인공지능 데이터 전처리 결과서

1. 개요

- **목적**
  - 사용 하려는 데이터는 아마존의 리뷰 및 메타 데이터로 굉장히 용량이 크고 많은 량의 Text데이터가 들어있어 모델이 학습하기에 너무 많은 양임.
  - 범위를 정하여 데이터를 축소하는 과정이 필요함
  - 학습시에 필요한 데이터만을 추출하는 과정이 필요함
  - 추출한 데이터에서 학습 과정시 물리적 환경을 고려하여 요약이 필요함

2. 데이터 설명

- **출처**
  - **Julian McAuley LAB - University of California San Diego**에서 수집한 아마존의 리뷰 및 메타데이터

    https://amazon-reviews-2023.github.io/
  - 연구소에서 수집하여 공개된 데이터로 활용 가능하며 kaggle에서도 다운로드 가능
- **특성**
  - 전체 데이터 크기 : 376GB, 34개의 카테고리

Category	Item Meta Size (MB)	Review Size (MB)	TotalSize (MB)
All_Beauty	213.00	327.00	540.00
Amazon_Fashion	1,454.08	1,075.20	2,529.28
Appliances	285.00	929.00	1,214.00
Arts_Crafts_and_Sewing	2,219.84	4,042.80	6,262.64
Automotive	5,478.40	8,900.32	14,378.72
Baby_Products	691.00	3,018.80	3,709.80
Beauty_and_Personal_Care	2,910.08	11,264.00	14,174.08
Books	15,059.20	20,518.40	35,577.60
CDs_and_Vinyl	949.00	3,360.16	4,309.16
Cell_Phones_and_Accessories	4,116.48	9,556.16	13,672.64

Category	Item Meta Size (MB)	Review Size (MB)	TotalSize (MB)
Clothing_Shoes_and_Jewelry	18,432.00	28,441.60	46,873.60
Digital_Music	67.10	78.80	145.90
Electronics	5,376.00	23,104.00	28,480.00
Gift_Cards	2.04	50.20	52.24
Grocery_and_Gourmet_Food	1,411.20	6,104.68	7,515.88
Handmade_Products	399.00	289.00	688.00
Health_and_Household	2,528.48	11,686.40	14,214.88
Health_and_Personal_Care	118.00	227.00	345.00
Home_and_Kitchen	12,099.20	32,153.60	44,252.80
Industrial_and_Scientific	1,155.52	2,403.20	3,558.72
Kindle_Store	7,024.48	16,179.20	23,203.68
Magazine_Subscriptions	4.10	33.30	37.40
Movies_and_TV	1,318.56	8,573.76	9,892.32
Musical_Instruments	632.00	1,597.44	2,229.44
Office_Products	2,201.60	5,904.64	8,106.24
Patio_Lawn_and_Garden	2,774.24	7,925.00	10,699.24
Pet_Supplies	1,606.08	8,576.00	10,182.08
Software	256.00	1,912.30	2,168.30
Sports_and_Outdoors	4,224.32	9,464.32	13,688.64
Subscription_Boxes	1.40	8.95	10.35
Tools_and_Home_Improvement	4,974.40	13,107.20	18,081.60
Toys_and_Games	2,703.36	7,484.16	10,187.52
Video_Games	675.00	2,743.04	3,418.04
Unknown	437.00	30,637.60	31,074.60

- 각 카테고리별 데이터 특성
  - Review

필드	타입	설명
rating	숫자형(float)	평점 1.0-5.0
title	문자형(str)	리뷰 제목
text	문자형(str)	리뷰 내용
images	문자형 배열(list)	유저가 올리는 제품의 실제 이미지각 이미지 사이즈는 상이하고

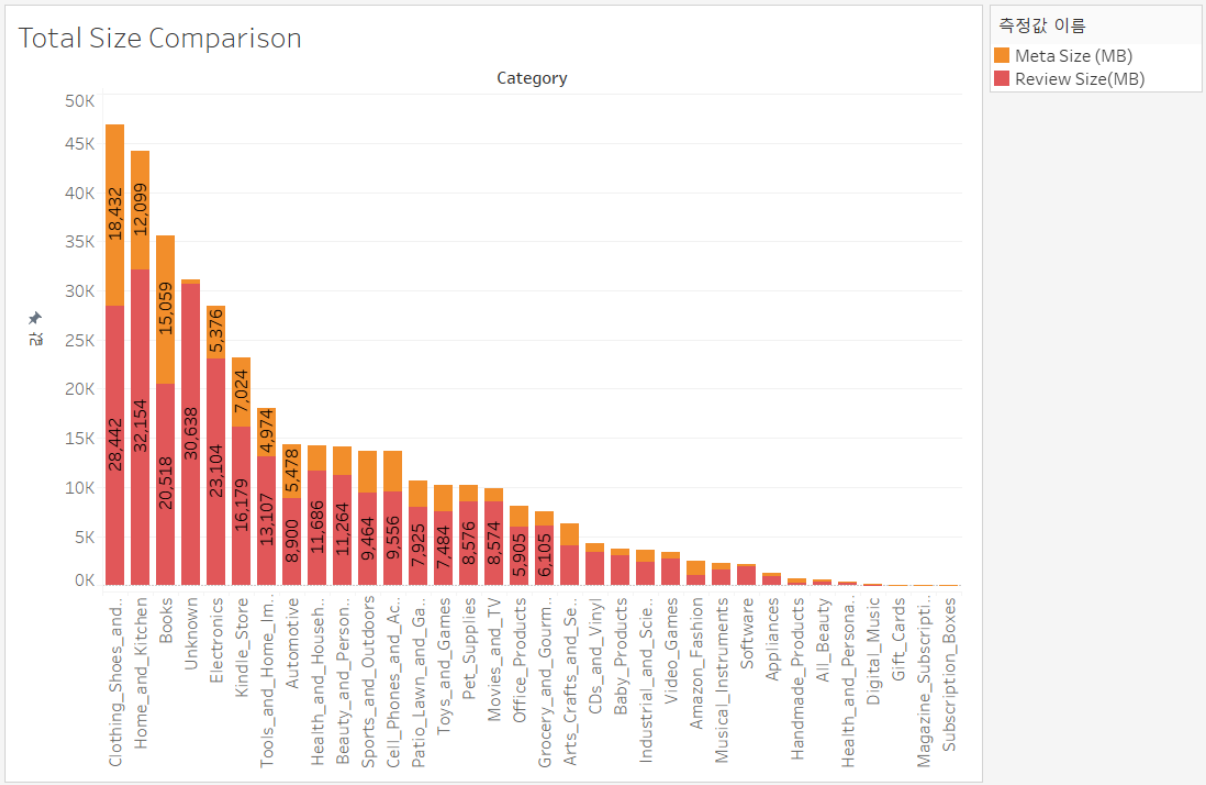
필드	타입	설명
		(small, medium, large), small_image_url, medium_image_url, large_image_url로 표기
asin	문자형(str)	제품 ID
parent_asin	문자형(str)	제품의 상위 ID 다른 색, 스타일, 사이즈들의 제품들은 보통 같은 상위 ID
user_id	문자형(str)	리뷰어 ID
timestamp	숫자형(int)	리뷰한 시간 (유닉스 시간)
verified_purchase	bool	구매 확인
helpful_vote	숫자형(int)	리뷰의 유용성 투표

- Meta

필드	타입	설명
main_category	문자형(str)	제품의 메인 카테고리
title	문자형(str)	제품 이름
average_rating	숫자형(float)	제품 페이지에 표기되는 평균 평점
rating_number	숫자형(int)	평점 수
features	문자형 배열(list)	글머리표 형식으로 된 제품 특징
description	문자형 배열(list)	제품 설명
price	숫자형(float)	크롤링 당시의 제품 가격 (US $)
images	문자형 배열(list)	제품 이미지들각 이미지의 사이즈가 상이하고 이미지의 위치는 “variant”필드로 표기
videos	문자형 배열(list)	제목과 url이 포함되어있는 제품 비디오들
store	문자형(str)	판매처 이름
categories	문자형 배열(list)	제품의 계층적 분류
details	dict	제품 상세 설명 (재질, 브랜드, 크기 등)
parent_asin	문자형(str)	제품의 상위 ID
bought_together	문자형 배열(llist)	웹사이트에서 추천하는 같이사면 좋은 제품들

두 데이터의 주요 특징은 대부분 문자형이며 결측값은 “”처럼 빈칸인 경우나 특수문자가 들어간 경우임.

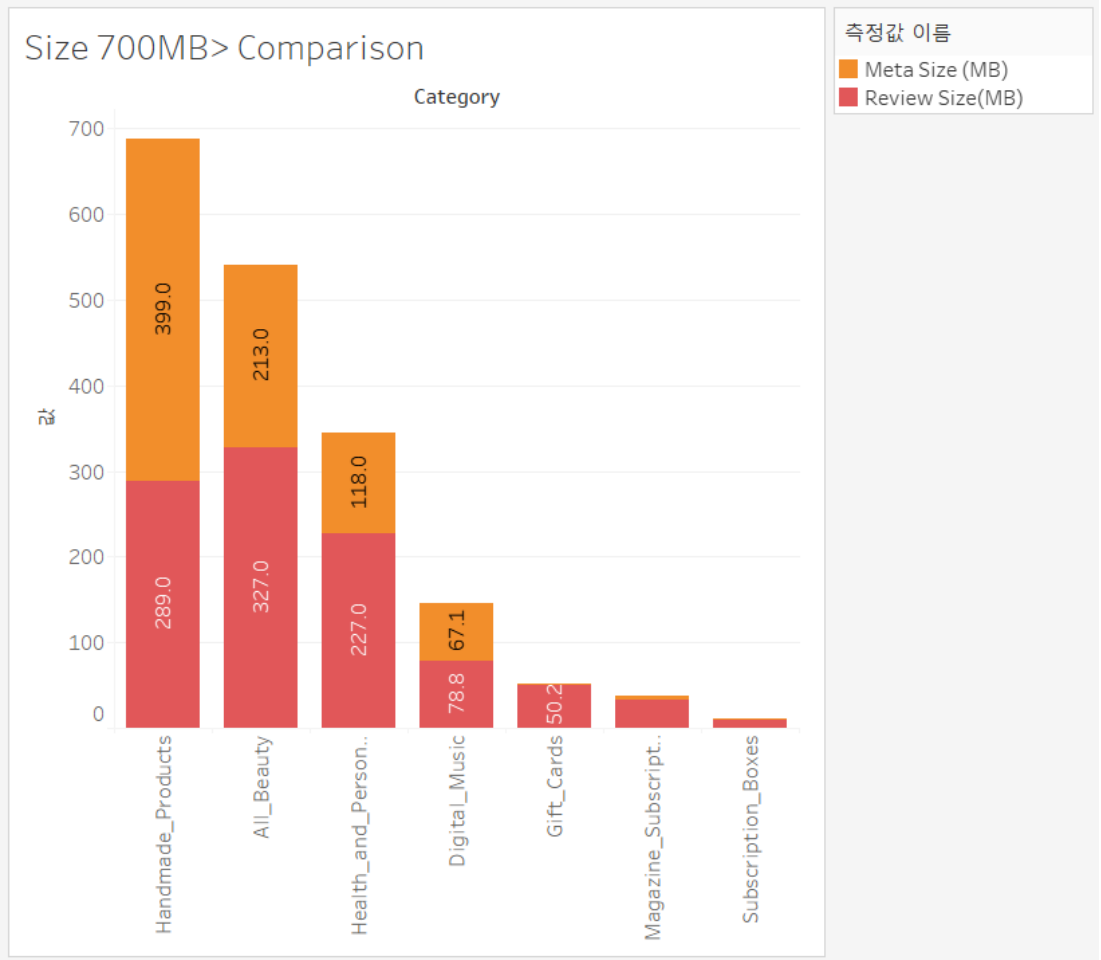
사용하려는 모델은 대부분 임베딩 과정을 거치기 때문에 전부 문자형으로 입력되어야 함



- 각 카테고리별 **Meta**데이터의 크기와 **Review**데이터의 크기
- 가장 큰 데이터는 **45GB이상**이고 작은 데이터는 **0.1GB**정도임 **모든** 카테고리를 전부 사용하기에는 문제가 있음
- 한 카테고리를 정하여 사용하기로 함

3. 카테고리 선정과정

- **Sample 데이터**는 크기가 작은 **gift_data**를 사용하여 진행, **50mb**정도의 데이터로 먼저 시범 학습하여 카테고리를 정하기로함
- **RTX 3070** 환경에서 33시간 소요(전체 데이터 학습 시)
  - 향후 **확장성**을 위해 약간 더 큰 데이터를 사용하되 너무 크지 않은 것으로 진행하기로 함
  - **0.2GB < X < 0.7GB**로 정하여 진행



- 후보는 **Handmade_products, All_Beauty, Health_and_Personal Care** 세가지가 되었음.
- **Handmade_products** : 수공예품 자체에서 다양한 상품이 존재 서로가 연관이 적은 경우 추천이 어렵다고 판단하여 제거
- **Health_and_Personal_Care** : 사용자의 특성을 고려하지 못 한채로 추천시에 문제가 생길 수도 있다고 판단하여 제거
- **All_Beauty** : 화장품은 추천 방향도 명확하다고 판단함

선택된 카테고리 : All_Beauty(meta: 213mb, review:327mb)

4. 전처리 과정

- **학습시에 필요한 데이터**
  - 유저와 상품의 연결 관계
  - 상품의 제목과 정보
- meta와 review에서 필요한 정보들을 전처리하여 후에 추출하여 새로운 파일을 만듦

결측치 처리

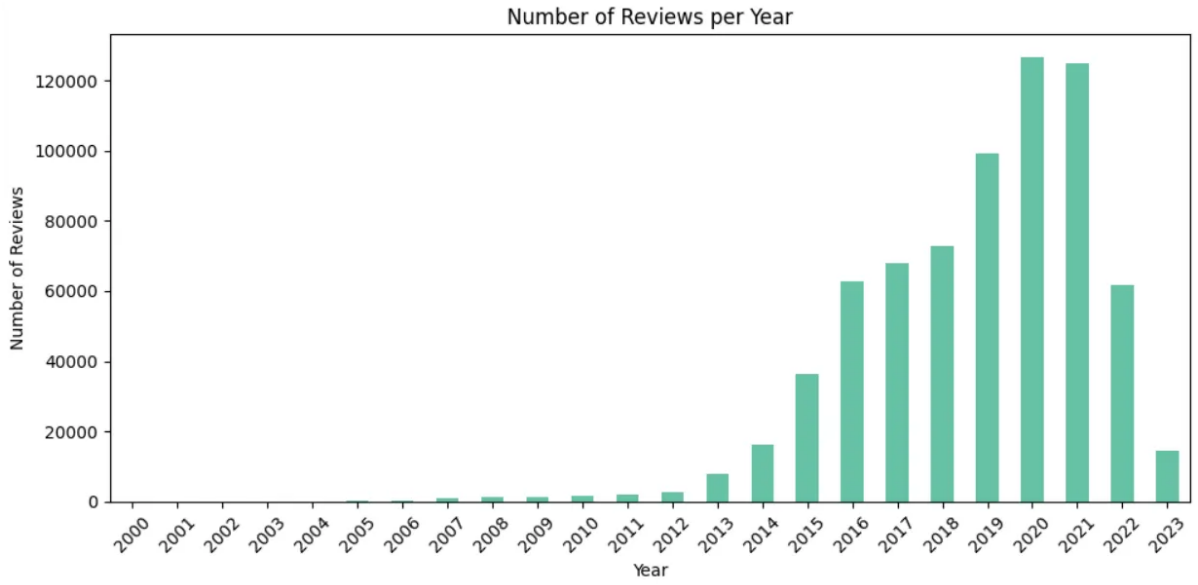
`description`, `title` 결측치 처리

```
try:
    # 설명 처리
    if 'description' in meta_dict[asin] and len(meta_dict[asin]['description']) == 0:
        name_dict['description'][itemmap[asin]] = 'Empty description'
    else:
        name_dict['description'][itemmap[asin]] = meta_dict[asin].get('description', ['Empty description'])[0]

    # 원본 타이틀 가져오기
    original_title = meta_dict[asin].get('title', 'No title')

    # 원본 타이틀을 name_dict에 할당
    name_dict['title'][itemmap[asin]] = original_title

except KeyError:
    pass
```

- `description` 결측치에는 'No description'로, `title` 결측치에는 'No title'로 대체해 결측치 처리

범위 제한

1. 기간 제한



- 총 1,416,231,267개의 리뷰 중 2000-2012년도에는 각 5,000개 미만의 리뷰가 존재
- All_Beauty의 전체 사이즈를 더 줄이고, 판매 중지된 아이템에 대한 추천을 방지하기 위하여 2013.01 - 2023.08 기간으로 데이터의 범위 축소

3. 리뷰 수 하한선 제한

- Threshold 값 설정 K = 12
  - 실질적인 구매 가능성이 높은 제품 추천을 위해 리뷰가 12개 이상인 아이템만 포함.
  - 개인 취향 반영을 위해 리뷰를 12개 이상으로 쓴 유저만 포함
- 추후에 K값을 낮춰가면서 데이터량을 증가시켜 학습 시킬 예정

데이터 가공

user_id, item_id 재설정

```
if rev in usermap:
    userid = usermap[rev]
else:
    usernum += 1
    userid = usernum
    usermap[rev] = userid
    User[userid] = []

if asin in itemmap:
    itemid = itemmap[asin]
else:
    itemnum += 1
    itemid = itemnum
    itemmap[asin] = itemid
User[userid].append([time, itemid])
```

- 복잡한 user_id, item_id 1부터 정수로 재설정

제품 이름 요약

```python
try:
    # OpenAI API 호출하여 텍스트 요약
    chat_completion = client.chat.completions.create(
        model=model,
        messages=[
            {
                "role": "system",
                "content": "You are an extremely efficient title summarizer. Your task is to summarize the provided text by
            },
            {
                "role": "user",
                "content": f"Here is the title: {title}. Summarize it by focusing on the most important product name and ess
            }
        ],
        max_tokens=30,
        temperature=0.7,
    )
    summary = chat_completion.choices[0].message.content.strip()
    return summary

except Exception as e:
    print(f"Error during GPT API call: {e}")
    return title  # 에러 발생 시 원래 타이틀 반환
```

- 긴 아이템 이름 ⇒ 긴 임베딩 ⇒ 연산 증가 ⇒ 시간 증가의 문제를 해결하기 위해 OpenAi로 제품 이름을 줄여 시간 축소

5. 전처리 결과 및 모델 영향

전처리 결과

All_Beauty_text_name_dict.json.gz - 234KB

필드	타입	설명
item_id	int	1부터 재설정한 item id
title	str	제품 이름
description	str	제품 설명

All_Beauty.txt - 95KB

필드	타입	설명
user_id	int	1부터 재설정한 유저 id
item_id	int	1부터 재설정한 상품 id

전처리 전 후 데이터 크기

540MB → 329KB

전처리 진행과정중 모델 성능 변화

점수가 적히지 않은 모델은 학습이 완료되지 못하거나 문제가 생긴 경우임

모델	num_user	num_item	description	eval
A-LLMRec-20240930	729	586	샘플 모델 (gift_card)	0.1875
A-LLMRec-2024-10-08 3rd	3389	7055	raw data 사용	
A-LLMRec-2024-10-08 3rd-2	4	4	날짜 범위 축소: 2023.07.01-2023.08.312달치 데이터 사용	
A-LLMRec-2024-10-08 3rd-3	3377	7031	날짜 범위 확대: 2010.01.01-2023.08.3114년치 데이터 사용	
A-LLMRec-2024-10-08 4th	3233	6755	날짜 범위 축소: 2018.08.01-2023.08.315년치 데이터 사용	
A-LLMRec-2024-10-08 4th	729	586	날짜 범위 축소: 2020.08.01-2023.08.313년치 데이터 사용(gift card일 가능성 농후	
A-LLMRec-2024-10-08 4th-2	366	514	날짜 범위 축소: 2022.01.01-2023.08.312년치 데이터 사용	
A-LLMRec-2024-10-08 5th	165	279	날짜 범위 축소: 2022.08.01-2023.08.311년치 데이터 사용	
A-LLMRec-2024-10-08 6th	15	40	날짜 범위 확대: 2000.08.01-2023.08.3123년치 데이터 사용Theshold = 50	
A-LLMRec-2024-10-08 7th	114	366	Theshold = 18	
A-LLMRec-2024-10-08 9th	127	439	Theshold = 15, gpt로 제품 이름 요약 도입	
A-LLMRec-2024-10-09 1st	259	881	Theshold = 12 (디폴트)날짜 범위 축소: 2013.08.01-	0.07

모델	num_user	num_item	description	eval
			2023.08.3110년치 데이터 사용	
A-LLMRec-2024-10-12 1st	1588	4099	Theshold = 5 (데이터 양 늘리기)날짜 범위 확대: 2013.01.01-2023.08.3111년치 데이터 사용	0.12
A-LLMRec-2024-10-12 3rd	1588	4099	`description` 패딩처리 (=0), 프롬포트 수정	
A-LLMRec-2024-10-12 4th	1588	4099	`description` 에 `details` 내용 삽입	0.12
A-LLMRec-2024-10-15 1st	1588	4099	`description` 패딩처리 (=0), 프롬포트 수정	0.12

- 데이터에 대한 리뷰 수 하한선을 늘림으로써 성능(Hit@1)가 향상됨을 볼 수 있음
- 추후에 데이터 가공을 통해 더 성능을 높일 수 있을 거라 예상

6. 결론

이번 전처리 과정에서는 전체 Amazon 상품 데이터셋에서 사이즈 문제로 인해 리뷰기반 추천이라는 배경을 고려해 **All_Beauty** 카테고리를 선택하였습니다.

제품 설명과 이름인 `description`, `title` 필드에 각각 “Empty description”, “No title”로 대체하여 결측치 처리를 하였고, All_Beauty의 전체 사이즈를 더 줄이고, 판매 중지된 아이템에 대한 추천을 방지하기 위하여 2013.01 - 2023.08으로 기간을 설정하였습니다.

리뷰데이터와 리뷰가 12개 이상인 아이템만을 추출하여 분석을 수행했습니다.

리뷰데이터의 제한 값은 학습이 진행 되면서 점차 완화하면서 진행할 예정입니다.

복잡하였던 user_id와 item_id를 1부터 정수로 재설정 하였습니다.

또한 시간 증가의 문제를 해결하기 위해 gpt-4o-mini이름을 줄여 시간 축소하였습니다.

이는 데이터 전처리를 거치며 All_Beauty_title_sum_match.jsonl 파일을 생성해 요약 부분을 확인 할 수 있습니다.

전처리 후에는 모델에 사용되는 파일 All_Beauty_text_name_dict.json.gz, All_Beauty.txt 두개가 생성됩니다.

결과적으로, 이를 통해 메모리 사용량과 모델 소요 시간을 절감하고, 개인취향에 맞는 제품 추천이 가능하도록 데이터를 최적화했습니다.

7. 향후 작업

- **데이터량 증가**

    - 리뷰 갯수 하한선을 완화화면서 데이터량을 늘려나갈 예정
- **데이터 가공 시도**
    - 학습시 사용되는 description의 부분에 결측치가 너무 많아 중요한 정보가 제공되고 있지 않음 원래 데이터에서 추출하여 description에 추가하여 더 많은 정보를 제공하고자 시도할 예정