Credit Card
사용자 신용도
분류 모델 해석

T2 _ 박민규 강윤지 정현지 피재희

CONTENTS

Introduction

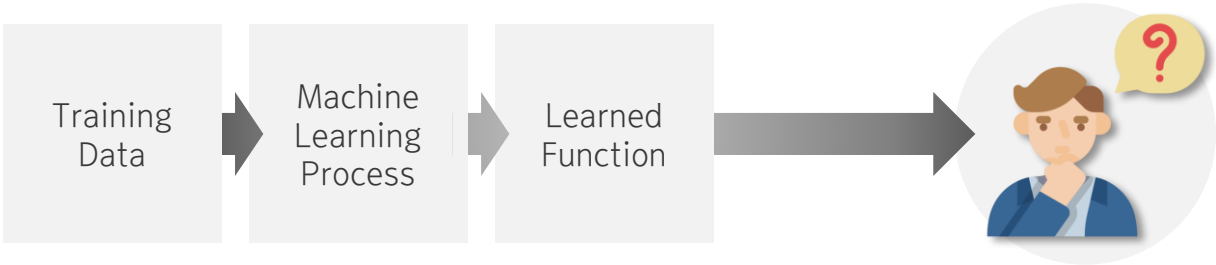
Introduction

Explainable AI

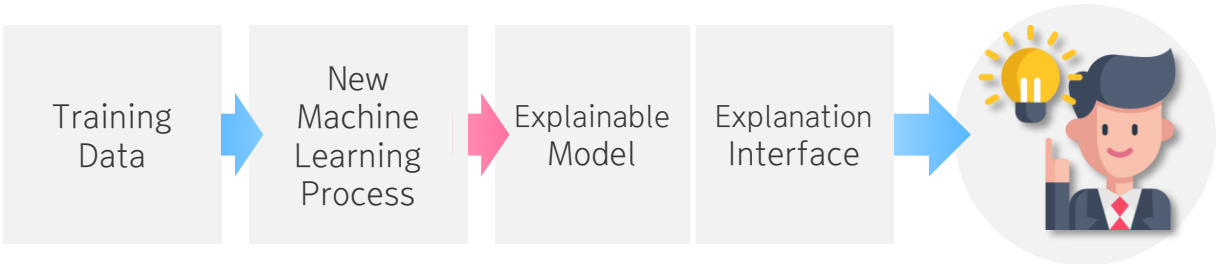
XAI

; Explainable AI, 즉 AI가 도출한 결과와 출력을 인간이 이해하고 신뢰할 수 있도록 해주는 기술

■ AI

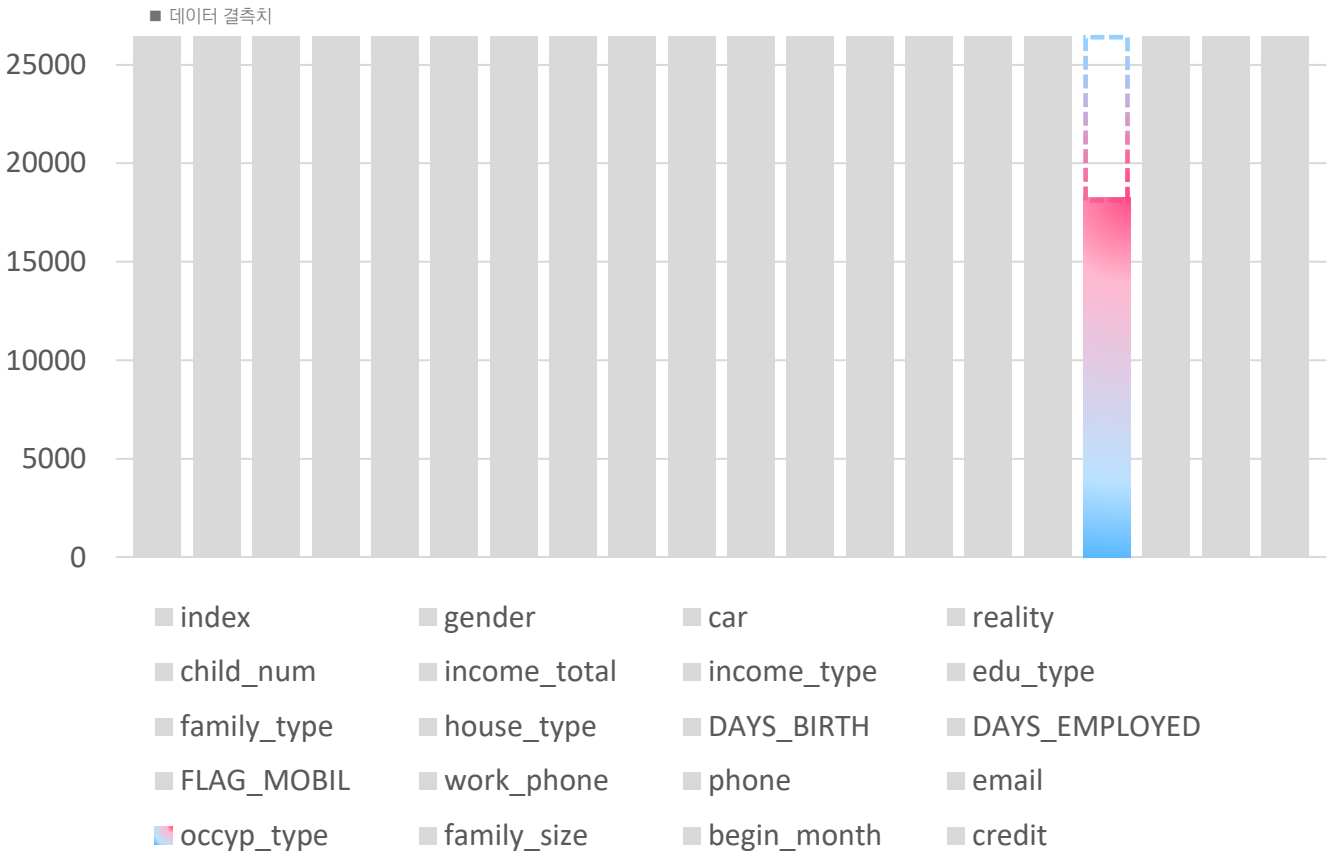


■ XAI



Research

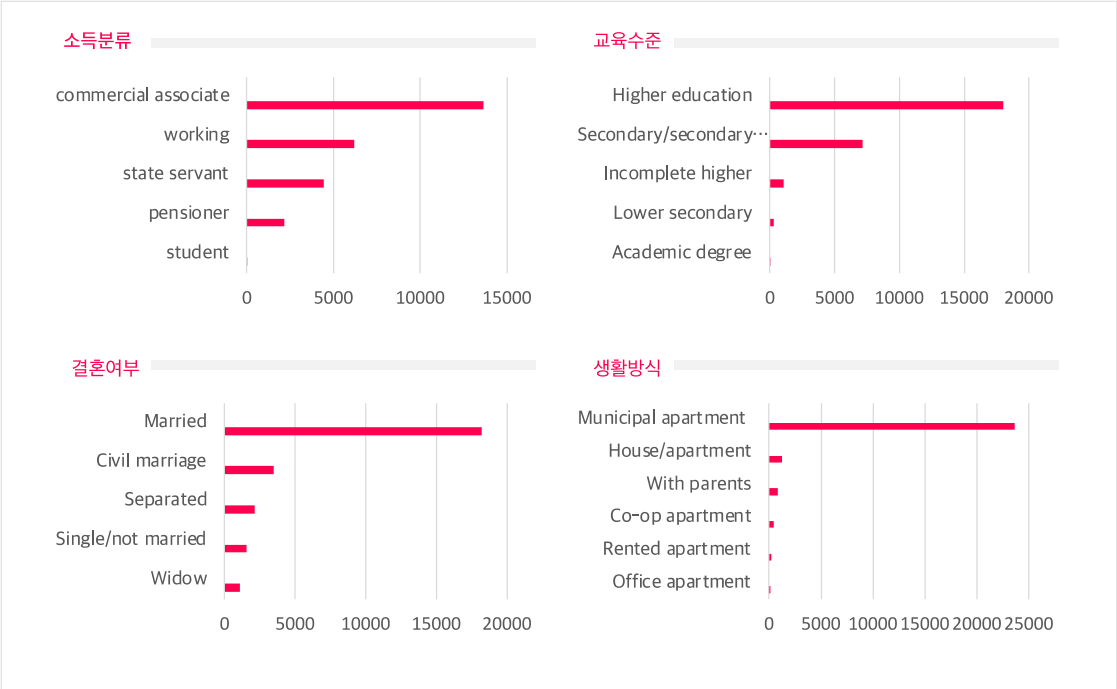
3-1. 데이터수집 및 EDA



Research

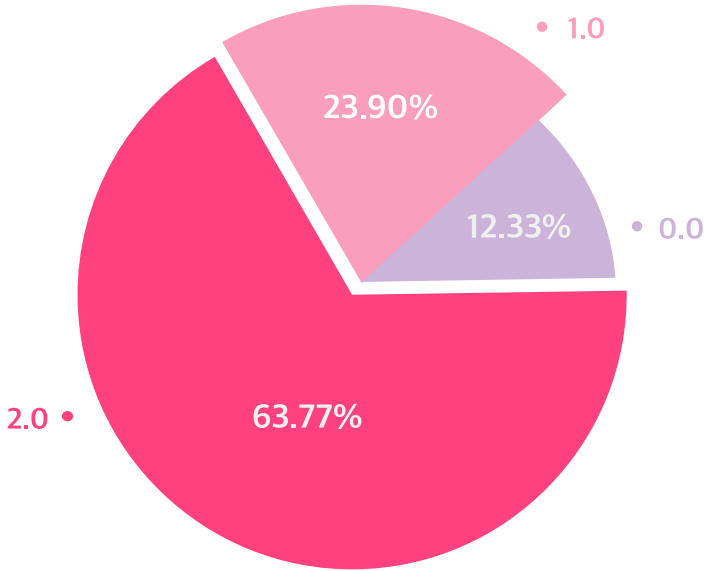
3-1. 데이터수집 및 EDA

■ column 데이터 별 unique 개수



Research

3-1. 데이터수집 및 EDA



※신용등급의 값이 낮을수록 높은 신용의 신용카드 사용자를 의미

► 데이터 불균형 발견

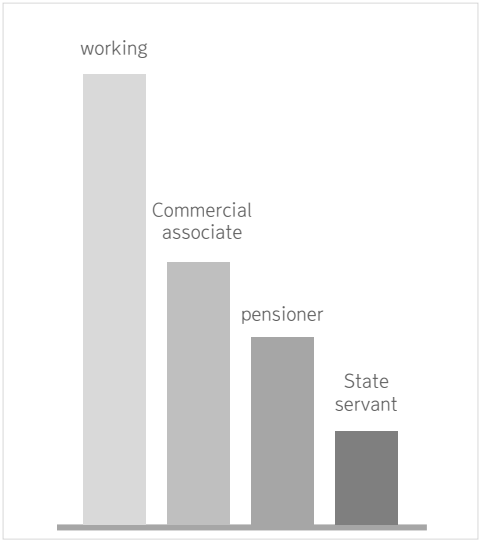
; 특정 값에 대한 데이터가 매우 높은 빈도로 나타나는 것, overfitting 초래

OVERSAMPLING

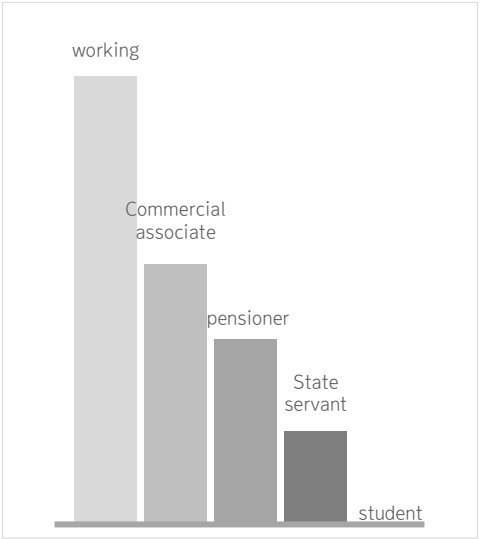
Research

3-1. 데이터수집 및 EDA

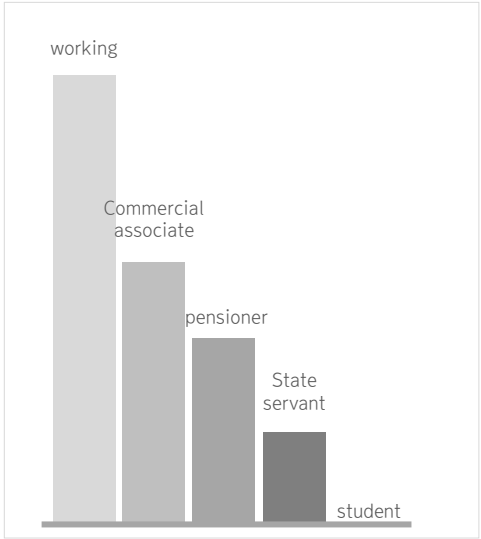
■ 소득분류에 따른 신용 등급 차이



Credit = 0



Credit = 1



Credit = 2

높은 신용 등급(0)에서는 학생이 존재하지 않음

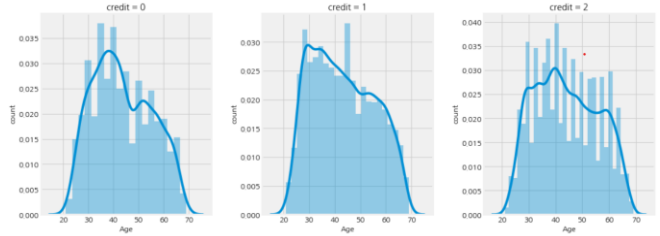
낮은 신용 등급(1,2)에서는 학생이 약간 존재하나 애초에 학생 수는 7명뿐 임

※ 위 그래프는 각 신용등급에 따른 상대적 수치를 나타냄

Research

3-1. 데이터수집 및 EDA

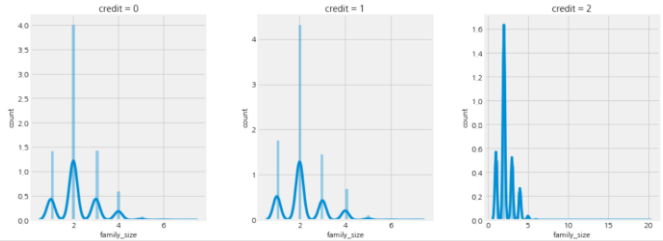
■ 연령대에 따른 신용등급



■ 신용등급에 따른 분포 차이 Class 별로 거의 비슷

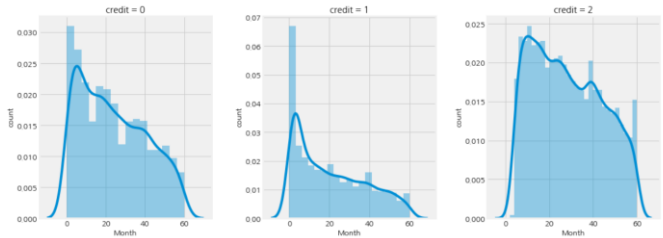
■ 전체적으로 20~30대 연령대 비율 낮음

■ 가족 수에 따른 신용등급



■ class2가 class0과 1에 비해 적은 가족 수 분포 가짐

■ 카드 발급 기간에 따른 신용등급



■ class2에서 카드 발급 기간이 긴 고객이 많음

Research

3-2. 데이터 전처리

- 결측치 -

Occpy_type = NaN

Pensioner	4440
Working	2312
Commercial Associate	1026
State servant	392
Student	1

Name : income_type, dtype : int64

결측치임에도 소득이 0인 경우 없음

Unknown 값도 충분히 의미 있는 데이터라고 판단

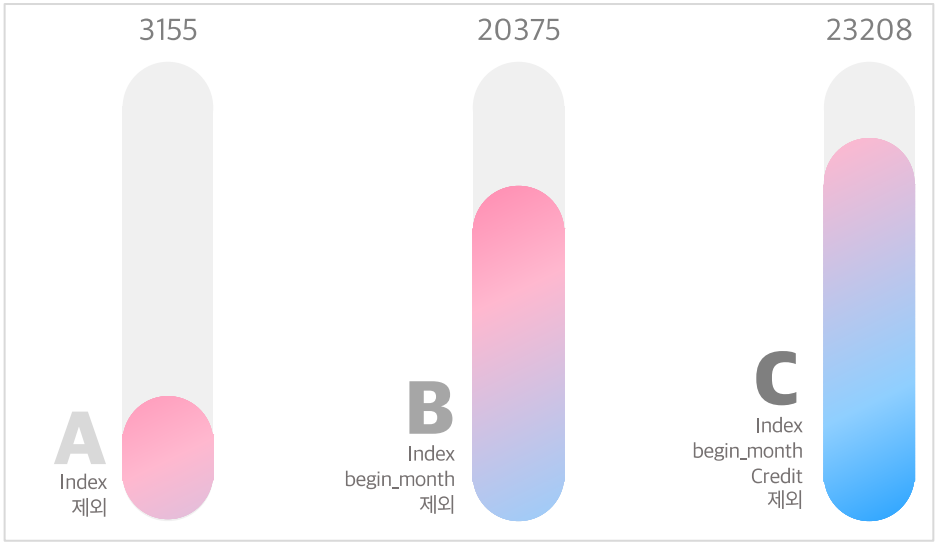
따라서 drop 하지 않고

NaN -〉 unknown 대체

Research

3-2. 데이터 전처리

- 중복값 및 이상치 -



■ Index 제외한 3155개 데이터 제거

Family_size 〉 7
Flag_MOBIL = 1 ▶ 제거

DAYS_EMPLOYED 〉 0 ▶ 0 처리

Research

3-2. 데이터 전처리

- Feature Engineering -

■ 수치형 파생 변수 생성 코드

```
for df in [train,test]:
  # before_EMPLOYED: 고용되기 전까지의 일수
  df['before_EMPLOYED'] = df['DAYS_BIRTH'] - df['DAYS_EMPLOYED']
  df['income_total_befofeEMP_ratio'] = df['income_total'] / df['before_EMPLOYED']
  df['before_EMPLOYED_m'] = np.floor(df['before_EMPLOYED'] / 30) - ((np.floor(df['before_EMPLOYED'] / 30) / 12).astype(int) * 12)
  df['before_EMPLOYED_w'] = np.floor(df['before_EMPLOYED'] / 7) - ((np.floor(df['before_EMPLOYED'] / 7) / 4).astype(int) * 4)

  #DAYS_BIRTH 파생변수- Age(나이), 태어난 월, 태어난 주(출생연도의 n주차)
  df['Age'] = df['DAYS_BIRTH'] // 365
  df['DAYS_BIRTH_m'] = np.floor(df['DAYS_BIRTH'] / 30) - ((np.floor(df['DAYS_BIRTH'] / 30) / 12).astype(int) * 12)
  df['DAYS_BIRTH_w'] = np.floor(df['DAYS_BIRTH'] / 7) - ((np.floor(df['DAYS_BIRTH'] / 7) / 4).astype(int) * 4)


  #DAYS_EMPLOYED_m 파생변수- EMPLOYED(근속연수), DAYS_EMPLOYED_m(고용된 달) ,DAYS_EMPLOYED_w(고용된 주(고용연도의 n주차))
  df['EMPLOYED'] = df['DAYS_EMPLOYED'] // 365
  df['DAYS_EMPLOYED_m'] = np.floor(df['DAYS_EMPLOYED'] / 30) - ((np.floor(df['DAYS_EMPLOYED'] / 30) / 12).astype(int) * 12)
  df['DAYS_EMPLOYED_w'] = np.floor(df['DAYS_EMPLOYED'] / 7) - ((np.floor(df['DAYS_EMPLOYED'] / 7) / 4).astype(int) * 4)

  #ability: 소득/(살아온 일수+ 근무일수)
  df['ability'] = df['income_total'] / (df['DAYS_BIRTH'] + df['DAYS_EMPLOYED'])

  #income_mean: 소득/ 가족 수
  df['income_mean'] = df['income_total'] / df['family_size']

  #ID 생성: gender, DAYS_BIRTH, income_total, income_type, edu_type, occyp_type 값들을 더해서 고유한 사람을 파악(*한 사람이 여러 개 카드
  df['ID'] = df['gender'].astype(str) + '_' + df['DAYS_BIRTH'].astype(str) + '_' + df['income_total'].astype(str) + '_'  + df['income_ty
```

executed in 129ms, finished 16:17:39 2022-06-15

신용등급 별 데이터 분포의 큰 차이 없음 파악

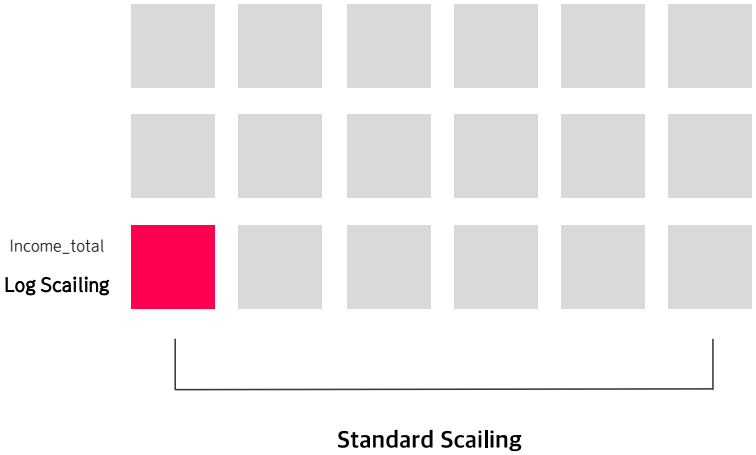
수치형 변수 활용 **총 12개 파생 변수** 생성

Child_num, DAYS_BIRTH, DAYS_EMPLOYED ; 파생 변수와 상관성이 높아 제거

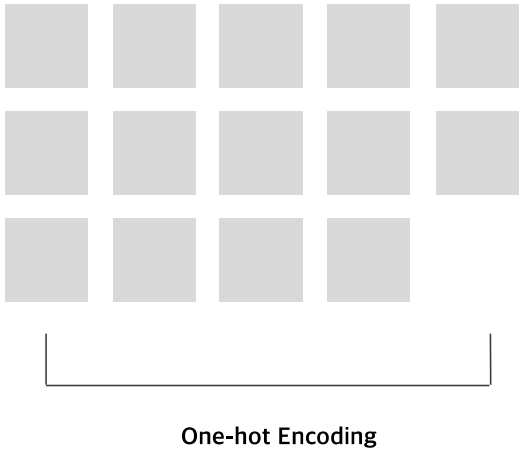
Research

3-2. 데이터 전처리

■ 수치형 변수



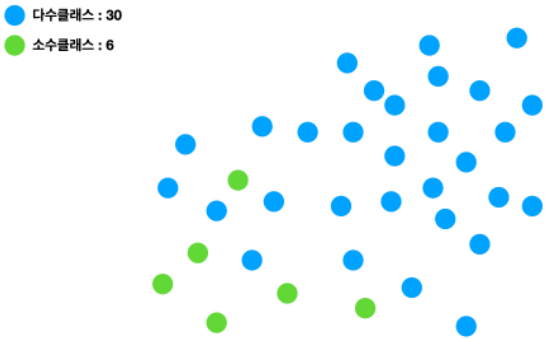
■ 범주형 변수



Research

3-2. 데이터 전처리

- Oversampling -



SMOTE-NC

; 수치형 변수와 범주형 변수가 함께 존재하는 경우 활용

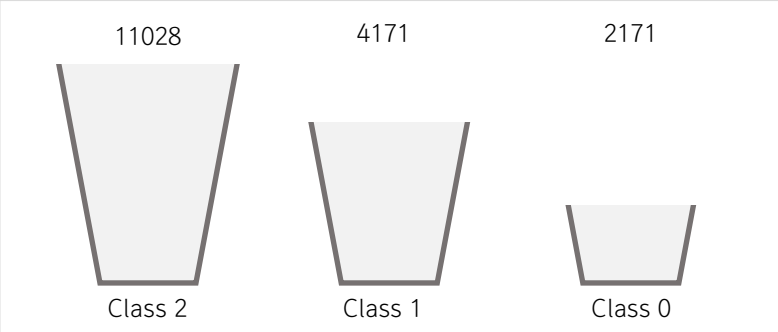
유클리디안 거리 계산
소수 클래스에서 각 샘플들의 KNN을 찾고
그 이웃들 사이에 선을 그어 데이터 무작위 생성

Research

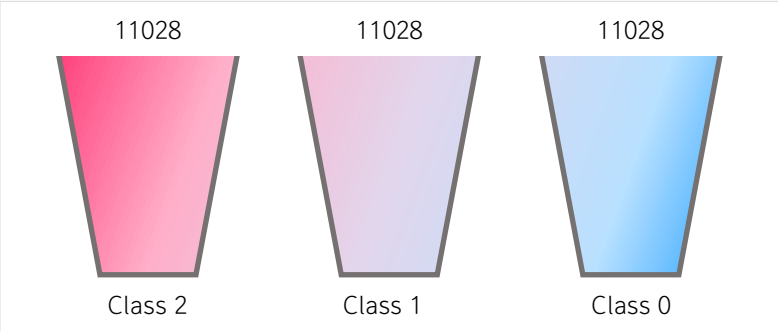
3-2. 데이터 전처리

- Oversampling -

■ 기존

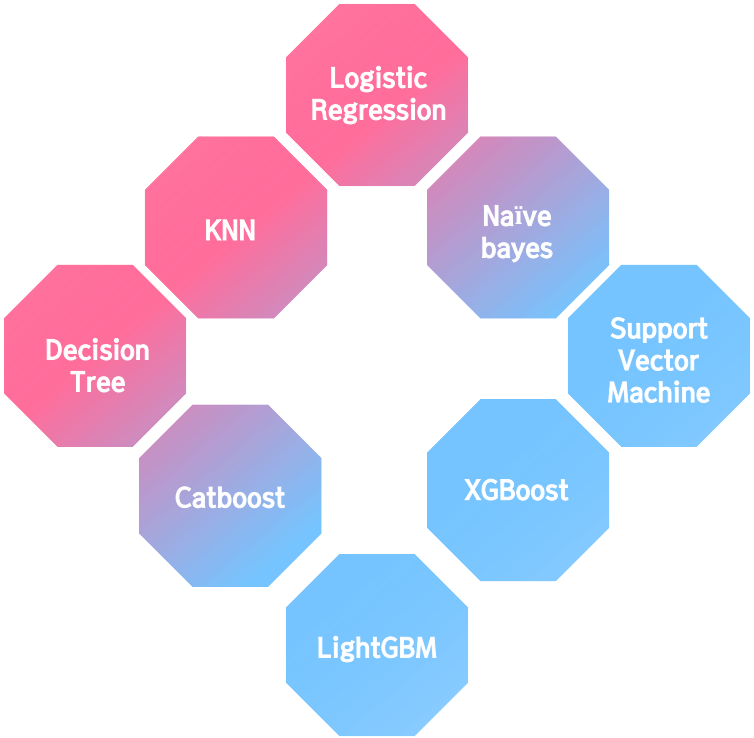


■ Oversampling



Research

3-3. 모델 생성 및 성능 비교



■ Parameter

n_est = 2000, seed = 42

■ StratifiedKFold

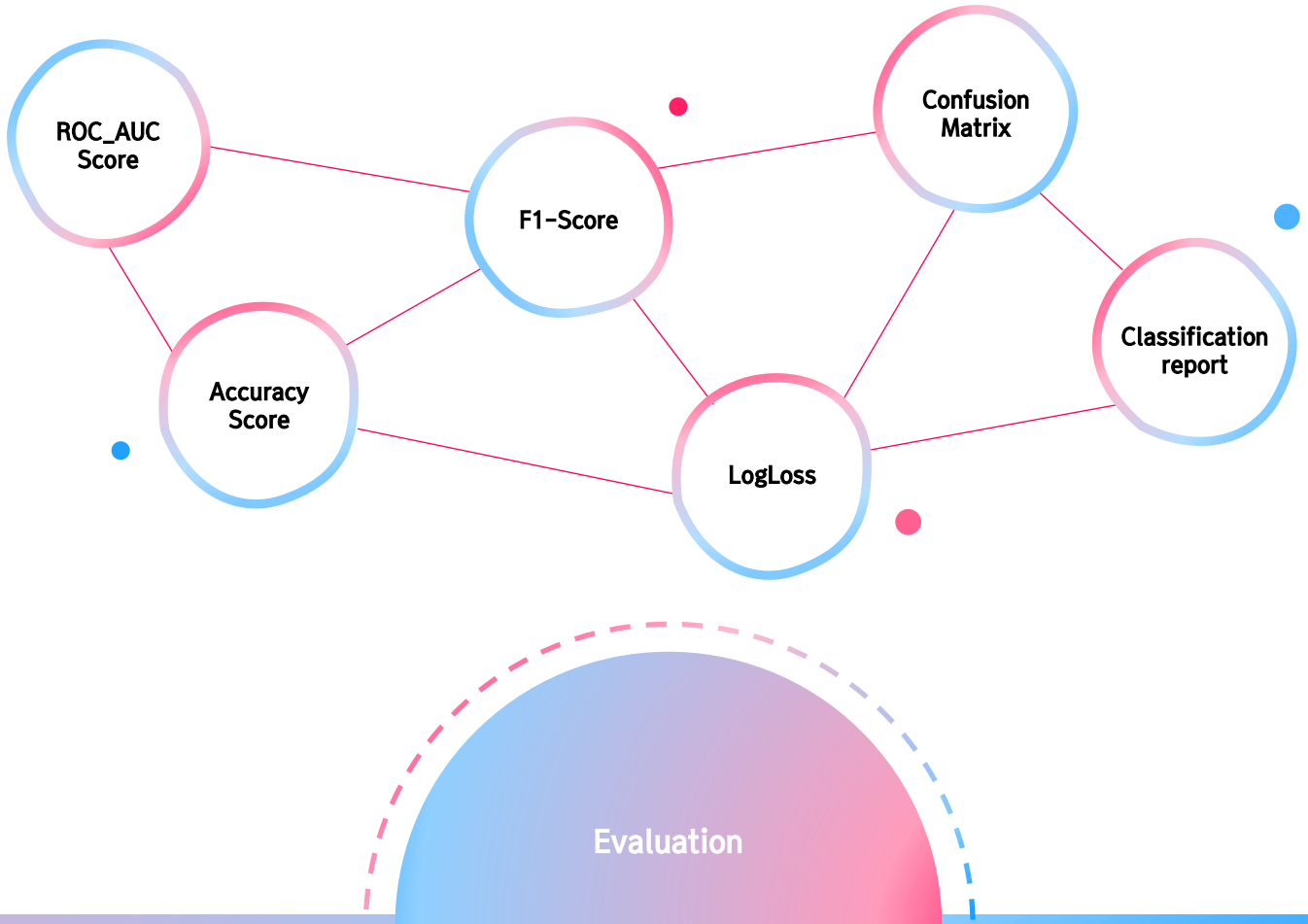
데이터 셋이 불균형 할 때 유용한 기법

Target이 고르게 분포될 수 있도록 fold를 나눔

Optimum Fold = 5

Research

3-3. 모델 생성 및 성능 비교



Research

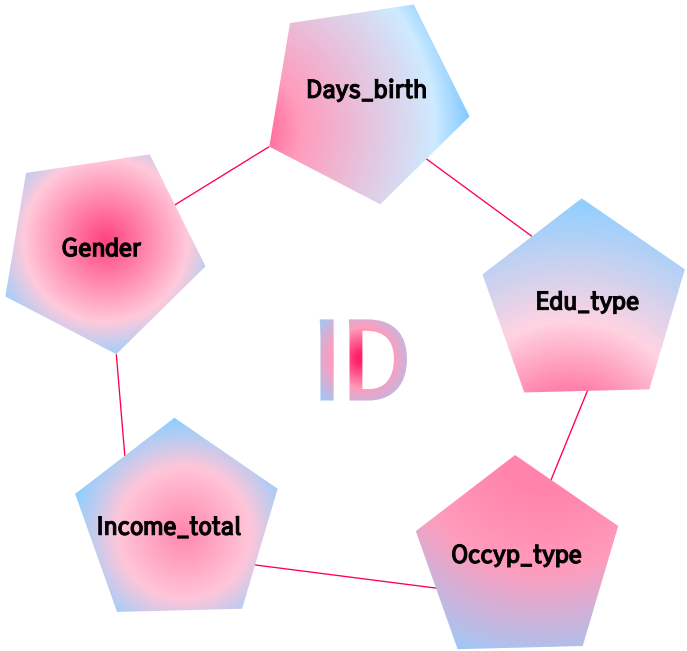
3-3. 모델 생성 및 성능 비교

Algorithm	Accuracy score	Algorithm	F1 score	Algorithm	Logloss	Algorithm	Roc auc score
LightGBM	**0.809062**	**LightGBM**	**0.808793**	**LightGBM**	**0.510125**	**LightGBM**	**0.921998**
Catboost	0.801808	Catboost	0.800959	XGBoost	0.542047	CatBoost	0.916272
XGBoost	0.793586	SVM	0.797224	Catboost	0.542547	XGBoost	0.912976
KNN	0.714031	XGBoost	0.694721	SVM	0.820924	KNN	0.885250
Decision Tree	0.657690	KNN	0.694721	Logistic	1.051366	SVM	0.876024
Logistic	0.455205	Decision Tree	0.657597	KNN	2.593647	Decision Tree	0.745790
SVM	0.422138	Logistic	0.450663	Naive Bayes	9.199845	Logistic	0.640787
Naive Bayes	0.384567	Naive Bayes	0.289933	Decision Tree	11.653466	Naive Bayes	0.595335

Research

3-4. 모델 성능 향상 및 최종 모델 선정

- ID -



Ex. F_12676_157500.0_State servant_Secondary / secondary special_Waiters/barmen staff

※ Begin_month : EDA과정에서 한사람이 여러 개의 카드를 만들 가능성 有, 활용하지 않음

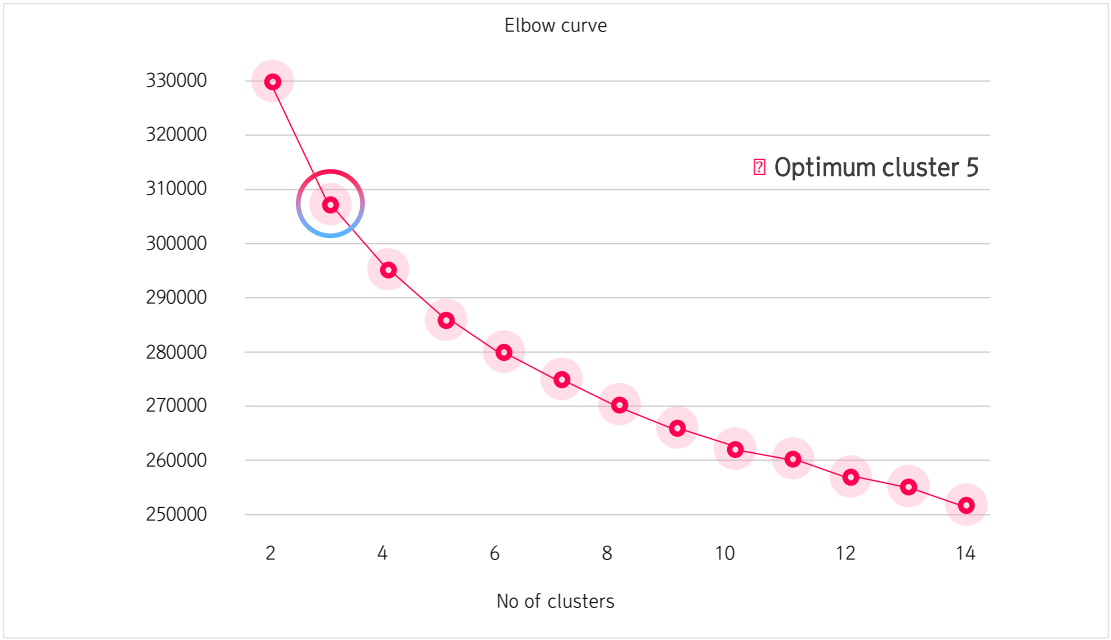
Research

3-4. 모델 성능 향상 및 최종 모델 선정

- Clustering -

K-prototype

연속형 자료 = 유클리디안 거리	범주형 자료 = 비유사도*가중치



Research

3-4. 모델 성능 향상 및 최종 모델 선정

- 최종 모델 선정 -

ID/Clustering 적용 후 각각 접합 ▶	XGBoost	LightGBM	Catboost
Categorical value	X	O	O
Encoding	O	X	Feature combination
Method	Level-wise	Leaf-wise	Oblivious decision tree
Velocity	Long	Fast	Fast
ID Issue	Variable ▲	Train 존재O, Test 존재X -> 직접처리	Order boosting -> 데이터 분포 문제 제거

Research

3-4. 모델 성능 향상 및 최종 모델 선정

- 모델 성능 비교 -

데이콘 타 참가자 상위그룹 logloss

DACON 커뮤니티 대회 교육 랭킹 더보기

대회안내 데이터 코드공유 토크 리더보드 제출

전체랭킹 〉

#	팀	팀 멤버	최종점수	제출수	등록일
1	소회의실		0.6581	77	일년전
2	Dswook		0.65862	66	일년전
3	Js4756		0.65913	77	일년전
4	초보산님		0.66003	23	일년전
...	...		...	...	...

T2 logloss

```
Valid Accuracy Score: 0.849444
Valid F1 Score:  0.849446
Valid Log Loss:  0.408876

Valid ROC_AUC_Score: 0.950088

classification_report - valid data
              precision    recall  f1-score   support

         0.0       0.88      0.89      0.89     11028
         1.0       0.86      0.81      0.83     11028
         2.0       0.80      0.85      0.83     11028

    accuracy                           0.85     33084
   macro avg       0.85      0.85      0.85     33084
weighted avg       0.85      0.85      0.85     33084


Confusion_matrix - valid data
[[9806  476  746]
 [ 605 8886 1537]
 [ 671  946 9411]]

Confusion_matrix_Normalize - valid data
[[0.88919115 0.04316286 0.06764599]
 [0.05486036 0.80576714 0.13937251]
 [0.06084512 0.08578165 0.85337323]]
```

► 25% ↑

Research

3-5. SHAP 분석 및 분석 결과

Shapley Additive exPlanations

SHAP

Shapley value

특정 값이 있을 때와 없을 때의 값 차이를 가능한 모든 조합으로 구한 것의 평균

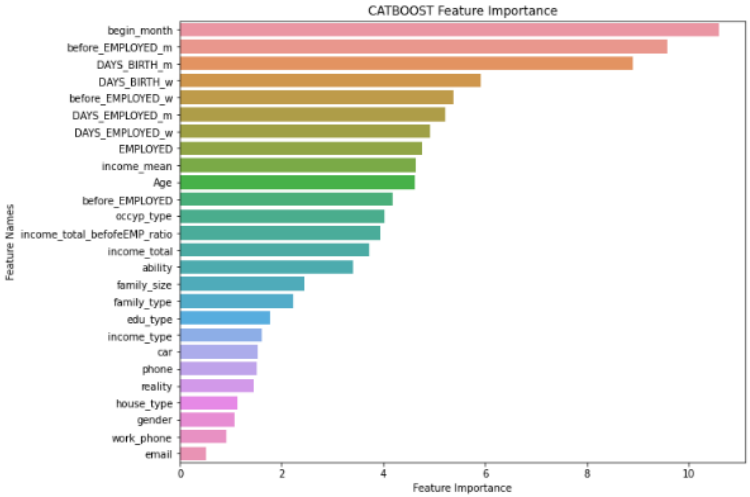
; Kernal SHAP, Deep SHAP, TreeSHAP

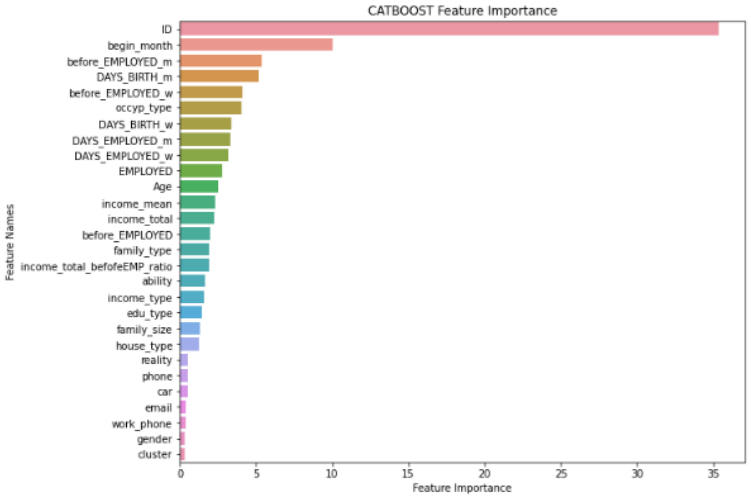
Research

3-5. SHAP 분석 및 분석 결과

- Catboost 자체 Feature Importance -

Catboost 자체 Feature Importance





■ class에 따른 각 feature의 **절대 영향도**

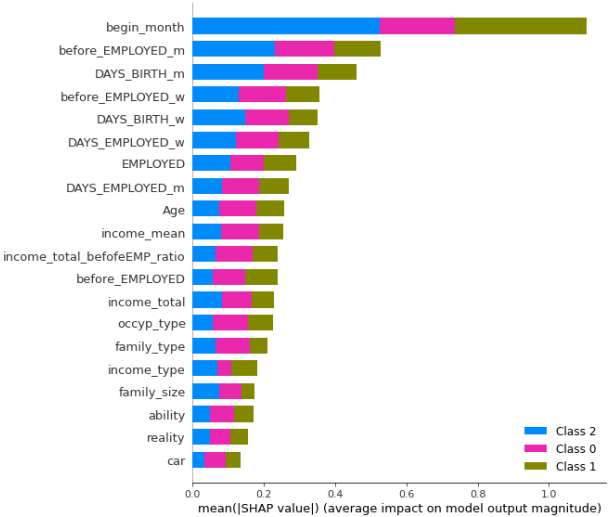
Research

3-5. SHAP 분석 및 분석 결과

- SHAP Feature Importance -

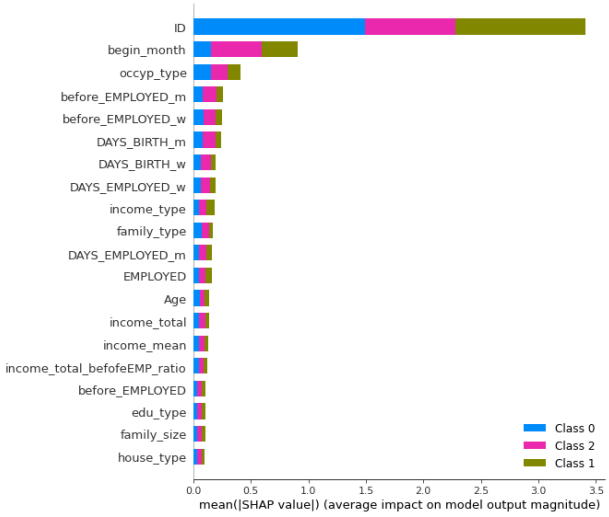
SHAP Feature Importance

ID & Cluster 생성 전



► Begin_month 압도적

ID & Cluster 생성 후



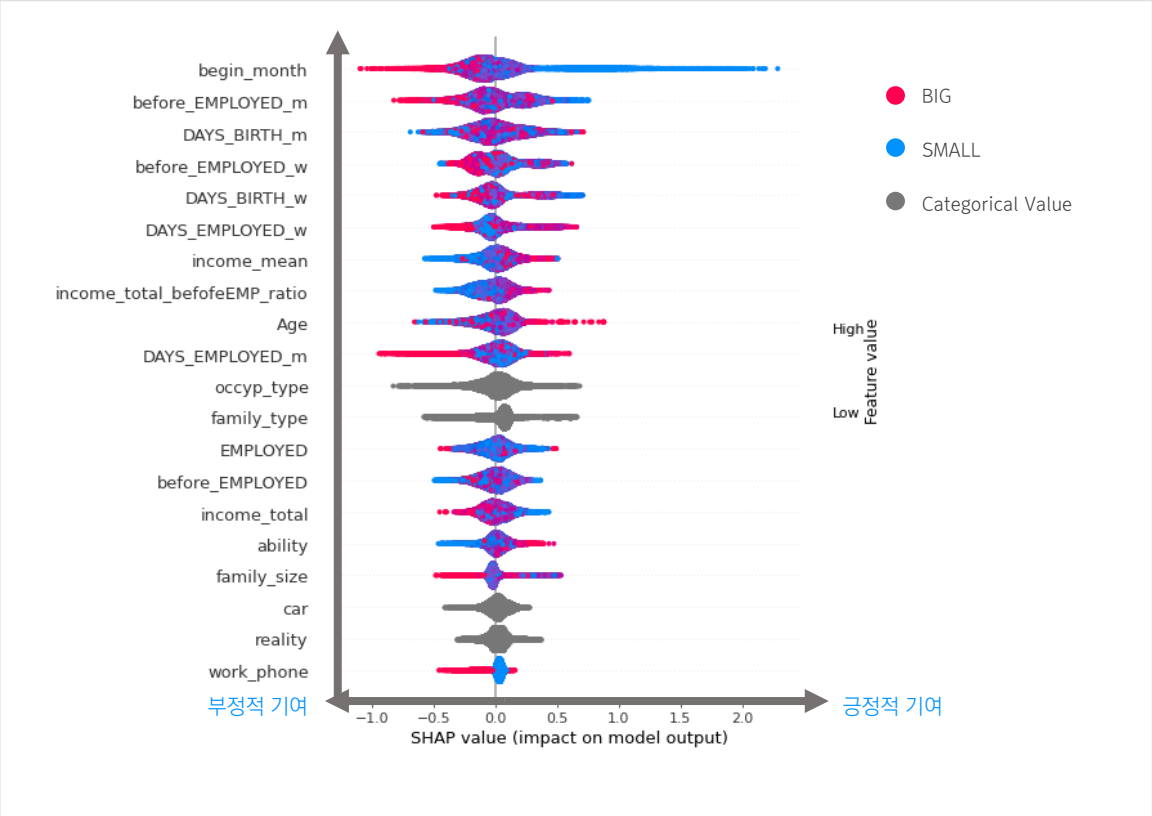
► ID 압도적

Research

3-5. SHAP 분석 및 분석 결과

- SHAP summary plot -

■ 그래프 설명

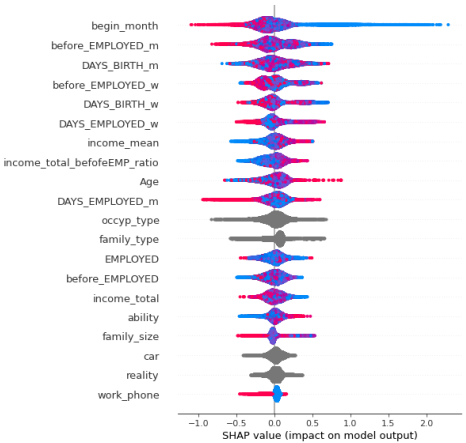


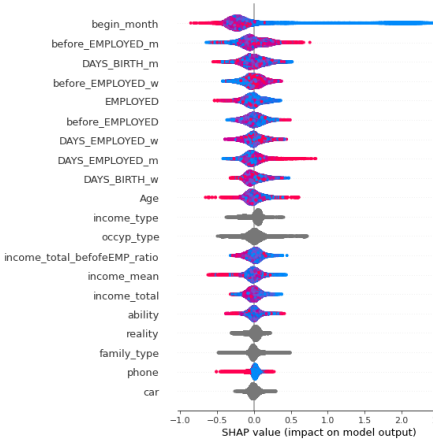
Research

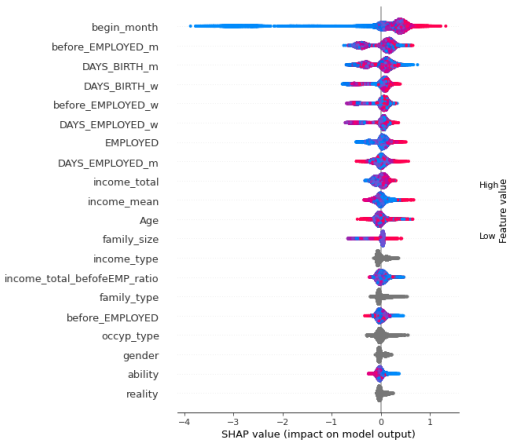
3-5. SHAP 분석 및 분석 결과

- SHAP summary plot -

ID / Cluster 생성 전







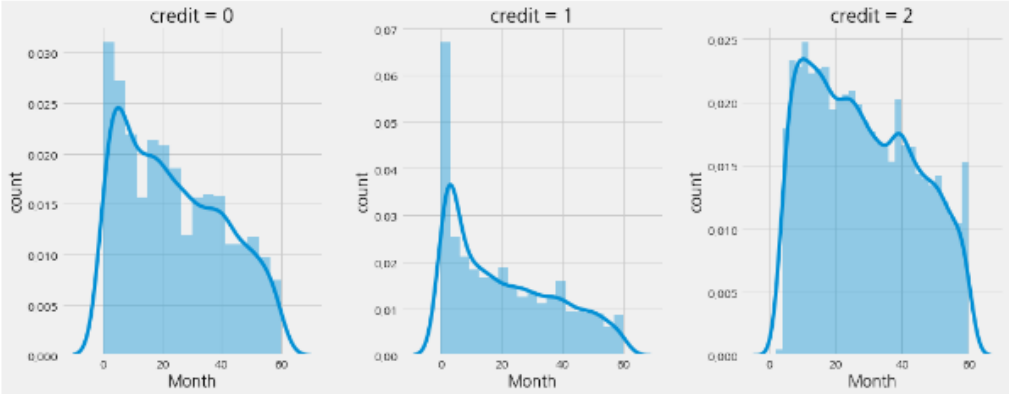
✓ Begin_month 정답 판별 기여도 큼

Research

3-5. SHAP 분석 및 분석 결과

- SHAP summary plot -

ID / Cluster 생성 전



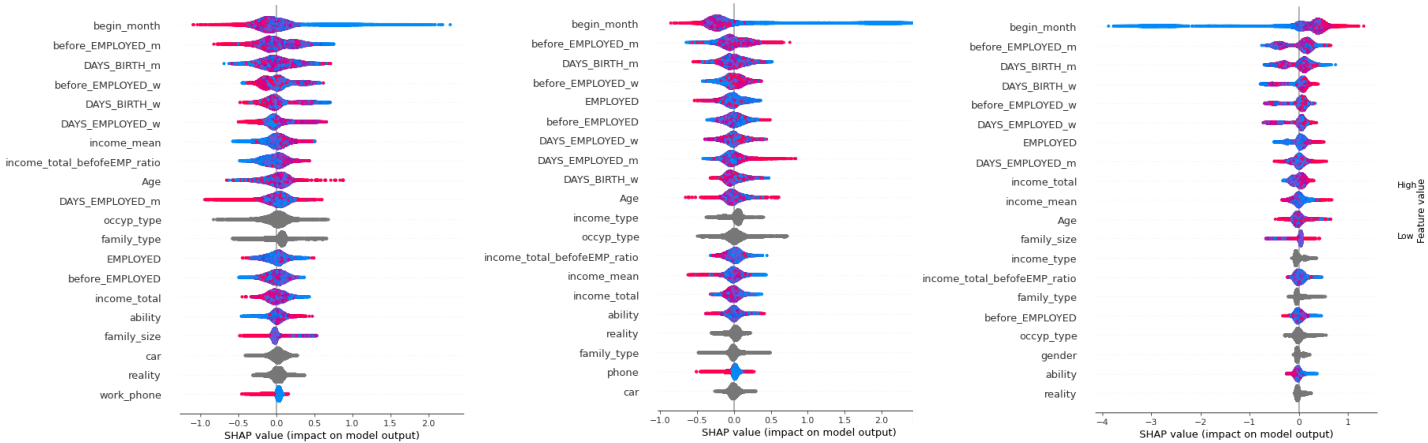
►Class2 에 begin_month 값이 높은 사람들의 데이터가 분포함

Research

3-5. SHAP 분석 및 분석 결과

- SHAP summary plot -

ID / Cluster 생성 전



일반적으로, 카드 발급 기간이 오래될수록 신용도가 높을 것이라 예측

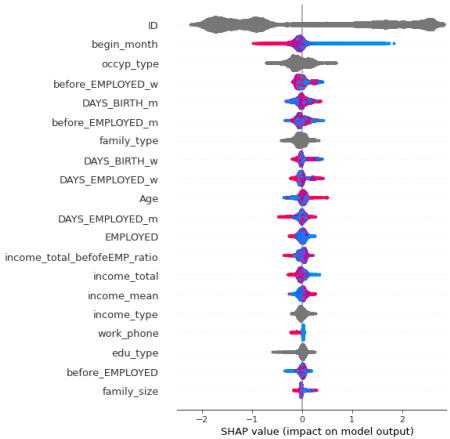
실제 데이터 ► ‘begin_month’ 가 높은 데이터가 class2에 주로 분포
모델 ► ‘begin_month’ 가 높은 값 class2로 분류

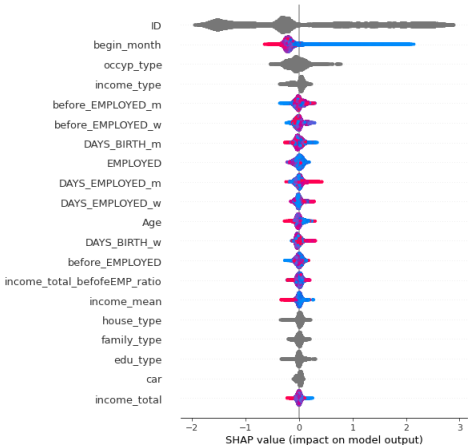
Research

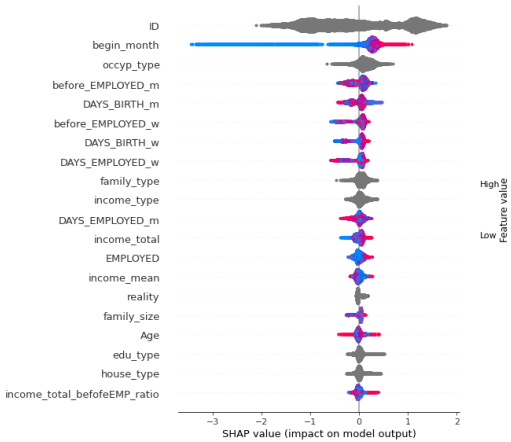
3-5. SHAP 분석 및 분석 결과

- SHAP summary plot -

ID / Cluster 생성 후







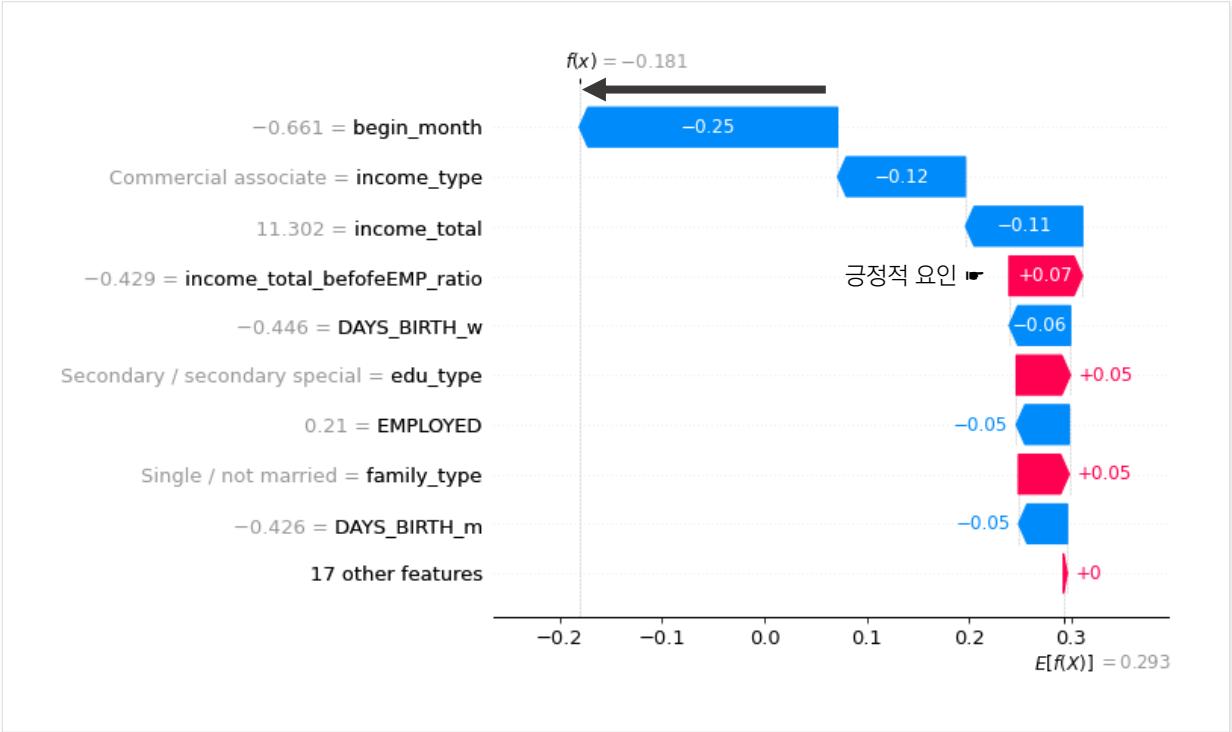
► ID의 feature importance가 제일 높음

Research

3-5. SHAP 분석 및 분석 결과

- SHAP waterfall plots -

■ 그래프 설명



► 요인 별 예측 기여도

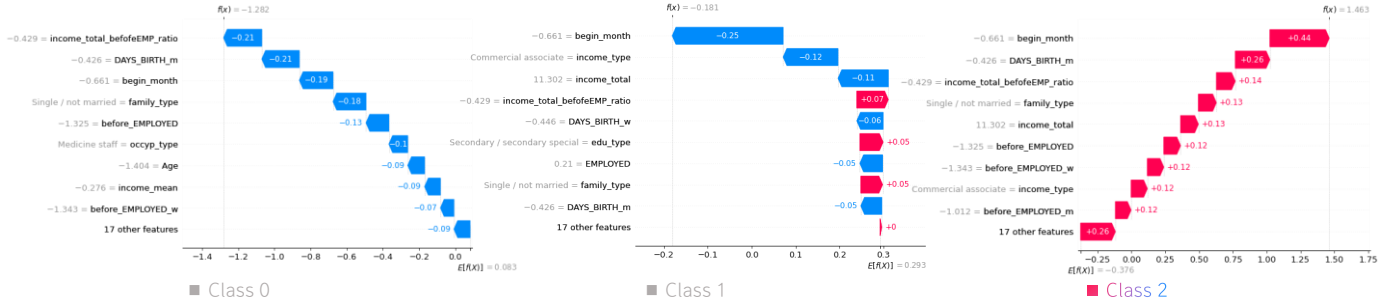
Research

3-5. SHAP 분석 및 분석 결과

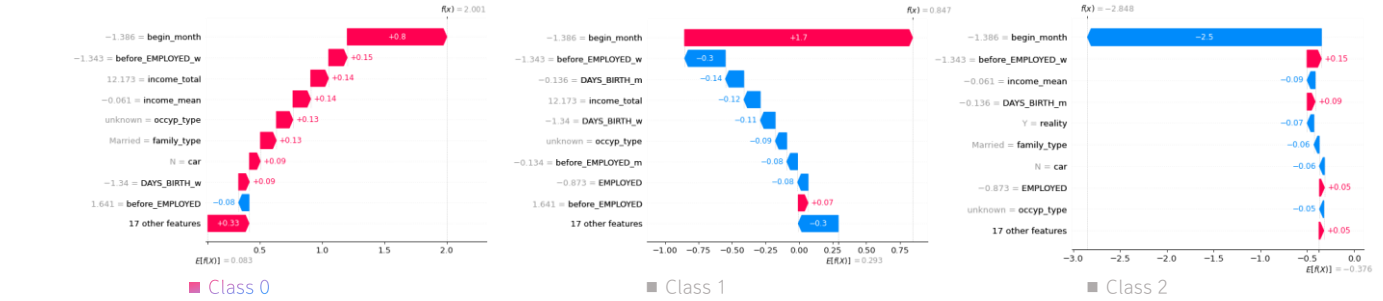
- SHAP waterfall plots -

ID / Cluster 생성 전

Row0(정답 : Class 2)



Row16540(정답 : Class 0)



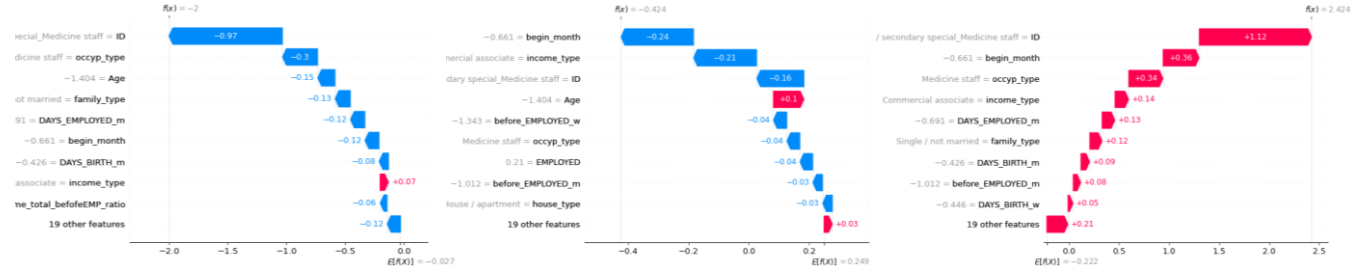
Research

3-5. SHAP 분석 및 분석 결과

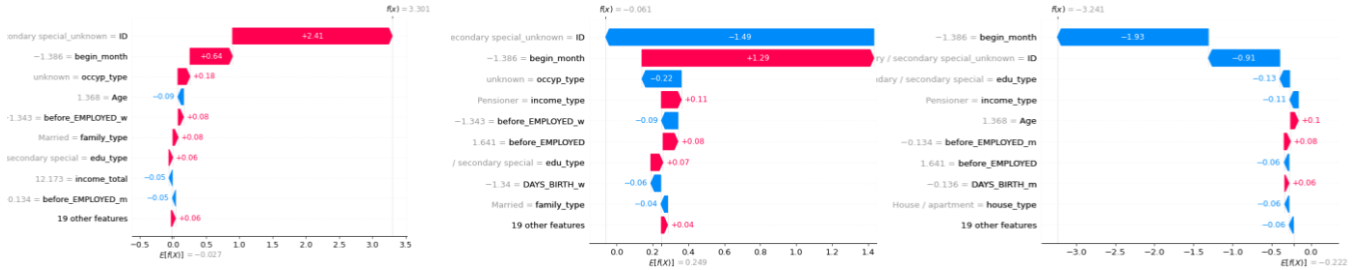
- SHAP waterfall plots -

ID / Cluster 생성 후

Row1(정답 : Class 2)

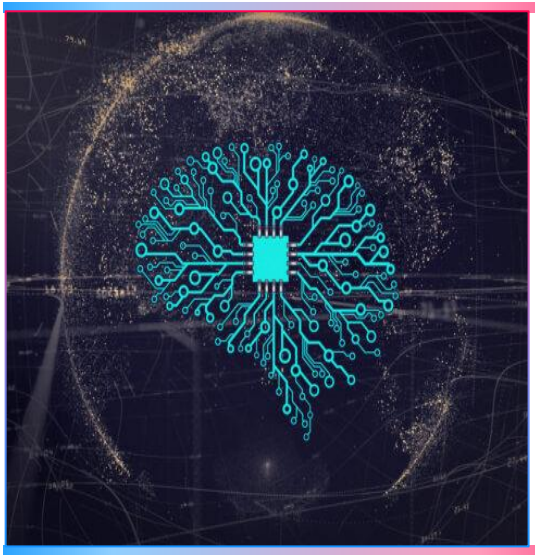


Row16540(정답 : Class 0)



Conclusion

한계점



- 데이터 노이즈 존재
- 실제 은행 데이터 용량과 차이
- 실제 은행 고객 데이터 특성 완벽 반영X

Conclusion

기대효과

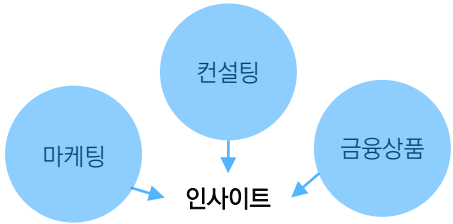
■ 고객입장

내 신용도가 왜 이래?

문제가 뭐지?

그럼 이렇게 하면 되겠다!

■ 은행입장



우리 은행 고객 신용도는 어떨까?

Credit Card
사용자 신용도
분류 모델 해석

T2 _ 박민규 강윤지 정현지 피재희