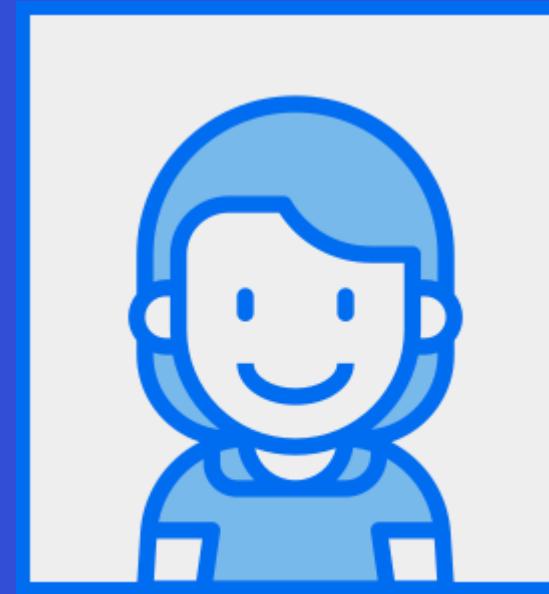
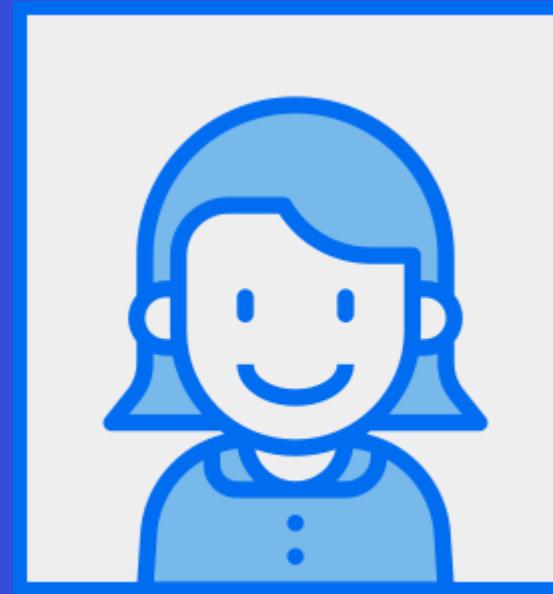


2 0 2 2 빅 콘 테 스 트 퓨 처 스 부 문

# 앱 사용성 데이터를 통한 대출신청 예측분석



TEAM Speaking Potato |

김수정 kstal38@snu.ac.kr  
박세현 ps\_hyen@snu.ac.kr  
조항범 poozx9@snu.ac.kr  
진수정 sjjin210@snu.ac.kr

# I N D E X

---

01

분석 배경

02

데이터 탐색

03

대출신청 예측 모델링

04

고객 군집 분석

05

의의 및 한계

06

부록

# 01 분석 배경

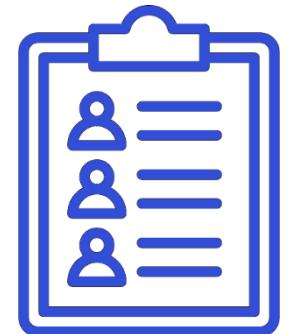
# 01 분석 배경



## 데이터 설명



대출 상품결과



사용자 스펙



사용자 로그

- 사용자가 신청한 대출별 금융사별 승인결과
- 승인한도와 승인금리 등이 있음
- 예측에 쓰일 신청 여부(타겟) 데이터가 있음

- 사용자에 대한 다양한 신용정보 데이터
- 사용자가 직접 입력 한 대출에 필요한 여러 정보 데이터
- 사용자가 입력한 데이터이므로 결측값이나 이상값 존재

- finda App 로그 정보 데이터
- 각 사용자별 finda App을 통하여 한 행동 데이터
- 행동 기록을 기준으로 향후 군집분석 시행

# 01 분석 배경

## 분석 배경



### ◦ 사용자 데이터를 이용한 효율적이고 정확한 추천

단순히 금리가 낮고 한도가 높은 대출 상품을 나열하여 고르게 하는 것은 쉽지만 불필요한 정보들까지 고객에게 전달한다는 단점이 있다. 이러한 단점을 해결하기 위하여 고객의 정보에 맞게 대출 신청을 할 것 같은 상품을 예측하는 모델을 만들어 추천하는 방식을 제안한다.

### ◦ 앱 사용성 데이터를 고객의 특성 파악 및 맞춤 메시지 추천

핀다 앱을 이용하는 사용자들은 각자가 목적이 다를 수 있다. 대출을 희망하는 고객부터 단순히 자신의 신용 정보를 확인하려는 고객까지. 때문에 군집분석을 통해 고객들 군집마다의 특성을 파악하여 맞춤 메시지를 추천한다.

## 분석 문제

### 1 대출 상품 신청 여부 예측

고객의 정보와 대출 상품의 정보를 기반으로 여러 머신러닝 알고리즘을 양상하여 대출 상품 신청 여부를 예측

### 2 모델 기반 고객 군집 분석

사용자 로그 데이터와 기본적인 사용자 정보를 이용하여 SOM 군집분석 이후 각 군집의 특징들을 파악 및 맞춤 메시지 제안

02 데이터 탐색

## 02 데이터 탐색

loan\_result log\_data user\_spec

### • 전처리 과정

application_id	loan_apply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied
1748340	2022-06-07 13:05:41	7	191	4.2e+07	13.6	NA

사용하지 않는 행  
제거

변수 변환

변수 추가

완성

대출 한도 / 대출 이자가 결측된 행  
유저 스펙 데이터에 없는 application\_id를 가진 행

loan\_apply\_insert\_time에서  
날짜만 추출하여 사용

loan\_limit이 1억보다 큰 경우를 나타내는 변수  
application\_id에 해당하는 user\_id 변수

application_id	loan_apply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	loan_limit_100million	date_cd	user_id
1748340	2022-06-07 13:05:41	7	191	4.2e+07	13.6	NA	FALSE	2022-06-07	430982

## 02 데이터 탐색

loan\_result log\_data user\_spec

### • 전처리 과정

application_id	loan_apply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	loan_limit_100million	date_cd	user_id
2157865	2022-05-09 08:44:59	54	235	2e+07	16.5	1	FALSE	2022-05-09	346970

변수 삭제

변수 변환

데이터 병합

완성

유의하지 않아 보이는  
운영체제, 어플리케이션 버전 변수 삭제

유저별 본인인증완료,  
한도조회결과확인 행동 횟수를 카운트

대출과 관련있는 행동인 카운트수 추가  
loan\_result 데이터에

application_id	loan_apply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	loan_limit_100million	date_cd	user_id	Complete IDCertification	EndLoanApply
2157865	2022-05-09 08:44:59	54	235	2e+07	16.5	1	FALSE	2022-05-09	346970	5	13

## 02 데이터 탐색

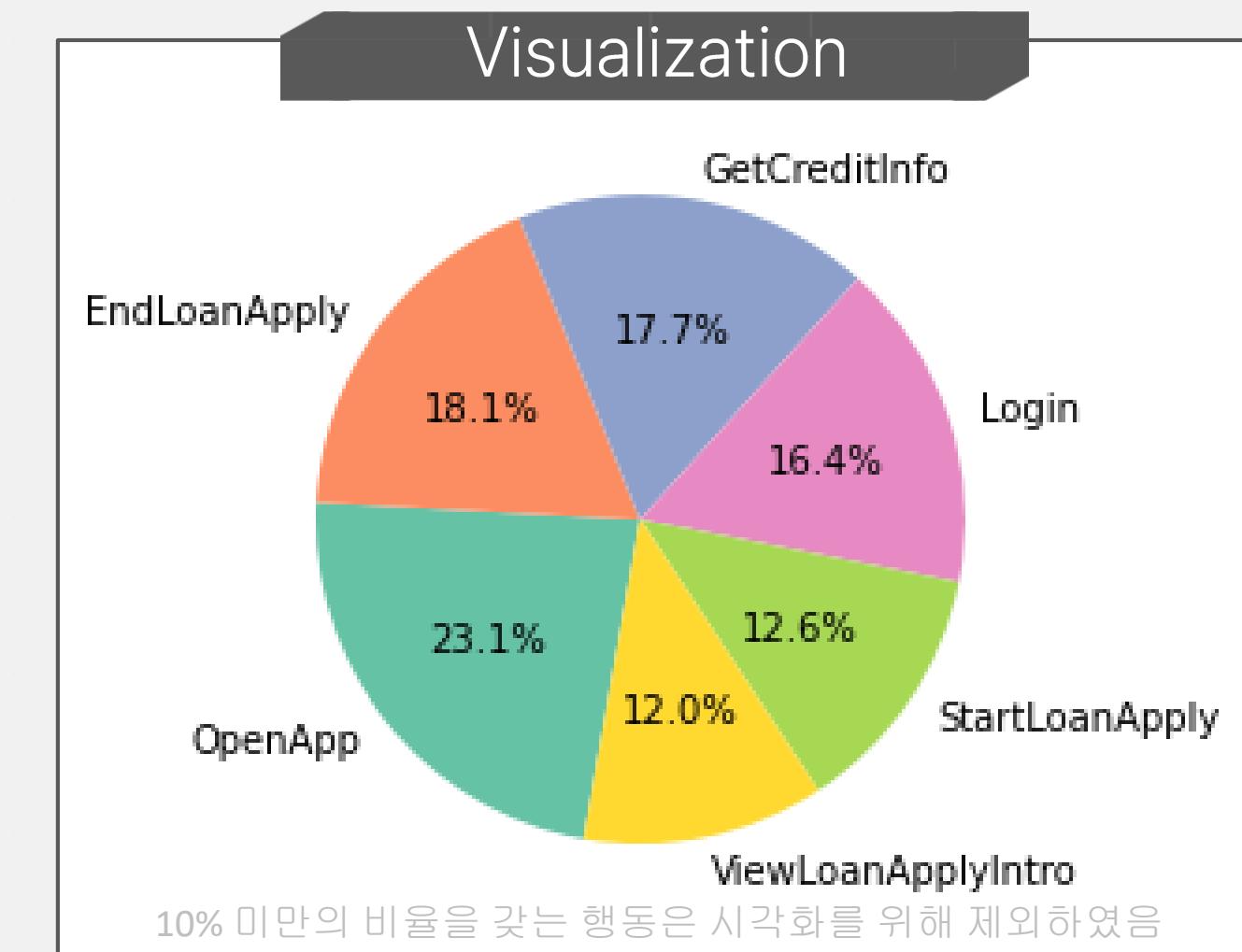
loan\_result

log\_data

user\_spec

### • 데이터 탐색

event_new	event description
SignUp	회원가입
OpenApp	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



- 한번만 하면 되는 회원가입과 본인인증완료, 한도조회 결과 확인 이후의 대출관리, 여윳돈 계산기 서비스 등은 낮은 비율을 보임
- 앱 실행부터 한도조회 시작하기까지의 일련의 과정은 비슷한 비율을 보임

## 02 데이터 탐색

loan\_result log\_data user\_spec

### • 전처리 과정

- 생년, 성별이 동시에 결측  
근로 형태, 고용형태, 주거소유형태, 목적 동시에 결측
- 근로 형태, 목적, 고용 형태, 대출희망금액, 주거소유형태 missing인 경우를 loan\_result.csv에서 찾을 수 없음 -> 삭제
- 대출상품 특성에 입각하여 합리적 방안으로 결측치 대체, 결과적으로 원본 데이터 손실을 최소화 함으로써 예측이 향상될 것이라고 기대함
  - user\_spec에 가장 결측된 자료가 많음, 적절한 수단을 통하여 대체
  - 개인회생자 여부와 개인회생자 납입 완료 여부가 결측인 경우 -> 해당 없음으로 판단, 각각을 0,1로 대체
  - 기대출수 = 0 인 자료가 없음 -> 결측된 기대출수에 0으로 대체

birth_year	12961	gender	12961	insert_time	0
credit_score	105115	yearly_income	90	income_type	85
company_enter_mo nth	171760	employment_type	85	houseown_type	85
desired_amount	85	purpose	85	personal_rehabilita tion_yn	587461
personal_rehabilita tion_complete_yn	1203354	existing_loan_cnt	198556	existing_loan_amt	313774

< 각 변수 별 결측치 수 >

# 02 데이터 탐색

loan\_result log\_data user\_spec

## • 전처리 과정

- 생년, 성별 : 한 user\_id에 전부는 결측이 되지 않은 경우, 결측되지 않은 행으로 대체 / 전부 결측된 경우, 연 수입이 비슷한 사람의 값들 (평균, 최빈값)로 대체
- 기대출금액 : 기대출횟수에 맞춰서 대체 (기대출금액 ~  $1.2e6 * \text{기대출횟수}$ )
- 신용 점수 : 한 개인의 신용 점수 중 결측 시기와 가장 비슷한 시기로 대체, 한 개인의 신용 점수가 모두 결측인 경우 가장 비슷한 연 소득을 가진 사람으로 대체
- 입사연월 : 대출 신청 시기와 입사연월의 차이를 통해 company\_time 변수 생성

성별 대체 예시

user_id <int>	gender <int>
49072	NA
49072	0
49072	0
49072	0
49072	NA
49072	0
49072	0
49072	0
49072	NA
49072	NA



user_id <int>	gender <dbl>
49072	0
49072	0
49072	0
49072	0
49072	0
49072	0
49072	0
49072	0
49072	0
49072	0

Company\_time 생성 예시

date_cd <chr>	company_enter_month <int>	company_time <dbl>
2022-03-12	201511	76
2022-03-04	201511	76
2022-05-03	201511	78
2022-05-31	201511	78
2022-04-02	201511	77
2022-03-28	201511	76
2022-03-02	201511	76
2022-06-02	201511	79
2022-05-06	201511	78
2022-04-18	201511	77

## 02 데이터 탐색

loan\_result log\_data user\_spec

- **데이터 통합** 시각화를 위해 여러 변수를 제외하였음.

is_applied	loan_limit	loan_rate	loan_limit_100million	CompleteIDCertification	EndLoanApply	credit_score	yearly_income	income_type
1	20000000.0	16.5	False	5	13	540.0	32000000.0	EARNEDINCOME
0	11000000.0	16.5	False	12	21	580.0	72000000.0	EARNEDINCOME
0	3000000.0	20.0	False	12	21	580.0	72000000.0	EARNEDINCOME
0	10000000.0	13.5	False	3	29	740.0	39000000.0	EARNEDINCOME
0	22000000.0	15.9	False	3	29	740.0	39000000.0	EARNEDINCOME

1

application\_id를 기준으로  
loan\_result.csv와 user\_spec.csv를 묶음

2

application\_id, user\_id는  
암호화된 숫자이므로 설명변수에서 제외

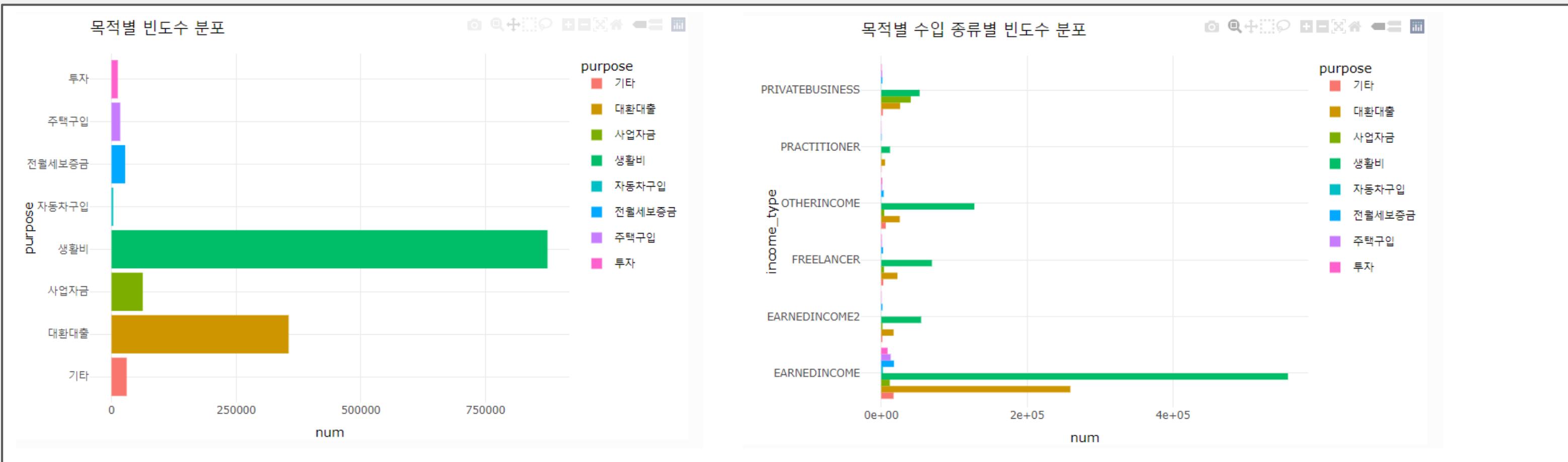
3

통합된 자료를 test, train 자료로 나누어 사용

# 02 데이터 탐색

loan\_result log\_data user\_spec

## 데이터 탐색 : purpose

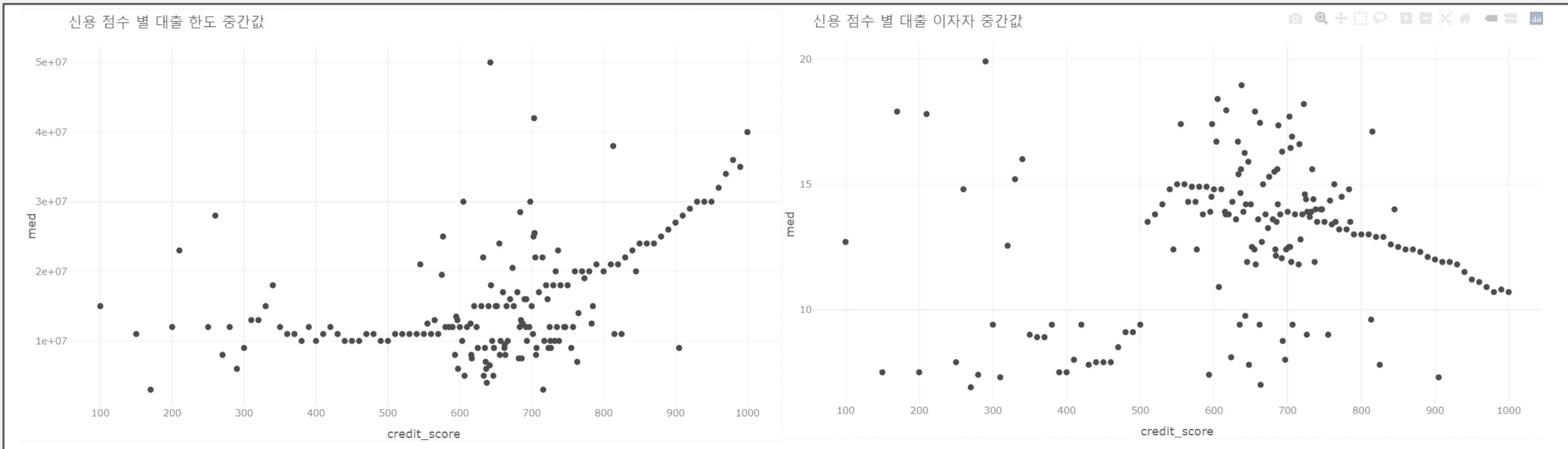


- 생활비와 대환대출의 비율이 상당히 높음
- 개인사업자의 경우, 사업자금 목적으로 받는 대출 비율이 유의미하게 높음

## 02 데이터 탐색

loan\_result log\_data user\_spec

### 데이터 탐색 : 신용 점수 vs 대출 조건

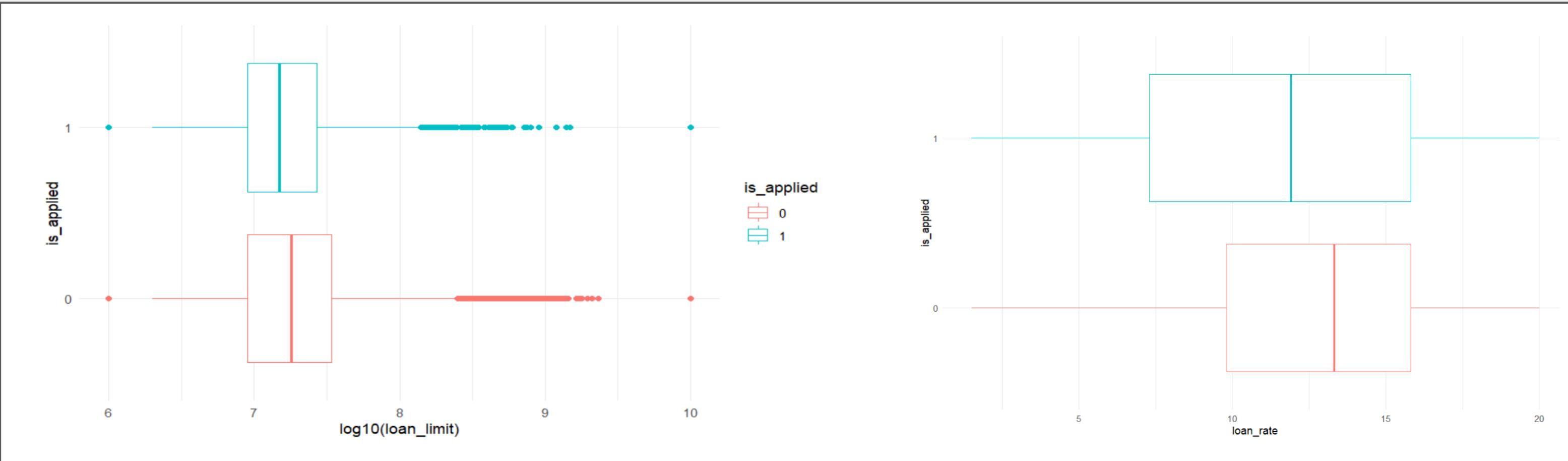


- 대출 한도와 이자의 outlier를 고려하여 median 사용. 신용 점수가 낮은 구간은 큰 경향성 보이지 않음
- 신용 점수가 일정 기준 이상일 때, 신용점수가 높을 수록 대출 한도는 증가, 대출 이자는 감소하는 경향을 보임.

## 02 데이터 탐색

loan\_result log\_data user\_spec

### 데이터 탐색 : 신청 여부 vs 대출 조건



- 대출 한도에 log 변환을 해줌에도 불구하고, outlier들이 많이 보임
- 다른 변수들을 무시하고서, 대출 한도와 대출 이자에 따라 신청 여부를 판단하기 어려워보임

03

# 대출신청 예측 모델링

# 03 대출신청 예측 모델링

목표      사용 데이터      모델

## • 유저스펙 데이터

: 데이터 탐색 결과 대출여부와 관련이 있는 대부분의 유저스펙 데이터의 컬럼을 추출하여 사용

birth_year	gender	credit_score	yearly_income	income_type	employment_type	houseown_type	desired_amount	purpose	personal_rehabilitation_yn	personal_rehabilitation_complete_yn	existing_loan_cnt	existing_loan_amt
유저 생년월일	유저 성별	한도조회 당시 유저 신용점수	연소득	근로형태	고용형태	주거소유형태	대출희망금액	대출 목적	개인회생자 여부	개인회생자 납입 완료 여부	기대출수	기대출금액



is_applied	loan_limit	...	birth_year	gender	credit_score	yearly_income	income_type	employment_type	houseown_type	desired_amount	purpose	personal_rehabilitation_yn	personal_rehabilitation_complete_yn	existing_loan_cnt	existing_loan_amt
0	0	...	1988	1	640	28000000	EARNEDINCOME	정규직	기타가족소유	5000000	생활비	0	0	6	35000000
1	0	...	1990	0	700	30000000	FREELANCER	기타	전월세	15000000	전월세보증금	0	1	1	10000000

→ Dummy variable로 처리

# 03 대출신청 예측 모델링

목표      사용 데이터      모델

## • 대출상품 결과테이블

: 데이터 탐색 결과 대출 여부와 관련이 있는 컬럼들을 일부 가공하여 사용

application_id	신청서 번호
loanapply_insert_time	한도조회 일시
bank_id	금융사 번호
product_id	상품 번호
loan_limit	승인한도
loan_rate	승인금리
is_applied	신청 여부(타겟)



is_applied	loan_limit	loan_rate	loan_limit_>100million	...
0	0	24000000	17.1	...
1	0	14000000	20	...

대출한도가 1억이 넘는지 여부  
1:초과, 0:미만



# 03 대출신청 예측 모델링

목표

사용 데이터

모델

## • 유저로그 데이터

: 행동명 중 본인인증완료, 한도조회 결과 확인 컬럼 사용

user_id	유저 번호
event	행동명
timestamp	행동일시
date_cd	일 코드



이벤트			
is_applied	...	CompleteIDCertification	EndLoanApply
0	...	FALSE	2
1	...	FALSE	1

# 03 대출신청 예측 모델링

Random Forest   Xgboost   Ensemble

## Random Forest

훈련 과정에서 구성한 다수의 의사결정트리를 만드는, 트리들의 상관성을 제거하는 간단한 방법

### 모델의 특성

- 배깅과 부스팅보다 더 많은 무작위성을 주어 여러 개의 weak learner인 의사결정트리를 생성하고, 이를 결합해 최종적으로 종속변수를 분류함
- Voting 방법을 적용함

### 모델 선택이유

- 많은 변수를 제거 없이 실행되고, 변수의 중요도 또한 확인할 수 있음

### 모델링 방법

- Grid search를 통한 f1-score 기준 최적의 하이퍼파라미터 생성

Hyperparameter	Meaning	Value
max_depth	트리의 최대 depth	10
min_samples_leaf	Leaf node에서 최소 데이터 개수	3
min_samples_split	Internal node에서 최소 데이터 개수	2
n_estimators	의사결정트리 갯수	200

# 03 대출신청 예측 모델링

Random Forest   XGBoost   Ensemble

## XGBoost *eXtreme Gradient Boost*

Gradient Boost 알고리즘에 병렬 학습이 지원되도록 하여 정확도와 속도를 높인 모델

### 모델의 특성

- 여러 개의 soft learner를 순차적으로 학습시켜 hard learner를 만드는 양상을 기법인 Boosting의 파생 모델
- 잘 못 예측한 데이터에 가중치를 부여하여 gradient를 개선해 나감

### 모델 선택이유

- 과적합 방지가 잘 되어있기 때문에(파라미터  $\gamma$ ,  $\lambda$ 가 존재) 예측력이 높음

### 모델링 방법

- Grid search를 통한 최적의 하이퍼파라미터 생성
- 모델 성능을 높이고 과적합을 방지하기 위해 탐색한 최적 하이퍼파라미터 조합은 아래와 같음 (이외에는 디폴트 값 사용)

Hyperparameter	Meaning	Value
n_estimators	학습기 갯수	200
learning_rate	학습률	0.2
gamma	Leaf Node 추가 분할 결정기준	1

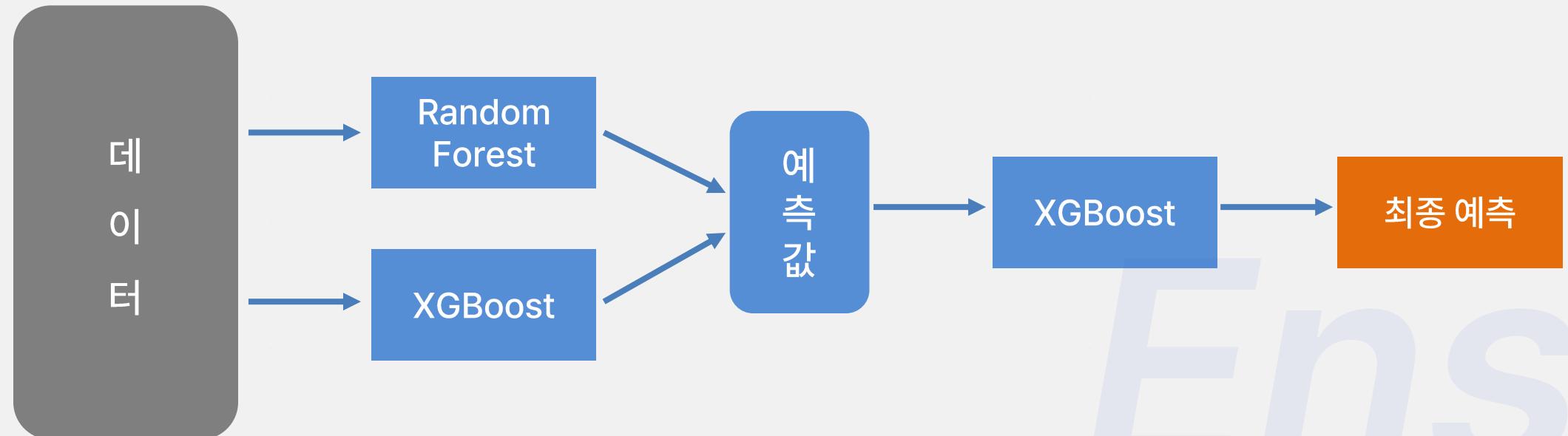
# 03 대출신청 예측 모델링

Random Forest XGBoost Ensemble

## Stacking Ensemble *Stacked generalization Ensemble*

여러 알고리즘에서 얻은 예측 값을 모델의 학습 데이터로 하여 일반화된 최종모델을 구성하는 방법

### 모델의 구성



### 모델의 특징

- 개별적인 여러 알고리즘을 서로 결합해 예측 결과를 도출한다는 점에서 배깅 및 부스팅과 공통점이 있음
- 일반적으로 성능이 비슷한 모델을 결합하면 더 나은 성능을 보여줌
- 너무 많은 모델을 결합할 시 과적합 위험이 있음

# 03 대출신청 예측 모델링

Random Forest XGBoost Ensemble 최종 모델

Model	단일 모델		스태킹
	Random Forest	XGBoost	XGBoost
Parameters used	<b>max_depth : 10 min_samples_leaf : 3 min_samples_split : 2 n_estimators : 200</b>	<b>n_estimators : 200 learning_rate : 0.2 gamma : 1</b>	<b>n_estimators : 200 learning_rate : 0.2 gamma : 1 Tree_method : 'exact'</b>
F1 score (cross validation)	<b>0.4861</b>	<b>0.5520</b>	<b>0.5558</b>
Accuracy score	<b>0.9459</b>	<b>0.9468</b>	<b>0.9469</b>

최종적으로 F1 스코어와 Accuracy 스코어가 가장 높게 나온, '스태킹 모델'을 선정

04

# 고객 구집 분석

# 04 고객 군집 분석

사용 데이터

SOM

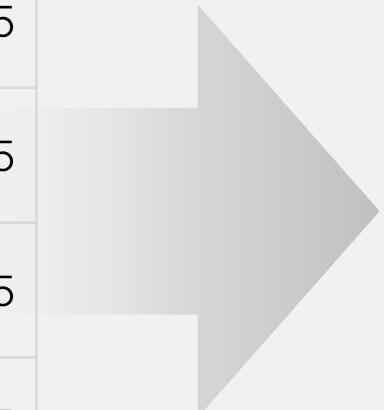
군집 특징 분석

서비스 메시지 제안

## • 유저로그 데이터

: 유저가 최종적으로 한 행동을 보기 위해 유저별로 관측된 이벤트를 통합한 후, 출생연도 / 성별 추가

user_id	event	timestamp	mp_os	mp_app_version	date_cd
576409	StartLoanApply	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
576409	ViewLoanApplyIntro	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
72878	OpenApp	2022-03-25 11:14:44	Android	3.8.4	2022-03-25
645317	UseLoanManage	2022-03-25 11:15:11	iOS	3.6.1	2022-03-25
645317	UseLoanManage	2022-03-25 11:41:53	iOS	3.6.1	2022-03-25



user_id	이벤트				Birth_year	Gender
	SignUp	...	GetCreditInfo	...		
576409	1	...	1	...	1969	1
72878	1	...	1	...	1969	1
645317	1	...	1	...	1957	1

어플 실행을 위해 입력해야 하는 필수 정보인

출생연도 / 성별 변수 추가



타임스탬프는 고려하지 않고 유저별로

각각의 이벤트를 행동했으면 1, 그렇지 않으면 0으로 인코딩

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

## SOM *Self Organized Map*

차원 축소와 군집화를 동시에 수행하는 기법

구체적으로 인공신경망과 유사한 방식의 학습을 통해 저차원 격자에 고차원 데이터를 대응시킴으로써 군집을 도출해내는 방법

### ✓ SOM 사용 근거

#### • 패턴 탐색

- 차원 축소를 통해 2차원 상에 클러스터를 생성함으로써 데이터가 암시하는 패턴을 탐색하는 데에 용이

#### • 계산 속도

- feed-forward flow 를 사용하여 속도가 빠르고, 이에 따라 큰 용량의 데이터도 비교적 빠르게 계산 가능

### ✓ 클러스터링 절차

Min-max scaling 을 통해 변수 간의 영향력을 통일



클러스터링을 진행한 후 여러 플랏을 통해 모델의 적절성을 확인한 다음,

각 클러스터별로 변수의 분포를 확인함으로써 고객의 특성 파악

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

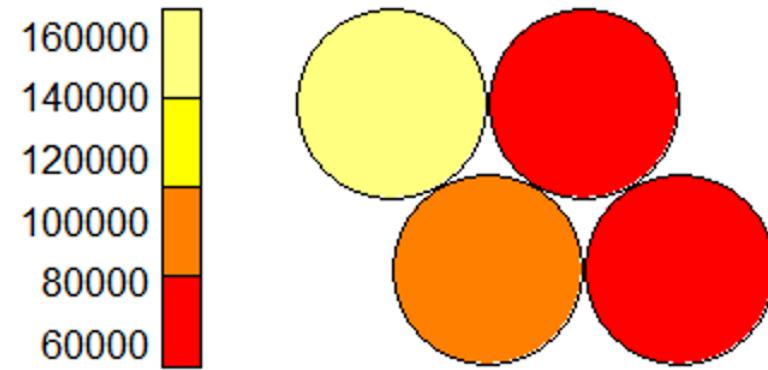
서비스 메시지 제안



플랫 확인

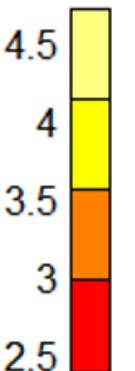
클러스터별 관측치 개수

**Node Counts**

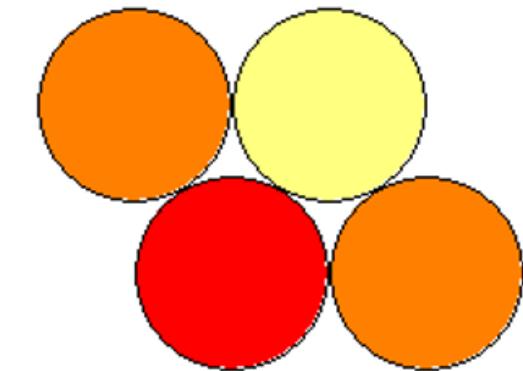


클러스터 사이의 거리

**SOM neighbour distances**

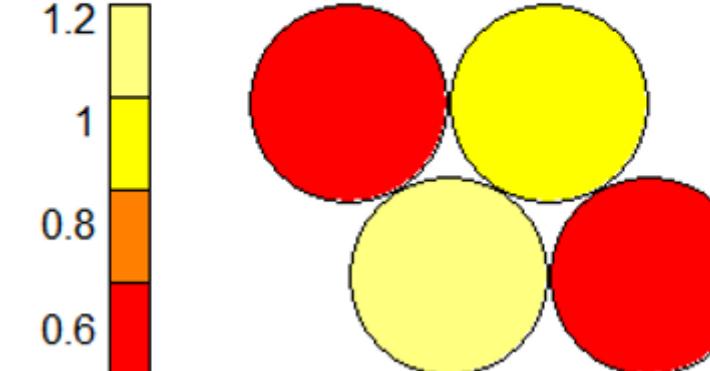
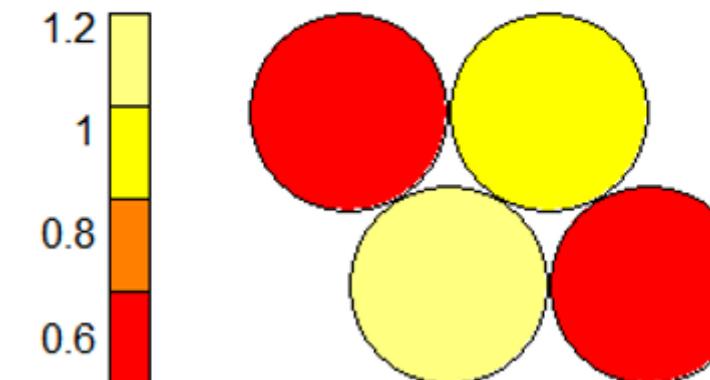


**SOM neighbour distances**



클러스터내 관측치 간의 유사도

**mapping quality**



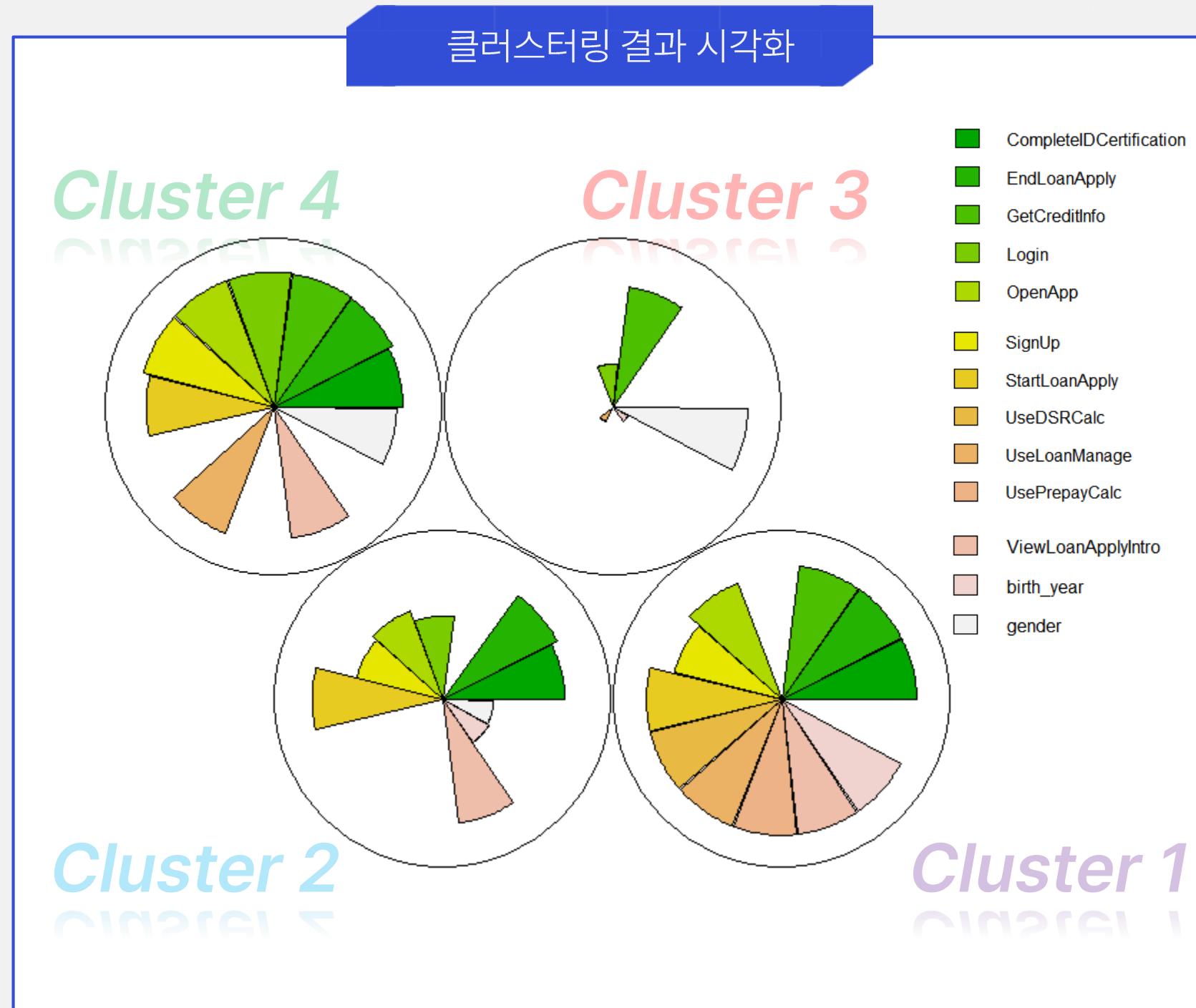
# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안



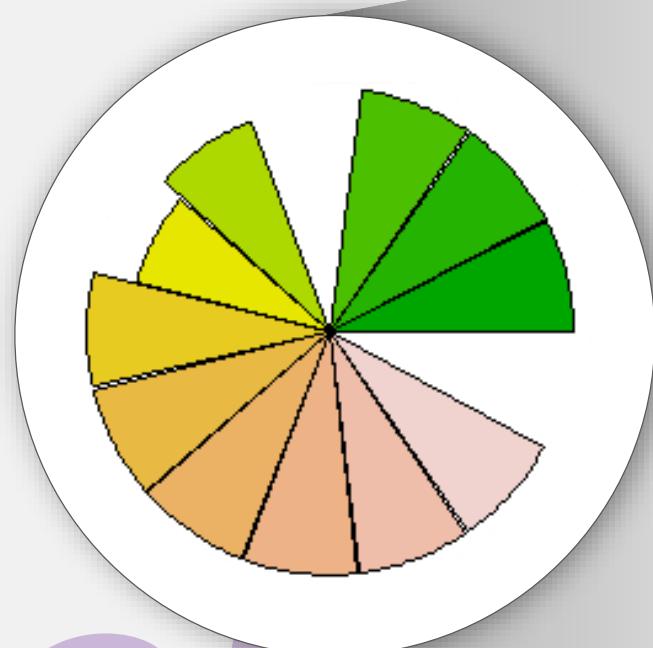
# 04 고객 군집 분석

사용 데이터

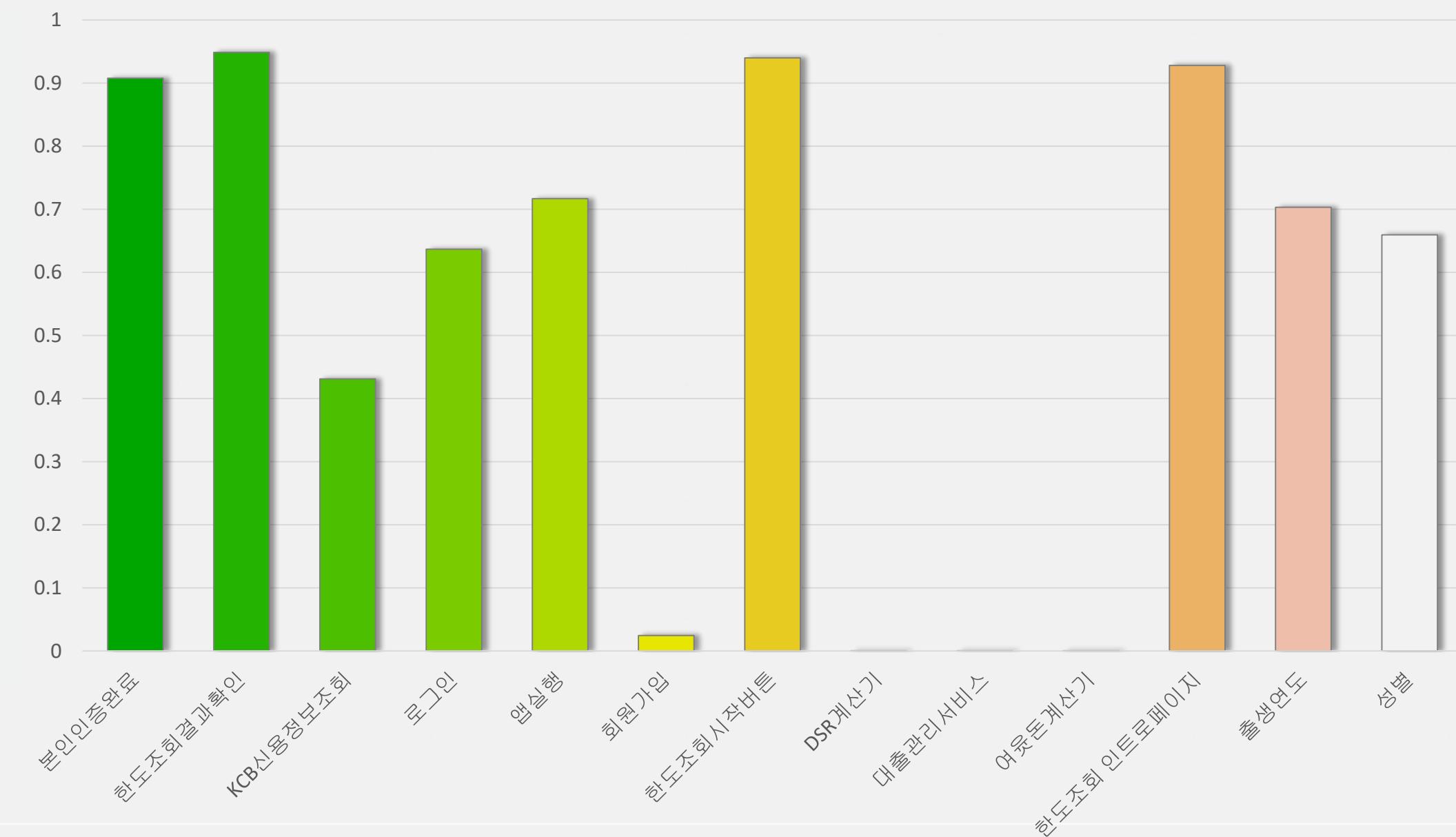
SOM

군집 특징 분석

서비스 메시지 제안



Cluster 1  
Overview



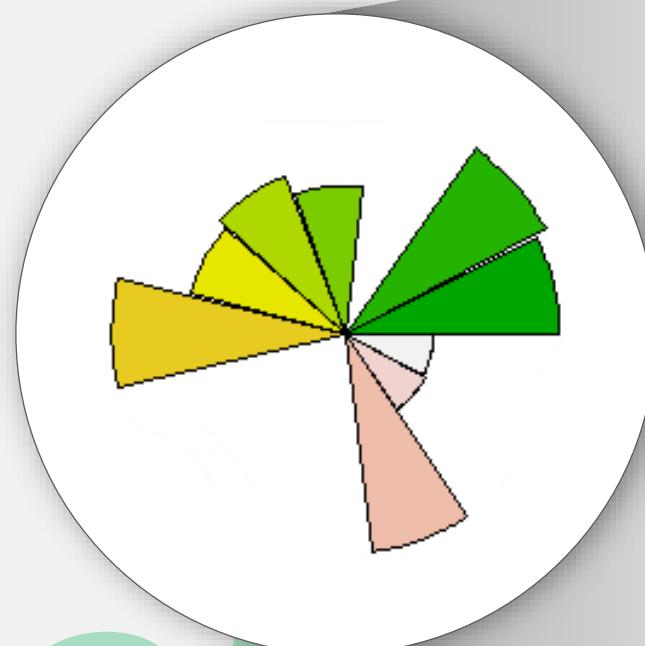
# 04 고객 군집 분석

사용 데이터

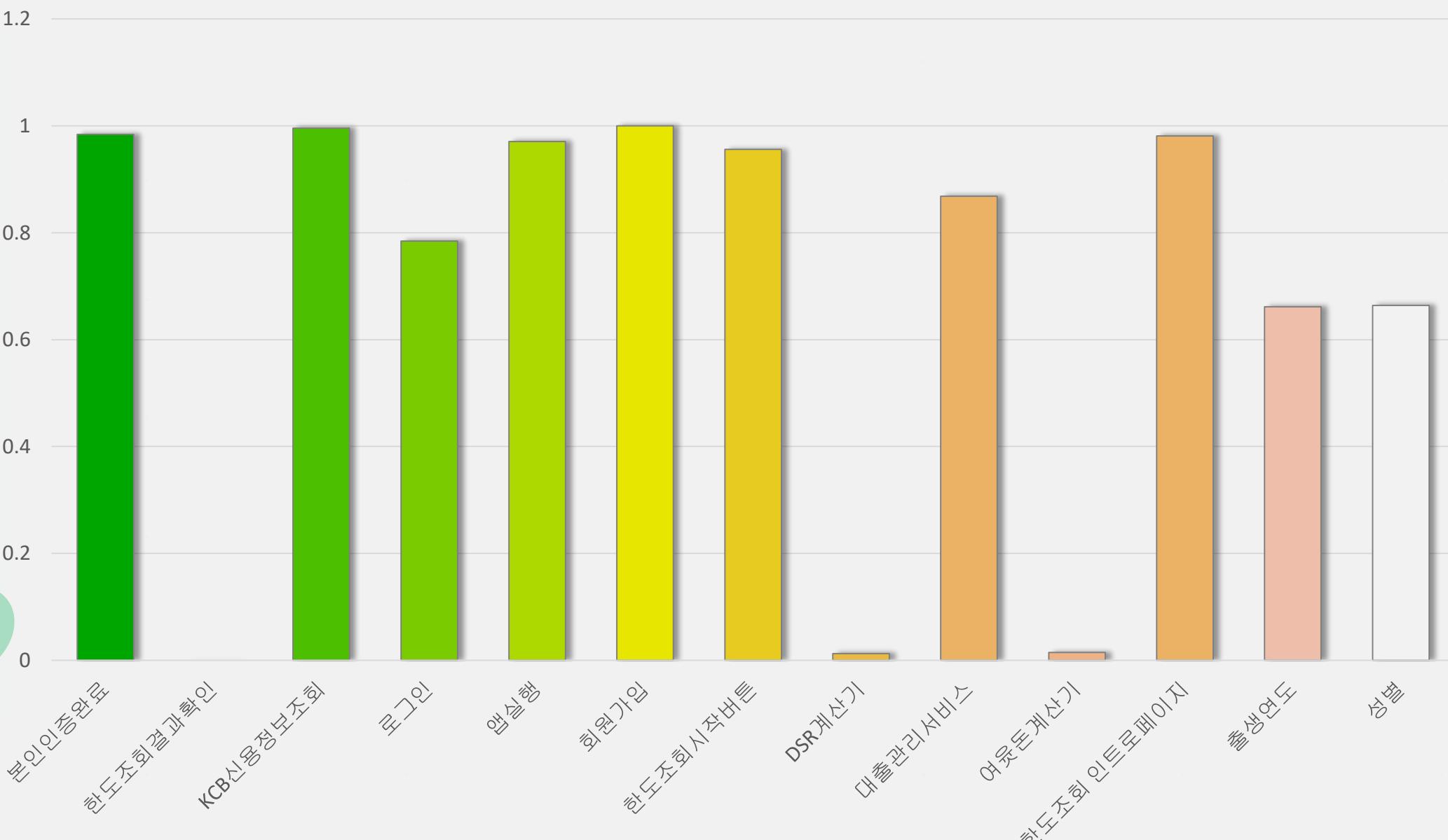
SOM

군집 특징 분석

서비스 메시지 제안



Cluster 2  
evidence



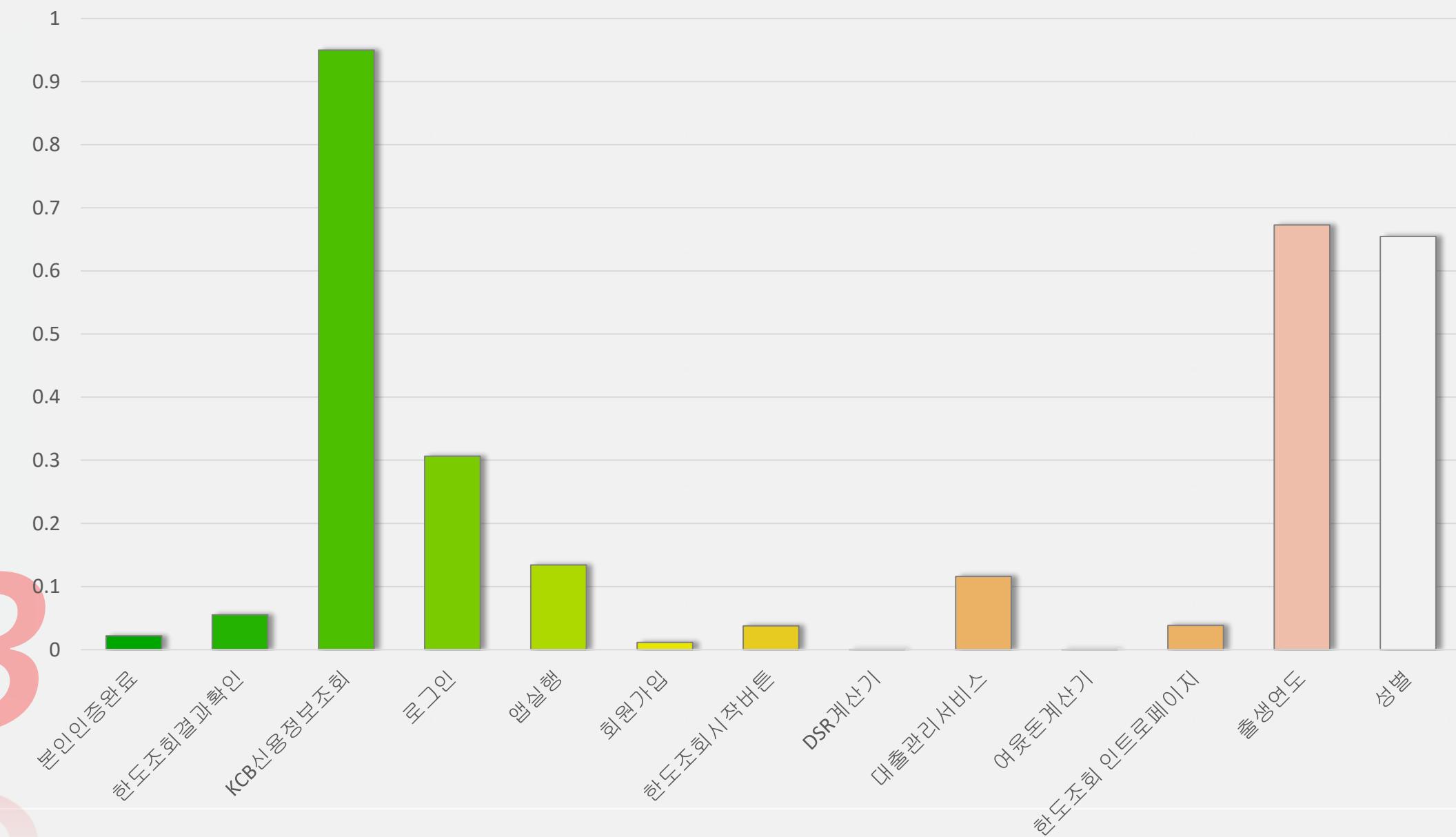
# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안



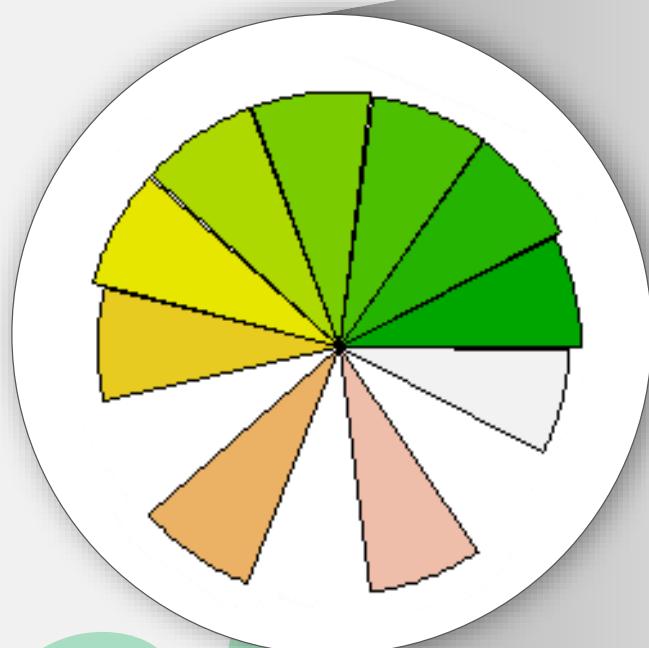
# 04 고객 군집 분석

사용 데이터

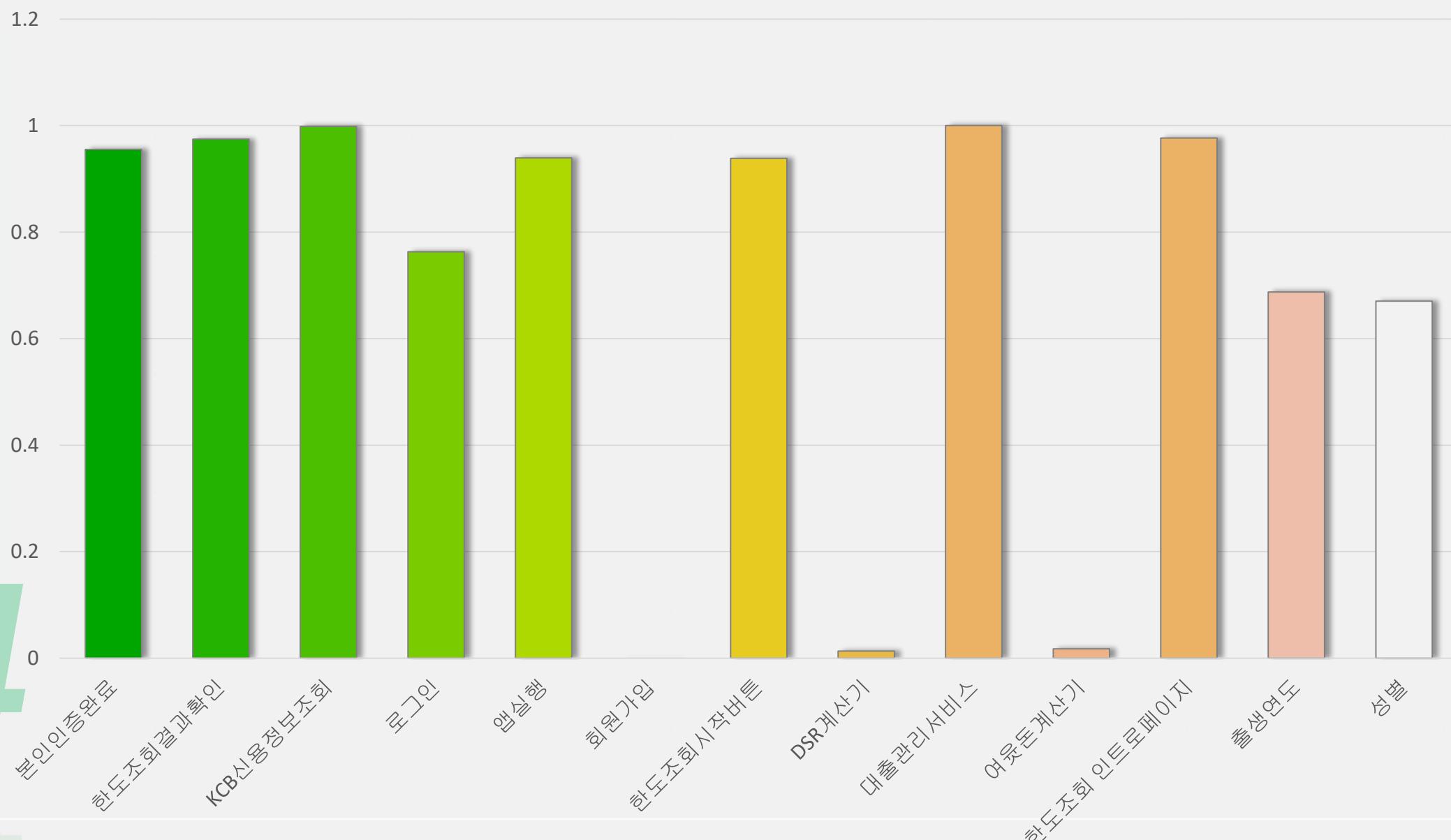
SOM

군집 특징 분석

서비스 메시지 제안



Cluster 4



# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

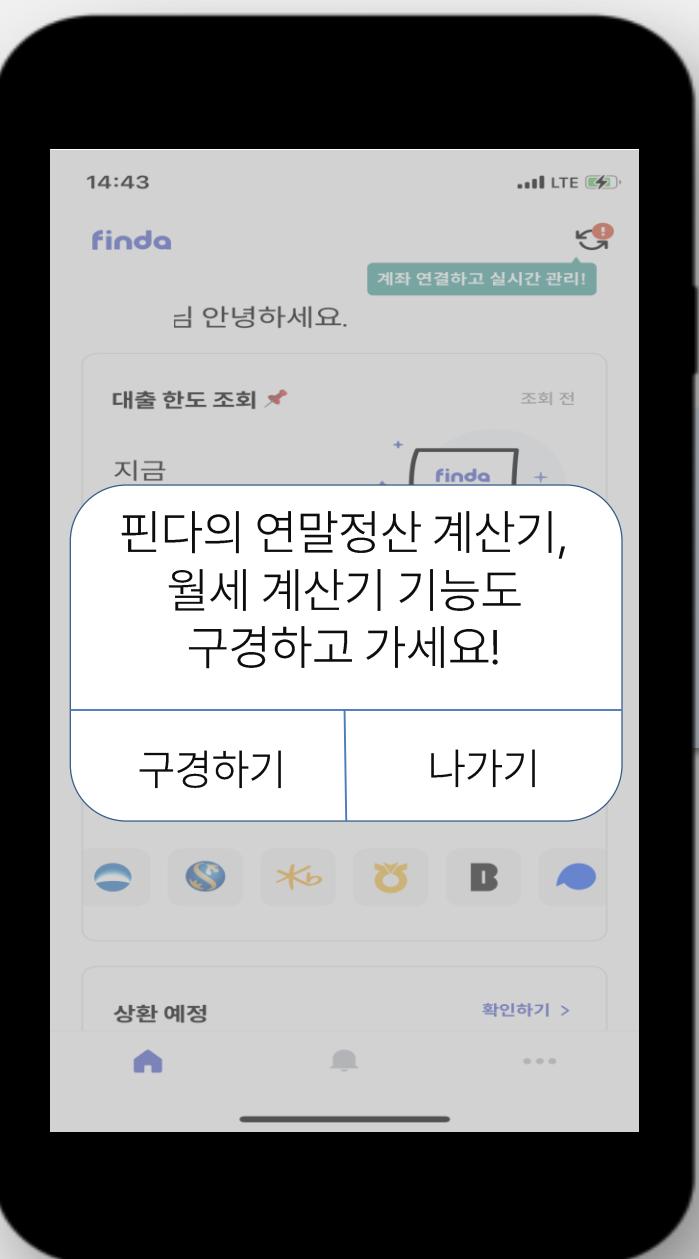
특징

Cluster 1

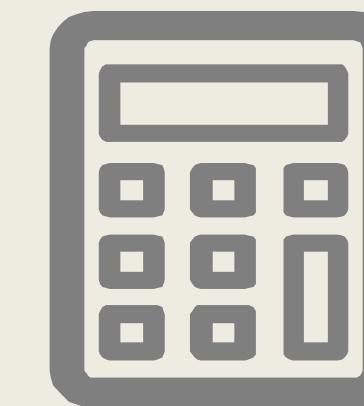
변수	그룹1
연령	청년
본인인증완료	많음
한도조회 결과 확인	많음
KCB 신용정보 조회	적당
핀다 앱 로그인	적당
핀다 앱 실행	적당
회원가입	적당
한도조회 시작하기 버튼 클릭	많음
DSR 계산기 서비스 이용	적음
대출관리 서비스 이용	적음
여윳돈 계산기 서비스 이용	적음
한도조회 인트로 페이지 조회	많음

첫번째 그룹은 청년층 그룹이고, 그룹의 크기가 작다. 이들은 한도조회 결과 확인은 많이 하지만 그 이후의 대출관리 서비스나 DSR 계산기, 여윳돈 계산기 기능을 잘 이용하지 않는 것으로 나타난다.

서비스 메시지 제안



기대 효과



대출관리 서비스나 DSR 계산기, 여윳돈 계산기가 아닌 청년층이 관심을 가질 만한 finda의 다른 서비스를 홍보함으로써 청년층의 활발한 앱 사용을 기대할 수 있음

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

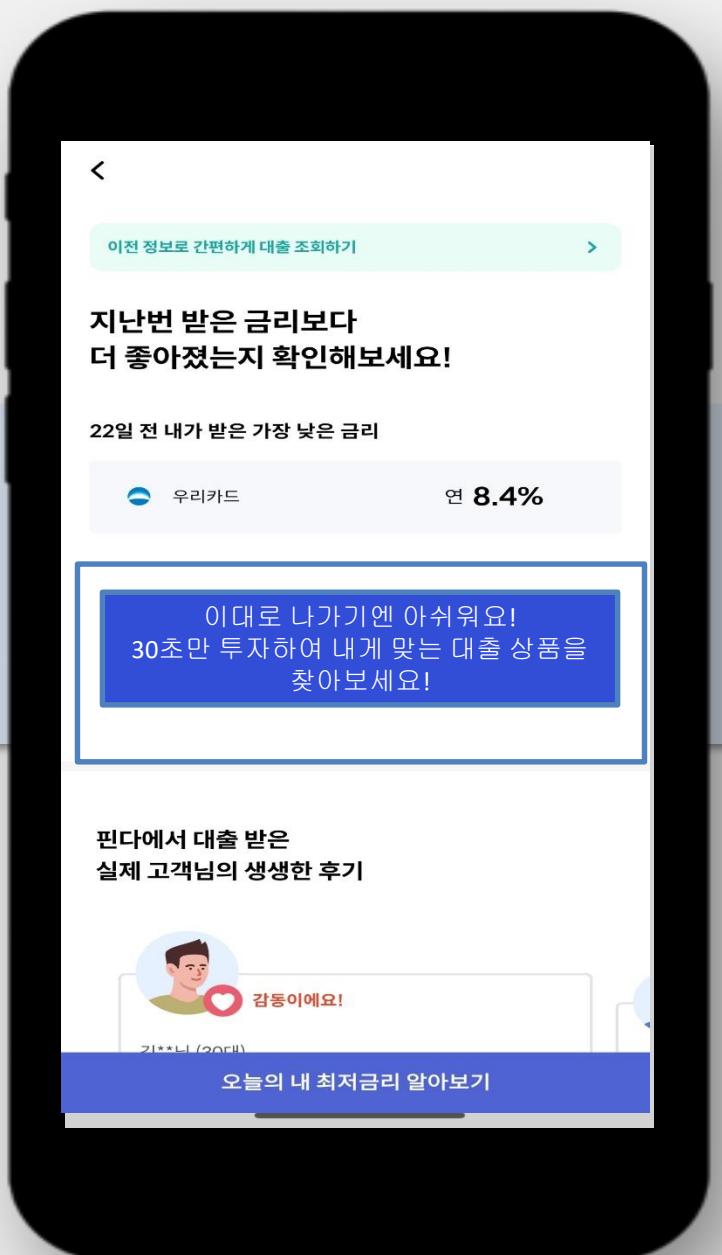
특징

## Cluster 3

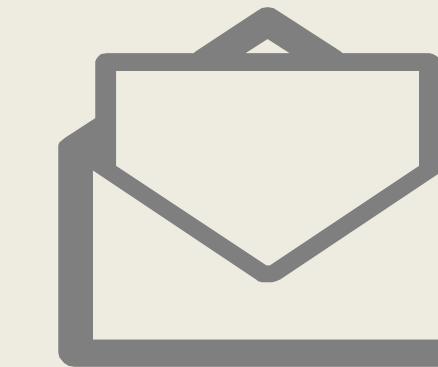
변수	그룹3
연령	중년
CompleteIDCertification (본인인증완료)	적음
EndLoanApply (한도조회 결과 확인)	적음
GetCreditInfo (KCB 신용정보 조회)	적음
Login (핀다 앱 로그인)	적음
OpenApp (핀다 앱 실행)	많음
SignUp (회원가입)	적음
StartLoanApply (한도조회 시작하기 버튼 클릭)	많음
UseDSRCalc (DSR 계산기 서비스 이용)	적음
UseLoanManage (대출관리 서비스 이용)	적음
UsePrepayCalc (여윳돈 계산기 서비스 이용)	적음
ViewLoanApplyIntro (한도조회 인트로 페이지 조회)	적음

- 세번째 클러스터
- 중년층
- 대부분의 앱 행동을 실행하지 않음
- 그룹원수(클러스터에 속한 사람 수)가 적다

### 서비스 메시지 제안



기대 효과



적은 시간 투자로 대출 상품을 알아 볼 수 있음을 한도조회 시작하기 버튼 누른 후 팝업 메시지로 띄움으로써 앱 사용이 어려운 고객의 앱 사용을 기대할 수 있음

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

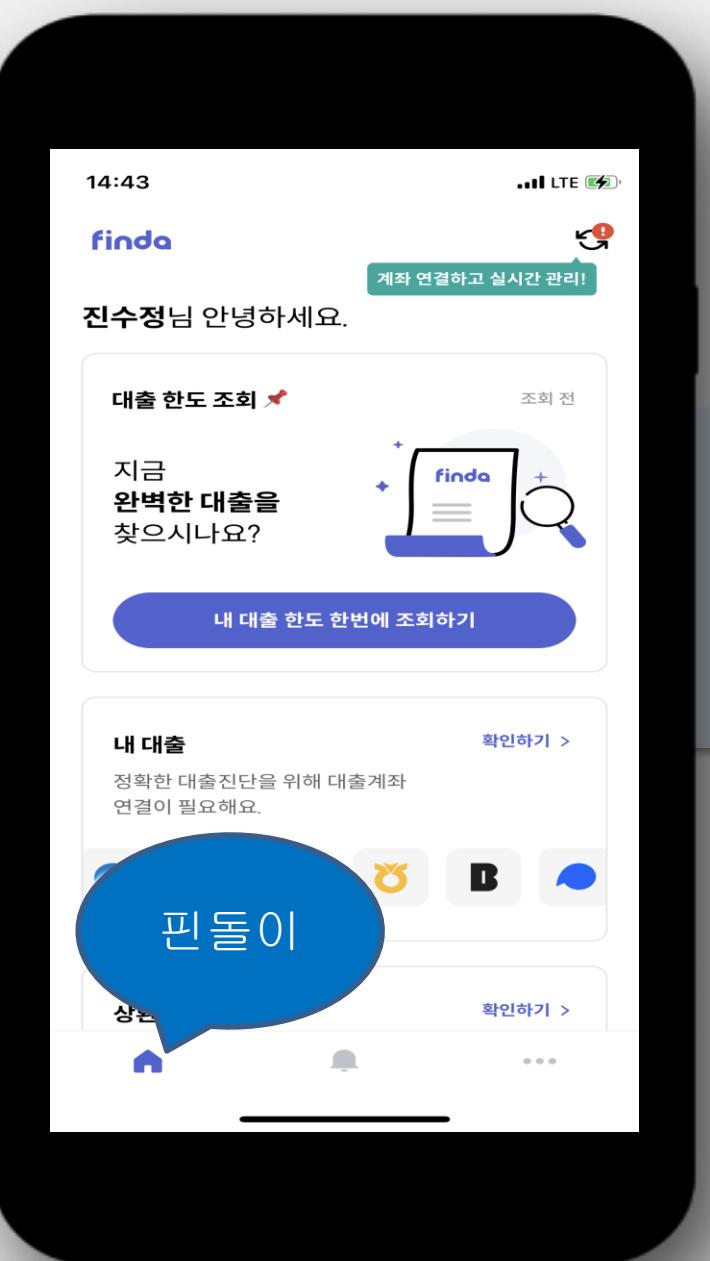
특징

Cluster 3

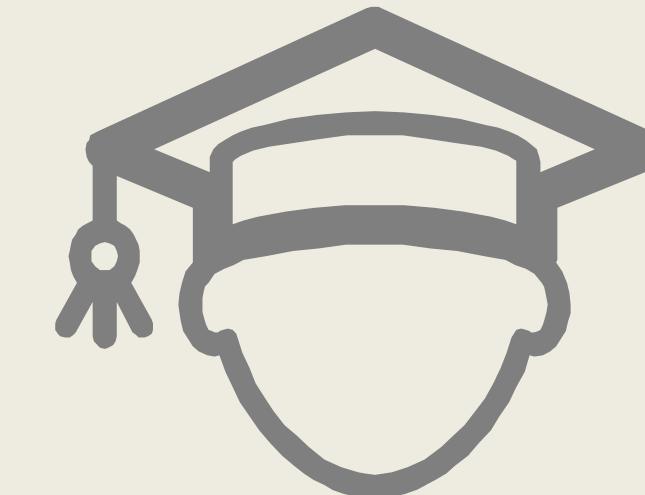
변수	그룹3
연령	중년
CompleteIDCertification (본인인증완료)	적음
EndLoanApply (한도조회 결과 확인)	적음
GetCreditInfo (KCB 신용정보 조회)	적음
Login (핀다 앱 로그인)	적음
OpenApp (핀다 앱 실행)	많음
SignUp (회원가입)	적음
StartLoanApply (한도조회 시작하기 버튼 클릭)	많음
UseDSRCalc (DSR 계산기 서비스 이용)	적음
UseLoanManage (대출관리 서비스 이용)	적음
UsePrepayCalc (여윳돈 계산기 서비스 이용)	적음
ViewLoanApplyIntro (한도조회 인트로 페이지 조회)	적음

- 세번째 클러스터
- 중년층
- 대부분의 앱 행동을 실행하지 않음
- 그룹원수(클러스터에 속한 사람 수)가 적다

서비스 메시지 제안



기대 효과



앱봇을 통한 스텝별 도움 제공을 통해 앱 사용 방법이 어려워 앱 설치만 하고 이용하지 못한 잠재 고객을 이용고객으로 확보할 수 있음

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

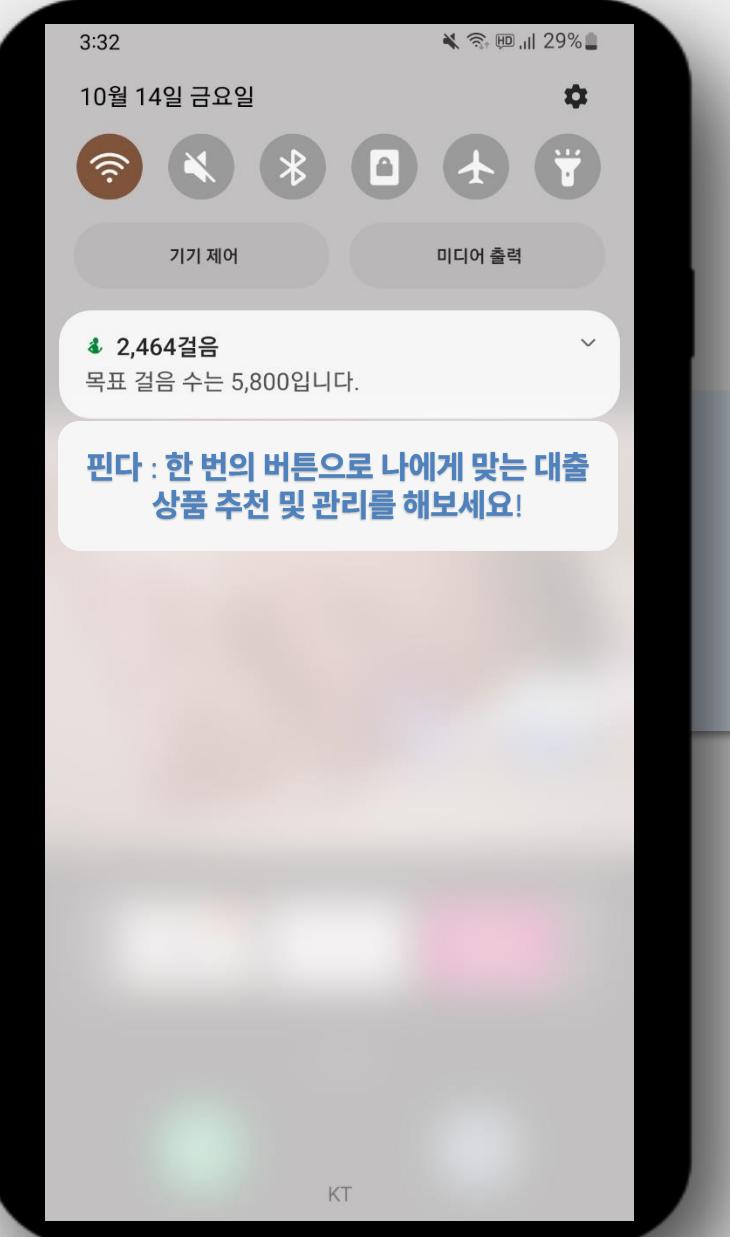
특징

Cluster 3

변수	그룹3
연령	중년
CompleteIDCertification (본인인증완료)	적음
EndLoanApply (한도조회 결과 확인)	적음
GetCreditInfo (KCB 신용정보 조회)	적음
Login (핀다 앱 로그인)	적음
OpenApp (핀다 앱 실행)	많음
SignUp (회원가입)	적음
StartLoanApply (한도조회 시작하기 버튼 클릭)	많음
UseDSRCalc (DSR 계산기 서비스 이용)	적음
UseLoanManage (대출관리 서비스 이용)	적음
UsePrepayCalc (여윳돈 계산기 서비스 이용)	적음
ViewLoanApplyIntro (한도조회 인트로 페이지 조회)	적음

- 세번째 클러스터
- 중년층
- 대부분의 앱 행동을 실행하지 않음
- 그룹원수(클러스터에 속한 사람 수)가 적다

서비스 메시지 제안



기대 효과



앱을 설치 후, 앱을 사용하지 않은 고객들을 팝업 메시지를 통해 앱 사용 장려할 수 있다.

# 04 고객 군집 분석

사용 데이터

SOM

군집 특징 분석

서비스 메시지 제안

## Cluster 2, 4

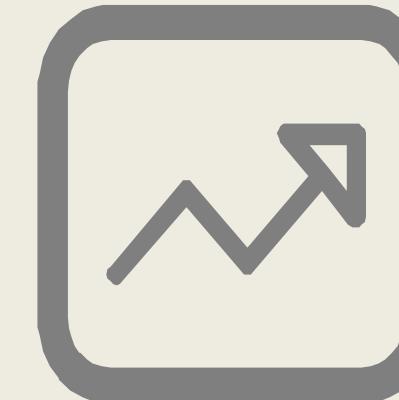
특징

	변수	그룹2	그룹4
연령	장년	중년	
본인인증완료	많음	많음	
한도조회 결과 확인	많음	많음	
KCB 신용정보 조회	많음	많음	
핀다 앱 로그인	많음	많음	
핀다 앱 실행	많음	적음	
회원가입	많음	적음	
한도조회 시작하기 버튼 클릭	많음	많음	
DSR 계산기 서비스 이용	적음	적음	
대출관리 서비스 이용	많음	많음	
여윳돈 계산기 서비스 이용	적음	적음	
한도조회 인트로 페이지 조회	많음	많음	

두번째, 네번째 그룹은 어플 실행 빈도가 높은 중장년층으로, 금융 계산기 기능을 제외한 모든 기능을 골고루 이용한다.  
어플 실행의 차이를 제외하고 두 그룹의 차이가 없어, 하나의 그룹으로 간주한다.



기대 효과



충분히 finda 어플을 사용하는 고객들에게 잘 보이지 않았던 기능을 제시함으로서 더 효율적인 대출 관리를 유도할 수 있음

# 04 고객 군집 분석

사용 데이터 SOM 군집 특징 분석 서비스 메시지 제안

- 클러스터링 활용 방안

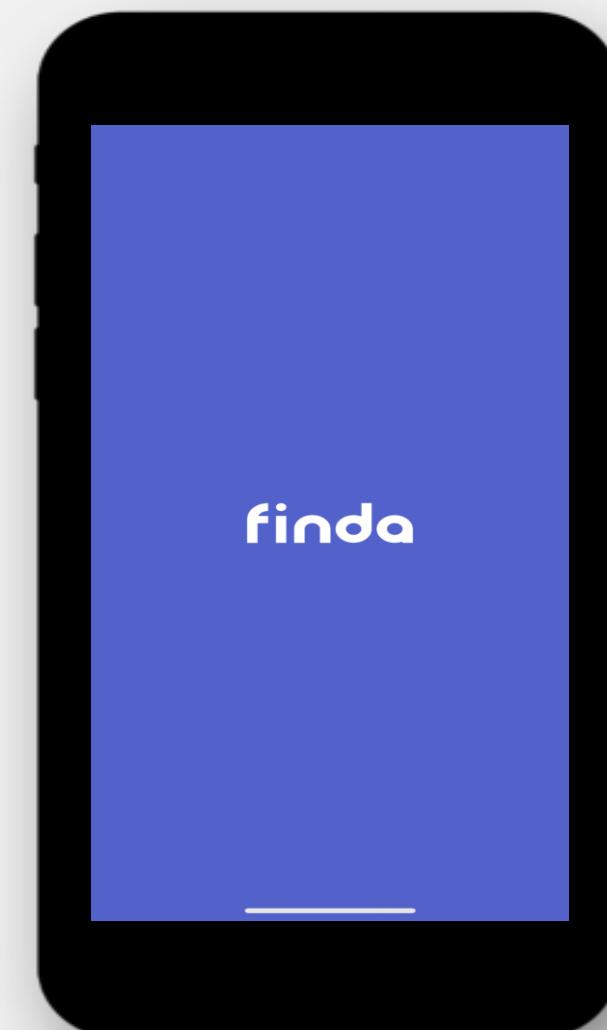
## 1 주기적인 업데이트

신규고객, 새로운 기능을 고려하기 위해 분기별로 재군집화

## 2 맞춤 메시지 업데이트

재군집화 이후 군집 특징의 변화, 새로운 기능의 영향력을 반영하여  
맞춤 메시지 업데이트 필요

*Update*



# 05 의의 및 한계

# 05 의의 및 한계

## 의의

- ✓ 데이터 특성을 고려한 변수(Feature)선택과 양상블로 예측력을 높인 분류기로 고객의 대출 상품 신청여부를 예측,  
보다 효율적인 고객관리를 할 수 있다
- ✓ SOM 군집분석의 해석을 통해 고객 행동을 분석 및 행동 별 맞춤 메시지 제안을 통해  
앱 사용성 향상 도모, 고객유입, 고객관리를 효과적으로 할 수 있다

## 한계

- ✓ 컴퓨팅 성능의 한계로 BART를 양상블하지 못하고 부록으로 기재
- ✓ 보다 좋은 컴퓨팅 성능과, 더 많은 데이터수로 더 좋은 모델을 만들지 못한 아쉬움이 있다

06 부록

# 06 추가 모델 제안 : BART

**BART** *Bayesian Additive Regression Trees*

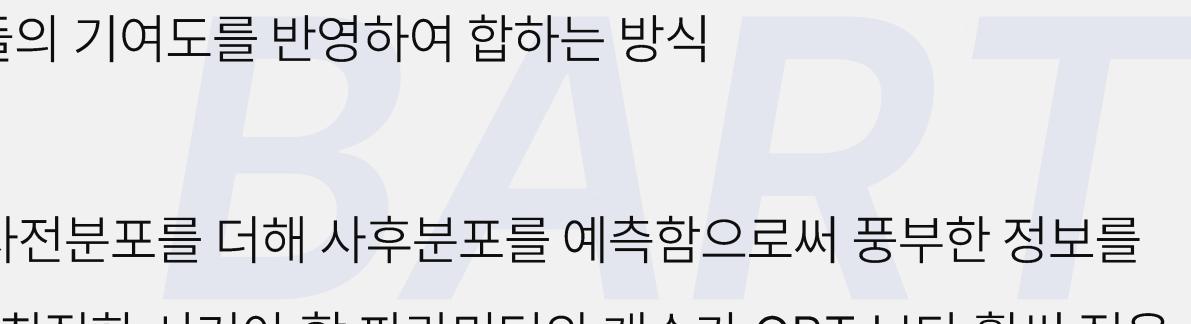
사전분포를 이용하여 여러 트리 모델들을 만든 후 트리의 합으로 근사하는 모델

## ✓ 모델의 특성

- 단순히 추정치들에 대해 averaging 을 하는 Random Forest 와 달리 'weak-learner' 들의 기여도를 반영하여 합하는 방식
- 주어진 데이터에 사전분포를 더해 사후분포를 예측함으로써 풍부한 정보를 활용할 수 있으며, 최적화 시켜야 할 파라미터의 개수가 GBT 보다 훨씬 적음

## ✓ 모델링 방법

예측 결과가 사전분포의 하이퍼파라미터에 robust 하므로 이를 디폴트 값으로 고정



3-fold cv 를 통해  
최적의 트리 개수, 체인 개수 선정

# 06 추가 모델 제안 : BART

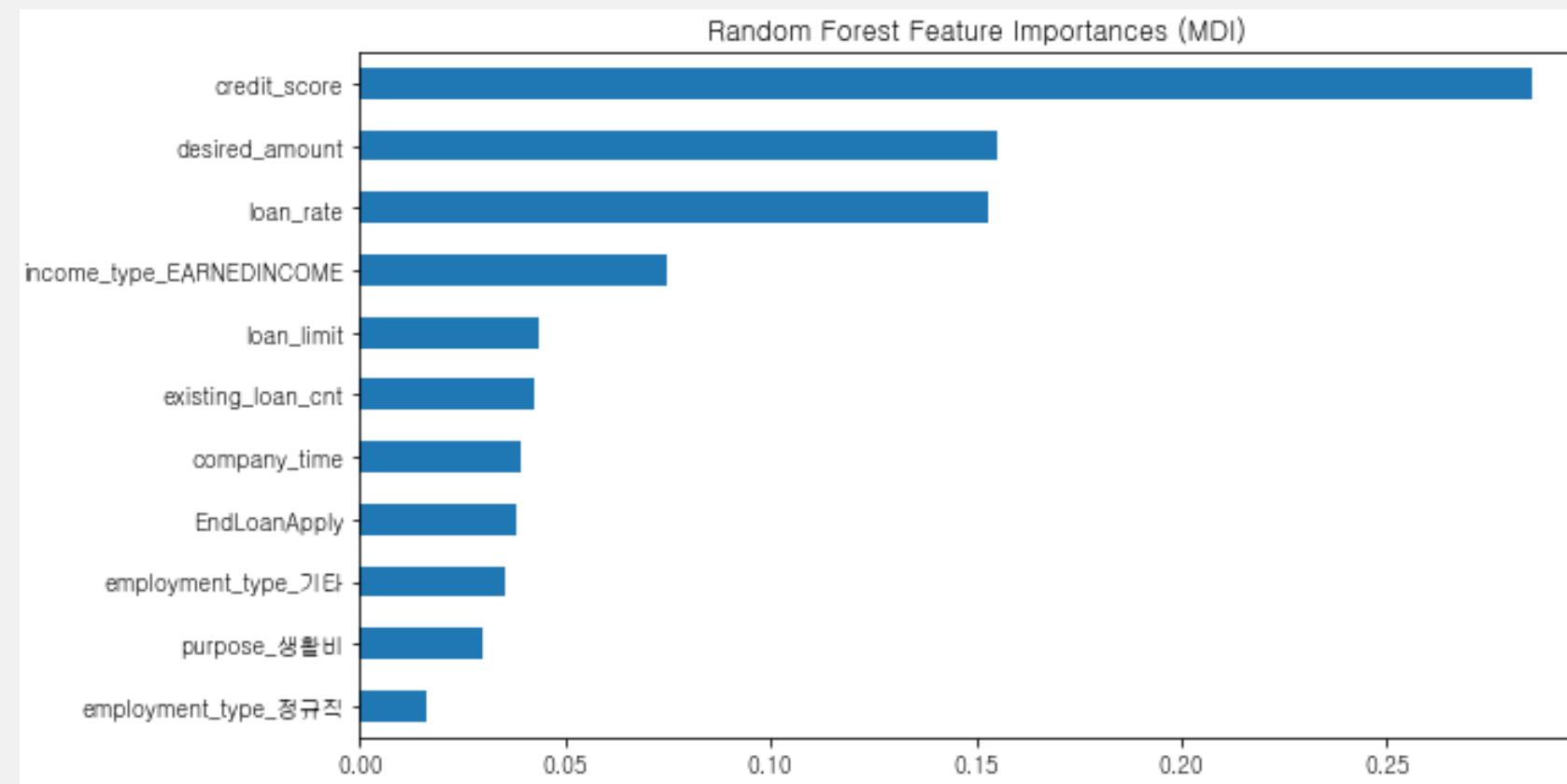
## 하이퍼 파라미터 설정

- **BART**

Hyperparameter	Meaning	Value
sigma_a	사전분포 $\sigma$ 의 shape 파라미터	0.001
sigma_b	사전분포 $\sigma$ 의 scale 파라미터	0.001
alpha	트리 구조에 관한 사전 모수	0.95
beta	트리 구조에 관한 사전 모수	2
n_trees	사용할 트리의 개수	100
n_burn	수렴에 이르기까지 기록하지 않고 돌릴 샘플의 수	8
thin	저장할 샘플의 비율로, 메모리를 세이브하기 위해 작게 설정	0.02

# 06 부록

## Random forest에서의 Variable importance



시각화를 위해 상위 11개 변수만 표시하였음.

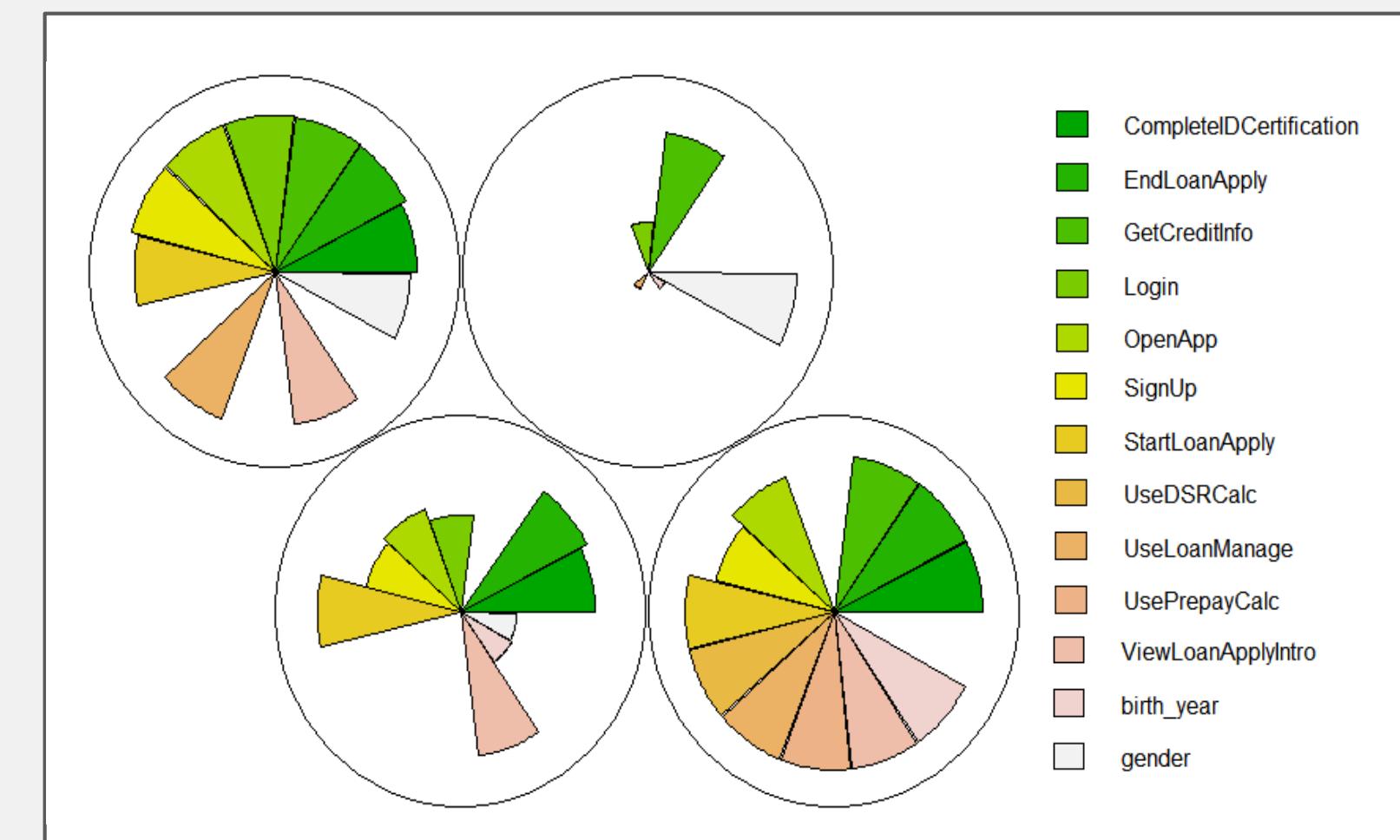
## MDI

각 변수가 split될 때 impurity 감소분의 평균으로,  
값이 클수록 중요도가 높음

- 신용 점수, 희망대출금액이 user spec에서 중요한 변수
- 이자율, 대출한도가 loan result에서 중요한 변수

# 06 부록

## SOM



- Output map grid 개수 -> parsimony 고려
- 전체 자료를 200번 학습시킴, 가장 가까운 unit의 평균 거리 감소

- Codebook : 각 입력 벡터가 각 grid에 얼마나 weight를 가지고 있는지, 학습의 대상이 됨
- 높은 weight를 가질 수록 각 grid에 연관있는 feature

# 06 부록

- 참고문헌

- [1] <https://projecteuclid.org/journals/annals-of-applied-statistics/volume-4/issue-1/BART-Bayesian-additive-regression-trees/10.1214/09-AOAS285.full>
- [2] <https://towardsdatascience.com/bayesian-additive-regression-trees-paper-summary-9da19708fa71>
- [3] <https://github.com/JakeColtman/bartpy>
- [4] <https://ratsgo.github.io/machine%20learning/2017/05/01/SOM>
- [5] <https://specialscene.tistory.com/87>
- [6] <https://hwi-doc.tistory.com/entry/스태킹Stacking-완벽-정리>
- [7] <https://loklee9.tistory.com/212>
- [8] [\[ML\] XGBoost 개념 이해 \(tistory.com\)](#)
- [9] [XGBoost Documentation — xgboost 1.6.2 documentation](#)

# THANK YOU!

