

Y냐 돈

손진원 김상현 김우정 노혜미



Member



응용통계학과 14학번
손진원



응용통계학과 13학번
김상현



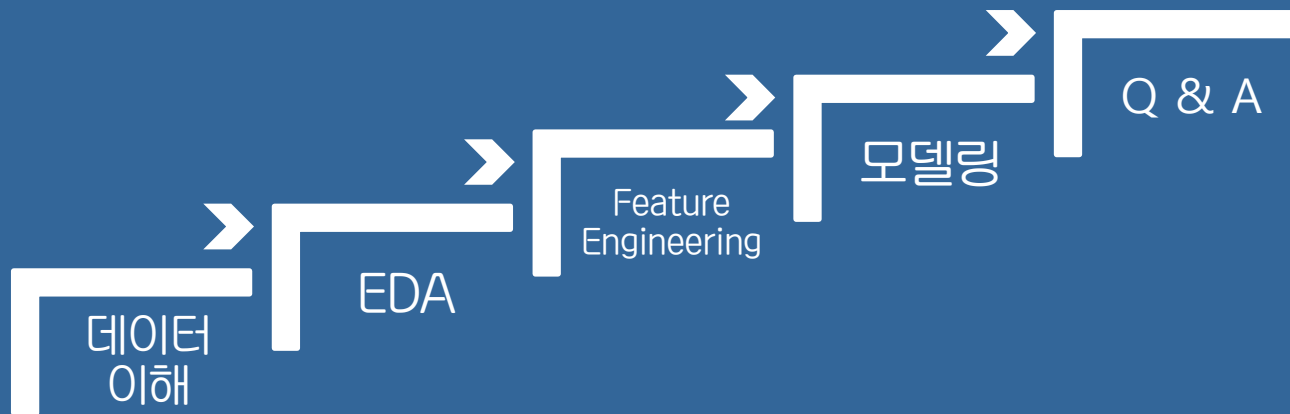
국어국문학과 16학번
노혜미



응용통계학과 14학번
김우정

소속 : 연세대 데이터분석학회 YBIGTA

Contents



데이터 이해

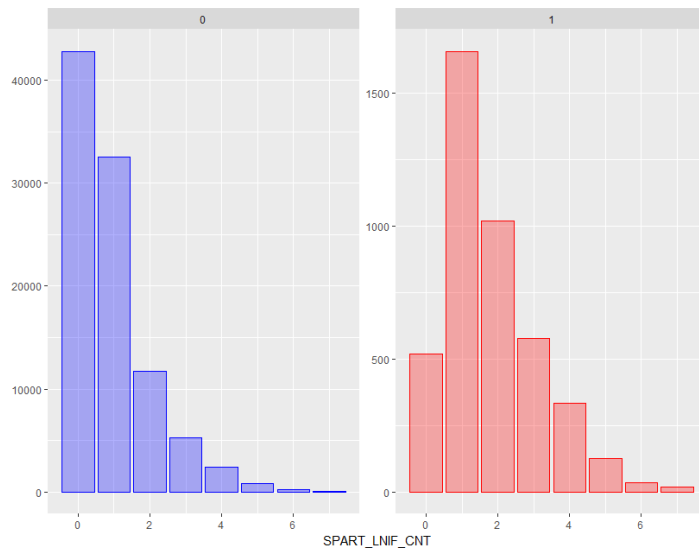
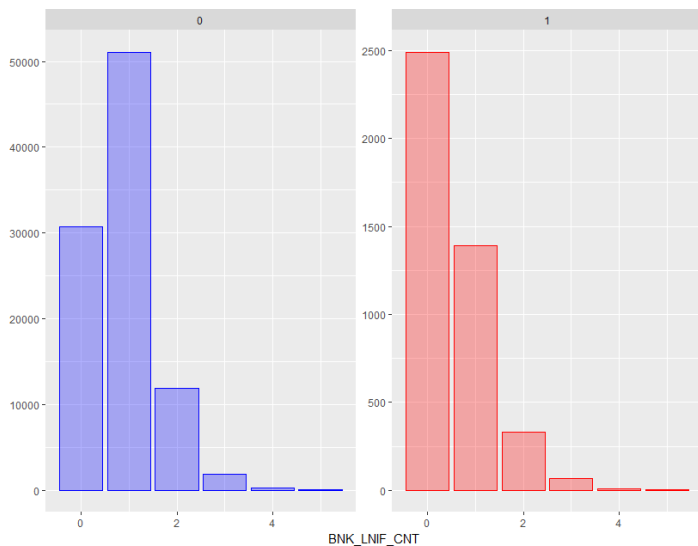
1. 매우 심한 TARGET 불균형

- 불균형 문제를 해결하기 위한 방안 필요
- TEST 데이터에 TARGET이 확률적으로 85명 내외

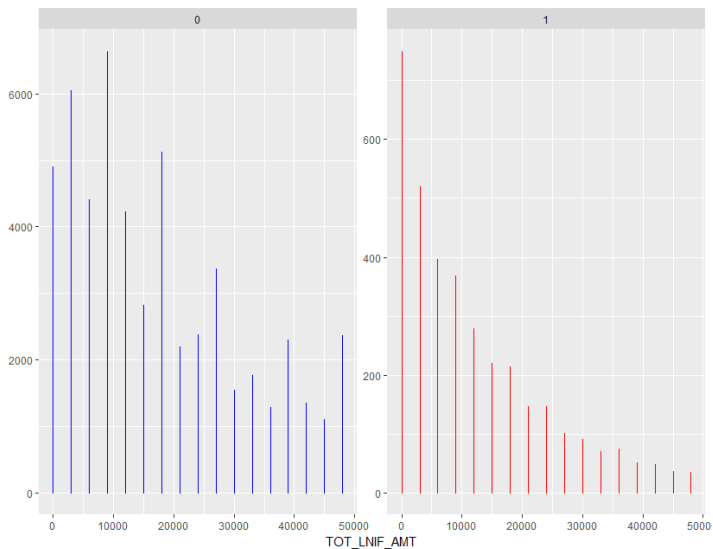
2. 비식별화로 인한 정보 손실

- * 처리로 인한 정보 손실
- 연속형 데이터의 범주화로 인한 정보 손실

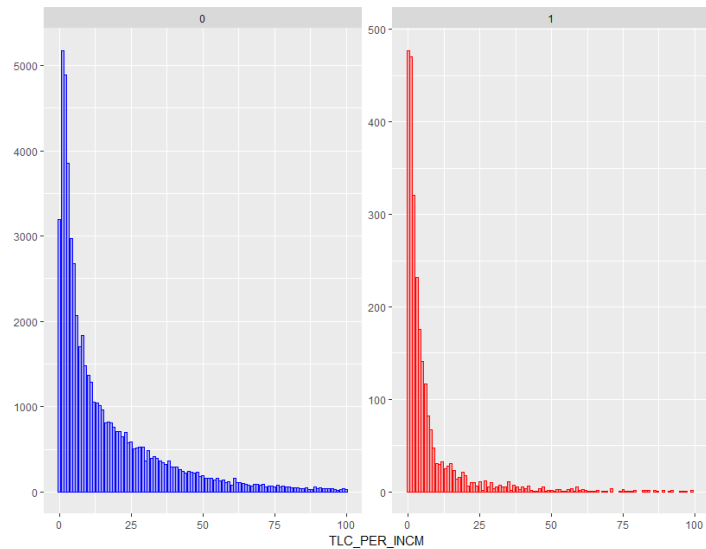
제2금융권 이하 대출 이력이 많을수록 TARGET일 확률이 높지 않을까??



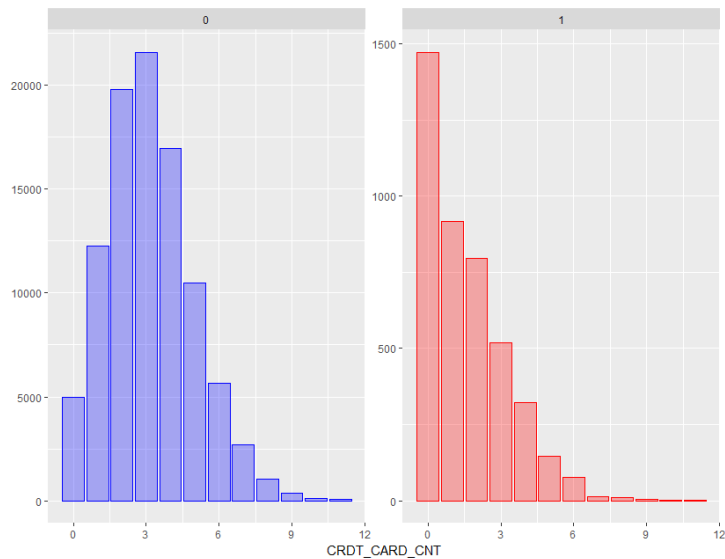
대출액이 많을수록 못 갚지 않을까?



소득에 비해서 많이 빌린다면?



신용카드가 0개면
TARGET 비율이 매우 높다!

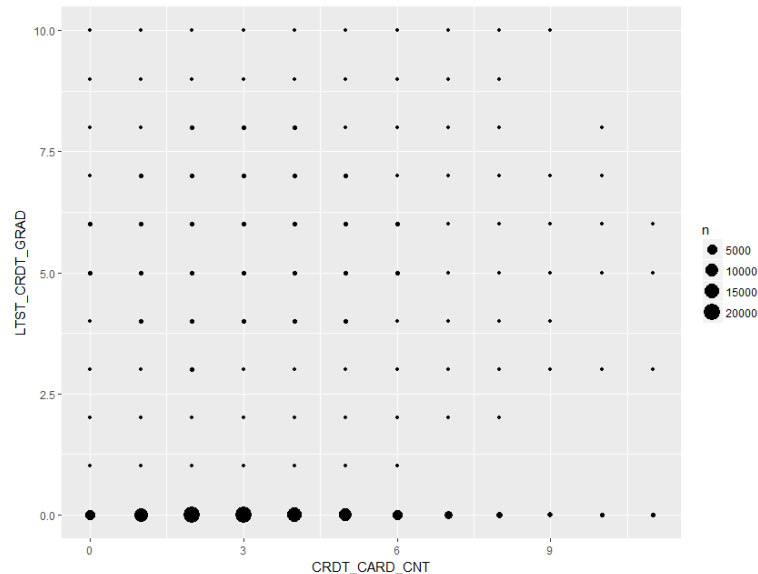
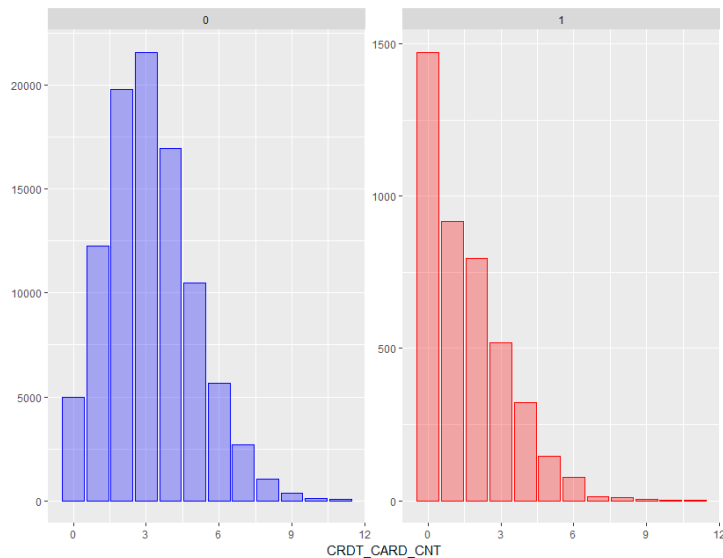


EDA

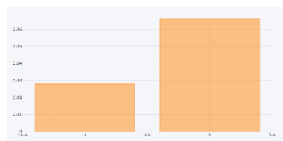
신용카드가 0개면
TARGET 비율이 매우 높다!



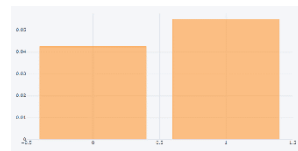
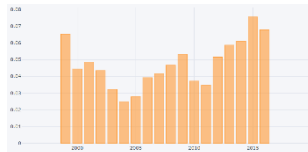
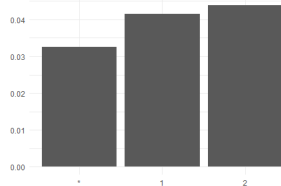
신용등급과 관련이 있는 것은 아닐까?



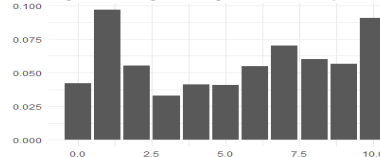
EDA



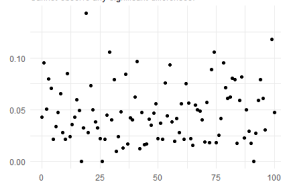
Proportion of Target by SEX
Cannot observe any significant differences.



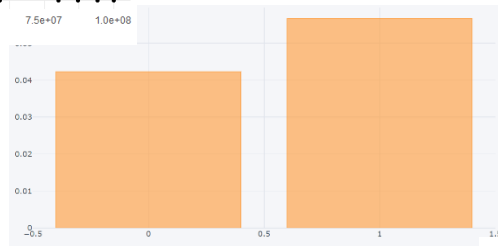
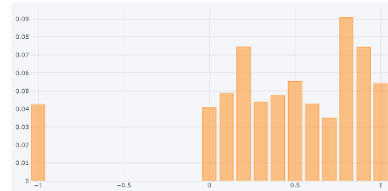
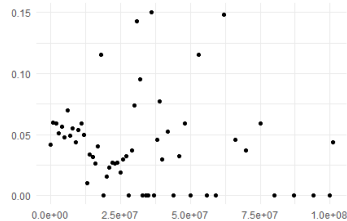
Bar-graph of LTST_CRDT_GRAD & Target
Target ratio look high on 1, 10 grade. But small sample.



Scatter plot of CRLN_OVDU_RATE & Target
Cannot observe any significant differences.

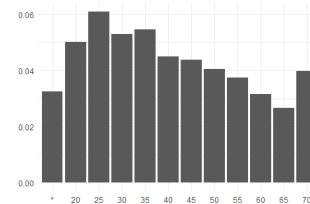


Scatter-plot of LT1Y_STLN_AMT & Target
No Pattern.



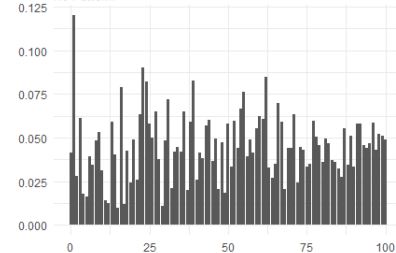
Proportion of Target by Age

Older, lower proportion. Why the target proportion jumps up at age 70?



Bar-plot of AVG_STLN_RATE & Target

No Pattern.



EDA



EDA를 통한

데이터 이해 및 유효 변수 예측 수행

Feature Engineering

1. 파생 변수 만들기

- $\text{SECOND_RATIO} : (\text{CPT_LNIF_CNT} + \text{SPART_LINF_CNT} + \text{ECT_LINF_CNT}) / \text{BNK_LNIF_CNT}$
- $\text{LNIF_AMT_PER_INCM} : \log_{10}(\text{TOT_LNIF_AMT} / \text{CUST_JOB_INCM} + 0.01)$
- $\text{AUTR_FAIL_MCNT_BINARY} : \text{자동이체 실패 월수가 0이면 1, 아니면 0}$
- $\text{CHNG_CRDT_GRAD} : \text{LTST_CRDT_GRAD} - \text{STRT_CRDT_GRAD}$
- $\text{AVG_ONE_CALL_TIME} : \text{AVG_CALL_TIME} / (\text{AVG_CALL_FREQ} + 1)$
- 약 90여 개의 파생 변수 생성

Feature Engineering

1. 파생 변수 만들기

- $\text{SECOND_RATIO} : (\text{CPT_LNIF_CNT} + \text{SPART_LINF_CNT} + \text{ECT_LINF_CNT}) / \text{BNK_LNIF_CNT}$
- $\text{LNIF_AMT_PER_INCM} : \log_{10}(\text{TOT_LNIF_AMT} / \text{CUST_JOB_INCM} + 0.01)$
- $\text{AUTR_FAIL_MCNT_BINARY}$: 자동이체 실패 월수가 0이면 1, 아니면 0
- $\text{CHNG_CRDT_GRAD} : \text{LTST_CRDT_GRAD} - \text{STRT_CRDT_GRAD}$
- $\text{AVG_ONE_CALL_TIME} : \text{AVG_CALL_TIME} / (\text{AVG_CALL_FREQ} + 1)$
- 약 90여 개의 파생 변수 생성


→ 그러나, 유의미한 F1 Score의 상승을 가져오지 못함

Feature Engineering

2. 외부 데이터 이용

- 경제성장률(1960 ~ 2017 2분기)
- 가계신용 전년동기증감율(%) (1997 ~ 2017 1분기)
- 가계대출 전년동기증감율(%) (1997 ~ 2017 1분기)

- 'TEL_CNTT_QTR' SKT 가입년월_분기
- 'MIN_CNTT_DATE' 최초대출날짜



두 종류를 조합하여
6개의 새로운 변수 생성

Feature Engineering

2. 외부 데이터 이용

- 경제성장률(1960 ~ 2017 2분기)
- 가계신용 전년동기증감율(%) (1997 ~ 2017 1분기)
- 가계대출 전년동기증감율(%) (1997 ~ 2017 1분기)
- 'TEL_CNTT_QTR' SKT 가입년월_분기
- 'MIN_CNTT_DATE' 최초대출날짜



두 종류를 조합하여
6개의 새로운 변수 생성

→ 그러나, 유의미한 F1 Score의 상승을 가져오지 못함
(Data_set의 날짜 변수는 실제 데이터의 관측 시점을 표현해주지 못함)

Feature Engineering

3. Outliers 처리

- **Isolation Forest** : Target을 제외한 나머지 데이터들로 비지도학습 수행
- Outlier에서의 Target 비율이 Inlier에서의 Target 비율보다 약 3.5배 높음
- 새로운 'OUTLIERS' 컬럼 추가

4. Missing-Value 처리

- **Miss Forest** : 결측값을 Target으로 생각하여 RF를 이용한 예측 수행
- OOB Error rate : $PFC = 0.2729$

Feature Engineering

5. 변수 선택

연속형 변수 : 금액, 시간, 비율 등 37개 / 범주형 변수 : 직업, 회선 상태 등 30개

- 1) 변수를 그대로 사용한 F1 Score와 더미화 한 F1 Score를 비교하여 더미화 여부 결정
- 2) 변수들에 대하여 피어슨 카이제곱 독립성 검정을 시행하여 P-Value ≤ 0.05 인 것을 선정

→ 더미화 시킨 180여개의 변수에서 최종적으로 87개 선택

→ 180개와 87개의 예측력이 큰 차이 없음 : 87개로 최종 확정!

Modeling



Original
TRAIN

Modeling

5 Cross Validation

Original
TRAIN

CV-TEST

Modeling

5 Cross Validation

Original
TRAIN

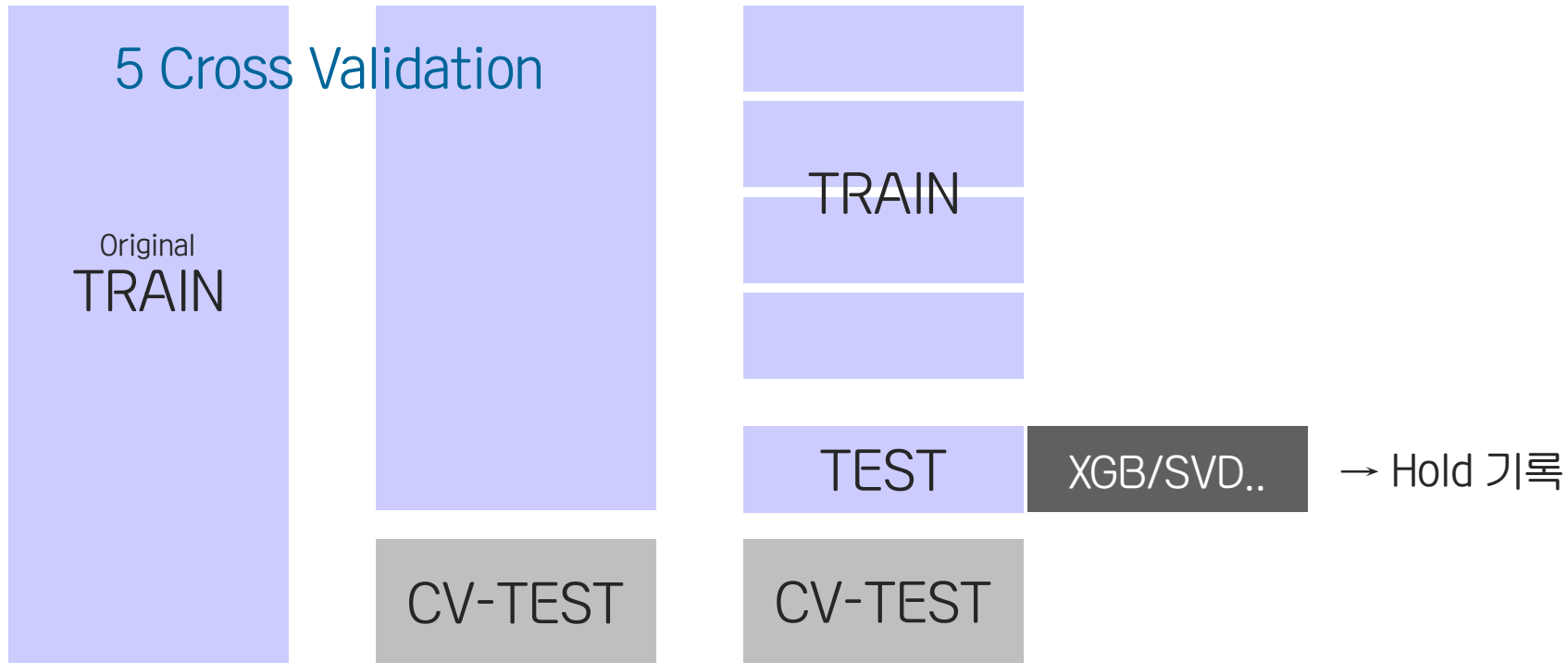
CV-TEST

TRAIN

TEST

CV-TEST

Modeling



Modeling

5 Cross Validation

Original
TRAIN

CV-TEST

TRAIN

TEST

CV-TEST

XGB/SVD..

XGB/SVD..

XGB/SVD..

XGB/SVD..

XGB/SVD..

→ Hold 기록

→ Hold 기록

→ Hold 기록

→ Hold 기록

→ Hold 기록

Modeling

5 Cross Validation

Original
TRAIN

CV-TEST

TRAIN

TEST

CV-TEST

XGB/SVD..

XGB/SVD..

XGB/SVD..

XGB/SVD..

XGB/SVD..

XGB/SVD..

→ Hold 기록

→ Hold 기록

→ Hold 기록

→ Hold 기록

→ Hold 기록

← Hold 평균

Modeling

	XGB/SVD..	RF/LOGIT..
	XGB/SVD..	RF/LOGIT..
ORIGINAL	XGB/SVD..	RF/LOGIT..
	XGB/SVD..	RF/LOGIT..
	XGB/SVD..	RF/LOGIT..

CV-TEST	XGB/SVD..	RF/LOGIT..
---------	-----------	------------

TRAIN SET



최적의 F1 Score 및
Threshold 기록

Modeling

No Stack

first_res

	Hold	F1_measure	Recall	Precision
gb	0.2075	0.478144	0.536601	0.431776
xgb1	0.2150	0.477381	0.530763	0.434366
ext	0.2000	0.456745	0.515316	0.410980
rf	0.2250	0.453018	0.497506	0.420489
bg	0.2650	0.438292	0.486450	0.401326
svc	0.0925	0.410741	0.430143	0.394721
logit	0.1900	0.400656	0.449404	0.366463
ada	0.5000	0.349256	0.254015	0.559281
mlp	0.4225	0.341391	0.355222	0.338845

1계층 Stack

second_res

	Hold	F1_measure	Recall	Precision
ext	0.2575	0.486035	0.535147	0.446361
rf	0.2725	0.479415	0.514733	0.452368
logit	0.2200	0.479217	0.517039	0.451536
xgb2	0.2275	0.474498	0.539533	0.426416
gb	0.2025	0.468066	0.549434	0.410192
svc	0.0400	0.465693	0.443569	0.490541

Modeling

2계층 Stack

```
stm.get_cv_result(CV_test_result)
```

	F1_measure	Hold	Precision	Recall
xgb4	0.491326	0.234286	0.454725	0.545621
logit	0.489871	0.267143	0.462952	0.523441
rf	0.484878	0.258571	0.450372	0.529522

```
stm.get_cv_result(not_meta_result)
```

	F1_measure	Hold	Precision	Recall
xgb4	0.474432	0.200000	0.423026	0.541160
rf	0.453242	0.225714	0.420524	0.496143
logit	0.402536	0.165714	0.342611	0.500599

5 Cross Validation의 결과,

- 2계층 Stack 모델의 결과가 더 좋음
- 대체로 Recall > Precision
- 그럼 이것이 최적의 결과인가?

Modeling

5 Cross Validation은 실제 TEST 예측에 적절하지 않음...

5 CV TRAIN : 80000명 중 연체자 3200명
 TEST : 20000명 중 연체자 800명

REAL TEST : 2000명 중 연체자 80명

Modeling

실제 TEST 환경과 유사한 49 CV를 해본다면?

```
count    49.000000
mean      0.501830
std       0.038331
min       0.406780
25%       0.475728
50%       0.506024
75%       0.527607
max       0.569767
Name: 1, dtype: float64
```

F1 Score : 0.40~0.56

```
count    49.000000
mean      0.257755
std       0.055949
min       0.160000
25%       0.210000
50%       0.250000
75%       0.290000
max       0.360000
Name: 0, dtype: float64
```

Threshold : 0.16~0.36

Threshold와 F1 Score의 변동이 매우 크다

Target Imbalance 문제를 해결하기 위한 시도

- Over Sampling
- Under Sampling
- SMOTE (범주형은 평균변환)
- Over and Under Sampling
- Data Ensembling

Modeling

Target Imbalance 문제를 해결하기 위한 시도

- Over Sampling
- Under Sampling
- SMOTE (범주형은 평균변환) → 효과가 없거나 오히려 예측력이 떨어짐
- Over and Under Sampling
- Data Ensembling

Modeling

실제 Test 데이터와 유사한 CV Test 데이터가 있을까?

→ 데이터 간의 유사도를 구해보자

Modeling

실제 Test 데이터와 유사한 CV Test 데이터가 있을까?

→ 데이터 간의 유사도를 구해보자

실제 Test $X=[2019 * 87]$

49 CV Test $X=[2047 * 87]$

Modeling

실제 Test 데이터와 유사한 CV Test 데이터가 있을까?

→ 데이터 간의 유사도를 구해보자

실제 Test

$$X = [2019 \ * \ 87]$$

$$X^T X = [87 \ * \ 87]$$

49 CV Test

$$X = [2047 \ * \ 87]$$

$$X^T X_1 = [87 \ * \ 87]$$

...

$$X^T X_{49} = [87 \ * \ 87]$$

Modeling

실제 Test 데이터와 유사한 CV Test 데이터가 있을까?

→ 데이터 간의 유사도를 구해보자

실제 Test $X=[2019 \ * \ 87]$

$$X^T X = [87 \ * \ 87]$$

49 CV Test $X=[2047 \ * \ 87]$

$$X^T X_1 = [87 \ * \ 87]$$

...

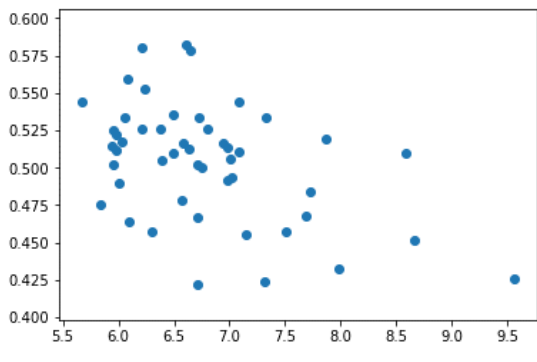
$$X^T X_{49} = [87 \ * \ 87]$$

Frobenius
Norm

Modeling

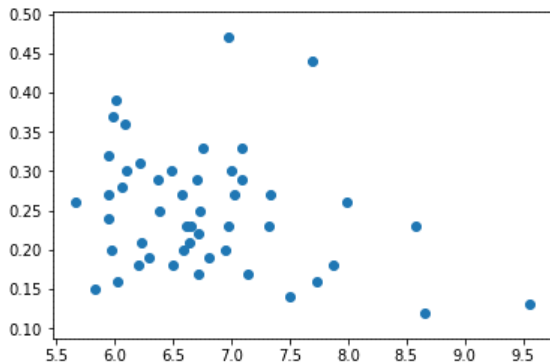
FB Norm ~ F1 Score

```
# 거리 vs 메저  
plt.scatter(xgb_res['norm'], xgb_res.iloc[:, 1])  
plt.show()
```



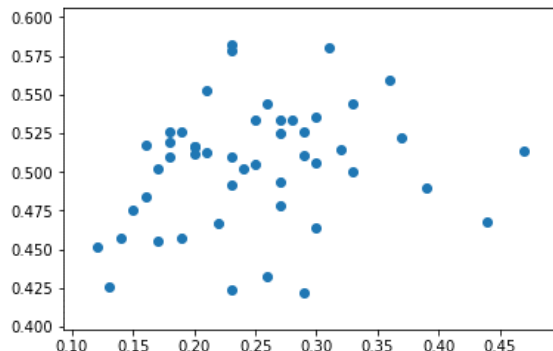
FB Norm ~ Threshold

```
# 거리 vs 출드  
plt.scatter(xgb_res['norm'], xgb_res.iloc[:, 0])  
plt.show()
```



F1 Score ~ Threshold

```
# 메저 vs 출드  
plt.scatter(xgb_res[0], xgb_res[1])  
plt.show()
```



- 거리가 가까울수록 F1 Score가 높음 → 실제 Test 데이터는 분류가 잘 되는 데이터인 편이다
- Threshold와 F1 Score가 약한 양의 상관관계가 있음 → 5CV보다 Threshold를 조금 높이자!

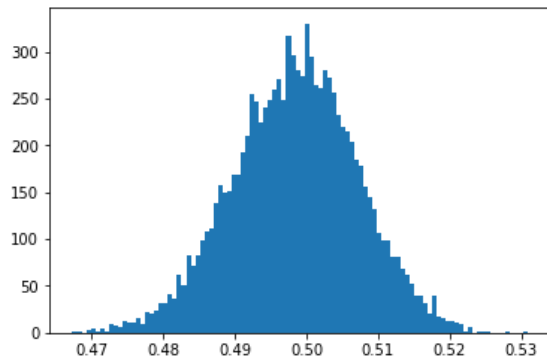
Modeling

FB Norm0이 6.5 미만인 49CV들로 최적의 Threshold 설정

- Threshold : 0.234 → 0.2605
- Target 수 / 데이터 수 : 109 / 2019 → 100 / 2019 (9명 차이)

F1 Score의 95% Bootstrap 신뢰구간

[0.481 , 0.515]

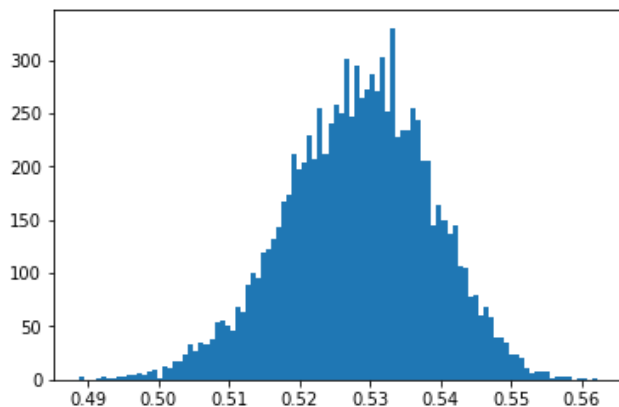


F1 bootstrap 95% CI
2.5 % : 0.481 , 97.5 % : 0.515

Modeling

Recall의 95% Bootstrap 신뢰구간

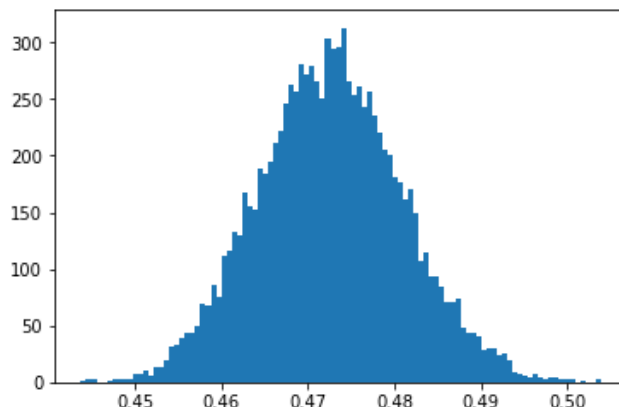
[0.507 , 0.547]



Recall bootstrap 95% CI
2.5 % : 0.507 , 97.5 % : 0.547

F1 Score의 95% Bootstrap 신뢰구간

[0.456 , 0.49]



Precision bootstrap 95% CI
2.5 % : 0.456 , 97.5 % : 0.49

About FN

		Actual Class	
		p	n
Predicted Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Totals:		P	N

FP : 연체자라고 판단하여 대출해주지 않았는데, 실제로는 연체자가 아닌 경우

FN : 연체하지 않을 거라고 판단하여 대출해주었는데 , 실제로는 연체한 경우

About FN

		Actual Class	
		p	n
Predicted Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Totals:		P	N

FP : 연체자라고 판단하여 대출해주지

않았는데, 실제로는 연체자가 아닌 경우



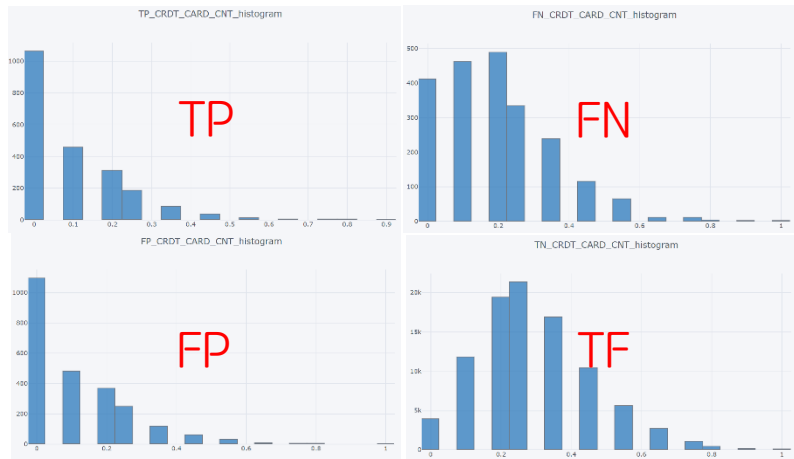
FN : 연체하지 않을 거라고 판단하여

대출해주었는데, 실제로는 연체한 경우

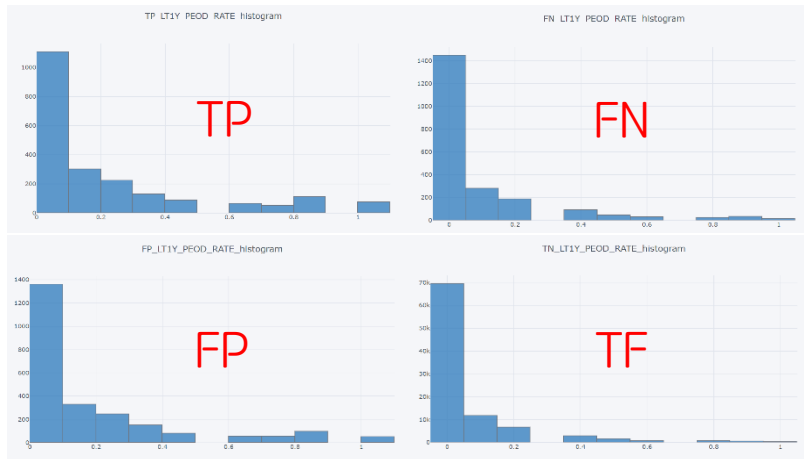
About FN

FN은 실수로 대출을 연체한 사람들이지 않을까??

CRDT_CARD_CNT



LT1Y_PEOB_RATE



About FN

FN을 낮추기 위한 제안

1. Threshold를 조정하여 Recall을 높인다
2. 더 유효한 변수를 만들어낸다
3. 더 많은 정보를 담고 있는 데이터를 사용한다

감사합니다!