WATCHAEK

2022-2 KUBIG 왓책피디아 팀

우명진 임정준 하예은

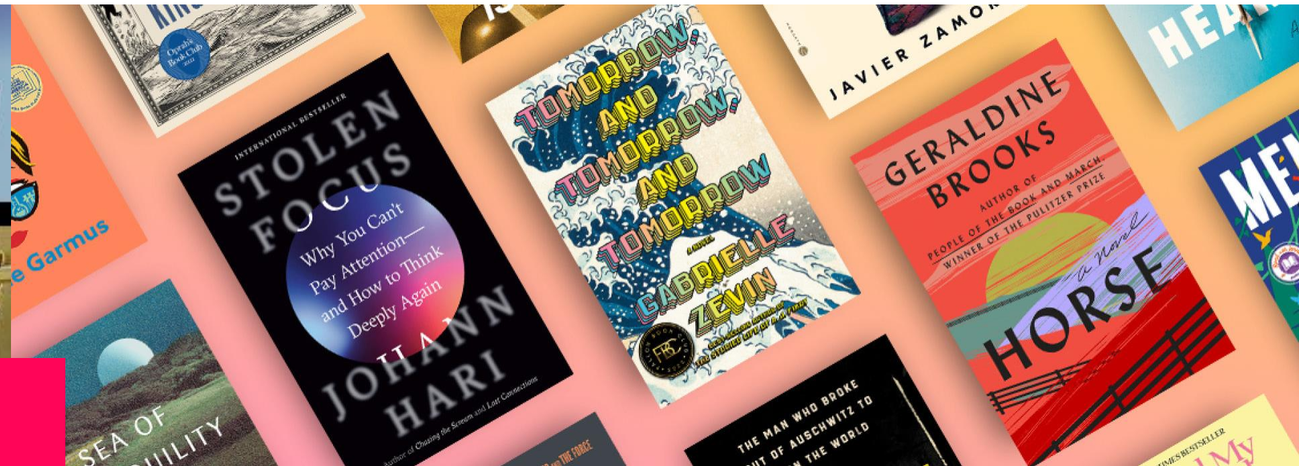
목차

01 연구 배경

사용자 영화취향을 반영한 크로스미디어 플랫폼 도서 추천 시스템(김성섭, 2021)

사용자 평점 기록이 적은 도서는 추천 정확도에 한계

사용자 평점 데이터가 풍부한 영화 평점 정보로
맞춤형 도서를 추천하는 추천 시스템 제안



02 데이터 확보

이동진 평론가가 평가한 영화의 정보
& 유저들의 코멘트 크롤링
2244 x 10 크기의 데이터 확보

이동진 평론가가 평가한 도서의 정보
& 유저들의 코멘트 크롤링
210 x 10 크기의 데이터 확보

03 데이터 전처리

1. ITEM DATA

(1) 영화 데이터 '장르' & 도서 데이터 '카테고리' 결측치 채우기

(2) 영화 데이터의 영화 제목 끝에 별표(*) 추가
: 영화 데이터와 도서 데이터를 병합했을 때 구분하기 위함

(3) 영화 데이터 '장르' & 도서 데이터 '카테고리' column명을 '카테고리'로 통일

(4) 제목, 연도, 카테고리, 내용 column 제외하고 나머지 column drop

(5) 영화 데이터 & 도서 데이터 병합하여 item data 생성

03 데이터 전처리

1. ITEM DATA

	제목	연도	카테고리	내용
0	물방울을 그리는 남자*	2020.0	다큐멘터리	50년간 묵묵히 '물방울'만을 그리며?물방울 작가로 사랑받은 화가 김창열. 침묵과 ...
1	2차 송환*	2022.0	다큐멘터리	"한국전쟁은 아직 끝나지 않았다!"?₩n₩n현재까지 남과 북은 정치 공작원들을 상호...
2	달이 지는 밤*	2020.0	모험	중년의 여인이 무주 시외버스 터미널에서 내린다. 그녀는 마을길을 지나 숲으로 들어간...
3	썬더버드*	2021.0	범죄/액션	돈이 미치게 필요한 태균₩n돈이 든 자동차를 잃어버린 태민₩n돈은 중요하지 않은 미...
4	기기묘묘*	2022.0	공포/액션/스릴러	땅을 둘러 싼 이웃들의 다툼,?엄마와 딸의 기이한 관계,?낙향한 청년과 괴인의 기묘...
...	...	...	...	...
205	0년	2016.0	역사/문화	1945년이라는 한 해를 대상으로 세계사를 써내려간 독특한 역사서이자 논픽션 다큐멘...
206	아날로그의 반격	2017.0	사회과학	디지털 라이프가 영구적인 현실이 된 지금, 새로운 얼굴을 한 아날로그가 유행하기 시...
207	마이클 케인의 연기 수업	2017.0	예술/대중문화	세계적인 명배우 마이클 케인이 배우를 꿈꾸는 이들에게 자신의 연기 노하우를 생생하게...
208	아프리카 우화집	2009.0	소설	야생의 땅 아프리카가 들려주는 옛이야기 스물아홉 편을 우화집으로 엮었다. 아프리카 ...
209	노마드랜드	2021.0	사회과학	미국에서 고정된 주거지 없이 자동차에서 살며 저임금 떠돌이 노동을 하는 사람들의 삶...

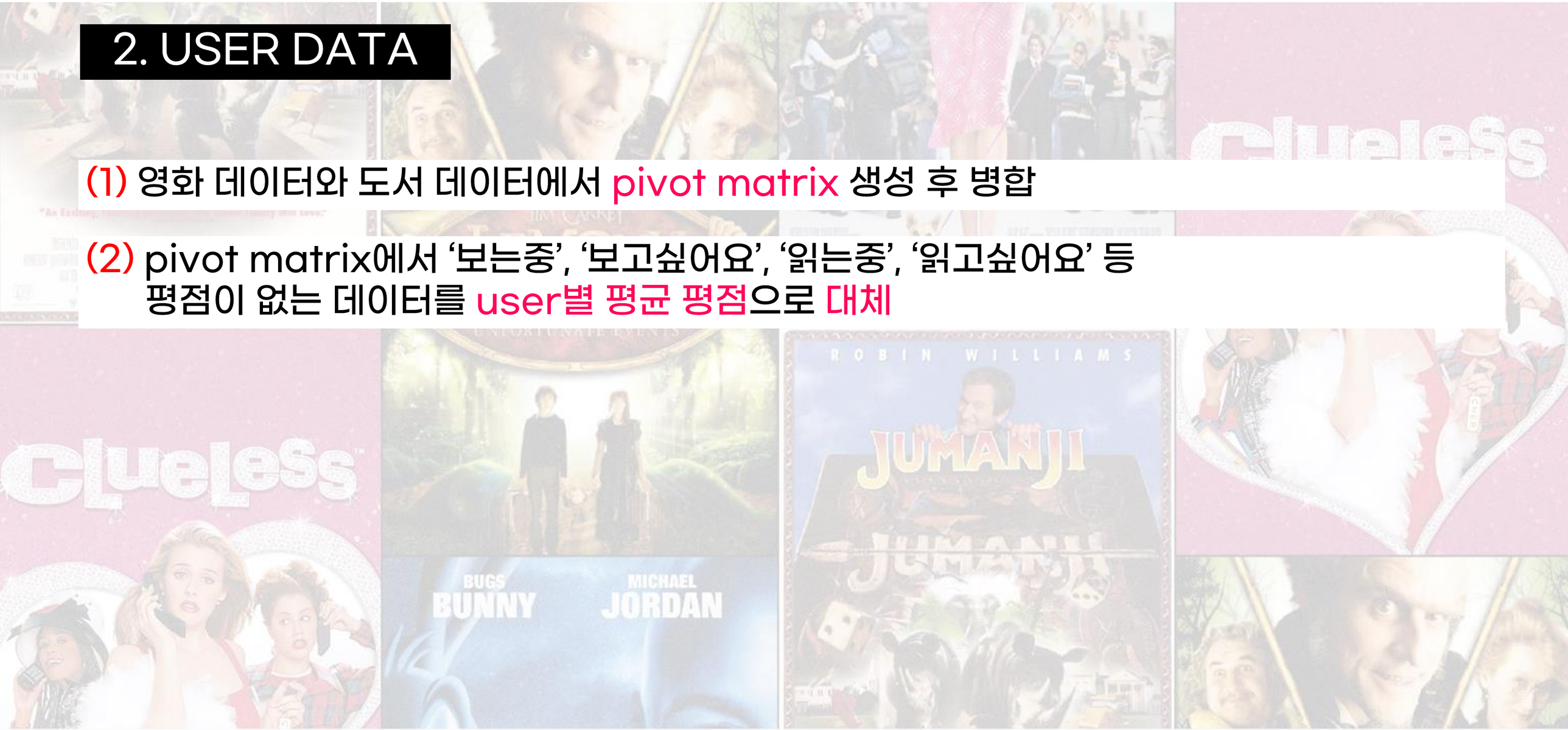
2454 rows × 4 columns

03 데이터 전처리

2. USER DATA

(1) 영화 데이터와 도서 데이터에서 pivot matrix 생성 후 병합

(2) pivot matrix에서 ‘보는중’, ‘보고싶어요’, ‘읽는중’, ‘읽고싶어요’ 등
평점이 없는 데이터를 user별 평균 평점으로 대체



03 데이터 전처리

2. USER DATA

	제목	이동진 평론가	E열표	석미인	Hana	성유	정선주	134340	성빵	Ziwoo	...	언블
0	물고기는 존재하지 않는다	5.0	4.5	4.5	4.0	5.0	5.0	3.5	5.0	5.0	...	NaN
1	예술가들의 이상심리	2.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
2	어느 독일인 이야기	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
3	세상에서 가장 널리 알려진 미신의 숫자 13	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
4	미신의 연대기	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
2449	메피스토	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
2450	앤트원 피셔	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4.5
2451	모베터 블루스	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
2452	스피드 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
2453	사고친 후에	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN

2454 rows × 8630 columns

03 데이터 전처리

3. TUPLE 변환

(1) item data의 경우 (item_id, [제목, 연도, 카테고리, 내용])으로 tuple 변환하여
rating_source에 저장

(2) user data의 경우 (사용자, movie_id, 평점)으로 tuple 변환하여
item_features_source에 저장

LightFM

(1) 콘텐츠 기반 필터링 + 협업 필터링인 hybrid 모델
(2)유저와 아이템 각각의 feature 둘 간의 상호작용 모두 고려
(3)Cold start 문제 완화
(4)Output은 해당 유저가 해당 아이템을 선호할 확률

→ item data와 user data 모두 사용!

04 모델링

1. Dataset Construction

(1) fit 메서드에 user id와 item id, flatten한 item feature를 인자로 넣음

(2) build_interactions 메서드에 user 정보가 담긴 rating_source를 넣어 interaction 데이터셋과 weight 생성

(3) build_item_features 메서드에 item 정보가 담긴 item_features_source를 넣어 item feature 데이터셋 생성

04 모델링

2. Hyperparameter Tuning

(1) LightFM parameter

- no_components : feature latent embedding의 차원
- learning_schedule : optimizer 결정 (adagrad/adadelta 중 선택)
- loss : 손실함수 결정 (logistic/bpr/warp/warp-kos 중 선택)
- learning_rate : adagrad 초기 학습율

04 모델링

2. Hyperparameter Tuning

(2) Tuning 과정

- no_components : 80, learning_rate : 0.01 → 0.7438의 AUC 도출

```
learning_rate=[1e-2, 5e-3, 1e-3]
n_components=[100,90,80]
epoch=10
auc=0.0
```

```
0.7515208 (0.01, 100)
0.75508016 (0.01, 90)
```

- no_components : 90, learning_rate : 0.01 → 0.7551의 AUC 도출

```
learning_rate=[1e-2, 5e-3, 1e-3]
n_components=[80,60,20]
epoch=10
auc=0.0
```

```
0.74382305 (0.01, 80)
```

04 모델링

2. Hyperparameter Tuning

(2) Tuning 결과
- no_components : 100, learning_rate : 0.05 → 0.9945의 AUC 도출

```
learning_rate=[1e-2, 5e-2, 1e-3]
n_components=[100,90,80]
epoch=20
auc=0.0

for lr in learning_rate:
  for n in n_components:
      model = LightFM(no_components= n,learning_schedule='adagrad',learning_rate=lr, loss='warp')
      model.fit(interactions=interactions,item_features=item_features,sample_weight=weights,epochs=epoch)
      train_auc = auc_score(model,interactions,item_features=item_features).mean()
      if train_auc>auc:
        auc=train_auc
        best_param=(lr, n)
        print(auc, best_param)
```

```
0.80218035 (0.01, 100)
0.9945072 (0.05, 100)
```

05 결과

(1) 매핑 딕셔너리 생성 : id(0, 1, 2,…)와 영화제목/사용자를 매핑하는 딕셔너리 생성

(2) 도서 데이터 중 추천 예측 : 평점이 0 이상인 도서를 제외하고 예측값 추출

(3) user_id를 입력하여 해당 사용자에 맞는 도서 추천 결과 출력

```
predict_book(df,model, 1,user_dict,item_dict)
```

```
['박시백의 조선왕조실록', '프로젝트 헤일메리', '노마드랜드', '휴먼카인드']
```

(4) 모든 user_id에 대해 추천 도서 목록 출력 후 csv 파일로 저장

(5) 도서 추천 웹 구현 : https://kubig-book.run.goorm.io/

의의

(1) 기존 도서 추천 시스템과 달리 영화의 취향을 고려하여 도서를 추천하는 새로운 방식 시도

(2) 크롤링을 통해 얻은 사용자의 정보와 아이템의 정보를 모두 활용한 하이브리드 모델 개발

(3) 하이퍼파라미터 조절을 통한 성능 개선. test set의 성능을 평가하지 못한 것이 아쉬움.

THANK YOU

2022-2 KUBIG 왓책피디아 팀

우명진 임정준 하예은