
데이터분석분야_퓨처스부문

빅콘테스트 결과보고서

팀명 시크릿곤두
팀장 오태연 oty0906@naver.com
팀원 권가영 kky3653@naver.com
팀원 박정민 park000103@gmail.com
팀원 전수연 junyeon8570@naver.com

순서

01

서론

데이터 소개 및 분석목표

1. 데이터 소개
 - 유저스펙 테이블
 - 대출 상품 결과 테이블
 - 유저 로그 데이터
2. 분석목표
 - 고객의 대출 상품 신청 여부 예측
 - 대출 신청, 미신청 고객을 분류하여 고객 특성분석

02

본론

분석 과정 및 결과

1. 예측
 - 데이터를 기반으로 고객의 대출상품 신청여부 예측 (2022년 3~5월 데이터를 기반으로 2022년 6월 예측)
2. 군집분석
 - 잠재적 채무 불이행자 식별
 - 대출 신청자 중 각 군집별 특성 분석

03

결론

결론 도출

1. 예측
 - 대출 신청 미신청자들 예측하여 예측정확도 향상
 2. 군집분석
 - 군집에 맞는 핀다 앱 서비스 제안
-

목차

01

데이터를 기반으로 고객의 대출상품 신청 여부 예측

02

군집분석으로 잠재적 채무 불이행자 식별

03

대출 조회자중 각 군집별 특성 분석

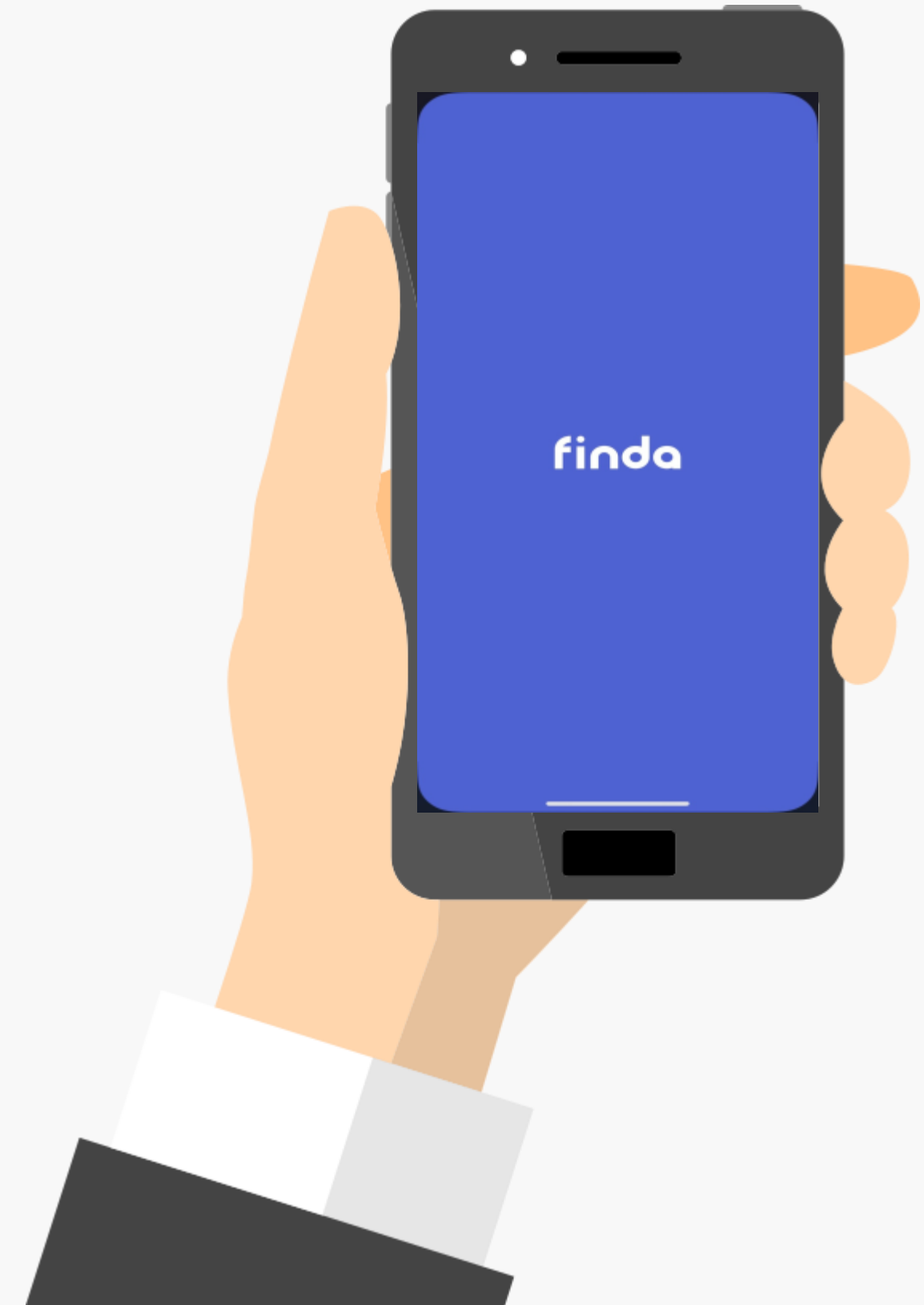
01

고객의 대출상품 신청 여부 예측

서론

1. 분석목표

대출 비교 중개 플랫폼인
핀다(finda)는 고객의 데이터를 수집하고 분석하여 고객에게 더 나은 서비스를 제공할 수 있도록 기반을 다지고 있다. 이 핀다앱 사용자 중 대출신청여부를 예측하여 핀다앱이 대출에 기여하는 실질적인 영향력을 예측하고자 한다.



서론

2. 데이터 소개 _ drop

'user_spec'에 있는데, 'loan_result' 에는 없는 application_id

→ 모두 거절되거나 중간에 이탈한 경우

'loan_result'에는 있는데, 'user_spec'에는 없는 application_id

→ 기간 차이로 인해 누락된 데이터

'loan_result'에 'loan_rate', 'loan_limit' 칼럼에 결측치(nan)

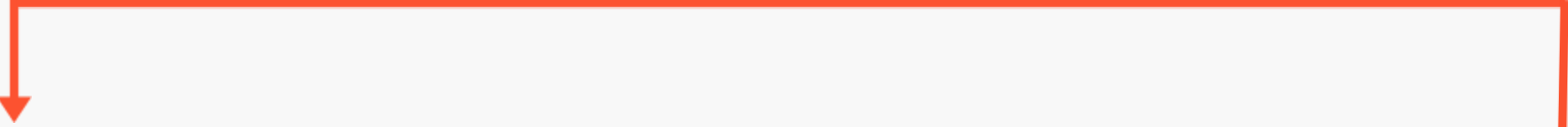
→ 금융사에서 값을 보내주지 않은 경우

01 고객의 대출상품 신청 여부 예측

서론

2. 데이터 소개 _ merge

'loan_result'에 'user_spec'을 합침



The diagram shows a red arrow pointing from the text 'user_spec' to the 'application_id' column of the 'loan_result' table, indicating the merge operation.

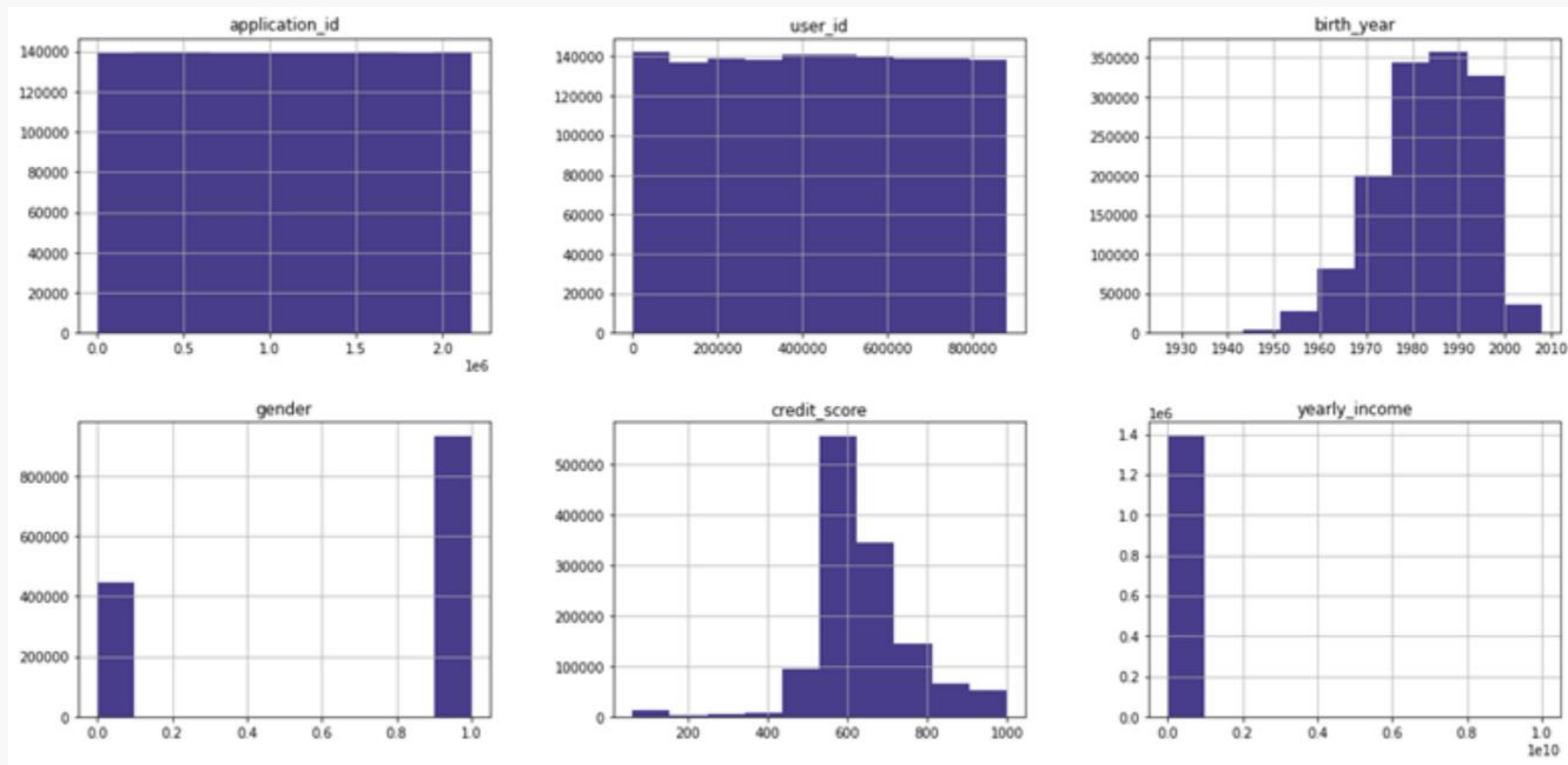
index	application_id	loanapply_insert_time	bank_id	product_id
0	1	2022-05-06 11:05:57	62	200
1	1	2022-05-06 11:05:42	59	150
2	1	2022-05-06 11:05:43	50	142
3	1	2022-05-06 11:05:56	38	16
4	1	2022-05-06 11:05:44	37	206
5	1	2022-05-06 11:05:43	38	223
6	2	2022-04-13 10:58:41	62	200
7	4	2022-06-20 10:24:38	7	191
8	4	2022-06-20 10:24:38	7	220
9	6	2022-04-26 02:02:44	59	150
10	6	2022-04-26 02:02:46	63	226
11	6	2022-04-26 02:02:45	35	29
12	6	2022-04-26 02:02:45	54	259

index	application_id	loanapply_insert_time	bank_id	product_id
0	1	2022-05-06 11:05:57	62	200
1	1	2022-05-06 11:05:42	59	150
2	1	2022-05-06 11:05:43	50	142
3	1	2022-05-06 11:05:56	38	16
4	1	2022-05-06 11:05:44	37	206
5	1	2022-05-06 11:05:43	38	223
6	2	2022-04-13 10:58:41	62	200
7	4	2022-06-20 10:24:38	7	191
8	4	2022-06-20 10:24:38	7	220
9	6	2022-04-26 02:02:44	59	150
10	6	2022-04-26 02:02:46	63	226
11	6	2022-04-26 02:02:45	35	29
12	6	2022-04-26 02:02:45	54	259
13	6	2022-04-26 02:02:45	41	222
14	6	2022-04-26 02:02:45	1	61
15	6	2022-04-26 02:02:45	10	149

01 고객의 대출상품 신청 여부 예측

본론

1. 데이터 시각화



본론

1. 데이터 시각화

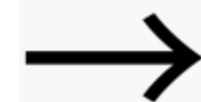
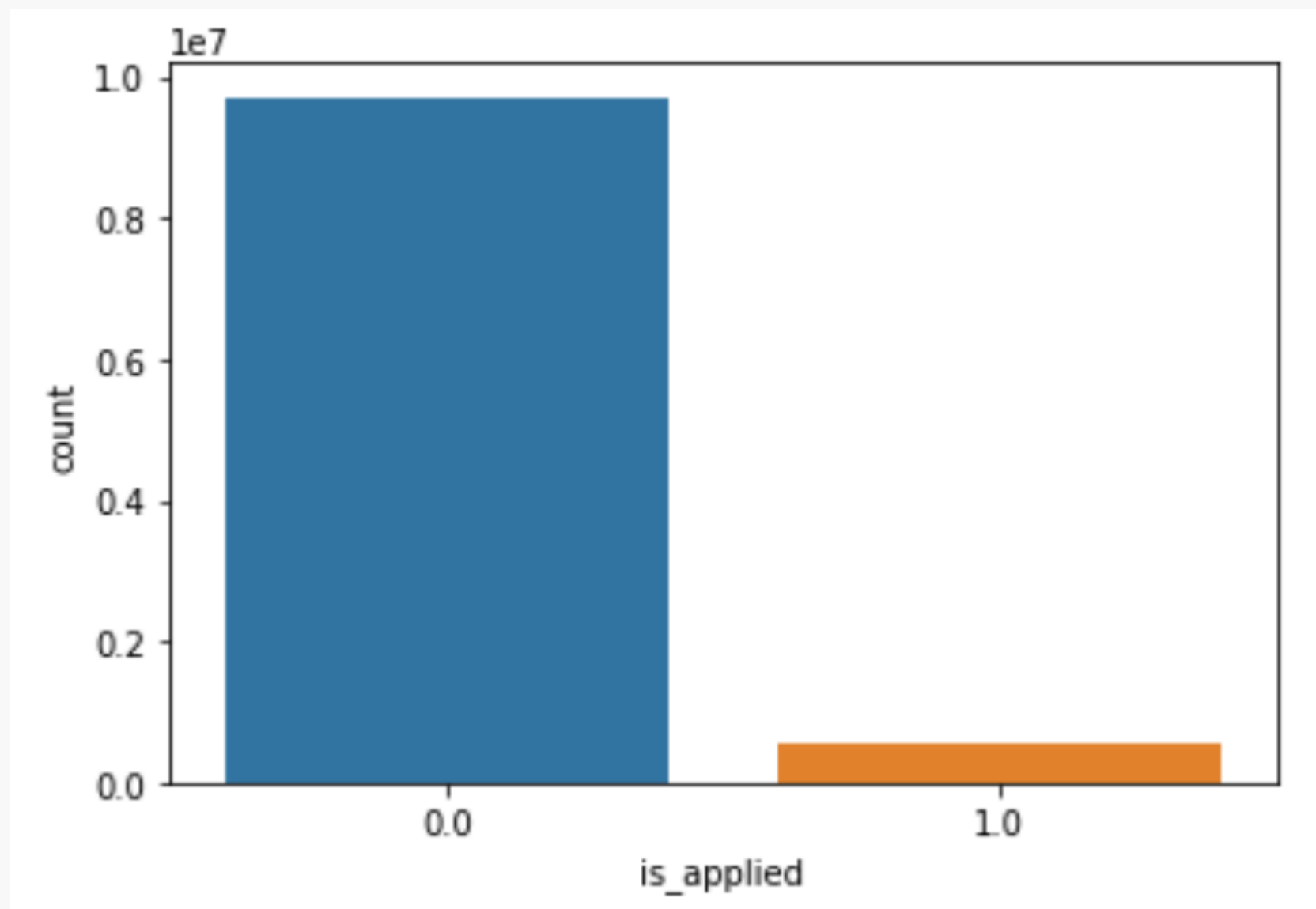
[시각화를 통해 알 수 있는 결과] → 가공필요



1. birth_year의 1990년생이 가장 많다.
2. 남자가 여자보다 대출조회하는 사람이 더 많다.
3. 대출조회하는 사람들의 신용점수는 600점이 가장 많다.
4. yearly_income (연소득) 이상치에 의해 그래프가 이상하게 나옴

본론

1. 데이터 시각화

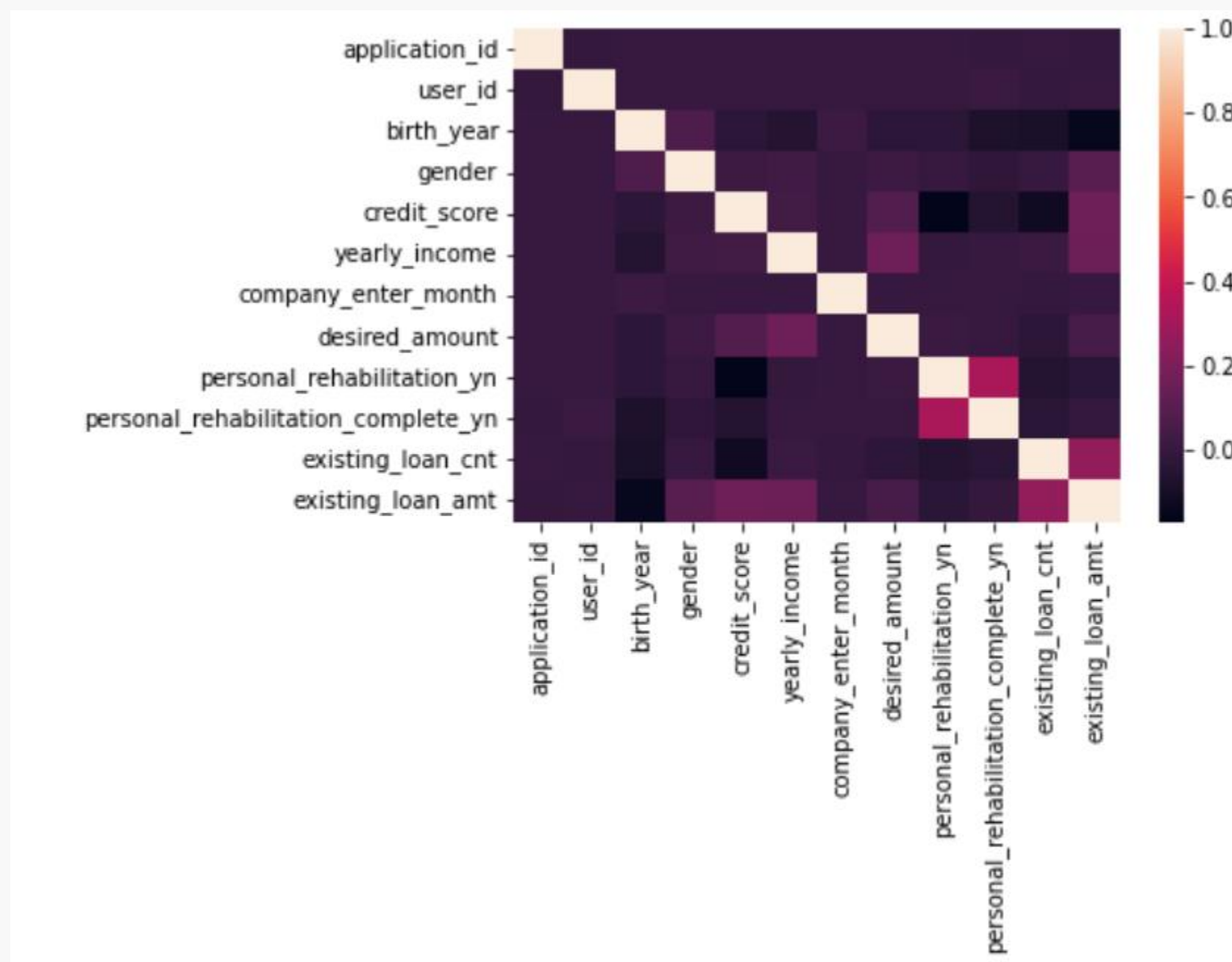


목표변수의 불균형에 의해 데이터 전처리 후 계층적k겹 교차검증을 위해 불균형 해소

특별히 상관관계 높은 변수
들 찾아볼 수 없음



변수 가공 후 다중공산성에 의해 예측도가 떨어지는 것을 방지하기 위해 상관관계가 높은 변수들은 제거

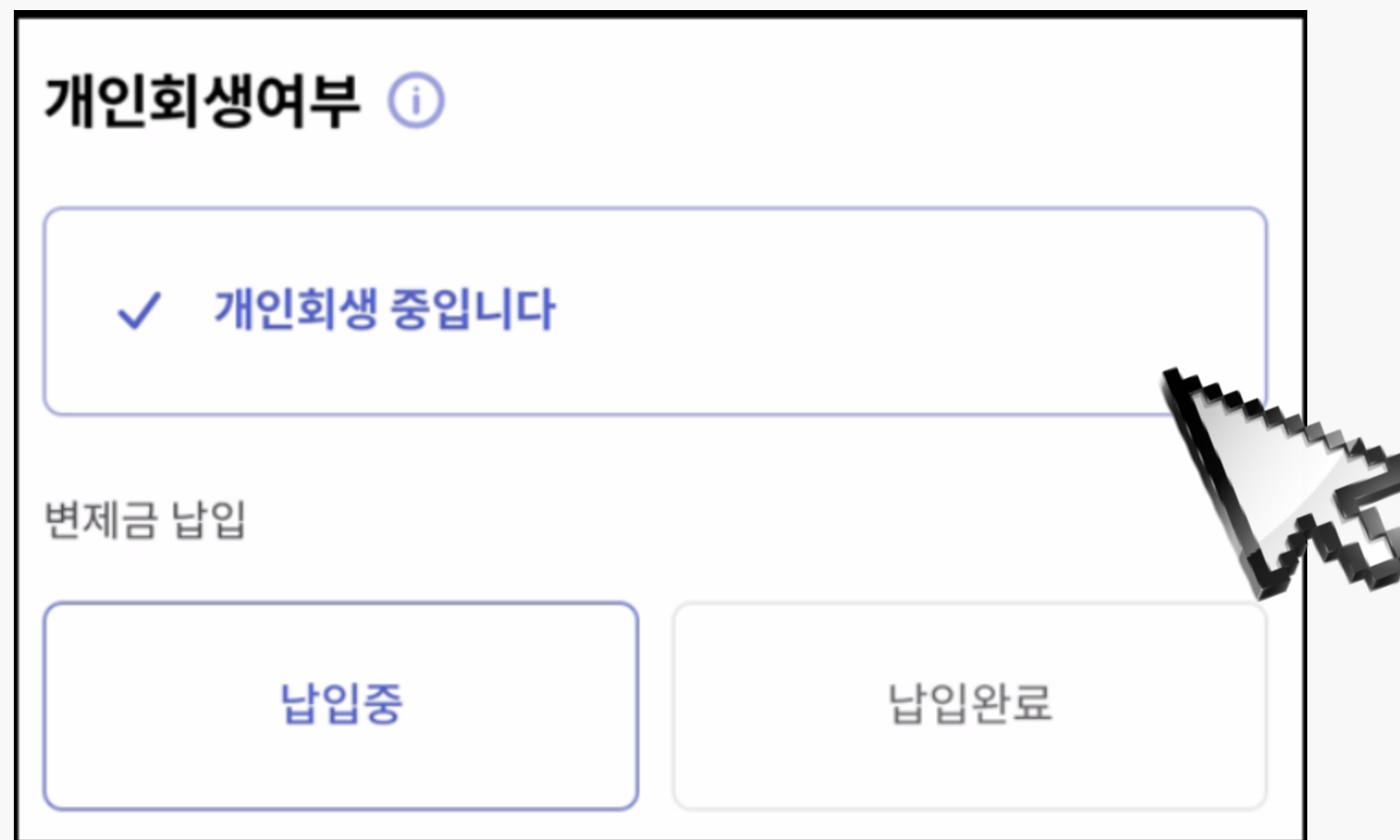


본론

2. 데이터 전처리

personal_rehabilitation_yn

personal_rehabilitation_complete_yn



The screenshot shows a mobile app interface for managing personal rehabilitation status. At the top, there is a title '개인회생여부' (Personal Rehabilitation Status) with an information icon. Below the title is a large button with a blue checkmark and the text '개인회생 중입니다' (I am currently in personal rehabilitation). Below this button, the text '변제금 납입' (Payment of installment) is displayed. At the bottom, there are two buttons: '납입중' (In payment) and '납입완료' (Payment complete). A mouse cursor is pointing at the '납입중' button.

[핀다앱 개인회생여부 페이지]

'개인회생중입니다' 버튼을 눌러야
'납입중' 또는 '납입완료' 누를 수 있음

01 고객의 대출상품 신청 여부 예측

유형] 결측치 처리 및 파생변수 생성(개인회생자 관련 변수 2개 1개로 합침)

방법]

개인회생자	개인회생자 납입완료	범주형 변수
X	X	0
X	0	1
0	X	2
0	0	3

새로운 개인회생자
범주형 변수 생성

본론

birth_year
gender



유형] 결측치 처리 및 파생변수 생성

- 방법]
- ① user_spec에 있는 칼럼 birth_year과 gender의 na값이 있는 행은 같음.
 - ② 같은 user_id 별 생년월일 및 성별은 같으므로 생년월일 및 성별이 있는 user는 같은 user_id에 있는 na값을 채워줌
 - ③ 생년월일(birth_year) 변수→age(나이)변수로 파생변수 생성 후 결측치 처리

01 고객의 대출상품 신청 여부 예측

본론

age

유형] 결측치

방법] train data에 age변수의 평균값을
구한 후 이 값으로 na 대체

existing_loan_cnt

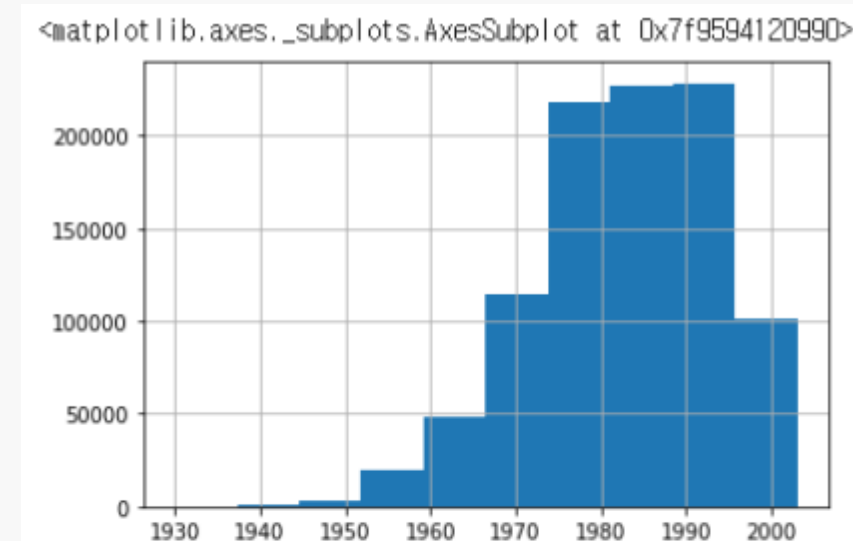
유형] 결측치

방법] user_spec 데이터에 기대출수가 0인 데이터가 없기 때문에 na값이 0임을 알 수 있음

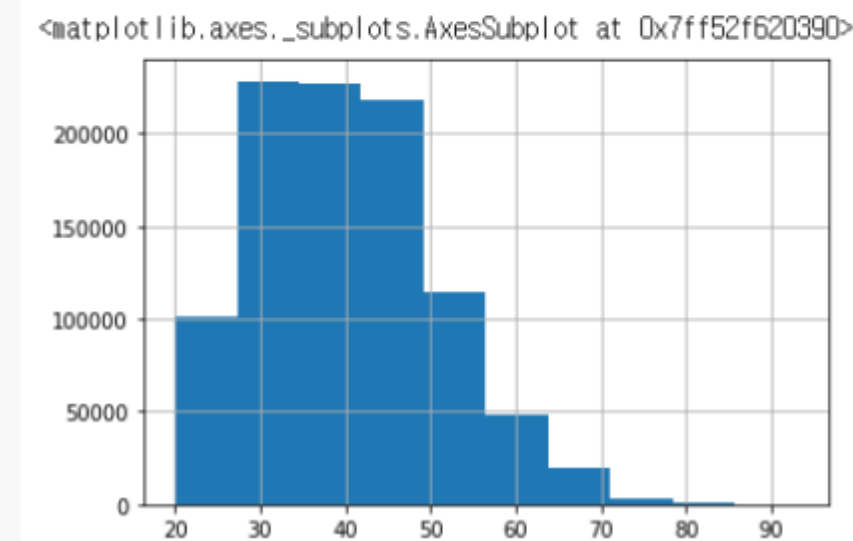
birth_year



age



df_user['age'].hist()



본론

existing_loan_amt

유형] 결측치

방법] 기대출수 0인 경우

기대출금액도 0

기대출수가 1이상,기대출금액이 na인 경우

마이너스통장을 가진 사람들이므로 기대출금액 na값을 0으로 대체

yearly_income

유형] 결측치

방법] train data에 1행으로 정규직이므로 이상치로 간주해 삭제

본론

company_enter_month

유형] 결측치, 파생변수 생성

방법] birth_year가 company_enter_month보다 최근인 사람 이상치로 간주하고 삭제
근무기간(단위:개월)로 파생변수 생성 → diff_month

diff_month

유형] 결측치

방법] income_type으로 개수 확인했을 때 대부분 OTHERINCOME에 속해있어 무직, 주부, 학생으로 간주해 0으로 대체

나머지 na값 train data의 diff_month의 평균값으로 대체

본론

기대출과다자 (lotloan)

유형] 파생변수

방법] 기대출금액이 연소득의 1.5배보다 크면 기대과다출자로 간주

application_counts

유형] user_id당 신청서 개수 (파생변수)

방법] 'loan_result' table에 'user_spec'의 application_id를 합친 테이블을 만든 후 user_id당 application_id 개수를 측정하여 지정함

본론

product_rate

유형] 한 user당 상품 비율 (파생변수)

방법] 한 user_id 당 상품(product)의 비율 구함

bank_rate

유형] 파생변수, 결측치

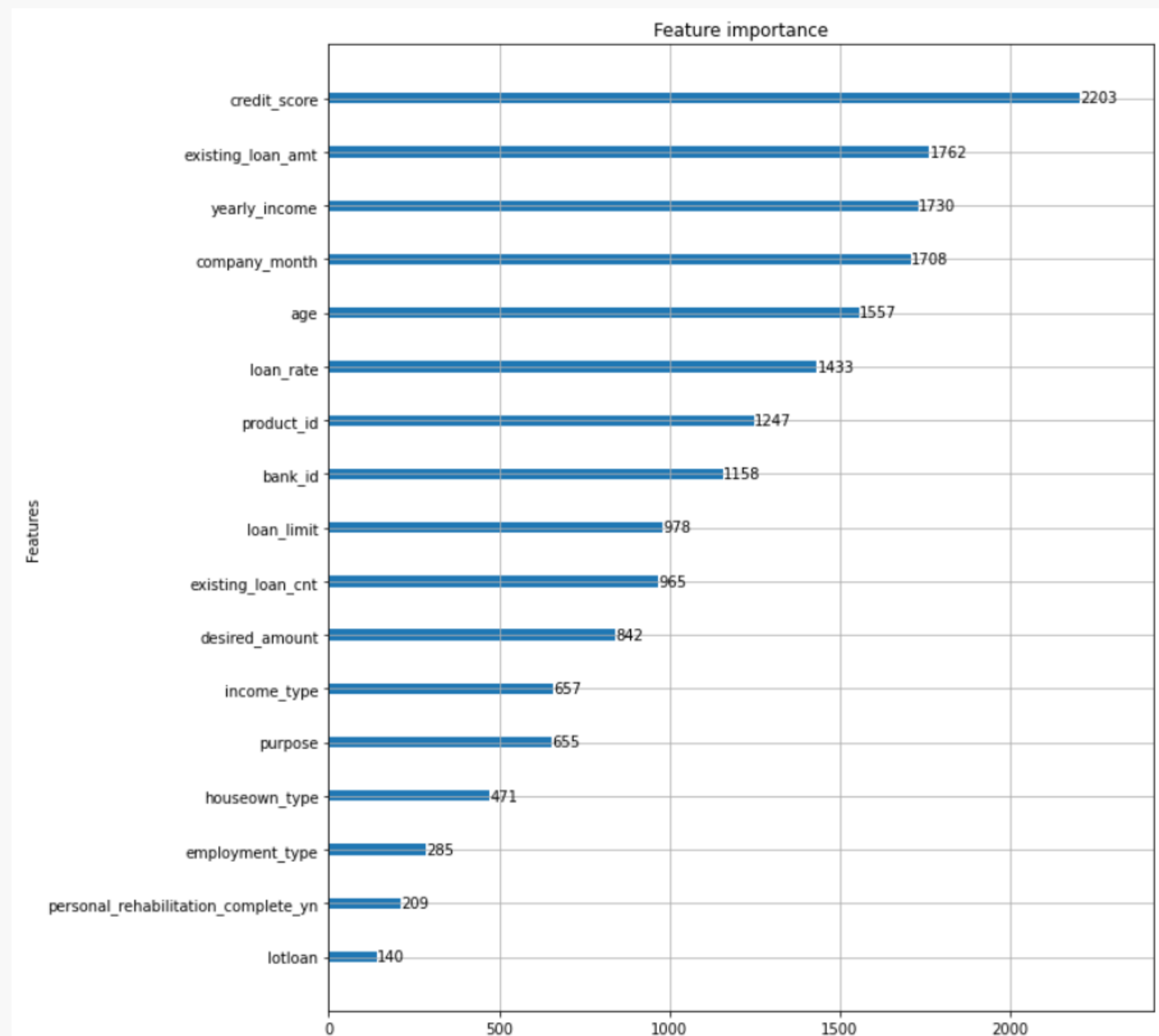
방법] 한 user_id당 bank_id 개수를 구해 비율을 구함.
결측치는 lightgbm 모델링을 통해 유의미한 변수들을 통해 예측함

01 고객의 대출상품 신청 여부 예측

본론

[bank_rate 변수와의 중요도]

변수 중요도를 통해 credit score와 연관성이 높다는 것을 알 수 있음



본론

credit_score

유형] 한 user당 상품 비율 (파생변수)

방법] bank_rate 변수와 신용점수와 연관성이 높은 변수들을 추출하여 Xgboost 모델링을 통해 결측치를 예측함.

desired_rate

유형] 대출희망금액당 승인한도 비율 (파생변수)

방법] loan_limit 변수에 desired_amount 변수를 나누어 두개의 단위를 맞춰주어 (백만 단위) 대출희망금액당 승인한도 변수를 만듦.

본론

3. 모델링 Lightgbm

Stratified Kfold (계층적 k교차검증)

↓
컷오프 조절

↓
parameter 조절

↓
lightgbm 모델링

```
[36] # StratifiedKFold 교차검증하기
      from sklearn.model_selection import StratifiedKFold

      skf = StratifiedKFold(n_splits=5)

      for train_idx, test_idx in skf.split(x, y):
          X_train = x.iloc[train_idx]
          X_test = x.iloc[test_idx]
          y_train, y_test = y[train_idx], y[test_idx]

          model = LGBMClassifier(n_estimators=300)
          lgb_eval = [(X_test, y_test)]
          model.fit(X_train, y_train,
                    eval_set=lgb_eval,
                    eval_metric='logloss',
                    verbose=True,
                    early_stopping_rounds=100)

          preds = model.predict(X_test)
          print(f1_score(y_test, preds))
```

03

대출 조회자중 각 군집별 특징 분석

01 서론

- 데이터 전처리

데이터 가공

- 대출조회자 군집분석을 위해 로그데이터와 user_spec 데이터 결합
- 로그데이터 : user_id 별 각 event 횟수 가 지도록 데이터 가공
- user_spec : 로그 데이터와 공통된 user_id 중 가장 최신인 행만 남기도록 가공

결측치 및 이상치 처리

- 데이터 특성을 고려해 결측치 처리
- MICE 를 이용해 결측치 처리
- purpose 범주 영어, 한글 통합

범주형 변수 가변수화

- MICE를 이용해 결측치 처리하는 과정에서 범주형 가변수화
- k-means 수행을 위해서도 필수적

Min-Max Scaling

- 최소값, 최대값을 이용해 데이터의 크기를 0과 1사이로 정규화 시켜줌
- 수치형 변수

K-means 수행

서론

1. 데이터 소개

log_data 가공

- ❶ user_spec에는 대출을 한번이라도 조회해본 사람의 정보만 있음.
- ❷ log_data에서 대출조회자의 정보만을 가져오기 위해 log_data와 user_spec에서 공통된 user_id를 가지는 행만을 추출
- ❸ 한 user_id 당 각 event 이용횟수를 구하기 위해 event 가변수화함.
- ❹ merge된 데이터프레임의 각 event별 개수를 구하기 위해 행을 합함.

03 대출 조회자중 각 군집별 특징 분석

서론

1. 데이터 소개

user_spec의 user_id(대출조회자)를 log_data에서 추출 후 log_data에 user_spec merge

user_spec의 user_id가 중복된 데이터가 있다면 가장 최근일시를 merge

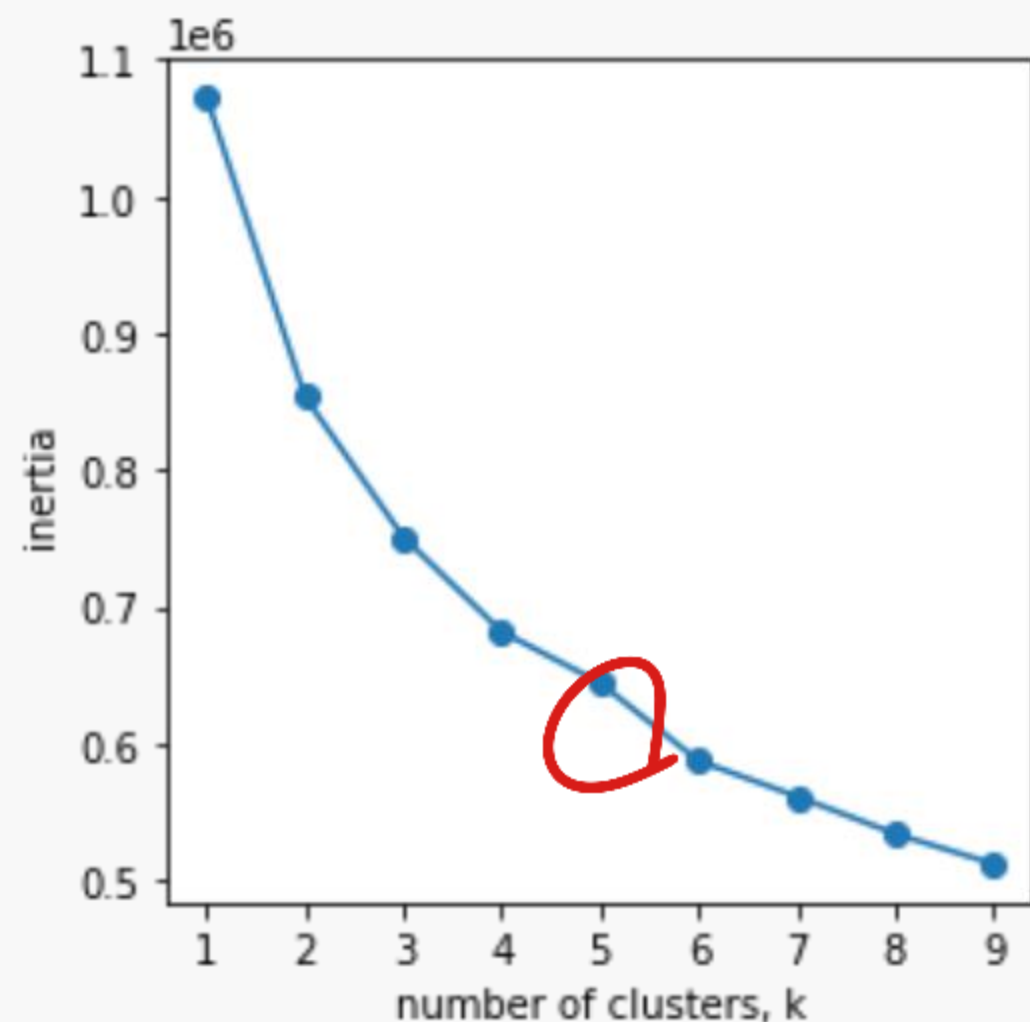
index	user_id	event	timestamp
0	1	GetCreditInfo	2022-06-16 23:58:42
1	1	Login	2022-06-16 23:58:41
2	1	UseLoanManage	2022-06-16 23:58:41
3	1	GetCreditInfo	2022-05-03 14:52:35
4	1	GetCreditInfo	2022-05-03 14:52:28
5	7	GetCreditInfo	2022-05-22 16:39:49
6	9	GetCreditInfo	2022-05-21 23:43:52
7	9	GetCreditInfo	2022-05-21 23:37:58
8	9	GetCreditInfo	2022-05-21 23:43:33
9	11	StartLoanApply	2022-06-13 21:43:11
10	11	ViewLoanApplyIntro	2022-06-13 21:43:11
11	11	CompleteIDCertification	2022-06-13 21:43:29
12	11	EndLoanApply	2022-06-13 21:45:22
13	11	UseLoanManage	2022-03-24 11:05:31

index	application_id	user_id	birth_year	gender	insert_time
0	1767402	1	2000.0	1.0	2022-05-03 14:53:33
1	1031304	1	2000.0	1.0	2022-05-03 14:51:00
2	1096882	7	1985.0	1.0	2022-05-22 16:42:47
3	248576	9	1996.0	1.0	2022-05-21 23:41:09
4	1521731	11	1997.0	1.0	2022-06-09 11:25:25
5	1894024	11	1997.0	1.0	2022-06-13 21:44:30
6	1180616	11	1997.0	1.0	2022-03-24 10:58:22
7	1751500	11	1997.0	1.0	2022-04-20

서론

2. 군집 분석 수행 : K- means 알고리즘 이용

(1) 최적의 k값 찾기 - elbow method 이용



- cluster 의 개수 k를 직접 지정해야 함
- 특정 k 이후 cost가 거의 변하지 않는 elbow point 선택
- k = 5로 선택해, 다섯개 군집으로 분류하는 5-means 군집분석 수행

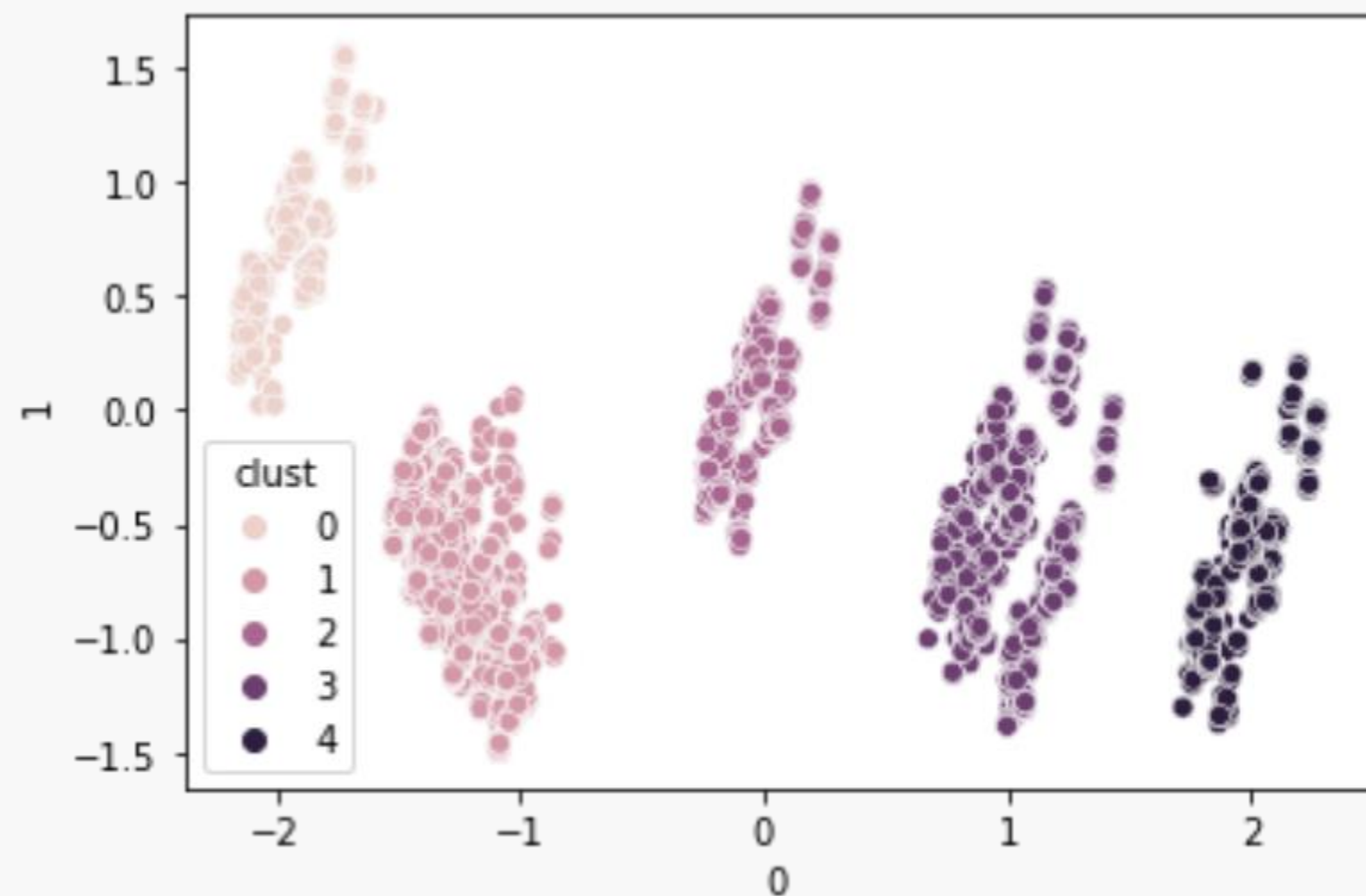
`clust_df.clust.value_counts()`

```
1    110015
2     85948
3     74660
4     56804
0     43008
Name: clust, dtype: int64
```

서론

2. 군집 분석 수행 : K- means 알고리즘 이용

(2) PCA를 이용해 5개의 군집을 2차원 시각화



- PCA를 이용해 50개 변수를 2개로 축소
- 2개의 변수를 X,Y축으로 하여 2차원으로 군집을 시각화
- 5개의 군집으로 나누어짐을 확인 가능
- 두번째 군집과 나머지 4개의 군집 간에 명확한 차이가 있음

03 대출 조회자중 각 군집별 특징 분석

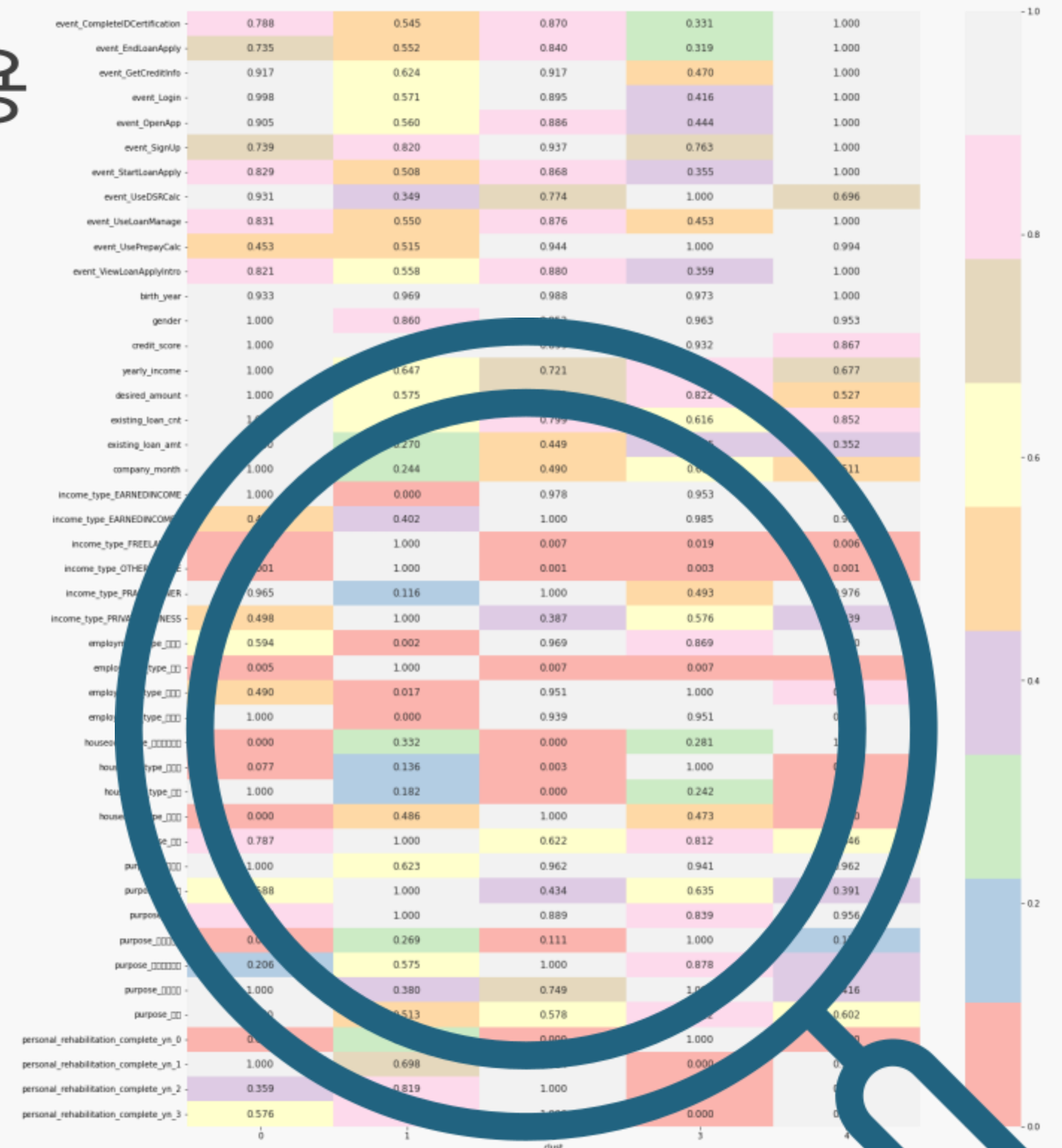
서론

2. 군집 분석 수행 : K- means 알고리즘 이용

(3) 각 군집별 평균값 히트맵을 이용한 군집 해석

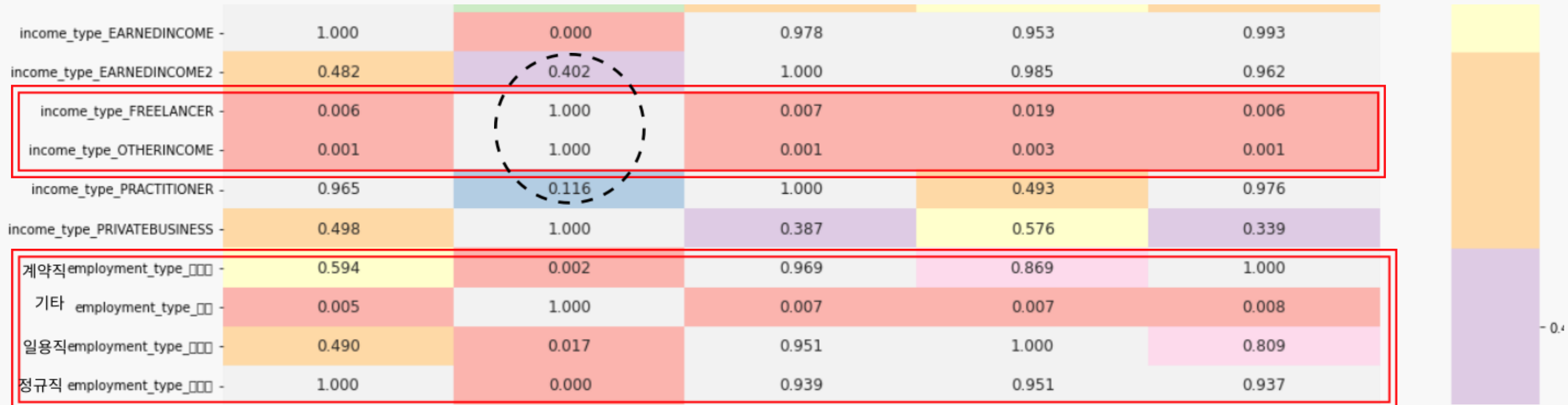
- 군집별 각 변수들의 평균값을 구한 후,
이 값을 각 변수별 최대 평균값으로 나누어
변수별 각 군집이 얼마나 비중을 차지하는지에 대한
히트맵 제작
- 각 변수에 대한 군집별 평균값 테이블과 비교하면서 해석

	event_CompleteIDCertification	event_EndLoanApply	event_GetCreditInfo
clust			
0	0.004562	0.003316	0.005571
1	0.003157	0.002491	0.003793
2	0.005036	0.003792	0.005572
3	0.001919	0.001440	0.002853
4	0.005790	0.004514	0.006075



3. 군집 분석 결과 해석

(1) 두번째 군집



- employment_type을 보면, 두번째 군집은 기타를 제외한 나머지에서 0에 가까운 값을 가지고, employment_type기타에서 두번째 군집을 제외한 나머지 군집은 0에 가까운 값을 가진다. 따라서, 두번째 군집은 employment_type이 기타인 데이터들로 구성된 군집임을 알 수 있다.
- income_type 또한 마찬가지로, FREELANCER와 OTHERINCOME에서 두번째 군집은 1의 값을 가지고, 나머지는 0의 값을 가진다. 따라서, FREELANCER와 OTHERINCOME 데이터들은 대부분 두번째 군집에 속한다는 것을 알 수 있다.
- 두번째 군집은 대부분 기타소득, 프리랜서, 기타 고용인 데이터들로 구성되어 있으며, 나머지 군집은 그 외의 소득, 고용 데이터들로 구성되어 있다는 점에서 차별점이 있다.

3. 군집 분석 결과 해석

(2) 첫번째, 세번째, 다섯번째 군집

기타가족 배우자 자가 전월세	employment_type_000	0.490	0.017	0.951	1.000	0.809
	employment_type_000	1.000	0.000	0.939	0.951	0.937
	houseown_type_000000	0.000	0.332	0.000	0.281	1.000
	houseown_type_000	0.077	0.136	0.003	1.000	0.049
	houseown_type_00	1.000	0.182	0.000	0.242	0.000
	houseown_type_000	0.000	0.486	1.000	0.473	0.000
	purpose_00	0.787	1.000	0.622	0.812	0.646

- 첫번째 군집의 houseown_type을 보면, 자가에서 1의 값을 가지고, 자가를 제외한 나머지 부분에서는 0에 가까운 값을 가짐을 알 수 있다. 따라서, 첫번째 군집은 자가 주택을 가지는 데이터들로 구성된 것을 알 수 있다.
- 세번째 군집의 houseown_type을 보면, 전월세에서 1의 값을 가지고, 전월세를 제외한 나머지 부분에서는 0에 가까운 값을 가짐을 알 수 있다. 따라서, 세번째 군집은 주택 전월세를 하는 데이터들로 구성된 것을 알 수 있다.
- 다섯번째 군집의 houseown_type을 보면, 기타가족에서 1의 값을 가지고, 기타가족을 제외한 나머지 부분에서는 0에 가까운 값을 가짐을 알 수 있다. 따라서, 다섯번째 군집은 기타가족의 주택에서 거주하는 데이터들로 구성된 것을 알 수 있다.

3. 군집 분석 결과 해석

(3) 네번째 군집

	0	1	2	3	4
personal_rehabilitation_complete_yn_0	0.000	0.288	0.000	1.000	0.000
personal_rehabilitation_complete_yn_1	1.000	0.698	0.984	0.000	0.987
personal_rehabilitation_complete_yn_2	0.359	0.819	1.000	0.001	0.904
personal_rehabilitation_complete_yn_3	0.576	0.869	1.000	0.000	0.890
	0	1	2	3	4
	dust				

- 네번째 군집의 personal_rehabilitation_complete_yn을 보면, 0일때 1의 값을 가지고, 나머지 값에서는 0에 가까운 값을 가짐을 볼 수 있다. 또한, 첫번째, 세번째, 다섯번째 군집들과 비교해보았을때, personal_rehabilitation_complete_yn_0은 네번째 군집에서만 나타남을 알 수 있다. 따라서, 개인회생 여부 기입을 패스한 데이터들이 네번째 군집에 몰려있음을 알 수 있다.

3. 군집 분석 결과 해석

(4) 결과 요약



03 대출 조회자중 각 군집별 특징 분석

4. 각 군집별 고객 맞춤 서비스 추천

(1) 첫번째 군집 -> Not 프리랜서, 기타소득 // 자가 주택 소유

- 기본적으로 앱 사용량이 많으며, 연소득, 기대출수 및 기대출 금액, 근무년수가 많음
- => but, 다른 군집들에 비해 event_UsePrepayCalc 사용량이 적음
- 아직 잘 알지 못 하고 있는 것으로 추정

첫번째 고객에게 추천하고자 하는 서비스!



여윳돈 계산기

- 여윳돈이 생겼을 때 어떤 대출부터 갚는 게 좋을지 알려주어 대출상환에 도움을 주는 기능
- 기대출수 및 금액이 많은 첫번째 군집에게 적합!

event_CompleteIDCertification	0.788
event_EndLoanApply	0.735
event_GetCreditInfo	0.917
event_Login	0.998
event_OpenApp	0.905
event_SignUp	0.739
event_StartLoanApply	0.829
event_UseDSRCalc	0.931
event_UseLoanManage	0.831
event_UsePrepayCalc	0.453
event_ViewLoanApplyIntro	0.821
birth_year	0.933
gender	1.000
credit_score	1.000
yearly_income	1.000
desired_amount	1.000
existing_loan_cnt	1.000
existing_loan_amt	1.000
company_month	1.000

03 대출 조회자중 각 군집별 특징 분석

4. 각 군집별 고객 맞춤 서비스 추천

(2) 두번째 군집 -> 프리랜서, 기타소득(주부,무직,학생)

- event_Signup 사용량이 특히 많은 편, event_USED SRCalc 사용 적음
- > 연소득이 타 군집에 비해 낮은 편으로, DSR 계산기는 연소득 대비 부채 상환액 비율을 계산해 대출 가능성을 알아보는 서비스이기때문에 소득이 적은 DSR 점수가 낮게 나올 것으로 예상되어 cluster2 군집은 이 서비스를 덜 사용한 것으로 보임.

두번째 고객에게 추천하고자 하는 서비스!



여윳돈 계산기 or 대출관리 서비스

- DSR 점수를 높이기 위해서는 연소득도 중요하지만 부채 상환 비율이 중요하므로, 대출 관리 서비스나 여윳돈 계산기를 추천해 이용자의 부채 상환 비율을 높이도록 유도!

event_CompleteIDCertification	0.545
event_EndLoanApply	0.552
event_GetCreditInfo	0.624
event_Login	0.571
event_OpenApp	0.560
event_SignUp	0.820
event_StartLoanApply	0.508
event_UseDSRCalc	0.349
event_UseLoanManage	0.550
event_UsePrepayCalc	0.515
event_ViewLoanApplyIntro	0.558

03 대출 조회자중 각 군집별 특징 분석

4. 각 군집별 고객 맞춤 서비스 추천

(3) 세번째 군집 -> Not 프리랜서, 기타소득 // 전월세 주택 거주

- 타군집보다 대출목적이 전월세보증금에 많다는 것을 알 수 있음.
- 타 event 사용량 대비, DSR 계산기의 사용량이 적은편으로 보인다.
- > DSR 계산기 추천 or 전세 vs 월세 계산기

세번째 고객에게 추천하고자 하는 서비스!

계산 결과

월세가 더 유리합니다.

매월 26,667원씩, 총 640,000원을 절약하실 수 있습니다.

전월세 비교

월세	매월 월세	400,000원
	매월 이자	106,667원
	총 비용 (2년)	12,160,000원
전세	매월 이자	533,333원
	총 비용 (2년)	12,800,000원

DSR계산기 or 전세 vs 월세 계산기

- DSR 계산기는 연소득 대비 부채 상환액 비율을 계산해 대출 가능성을 알아보는 서비스
- 전세와 월세 중 뭐가 유리한지 알아보는 서비스

event_CompleteIDCertification -	0.870
event_EndLoanApply -	0.840
event_GetCreditInfo -	0.917
event_Login -	0.895
event_OpenApp -	0.886
event_SignUp -	0.937
event_StartLoanApply -	0.868
event_UseDSRCalc -	0.774
event_UseLoanManage -	0.876
event_UsePrepayCalc -	0.944
event_ViewLoanApplyIntro -	0.880

03 대출 조회자중 각 군집별 특징 분석

4. 각 군집별 고객 맞춤 서비스 추천

(4) 4번째 군집 -> Not 프리랜서, 기타소득 //

개인회생에 응답을 하지 않고 대출조회를 한 사람들

- 대출한도 조회 관련 서비스는 이용이 적으나, DSR계산기와 여윳돈 계산기를 많이 사용한 것으로 보임.

4번째 고객에게 추천하고자 하는 서비스!



대출한도조회 서비스

- 대출희망금액은 높고, 대출목적도 다양한 것으로 보아 대출한도조회가 도움이 될 것으로 보임.

- 대출한도조회 서비스에 보다 관심을 가질 수 있도록 홈화면에 많이 노출하도록 함

event_CompleteIDCertification	-	0.331
event_EndLoanApply	-	0.319
event_GetCreditInfo	-	0.470
event_Login	-	0.416
event_OpenApp	-	0.444
event_SignUp	-	0.763
event_StartLoanApply	-	0.355
event_UseDSRCalc	-	1.000
event_UseLoanManage	-	0.453
event_UsePrepayCalc	-	1.000
event_ViewLoanApplyIntro	-	0.359

03 대출 조회자중 각 군집별 특징 분석

4. 각 군집별 고객 맞춤 서비스 추천

(5) 5번째 군집 -> Not 프리랜서, 기타소득 // 기타가족 소유 주택 거주

- 전체적인 앱 사용량 많으나, DSR계산기에만 사용량이 적음
- 다른 군집에 비해 대출 희망 금액이 적고, 기대출횟수 대비 기대출금액은 적음.

-> 소액대출을 선호하는 고객군집들로 보임.

5번째 고객에게 추천하고자 하는 서비스!



소액 대출 관련 대출 조회 기능

- 소액대출을 선호하는 고객 군집이므로,
소액 대출에 초점을 맞춘 대출 조회 기능 추천

event_CompleteIDCertification	1.000
event_EndLoanApply	1.000
event_GetCreditInfo	1.000
event_Login	1.000
event_OpenApp	1.000
event_SignUp	1.000
event_StartLoanApply	1.000
event_UseDSRCalc	0.696
event_UseLoanManage	1.000
event_UsePrepayCalc	0.994
event_ViewLoanApplyIntro	1.000