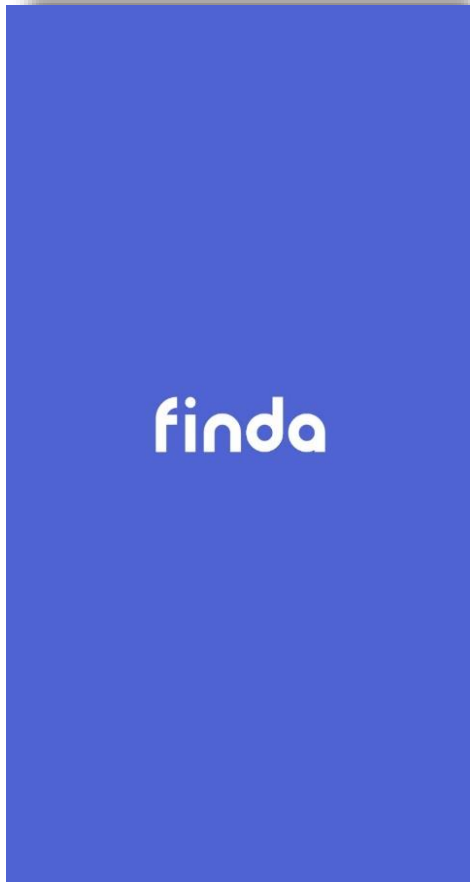


팀명: nan알아요, 팀원: 김희경, 이영아, 이지수, 최윤서

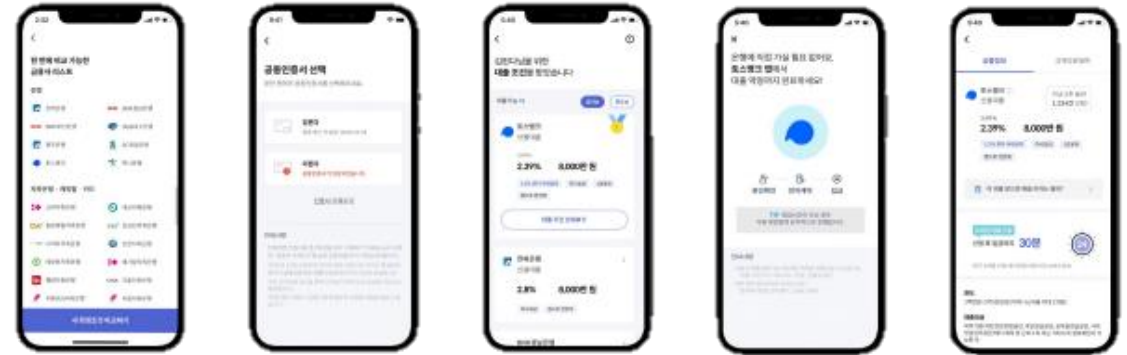
2022 빅콘테스트 제 10회 (BIG-DATA)

데이터 분석 리그 - 퓨처스 분야 - 앱 사용성 데이터를 통한 대출신청 예측 분석

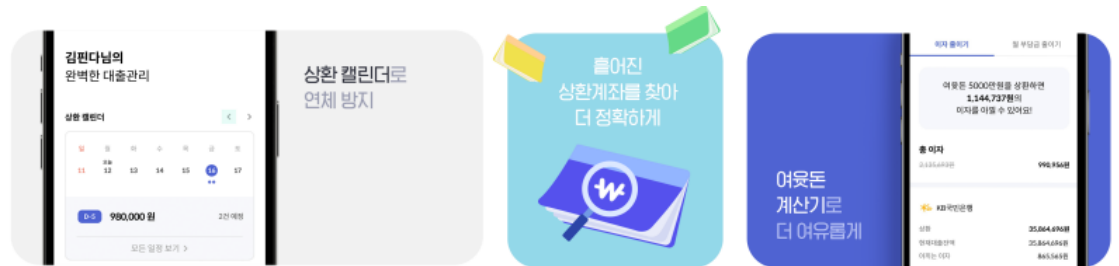
세상에 없던 대출비교 플랫폼 finda



1. 비교대출 서비스

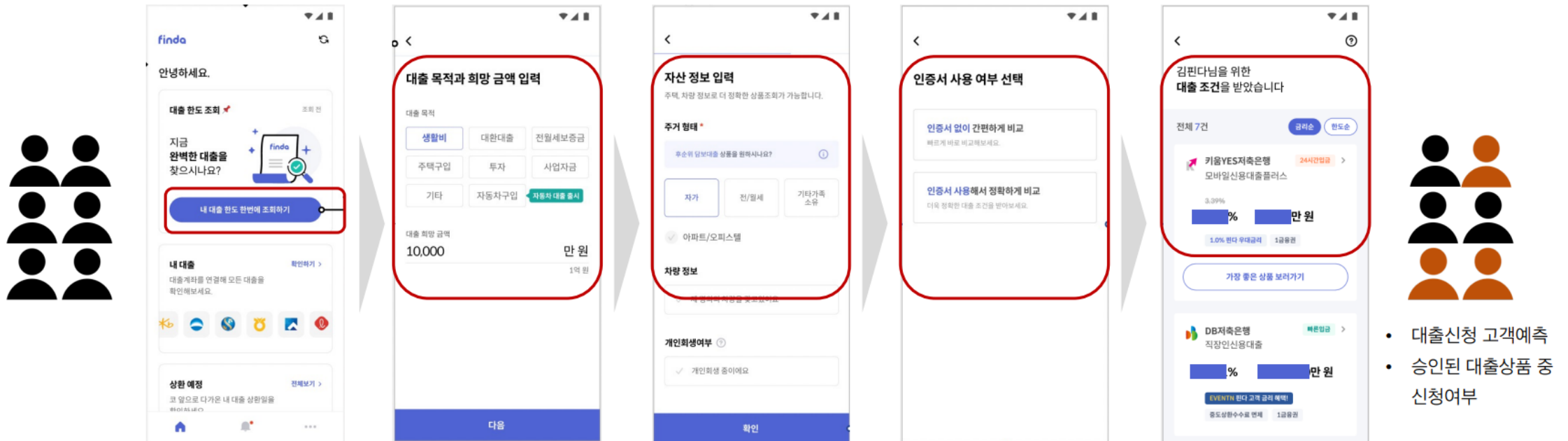


2. 나의 대출관리



문제1. 대출신청 고객 예측

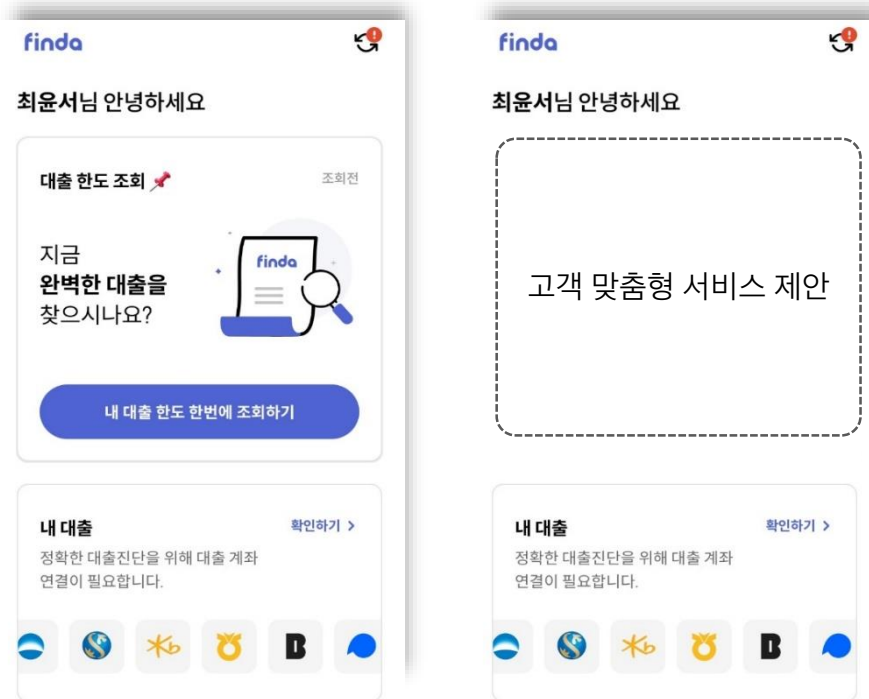
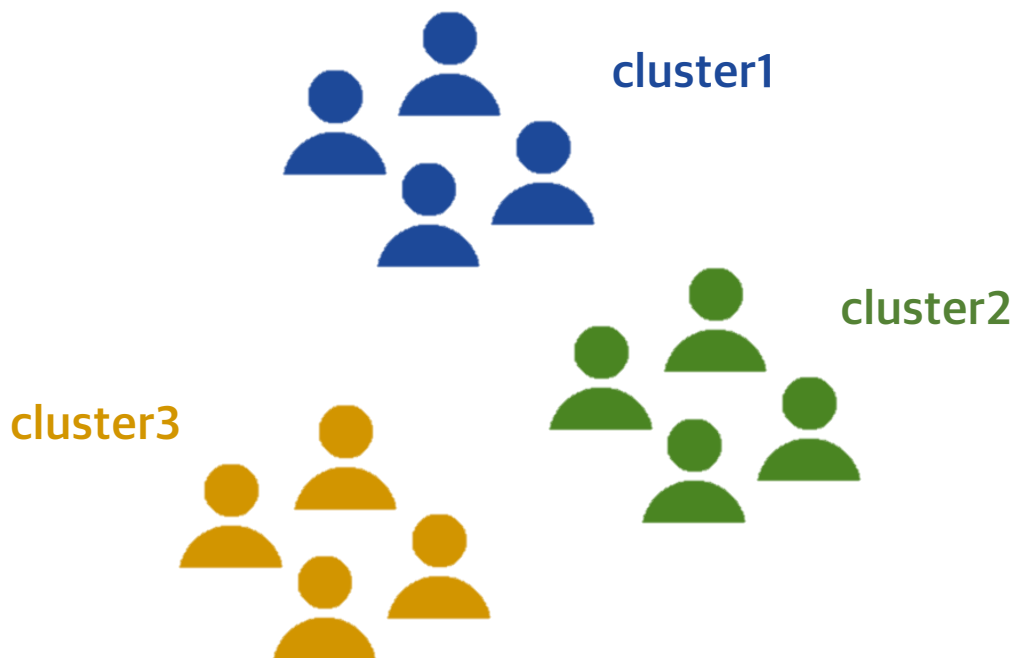
- 사용자의 대출 상품 조회 신청서 정보와 승인된 대출 상품 정보를 바탕으로 대출 신청 고객과 승인된 대출 상품을 예측



- 고객의 대출 신청 여부를 예측하여 고객이 신청할 가능성이 높은 대출을 고객에게 노출시켜 서비스 향상을 도모할 수 있음.
- 대출을 신청하는 고객의 특성을 파악하여 고객 개인 맞춤형 서비스의 기반 자료로 활용 가능

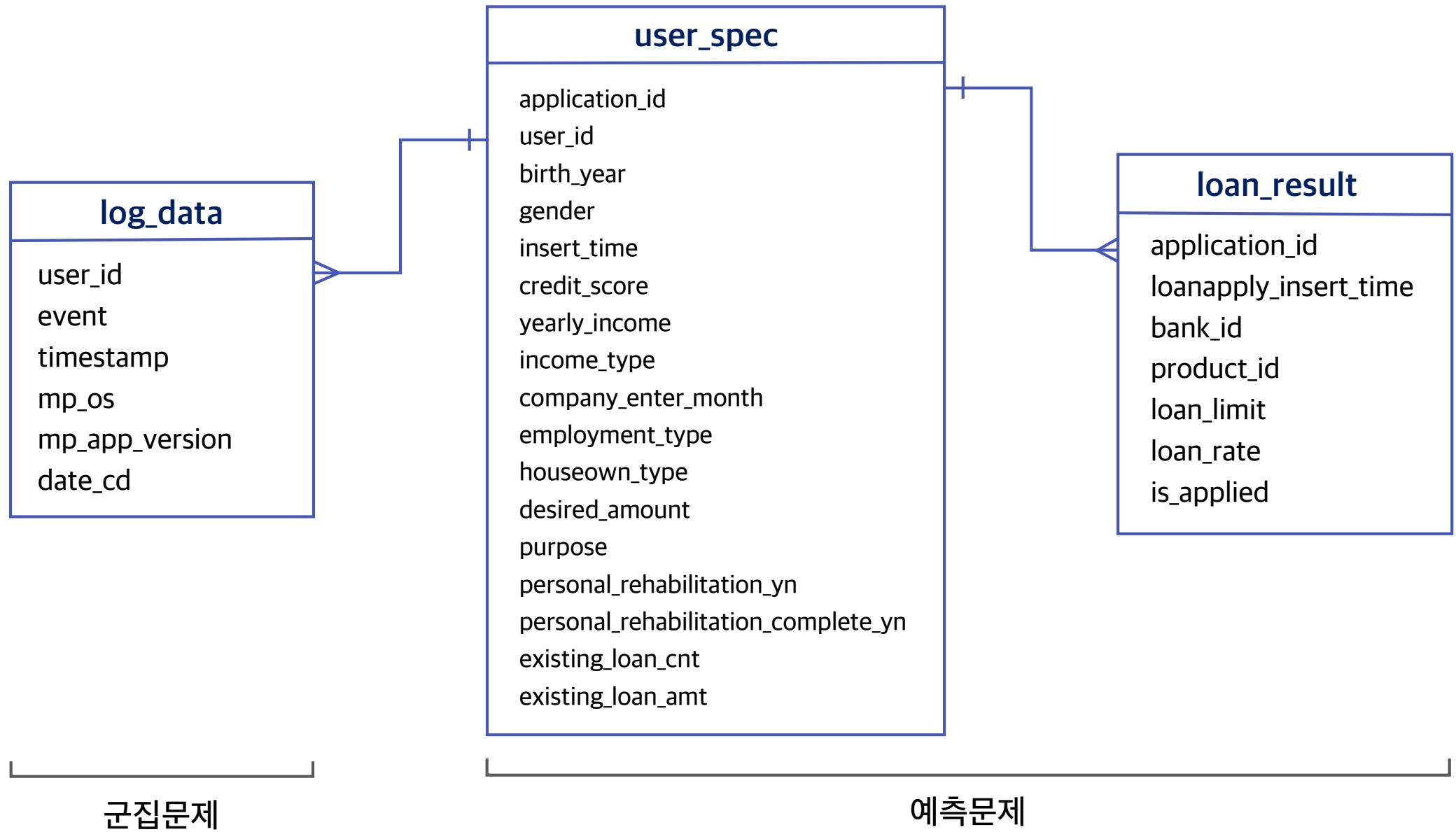
문제2. 모델 기반 고객 군집 분석

- 사용자 사용 기록(log data)을 이용한 군집 결과를 바탕으로 앱 메인 화면 상단에 노출될 서비스 메시지를 제안.



- 어플의 첫 화면을 개인화 시켜 고객에게 더 빠르게 필요한 서비스를 제공할 수 있는 기반을 마련할 수 있음.
- 노출 빈도가 가장 높은 메인 화면의 활용도를 높여 핀다의 서비스 질 향상과 서비스 홍보에 도움이 될 수 있음.

제공 데이터



예측 문제

데이터 전처리 - USER_SPEC - 데이터 소개

- 유저 스펙 테이블 (User_Spec)

birth_year	일반정보	age	gender	employment_period	employment_type
gender		37	1	7	기타
		54	1	15	정규직
company_enter_month	고용정보	insert_time - birth_year		insert_time - company_enter_month	4개 범주 (one-hot encoding)
employment_type					
credit_score	소득정보	credit_score	yearly_income	income_type	houseown_type
yearly_income		660	108000000	PRIVATEBUSINESS	자가
income_type		870	30000000	PRIVATEBUSINESS	기타가족소유
houseown_type				6개 범주 (one-hot encoding)	4개 범주 (one-hot encoding)
desired_amount	대출관련정보	desired_amount	purpose	existing_loan_cnt	existing_loan_amt
purpose		1000000	기타	4	162000000
personal_rehabilitation / complete_yn		30000000	대환대출	1	27000000
existing_loan_cnt / amt				8개 범주 (one-hot encoding)	
		personal_rehabilitation_yn		personal_rehabilitatioin_complete_yn	
		0		0	
		0		0	

데이터 전처리 - USER_SPEC - 결측치 처리

- data cleaning 및 기존 대출 관련 변수 결측치 처리

yearly_income	90
income_type	85
employment_type	85
housewown_type	85
desired_amount	85
purpose	85

EDA) 결측이 같은 사용자에게서 동시에 발생함.

step 1

- 신청서를 불성실하게 작성한 이용자의 데이터.

→ 신뢰성이 떨어지는 데이터로 row data 삭제

step2

- step1 과정 후에 남은 데이터의 yearly_income의 결측은 같은 user_id의 yearly_income 값의 평균으로 대체

사용자가 직접 작성한 신청서 정보를 저장하는 방식으로 데이터가 수집된다

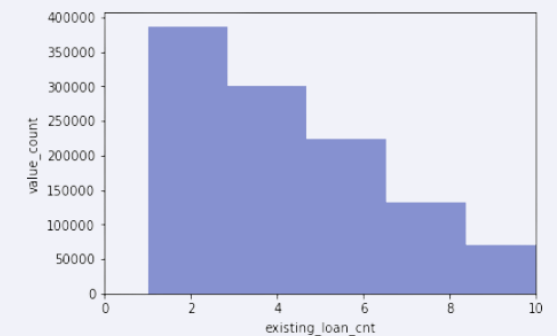
existing_loan_cnt	198556
existing_loan_amt	313774

EDA) raw data에 cnt가 0인 data가 존재하지 않음

step1

- 데이터 수집 시 cnt가 0인 데이터가 nan으로 처리된 것으로 추측됨.

→ cnt가 nan인 데이터는 cnt, amt 모두 0으로 대체



데이터 전처리 - USER_SPEC - 결측치 처리

- 개인회생 관련 변수 결측치 처리

personal_rehabilitation_yn	587351
personal_rehabilitation_complete_yn	1203174

		개인회생여부	
		0	1
개인회생상환여부	0	178139	11356
	1	4	1344

EDA 및 데이터 수집 과정 분석

- 논리적으로 나올 수 있는 경우는 3가지이다.
 - > (0,0) : 개인회생 신청X, (1,0) : 개인회생 신청O, 변제금 납입X, (1,1) : 개인회생 신청O, 변제금 납입O
- (0,0)의 조합이 가장 많은 경우로 나타난다.
- 개인회생여부 여부에 체크를 하지 않더라도 다음 페이지로 넘어갈 수 있다.

step 1

- 논리적인 오류가 있는 (0,1) 조합은 (0,0)으로 대체

step 2

- (0, nan) 인 데이터는 논리에 따라 (0,0)으로 대체

step 3

- (nan,nan) 인 데이터는 개인회생여부에 표시하지 않은 데이터로 간주하고 (0,0)으로 대체

개인회생여부를 체크하지 않아도 다음 버튼이 활성화 됨.

개인회생여부를 체크하지 않으면 변제금 납입 체크창이 활성화되지 않는다.

데이터 전처리 - USER_SPEC - 결측치 처리

- 수치형 변수 결측치 처리

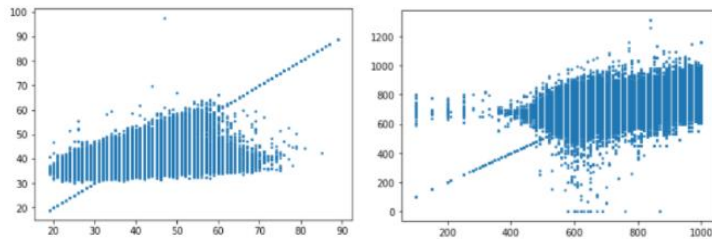
credit_score	105014
existing_loan_cnt	115148
existing_loan_amt	171578
age	12959

EDA 및 결측치 처리 방법

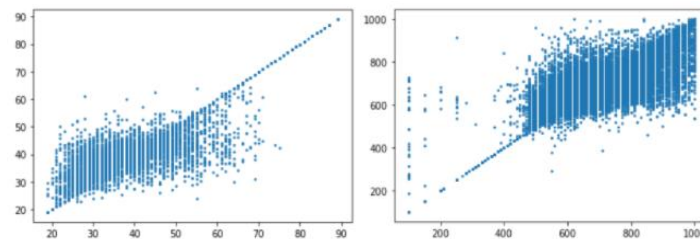
- 결측 데이터와 기존 데이터의 분포가 크게 차이 나지 않음
- 무작위결측(MAR) 가정 → MICE를 이용한 결측치 대체

• MICE(Multiple Imputation with Chained Equations, 다중대체법)

- 분포에 대한 가정 없이 연속된 회귀방정식을 통해 값을 채워나가는 방법.
앞서 채워진 변수는 다음 채워지는 변수의 독립변수로 활용되는 방식
- Python의 IterativeImputer 함수를 활용하여 MICE 방법 적용
- 사용되는 회귀모델은 다양한 모델을 적합 후 실제값과 대체값의 quantile plot을 이용하여 가장 $y = x$ 그래프와 비슷한 모양을 가지는 ExtraTree 모델로 선택

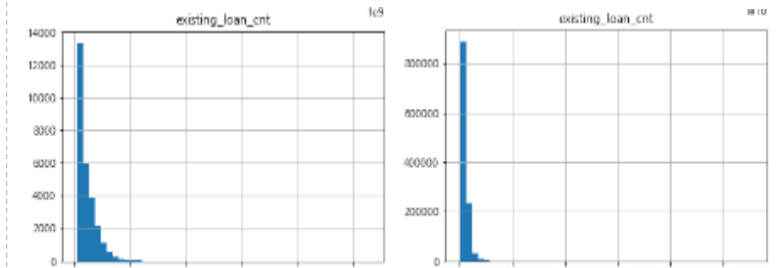
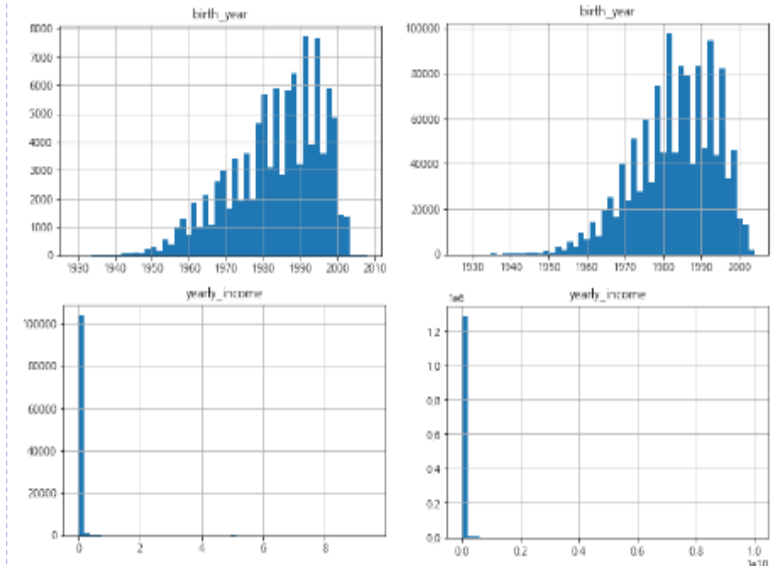


<linear regression 적용 시의 quantile plot>



<ExtraTree 적용 시의 quantile plot>

<credit score 결측 여부에 따른 변수별 분포>



결측X

결측O

데이터 전처리 - USER_SPEC - 결측치 처리

- gender 변수 결측치 처리

gender 12959

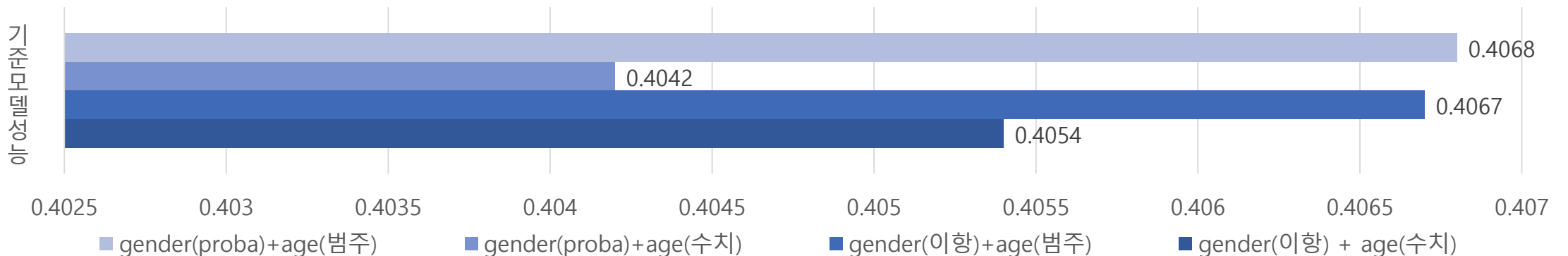
결측치 처리 방법

- 앞선 단계들에서 결측치가 채워진 변수를 이용하여 gender 예측
→ logistic regression 모형 적합 후 예측 확률로 gender 값을 대체함.

- age(수치형) 변수 범주화

30대 미만	1
30대 이상 40대 미만	2
40대 이상 50대 미만	3
50대 이상 60대 미만	4
60대 이상 70대 미만	5
70대 이상 80대 미만	6
80대 이상	7

- gender와 age변수의 전처리는 각 경우의 수를 고려하여 기준모델(rf)에서 성능이 가장 좋은 조합을 사용함.



데이터 전처리 - LOAN_RESULT

- 대출상품결과테이블

bank_id	상품 식별 아이디
product_id	
loan_limit	대출 상품 정보
loan_rate	
is_applied	타겟변수

1. 같은 application_id로 이루어진 행들이 나열된 데이터

- 각 행이 독립이 아니다.
- 서로의 정보가 각각의 행의 예측에 도움이 될 수 있다.
ex) 승인된 더 좋은 상품이 존재한다면 0일 가능성이 높아짐.

2. loan_rate, loan_result가 nan인 데이터의 규칙

- 두 변수가 nan이면 is_applied는 모두 1이다.
- 최종 예측에서 두 변수가 nan인 데이터는 1로 예측


application_id	bank_id	product_id	loan_limit	loan_rate	is_applied
1748340	7	191	42000000	13.6	0
1748340	25	169	24000000	17.9	1
1748340	2	7	24000000	18.5	0
1748340	4	268	29000000	10.8	0
1748340	11	118	5000000	16.4	0
1748340	35	168	21000000	15.2	1
1748340	44	8	3000000	14.8	0
1748340	28	217	10000000	18.0	0
...
1222550	13	262	nan	nan	1
135727	10	149	nan	nan	1
687402	51	21	nan	nan	1
1719382	13	262	nan	nan	1

데이터 전처리 - LOAN_RESULT - 파생변수 생성

- 파생변수 생성

- loan_limit, loan_rate의 min, max, mean 변수 생성
 - 같은 application_id간 승인된 대출 상품 정보 공유를 위해 application_id별 loan_limit, loan_rate의 min, max, mean 변수를 추가.
 - 파생변수 추가 후 valid set의 f1-score가 0.03 향상됨.

application_id	loan_limit	loan_rate	is_applied
1748340	42000000	13.6	0
1718340
1748340	10000000	18.0	0
1357275	21000000	14.8	1



application_id	loan_limit_min	loan_limit_max	loan_limit_mean	loan_rate_min	loan_rate_max	loan_rate_mean
1718340	3000000	42000000	19750000	10.8	18.5	17.89
1718340	3000000	42000000	19750000	10.8	18.5	17.89
1718340	3000000	42000000	19750000	10.8	18.5	17.89
1718340	3000000	42000000	19750000	10.8	18.5	17.89
...

데이터 전처리 - LOAN_RESULT - 파생변수 생성

- bank_id, product_id 전처리 및 파생변수 생성

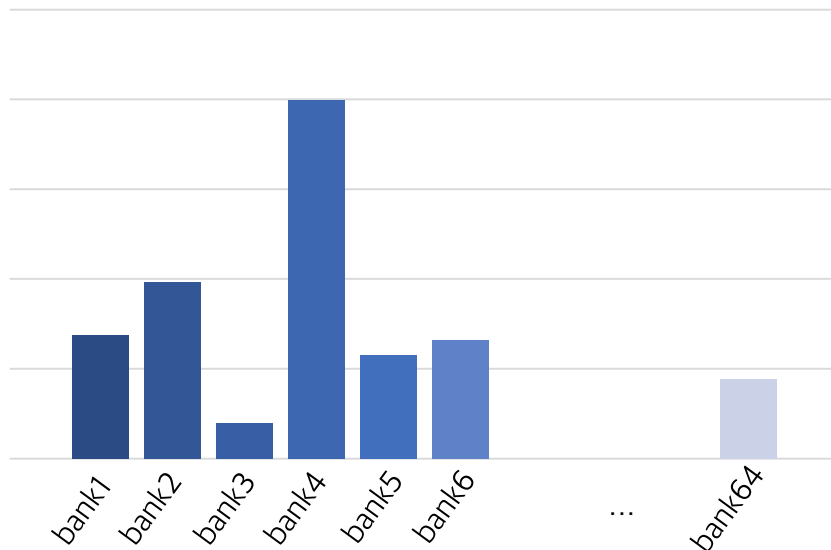
상품 식별 아이디

bank_id	은행별 식별 번호 (총 63개의 범주)	levels : 1, 2, ..., 64
product_id	상품별 식별 번호 (총 270개의 범주)	levels : 1, 2, ..., 270

→ one-hot encoding 시 변수가 너무 많아져 메모리 부족, 계산 시간 오버 문제가 발생

bank_apply_prop

파생변수



- bank_id 별 is_applied 비율을 계산한 'bank_apply_prop' 변수 생성
- product_id의 경우 사용자가 대출을 신청할 때 개설 은행의 영향은 많이 받을 수 있지만 대출 상품(금리, 한도를 제외한 상품명 등)의 영향은 많이 받지 않을 것이라는 가정으로 변수 제거.
 - 실제로 product_id에 대하여 bank_id에 적용한 방식으로 파생변수를 생성해보았지만 큰 효과가 없었음.

데이터 전처리 - merge

- user_loan 테이블 생성

- 유저스펙테이블 (1394216 x 17)
- 대출상품결과테이블 (13527363 x 7)

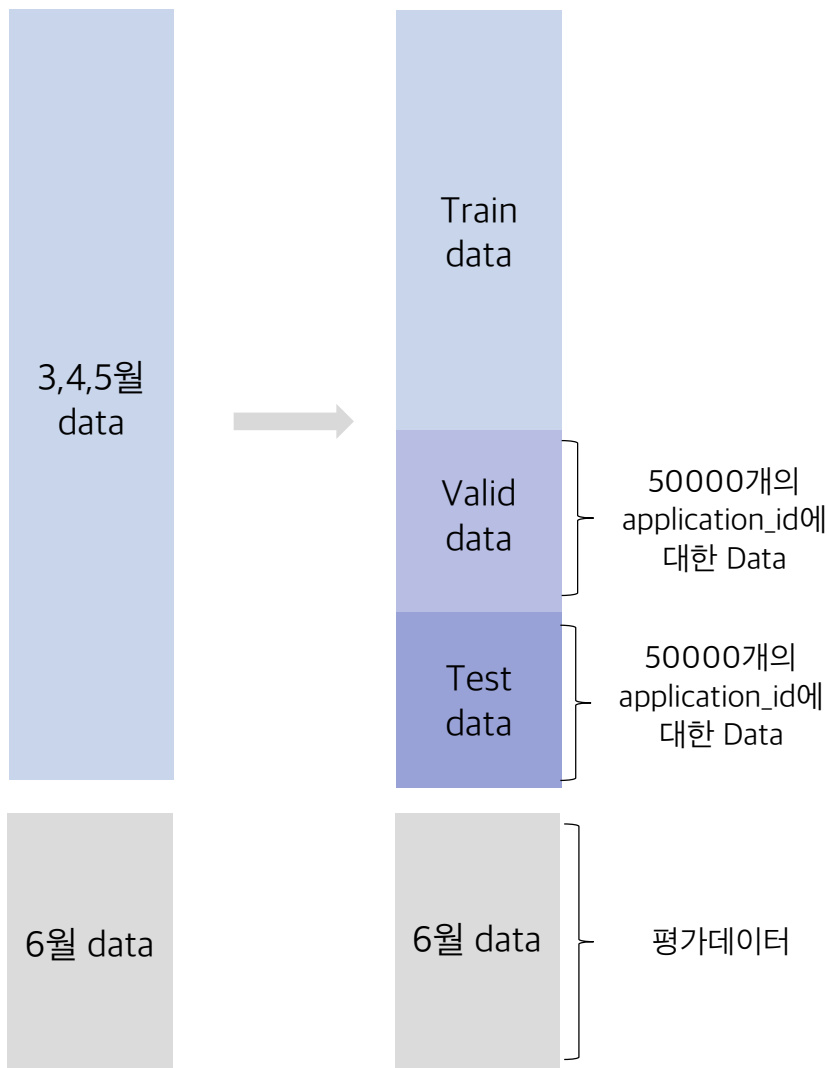
application_id	user_id	age_cat	credit_score	...	purpose	application_id	bank_id	...	loan_rate	is_applied
1748340	848651	3	660		기타	1748340	3		18.5	0
1357275	601384	3	870		대환대출	1357275	45		14.7	1

merge

- User_Loan 테이블 (13527229 x 49)

application_id	user_id	age_cat	credit_score	loan_limit	loan_rate	...	bank_apply_prob	is_applied
178340	848651	3	660	42000000	18.5		0.0463	0
178340	848651	3	660	42000000	14.7		0.0654	1
1357275	601384	3	870	10000000	15.6		0.0324	0

Data Split



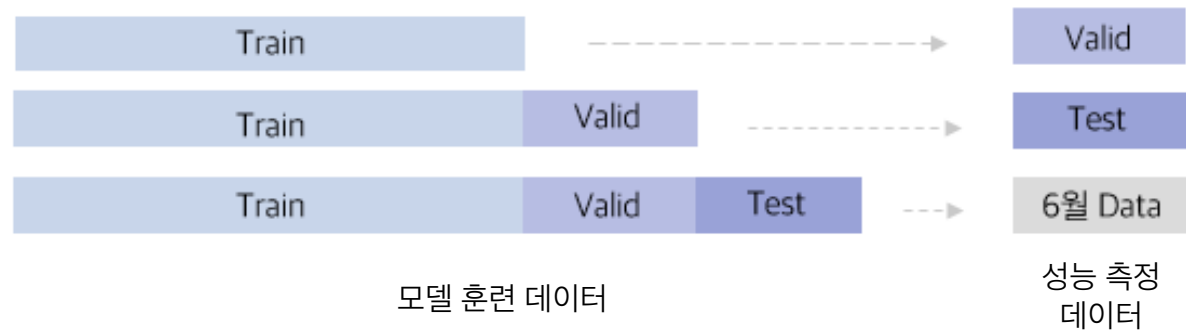
- application_id를 index로 한 Data Split



→ application_id를 기준으로 하여 train / valid / test 할당

→ valid, test의 application_id는 각각 50,000개로 설정

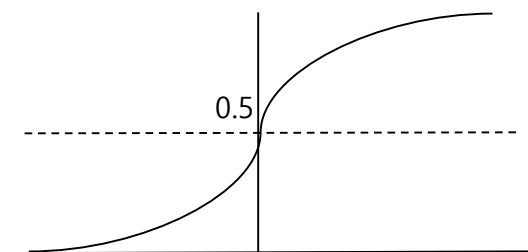
- Modeling 성능 확인



Modeling

- 단일 모델 성능 비교

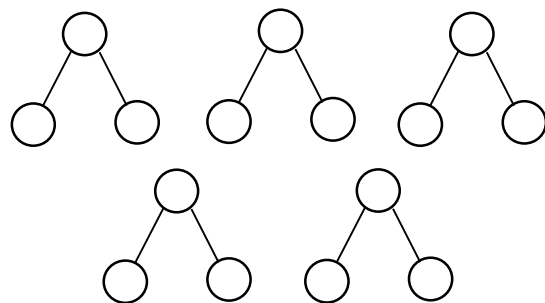
Logistic Regression



C = 1.0
Solver = "lbfgs"
Penalty "l2"

train	validation
0.1170	0.1232

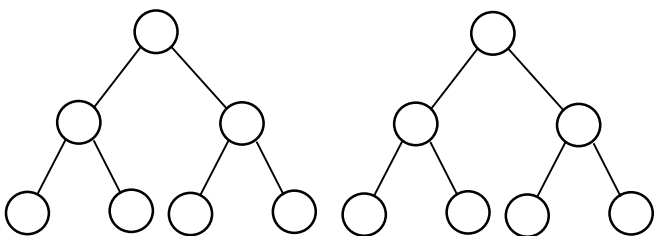
XGBoost



n_estimators = 500
learning_rate = 0.1
max_depth = 10
colsample_bytree & bylevel = 0.7
subsample = 0.8

train	validation
0.4139	0.3751

RandomForest



n_estimators = 200
min_samples_split = 30
class_weight = "balanced"

train	validation
0.7083	0.4567

Modeling

- 이중 모델 프로세스

RandomForest

	예측(0)	예측(1)
실제(0)	632120	36567
실제(1)	16189	22177

XGBoost

	예측(0)	예측(1)
실제(0)	566439	102248
실제(1)	5904	32462

➔ 두 개의 모델이 각각 잘 분류하는 label이 다르게 나타난다. (RandomForest는 0을 잘 예측, XGB는 1을 잘 예측한다)

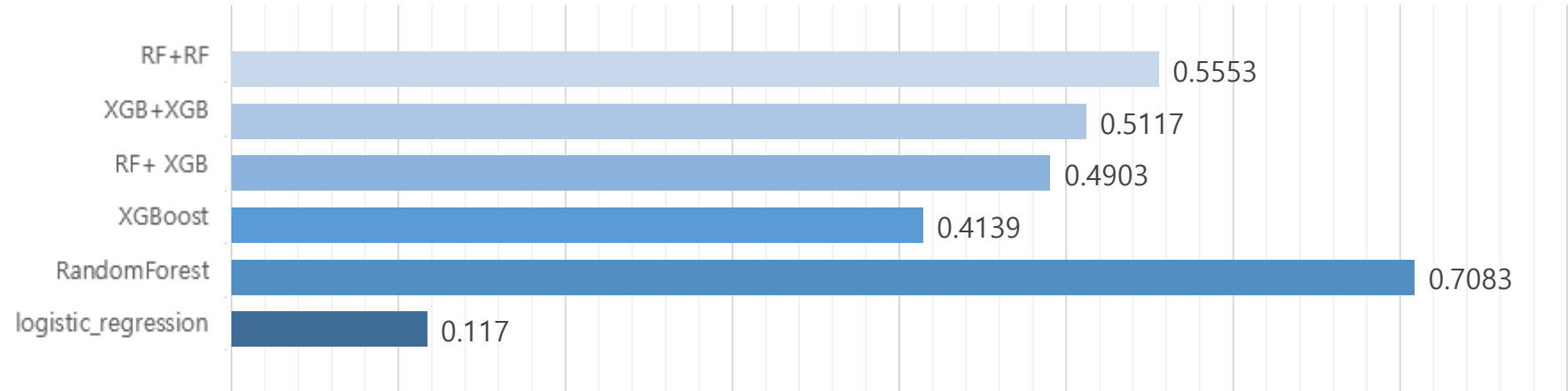


➔ RandomForest 모델과 XGBoost 모델을 결합하여 이중 모델 프로세스를 적용하여 성능 측정

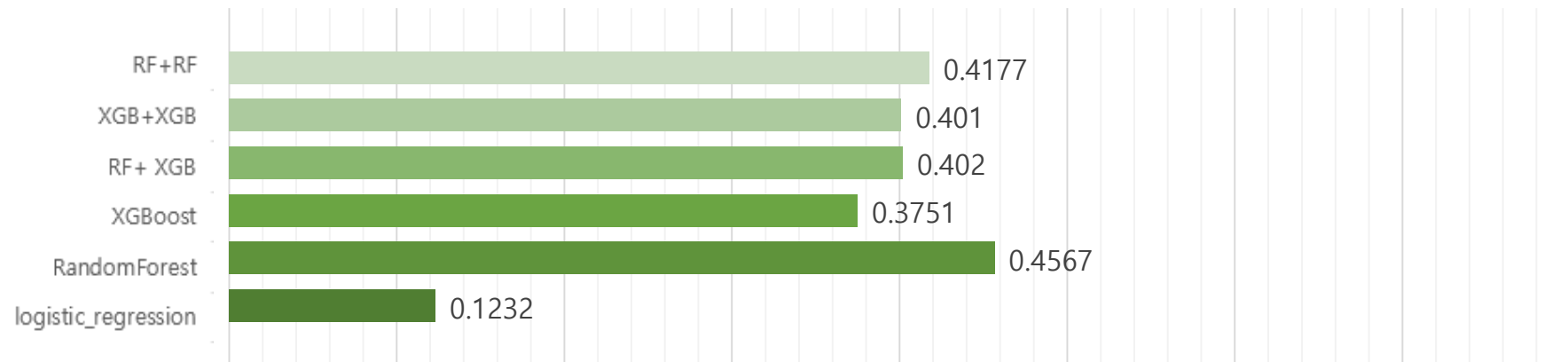
Modeling

- 모델 성능 비교

train data
f1-score



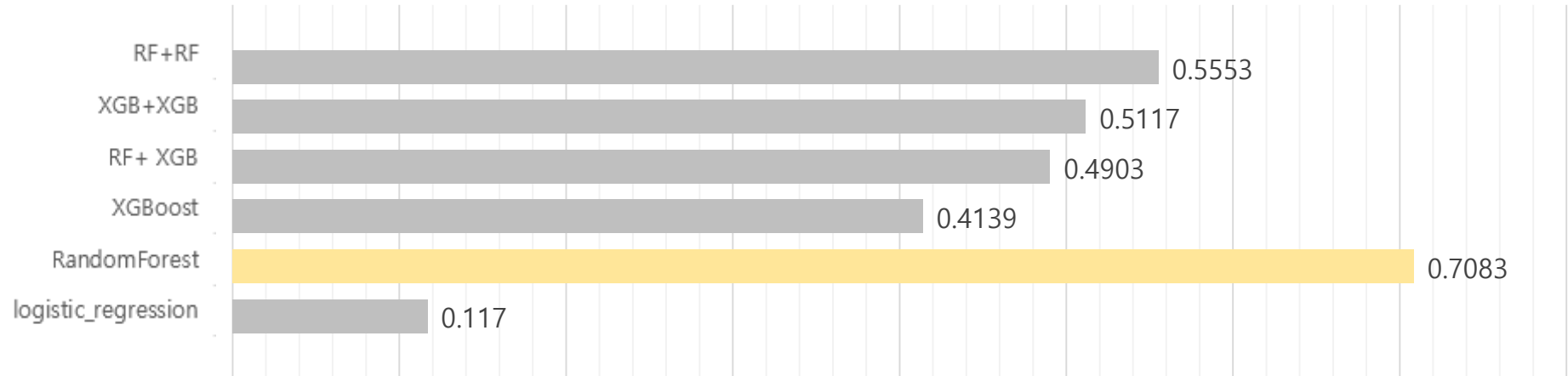
validation
f1-score



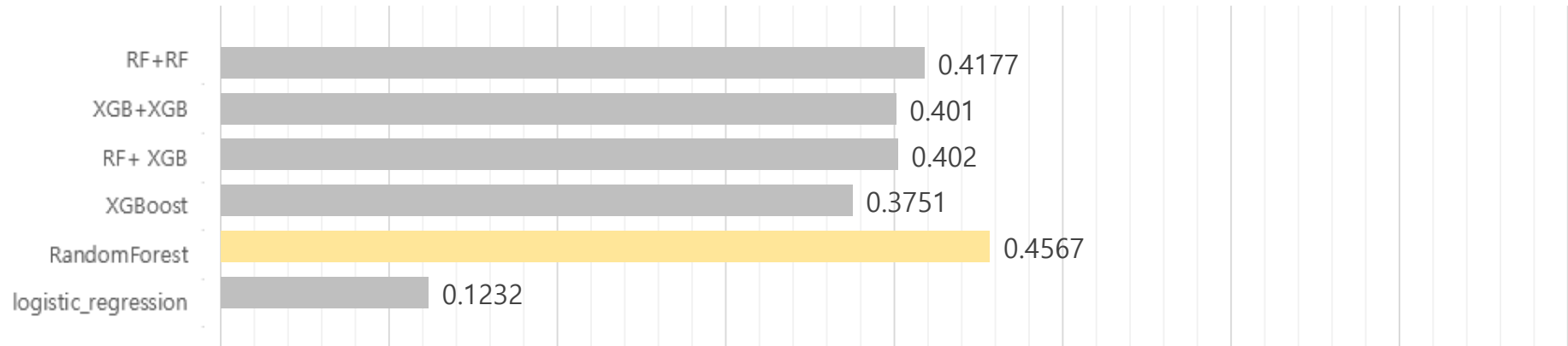
Modeling

- 모델 성능 비교 - 최종 모델 결정(RF) 및 test 데이터 성능

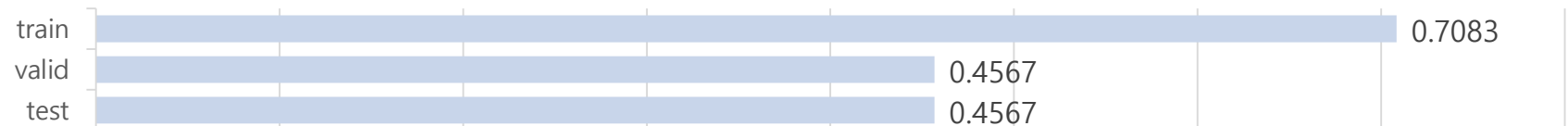
train data
f1-score



valid data
f1-score



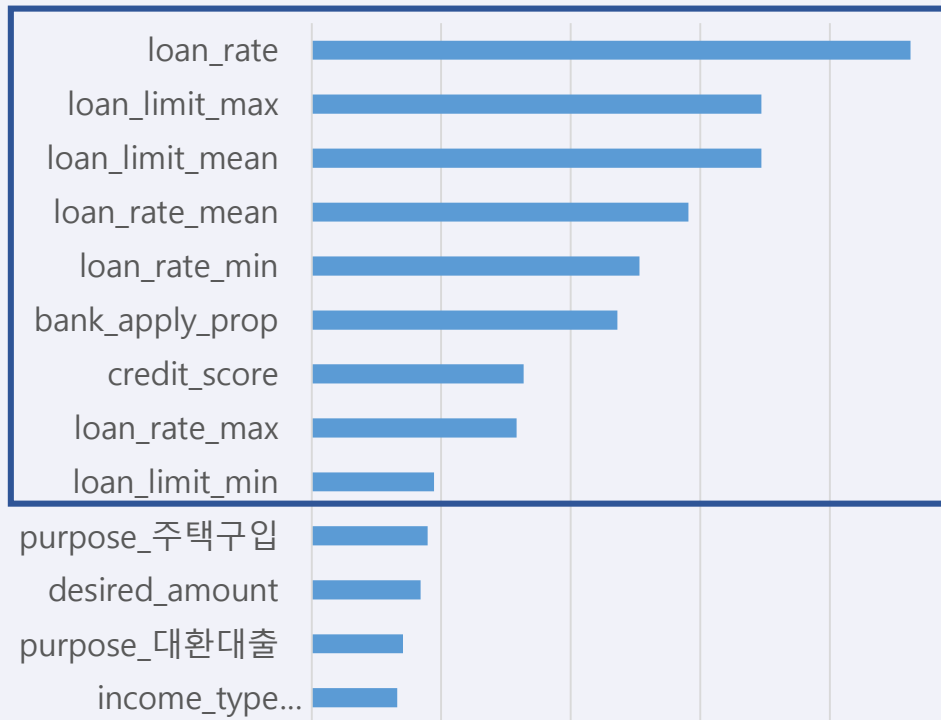
최종 모델 RF 성능



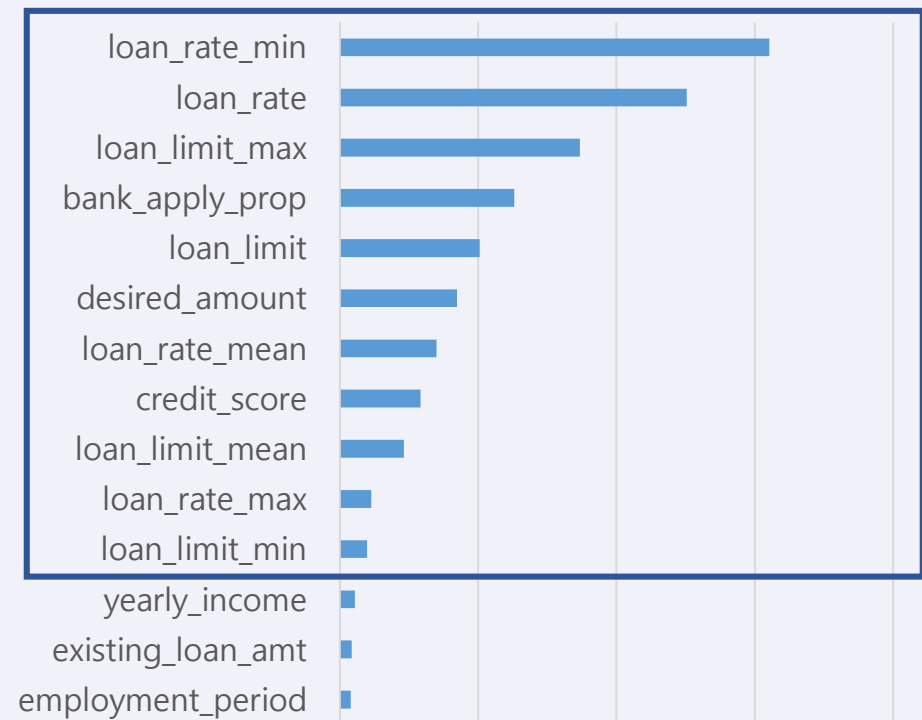
Modeling

- 모델 해석

Logistic Regression Coefficient(abs) (Stepwise)



Permutation Importance (RF)

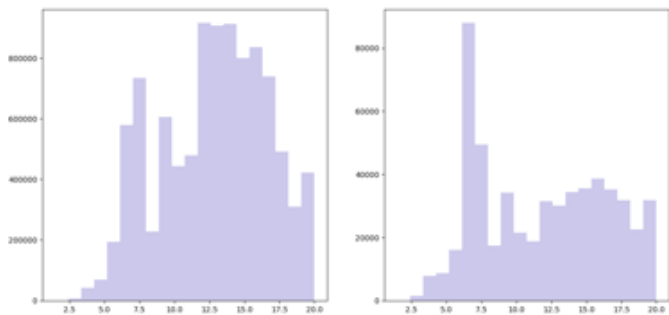


- Linear Regression 의 coefficient 의 절댓값과 Random Forest의 Permutation Importance 의 top 9 변수가 동일하게 나타난다.
- 앞서 생성한 파생 변수가 위의 두 모델 모두에서 예측에 중요한 변수로 나타나는 것을 볼 수 있다.
- 즉, 각 application 간에 정보를 공유하는 파생 변수가 예측 정확도에 영향을 준다고 할 수 있다.

Modeling

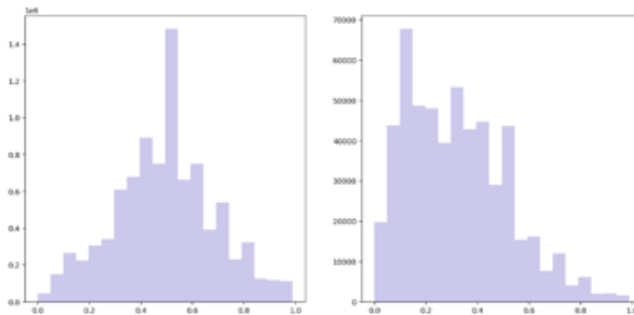
- 모델 해석 - 중요도 상위 6개 변수와 is_applied와의 관계 (좌: is_applied = 0, 우: is_applied = 1)

loan_rate



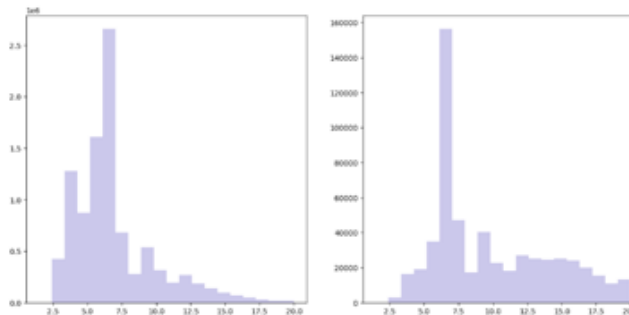
loan_rate가 낮은 상품을 선호한다.

loan_limit_max



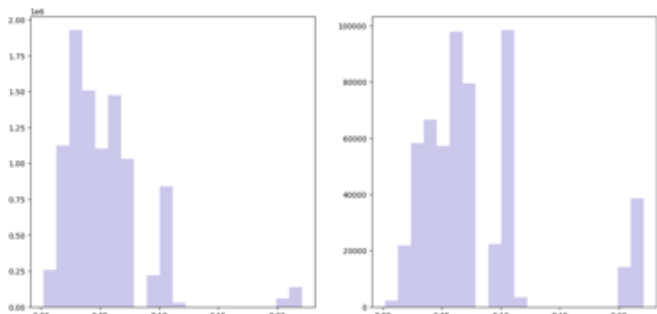
승인된 대출의 한도 최대 금액이 낮은 유저가
대출 신청한 비율이 높음을 볼 수 있다.

loan_rate_min



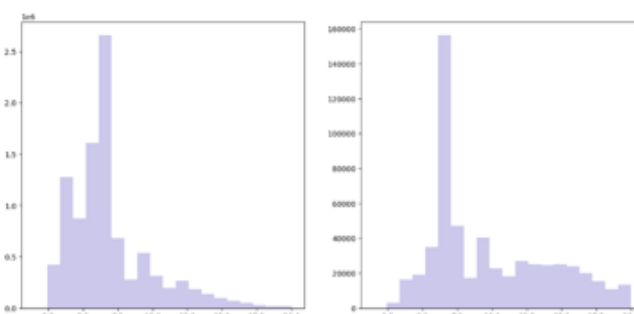
승인된 대출의 최소 금리가 높은 유저가
대출 신청 비율이 높다는 것을 짐작할 수 있다.

bank_apply_prop



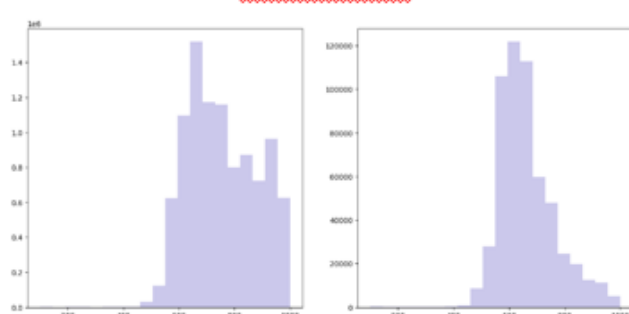
is_applied가 1인 데이터는 대출 신청
비율이 높은 은행에 대출 신청하는 경향이 있다.

loan_rate_mean



승인된 대출의 금리 평균이 높은 유저가
대출을 고려하는 경향이 있다.

credit_score



credit_score가 높은 유저는
대출 신청 비율이 높다는 것을 알 수 있다.

→ 중요도가 높은 변수에서 is_applied 별로 분포가 다르게 나타나는 것을 볼 수 있다.

군집 문제

데이터 전처리

- log data 전처리

event

- Sign up : 회원가입
- Open App : 핀다 앱 실행
- Login : 핀다 앱 로그인
- ViewLoanApplyIntro : 한도조회 인트로 페이지 조회
- StartLoanApply : 한도조회 시작하기 버튼 클릭
- CompleteIDCertification : 본인인증완료
- EndLoanApply : 한도조회 결과 확인
- UseLoanManage : 대출관리 서비스 이용
- UsePrepayCalc : 여윳돈 계산기 서비스 이용
- UseDSRCalc : DSR 계산기 서비스 이용
- GetCreditInfo : KCB 신용정보 조회

- log 형식의 테이블을 각 user_id 당 1개의 행에 존재하도록 bag of events(각 사용자별 event count) 형식의 테이블로 변환

user_id	event	timestamp	date_cd
1469	SignUp
1469	Login
1469	UsePrepayCalc
1673	UseLoanManage
...			
867542	GetCreditInfo



user_id	SignUp	Login	...	GetCreditInfo
1	0	1	...	0
7	0	0	...	1
9	1	2	...	3
11	1	15	...	5
...
879696	0	0	...	3

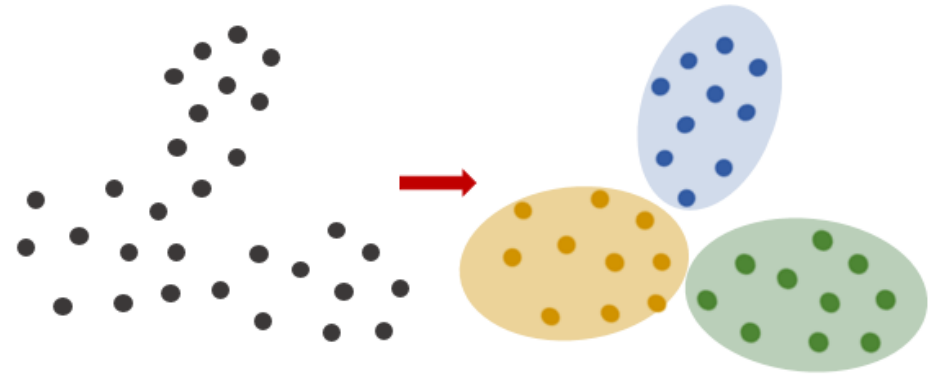
불필요한 데이터 삭제

각 event의 도수

데이터 전처리

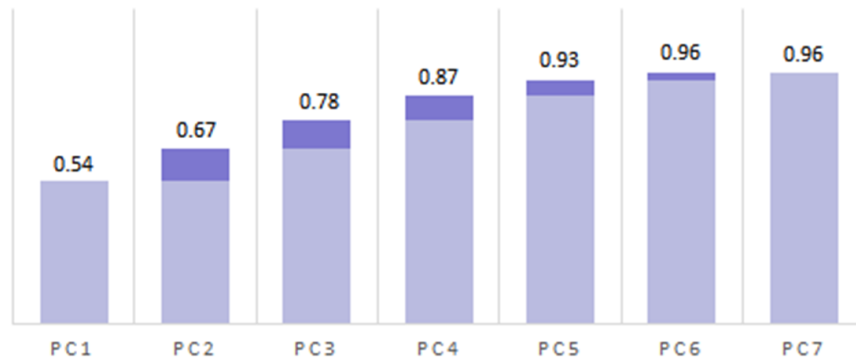
- 사용 군집 알고리즘 : K-means 알고리즘

- 주어진 데이터를 k개의 클러스터로 묶는 알고리즘
- 각 클러스터와 거리 차이의 분산을 최소화하는 방식
- $O(tkn)$ 의 복잡도를 가진 알고리즘으로 빠른 실행이 가능한 알고리즘.
(where n : # of objects, k : # clusters, t : # iterations)

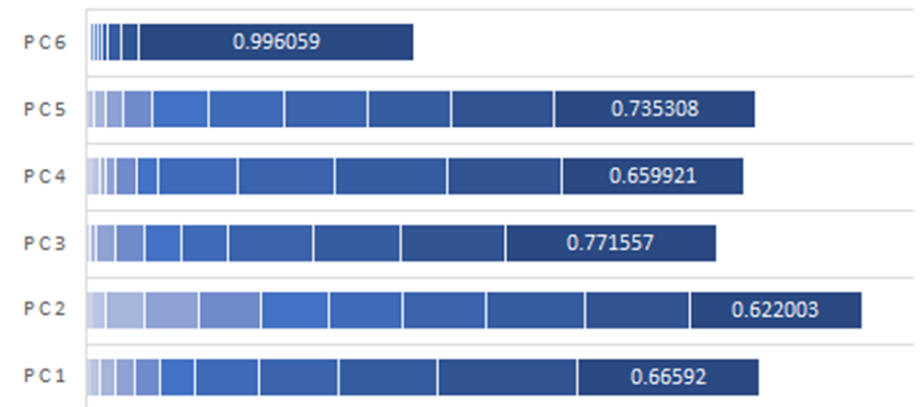


- 데이터 전처리 : 정규화 → PCA

- 거리 기반 군집 알고리즘은 차원이 커질 수록 효과가 떨어지므로 PCA를 통해 차원을 줄이고자 함.
- PC의 개수는 각 PC의 의미와 cumulative variance를 고려하여 PC6까지 사용.
- PCA전 정규화를 진행.



<주성분 별 cumulative variance>

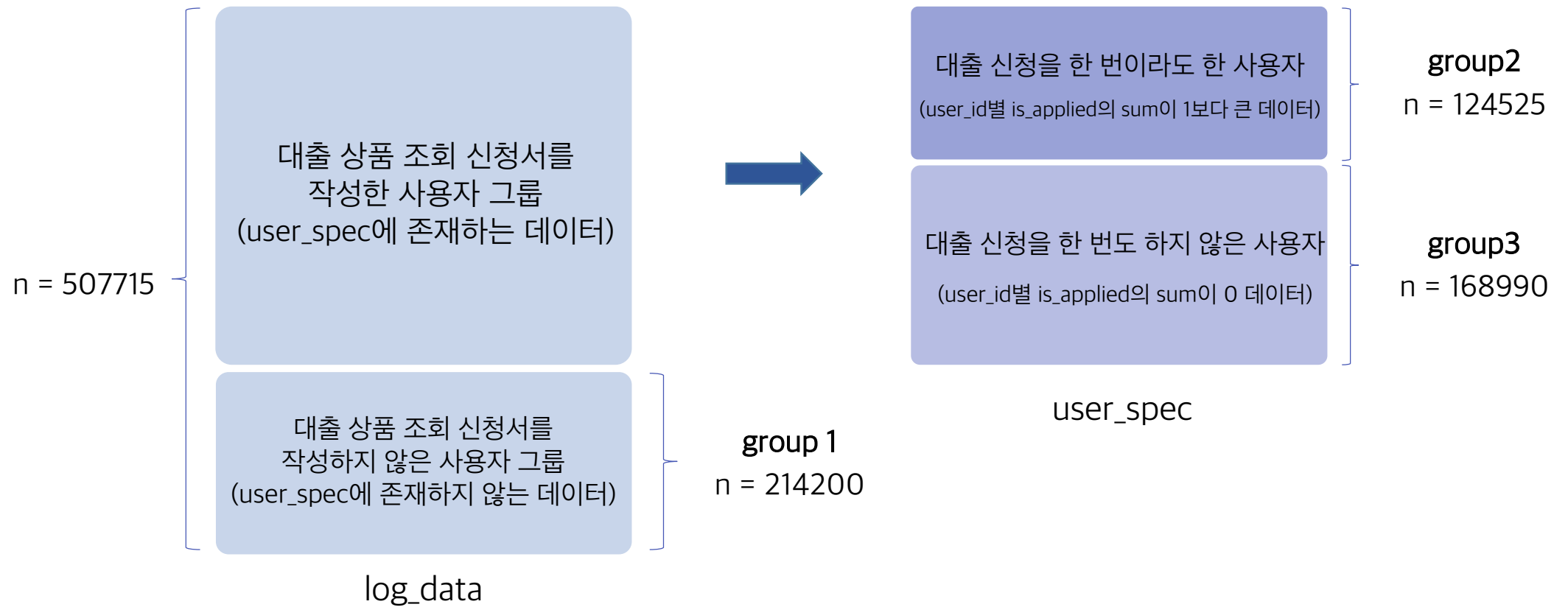


<주성분 별 변수 기여도>

데이터 전처리

- data split

- 대출 신청 미신청 / 신청 고객 별로 다른 특성이 나타나는지를 알아보기 위해 아래와 같이 3개의 그룹으로 분리하여 군집을 각각 진행함.
- 대출 상품 조회 신청서를 작성하지 않은 사용자 / 대출 신청서를 작성하고 대출 신청을 한 번이라도 한 사용자 / 대출 신청서를 작성했지만 대출 신청을 한 번도 하지 않은 사용자 그룹으로 분리.

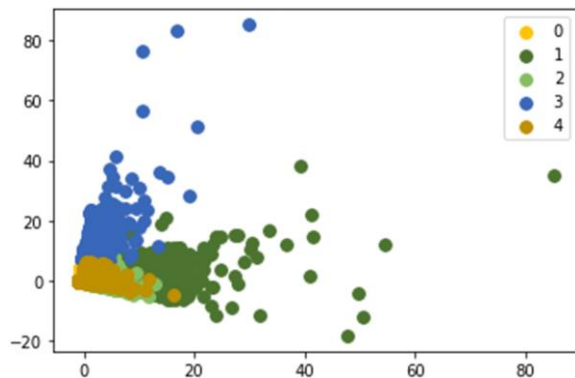
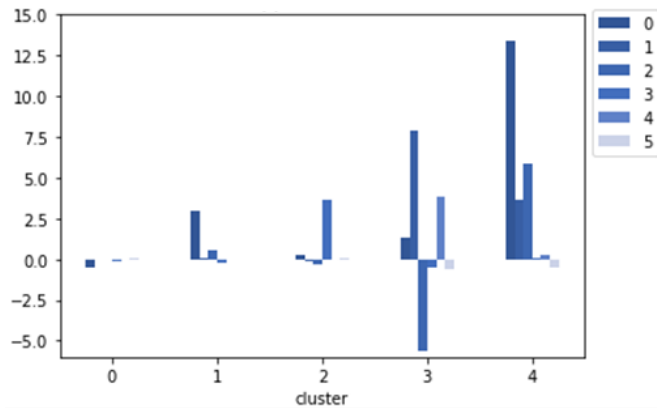


군집 결과

- 군집별 PC값의 평균 및 PC1, 2 에서의 각 군집의 분포 시각화

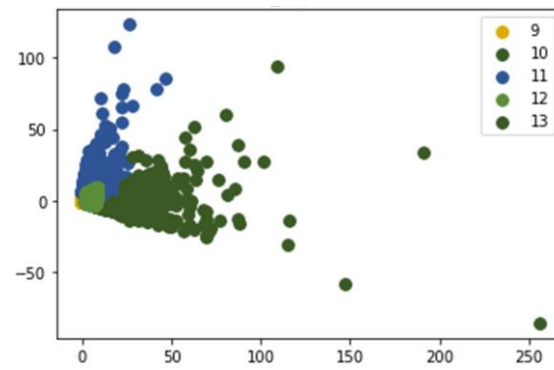
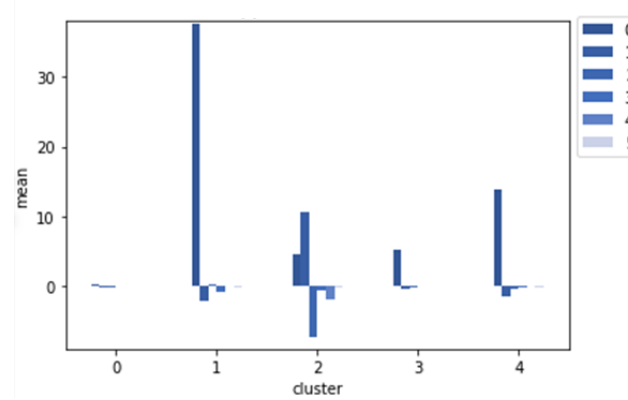
group 1

신청서를 작성하지 않은 그룹



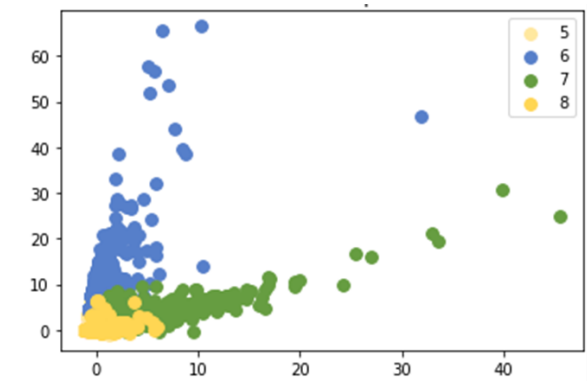
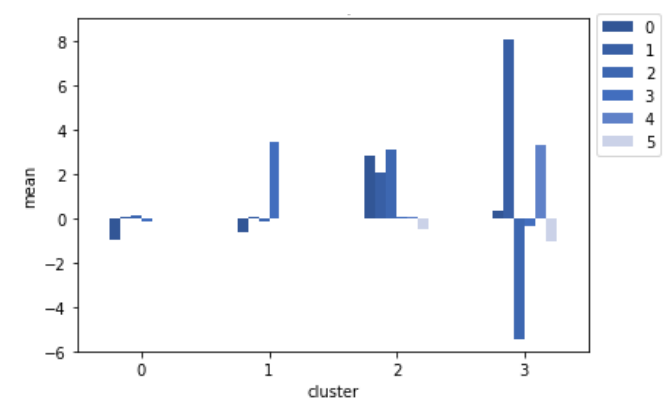
group 2

신청서 작성 후 대출을 신청한 그룹



group 3

신청서 작성 후 대출을 신청하지 않은 그룹



군집 결과

- 군집 결과 해석 : Group1

→ 신청서를 작성하지 않은 그룹

	SignUp	OpenApp	Login	ViewLoanApply Intro	StartLoan Apply	CompleteID Certification	EndLoan Apply	UseDSR Calc	UsePrepay Calc	UseLoan Manage	GetCredit Info
Cluster 1	0	0.0485	0.0381	0.0642	0.0638	0.0473	0.0028	0	0.0020	0.02501	0.0365
Cluster 2	0.906	0.0495	0.0297	0.0936	0.1172	0.6282	0.9435	0.0002	0.0017	0.0446	0.0535
Cluster 3	0.0433	0.7510	0.9117	0.5775	0.5409	0.2220	0.0308	0.0003	0.0183	0.6917	0.7225
Cluster 4	0.0507	0.1510	0.0205	0.2648	0.2781	0.1025	0.0229	0.9995	0.9779	0.2388	0.1876

Cluster 1

신청서를 작성하지 않고, 다른 군집보다 현저히 적은 event를 가지는 것을 보아 신규 User로 추측 가능

Cluster 2

회원가입의 비율이 높고 한도조회를 끝까지 마친 User

Cluster 3

어플을 자주 사용하며 대출 신청 중 신청서 이탈률이 높은 고객

Cluster 4

대출 관련 계산기 기능을 많이 사용하는 대출관리가 목적인 User

군집 결과

- 군집 결과 해석 : Group2

→ 신청서 작성 후 대출을 신청한 그룹

	SignUp	OpenApp	Login	ViewLoanApply Intro	StartLoan Apply	CompleteID Certification	EndLoan Apply	UseDSR Calc	UsePrepay Calc	UseLoan Manage	GetCredit Info
Cluster 1	0.0540	0.0181	0.0174	0.0194	0.0200	0.0198	0.0159	0.0032	0.0023	0.0156	0.0183
Cluster 2	0.1045	0.0793	0.006	0.0848	0.0772	0.0763	0.0677	0.9270	0.9442	0.1651	0.1246
Cluster 3	0.1700	0.0825	0.0833	0.0847	0.0856	0.0840	0.0749	0.0079	0.0063	0.0769	0.0762
Cluster 4	0.2275	0.2160	0.2311	0.2260	0.2240	0.2277	0.2135	0.0129	0.0116	0.1923	0.1950
Cluster 5	0.4440	0.6041	0.6623	0.5850	0.5932	0.5921	0.6280	0.0491	0.0357	0.5502	0.5858

Cluster 1,3

다른 군집에 비해 현저히 적은 log를 가지고 있기에 휴면 또는 활동량이 적은 User

Cluster 2

대출 관련 계산기 기능을 많이 사용하는 대출관리가 목적인 User

Cluster 5

대출을 신청하진 않았지만, 한도조회와 대출서비스를 이용하는 잠재고객으로 판단하여 대출 임박 유저로 분류

군집 결과

- 군집 결과 해석 : Group3

→ 신청서 작성 후 대출을 신청하지 않은 그룹

	SignUp	OpenApp	Login	ViewLoanApply Intro	StartLoan Apply	CompleteID Certification	EndLoan Apply	UseDSR Calc	UsePrepay Calc	UseLoan Manage	GetCredit Info
Cluster 1	0.0468	0.0086	0.008	0.0166	0.0160	0.0190	0.0126	0.003	0.0013	0.008	0.0104
Cluster 2	0.075	0.0328	0.0021	0.0507	0.0403	0.0462	0.0318	0.9517	0.9008	0.0807	0.0546
Cluster 3	0.191	0.0677	0.0681	0.1015	0.1003	0.1057	0.0959	0.0069	0.0051	0.0696	0.0648
Cluster 4	0.2171	0.1981	0.2151	0.2611	0.2603	0.2720	0.2590	0.0112	0.0054	0.2026	0.1958
Cluster 5	0.4701	0.6927	0.7067	0.5701	0.5831	0.5569	0.6006	0.027	0.0874	0.6392	0.6743

Cluster 1,3

다른 군집에 비해 현저히 적은 log를 가지고 있기에 휴면 또는 활동량이 적은 User

Cluster 2

대출 관련 계산기 기능을 많이 사용하는 대출관리가 목적인 User

Cluster 5

대출을 신청하진 않았지만, 한도조회와 대출서비스를 이용하는 잠재고객

서비스 제안

고객 군집별 서비스 제안



휴먼 or 활동량이 적은 유저

Group 2-1, Group 2-3,
Group 3-1, Group 3-3



계산기 사용 유저

Group1-4, Group2-2,
Group3-3



신규 유저

Group1-1



대출 관심 유저

Group1-2, Group1-3,
Group 2-5, Group 3-5



대출 임박 유저

Group2-5

고객 군집별 서비스 제안

- 핀다의 홈 화면의 기능을 설명해주는 매뉴얼 서비스를 제공



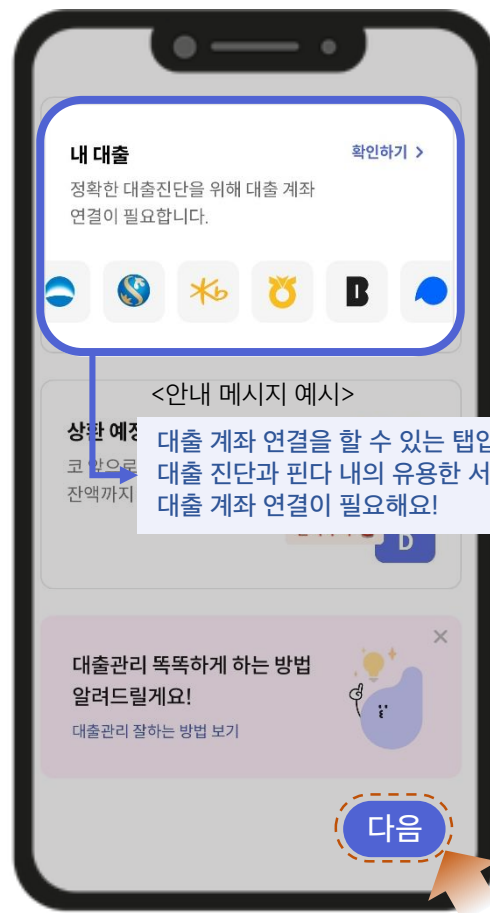
휴먼 or 활동량이 적은 유저



신규 유저

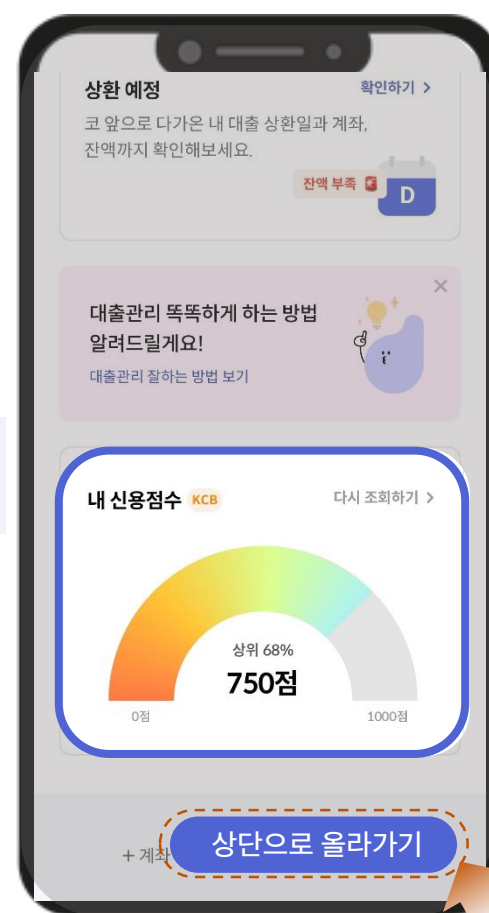


1. 상단에 '홈 화면 빠르게 살펴보기' 버튼을 배치한다.



다음 스크롤로 넘어간다

2. 버튼을 클릭하면 홈 화면이 한 배너 씩 스크롤되며 안내 메시지와 함께 홈 화면에 배치된 기능들의 사용법이 설명된다.



가장 상단으로 올라간다



3. 가장 마지막에 상단으로 올라가기 버튼을 누르면 상단 페이지로 이동되고, 대출 한도 조회 신청서 작성을 유도하는 배너를 배치한다.

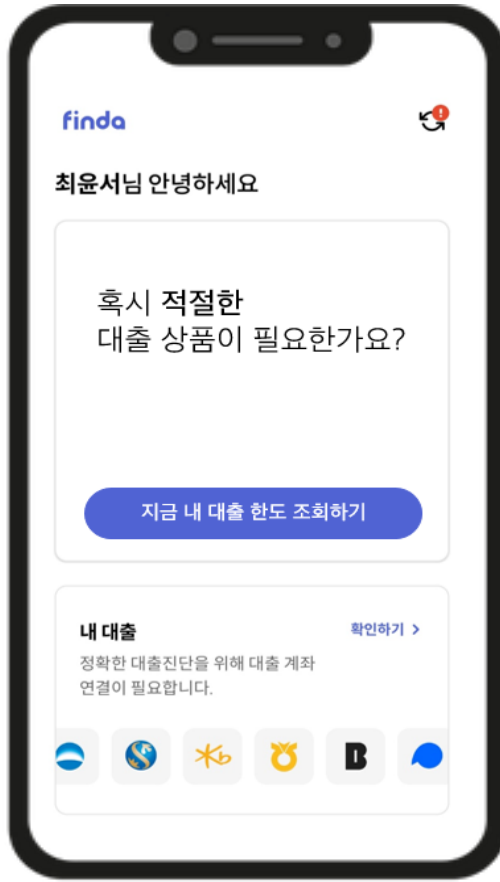
고객 군집별 서비스 제안



대출 관심 유저

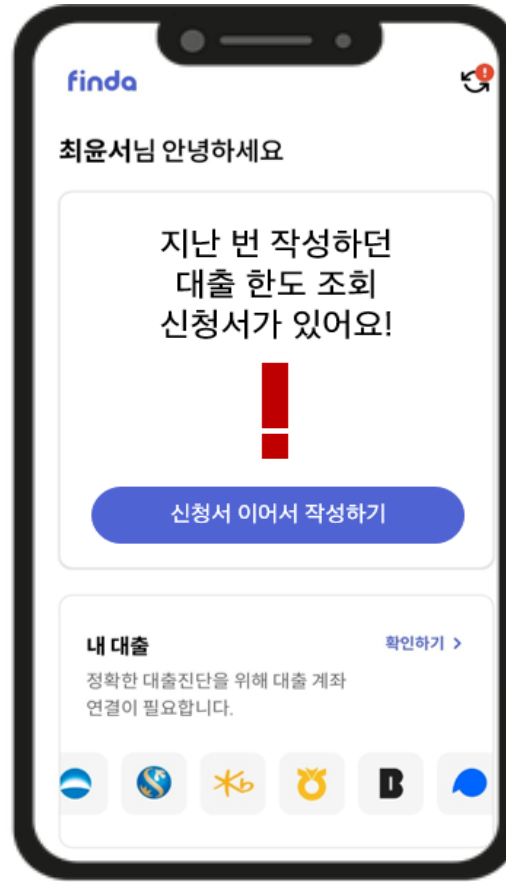
- 대출 관련 정보를 담은 배너를 홈 화면 최상단에 배치

group1 - 2



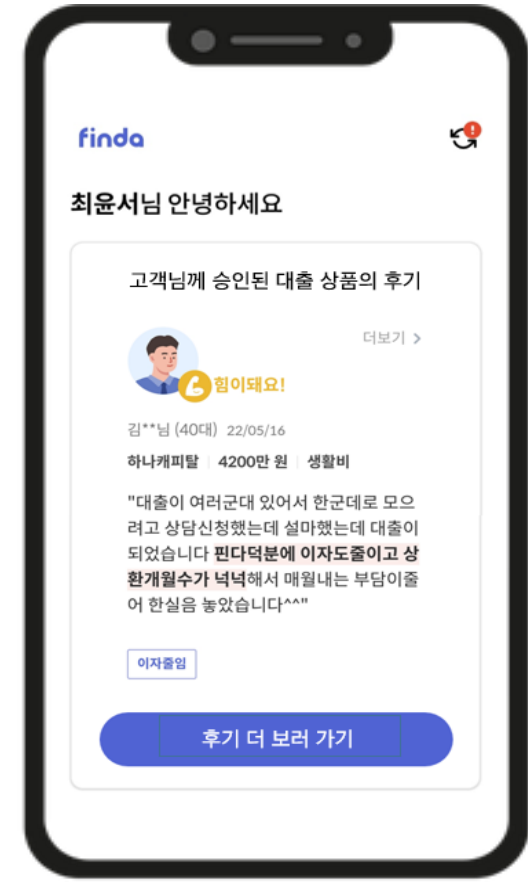
고객이 승인된 대출 상품 목록을 볼 수 있도록 대출 한도 조회 신청서 작성을 유도하는 배너 배치

group1 - 3



신청서 작성 중 이탈했던 고객은 다시 이어서 신청서를 작성할 수 있도록 하는 배너 배치

group2-5, group3-5



승인된 대출 목록 중 대출 승인 예측 모델을 활용하여 대출 상품을 추천하고 관련 후기를 노출시키는 메시지 배치

고객 군집별 서비스 제안



계산기 사용 유저



대출 임박 유저

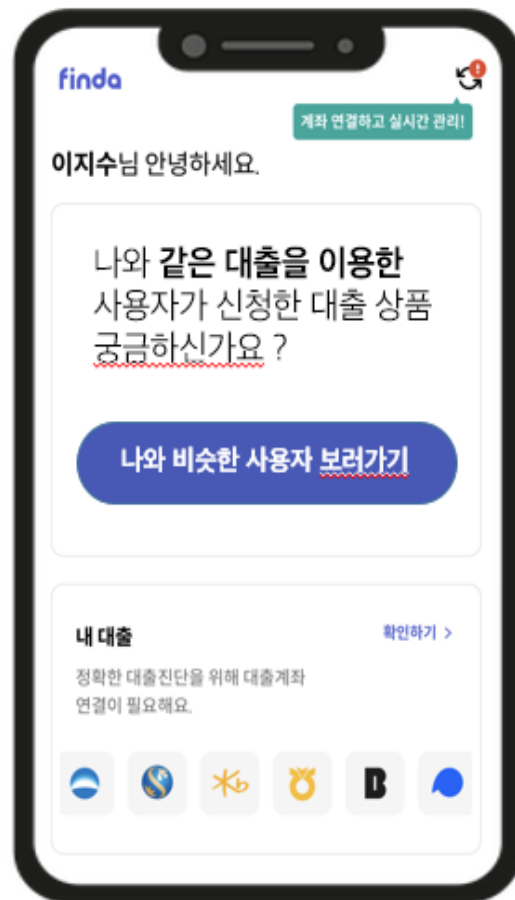
- 대출 관련 정보를 담은 배너를 홈 화면 최상단에 배치

group1-4, 2-2, 3-3



대출관련 계산기를 이용하는 고객에게 메인화면에 자주 이용하는 서비스 바로가기 배치

group3-2



기존 대출 신청 이력이 있는 고객에게 유사 사용자가 과거에 신청한 상품 추천 메시지 배치

문제 1) 대출신청 고객 예측

1. 결측치대체

- EDA를 통해 MNAR(비무작위 결측)과 MAR(무작위 결측)을 가정하고, 각 결측치에 적절한 방법으로 처리함.

2. 데이터의 특성을 고려한 파생변수 생성

- application_id간 독립이 아닌 점을 고려하여 각 행 간의 상관성을 반영하기 위한 loan_limit, loan_rate의 min, max, mean 변수가 예측 성능에 도움을 줌.
- practical하게 반영할 수 없는 bank_id의 다범주 데이터를 수치형으로 변형함으로써 같은 정보를 표현할 수 있도록 전처리함.

3. 훈련, 검증, 테스트 데이터 간의 독립성 유지

- train, valid, test 데이터를 분할할 때 단순 index가 아닌 application_id를 기준으로 분리함으로써 각 데이터셋이 독립성을 유지할 수 있도록 노력하였으며, 그 결과 신뢰성이 높은 검증을 수행할 수 있었음.

문제 2) 앱 사용자 데이터에 따른 모델 기반 군집 분석 및 서비스 메시지 제안

1. 고객의 신청서 작성 / 대출 신청 여부 별 그룹을 나누어 군집을 진행

- 전체 사용자 데이터를 일괄적으로 군집한 것이 아닌 대출 한도 조회 신청서 작성 / 대출 신청 여부를 기반으로 한 단계 먼저 사용자를 분리하고 군집을 진행함으로써 군집에 대한 좀 더 폭넓은 해석이 가능해짐.
- 이에 따라 사용자의 상황에 맞은 더 구체화된 서비스 제안이 가능해졌다.

2. PCA를 통한 차원 축소

- 거리 기반 군집 알고리즘인 kmeans를 사용한다는 점을 고려하여 PCA를 통해 차원을 축소함으로써 좀 더 식별가능한 군집 결과를 얻을 수 있었다.

활용 방안 및 기대효과

- 고객의 log data만을 활용하여 고객의 현재 니즈를 파악할 수 있고, 이에 맞는 서비스 메시지를 제안할 수 있다.
- 고객 개인 맞춤 서비스를 제공함으로써 고객의 만족도를 높이고 어플 체류 시간을 증가시킬 수 있다.
- 예측 모델과 군집 결과를 함께 사용하여 대출 신청 잠재 고객을 타겟팅할 수 있다.

감사합니다