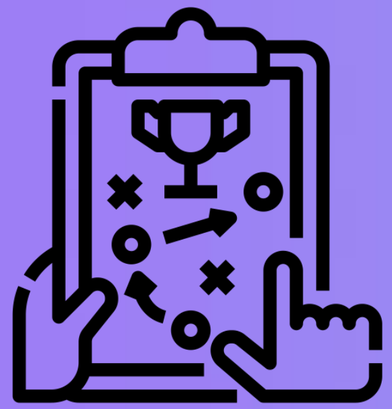
빅콘테스트 데이터 분석분야 퓨처스리그

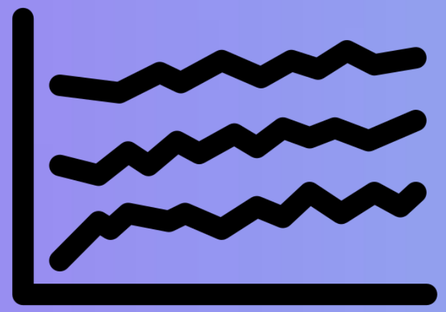
앱 사용성 데이터를 통한 대출신청 예측 분석
부제 뭘로할까요

팀_중고신입

고려대학교 고건호
고려대학교 정해원
중앙대학교 김진재
중앙대학교 김효진
중앙대학교 유나현



문제정의 및 분석배경



데이터 전처리&EDA

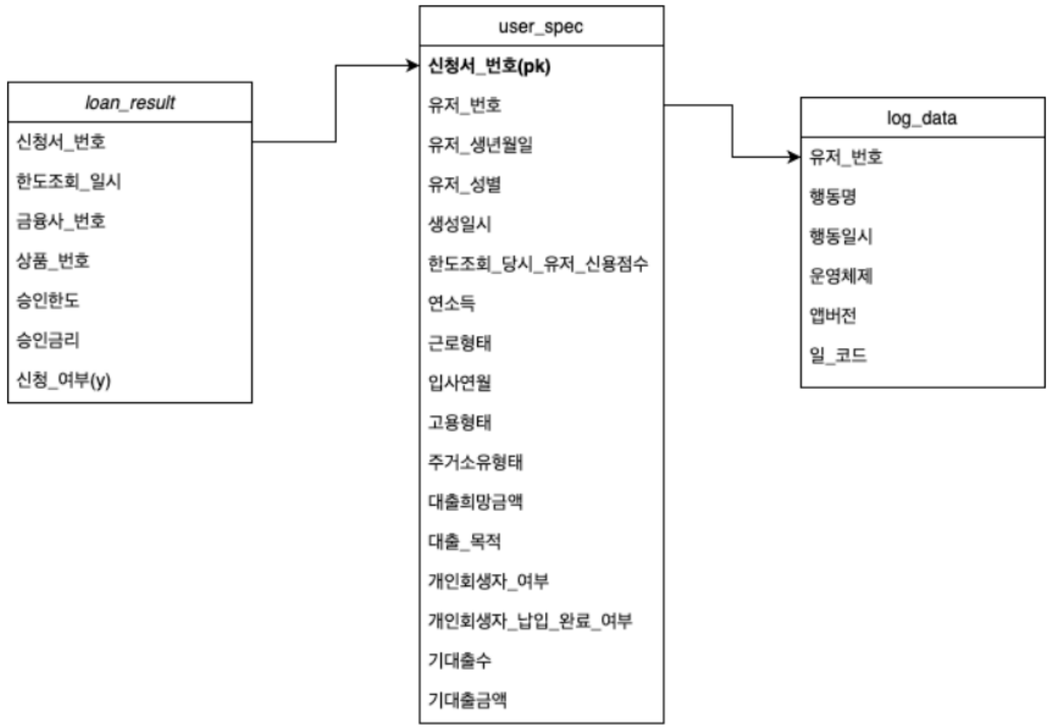
모델링

군집분석&서비스제안

문제 정의 및 분석배경

01 문제 정의 및 분석 배경

문제 : 가명화된 데이터를 기반으로
대출 상품별 고객의 대출신청 여부 예측



- 핀다의 고객 3~6월까지 대출 데이터
  - loan_result.csv (대출상품결과테이블)
  - user_spec.csv (유저스펙테이블)
  - log_data.csv (유저로그데이터)
- 6월 고객 대출 승인여부를 예측
- 예측모델을 활용하여 EDA 수행
- 고객의 특성 분석결과 도출

01 문제 정의 및 분석 배경

핀다는 고객 정보와 자체 신용평가 모델을 이용하여
가능한 대출 상품을 추출하고, 대출 가능여부를 결정할 것이다.

따라서, 대출 상품 결정 요인과 신용평가 결정 요인을 찾아
모델을 훈련시켜 예측 성능 평가 지표인 F1 score를 높인다

제시된 정보만으로 두 요인을 설명하기에 부족하다고 판단,
컬럼들에 대한 적절한 처리와 새로운 특성을 추가하였다

전처리&EDA

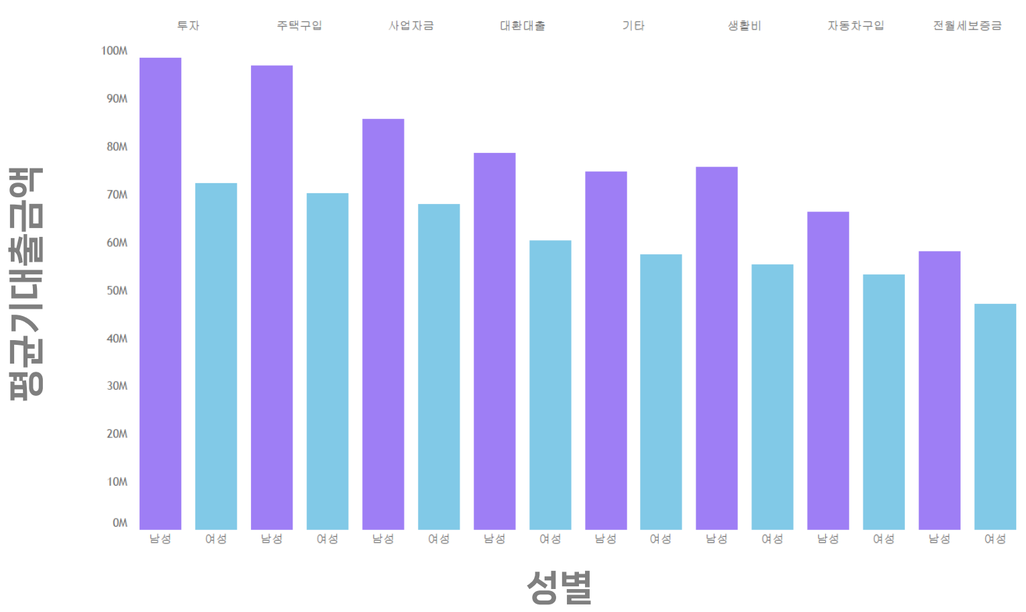
시각화 결과

결측치 처리

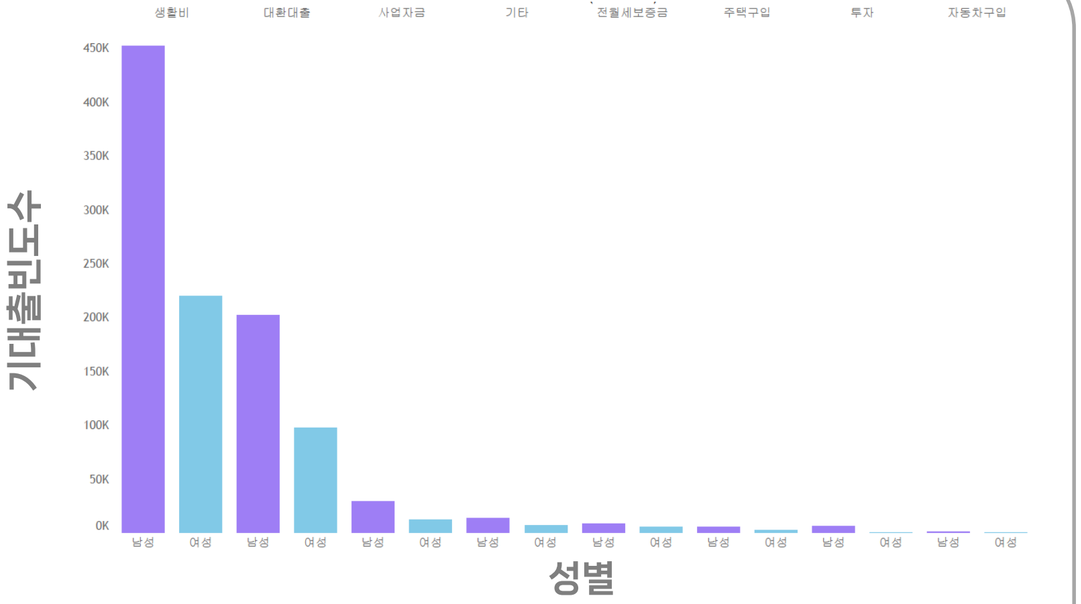
Feature Engineering

02 전처리 & EDA

Tableau를 이용한 시각화

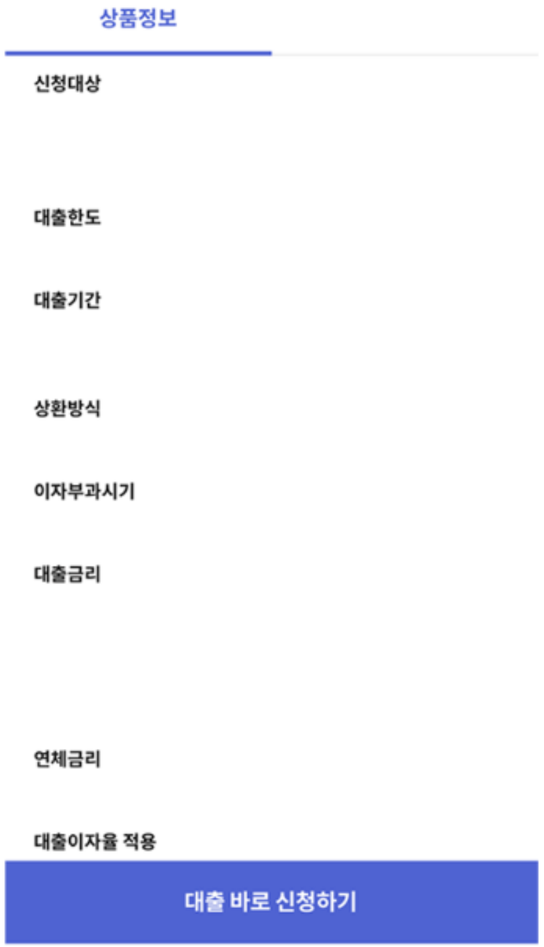


대출 목적과 성별로 평균 기대출 금액을 시각화 한 결과 모든 항목에서 남성의 **평균 기대출 금액이 높았고, 대출 목적으로는 투자, 주택구입, 자업자금 순으로 높은 것을 확인할 수 있었다.**



대출 목적과 성별로 기대출 금액 빈도수로 시각화 한 결과 **생활비를 목적으로 대출하는 사용자가 과반수인 것을 확인할 수 있었다.**

02 전처리 & EDA

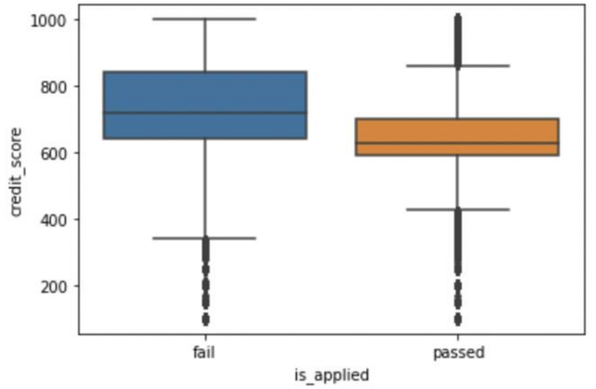


대출 상품 결정 요인을 결정

제공된 항목	새로운 항목
Income_type Employment_type Houseown_type Purpose	Rehabilitation

핀다 어플을 이용하여 대출 상품을 조회,
대출 상품마다 조건이 기제되어 있음

신용평가 모델 요인을 결정



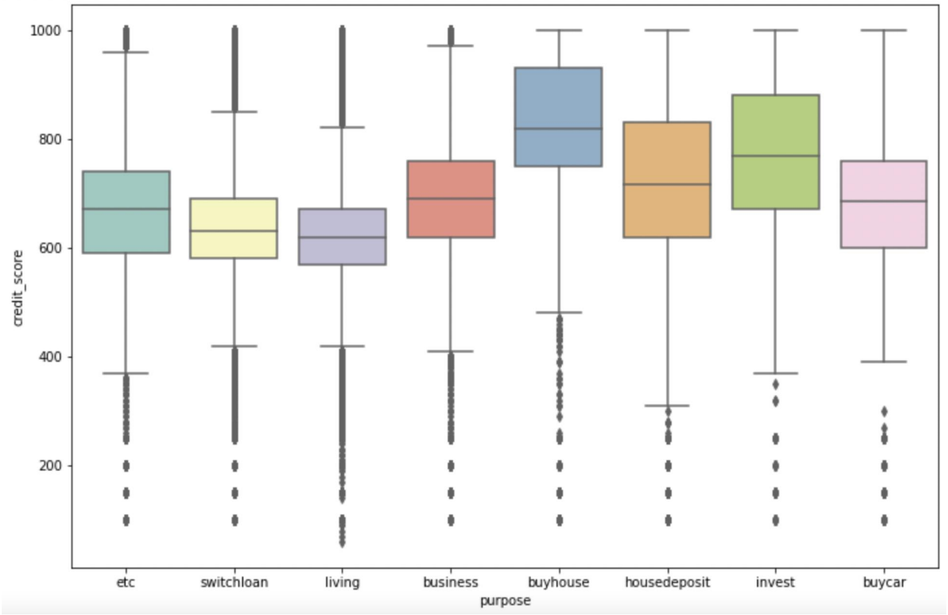
신용점수와 승인비율은 관계가 없음

제공된 항목	새로운 항목
Credit_score	Existing_loan (기대출여부) Over_loan (기대출과다여부)

02 전처리 & EDA

결측치 처리, Feature Engineering

<신용점수>



대출 목적별 신용점수를 확인한 결과,
유의미한 차이가 있었음

목적별 신용점수 통계량으로 대체
항목별 분포가 다르기 때문에 중앙값 사용

개인회생여부와 변제금 납부 여부를 조합하여
범주형 컬럼 rehabilitation을 생성

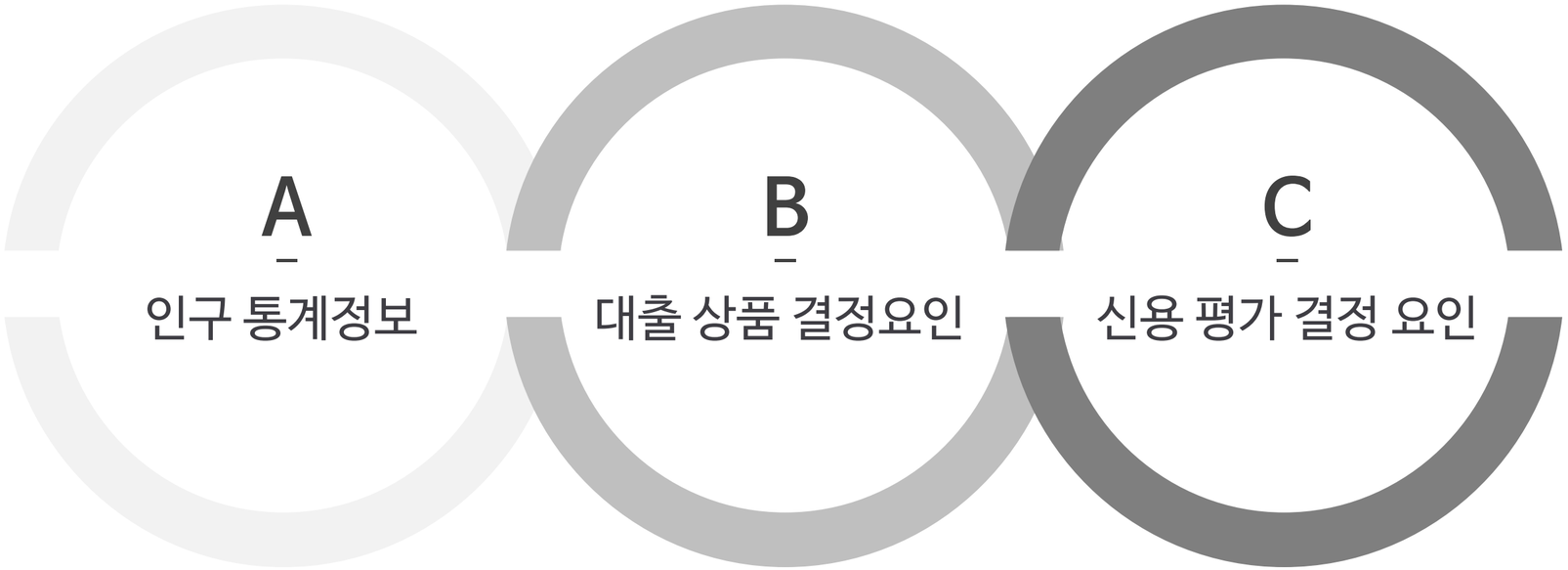
기대출 개수와 기대출 잔액을 조합하여
범주형 컬럼 Existing_loan을 생성

연소득 대비 기대출 금액이 100% 이상여부로
Over_loan을 생성

대출상품결정요인	신용평가결정요인	
Rehabilitation	Existing_loan	Over_loan
Re_com	Ex_large	Non_over
	Ex_non_info	
Re_ing	Ex_small	Over
Re_not	Ex_com	

02 전처리 & EDA

컬럼 설명



Age

Income_type

Employment_type

Houseown_type

Purpose

Rehabilitation

Credit score

Existing_loan

Over_loan

Loan_limit

Loan_rate

모델링

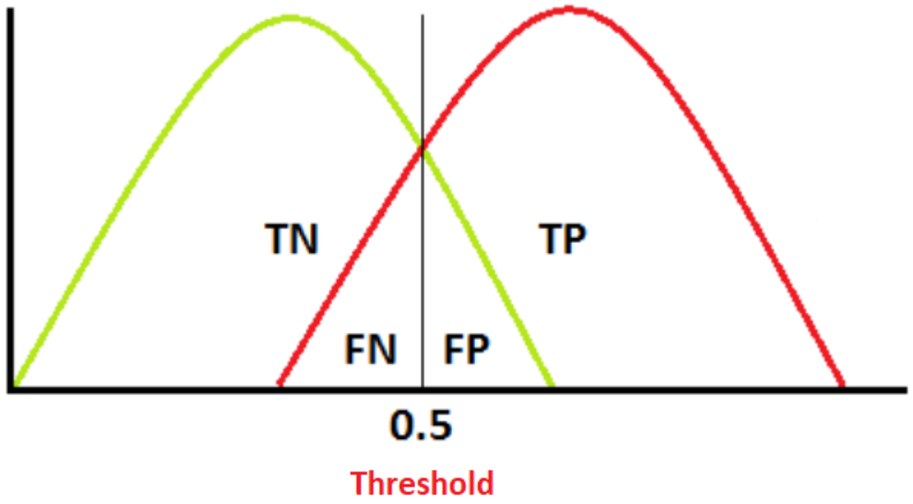
주 모델
RANDOMFOREST

이외 학습 모델
- CATBOOST
- LGBM
- XGBoost

03 모델링

평가 지표) F1 Score

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$



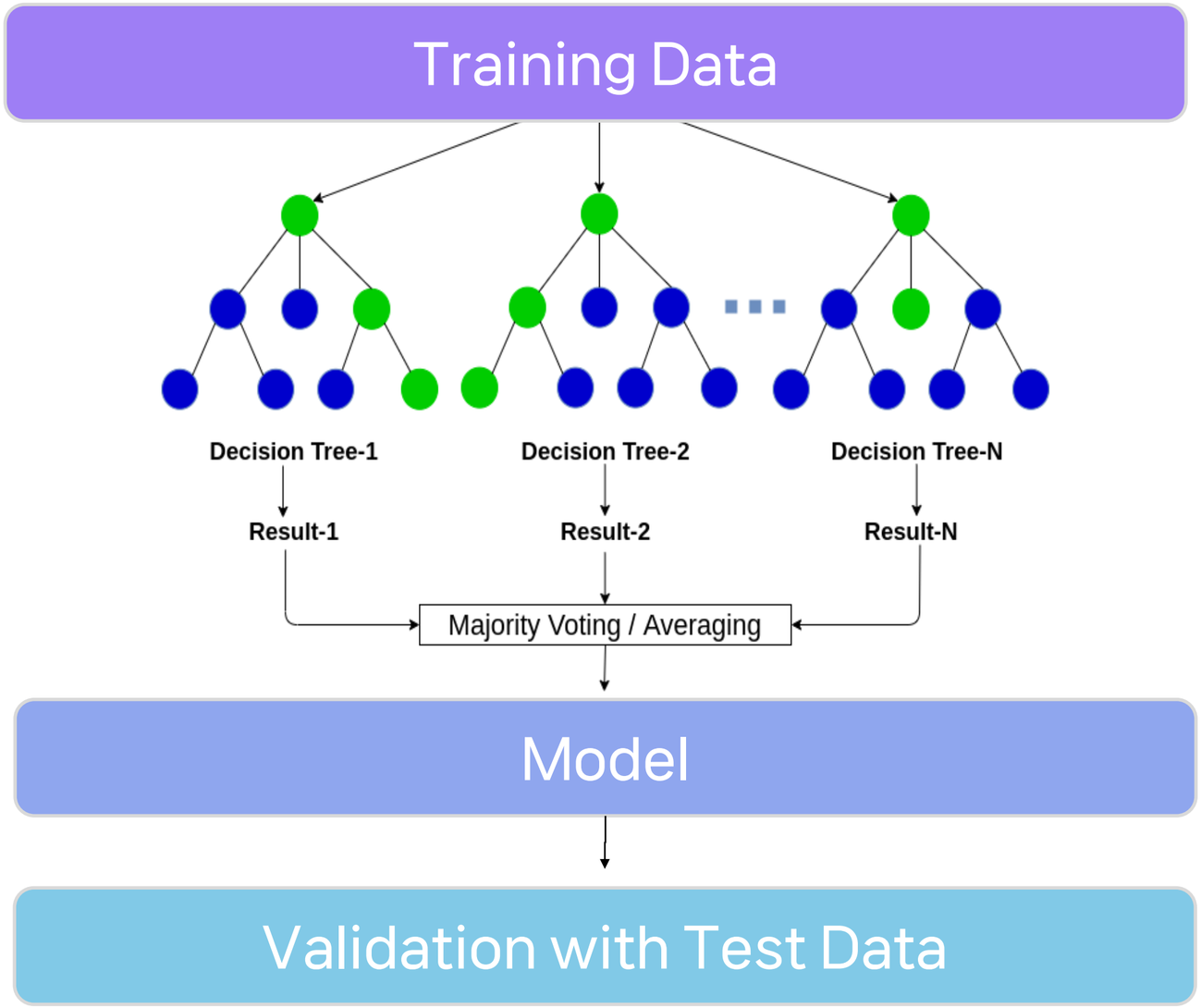
지표 선택 이유

Dataset = 불균형한 데이터

- 전체 데이터에서 약 5.04%만 대출 승인
- 모든 데이터를 0으로 예측하면 정확도가 매우 높아지지만, 의미가 없는 모델
- 모든 데이터를 1으로 판정하면 반대로 Recall이 매우 높아짐
- 서로 trade-off 관계인 Precision과 Recall의 조화 평균인 F1 Score를 이용해 모델의 성능을 평가

03 모델링

1. Random Forest Model



모델 선택 이유

- Classification 에 적합하다
- 과적합이 발생할 확률이 낮다
- 대용량 데이터 처리에 효과적이다
- 다만, 너무 많은 시간이 걸리는 것을 방지하기 위해 결정트리 수(n_estimators)의 수를 적절히 조절해야한다

03 모델링

1. Random Forest Model

Explanatory Variables

Response V.

	credit_score	yearly_income	income_type	employment_type	houseown_type	purpose	age	rehabilitation	existing_loan	over_loan
0	670.00	50000000.0	earnedincome2	etc	monthlyrent	switchloan	44	re_not	ex_large	non_over
1	670.00	50000000.0	earnedincome2	etc	monthlyrent	switchloan	44	re_not	ex_large	non_over
2	730.00	95000000.0	earnedincome	fulltime	monthlyrent	switchloan	44	re_not	ex_large	non_over
3	730.00	95000000.0	earnedincome	fulltime	monthlyrent	switchloan	44	re_not	ex_large	non_over
4	730.00	95000000.0	earnedincome	fulltime	monthlyrent	switchloan	44	re_not	ex_large	non_over
...	...	...	...	...	...	...	...	...	...	...
10264371	625.48	35000000.0	freelancer	etc	own	living	46	re_not	ex_non_info	non_over
10264372	625.48	35000000.0	freelancer	etc	own	living	46	re_not	ex_non_info	non_over
10264373	625.48	35000000.0	freelancer	etc	own	living	46	re_not	ex_non_info	non_over
10264374	625.48	35000000.0	freelancer	etc	own	living	46	re_not	ex_non_info	non_over
10264375	625.48	35000000.0	freelancer	etc	own	living	46	re_not	ex_non_info	non_over

10264376 rows × 13 columns

예측

is_applied
0.0
0.0
0.0
0.0
0.0
...
0.0
0.0
0.0
0.0
1.0

모델의 성능 검증을 위해 기존 Training Data 내에서 학습 및 예측 검정을 시행한다

03 모델링

1. Random Forest Model

범주형 변수들 각각에 대해 Dummy Variable로 변환 후 학습 및 예측에 사용

```
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size=0.3)
```

```
df_dummy1 = pd.get_dummies(df, columns=['income_type','employment_type','h
display(df_dummy1)
```

TRAINING

TEST

	credit_score	yearly_income	age	loan_limit	loan_rate
0	670.00	50000000.0	44	3000000.0	14.5
1	670.00	50000000.0	44	1000000.0	19.9
2	730.00	95000000.0	44	11000000.0	15.1
3	730.00	95000000.0	44	15000000.0	9.9
4	730.00	95000000.0	44	3000000.0	15.9
...	...	...	...	...	...
10264371	625.48	35000000.0	46	8000000.0	20.0
10264372	625.48	35000000.0	46	50000000.0	18.9
10264373	625.48	35000000.0	46	16000000.0	14.2
10264374	625.48	35000000.0	46	4000000.0	12.9
10264375	625.48	35000000.0	46	25000000.0	10.9

10264376 rows × 30 columns

학습

예측

is_applied
0.0
0.0
0.0
0.0
0.0
...
0.0
0.0
0.0
0.0
1.0

03 모델링

1. Random Forest Model

```
full_RF = RandomForestClassifier(n_estimators=100)
full_RF.fit(x1_train, y1_train)

#threshold 조절 후 예측
threshold = 0.3

predicted_proba = full_RF.predict_proba(x1_test)
y1_pred = (predicted_proba [:,1] >= threshold).astype('int')
```

n_estimators = 100으로 설정 후
Random Forest Model 학습

Threshold = 0.3
Test Data를 이용해 대출 승인 여부 예측

Result

	Predicted 1	Predicted 0
Real 1	53193	113274
Real 0	95642	2817204

	Score
Accuracy	0.932
Precision	0.357
Recall	0.32
F1 Score	0.337

03 모델링

1. Random Forest Model

결과로 나온 f1 score를 기반한 모델을 이용하여
3,4,5월 데이터 전부를 Training Data,
6월 데이터 전부를 TEST로 사용

TEST

```
new_Y = pd.merge(trainY, testY2, how='inner', left_on=['application_id', 'product_id'], right_on=['application_id', 'product_id'])
display(new_Y)
```

	Unnamed: 0	application_id	credit_score	yearly_income	income_type	employment_type	houseown_type	purpose
0	0	954900	870.0	30000000.0	privatebusiness	fulltime	otherfamily	switchloan
1	1	954900	870.0	30000000.0	privatebusiness	fulltime	otherfamily	switchloan
2	2	954900	870.0	30000000.0	privatebusiness	fulltime	otherfamily	switchloan
3	3	954900	870.0	30000000.0	privatebusiness	fulltime	otherfamily	switchloan
4	4	954900	870.0	30000000.0	privatebusiness	fulltime	otherfamily	switchloan
...	...	...	...	...	...	...	...	...
3255189	13518243	242374	660.0	78000000.0	earnedincome	fulltime	own	living
3255190	13518244	242374	660.0	78000000.0	earnedincome	fulltime	own	living
3255191	13518245	242374	660.0	78000000.0	earnedincome	fulltime	own	living
3255192	13518246	242374	660.0	78000000.0	earnedincome	fulltime	own	living
3255193	13518247	242374	660.0	78000000.0	earnedincome	fulltime	own	living

3255194 rows × 16 columns

예측

is_applied
NaN
NaN
NaN
NaN
NaN
...
NaN
NaN
NaN
NaN
NaN

HYPERPARAMETER : n_estimator=100 기준으로 이상으로 가면
시간이 너무 오래걸리고 F1 상승이 미약했다. 그보다 작으면 Underfitting되었다.

03 모델링

1. Random Forest Model

Model Fitting with Training Data

```
full_RF = RandomForestClassifier(n_estimators=100)
full_RF.fit(x1, y1)
```

```
RandomForestClassifier()
```

Predicting the Test Data

```
threshold = 0.3

predicted_proba = full_RF.predict_proba(x1_test)
y1_pred = (predicted_proba [:,1] >= threshold).astype('int')
print(y1_pred)
```

F1 Score 이용하여 모델 평가

6월 대출 승인여부를 예측

```
testY2['is_applied'] = y1_pred
display(testY2)
```

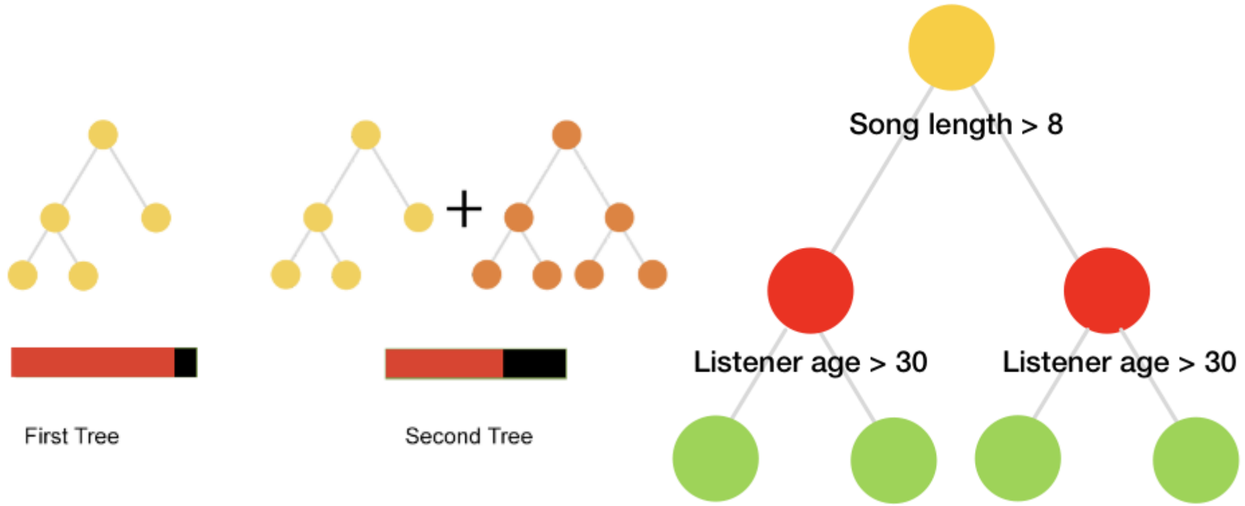
	application_id	product_id	Predicted Y values
0	4	220	0
1	4	191	0
2	8	29	0
3	8	159	0
4	8	85	0
...	...	...	...
3255189	2167778	258	0
3255190	2167791	29	0
3255191	2167822	149	0
3255192	2167822	157	0
3255193	2167822	65	0

3255194 rows × 3 columns

03 모델링

2. CATBoost Model

Training Data



Model

Validation with Test Data

모델 선택 이유

- 범주형 변수가 많은 데이터셋에서 성능이 우수하다
- 과적합이 발생할 확률이 낮다
- 학습 속도가 빠르다
- 다만 수치형 변수에서는 학습속도가 느리다

03 모델링

2. CATBoost Model

그룹화 모델

```
X_train2 = X_train.groupby('application_id').max().reset_index('inplace' == True, 'drop' == True)
X_test2 = X_test.groupby('application_id').max().reset_index('inplace' == True, 'drop' == True)
y_test2 = y_test.groupby('application_id').max().reset_index('inplace' == True, 'drop' == True)
y_train2 = y_train.groupby('application_id').max().reset_index('inplace' == True, 'drop' == True)
```

TRAINING

	0	application_id	credit_score	yearly_income	income_type	employment_type	houseown_type	purpose
0	0	1945260	-0.559220	17.727534	1	1	1	1
1	1	1945260	-0.559220	17.727534	1	1	1	1
2	2	1019382	-0.077456	18.369387	2	2	1	1
3	3	1019382	-0.077456	18.369387	2	2	1	1
4	4	1019382	-0.077456	18.369387	2	2	1	1
...	...	...	...	...	...	...	...	...
9999995	9999995	559973	2.090478	17.875954	2	2	1	3
9999996	9999996	559973	2.090478	17.875954	2	2	1	3
9999997	9999997	559973	2.090478	17.875954	2	2	1	3
9999998	9999998	559973	2.090478	17.875954	2	2	1	3
9999999	9999999	559973	2.090478	17.875954	2	2	1	3

10000000 rows × 15 columns

학습

is_applied
0.0
0.0
0.0
0.0
0.0
...
0.0
0.0
0.0
0.0
0.0

TEST

	0	application_id	credit_score	yearly_income	income_type	employment_type	houseown_type	purpose
10000000	10000000	559973	2.090478	17.875954	2	2	1	3
10000001	10000001	559973	2.090478	17.875954	2	2	1	3
10000002	10000002	559973	2.090478	17.875954	2	2	1	3
10000003	10000003	559973	2.090478	17.875954	2	2	1	3
10000004	10000004	559973	2.090478	17.875954	2	2	1	3

예측

Application_id별로 그룹화시켜 예측을 먼저 진행하였다.

03 모델링

2. CATBoost Model

그룹화 모델

---

iterations= 20으로 설정 후
CatBoost 학습

Learning_rate = 0.1
Test Data를 이용해 대출 승인 여부 예측

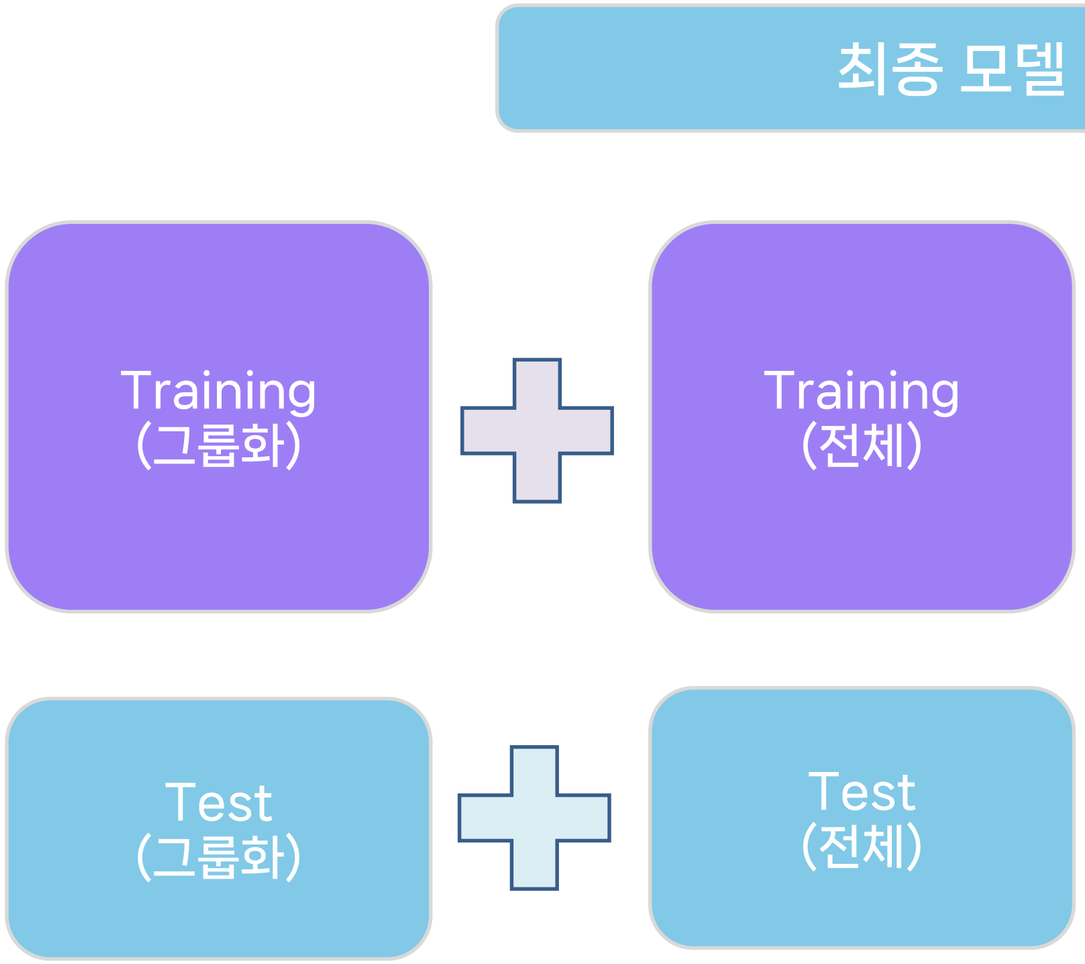
Result

	Predicted 1	Predicted 0
Real 1	9348	397
Real 0	7998	1029

	Score
Accuracy	0.553
F1 Score	0.690

03 모델링

2. CATBoost Model



is_applied
0.0
0.0
0.0
0.0
0.0
...
0.0
0.0
0.0
0.0
0.0

그룹화 모델의 결과와 전체 모델의 결과를 합하여 최종 예측을 진행하였다.

03 모델링

2. CATBoost Model

최종 모델

---

iterations= 20으로 설정 후
CatBoost 학습

Learning_rate = 0.1
Test Data를 이용해 대출 승인 여부 예측

Result

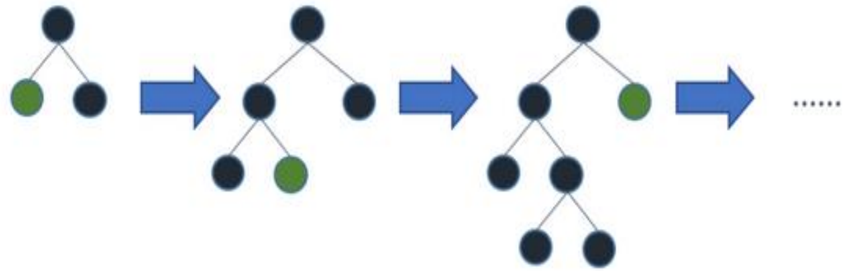
	Predicted 1	Predicted 0
Real 1	7981	6523
Real 0	41266	208606

	Score
Accuracy	0.819
F1 Score	0.250

03 모델링

3. LGBM & XGBoost

LGBM

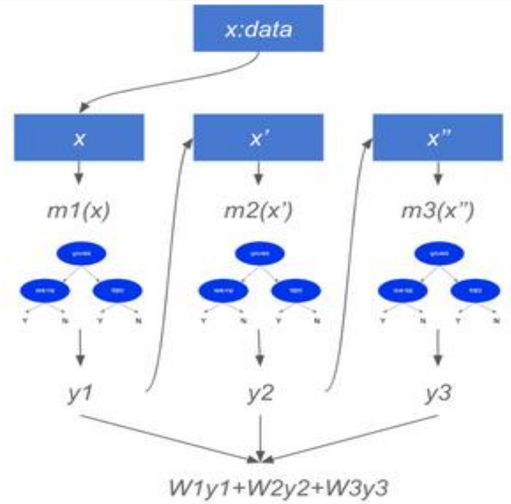


Leaf-wise tree growth

모델 선택 이유

- 빠른 속도
- 적은 메모리 사용
- 다만 과적합 가능성이 존재

XGBoost



모델 선택 이유

- 분류와 회귀에 있어 뛰어난 성능
- 과적합 규제 기능
- 학습, 분류속도가 높음
- 조기종료 기능

03 모델링

3. LGBM & XGBoost

LGBM 모델

n_estimators = 30으로 설정

Objective = binary

	Predicted 1	Predicted 0
Real 1	79	110676
Real 0	45	1942076

	Score
Accuracy	0.946
F1 Score	0.001

XGBoost 모델

n_estimators = 30으로 설정

Learning rate = 0.1, alpha = 10
Colsample_bytree = 0.2

	Predicted 1	Predicted 0
Real 1	16285	94322
Real 0	161200	1781069

	Score
Accuracy	0.876
F1 Score	0.113

군집분석

문제 & 분석 대상 정의

SOM

서비스 제안

04 군집 분석 & 서비스 제안

문제정의

핀다 홈 화면 진입 고객을 대상으로
군집분석을 시행하고 서비스 메시지 제안

진입 고객을 가장 일반적인 형태로 정의한 후,
군집 별 서비스 제안 목표를 설정

04 군집 분석 & 서비스 제안

분석 대상 정의

핀다 이용고객 분석

```
living          0.6316
switchloan      0.2568
business        0.0453
etc             0.0220
housedeposit    0.0199
buyhouse        0.0127
invest          0.0091
buycar          0.0025
```

약 86%의 고객이 생활비, 대환대출 목적으로 대출을 신청함.

```
기대출 금액 중앙값: 45000000.0 원
기대출 갯수 중앙값: 4.0 개
```

고객은 평균4개, 4500만원 정도의 기대출을 가지고 있음.

전처리

1. 이상치 제거 및 정규화

연속형 변수 중 1-3분위수의 값만 사용

Minmax scaling 진행

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

col = ['yearly_income', 'desired_amount']
x = user[col].values
x_scaled = scaler.fit_transform(x)
```

일반적인 고객 도출

핀다 화면에 진입하는 고객은 평균적으로

생활비, 대환대출 목적

을 가지며

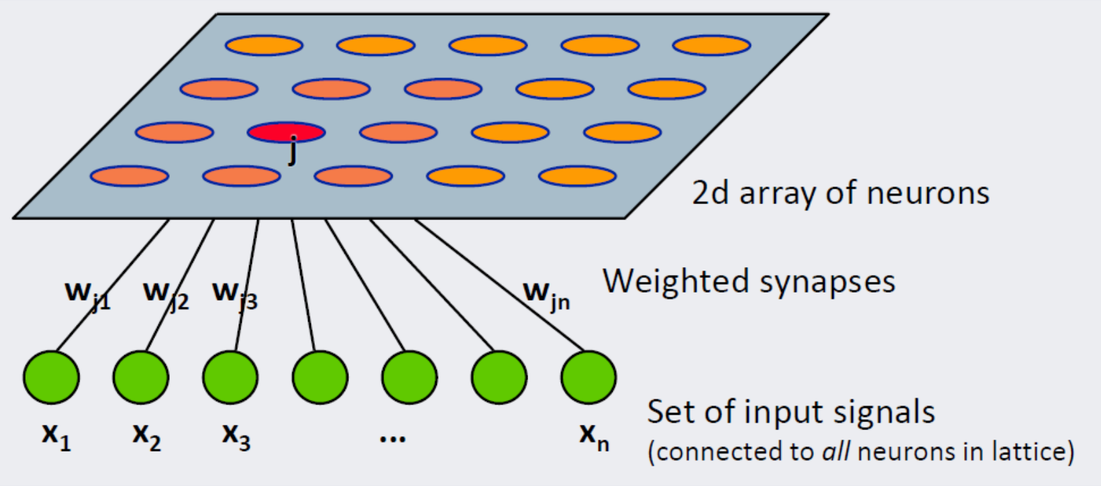
약 4개, 4,500만원의 기대출

을 가지고 있다고 가정한다.

04 군집 분석 & 서비스 제안

모델 선정

SOM(자기조직화 지도)



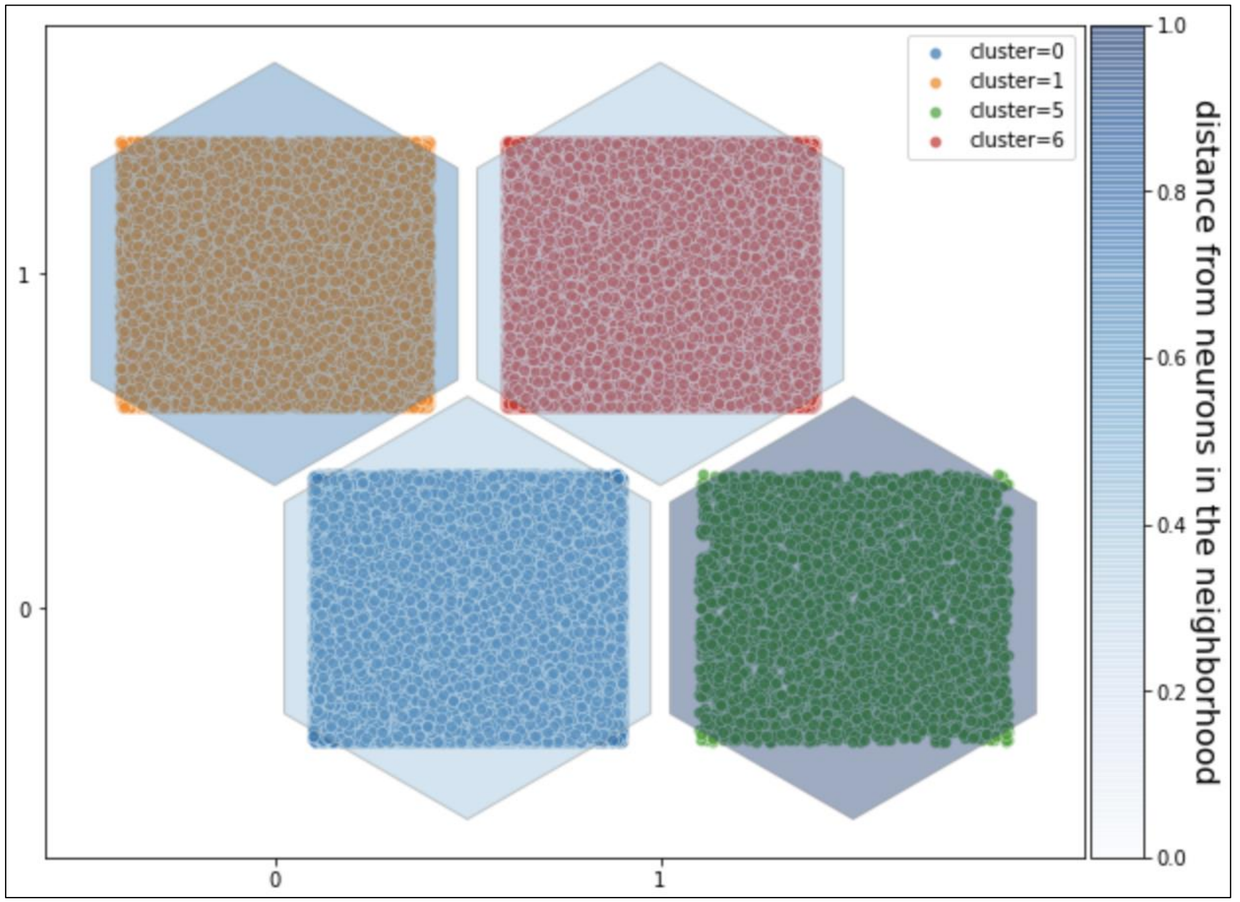
특징

- 인공신경망과 유사한 방식의 학습을 통해 군집을 도출해내는 기법
- 차원축소 시 유사도 보존에 강력함
- 학습 결과를 2차원의 지도로 표현하여 시각화에 유리함

데이터가 많은 고객 군집 분석에 유리함.

04 군집 분석 & 서비스 제안

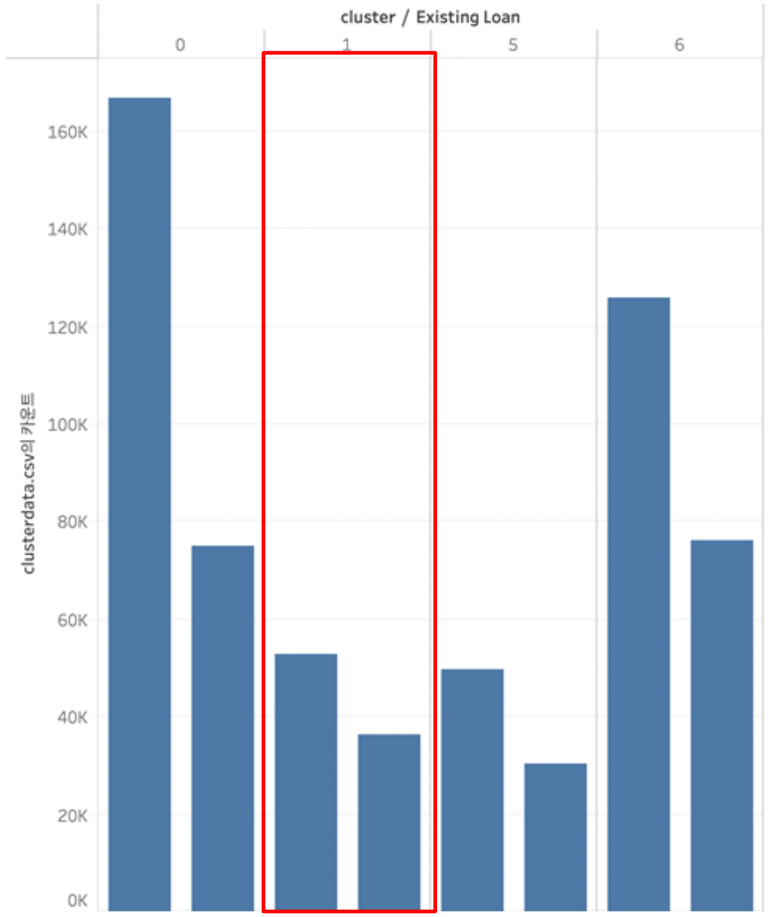
훈련결과

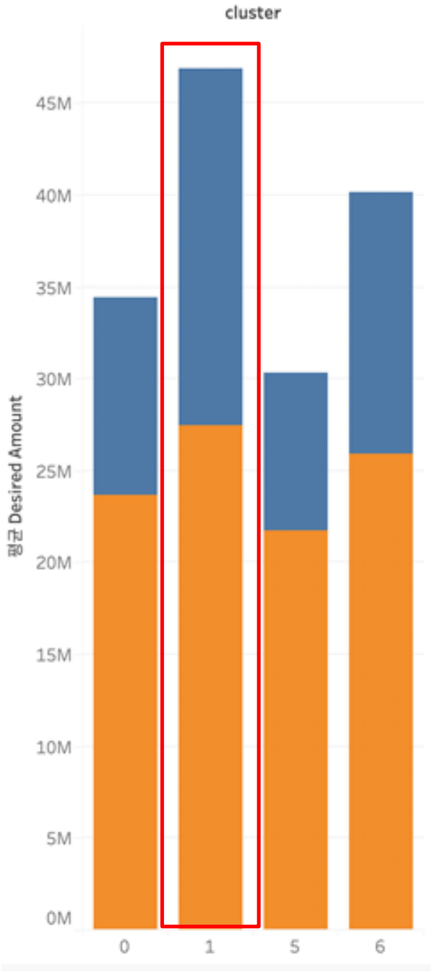


4개의 군집을 선택하여 훈련, 특히 cluster 1,5가 잘 구분된 것을 알 수 있다.

Cluster 5 | Cluster 0, 6 | **Cluster 1**

신용도 낮음 — 신용도 높음





특징
1) 신용도가 높음
2) 기대출 비율이 높지 않음
3) 상대적으로 대환대출보다 생활비 대출이 많음

핀다 **이용률이 적은 고객임을 확인**

[서비스 제안] **서비스 체류 기간 연장**

대출상품 전환을 통한 금리인하 & 다양한 대출상품 조회와 관련한 핀다 어플의 유용성에 대한 메시지 제안

**" 핀다 고객의 몇 %는
대출 금리를 ~만큼 낮췄어요"**

**" 이제는 제 1금융권의 대출상품까지,
한번에 조회해보세요 "**

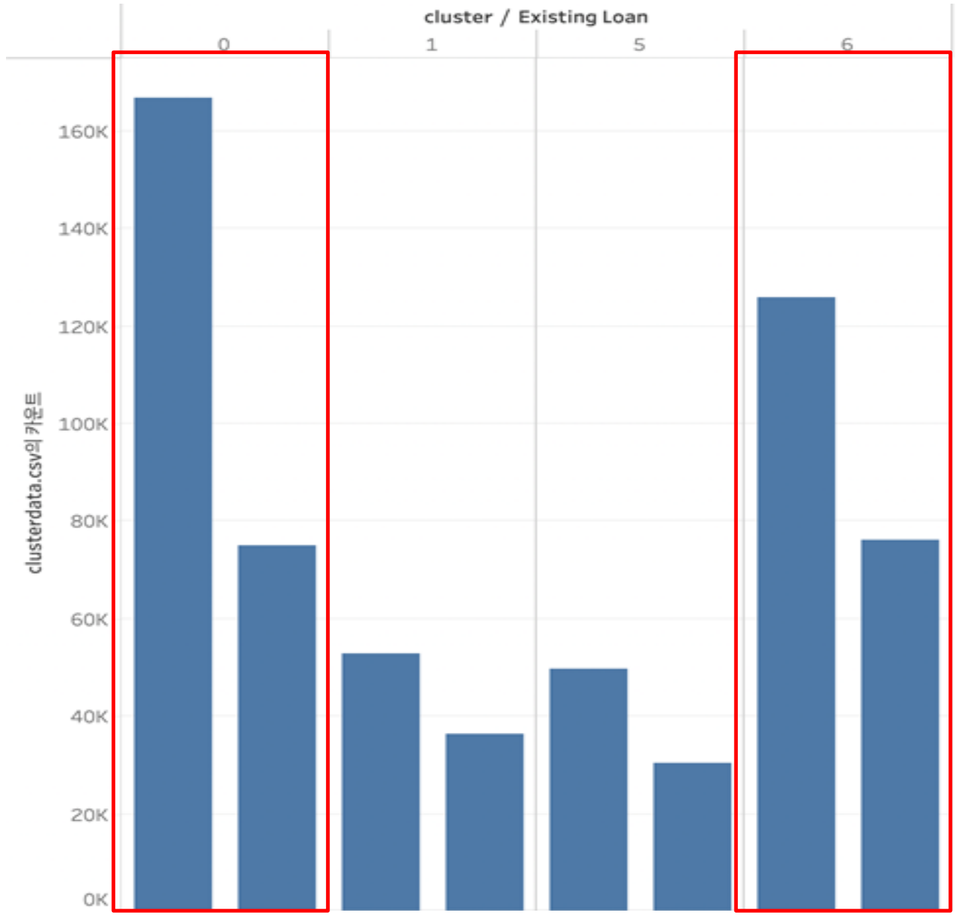
Cluster 5

Cluster 0, 6

Cluster 1

신용도 낮음

신용도 높음



특징
1) 중신용도, 중저금리 이용자
2) 기대출(금액, 회수) 비율이 높음

핀다 이용률이 가장 높은 **충성고객**이라고 판단

[서비스 제안] : 기대출 관리 서비스 이용 확대

대출관리서비스를 강조하여
기존 대출을 통합적으로 관리할 수 있다는 것을 강조
대환대출 서비스를 통한 대출금리 인하를 어필

" 내 대출을 한 데 모아서 확인해요"

"기존 대출의 금리를 낮출 수 있어요 "

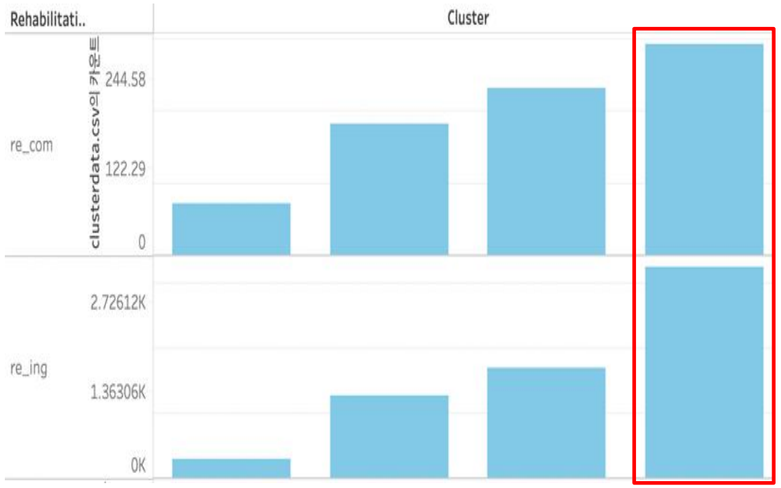
Cluster 5

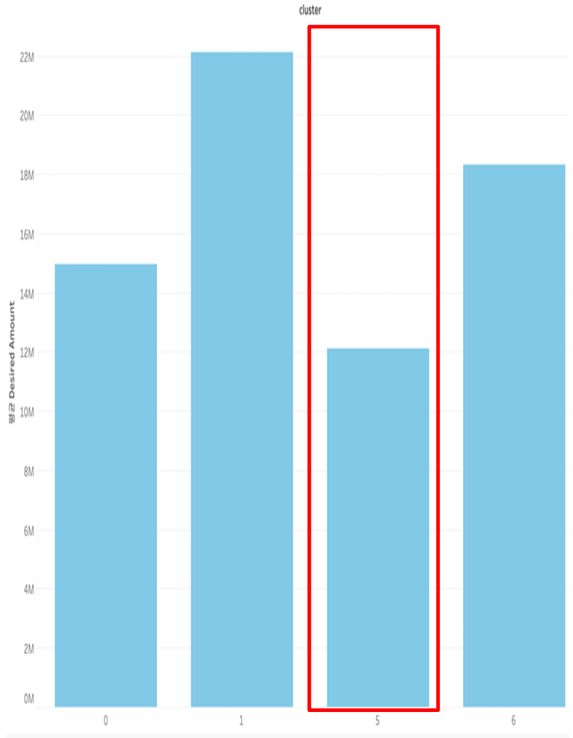
Cluster 0, 6

Cluster 1

신용도 낮음

신용도 높음





특징
1) 클러스터 중 가장 적은 인원
2) 저신용도 고객 비율이 높음
3) 희망대출금액이 상대적으로 적음
4) 개인회생자의 비율이 상대적으로 많음

대출에 대해서 **소극적인 태도**를 가질 확률이 높다고 판단

[서비스 제안] 핀다 이용 신뢰도 제고

신용점수 조회를 한 대상에게, 본인과 비슷한 신용점수대에 있는 사람들의 대출기록(대출횟수, 상품개수)을 보여줌으로써 핀다 이용에 대한 신뢰도를 높이기

"나와 비슷한 신용점수대의 사람들이 이번 달에 ~만큼 대출을 이용했어요"

"나와 비슷한 시용점수대의 사람들은 ~만큼 금리를 줄였어요 "

Cluster 5

finda

나와 비슷한 신용점수대의
사람들은 이번 달에 ~만큼
대출을 이용했어요

Cluster 0, 6

finda

내 대출을
한데 모아서 확인해요

Cluster 1

finda

3분만에, 간편하게
‘나를 위한 ' 대출 상품을
조회해보세요

감사합니다 ☺

팀_중고신입

고려대학교 고건호
고려대학교 정해원
중앙대학교 김진재
중앙대학교 김효진
중앙대학교 유나현