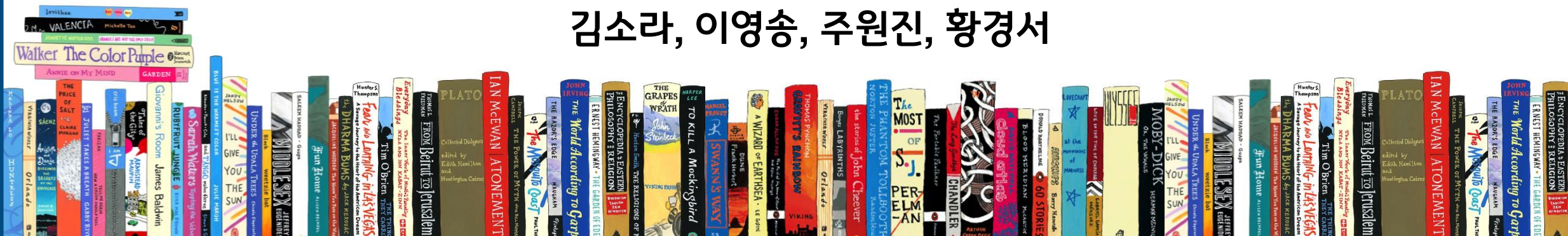


Thompson Sampling을 통한 맞춤형 책 추천 시스템

마음의 양식 with YES^{!!}24.COM
김소라, 이영송, 주원진, 황경서





목차

YES24.COM

I. 프로젝트 배경

II. 데이터 설명

III. 모델링 결과

IV. 결론

V.부록



I. 프로젝트 배경

A. 프로젝트 개요

1) As-is yes24 추천 시스템

국내도서 외국도서 eBook 웹소설/코믹 CD/LP DVD/BD 영화 공연 문구/GIFT 중고샵 iSTYLE24 로그인 회원

YES24.COM

국내도서 류근관

스타워즈세트

빠른분야찾기 베스트 신상품

통합검색 275

국내도서 102

eBook 68

CD/LP 15

다운로드 15

중고샵 86

리뷰 1

류근관

류근관 법문사

류근관 저

류근관 korea

류근관 hill

류근관 mcgraw

류근관 맥그로힐

[도서] 통계학
류근관 저 | 법문사 | 2013년 02월
35,000원 → 35,000원(0% ↓ +3% P)

[도서] 전자회로
Albert Malvino, David J. Bates 공저/진경복, 류근관, 박진현, 정연호
공역 | 한국맥그로힐(McGraw-Hill KOREA) | 2016년 01월
38,000원 → 38,000원(0% ↓ +0% P)

[도서] 해방 이후 한국기업의 진화 1 : 1976~2005년간의 통계의
구축과 기초분석 서울대학교 규장각 한국학연구원 한국학 자료총
서 07/서울대학교 규장각한국학연구원 한국학자료총서 07
이근, 류근관, 이창윤 등저 | 서울대학교출판부 | 2007년 11월
12,000원 → 12,000원(0% ↓ +0% P)

1 2 검색결과 5건

기능끄기

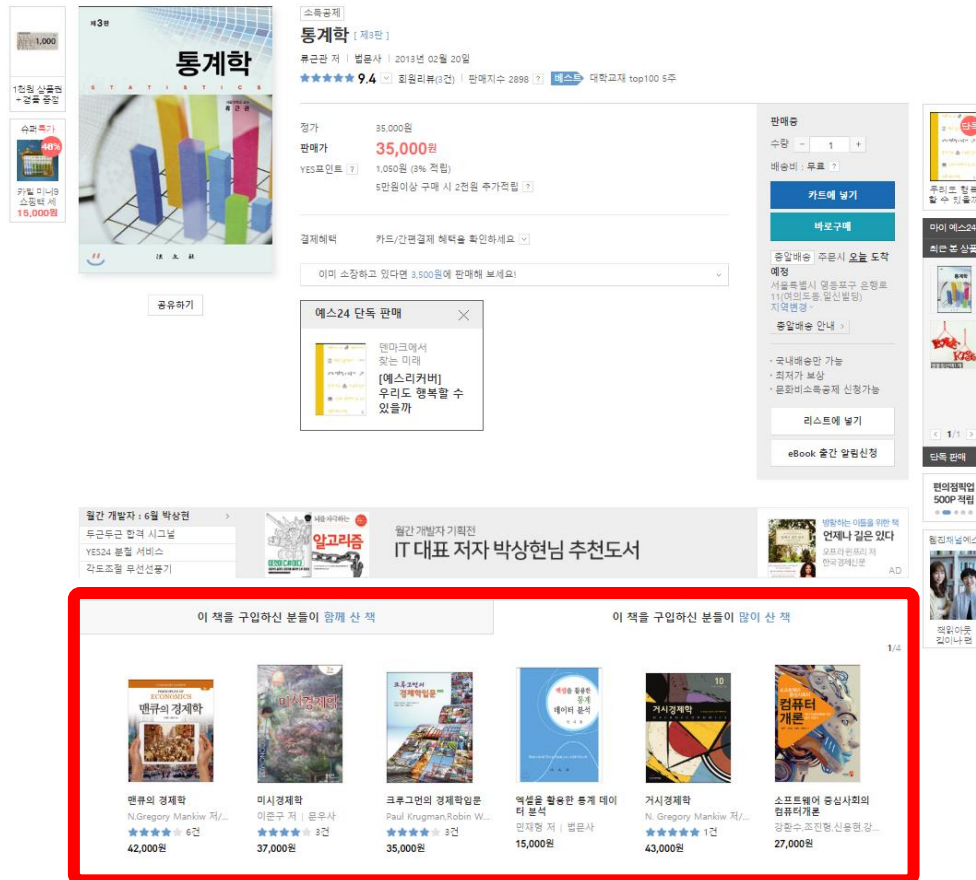
서비스소개 및 의견수집

특정 조건의 키워드 검색 혹은 추천 상품의 클릭 등을 통해 유입

I. 프로젝트 배경

A. 프로젝트 개요

1) As-is yes24 추천 시스템



상세페이지 내의 추천 종류

[이 책을 구입한 분들이 함께 산 책]
[이 책을 구입한 분들이 많이 산 책]

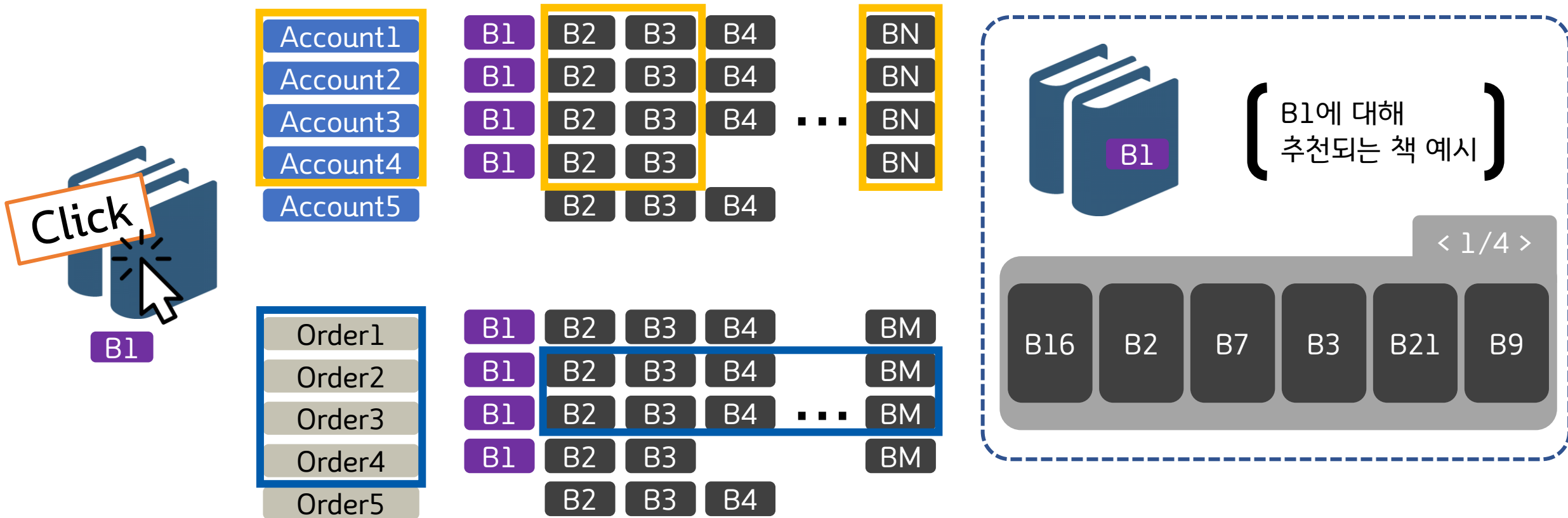
상세페이지 내의 추천 알고리즘

1. 해당 책을 기준으로 주문ID 매개로 함께 구매한 책/
고객 Account 매개로 많이 산 책들의 구매 이력 종합
2. 판매수가 높은 상품 24권을 추출한 후
랜덤으로 유저에게 노출
cf. 6권 묶음으로 1~4페이지도 랜덤하게 노출

I. 프로젝트 배경

A. 프로젝트 개요

1) As-is yes24 추천 시스템



I. 프로젝트 배경

A. 프로젝트 개요

2) As-is yes24 추천 시스템의 문제점

- 맞춤형 추천이 어려움
→보편적으로 가장 많이 읽히는 책이 추천될 가능성이 높음
- Diversity, Novelty Problem
→ 신규 도서나 마니아를 위한 책이 탐색될 가능성이 적음
- Update Problem
→ 추천 된 상품의 랭킹에 따른 결과 누적이 되지 않음

I. 프로젝트 배경

B. 프로젝트 주제

1) To-be 추천 시스템



#30살 남자
#판타지 러버
#한 달에 1권



#16살 여자
#SF빠
#생각날 때 가끔



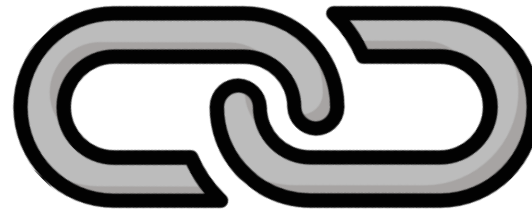
I. 프로젝트 배경

B. 프로젝트 주제

1) To-be 추천 시스템



책을 만드는 사람



연결고리



사람을 만드는 책

I. 프로젝트 배경

B. 프로젝트 주제

- ① 수 많은 책에 대한 Multi armed bandit problem
- ② 수 많은 유저와 책의 Context를 반영
- ③ Bayesian inference 바탕의 Thompson Sampling 알고리즘의 적용을 통해 Context 상황에서 Bandit의 Reward를 극대화
- ④ 추천 풀 vs 랭킹, 두 가지 방식으로 적용 가능
- ⑤ Yahoo, Amazon, Netflix 같은 IT 기업을 통한 간접적인 실증

II. 데이터 설명

A. 데이터 설명

1) Raw Data(3개월)

- Accounts.json : 고객 데이터
- Products.json : 책 정보 데이터
- Clicks.json : 고객ID데이터 별 상품(책) 클릭 로그 데이터
- Orders.json : 상품(책) 주문 데이터

2) Processed Data

- Book_train : 3월,4월 데이터
- Book_test : 5월 데이터

II. 데이터 설명

B. 데이터 변수 설명

1) Accounts.json (1741578 rows x 5 columns)

변수명	변수 설명	데이터 타입
account_id	고객 아이디	int64
gender	성별(Female, Male)	object
age	나이	float64
address	주소	object
last_login_dts	최근 로그인 날짜(Java script Date)	float64

II. 데이터 설명

B. 데이터 변수 설명

2) Products.json (1745066 rows x 6 columns)

변수명	변수 설명	데이터 타입
product_id	상품 아이디	int64
product_name	상품명	object
category_id	카테고리 아이디	int64
published_at	출판일(Java script Date)	object
shop_price	상품가격(₩)	float64
maker_name	출판사	object

II. 데이터 설명

B. 데이터 변수 설명

3) Clicks.json (60265430 rows x 5 columns)

변수명	변수 설명	데이터 타입
request_date_time	클릭 날짜, 시간(Java script Date)	object
account_id	고객 아이디	int64
device_type	사용 기기(PC, Mobile)	object
product_id	상품 아이디	int64
before_product_id	직전 클릭 상품 아이디	int64

II. 데이터 설명

B. 데이터 변수 설명

4) Orders.json (8382514 rows x 5 columns)

변수명	변수 설명	데이터 타입
orders_id	주문 아이디	int64
account_id	고객 아이디	int64
product_id	상품 아이디	int64
price	주문 가격(₩)	float64
created_at	주문 날짜(Java script Date)	int64

II. 데이터 설명

C. Processed Data

구매량 기준 TOP 24개의 책 기준으로 3개월 동안의 데이터 추출(총 1,148,637 건)
책 상세페이지 Click = '추천'을 받은 상황

	product_id	year	gender	purchase	month
0	84659792	0	F	0.0	5
1	86895523	1	F	1.0	5
2	81503945	0	F	0.0	5
3	81503945	0	F	0.0	5
4	86591861	1	F	0.0	3
...

- **product_id**
각 책의 일련번호
- **Year**
신규도서여부
(2020년 출판=1, 2020 이전 출판 = 0)
- **gender**
성별
- **purchase**
구매 여부 (구매 =1, 비구매=0)
- **month**
사용자가 해당 도서를 클릭한 달
(사용자에게 해당 도서가 추천된 달)

II. 데이터 설명

C. Processed Data

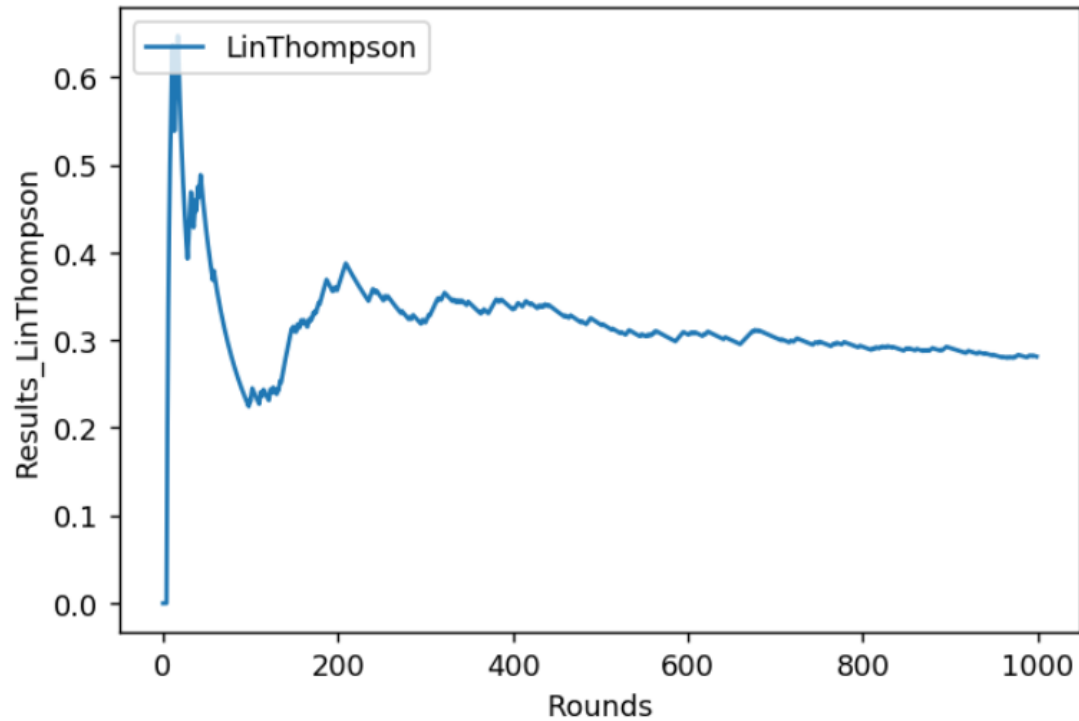
구매량 기준 TOP 24개의 책 기준으로 3개월 동안의 데이터 추출(총 1,148,637 건)

						product_id	purchase	month	F/O/0	F/O/1	...	M/N/14	M/N/15	M/N/16	
0	84659792	0	F	0.0	5	0	12	0.0	5	0	0	...	0	0	0
1	86895523	1	F	1.0	5	1	19	1.0	5	0	0	...	0	0	0
2	81503945	0	F	0.0	5	2	10	0.0	5	0	0	...	0	0	0
3	81503945	0	F	0.0	5	3	10	0.0	5	0	0	...	0	0	0
4	86591861	1	F	0.0	3	4	17	0.0	3	0	0	...	0	0	0
...

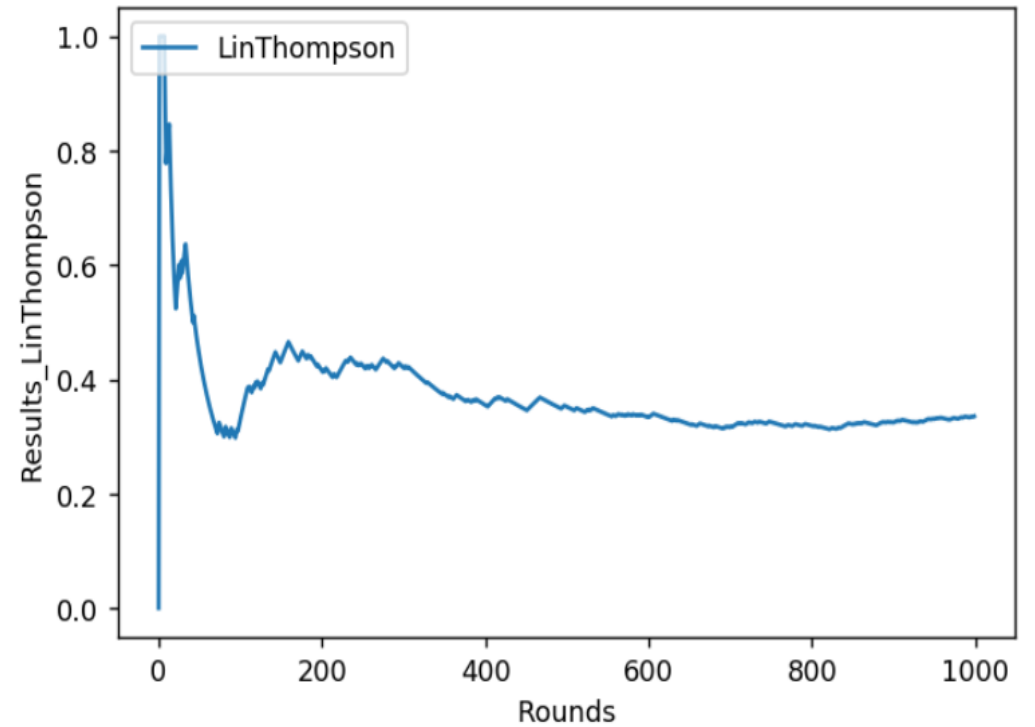
User Feature, Book Feature, 각 Book의 권 수를 조합하여 Dummy 96개 변수 생성
(ex. [여성, 올해 출판 도서, 5번째 Book] ► [F/N/5])

III. 모델링 결과

A. 톰슨 적용 후 데이터 결과 값 비교



Train 수렴 과정



Test 수렴 과정

III. 모델링 결과

A. 톰슨 적용 후 데이터 결과 값 비교

	Yes 24	Linear Thompson
3,4월 구매 기대값	33.09%	33.8%
5월 구매 기대값	28.73%	33.6%

- All Data
-> 상세 정보를 클릭한 경우 = 추천을 받음
- Yes24
클릭 후(추천을 받고) 구매한 비율
- Linear Thompson
0과 1을 예측하여 True = Prediction이 된 비율

IV. 결론

A. 앞으로 나아갈 방향

- 각 추천 시스템 별 평균 Reward 결과값으로 수익성이 높은 추천 시스템 선택 가능
- 추천 대상을 현재 24권에서 N권으로 확장
- Feature Engineering을 통한 개인화 변수 추가로 정교한 추천 시스템 구축 가능
- Non-linear 모델로의 확장 (Gibbs sampling 기법 사용)
- Cholesky Decomposition을 사용하여 Sampling 단계에서 노이즈 추가

IV. 결론

B. 최종 발표를 위한 궁금증

- Context vector의 효율적인 구성
 - sparse matrix 및 차원의 저주 해결 방안
 - Feature Engineering 계획 실현 가능성 정도
- Clustering을 통해 유저와 책의 Feature 단순화 방안
 - 타당성 여부(1 to 1이 가능한가?)
 - Clustering 기준
- 추천 방법론
 1. N권에서 후보 책 선정 시 T/S 사용
 2. 후보 책 선정 후 노출 순서 결정 시 T/S 사용
- Off-Policy 상황에서의 평가 방법

IV. 결론

B. 최종 발표를 위한 궁금증

- Feature Engineering 예시

Book Feature	설명	비고
카테고리	중분류 34개	One-Hot인코딩
● 출판일	6개월 이전/6개월~1년 이내/1년~3년 이내/3년 이후	올해/이전
가격	1. 정규화 분류 2. 평균값 중간값 최반값 사분위수 3. 배송비 포함/불포함 최저 가격	
베스트셀러	1. 전체 기간 베스트셀러 해당 여부 2. 각 달 별 베스트 셀러 해당 여부 3. 구매자 수	One-Hot인코딩 One-Hot인코딩

IV. 결론

B. 최종 발표를 위한 궁금증

- Feature Engineering 예시

User Feature	설명	비고
● 성별	남/여	
나이	1. 정규화 분류 2. 기존 Clustering한 방법론 사용 3. 19세 이하, 20~30, 31~42, 43~50, 51~64, 65세 이상	주관적 분류
주소	수도권(서울,경기도,인천) / 비수도권	
기기	PC/Mobile	
책 Click 수	책에 대한 관여도 = Order 수 / Click 수	추천 민감도
활동 시간	Day/Balance/Night	

IV. 결론

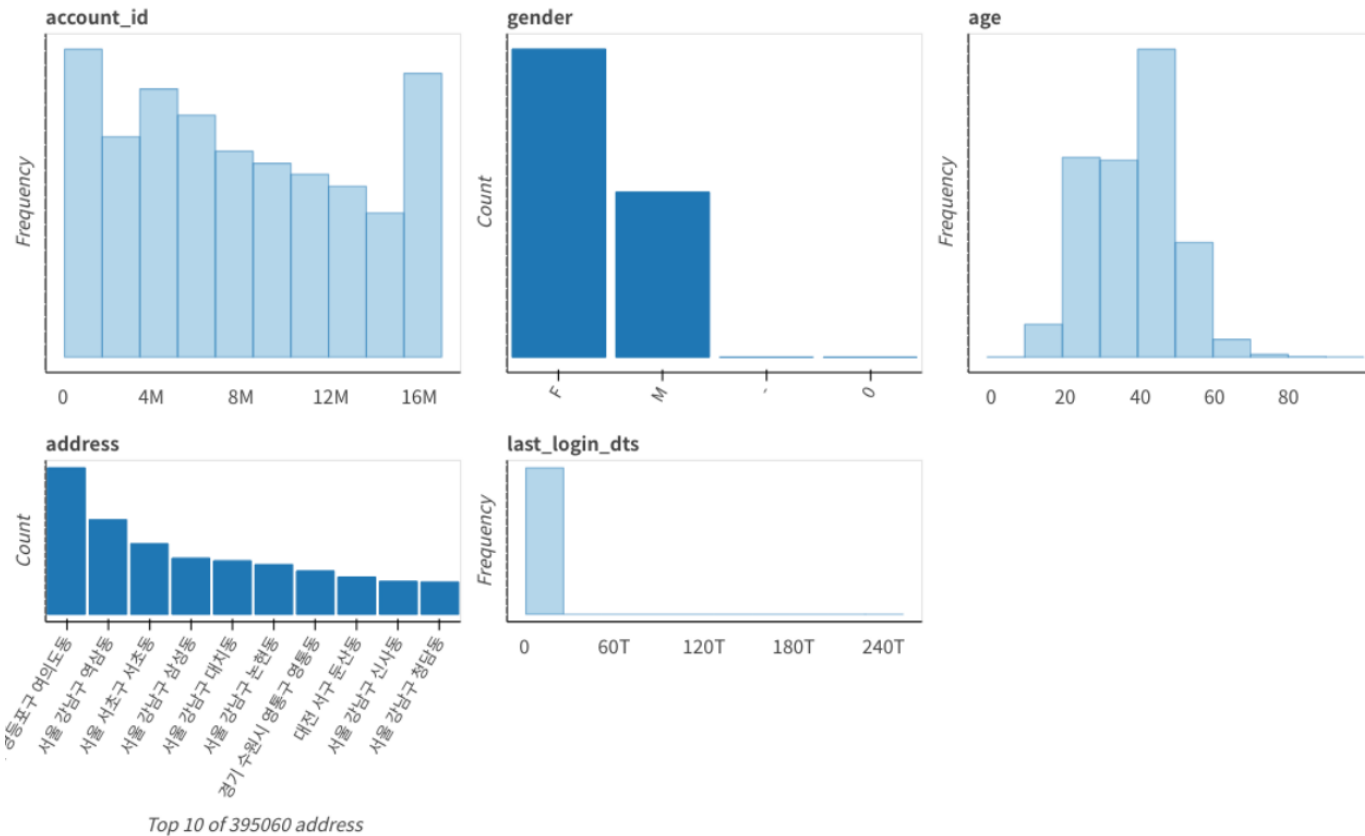
B. 최종 발표를 위한 궁금증

- Feature Engineering 예시

User Feature	설명	비고
활동 요일	주중 / 주말(금,토,일)	
구매 주기	짧은 주기 / 긴 주기 (포아송)	
카테고리 선호도	- 문학 / 비문학 - 수험서 / 비수험서 - 신권 구매 민감 유무	
...		
월별 구매 금액	평균 이상 / 평균 이하	

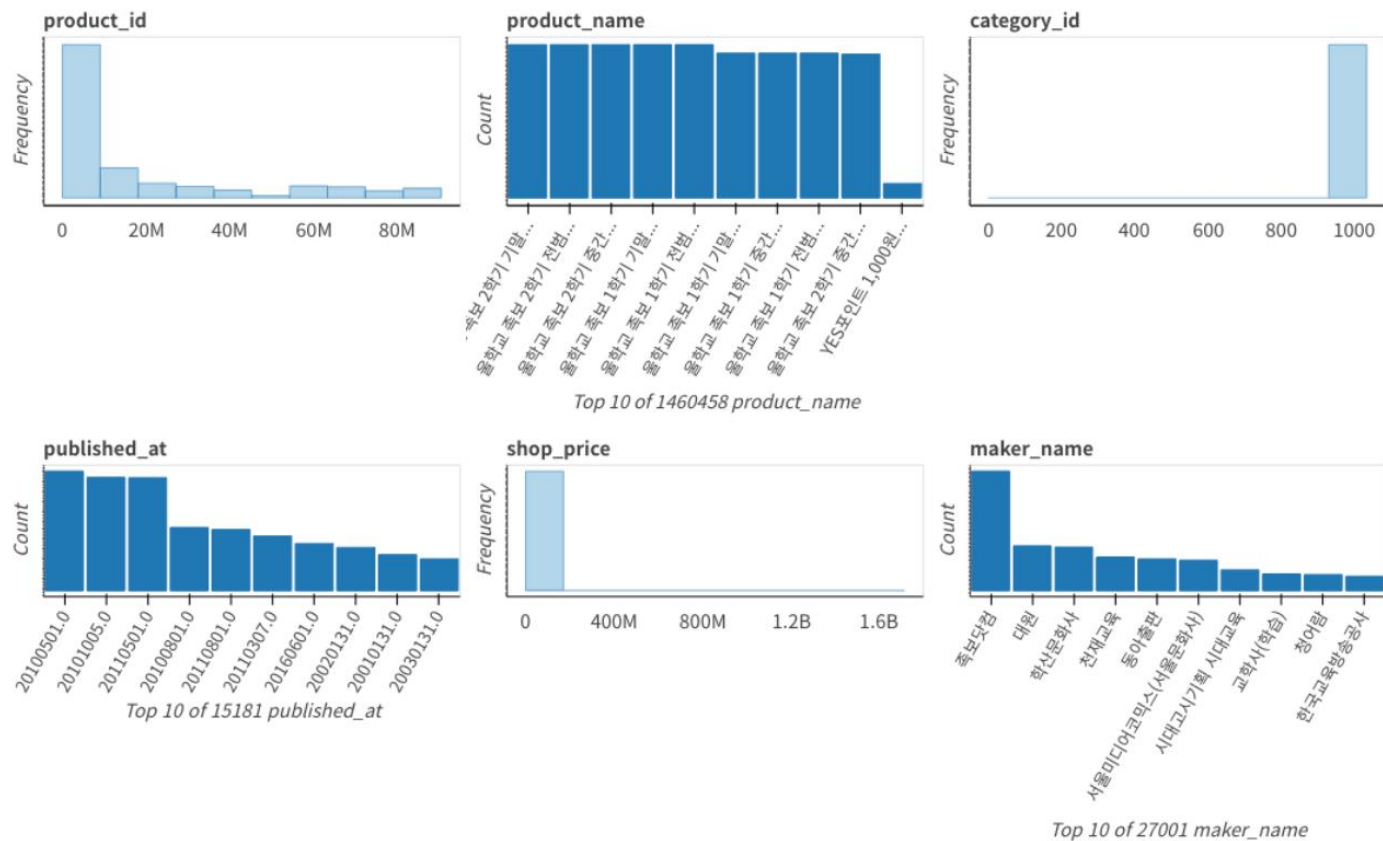
V. 부록

- 데이터 EDA
 - 1. Accounts



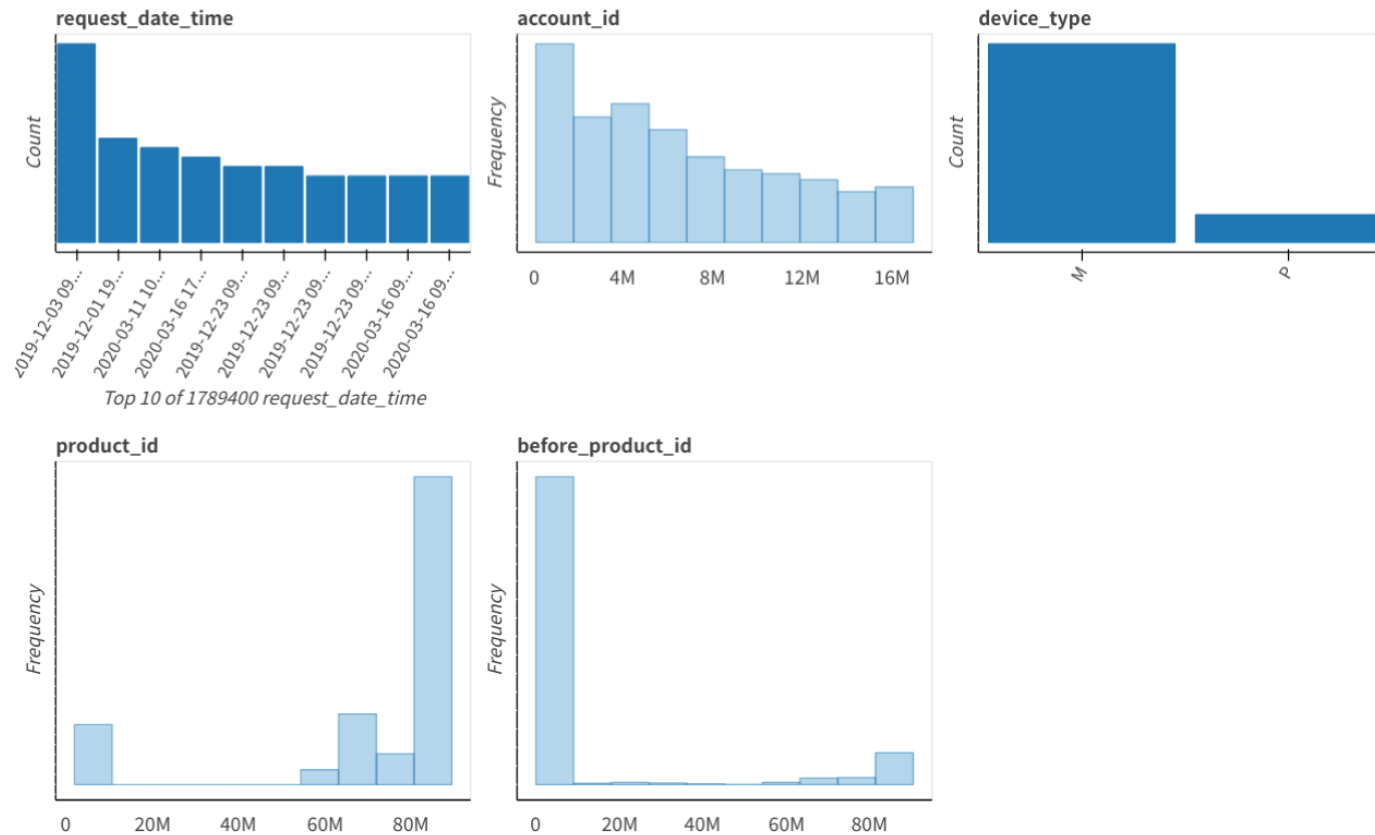
V. 부록

- 데이터 EDA
2. Products



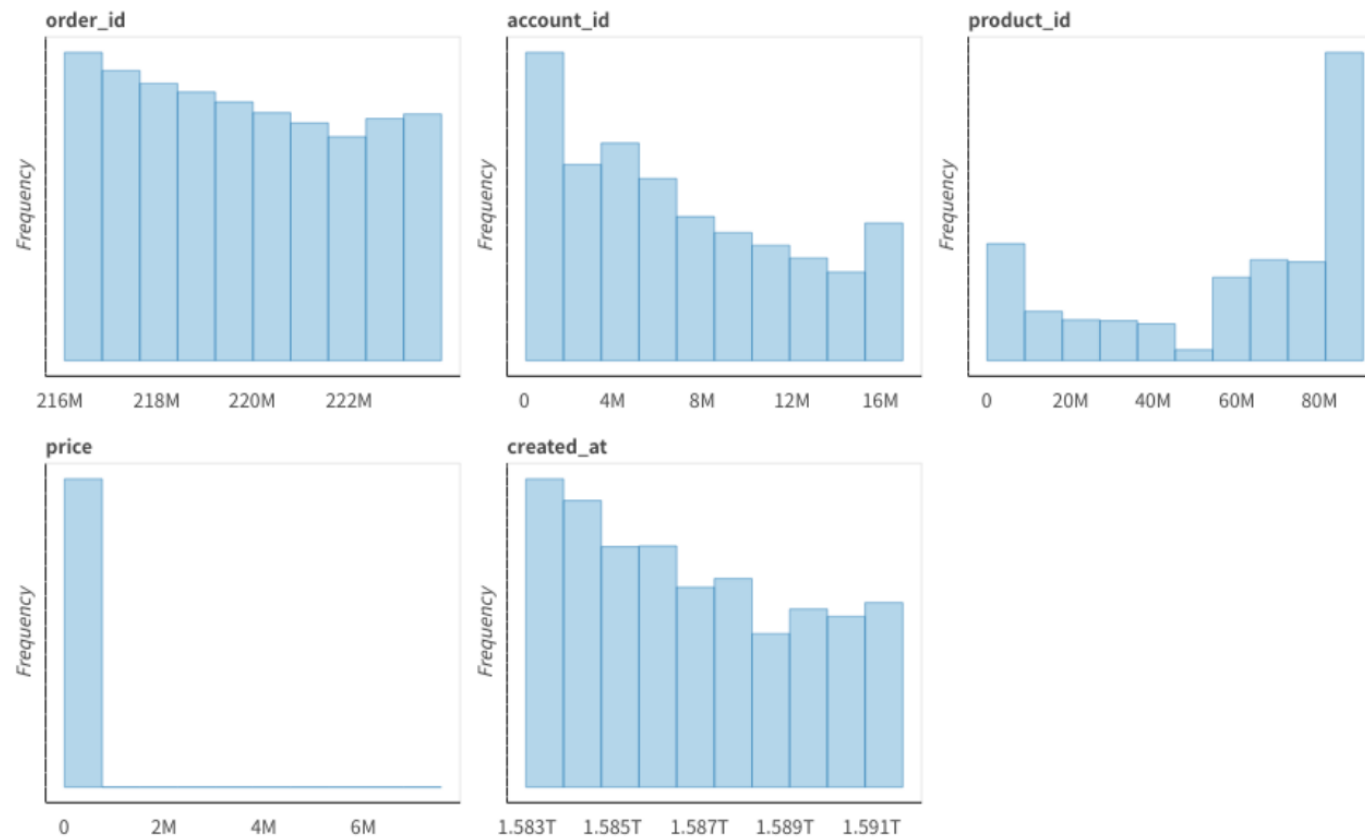
V. 부록

- 데이터 EDA
 - 3. Clicks



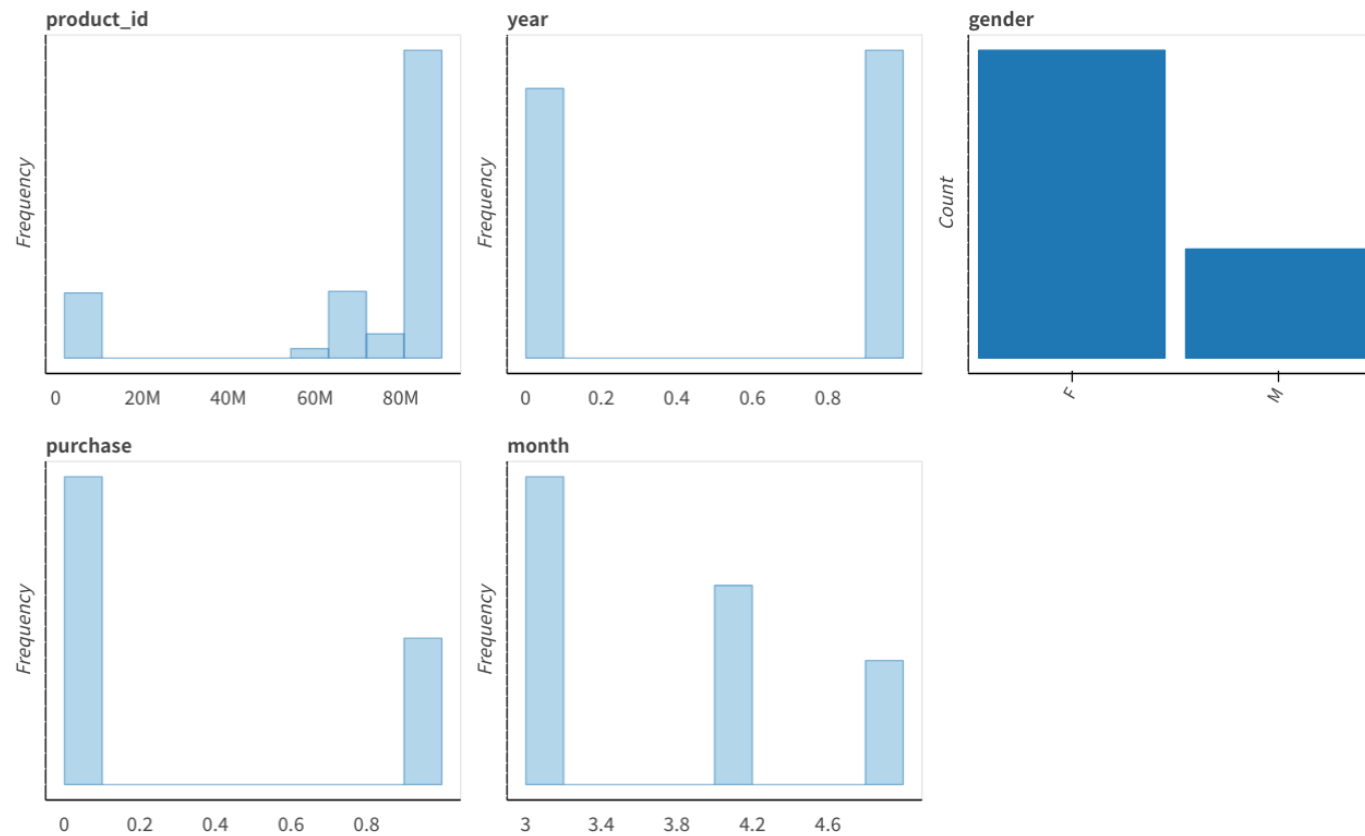
V. 부록

- 데이터 EDA
4. Orders



V. 부록

- 데이터 EDA
5. Book24



IV. 부록

- 알고리즘 구조
 - **Class** MAB(ABC) > **Class** LinThompson(MAB)
 - **def** off-policy evaluation (mab, arms, rewards, contexts, nrounds=None)
 - **def** evaluation(T, params)
- 모델의 기본 가정
 - Linear ($y^* = y$) [*All models are wrong but some are useful*]
 - Bayesian Inference를 따라 $f(\text{베타} \mid x, y)$ 는 $f(y \mid x, \text{베타}) * f(\text{베타})$ 에 비례함
 - 이 때 베타의 사전 분포는 $N(0,1)$ 를 따르며
 $y = x \cdot \text{베타} + (\text{noise})$, $\text{noise} \sim N(0,1)$ 을 가정할 때, $N(x \cdot \text{베타}, 1)$ 을 따른다.

IV. 부록

1. 사전 분포

- 표준 편차
`self.B = np.identity(ndims)`
`inv(self.B)`
- 평균
`self.mu_hat = np.zeros((ndims,1))`
- Y
`리워드 = {1,0,0,0,0,1..}`
- $X.T$
`arm_context (4,1) of (4,24)`
- X
`context[0][i:i+4] (1,4) * 24bandits`

IV. 부록

2. 사후 분포

- 표준 편차 (cov)

$$B = \text{self.B} + \text{arm_context} \cdot \text{arm_context.T}$$
$$\text{inv}(B)$$

- 평균(mean)

$$\text{mu_hat} = \text{inverse}(\text{self.B}) \cdot \text{self.f}$$

IV. 부록

3. Policy

- $T=t$ 라운드
- `Multivariate_normal(mean,cov)`에서 4가지 feature에 대한 베타(beta)를 Thompson sampling 방식을 통해 추출
- 추출한 베타 값과 `arm_context.T`를 내적
- arm 1~24의 reward 기댓값을 계산 후 `max(arm)`을 선택
- 이 과정을 반복하여 분포가 누적 계산됨

IV. 부록

4. Test

- Train(3,4월) – Test(5월)
- Train(iterations=1,000)을 통해 사후적으로 얻은 각 feature의 베타를 사용
- Test(iterationas=1,000) 정도에서 수렴하는 것을 확인
- 평가

Yes 24 추천에 대한 1 비율 (약 19%)

LinThompson 모델에 대한 average reward(약 30%)