부도예측모형 with Lending Club Data

4 조

김태훈
남경혜
방수영
이상윤
정재영

목차

1. 분석 목적

Lendingclub이란?

- 세계 최대 p2p 대부업체
- 투자자들과 차입자가 Lendingclub 플랫폼을 통해 자유롭게 대출 금액과 이자율을 결정
- 투자자들에게 차입자들의 신용관련 정보들을 제공 후 수수료 이익을 얻음

이윤 결정 요소 ⇨ **대출의 성사 건수**

1. 분석 목적

$$E\pi = (1 - P)R - P(1 - RR) - rf$$

Eπ : 투자 1원당 기대수익
P : 부도확률
RR : 회수율(0.66 가정)
R : 차입자와 투자자가 결정한 이자율(lendingclub data 상의 이자율 평균 0.126)
rf : 무위험 이자율 (한국은행 기준금리 0.5%로 가정)

Eπ = 0 이 되는 P가 threshold

산출된 threshold = 0.26
P의 예측치가 threshold 보다 크면 부도 위험 signal 전달

⇨ **목적 : 투자자 이윤 극대화**

2. 데이터 전처리

2. 데이터 전처리

Data shape ⇨ 109291 x 333 형태의 데이터

	loan_amnt	funded_amnt	funded_amnt_inv	int_rate	installment	annual_inc	dti	delinq_2yrs	fico_range_low
0	19000	19000	19000.0	0.0916	605.62	65000.0	16.36	1	670
1	10000	10000	10000.0	0.0789	312.86	58000.0	5.03	0	690
2	6000	6000	6000.0	0.1147	197.78	46900.0	24.23	2	665
3	25200	25200	25200.0	0.1199	836.89	76280.0	32.87	0	685
4	8000	8000	8000.0	0.1299	269.52	29000.0	20.28	0	770
...	...	...	...	...	...	...	...	...	...
1092914	24000	24000	24000.0	0.1199	797.03	79000.0	3.90	0	660
1092915	10000	10000	10000.0	0.1199	332.10	31000.0	28.69	0	670
1092916	12000	12000	12000.0	0.1999	317.86	64400.0	27.19	1	695
1092917	13000	13000	13000.0	0.1599	316.07	35000.0	30.90	0	680
1092918	20000	20000	20000.0	0.1199	664.20	100000.0	10.83	0	675

1092919 rows × 333 columns

2. 데이터 전처리

Features

다양한 형태(연속형, 범주형 등)의 특성들을 갖는다.

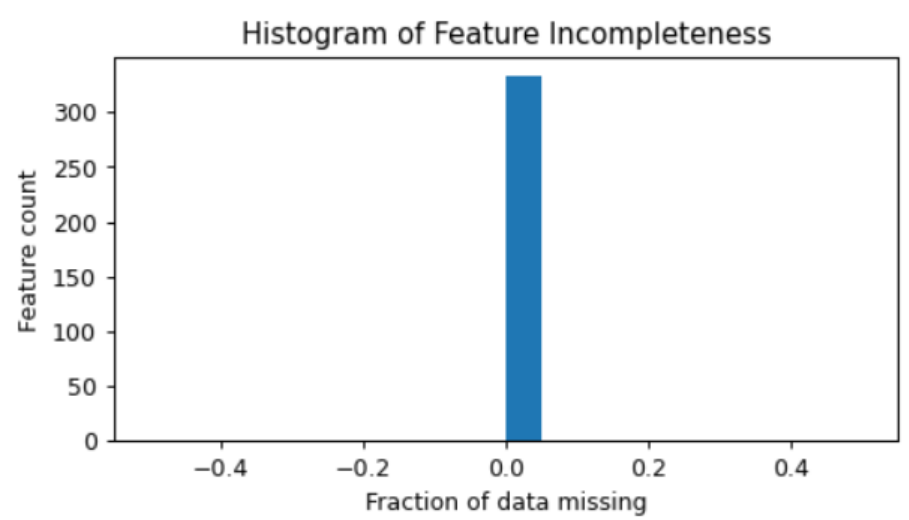
변수명
funded_amnt
int_rate
installment
annual_inc
dti
delinq_2yrs
fico_range_low
inq_last_6mths
open_acc
pub_rec
revol_bal
revol_util
...

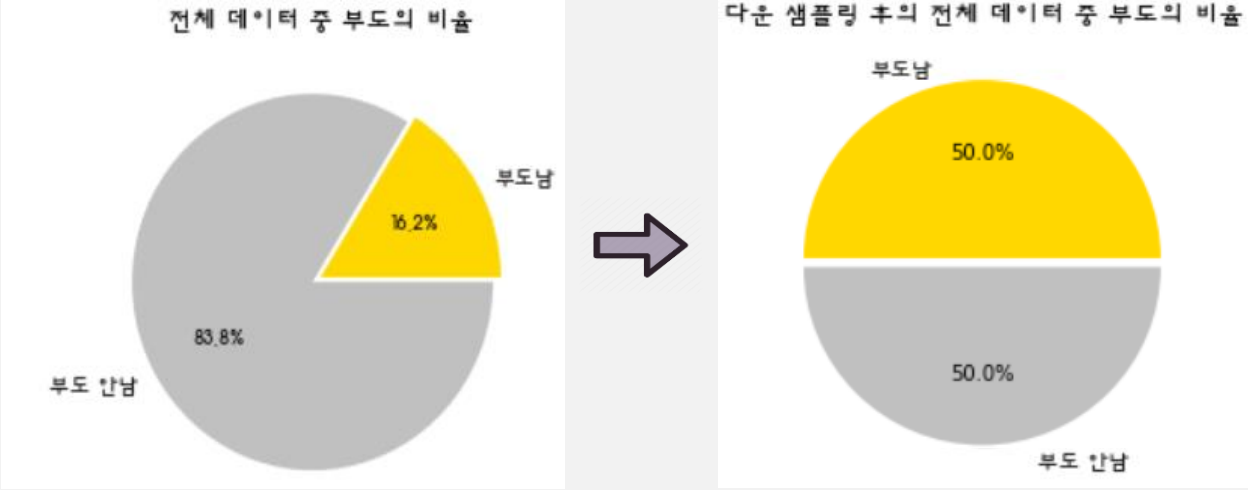
범주형
emp_length(1~12)
home_ownership(1~6)
verification_status(1~3)
purpose(1~14)
addr_state(1~51)
initial_list_status(1~2)
mths_since_last_delinq(1~11)
mths_since_last_major_derog(1~11)
mths_since_last_record(1~11)
mths_since_rcnt_il(1~11)
...

종속 변수
depvar

2. 데이터 전처리

Missing Value & Down Sampling





결측치가 존재하는 Feature가 없다.

종속변수가 편향되어 예측에 영향을 미침

후행 변수 처리

후행 변수	
out_prncp	collection_recovery_fee
out_prncp_inv	last_pymnt_amnt
total_pymnt	last_fico_range_high
total_pymnt_inv	last_fico_range_low
total_rec_prncp	delinq_amnt
total_rec_int	elapsed_t
total_rec_late_fee	debt_settlement_flag1
recoveries	term1

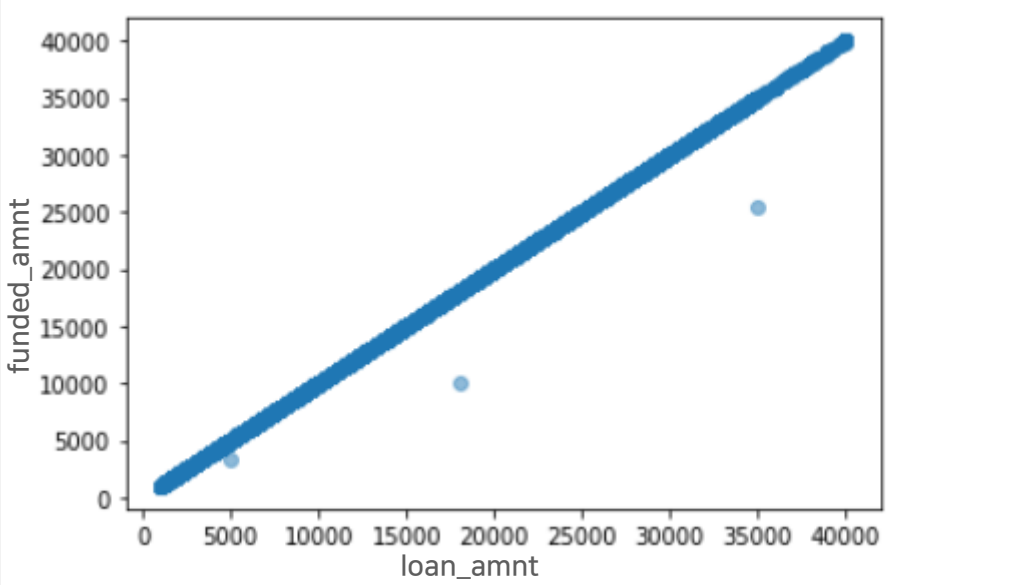
```
for i in range(len(df['recoveries'])):
  if df['recoveries'][i]!=0:
    result.append(1)
  else :
    result.append(0)

mat = confusion_matrix(df['depvar'],result)
mat
```

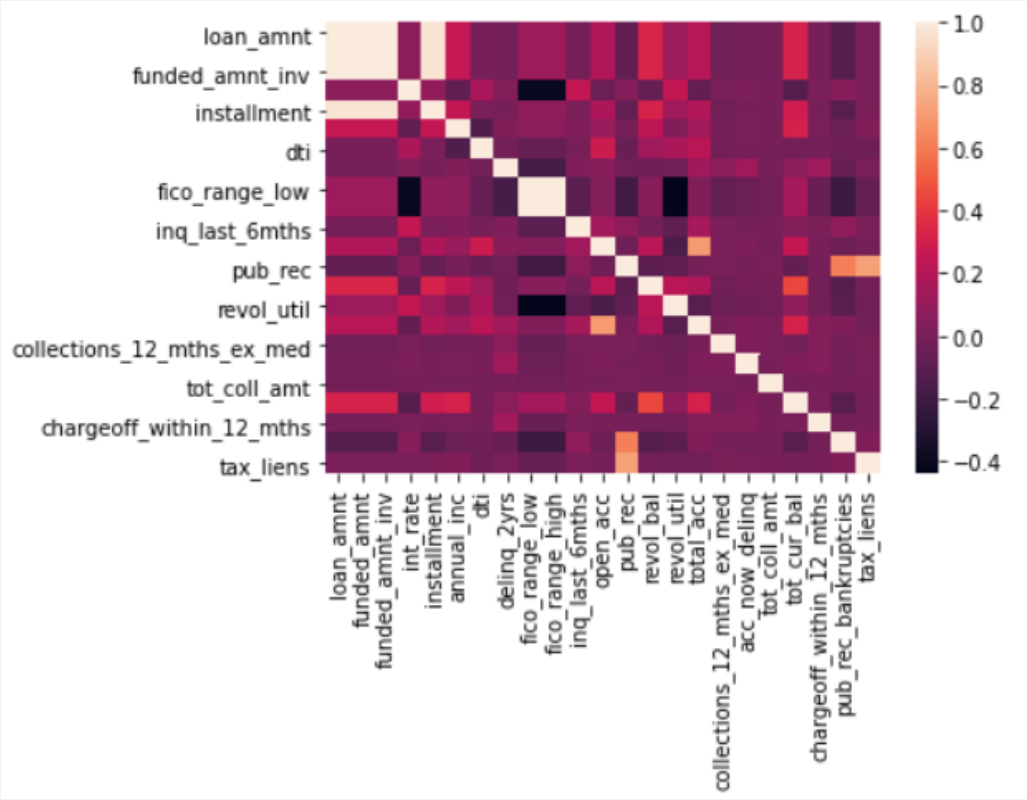
```
array([[916095,      0],
       [ 37777, 139047]])
```

대출자의 부도가 결정 난 이후에 수집되는 변수(=후행변수)가 존재.
후행 변수가 종속변수에 미치는 영향을 검증한 후 일괄 삭제 처리.

2. 데이터 전처리

Correlation





산점도와 히트맵을 통해 연속형 변수간의 상관성을 확인

다중 공선성 확인

다중 공선성을 띄는 변수
loan_amnt & funded_amnt & funded_amnt_inv & installment
fico_rance_low & fico_range_high
total_pymnt & total_pymnt_inv
open_acc & total_acc
revol_util & revol_bal
tax_liens & pub_rec_bankruptcies

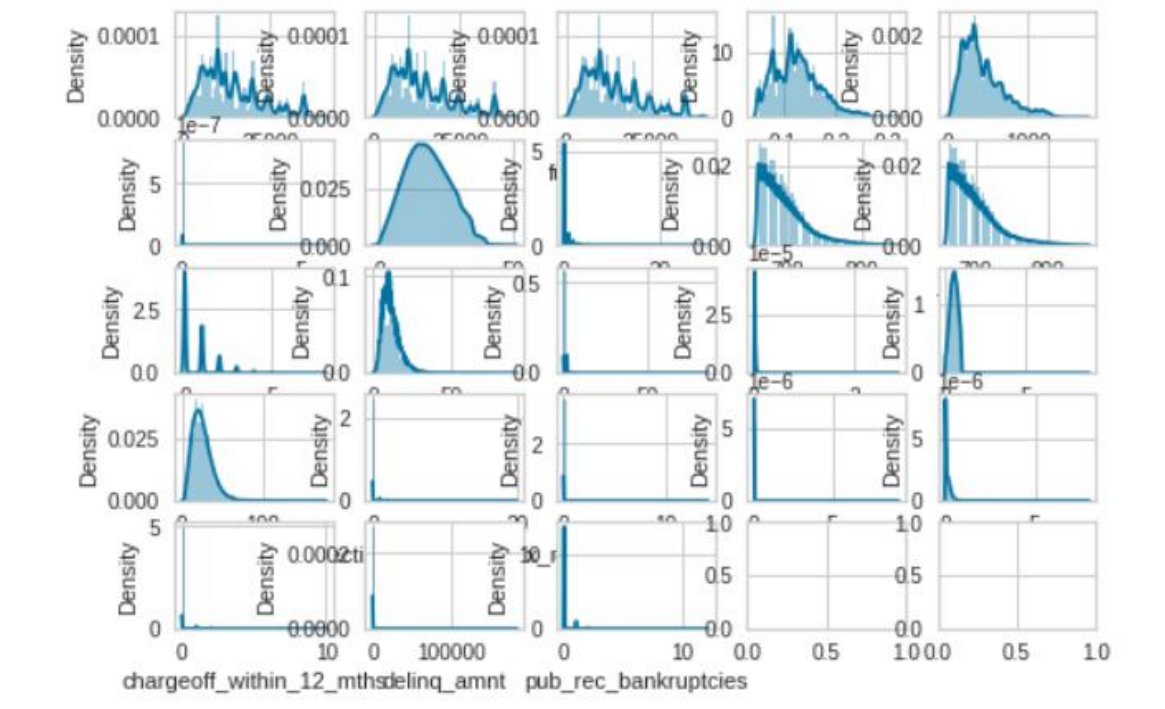
선택한 변수

	VIF Factor	features
0	15.4	Intercept
1	454930.4	loan_amnt
2	535413.0	funded_amnt
3	79481.9	funded_amnt_inv
4	1.0	int_rate
5	1.1	annual_inc
6	1.0	dti

	VIF Factor	features
0	15.4	Intercept
1	1.1	funded_amnt
2	1.0	int_rate
3	1.1	annual_inc
4	1.0	dti

변수 탐색 시 데이터의 유사도가 높은 것으로 추정되는 변수들의 다중 공선성을 VIF로 확인해본 결과, 극단적으로 높은 값을 띄어 제거해도 되는 변수라고 판단.

스케일링



연속형 변수의 데이터 구성을 살펴 보았을 때, 오른쪽으로 꼬리가 긴 비대칭 양상을 띄고 있음. (왜도>0)
그러므로 **이상치에 영향을 받지 않는 robust Scaler**를 사용하여 스케일링을 진행.

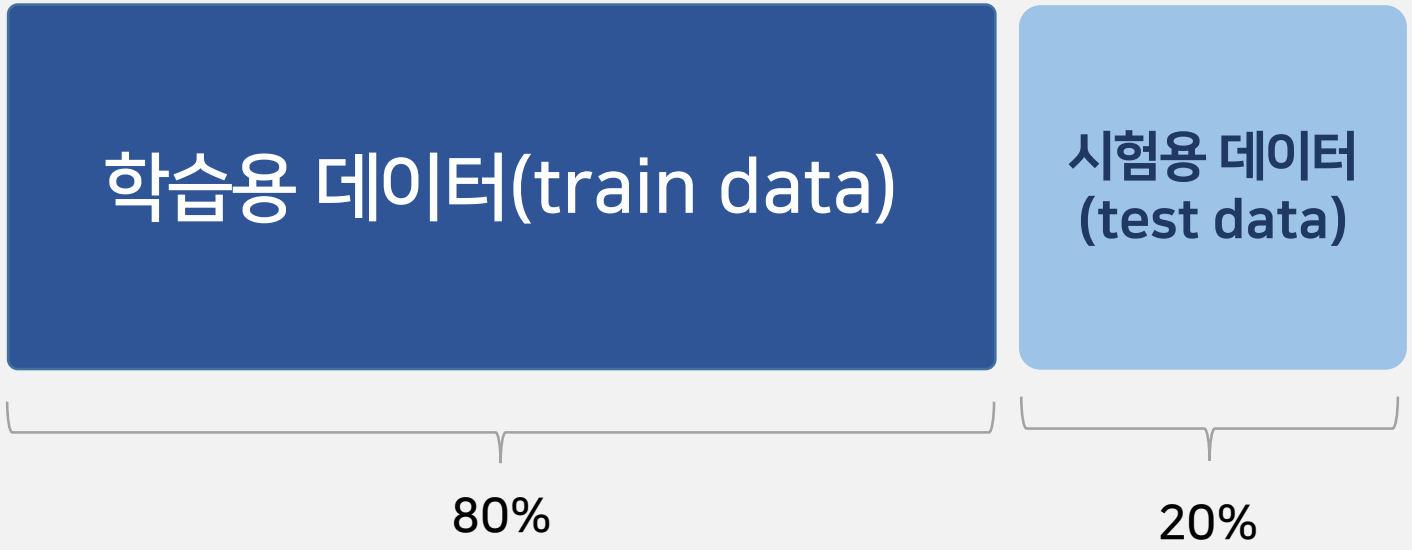
최종 선택 변수

변수명		
funded_amnt_inv	tot_coll_amt	mths_since_last_record(1~11)
int_rate	tot_cur_bal	mths_since_rcnt_i(1~11)
annual_inc	chargeoff_within_12_mths	mths_since_recent_bc(1~11)
dti	delinq_amnt	mths_since_recent_bc_dlq(1~11)
delinq_2yrs	tax_liens	mths_since_recent_inq(1~10)
fico_range_low	emp_length(1~12)	mths_since_recent_revol_delinq(1~11)
inq_last_6mths	home_ownership(1~6)	initial_list_status1
pub_rec	verification_status(1~3)	depvar
revol_bal	purpose(1~14)	
total_acc	addr_state(1~51)	
collections_12_mths_ex_med	mths_since_last_delinq(1~11)	
acc_now_delinq	mths_since_last_major_derog(1~11)	

총 191개의 설명변수와 1개의 종속변수

2. 데이터 전처리

Data split 모델의 정확도 확인 및 과적합 방지를 위해 데이터를 나눠야 함!



8:2의 비율로 학습/시험용 데이터를 나눔

※Validation을 나누지 않은 이유 : Grid search로 하이퍼 파라미터 추정하여 따로 Validation Set은 설정하지 않음

3. 모델 생성

3. 모델 생성

사전 모델 선정 기준 → accuracy & precision & recall 비교 시, 가장 높은 분류 모델 선택

초기 모델 후보

- LPM
- Logistic
- Ridge & Lasso
- Decision Tree
- KNN
- SVM

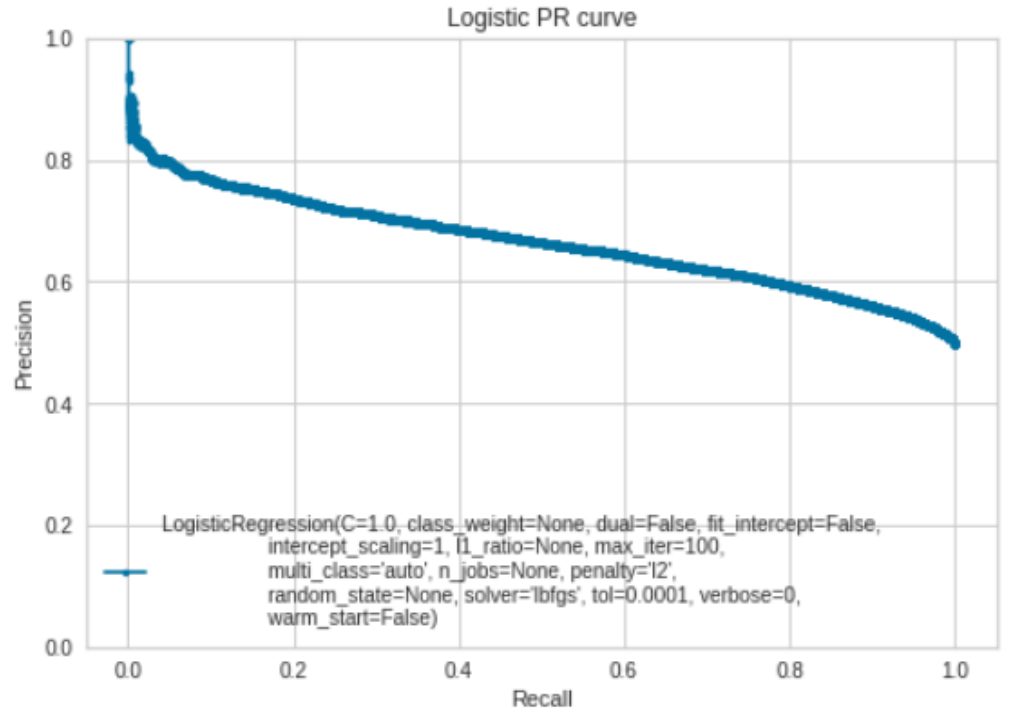
최종 비교 모델

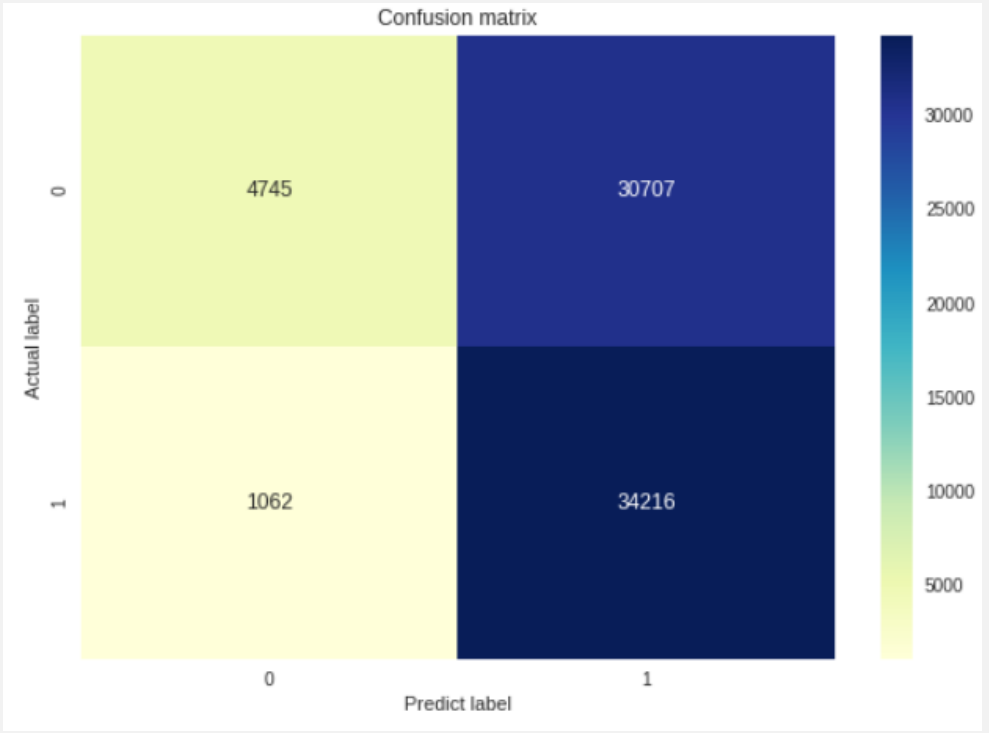
- Logistic
- Ridge
- Lasso
- Decision Tree
- Random Forest

현실성 고려(하드웨어적 문제) & 조별 토론

3. 모델 생성

Logistic

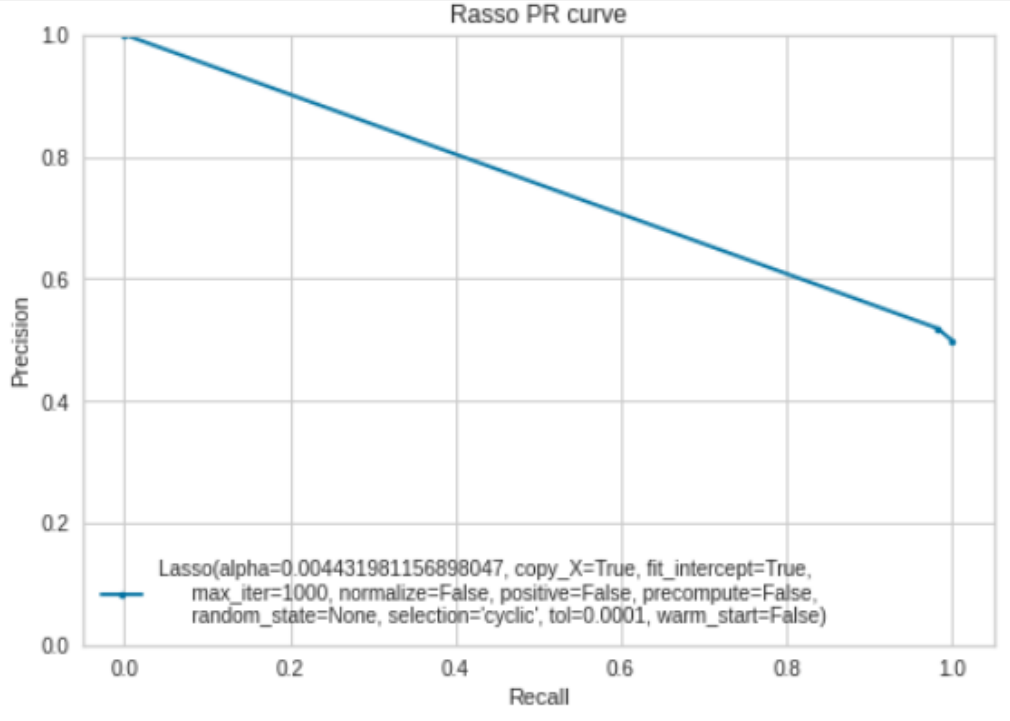


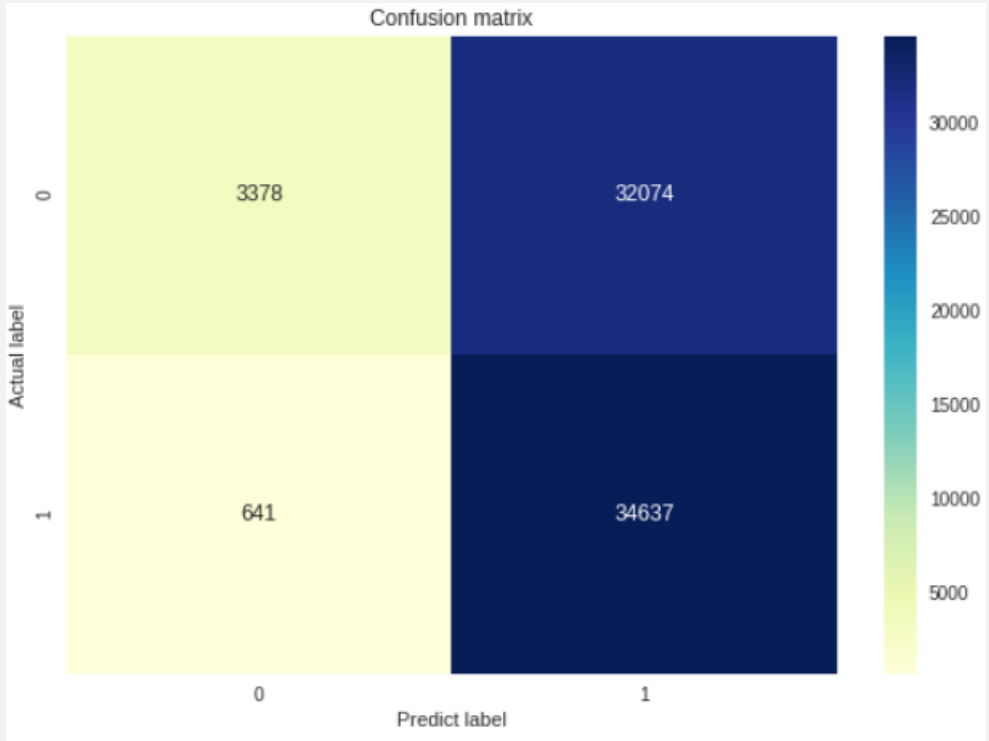


Accuracy score =0.553 Precision score = 0.53 Recall: 0.97

3. 모델 생성

Lasso

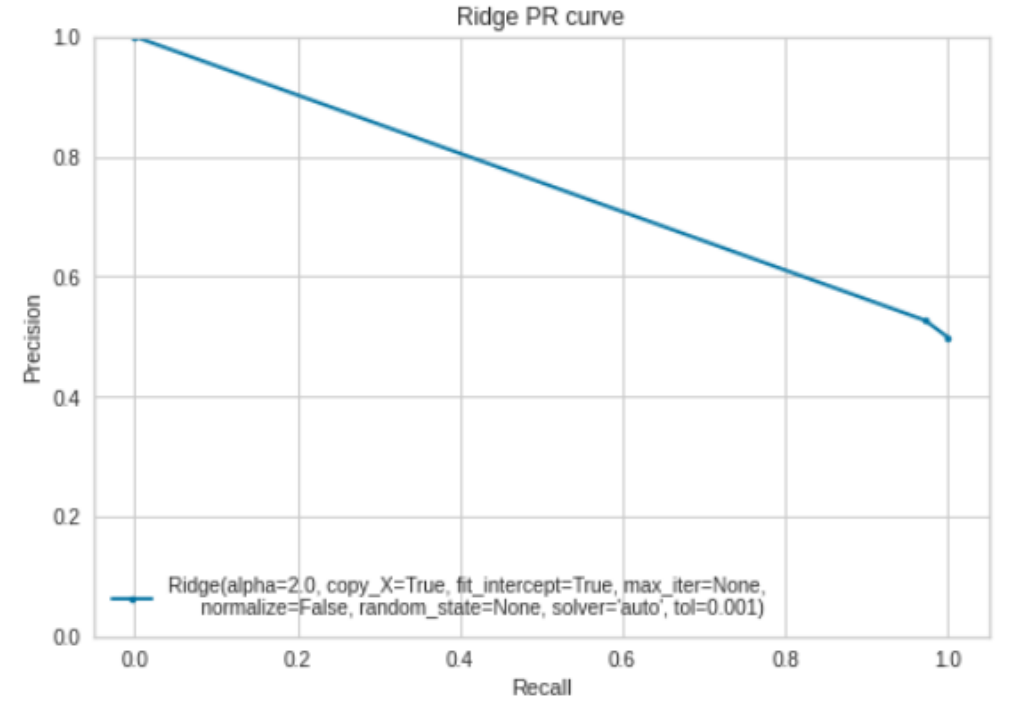


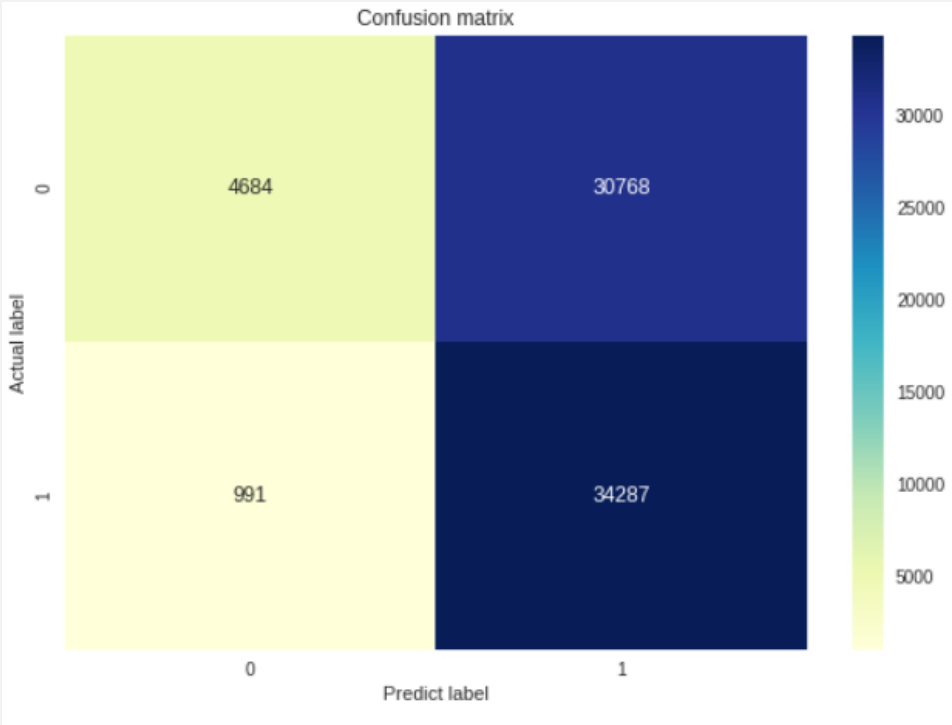


Accuracy score =0.537 Precision score = 0.519 Recall: 0.99

3. 모델 생성

Ridge

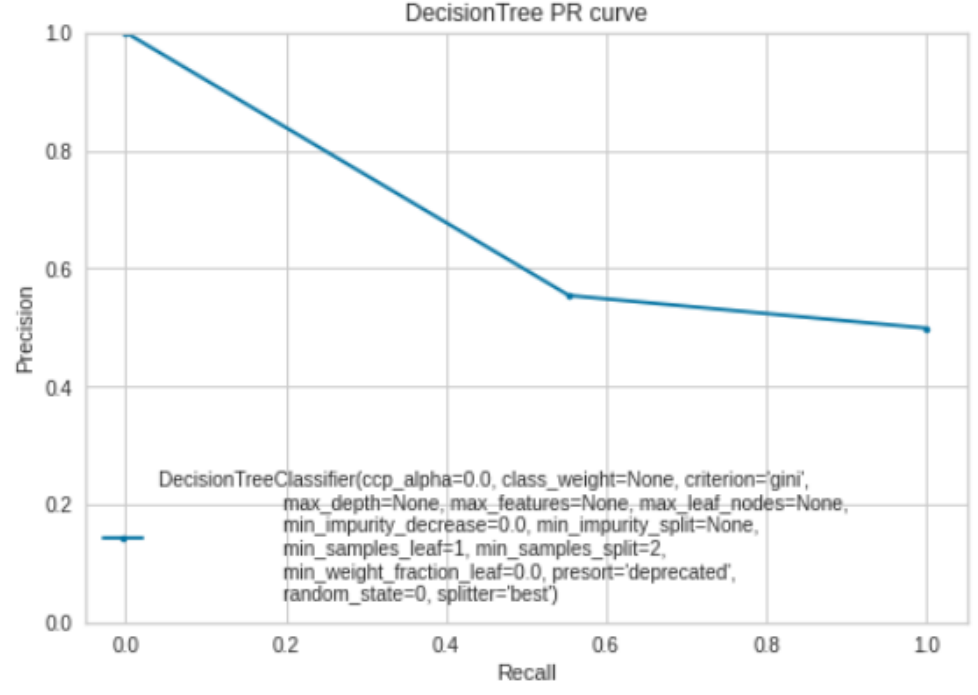


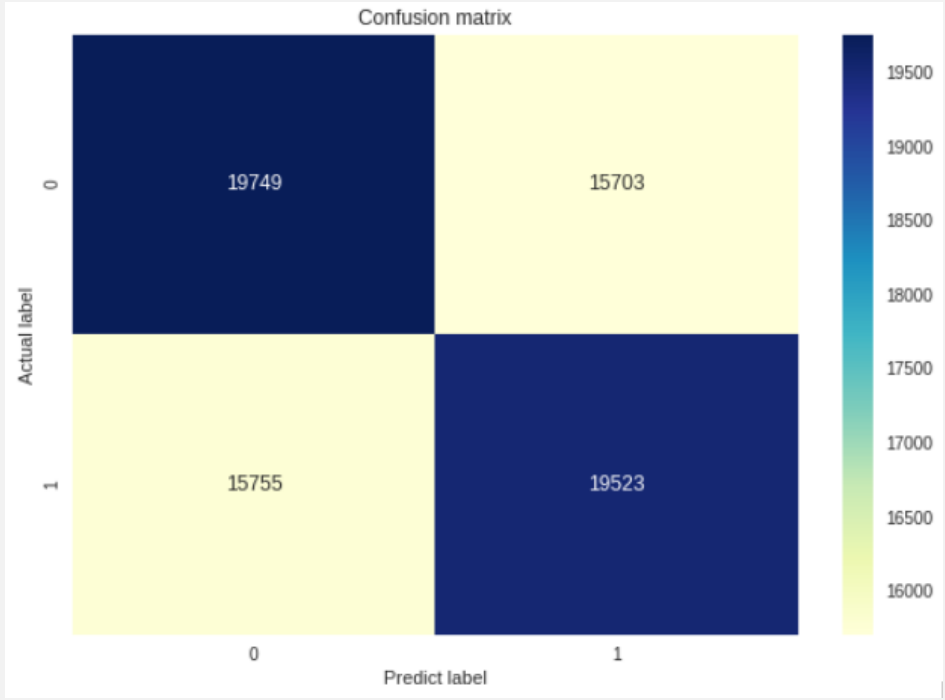


Accuracy score =0.551 Precision score = 0.532 recall: 0.972

3. 모델 생성

Decision Tree

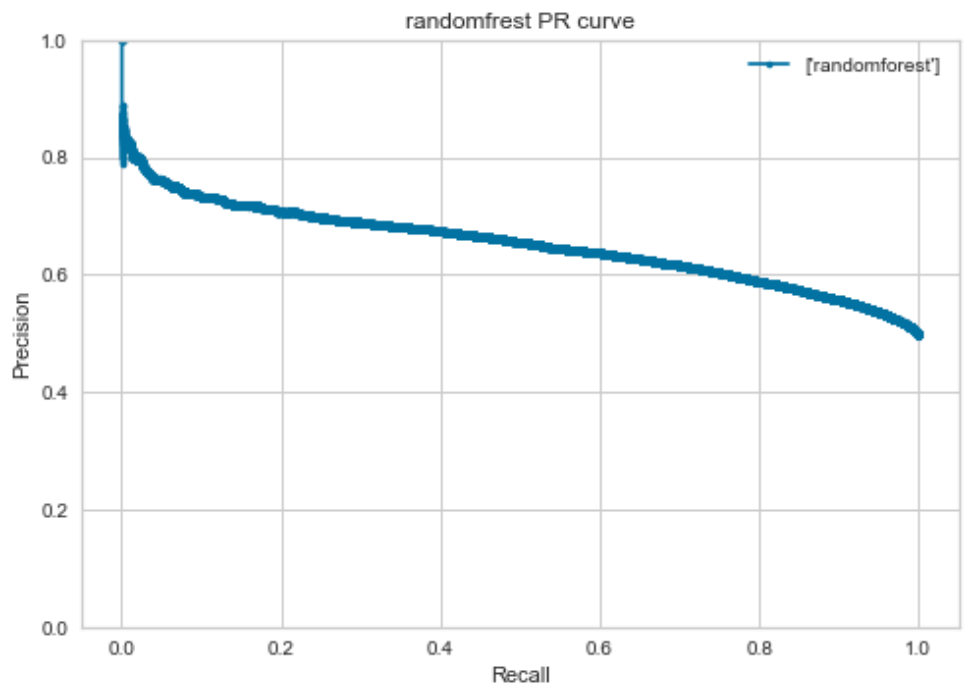


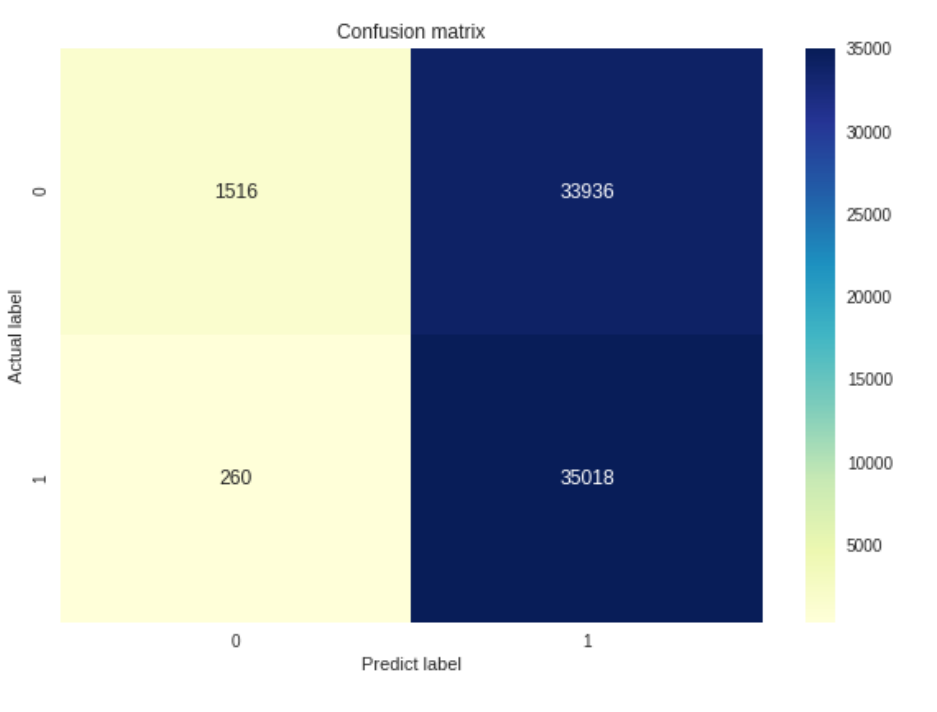


Accuracy score = 0.555 Precision score = 0.527 Recall= 0.945

3. 모델 생성

Random Forest





Accuracy score = 0.502 Precision score = 0.501 Recall: 0.993

4. 모델 비교

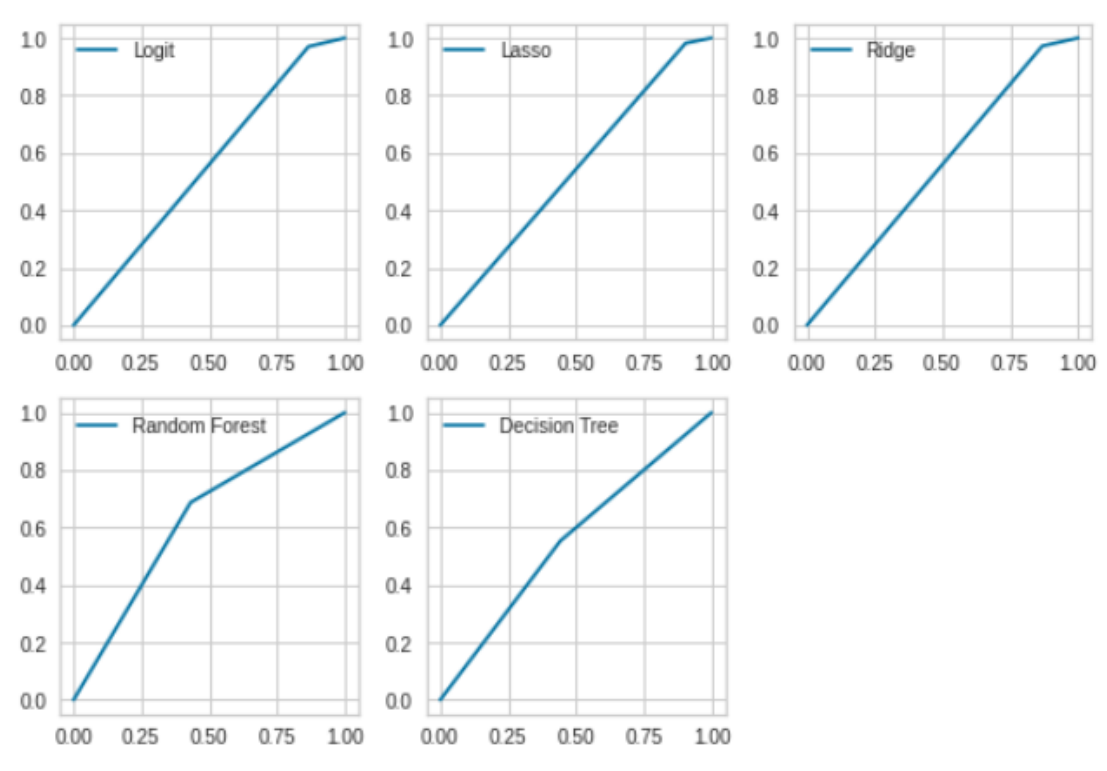
4. 모델 비교

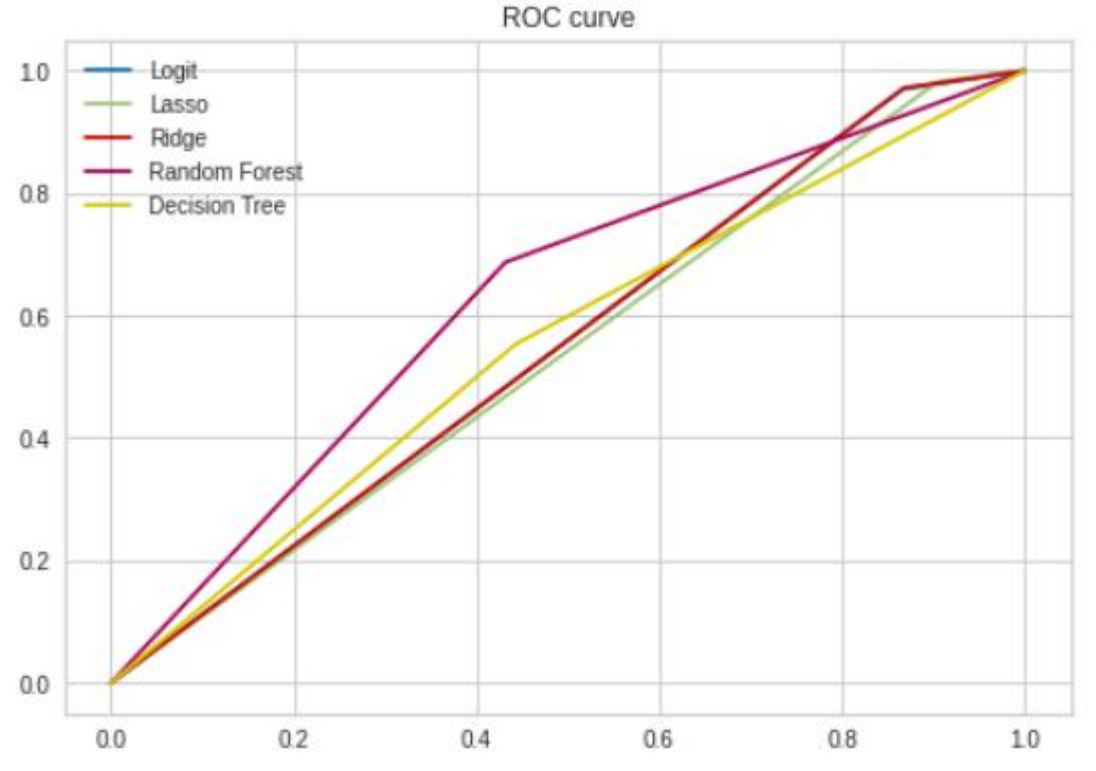
주요 지표 확인

모델명		Logit	Lasso	Ridge	Decision Tree	Random Forest
주요 지표 결과	accuracy	0.553	0.537	0.551	0.555	0.502
	Precision	0.53	0.519	0.532	0.527	0.501
	Recall	0.97	0.99	0.972	0.945	0.993
모델 평가	accuracy	높음	보통	높음	높음	낮음
	Precision	높음	보통	높음	높음	낮음
	Recall	높음	높음	높음	낮음	높음

4. 모델 비교

ROC curve 확인





5. 결론

5. 결론

투자 시 '부도 = 1'과 '부도 = 0' 에서 발생하는 투자자의 손실과 수익이 비대칭적.
기준식 [Eπ = (1 - P)R - P(1 - RR) – rf = 0] 에서는 이를 반영하여 p(=0.34의 가중치)에 (1-p)(=0.126의 가중치)보다 큰 가중치가 부여되어 0.26의 낮고 엄격한 threshold가 도출됨.
이에 따라, 전반적으로 모형들의 accuracy와 precision이 낮아지고 대신 recall이 아주 높아짐.
낮은 threshold를 통해 이러한 결과를 의도한 만큼 accuracy, precision, recall 값의 단순 비교로 모형 간의 우월성을 판단하기는 어렵지만, 타 모형 대비 accuracy, precision, recall 모두 준수한 결과를 보이는 Ridge를 최고의 모형으로 선정할 수 있을 것 같다.

5. 결론

이전 발표로부터의 개선 사항

- 0의 회수율(RR = 0)가정이 비현실적이라고 판단되어, RR=0.66으로 수정
  -> 지나치게 낮게 설정되었던 0.1정도의 threshold가 상방 조정됨.

- 후행변수 제거
  -> 지나치게 높은 설명력을 가진 후행변수를 제거함으로써 accuracy 정상화
  (accuracy : 0.91~0.92 => 0.5~0.55)

감사합니다