

| 올리브영 리뷰 요약 데이터를 활용한 추천시스템 성능 개선

최종 발표

4조

이영현
강채원
김주은
임세은

목 차

1. 팀 소개

2. 주제 선정

3. 데이터 수집

4. 전처리 & EDA

5. 리뷰 데이터 처리

a. summarization : Kobart, lexrank, chatGPT

b. Embedding : tfidf, kobart, word2vec

6. GNN 기반 추천 시스템

7. 실험 결과

8. 프로젝트 의의 & 활용 방안

팀 소개



이영현
분석 20기



강채원
분석 20기



김주은
분석 20기



임세은
분석 20기

주제 선정

기존 추천 시스템 - ex) MovieLens 100k movie ratings

사용자 정보

id, 성별, 연령, 직업 등

아이템 정보

id, 제목, 개봉일,
장르별 원핫인코딩 컬럼

구매 정보

사용자 id, 아이템 id,
평점, timestamp

자연어 데이터를 크게 활용하지 않음

" 추천 시스템에 리뷰 데이터를
활용할 수는 없을까? "

- KoBART를 이용한 요약
- Lexrank를 이용한 요약
- Word2Vec를 이용한 임베딩
- TF-IDF를 이용한 임베딩
- ...

데이터 수집

상품 정보


[아누아 >](#) [스킨케어 >](#) [토너/로션/올인원 >](#) [스킨/토너 >](#)

[단독기획] 아누아 어성초 77 수딩 토너 350ml 기획(+토너40ml+패드2매+선크림10ml 증정)

30,500원 **22,800원** [혜택 정보](#)

[세일](#) [오늘드림](#)
75명이 보고있어요

내용물의 용량 또는 중량	토너350ml+패드2매+토너40ml+선크림10ml
제품 주요 사양	모든 피부용.


최고

총 6,136 건
4.7 점
★★★★★

80%
5점

15%
4점

4%
3점

1%
2점


1%
1점

피부타입
건성에 좋아요 23%
복합성에 좋아요 63%
지성에 좋아요 14%

피부고민
보습에 좋아요 22%
진정에 좋아요 78%
주름/미백에 좋아요 1%

자극도
자극없이 순해요 77%
보통이에요 23%
자극이 느껴져요 0%


사용자 정보 및 구매 정보



핑크올립
TOP1371
지성 · 원톤 · 민감성

364 도움 1 팔로워 1 팔로잉


팔로우

컬렉션 2


남남


패드맛집

누적 리뷰 183 [최근작성순](#)


필리밀리
필리밀리 아쿠아핏 여성용 면도기 (색상랜덤발송)

★★★★★ 작성일자 2023.05.22

한달이상사용

물로만 면도가 가능하고 5종 면도날이여서 깔끔하게 제모를 해줍니다

도움이 돼요 1

신고하기

상품 376개, 사용자 3576명, 구매 정보 5427건에 대한 데이터 수집

전처리 & EDA

구매 정보

리뷰자	상품명	작성일자	평점	리뷰
eHNCNmFVY1ladXNQ V21xcS9zT0J1QT09	라운드랩 자작나무 수분 마스크 1매	2023.05.18	4	이벤트 할때 한장사서.써보고 괜찮아 서 다시 구입합니다..저는 지성피부에 요..나이 50인데도 아직 피부가.번들거 리죠....(생략)

- 리뷰자 id 및 상품명 : 숫자로 매핑
- 상품명이 상품 정보에 존재하는 경우만 사용

상품 정보

goodsno	상품명	브랜드	가격	용량	주요사양	성분	평점	증정 여부
A000000 006564	우르오스 스킨로션 200ml	우르오스	23,700	200ml	지복합성 피부를 위한 워터타입 스킨케어 [2 in 1]	정제수, 에탄올, 펜틸렌글라이콜, 글리세린, 베타인, 피이지-6, 피이지-32, ...	4.8	0

- One hot encoding
 - 성분 : 30번 이상 등장하는 성분
 - 브랜드
- 평점 : 최소값으로 fillna
- 총 152개 feature

전처리 & EDA

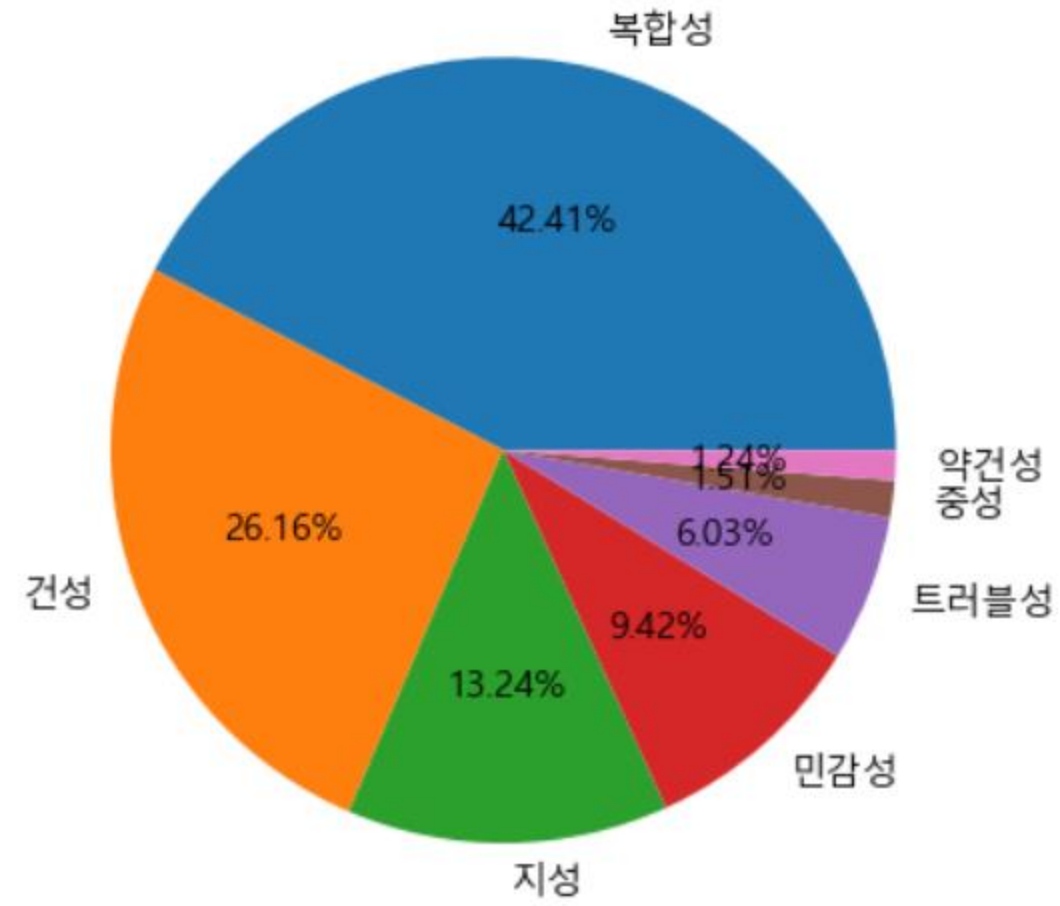
사용자 정보

리뷰자	type1	type2	type3	type4	랭킹	리뷰
eHNCNmFVY1ladXNQV21xcS9zT0J1QT09	지성	웜톤	모공	트러블	0.056	이벤트 할때 한장사서.써보고 괜찮아서 다시 구입합니다..저는 지성 피부예요..나이 50인데도 아직 피부가.번들거리죠....(생략)

- type1: 피부타입(건성, 지성, 복합성...)
- type2 : 퍼스널컬러(웜톤, 쿨톤, 겨울쿨톤...)
- type3 & 4 : 피부고민 → 합치기
- One-hot encoding : type 1 ~ 4
- 랭킹 : 숫자가 클수록 순위가 높도록
- 리뷰 : 구매 정보의 리뷰를 사용자별로 groupby

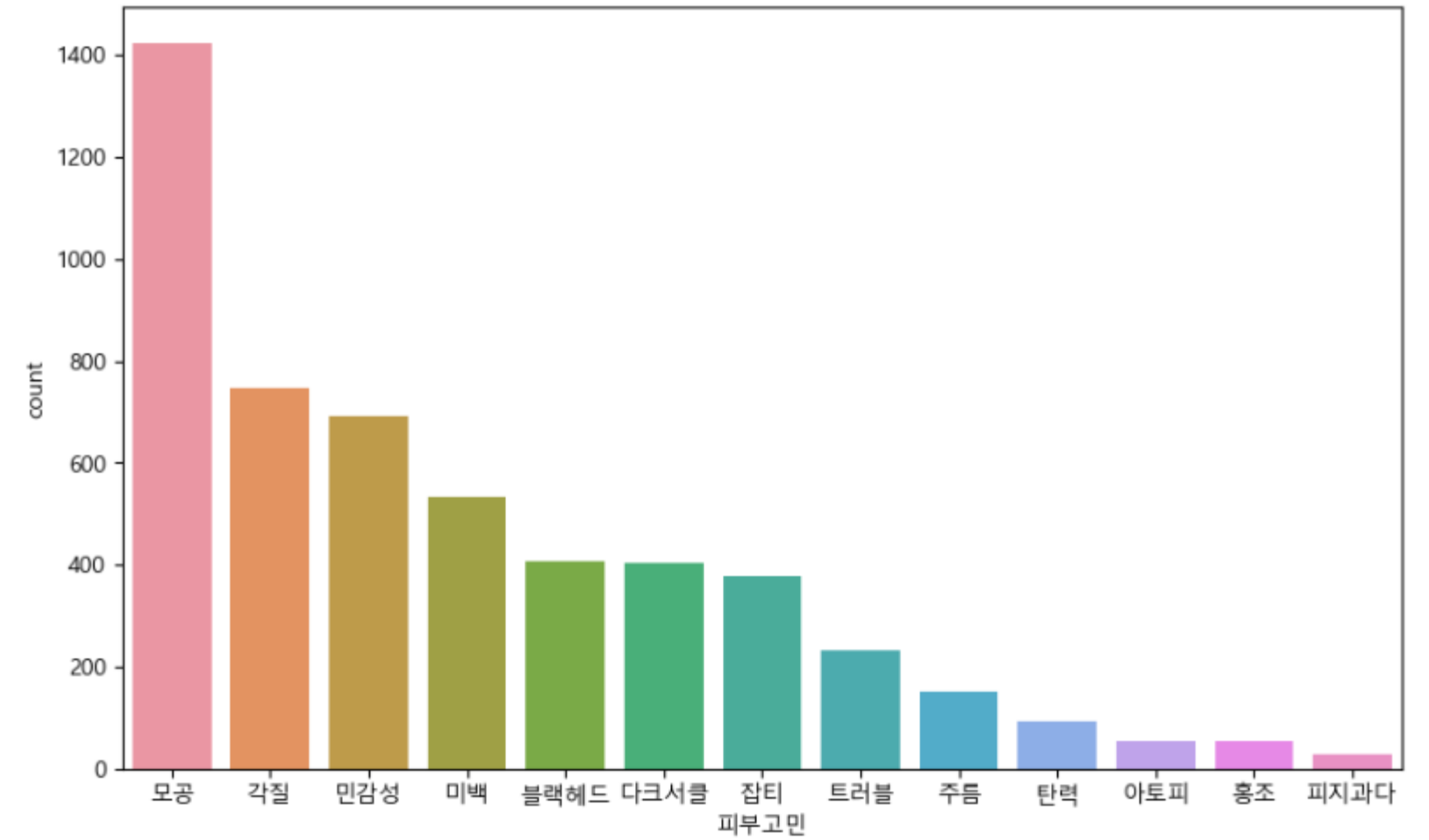
전처리 & EDA

사용자 피부타입 분포



- 복합성 > 건성 > 지성 > 민감성 > 트러블성 > 중성 > 약건성

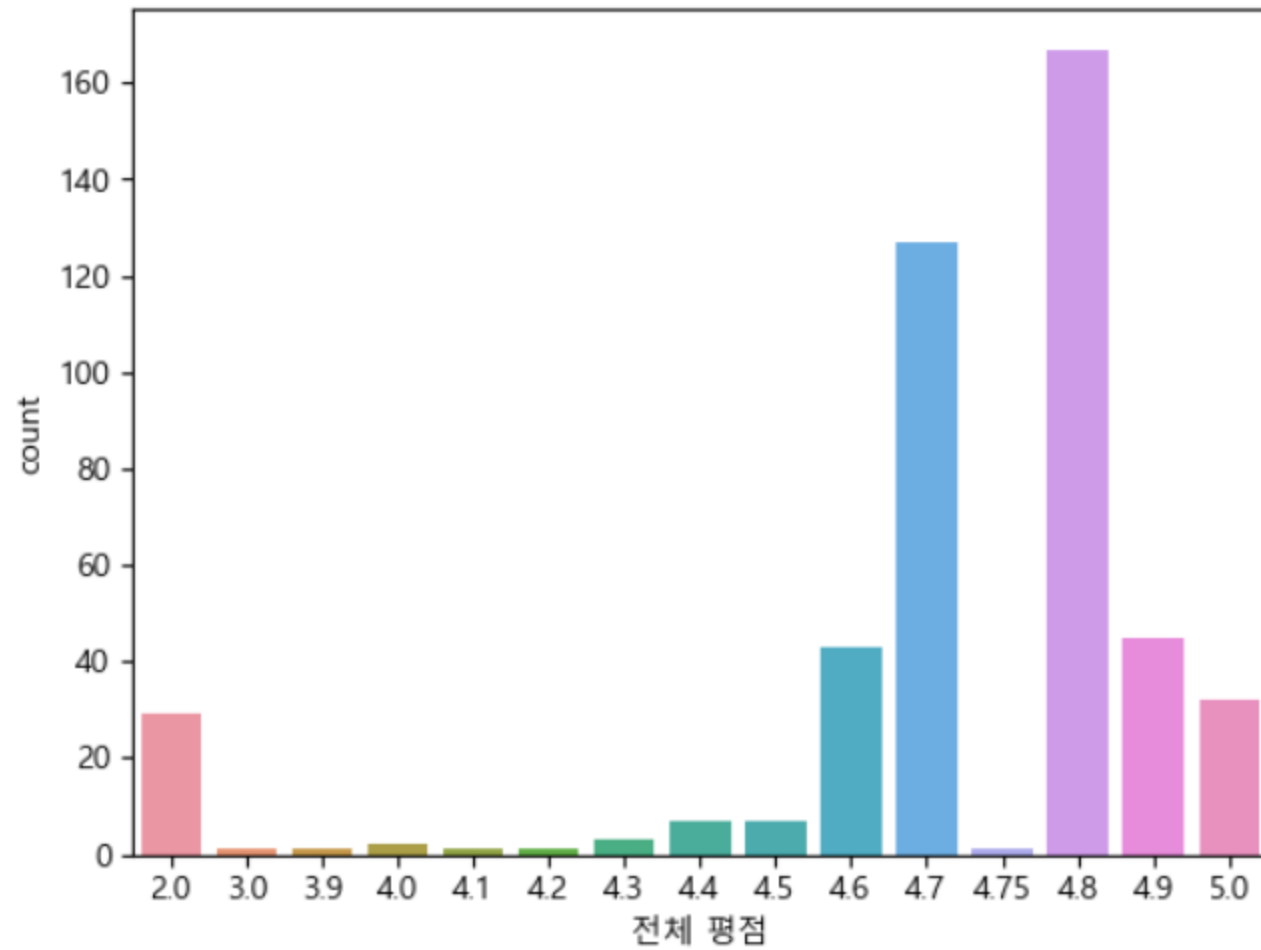
사용자 피부 고민



- 모공 > 각질 > 민감성 > 미백 > 블랙헤드 > 다크서클 > 잡티 > 트러블 > 주름 > 탄력 > 아토피 > 홍조 > 피지과다

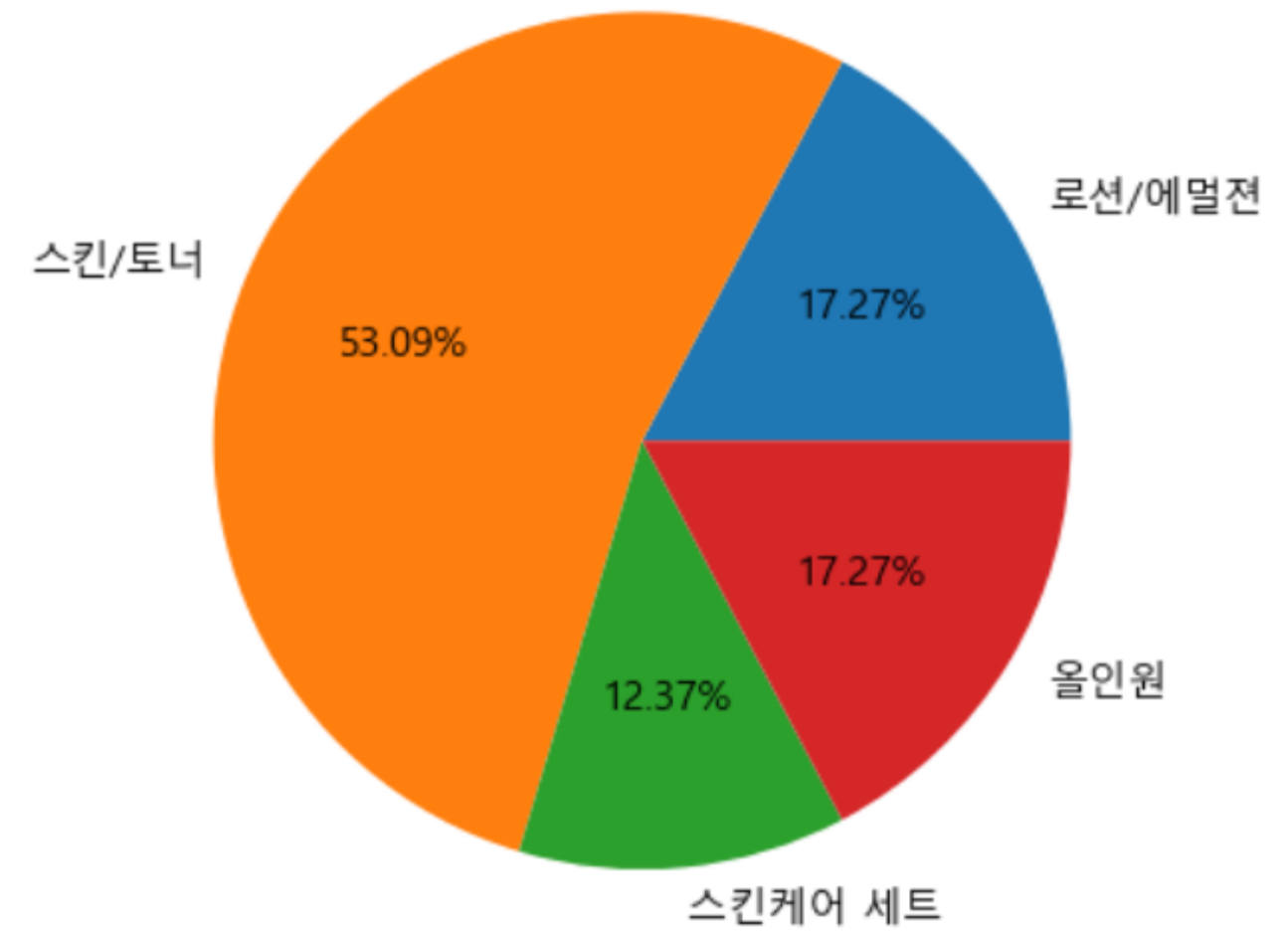
전처리 & EDA

상품별 평균 평점



- 평균 평점이 4.8 이상인 상품: 약 52.25 %
- 전반적으로 높은 평점

카테고리 분포



- 스킨/토너 > 울인원 = 로션/에멀전 > 스킨케어 세트

리뷰 데이터 처리

사용자별 리뷰 concat



Summarization

- 1. KoBART
- 2. Lexrank
- 3. chatGPT



Embedding

- 1. KoBART
- 2. TF-IDF
- 3. Word2Vec



Embedding 결과 : 사용자별 Feature로 사용

리뷰 데이터 처리

Summarization

- KoBART : 사전학습된 huggingface 모델
- ChatGPT : OpenAI의 API
- Lexrankr : Lexrank의 한국어 버전
 - Google의 초기 검색엔진에 적용되었던 PageRank 알고리즘 적용
 - extractive summarization
 - 문장 간 유사도 및 중요도를 기반으로 요약에 포함할 문장 추출
 - TF-IDF 기반 코사인 유사도 계산
 - 유사도가 높은 문장 = 중요도가 높은 문장
 - 전처리 시 주의사항
 - 문장을 추출하기 때문에 텍스트에 escape string \n이 포함되어야

본문	KoBART	ChatGPT	Lexrankr
처음 바를때는 따가웠는데 자고 일어나도 유분기 없이 보송하게되어서좋고...	처음 바를때는 따가웠는데 자고 일어나도 유분기 유분 기 유분기 없이 보송하게 되어서좋고대용량이라 더 좋아용	처음 바를 때 따가웠지만 자고 일어나면 보송하게 되어 좋고, 대용량이라 더 좋아용	처음 바를때는 따가웠는데 대용량이라 더 좋아용
엄마 선물로 사드렸는데 보 습력도 좋고 향도 만족한다 며 좋아하시네요...	엄마 선물로 사드렸는데 보 습력도 좋고 향도 만족한다 며 좋아하시네요 가격도 용 량 대비 괜찮은 거 같아요 엄마 선물로 사드렸는데 엄 마 선물로 사드렸는데	엄마 선물로 샀는 보습력과 향이 만족하며 가격도 용량 대비 괜찮은 제품이라 엄마 가 써보고 좋다고 하셨다	엄마 선물로 사드렸는데 가 격도 용량 대비 괜찮은 거 같아요 엄마 선물로 샀어요
신랑이 꾸준히 사용하는 미 프 올인원로션이에요 가성비 비 넘치는 구성이라 이깁없 이 짹 사용해도 부담없는 알찬구성이에요	끈적임 없어 기초제품 꾸준 히 못쓰는 신랑도 미프는 잘 쓰더라구요 보습도 좋아 찬바람 부는 요즘날씨에 피 부관리에 최적입니다	신랑이 꾸준히 사용하는 미 프 올인원로션은 가성비가 뛰어나고 끈적임 없고 자극 없이 순하고 복합성 피부가 무난하게 사용할 수 있어 여름철까지 잘 쓸 수 있다.	신랑이 꾸준히 사용하는 미 프 올인원로션이에요 넘 좋 습니다 끈적임 없어 기초제 품 꾸준히 못쓰는 신랑도 미프는 잘 쓰더라구요

리뷰 데이터 처리

Embedding

- TF-IDF : shape (3576, 3000+)
- KoBART : last hidden state의 평균값 사용
 - shape (3576, 768)

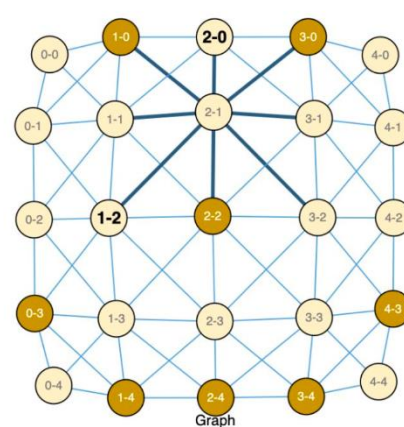
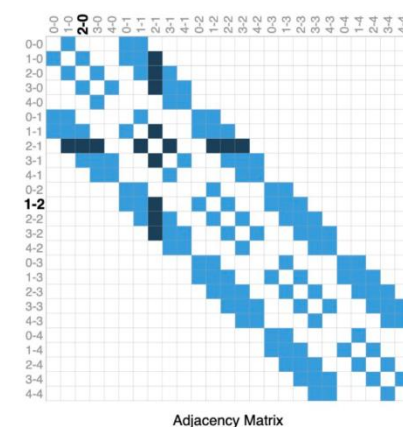
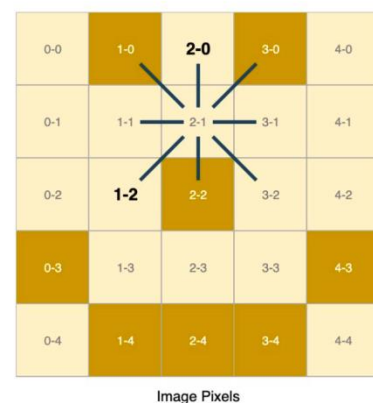
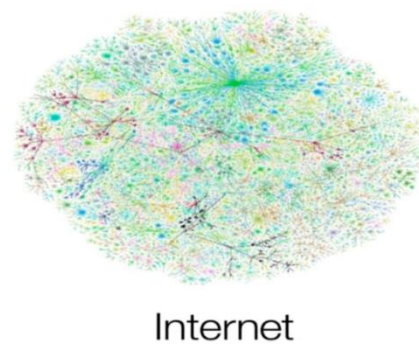
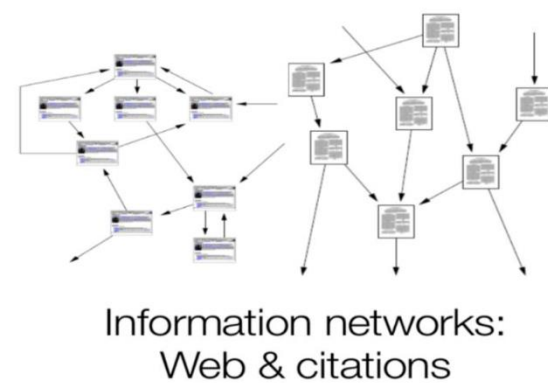
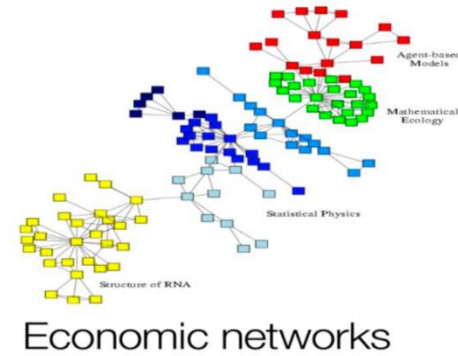
```
def get_embedding(txt, num = 768):
    embeddings = []
    tokens = kobart_tokenizer.tokenize(txt)
    input_ids = kobart_tokenizer.convert_tokens_to_ids(tokens)
    input_ids = torch.tensor([input_ids])
    try:
        with torch.no_grad():
            lhs = model(input_ids)[0].mean(dim=1) #평균 이용
    except IndexError: #리뷰가 없는 경우 0으로 채워진 array 생성
        lhs = np.zeros(num)
    embeddings.append(lhs.tolist())
    return embeddings[0]
```

- Word2Vec : 단어 벡터의 평균값 사용
 - shape (3576, 100)

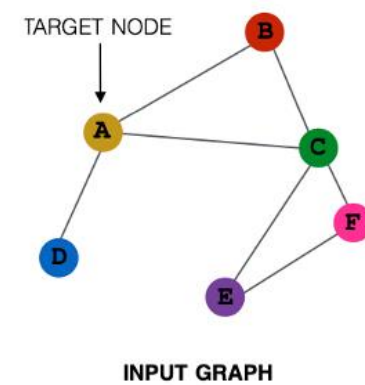
```
def get_features(words, model, num_features):
    # 출력 벡터 초기화
    feature_vector = np.zeros((num_features), dtype=np.float32)
    num_words = 0
    # 어휘사전 준비
    index2word_set = set(model.wv.index_to_key)
    for w in words:
        if w in index2word_set:
            num_words += 1
            # 사전에 해당하는 단어에 대해 단어 벡터를 더함
            feature_vector = np.add(feature_vector, model.wv[w])
    # 문장의 단어 수만큼 나누어 단어 벡터의 평균값을 문장 벡터로 함
    feature_vector = np.divide(feature_vector, num_words)
    return feature_vector

def get_dataset(reviews, model, num_features):
    dataset = list()
    for s in reviews:
        dataset.append(get_features(s, model, num_features))
    reviewFeatureVecs = np.stack(dataset)
    return reviewFeatureVecs
```

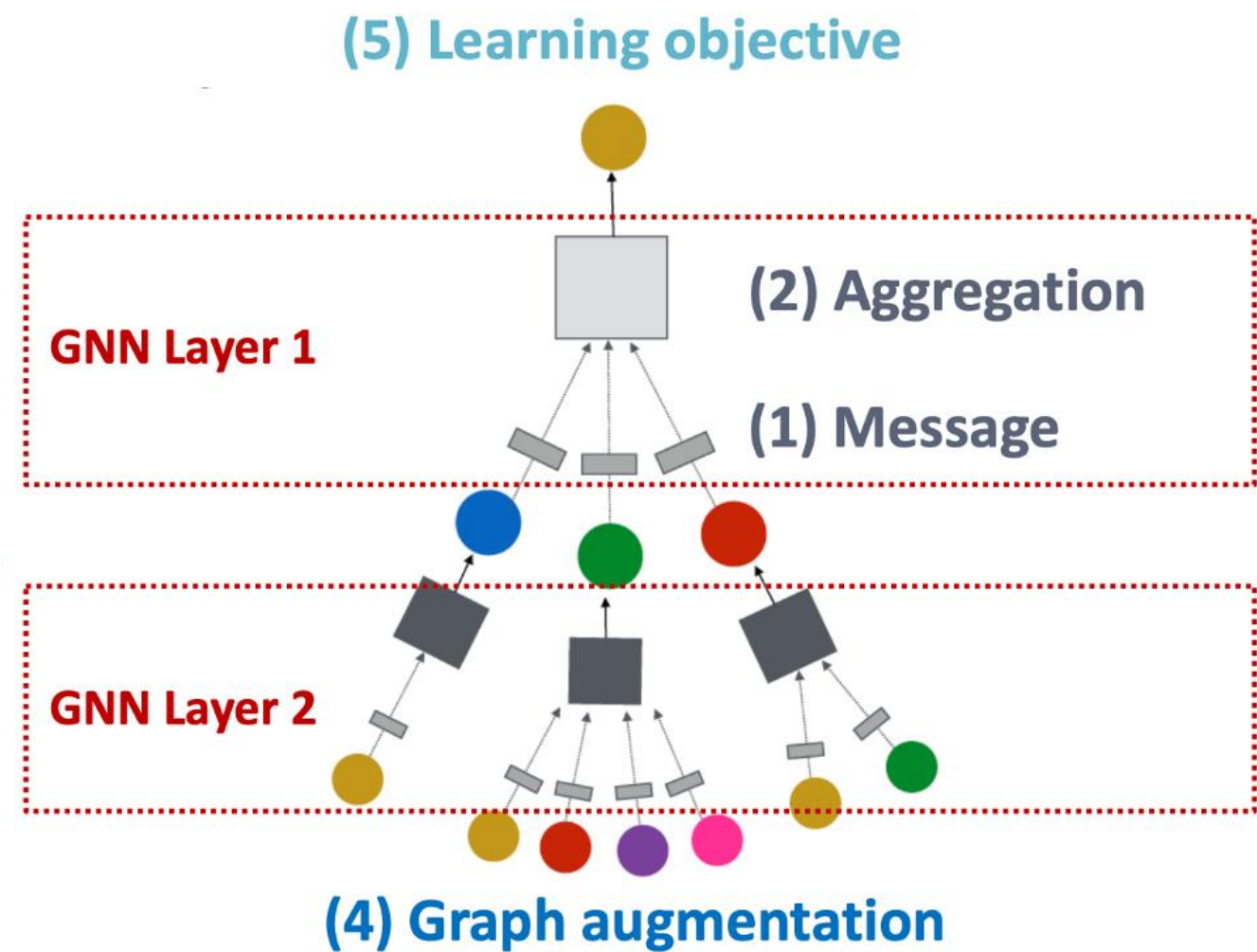

GNN(Graph Neural Network)



<Graph Data>

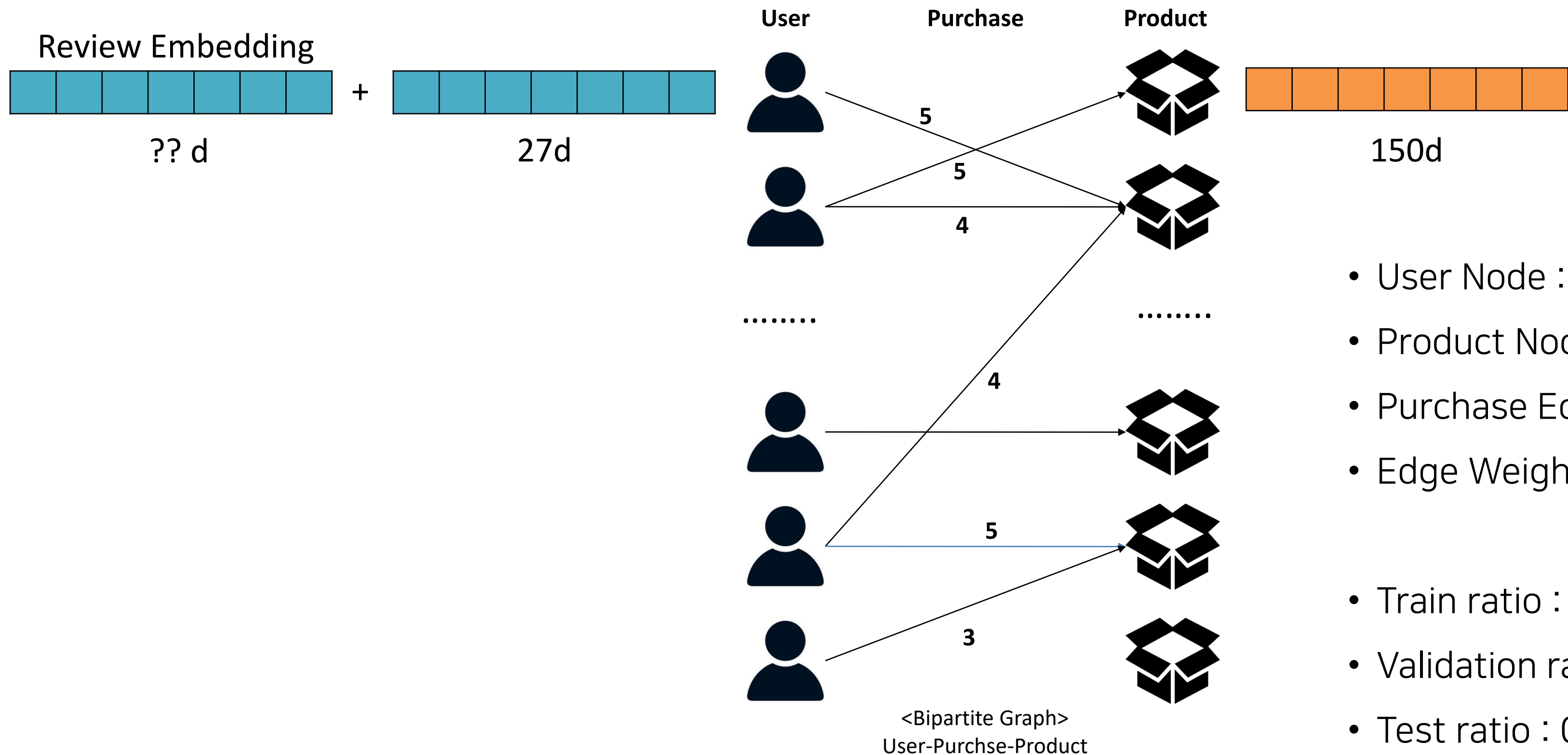


(3) Layer connectivity



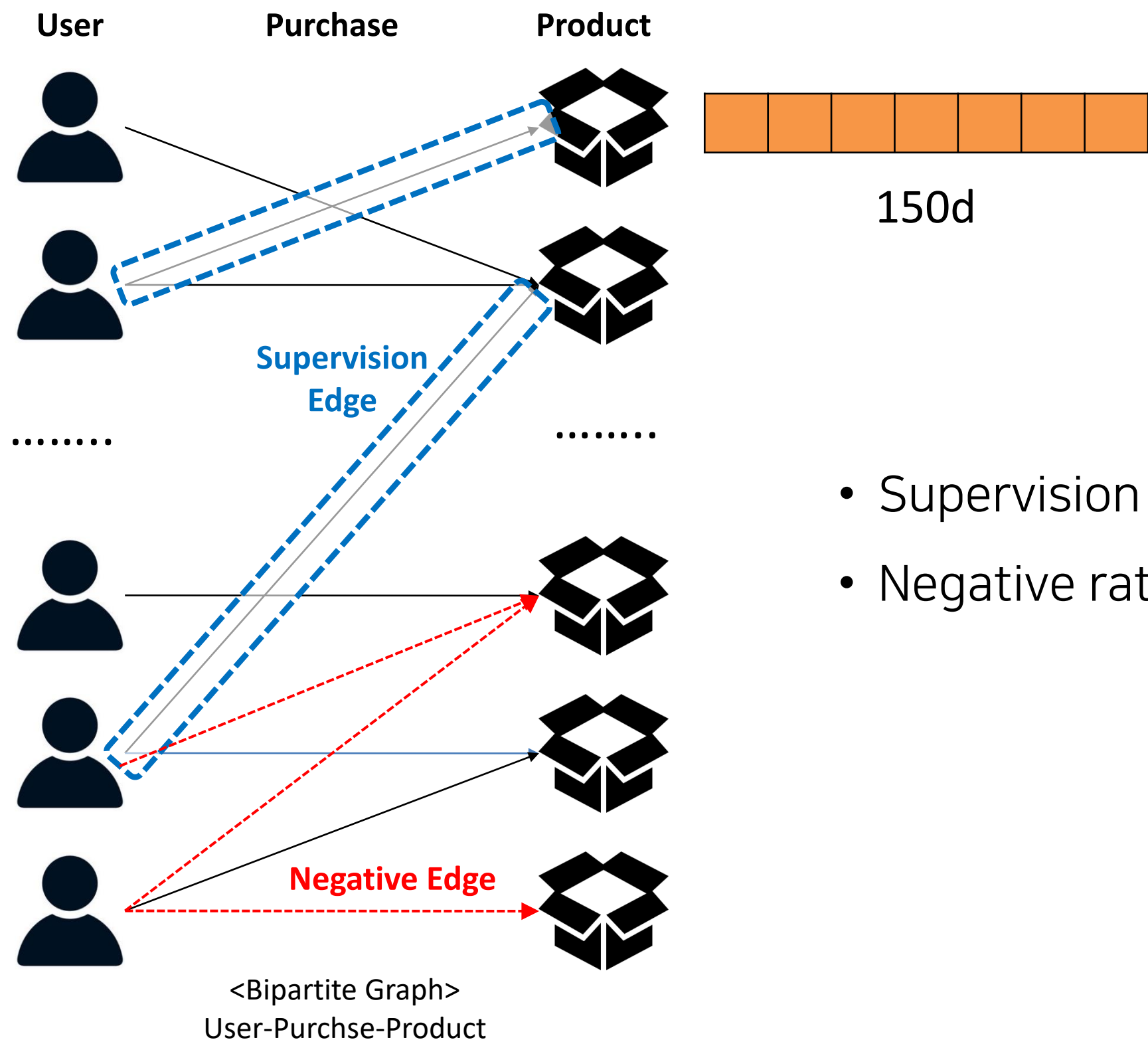
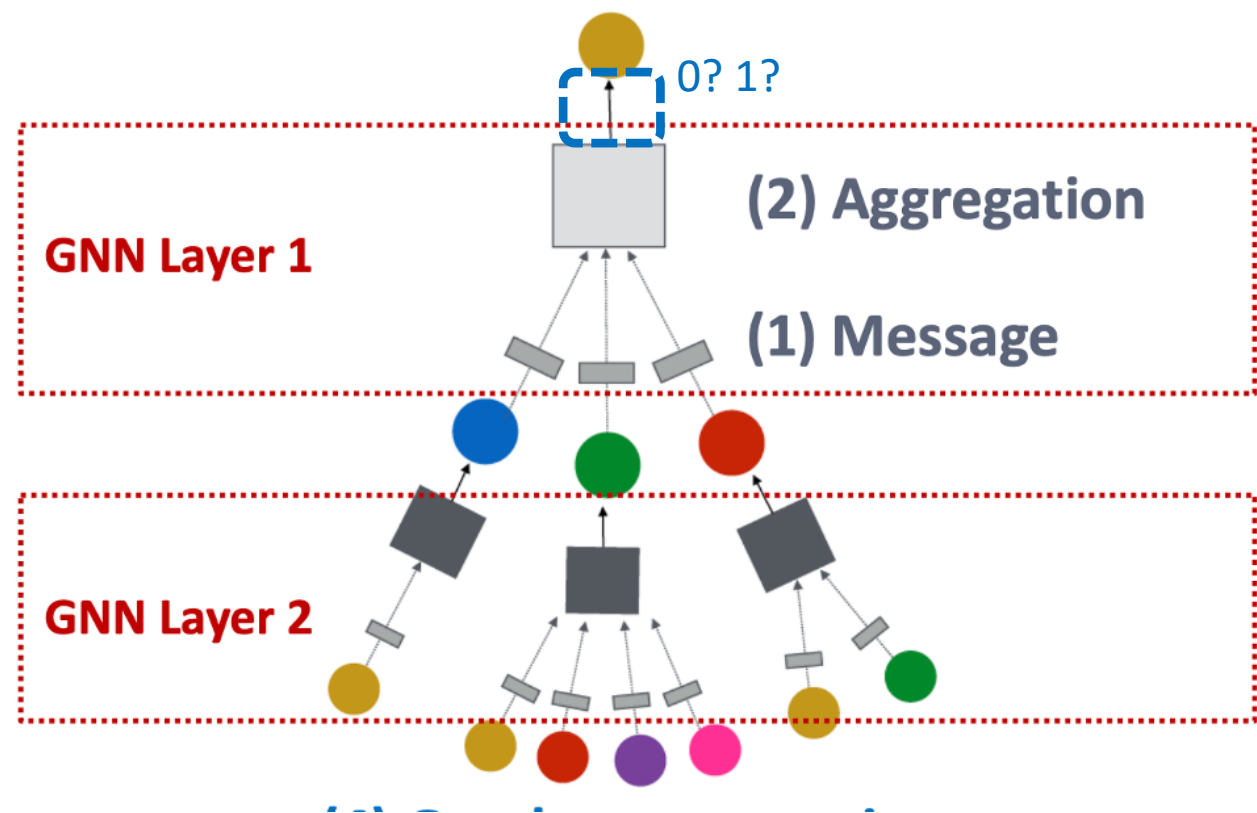
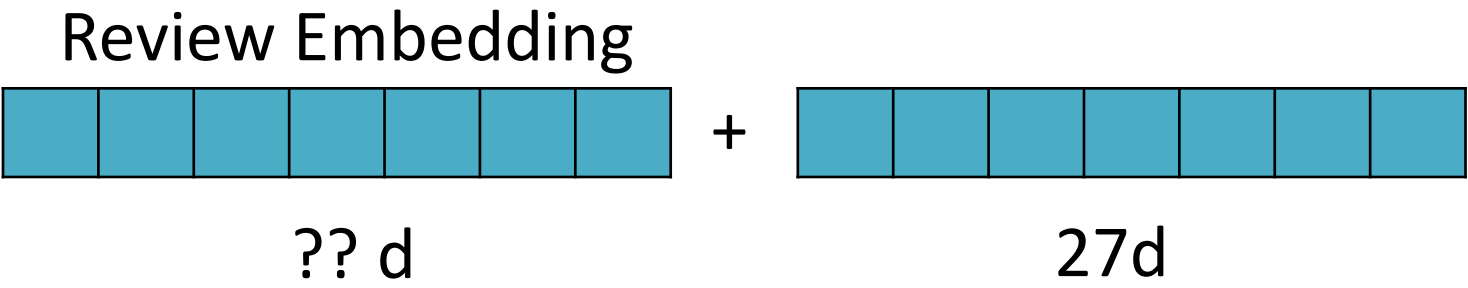
<GNN Layer>

Graph 데이터 구성



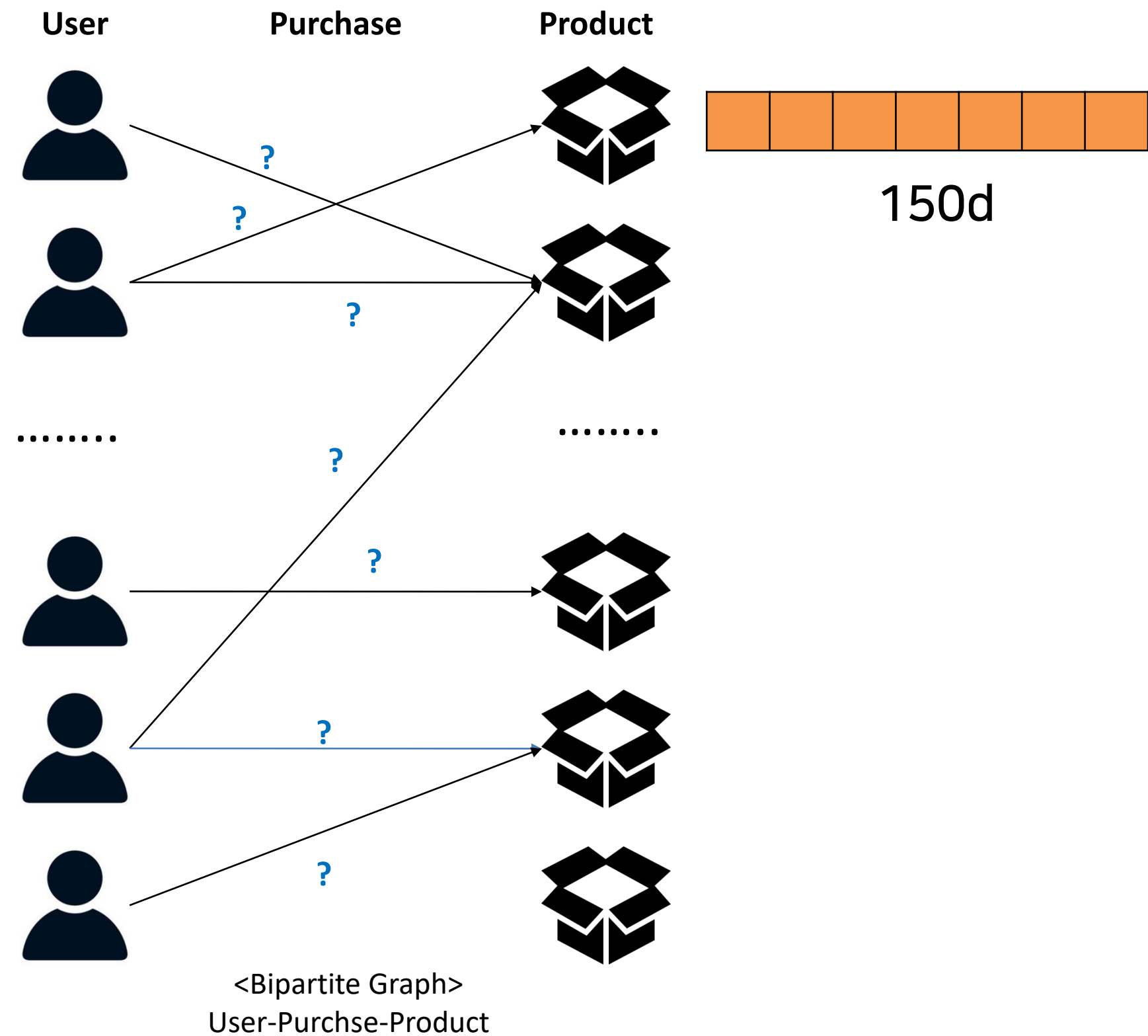
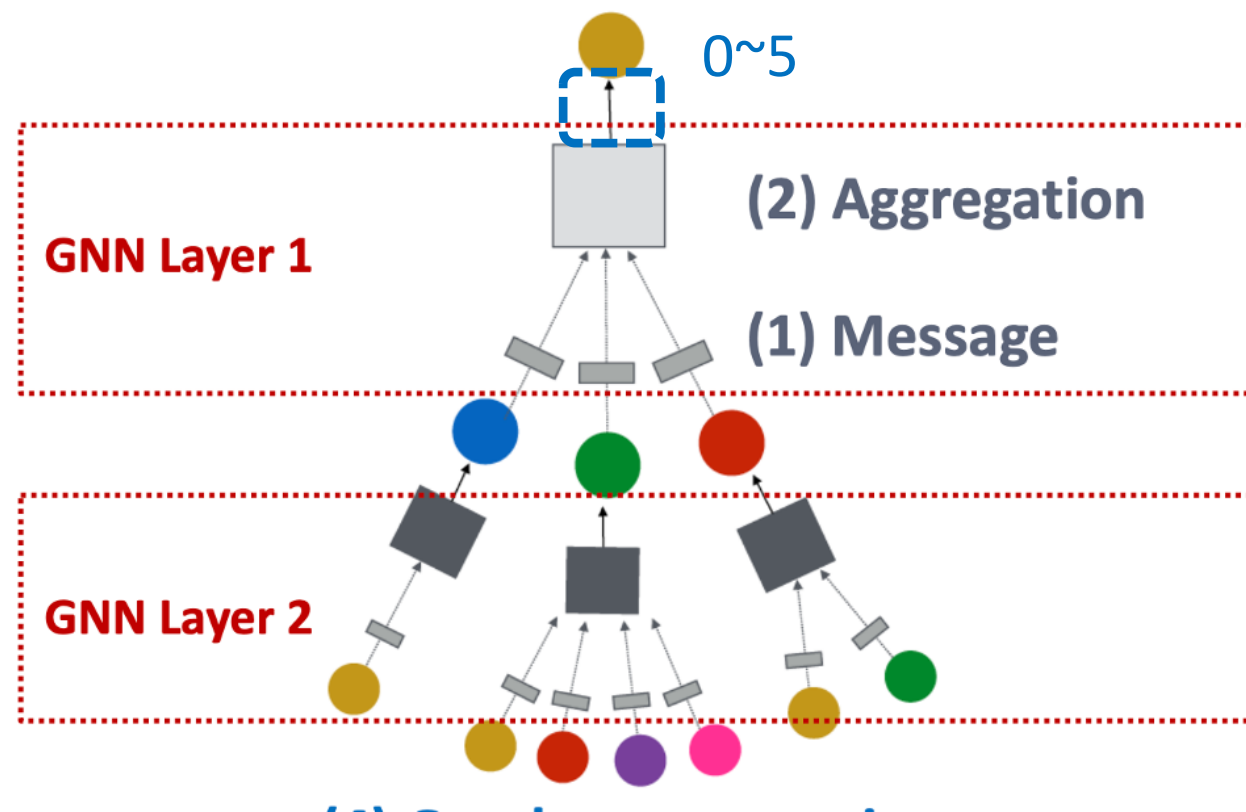
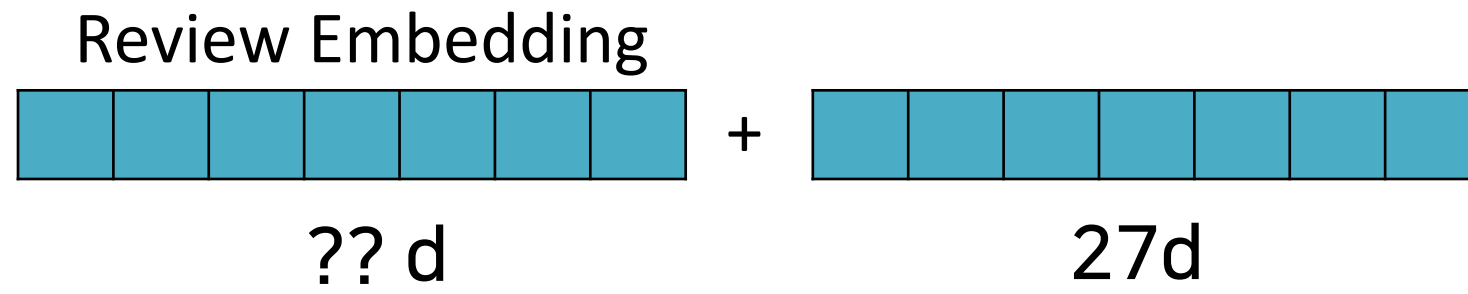
- User Node : 3576
- Product Node : 468
- Purchase Edge : 5424
- Edge Weight : 1~5
- Train ratio : 0.8
- Validation ratio : 0.1
- Test ratio : 0.1

GNN Task : Link Prediction



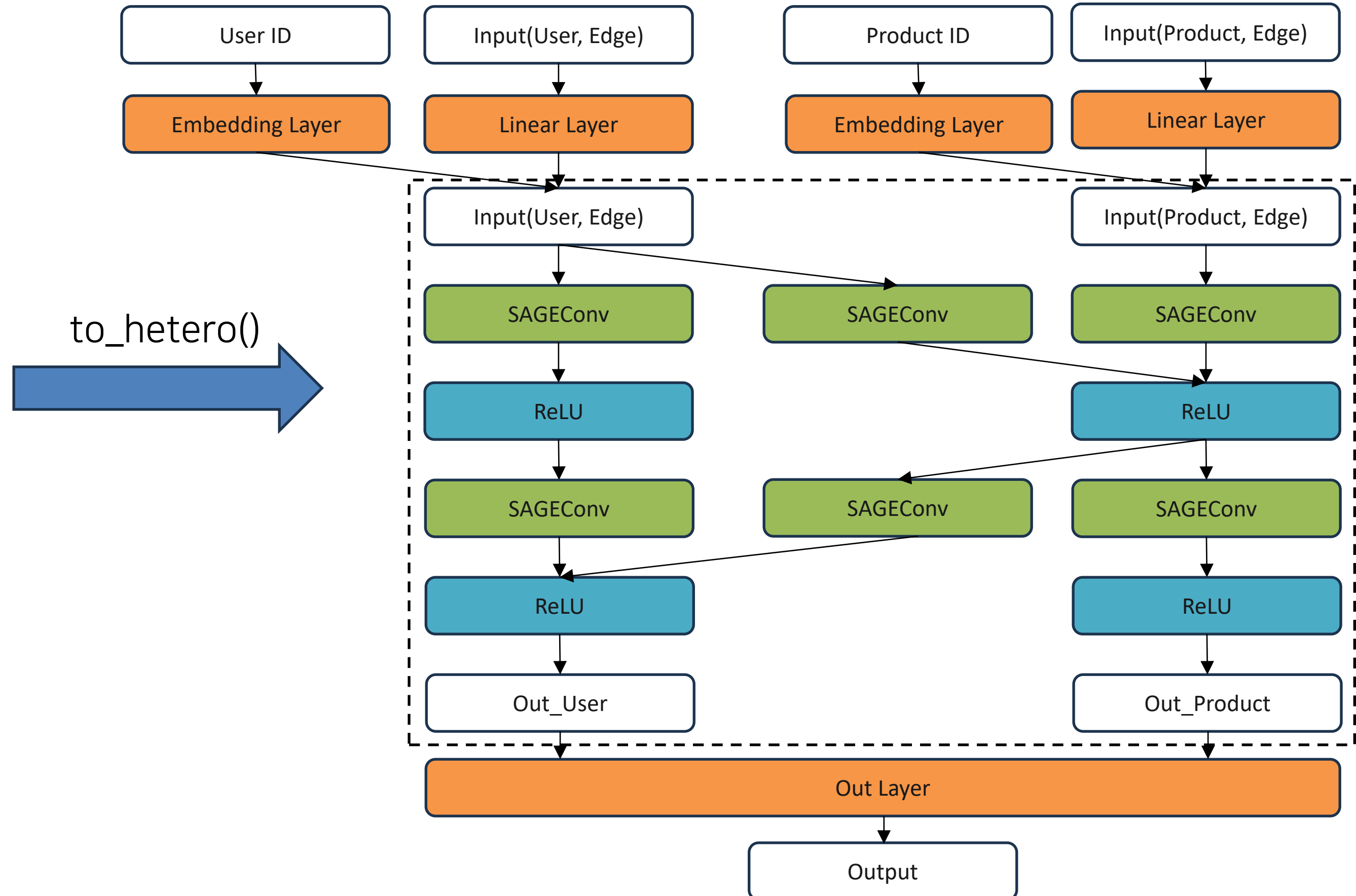
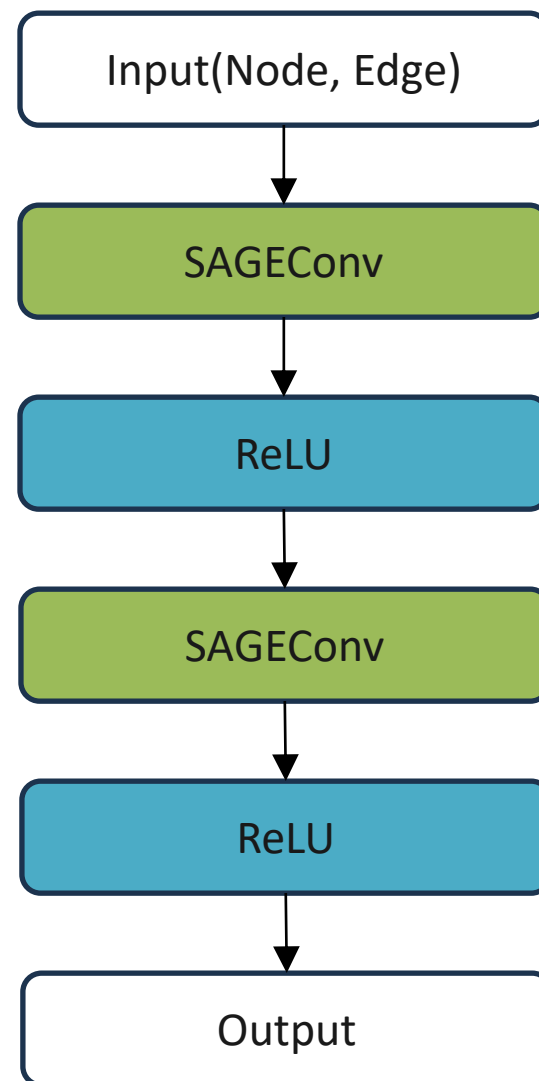
- Supervision ratio : 0.3
- Negative ratio : 2

GNN Task : Link Regression



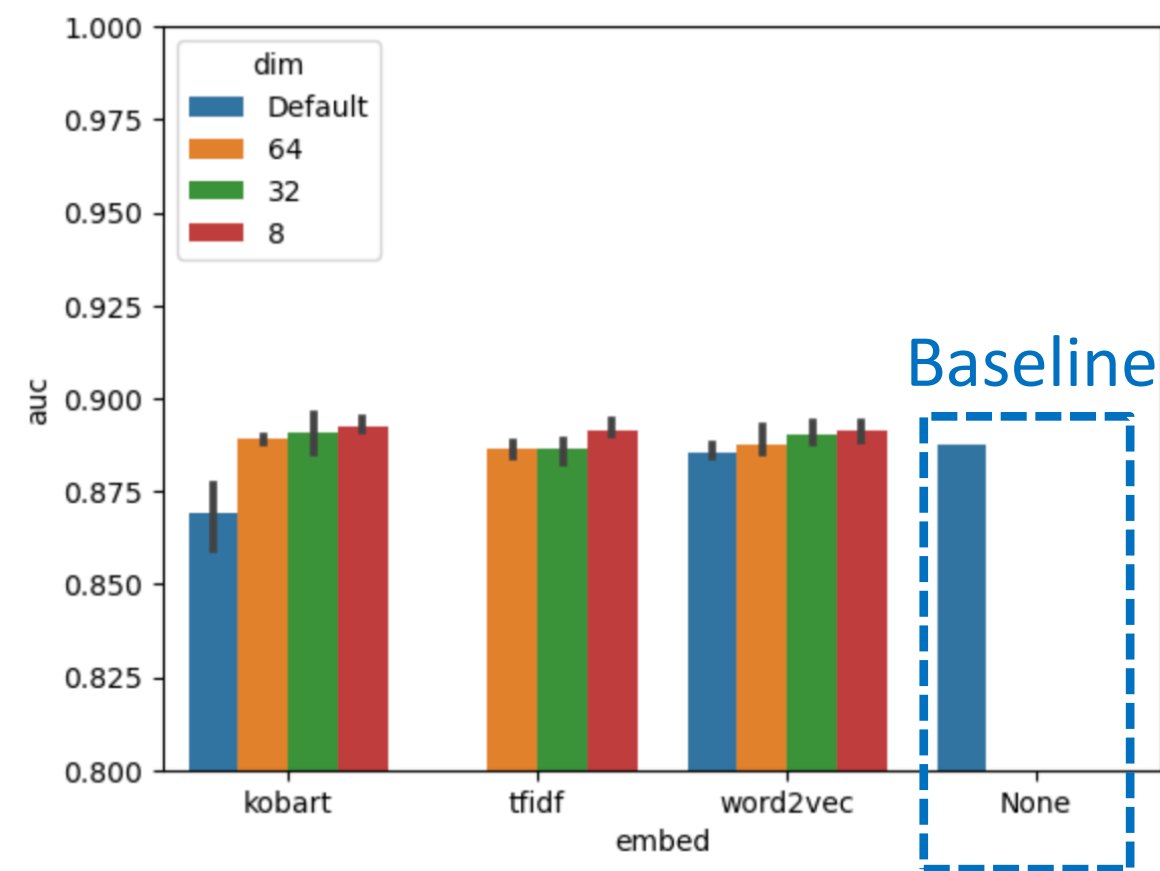
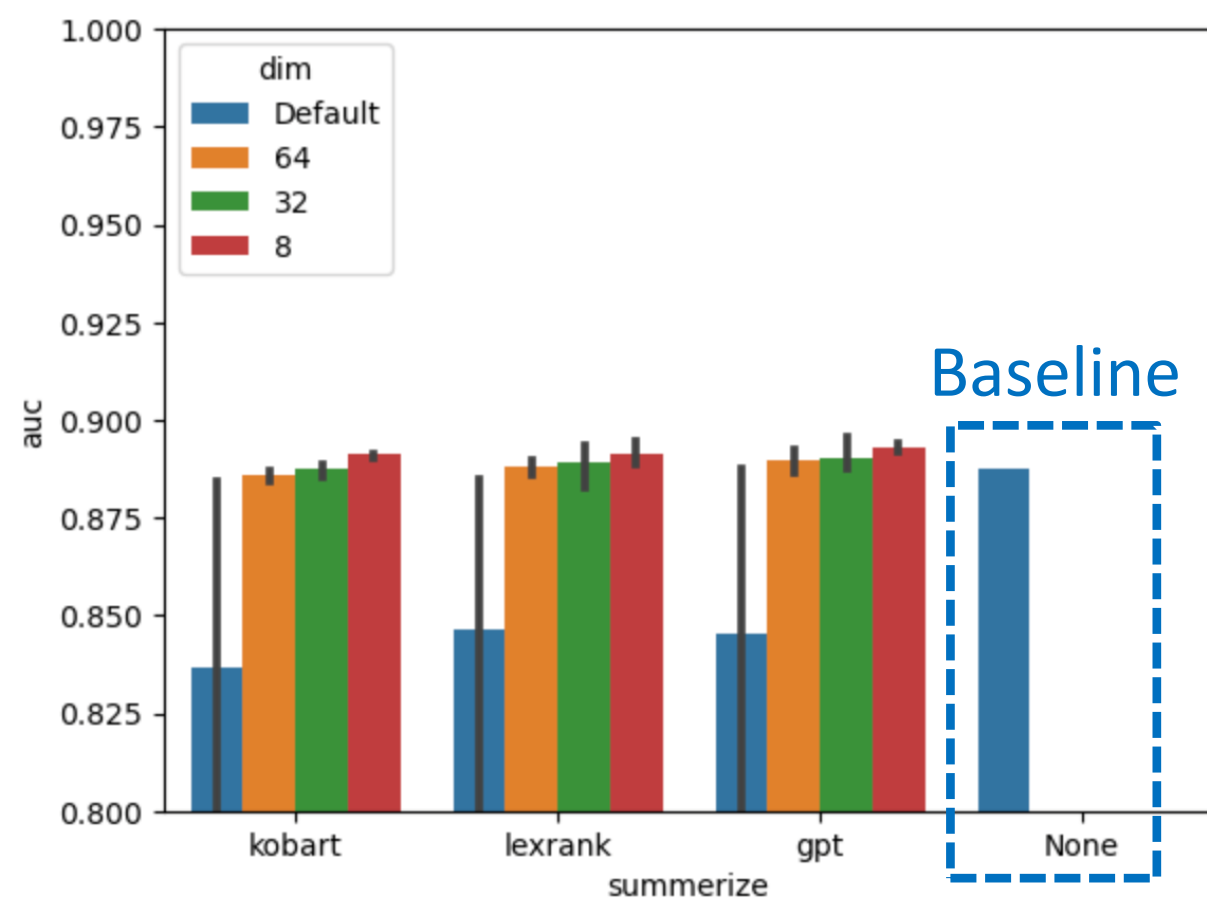
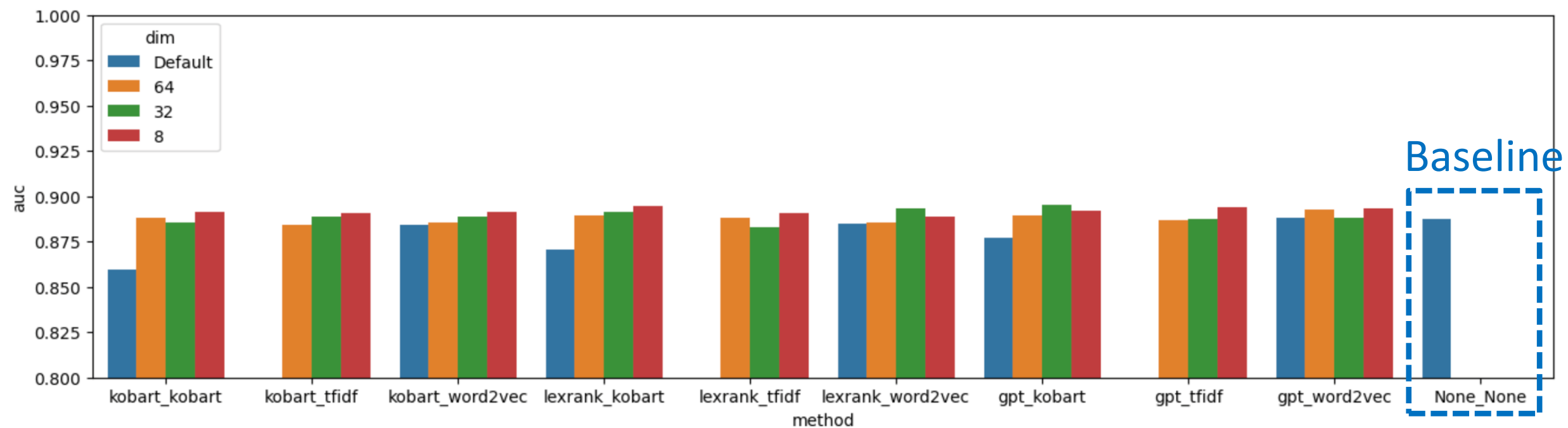
Heterogeneous GNN Model

- Homogeneous Model



실험 결과

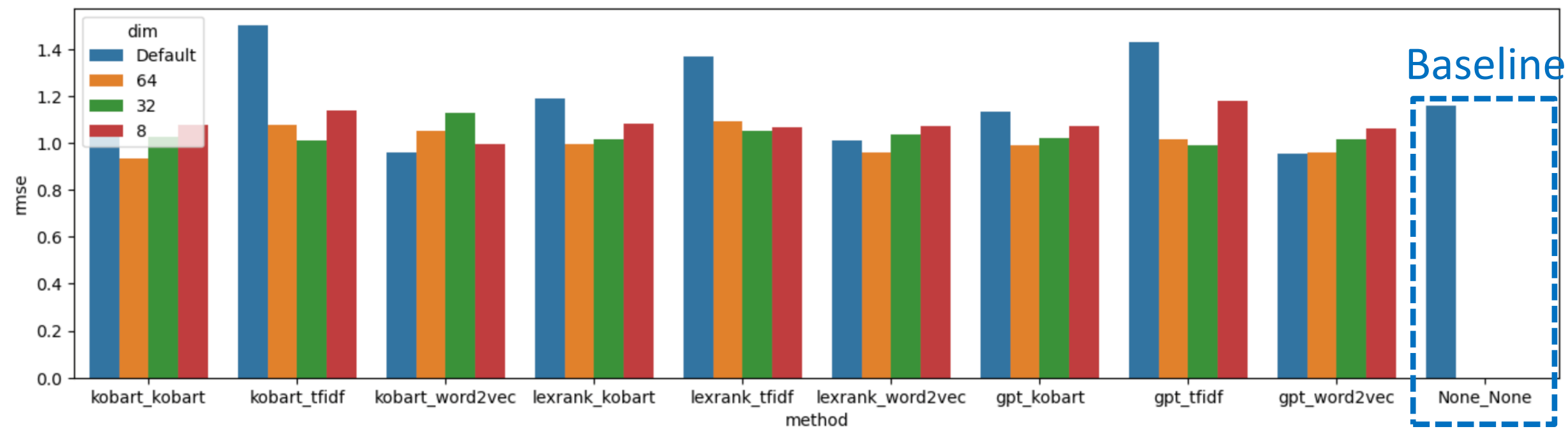
GNN Task : Link Prediction



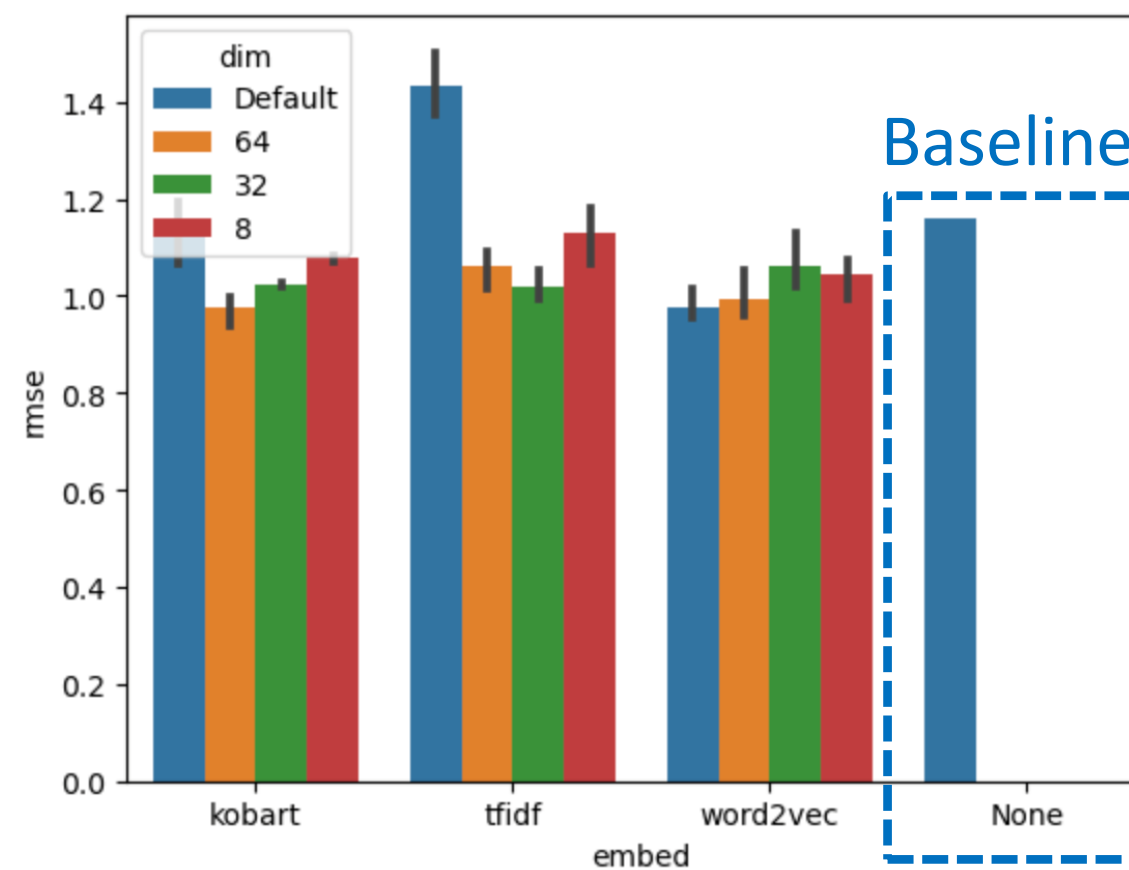
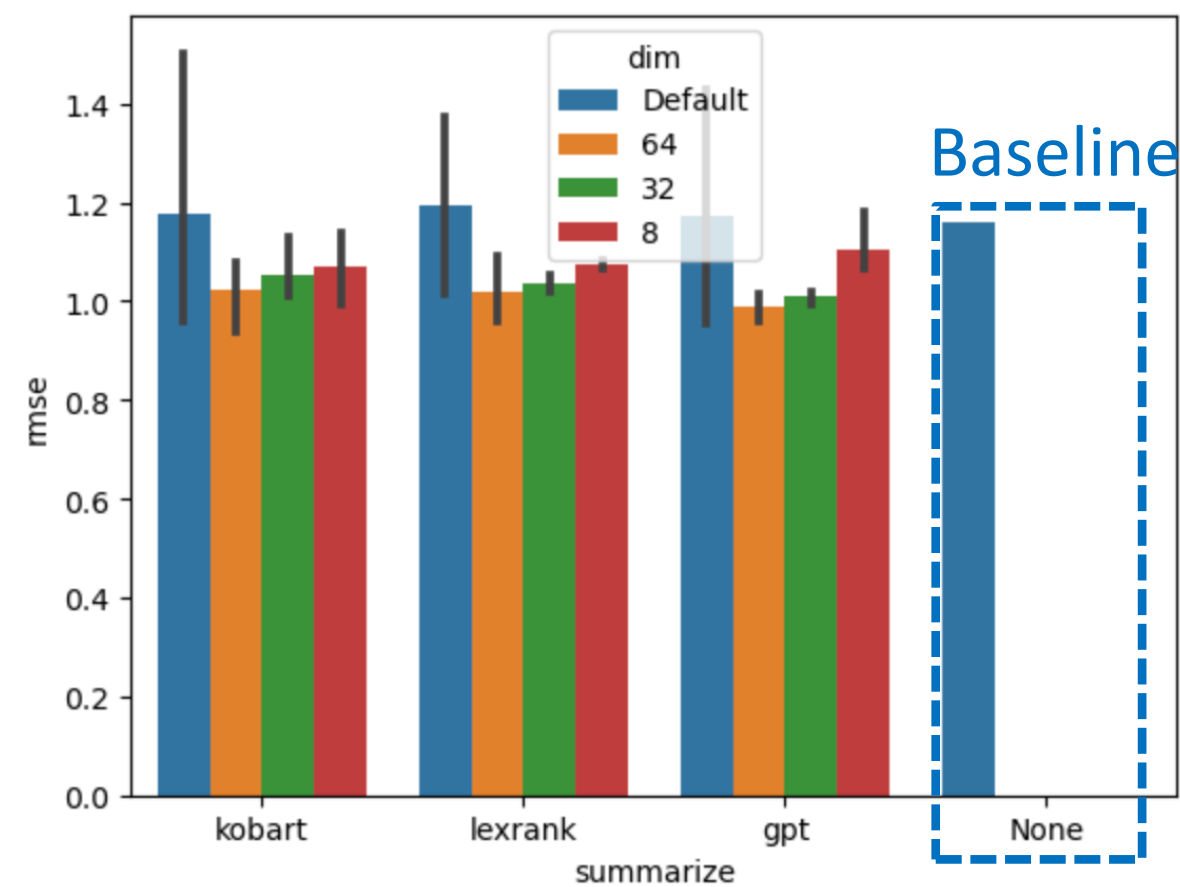
- Metric: AUC(higher is better)
- Baseline 과 비슷한 성능
- Review Embedding의 차원이 클수록 성능이 떨어짐
- Summarize에서는 chatGPT
- Embedding에서는 word2vec

실험 결과

GNN Task : Link Regression



- Metric : RMSE(lower is better)
- 대체적으로 Review embedding을 추가 했을 때 RMSE가 낮음



- Review Embedding의 차원은 32
- Summarize에서는 비슷한 성능
- Embedding에서는 word2vec과 koBART

실험 결과

GNN Task : Link Regression(Baseline)

1	result_df[result_df.target==5.0]				
✓	0.0s				
	userId	productId	rating	target	
1	152	1	3.633228	5.0	
2	1548	60	4.464217	5.0	
3	2606	225	4.844783	5.0	
4	399	40	4.218954	5.0	
5	168	1	5.000000	5.0	
...	
537	1956	77	5.000000	5.0	
538	3109	133	5.000000	5.0	
539	1773	385	4.571058	5.0	
540	162	1	3.633228	5.0	
541	1952	346	5.000000	5.0	

1	result_df[result_df.target==4.0]				
✓	0.0s				
	userId	productId	rating	target	
21	1369	293	3.874032	4.0	
48	3288	137	4.881904	4.0	
61	101	26	4.691068	4.0	
62	2586	103	5.000000	4.0	
64	1036	52	5.000000	4.0	
65	1741	67	3.496593	4.0	
114	875	377	3.737224	4.0	
129	701	41	5.000000	4.0	
133	3554	149	2.777737	4.0	
143	318	8	4.169987	4.0	
165	1790	70	3.822004	4.0	
180	3369	144	5.000000	4.0	
186	3228	167	4.183375	4.0	
198	2205	85	4.992976	4.0	
237	2094	81	3.286409	4.0	
255	2649	109	2.550093	4.0	
269	776	226	2.833604	4.0	
271	3343	143	3.121306	4.0	
297	1202	42	5.000000	4.0	
298	2500	100	3.939670	4.0	
306	1258	53	4.804395	4.0	
345	1693	66	5.000000	4.0	
347	703	23	2.516837	4.0	
372	2447	99	4.250609	4.0	
383	308	27	4.208152	4.0	
398	320	8	4.438456	4.0	
399	1887	74	5.000000	4.0	
405	761	26	3.286736	4.0	
406	3456	147	5.000000	4.0	
413	2936	126	2.828454	4.0	
421	629	155	4.153125	4.0	
429	3001	128	4.727071	4.0	
432	3397	145	4.943103	4.0	
446	2406	145	5.000000	4.0	
449	3240	137	4.693834	4.0	

1	result_df[result_df.target==3.0]				
✓	0.0s				
	userId	productId	rating	target	
0	3306	53	3.577972	3.0	
18	1660	66	4.691576	3.0	
26	2832	120	5.000000	3.0	
35	1152	440	5.000000	3.0	
50	478	13	4.875893	3.0	
53	749	421	4.682847	3.0	
56	1486	58	5.000000	3.0	
122	1375	52	3.955114	3.0	
127	3386	51	5.000000	3.0	
131	3229	33	5.000000	3.0	
170	2735	377	4.031892	3.0	
221	386	97	5.000000	3.0	
285	3055	107	3.316368	3.0	
325	2209	85	5.000000	3.0	
327	1128	40	5.000000	3.0	
331	1665	66	5.000000	3.0	
334	1693	103	5.000000	3.0	
341	2433	98	5.000000	3.0	
378	2890	2	5.000000	3.0	
436	1647	440	5.000000	3.0	
444	1168	58	5.000000	3.0	
454	617	16	4.380340	3.0	
533	811	27	5.000000	3.0	

1	result_df[result_df.target==2.0]				
✓	0.0s				
	userId	productId	rating	target	
120	3012	38	5.000000	2.0	
160	29	45	5.000000	2.0	
173	1705	67	5.000000	2.0	
218	3012	13	5.000000	2.0	
299	1492	128	5.000000	2.0	
329	1027	41	4.817318	2.0	
348	2974	34	5.000000	2.0	
354	3069	129	2.983785	2.0	

1	result_df[result_df.target==1.0]				
✓	0.0s				
	userId	productId	rating	target	
42	3341	143	5.000000	1.0	
228	3436	146	2.394918	1.0	
256	2482	100	2.916188	1.0	
309	29	2	5.000000	1.0	
335	649	41	5.000000	1.0	

- 대부분의 rating이 5점이며, 낮은 rating의 데이터가 많지 않음
- Baseline 모델은 낮은 rating의 데이터에 대해 잘 예측하지 못하는 편

실험 결과

GNN Task : Link Regression(kobart_kobart)

```
2 result_df[result_df.target==5.0]
```

✓ 0.1s

	userId	productId	rating	target
0	1754	97	4.150192	5.0
1	2450	99	2.999762	5.0
2	1478	57	3.665874	5.0
3	26	2	3.446088	5.0
4	2997	181	4.325217	5.0
5	1304	419	4.693791	5.0
6	3531	148	2.713874	5.0
7	2737	377	3.123627	5.0
8	3391	145	3.053062	5.0
9	1192	42	3.742921	5.0
10	2817	120	2.638542	5.0
11	2076	81	2.004297	5.0
12	892	20	3.423575	5.0
13	3466	260	5.000000	5.0
14	1897	74	5.000000	5.0
15	353	154	3.365428	5.0
16	2201	157	2.553615	5.0
17	2116	8	5.000000	5.0
18	588	15	4.528852	5.0
20	2855	122	3.328531	5.0
21	984	12	3.223139	5.0
23	3164	14	5.000000	5.0
24	3317	143	3.206159	5.0
25	653	22	3.144448	5.0
26	2313	13	4.258848	5.0
27	3153	133	5.000000	5.0
28	3000	128	3.058114	5.0
29	1105	87	5.000000	5.0

```
1 result_df[result_df.target==4.0]
```

✓ 0.1s

	userId	productId	rating	target
19	1670	66	2.515708	4.0
22	2692	112	3.674885	4.0
33	3205	55	3.708409	4.0
102	3554	149	1.954402	4.0
112	270	5	3.644935	4.0
116	1480	44	5.000000	4.0
127	3461	440	2.639439	4.0
145	991	19	5.000000	4.0
146	320	8	5.000000	4.0
148	2966	155	5.000000	4.0
166	2273	61	3.230604	4.0
169	3518	148	2.986852	4.0
178	2961	127	2.908696	4.0
184	226	428	1.997170	4.0
194	1381	55	3.635383	4.0
204	2815	329	2.414355	4.0
210	626	155	5.000000	4.0
234	285	71	4.847541	4.0
236	3404	145	3.213202	4.0
237	488	13	5.000000	4.0
243	781	223	2.534193	4.0
266	825	27	5.000000	4.0
269	1693	66	3.967039	4.0
271	1036	52	5.000000	4.0
282	107	0	3.149905	4.0
289	112	30	1.871232	4.0
290	856	40	2.514253	4.0
297	1063	39	4.764365	4.0
299	1285	267	3.414635	4.0
301	1070	39	3.584456	4.0
305	91	240	3.954740	4.0
307	1458	87	3.893723	4.0
316	1910	75	2.458069	4.0
320	2406	145	4.348644	4.0

```
1 result_df[result_df.target==3.0]
```

✓ 0.0s

	userId	productId	rating	target
39	2202	157	3.885992	3.0
132	1308	48	5.000000	3.0
259	1152	45	4.554935	3.0
372	3386	144	3.492504	3.0
381	395	181	2.862740	3.0
391	1410	55	4.877378	3.0
392	2433	98	3.239474	3.0
463	1709	67	4.663021	3.0
502	3320	143	2.129749	3.0
511	2719	247	3.322129	3.0
531	1129	41	4.674100	3.0

```
1 result_df[result_df.target==2.0]
```

✓ 0.0s

	userId	productId	rating	target
78	29	45	4.964735	2.0
119	2012	79	1.761225	2.0
142	1705	67	5.000000	2.0
224	884	342	4.176741	2.0
227	3338	143	3.223336	2.0
327	1285	51	2.410527	2.0
487	3221	49	3.521013	2.0
524	2050	38	2.742182	2.0

```
1 result_df[result_df.target==1.0]
```

✓ 0.0s

	userId	productId	rating	target
226	2961	216	2.221242	1.0
291	576	15	4.398277	1.0

- Review embedding을 추가했을 때, 낮은 rating의 데이터에 대해 Baseline 대비 상대적으로 잘 예측

프로젝트 의의 & 활용 방안

Conclusion

- GNN 기반 추천시스템에, 사용자의 Review 데이터를 추가하여 예측 성능을 비교
- Link Prediction에서는 성능향상이 없었으며, 기존 데이터만으로도 좋은 성능을 보임
- Link Regression에서는 Baseline 대비 낮은 RMSE를 보였으며, 상대적으로 낮은 rating을 잘 예측
- Summarize 방법은 비슷한 성능들을 보였으며, Embedding 방법은 word2vec이 준수한 성능을 보임

활용방안

- 일반적으로 사용자들은 높은 rating을 주기 때문에, 낮은 rating을 받은 상품에 대한 데이터가 적음
- 사용자의 Review 데이터를 통해 성향을 파악하고, 낮은 rating에 대한 예측성능을 향상시킴으로써 잘못된 추천의 비율을 낮출 수 있을 것으로 기대

참고 자료

- 설진석, 이상구.(2016).lexrankr: LexRank 기반 한국어 다중 문서 요약.한국정보과학회 학술발표논문집,(),458-460.
- <https://excelsior-cjh.tistory.com/194>
- https://heung-bae-lee.github.io/2020/01/30/NLP_04/

감사합니다

4조

이영현
강채원
김주은
임세은