앱 사용성
데이터 분석을 통한
대출 신청 예측

최성수 : ttjh1234@naver.com
곽민섭 : minsub980422@naver.com
심범수 : sbs980205@naver.com

CONTENTS

대출신청 예측 메커니즘에 대한 요인 분석

: 개인의 대출 신청을 결정하는 요인은 무엇일까?

Chap 1

BackGround

주제선정배경

대출 신청 예측의 가치 ?

예측 모델 개발을 통해 가장 적절한 대출 상품을 고객에게 추천
모형 해석을 통해 주요 요인을 찾아 대출 상품을 개발하거나 마케팅 전략을 수립할 때에 이용할 수 있다.

01 금리의 지속적인 상승으로 이자 부담 증가 -> 대출조건 비교 필수적

02 대출 상품의 다양성 : 62개 은행의 수백가지 대출 상품

03 시간적 부족 -> 최적의 상품을 찾아보기에 제한적

Finda

최대한도, 최대금리, 한도 등
대출상품 주요 정보 수집 후
고객에게 정보 제공

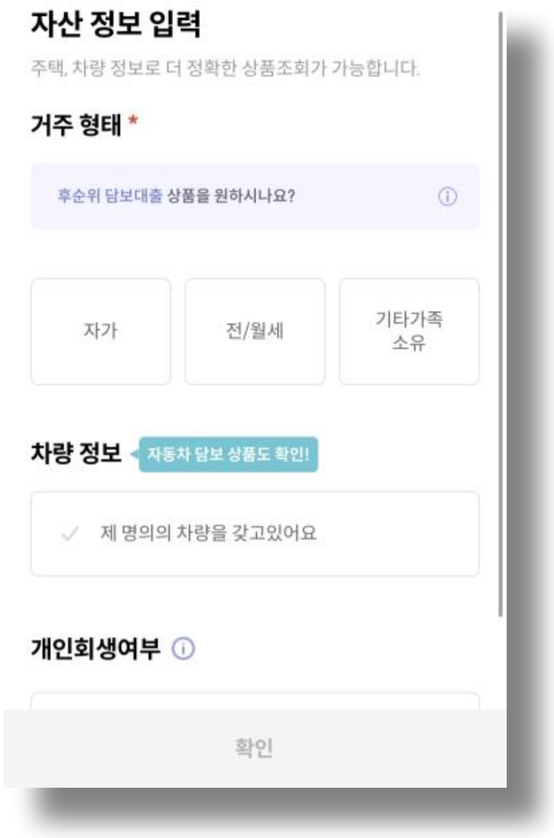
모형을 통한 예측으로 가장 적절한 상품 추천 + 마케팅 비용 절감 효과

Finda work-flow

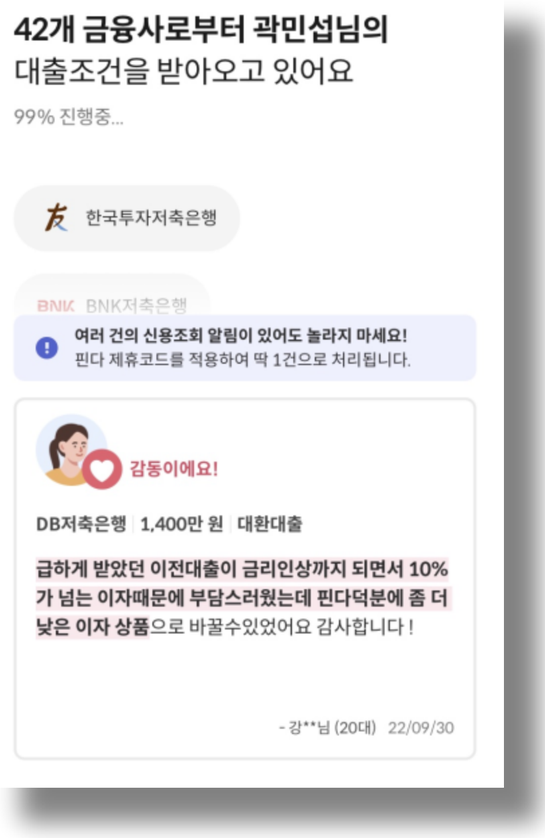
데이터 수집 구조 파악

Finda앱의 work-flow를 파악하여 데이터를 이해한 후 분석을 진행

01 개인 정보 입력



02 대출 가심사



03 대출 상품 추천

Chap 2

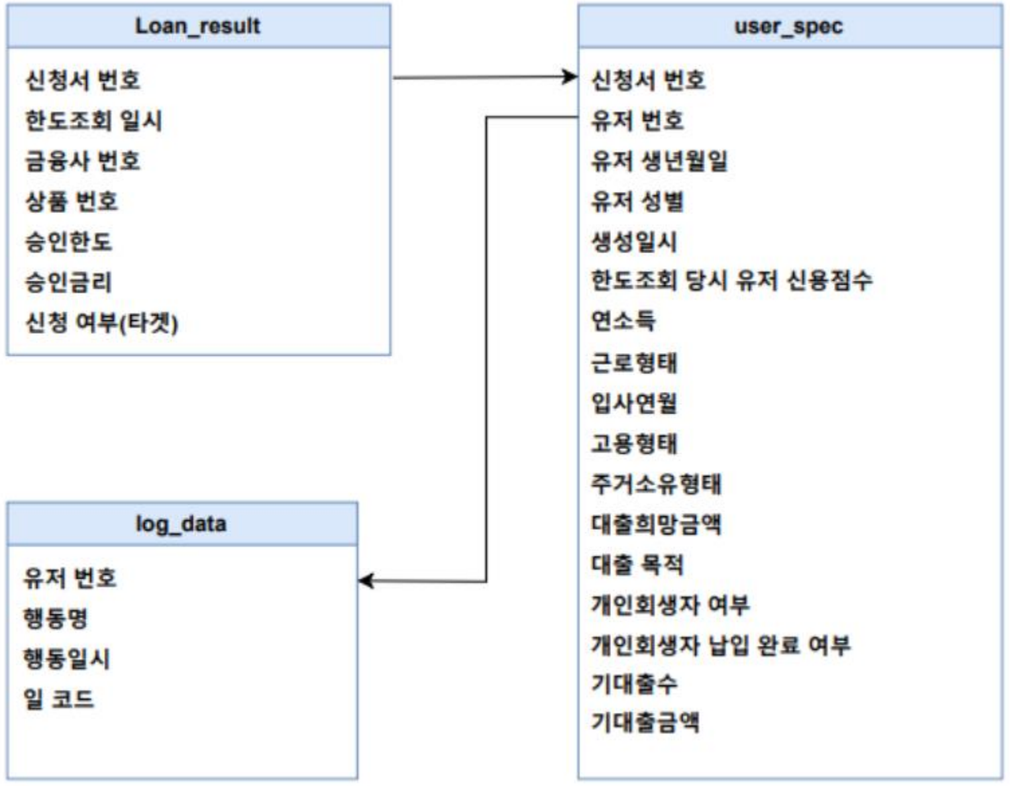
EDA

데이터 소개

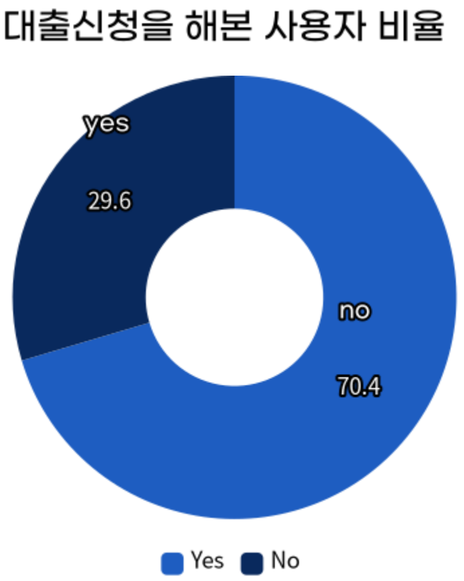
데이터 소개

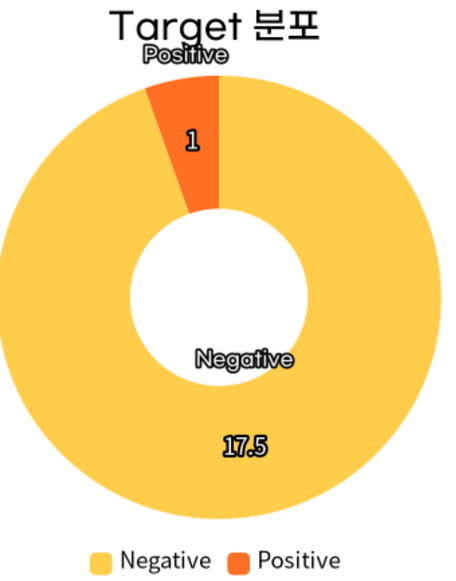
Finda 앱 사용성 데이터

제공 데이터



Data_Report





데이터 전처리

결측치 처리방식

결측치는 데이터의 구조적 논리성을 이용한 방식을 우선함.
논리적으로 처리할 수 없는 데이터의 경우 모델링을 이용하여 대치.

생년월일, 성별은 user_id 별로 같아야 하므로
user_id를 조회하여 값이 존재하면 해당 값으로 대치.

user_id	insert_time	birth_year	gender	crdit_score	personal_ rehabilitation_yn	personal_rehabilitation_complete_yn	existing_loan_cnt
1	2012-01-01-00:00	19900101	1	nan	0	nan	1
1	2012-02-01-00:00	nan	1	400	nan	0	nan
1	2012-03-01-00:00	nan	nan	350	0	0	2
1	2012-04-01-00:00	19900101	nan	420	1	0	4
1	2012-05-01-00:00	19900101	nan	400	1	0	nan

개인회생자 여부, 개인회생자 납입 여부는
user_id와 insert_time을 고려하여 대치.

데이터 전처리

결측치 처리방식

결측치는 데이터의 구조적 논리성을 이용한 방식을 우선함.
논리적으로 처리할 수 없는 데이터의 경우 모델링을 이용하여 대치.

credit_score는 같은 유저라도 조회 시간에 따라 달라질 수 있으므로,
insert_time을 고려하여 가중평균한 값으로 대치

user_id	insert_time	birth_year	gender	crdit_score	personal_ rehabilitation_yn	personal_rehabilitation_complete_yn	existing_loan_cnt
1	2012-01-01-00:00	19900101	1	nan	0	nan	1
1	2012-02-01-00:00	nan	1	400	nan	0	nan
1	2012-03-01-00:00	nan	nan	350	0	0	2
1	2012-04-01-00:00	19900101	nan	420	1	0	4
1	2012-05-01-00:00	19900101	nan	400	1	0	nan

기대출수는 0이 nan으로 입력되어 있으므로 0으로 대치

데이터 전처리

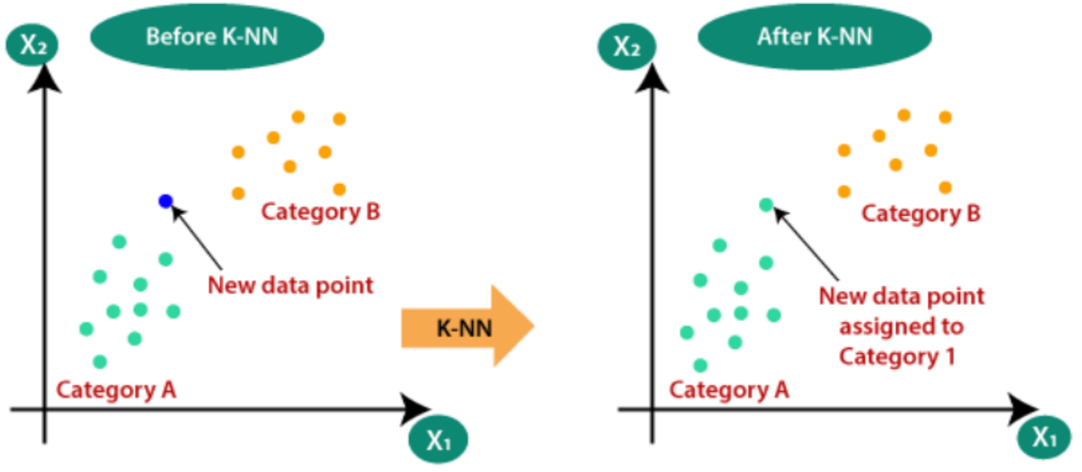
결측치 처리방식

결측치는 데이터의 구조적 논리성을 이용한 방식을 우선함.
논리적으로 처리할 수 없는 데이터의 경우 모델링을 이용하여 대치.

모델링 대치 : KNN-Imputer -> XgboostRgressor

KNN-Imputer란 ?
누락되지 않은 피처가 모두 가까운 경우에 두 샘플을 이웃으로 평가하고,
각 표본의 결측치는 학습 세트에서 찾은 k개의 이웃의 가중평균을 사용하여 대치한다.

범주형의 경우 voting 을 사용 !



상대적으로 중요하다고 판단한 existing_amt 변수 외 변수를 모두 대치 후 existing_amt 를 XGBRegression 알고리즘으로 대치.

MAE = 5000000 , R^2 = 0.5

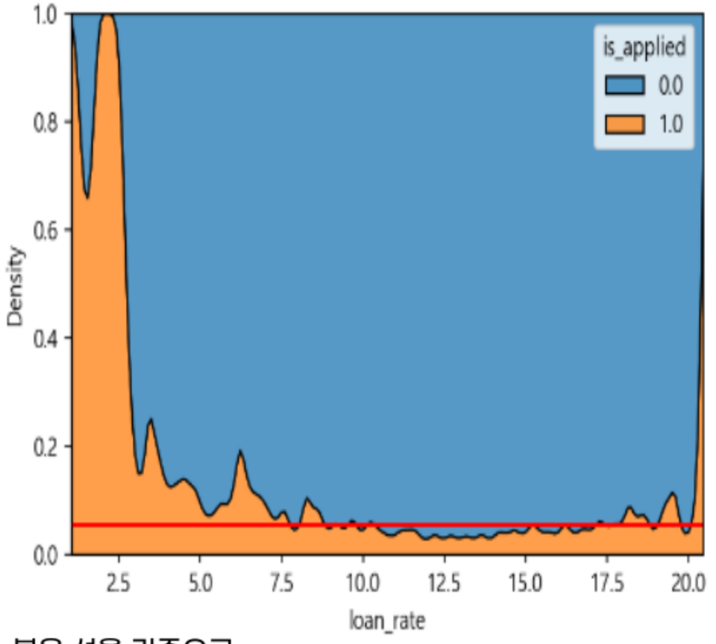
데이터 분석

연속형 변수 분석

이변량 분석, 히스토그램을 통해 target과의 연관성을 파악

연속형 변수

로지스틱 회귀모형에 적합한 뒤 회귀계수의 유의성을 t분포를 이용해 검정

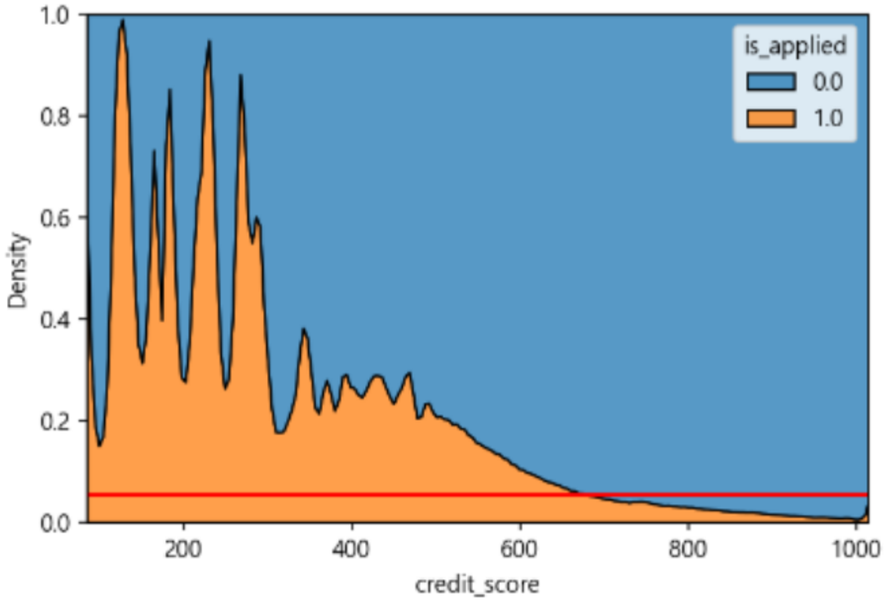


붉은 선을 기준으로
분포가 불안정 할수록 높은 연관성

Loan_rate

t-통계량	p-value
약 -1915	< 0.0001

-> 연관성 존재



Credit_score

t-통계량	p-value
-2007	< 0.0001

-> 연관성 존재

데이터 분석

범주형 변수 분석

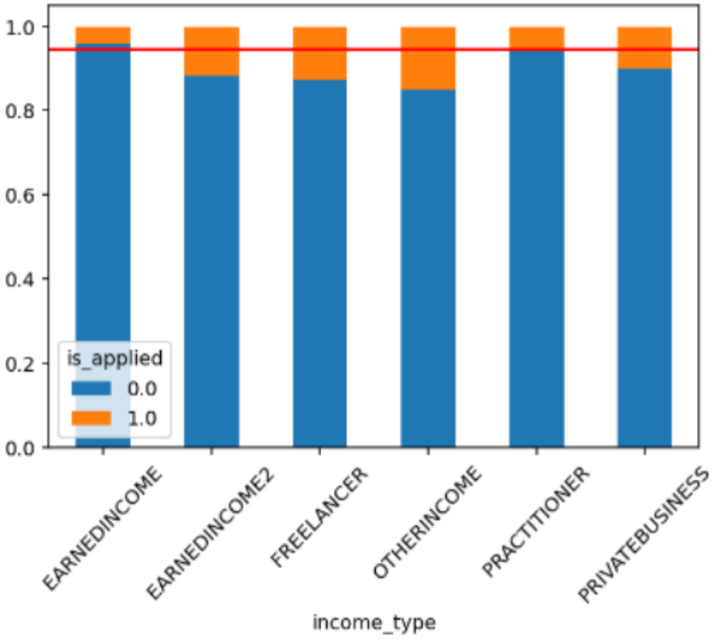
카이제곱 독립성 검정으로 target과의 연관성을, 박스플롯으로 분포를 파악

범주형 변수

카이제곱 독립성 검정 : 검정통계량 $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$ 을 이용하여 두 변수의 독립성을 검정한다.

단, 모든 셀의 기대빈도가 5이상인 경우에만 통계적 가정을 만족하므로 이를 준수하여 검정을 진행하였다.

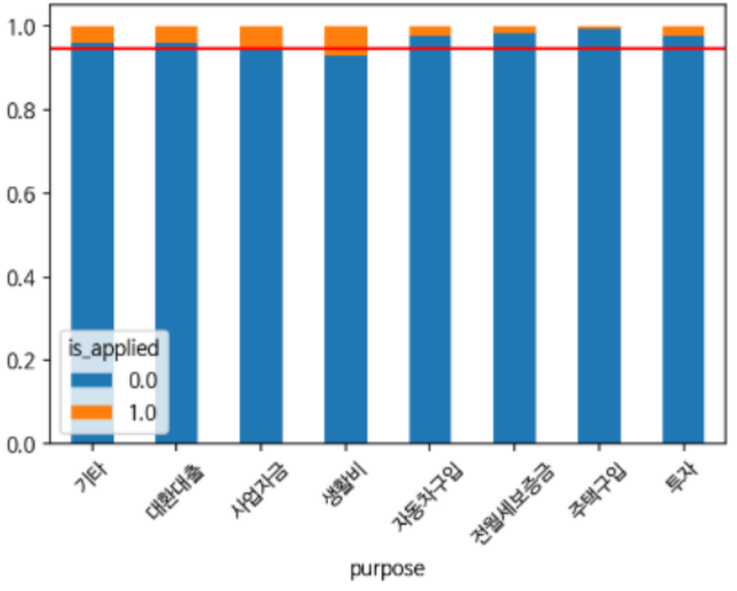
범주별 0 : 1 비율



Income_type

카이제곱 통계량	p-value
약 155223	< 0.0001

-> 연관성 존재



purpose

t-통계량	p-value
74609	< 0.0001

-> 연관성 존재

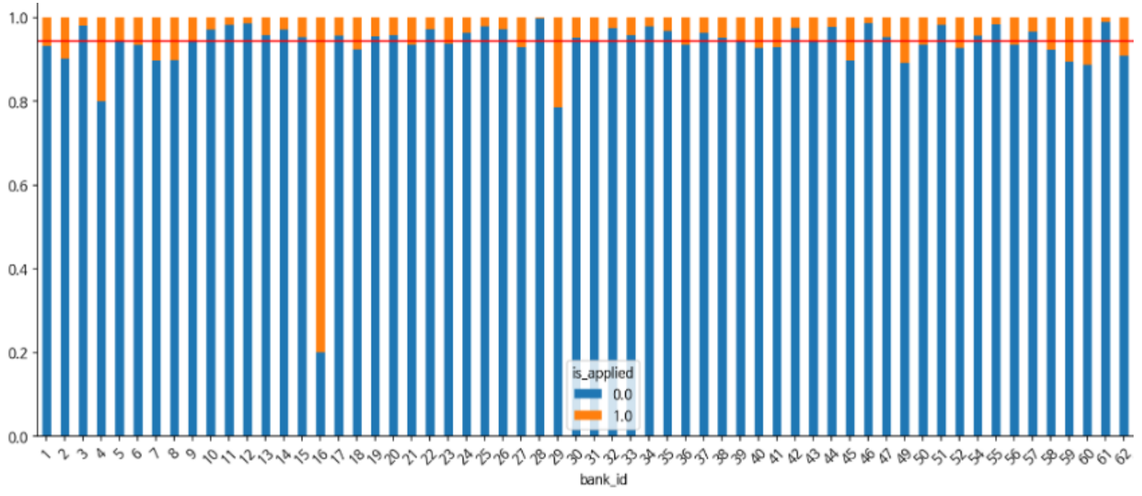
데이터 분석

범주형 변수 분석

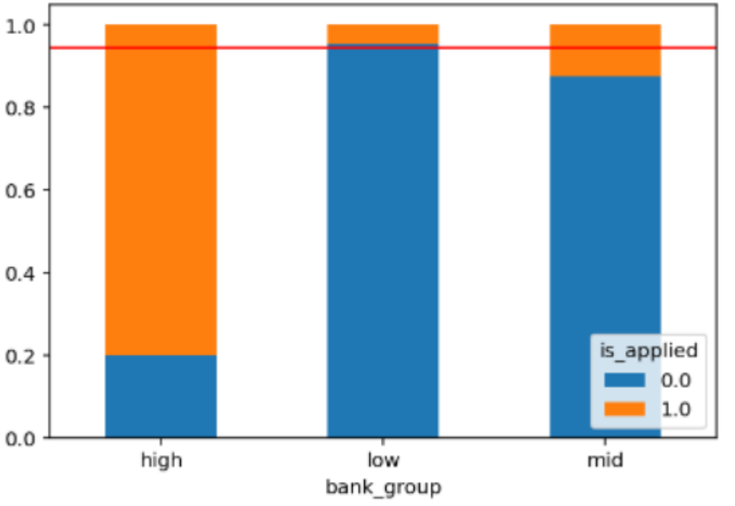
one-hot-encoding : 범주형 변수를 [1, 0, 0, 0] 의 벡터형태로 표현하는 기법

범주형 변수 구획화

특정 범주형 변수는 범주의 갯수가 너무 많아 one-hot-encoding처리가 어려웠음.
-> target과의 연관성을 기준으로 high, mid, low로 구획화



변환 전	변환 후
62개	3개



파생변수 생성

파생변수의 유형

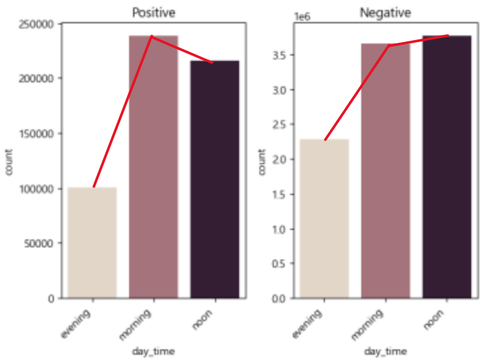
①시간 관련 파생변수, ②고객 관련 파생변수

시간 관련 파생변수

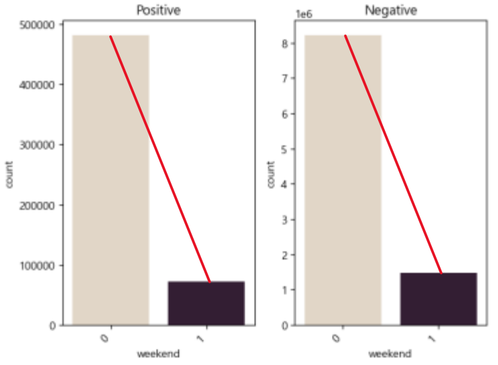
모형에 시간 관련 정보를 학습시키기 위하여 생성.
분석결과 day_time과는 상관이 존재, 주말과는 큰 연관성이 없다고 판단

feature	decription
daytime	한도 조회 시간에 따라 morning, noon, evening 으로 분할
weekend	한도 조회 시간에 따라 주말이면 1 그렇지 않으면 0

다른 분포 형태



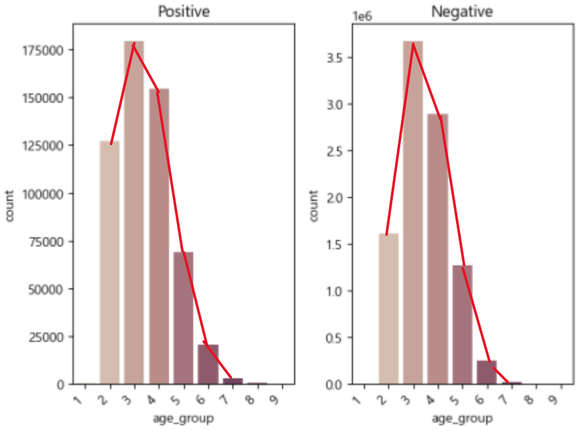
동일한 분포 형태



고객 관련 파생변수

고객 관련 정보를 기반으로 연령대, 근속개월 변수를 추가

feature	decription
age_group	birth_year를 고려하여 고객의 나이를 10단위로 범주화
serving_term	loanapply_insert_time과 company_enter_month을 이용하여 근속개월 변수 추가



다른 분포 형태

파생변수 생성

파생변수의 유형

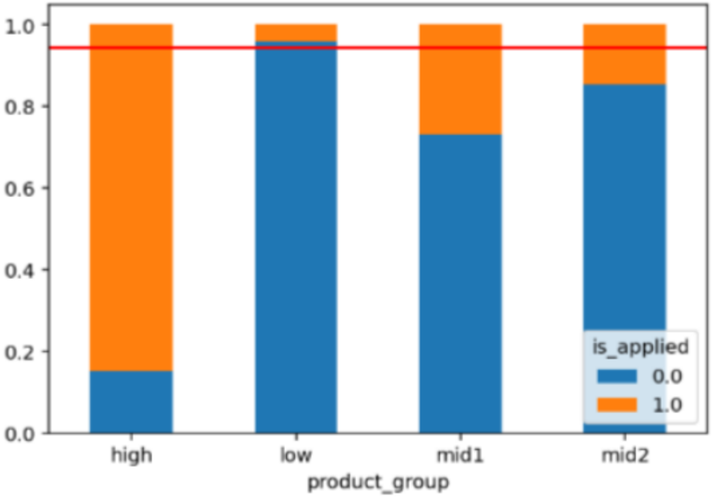
③대출 관련 파생변수

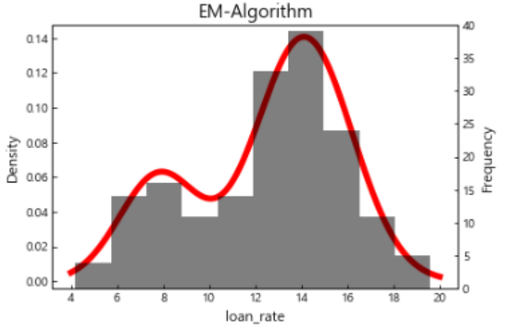
대출 관련 파생변수

대출 관련 정보를 기반으로 이자 관련 변수를 추가
대출신청 비율에 따른 주요 금융사와 대출 상품을 인식하기 위해 생성

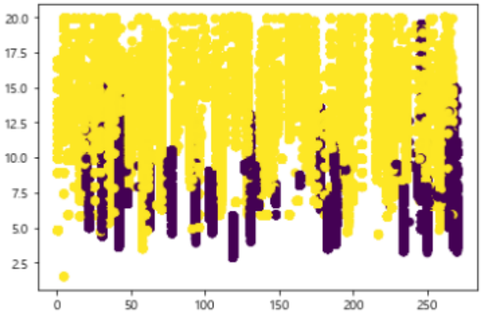
feature	decription
interest	loan_limit * loan_rate 으로 연 이자 고려
deinterest	desired_amount * loan_rate로 희망금액과 금리의 교호작용 고려
bank_group	범주형 변수 분석 결과에 따라 bank_id를 구획화
product_group	범주형 변수 분석 결과에 따라 product_id를 구획화
pr_group	EM Algorithm을 통해 loan_rate로 product_id를 구획화

변환 전	변환 후
168개	4개





대출 상품 별 평균 Loan rate의 분포



구획화 후 대출 상품 별 Loan rate의 분포

파생변수 생성

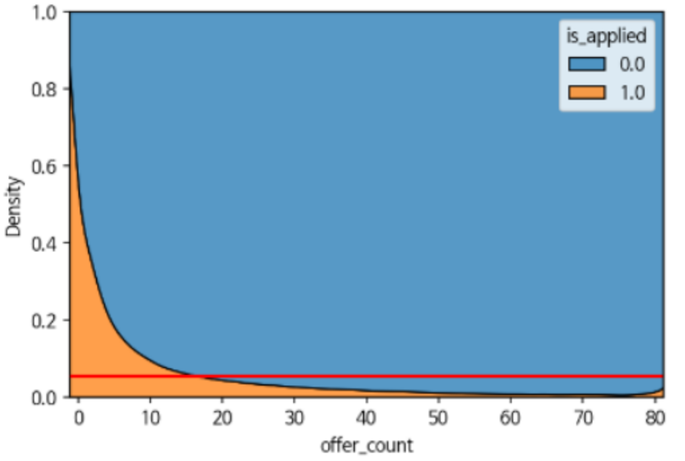
파생변수의 유형

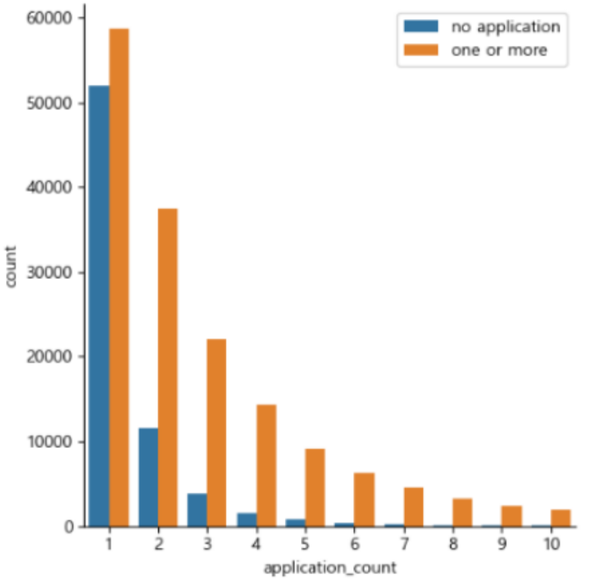
④대출-고객 관련 파생변수

대출-고객 관련 파생변수

대출 상품 데이터는 고객 특성을 제대로 고려하지 못함.
고객 데이터는 대출 상품의 특성을 제대로 고려하지 못함.
-> 대출 상품 정보와 고객 정보는 서로 유기적인 관계가 존재하여 이를 모델링에 반영해야 함

feature	decription
offer_count	추천받은 상품의 수
dm_limit	desired_amount - loan_limit 로 대출 금액 충족도 고려
enough_desired	max(dm_limit , 0) 로 대출금액 충족 여부 고려
a_DTI	(amt + loan_limit) / yearly_income (기존 대출과 추천받은 대출금액을 갚을 수 있는 능력)
a_DTI2	(amt + desired_amount) / yearly_income (기존 대출과 추천받은 대출금액을 갚을 수 있는 능력)
application_count	과거 대출 상품을 조회한 횟수





대출 조회를 많이 할수록 대출 신청 비율이 증가

파생변수 생성

파생변수의 유형

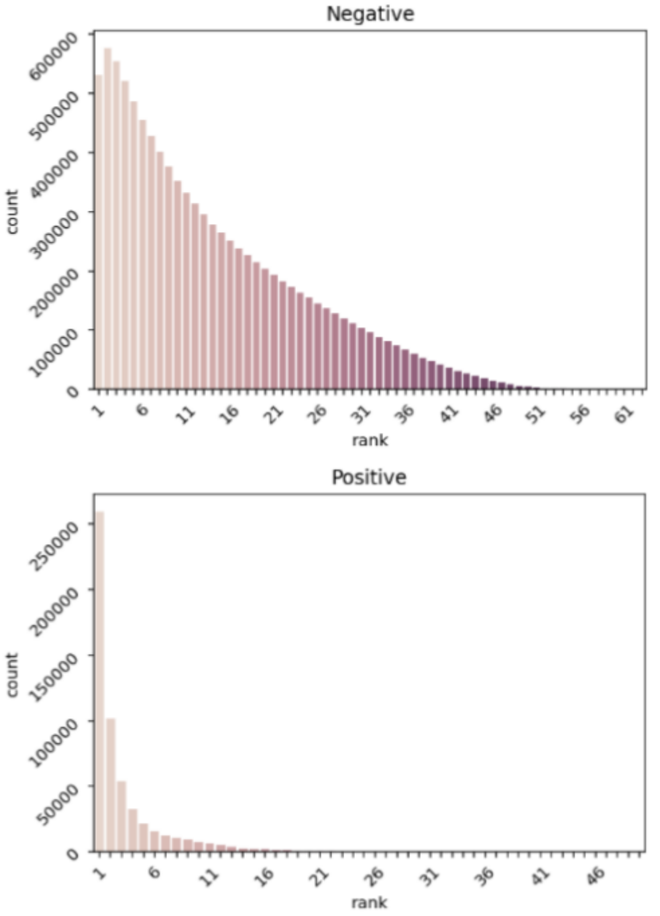
④대출-고객 관련 파생변수

대출-고객 관련 파생변수

대출 상품 데이터는 고객 특성을 제대로 고려하지 못함.
고객 데이터는 대출 상품의 특성을 제대로 고려하지 못함.
-> 대출 상품 정보와 고객 정보는 서로 유기적인 관계가 존재하여 이를 모델링에 반영해야 함

feature	decription
rank	desired_amount - loan_limit 로 대출 금액 충족도 고려
loan_rate max, min, mean	추천받은 상품 금리의 최솟값, 최댓값, 평균값
loan_limit max, min, mean	추천받은 상품 한도의 최솟값, 최댓값, 평균값
cum_applied	과거 대출 상품을 신청한 횟수

음성과 양성의 분포 차이가 극명함



Chap 3

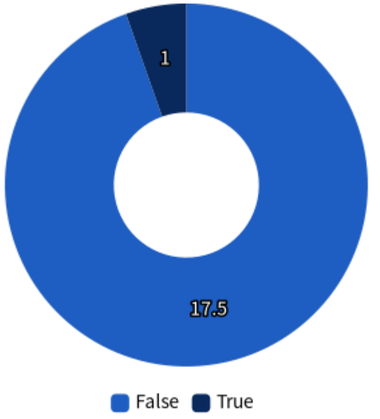
Modeling

모델링 목표

목표 수립 과정

모델링은 타겟 불균형 해소와 해석을 중점적으로 고려

Target 분포



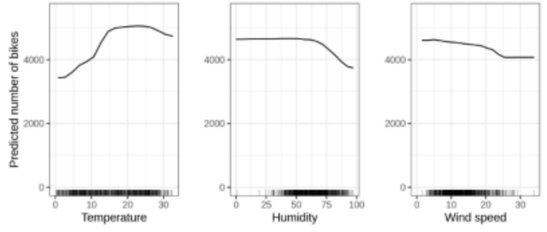
1. 타겟 불균형 문제 해소

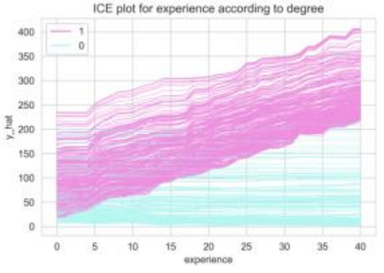
주어진 데이터는 is_applied 의 비율이 1 : 17.5
이 경우 모든 데이터를 False 라고 예측 했을때 모델의 accuracy는 97% 임.
그러나, 주어진 task가 대출 신청 예측이므로 실제 True를 False라고 예측하는 오류가 치명적임.

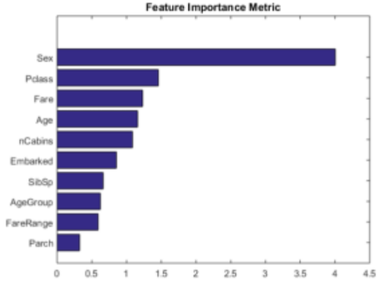
=> accuracy만 이용하기보다 recall, precision 등 다양한 평가 지표로 모형을 선정.

2. 해석력

모형 적합 후 해석을 통해서 대출 신청 예측 결정 요인을 파악하는 것을 목적으로 함.
따라서 해석이 가능한 모델을 이용하고 PDP, ice ,ALE, Feature_importance 등 지표로 요인 분석







모델 개요

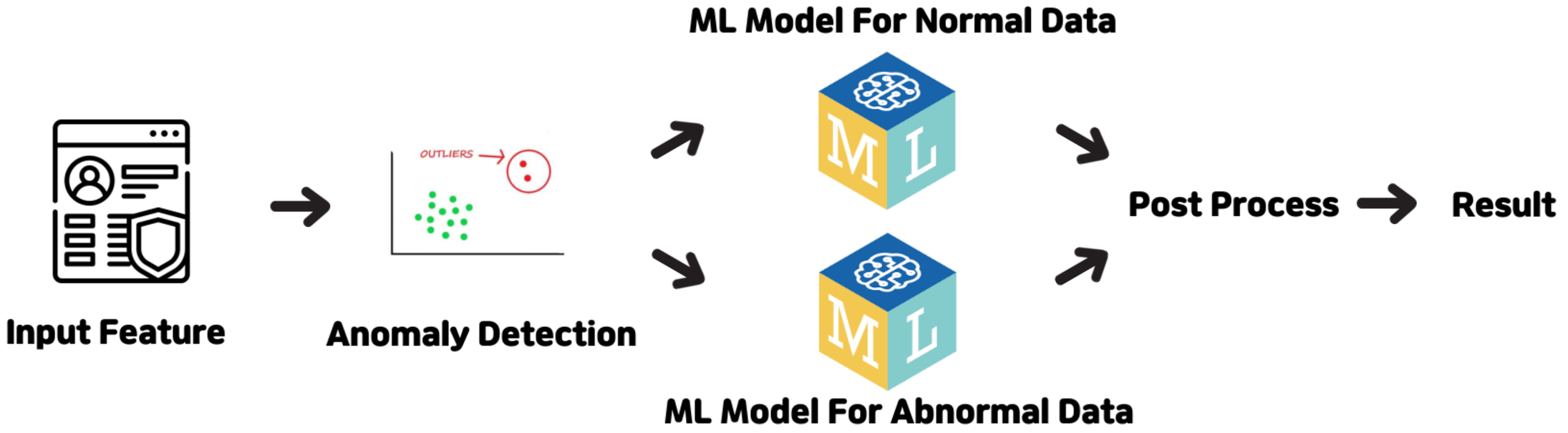
AnoImaly Detection

모델링은 타겟 불균형 해소와 해석을 중점적으로 고려

전반적인 모델링 순서도

초기 Input Feature들로 부터 학습을 진행한 결과, 대출 신청 여부를 판단하는 데 있어 불규칙적인 특성을 가지는 데이터가 존재한다고 판단.
전체 데이터들에서 일반화된 결과를 도출하는 것을 막는 Data Instance들을 분리하여 두 개의 독립적인 모델을 생성하고, 결과를 집계.

=> Anomaly Detection 결과로 부터 입력 데이터 간의 이질성을 해소하고 모델링 후처리를 통해 결과 생성.



모델링 방법

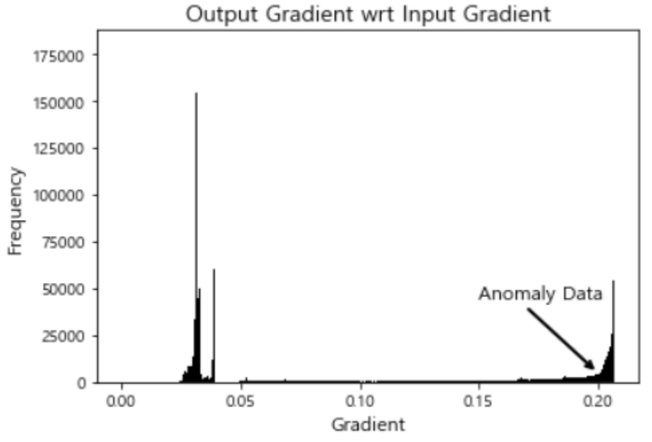
Deep Learning Based Anomaly Detection

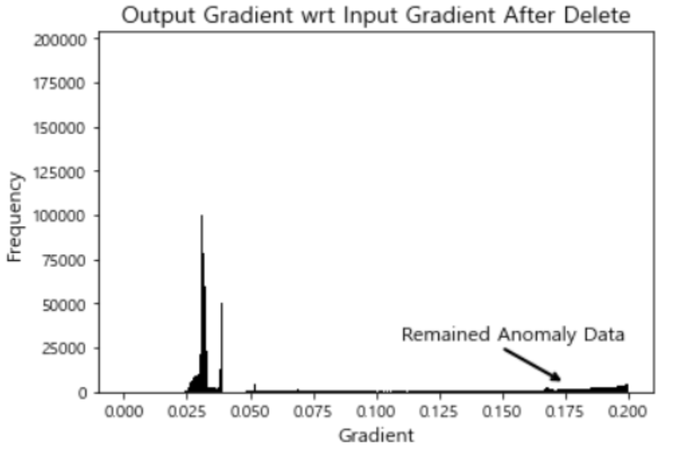
딥러닝 기반 학습 과정에서의 gradient 분포를 통한 이상치 탐색

Anomaly Detection - ODIM Algorithm

딥러닝 모델을 학습 할 때, 초기 학습 단계에서의 Gradient Vector를 통해 Outlier를 탐색하는 방법.
Loss Function을 최적화하는 초기 단계에서, 흔히 등장하는 데이터들이 Deep Learning Model의 parameter 변화에 영향을 많이 줌.
Outlier와 같은 데이터들은 이러한 학습 과정에서 parameter 변화에 기여하는 정도가 적고, 결과적으로, 학습의 초기 단계에서 모델의 output에 관한 Input의 gradient가 큰 경향을 가짐. (해당 방법을 제안한 논문에서는 이 현상을 Inlier-Memorization effect로 정의.)

=> 간단한 딥러닝 모형으로부터 초기 학습과정에서 Output에 대한 Input의 gradient를 이용하여 Anomaly Detection을 수행.
이때, Gradient Vector의 Norm을 이용해서 gradient 분포가 다른 임계점을 적절히 잡아 Abnormal Data로 분리.





모델링 방법

언더샘플링 기반의 앙상블 모델 사용

언더샘플링이 음성 샘플을 전부 사용하지 못한다는 점을 보완하기 위해 모델 10개 훈련

독립적인 모델 사용

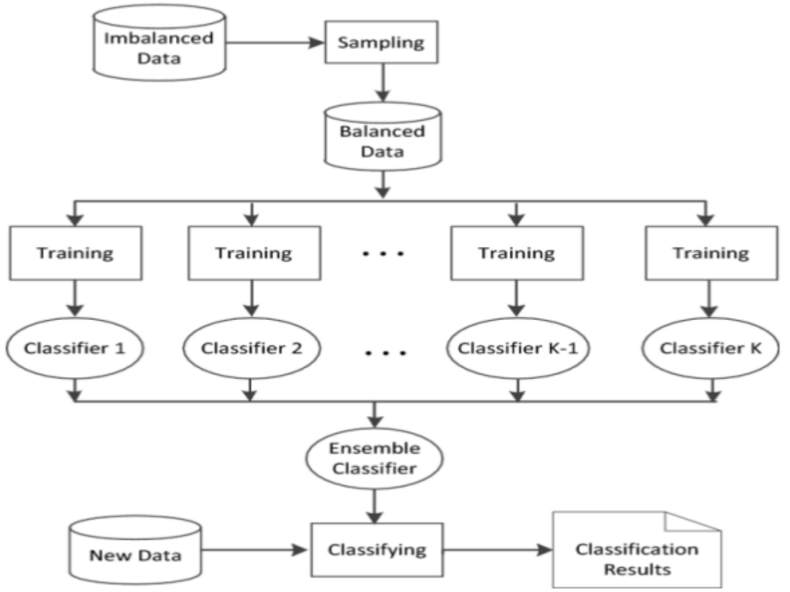
Anomaly Detection을 통해 전체 Input Data를 두 가지 그룹으로 나눔.
서로 다른 Data Distribution을 하나의 모형으로 설명하는 것은 제한이 있다고 판단하여,
두 가지 모델로 분리하여 개별 학습을 진행.

=> 각 그룹의 분포로부터 최적의 결정 경계를 찾기 위해 개별 모델을 사용함으로 성능 향상.

언더샘플링 기반의 앙상블 모델

Target 불균형을 해소하기위해, UnderSampling을 사용하여, 1 : 1 비율로 모델을 학습시킴.
이때, Target이 0인 데이터가 주는 정보의 손실을 최소화하고자, 랜덤 비복원 추출을 이용.
10개의 모델을 서로 다른 데이터들로 사용하게 하면서, 모델의 Robustness를 향상.

=> Target 불균형을 해소하여, 모델이 일반화된 성능을 가지도록 설계.



모델링 후처리

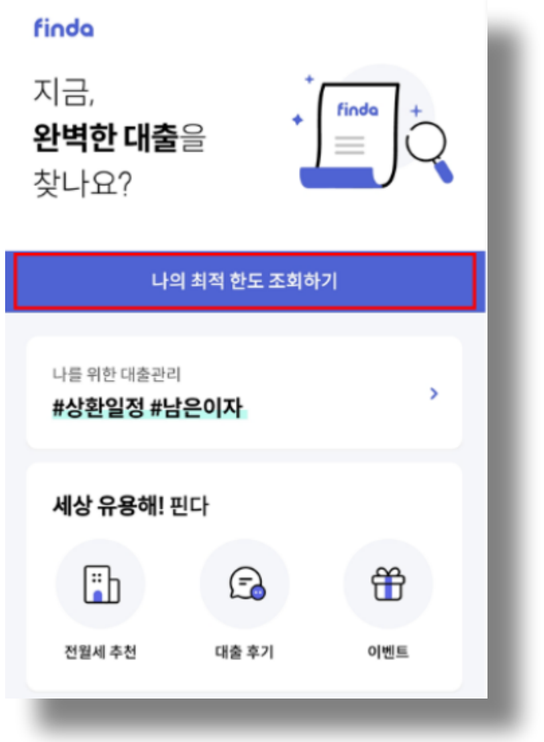
모델 예측값 진단 결과

모델링 후 예측값의 진단을 통해 모델 개선 가능성을 발견함

모델 예측값 진단

일반적으로 하나의 application_id에서 1~2개의 상품을 신청하는 경향성을 가짐.
반면, 모델은 하나의 application_id에서 훨씬 많은 상품을 신청할 것이라고 예측함.

application_id	product_id	is_applied	predict
1146852	26	1	1
1146852	27	0	1
1146852	28	0	1
1146852	29	0	0
1146852	30	0	0



모델의 recall을 중요시 하다 보니 양성클래스로 많이 분류하는 문제의 해결

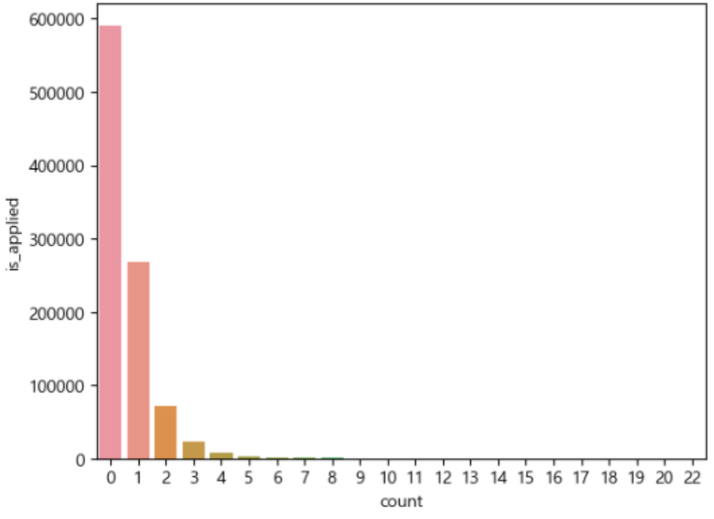
모델링 후처리

모델 예측 진단 결과

application_id 별 신청횟수에 대한 분석

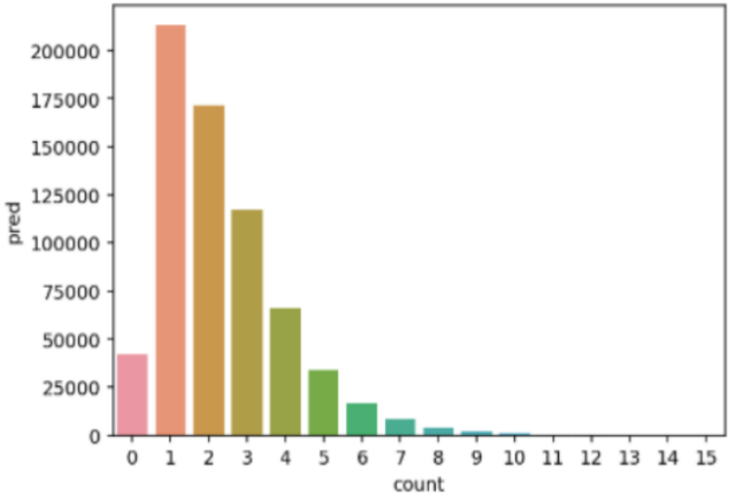
application_id 별 target의 분포

Train Distribution



group	count
0	591221
1	267622
2	71144
3	23310
4	8721
5	3593
6	1590
7	717
8	387
9	168

Predict Distribution



group	count
0	41848
1	213064
2	171592
3	116881
4	66219
5	33812
6	16821
7	8347
8	3884
9	1728
10	732

약 12.5배

훈련데이터의 분포는 5개 이상이 전체 데이터의 0.7%도 안되는데 모델은 약 9%로 예측

모델링 후처리

모델 예측 진단 결과

application_id 별 신청횟수에 대한 조건 부여

예측 오류 해소 방법

훈련데이터의 분포를 기준으로 5개 이상 신청한 그룹은 4개까지만 신청할 수 있도록 조치를 취함.
모델의 예측을 [1 , 0] 이 아닌 확률로 출력하여 각 application_id 그룹별로 높은 확률 3개를 선택.

application_id	product_id	is_applied	existing_pred	proba	new_pred
1146852	26	1	1	0.95	1
1146852	27	1	1	0.90	1
1146852	28	0	1	0.75	1
1146852	29	0	1	0.6	0
1146852	30	0	1	0.7	1

1개의 오분류 감소 효과 !!

모델의 recall을 최대한 유지하면서 precision을 높이는 효과

Chap 4

Conclusion

최종 모델 평가

후보 모델과의 비교

대출 신청 예측 모델의 경우 양성 샘플을 음성이라고 예측하는 오류가 치명적임에 유의

최종 모델 평가

모델의 제 1목적에 집중하여 recall 수치의 하한선을 0.9로 설정

단순히 F1_score가 높거나, accuracy가 높은 모델 보다는 모형의 경제적 효용성에 초점을 두어 판단함.

또한, 현재 모형을 2개 사용하는 구조 이므로, 두 모델의 혼동행렬의 합을 이용하여 모형을 평가하였다.

	Accuracy	Recall	Precision	F1_score
DT	0.8315	0.8456	0.222	0.3517
LGBM	0.8311	0.8464	0.2232	0.3532
XGBM	**0.8340**	**0.8470**	0.2249	0.3554

튜닝 및 후처리

성능지표	XGBM
Accuracy	0.8493
Recall	0.8508
Precision	0.2438
F1_score	0.3790

모든 평가지표를 종합적으로 고려하였을 때 XGBM을 선택함

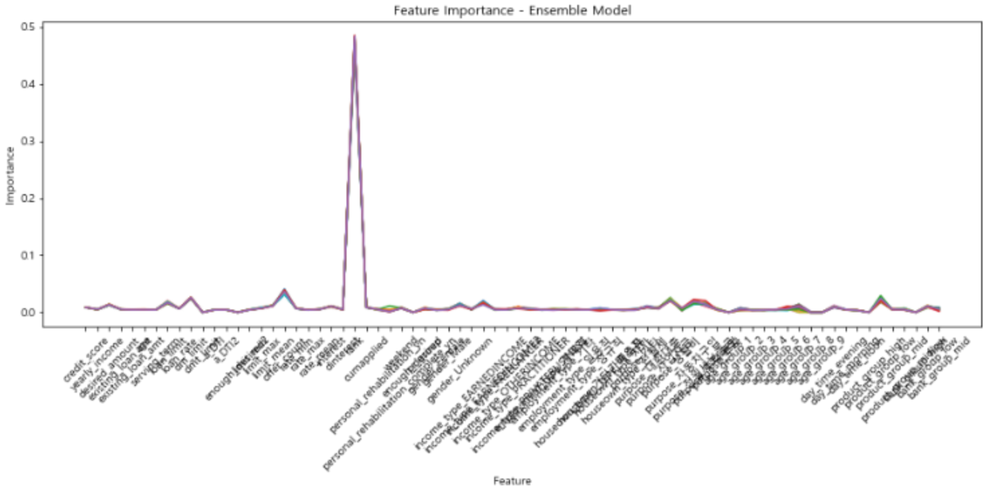
모델링 해석

어떤 변수가 개인의 대출 신청 여부에 영향을 끼칠까?

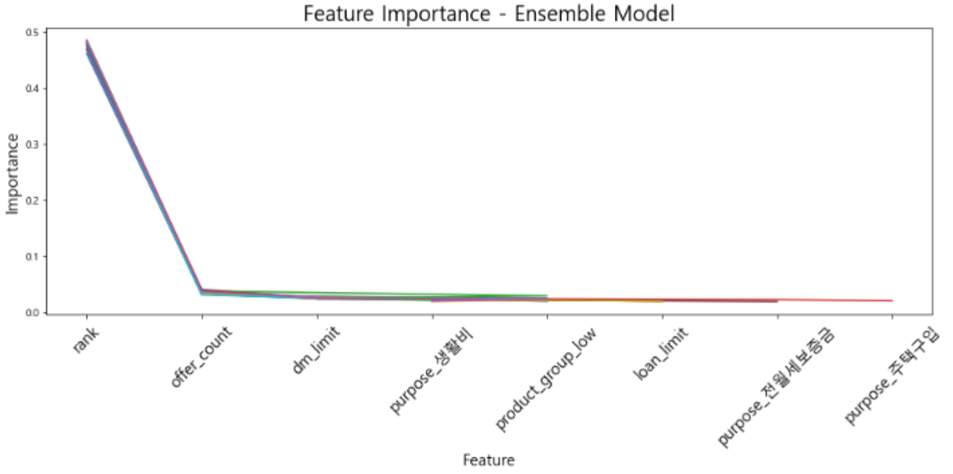
Feature Importance, PDP , ICE_Plot 중심의 해석

적합 모델 해석

모델이 타겟 정보를 예측하는데 기여한 상위 3개의 Feauture들을 중점으로 해석.
주어진 Data에서 논리적 결함이 있는 데이터, 많은 누락값, Abnormal한 데이터들이 많아 평균적인 영향력에 중점.



전체 변수의 Feature Importance



상위 7개 변수의 Feature Importance

상위 3개 변수를 중점으로 각 요인들이 모델에 어떻게 영향을 끼치는지 해석

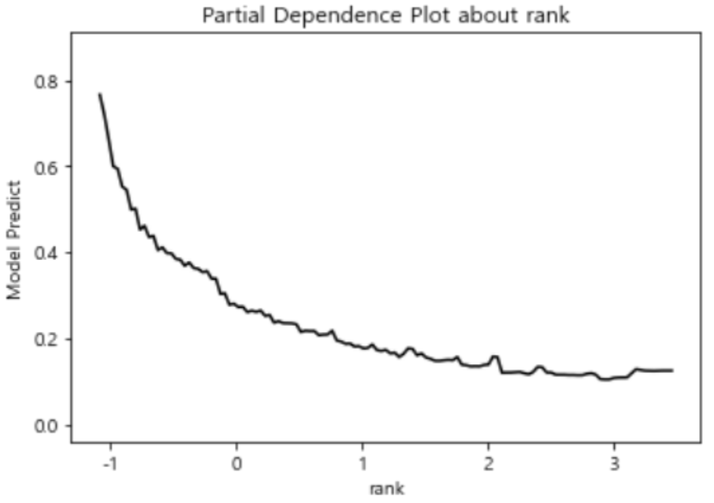
모델링 해석

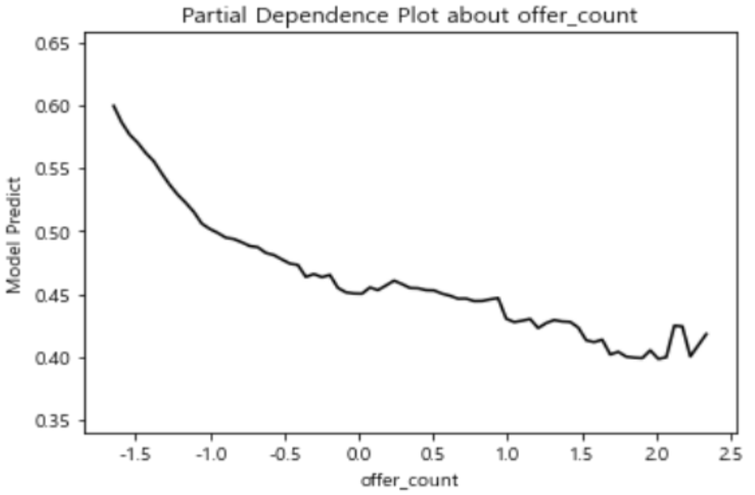
어떤 변수가 개인의 대출 신청 여부에 영향을 끼칠까?

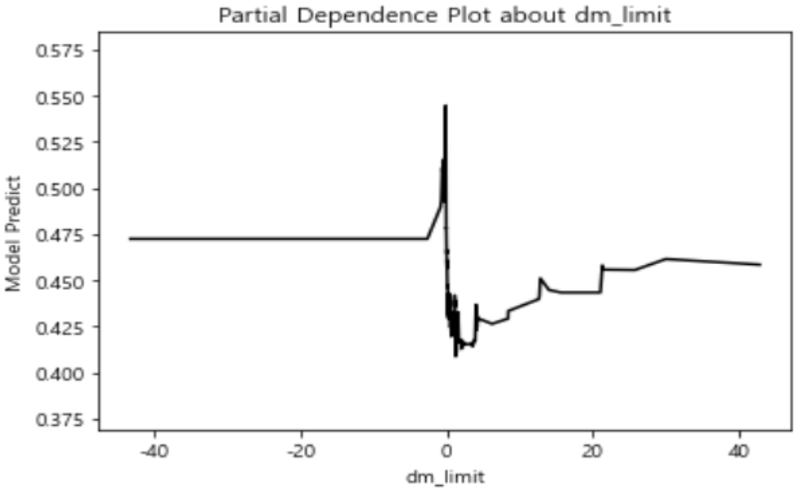
Feature Importance, PDP , ICE_Plot 중심의 해석

Partial Dependence Plot

특정 개인이 여러 개의 상품을 추천 받는다면, 평균적으로 가장 대출 금리가 낮은 것을 선택할 경향이 있음.
특정 개인이 추천 받은 상품의 개수가 적으면 적을 수록, 평균적으로 신청 확률이 높음.
추천받은 대출 상품의 승인 한도가 특정 개인의 대출 희망 금액보다 비슷하거나 조금 클 경우 평균적으로 신청 확률이 높음.





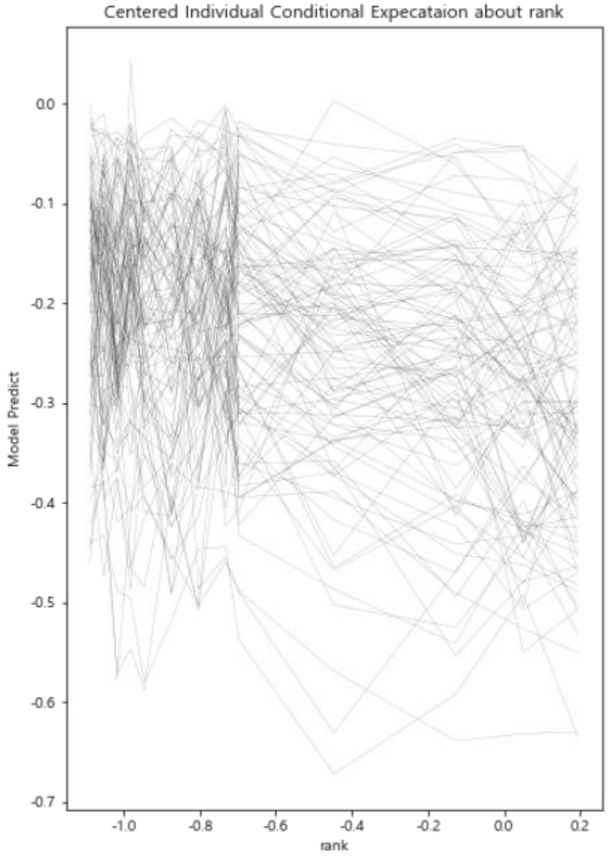


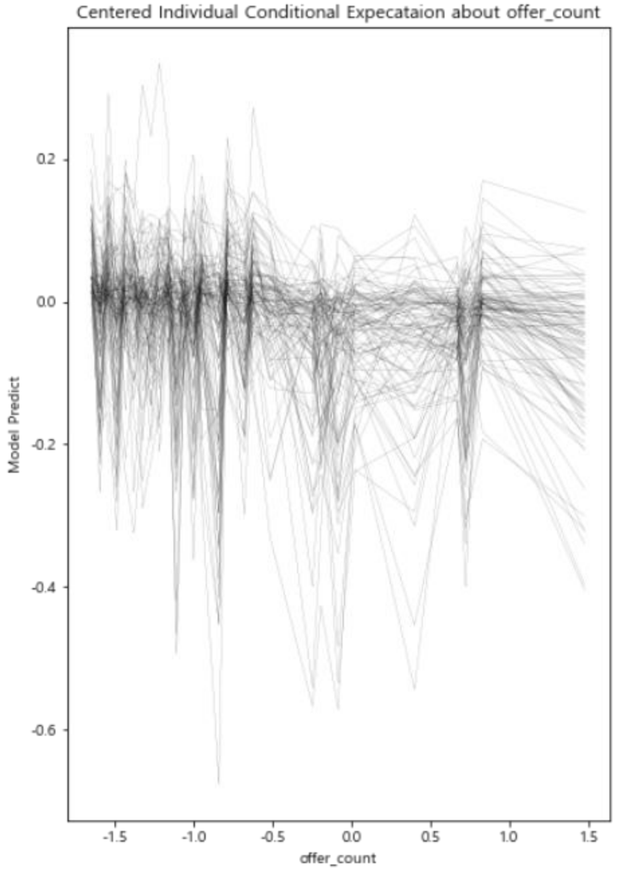
모델링 해석

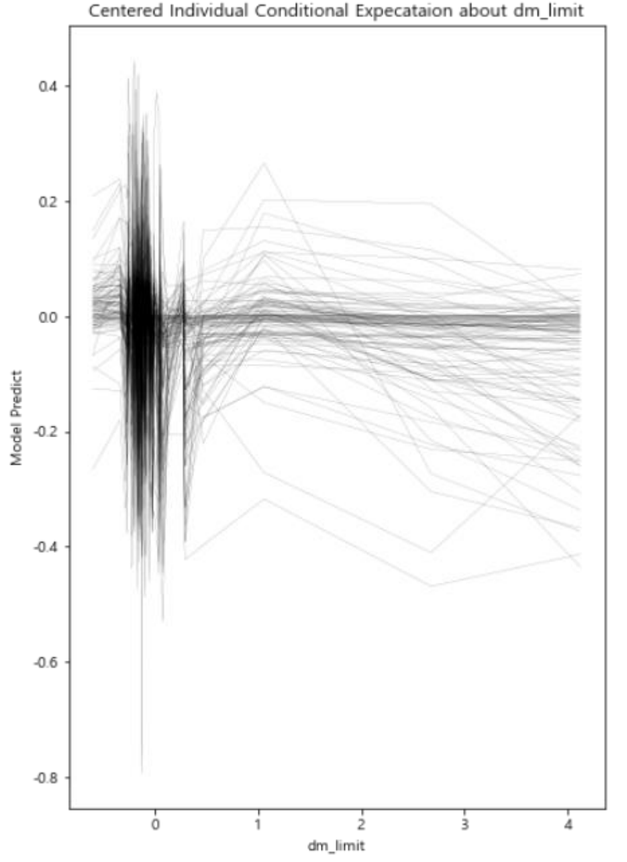
개별 Instance 마다의 상위 Feature의 영향력 분석

어느 정도 경향성을 띄면서도, 데이터의 Noise가 반영.

Individual Conditional Expectation Plot







활용 방안

1. 고객 만족도 증가

모형 기반 대출 상품 추천 시스템으로 고객에게 더욱 적절한 상품을 추천해주어 고객 만족도를 높일 수 있음.

=> 고객 이탈 최소화 효과

2. 대출 상품 개발

신청 예측 모델의 해석을 통해 어떤 요인이 사용자의 대출 신청에 영향을 주는지 분석

=> 이를 바탕으로 대출 상품 개발

3. 대출 상품 판매 이익 증가

모델을 통해 신청 확률이 높은 상품을 추천해주어 더 많은 상품 판매 이익을 기대할 수 있음.

소비자, 판매자, 연구자 모두에게 긍정적인 효과를 기대할 수 있다.

의의

1. 경제적 효용성 고려

recall값에 높은 하한값을 두어 양성 샘플을 음성이라고 예측하는 오류를 줄이고,
기업입장에서의 실질적인 이익을 고려하였음.

2. 모형 진단을 통한 개선

모델의 훈련, 예측, 평가에 그치지 않고, 예측값의 진단을 통해 모형을 개선하였음.

3. 새로운 방법론 제시

딥러닝 알고리즘을 도입하여 데이터를 분리하고 이를 통해 두가지의 모형을 각각 학습시켜 성능을 개선하였음.

참고 자료

1. Interpretable Machine Learning - A Guide for Making Black Box Models Interpretable, Christoph Molnar
2. ODIM: a method to identify inliers via inlier-memorization effect of deep generative models, (Dongha Kim, Proceedings of the Korean Data Analysis Society July 7-8, 2022)
3. XGBoost: A scalable Tree Boosting System
4. https://neptune.ai/blog/how-to-deal-with-imbalanced-classification-and-regression-data