주택 담보 대출 데이터 분석을 통한 대출 상환 여부 분류

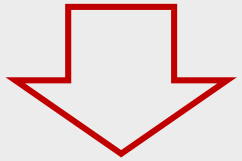
이제승, 진영봉

Seoul National University

FRE LAB

CONTENTS

- **Problem Definition**
  - 신용평가 모델은 대출자(금융기관)이 금융 채무 상환능력에 대한 모델을 바탕으로 대출 신청 수용 여부를 결정하는 데 도움 주는 모델
  - 많은 ML과 AI 기술들이 대출 수용 여부를 예측하는데 좋은 성능을 보이지만, Black-box 이기에 예측에 대한 근거를 설명하지 않고 대출자와 신청자로 하여금 결과에 대한 해석 제공하지 않음
  - 금융 전문가들이 정당성이 없는 모델의 예측을 신뢰하지 않을 가능성



- **Aims and Objectives**
  - ML 모델을 사용하여 **2년의 기간동안 90일이상의 연체기록**에 해당하는 위험을 분류하고, 원인을 분석하고자 함
  - XAI 기법을 통해 예측 결과를 해석할 수 있는 데이터 분석 실행
  - White-box vs Black-box 모델 비교

- **Home Equity Line Of Credit (HELOC)**

- 주택지분(Home Equity) = 현재시점 주택가격 – 주택담보대출잔액
- 위의 자본을 담보로 하여 한번에 대출금 전부를 받는 **고정금리***

  대출상품 –〉 Home Equity Loan
- 필요 시마다 한도 내에서 인출 받을 수 있는 **변동금리***

  대출상품 –〉 Home Equity Line Of Credit
- 자금 용도에 구애 받지 않으며 일반적으로 초기 10년까지 한도 내에서 신용카드처럼 이용

		2019년		2020년	
		잔액(십억 달러)	비중(%)	잔액(십억 달러)	비중(%)
주택 관련대출	HELOC	420	72.1	374	72.6
	Home equity	124		120	
	Mortgage	9,610		10,310	
기타 대출 (신용대출 등)	Auto loan&lease	1,300	27.9	1,350	27.4
	Personal loan	305		324	
	Student loan	1,400		1,570	
	Credit card	829		756	
	Retail credit card	90		80	
합계		14,078	100.0	14,884	100.0

19/20년 소비자 대출 잔액

• Data Descriptions (Feature Select한 컬럼)

: Data Set의 고객은 $5,000 – $15,000 범위의 HELOC 대출을 요청

–〉 그에 따라 금융기관은 신청자의 신용 보고서를 활용
–〉 2년 동안 90일이내 상환 (Good) / 90일이후 상환 (Bad) 분류

- RiskPerformance (Target): 대출 위험에 따라 “Good”/”Bad”로 분류
- ExternalRiskEstimate: FICO에서 부여한 스코어 값으로, 높을수록 대출위험이 적음
- NumSatisfactoryTrades: 만족스러운 거래 수
- PercentTradesNeverDelq: 연체하지 않는 거래 비율
- PercentInstallTrades: 할부 거래 비율
- NumInqLast6M: 6개월간 신용점수 조회 횟수
- MaxDelq2PublicRecLast12M: 12개월 동안의 최대 연체 기록
- NetFractionRevolvingBurden: 회전 잔액을 신용한도로 나눈 값
- NumRevolvingTradesWBalance: 잔액이 있는 회전 거래 수

```
RangeIndex: 10459 entries, 0 to 10458
Data columns (total 24 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   RiskPerformance                 10459 non-null  object
 1   ExternalRiskEstimate            10459 non-null  int64
 2   MSinceOldestTradeOpen           10459 non-null  int64
 3   MSinceMostRecentTradeOpen       10459 non-null  int64
 4   AverageMInFile                  10459 non-null  int64
 5   NumSatisfactoryTrades           10459 non-null  int64
 6   NumTrades60Ever2DerogPubRec     10459 non-null  int64
 7   NumTrades90Ever2DerogPubRec     10459 non-null  int64
 8   PercentTradesNeverDelq          10459 non-null  int64
 9   MSinceMostRecentDelq            10459 non-null  int64
 10  MaxDelq2PublicRecLast12M        10459 non-null  int64
 11  MaxDelqEver                     10459 non-null  int64
 12  NumTotalTrades                  10459 non-null  int64
 13  NumTradesOpeninLast12M          10459 non-null  int64
 14  PercentInstallTrades            10459 non-null  int64
 15  MSinceMostRecentInqexcl7days    10459 non-null  int64
 16  NumInqLast6M                    10459 non-null  int64
 17  NumInqLast6Mexcl7days           10459 non-null  int64
 18  NetFractionRevolvingBurden      10459 non-null  int64
 19  NetFractionInstallBurden        10459 non-null  int64
 20  NumRevolvingTradesWBalance      10459 non-null  int64
 21  NumInstallTradesWBalance        10459 non-null  int64
 22  NumBank2NatlTradesWHighUtilization  10459 non-null  int64
 23  PercentTradesWBalance           10459 non-null  int64
```

데이터 컬럼 정보

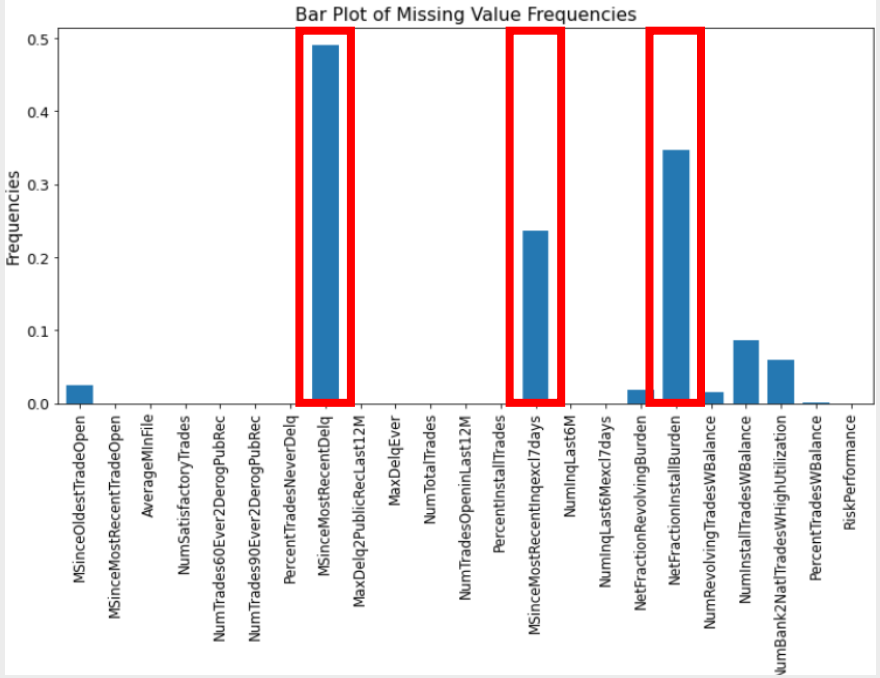
- **EDA**

  * special values -9: No Bureau Record or No Investiagtion
    -8: No Usable/Valid Accounts Trade of Inquiries
    -7: Condition not Met (e.g. No Inquiries, No Delinquencies)

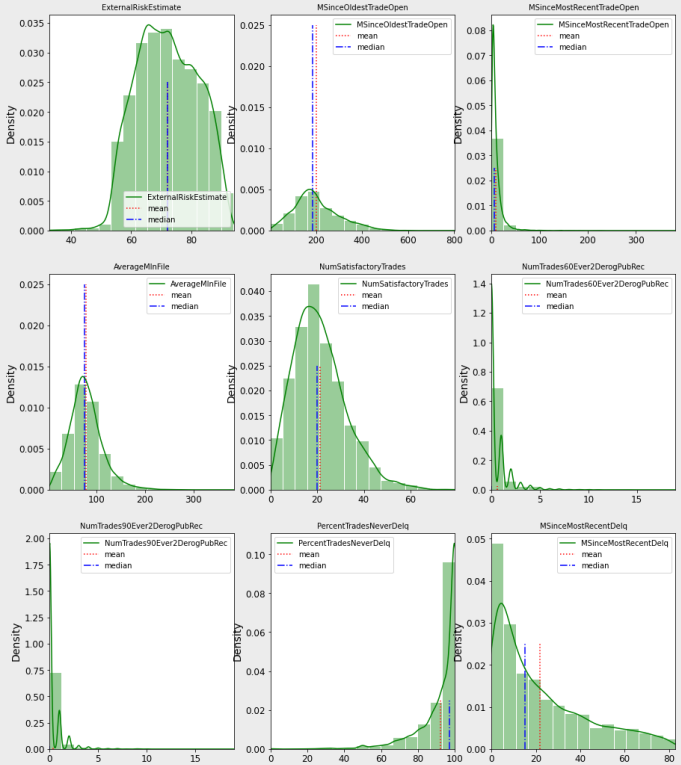
  => 모두 Missing value로 처리 (Nan)

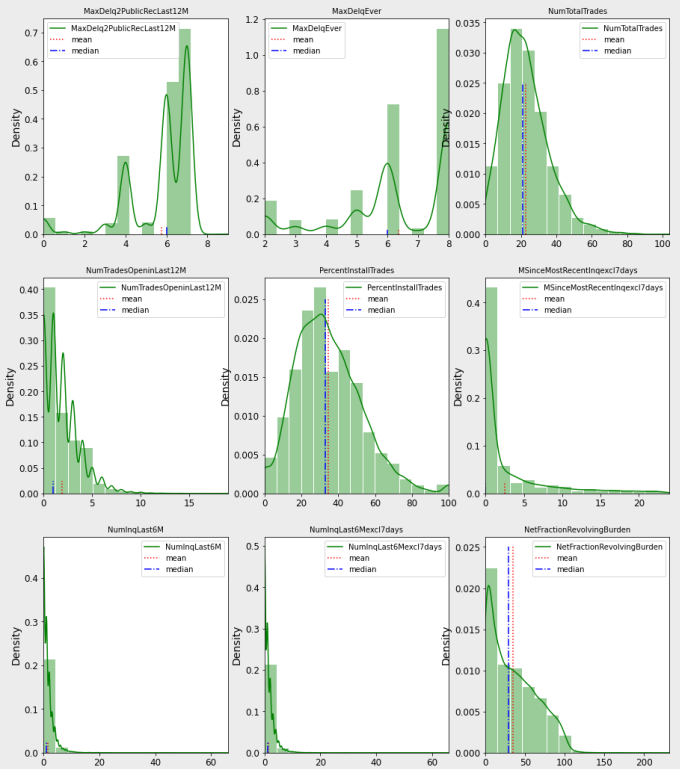
  - special values를 missing value로 처리한 후, 그 비율을 살펴 봄
  - ("MSinceMostRecentDelq", "MSinceMostRecentInqexcl7days", "NetFractionInstallBurden")에서 각각 (49%, 23.6%, 34.6%) 의 missing value비율을 보임
  - 총 8개의 컬럼에서 missing value 존재

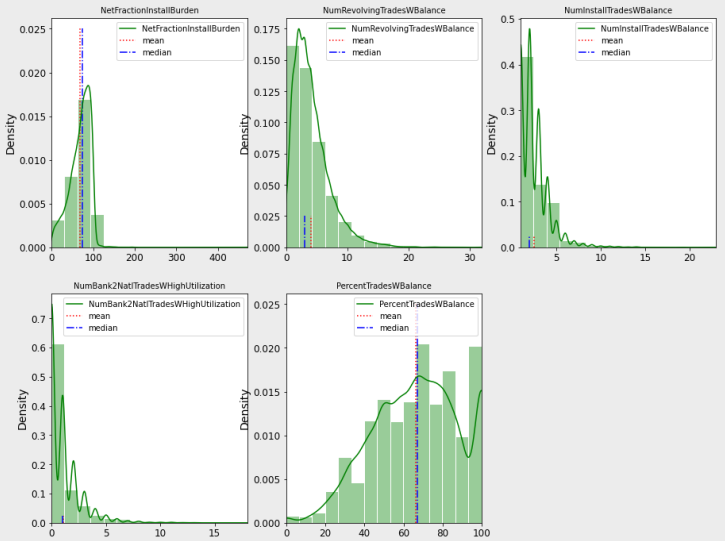
  -> 이외의 컬럼들에 대해서는 <u>분포를 확인 후 imputation 실행</u>



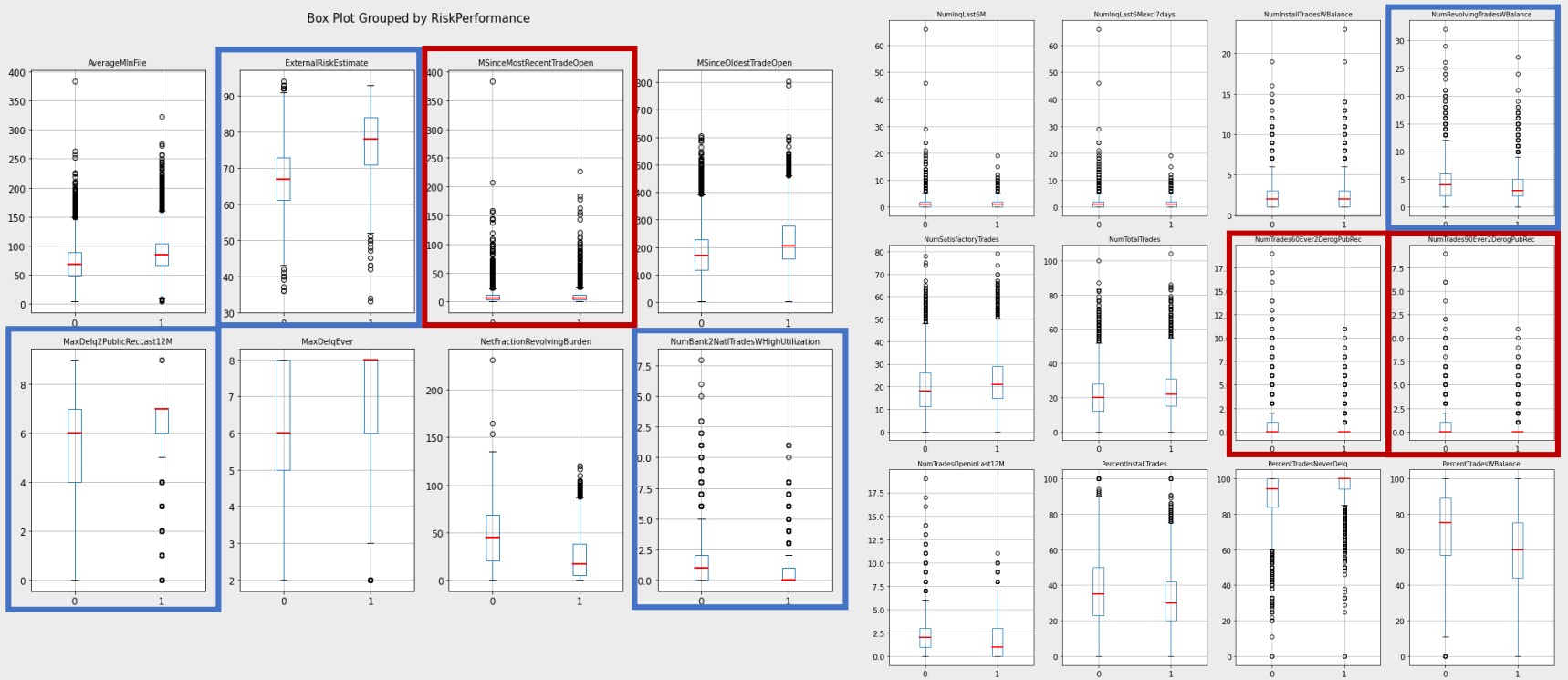
- **데이터 EDA – 각 feature의 분포 확인 및 imputation**
  - Missing value가 과도하게 높게 되면 편향되거나 잘못된 결과를 유도할 가능성이 높음 -〉 매우 큰 비율을 갖는 칼럼 제거
  - Missing value가 존재하는 컬럼들의 y값에 따른 데이터 분포 확인하고 다른 기준을 이용하여 data cleaning, imputation 수행
  - 일부 features에 대해 skewness가 큰 경우가 있음 -〉 이러한 경우 imputation을 평균이 아닌 중앙값을 사용하고자 함





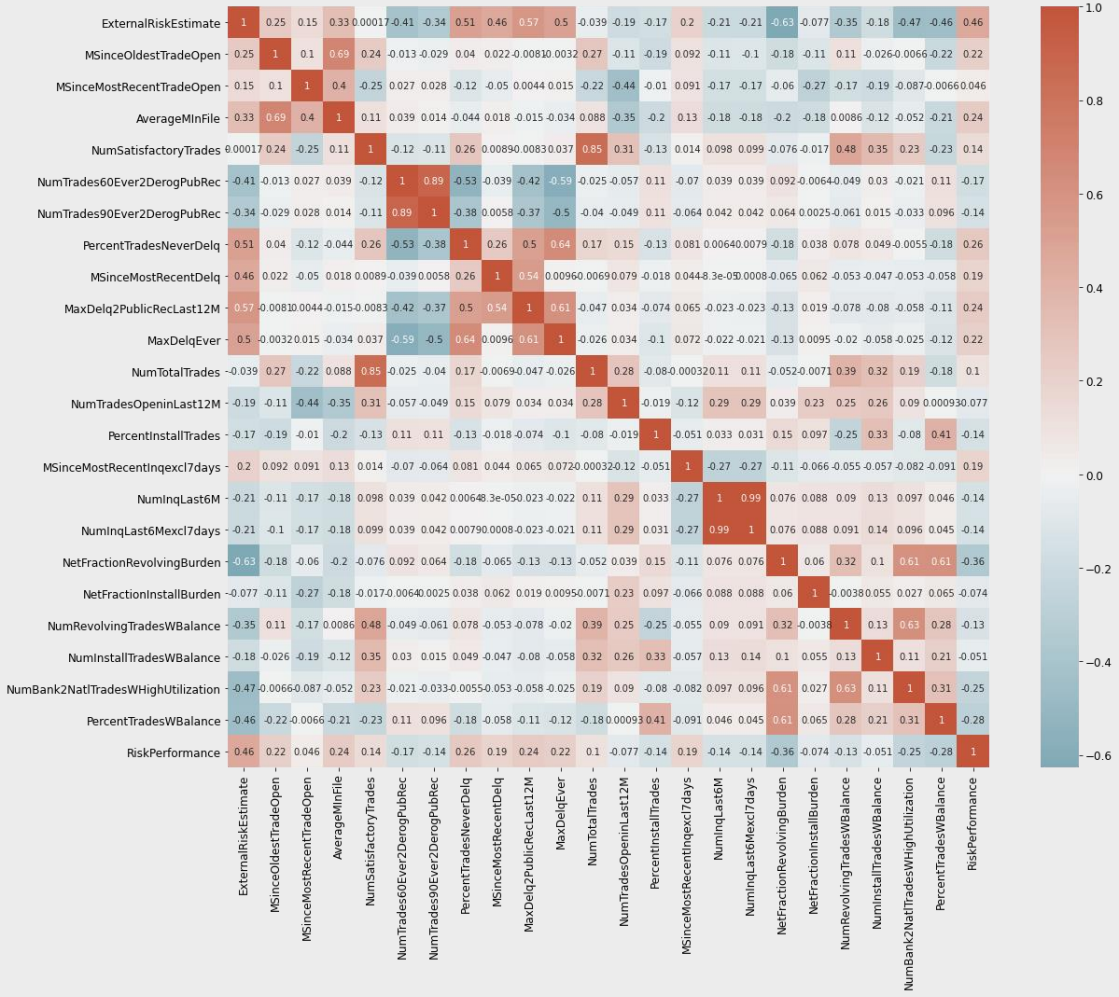


- **데이터 EDA – 각 label에 따른 데이터 분포 확인**
  - 우리의 target label에 따른 컬럼별 boxpolt으로 데이터 분포 및 이상치를 탐색함
  - “ExternalRiskEstimate”와 같은 경우 label에 따른 분포가 차이가 있어 target을 예측하는데 주요하게 작용할 것으로 예상
  - 반면, MSinceMostRecentTradeOpen”과 같은 데이터는 분포의 차이도 없을 뿐더러 수많은 이상치로 구성되어있어 모델 예측에 크게 도움되지 않을 것으로 예상됨 –〉 Feature Selection을 통해 불필요한 컬럼 filtering 필요성



- **데이터 EDA – Multicolinearity**

  – Meaningful & independent features를 선택하기 위해 각 feature간의 correlation을 확인함

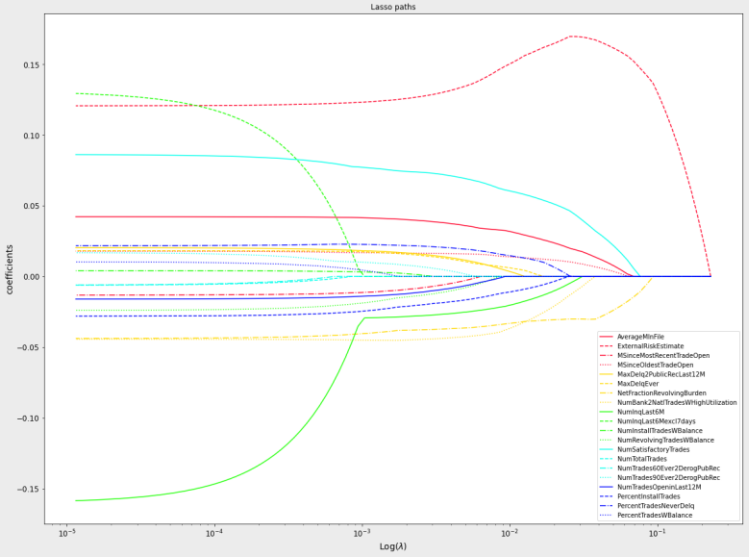


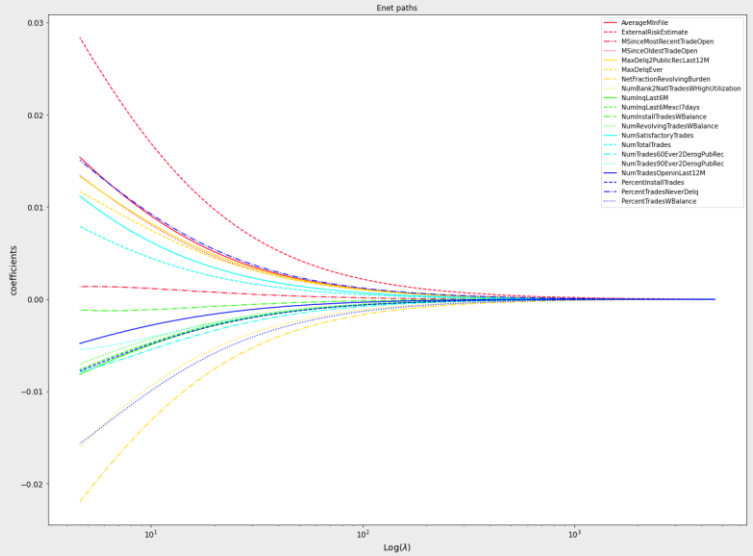
* High correlation features

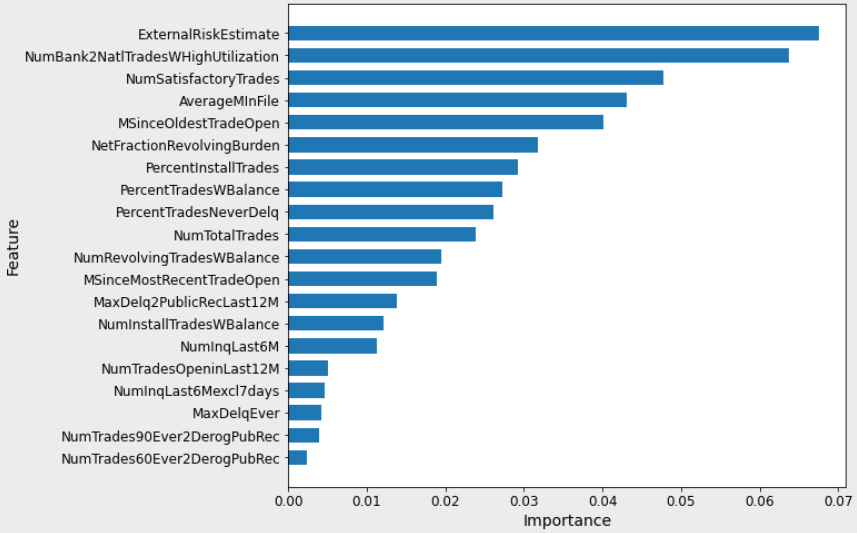
- AverageMinFile – MSinceOldestTradeOpen (0.69)
- NumTotalTrades – NumSatisfactoryTrades (0.85)
- NumTrades60Ever2DerogPubRec – NumTrades90Ever2DerogPubRec (0.99)
- NumInqLast6Mexcl7days – NumInqLast6M (0.99)

- **Feature selection**
  - Model based method (Lasso, Elastic Net)
  - 매개변수 λ에 따라서 feature에 해당하는 계수를 조절하여 어떤 feature를 선택할지 결정할 수 있음
  - 이외에도, XGBoost의 permutation importance를 사용하는 방법등을 고려함

✓feature selection을 수행하기 위해, lasso 방법과 XGBoost를 이용한 방법을 선택함







• Model – Regression based

$$logit(p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \quad logit(p) = \beta_0 + f_1(x_1) + \cdots + f_p(x_p)$$

▪ Logistic regression

✓ 모든 features 사용

– Accuracy : 0.717

– F1 score : 0.717

▪ Logistic GAM

✓ linear splines (Lasso)

– Accuracy : 0.728

– F1 score : 0.727

✓Logistic regression에서 모든 features를 사용하는 경우 모델 성능이 좋지 않음

– Regression 학습을 방해하는 feature가 있을 것으로 생각됨

✓ Lasso feature selection

– Accuracy : 0.717

– F1 score : 0.717

✓ B-spline with degree 3 (Lasso)

– Accuracy : 0.725

– F1 score : 0.725

✓단순 logistic regression 보다 GAM에서 outperform

– Feature간 비선형성을 더욱 잘 반영한 것으로 보임

• Model – Tree based

▪ Decision tree

✓ 모든 features 사용

- Accuracy : 0.706
- F1 score : 0.705

✓ XGBoost feature selection

- Accuracy : 0.705
- F1 score : 0.705

▪ Random Forest

✓ 모든 features 사용

- Accuracy : 0.718
- F1 score : 0.717

✓ XGBoost feature selection

- Accuracy : 0.713
- F1 score : 0.712

✓Tree 기반의 방법은 regression 방법과 다른 결과

– 모든 feature를 사용했을 때, 더 좋은 결과를 보여줌

✓DT에서 CV를 수행한 결과보다 RF의 결과가 outperform

– 앙상블기반의 방법이 단일 모델보다 우위

- **Model – XGBoost**

✓ 모든 features 사용

– Accuracy : 0.728

– F1 score : 0.727

✓XGBoost방법 또한 여러 개의 DecistionTree를 조합하기 때문에 모든 feature를 사용했을 때, 더 좋은 결과를 보여주는 것으로 생각됨

– 모델의 오류를 순차적으로 보완해가며 학습하는 방식이기 때문에 기본적인 모델보다 성능이 좋은 것으로 생각됨

✓ XGBoost feature selection

– Accuracy : 0.719

– F1 score : 0.719

✓현재까지 결과에서 가장 좋은 성능을 보여준 모델은 Logistic GAM(linear spline/lasso features; acc 0.730)방법과 XGBoost(all feautures ; acc 0.732) 임

– Logistic GAM은 이미 튜닝된 결과임

– XGBoost에 대해 하이퍼 파라미터 튜닝 수행

- **Model – XGBoost (모든 feature 사용)**

✓기존 결과

- Accuracy : 0.728
- F1 score : 0.727

✓ Bayesian optimization 사용 결과

- Accuracy : 0.729
- F1 score : 0.728

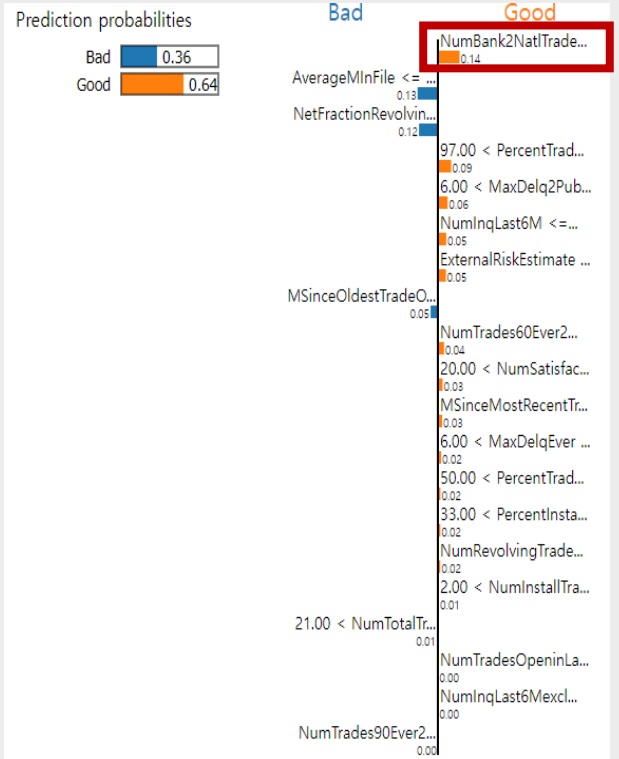
✓모델 학습에 주요한 영향을 미칠 것으로 생각되는 파라미터를 선정하여, Bayesian optimization을 수행함

- 기존 하이퍼파라미터를 갖는 모델보다 성능이 개선되는 모습을 보여줌
- 추가적으로 어떤 feature가 모델 학습에 주요한 영향을 주었는지 LIME 기법을 통해 확인함

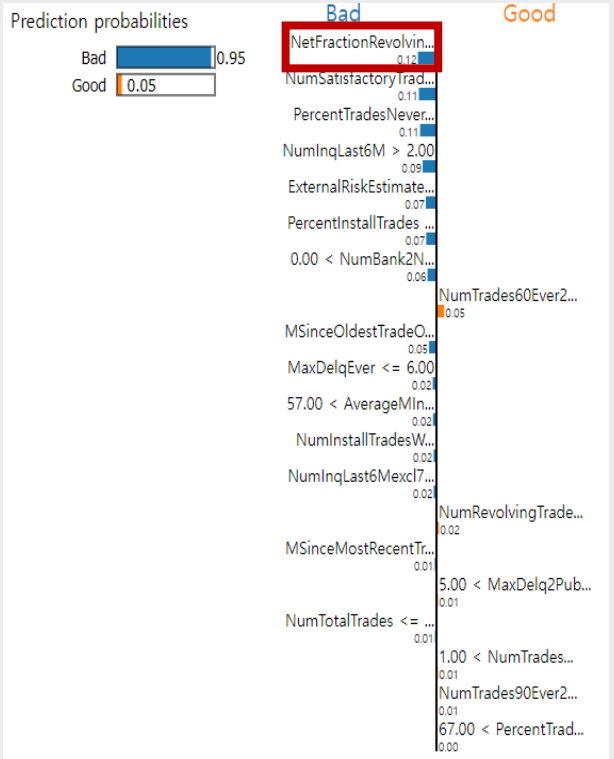
- **Model – XGBoost LIME & SHAP**

▪ LIME (Local Interpretable Model-Agnostic Explnations)

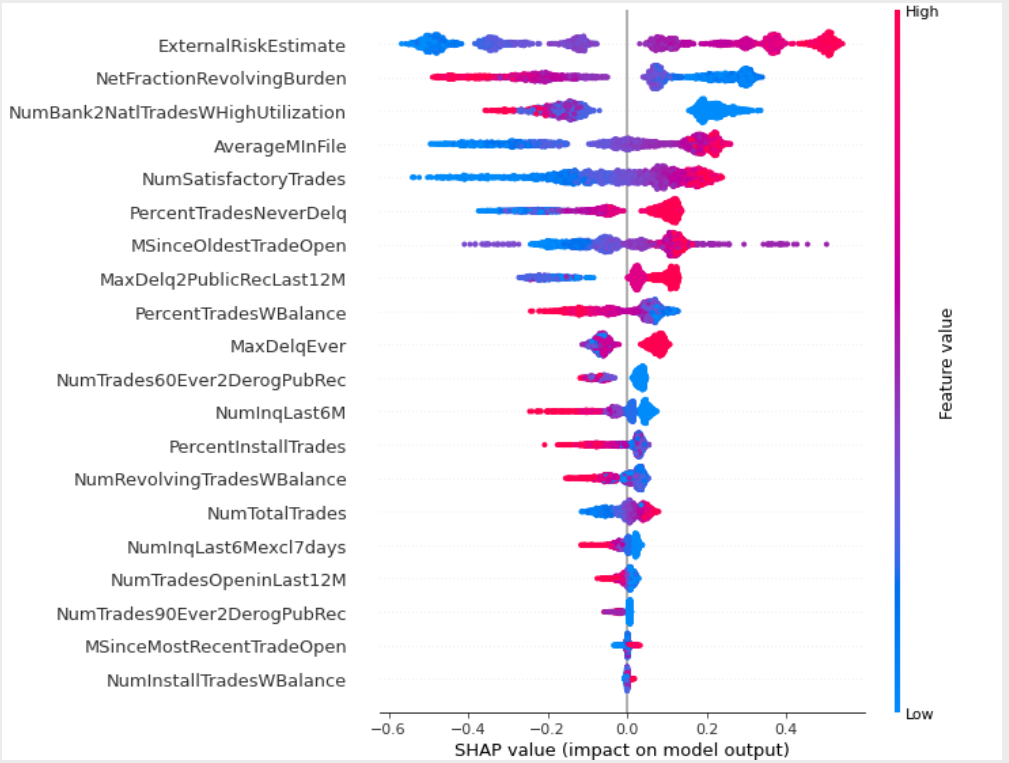
✓ worst case (real: Bad)



✓ best case (real: Bad)



▪ SHAP (Shapley Additive exPlanation)



- **요약 및 결론**

*"대출 신청자의 **2년의 기간의 신용 보고서"**에 해당하는 기록으로 "RiskPerformance"에 대한 분류를 수행하여 대출 연체 여부 예측"*

- EDA

– Missing value에 대하여 각 feature 특징에 따라 다른 방법을 사용하여 imputation을 수행

– Target label에 따른 분포를 확인해보고 feature간 correlation 분석을 통해 feature selection을 진행

✓ Lasso와 XGBoost의 permutation importance를 이용하여 feature selection

- **성능 비교**

Model	Feature Selection	Additional Technique	Acc	F1
XGBoost	All	Bayesian Opt	0.729	0.728
Logistic GAM	Lasso	Linear Spline	0.728	0.727
Random Forest	All		0.718	0.717

- LIME & SHAP

- 추가적으로, 모델이 어떤 feature를 사용하여 학습되었는지 그 중요도를 살펴봄
- 가장 부정확한 sample과 정확한 sample에 대해 LIME 기법을 적용해본 결과, 변수 중요도에 있어 그 차이가 눈에 띔
- 전체 모델에 대한 feature 중요도를 확인해본 결과, “ExternalRiskEstimate”가 가장 높은 영향력을 “NuminstallTradesWBalance”가 가장 낮은 영향력을 보여주었지만 대부분의 feature 의 SHAP값이 ± 0.5이하로 나타남

- **결론**

- 다양한 모델을 적용한 결과, 부스팅 기반의 XGBoost가 가장 좋은 성능을 보여주며, logisticGAM 또한 좋은 성능을 보여주었음
- 모델의 정확도 측면에서는 black-box 모델이 XGBoost가 강점을 보였지만, 해석력 측면에서 logisticGAM에 우위를 점하기는 힘들 것으로 예상됨
- 하지만, XGBoost 모델에 LIME, SHAP와 같은 방법론을 적용하여 모델의 작동 원리를 살펴볼 수 있었음
- 추후, robust & reliable ML 모델에 대한 연구를 수행하고자 함

References

- Demajo, Lara Marie, Vince Vella, and Alexiei Dingli. "Explainable ai for interpretable credit scoring." *arXiv preprint arXiv:2012.03749* (2020).

- Torrent, Neus Llop, Giorgio Visani, and Enrico Bagli. "PSD2 Explainable AI Model for Credit Scoring." *arXiv preprint arXiv:2011.10367* (2020).

- Han, Jesun. "주택을 담보로 신용카드처럼 쓴다? 미국의 HELOC 대출을 알아보자!" *주택을 담보로 신용카드처럼 쓴다? 미국의 HELOC 대출을 알아보자! : 네이버 포스트*, Woori Bank, 24 June 2021, https://post.naver.com/viewer/postView.naver?volumeNo=31831807&memberNo=38946978&vType=VERTICAL.

감사합니다.

Q&A