BigContest futures

팀 노란오리
팀장 : 전주혁 

팀원 : 반소희 박민정 김수연

Contents

Intro

1. 주제

앱 사용성 데이터를 통한 대출신청 예측분석

* 대출신청, 미신청 고객을 분류하여 고객의 특성 분석결과 도출

* 고객의 특성에 따라 대출 신청, 미신청 분류가 쉬워진다면 더욱 효율적인 대출 상품 기대 가능

* 대출 신청 확률이 작은 고객들에게 쓰는 시간적, 인적 자원이 줄어 들어 효율적인 영업 가능

Intro

2. Data Access

1. 3개의 csv 파일을 merge 시킨 후 코드 구현 시작
→ RAM 초과 오류 발생

2. csv 파일 각자 전처리 후 데이터 타입 변경으로 용량 축소 후 merge
→ RAM 초과 오류 발생

3. csv 파일 PCA 후 용량 축소 후 merge
→ **Success**

Part 2,

EDA

Part 2,

EDA – 결측치처리

```
merge_1_2_train.isnull().sum() # 결측치 확인 : 결측치 모두 int or float type
```

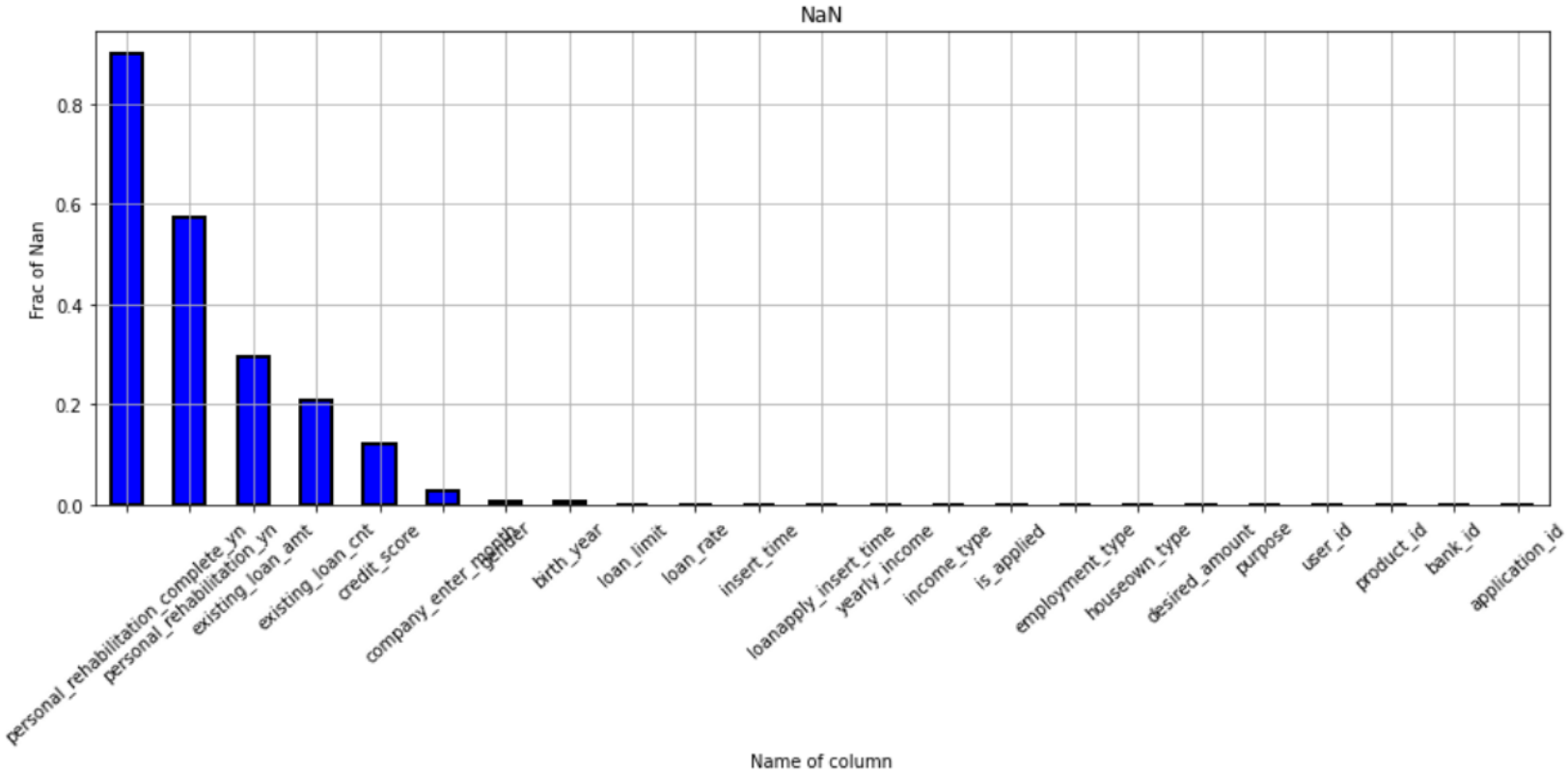
```
application_id                      0
user_id                             0
birth_year                      91626
gender                          91626
insert_time                         0
credit_score                  1243812
yearly_income                       0
income_type                         0
company_enter_month            303568
employment_type                     0
houseown_type                       0
desired_amount                      0
purpose                             0
personal_rehabilitation_yn    5873229
personal_rehabilitation_complete_yn  9232232
existing_loan_cnt             2143811
existing_loan_amt             3044140
loanapply_insert_time               0
bank_id                             0
product_id                          0
loan_limit                       5625
loan_rate                        5625
is_applied                          0
dtype: int64
```

결측치 처리 해당 컬럼

- birth_year (생년월일)
- gender(성별)
- credit_score(한도조회 당시 유저 신용점수)
- company_enter_month(입사연월)
- personal_rehabilitation_yn(개인회생자 여부)
- personal_rehabilitation_complete_yn (개인회생자 납입 완료 여부)
- existing_loan_cnt (기대출수)
- existing_loan_amt (기대출금액)
- loan_limit(승인한도)
- loan_rate(승인금리)

Part 2,

EDA – 결측치처리



데이터 결측치 분포 파악

- personal_rehabilitation_complete_yn
개인회생자 납입 완료 여부(90%)

- personal_rehabilitation_yn
개인회생자 여부
결측치 비율(57%)

- existing_loan_amt
기대출금액(30%)

- existing_loan_cnt
기대출수(21%)

- credit_score
한도조회 당시 유저 신용점수(12%)

Part 2,

EDA – 결측치처리

데이터 결측치 처리 방식

- personal_rehabilitation_complete_yn(90%) → 많은 결측치 drop

- personal_rehabilitation_yn(57%) → 많은 결측치 drop

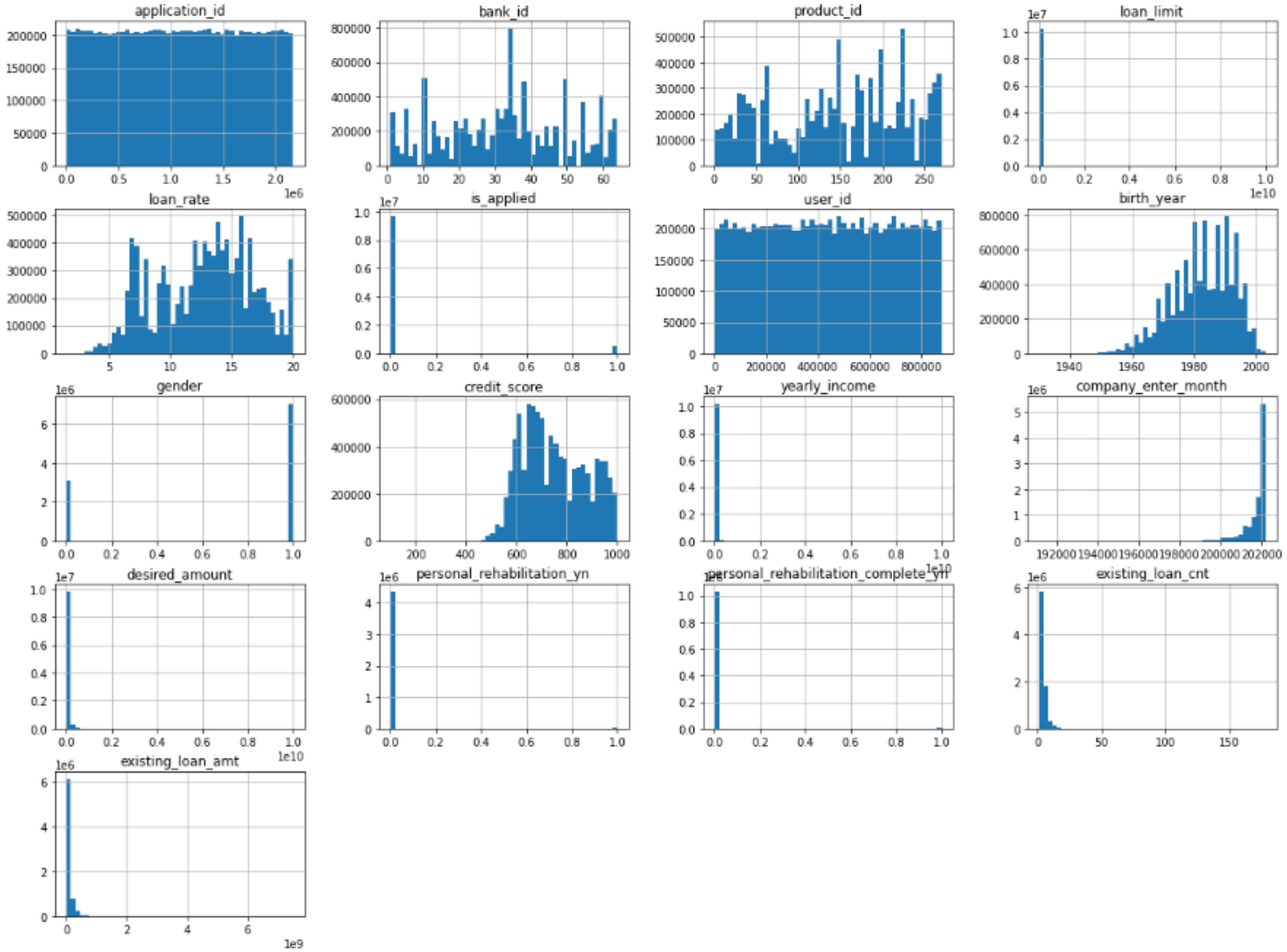
- existing_loan_amt(30%) → 많은 결측치 drop

- company_enter_month(3%) → 많은 결측치 drop
(column 내에서는 3%이지만 전체 데이터에 비해 결측치가 많은편)

이 외의 결측치 coiumn → 분포 파악하여 결측치 처리

Part 2,

EDA – 결측치처리



결측치 처리 방식 선택을 위한 데이터 분포 histogram 파악

+

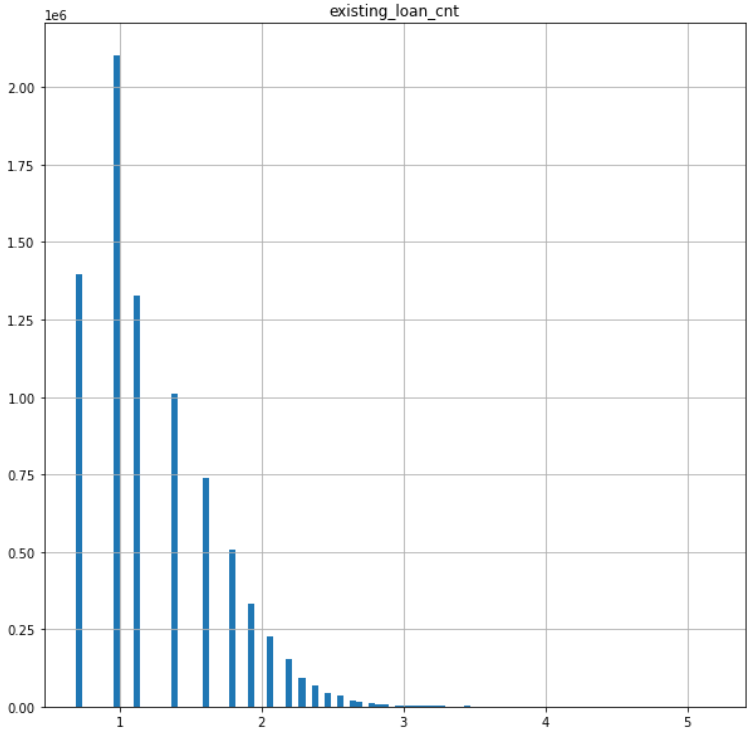
한쪽으로 치우친
데이터 분포 확인

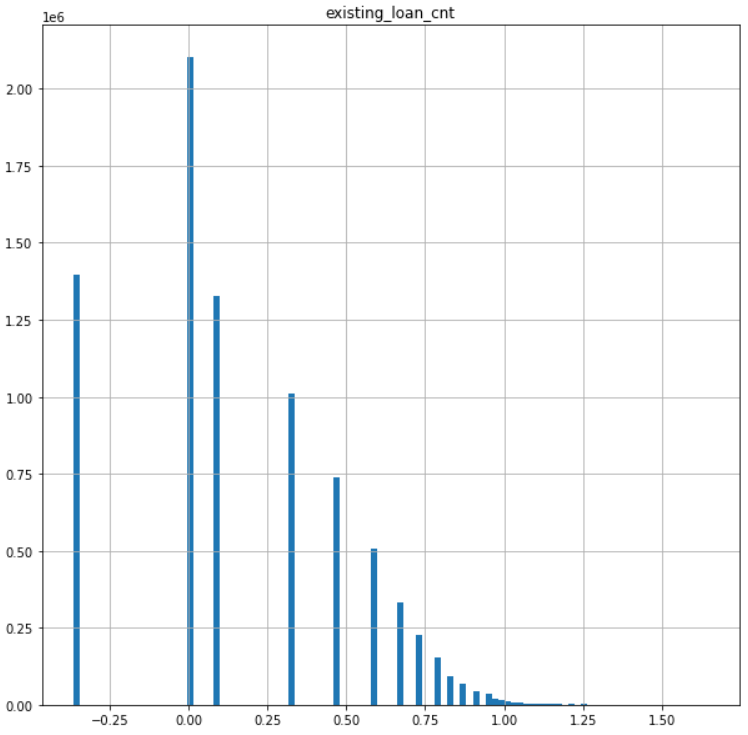
-> log변환 방식 사용

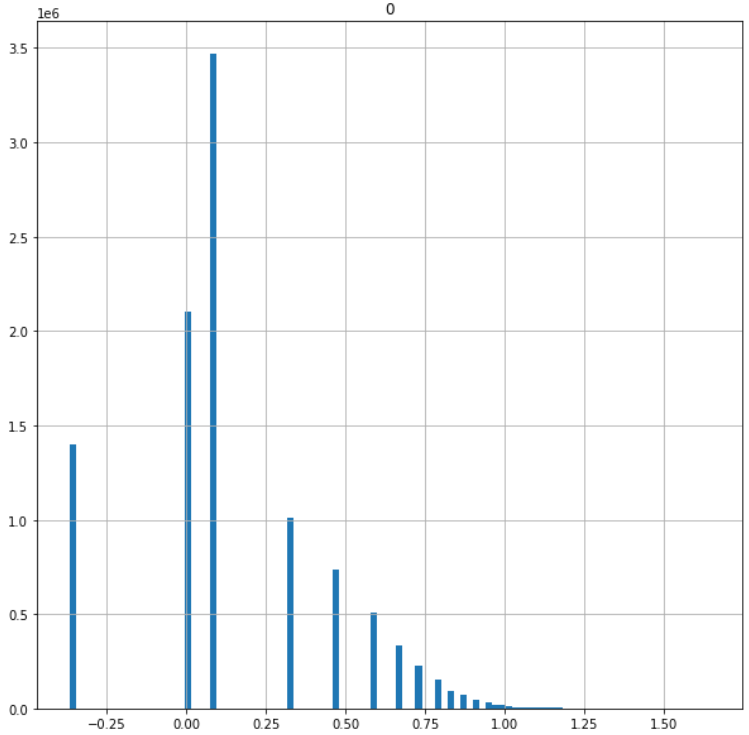
Part 2,

existing_loan_cnt 결측치 처리

데이터 histogram 파악







log변환한 히스토그램
- 왼쪽으로 치우친 분포

log변환 한번 더 실시
- 상대적으로 정규분포에 가까워짐

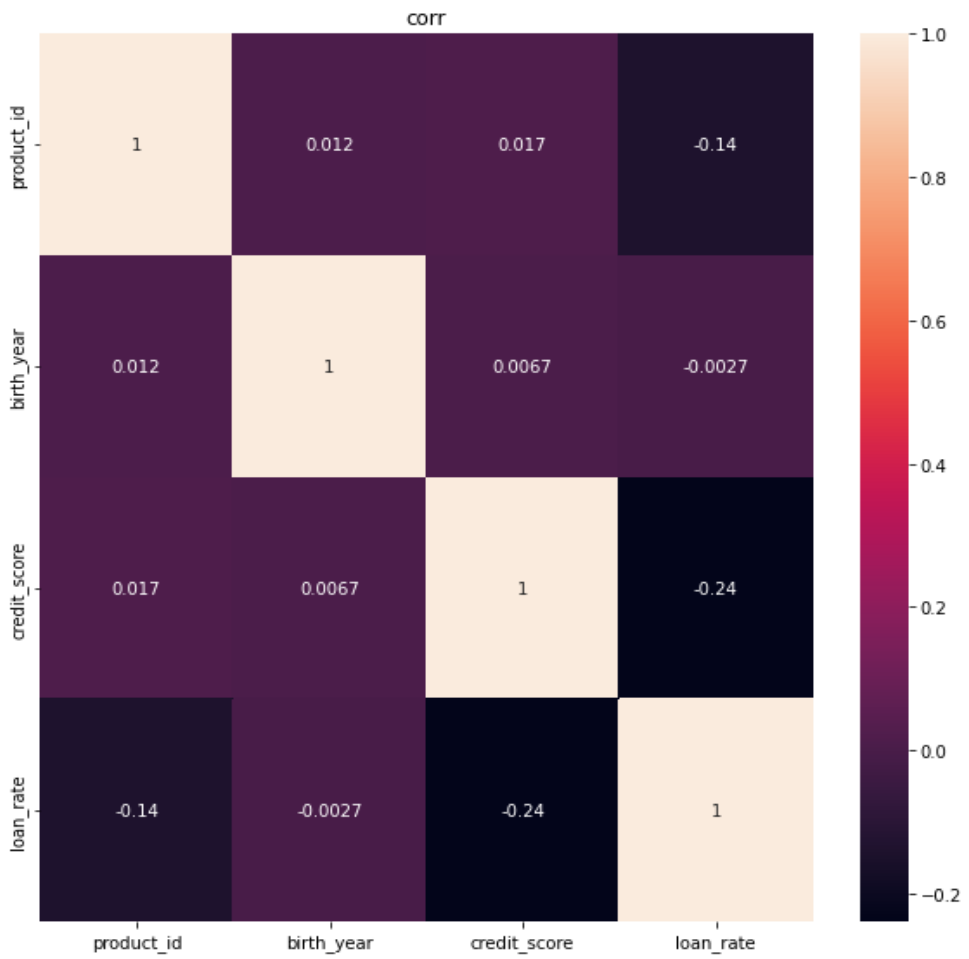
분포 특성에 맞춰
결측치를 **평균값**으로 대체한 분포
- 더욱 정규분포에 근사

Part 3,

Feature Engineering

(1) – 1 상관계수 기반 Feature Engineering

상관계수 heatmap



상관계수 상대적으로 높은 column heatmap

product_id, birth_year
credit_score, loan_rate
변수 사용

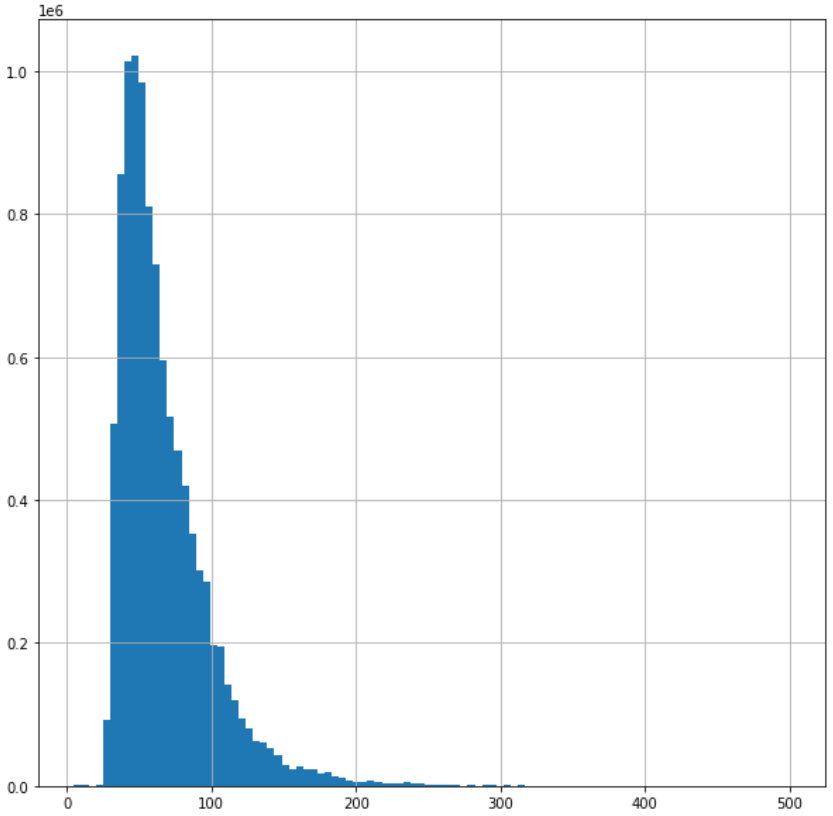
→ 가장 상관성이 높은 (-0.24)
credit_score , loan_rate

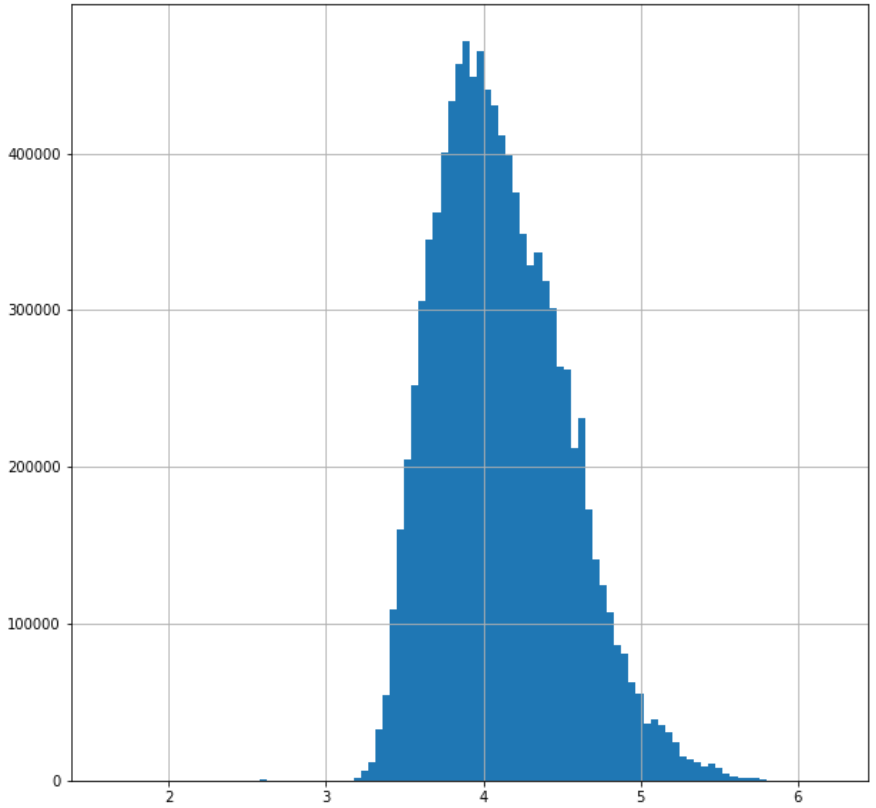
→ 두번째로 높은(-0.14)
product_id, loan_rate

Part 3,

credit_score , loan_rate

유의미한 feature 생성





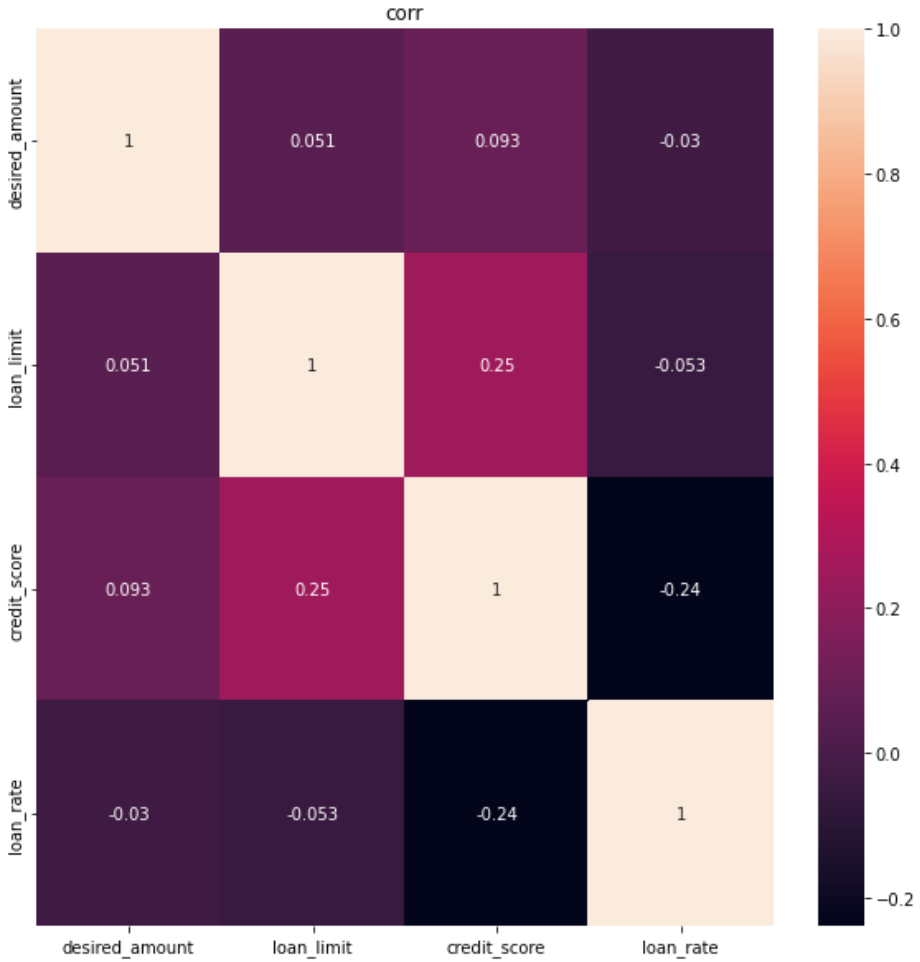
credit_score/loan rate 히스토그램
- 왼쪽으로 치우친 데이터 분포

credit_score/loan rate log변환
- 정규분포에 매우 근사한 분포

Part 3,

(1) – 2 상관계수 기반 Feature Engineering

상관계수 heatmap



상관계수 상대적으로 높은 column heatmap

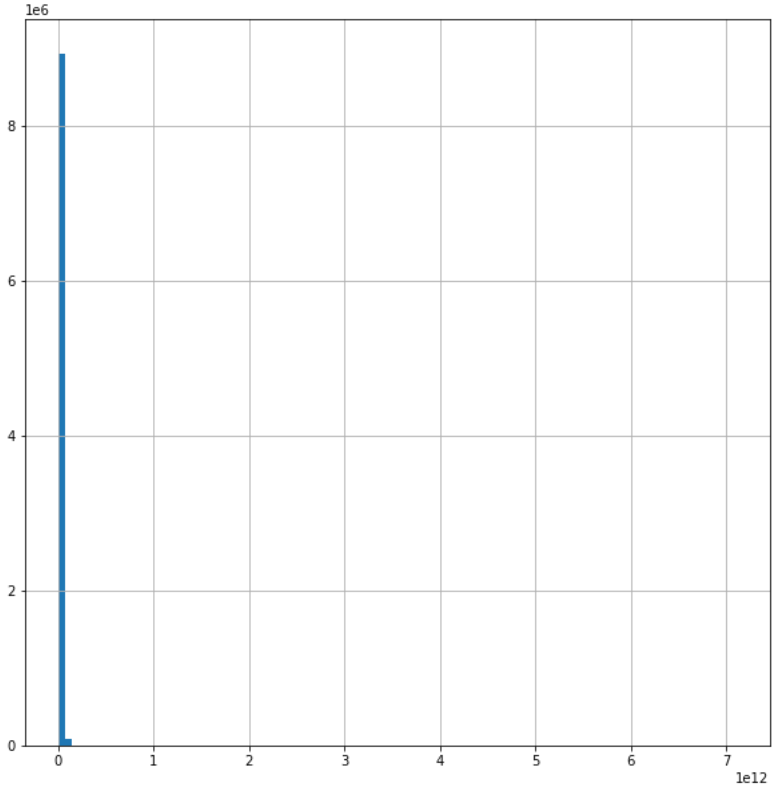
desired_amount, loan_limit
credit_score, loan_rate
변수 사용

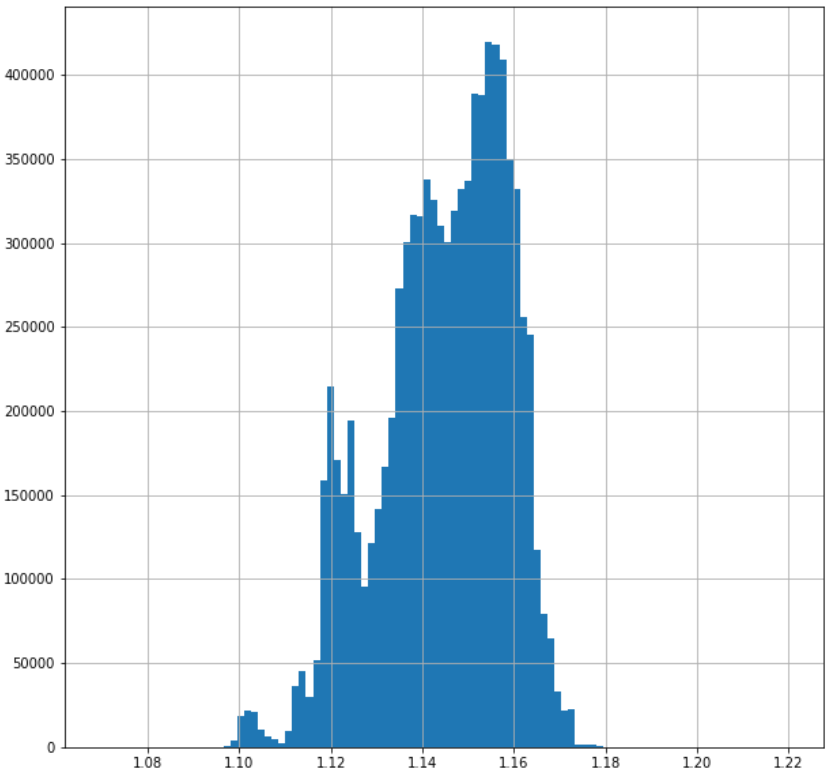
→ 가장 상관성이 높은 (0.25)
credit_score , loan_limit

Part 3,

credit_score, loan_limit

파생변수 생성



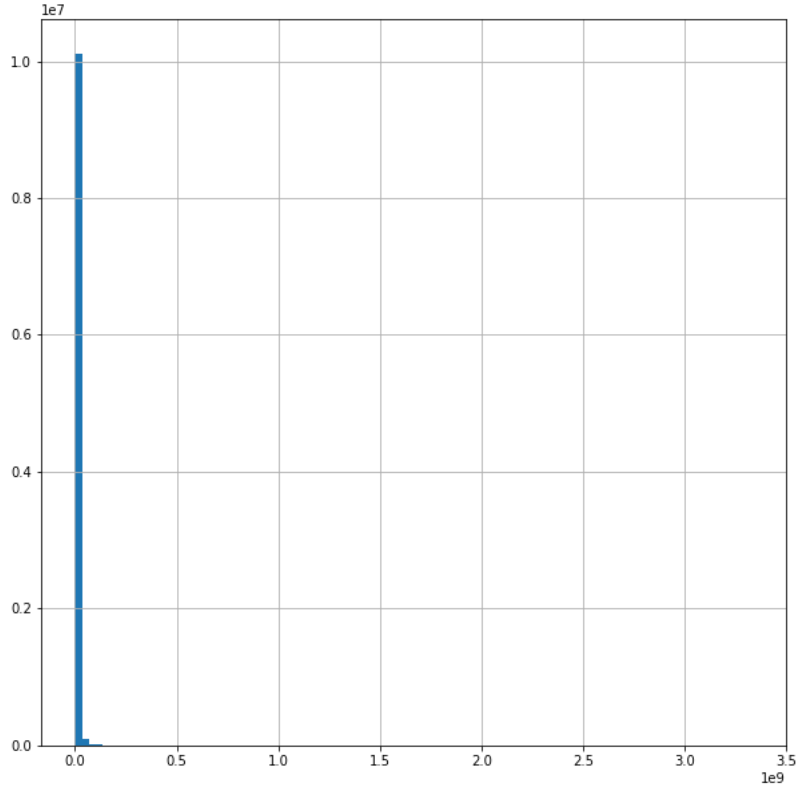


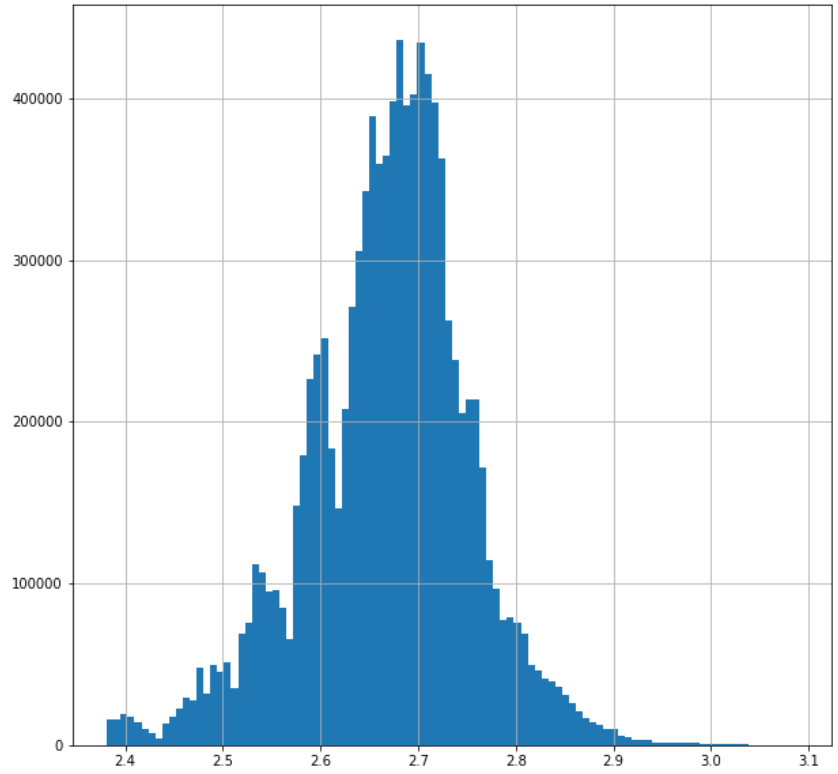
credit_score*loan_limit 히스토그램
- log변환 필요

credit_score*loan_limit log변환
– 정규분포에 가까워져 분석에 적합

(2) – 1 가설 기반 Feature Engineering

Hypothesis : 대출희망금액이 높은데 금리가 낮으면 빌릴 확률이 높을 것이다.



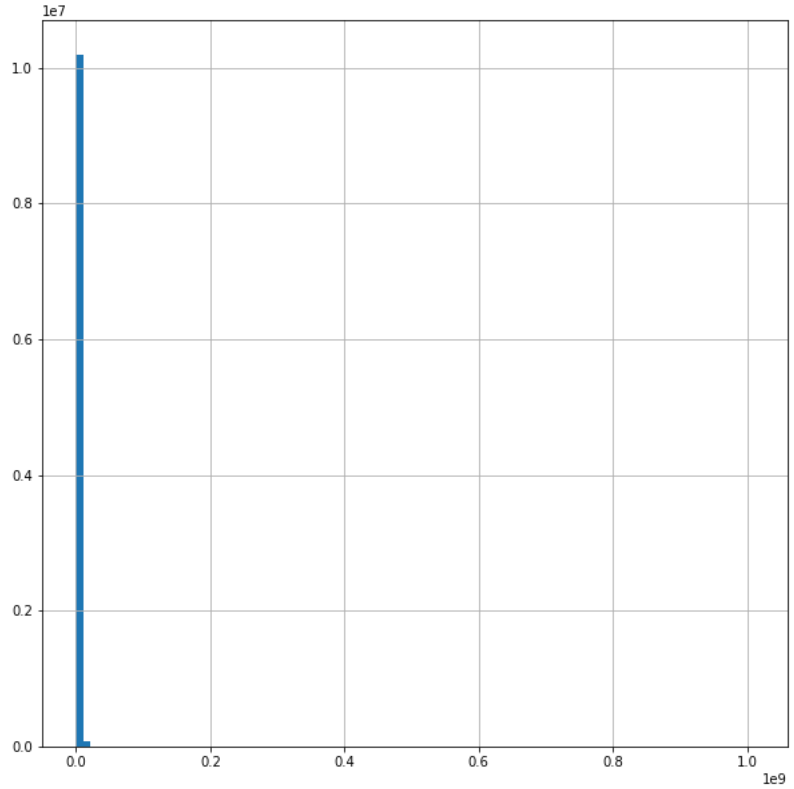


desired_amount/loan_rate 히스토그램
- log변환 필요

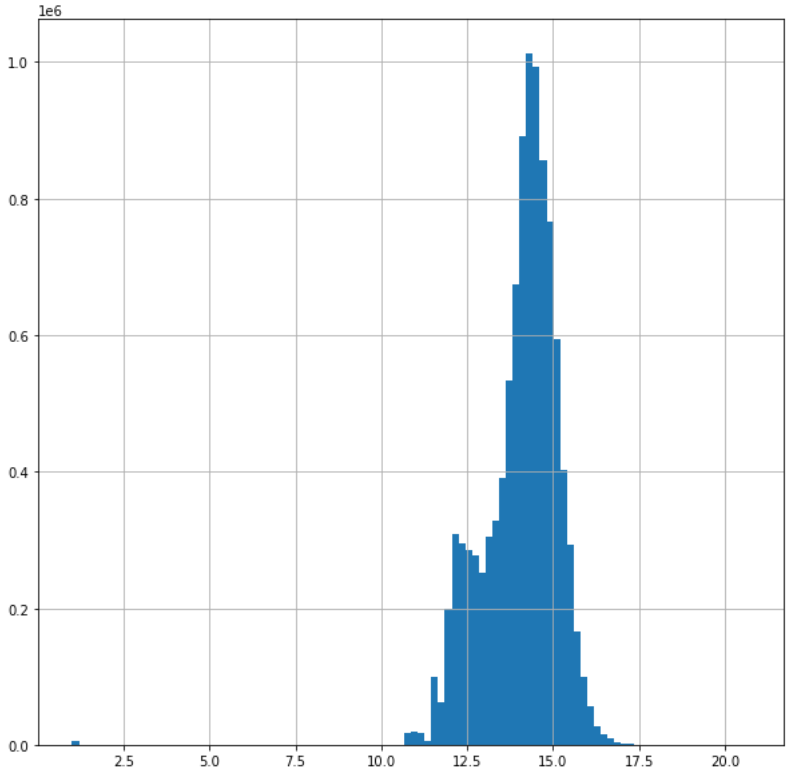
desired_amount/loan_rate log 변환
- 정규분포에 가까워져 분석에 적합

(2) – 2 가설 기반 Feature Engineering

Hypothesis : 승인금리가 낮을수록 승인한도가 높을 것이다.



loan_limit/loan_rate 히스토그램
- log변환 필요



loan_limit/loan_rate log 변환
- 정규분포에 근사

Feature Engineering

object 변수 encoding

데이터의 크기가 너무 커지는 문제 발생

one-hot encoding

머신러닝이 독립적인 변수들을 종속적인 관계로 학습을 할 수 있음

Label Encoding

트리계열 모델

두 문제 모두 해결 가능한
트리계열 모델
사용 결정 !

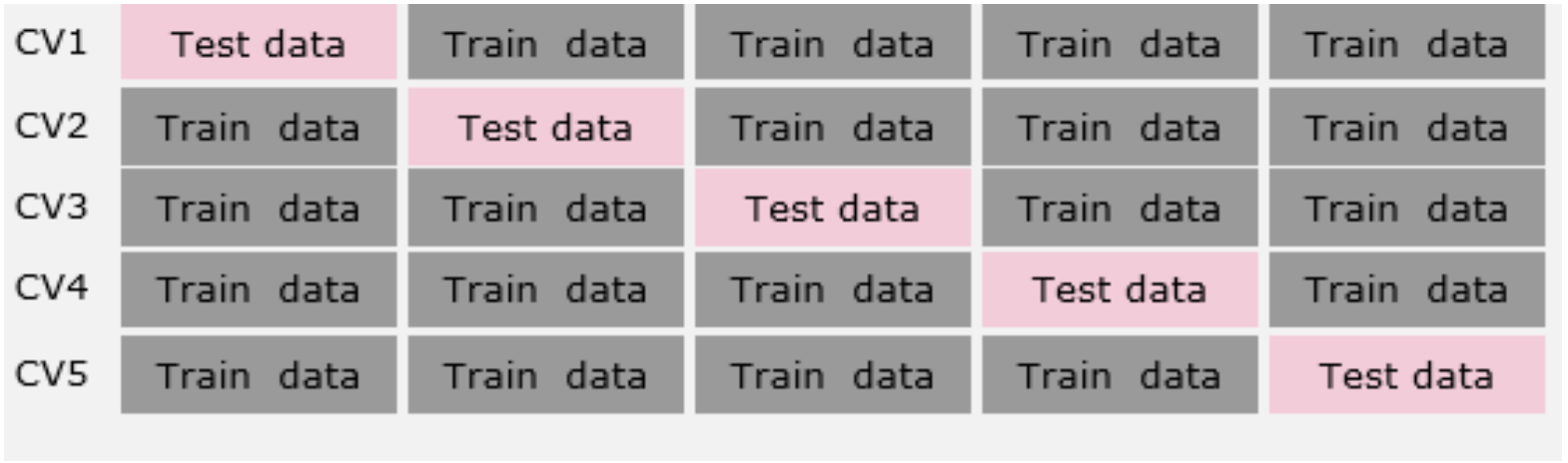
Part 4,

Validation

Validation

계층별 K-fold 교차 검증 활용

(회귀) 일반적 방법론 – KFOLD 교차 검증



장점

- 특정 데이터셋에 과적합 방지
- 일반화된 모델 생성
- 과소적합 방지(데이터셋 규모가 작을경우)

단점

- 데이터 클래스가 불균형한 경우 학습 데이터가 고루 분할되지 못함

Validation

계층별 K-fold 교차 검증 활용

활용 방법론 – 계층별 KFOLD 교차 검증

-> 주기가 다른 2개의 사이클을 지닌 데이터 특성을 반영하기 위해
원본 데이터의 분포를 반영하는 계층별 KFOLD 교차 검증 활용

Part 5,

Modeling

Part 5,

Modeling

pycaret

모델 별 평가지표를 쉽게 보기 위해 pycaret 사용

pycaret 이란?

- Machine Learning Workflow를 자동화하는 오픈소스 라이브러리

- Classification, Regression, Clustering 등의 Task에서 사용하는 여러 모델들을 동일한 환경에서 실행을 자동화한 라이브러리

- 여러 모델을 비교 가능

Modeling

Model Metrics

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9462	0.9003	0.0598	0.5733	0.1084	0.0991	0.1728	752.286
gbc	Gradient Boosting Classifier	0.9455	0.8495	0.0285	0.5077	0.0540	0.0485	0.1106	1932.224
dt	Decision Tree Classifier	0.9176	0.6155	0.2764	0.2602	0.2680	0.2244	0.2245	197.398
et	Extra Trees Classifier	0.7562	0.7026	0.0284	0.3863	0.0529	0.0473	0.0958	666.226

→ 트리계열 중 Accuracy와 Recall, F1 평가지표가 제일 좋은 DT를 사용

Modeling

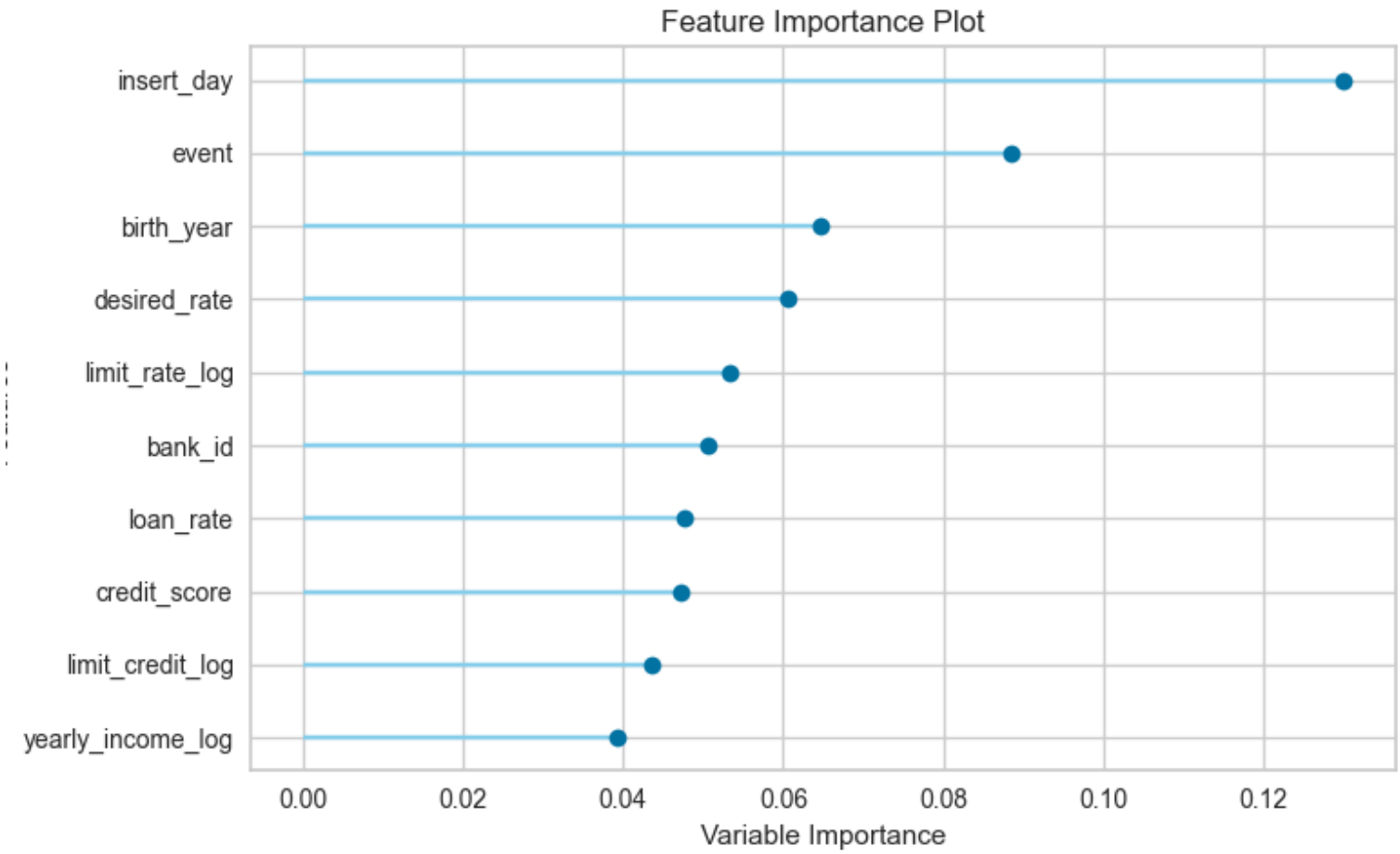
validation

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9158	0.6202	0.2829	0.2625	0.2723	0.2277	0.2279
1	0.9160	0.6206	0.2831	0.2631	0.2728	0.2282	0.2284
2	0.9155	0.6191	0.2804	0.2599	0.2698	0.2250	0.2252
3	0.9160	0.6206	0.2835	0.2633	0.2730	0.2285	0.2286
4	0.9161	0.6204	0.2829	0.2636	0.2729	0.2285	0.2286
Mean	0.9159	0.6202	0.2826	0.2625	0.2722	0.2276	0.2277
Std	0.0002	0.0006	0.0011	0.0013	0.0012	0.0013	0.0013

→ **Decision Tree 모델 사용, Straitified K - fold Cross Validation 결과**

Feature Importance

Feature Importance



Feature Importance 결과

“ 감사합니다. ”