상 2점이하 평점 제품 핵심 키워드>

트러블 올라오
피부 진정 효과
자극 없이
수분 크림
구매 의사
효과 전혀
민감 피부
자극 없이
제품 사용
지성 피부
피부 타입
피부 예민하
스킨 케어
바르 피부
제품 사용
지성 피부
피부 타입
피부 예민하
스킨 케어

<1점 이하 평점 제품 핵심 키워드>

트러블 올라오
피부 진정 효과
자극 없이
수분 크림
구매 의사
효과 전혀
민감 피부
자극 없이
제품 사용
지성 피부
피부 타입
피부 예민하
스킨 케어
바르 피부
제품 사용
지성 피부
피부 타입
피부 예민하
스킨 케어

TF-IDF 벡터화 후 평점 Bi-gram 키워드 추출

<Category별 제품별 긍정, 부정 핵심 키워드(로션)>

```
Category: 로션, Ranking: 1, Positive
인원 제품      2.920679
남자 친구      1.791217
스킨 로션      1.283893
바르 편하      1.187239
인원 바르      1.128718
Name: Scaled_Weight, dtype: float64
-----밀은 단점-----
Category: 로션, Ranking: 1, Negative
피부 뒤집어지   0.126491
둘려놓 피부    0.063246
뒤집어지 바로  0.063246
뒤집어지 어떨  0.063246
바로 쓰레기    0.063246
Name: Scaled_Weight, dtype: float64
```

<Category별 제품별 긍정, 부정 핵심 키워드(에센스)>

```
Category: 에센스/세럼/앰플, Ranking: 2, Positive
마녀 공장      3.597185
꾸준히 사용    1.966420
흡수 빠르      1.514400
자극 없이      1.502313
바르 피부      1.327728
Name: Scaled_Weight, dtype: float64
-----밀은 단점-----
Category: 에센스/세럼/앰플, Ranking: 2, Negative
사용 중단      0.122353
모공 효과      0.108090
올라오 사용    0.098657
느끼 바르      0.089443
따끔거리 자극  0.089443
```

<Category별 제품별 긍정, 부정 핵심 키워드 핵심 키워드(스킨)>

```
Category: 스킨/토너, Ranking: 1, Positive
자작 나무      3.364683
자극 없이      1.615438
수분 토너      1.247247
나무 수분      1.119684
없이 순하      1.011792
Name: Scaled_Weight, dtype: float64
-----밀은 단점-----
Category: 스킨/토너, Ranking: 1, Negative
민감 피부      0.097575
산성 파랑      0.081650
세럼 이상      0.081650
암모니아 냄새  0.081650
이상 암모니아  0.081650
Name: Scaled_Weight, dtype: float64
```

<Category별 제품별 긍정, 부정 핵심 키워드 핵심 키워드(미스트)>

```
Category: 미스트/오일, Ranking: 1, Positive
녹스 오일      2.997036
머리 카락      1.493291
사용 너무      1.128188
너무 건조      1.098034
뿌리 너무      0.917579
Name: Scaled_Weight, dtype: float64
-----밀은 단점-----
Category: 미스트/오일, Ranking: 1, Negative
난들 주변      0.075593
노즐 타입      0.075593
마음 난들      0.075593
분사 노즐      0.075593
여간 불편      0.075593
Name: Scaled_Weight, dtype: float64
```


Chat- prompt & fine-tuning

<Prompting>

```
'messages': [  
  {"role": "system",  
   "content": "너는 추천시스템 챗봇이야. 실제 대화처럼 60token 내외로 답을하고 친근한 구어체를 사용해줘. 그리고 유사한 제품을 추천해 주고  
   "rate보다 final_score가 높으면서 final_score 5점이상이고 그 추천한 제품의 특징을 설명해줘"}]
```

Prompt: 모델이 어떻게 응답할지 제어(명령)

```
"messages": [  
  {"role": "user",  
   "content": "이 제품과 유사한 제품 3개만 추천해줘."}]
```

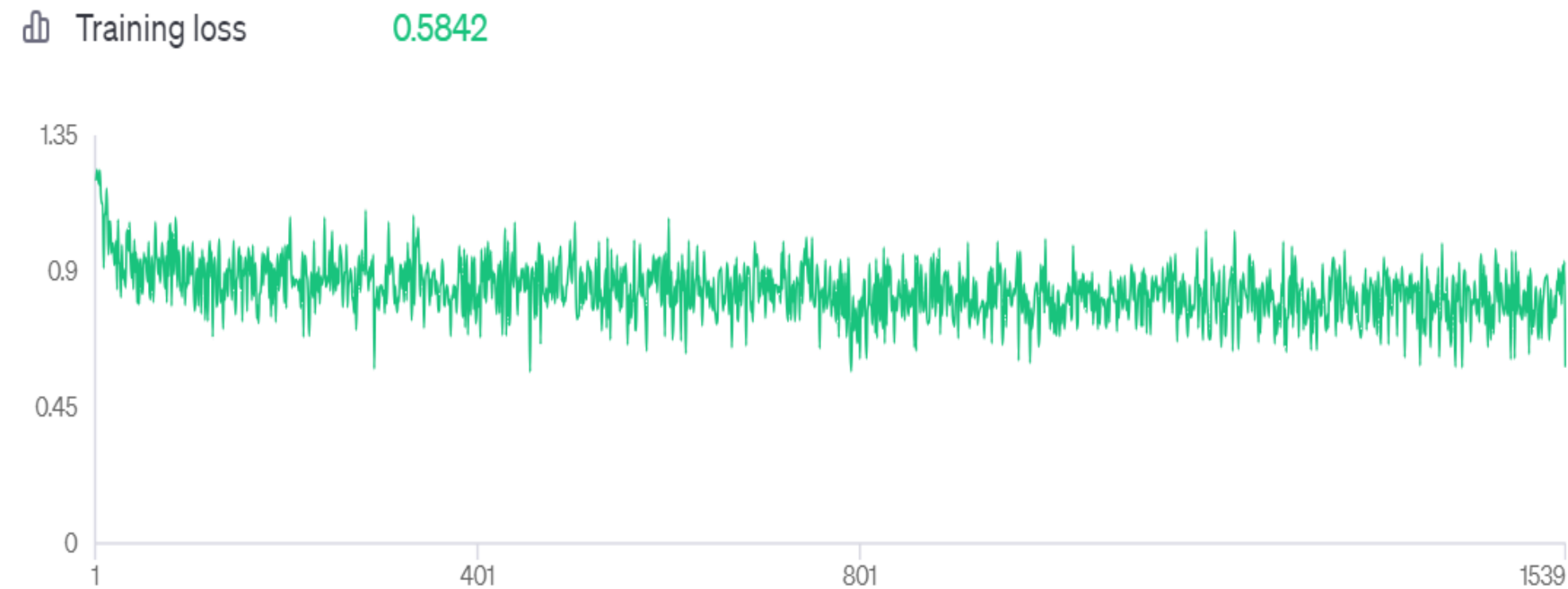
query: 특정제품과 유사한 제품을 추천 질문

```
{"role": "assistant",  
 "content": "f0이 {name} 제품의 특징은 {select_1_content}, {select_2_content}, {select_3_content}이고, 기존 평점은 {rate}이지만 최종평점은 {final_score}입니다.  
최종 평점이 높은 제품을 추천해줘 그리고 제품의 리뷰는 {txt}입니다."}]
```

대답: 질문에 답변 형식

Chat-gpt fine-tuning 학습

<fine-tuning 학습 결과>



Chat-gpt fine-tuning 학습

```
[34] ##### 영어로 추천시스템(10000개 데이터로 모델 파인 튜닝) #####
import openai

openai.api_key = 'sk-proj-J8ISyFnhFRmj4RmltmqIT3BlbkFJzq9Pn44eJNcWjLLKYem0' # Replace 'your-api-key' with your actual OpenAI API key
completion = openai.ChatCompletion.create(
    model="ft:gpt-3.5-turbo-1106:personal:test-realim:9cSQqC9G", # Update this to the correct model identifier
    messages=[
        {"role": "system", "content": "You are a chatbot that identifies reviews and strengths of cosmetics, recommends products with similar performance, and explains why you recommend that product."},
        {"role": "user", "content": "AHC OnlyForMan Pore Fresh All-in-One Essence Double Planning"}
    ]
)

# Directly print the content of the message, decoding Unicode escapes if necessary
message_content = completion.choices[0].message['content']
print(message_content.encode('utf-8').decode('unicode_escape'))
```

➡ Based on the analysis of the provided review, I think the product you would like is [AHC OnlyForMan Pore Fresh All-in-One Essence Set](<https://www.sephora.com/product/ahc-onlyforman-pore-fresh-all-in-one>)

The strengths of this product are Brightening, Good Value, Lightweight, Non-greasy, Absorbs Quickly, and Lasting Hydration.

I recommend [AHC OnlyForMan Pore Fresh All-in-One Essence Set](<https://www.sephora.com/product/ahc-onlyforman-pore-fresh-all-in-one-essence-set-P471495>) because it has similar strengths and also because

특정제품을 추천해주고 추천해주는 이유를 설명이 가능(다음페이지 계속)

Chat-gpt fine-tuning 학습

<입력 ITEM>



AHC >

AHC 온리포맨 올인원에센스 120ml

38,000원

👁 1명이 보고있어요

배송정보 ⓘ

일반배송 | 2,500원 (20,000 원 이상 무료배송)
올리브영 배송 | 평균 3일 이내 배송

결제혜택

THE CJ 카드 추가 10%할인 ⓘ

CJ ONE 포인트 최대 1% 적립 예상 ⓘ

구매수량

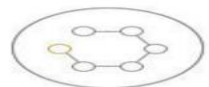
-

1

+

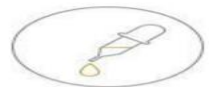
상품금액 합계

38,000원



모든 케어를 한번에,
All-In-One 에센스

스킨케어 단계를 하나로 줄여 모든 남성들의 피부
고민들을 한번에 해결해주는 올인원 에센스입니다.



집중 영양을 공급해주는
블랙 컴플렉스 함유

피부에 부족한 영양을 공급해주어 탄탄하고
매끈한 피부결로 관리해줍니다.



보습 및 진정 케어해주는
모이스트 카밍 컴플렉스 함유

모이스트 카밍 컴플렉스가 피부를 편안하게
진정시켜주고 보습 효과를 부여합니다.

<추천 ITEM>



SSG.COM 뷰티

신세계백화점

AHC >

AHC 온리포맨 포어 프레쉬 올인원 에센스 기획세트

★★★★★ 4.5 (2건) 🍌

5% 36,100원 38,000원

할인내역 ⌵

SHINSEGAE
UNIVERSE

유니버스 클럽은 가입하자마자 3만원에
매월 최대, 11만원 할인까지

신세계 유니버스 클럽 시작하기 >

카드혜택가 36,100원 ⌵

무이자 카드사별 무이자 혜택 ⌵

상품쿠폰

15% 상품쿠폰

최대 1만원

시코르위크 15% 상품쿠폰
3만원 이상 구매 시 사용가능(-06/23)



ID당 2매

상품쿠폰 유의사항 (?)

Chat-gpt fine-tuning 학습

```
##### 영어로 추천 시스템(10000개 데이터로 모델 파인 튜닝) #####
```

```
import openai
```

```
openai.api_key = 'sk-proj-J8ISyFnhFRmj4RmltmqiT3BIbkFJzq9Pn44eJNcWjLLKYem0' # Replace 'your-api-key' with your actual OpenAI API key
```

```
completion = openai.ChatCompletion.create(
```

```
    model="ft:gpt-3.5-turbo-1106:personal:test-realm:9cSQqC9G", # Update this to the correct model identifier
```

```
    messages=[
```

```
        {"role": "system", "content": "You are a chatbot that identifies reviews and strengths of cosmetics, recommends products, and explains why you recommend that product."},
```

```
        {"role": "user", "content": "Could you recommend similar to Round Lab 1025 Dokdo Lotion?"}
```

```
    ]
```

```
)
```

```
# Directly print the content of the message, decoding Unicode escapes if necessary
```

```
message_content = completion.choices[0].message['content']
```

```
print(message_content.encode('utf-8-sig').decode('unicode_escape'))
```

```
I»ﺯ I recommend the A'pieu Madecassoside Fluid as a similar product to the Round Lab 1025 Dokdo Lotion. Both of these products are hydrating lotions suitable for all skin types. They contain mild formula
```

I recommend the A'pieu Madecassoside Fluid as a similar product to the Round Lab 1025 Dokdo Lotion.

Both of these products are hydrating lotions suitable for all skin types.

They contain mild formulations that are ideal for sensitive or easily irritated skin.

If you like the Round Lab 1025 Dokdo Lotion, you may also enjoy trying the A'pieu Madecassoside Fluid for its lightweight and soothing properties.

특정제품을 추천해주고 추천해주는 이유를 설명이 가능(다음페이지 계속)

Chat-gpt fine-tuning 학습

<입력 ITEM>



라운드랩 >

라운드랩 1025 독도 로션 (본품200ml+50ml 추가 증정)

20,000원 **17,900원** 혜택 정보

세일 오늘드림

3명이 보고있어요

배송정보

일반배송 | 2,500원 (20,000 원 이상 무료배송)
올리브영 배송 | 평균 3일 이내 배송

오늘드림 | 2,500원 또는 5,000원

픽업 | 배송비 조건 없음

결제혜택

THE CJ 카드 추가 10%할인

CJ ONE 포인트 최대 1% 적립 예상

<추천 ITEM>

Mireclim
한국코스메틱 Korean Cosmetics

A'pieu

해외직구 원학이커머스1

APIEU MADECASSOSIDE 아피유 매카소사이드
플루이드 2X 300ml

★★★★★ (0)

14% 71,000원
61,060원

카드사별 무이자할부 / 최대 5개월

해외배송 상품 배송정보

무료배송 | 우체국택배

스마일카드 최대 2% 캐시 적립

신세계포인트 적립

카드추가혜택

본 상품은 국내배송만 가능합니다

원산지 상세설명 참조

할인가능 쿠폰 : (1)장

쿠폰받기

연구 한계

연구 한계

1. **Cost** 문제로 인한 수집한 데이터를 전부 학습 불가능
2. 정확하게 무슨 이유로 특정제품을 추천해 주는지에 대한 의문
3. 학습데이터를 잘 활용하는지에 대한 의문

추후 연구 진행이 더필요

THANK YOU