

Reciprocal Recommendation System(RRSs)

p e o p l e t o p e o p l e 상호 추천 시스템 구현

CONTENTS

- 01** 프로젝트 기획 배경 & 목표
- 02** 프로젝트 내용(프로젝트 구성)
- 03** 프로젝트 발전 가능성
- 04** 프로젝트 후기

01

프로젝트 기획 배경 & 목표

- a. 프로젝트 기획 배경
- b. 프로젝트 목표

01 프로젝트 기획 배경 & 목표

a. 프로젝트 기획 배경

b. 프로젝트 목표

사람 <-> 사람 추천 알고리즘의 한계점 해결

현재 매칭 플랫폼에서 유저에게 유저를 추천해주는
협업필터링 기반 People to People 추천 알고리즘을 서비스화 하여 제공 중
그러다, **실제 매칭서비스 내에서 해당 추천 방식의 한계점을 발견함**

사람에게 아이템(item)을 추천하는 것은 추천 받는 유저의 '선호'만 고려하면 되지만,
매칭 플랫폼에서의 추천의 성공은 **추천 대상과의 '매칭' 이 더 좌우함**
=> **추천을 받는 유저(A) 의 선호 뿐 아니라, 추천되는 유저(B) 의 선호에도 맞아야함**

전통적인 people to people 추천은

- (1) 추천(item/user)를 받아들이는 유저 A에 의해 결정
- (2) 추천 받는 유저와 추천되는 유저의 선호에 동일한 가중치를 주는
'semi-personalized(반개인화)' 추천에 가까움

01 프로젝트 기획 배경 & 목표

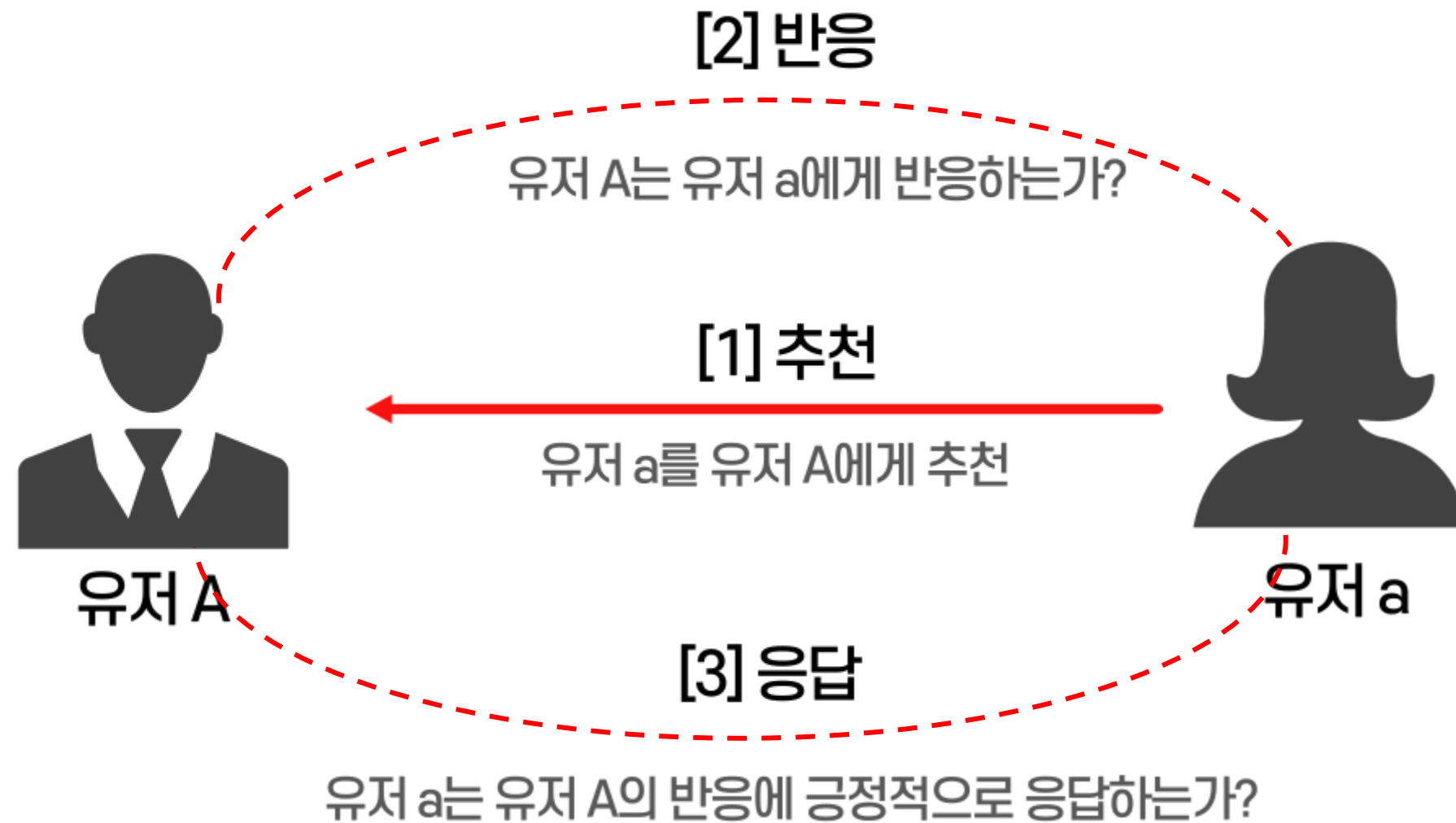
a. 프로젝트 기획 배경

b. 프로젝트 목표

상호 추천 알고리즘의 발견(RRSs)

추천 받는 유저(A), 추천 되는 유저(a)

두 유저 간 'successful interaction' (성공적인 상호 작용) 을 기반으로 한 추천 알고리즘



01 프로젝트 기획 배경 & 목표

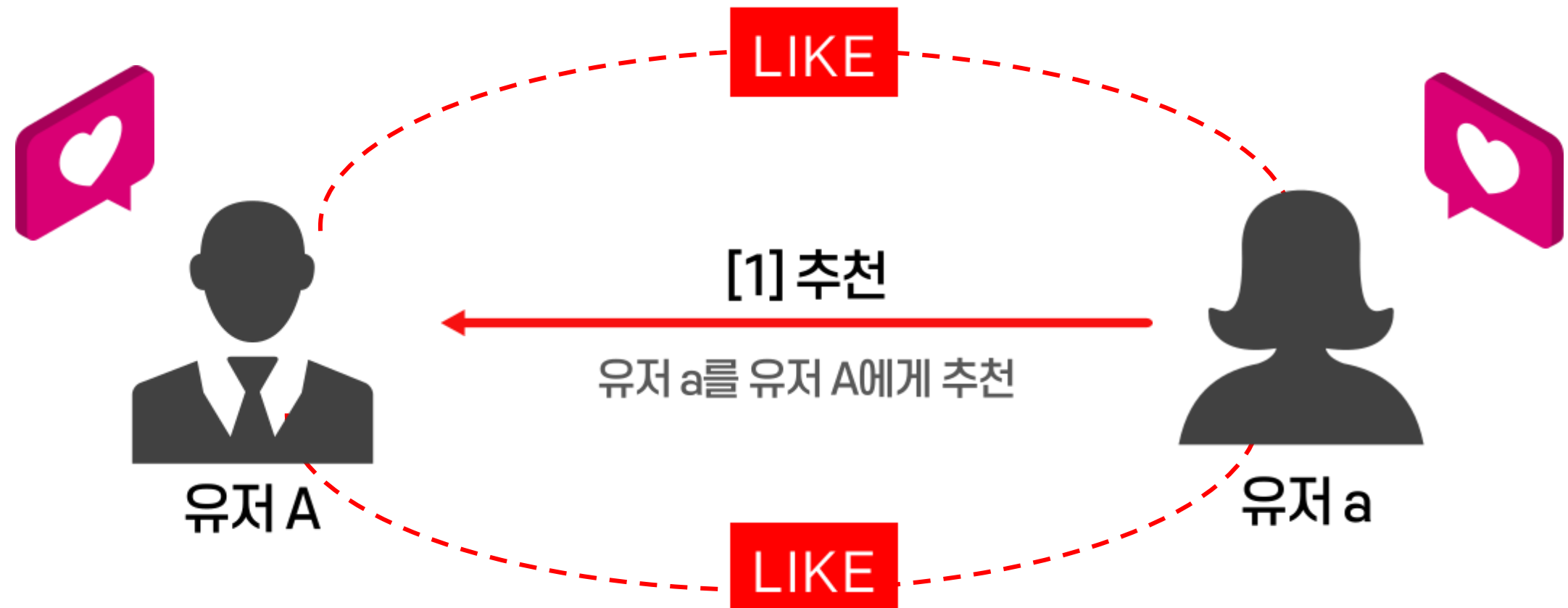
a. 프로젝트 기획 배경

b. 프로젝트 목표

상호 추천 알고리즘의 목표(Goals)

[상호 추천 알고리즘의 2가지 목표]

- [1] 유저 A(추천을 받는 유저)의 선호에 맞는 상대 유저 a 추천
- [2] 추천된 상대 유저 a 또한 유저 A가 선호에 맞아야 함



그러나 위의 CASE와 같이 People-to-Peple 사이의
긍정적 상호작용은 항상 일치하지 않음!

01 프로젝트 기획 배경 & 목표

a. 프로젝트 기획 배경

b. 프로젝트 목표

상호 추천 알고리즘의 성공적인 구현이 목표!

유저에 따라 추천된 유저 선호의 고려 정도의 차이가 달라,
완전한 개인화 추천 시스템과 더불어 상호 유저의 선호를 반영한
'상호 추천 시스템'의 구현을 목표로 프로젝트 진행

2021년 The Journal of Personalization Research 의
User Modeling and User-Adapted Interaction 에 게재된
Akiva Kleinerman (2021) 의

Supporting users in finding successful matches in reciprocal recommender systems
논문을 기반으로 해당 추천 시스템을 구현해볼

https://s3.us-west-2.amazonaws.com/secure.notion-static.com/96588717-b791-4ede-9d26-360e4f781753/Kleinerman2021_Article_SupportingUsersInFindingSuccess.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Credential=AKIAT73L2G45EIPT3X45%2F20221104%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20221104T103639Z&X-Amz-Expires=86400&X-Amz-Signature=39f76c4a038c4272fad51050239644e6337cde721170a22aa0e4f25db9d15678&X-Amz-SignedHeaders=host&response-content-disposition=filename%3D%22Kleinerman2021_Article_SupportingUsersInFindingSuccess.pdf%22&x-id=GetObject

02

프로젝트 내용

- a. 활용한 데이터
- b. 데이터 EDA
- c. 데이터 전처리
- d. 알고리즘 구현

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

활용한 데이터

(1) 유저 profile 데이터

	회원번호	성별	나이	지역	신장	체중	혈액형	연봉	재산	흡연유무	...	학력	종교	스타일	성격	이상형	사진개수	정보업데이트수	로그인수	서비스이용시간	선호하는상대지역
0	1	f	27	b	176	72	1	12	b	n	...	2	5	1	4	5	1	2	261	43055	b,
1	2	m	28	i	171	75	3	4	d	b	...	8	0	8	11	7	8	8	33	14854	i
2	3	m	24	b	176	80	3	10	c	c	...	6	5	1	4	4	2	31	2221	208975	b,

-> 회원 번호 포함 회원 특성이 담긴 22개의 컬럼으로 구성

(성별, 나이, 지역, 신장, 체중, 혈액형, 연봉, 재산, 흡연유무, 음주여부, 운동여부, 학력, 종교, 스타일, 성격, 이상형, 사진개수, 정보업데이트수, 로그인수, 서비스이용시간, 선호하는상대지역 등)

(2) 유저-유저 간 메시지 송-수신 데이터

-> send_user(메시지 보낸 유저), sent_user(메시지 받은 유저)

	send_user	sent_user
0	13876	11786
1	20615	6689

(3) 유저-유저간 프로필 열람 데이터

-> view_user(프로필을 열람한 유저), veiwed_user(프로필을 열람 당한 유저)

	view_user	viewed_user
0	24301	29736
1	7584	17553

02 프로젝트 내용

데이터 EDA

pandas_profiling의 ProfileReport 라이브러리 활용하여 EDA

-> 파일 업로드 용량이 10MB로 한정되어 있어, minimal = True 값으로 주어 기본적인 EDA값 확인
업로드 하지 않았지만 minimal = False 값으로 Correlation 및 missing values 확인 함

a. 활용한 데이터

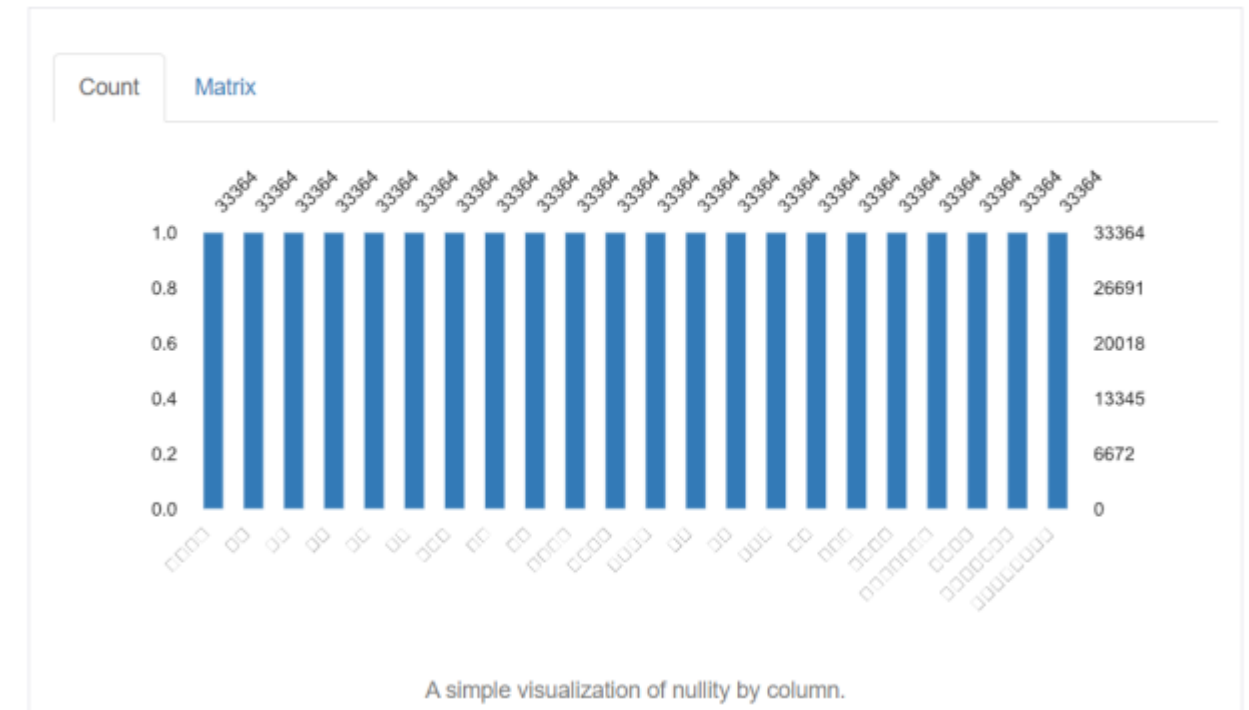
b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현



Missing values



02 프로젝트 내용

데이터 전처리

step 1. 유저 profile 데이터를 변수 특성에 맞도록 데이터 유형 변환

-> 첨부된 주피터 파일 10번 줄에서 데이터프레임 전처리 preprocessing 클래스로 만들어 변환

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

NO	컬럼 명	데이터 성격	전처리 방식	NO	컬럼 명	데이터 성격	전처리 방식
1	성별	범주형(명목형)		10	음주여부	범주형(명목형)	여러 attribute를 가지고 있어 유/무 (1/0) 수치형으로 전처리
2	나이	수치형(연속형)		11	운동여부	범주형(명목형)	여러 attribute를 가지고 있어 유/무 (1/0) 수치형으로 전처리
3	지역	범주형(명목형)		12	학력	범주형(순서형)	할당된 변수의 수치가 높을 수록 학력이 높음, int형으로 데이터형식 바꾸는 전처리
4	신장	수치형(연속형)		13	종교	범주형(순서형)	현재 데이터가 int형이라 범주형으로 전처리
5	체중	수치형(연속형)		14	스타일	범주형(명목형)	현재 int형으로 되어 있어, 범주형으로 전처리
6	혈액형	범주형(명목형)	현재 데이터가 int형이라 범주형으로 전처리 예정	15	성격	범주형(명목형)	현재 int형으로 되어 있어, 범주형으로 전처리
7	연봉	범주형(순서형)	할당된 변수의 수치가 높을 수록 연봉이 높음	16	이상형	범주형(명목형)	현재 int형으로 되어 있어 범주형으로 전처리
8	재산	범주형(순서형)	할당된 변수의 수치가 높을 수록 재산이 높은 순서형 데이터, int형으로 데이터 형식 바꾸는 전처리	17,18,19,20	수치형 (이산형)		
9	흡연유무	범주형(명목형)	여러 attribute를 가지고 있어 유/무(1/0) 수치형으로 전처리 예정	21	선호하는 상대지역	범주형(명목형)	데이터에 ' ' 가 섞여있어 텍스트 전처리 예정

step 2. 유저 간 메시지, 프로필 열람 데이터 전처리

02 프로젝트 내용

데이터 전처리

[데이터 전처리 전] sample 3개

	회원번호	성별	나이	지역	신장	체중	혈액형	연봉	재산	흡연유무	...	학력	종교	스타일	성격	이상형	사진개수	정보업데이트수	로그인수	서비스이용시간	선호하는상대지역
0	1	f	27	b	176	72	1	12	b	n	...	2	5	1	4	5	1	2	261	43055	b
1	2	m	28	i	171	75	3	4	d	b	...	8	0	8	11	7	8	8	33	14854	i
2	3	m	24	b	176	80	3	10	c	c	...	6	5	1	4	4	2	31	2221	208975	b

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

[데이터 전처리 후] sample 3개

	회원번호	성별	나이	지역	신장	체중	혈액형	연봉	재산	흡연유무	...	학력	종교	스타일	성격	이상형	사진개수	정보업데이트수	로그인수	서비스이용시간	선호하는상대지역
0	1	f	27	b	176	72	a	12	0	0	...	2	d	s1	c4	m5	1	2	261	43055	b
1	2	m	28	i	171	75	c	4	2	0	...	8	etc	s8	c11	m7	8	8	33	14854	i
2	3	m	24	b	176	80	c	10	1	1	...	6	d	s1	c4	m4	2	31	2221	208975	b

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현 - 알고리즘 도식화

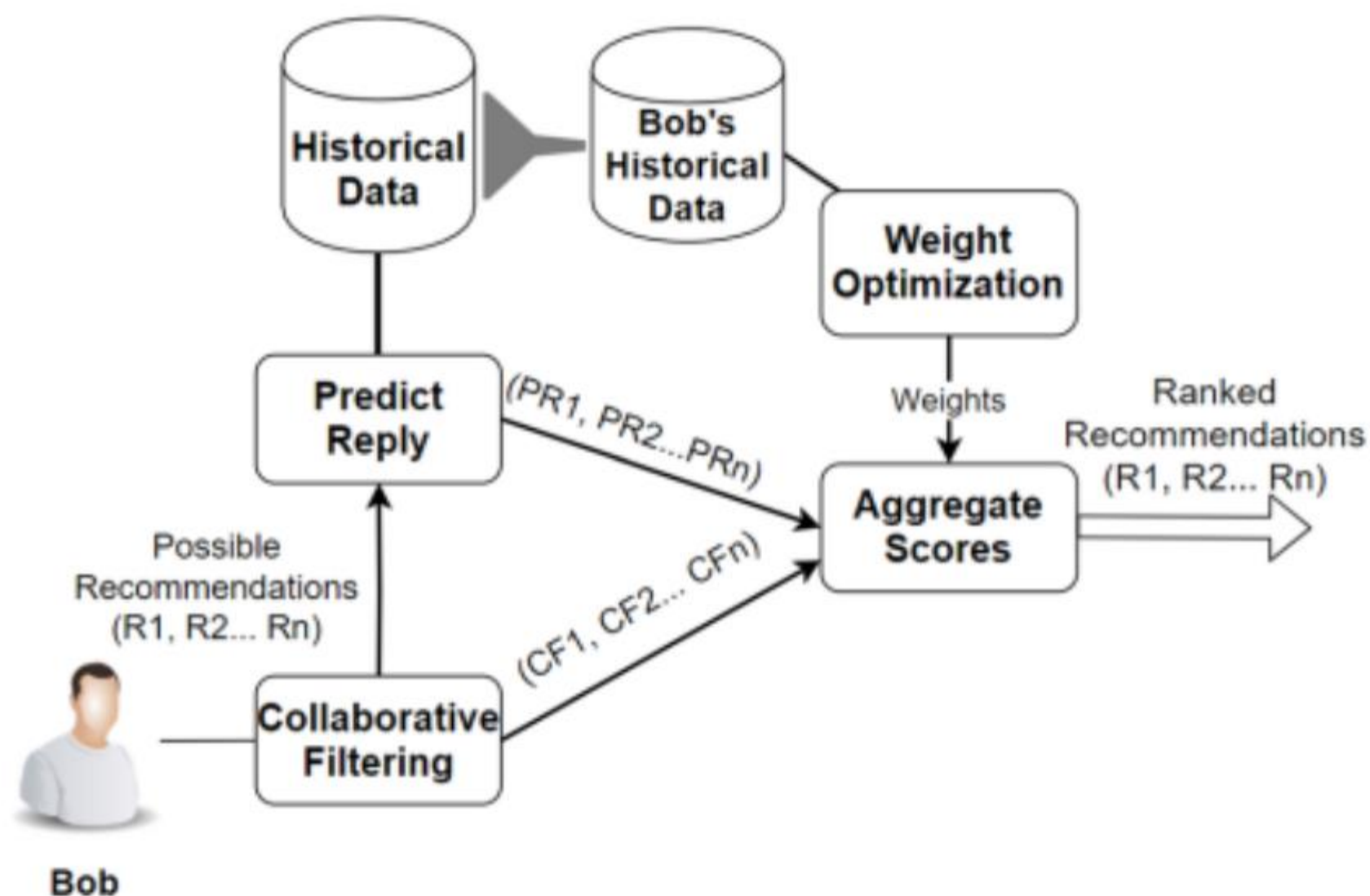


Fig. 6 A diagrammatic representation of RWS. The "Predict Reply" component is described in Sect. 6.2, and the "Weight Optimization" component is described in Sect. 6.4

paper의 p.562 참조, 주피터 파일의 13번 cell 위 참조

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현 - 프로세스 설명

paper에 따르면 상호 추천 알고리즘을 구현하기 위해서는

- [1] 상호 협업 필터링(Reciprocal Collaborative Filtering Recommendation) 프로세스 구현
- [2] 유저의 응답 예측 모델(Predicting replies of Recommended users) 학습 및 생성
- [3] Reciprocal Explanations (상호 추천 방법론) 구현
- [4] Correlation-based Explanation (상관관계 기반 설명론) 구현
- [5] Optimal Weighted Score Recommendations Schema (추천 점수 가중치 찾기)

위의 [1] - [5] 를 통해서 종합적인 상호 추천 알고리즘이 구현이 가능함 !

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현- STEP 1. RCFR

상호 협업 필터링(Reciprocal Collaborative Filtering Recommendation) - RCFR

Algorithm 1 Reciprocal Collaborative Filtering Recommendation

Input: service user x

Output: top-k recommendations

```

1:  $Recs \leftarrow \emptyset$ 
2: for all  $y \in RecommendationCandidates$  do
3:    $score_{x,y} \leftarrow 0, score_{y,x} \leftarrow 0$ 
4:   for all  $u \in SentTo_y$  do ▷ calculate  $x$ 's interest in  $y$ 
5:      $score_{x,y} \leftarrow score_{x,y} + Similarity_{x,u}$ 
6:   for all  $v \in SentTo_x$  do ▷ calculate  $y$ 's interest in  $x$ 
7:      $score_{y,x} \leftarrow score_{y,x} + Similarity_{y,v}$ 
8:    $score_{x,y} \leftarrow \frac{score_{x,y}}{|SentTo_y|}$  ▷ normalize scores
9:    $score_{y,x} \leftarrow \frac{score_{y,x}}{|SentTo_x|}$ 
10:  if  $score_{x,y} = 0$  or  $score_{y,x} = 0$  then
11:     $reciprocalScore_{x,y} \leftarrow 0$ 
12:  else
13:     $reciprocalScore_{x,y} \leftarrow \frac{2}{score_{y,x}^{-1} + score_{x,y}^{-1}}$  ▷ save the harmonic mean of both scores
14:   $Recs \leftarrow Recs + (y, reciprocalScore_{x,y})$ 
15: sort  $Recs$  and return top-k

```

추천 받을 유저 x 와 추천 후보 집단 유저 y 간의 상호 관심도를 연산하는 알고리즘

x 에 대한 y 의 관심도, y 에 대한 x 의 관심도를 구하여 서로 조화 평균을 내어 내림차순으로 추천 리스트 순위를 만들

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

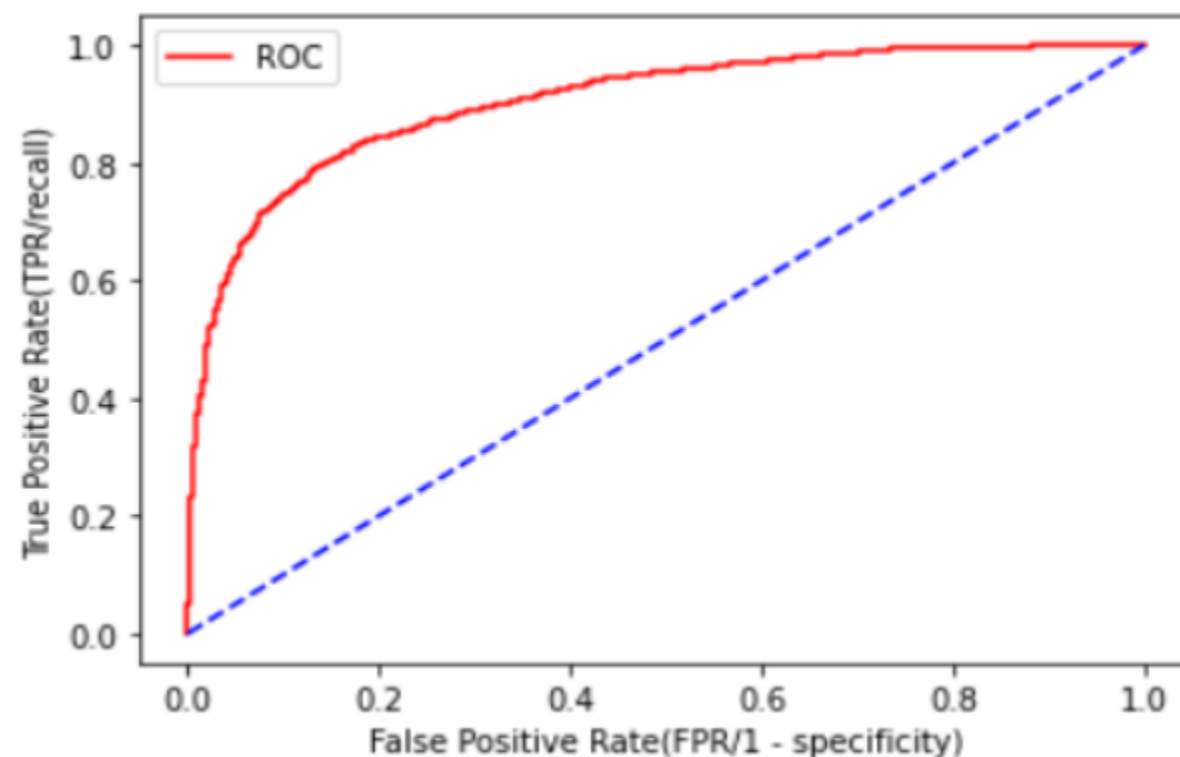
d. 알고리즘 구현

상호 추천 알고리즘 구현- STEP 2. PR

유저의 응답 예측 모델(Predicting replies of Recommended users) 생성

- 특정 유저 x와 잠재적인 x의 추천 유저 y에 대하여 기존의 활동 로그 (로그인 수, 정보업데이트 수, 긍정 응답률 등)와 프로필 특성(나이, 키, 체중, 재산, 연봉 등)을 통해 유저 y가 유저 x에 긍정적으로 응답할 확률을 예측하는 모델 생성 - **AdaBoost 모델 활용!**
- (응답 1, 무응답 0인 이진 분류 모델로 학습하여 proba 값 사용)

* 유저 응답 데이터가, 무응답 데이터에 비해 현저히 작아 불균형 데이터 처리 **SMOTE 기법 활용***



AUC Score : 0.905

Recall : 0.798

Precision : 0.860

f1 score : 0.828

의 AdaBoost 모델 활용 응답률 모델 학습 및 생성

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현- STEP 2. PR

유저의 응답 예측 모델(Predicting replies of Recommended users) 생성

Weight	Feature
0.1276 ± 0.0105	send_cnt
0.1024 ± 0.0107	receiver_send_cnt
0.0779 ± 0.0085	receiver_sent_cnt
0.0515 ± 0.0080	sent_cnt
0.0165 ± 0.0058	receiver_viewed_cnt
0.0045 ± 0.0031	receiver_view_cnt
0.0045 ± 0.0058	viewed_cnt
0.0040 ± 0.0032	신장
0.0029 ± 0.0013	receiver_서비스이용시간
0.0028 ± 0.0029	재산
0.0027 ± 0.0015	체중
0.0020 ± 0.0011	receiver_신장
0.0017 ± 0.0030	서비스이용시간
0.0016 ± 0.0022	성격_c1
0.0016 ± 0.0044	성별_f
0.0015 ± 0.0015	학력
0.0012 ± 0.0015	나이
0.0012 ± 0.0015	성별_m
0.0011 ± 0.0006	지역_b
0.0010 ± 0.0015	로그인수
... 222 more ...	

생성한 모델을 eli5 에서 제공하는

PermutationImportance 라이브러리를 활용하여

해당 PR 모델에서의 feature 변수 중요도를 알아봄

(1) send_cnt : 메시지 받은 수

(2) sent_cnt : 메시지 보낸 수

당연스럽게 생각되는 메시지 송,수신 feature 외에

회원 프로필에서 '신장', '재산', '체중' 변수가 중요함을 알

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현- STEP 3. RE + STEP 4. CBE

[Reciprocal Explanations (상호 설명론) + Correlation-based Explanation (상관관계 기반 설명론)]

Algorithm 2 Reciprocal Explanations

Input: User x , *GenerateRecommendations*: a Recommendation method, returns a list of recommended matches, *Explain*: an explanation method

- 1: $Output \leftarrow \emptyset$
- 2: $R \leftarrow \text{GenerateRecommendations}(x)$
- 3: **for all** $r \in R$ **do**
- 4: $e_{x,r} \leftarrow \text{Explain}(x, r)$
- 5: $e_{r,x} \leftarrow \text{Explain}(r, x)$
- 6: $re_{x,r} \leftarrow (e_{x,r}, e_{r,x})$
- 7: $Output = Output \cup (r, re_{x,r})$
- 8: **return** $Output$

Algorithm 3 Correlation-based Explanation Method

Input: two users x and y , number of attributes for explanation k .

- 1: $temp \leftarrow \emptyset$
- 2: obtain m_x
- 3: **for all** attributes $a \in A$ **do**
- 4: obtain the value a_v of attribute a in A_y
- 5: obtain s_{x,a_v}
- 6: $w_{a_v} = \text{CORRELATION}(m_x, s_{x,a_v})$
- 7: $temp = temp \cup (a_v, w_{a_v})$
- 8: **sort** $temp$ by the values w_{a_v}
- 9: $e_{x,y} = \text{top-}k \text{ attribute values of } temp$
- 10: **return** $e_{x,y}$

- 상호 설명론(RE): 유저 x 가 추천 후보 y 를 추천 받은 이유에 대해 설명하는 방법

기존 x, y 유저가 메시지를 보낸 유저의 모든 특성값에 대해 응답 유무에 따라 유저 이진 벡터 값을 추출해서, 유저 이진벡터와 특성 값의 상관관계를 측정하여 상관관계가 가장 큰 특성값 상위 k 개 추출해 해당 추천에 대한 이유를 설명하는 프로세스

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현- STEP 5. OWSR

Optimal Weighted Score Recommendations Schema (추천 점수 가중치 찾기)

Algorithm 4 Reciprocal Weighted Score Recommendations Scheme

Input: service user x

Output: top-k recommendations

```

1:  $Recs \leftarrow \emptyset$ 
2: for all  $y \in RecommendationCandidates$  do
3:    $CF_{x,y} \leftarrow 0$ 
4:   for all  $n \in SentTo_y$  do                                ▷ calculate  $x$ 's interest in  $y$ 
5:      $CF_{x,y} \leftarrow CF_{x,y} + Similarity_{x,n}$ 
6:    $CF_{x,y} \leftarrow \frac{CF_{x,y}}{|SentTo_y|}$                                 ▷ normalize score
7:   if  $CF_{x,y} = 0$  then
8:      $reciprocalScore_{x,y} \leftarrow 0$ 
9:   else
10:     $\mathbf{x} \leftarrow ServiceUserFeatures(x)$ 
11:     $\mathbf{y} \leftarrow RecommendedUserFeatures(y)$ 
12:     $PR_{y,x} \leftarrow PredictReply(\mathbf{x}, \mathbf{y})$                                 ▷ predict  $y$ 's response to  $x$ 
13:     $\alpha \leftarrow OptimizedWeight(x)$ 
14:     $reciprocalScore_{x,y} \leftarrow (\alpha \cdot CF_{x,y} + (1 - \alpha) \cdot PR_{y,x})$                                 ▷ aggregate scores
15:   $Recs \leftarrow Recs + (y, reciprocalScore_{x,y})$ 
16: sort  $Recs$  and return top-k

```

유저의 특성에 따라 [1] 관심도, [2] 상대 유저의 응답률을 중시하는 비중이 다름

- 특정 유저 x에게 과거에 타 유저와 긍정적 상호작용에 대한 데이터가 있을 경우, 가중치를 추출하여 위에서 구한 RCF, PR score에 각각 곱하고 최종 점수를 도출함

- paper 에서 는 default 값으로 알파 0.3978 제시하였으나, 구해보니 이 데이터는 0.2 가 최적의 가중치

02 프로젝트 내용

a. 활용한 데이터

b. 데이터 EDA

c. 데이터 전처리

d. 알고리즘 구현

상호 추천 알고리즘 구현 결과물

주피터 파일 62 cell에서 `get_reciprocal_score` 메소드를 통하여, 최종 알고리즘 구현 확인 가능!

예시로 임의 회원 번호 20806을 넣었을 때,
최적의 가중치(optimal weight)는 0.2 이고

```
get_reciprocal_recommendation(x)
```

```
optimal weight : 0.2
```

```
my_candidate_num : 1760
```

```
20806와 상관관계가 가장 높은 feature는 : 성격, corr value : 0.49
```

```
19156와 상관관계가 가장 높은 feature는 : 신장, corr value : nan
```

```
((19156, 1.1535785726901626)),
```

```
[('유저 20806에게 유저 19156가 잘 어울리는 이유 =====> 1순위 성격, 2순위 사진개수, 3순위 연봉',  
  '유저 19156에게 유저 20806가 잘 어울리는 이유 =====> 1순위 신장, 2순위 체중, 3순위 혈액형'))]
```

20806 유저는 추천회원으로 유저 19156를 추천받게 되고 그 이유는 성격, 사진개수, 연봉 때문임
추천되어지는 19156 유저는 자신이 추천된 상대 20806과는 신장, 체중, 혈액형으로 인해 잘 어울릴 것
이라고 설명해주는 것!

03

프로젝트 발전 가능성

a. 프로젝트 발전 가능성

03 프로젝트 발전 가능성

a. 프로젝트 발전 가능성

프로젝트 발전 가능성

- 기존의 추천 시스템에서 고려 받지 않았던 추천 되는 유저의 선호를 고려하여, 완전한 개인화 추천 시스템으로 구현될 수 있지 않을까 하는 기대감?
- 추천 시스템에서 설명을 제공하는 것은 추천의 수용률을 증가시키는데 효과적이라는 사실이 입증되어 옴에 따라, 추천되는 이유를 양 유저에게 설명함으로써 유저 만족감 증가 예상

04

프로젝트 후기

- a. 프로젝트 후기
- b. 업데이트 계획

04 프로젝트 후기

a. 프로젝트 후기

프로젝트 후기& 업데이트 계획

Paper에서 말하고 있는 상호추천 시스템의 중요 포인트는 '구체적인 특징을 포함하는 설명은 효과적이지만 **정보 과부화(information overload)**로 올 역효과를 피하기 위해서 '설명'의 길이를 제한' 이 필요하다고 한다.

회원 개인을 나타내는 특성의 수를 제한해야 하는데, 이러한 부분에 대해서 어느 부분을 제거하고, 추가(파생 변수) 해야 하는지에 대한 심도 있는 고민이 필요할 것 같다.

또, RRSs 추천 방식이 정말로 효과가 있는지 일반 추천과의 수용률을 알기 위해서는 A/B 테스트가 필요할 것 같은데.. 기회가 된다면 실서비스 릴리즈해서 비교해 보고 싶음

추후 보안 방법으로 선호도 측정 점수 데이터를 받아서 효과를 입증해보고 싶다..(?)

04 프로젝트 후기

a. 활용한 데이터

b. 업데이트 계획

업데이트 계획

위에서 구현한 데이터들은 실제 데이터들을 변형한 데이터라서,
실제 데이터 구현을 해보면 또 다른 문제점이 발생할 수 있을 것 같습니다.

특히 실서비스에서 구현시 고려되는 것은 정확도 뿐 아니라
(1) 유저 한 명당 추천 상대를 추출해 내는데 걸리는 시간
(2) 구현한 모듈 및 모델이 점유하고 있는 메모리
등이 고려되어야 하는데, 사이드 프로젝트 용으로 구현해 놓은 상태라
위의 두 사안이 완벽히 고려하지는 않아서
실제 서비스를 위해서 실 데이터를 가지고 테스트를 해보면서 지속적으로
업데이트 해볼 생각입니다

끝 ~
