패션커머스 고객 TPO*에 맞는 맞춤추천 시스템 개발

*TPO(Time, Place, Occasion의 약자)

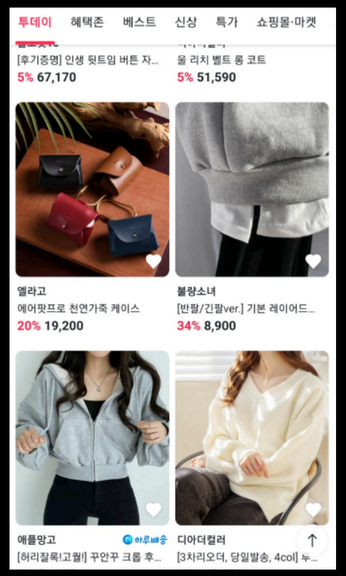
김현재 21.3.16

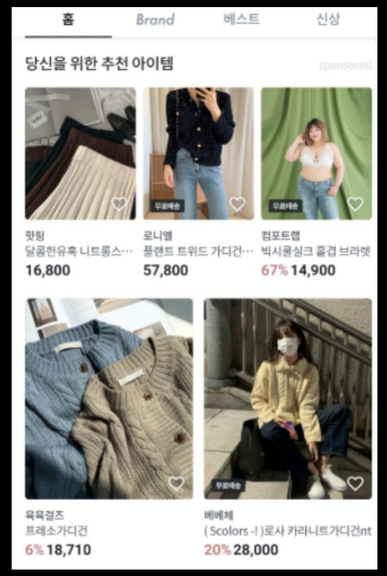
목차

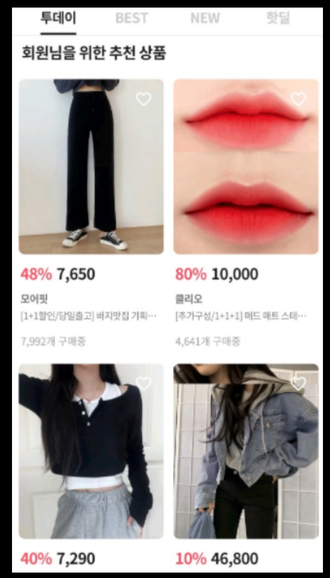
- 프로젝트 배경
- 데이터 소개
- 추천모델 주요 개념 소개
- 본 프로젝트 가설, 분석 모델, 결과 소개
- 요약 및 향후 보완할 점

왜? 고객의 TPO를 잡아야 할까요?

- 패션 커머스 앱 대부분 ‘아이템’기반 추천을 하고 있는 상황







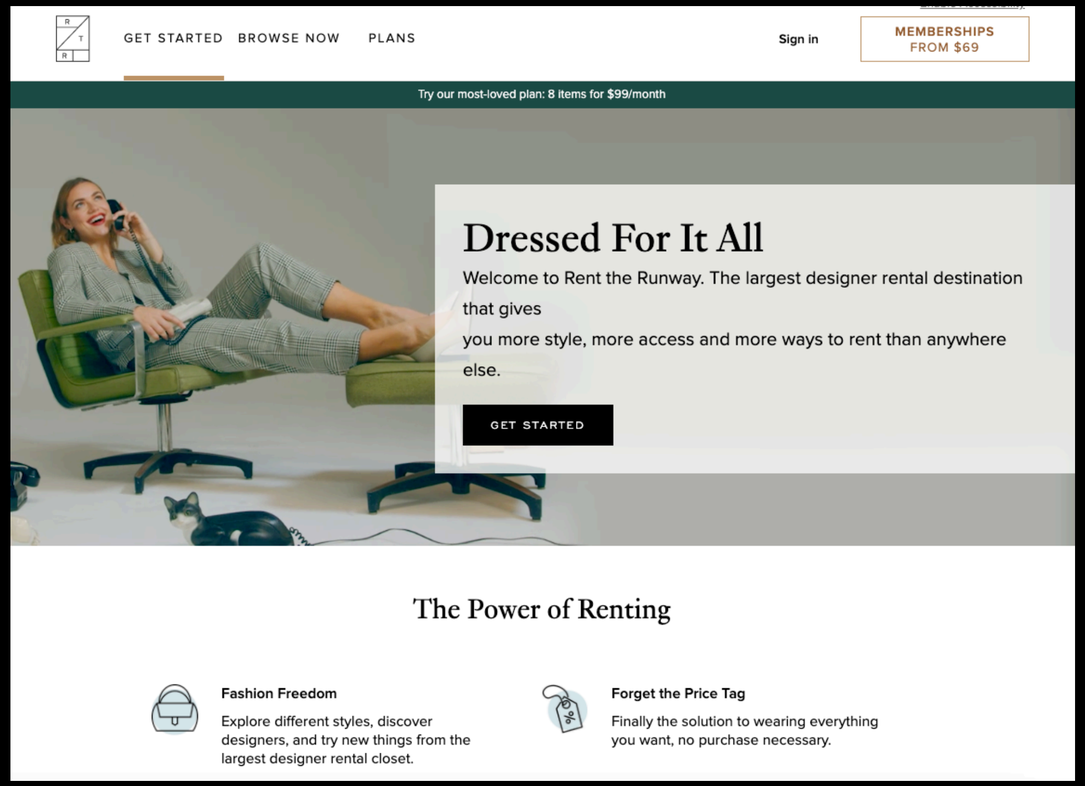
=> 놀랍게도.. 모두 다른 패션앱!

(달라보이시나요?)

- 잘 나가는 ‘유튜버’도, 인스타그래머도 고객을 사로잡는 방법 ‘TPO별 룩(look)&코디 제안’!

차별화, 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 >

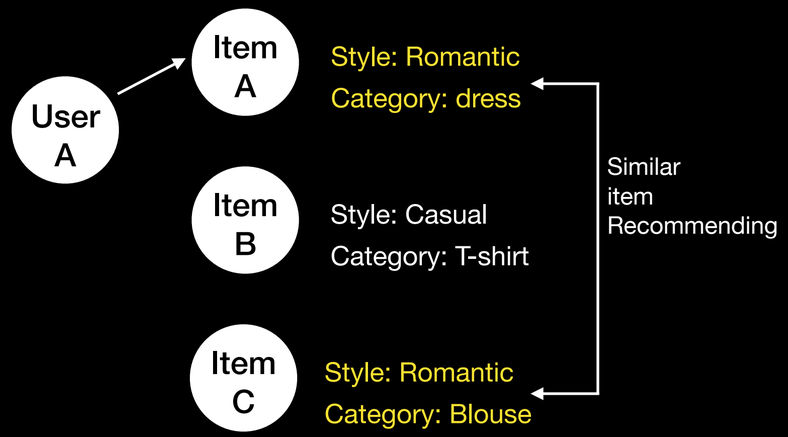
- user_id: a unique id for the customer
- item_id: unique product id
- rating: rating for the product
- rating_average : popularity를 측정하기 위한 것으로 아이템별 rating 평균값.
- weight: weight measurement of customer
- rented for: purpose clothing was rented for => TPO 데이터로 사용
- body type: body type of customer
- 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

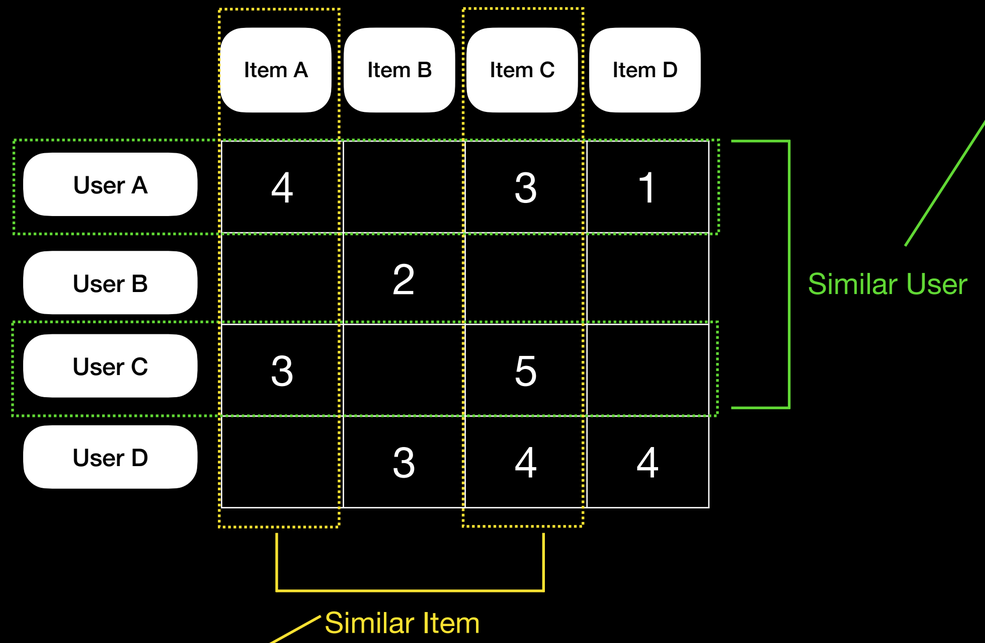
(content based filtering)



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

2. 협업필터링

(CF:Collaborative Filtering)



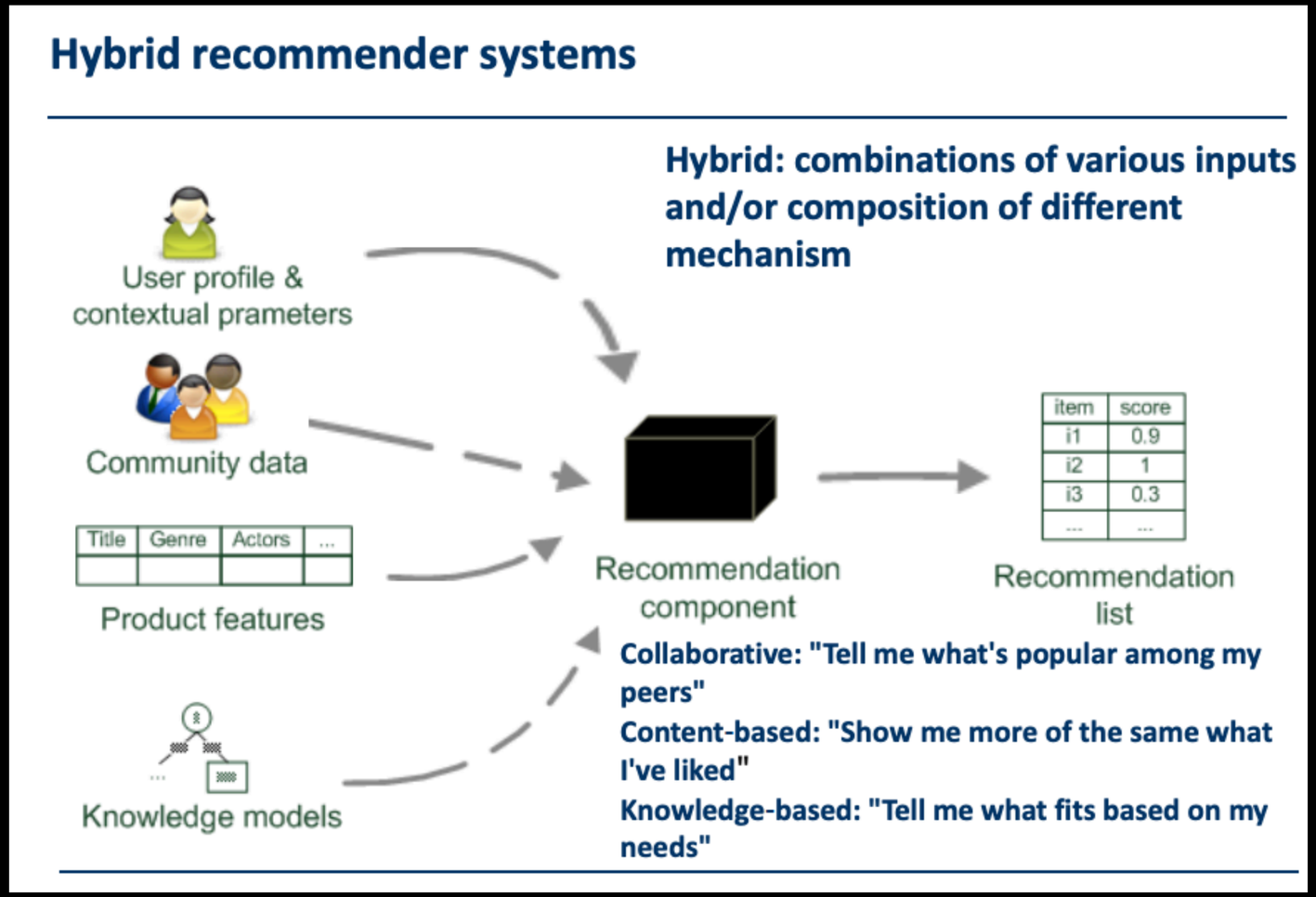
<item-to-item>

- 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 컨텐츠를 추천하기 때문에, 소비자 특성이 잘 분류되는 시장에서 만족도 높은 컨텐츠를 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.

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

추천모델 주요 개념 소개 - 평가지표

1. 모델의 성능지표

RMSE(Root Mean Square Error)

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

MAE(Mean Absolute Error)

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

<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

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

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

Precision(정밀도) :

추천된 K개의 아이템 중 유사도가 높은 아이템들의 비중.

(전체 추천목록 내에 유사도가 있는 추천 아이템들이 얼마나 되는지 측정,

추천목록 내 유사도 비중을 높일 때 정밀도를 높이면 됨.)

Recall(재현율):

유사도가 높은 아이템들 중 추천된 K개의 아이템들의 유사도 비중

(전체 추천목록 자체가 얼마나 유사도가 높은지 측정,

추천할 때마다 추천목록의 전체 유사도를 유지하고 싶을 때 사용.)

본 프로젝트 가설, 분석 방법 소개

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

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

■ (가설2) 협업필터링의 추천모델 목록보다, 하이브리드 추천모델(고객의 TPO,Category,popularity 등을 추가한)의 추천목록의 만족도(실제 평점, 구매여부)가 더 높을 것이다.

=> 협업필터링 추천목록의 실제 평점과 하이브리드 추천목록의 실제 평점 (평균 등)을 비교. (보조지표: recall, precision)

가설(1) 검증결과

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

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

	Algorithm	test_rmse	test_mae
0	KnnBaseline(item-based)	0.709493	0.564006
1	KnnBaseline(user-based)	0.709543	0.564261
2	SVD(CF)	0.708741	0.560561
3	SVDpp(CF)	0.707846	0.556635
4	gridsearch(SVDpp)	0.706181	0.563741

- 사용한 모듈: Surprise (python library)
- Knn알고리즘: 훈련셋의 벡터와 '근접한 거리에 있는' 벡터를 예측값으로 출력.
- SVD알고리즘: 유저와 아이템의 상관관계(평점)가 있는 행렬매트릭스에서 유사도를 측정하여 예측하는 모델. (Matrix factorization 방식)
- 5가지 알고리즘 중 **'협업필터링 알고리즘 (SVDpp, gridsearchCV로 찾은 SVDpp)' 모델 성능이 가장 우수.**

가설(2) 검증결과

■ (가설2) 협업필터링의 추천모델 목록보다, 하이브리드 추천모델(고객의 TPO,Category,popularity 등을 추가한)의 추천목록의 만족도(실제 평점, 구매여부)가 더 높을 것이다.

=> 협업필터링 추천목록의 실제 평점과 하이브리드 추천목록의 실제 평점 (평균 등)을 비교. (보조지표: recall, precision)

	Algorithm	test_rmse	test_mae	Precision	recall	F-measure	nDCG_score
0	KnnBaseline(item-based)	0.704433	0.561753	0.933642	0.946020	0.939790	1.000000
1	KnnBaseline(user-based)	0.703550	0.561565	0.933671	0.945977	0.939783	1.000163
2	SVD(CF)	0.704298	0.557937	0.933316	0.945001	0.939122	1.000363
3	SVDpp(CF)	0.704978	0.555545	0.930340	0.940613	0.935448	0.999756
4	gridsearch(SVDpp)	0.702520	0.562151	0.933909	0.946093	0.939961	0.999253
5	Hybrid(CF+CBF)	0.708670	0.548347	0.931045	0.941751	0.936368	0.999966

사용자의 실제 평점 데이터는 아쉽게도 확인이 불가하여,
보조지표로 recall 과 precision 값을 비교. 하이브리드의 모델 추천목록도 아이템 유사도에 크게 떨어지지 않음.

가설(2) 검증결과

Hybrid 모델의 추천 목록
(testset user = 222811에 대한 추천 목록)

	item_id	est	Model	category	TPO
18	948396	4.846413	Popularity(rating_average)	[gown, gown, gown, gown, gown, gown, gown, gow...	[wedding, formal affair, formal affair, party,...
5	1202983	4.767246	Item similarity	[dress, dress, dress, dress, dress, dress, dress]	[work, work, work, work, work, work, party]
12	1529884	4.742492	Popularity(rating_average)	[sheath, sheath]	[party, party]
16	1883531	4.739021	Popularity(rating_average)	[sheath, sheath, sheath, sheath, sheath, sheat...	[wedding, other, wedding, date, wedding, party...
19	1692989	4.715691	Popularity(rating_average)	[gown, gown, gown, gown, gown, gown]	[party, formal affair, formal affair, wedding,...
4	614587	4.714849	Item similarity	[dress, dress]	[work, everyday]
11	1935592	4.713615	Popularity(rating_average)	[dress, dress, dress]	[party, party, work]
17	1881176	4.706546	Popularity(rating_average)	[gown, gown]	[party, formal affair]
0	1527946	4.693832	CBF + CF	[sheath, sheath, sheath, sheath, sheath]	[party, party, party, wedding, date]
8	2390516	4.683958	Item similarity	[top, top, top, top, top]	[work, everyday, work, other, other]

기존 추천목록이 '아이템'만 추천하는 반면,
카테고리와 TPO도 함께 추천할 수 있다는 장점!

요약 및 향후 보완할 점

<개인화된 맞춤 시스템으로 지속적으로 진화하기 위해 필요한 점>

1. 유저의 구매고려사항에 대한 데이터 확보 (해당 아이템을 살 때 고려하는 사항들의 우선순위)

2. 추천 시스템에 대한 유저의 취향 파악 (신선도:유사도 비중이 어느 정도일 때 만족하는지)

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

<코드구현 참고자료>

Surprise 모듈
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