



모델링 A조 발표

대출 의사결정 과정을 반영한 대출 신청 예측

강건우 김원 도승범 조찬형

목차

파트 1

데이터 설명

파트 2

전처리 과정

파트 3

EDA

파트 4

예측 모델

파트 5

예측 결과 및 해석

1. 데이터 설명

데이터 분석 전 개요 탐방 : 가진 데이터셋 어떤지

빅콘테스트 퓨처스 부문 데이터셋 :
암호화된 Finda 앱 사용자들의
log data, user data, loan data

Target value는 is_applied.
자세한 구성은 이후 지면 할애해 설명.

어떤 대출상품이 신청될 것인지
예측하는 task.

세 가지 데이터셋 자유롭게 사용해서 어떤 사용자가
어떤 대출 제안에 긍정적으로 반응할 지 예상.

데이터 특성 상 실제 데이터는 공개 불가

설명 시에는 인공 예시 데이터를 사용

방대한 양의 데이터

병합된 최종 데이터는
대략 10,000,000 * 30

1. 데이터 설명

1) User_log Dataset: 12,907,328*6

user_id	event	timestamp	date_cd
1715630	GetCreditInfo	44629.64088	#####
1715258	Login	44629.64109	#####
1715258	Login	44629.64117	#####
364608	OpenApp	44629.64648	#####
1511253	UseLoanManage	44629.64737	#####

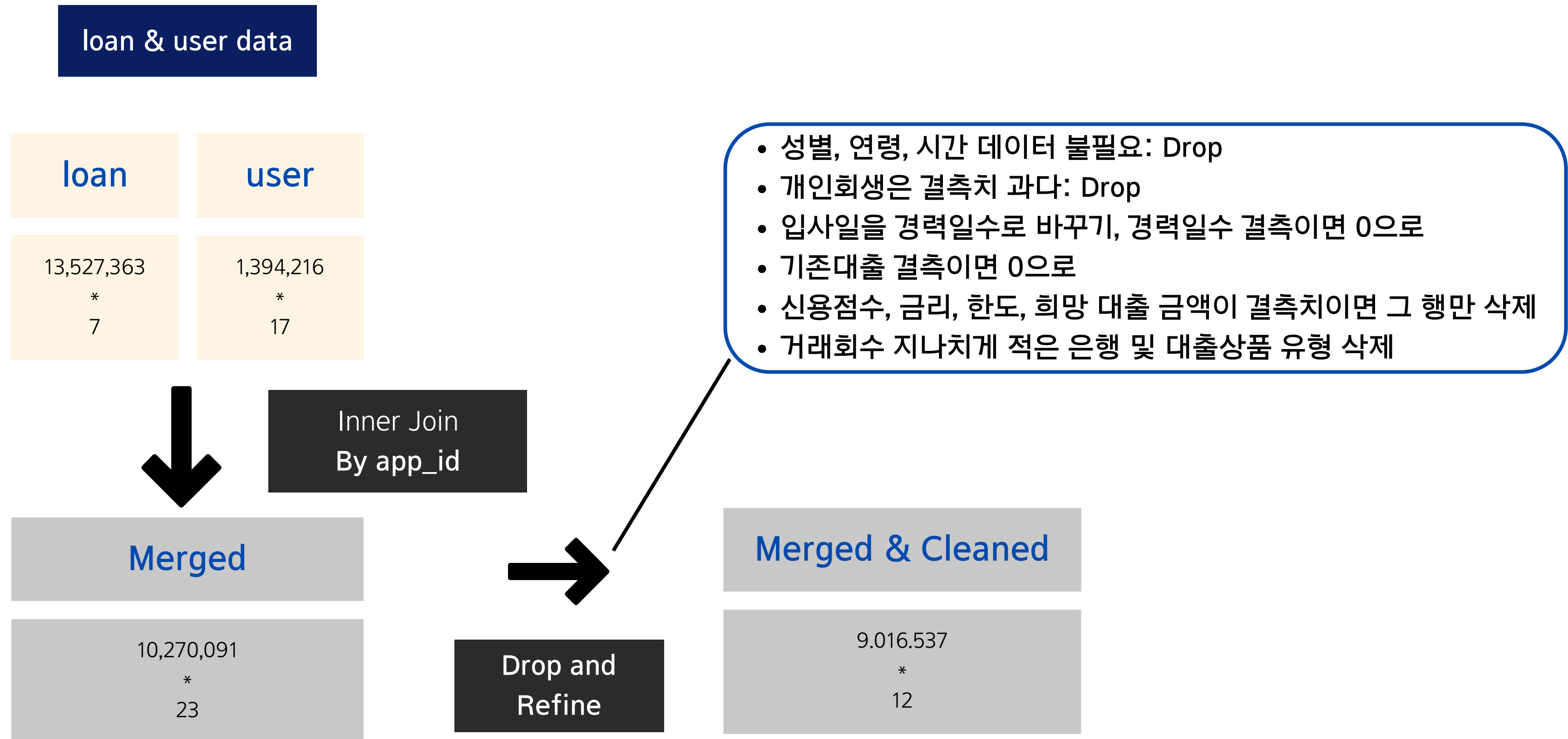
2) User_spec Dataset: 1,394,216*17

application_id	user_id	birth_year	gender	insert_time	credit_score	yearly_income	income_type	company_enter_month	employment_type	houseown_type	desired_amount	purpose
3863817	341318	1971	1	#####	710	30000000	EARNEDINCOME	201901	계약직			
3866840	524359	1976	0	#####	590	42000000	EARNEDINCOME	201807	정규직	자가	2000000	생활비
3869011	524359	1976	1	#####	930	36000000	EARNEDINCOME	201807	정규직	기타가족소유	5000000	생활비
3866350	733387	1999	1	#####	580	30000000	EARNEDINCOME2	202111	정규직	자가	10000000	생활비
3866526	801057	1993	0	#####	660	48000000	EARNEDINCOME	202104	정규직	전월세	5000000	생활비
										전월세	20000000	대환대출

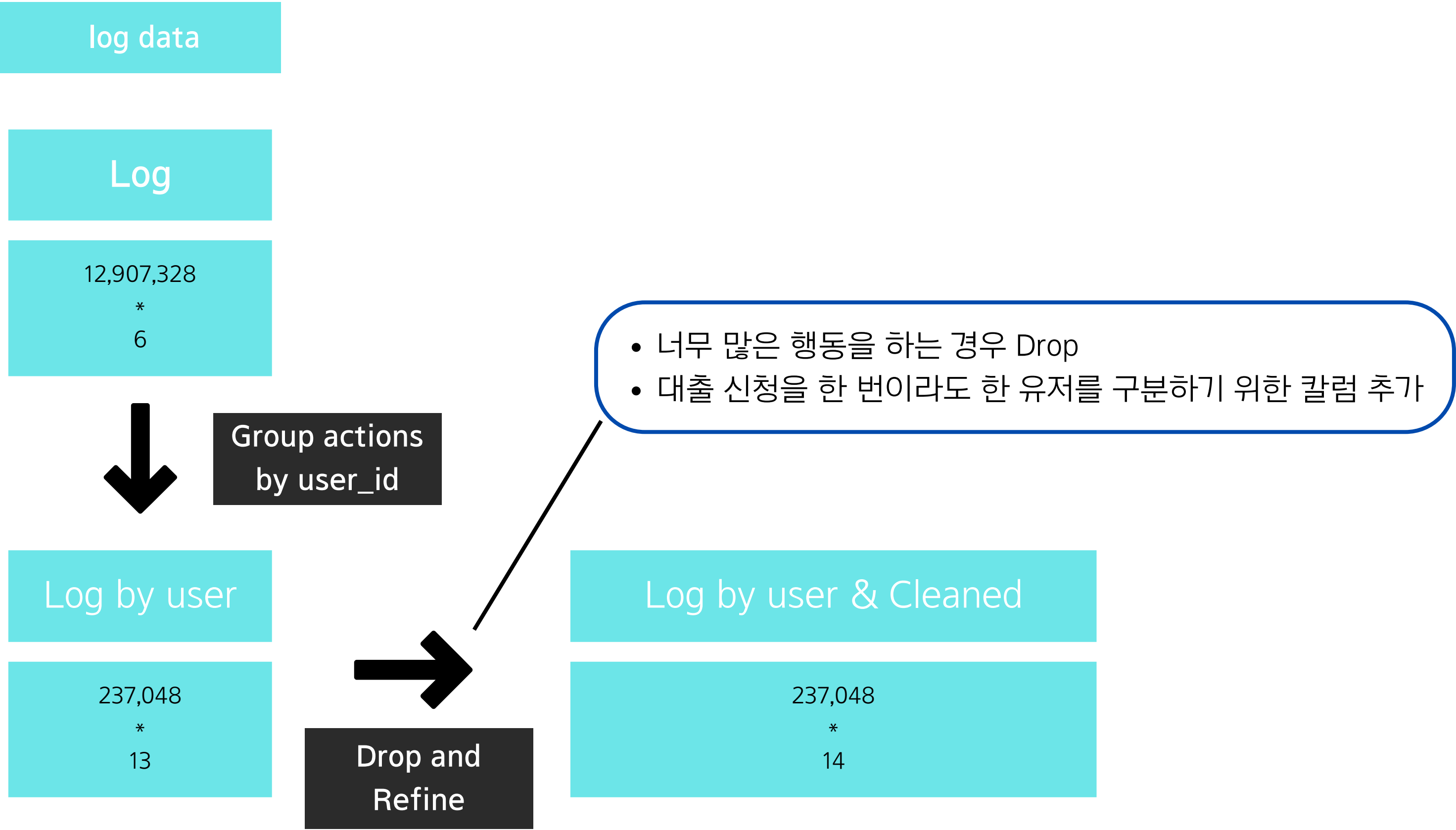
3) Loan_result Dataset: 13,527,363*7

application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied
5072400	2022-06-21 8:31	27	911001	15000000	7.9	0
5072400	2022-06-21 8:31	41	906002	45000000	12.7	0
5072400	2022-06-21 8:31	50	932001	43900000	13.3	0
5072400	2022-06-21 8:31	499	927001	43900000	12.4	0
5072400	2022-06-21 8:31	30	942001	17000000	11.4	0

2. 데이터 전처리



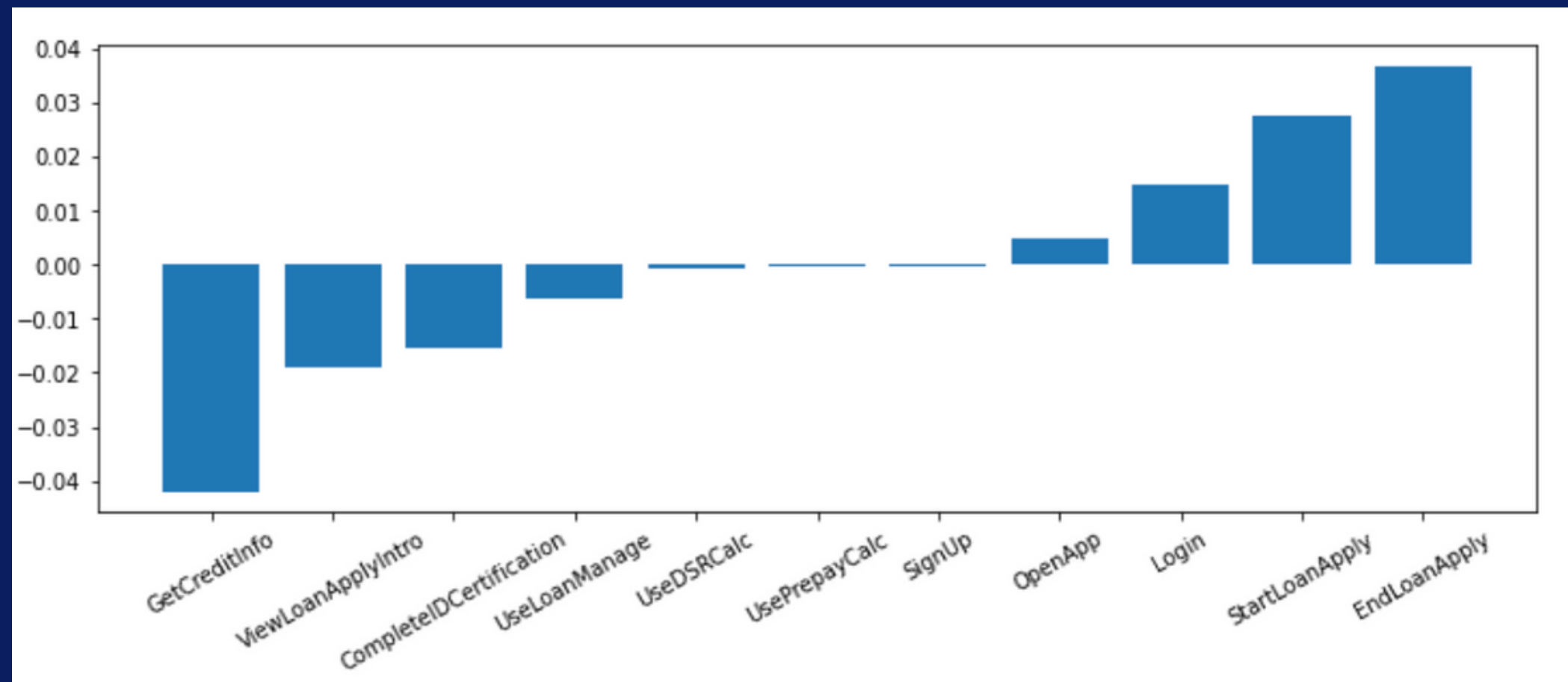
2. 데이터 전처리



3. EDA

1) log data 분석

대출 신청 기록이 있는 유저와 그렇지 않은 유저 집단의 평균적 행동 수 차이(%)



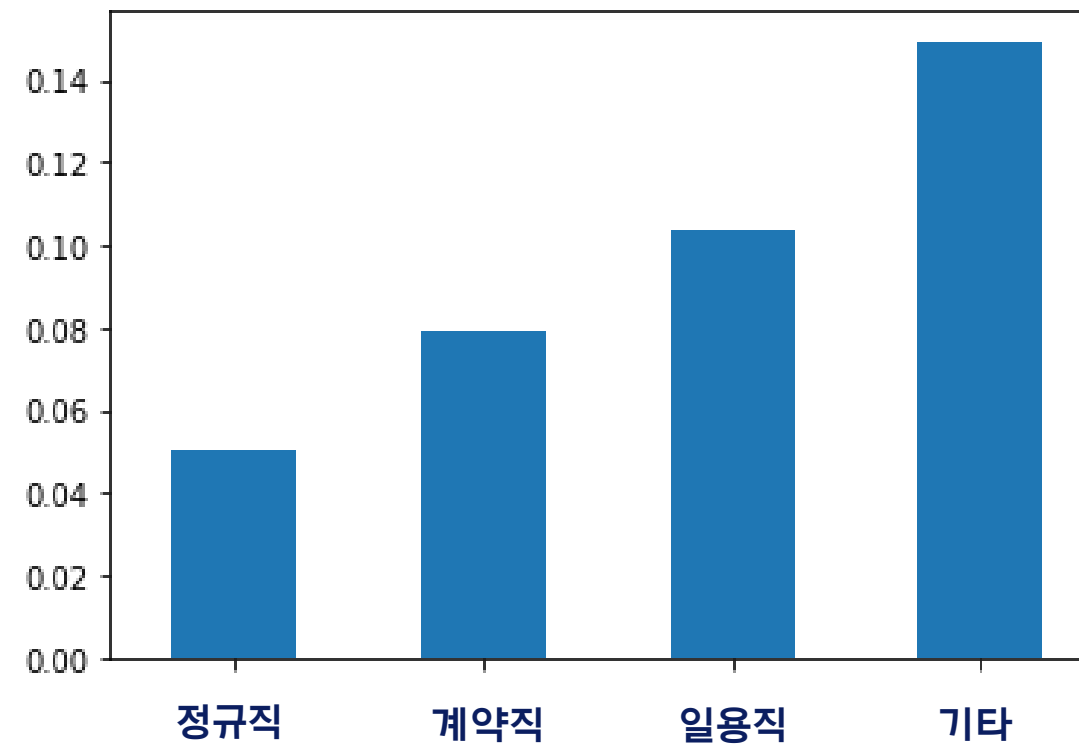
대출 신청 기록이 있는 유저는 대출상품조회, 상품조회신청, 로그인 등 반복적으로 대출상품목록을 조회하는 행동을 보이는 경향

3. EDA

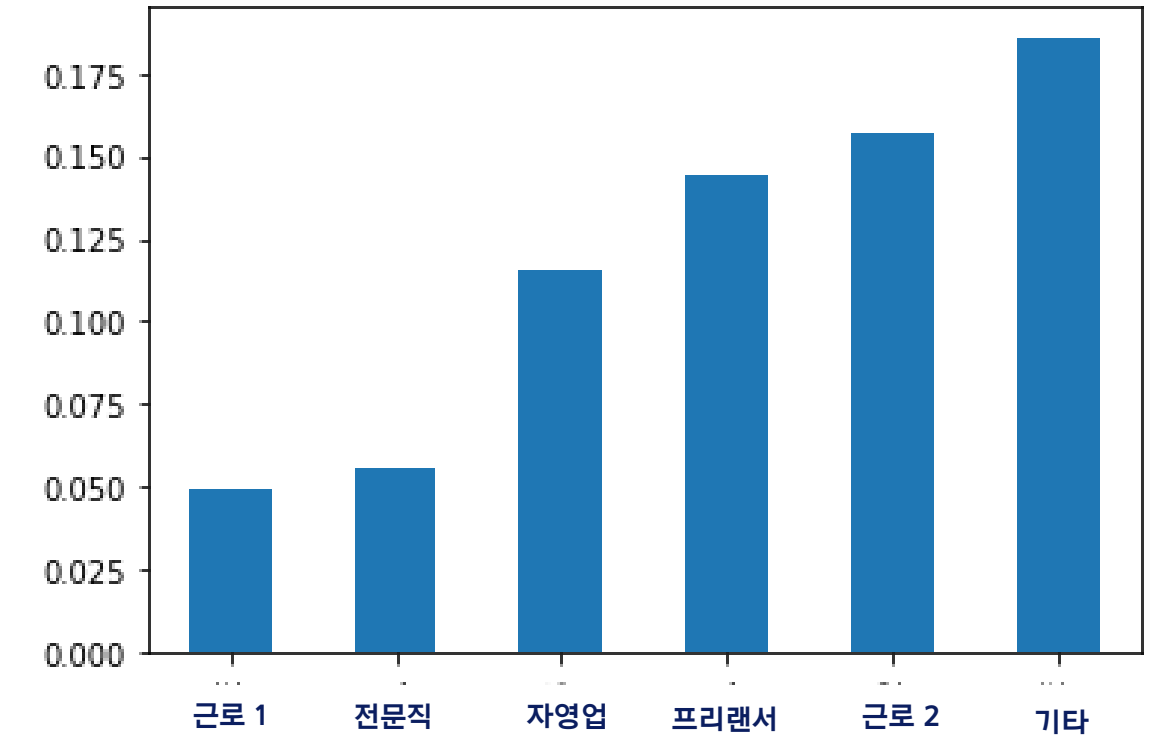
2) user 데이터 분석

- 고용형태 및 소득형태가 불안정할수록 대출신청률 높음
- 전월세 거주자가 자가거주자보다 대출신청률 높음
- 생활비 및 사업자금, 대환대출 목적일 경우 주택 관련 대출보다 신청률 높음

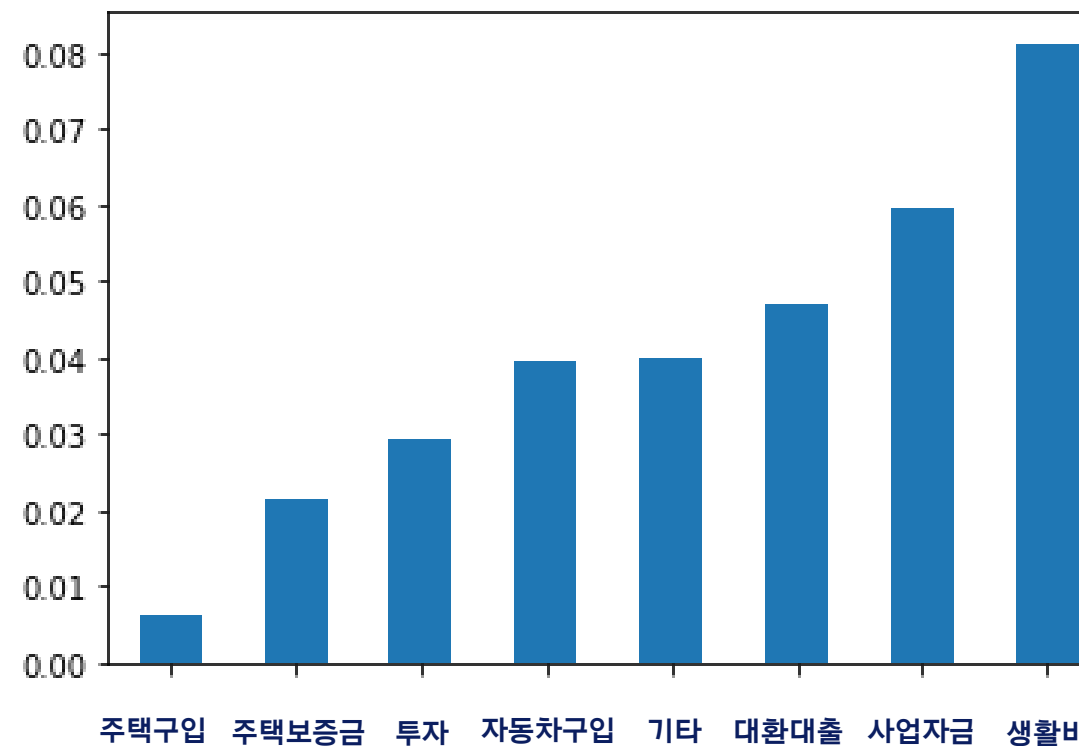
고용형태별 대출신청률



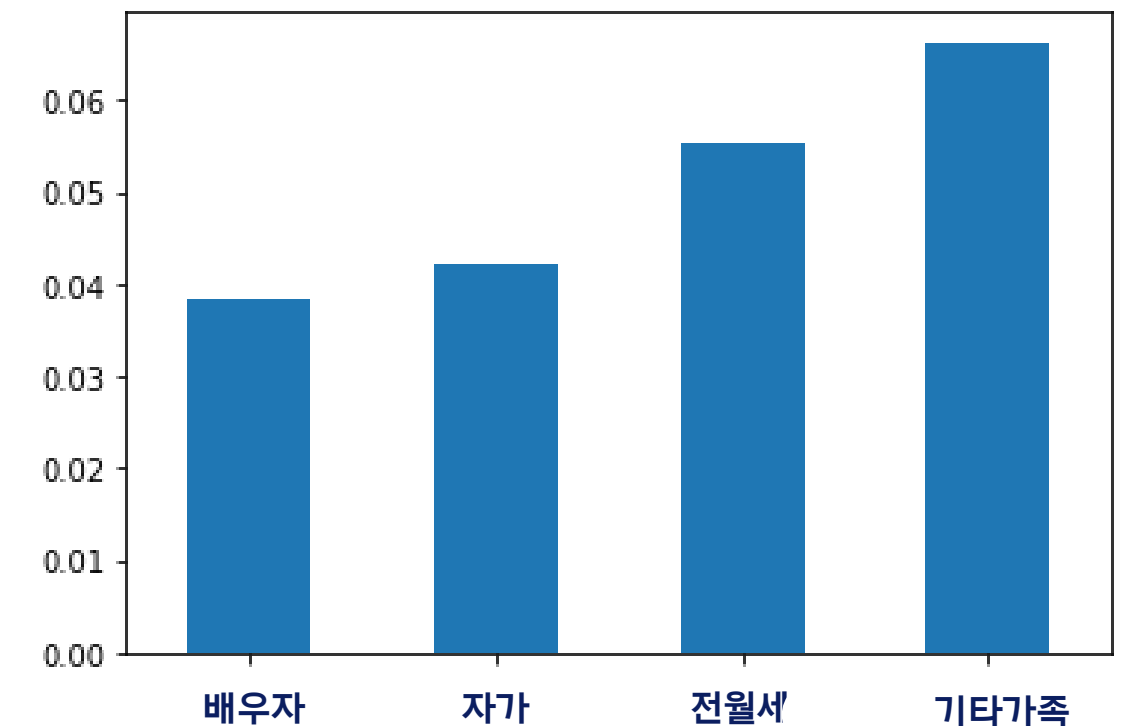
소득형태별 대출신청률



대출목적별 대출신청률



주거형태별 대출신청률

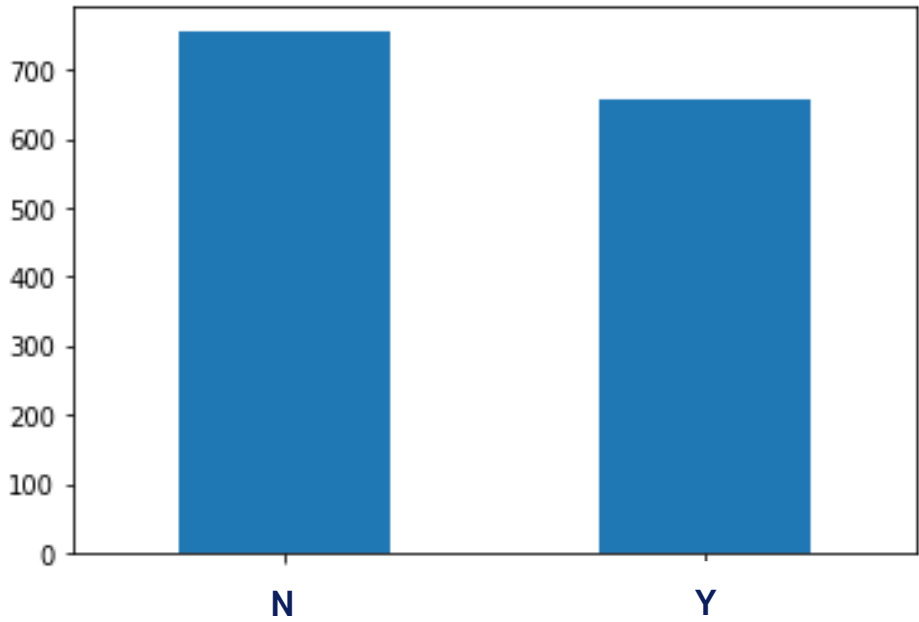


3. EDA

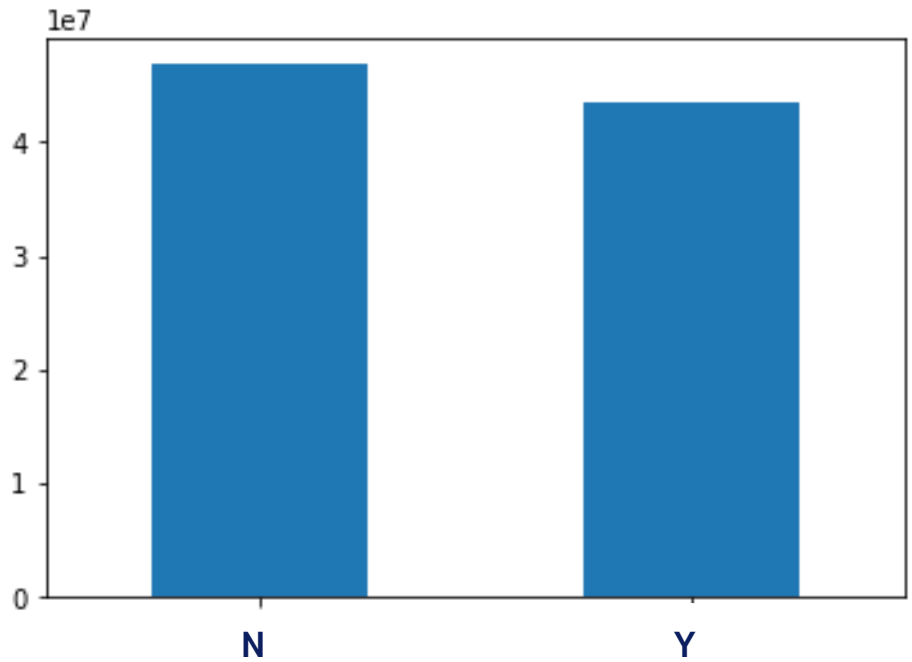
2) user 데이터 분석

- 대출을 신청한 집단의 평균 신용점수가 더 낮음
- 대출을 신청한 집단의 평균 소득이 더 낮음
- 대출을 신청한 집단의 평균적인 대출 희망한도가 더 낮음

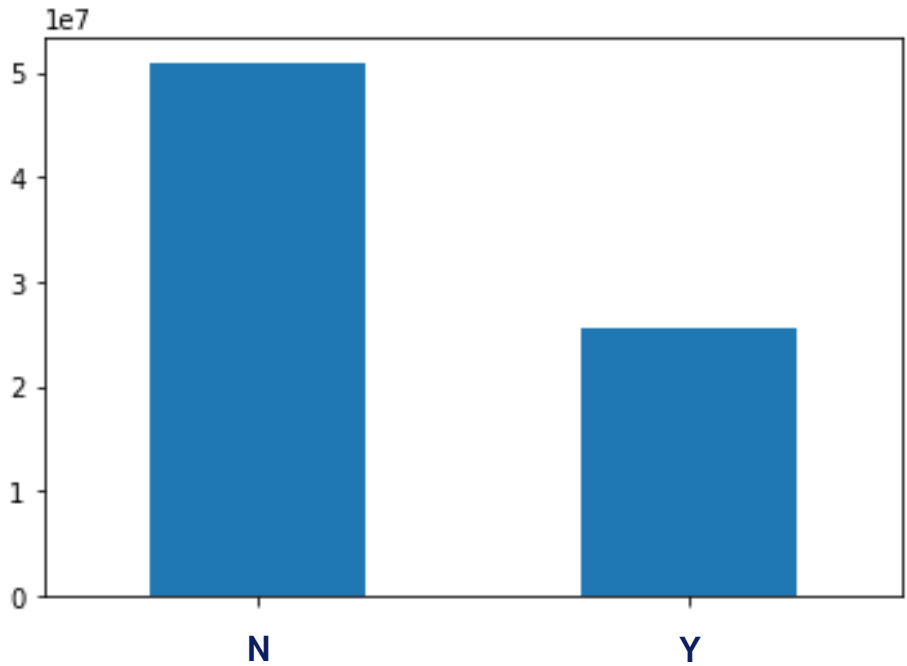
대출 신청 여부에 따른 평균 신용점수



대출 신청 여부에 따른 평균 소득



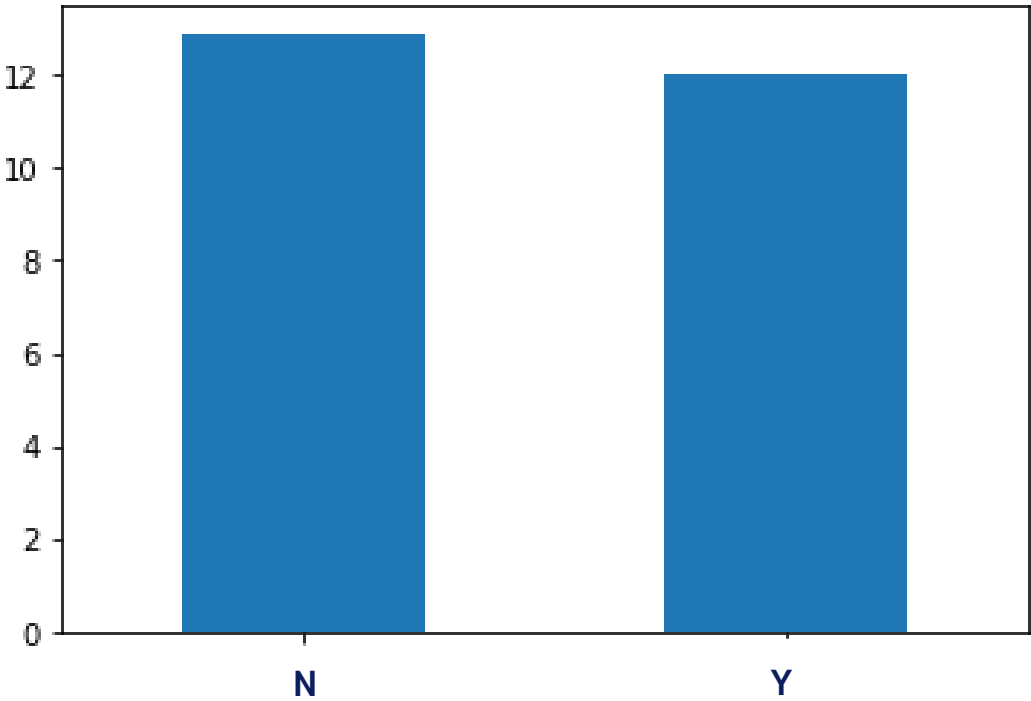
대출 신청 여부에 따른 평균 희망한도



3. EDA

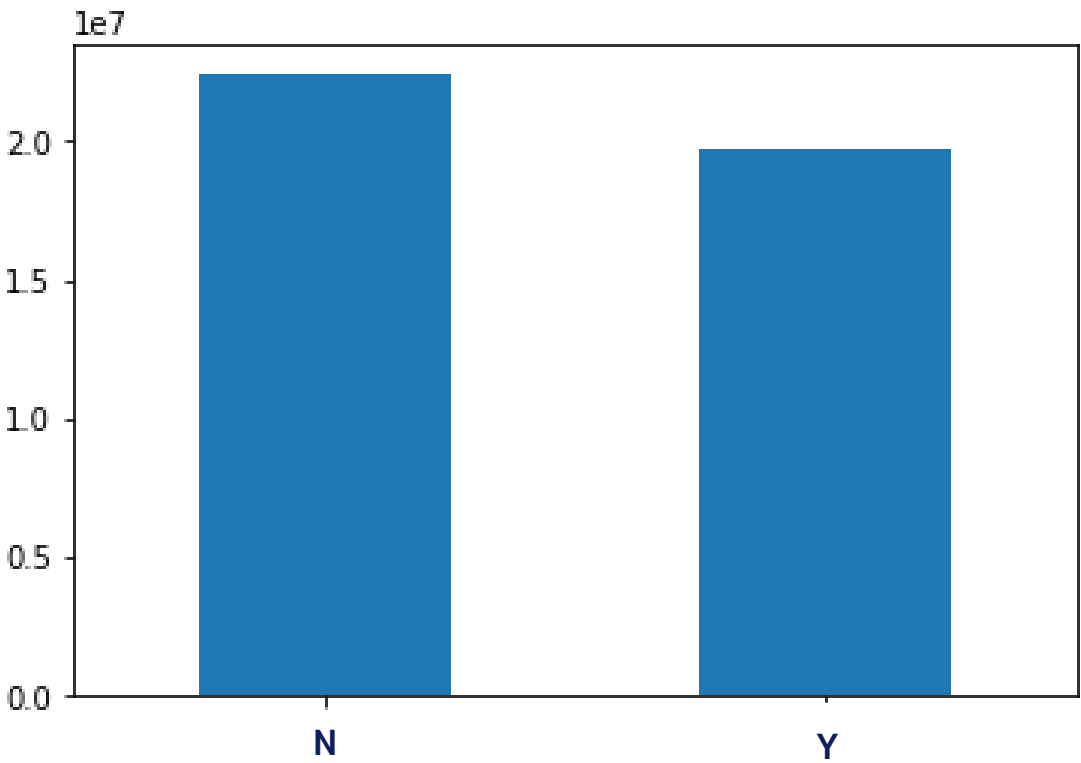
3) loan data 분석

대출 이자율



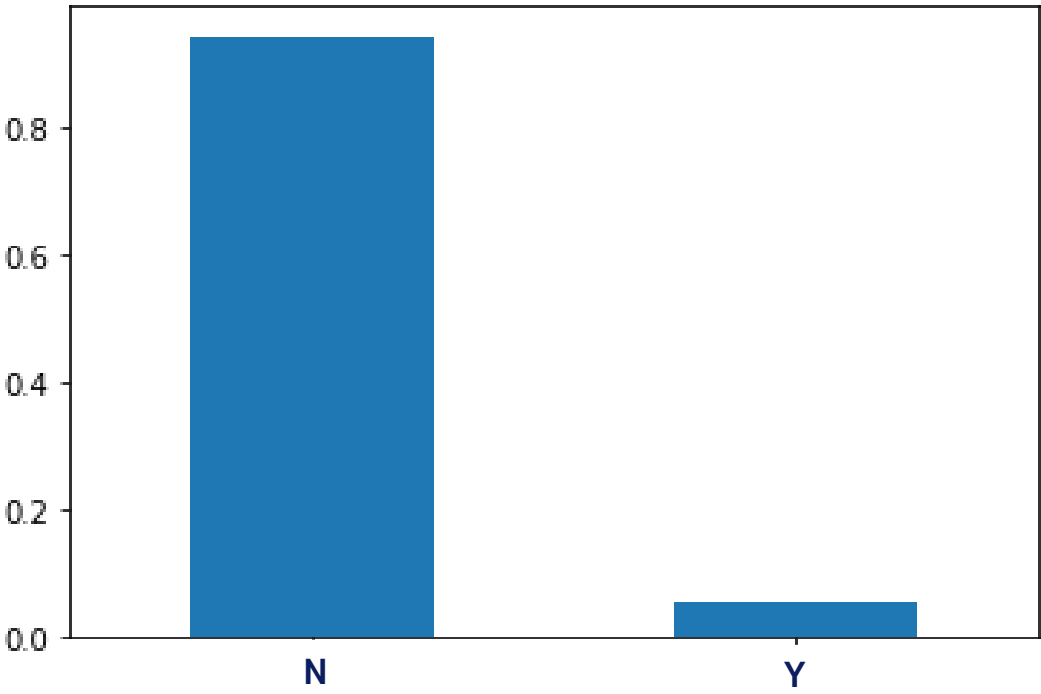
- 대출이 신청된 상품의 평균적인 금리가 더 낮음

대출 한도 분포



- 대출이 신청된 상품의 평균적인 한도가 더 낮음

대출신청률 분포

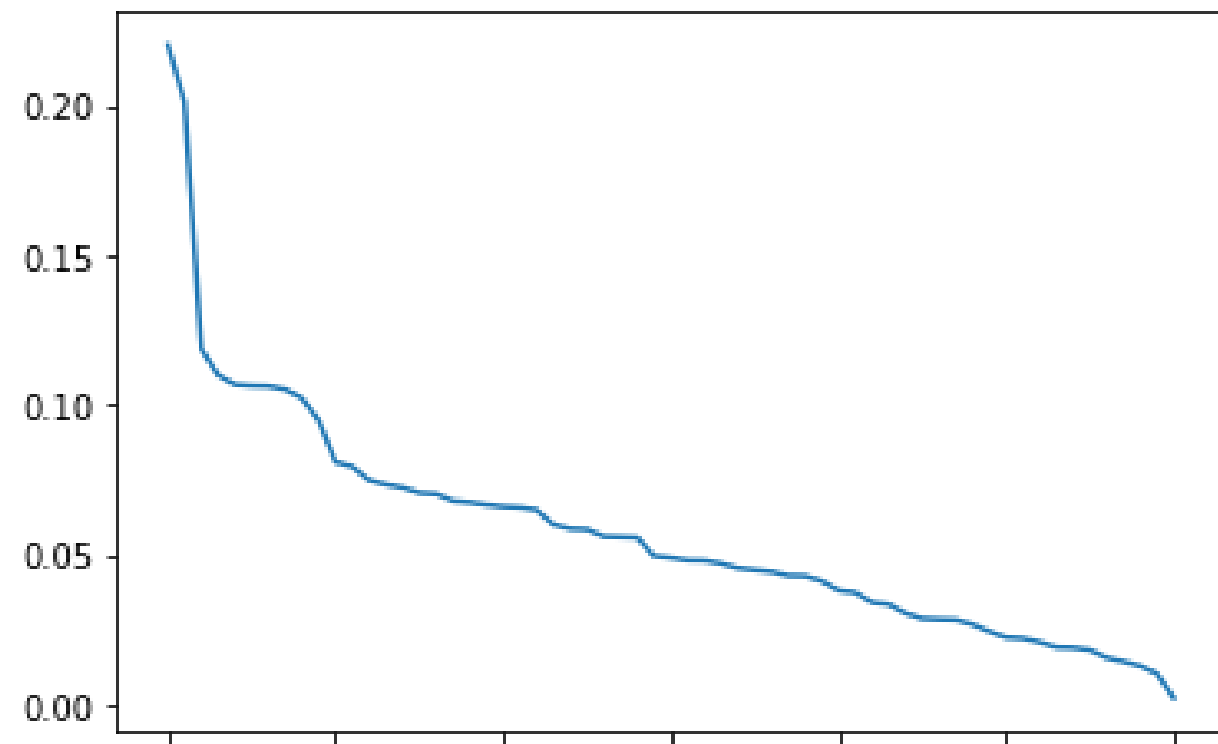


- 대출이 신청될 확률은 매우 불균형한 분포
- 대출신청률은 전체에서 5%

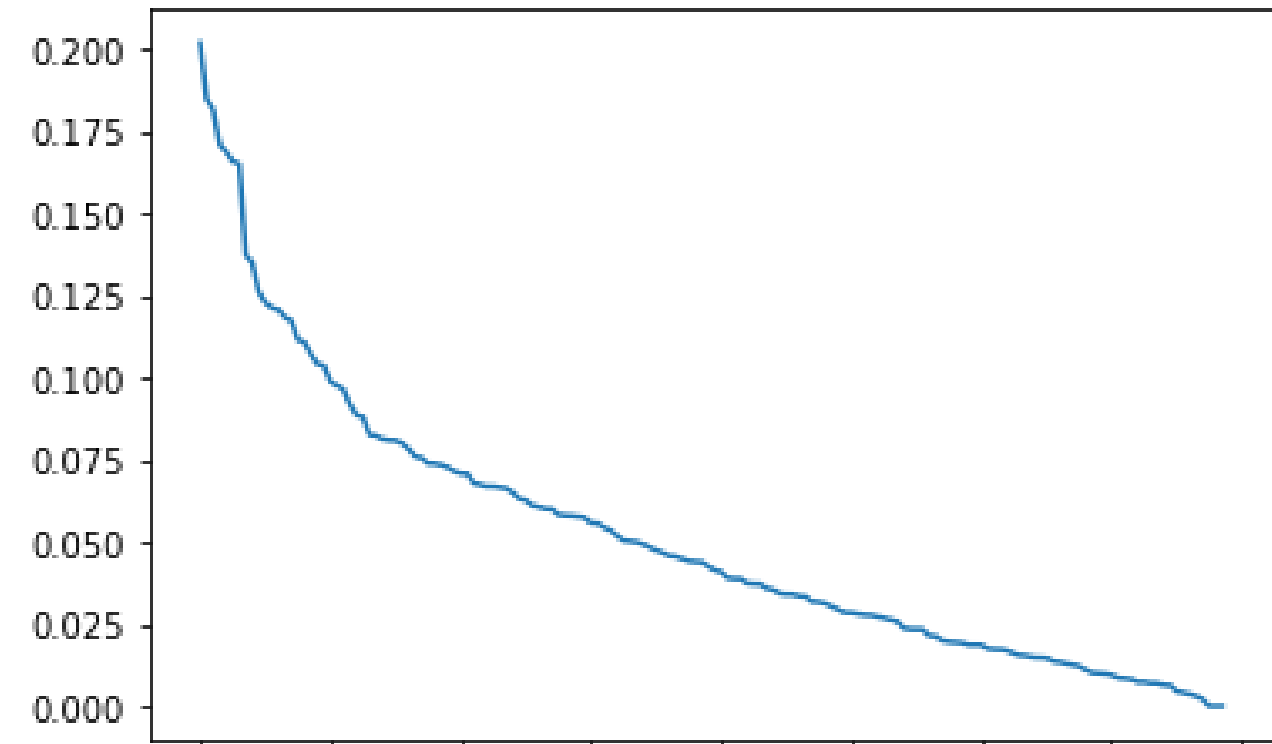
3. EDA

3) loan data 분석

은행별 대출신청률(내림차순)



상품유형별 대출신청률(내림차순)



- 유저들의 선호도는 특정한 은행 및 대출상품 유형으로 쏠려있는 것을 확인할 수 있음

3. EDA

4) EDA 결과 종합

EDA 결과 요약

Log data

특정한 행동들은 target 변수와 상관관계를 보임

- 대출상품조회 등을 반복적으로 실행한 유저는 대출 신청으로 이어질 확률 높음

Loan data

대출상품의 조건들은 target 변수에 유의미한 영향력을 미침

- 금리가 낮을수록, 한도가 낮을수록 신청될 확률 높음
- 특정한 은행 및 상품 유형이 유저들에게 인기 많음

User data

대출을 신청한 유저의 spec이 target 변수와 상관관계를 보임

- 연 수입 낮을수록, 소득 형태 불안정할수록, 신용점수 낮을수록, 신청확률 높음
- 생활비 대출, 대환대출의 경우 신청확률 높음
- 주택을 보유하지 않은 유저들의 대출 신청확률 높음

모델 반영?

- EDA를 통해 유의한 상관관계를 보인 특성들을 모델에 투입하자

3. EDA

4) EDA 결과 종합

EDA 결과 요약

target value의 불균형한 분포

- 전체 샘플에서 대출을 신청한 경우는 5%
- 대출을 신청하는 경우는 전체 샘플에서는 예외적인 경우

모델 반영?

- 대출을 신청하는 경우는 전체 샘플 차원에서 무언가 다른 패턴을 보이지 않을까?
- 이상치 탐지 모델로 접근해보자

4. 예측 모델

1) 이상치 탐지 모델: Auto encoder

Credit Card Fraud Detection

Data

Code (3758)

Discussion (104)

Metadata

▲

9438

New Notebook

Download (69 MB)

🏆

🔍

Search notebooks

⌵ Filters


All

Your Work

Shared With You

Bookmarks

Most Votes



Credit Fraud || Dealing with Imbalanced Datasets

Updated 3Y ago


623 comments · Credit Card Fraud Detection

▲

4175

🏆 Gold

...



In depth skewed data classif. (93% recall acc now)

Updated 6Y ago


124 comments · Credit Card Fraud Detection

▲

759

🏆 Gold

...



Outlier!!! The Silent Killer

Updated 1y ago


105 comments · Titanic - Machine Learning from Disaster +16

▲

695

🏆 Gold

...



Semi Supervised Classification using AutoEncoders

Updated 4Y ago


67 comments · Titanic - Machine Learning from Disaster +1

▲

398

🏆 Gold

...



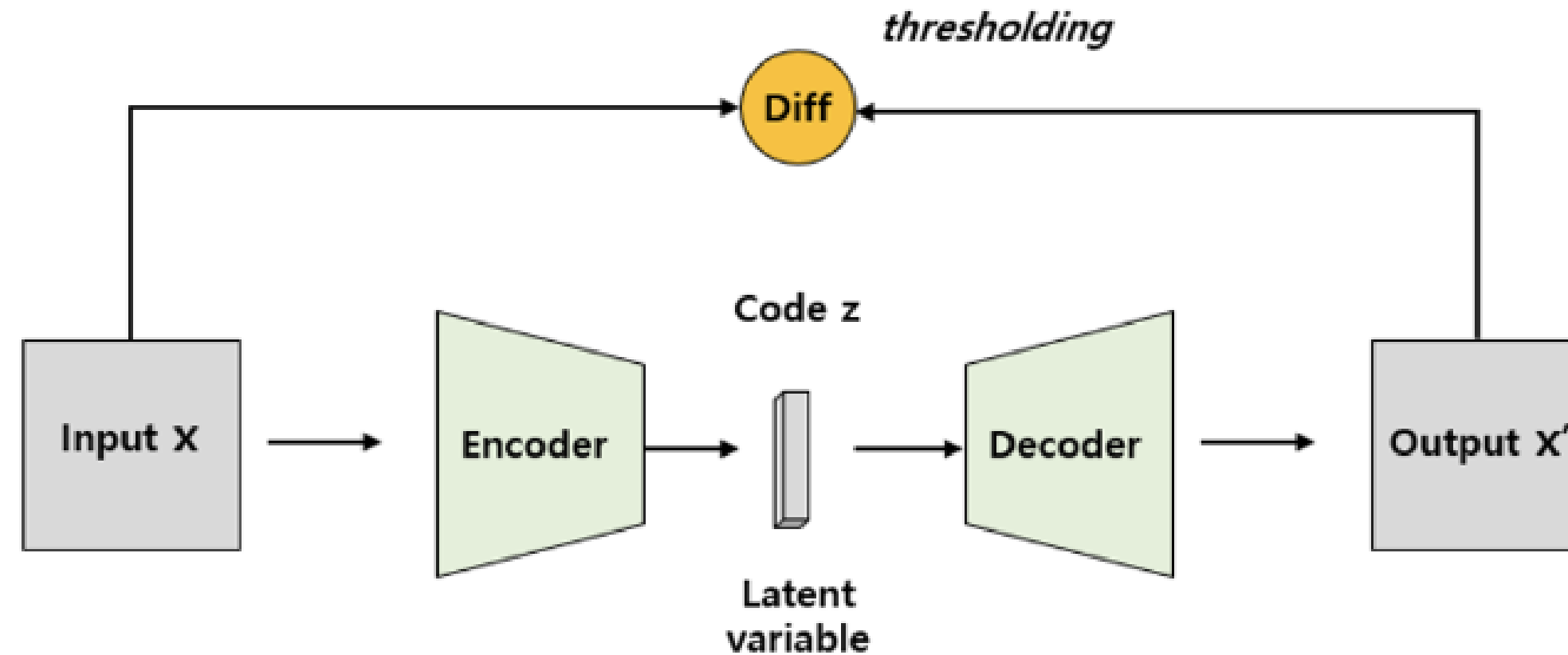
Automatic EDA Libraries 📊 Comparisson

▲

333

4. 예측 모델

1) 이상치 탐지 모델: Auto encoder



- Deep Neural Network와 PCA를 결합한 모델
 - input 벡터를 작은 차원으로 축소 후(Encoding)
 - 다시 원 차원으로 복원(Decoding)해 output 벡터 반환
 - input과 output의 오차를 DNN으로 줄여나가는 모델
 - 정보 소실은 적으면서 효과적인 차원 축소를 유도

- 모델 학습과정에서 데이터의 가장 핵심적인 패턴들만을 추출하기 때문에, output 벡터는 데이터의 주요한 특징들을 반영하고 noise들은 제거된 상태
- 만약 input 벡터와 output 벡터의 차이가 크다면, input 벡터에 noise가 많이 포함되어 있다는 것, 즉 이상치

4. 예측 모델

1) 이상치 탐지 모델: Auto encoder

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	6400
batch_normalization (Batch Normalization)	(None, 256)	1024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
batch_normalization_2 (Batch Normalization)	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Total params: 141,313
Trainable params: 139,777
Non-trainable params: 1,536

confusion_matrix	confusion_matrix_ideal
[0.8973, 0.0496] [0.0499, 0.0031]	[0.94431, 0.] [0. , 0.05569]

accuracy	precision	recall	f1_score
0.90040	0.05882	0.05849	0.05866

4. 예측 모델

2) 왜 결과가 좋지 않았을까?

(1) 사람들은 자신이 제시 받은 대출 조건 내에서 '만' 의사결정을 내린다

신용 상태가 사람들마다 다르기 때문에 각자 제시 받은 금리는 다르다

- 어떤 사람에게는 10% 금리가 낮다고 생각하지만 어떤 사람은 5% 금리도 높다고 느낌

그러나 모델은 이런 사실을 알지 못한다

- 어떤 경우 금리가 높는데 대출상품이 선택되고 어떤 경우 낮은데도 거절되는 것을 이해할 수 없음

-> 대출조건을 표준화하되, 각 유저가 제시 받은 대출조건 집합 내에서만 표준화하자

ex) i번째 사람이 받은 j번째 대출 상품의 표준화 금리

$$= \frac{(j\text{번째 대출 상품의 금리} - i\text{번째 사람이 제시 받은 금리 집합의 평균})}{(i\text{번째 사람이 제시 받은 금리 집합의 표준편차})}$$

-> 표준화된 금리의 값은 낮을수록 대출 확률이 높아지며
모델이 금방 소화해낼 수 있는 단순한 패턴이 됨

4. 예측 모델

2) 왜 결과가 좋지 않았을까?

(2) 애초에 대출을 받지 않으려는 사람들도 있다

대출을 받을 생각으로 어플을 실행한 유저도 있지만 시험 삼아 또는 장난 삼아 해보는 유저들도 있음
모델은 표준화된 조건이 이렇게 유리한데도 신청되지 않는 경우를 이해할 수 없음

-> 대출 상품이 선택될 확률을 두 가지 요소로 분해하자

$P(\{\text{대출상품 } j \text{가 신청됨}\})$

$= P(\{\text{대출상품 } j \text{가 신청됨}\} \cap \{\text{유저 } i \text{가 대출을 받고자 함}\})$

$= P(\{\text{유저 } i \text{가 대출을 받고자 함}\}) * P(\{\text{대출상품 } j \text{가 신청됨}\} | \{\text{유저 } i \text{가 대출을 받고자 함}\})$

두 확률에 대하여 최적화된 모델을 적용하여 추정하고, 추정된 두 확률을 곱하자

-> 대출을 받고자 하는 유저들로만 한정하면, 표준화된 대출조건이 유리하면 무조건 확률이 높아지므로
모델이 쉽게 소화해낼 수 있는 단순한 패턴이 됨

4. 예측 모델

3) 두 확률 모델

target value

$$= P(\{ \text{대출상품 } j \text{가 신청됨} \})$$

$$= P(\{ \text{대출상품 } j \text{가 신청됨} \} \cap \{ \text{유저 } i \text{가 대출을 받고자 함} \})$$

$$= \underbrace{P(\{ \text{유저 } i \text{가 대출을 받고자 함} \})}_{\text{Decision Tree}} * \underbrace{P(\{ \text{대출상품 } j \text{가 신청됨} \} \mid \{ \text{유저 } i \text{가 대출을 받고자 함} \})}_{\text{Decision Tree}}$$

Decision Tree

- log features
- user features

Decision Tree

- 표준화된 loan features
- 은행 및 유형별 선호도(train set)

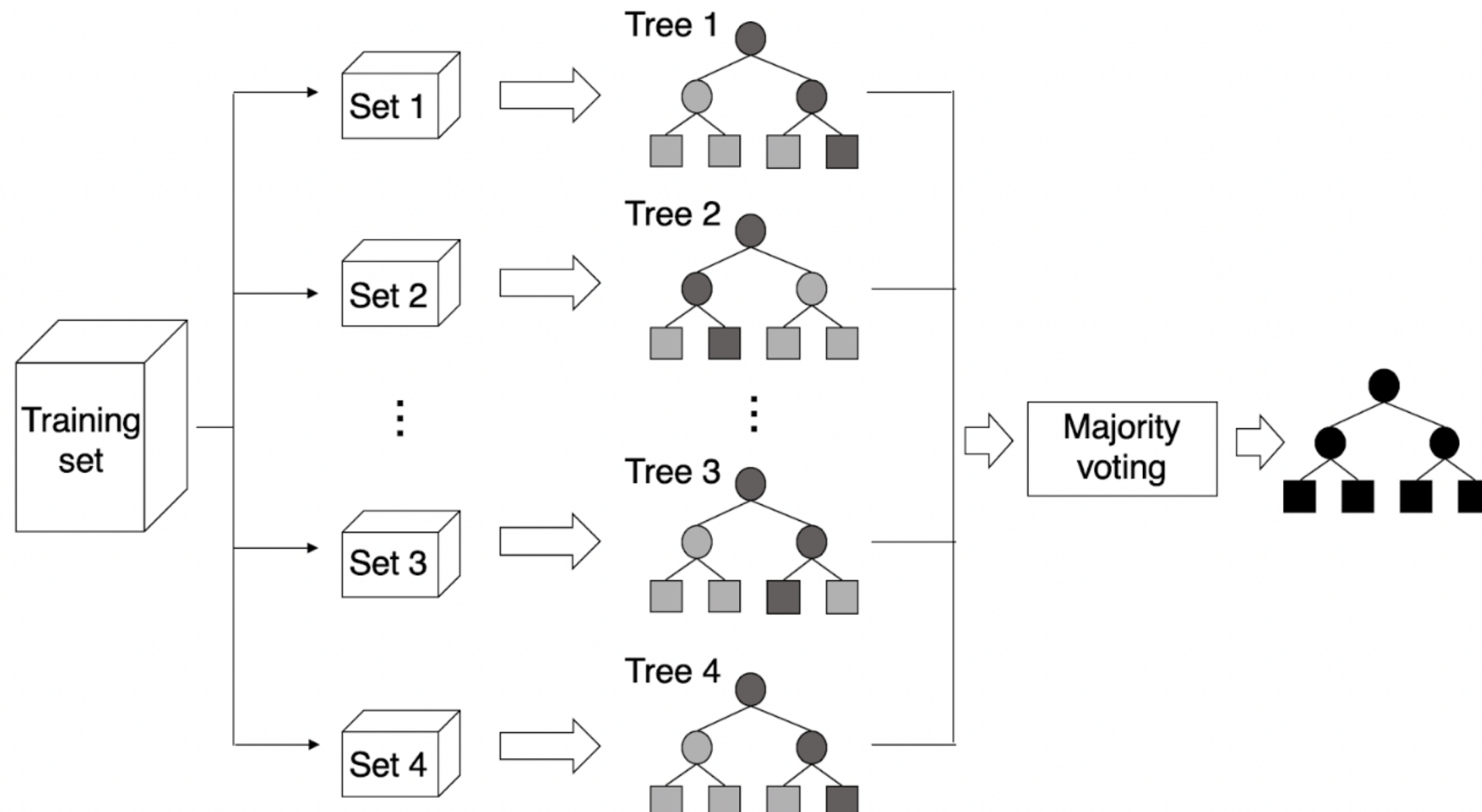
어떤 Decision Tree model?

4. 예측 모델

3) 두 확률 모델

- 어떤 분류기를 쓸까

Random Forest Model



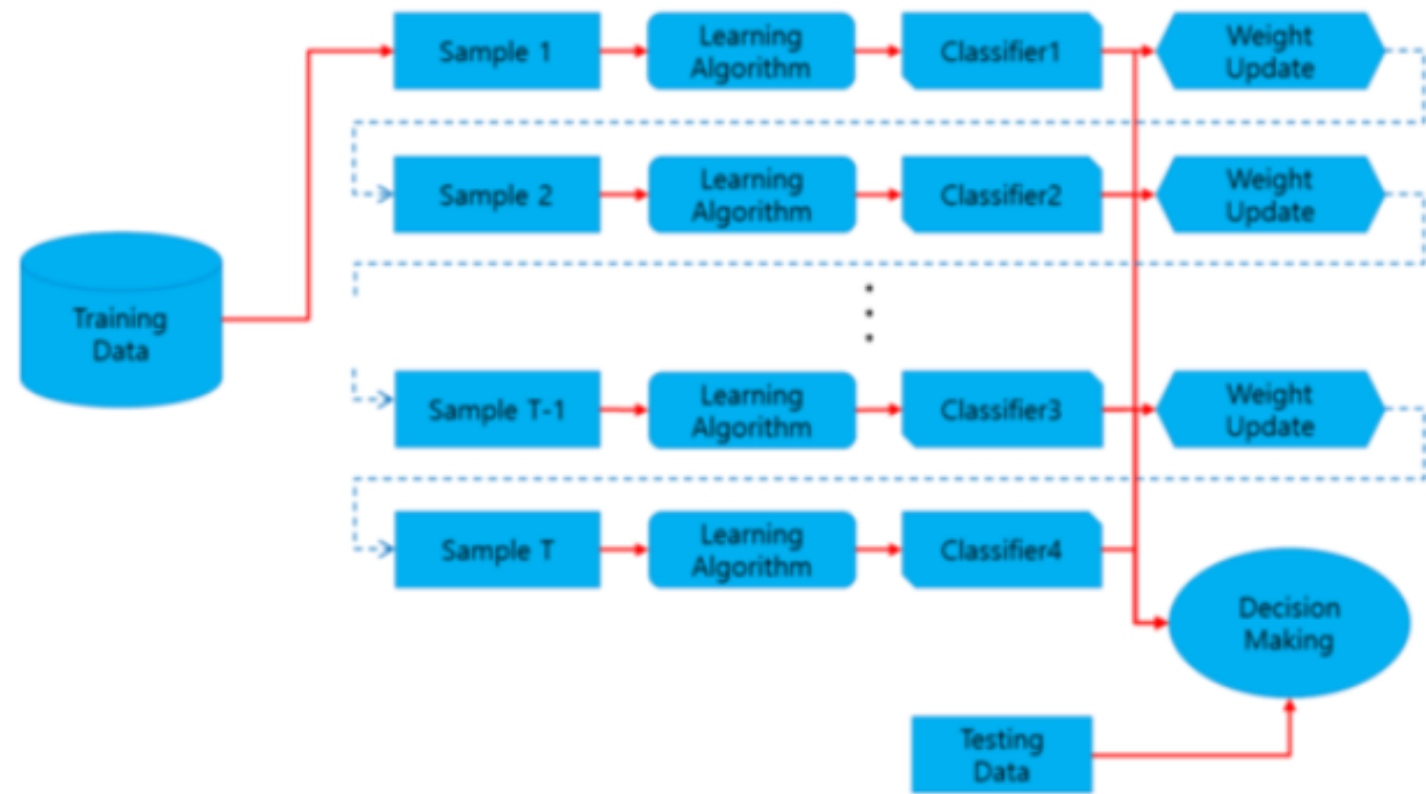
- 여러 개의 의사결정 트리
 - 샘플의 무작위 추출
 - 특성변수도 무작위로 추출한다
- 여러 의사결정 트리의 결과를 평균하여 최종 예측값 제시
- 데이터 크기 및 특성의 차원이 큰 경우에도 과대적합 문제로부터 자유로움

4. 예측 모델

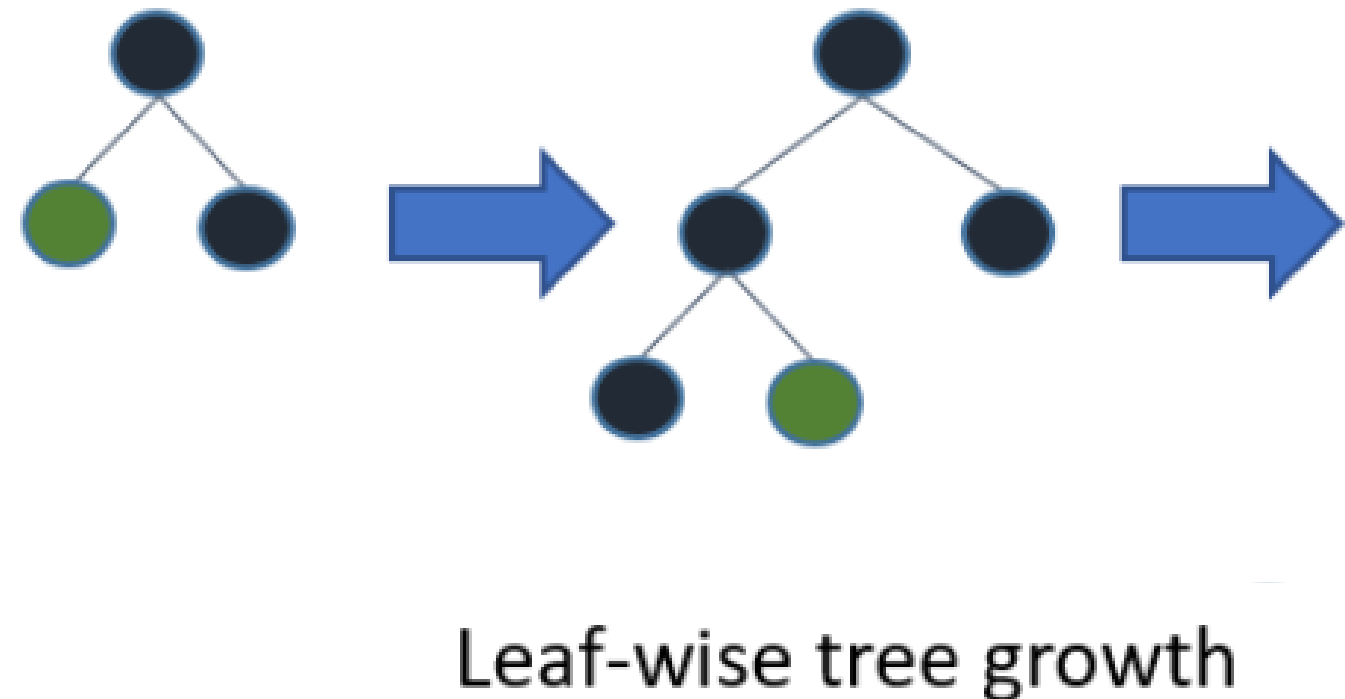
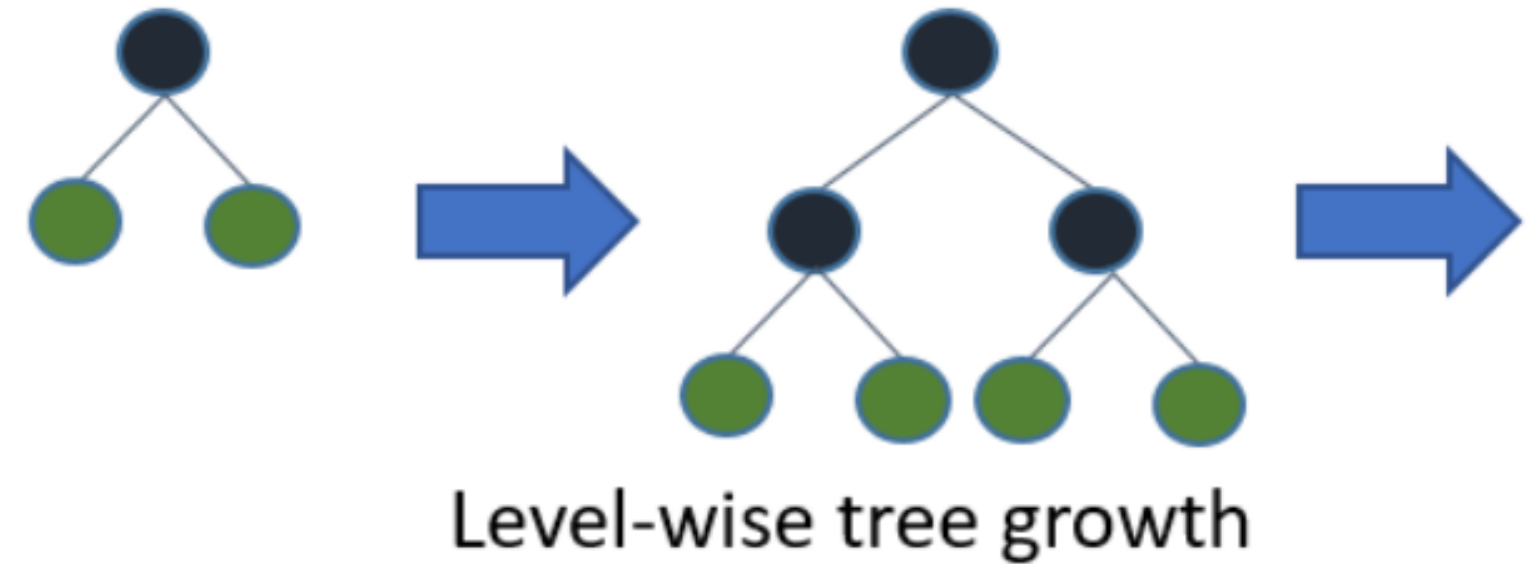
3) 두 확률 모델

- 어떤 분류기를 쓸까

Light GBM



- 하나의 의사결정 트리를 가중치 학습하여 발전시킨 여러 의사결정 트리의 앙상블
- 여러 의사결정 트리가 서로 상관되므로 과대적합의 문제가 있음



- 일반적인 Boost 모델은 Level-wise 방식
- LGBM은 Leaf-wise 방식으로 연산 속도 빠름

5. 예측 결과 및 해석

1) 훈련 방법

(1) 대출 신청 '유저' 예측 훈련

- 대출을 받은 적이 있는 경우 $y = 1$,
- 그렇지 않은 경우 $y = 0$

(2) 대출 신청 '상품' 예측 훈련

- train set 내에서 대출을 받은 적이 있는 유저 그룹 내에 대해서만 훈련

Log_data
CompleteIDCertification
GetCreditInfo
EndLoanApply
StartLoanApply
Login
OpenApp
SignUp
UseDSRCalc
UseLoanManage
UsePrepayCalc
ViewLoanApplyIntro



User_spec
Credit_score
Yearly_income
Income_type
Employment_type
Houseown_type
Desired_amount
Purpose

Decision Tree Model

'유저'가 대출 받을 확률

Loan_result
Rate_s
Limit_s
Bank_applied
Product_applied

Decision Tree Model

대출을 받는 유저가
이 '상품'을 선택할 확률

5. 예측 결과 및 해석

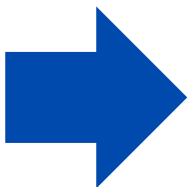
2) 예측 전략

(1) 대출 신청 '유저' 예측

Log_data
CompleteIDCertification
GetCreditInfo
EndLoanApply
StartLoanApply
Login
OpenApp
SignUp
UseDSRCalc
UseLoanManage
UsePrepayCalc
ViewLoanApplyIntro



User_spec
Credit_score
Yearly_income
Income_type
Employment_type
Houseown_type
Desired_amount
Purpose

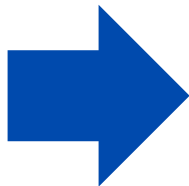


user_1	0.6
user_2	0.2



(2) 대출 신청 '상품' 예측

Loan_result
Rate_s
Limit_s
Bank_applied
Product_applied



user_1	loan_a	0.03
	loan_b	0.9
user_2	loan_c	0.2
	loan_d	0.01



user_1	loan_a	0.018
	loan_b	0.54
user_2	loan_c	0.04
	loan_d	0.002

(3) 두 확률의 곱

5. 예측 결과 및 해석

2) 예측 전략

(4) 예시

	user_id	application_id	bank_id	product_id	is_applied	user_pred	loan_pred	y_pred
5503526	851700	1924079	11	170	0.000	0.840	0.000	0.000
5503525	851700	1924079	62	200	0.000	0.840	0.000	0.000
5503524	851700	1924079	21	196	0.000	0.840	0.000	0.000
5503523	851700	1924079	35	267	0.000	0.840	0.000	0.000
...
5193994	53865	1920504	47	138	1.000	1.000	1.000	1.000
2034172	305038	444452	49	136	1.000	1.000	1.000	1.000
867931	829978	2166142	27	176	1.000	1.000	1.000	1.000
5473559	448288	1869828	24	264	1.000	1.000	1.000	1.000
8703711	485090	1019855	1	61	1.000	1.000	1.000	1.000

5. 예측 결과 및 해석

3) 예측 결과

model	accuracy	precision	recall	f1_score
2 prob (rf+rf)	0.96290	0.73239	0.48689	0.58493
2 prob (lgbm+rf)	0.95922	0.69731	0.40243	0.51033
2 prob (rf+lgbm)	0.94765	0.59767	0.02535	0.04864
2 prob (lgbm+lgbm)	0.94748	0.57589	0.01930	0.03735
auto encoder	0.90040	0.05882	0.05849	0.05866

confusion_matrix_(rf+rf)	confusion_matrix_ideal
[0.9369, 0.0095]	[0.94431, 0.]
[0.0274, 0.026]	[0., 0.05569]

5. 예측 결과 및 해석

3) 시사점

(1) 대출 서비스 수준 향상: 본 분석 결과를 활용 대출 서비스의 질적 향상 도모

- 대출 제공자의 경쟁력 향상

- 매력적인 대출 상품의 조건을 이해함으로써 경쟁력을 제고할 수 있음
- 다수의 대출 제공자들의 상품 제시 능력 향상은 대출 상품 소비자의 효용 증가를 의미

- 플랫폼 운영자의 대출 연계 능력 향상

- 유저에 따른 선호 대출 상품을 파악하여 예상 선호 순서로 대출 상품 매칭
- 대출 신청 확률이 높은 유저들을 파악하여 효과적 권유 시스템 구축

5. 예측 결과 및 해석

3) 시사점

(2) 데이터 예측 방법론: 실제 대출 의사결정 과정을 반영하는 새로운 예측 방법론 제시

- 단순히 상관성이 높은 입력변수들을 검증된 모델들에 투입해서는 해결할 수 없는 문제
 - 어떤 대출 상품이 신청될 확률은 그 대출 상품이 유저에게 유리한지 뿐만 아니라 유저가 대출을 받을 의향이 있는 것인지를 암묵적으로 반영함
 - 대출을 결정한 시점에서 중요한 것은 제시 받은 대출 조건 집합 내에서의 상대적 매력도임
- 본 분석에서는 대출 의사결정 과정을 반영하여 모델을 구축하였음
 - 유저가 대출 의향이 있을 확률과, 대출 의향 있는 유저가 해당 상품을 신청할 확률로 분해함으로써 대출 상품 신청 확률에 포함된 두 가지 요소를 명시적으로 고려
 - 유저가 제시 받은 대출 조건 간의 상대적 매력도만을 비교하기 위하여, 유저가 제시 받은 대출 조건 집합 내에서 각 대출 조건을 표준화

감사합니다

2. 데이터 전처리

Raw Data Shape

log

loan

user

17,843,993

*

6

1,3527,363

*

7

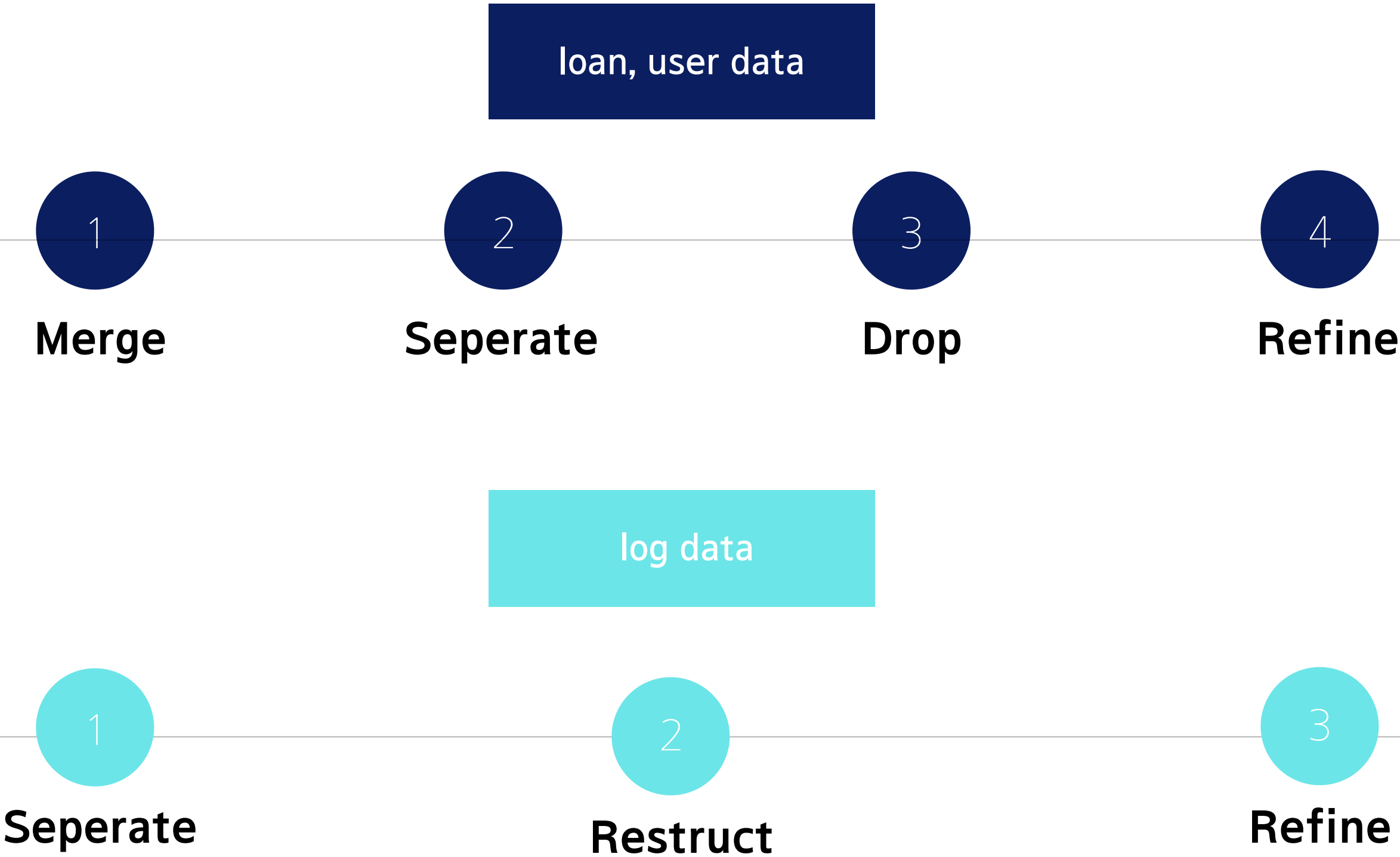
1,394,216

*

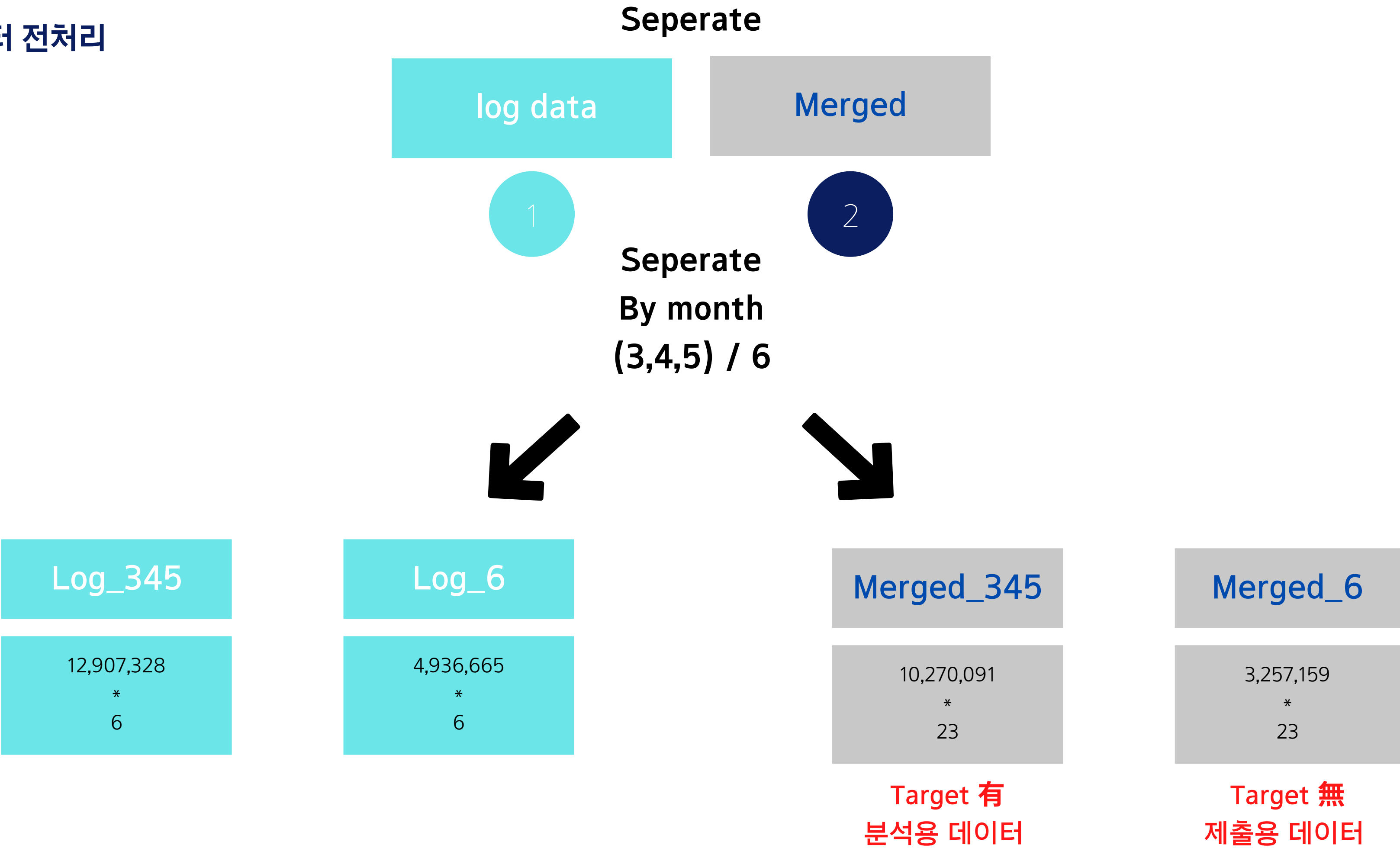
17

전처리 로드맵

전처리 과정은 log, user, loan data 따로 그러나 연계되어 진행됩니다.



2. 데이터 전처리



4. 예측 모델

3) 두 확률 모델

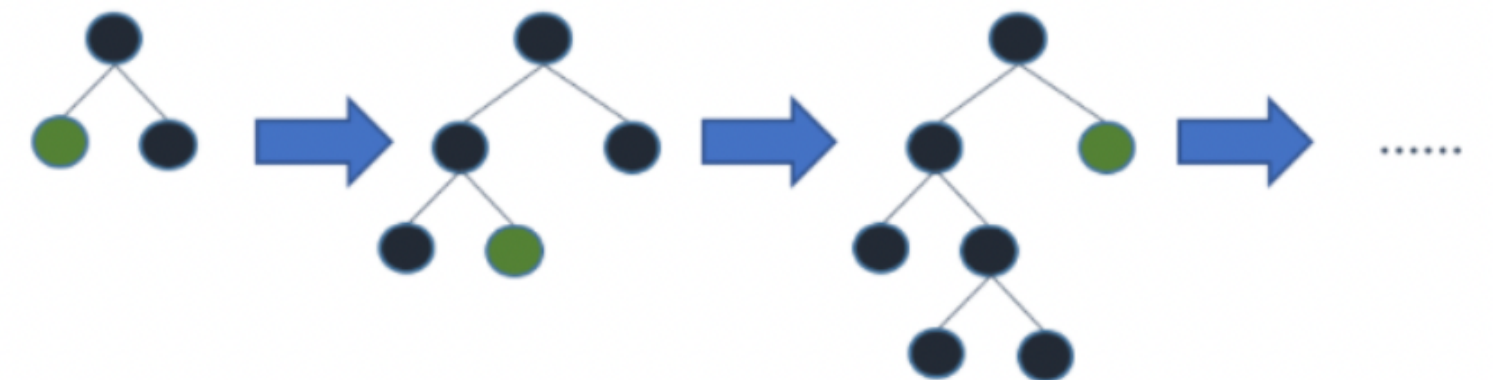
- 어떤 분류기를 쓸까

GBM 기반의 모델, 그러나
기존 Level-wise tree와의 차별,
Leaf-wise 모델 사용



Level-wise tree growth

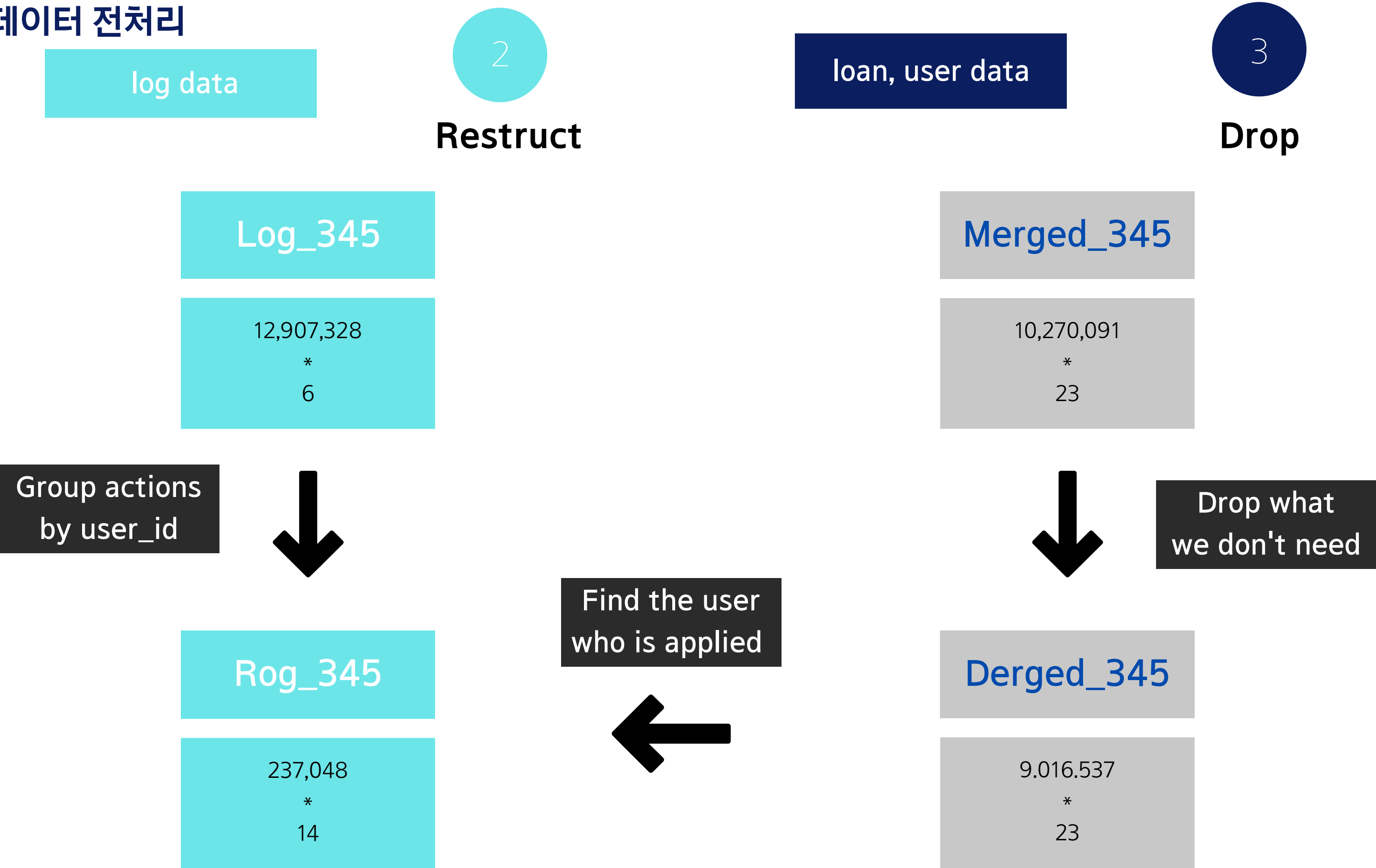
Level-wise tree growth in XGBOOST.



Leaf-wise tree growth

Leaf wise tree growth in Light GBM.

2. 다시 돌아온 데이터 전처리



2. 다시 돌아온 데이터 전처리

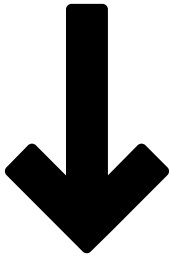
loan, user data

4

Refine

Derged_345

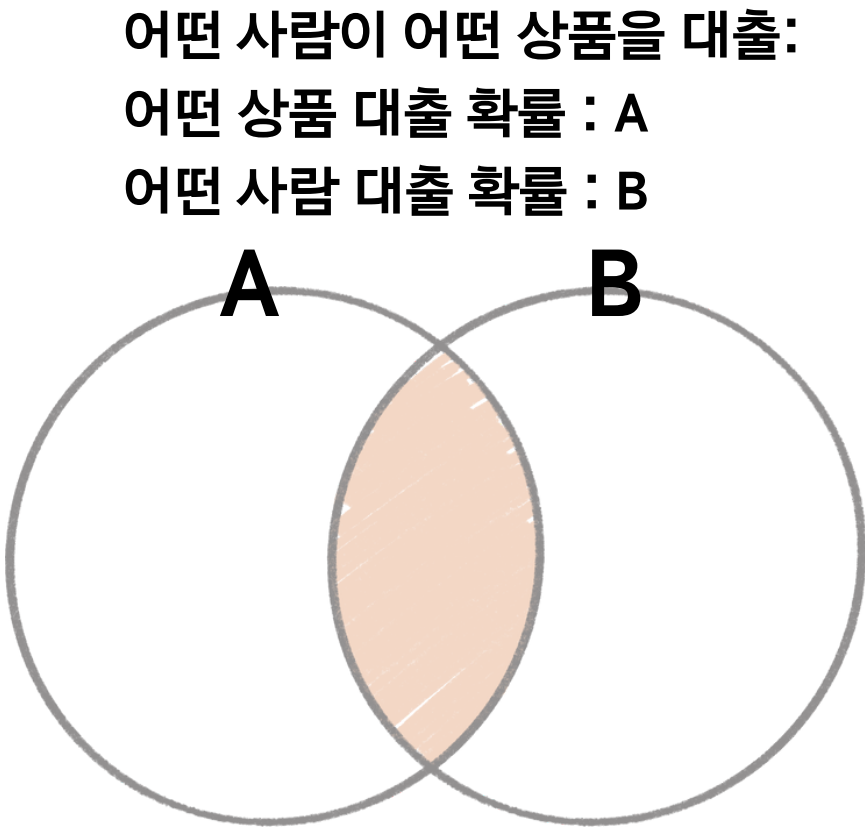
9,016,537
*
23



Find who'd
ever applied
for find P(A)

Derged_345

6,861,422
*
23



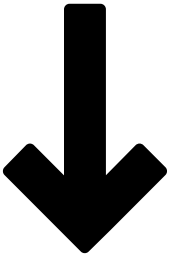
log data

3

Refine

Rog_345

237,048
*
14



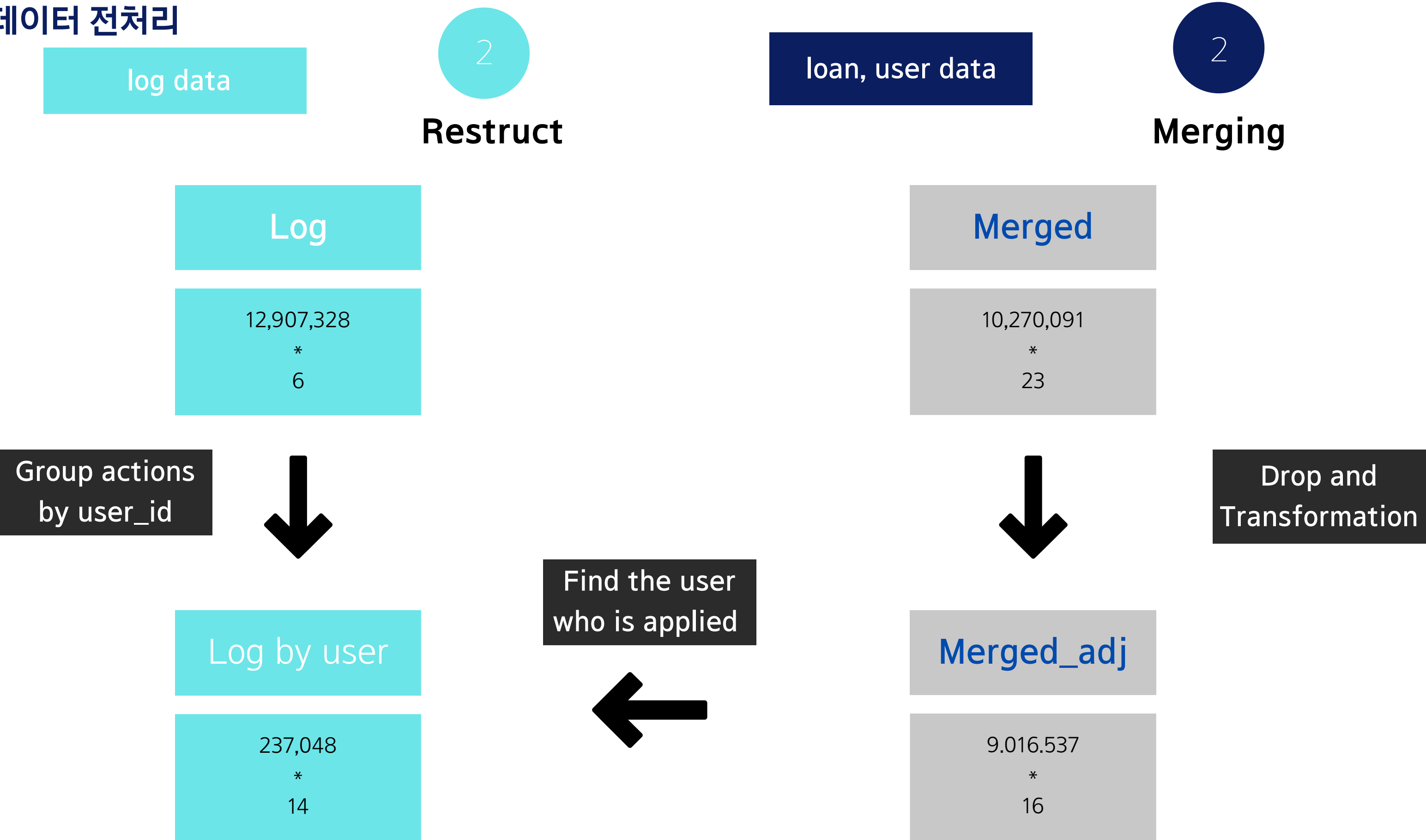
Rog_345

236,395
*
14

Abnormal
Detection

Log Data for
P(B)

2. 다시 돌아온 데이터 전처리



3. EDA

EDA 결과 분석

모델 반영!

나아가 생각해보면, is_applied 변수와 타겟이 아닌 변수들의 관계가 명확하지 않다는 점에서, 우리는 분류 트리 등의 모델을 정말 '도구적'으로 사용하기로 결정했다.

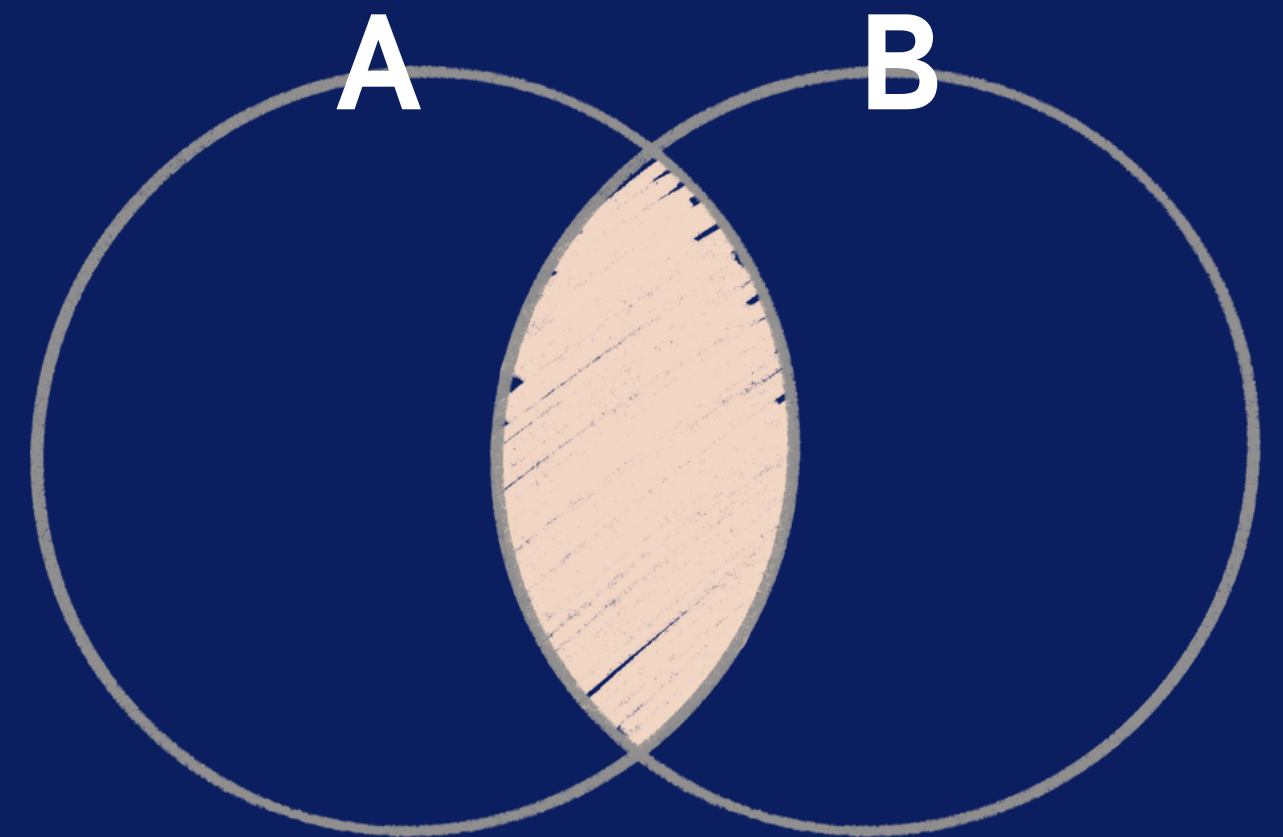
단순한 분류 문제에서 벗어나, 사람이 대출을 받는다고 생각할 때, 합리적인 사고 흐름을 모델에 녹여내기로 했다.

어떤 사람이 어떤 상품 대출을 받는다고 가정하면, 대출 받을 **상품이 어떤 것인지**와, 대출 받는 **사람이 어떤 상황**인지가 중요한 변수로 작용할 것이다.

어떤 사람이 어떤 상품을 대출:

어떤 상품 대출 확률 : A

어떤 사람 대출 확률 : B



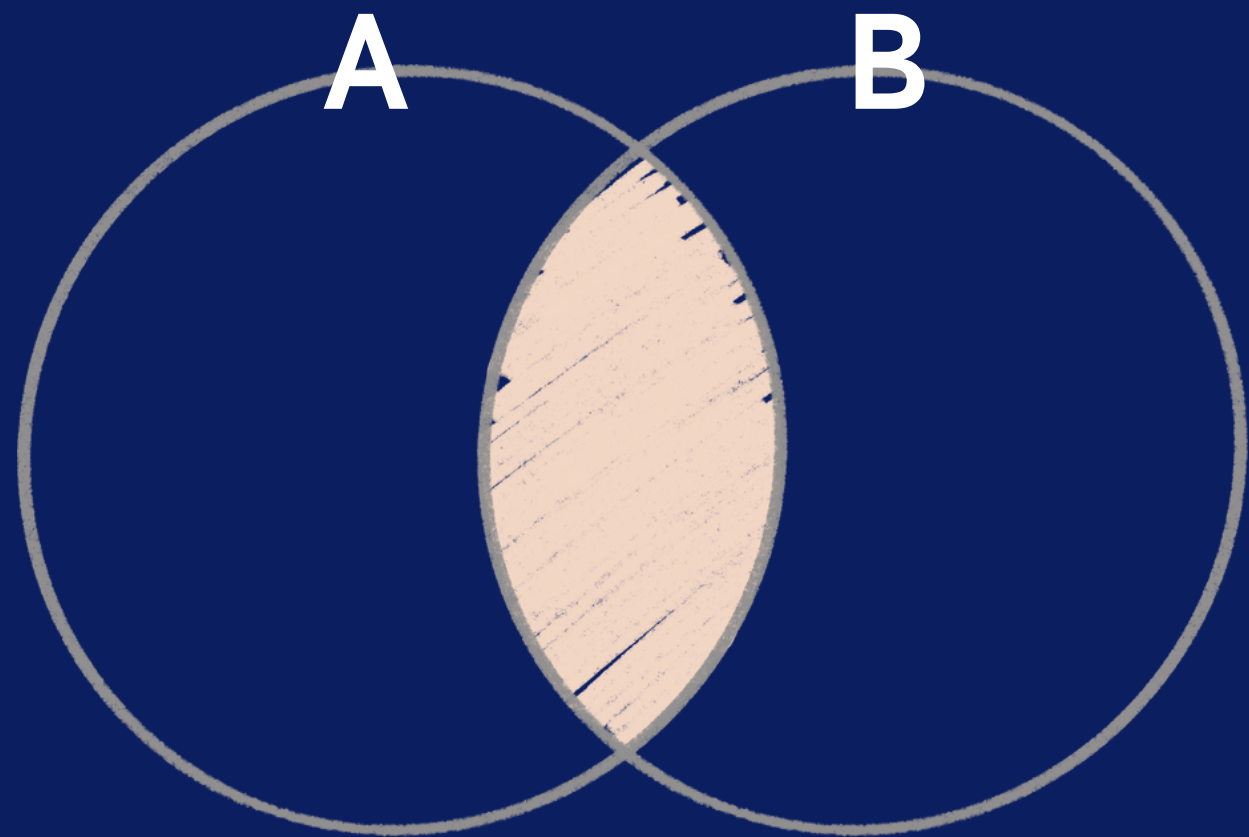
3. EDA

EDA 결과 분석

어떤 사람이 어떤 상품을 대출:

어떤 상품 대출 확률 : A

어떤 사람 대출 확률 : B



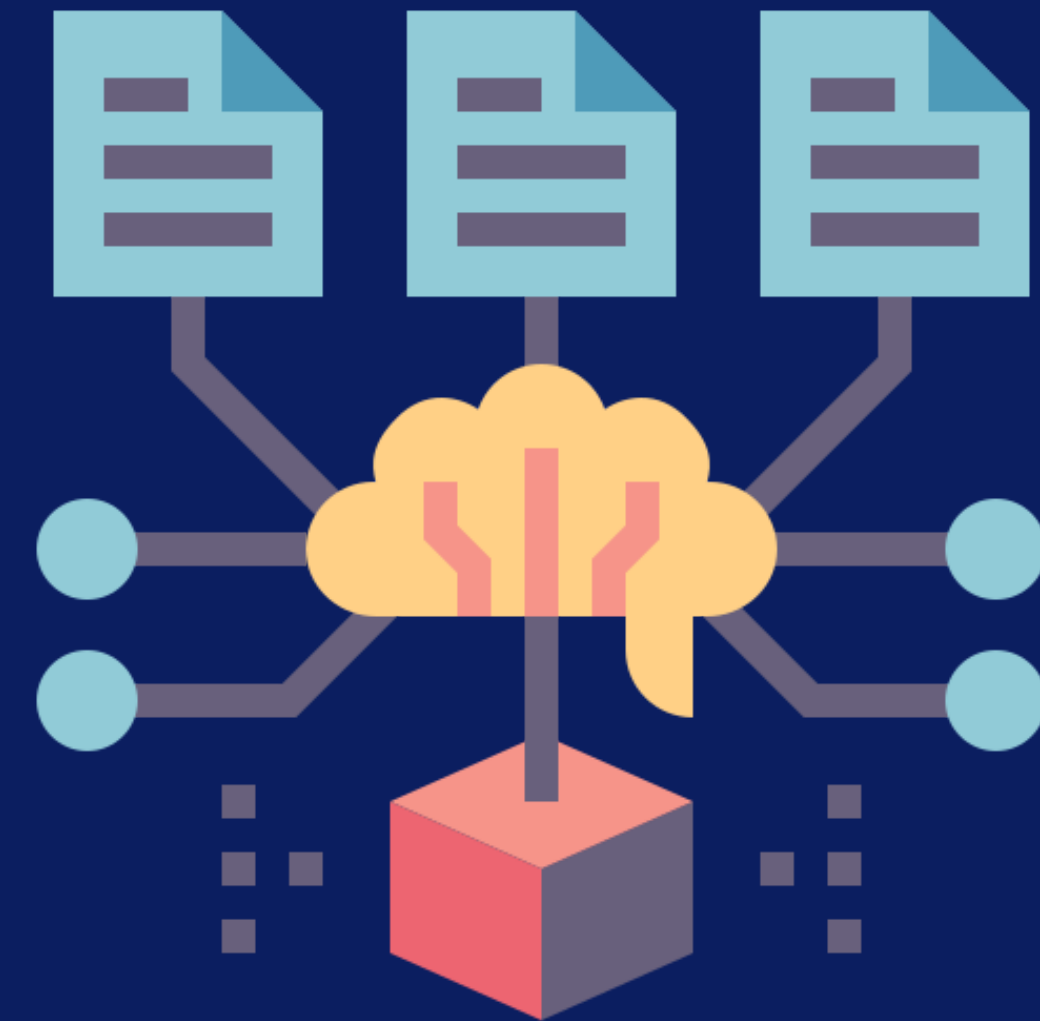
$P(A)$ 와 $P(B)$ 가 서로 독립이라는 가정 하에, 두 사건의 교집합은 확률곱으로, 그렇지 않다면 조건부 확률로 나타낼 수 있을 것이다.

이제 우리의 Task는 단순한 분류가 아닌, 두 사건 각각이 독립이 되는 조건(변수조합)을 찾고, 두 사건의 확률을 잘 설명할 수 있는 모델을 만들어내는 것이다.

먼저 우리는 유저 개인의 행동에서 비롯되는 확률인 $P(B)$ 값을 행동 데이터셋인 로그 데이터에서, 어떤 상품이 매력적인가에 대한 이야기인 $P(A)$ 값을 론 데이터에서 찾고, 유저 변수를 사용하는 방향에 대해 고민할 것이다.

4. 예측 모델

어떤 모델이 우리의 생각을 대변할 수 있을까?



가장 먼저, 생각을 모델에

앞서 말했던 확률값 형태의 모델을, 실제로 구현하기 위해서 우리는 로그 데이터와 머지드 데이터에 각각 서로다른 모델을 적용하고, 가장 설명력이 높은 두 proba 값을 곱해 최종 의사 결정을 내리기로 했다.

생각의 전환

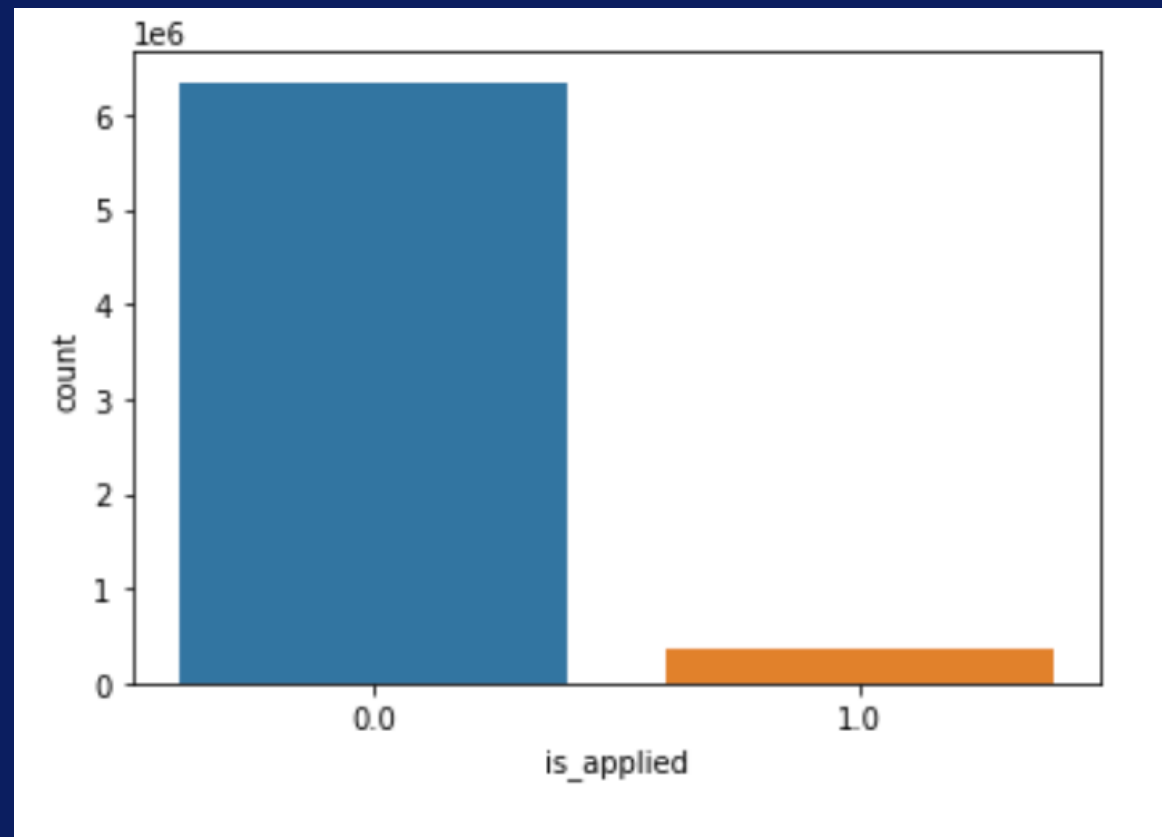
즉, 우리는 0과 1로 이뤄진 자료형을 분류하는 작업이 아닌, 1 속성을 가지게 하는 피쳐들을 분석해, 1 피쳐들을 구해내고자 하는 것이다.

회귀? 분류?

어떤 모델을 선택하든 우리가 결과적으로 구하고자 하는 것은 서로 다른 두 모델에서의 '확률'이다. 몇 가지 모델들을 **변수를 달리하여 적용**해보고 valid 셋에서 가장 높은 설명력을 가지는 모델을 선택하기로 하자.

3. EDA

데이터 불균형 문제



```
0.0    0.945432
1.0    0.054560
Name: is_applied, dtype: float64
```

```
[37] import lightgbm as lgb

lgb_clf = lgb.LGBMClassifier( objective='binary', metric='binary_logloss')
lgb_clf.fit(X_train, y_train)
y_pred = lgb_clf.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.9443423628987785
```

```
[39] from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)/len(y_pred)
print(cm)
```

```
[[0.94305561 0.00101883]
 [0.05463881 0.00128675]]
```

데이터의 불균형을 고려하지 않았음 ..
첫 분류기 학습에도 accuracy가 높았지만,
confusion matrix를 그려보니 신청 대출을
맞추는 (1,1)성능이 전체의 1%에 불과

-> 초비상 공모전 이대로 시마이?

에잇 씨팔



2. 데이터 전처리

loan, user data

1

Merge

application_id	user_id	birth_year	gender	insert_time	credit_score	yearly_income	income_type	company_enter_month	employment_type
3863817	341318	1971	1	#####	710	30000000	EARNEDINCOME	201901	계약직
3866840	524359	1976	0	#####	590	42000000	EARNEDINCOME	201807	정규직
3869011	524359	1976	1	#####	930	36000000	EARNEDINCOME	201807	정규직
3866350	733387	1999	1	#####	580	30000000	EARNEDINCOME2	202111	정규직
3866526	801057	1993	0	#####	660	48000000	EARNEDINCOME	202104	정규직

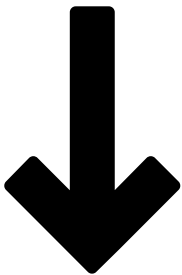
application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied
5072400	2022-06-21 8:31	27	911001	15000000	7.9	0
5072400	2022-06-21 8:31	41	906002	45000000	12.7	0
5072400	2022-06-21 8:31	50	932001	43900000	13.3	0
5072400	2022-06-21 8:31	499	927001	43900000	12.4	0
5072400	2022-06-21 8:31	30	942001	17000000	11.4	0

loan

13,527,363
*
7

user

1,394,216
*
17



Inner Join
By app_id

Merged_adj

9.016.537
*
16

Merged

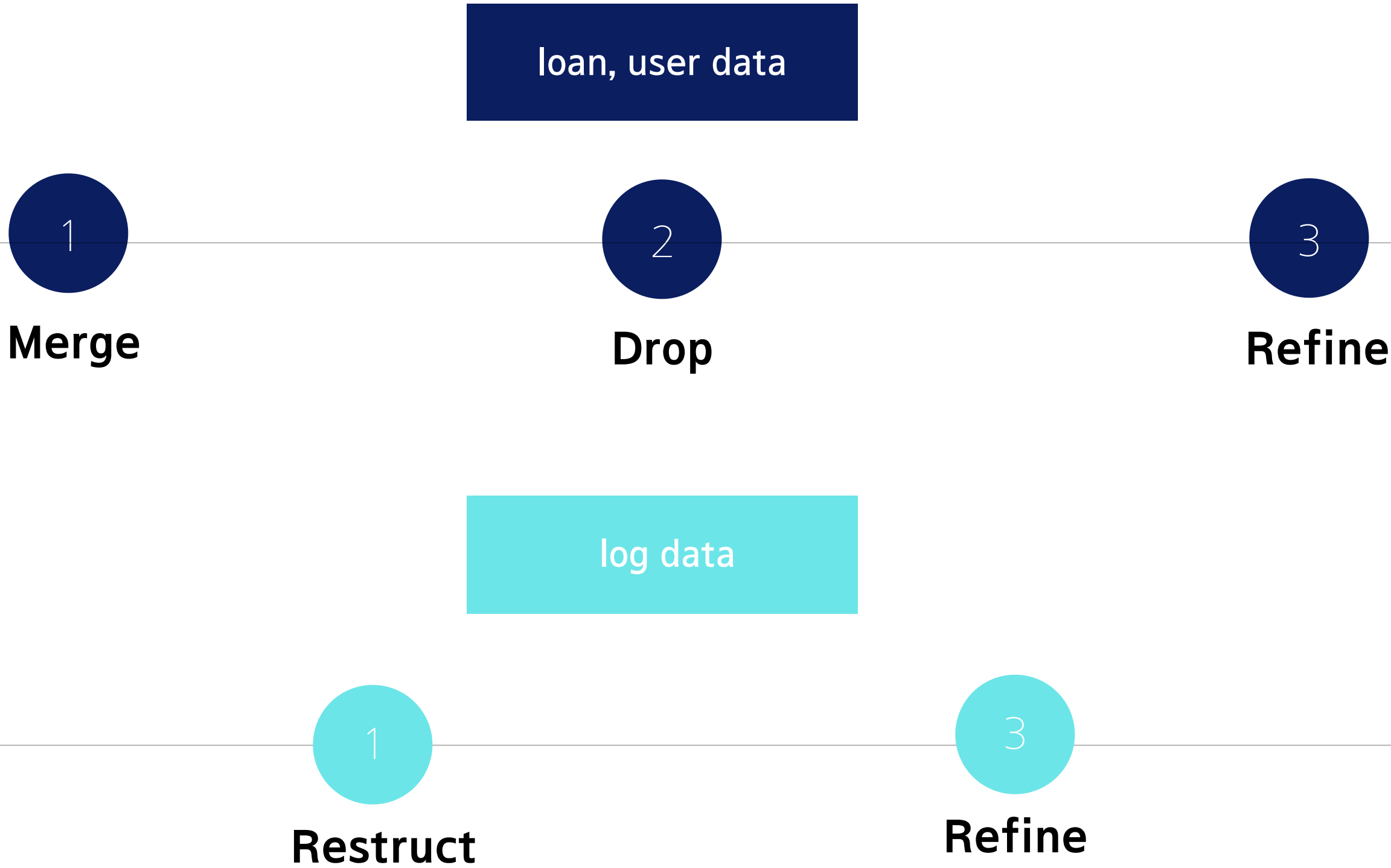
10,270,091
*
23

2. 데이터 전처리

Raw Data Shape

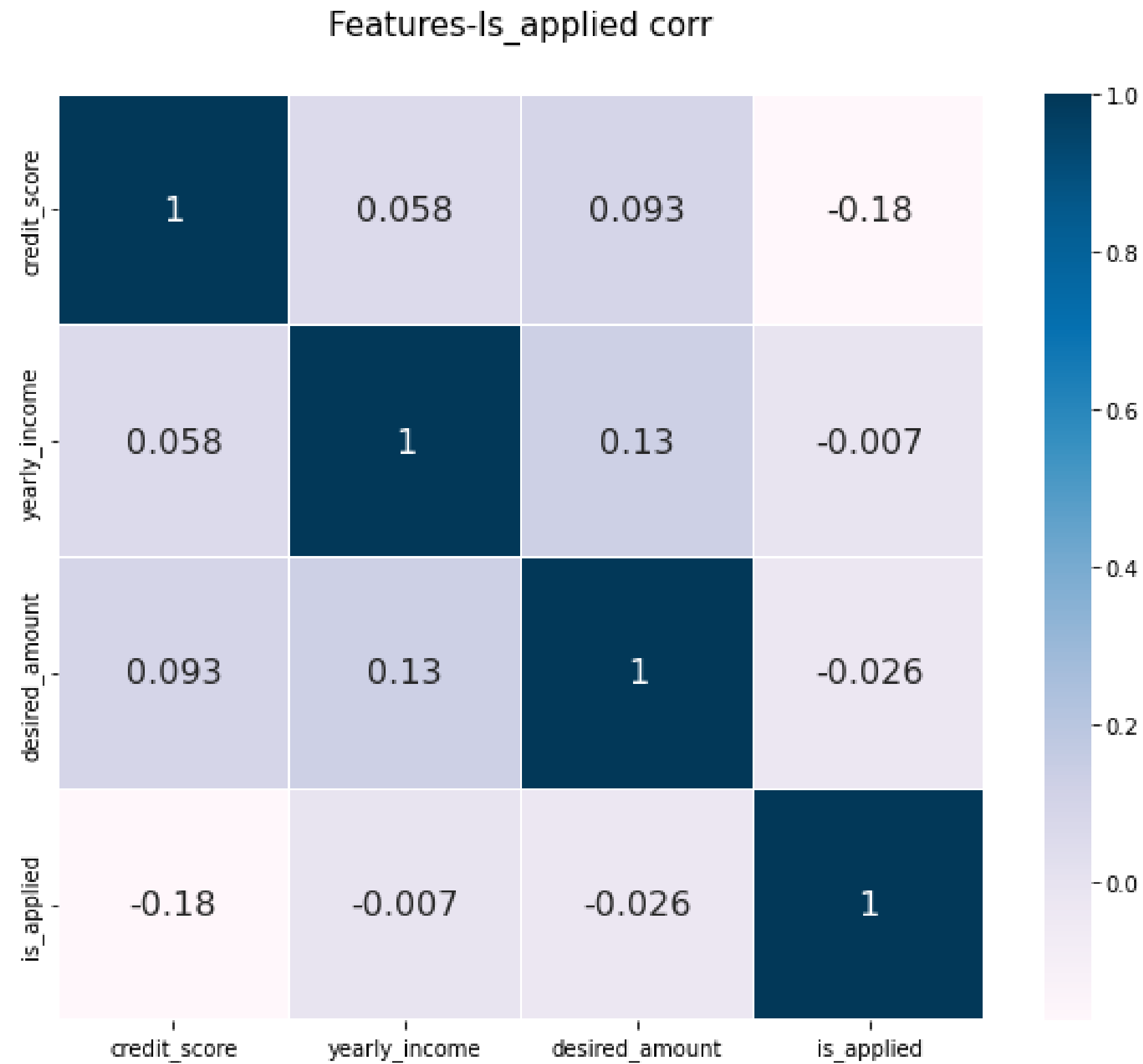
log	loan	user
17,843,993 * 6	1,3527,363 * 7	1,394,216 * 17

전처리 로드맵



3. EDA

2) user 데이터 분석



3. EDA

기초통계량 분석

log, user, loan data의
기초 통계량 분석



변수 간 상관 분석

Heat map을 통해 is_applied
(target)변수와 feature 간의
상관계수 분석

EDA 결과 분석

데이터의 특성을 바탕으로
모델 구성 방향 논의

