데이터 분석 보고서

단골 브랜드 데이터 활용 유사도 기반 추천 시스템

1. 개요
   -. 현 상황 진단
   -. 선행 연구 탐색 및 연구가설 설정

2. 분석 결과
   -. 추천 알고리즘 요약
   -. 데이터 현황
   -. 데이터 분석 결과
   -. 성능 평가

3. 결론

개요

개요

1. 현 상황 진단

〈'상품이 적다' VOC 지속 접수〉

바이어(특히 해외바이어)로부터
'상품(SKU)가 적다' 라는 VoC가 **지속적으로 접수**

〈검색어 결과 품질 이슈〉

바이어(특히 해외 바이어)의 경우 **검색 결과**를 **신뢰할 수 없어**
카테고리별 상품 탐색을 진행하는 경향이 있음

〈C레벨 희망 메인화면 개편 방향〉

카테고리를 넘어 **주제별 종합 큐레이션**이 가능한 **메인화면**이 **구성**되었으면 하는 방향성을 제시

목표 설정

카테고리를 넘어 **나와 비슷한 다른 바이어의** 연관 상품이 **나에게도 노출**되도록 **추천 알고리즘 구성** 및 제공

2. 선행 연구 검토 및 연구 가설 설정

1) 선행 연구 탐색

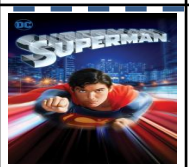
-. 컨텐츠 기반 필터링

① **아이템** 자체의 **특성**을 이용하여 **유사한 상품**을 추천하는 방법론

② 가령, 영화라고 한다면 영화에 대한 **'장르', '감독명', '출연 배우** ' 등이 **영화 A**라는 컨텐츠의 **특성**으로 작용

③ 컨텐츠 기반 필터링은 다음과 같은 **세 단계(부분)**으로 **일반화**가 가능함(S.M Mahdi Seyednezhad et.al, 2019)

- **Content Analyzer** : 컨텐츠의 특성을 분석하는 부분. Raw데이터를 정형화된 정보로 변환한다
- **Profile Learner** : Content Analyzer에서 추출한 컨텐츠의 특성을 추전 목표인 유저에 맞게 재가공하는 부분
- **Filtering Component** : Profile Learner에서 나온 user Profile 기반으로 가장 유사한 상품을 추천하는 부분. 다양한 유사도 측정 방법론 활용 가능

		블록버스터	아동	SF	Over 1 million
선택영화		**Yes**	No	No	**Yes**
		No	No	**Yes**	**Yes**
		No	No	No	No
추천영화		**Yes**	No	No	**Yes**

유사도 높음

〈컨텐츠 기반 필터링 예시〉

2. 선행 연구 검토 및 연구 가설 설정

1) 선행 연구 탐색

-. 협업 필터링

① **같은 영역**에 존재하는 **다른 공간**의 **데이터**를 이용하여 추천 사항을 추론하는 방법론

② 가령, **'유저의 구매 내역 '** 이라는 **같은 영역**에 존재하는 **'유저A'와 '유저B'**의 경우, 둘이 **충분히 유사**하다면 **A에게 B**의 **내역**을 **추천** 가능

③ 협업 기반 필터링은 아래와 같이 분류 가능(S.M Mahdi Seyednezhad et.al, 2019)

- **메모리 기반 협업 필터링** : User - Item 행렬에서만 작동. 추천 직전에 모든 평점을 직접적으로 업데이트하여 활용(배치성 활용)
- **모델 기반 협업 필터링** : 학습 가능한 파라미터를 가진 모델을 활용하여 학습한 패턴을 기반으로 추천을 수행 (Paul Convington, Jay Adams and Emre Sargin, 2016)

<협업 필터링 예시>

2. 선행 연구 검토 및 연구 가설 설정

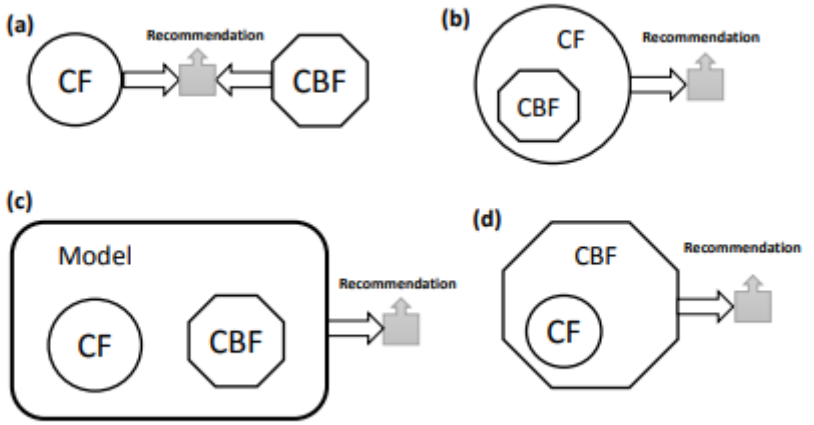
1) 선행 연구 탐색

-. 하이브리드 방법론

① **컨텐츠 기반 필터링**과 **협업 기반 필터링**은 각자 **고유**한 **장단점**을 보유

② 주류 추천시스템은 **협업 기반 필터링**과 **컨텐츠 기반 필터링**을 **결합(Hybrid)**하여 추천을 수행(Keunho Choi et.al)

③ 우측 그림에 대한 설명은 아래와 같음
- **(a)의 경우** : **협업 필터링**과 **컨텐츠 기반 필터링**의 결과물을 **가중치**를 통해 결합
- **(b)의 경우** : **컨텐츠 기반 필터링**을 통해 도출된 **컨텐츠 특성**을 기반으로 **협업 기반 필터링**이 **유저간 유사도**를 측정
- **(c)의 경우** : **협업 필터링**과 **컨텐츠 기반 필터링**의 Output을 또다른 **예측 모델**에 **Input**으로 투입
- **(d)의 경우 : 컨텐츠 기반 필터링**의 **User Profile 과정**에 **협업 필터링의 결과물**을 활용하여 품질을 향상



〈하이브리드 방식 개념도, S.M Mahdi Seyednezhad et.al, 2019〉

2. 선행 연구 검토 및 연구 가설 설정

2) 가설 설정

-. 가설 1 : 하이브리드 방법론 (d)를 차용한 추천 시스템은 단순 user-user 협업 필터링에 비해 실제 서비스에 활용 가능한 준수한 성능을 보일것이다

-. 가설 2 : 유저의 단골브랜드 추가 내역, 위시리스트 상품 추가 내역, 구매 내역을 모두 활용한 모델은 준수한 성능을 보일 것이다

본문

본문

1. 추천 알고리즘 요약

1) 추천 A-1안		2) 추천 A-3안
-. **공통점** ① **유저 - 유저 협업 필터링**인 '유저 프로파일링'단계 존재 : **단골브랜드 기반**으로 유저간 유사도가 가장 높은 **Top - K (혹은 > threshold)** 바이어를 **후보 바이어로** 추출 ② **공통 카테고리 추출** 단계 존재 : **Target Buyer**의 **상품 카테고리**와 **일치**하는 후보 바이어들의 상품들을 **후보 상품**으로 추출 ③ **상품 컨텐츠 기반 필터링** 존재 : 상품의 특성(**위시리스트를 추가한 바이어**)을 기반으로 **Target Buyer의 상품별**로 가장 **유사도**가 **높은** 후보 상품 **Top - K (혹은 > Threshold)** 상품들을 최종 도출		
-. **차이점(A-1안) : 구매 내역** ① Target Buyer의 상품과 후보 상품을 '공통 카테고리 추출 단계', '상품 컨텐츠 기반 필터링' 단계에서 추출 / 계산할 때 각 바이어의 **구매 내역 기반**으로 동작	**VS**	-. **차이점(A-3안) : 위시리스트** ① Target Buyer의 상품과 후보 상품을 '공통 카테고리 추출 단계', '상품 컨텐츠 기반 필터링' 단계에서 추출 / 계산할 때 각 바이어의 **위시리스트 추가 내역 기반**으로 동작

2. 데이터 현황

1) 데이터셋 정의

번호	컬럼명	타입	설명
1	brandId	Integer	브랜드ID
2	BuyerId	Integer	바이어ID
3	Yearmon	Datetime	추가일자
4	Value	Integer	내부 처리용(1 고정)

〈단골브랜드 추가 내역〉

번호	컬럼명	타입	설명
1	brandId	Integer	브랜드ID
2	buyerId	Integer	바이어ID
3	productid	Integer	상품ID
4	createdAt	Integer	추가일자

〈위시리스트 추가 내역〉

번호	컬럼명	타입	설명
1	brandId	Integer	브랜드ID
2	BuyerId	Integer	바이어ID
3	productid	Integer	상품ID
4	initQuantity	Integer	내부 처리용(1 고정)
5	Status	Integer	주문 상태
6	createdAt	Datetime	주문 일자

〈주문 내역〉

번호	컬럼명	타입	설명
1	productId	Integer	브랜드ID
2	name	String	상품명
3	Category	Array	상품 카테고리 목록
4	Images	Json	상품 이미지 목록

〈상품 목록〉

3. 데이터 분석 결과

0) 데이터 EDA

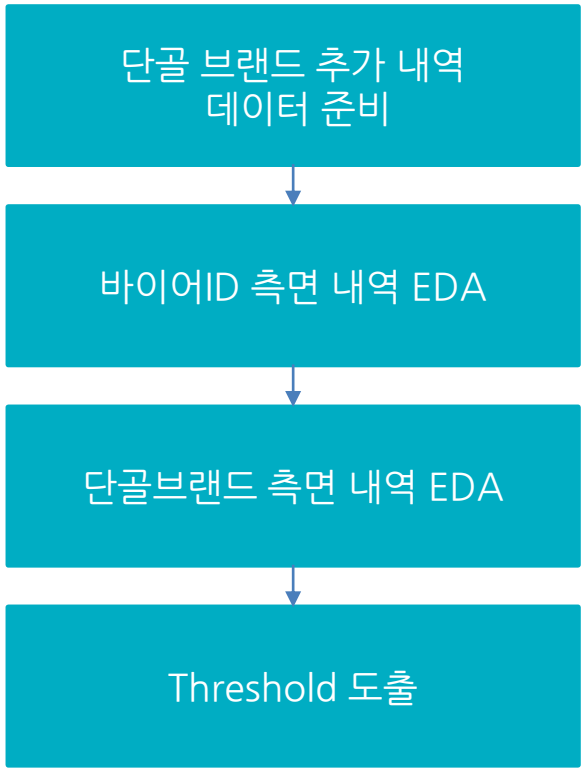
-. 단골 브랜드 소수 사례 절단 EDA(요약)

① 단골 브랜드를 **매우 적게 추가**한 **바이어ID**, 혹은 바이어에 의해 **추가된 건수**가 **매우 적은 브랜드 존재**

② **(-1 ~ 1) 사이의 유사도**를 기반으로 작동하는 이번 추천시스템 성격상 **해당 사례들**은 **데이터**에서 **제거**해야 전반적인 품질을 향상 가능

③ **절단이 가능한 Threshold**를 기반으로 데이터 절단이 이루어질 수 있도록 **탐색** 진행

- 탐색 결과, 바이어ID 기준으로는 단골브랜드 추가 건수가 **(log(X) > 1.3) ≈ 4 이상인 바이어ID** 추출 수행
- 단골브랜드 기준으로는 **(log(X) > 1.3) ≈ 7.3 이상인 단골브랜드**만 추출 수행



<EDA 절차도>

3. 데이터 분석 결과

0) 데이터 EDA

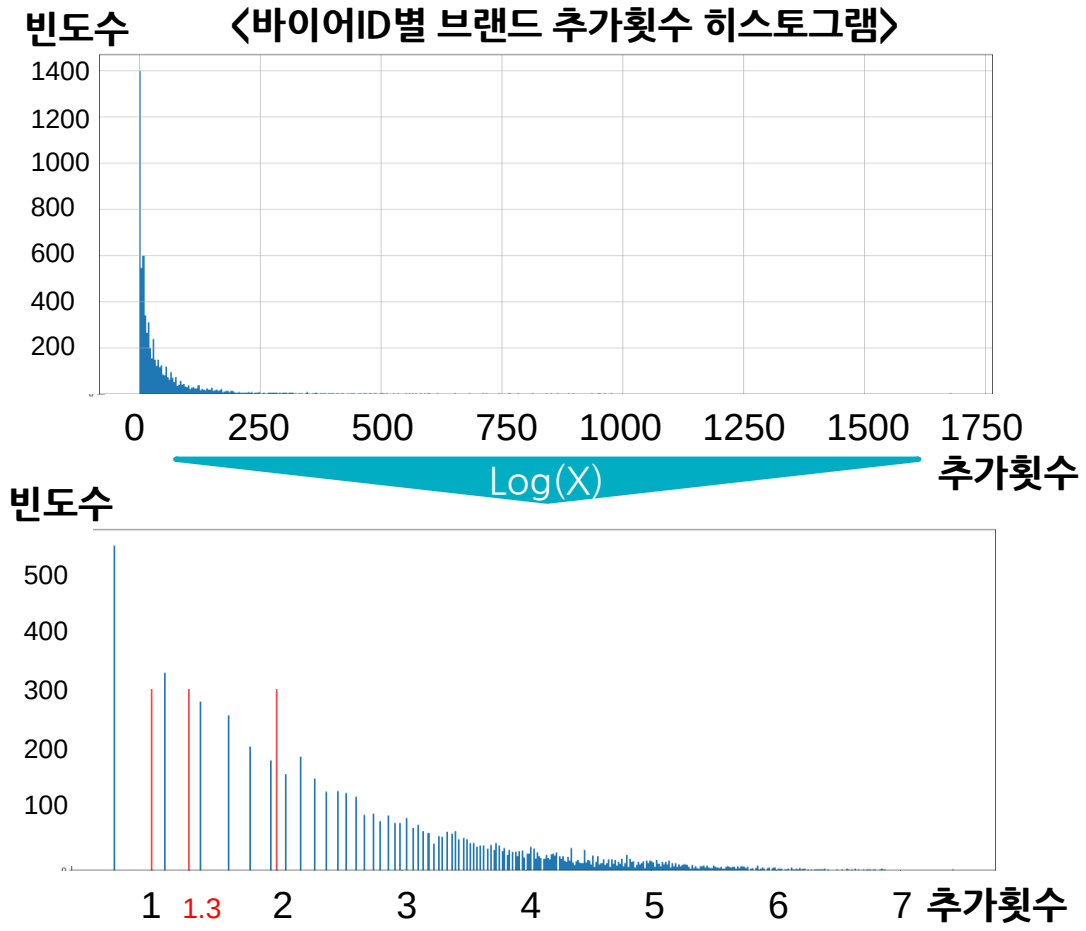
-. 바이어ID 측면 단골브랜드 내역 EDA

① 바이어 ID별 단골 브랜드 추가 개수를 히스토그램으로 시각화

② 0 ~ 250 구간에 전체 샘플의 98%가 몰려 있어 계급 차이가 명확하게 드러나지 않음

③ 빈도수를 로그 변환하여 계급 차이를 좀 더 명확화

④ **(log(X) > 1.3) ≈ 4** 이상일 때 **제외되는 사례 865**건, **포함되는 사례 5618건**으로 적절한 Threshold로 판단



3. 데이터 분석 결과

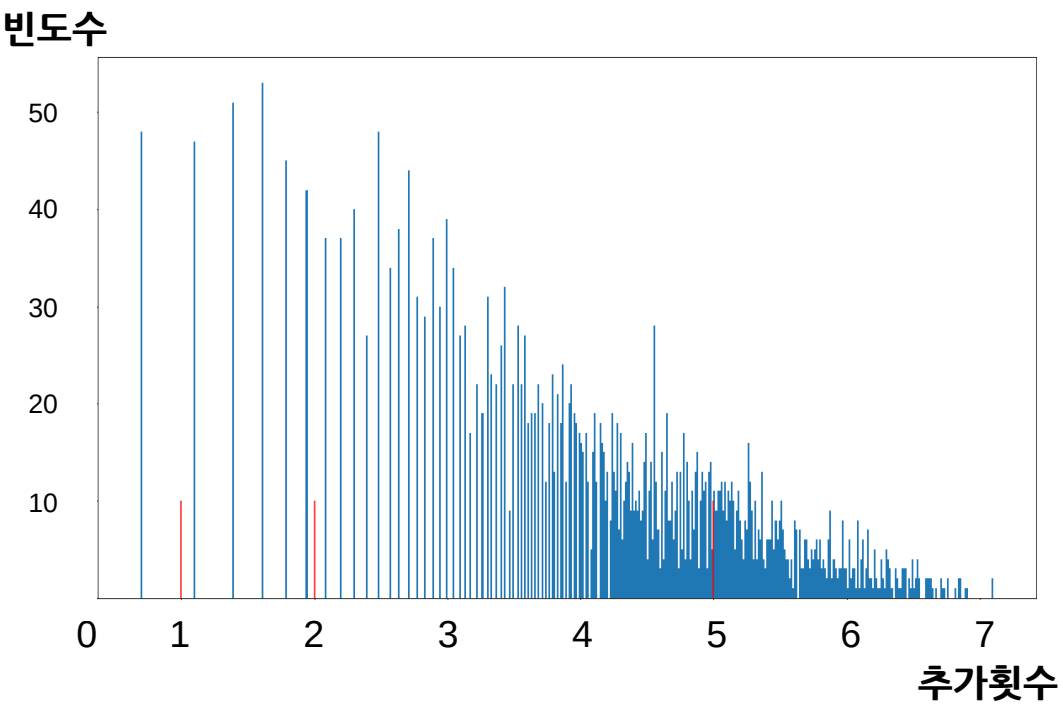
0) 데이터 EDA

-. 단골브랜드 측면 내역 EDA

① 단골 브랜드별 추가된 내역에 대하여 EDA 수행
- 단골브랜드 추가 내역 행렬을 전치(Transpose)
- 브랜드ID를 행으로, 바이어ID를 열로 하여 횟수 Count

② **X >= 2**일 때 **제외되는 사례 286건**, **포함되는 사례 2664**건으로 적절한 Threshold로 판단

<바이어ID별 브랜드 추가횟수 히스토그램>



3. 데이터 분석 결과

1. 유저 프로파일링

2. 동일 카테고리 도출

3. 상품 프로파일링

1) 추천 A-1안

-. 유저 프로파일링

① 바이어의 **단골 브랜드 추가 내역**을 기반으로 **유사한 바이어**를 도출하는 단계

② **각 바이어별**로 **순회**를 돌면서 **Top-K** or **역치(Threshold)** 이상의 **유사 브랜드**를 **코사인 유사도** 기반으로 추출

③ 해당 바이어들은 '**유사**한 **디자인 특성**을 **선호**한다' 라는 가정하에 다음 절차를 진행

브랜드ID / 바이어ID	2123	5211	33321	18620
1215	1	0	0	1
1224	1	1	0	1
1351	0	0	0	0
1455	1	1	1	1
3851	0	0	0	1

〈유저 - 브랜드 협업 행렬〉

유사도 계산

	1215	1224	1351	1455	3851
1215	1	0.82	0	0.71	0.71
1224	0.82	1	0	0.86	0.58
1351	0	0	0	0	0
1455	0.70	0.86	0	1	0.5
3851	0.	0.5	0	0.58	1

〈유저간 코사인 유사도 행렬〉

Top-K 추출(K = 1 예시)

1215	1224
3851	1455

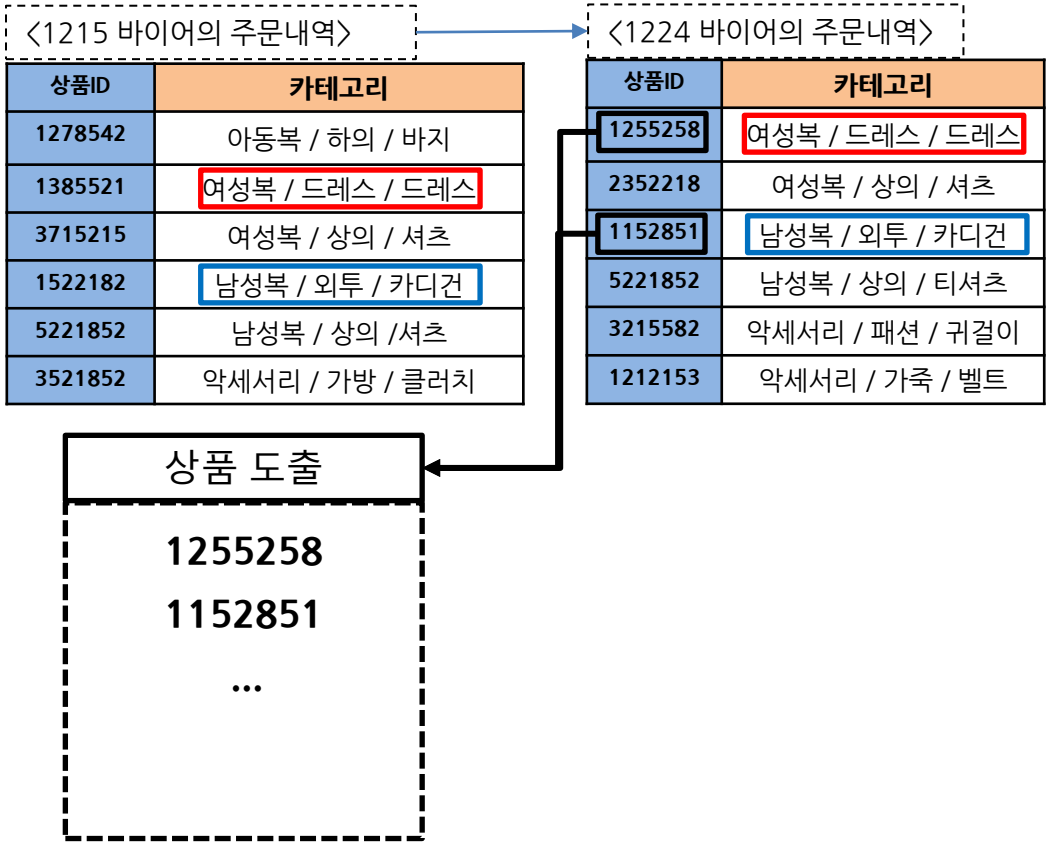
3. 데이터 분석 결과

1. 유저 프로파일링
2. 동일 카테고리 도출
3. 상품 프로파일링

1) 추천 A-1안

-. 동일 카테고리 추출

① **각 바이어별**로 **순회**를 돌면서 바이어가 구매한 각 상품별로 3-Depth까지 동일한 유사 바이어의 상품들을 도출



3. 데이터 분석 결과

1. 유저 프로파일링

2. 동일 카테고리 도출

3. 상품 프로파일링

1) 추천 A-1안

-. 상품 프로파일링

① 도출된 **동일 카테고리**내 **유사 상품**들 대상

② **상품 특성**들을 이용하여 **Top - K**의 상품들을 **유사도 추출**
. 현재는 상품을 추가한 바이어의 **위시리스트**를 상품 특성으로 활용
. 향후 **상품 특성**이 더 뚜렷하게 드러나는 **다른 특성(이미지, 텍스트 등)**으로 **교체** 가능

③ 추출한 상품들을 **유사도기준**으로 **Sorting**하여 최종 표출

바이어ID	2123	5211	33321	18620	…
상품ID					
1278542	1	0	0	1	
1385521	1	1	0	1	
3715215	0	0	0	0	…
1255258	1	1	1	1	
1152851	0	0	0	1	
…					

〈상품 특성 행렬(위시리스트)〉

유사도 계산

	1255258	1152851	…
1278542	0.23	0.33	
1385521	0.10	0.12	…
3715215	0.08	0.02	

리스트 도출

바이어 1226

상품ID	상품명	유사도
1255258	니트 스커트	0.33
1152851	신상 남성 카디건	0.12

3. 데이터 분석 결과

1. 유저 프로파일링

2. 동일 카테고리 도출

3. 상품 프로파일링

2) 추천 A-3안

-. 유저 프로파일링

① 바이어의 **단골 브랜드 추가 내역**을 기반으로 **유사한 바이어**를 도출하는 단계

② **각 바이어별**로 **순회**를 돌면서 **Top-K** or **역치(Threshold)** 이상의 **유사 브랜드**를 **코사인 유사도** 기반으로 추출

③ 해당 바이어들은 '**유사**한 **디자인 특성**을 **선호**한다' 라는 가정하에 다음 절차를 진행

브랜드ID / 바이어ID	2123	5211	33321	18620
1215	1	0	0	1
1224	1	1	0	1
1351	0	0	0	0
1455	1	1	1	1
3851	0	0	0	1

〈유저 - 브랜드 협업 행렬〉

유사도 계산

	1215	1224	1351	1455	3851
1215	1	0.82	0	0.71	0.71
1224	0.82	1	0	0.86	0.58
1351	0	0	0	0	0
1455	0.70	0.86	0	1	0.5
3851	0.	0.5	0	0.58	1

〈유저간 코사인 유사도 행렬〉

Top-K 추출(K = 1 예시)

1215	1224
3851	1455

3. 데이터 분석 결과

1. 유저 프로파일링
2. 동일 카테고리 도출
3. 상품 프로파일링

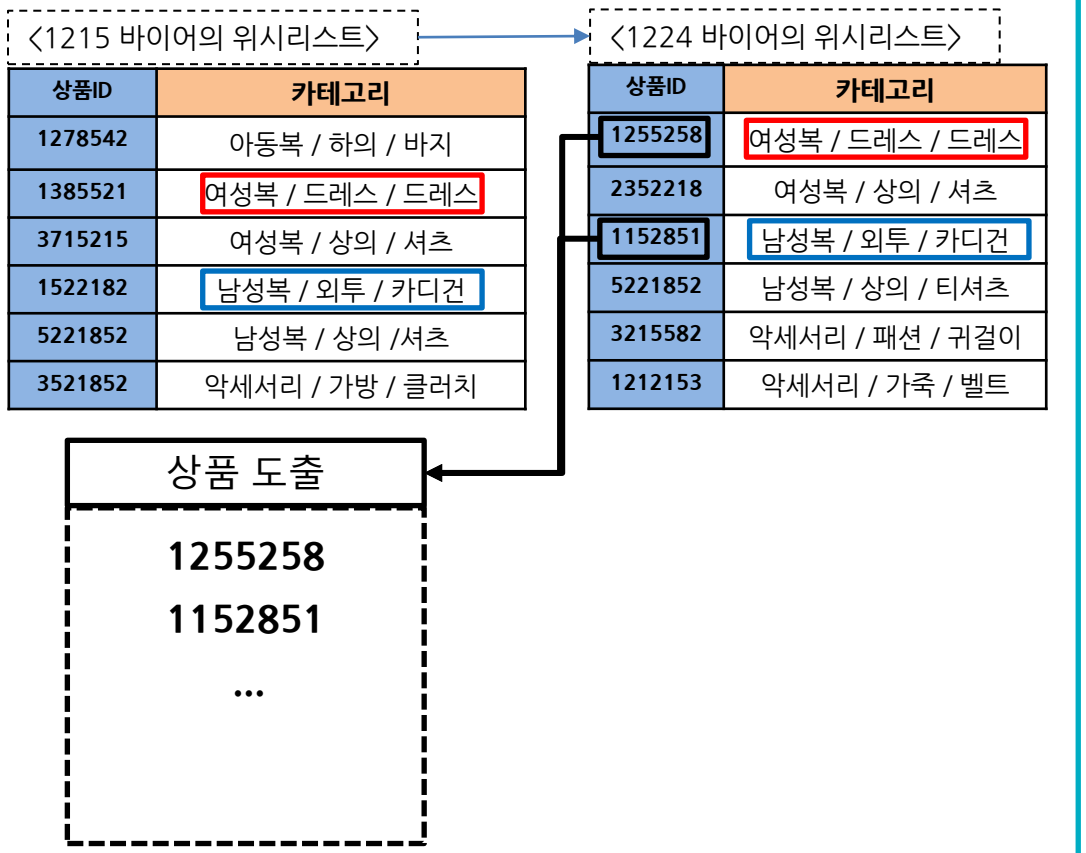
2) 추천 A-3안

-. 동일 카테고리 추출

① **각 바이어별**로 **순회**를 돌면서 바이어가 위시리스트로 추가한 각 상품들의 카테고리 도출

② 유저 프로파일링 단계에서 도출한 **후보 바이어**들의 **위시리스트 추가 상품들**의 카테고리 도출

③ 3-Depth까지 동일한 유사 바이어의 상품들을 도출



3. 데이터 분석 결과

1. 유저 프로파일링

2. 동일 카테고리 도출

3. 상품 프로파일링

2) 추천 A-3안

-. 상품 프로파일링

① 도출된 **동일 카테고리**내 **유사 상품**들 대상

② **상품 특성**들을 이용하여 **Top - K 유저**의 위시리스트 상품들의 **유사도 추출**
. 현재는 상품을 추가한 바이어의 **위시리스트**를 상품 특성으로 다시 활용
. 향후 **상품 특성**이 더 뚜렷하게 드러나는 **다른 특성(이미지, 텍스트 등)**으로 **교체** 가능

③ 추출한 상품들을 **유사도기준**으로 **Sorting**하여 최종 표출

바이어ID	2123	5211	33321	18620	…
상품ID					
1278542	1	0	0	1	
1385521	1	1	0	1	
3715215	0	0	0	0	…
1255258	1	1	1	1	
1152851	0	0	0	1	
…					

〈상품 특성 행렬(위시리스트)〉

유사도 계산

	1255258	1152851	…
1278542	0.23	0.33	
1385521	0.10	0.12	…
3715215	0.08	0.02	

리스트 도출

바이어 1226

상품ID	상품명	유사도
1255258	니트 스커트	0.33
1152851	신상 남성 카디건	0.12

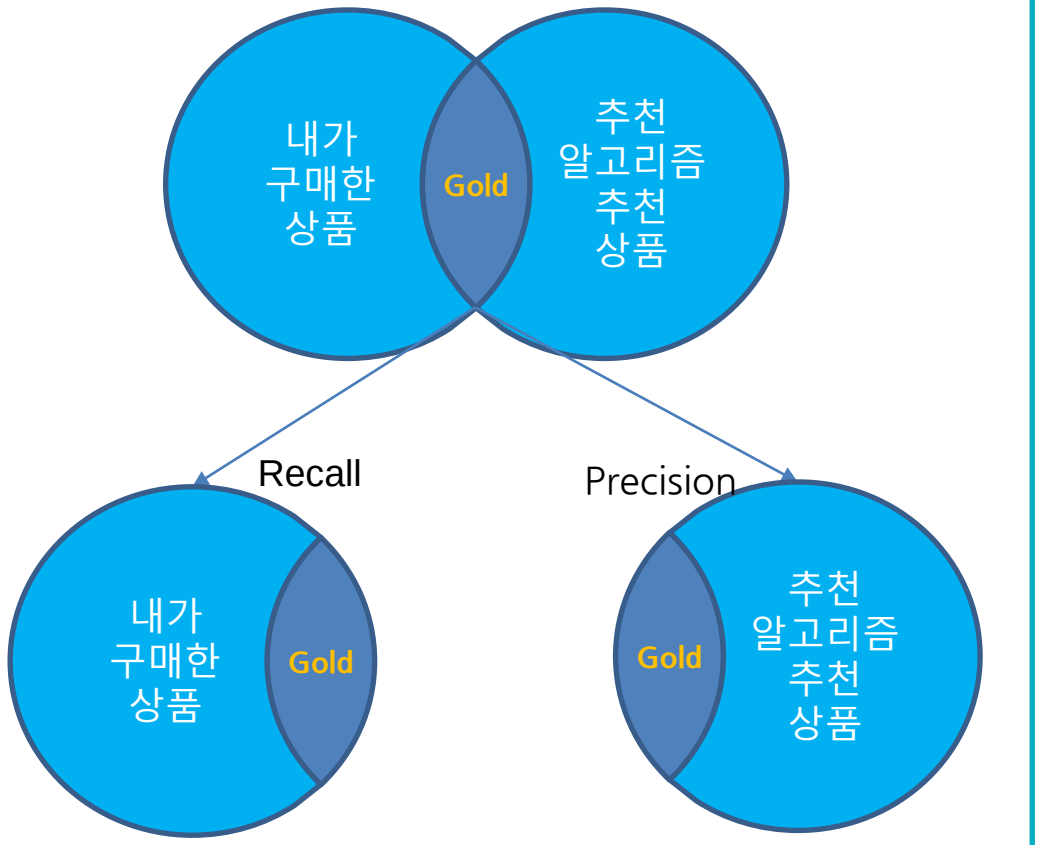
4. 성능 평가

1) 성능 평가 지표

-. 성능 측정 결과

① 내가 실제 구매한 상품 중 추천 알고리즘이 추천한 상품의 비율 -> Recall(재현율)

② 내가 구매했다고 추천한 상품 중 실제 내가 구매한 상품의 비율 -> Precision(정밀도)



4. 성능 평가

2) 추천 A-1안 성능 측정 결과

-. 성능 측정 결과

① 내가 실제 구매한 상품 중 추천 알고리즘이 추천한 상품의 비율 -> Recall(재현율)

② 내가 구매했다고 추천한 상품 중 실제 내가 구매한 상품의 비율 -> Precision(정밀도)

③ 추천 알고리즘을 가동하면서 데이터 부족으로 전처리 불가 / 타겟 상품 도출 불가 / 후보 상품 도출 불가 / 추천 결과 도출 불가로 제외된 빈도

지표	값
Precision	0.012
Recall	0.029

<A-1안 성능 평가표>

지표	값	
총 명수	6,446명	
전처리 후	4,641명	
추천 도출 제외	타겟 상품 X	19
	후보 상품 X	0
	추천 결과 X	691
	위시리스트 X	0
추천 수행	3,931명	

<A-1안 콜드 스타트 사례 빈도표(23.03.07 기준)>

4. 성능 평가

3) 추천 A-3안 성능 측정 결과

-. 성능 측정 결과

① 내가 실제 구매한 상품 중 추천 알고리즘이 추천한 상품의 비율 -> Recall(재현율)

② 내가 구매했다고 추천한 상품 중 실제 내가 구매한 상품의 비율 -> Precision(정밀도)

③ 추천 알고리즘을 가동하면서 데이터 부족으로 전처리 불가 / 타겟 상품 도출 불가 / 후보 상품 도출 불가 / 추천 결과 도출 불가로 제외된 빈도

지표	값
Precision	0.014
Recall	0.046

<A-3안 성능 평가표>

지표	값	
총 명수	6,446명	
전처리 후	4,641명	
추천 도출 제외	타겟 상품 X	165
	후보 상품 X	34
	추천 결과 X	684
	위시리스트 X	16
추천 수행	3,740명	

<A-3안 콜드 스타트 사례 빈도표(23.03.07 기준)>

4. 성능 평가

4) 비교군 : user - user 협업필터링

-. 성능 측정 결과

① A-1, A-3안과 비교했을 때 성능지표와 콜드스타트 양쪽 모두 USER - USER 협업 필터링이 우수한 것으로 언뜻 보임

② 그러나, 실제 사용자 측면에서 추천 결과를 확인해봤을 때 다음의 절차 결여로 매력적인 추천 결과를 뽑아내지 못하는 것으로 보임

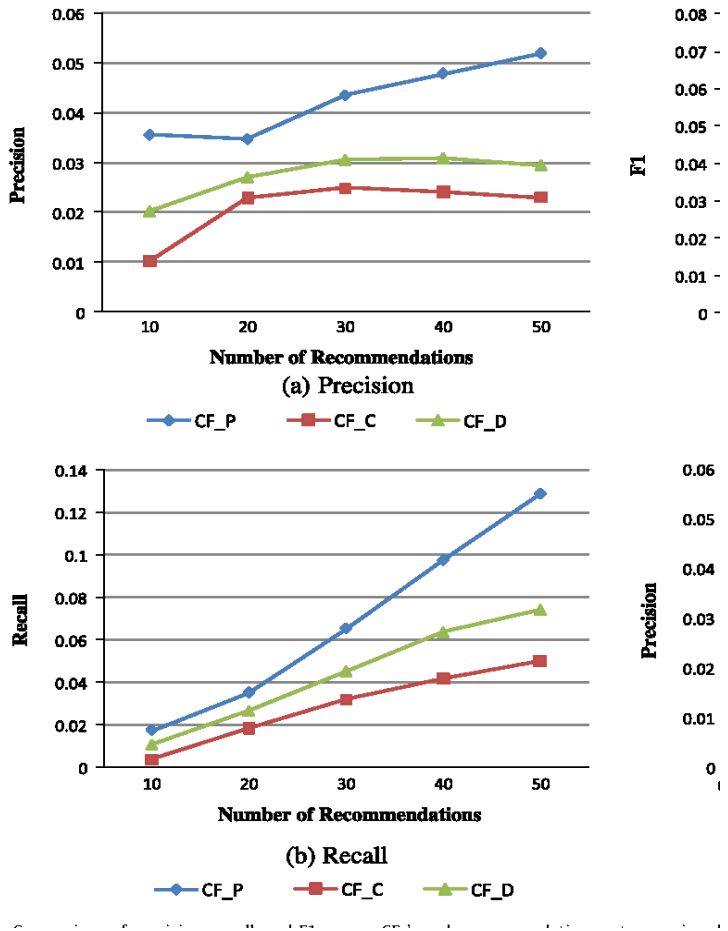
- **카테고리 기반 필터링 부재** : Target-buyer의 구매 상품별 **카테고리**를 **고려 X**-> Target-buyer가 별로 구매하지 않은 카테고리의 상품을 다수 추천
- **추천 결과 부재 多** : A-1, A-3안 모두 **추천결과가 없는** 경우가 **7~800건 수준**이나, A-3안 같은 경우는 **1778건으로 2.5배 수준**
  -> A-1, A-3안의 경우 **wishlist, 단골브랜드 내역 추가**를 통해 **Cold-start 해결이** 가능하나 이 방법의 경우 **유저가 구매**를 **더 많이** 하지 않는 한 해결 불가능

지표	값
Precision	0.287
Recall	0.11

<협업 필터링 성능 평가표>

지표	값	
총 명수	6,446명	
전처리 후	-	
추천 도출 제외	타겟 상품 X	-
	후보 상품 X	-
	추천 결과 X	1778
	위시리스트 X	-
추천 수행	4,668명	

<협업 필터링콜드 스타트 사례 빈도표(23.03.07 기준)>



A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis

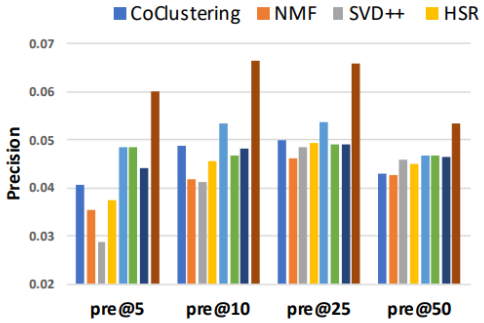


Fig. 2. Precision@K on *Apparel*.

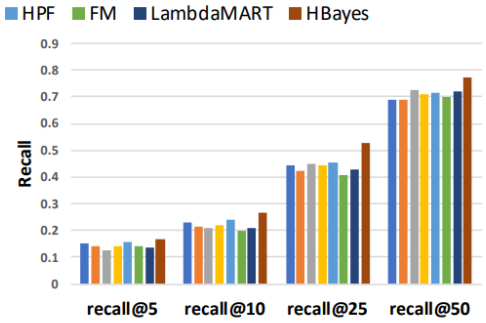


Fig. 3. Recall@K on *Apparel*.

Hierarchical Bayesian Personalized Recommendation: A Case Study and Beyond

결론

결론

LinkShops

1. 결론 요약

결론 1

-. **정량적 지표**로만 확인했을 땐 단순 **user-user 협업필터링**이 **A-1, A-3안**보다 **더 우수**
-. But **정성적**으로 확인했을 때 다음 사유로 **A-1, A-3이 더 우수**한 것으로 판단
① **추천 적정성** : 사용자 경험 차원에서 **A-1, A-3**이 **user-user 협업 필터링**보다 **더 적절한 결과**를 추천
② **성능 개선 가능성** : **추천 실패 사례수**가 **2.5배**에 이르고 **추가적인 개선**이 **불가능**하여 A-1, A-3안이 더 우수한 것으로 보임

결론 2

-. **정량적 지표**로 확인했을 때 **Precision / Recall은 0.02 / 0.04** 수준으로 **유사 연구**들과 비교해봤을 때 **평균적인 성능**으로 판단
-. **정성적**으로 확인했을 때 **유사한 스타일**과 **같은 카테고리**를 기반으로 추천을 수행하기 때문에 **사용자 경험 차원**에서 **적절한 추천 결과**를 보여주는 것으로 판단

결론

2. 연구의 한계 향후 시사점

연구의 한계

-. 연구 환경의 한계 : GPU 활용이 불가능하고, 메모리가 작은 현 연구 환경에서 좀 더 장기간 / 좀 더 많은 변수를 활용 / Matrix Factorization같은 좀 더 상호간의 연결성을 강조하는 방법론 활용에 한계

향후 시사점

-. 향후 연구 환경이 개선되고, 유저들이 Wishlist / 단골 브랜드 추가를 더 적극적으로 수행할 경우 추천 성능과 Cold-start 문제 해결에 진전이 있을 것으로 기대

-.

참고문헌

- S.M Mahdi Seyednezhad et.al, “A Review on Recommendation Systems: Context-aware to Social-based”, Arxiv(2018),
- Zitao Liu Et.al, “Hierarchical Bayesian Personalized Recommendation”, Arxiv(2019)
- Keunho Choi et.al, “A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis”, Electronic Commerce Research and Applications Vol 11, Issue 4(2012), pp 309-317
- Paul Convington, Jay Adams and Emre Sargin, “Deep Neural Networks for YouTube Recommendations”, RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems(2016), pp 191-198

부록

데이터 흐름

1. A-1 알고리즘 데이터 흐름

1) 흐름도 얼개

-. 데이터 처리 프로세스

① **상품 정보 전처리** : 카테고리 정보를 Json에서 추출하여 orderseries와 Merge

②-1 **카테고리 데이터 전처리** : 카테고리 정보를 Pivoting하여 A_1 알고리즘에 전달

②-2 **위시리스트 전처리** : orderseries와 wishlist를 Merge

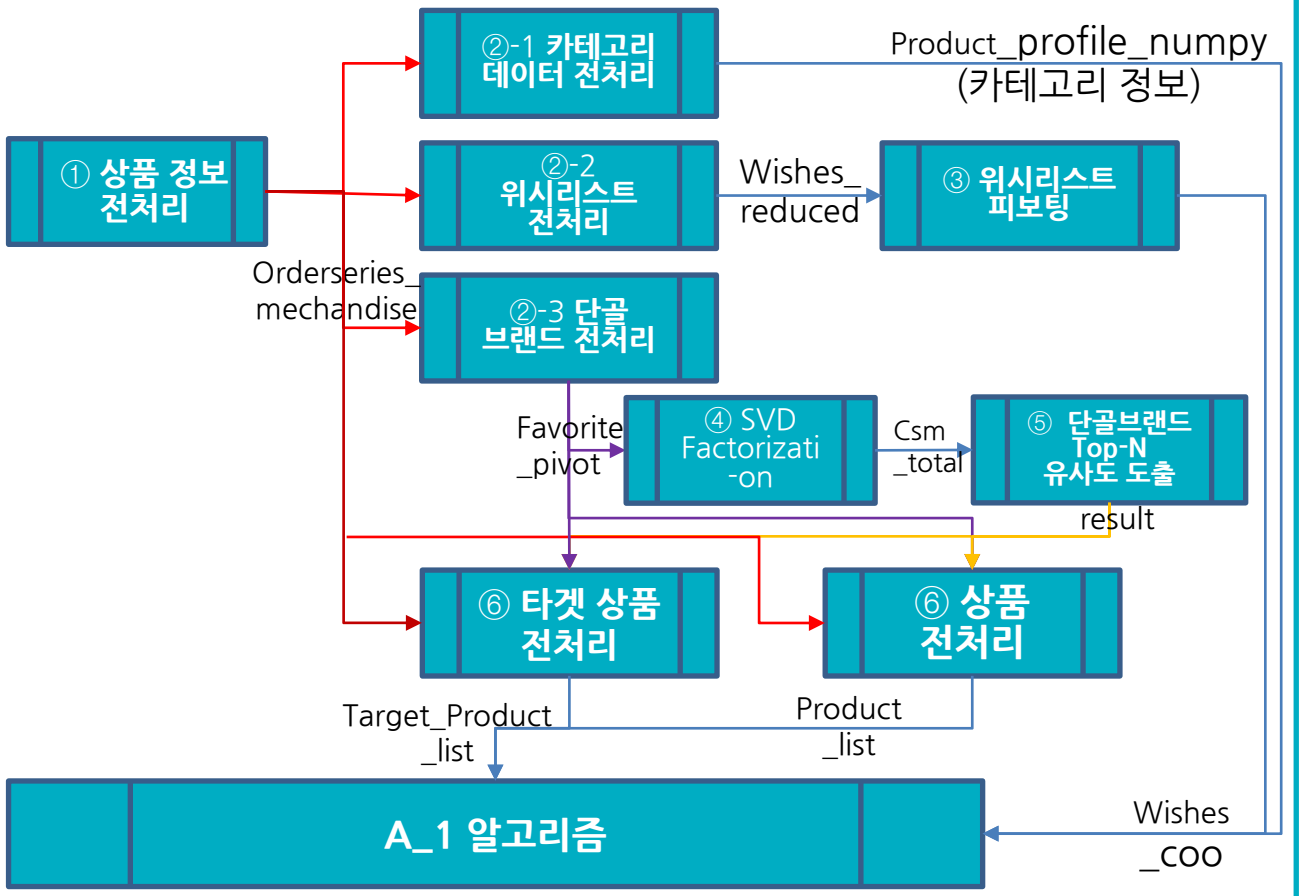
②-3 **단골브랜드 전처리** : orderseries와 단골브랜드List를 Merge 후 기준에 따라 소수 사례는 Filtering 수행

③ **위시리스트 피보팅** : wishes_reduce를 피보팅하여 희소행렬(coo_matrix)로 변환

④ **SVD Matrix Factorization** : 암시적 정보(Implicity Information)를 표면화하기 위해 SVD 행렬분해 후 재조합 수행

⑤ **단골브랜드 Top -N 유사도 도출** : 단골브랜드(Favorite_pivot) 기반으로 각 바이어ID별 최다 유사도 바이어ID 짝을 도출

⑥ **타겟 상품 전처리 / 상품 전처리**: 단골브랜드(Favorite_pivot) 기반으로 각 바이어ID별 최다 유사도 바이어ID 짝을 도출



데이터 흐름

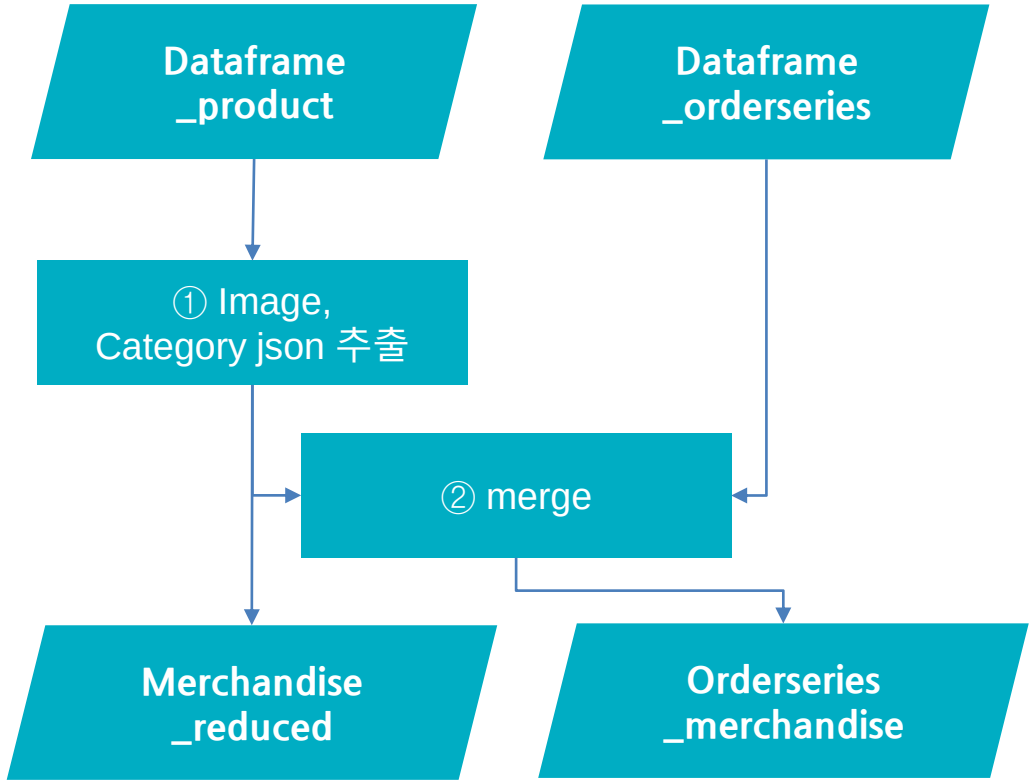
1. A-1 알고리즘 데이터 흐름

2) ① 상품 정보 전처리

-. 투입 데이터
① dataframe_product(상품 데이터프레임
② dataframe_orderseries(주문 데이터 프레임)

-. 데이터 처리 프로세스
① dataframe_product에서 이미지 정보, 카테고리 정보를 Json에서 추출 ➔ merchandise_reduced
② dataframe_orderseries와 merchandised_reduced merge ➔ orderseries_merchandise

-. 산출 데이터
① merchandise_reduced(pandas DataFrame)
② orderseries_merchandise(pandas DataFrame)



데이터 흐름

#LinkShops

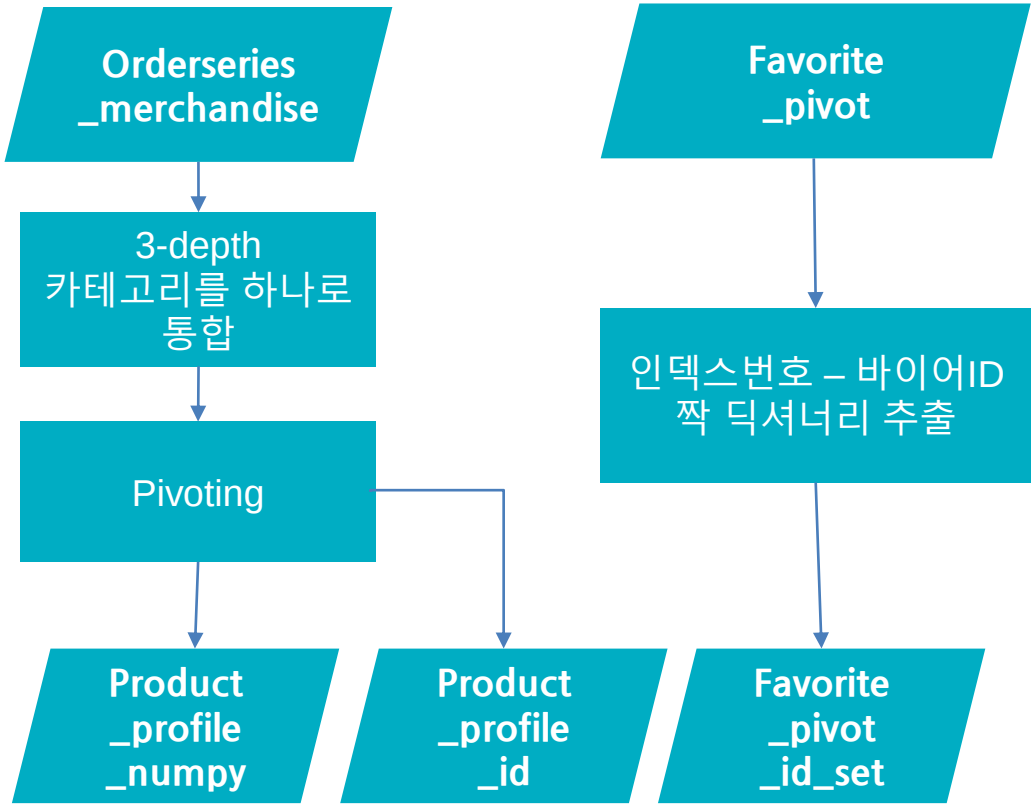
1. A-1 알고리즘 데이터 흐름

3) ②-1 카테고리 데이터 전처리

-. 투입 데이터
① orderseries_merchandise
② favorite_pivot

-. 데이터 처리 프로세스
① orderseries_merchandise에서 3-depth까지의 카테고리 내용을 하나로 통합 후 상품ID를 인덱스로, 카테고리(통합)을 열로 갖는 테이블로 Pivoting ➔ product_profile_numpy
② product_profile_numpy에서 인덱스번호 - 상품ID 짝 딕셔너리를 추출 ➔ product_profile_Id
③ favorite_pivot에서 인덱스번호 - 바이어ID 짝 딕셔너리를 추출 ➔ favorite_pivot_Id_set

-. 산출 데이터
① product_profile_numpy(numpy array)
② product_profile_Id(dict)
③ favorite_pivot_Id_set(dict)



데이터 흐름

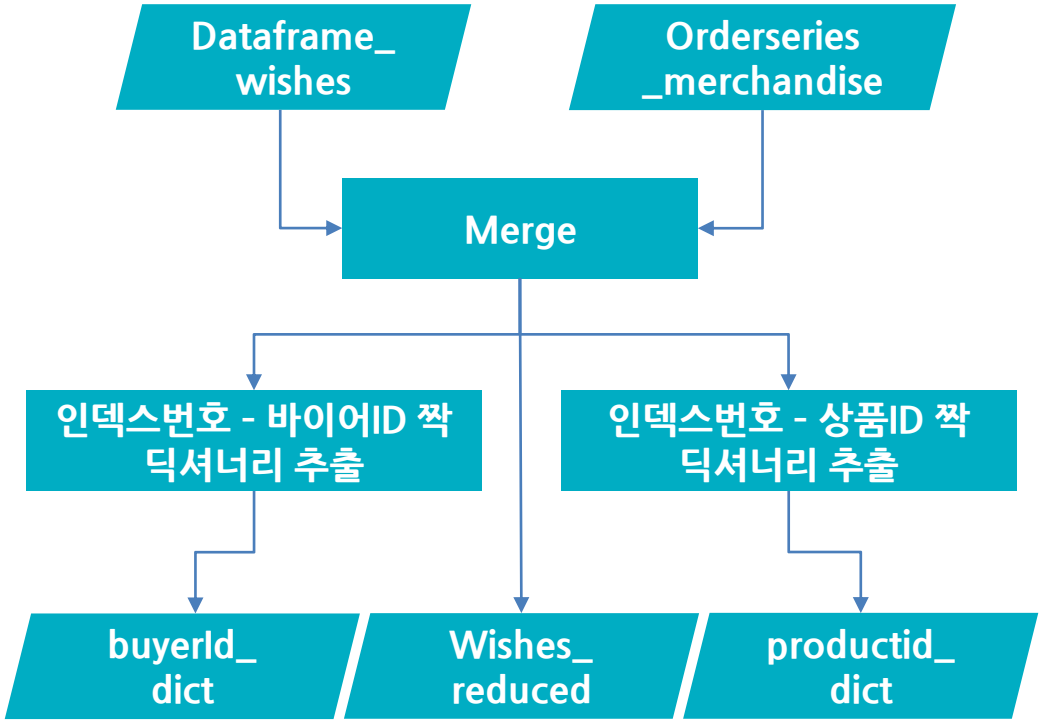
1. A-1 알고리즘 데이터 흐름

4) ②-2 위시리스트 전처리

-. 투입 데이터
① dataframe_wishes(위시리스트 추가 내역)
② orderseries_merchandise

-. 데이터 처리 프로세스
① favorite_brand와 orderseries_merchandise Merge
② 바이어ID와 상품ID를 각각 인덱스와 짝지은 딕셔너리 추출

-. 산출 데이터
① wishes_reduced(pandas Dataframe)
② buyerId_dict(dict)
③ productid_dict(dict))



데이터 흐름

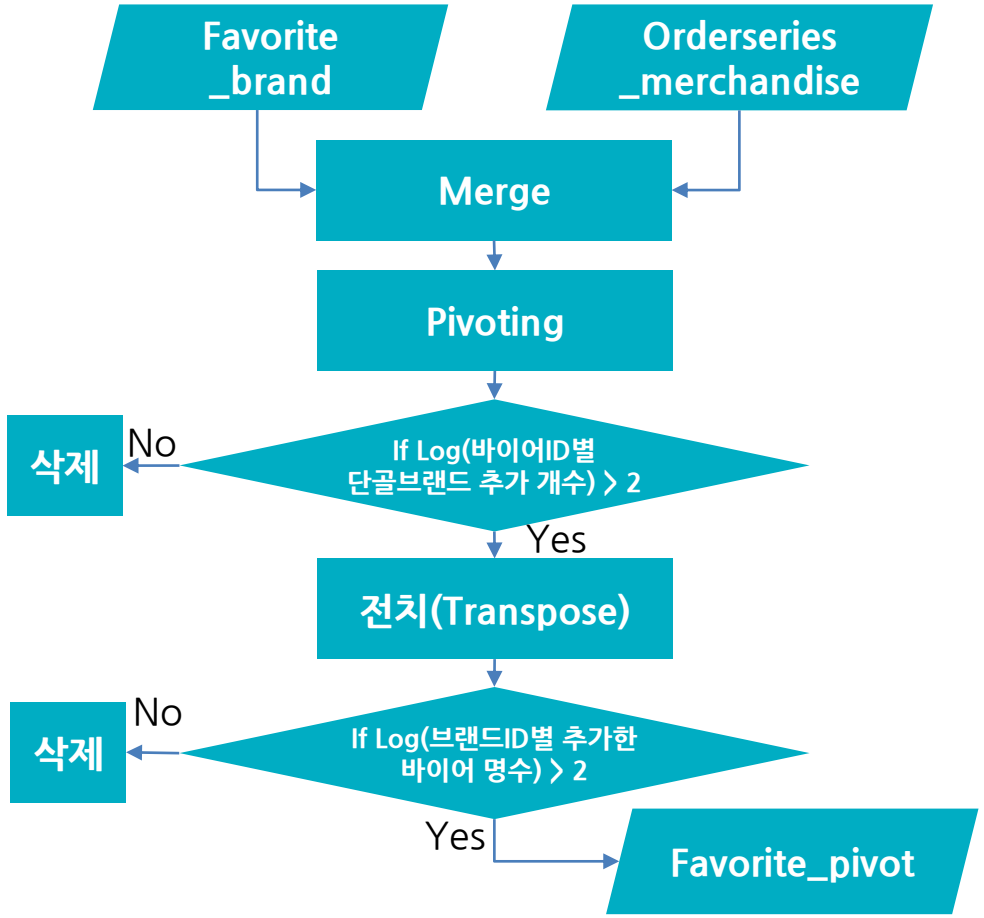
1. A-1 알고리즘 데이터 흐름

5) ②-3 단골브랜드 전처리

-. 투입 데이터
① favorite_brand
② orderseries_merchandise

-. 데이터 처리 프로세스
① favorite_brand와 orderseries_merchandise Merge
② 바이어ID를 인덱스로, 브랜드ID를 열로 갖는 행렬로 Pivoting
③ log(바이어ID별 단골브랜드 추가 개수) > 1.3 이면서 log(브랜드ID별 추가한 바이어명수) > 2인 행만 필터링

-. 산출 데이터
① favorite_pivot



데이터 흐름

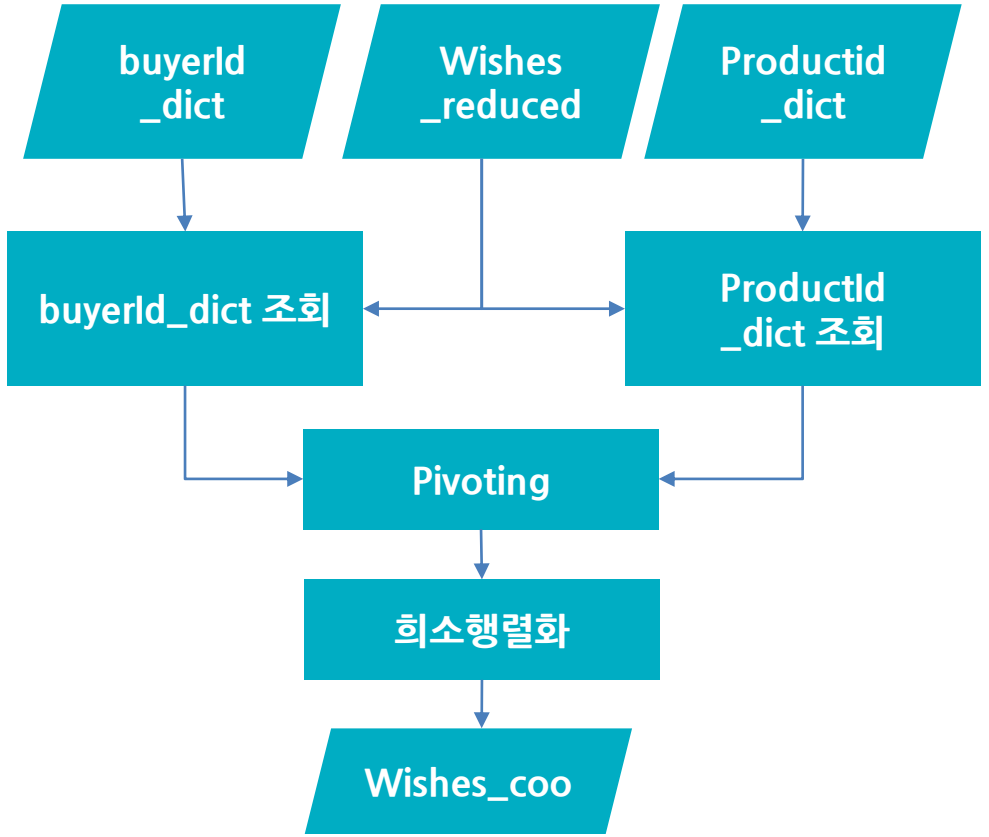
1. A-1 알고리즘 데이터 흐름

6) ③ 위시리스트 피보팅

-. 투입 데이터
① orderseries_merchandise
② favorite_pivot
③ wishes_reduced

-. 데이터 처리 프로세스
① wishes_reduced의 인덱스 정보를 buyerId_dict에서 조회하여 바이어ID 정보로 전환
② wishes_reduced의 인덱스 정보를 productId_dict에서 조회하여 상품ID 정보로 전환
③ scipy의 coo_matrix를 이용하여 pivoting과 동시에 희소행렬화 수행

-. 산출 데이터
① wishes_coo



데이터 흐름

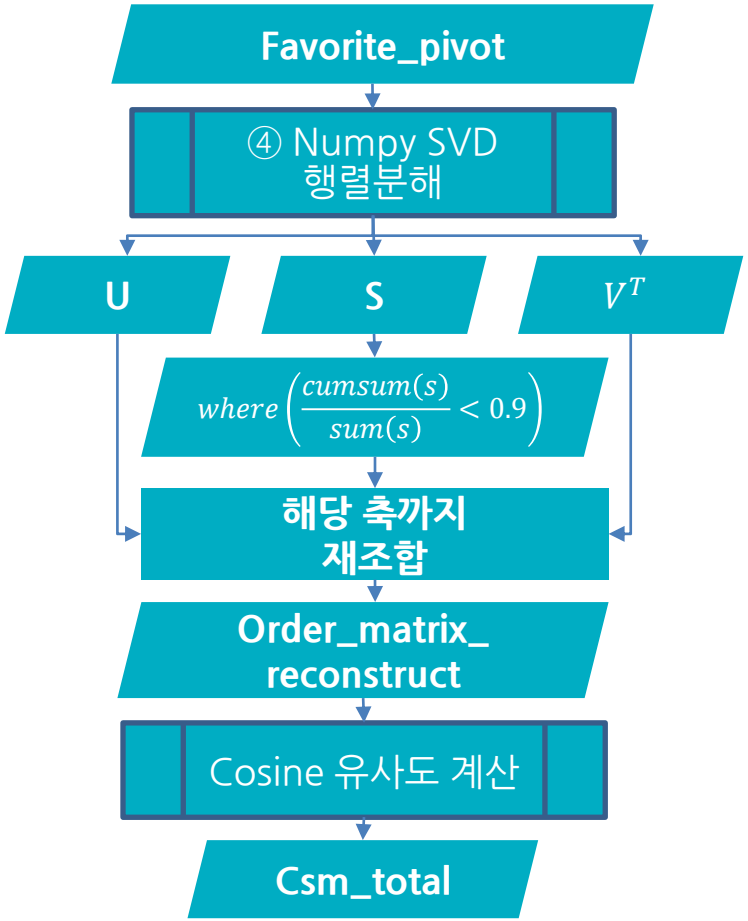
1. A-1 알고리즘 데이터 흐름

7) ④ SVD Matrix Factorization + 코사인 유사도

-. 투입 데이터
① favorite_pivot

-. 데이터 처리 프로세스
① 특이값 분해(SVD)를 통해 U, S, V^T 행렬을 도출
② 특이값의 누적합이 90%인 축을 도출
➔ 평가가 이루어지지 않은 암묵적 선호(Implicity Favorite)를 재조합 행렬에서 표출화하기 위해
➔ 특이값(Singular Value)이 큰 순으로 도출되는 SVD 특성상 비중이 낮은 기저(= 10% 노이즈)를 제거하기 위해
③ 해당 축까지의 특이벡터와 특이값을 필터링하여 재조합
④ 재조합 행렬의 코사인 유사도 행렬 도출

-. 산출 데이터
① csm_total(재조합 행렬의 코사인 유사도 행렬)



데이터 흐름

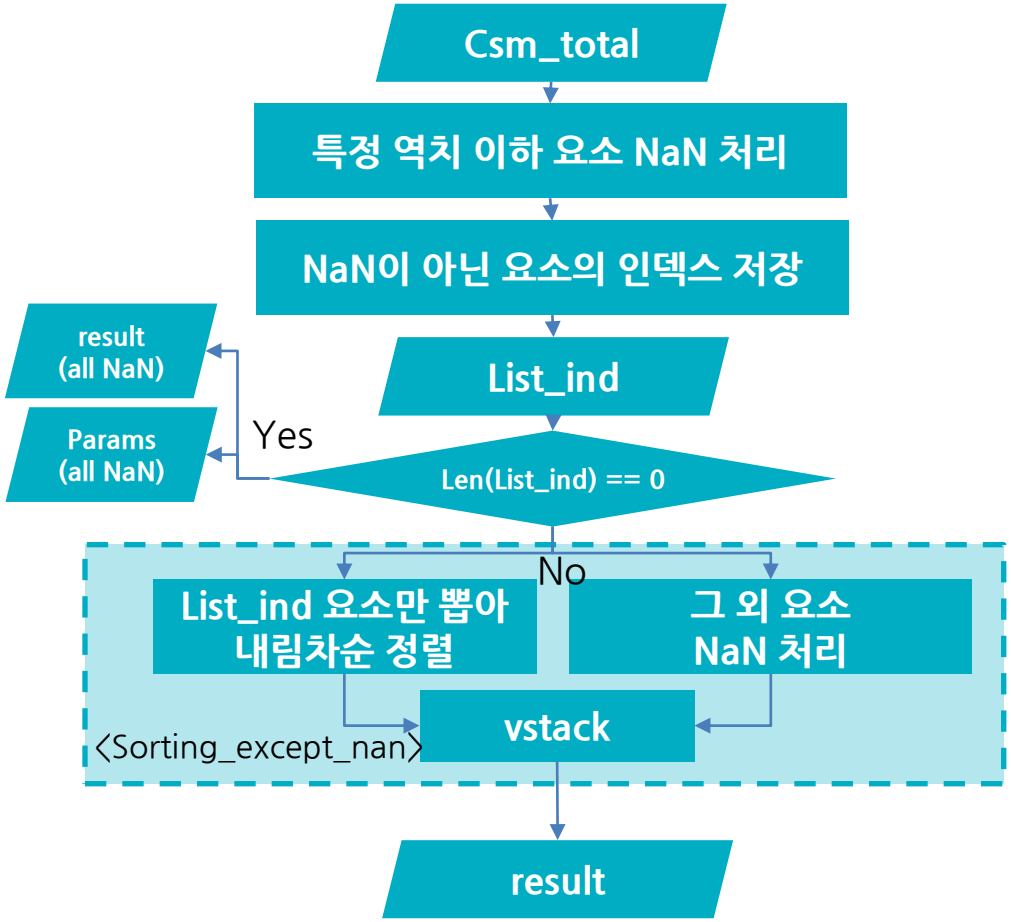
1. A-1 알고리즘 데이터 흐름

8) ⑤ 단골브랜드 Top-N 유사도 도출(**Profiling_engine**)

-. 투입 데이터
① csm_total

-. 데이터 처리 프로세스
① 특정 역치 이하의 값은 내부처리를 위해 NaN으로 처리
② NaN이 아닌 요소의 인덱스 정보 저장 ➔ **list_ind**
③ 전부 다 NaN인경우(=list_ind의 길이가 0) ➔ **result, params를 전부 NaN으로 처리하고 반환**
④NaN값을 포함하는 내림차순 정렬 서브루틴 (**sorting_except_nan**) 실행
➔ NaN값을 포함한 상태로 numpy sorting 알고리즘을 활용할 경우 내림차순 정렬이 제대로 되지 않음

-. 산출 데이터
① result(각 바이어별 top-n 유사도 바이어 ID 목록, dict)



데이터 흐름

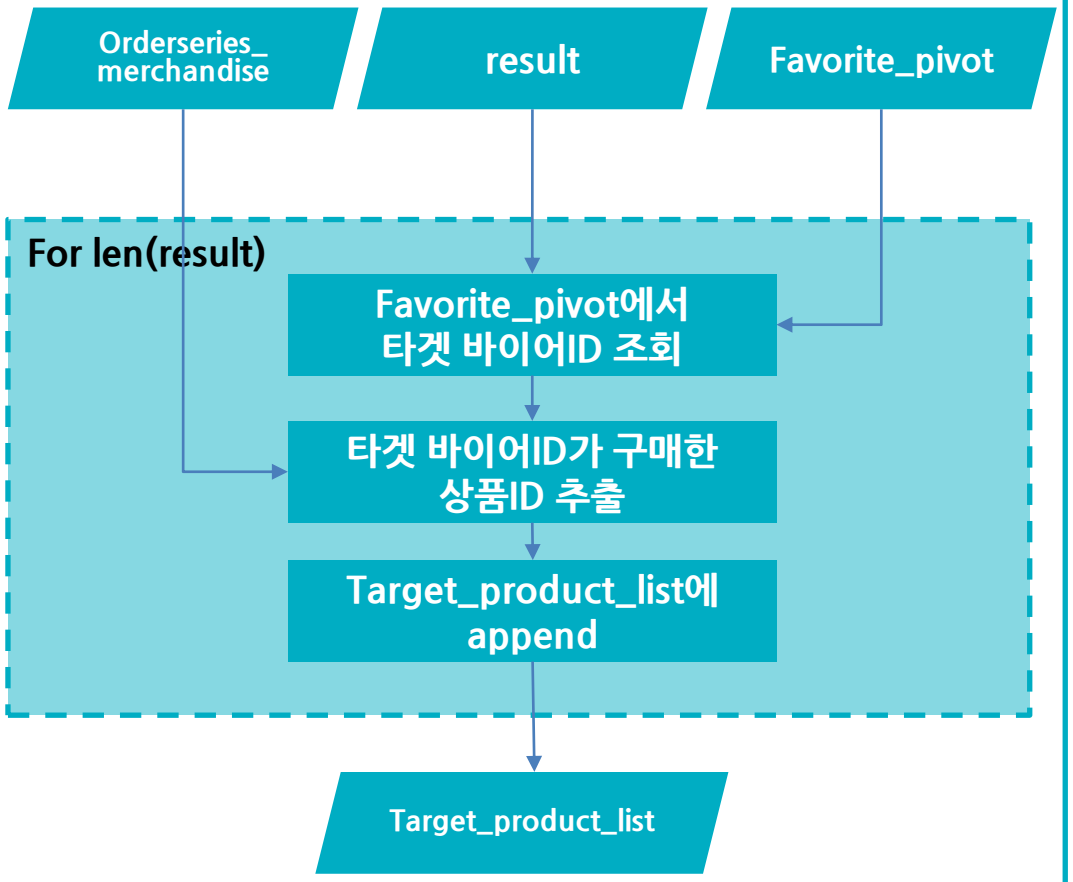
1. A-1 알고리즘 데이터 흐름

9) ⑥ 타겟 상품 전처리

-. 투입 데이터
① orderseries_merchandise
② favorite_pivot
③ result

-. 데이터 처리 프로세스
① result의 요소 개수(= 대상 바이어ID의 수)만큼 순회
② favorite_pivot에서 역으로 현재 순서의 바이어ID(=타겟 바이어ID)를 조회
③ 타겟 바이어 ID가 구매한 상품ID를 orderseries_merchandise에서 추출
④ 상품ID 목록을 Target_product_list에 append

-. 산출 데이터
① target_product_list



데이터 흐름

#LinkShops

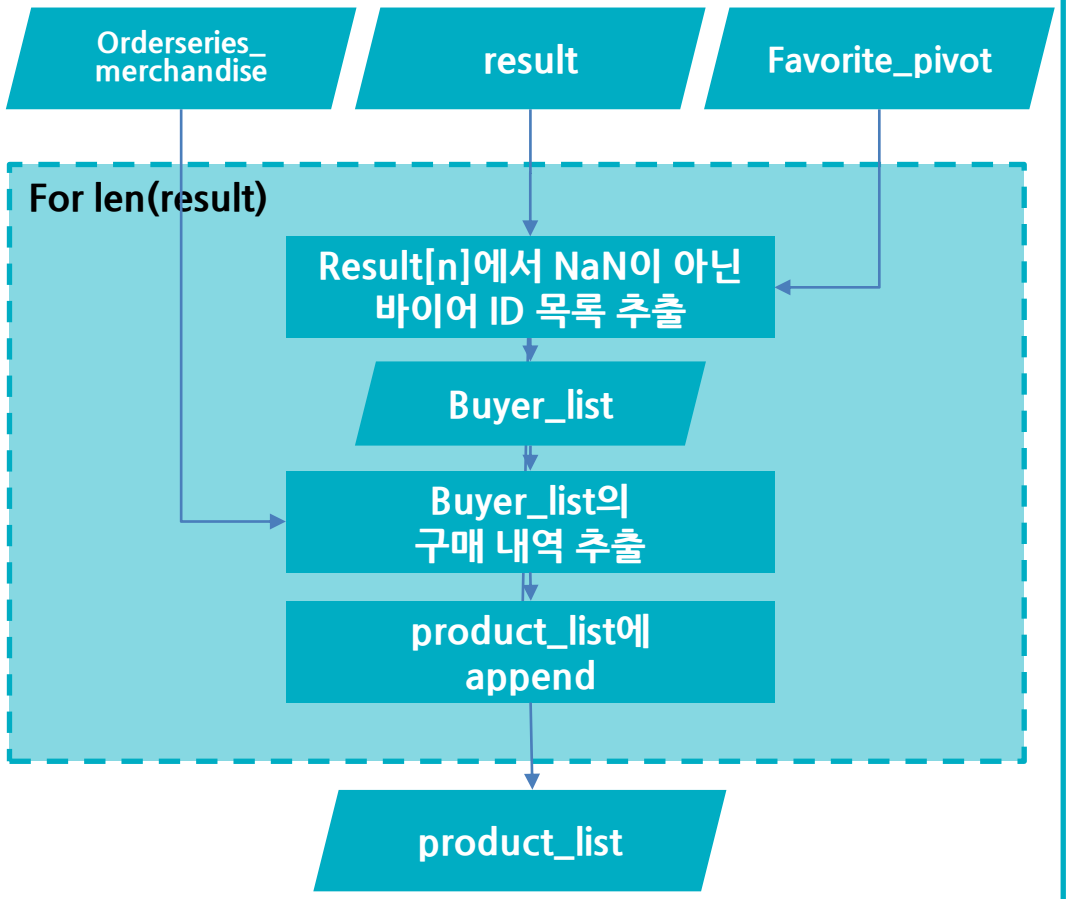
1. A-1 알고리즘 데이터 흐름

10) ⑥ 상품 전처리

-. 투입 데이터
① orderseries_merchandise
② favorite_pivot
③ result

-. 데이터 처리 프로세스
① result의 요소 개수(= 대상 바이어ID의 수)만큼 순회
② favorite_pivot에서 타겟 바이어ID와 유사하다고 판단한 Top-N 바이어 ID 목록을 추출 ➔ buyer_list
③ buyer_list들이 구매한 구매 내역을 추출
④ 상품ID 목록을 product_list에 append

-. 산출 데이터
① product_list



데이터 흐름

1. A-1 알고리즘 데이터 흐름

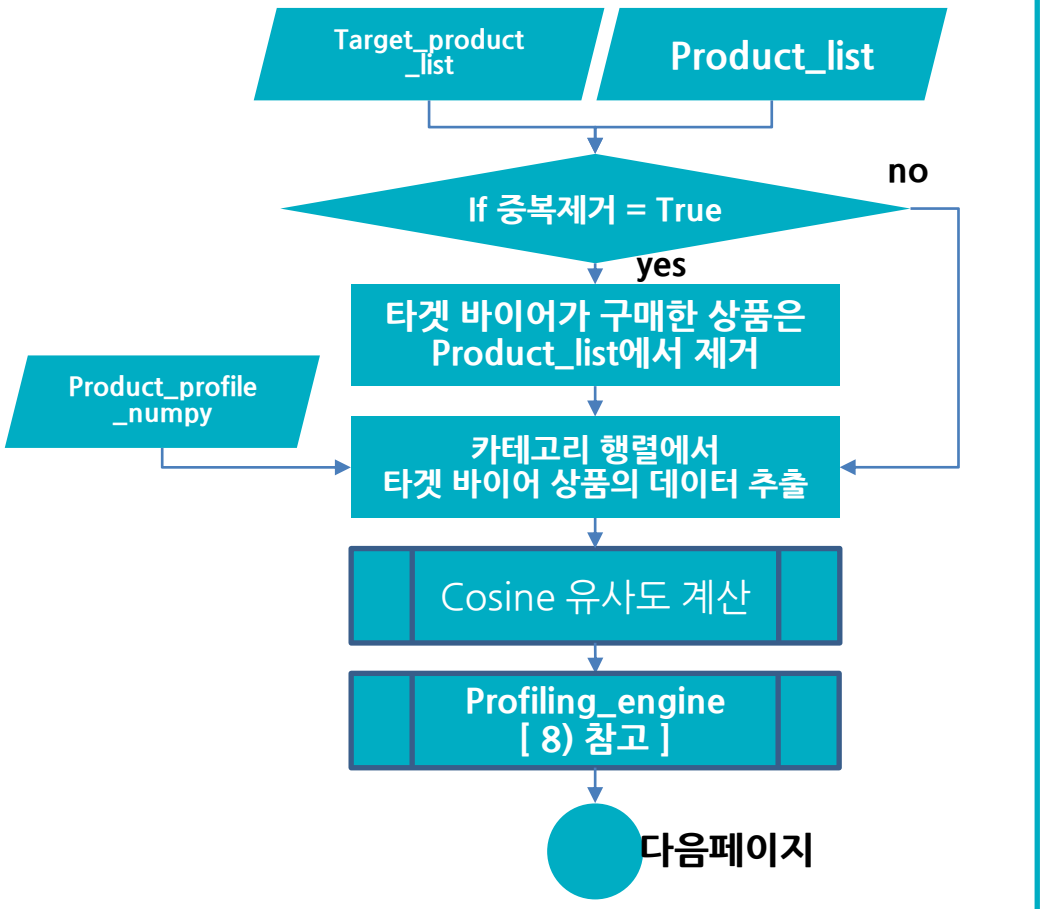
11) A-1 알고리즘부

-. 투입 데이터

① target_product_list
② product_list
③ product_profile_numpy
④ product_profile_ID
⑤ favorite_pivot_id_set
⑥ productid_set
⑦ wishes_coo

-. 데이터 처리 프로세스

① target_product_list, product_list 준비
② productId 리스트에 해당하는 카테고리 행렬 추출
③ 코사인 유사도를 이용해 동일 카테고리 추출
- 모든 상품을 대상으로 카테고리 항목 유사도를 계산
- product_engine을 통해 100% 동일한 카테고리를 갖는 상품목록 추출



데이터 흐름

1. A-1 알고리즘 데이터 흐름

11) A-1 알고리즘부(계속)

-. 투입 데이터

① target_product_list
② product_list
③ product_profile_numpy
④ product_profile_ID
⑤ favorite_pivot_id_set
⑥ productid_set
⑦ wishes_coo

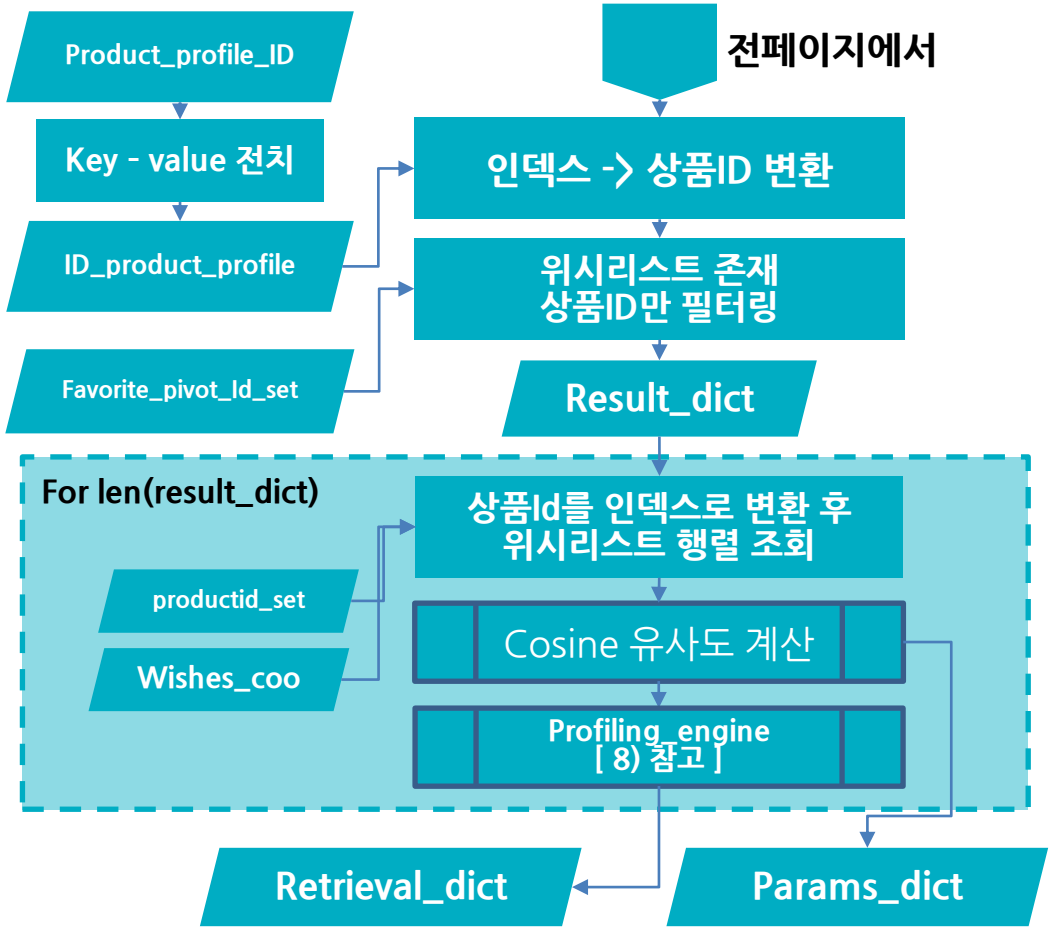
-. 데이터 처리 프로세스

④ 직전 추출한 상품목록 중 위시리스트 존재 상품만 필터링
⑤ result_dict 요소 개수만큼 순회
- 상품ID로 위시리스트 행렬 데이터 추출
- 코사인 유사도 계산 후 profiling_engin을 통해 [Top-K] or [x > Threshold]인 상품 목록 추출 ➔ retrieval_dict

⑥ 상품목록에 해당하는 코사인 유사도 추출 ➔ params_dict

-. 산출 데이터

① retrieval_dict, params_dict



데이터 흐름

2. A-3 알고리즘 데이터 흐름

1) 흐름도 얼개

-. 데이터 처리 프로세스

① **상품 정보 전처리** : 카테고리 정보를 Json에서 추출하여 orderseries와 Merge

②-1 **카테고리 데이터 전처리** : 카테고리 정보를 Pivoting하여 A_1 알고리즘에 전달

②-2 **위시리스트 전처리** : orderseries와 wishlist를 Merge

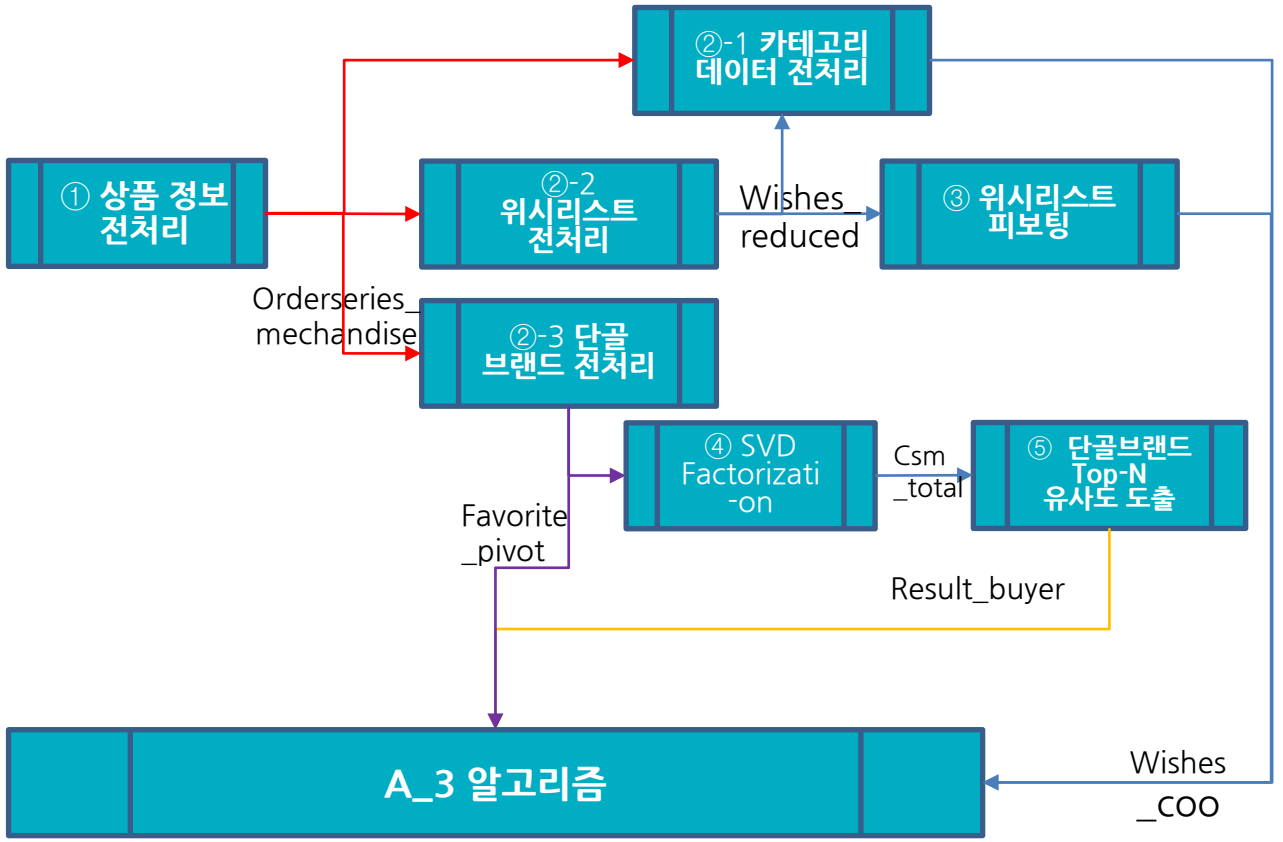
②-3 **단골브랜드 전처리** : orderseries와 단골브랜드List를 Merge 후 기준에 따라 소수 사례는 Filtering 수행

③ **위시리스트 피보팅** : wishes_reduce를 피보팅하여 희소행렬(coo_matrix)로 변환

④ **SVD Matrix Factorization** : 암시적 정보(Implicity Information)를 표면화하기 위해 SVD 행렬분해 후 재조합 수행

⑤ **단골브랜드 Top -N 유사도 도출** : 단골브랜드(Favorite_pivot) 기반으로 각 바이어ID별 최다 유사도 바이어ID 짝을 도출

※ <①, ②-1, ②-2, ②-3, ③, ④, ⑤>는 A-1 알고리즘의 설명 참조



데이터 흐름

1. A-3 알고리즘 데이터 흐름

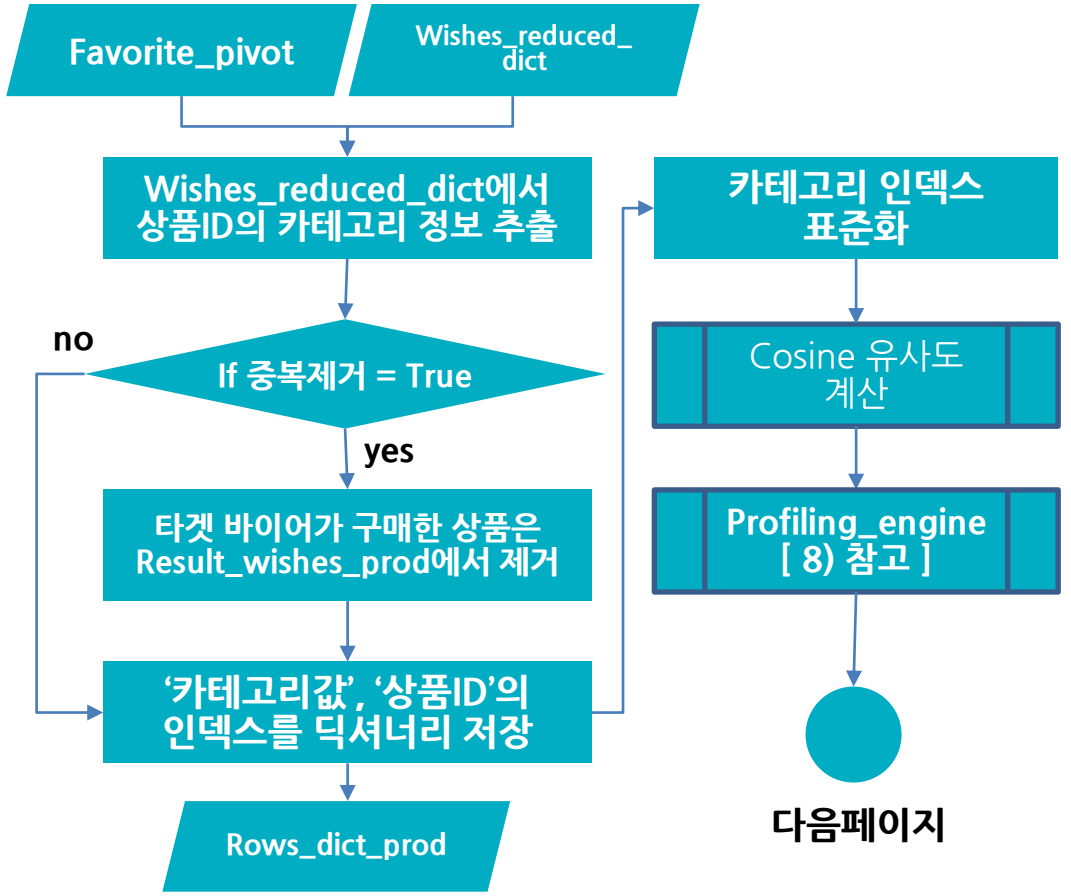
1) A-3 알고리즘부

-. 투입 데이터

① favorite_pivot
② wishes_coo
③ result_buyer
④ wishes_result_dict
⑤ productid_dict
⑥ IdBuyer_dict

-. 데이터 처리 프로세스

① favorite_pivot, wishes_reduced_dict 준비
② 타겟바이어, 유사바이어들이 위시리스트에 추가한 상품들의 카테고리 정보들을 추출 후 Flattened ➔ 카테고리 정보 행렬
③ 카테고리 정보 행렬의 '카테고리값', '상품ID'의 인덱스-값 딕셔너리를 저장
④ 카테고리 인덱스 표준화
- 각각의 상품ID가 갖고있는 카테고리 정보가 다르기 때문에 인덱스 정보가 다름
- 이를 하나로 통합(표준화)하고 빈공간은 0행렬로 채움



데이터 흐름

1. A-3 알고리즘 데이터 흐름

11) A-3 알고리즘부(계속)

-. 투입 데이터

① favorite_pivot
② wishes_coo
③ result_buyer
④ wishes_result_dict
⑤ productid_dict
⑥ IdBuyer_dict

-. 데이터 처리 프로세스

④ 직전의 상품목록 인덱스를 상품ID로 변환 ➔ result_dict
⑤ result_dict 요소 개수만큼 순회
- 상품ID로 위시리스트 행렬 데이터 추출
- 코사인 유사도 계산 후 profiling_engin을 통해 [Top-K] or [x > Threshold]인 상품 목록 추출 ➔ final_dict

⑥ final_dict 처리
- 인덱스를 바이어ID로 변환 ➔ retrieval_dict
- Final_dict 상품ID에 해당하는 코사인 유사도 ➔params_dict

-. 산출 데이터

① retrieval_dict, params_dict