대출 목적에 따른 부도 예측

7조: 박보현, 안시완, 조성혜, 함태욱, 홍문기

2021.02.07

기존 보고서와 변경 사항

01.19 피드백

1. 데이터 셋 분할
   - 훈련 : 검증 : 테스트 = 6 : 2 : 2로 분할하여 파라메터를 학습 시킬 것.

2. 데이터 전처리
   - addr_state1-addr_state51 중요하지 않다고 판단해 제거한 주에 대한 변수는 보이지 않는 특성을 가지고 있을 수 있음.

3. 모델 적합 및 평가
   - 대출 목적에 대한 차이점이 부족함. 또한 모델의 AUC가 높은 이유는 Threshold가 매우 낮을 가능성이 있음. 확인 해야함.

02.07 변경 사항

1. 데이터 셋 분할
   - 훈련 : 검증 : 테스트 = 6 : 2 : 2로 변경하여 모델 학습을 다시 함.

2. 데이터 전처리
   - addr_state1-addr_state51 변수를 추가함.

3. 모델 적합 및 평가
   - 대출 목적에 대한 차이점에 대해 기대 수익 측면과 로지스틱의 Threshold에 대한 설명을 추가함.
   - 시간의 제약으로 파라메타를 추정하는데 한계가 있어 예측력을 높이지 못한 SVM과 다층 퍼셉트론을 제거함.

목차

1. 연구 목적

연구 목적

- 대출 목적에 따른 부도 여부 예측.

1. 부채 	2. 신용카드 	3. 집 	4. 자동차 	5. 사업 
6. 의료 	7. 휴가 	8. 결혼 	9. 기타  	

연구 수행 계획

〈STEP 1〉

- 연구 목적 고민

〈STEP 2〉

- 데이터 이해
- 데이터 전처리
- 데이터 탐색
- 데이터 수정

〈STEP 3〉

- 모형 구축
- 모형 평가

〈STEP 4〉

- 최종결론

2. 데이터 소개 및 전처리

데이터 소개

1. 데이터 소개: LendingClub 데이터는 1개의 Depvar(반응 변수)와 54개의 설명변수(원핫인코딩-333개)로 구성됨.

1	acc_now_delinq	19	inq_last_6mths	37	out_prncp_inv
2	addr_state1 - addr_state51	20	installment	38	pub_rec
3	annual_inc	21	int_rate	39	pub_rec_bankruptcies
4	chargeoff_within_12_mths	22	issue_d1 -issue_d118	40	Purpose1-purpose14
5	collection_recovery_fee	23	last_fico_range_high	41	recoveries
6	collections_12_mths_ex_med	24	last_fico_range_low	42	revol_bal
7	debt_settlement_flag1	25	last_pymnt_amnt	43	revol_util
8	delinq_2yrs	26	loan_amnt	44	tax_liens
9	mths_since_last_major_derog1- mths_since_last_major_derog11	27	mths_since_last_delinq1- mths_since_last_delinq11	45	mths_since_recent_inq1- mths_since_recent_inq10
10	mths_since_recent_bc_dlq1- mths_since_recent_bc_dlq11	28	mths_since_last_record1- mths_since_last_record11	46	mths_since_recent_bc1- mths_since_recent_bc11
11	mths_since_rcnt_il1-mths_since_rcnt_il11	29	mths_since_recent_revol_delinq1- mths_since_recent_revol_delinq11	47	verification_status1-verification_status3
12	emp_length1 - emp_length12	30	elapsed_t	48	total_acc
13	fico_range_high	31	Dti	49	total_pymnt
14	fico_range_low	32	term1	50	total_pymnt_inv
15	funded_amnt	33	tot_coll_amt	51	total_rec_int
16	funded_amnt_inv	34	delinq_amnt	52	total_rec_late_fee
17	home_ownership1 - home_ownership6	35	open_acc	53	total_rec_prncp
18	initial_list_status1 - initial_list_status2	36	out_prncp	54	tot_cur_bal

데이터 전처리

1. 변수 삭제

- 변수 설명이 없거나 설명이 불명확한 경우.

1	elapsed_t	변수 설명 없음.
2	collections_12_mths_ex_med	의료수집을 제외한 12개월 동안의 수집수.
3	initial_list_status1, initial_list_status2	대출의 초기 목록 상태(W, F).

- 사후 변수: 대출해주는 시점에 관찰이 불가능 하고, 대출 후 관찰 가능한 경우.

1	delinq_amnt	현재 채무불이행"인 채무자의 계좌 연체금액
2	total_pymnt	현재 까지 수령한 자금 총액
3	total_pymnt_inv	투자자가 자금을 지원한 총액의 일부에 대해 현재까지 수령한 지급액
4	total_rec_prncp	현재까지 수령한 원금
5	total_rec_late_fee	현재까지 수령한 연체료
6	total_rec_int	현재까지 받은 이자

데이터 전처리

1. 변수 삭제

- 대출해주는 시점에 관찰이 불가능 하고, 대출 후 관찰 가능한 경우.

7	tot_coll_amt	총징수액
8	recoveries	총 신용 회전 잔액
9	pub_rec_bankruptcies	공공 기록 파산 건수
10	issue_d1-issue_d118	대출 자금이 지원된 달
11	collection_recovery_fee	90일 이상 미지급된 세금 및 수수료 징수 비용
12	debt_settlement_flag1	채무 정산(제3자가 제공)
13	mths_since_last_delinq1-11	대출자가 마지막으로 연체된 이후 월
14	mths_since_last_major_derog1-11	가장 최근의 90일 또는 그 이하의 등급 이후 월
15	mths_since_last_record1	마지막 공개 기록 이후 월
16	mths_since_rcnt_il1	가장 최근 할부 계정이 개설된 이후 월
17	mths_since_recent_bc_dlq1	가장 최근의 은행 카드 연체 이후 월
18	mths_since_recent_bc1	가장 최근의 은행 카드 계좌가 개설된 지 몇 개월 후
19	mths_since_recent_inq1	최근 문의 몇 개월
20	mths_since_recent_revol_delinq1	최근 회전 연체 이후 몇 개월

데이터 전처리

1. 사후 변수 예시: 대출해주는 시점에 관찰이 불가능 하고, 대출 후 관찰 가능한 경우.

- collection_recovery_fee: 0보다 큰 경우 부도율이 100%. 이처럼 극단적인 값을 지니는 변수는 사후적으로 관찰될 수 있는 변수라는 것을 입증함.

- debt_settlement_flag1: 0일 때 , 부도율이 99.xx%으로 거의 100%에 가깝음.

depvar collection_recovery_fee	0.0	1.0	All	p
0.0	916095	42919	959014	4.475326
0.0036	0	1	1	100.000000
0.018	0	1	1	100.000000
0.0252	0	1	1	100.000000
0.036	0	1	1	100.000000
...	...	...	...	...
6184.2942	0	1	1	100.000000
6404.7384	0	1	1	100.000000
6584.1372	0	1	1	100.000000
6687.6228	0	1	1	100.000000
All	916095	176824	1092919	16.179058

collection_recovery_fee

depvar debt_settlement_flag1	0.0	1.0	All	p
0	9	25569	25578	99.964814
1	916086	151255	1067341	14.171197
All	916095	176824	1092919	16.179058

debt_settlement_flag1

데이터 전처리

2. 변수 그룹화

- 변수: verification_status(소득 확인 여부)

1	소득 확인 안됨
2	소득 출처 확인 됨
3	소득 확인

0	소득 확인 안됨
1	소득 출처 확인 됨, 소득 확인

데이터 전처리

2. 변수 그룹화

- 변수: home_ownership(집 소유)

1	ANY
2	대출
3	없음
4	OTHER
5	주인
6	임차인

1	대출
2	주인
3	임차인
4	집 없음
5	정보없음(ANY, OTHER)

데이터 전처리

2. 변수 그룹화

- 변수: emp_length(근무기간)

1	1년
3	2년
4	3년
⋮	⋮
10	9년
2	10년 이상
12	NA

1	1년 미만
2	1년 이상 5년 미만
3	5년 이상 10년 미만
4	10년 이상
5	정보없음(NA)

분석에 사용한 데이터

1. 데이터를 대출 목적에 따라 9개로 분할함.

- 데이터 셋은 각각 반응변수 1개, 설명변수 45개.
- 데이터 셋은 각각 훈련 데이터 60%, 검증 데이터 20%, 테스트 데이터 20%로 분할.
- 검증 데이터를 10개로 분할하여, 파라메터 추정에 사용함.

		훈련 데이터	검증 데이터	테스트 데이터	N
1	부채	377,946	125,983	125,983	629,912
2	신용카드	152,190	50,730	50,730	253,650
3	집	64,355	16,089	16,089	80,442
4	자동차	6,616	2,206	2,206	11,028
5	사업	6,804	2,269	2,269	11,341
6	의료	7,437	2,479	2,479	12,395
7	휴가	4,647	1,549	1,550	7,746
8	결혼	525	175	175	875
9	기타	68,420	17,105	17,105	85,526

3. 대출 목적별 모형 구축 및 평가

모델 평가 지표

- P2P 대출 예측의 Confusion matrix

	부도라고 예측 안함. depvar=0	부도라고 예측 함. depvar=1
실제로 부도가 일어나지 않음	TN	FP
실제로 부도가 일어남	FN	TP

- 주로 목적함수로 정확도 = $\frac{TN+TP}{TN+TP+FP+FN}$ 를 사용함.

- 하지만, 투자자들이 원하는 목적 함수는 P2P 투자에서 최대 이익을 얻는 것.

모델 평가 지표

- 기댓 수익값 목적 함수: $f(x) = X + Y + Z + W$

	부도라고 예측 안함. depvar=0	부도라고 예측 함. depvar=1
실제로 부도가 일어나지 않음	TN	FP
실제로 부도가 일어남	FN	TP

- N: TN + TP + FP + FN
- X(이자 받음): $\frac{TN}{N}$ ×(투자금 × 대출이자).
- Y(손해 없음): $\frac{TP}{N}$ × 0.
- Z(전액 손실): $\frac{FN}{N}$ × (-투자금).
- W(예측실패) : $\frac{FP}{N}$ × (-투자금*미국 기준 금리). → 돈을 가만히 둬서 이자율 상승만큼 손해
- 이때, 투자금은 10,000원이라 가정함.

모델 구축 과정

1. 로지스틱 모형

1.1 훈련 데이터: 훈련 데이터를 사용하여 모델을 적합함.

```
Logit_model_df1 = LogisticRegression(fit_intercept=False)
Logit_result_df1 = Logit_model_df1.fit(df1_x_train, df1_y_train)
```

1.2 검증 데이터: threshold를 선택하는데 사용함.

- threshold 값은 0.001 ~ 0.99까지 총 99개를 사용함.
- 검증 데이터를 랜덤하게 10개로 분할함. 10개의 검증 데이터로 각각 예측값을 구하고, Threshold 에 대한 기대수익값을 계산함.
- 각 Threshold 에 대한 10개의 검증데이터 기대수익값 평균을 구하여 ep_mean 변수를 생성함.
- ep_mean변수에서 가장 높은 기대수익값을 갖는 threshold를 선택함.

모델 구축 과정

1. 로지스틱 모형

1.2 검증 데이터: threshold를 선택하는데 사용함.

- 행은 threshold
- 열은 10개의 검정데이터와 ep_mean(10개의 행 평균), threshold

	0	1	2	3	4	5	6	7	8	9	ep_mean	threshold
0	-43.550296	-42.943793	-43.460076	-43.694422	-44.069814	-43.466023	-43.189756	-43.244873	-42.671073	-43.152401	-43.344253	0.00
1	507.102700	489.712159	484.532405	507.615626	503.245731	500.279243	488.876244	490.349721	486.781324	492.374599	495.086975	0.01
2	567.679138	537.179195	556.015268	568.601251	571.293256	555.609873	549.168606	554.488706	548.737796	545.469476	555.424256	0.02
3	615.296811	587.086507	608.574141	625.544839	618.175958	609.790896	597.634981	604.995712	600.347275	593.886617	606.133374	0.03
4	648.532455	623.259406	643.861047	660.533601	656.925243	638.847696	642.829679	645.765590	635.495630	629.554925	642.560527	0.04
...	...	...	...	...	...	...	...	...	...	...	...	...
95	-131.500292	-278.319977	-145.387784	-133.132507	-27.758227	-176.521714	-230.756138	-184.132296	-394.805125	-197.243462	-189.955752	0.95
96	-134.970645	-285.922783	-149.190065	-137.174544	-31.724907	-180.542623	-230.756138	-184.132296	-398.693150	-197.243462	-193.035061	0.96
97	-134.970645	-285.922783	-156.794628	-137.174544	-31.724907	-180.542623	-230.288756	-184.132296	-398.693150	-197.243462	-193.748779	0.97
98	-134.970645	-285.922783	-156.794628	-137.174544	-31.724907	-180.542623	-234.169237	-184.132296	-398.222832	-197.243462	-194.089796	0.98
99	-134.970645	-285.438246	-160.596909	-137.174544	-31.724907	-180.542623	-233.234471	-184.132296	-398.222832	-196.741092	-194.277857	0.99

1.3 테스트 데이터: 1.2에서 선택된 threshold를 이용하여 모델 평가 지표와 기대수익값을 예측함.

performance	accuracy	recall	precision	expect_price
threshold				
0.16	0.864006	0.928075	0.530236	806.144346

모델 구축 과정

2. 의사결정나무

2.1 훈련 데이터: 훈련 데이터를 사용하여 모델을 적합함.

```
model = DecisionTreeClassifier(criterion="entropy", max_depth = i, max_features=j)
result = model.fit(df1_x_train, df1_y_train)
```

2.2 검증 데이터: max_depth와 max_feature를 선택하는데 사용함.

- depth 값은 5 ~ 10 를 사용함.
- max_feature 값은 1~5를 사용함.
- 검증 데이터를 랜덤하게 10개로 분할함. 10개의 검증 데이터로 각각 예측값을 구하고, max_depth와 max_featur의 경우의 수에 대한 기대수익값을 계산함.
- 각 max_depth와 max_feature에 대한 10개의 검증데이터 기대수익값 평균을 구하여 ep_mean 변수를 생성함. ep_mean변수에서 가장 높은 기대수익값을 갖는 max_depth와 max_feature를 선택함.

모델 구축 과정

2. 의사결정나무

2.2 검증 데이터: max_depth와 max_feature를 선택하는데 사용함.

- 행은 max_depth와 max_feature
- 열은 10개의 검정데이터와 ep_mean(10개의 행 평균), max_depth와 max_feature

	0	1	2	3	4	5	6	7	8	9	ep_mean	depth	feature
	-428.913779	-481.096952	-513.363961	-446.416640	-337.200309	-371.481540	-492.164358	-354.761701	-557.049616	-435.183276	-368.074633	5	1
	-430.538717	-310.734156	-501.977321	-421.719962	-330.911000	-371.481540	-498.306864	-354.761701	-80.022690	-220.377969	-293.170000	5	2
	-428.913779	-327.861515	-104.245938	18.764961	260.056960	-371.481540	-498.306864	-354.761701	-351.904363	-432.193715	-215.496744	5	3
	-397.915639	-468.486611	-508.071166	-323.217514	135.418209	-353.233365	99.831839	-342.408396	-427.439833	-250.019578	-235.829248	5	4
	-422.714151	-410.429765	-453.541110	662.803179	-238.039902	-341.878411	-498.306864	514.477153	-557.049616	-63.710856	-150.049135	5	5
	-429.726248	-484.904694	-1.736825	-446.416640	-338.852913	-371.481540	-492.977997	-354.761701	-557.860374	-461.199110	-328.135174	6	1
	-327.070100	-484.904694	-513.363961	-337.024965	109.755785	-268.885710	-499.120503	-324.691717	-533.410496	-424.242652	-299.939184	6	2
	-197.178033	-444.098406	10.106970	-446.416640	-258.239320	-239.683821	-498.306864	-354.761701	-557.049616	-460.372193	-286.760915	6	3
	-428.913779	708.112714	-432.856426	-446.416640	-118.358965	279.418126	465.655120	255.057758	-190.547885	362.019980	38.637261	6	4
	-348.406512	-311.088794	-373.834623	-446.416640	-215.682687	172.344534	-145.948838	46.369849	-484.447163	212.540313	-157.156479	6	5
▶	-418.139460	-484.904694	-508.071166	-447.270722	-301.117064	10.157232	60.008420	-354.761701	103.542126	123.933594	-184.274500	7	1

2.3 테스트 데이터: 2.2에서 선택된 max_depth와 max_feature 를 이용하여 모델 평가 지표와 기대수익값을 예측함.

performance	accuracy	recall	precision	expect_price
1	0.87034	0.540507	0.645267	216.056463

모델 구축 과정

3. 랜덤포레스트

3.1 훈련 데이터: 훈련 데이터를 사용하여 모델을 적합함.

```
model = RandomForestClassifier(criterion="entropy", max_features='auto', n_estimators = i, max_depth=j)
result = model.fit(df1_x_train, df1_y_train)
```

3.2 검증 데이터: n_estimators와 max_depth을 선택하는데 사용함.

- n_estimators 값은 5 ~ 10 를 사용함.
- max_depth 값은 5~10를 사용함.
- 검증 데이터를 랜덤하게 10개로 분할함. 10개의 검증 데이터로 각각 예측값을 구하고, n_estimators와 max_depth의 경우의 수에 대한 기대수익값을 계산함.
- 각 n_estimators와 max_depth 에 대한 10개의 검증데이터 기대수익값 평균을 구하여 ep_mean 변수를 생성함. ep_mean변수에서 가장 높은 기대수익값을 갖는 n_estimators와 max_depth 를 선택함.

모델 구축 과정

3. 랜덤포레스트

3.2 검증 데이터: n_estimators와 max_depth을 선택하는데 사용함.

- 행은 n_estimators와 max_depth
- 열은 10개의 검정데이터와 ep_mean(10개의 행 평균), n_estimators와 max_depth

	0	1	2	3	4	5	6	7	8	9	ep_mean	n_estimators	max_depth
0	33.119146	198.939974	-716.222559	-598.815259	-511.476436	-503.037582	-474.940972	259.722906	555.184116	160.877635	-159.664903	5	5
1	-559.458616	-142.326545	519.951650	-303.846764	-206.720200	-132.226074	411.165723	319.598228	-2.858010	61.835364	-3.488524	5	6
2	-189.191873	-77.037629	108.819898	307.589949	291.904238	535.364734	-116.938509	233.158815	478.064131	305.432372	187.716613	5	7
3	451.165258	181.280744	59.492933	384.635482	397.998925	425.271256	480.381866	568.338310	462.347038	429.829049	384.074086	5	8
4	456.226856	446.863436	475.890530	522.154375	510.298349	373.717856	485.826500	546.599258	403.853760	257.266027	447.869695	5	9
5	399.794410	549.248280	542.633516	442.180235	604.815304	460.312798	546.644048	515.210129	633.769578	500.309978	519.491828	5	10
6	12.256057	-394.823685	-294.030150	-280.431867	144.080822	-381.433545	83.824176	24.503483	318.472084	-214.468922	-98.205155	6	5

3.3 테스트 데이터: 3.2에서 선택된 n_estimators와 max_depth 를 이용하여 모델 평가 지표와 기대수익값을 예측함.

performance	accuracy	recall	precision	expect_price
1	0.92172	0.734494	0.793366	573.538576

대출 목적 별 모델 평가

1. 대출 목적: 부채

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.18	0.87	0.94	0.58	816.44
의사결정나무	max_depth 10 max_feature 5	0.87	0.54	0.65	216.06
랜덤포레스트	n_estimators 8 max_depth 10	0.92	0.73	0.79	573.54

- 랜덤포레스트의 정확도가 로지스틱 보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

대출 목적 별 모델 평가

2. 대출 목적: 신용카드

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.14	0.88	0.94	0.53	749.52
의사결정나무	max_depth 10	0.89	0.51	0.34	257.52
	max_feature 5				
랜덤포레스트	n_estimators 8	0.93	0.69	0.79	514.12
	max_depth 10				

- 랜덤포레스트의 정확도가 로지스틱 보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

대출 목적 별 모델 평가

3. 대출 목적: 집

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.16	0.86	0.93	0.53	806.14
의사결정나무	max_depth 8	0.87	0.52	0.59	274.43
	max_feature 5				
랜덤포레스트	n_estimators 8	0.92	0.62	0.79	470.94
	max_depth 10				

- 랜덤포레스트의 정확도가 로지스틱 보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

대출 목적 별 모델 평가

4. 대출 목적: 자동차

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.13	0.88	0.95	0.51	839.03
의사결정나무	max_depth 7	0.90	0.63	0.59	504.70
	max_feature 5				
랜덤포레스트	n_estimators 10	0.92	0.57	0.77	466.16
	max_depth 8				

- 랜덤포레스트의 정확도가 높지만, 재현율이 낮아 손해가 심해 기대수익값이 제일 낮음.

기대수익값 : 로지스틱 > 의사결정나무 > 랜덤포레스트

대출 목적 별 모델 평가

5. 대출 목적: 사업

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.19	0.83	0.94	0.59	815.29
의사결정나무	max_depth 10	0.85	0.69	0.68	352.29
	max_feature 4				
랜덤포레스트	n_estimators 10	0.89	0.79	0.77	622.53
	max_depth 8				

- 랜덤포레스트의 정확도가 로지스틱보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

대출 목적 별 모델 평가

6. 대출 목적: 의료

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.21	0.86	0.89	0.57	754.28
의사결정나무	max_depth 7	0.89	0.74	0.69	561.87
	max_feature 5				
랜덤포레스트	n_estimators 10	0.89	0.52	0.79	208.32
	max_depth 10				

- 랜덤포레스트와 의사결정나무의 정확도가 같지만, 랜덤포레스트의 재현율이 낮아 기대수익값이 낮음.

기대수익값 : 로지스틱 > 의사결정나무 > 랜덤포레스트

대출 목적 별 모델 평가

7. 대출 목적: 휴가

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.16	0.86	0.93	0.54	843.52
의사결정나무	max_depth 9	0.87	0.56	0.60	322.87
	max_feature 5				
랜덤포레스트	n_estimators 10	0.88	0.44	0.77	195.51
	max_depth 8				

- 랜덤포레스트의 정확도가 높지만, 재현율이 낮아 손해가 심해 기대수익값이 제일 낮음.

기대수익값 : 로지스틱 > 의사결정나무 > 랜덤포레스트

대출 목적 별 모델 평가

8. 대출 목적: 결혼

	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.13	0.83	0.88	0.46	1050.27
의사결정나무	max_depth 8	0.83	0.36	0.41	439.99
	max_feature 5				
랜덤포레스트	n_estimators 6	0.87	0.40	0.59	558.19
	max_depth 8				

- 랜덤포레스트의 정확도가 로지스틱보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

대출 목적 별 모델 평가

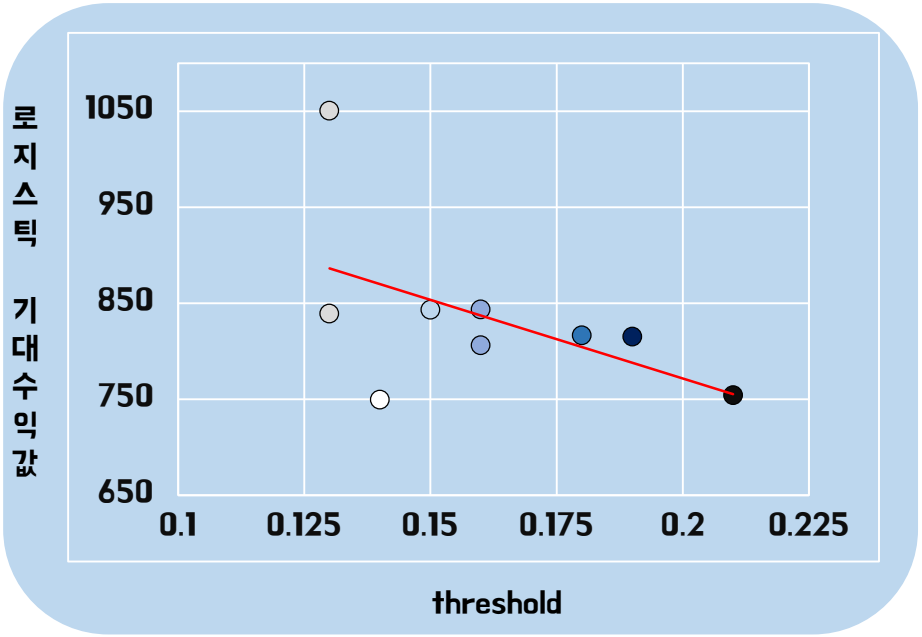
9. 대출 목적: 기타

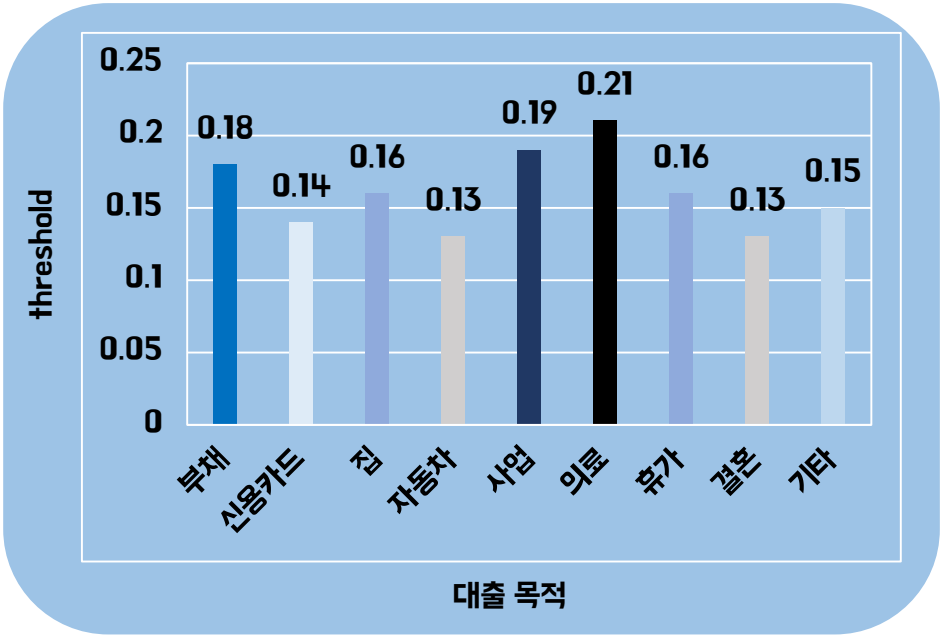
	파라메터	정확도 (accuracy)	재현율 (recall)	정밀도 (precision)	기대수익값
로지스틱	threshold 0.15	0.85	0.93	0.52	843.09
의사결정나무	max_depth 10	0.85	0.47	0.57	188.47
	max_feature 5				
랜덤포레스트	n_estimators 10	0.91	0.69	0.77	585.82
	max_depth 10				

- 랜덤포레스트의 정확도가 로지스틱보다 높지만, 재현율이 낮아 손해가 심해 기대수익값이 낮음.

기대수익값 : 로지스틱 > 랜덤포레스트 > 의사결정나무

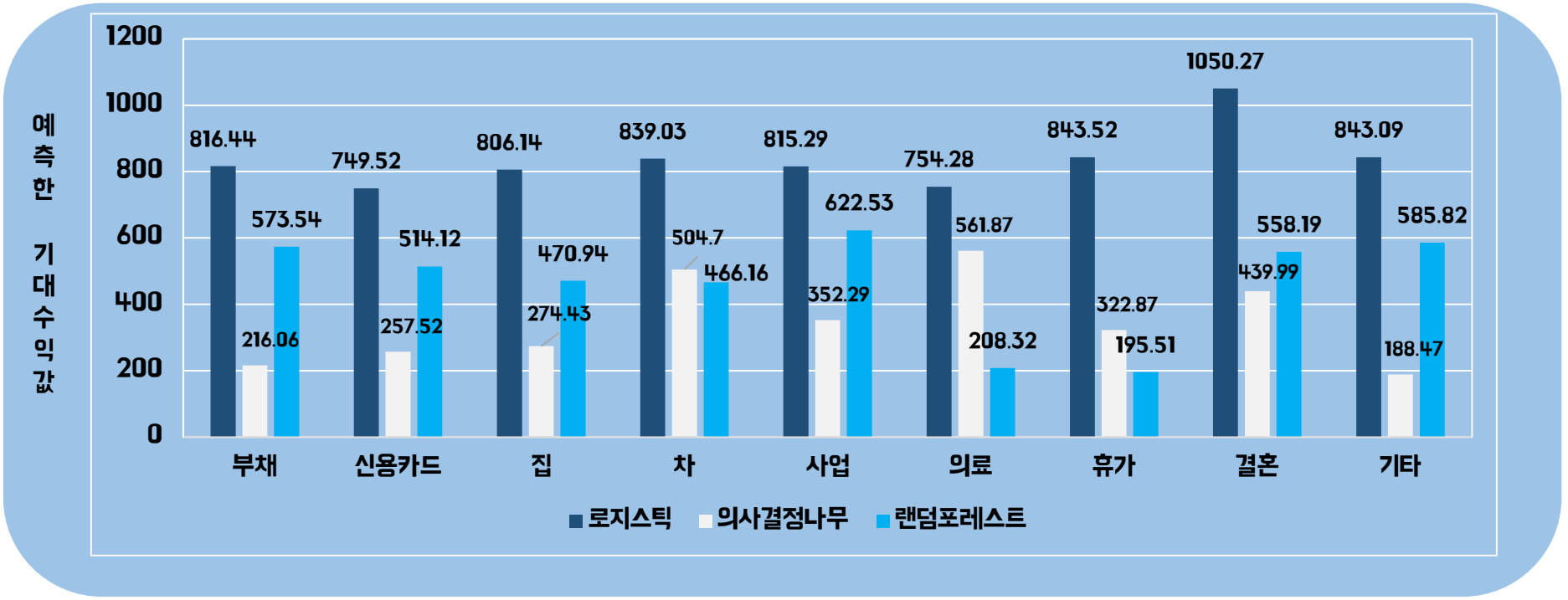
대출 목적 별 threshold와 기대수익 비교





- Threshold가 높아질 수 록 로지스틱이 예측한 기대수익값이 감소하는 추세를 보이고 있음.
- Threshold가 높을 때 예측된 기대수익값이 감소하기 때문에, 기대수익을 극대화하기 위해서는 대출태도를 보수적으로 가져야함. 즉, 위험성이 높음을 의미함.
- 따라서, P2P 업체가 더 많은 대부자를 플랫폼으로 유도하기 위해서는, 위험성이 낮고(threshold가 낮고), 기대수익이 높은 목적을 중심으로 대출을 구성하는 것이 바람직함.
- 특이하게 신용카드 목적의 대출은 threshold가 낮은데 기대수익은 낮은 것으로 나타났음.

대출 목적 별 기대 수익값 비교



- 9개의 대출 목적에 대해서 각각 10,000원을 투자했다고 가정하였을때,
  - 대출 목적이 결혼이고 로지스틱으로 부도를 예측했을 때, 예측 된 추가 기대수익값이 1,050원으로 가장 큰 기대 수익값을 갖음.
  - 대출 목적이 기타이고 의사결정나무로 부도를 예측했을 때, 예측 된 추가 기대수익값이 188.47원으로 가장 적은 기대 수익값을 갖음.
- 앞서, 위험성을 판단할 수 있는 threshold와 예측된 기대수익값 을 기준으로 판단할때,
  - 우선적으로 투자할 대출 목적은 결혼, 자동차, 휴가가 있음.
  - 투자를 고려해야할 대출 목적은 의료, 사업, 신용카드가 있음.

6. 결론

결론

- 연구목적: 대출 목적에 따른 부도 여부 예측.
- 데이터 전처리: 변수 삭제(사후 변수, 설명 부족),
  범주형 변수 그룹화.
- 데이터 분석 : 9개의 대출 목적에 따라 분석을 진행함.
- 모형 구축 및 평가:
  - 3개의 모형(로지스틱, 의사결정나무, 랜덤포레스트)을 적합함.
  - 모델 평가지표는 정확도, 재현률, 정밀도를 구함.
  - 기댓수익값을 예측함.

결론

- 대출 목적별 threshold와 기대수익값 비교:

  - Threshold가 높아질 수 록 로지스틱이 예측한 기대수익값이 감소하는 추세가 있음.
  - Threshold가 높을 때 예측된 기대수익값이 감소하기 때문에, 위험성이 높음을 의미함.
  - 따라서, P2P 업체가 더 많은 대부자를 플랫폼으로 유도하기 위해서는 위험성이 낮고(threshold가 낮고), 기대수익이 높은 목적을 중심으로 대출을 구성하는 것이 바람직함.
  - 따라서,
    - ✓ 우선적으로 투자할 대출 목적은:

      "결혼", "자동차", "휴가"

    - ✓ 투자를 고려해야할 대출 목적:

      "의료", "사업", "신용카드"