2023 빅콘테스트 제 11회 (BIG-DATA)

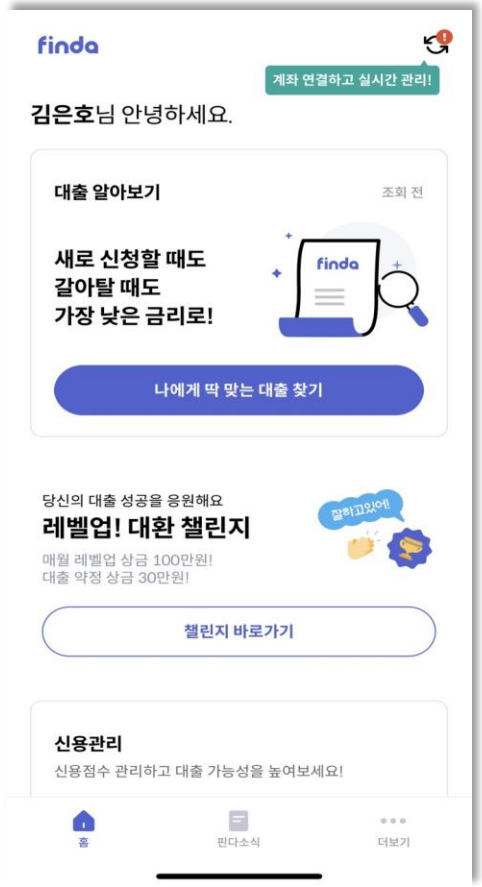
생성형 AI 분야 - 생성형 AI를 활용한 데이터 분석 자동화 및 레포트 제출

팀명: NaN괜찮아

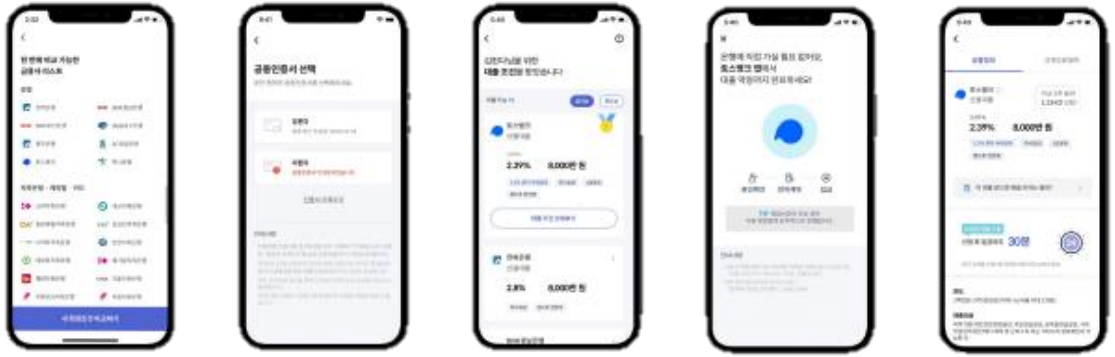
팀원: 김은호 (eunho9703@gmail.com)
김희경 (human09129@sookmyung.ac.kr)
이지수 (jisujisu1012@naver.com)
이채윤 (caron1517@naver.com)

세상에 없던 대출비교 플랫폼 finda

finda



1. 비교대출 서비스



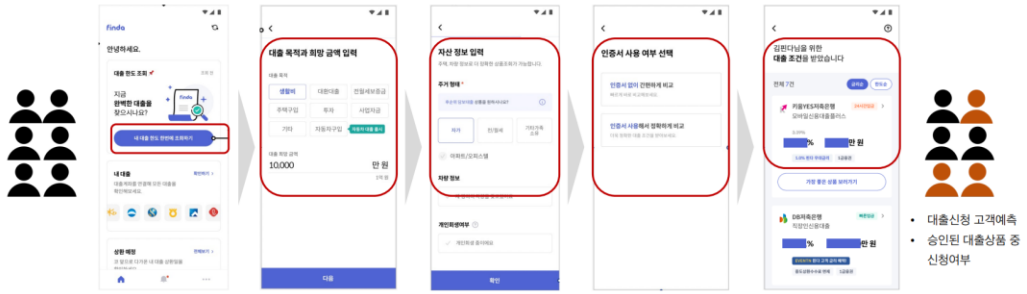
2. 나의 대출관리

문제 정의

1.

대출신청 고객 예측을 위한 자동화 분석서

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



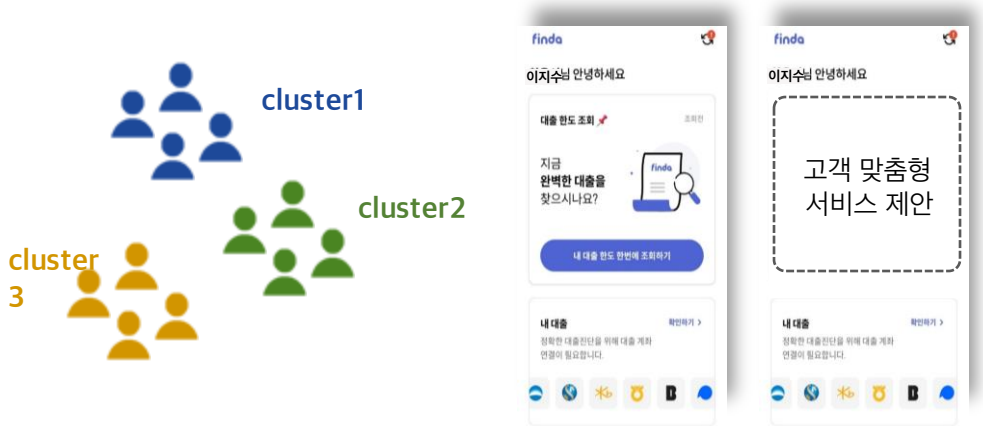
→ 대출 신청 고객의 특성을 자동화된 분석서로 작성하여 분석 업무의 자동화 및 고객 맞춤형 서비스 기반 자료 확보 가능

→ 고객의 대출 신청 여부를 예측하여 고객이 신청할 가능성이 높은 대출을 고객에게 노출시켜 서비스 향상을 도모할 수 있음.

2.

고객 군집 분석 , 군집별 서비스 메시지 제안 자동화 분석서

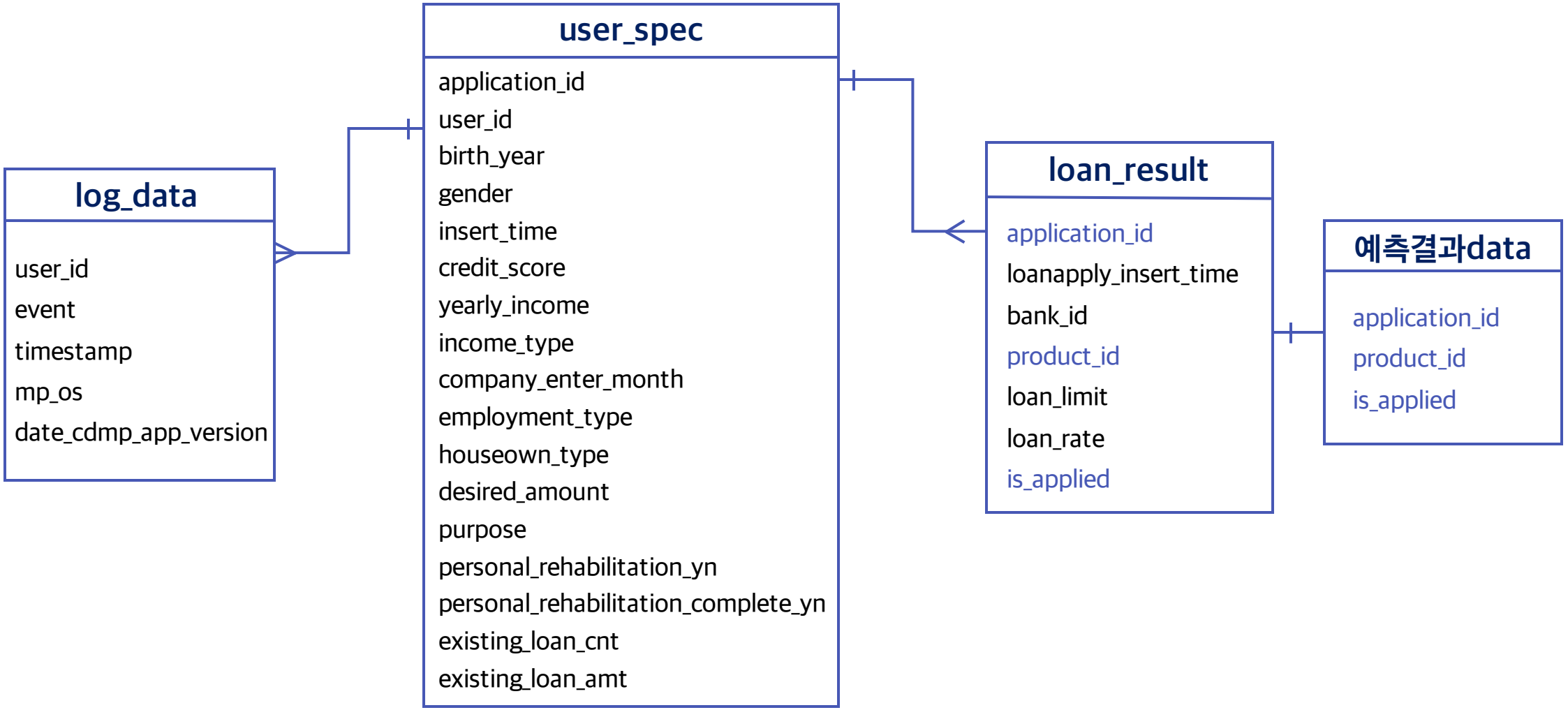
- 사용자 사용 기록(log data)을 이용한 군집 결과를 바탕으로 앱 메인 화면 상단에 노출될 서비스 메시지를 제안.



→ 군집 분석 결과를 통해 어플의 첫 화면을 개인화하여 고객에게 맞춤형 서비스 제안

→ 군집의 특성을 반영한 자동화된 메시지 제안을 통해 핀다의 서비스 질 향상과 효과적인 서비스 홍보를 기대할 수 있음.

제공 데이터


user_spec
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
personal_rehabilitation_yn
personal_rehabilitation_complete_yn
existing_loan_cnt
existing_loan_amt
log_data
user_id
event
timestamp
mp_os
date_cdmp_app_version
loan_result
application_id
loanapply_insert_time
bank_id
product_id
loan_limit
loan_rate
is_applied
예측결과data
application_id
product_id
is_applied
군집문제
예측문제
검증

자동화 분석서 - 생성형 AI 모델

자동화 분석서

예측 모델 개발을 위한 데이터 분석의 각 단계를 모두 생성형 AI를 활용,

진행한 각 단계의 과정 또한 생성형 AI 모델을 활용하여 예측 보고서로 작성

(예측/군집 나누어 발표자료 첨부)

생성형 AI 모델

gpt 3.5 turbo

GPT 3.5 모델 중 가장 성능이 뛰어나고 최적화되어 있음,

davinci - 003에 비해 비용 1/10,4096개까지 토큰 수행가능

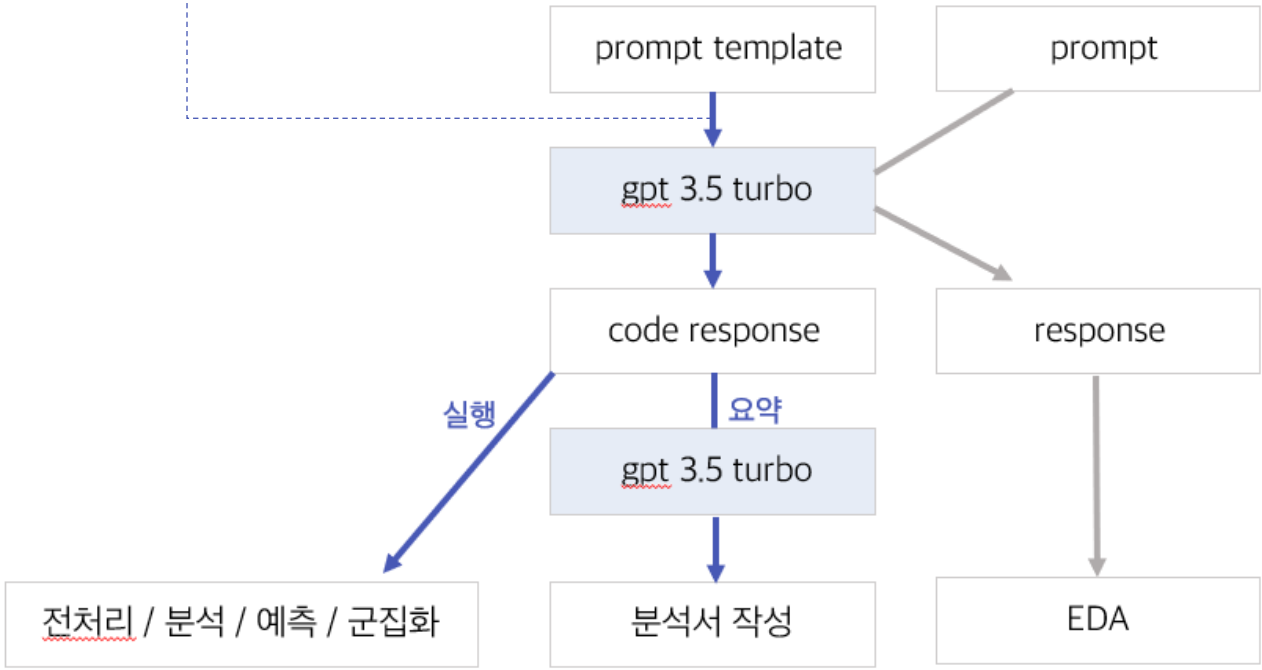
- Transformer 아키텍처를 활용하여 텍스트 시퀀스를 입력으로 받고, 해당 시퀀스에 대한 문맥을 인식해 자동 답변 생성
- 자연어의 어휘, 문법 및 의미론적 관계를 이해하고, 이를 기반으로 응답 생성하여 자연어 질문 응답, 텍스트 생성, 번역, 대화 생성 등의 작업에서 뛰어난 성능을 보인다.

OpenAI - gpt 3.5 turbo

few shot learning :

gpt 모델이 부연 설명 없이 code 자체만을 결과값으로 출력하도록 학습,

프롬프트문 입력 후 해당 모델이 생성한 code를 실행하여 각 단계 진행



생성형 모델

OpenAI - gpt 3.5 turbo

few shot learning

code_examples

= [{ **“question”:**

logistic regression classification 코드 출력

“ answer”:

"""from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
… ”””} …]

code_example_prompt

= [{ **“PromptTemplate**(template =

"My Question : {question}

AI Answer : {answer}",

input_variables = ["question", "answer"])

few_shot_code_prompt

= **FewShotPromptTemplate** (

examples = code_examples,

example_prompt = code_example_prompt,

suffix ="Question : {input}",

input_variables = ["input"])

LLM Chain

llm = OpenAI (temperature=0, model_name = "gpt-3.5-turbo")

code_chain = **LLMChain** (llm=llm,prompt=few_shot_code_prompt)

question = """ *전처리 요청 프롬프트문…* “””

code_response = code_chain.run (question)

exec (code_response)

예측

데이터 전처리 - USER_SPEC - 데이터 소개

- 유저 스펙 테이블 (User_Spec)

birth_year	*일반정보*
gender	
company_enter_month	*고용정보*
employment_type	
credit_score	*소득정보*
yearly_income	
income_type	
houseown_type	
desired_amount	*대출관련정보*
purpose	
personal_rehabilitation / complete_yn	
existing_loan_cnt / amt	

age	gender	employment_period	employment_type
37	1	7	기타
54	1	15	정규직

insert_time - birth_year

insert_time - company_enter_month

4개 범주 (one-hot encoding)

credit_score	yearly_income	income_type	houseown_type
660	108000000	PRIVATEBUSINESS	자가
870	30000000	PRIVATEBUSINESS	기타가족소유

6개 범주 (one-hot encoding) ***4개 범주 (one-hot encoding)***

desired_amount	purpose	existing_loan_cnt	existing_loan_amt
1000000	기타	4	162000000
30000000	대환대출	1	27000000

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 변수 결측치 처리

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

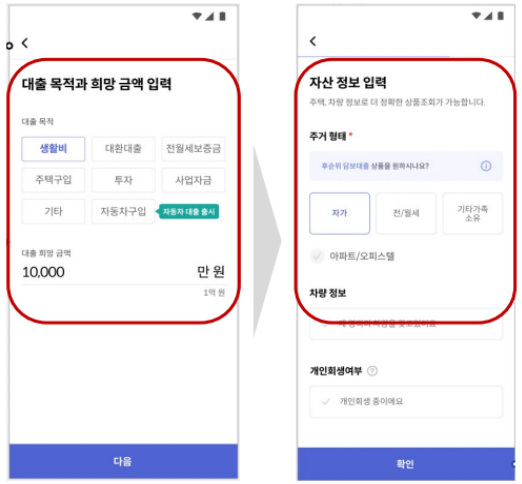
EDA) 결측이 같은 사용자에게서 동시에 발생

step 1 신청서를 불성실하게 작성한 이용자의 데이터로 판단

→ 신뢰성이 떨어지는 데이터로 row data 삭제

step 2 step1 과정 후에 남은 데이터의 yearly_income의 결측

→ 동일 user_id의 yearly_income 값의 평균으로 대체



사용자가 직접 작성한 신청서 정보를 저장하는 방식으로 데이터가 수집된다

existing_loan_cnt	*198556*
existing_loan_amt	*313774*

기존 대출 여부 관련 정보 기입을 skip 한 user라고 판단

→ cnt, amt 가 모두 nan 인 행은 두 컬럼 값 모두 0으로 대체

gender	*12961*

성별에 따라 대출 여부에 유의미한 차이 X

→ 변수 제거

company_enter_month	*171760*
birth_year	*12961*

같은 user의 경우 동일한 출생연도 및 입사일을 가질 것으로 판단

→ 동일 user_id의 birth_year 값 및 company_enter_month 값으로 대체

데이터 전처리 - USER_SPEC - 결측치 처리

- credit_score 결측치 처리

credit_score	*105115*

→ **사용자별 credit_score의 unique한 값의 개수에 따른 결측치 처리**

Case 1) unique한 값이 1개인 경우
user_id별로 groupby 후 transform('first') 적용
→ 같은 user_id의 **첫번째 값**으로 대체

Case 2) unique한 값이 2개인 경우
user_id별로 groupby 후 transform('mean') 적용
→ 같은 user_id의 **평균값**으로 대체

Case 3) unique한 값이 존재하지 않는 경우
' XgboostRegressor ' 를 이용한 결측치 예측 수행

- age 결측치 처리

age	*9723*

step 1 'cluster_age' 생성 - 나이를 **범주화하여 생성된 연령대변수**

age	Cluster_age
20대 미만	*0*
20 대 이상 30대 미만	*1*
30대 이상 40대 미만	*2*
⋮ →	⋮
80대 이상 90대 미만	*7*
90대 이상	*8*
NaN	*9*

step 2 'cluster_age' 결측치 대체 -

' XgboostClassifier '를 이용한 결측치 예측 수행

step 3 'age' 결측치 대체

```
user_spec['age'].fillna(20 + 10 * user_spec['cluster_age'])
```

데이터 전처리 - USER_SPEC - 결측치 처리

- 수치형 변수 결측치 처리

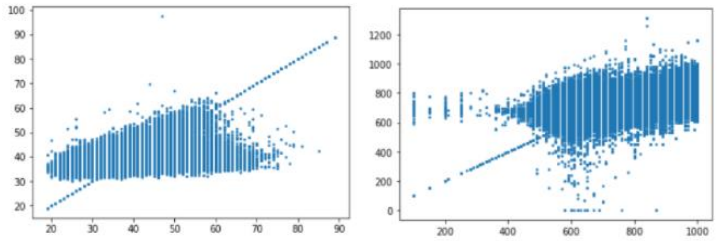
existing_loan_amt	*115149*
employment_period	*118778*

EDA 및 결측치 처리 방법

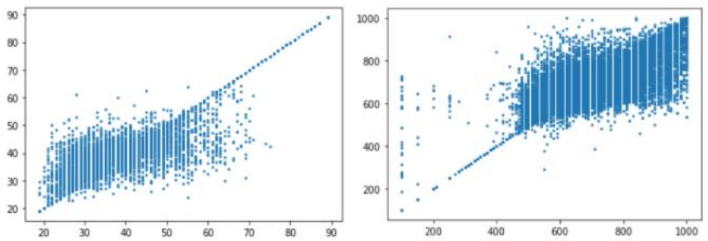
결측 데이터와 기존 데이터의 분포가 크게 차이 나지 않음

→ 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

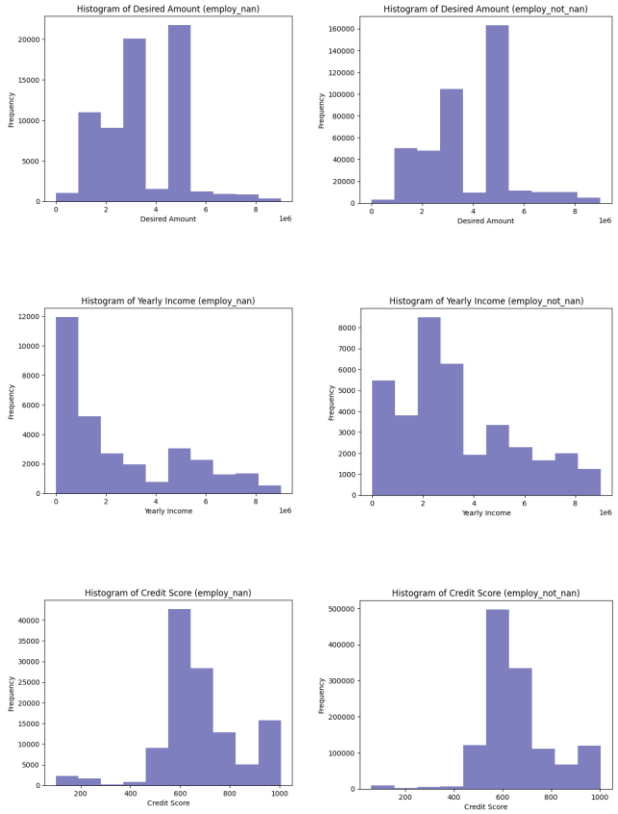


<linear regression 적용 시의 quantile plot>



<ExtraTree 적용 시의 quantile plot>

[employment period 결측 여부에 따른 변수별 분포]



결측X 결측O

데이터 전처리 - USER_SPEC - 파생변수 생성

- 개인회생 관련 파생변수 생성

rehabilitation | *범주형 파생변수 (해당없음 / 진행중 / 회생완료 / 모름)*

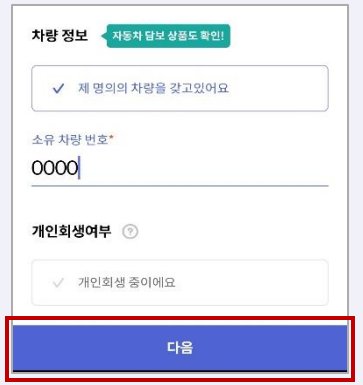
personal_rehabilitation_yn	*587351*
personal_rehabilitation_complete_yn	*1203174*

개인회생여부

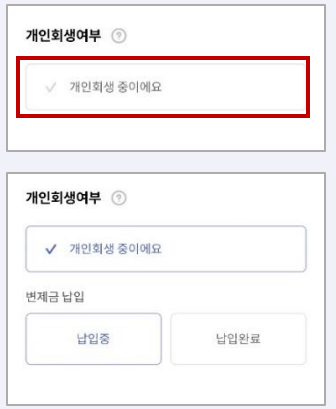
개인회생상환여부	0	1
0	178139	11356
1	4	1344

EDA 및 데이터 수집 과정 분석

- (0,0)의 조합이 가장 많은 경우로 나타남
- 카이 제곱 검정 결과 해당 파생변수의 유의성 증명
- 개인회생여부 여부에 체크를 하지 않더라도
  다음 페이지로 넘어갈 수 있는 사실 확인
- 논리적인 오류가 있는 (0,1) 혹은 (0, nan)은
  개인 회생 여부를 체크하지 않은 것에
  비중을 두고 해당 없는 user라고 판단하였음



개인회생여부를 체크하지 않아도 다음 버튼이 활성화 됨.



개인회생여부를 체크하지 않으면 변제금 납입 체크창이 활성화되지 않는다.

파생변수 생성 과정

- (0,0), (0,1), (0,nan) : 개인회생 신청 X → " 해당없음 "
- (1,0) : 개인회생 신청 O, 변제금 납입 X → " 진행중 "
- (1,1) : 개인회생 신청 O, 변제금 납입 O → " 회생완료 "
- (nan,nan) : 알 수 없음 → " 모름 "

데이터 전처리 - LOAN_RESULT - 데이터 소개

- loan_result 테이블

bank_id	*상품 식별 아이디*
product_id	
loan_limit	*대출 상품 정보*
loan_rate	
is_applied	*target 변수*

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

- bank_cluster 파생변수 생성

bank_id	**은행별 식별 번호** *(총 63개의 범주)*
product_id	**상품별 식별 번호** *(총 270개의 범주)*

→ *one-hot encoding 시 늘어난 변수 개수 문제로 메모리 부족, 계산 시간 오버 문제가 발생*

bank_cluster *파생변수*

- 은행별로 금리와 한도가 상이한 것을 확인하였으며, 은행사에 따른 고객의 선호도 반영 목적
- **loan_limit, loan_rate** 기준 **Kmeans (k=5)** 로 'bank_id' 클러스터링 , 'bank_cluster' 변수 생성
- product_id의 경우 사용자가 대출을 신청할 시 개설 은행의 영향은 많이 받을 수 있지만 대출 상품(금리, 한도를 제외한 상품명 등)의 영향은 많이 받지 않을 것이라는 판단으로 변수 제거
  → 실제로 product_id는 feature importance 측정 시 중요 변수로 추출되지 않음.

데이터 전처리 - LOAN_RESULT - 파생변수 생성

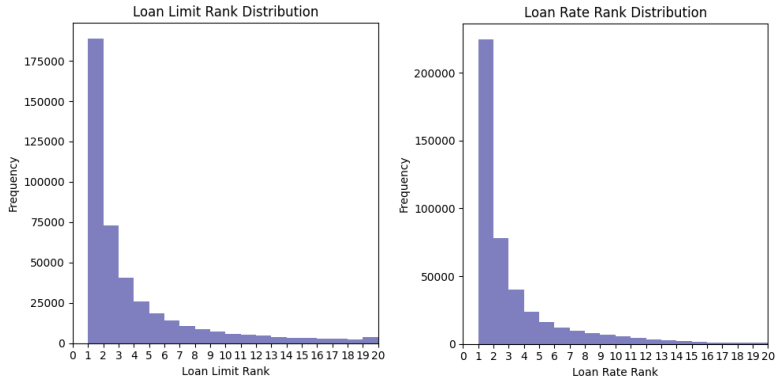
- loan_limit, loan_rate 비교 우위 파생변수 생성

loan_rate_per	$\frac{\text{loan\_rate\_min}}{\text{loan\_rate}}$
loan_limit_per	$\frac{\text{loan\_limit}}{\text{loan\_limit\_max}}$

- 동일한 application_id의 대출 상품 사이의 우위를 비교 및 반영하기 위해 application_id별로 groupby 후 min(loan_rate)과 max(loan_limit) 추출
- 0 - 1 사이의 값을 가지며, 1에 가까울수록 우위 상품 (낮은 금리, 높은 대출 한도)

EDA) 승인 한도는 높을수록, 금리는 낮을수록 대출을 신청하는 경향이 존재함을 우측 그래프 rank distribution을 통해 확인

- Feature importance 결과 생성한 두 파생변수가 가장 중요도가 높게 측정
- t-test 검정 결과 두 변수 모두 대출 여부에 대하여 유의한 변수임을 확인
- 파생변수 추가 후 valid set의 f1-score가 0.052 향상되었음



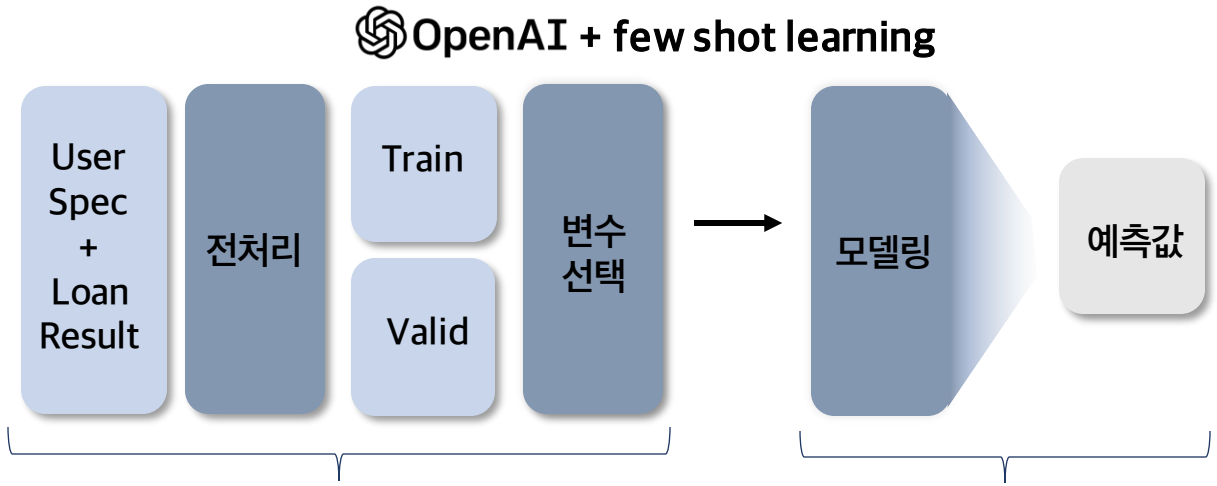
application_id	loan_limit_per	loan_rate_per	loan_limit	loan_rate	loan_limit_max	loan_rate_min
1718340	1	1	42000000	13.6	42000000	13.6
1718340	0.238	0.755	10000000	18.0	42000000	13.6
1718340	0.5	0.918	21000000	14.8	42000000	13.6
…	…	…	…	…	…	…

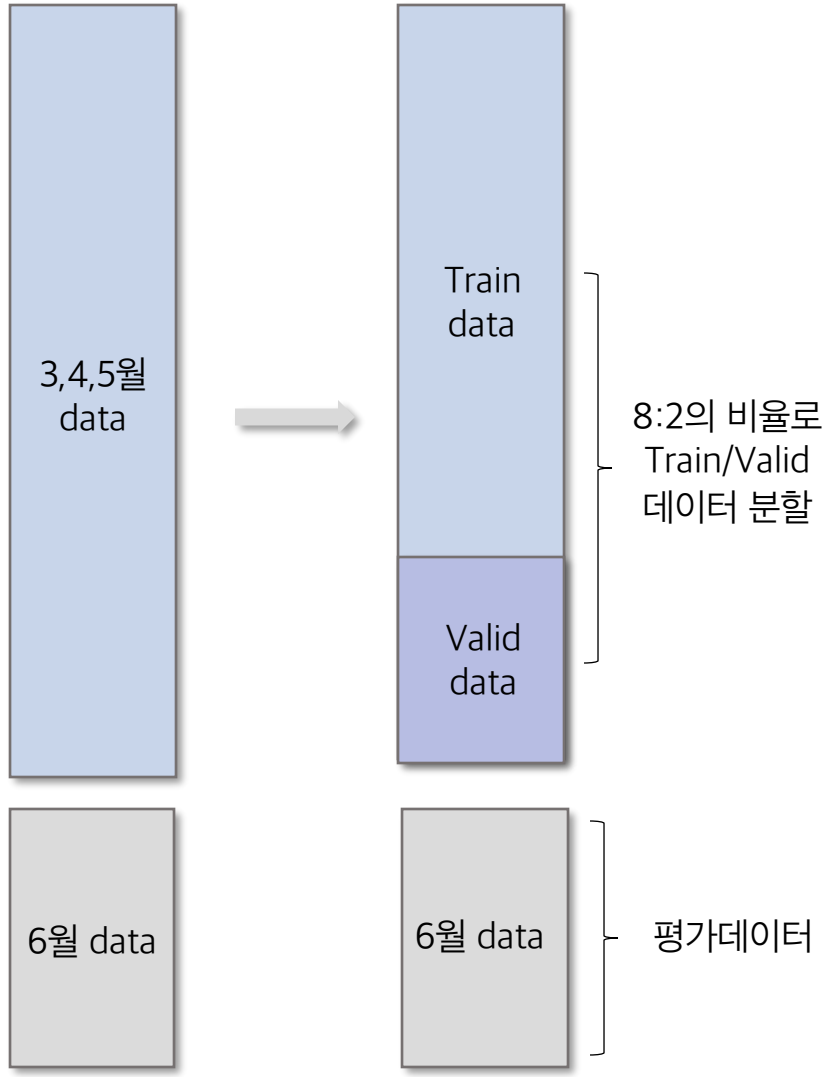
Data Split

- **application_id를 index로 한 Data Split**

→ application_id를 기준으로 user_spec과 loan_result 데이터 병합
→ 데이터의 독립성을 보존하기 위해 application_id 기준 데이터 분할

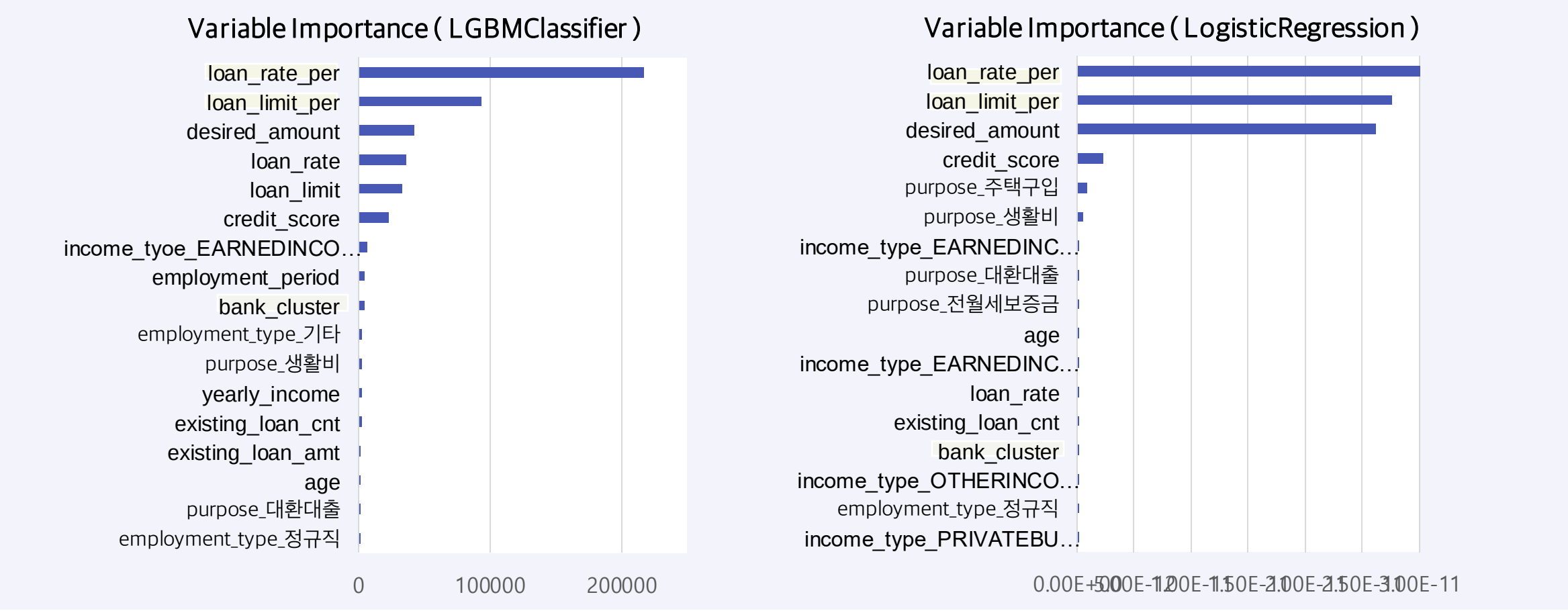
- **예측 흐름도**





Modeling

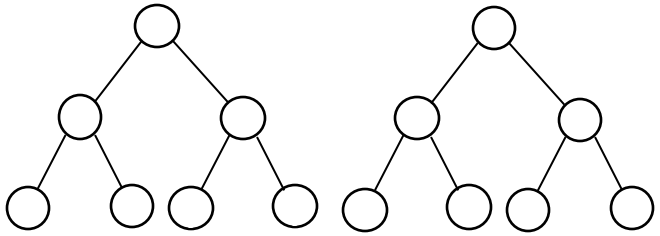
- 변수 선택



- 변수중요도가 높은 20개의 변수를 선택하여 모델링에 사용하였다.
- 생성한 파생 변수 (loan_rate_per , loan_limit_per , bank_cluster)가 위의 모델에서 예측에 중요한 변수로 나타나는 것을 확인할 수 있다

Modeling

RandomForest



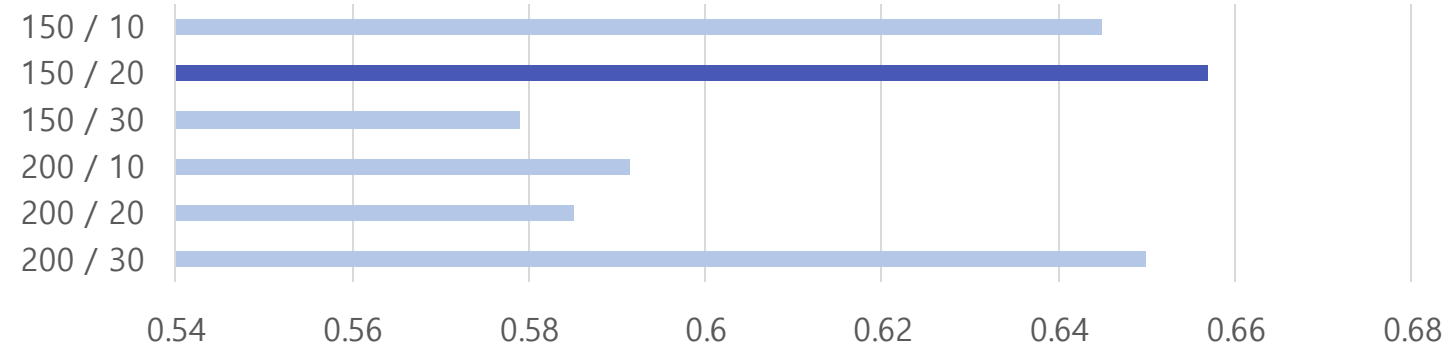
n_estimators = 150
min_samples_split = 20
class_weight = "balanced"

Train f1-score	Valid f1-score
0.656	0.457

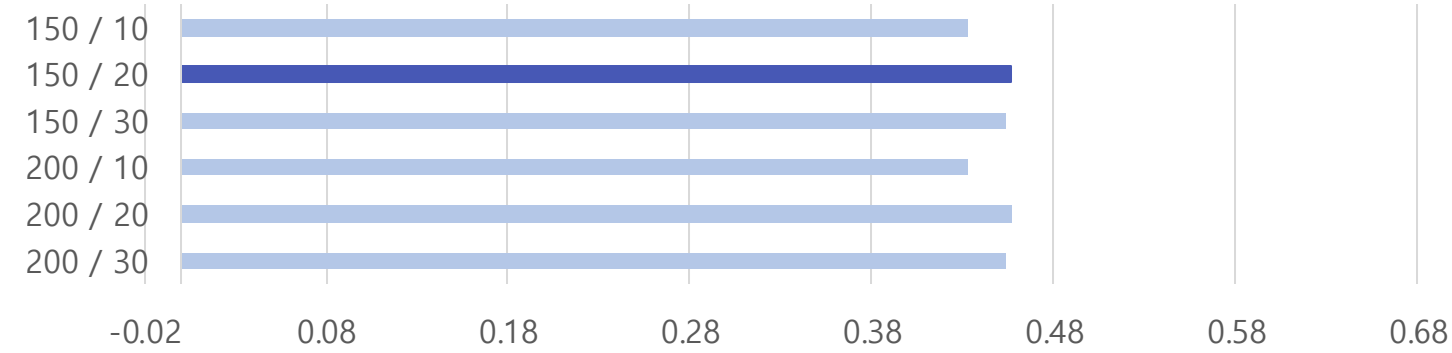
Grid Search

Train f1 - score

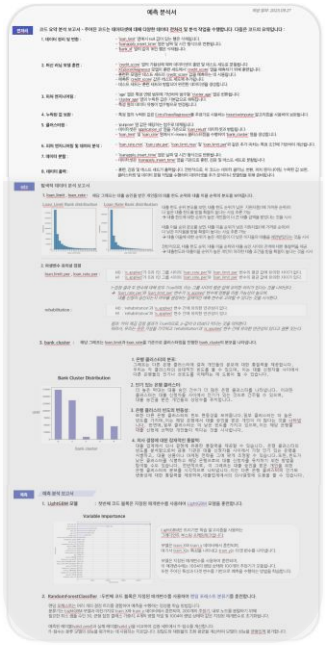
n_estimator/min samples split



Valid f1 - score



자동화 분석서



예측 분석서

작성 일자: 2023.09.27

전처리

전처리 분석 보고서 - 주어진 코드는 데이터셋에 대해 다양한 데이터 전처리 및 분석 작업을 수행합니다. 다음은 코드의 요약입니다 :

1. 데이터 정리 및 변환 :

- 'loan_limit' 열에서 null 값이 있는 행은 삭제됩니다.
- 'loanapply_insert_time' 열은 날짜 및 시간 형식으로 변환됩니다.
- 'bank_id' 열의 값이 16인 행은 삭제됩니다.

⋮

2. 머신 러닝 모델 훈련 :

- 'credit_score' 열의 가용성에 따라 데이터셋이 훈련 및 테스트 세트로 분할됩니다.
- XGBoostRegressor 모델이 훈련 세트에서 'credit_score' 열을 예측하기 위해 훈련됩니다.
- 훈련된 모델은 테스트 세트의 'credit_score' 값을 예측하는 데 사용됩니다.
- 예측된 'credit_score' 값은 테스트 세트에 추가됩니다.
- 테스트 세트는 훈련 세트와 병합되어 완전한 데이터셋을 생성합니다.

3. 피처 엔지니어링 :

- 'age' 열은 특정 연령 범위에 기반하여 범주형 'cluster_age' 열로 변환됩니다.
- 'cluster_age' 열의 누락된 값은 기본값으로 채워집니다.
- 특정 열의 데이터 유형이 범주형으로 변경됩니다.

4. 누락된 값 보완 :

- 특정 열의 누락된 값은 ExtraTreesRegressor를 추정기로 사용하는 IterativeImputer 알고리즘을 사용하여 보완됩니다.

5. 클러스터링 :

- 'purpose' 열 값은 해당하는 범주로 대체됩니다.
- 데이터셋은 'application_id' 열을 기준으로 'loan_result' 데이터셋과 병합됩니다.
- 'loan_limit' 및 'loan_rate' 열에서 K-means 클러스터링을 수행하여 'bank_cluster' 열을 생성합니다.

6. 피처 엔지니어링 및 데이터 분석 :

- 'loan_rate_min', 'loan_rate_per', 'loan_limit_max' 및 'loan_limit_per'와 같은 추가 피처는 특정 조건에 기반하여 계산됩니다.

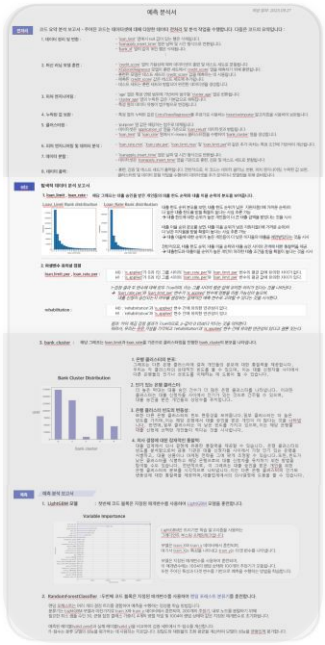
7. 데이터 분할 :

- 'loanapply_insert_time' 열은 날짜 및 시간 형식으로 변환됩니다.
- 데이터셋은 'loanapply_insert_time' 열을 기준으로 훈련, 검증 및 테스트 세트로 분할됩니다.

8. 데이터 출력 :

- 훈련, 검증 및 테스트 세트가 출력됩니다. 전반적으로, 이 코드는 데이터 클리닝, 변환, 피처 엔지니어링, 누락된 값 보완, 클러스터링 및 데이터 분할 작업을 수행하여 데이터셋을 추가 분석이나 모델링을 위해 준비합니다.

자동화 분석서

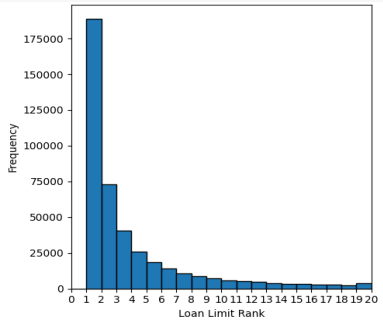


eda

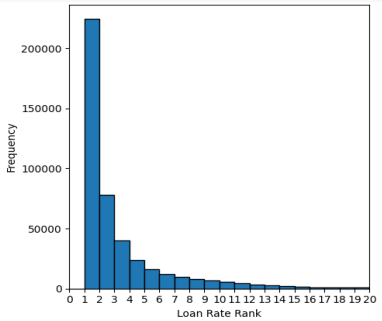
탐색적 데이터 분석 보고서

1. loan_limit , loan_rate : 해당 그래프는 대출 승인을 받은 개인들의 대출 한도 순위와 대출 이율 순위의 분포를 보여줍니다.

Loan_Limit Rank distribution



Loan_Rate Rank distribution



대출 한도 순위 분포를 보면, 대출 한도 순위가 낮은 지원자들(1에 가까운 순위)이 더 높은 대출 한도를 받을 확률이 높다는 사실 추론 가능
→ 대출 한도에 대한 순위가 높은 개인들이 더 큰 대출 금액을 받았다는 것을 시사

대출 이율 순위 분포를 보면, 대출 이율 순위가 낮은 지원자들(1에 가까운 순위)이 더 낮은 이자율을 받을 확률이 높다는 사실 추론 가능
→ 대출 이율에 대한 순위가 높은 개인들이 더 낮은 이자율의 대출을 제안받았다는 것을 시사

전반적으로, 대출 한도 순위, 대출 이율 순위와 대출 승인 사이의 관계에 대한 통찰력을 제공
→ 대출한도와 대출이율 순위가 높은 개인이 유리한 대출 조건을 받을 확률이 높다는 것을 시사

2. 파생변수 유의성 검정

loan_limit_per , loan_rate_per :

H0 : 'is_applied'가 0과 1인 그룹 사이의 'loan_rate_per'와 'loan_limit_per' 변수의 평균 값에 유의한 차이가 없다.
H1 : 'is_applied'가 0과 1인 그룹 사이의 'loan_rate_per'와 'loan_limit_per' 변수의 평균 값에 유의한 차이가 있다.

t-검정 결과 두 변수에 대해 모두 True이며, 이는 그룹 사이의 평균 값에 유의한 차이가 있다는 것을 나타낸다.
→ *'loan_rate_per'와 'loan_limit_per' 변수가 'is_applied' 변수에 영향을 미칠 가능성이 높으며,*
대출 신청이 승인되는지 여부를 결정하는 잠재적인 예측 변수로 고려될 수 있다는 것을 시사한다.

rehabilitation :

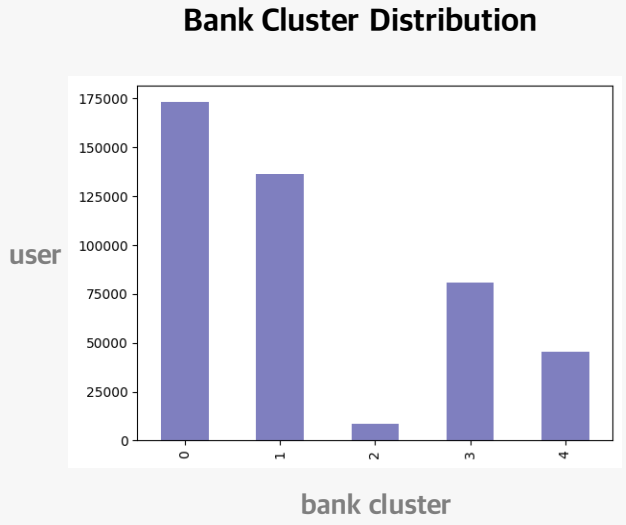
H0 : 'rehabilitation'과 'is_applied' 변수 간에 유의한 연관성이 없다.
H1 : 'rehabilitation'과 'is_applied' 변수 간에 유의한 연관성이 있다.

결과: 카이 제곱 검정 결과가 True이므로, p-값이 0.05보다 작다는 것을 의미한다.
따라서, 우리는 귀무 가설을 기각하고 'rehabilitation'과 'is_applied' 변수 간에 유의한 연관성이 있다고 결론 짓는다.

자동화 분석서

⋮

3. bank_cluster : 해당 그래프는 loan_limit과 loan_rate를 기준으로 클러스터링을 진행한 bank_cluster의 분포를 나타냅니다.



1. 은행 클러스터의 분포:
그래프는 다른 은행 클러스터에 걸쳐 개인들의 분포에 대한 통찰력을 제공합니다. 우리는 각 클러스터의 상대적인 빈도를 볼 수 있으며, 이는 대출 신청자들 사이에서 다른 은행들의 인기나 선호도를 이해하는 데 도움이 될 수 있습니다.

2. 인기 있는 은행 클러스터:
더 높은 막대는 대출 승인 건수가 더 많은 은행 클러스터를 나타냅니다. 이러한 클러스터는 대출 신청자들 사이에서 인기가 있는 것으로 간주될 수 있으며, 대출 승인을 받은 개인들의 상당수를 유치합니다.

3. 은행 클러스터 빈도의 변동성:
또한 다른 은행 클러스터의 빈도 변동성을 보여줍니다.일부 클러스터는 더 높은 빈도를 가지며,이는 해당 은행에서 대출 승인을 받은 개인이 더 많다는 것을 나타냅니다. 반면에,일부 클러스터는 더 낮은 빈도를 가지고 있으며,이는 해당 은행을 대출 신청에 선택한 개인들이 적다는 것을 시사합니다.

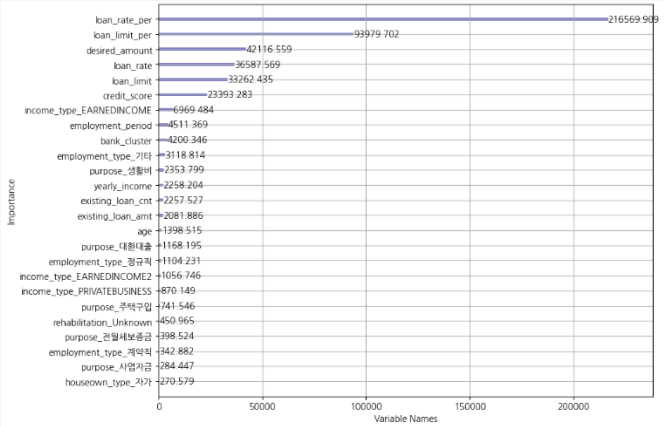
4. 의사 결정에 대한 잠재적인 통찰력:
대출 업계에서 의사 결정에 유용한 통찰력을 제공할 수 있습니다. 은행 클러스터의 빈도를 분석함으로써 금융 기관은 대출 신청자들 사이에서 가장 인기 있는 은행을 식별하고, 대출 상품이나 마케팅 전략을 그에 맞게 조정할 수 있습니다.또한,빈도가 낮은 클러스터를 식별하고 해당 은행으로의 대출 신청자를 유치하기 위한 방법을 탐색할 수도 있습니다. 전반적으로, 이 그래프는 대출 승인을 받은 개인을 위한 은행 클러스터의 분포를 시각적으로 나타냅니다.이는 다른 은행 클러스터의 인기와 변동성에 대한 통찰력을 제공하며,대출업계에서의 의사결정에 도움을 줄 수 있습니다.

자동화 분석서

예측 예측 분석 보고서

1. LightGBM 모델 : 첫번째 코드 블록은 지정된 매개변수를 사용하여 LightGBM 모델을 훈련합니다.

Variable Importance



LightGBM은 트리기반 학습 알고리즘을 사용하는
그래디언트 부스팅 프레임워크입니다.

모델은 train_X와 train_y 데이터에서 훈련되며,
여기서 train_X는 특성을 나타내고 train_y는 타겟 변수를 나타냅니다.

모델은 지정된 매개변수를 사용하여 훈련되며,
이 매개변수에는 1004의 랜덤 상태와 100개의 추정기가 포함됩니다.
또한 주어진 특성과 타겟 변수를 기반으로 예측을 수행하는 방법을 학습합니다.

2. RandomForestClassifier :두번째 코드 블록은 지정된 매개변수를 사용하여 랜덤 포레스트 분류기를 훈련합니다.

랜덤 포레스트는 여러 개의 결정 트리를 결합하여 예측을 수행하는 앙상블 학습 방법입니다.
분류기는 LightGBM 모델과 마찬가지로 train_X와 train_y 데이터에서 훈련되며, 200개의 추정기, 내부 노드를 분할하기 위해
필요한 최소 샘플 수인 30, 균형 잡힌 클래스 가중치, 4개의 병렬 작업 및 1004의 랜덤 상태와 같은 지정된 매개변수로 초기화됩니다.

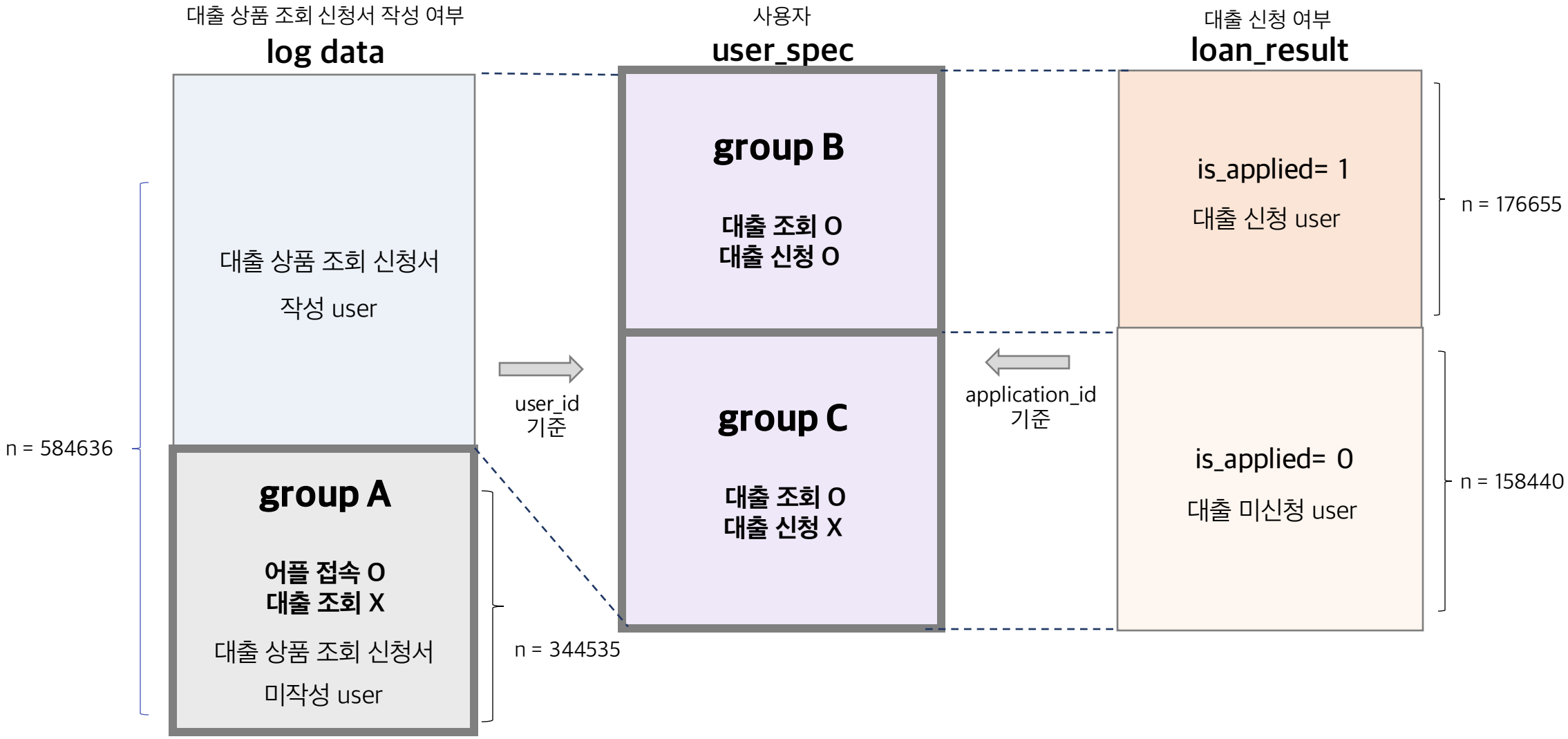
예측된 레이블(valid_pred)과 실제 레이블(valid_y)을 비교하여 검증 세트에서 f1-점수를 계산합니다.
f1-점수는 분류 모델의 성능을 평가하는 데 사용되는 지표입니다. 정밀도와 재현율의 조화 평균을 계산하여 모델의 성능을 균형있게 평가합니다.

군집 문제

Data Split

(group B, C - 추가 군집 분석 진행)

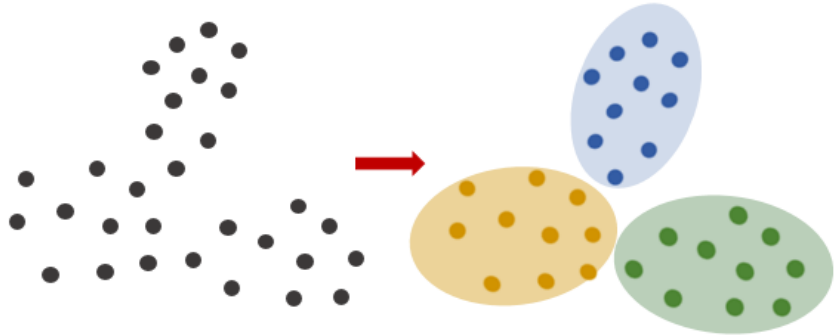
- 대출 상품 조회 신청서를 작성하지 않은 user / 대출신청서 작성 후 대출 신청을 한 user / 대출신청서 작성 후 대출 신청을 하지 않은 user group으로 분리



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)

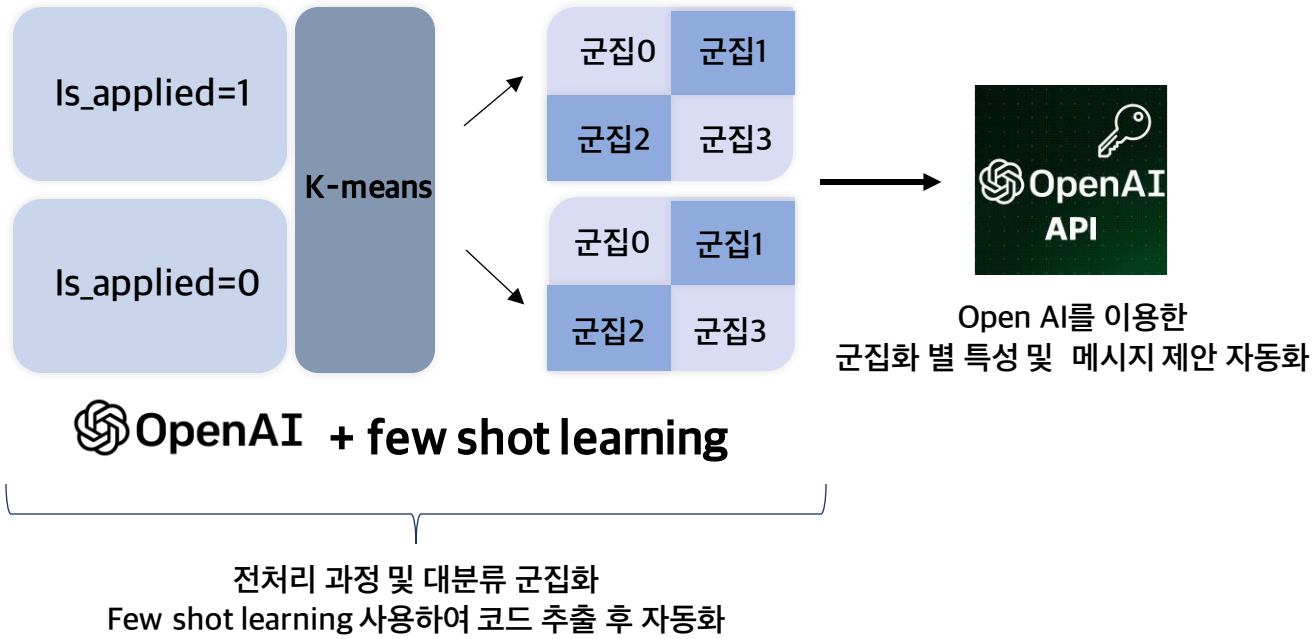


- 사용 데이터

is_applied=1인 데이터	is_applied=0인 데이터
yearly_income, desired_amount, credit_score, existing_loan_amt, age 연수입, 희망 대출금액, 신용점수, 기대출 금액, 연령	
Event_LoanManage 대출관리 횟수	Other count service 대출관리 이외의 서비스 사용 횟수 합

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

- 군집화 흐름



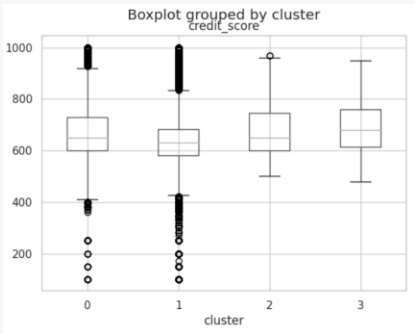
군집 분석서

군집 | 군집 분석 보고서

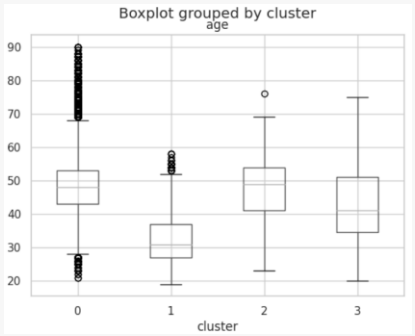
대출을 신청한 그룹 군집화 결과 (group B)

cluster	credit_score	yearly_income	desired_amount	existing_loan_amt	age	event_UseLoanManage
0	680.048814	5.212805e+07	2.834628e+07	9.589864e+07	48.543835	9.206077
1	640.701994	3.422755e+07	1.800349e+07	3.091602e+07	32.038554	7.785778
2	675.870178	5.306651e+09	7.835137e+08	5.592233e+07	47.630137	8.650685
3	688.541349	2.306918e+08	6.831799e+09	7.080818e+07	42.553459	7.566038

▲ 군집별 변수 평균



▲ 군집별 신용점수 상자그림



▲ 군집별 연령 상자그림

군집 분석서

작성 일자: 2023.09.27

군집1 신용 저조 군집

- 신용점수가 낮고 연소득이 상대적으로 낮음
- 대출 희망 금액 상대적으로 낮음, 기존 대출 금액량 적음
- 연령은 30대 초반으로 비교적 젊은 사람들
- 대출 관리 서비스에 더 많이 의존할 수 있으며, 대출 어플에서 저렴한 대출 상품을 제공하는 것이 유용

"당신의 대출 상환을 효율적으로 관리해드립니다"
"대출 이자 계산기를 사용하여 최적의 이자율을 확인하세요"

군집2 고소득 중년 군집

- 신용점수가 상대적으로 높고 연소득도 높음
- 대출 희망 금액과 기존 대출 금액량도 상당히 높음
- 연령은 40대 후반에서 50대 초반
- 대출 관리 서비스보다는 다양한 대출 상품을 제공

"당신의 신용을 개선하는 방법을 알려드립니다"
"저희 대출 상환 계획 관리 서비스로 빠르게 대출을 상환하세요"

군집3 영앤리치

- 신용점수가 매우 높고 연소득도 매우 높음
- 대출 희망 금액은 상대적으로 낮지만, 기존 대출 금액량은 상당히 높음
- 연령은 40대 중반
- 고금리 대출 상환을 최소화할 수 있는 상품을 제공.

"당신의 대출 한도를 상향 조정해드립니다"
"VIP 대출 혜택을 누려보세요"

군집4 대출 상당 존재 중년 군집

- 신용점수가 높고 연소득도 매우 높음
- 대출 희망 금액은 비교적 높음
- 기존 대출 금액량은 상대적으로 낮음.
- 연령은 40대 후반에서 50대 초반
- 신속하고 간편한 대출 신청 프로세스 제공 필요

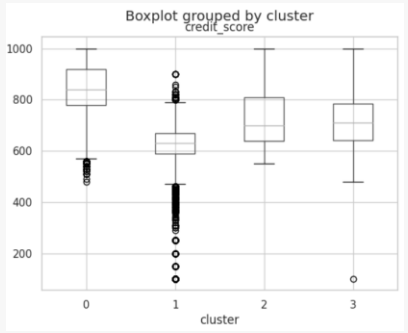
"당신에게 최적화된 대출 이자 혜택을 제공합니다"
"저희 대출 상환 계획 관리 서비스로 효율적으로 대출을 상환하세요"

군집 분석서

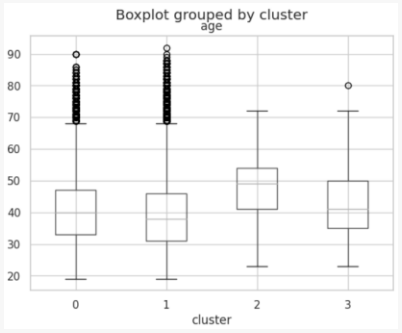
대출을 신청하지 않은 그룹 군집화 결과 (group C)

cluster	credit_score	yearly_income	desired_amount	existing_loan_amt	age	other_cnt_service
0	843.349598	5.658085e+07	7.226409e+07	1.117661e+08	40.501666	15.171248
1	630.822171	3.876261e+07	2.784620e+07	4.850588e+07	39.073688	32.453959
2	732.669301	6.407792e+09	7.853854e+08	3.579385e+07	47.010417	19.000000
3	726.558349	2.714513e+08	7.037925e+09	8.174159e+07	42.185841	22.292035

▲ 군집별 변수 평균



▲ 군집별 신용점수 상자그림



▲ 군집별 연령 상자그림

군집1 신용 높은 소득층 군집

- 신용점수 평균적으로 중간
- 연간 소득은 높은 편이며, 대출 희망 금액도 상당히 높음
- 기존 대출 금액량도 크고, 연령은 중년층에 해당
- 필요 서비스:대출 상환 계획 관리와 대출 이자

"MZ들을 위한 저렴한 대출 상품,지금 신청하세요!"

군집2 젊은 소득층 군집

- 신용점수가 낮은 편, 연간 소득도 중간 수준
- 대출 희망 금액과 기존 대출 금액량 모두 적은 편
- 연령은 젊은 층에 해당
- 필요 서비스: 신용 개선 방법 안내, 대출 상환계획 관리

"고수익층을 위한 다양한 대출 상품,
원하는 금액을 신속하게 대출받으세요!"

군집3 고소득층 군집

- 신용점수가 매우 높고, 연간 소득도 높음
- 대출 희망 금액은 매우 크나
  기존 대출 금액량은 상대적으로 적음
- 연령은 어린 층에 해당.
- 대출한도 상향 조정 안내와 VIP 대출혜택 안내 필요 예상

"고신용 고소득자를 위한 저금리 대출 상품,
이자 부담을 줄이세요!"

군집4 대출 금액 높은 소득층 군집

- 신용점수가 평균적으로 높고 연간 소득도 매우 높음
- 대출 희망 금액 상대적으로 낮고,기존 대출 금액량도 적음
- 연령은 중년층에 해당
- 필요 서비스: 대출 이자 혜택 안내와 대출 상환 계획 관리

"고신용 고소득자를 위한 간편 대출 신청,
원하는 금액을 빠르게 대출받으세요!"

서비스 제안

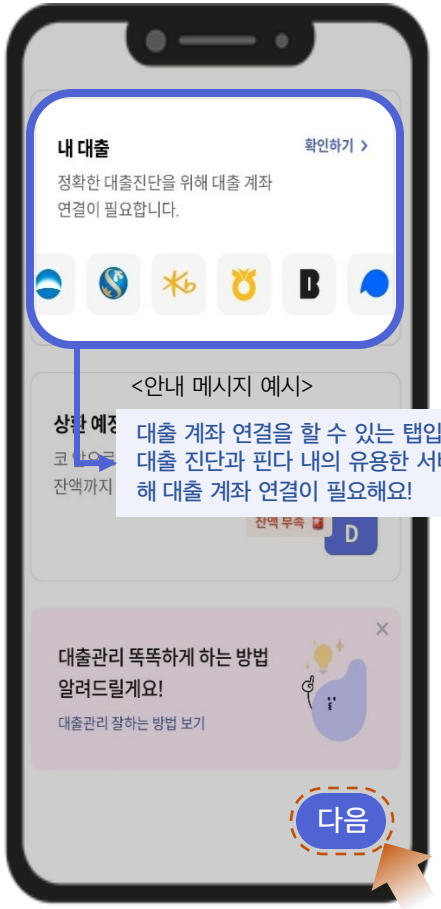
고객 군집별 서비스 제안

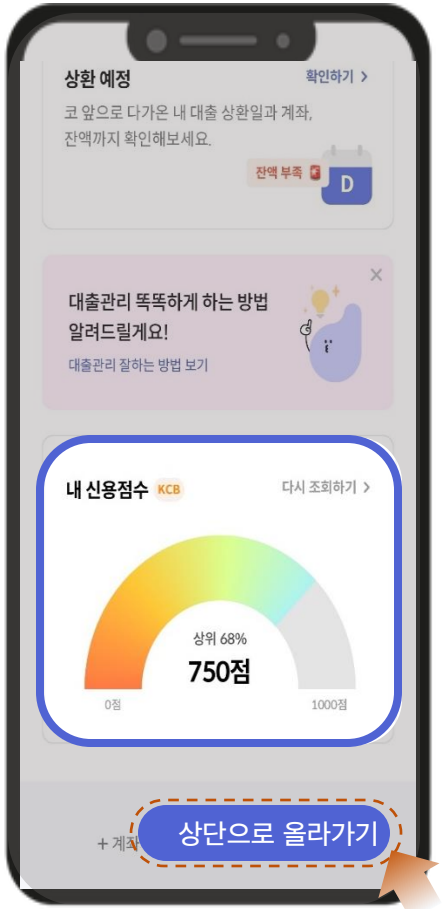
- 핀다의 홈 화면의 기능 설명 및 가이드 매뉴얼 서비스 제공

금융 입문자

- 저신용, 저소득
- 적은 희망금액,적은 기존대출
- 상대적으로 연령층 젊음
- 경험 부족으로 판단
- 가이드 매뉴얼 제공 및 재방문 유도

1. 상단에 '홈 화면 빠르게 살펴보기' 버튼을 배치한다.





다음 스크롤로 넘어간다

가장 상단으로 올라간다

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

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

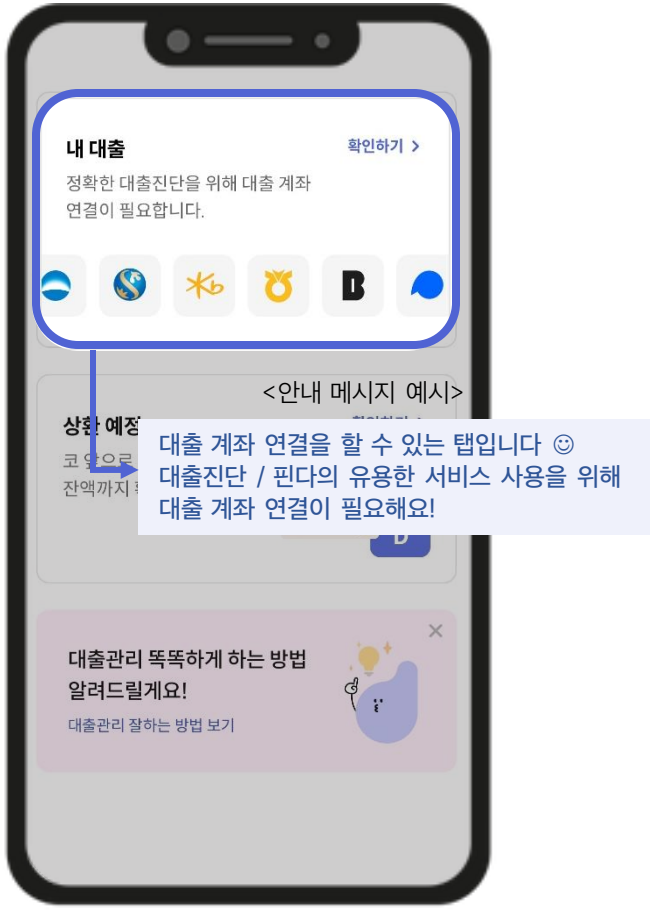
고객 군집별 서비스 제안

- 기존 대출 관리 기능의 편리한 활용을 주요 서비스로 제공

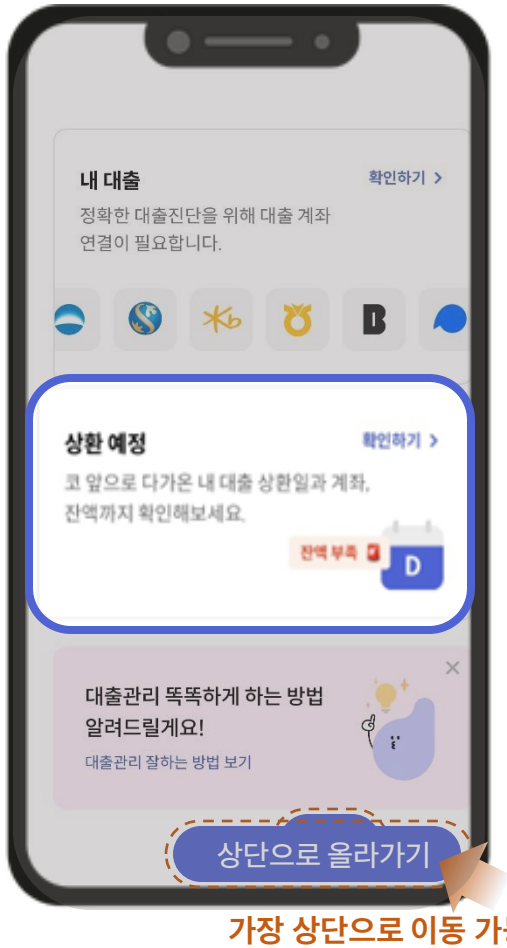
금융 전문가

- 고신용, 고소득
- 높은 희망금액, 높은 기존대출
- 중년 user
- 한도가 높은 상품 희망 예상

1. 현재 자신의 한도를 편리하게 조회할 수 있도록 홈화면에 배너를 배치



2. 아래로 스크롤 시 하단에 기존 대출 관리 서비스를 통해 편의성 제공



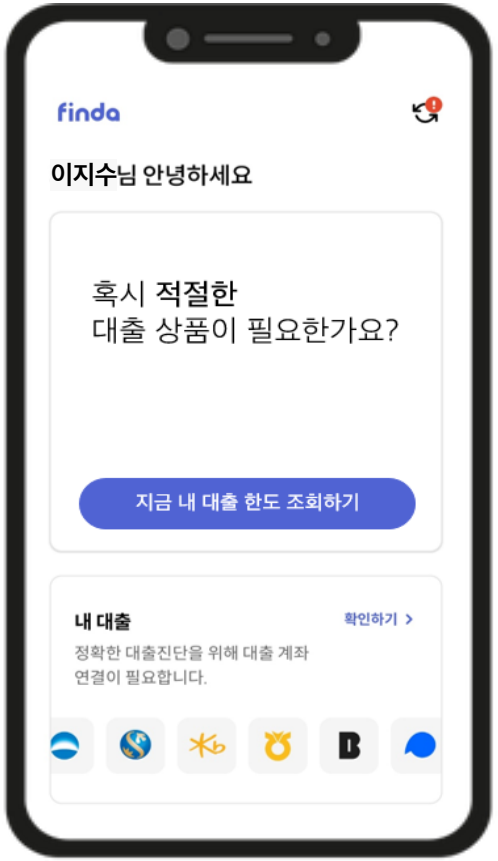
3. 대출 상환일, 잔액 등 기존 대출 관리에 대한 편의성 제공

고객 군집별 서비스 제안

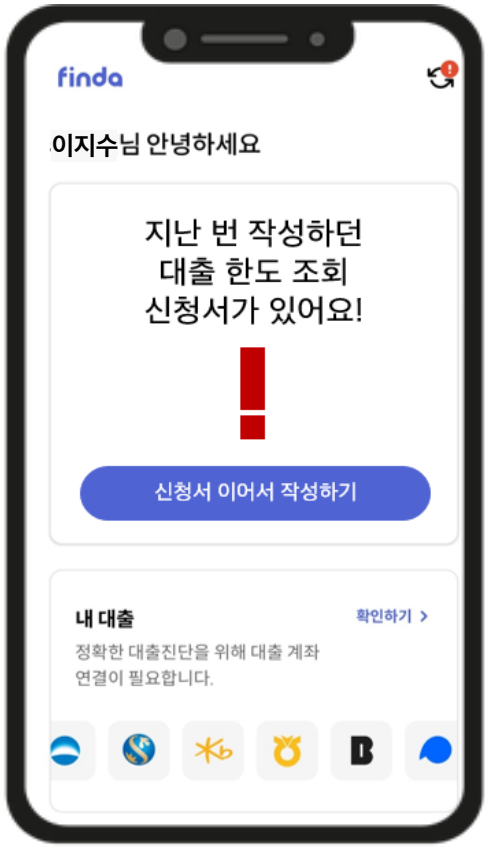
- 맞춤형 대출 상품 관련 정보를 담은 배너를 홈 화면 최상단에 배치 , 유도

재정 안전층 청년

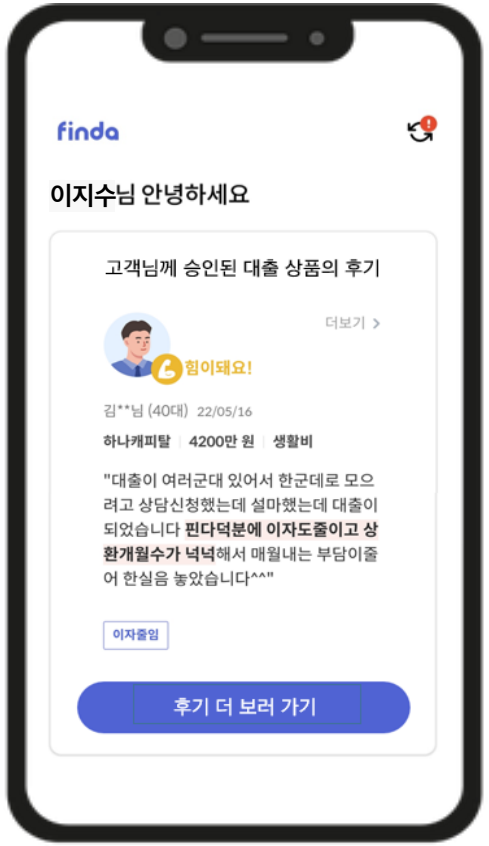
- 고신용, 고소득
- 높은 희망금액, 적은 기존대출
- 상대적으로 연령층이 낮음
- 경험은 없으나 가능성이 높으며 잠재 우량 고객이라고 판단 가능
- 한도나 금리 측면에서 유리한 대출 상품 및 서비스 제안 필요



맞춤형 상품을 메인 화면에 제시하며 적절한 대출 상품 제안



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



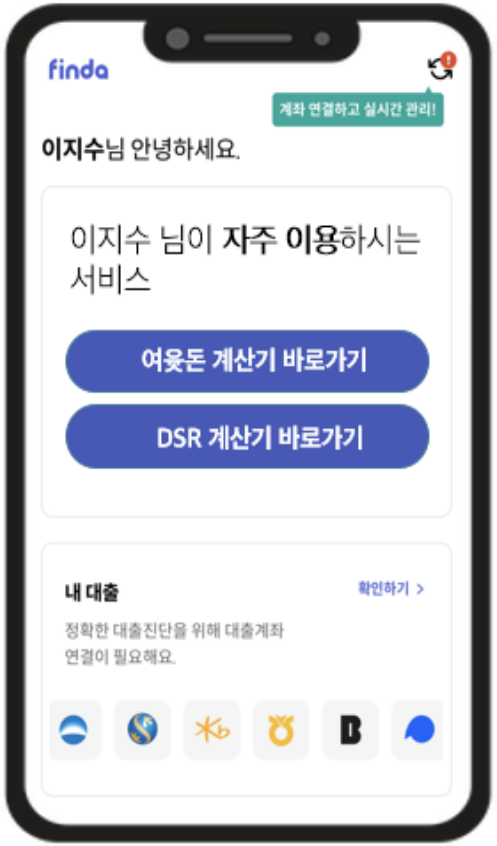
승인된 대출 목록 중 대출 승인 예측 모델을 활용하여 대출 상품을 추천하고 관련 후기를 노출시키는 메시지 배치

고객 군집별 서비스 제안

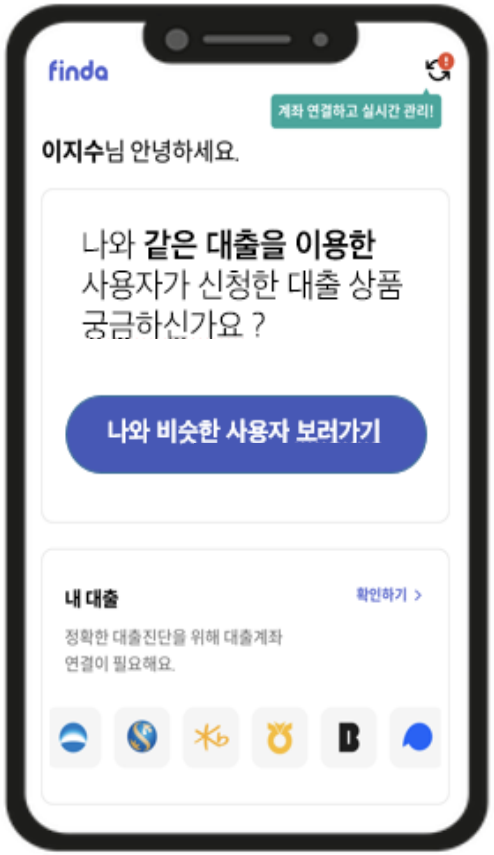
- 재방문율을 높일 수 있는 서비스 바로가기 및 유사 사용자 대출 상품 추천

재정 안정층 중년층

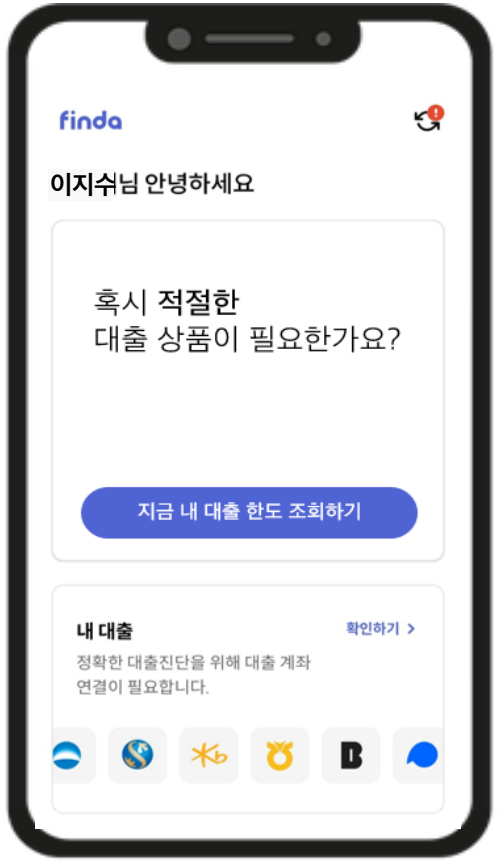
- 고신용, 고소득
- 낮은 희망금액, 적은 기존대출
- 중년 user
- 희망금액이 낮은 상품 선호
- 희망금액대의 상품 추천과 재방문 유도를 위한 계산기 서비스 등 자주 찾는 서비스 바로가기 제안



대출관련 계산기를 이용하는 고객에게 메인화면에 자주 이용하는 서비스 바로가기 배치



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



맞춤형 상품을 메인 화면에 제시하며 적절한 대출 상품 제안