

패션커머스 고객 **TPO***에 맞는 맞춤추천 시스템 개발

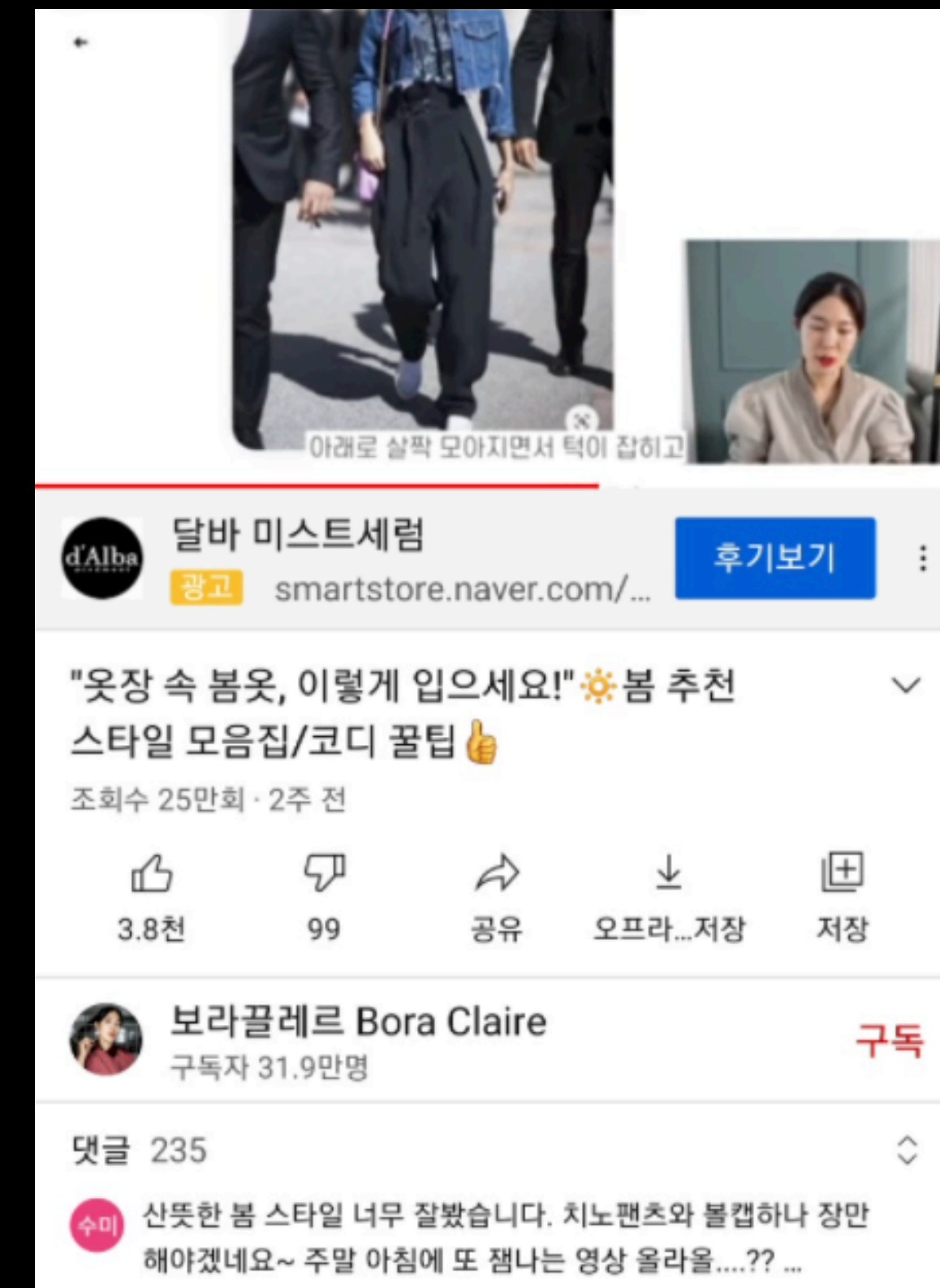
*TPO(Time, Place, Occasion의 약자)

목차

- 프로젝트 배경
- 데이터 소개
- 추천모델 주요 개념 소개
- 본 프로젝트 가설, 분석 모델, 검정 결과
- 결과 요약 및 향후 보완할 점

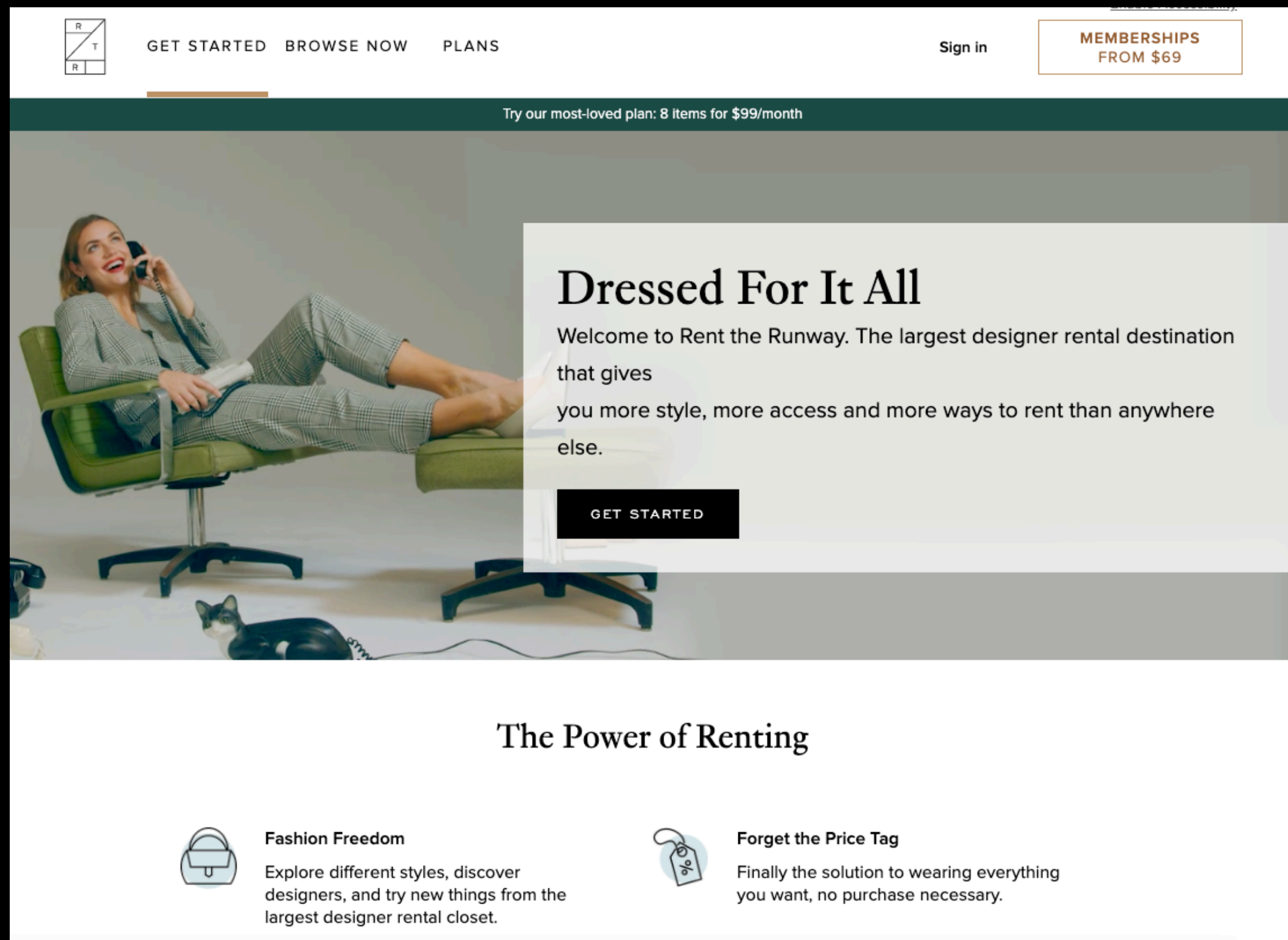
왜? 고객의 TPO를 잡아야 할까요?

- 잘 나가는 '패션 유튜버'도, 인스타그램어도 고객을 사로잡는 방법 'TPO별 룩(look)&코디 제안'!



고객 TPO별 다양한 룩을 제안하여,
추가 구매 전환 기회가 많은 곳

데이터 소개 - 'RentTheRunway' 데이터



< 데이터 형태 >

- Number of customers: 105,508
- Number of products: 5,850
- Number of transactions: 192,544

< 데이터 특성 >

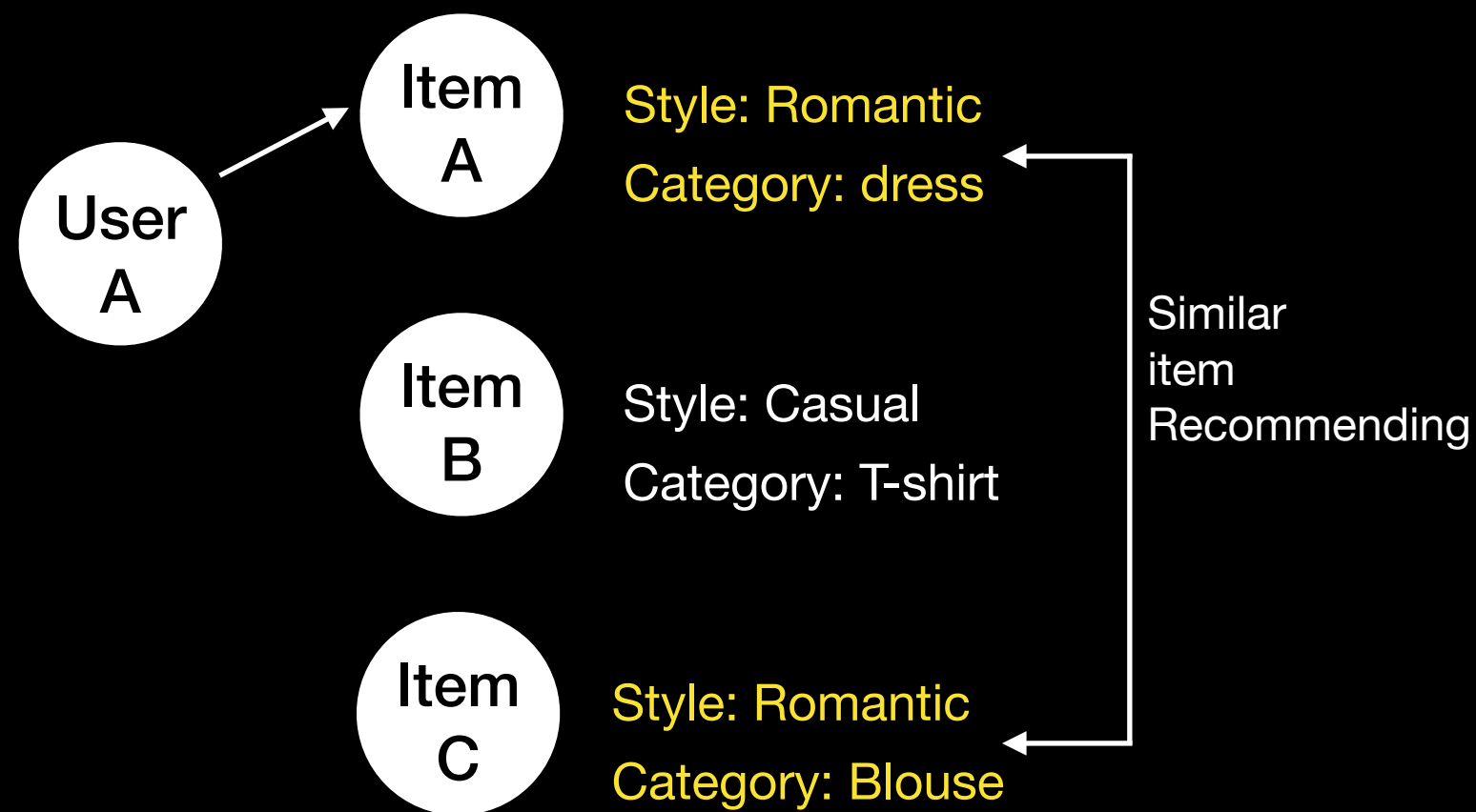
- **user_id**: a unique id for the customer
- **item_id**: unique product id
- **rating**: rating for the product
- **rating_average** : popularity를 측정하기 위한 것으로 아이템별 rating 평균값.
- **weight**: weight measurement of customer
- **rented for**: purpose clothing was rented for => TPO 데이터로 사용
- **body type**: body type of customer
- **review_text**: review given by the customer
- **review_summary**: summary of the review
- **size**: the standardized size of the product
- **age**: age of the customer
- **category**: the category of the product
- **bust size**: bust measurement of customer
- **height**: height of the customer
- **fit**: fit feedback
- **review_date**: date when the review was written

※출처: 캐글(Kaggle)에 공개된 'RentTheRunway' 여성 의류쇼핑몰 데이터셋

<https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation>

추천모델 주요 개념 소개 - 모델의 유형

1. 콘텐츠기반 필터링 (content based filtering)



- 장점: 모델이 단순하고 빠름.
- 단점: 식상한 콘텐츠 추천이 될 수 있음.

2. 협업필터링 (CF: Collaborative Filtering)

	Item A	Item B	Item C	Item D
User A	4		3	1
User B		2		
User C	3		5	
User D		3	4	4

Similar User

Similar Item

<item-to-item>

- 장점: 아이템이 매우 많은 경우, 유저베이스에 비하여 더 빠르고 안정적.
- 단점: 특정 아이템을 좋아한다고 해당 아이템과 연관된 모든 아이템을 좋아하지 않을 수 있고, 식상한 추천이 될 수 있음.

<user-to-user>

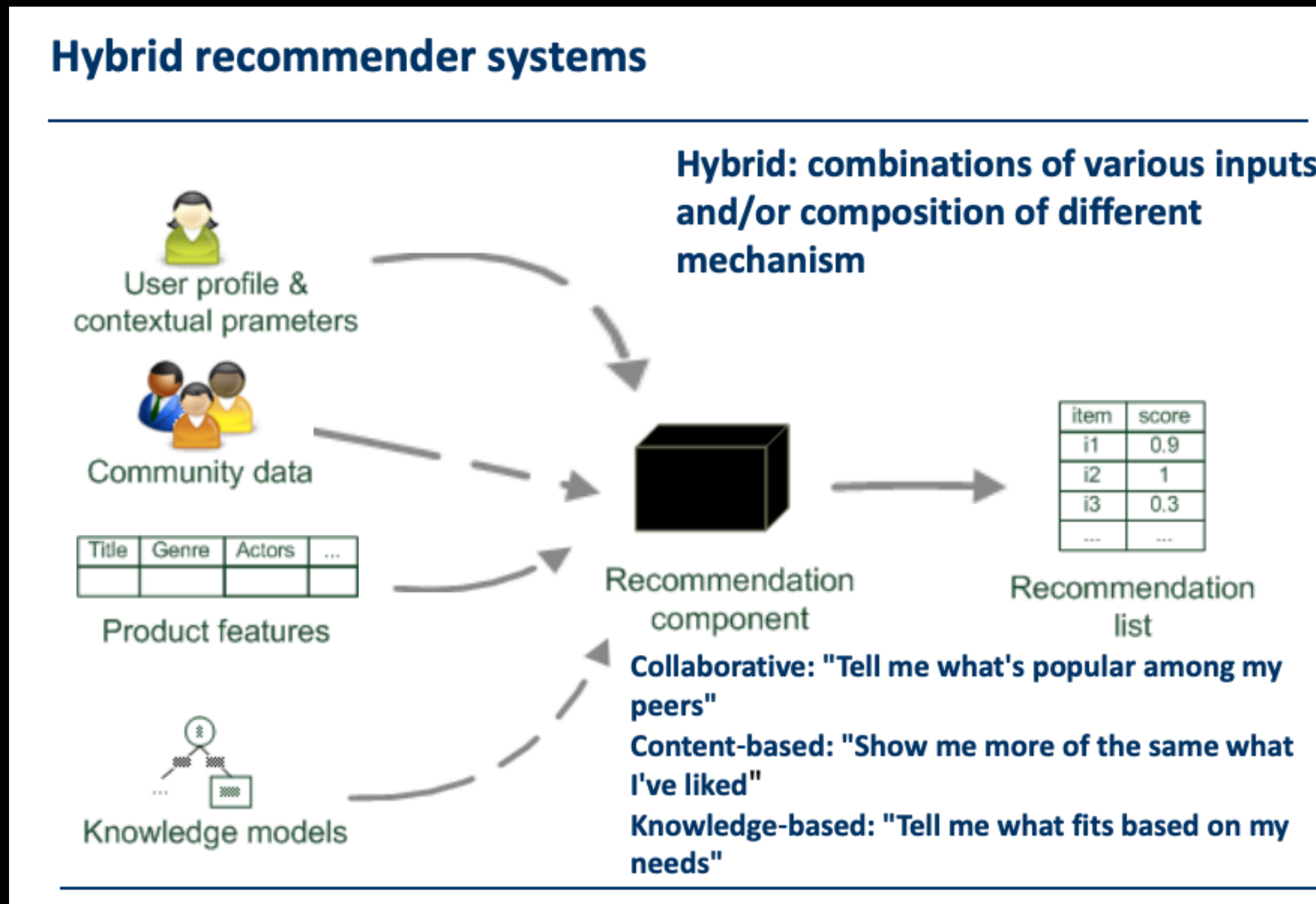
- 장점: 비슷한 유저 그룹이 소비한 다른 콘텐츠를 추천하기 때문에, 소비자 특성이 잘 분류되는 시장에서 만족도 높은 콘텐츠를 추천.
- 단점: 위 모델에서는 오직 평점을 기준으로 유사그룹을 측정한다는 점. 평점 이외 다른 요소를 고려하지 않음.

추천모델 주요 개념 소개 - 모델의 유형

3. 하이브리드 추천 모델

(Hybrid recommendation)

- 장점: 유저의 평점 데이터로만 추천하는 것이 아니라, 유저와 아이템의 다양한 다른 특성(ex. Age, style needs, item-color/design/fit/category, etc.) 을 고려하여 추천하기 때문에 ‘유사도가 있으면서, 신선도가 있는 추천이 가능’
- 단점: 모델 구현이 복잡하고, 해당 시장의 특성에 따라 적절한 솔루션을 찾아야 한다는 점.



추천모델 주요 개념 소개 - 평가지표

1. 모델의 성능지표

RMSE(Root Mean Square Error)

$$\text{RMSE} = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}$$

MAE(Mean Absolute Error)

$$\text{MAE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

<RMSE, MAE의 의미>

- 예측 평점과 실제 평점 사이의 오차(손실)
- 크기가 작을수록 모델 예측 성능이 좋음을 의미.
- RMSE는 오차에 대해서 MAE보다 더 큰 값으로 보여주기 때문에, 일반적인 머신러닝 모델에서 성능개선을 더 정확하게 해야 할 때 사용.

2. 추천목록의 평가지표

$$\text{Precision@k} = \frac{|\{\text{Recommended items that are relevant}\}|}{|\{\text{Recommended items}\}|}$$
$$\text{Recall@k} = \frac{|\{\text{Recommended items that are relevant}\}|}{|\{\text{Relevant items}\}|}$$

Precision(정밀도) :

- 추천된 K개의 아이템들 중 유사도가 높은 아이템들의 비중.
- 추천목록 내에서 유사도 높은 아이템들이 얼마큼 있는지 의미.

Recall(재현율):

- 유사도가 높은 아이템들 중 추천된 K개의 아이템들의 비중
- 추천목록 전체가 유사도 높은 아이템들 중에서 얼마큼의 비중으로 추천 되었는지에 대한 수치.

본 프로젝트 가설, 분석 방법 소개

- (가설1) **아이템 기반 협업필터링 추천모델보다, SVD 협업필터링 모델의 성능이 더 높을 것이고, 가장 성능이 좋은 모델로 추천 목록을 고른다.**

=> 분석방법: 아이템 기반 협업필터링 추천모델과 SVD 협업필터링 모델의 RMSE, MAE 수치 비교.

- (가설2) 협업필터링의 추천모델 목록보다, **하이브리드 추천모델(고객의 TPO, Category, popularity 등을 추가한)의 추천목록의 순위평가지표가 더 높을 것이다.**

=> 분석방법: 아이템 기반 협업필터링 모델과 SVD 협업필터링 모델의 RMSE, MAE 수치 비교.

가설(1) 검증결과

■ (가설1) **아이템 기반 협업필터링 추천모델보다, SVD 협업필터링 모델의 성능이 더 높을 것이다.**

=> 분석방법: 아이템 기반 협업필터링 모델과 SVD 협업필터링 모델의 RMSE, MAE 수치 비교.

	Algorithm	test_rmse	test_mae
0	KnnBaseline(item-based)	0.718082	0.564715
1	KnnBaseline(user-based)	0.716891	0.563900
2	SVD(CF)	0.718168	0.560152
3	SVDpp(CF)	0.716266	0.556810
4	gridsearch(SVDpp)	0.715593	0.564648
5	Hybrid(CF+CBF)	0.717355	0.561553

- 사용한 모듈: Surprise (python library)
- KnnBaseline(item-based)모델: 기본 협업필터링 알고리즘 +Baseline을 추가하여 사용자별 평점 편향을 줄인 모델
- SVD모델: 특이값 분해(SVD)방식으로 계산한 협업필터링 모델
- SVDpp모델: SVD모델에 암시적 평점을 고려한 모델.
- Hybrid모델: SVDpp모델과 KnnBaseline을 4:6비중으로 가중 평균합하여 rmse와 mae를 계산, 고객TPO특성이 합쳐진 모델.
- 5가지 알고리즘 중 근소한 차이로 SVDpp 모델 성능이 가장 우수. 다만, 차이가 너무 근소하여 SVDpp모델의 성능 증명이 명확하다고 보기 어려워보임. 데이터의 Sparsity가 높을 경우 SVD, SVDpp의 성능이 기대보다 높게 나오지 않는다는 한계점을 배움.

가설(2) 검증결과

- (가설2) 협업필터링의 추천모델 목록보다, 하이브리드 추천모델(고객의 TPO,Category,popularity 등을 추가한)의 추천목록의 순위평가지표가 더 높을 것이다.

	Algorithm	test_rmse	test_mae	Precision	recall
0	KnnBaseline(item-based)	0.718082	0.564715	0.931839	0.943002
1	KnnBaseline(user-based)	0.716891	0.563900	0.932373	0.943399
2	SVD(CF)	0.718168	0.560152	0.931827	0.942827
3	SVDpp(CF)	0.716266	0.556810	0.930189	0.940228
4	gridsearch(SVDpp)	0.715593	0.564648	0.932217	0.943309
5	Hybrid(CF+CBF)	0.717355	0.561553	0.931179	0.941893

- Recall: 수치가 높을수록 추천목록 전체가 유사도 높은 아이템들 중에서 많이 선택 되었다는 의미.
- 근소한 차이로 KnnBaseline(user-based)모델과 gridsearchCV로 조정한 SVDpp모델이 Recall값이 가장 높으나, 수치 차이가 매우 근소하다는 한계점.

Hybrid 추천목록의 장점

Hybrid 추천 목록 결과 (testset user = 817402)

	item_id	est	Model	category	TPO
2	543189	4.772779	Popularity(rating_average)	[gown, gown, gown, gown, gown, gown, gown, gown, gown, gown]	[wedding, wedding, formal affair, formal affair, formal affair]
3	633179	4.752781	Popularity(rating_average)	[maxi, maxi, maxi, maxi, maxi, maxi, maxi, maxi, maxi, maxi]	[wedding, party, wedding, party, formal affair, formal affair, formal affair, formal affair]
1	2337876	4.731897	Popularity(rating_average)	[skirt, skirt, skirt, skirt, skirt]	[other, party, party, other, party]
6	2276741	4.728079	Popularity(rating_average)	[jumpsuit, jumpsuit, jumpsuit]	[other, party, other]
4	675264	4.715891	Popularity(rating_average)	[gown, gown, gown, gown, gown]	[other, wedding, formal affair, wedding, formal affair]
9	708493	4.680326	Popularity(rating_average)	[dress, dress, dress]	[other, everyday, work]
0	330893	4.661841	Popularity(rating_average)	[dress, dress]	[party, other]
5	1011971	4.640780	Popularity(rating_average)	[dress, dress]	[other, formal affair]
7	328677	4.631116	Popularity(rating_average)	[dress, dress, dress]	[other, party, party]
8	1285191	4.589043	Popularity(rating_average)	[dress]	[other]

기존 추천목록에서는 ‘아이템’ 추천목록만 확인 가능하다면,

하이브리드 모델의 결과는 아이템별 카테고리나 TPO를 확인할 수 있어 더 다양한 추천으로 이어질 수 있다는 장점.

추천알고리즘에서 ‘신선도’, ‘다양성’ 등이 중요한 이슈인만큼 모델의 성능차이가 근소하다면 하이브리드 모델을 사용하는 것이 다양성과 개인화 니즈를 만족시킬 수 있는 모델에 가까울 것으로 기대.

결론 요약 및 향후 보완하면 좋을 점

[결론 요약]

1. 가설1 검정결과 : 아이템기반 모델(Knn-baseline_item-based)보다 SVDpp모델의 RMSE수치가 근소하게 낮음(성능이 좋음)

■ KnnBaseline(item-based) RMSE: 0.718082 > SVDpp RMSE: 0.716266

==> 차이가 너무 근소하여 가설검정에 유의미하다고 보기 어려울 수 있다는 한계점.

2. 가설2 검정결과: KnnBaseline(user-based)모델과 gridsearchCV로 조정한 SVDpp모델이 Recall값이 가장 높음.

■ KnnBaseline(item-based) recall: 0.943002 < SVDpp(gridsearchCV) recall: 0.943309

==> 차이가 너무 근소하여 가설검정에 유의미하다고 보기 어려울 수 있다는 한계점.

[향후 보완하면 좋을 점]

1. 유저의 구매고려사항에 대한 데이터 확보(해당 아이템을 살 때 고려하는 점의 우선순위 등)
2. 추천 시스템에 대한 유저의 취향 파악 (신선도 vs 유사도 비중을 어느 정도로 좋아하는지 등)
3. 실제 유저가 반응한 유저행동데이터(구매/클릭/체험한 시간 등) 를 고려한 추천

참고자료

<추천모델의 개념>

“Build a Recommendation Engine With Collaborative Filtering”

<https://realpython.com/build-recommendation-engine-collaborative-filtering/>

“Hybrid recommendation approach”

https://www.math.uci.edu/icamp/courses/math77b/lecture_12w/pdfs/Chapter%2005%20-%20Hybrid%20recommendation%20approaches.pdf

"Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System"

<https://bit.ly/3tenCml>

<코드구현 참고자료>

Python Surprise 라이브러리

<https://surprise.readthedocs.io/en/stable/index.html>

hybrid_model_CF and Latent factor models

https://github.com/prakruti-joshi/Movie-Recommendation-System/blob/master/Code/hybrid_model.ipynb