샘플링 비율 탐색과
불균형 데이터 전처리 방법을 활용한
부스팅 알고리즘 모형 비교

- 기업 부도 여부 예측 중심으로-

비즈니스 인포매틱스 석사2기
2021155151 이준희

목차

1장 서론

2장 이론적 배경

3장 연구 방법론

4장 연구 결과

5장 결론

부록

연구 배경

- 분류 알고리즘 → 데이터의 클래스(레이블)에 속한 데이터가 균등하게 분포되어 있다는 가정 하에 학습과 평가가 진행

- 불균형 데이터 → 다수 클래스에 가중치를 둔 학습으로 인한 왜곡된 결과 및 잘못된 판단 유발
  ex.) 높은 정확도, But 극히 불균형한 재현율과 정밀도

- 해결 방안 → 표본 수준의 접근방법
  ex.) 오버샘플링(OverSampling),언더샘플링(UnderSampling)
  오버+언더샘플링(Over+UnderSampling)

연구 목적

- 기업의 부도 여부 예측 중심으로 표본 수준의 접근방법(오버샘플링, 오버+언더샘플링) 사용
- 샘플링 비율 탐색 과정(0.7~1.0, 0.5단위 증가)
- 트리 계열의 부스팅 알고리즘(Gradient Boosting Machine, XGBoost, Light GBM) 사용
- 불균형 데이터에 경건한 평가지표로 종합적 판단(Balanced Accuracy, G-Mean, Matthew's Correlation coefficient)
- 변수 중요도 탐색으로 부도 예측에 있어서 변수들의 영향력 비교

연구 구성

- 이론적 배경 → 연구를 진행하기 위한 주요 개념 및 방법론 서술
  [불균형 데이터 전처리 기법, 부스팅 알고리즘, 평가지표]

- 연구 방법론 → 이론적 배경에서 서술된 개념의 활용 방식
  연구에 사용될 데이터 소개
  연구 가설 설정 및 연구 모형 설계

- 연구 결과 → 연구 방법론 기반의 결과 및 연구를 통한 발견점 나열

- 결론 → 연구 요약
  연구의 한계점 및 향후 연구 방향 서술

불균형 데이터

- 정의 → 다수 클래스(Majority Class) > 소수 클래스(Minority Class)
- IR → Imbalanced Ratio. 비율이 1을 초과할 경우 클래스의 불균형 판단
  (일반적으로 IR 10 초과 시 불균형이 심한 것으로 판정)

$$IR = \frac{n^+}{n^-}$$

n^+ : 다수 범주의 개수

n^- : 소수 범주의 개수

- 성능의 영향 → 균형 데이터에 초점이 맞춰진 분류 알고리즘의 성능 왜곡
- 불균형 데이터 예시 → ex.)기업의 부도 여부, 신용카드 부정거래, 보험 사기
- 현실 데이터의 특징 → 일반적으로 소수 클래스의 비율은 매우 적음
- 본 연구의 문제 → 소수 클래스를 얼마나 제대로 분류할 수 있는지에 대한 탐색

불균형 데이터 전처리 기법1. OverSampling

SMOTE(Synthetic Minority OverSampling)

- KNN(K-Nearest Neighbor) 원리로 소수 클래스 데이터의 특성 공간에서 합성데이터 생성
  1. 소수 클래스의 임의의 하나의 데이터와 K최근접 이웃의 데이터 선택 & Euclidean Distance 이용
  2. Euclidean Distance에 난수 곱셈 & 기존에 임의로 선택된 소수 클래스의 데이터와 더함
  3.다수 클래스와의 비율이 맞춰질 때까지 이 과정을 반복

- 단순한 복원추출로 인한 데이터 생성이 아니므로 과적합 문제 완화
- 주변 데이터의 고려 없는 진행으로 추가적인 노이즈와 일반화 발생 위험

ADASYN(Adaptive Synthetic Sample Approach)

- SMOTE의 노이즈 및 일반화 위험의 단점을 보완
  1. KNN의 공간특성이 아닌, 해당 데이터의 밀도 기반으로 합성 데이터 생성
  2. 밀도가 적을수록 합성데이터는 많이, 밀도가 높을수록 합성데이터는 적게 생성

- 밀도 기반의 데이터 생성으로 SMOTE보다 체계적인 샘플링 가능

불균형 데이터 전처리 기법1. OverSampling

ROSE(Random OverSampling)

- 가장 단순한 오버샘플링 기법

  1. 소수 클래스에 대한 무작위 복원 추출을 통한 다수 클래스와의 비율 일치

- 데이터의 복제로 인한 중복데이터의 생성 및 검증데이터에서의 과적합 발생

Borderline - SMOTE

- SMOTE의 노이즈 및 일반화 위험의 단점을 보완

  1. 다수&소수 클래스를 구분하는 경계에서 소수 데이터를 기준점으로 선택

  2 선택된 기준점에서 합성데이터 생성

- 상대적으로 구분하기 어려운 경계선 상의 데이터의 분류가 제대로 되면, 다른 데이터 역시 클래스 간 오분류가 낮을 것이라는 아이디에서 비롯된 기법

불균형 데이터 전처리 기법1. OverSampling

SVM - SMOTE

- Borderline - SMOTE의 변형된 방법
  1. Borderline SMOTE에서 기존의 KNN이 아닌 SVM(Support Vector Machine)을 이용
  2. 클래스 간 경계선은 훈련된 데이터의 Support Vector로 예측
  3. Support Vector와 가까운 소수 클래스의 데이터가 합성데이터 생성의 기준 역할
  4. 소수 클래스의 Support Vector와 K개의 근접 이웃들을 연결하는 선에 따라 합성데이터 무작위 생성

- BorderLine – SMOTE에 비해서 클래스 간 경계선에 더 높은 주의를 기울인 기법

불균형 데이터 전처리 기법2. Over+UnderSampling

SMOTEENN(SMOTE + ENN)

- SMOTE(오버샘플링) + ENN(언더샘플링)
  1. SMOTE로 오버샘플링한 데이터에 대한 추가적인 필터링 작업을 수행

- ENN → 1. 각 데이터에 대한 K최근접 이웃의 데이터 집합의 과반수 이상의 클래스가 기존의 데이터의 클래스와 일치하는지 확인.
  2. 클래스가 다를 시, 관측된 데이터 & K최근접 이웃 데이터 삭제

SMOTETomek(SMOTE + Tomek Link)

- SMOTE(오버샘플링) + Tomek Link(언더샘플링)
  1. SMOTE에 Tomek Link를 추가해서 소수&다수 클래스 간 구분하기 어려운 특정 데이터 삭제

- Tomek Link → 1. Euclidean Distance를 활용한 규칙으로 언더샘플링 진행
  2. 다수 클래스에 속한 무작위의 데이터 집단과 소수 클래스와의 데이터와 구분이 모호하면 해당 다수 클래스의 데이터 삭제

부스팅 기반 알고리즘

Gradient Boosting Machine(GBM)

- 부스팅(순차적인 오류 가중치의 부여) + 경사하강법(비용함수의 최적화)
- 잔차를 직접 예측하고 앞선 모델이 예측하지 못한 차이를 추가모델에서 보상하는 구조
- 노이즈까지 모델링 & 과적합 위험(해결책→ Subsampling, Shrinkage, Early Stopping)

XG-Boost

- GBM의 수행 시간 및 과적합 문제 해결을 위한 기법(오차의 가중치 처리를 병렬 수행으로 진행)
- 자체적인 과적합 규제, 나무 가지치기, 내장된 교차검증, 결손값 처리 기능 보유

Light GBM

- GBM의 모든 데이터와 특성 변수 사용의 비효율성 해결을 위한(GOSS[데이터 제거], EFB[특성 변수 제거])
- XG-Boost보다 빠른 학습 시간
- Leaf 중심의 비대칭 규칙 트리의 생성으로 기존의 Tree분할 방식보다 예측 오류의 손실 최소화

평가지표

		예측 범주	
		Positive	Negative
실제 범주	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

<표 2-5>

** TP: 예측과 실제가 모두 Positive FP : 예측은 Positive, 실제는 Negative

TN: 예측과 실제가 모두 Negative FN: 예측은 Negative, 실제는 Positive

정확도(Accuracy) = (TP + FP) / (TP + FP + TN + FN)

정밀도(Precision) = TP / (TP + FP)

재현율(Recall) = TP / (TP + FN)

*F1-Score = (2*Precision*Recall) / (Precision + Recall)*

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Balanced Accuracy = (Sensitivity * Specificity) / 2

$$G - Mean = \sqrt{Sensitivity \times Specificity}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

평가지표

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

- 기존의 F1-Score에서 정밀도(*Beta*=0.5), *또는 재현율(Beta*=2.0)에 더 큰 가중치의 비중을 둘 경우의 평가지표

  ex.) 보험사기 이용자를 정상 고객으로 분류하는 FN을 낮춰야 되는 문제 → Recall(재현율)에 더 비중을 두는 Beta값 사용

$$Balanced\ Accuracy = (Sensitivity * Specificity) / 2$$

- 민감도(TPR)와 특이도(TNR)의 평균으로 다수&소수 클래스의 평균 정확도 측정 (TPR & TNR 모두 고려)
- 기존의 불균형 데이터의 정확도보다 낮은 수치(불균형의 왜곡 완화)

$$G - Mean = \sqrt{Sensitivity \times Specificity}$$

- 민감도(TPR)와 특이도(TNR)의 기하평균(Geometric Mean)으로 다수&소수 클래스데 대한 성능의 균형 측정
- 높을수록 소수 클래스에 대한 분류 성능이 좋음 (TPR & TNR 모두 고려)
- 다수 클래스에 대한 과적합, 소수 클래스에 대한 과소적합 피하기 위한 판단의 역할

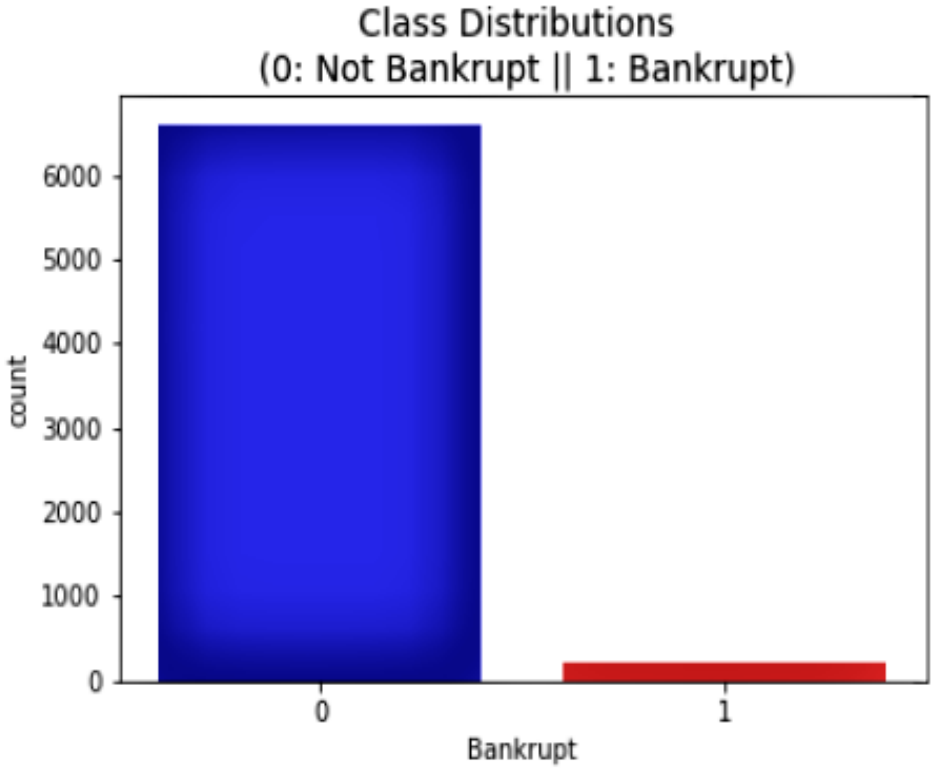
$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$ (Matthew's Correlation Coefficient)

- 오차행렬의 모든 요소를 사용한 불균형으로부터 가장 영향을 덜 받는 평가지표 중 하나
- MCC의 범위는 피어선 상관계수와 같이 -1에서 +1(-1은 최악의 성능, 0은 랜덤 예측의 성능, +1은 최고 성능)

데이터 탐색 및 이해

데이터

- 데이터 수집 →Company Bankruptcy Prediction (Kaggle)
- 데이터 구성 → [6819행,96열], 종속변수['Bankrupt'] & 95개의 재무변수(이익&재무위험 특징)



- ✓ **Class=0 (Negative) : 6599(96.77%)**
- ✓ **Class=1 (Positive) : 220(2.23%)**
- ✓ **IR(Imbalanced Ratio) : 29.99(약 30)**

연구 가설 설정

1. H0: Lasso Penalty 변수선택법은 모형의 성능을 헤친다.

   H1: Lasso Penalty 변수선택법은 기존의 성능과 비슷하거나 성능을 향상시킨다.

*Lasso Penalty(L1 Penalty) : 중요도가 낮은 변수의 영향력을 0으로 만드는 규제 기법

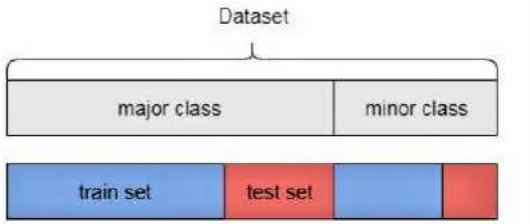
2. H0: 샘플링 비율이 높을수록 모형의 성능이 증가한다.

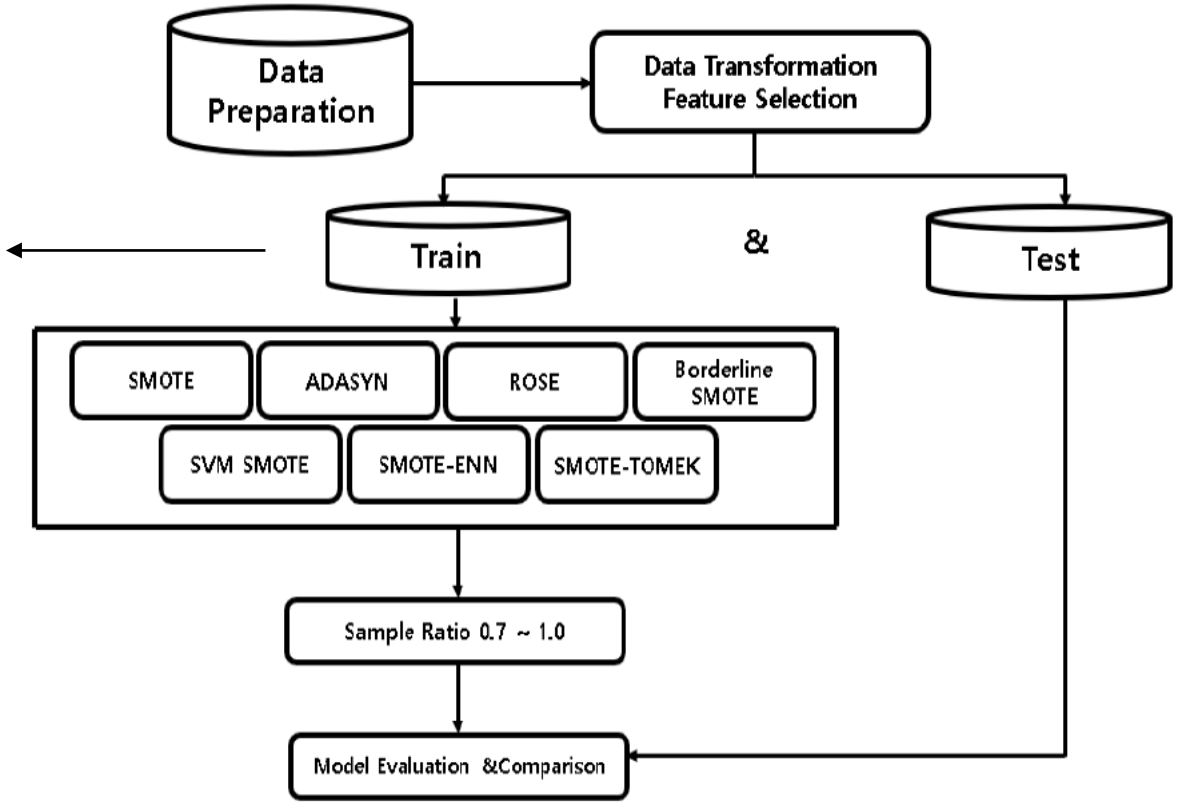
   H1: 사용하는 모델과 샘플링 비율에 따라 과적합 발생 및 성능 개선이 나타나는 지점이 각각 다르다.

3. H0: 통계적으로 유의한 변수들은 모형의 성능에 있어서 크게 기여하지 않은 변수들이다.

   H1: 통계적으로 유의한 변수들이 모형의 설명력과 성능에 중요한 영향력을 끼친다.

연구 모형 설계





- 데이터 특징 파악
- Stratified Train/Test Split
- 샘플링 방법론 & 부스팅 알고리즘
- 설정된 가설 중심의 진행

연구 결과

1. 변수선택법에 따른 모형별 분류 성능 비교

   [93 or 34 variables ?]

2. 샘플링 비율에 따른 모형별 분류 성능 비교

   [Performance 1.0 > 0.95 > 0.9 > 0.85 > 0.8 > 0.75 > 0.7 ?]

   [Top 10 Models]

3. 설명 변수의 중요도 분석

   [P-value < 0.05 variables = Top Important Features ?]

1. 변수선택법에 따른 모형별 분류 성능 비교

- L1(Lasso) Penalty alpha =0.1-

ROA(C) before interest and depreciation before interest	ROA(B) before interest and depreciation after tax	Pre-tax net Interest Rate	Operating Expense Rate	Net Value Per Share (B)
Persistent EPS in the Last Four Seasons	Regular Net Profit Growth Rate	Total Asset Growth Rate	Cash Reinvestment %	Quick Ratio
Total debt/Total net worth	Debt ratio %	Net worth/Assets	Borrowing dependency	Inventory and accounts receivable/Net value
Total Asset Turnover	Accounts Receivable Turnover	Fixed Assets Turnover Frequency	Net Worth Turnover Rate (times)	Revenue per person
Operating profit per person	Cash/Total Assets	Cash/Current Liability	Current Liability to Assets	Inventory/Working Capital
Working Capital/Equity	Total expense/Assets	Quick Asset Turnover Rate	Cash Turnover Rate	Fixed Assets to Assets
Net Income to Total Assets	Total assets to GNP price	No-credit Interval	Degree of Financial Leverage (DFL)	

- Net Income Flag, Liability Assets Flag 제거(동일 값 아닌 연속형 변수만 보기) → 기존 변수 93개
- Logistic Regression으로 Lasso 진행 → 규제 목적 한정 데이터 표준화 진행. [' C ' : 0.1]
- 93개에서 최종 34개의 변수 남음

1. 변수선택법에 따른 모형별 분류 성능 비교

- L1(Lasso) Penalty alpha =0.1-

Lasso 적용 X

SMOTE TOMEK

샘플링 비율 =1.0

	Accuracy	Precision	Recall	AUC	F1	F2	Bal_Acc	G-Mean	MCC
GBM	0.934	0.284	0.697	0.927	0.404	0.540	0.819	0.810	0.418
LightGBM	0.960	0.407	0.530	0.930	0.461	0.500	0.752	0.719	0.444
XGBoost	0.960	0.418	0.576	0.929	0.484	0.535	0.774	0.749	0.471

Lasso 적용 이후

SMOTE TOMEK

샘플링 비율 =1.0

	Accuracy	Precision	Recall	AUC	F1	F2	Bal_Acc	G-Mean	MCC
GBM	0.923	0.254	0.712	0.922	0.375	0.523	0.821	0.814	0.396
LightGBM	0.955	0.380	0.621	0.927	0.471	0.551	0.794	0.775	0.464
XGBoost	0.958	0.400	0.606	0.918	0.482	0.549	0.788	0.767	0.472

- 트리 계열 부스팅 알고리즘은 기본적으로 자동으로 변수 중요도 선택 기능 존재(ex. max_features = 'auto','sqrt')
- L1 규제 적용 전후의 예시를 보면 별다른 성능의 차이가 없음
- 연구 진행의 시간 단축 & 성능의 유지 및 증가

  → 1번 가설 기각

❖ 성능의 수치는 10번의 반복시행의 평균

[매번 생성된 합성데이터의 변동성 및 클래스 비율 고려]

1. 변수선택법에 따른 모형별 분류 성능 비교

- L1(Lasso) Penalty alpha =0.1-

클래스 집단 간 통계적으로 유의한 변수들

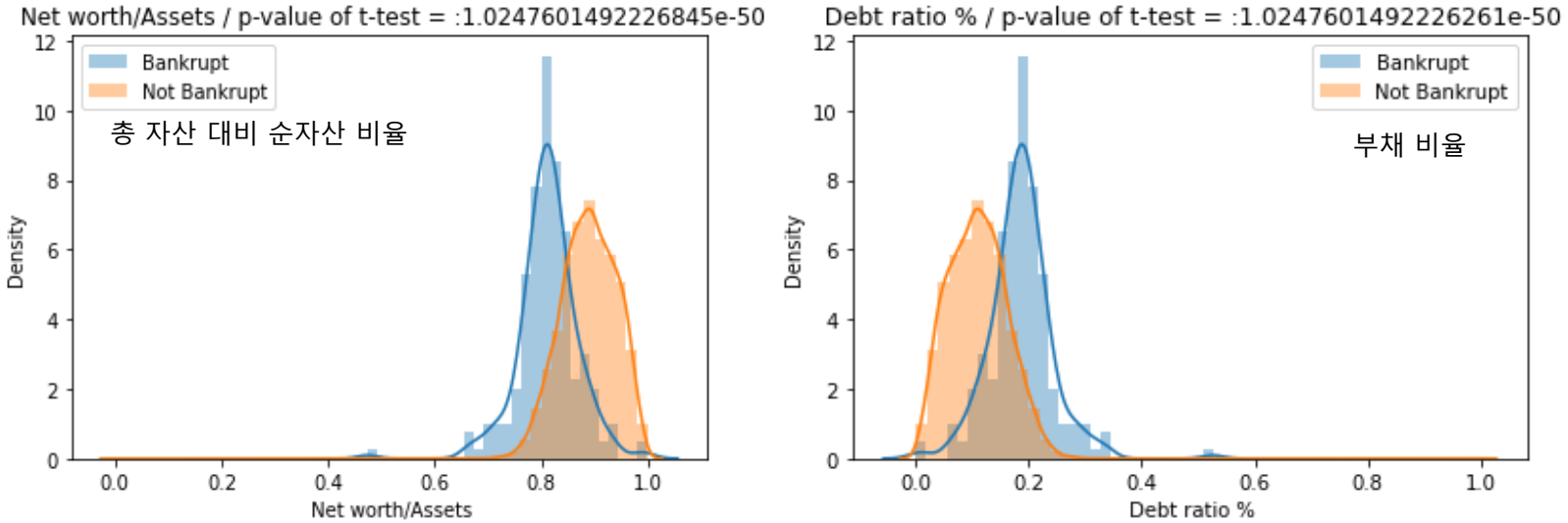
ROA(C) before interest and depreciation before interest ***	ROA(B) before interest and depreciation after tax***	Pre-tax net Interest Rate***	Net Value Per Share (B)***	Persistent EPS in the Last Four Seasons***
Total Asset Growth Rate***	Cash Reinvestment %*	Debt ratio %***	Net worth/Assets***	Borrowing dependency***
Inventory and accounts receivable/Net value***	Total Asset Turnover***	Fixed Assets Turnover Frequency***	Net Worth Turnover Rate (times)*	Operating profit per person***
Cash/Total Assets***	Cash/Current Liability***	Current Liability to Assets***	Working Capital/ Equity***	Total expense/ Assets***
Quick Asset Turnover Rate***	Cash Turnover Rate*	Net Income to Total Assets***	*** p-value < 0.05 * p-value < 0.1	

- 34개의 변수에 대한 집단간 T-Test를 실시 (다수 클래스의 데이터 임의 추출 이후 소수 클래스와의 비율 맞추기)
- 34개 중에서 23개의 변수가 T-Test에서 통계적 유의성을 나타냄
- 다른 변수에 비해서 부도 예측에 더 효율적인지 알아보기 위한 3번 가설에 추후 사용 예정

1. 변수선택법에 따른 모형별 분류 성능 비교

- L1(Lasso) Penalty alpha =0.1-

T – Test 결과 예시



해석 예시

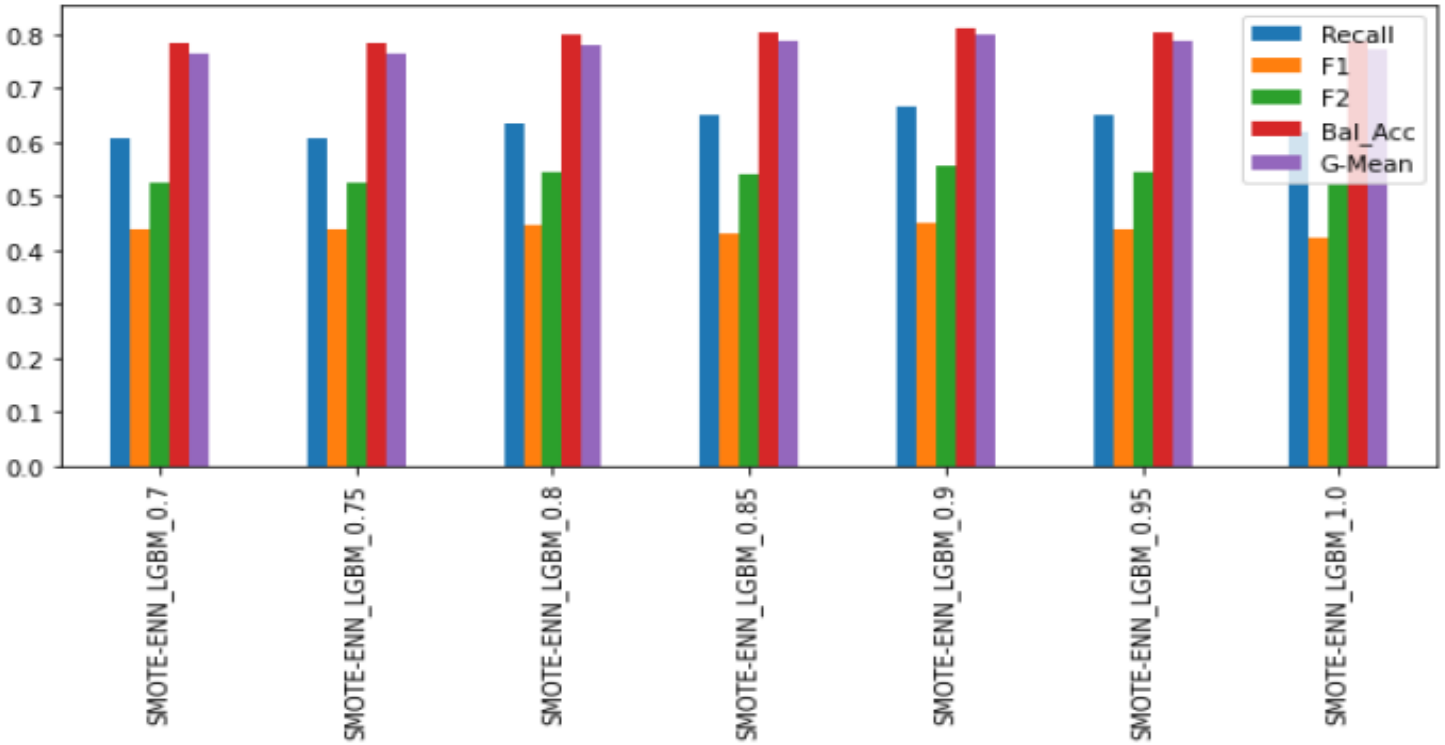
1. 총자산 대비 순자산 비율에 대한 두 집단(Bankrupt = 0, Bankrupt =1)의 유의한 차이
   → 순자산 비율이 낮을수록 기업의 재무건전성이 악화

2. 부채 비율에 대한 두 집단(Bankrupt = 0, Bankrupt =1)의 유의한 차이
   → 부채 비율이 높을수록 기업의 재무건전성이 악화

2. 샘플링 비율에 따른 모형별 분류 성능 비교

- Sampling_Strategy=0.7~1.0 -

Ex.) SMOTE-ENN Light GBM



- 샘플링 비율 0.7 ~ 1.0(증가 폭 = 0.05) 모든 방법론에 적용
- 샘플링 비율은 사용 데이터와 방법론에 따라 전부 다른 결과를 초래
- 연구의 데이터는 샘플링 비율이 증가할수록 성능 또한 증가한다는 명확한 규칙 없음

❖ 성능의 수치는 10번의 반복시행의 평균

[매번 생성된 합성데이터의 변동성 및 클래스 비율 고려]

2. 샘플링 비율에 따른 모형별 분류 성능 비교

Baseline
(기존의 불균형 비율을 유지한 데이터)

	Accuracy	Precision	Recall	AUC	F1	F2	Bal_Acc	G-Mean	MCC
GBM	0.965	0.417	0.227	0.924	0.294	0.250	0.608	0.474	0.291
LightGBM	0.972	0.667	0.242	0.933	0.356	0.278	0.608	0.474	0.291
XGBoost	0.973	0.708	0.258	0.933	0.378	0.295	0.627	0.507	0.417

Top10 성능 결과

	Sampling	Model	ratio	Precision	Recall	F1	F2	Bal_Acc	G-Mean	MCC
1	SMOTE	LightGBM	0.9	0.424	0.636	0.509	0.579	0.804	0.786	0.500
2	SMOTE	XGBoost	0.7	0.400	0.636	0.491	0.569	0.802	0.785	0.484
3	SMOTE	XGBoost	1.0	0.429	0.636	0.515	0.580	0.804	0.786	0.503
4	ADASYN	LightGBM	0.7	0.408	0.636	0.497	0.572	0.803	0.785	0.489
5	ADASYN	XGBoost	0.7	0.433	0.636	0.515	0.582	0.804	0.787	0.506
6	ADASYN	XGBoost	1.0	0.406	0.652	0.500	0.581	0.810	0.794	0.494
7	SVM-SMOTE	GBM	0.7	0.382	0.591	0.464	0.533	0.780	0.756	0.454
8	SMOTE-ENN	XGBoost	0.7	0.370	0.667	0.476	0.574	0.814	0.801	0.475
9	SMOTE-TomekLink	LightGBM	0.8	0.402	0.621	0.488	0.560	0.795	0.776	0.479
10	SMOTE-TomekLink	XGBoost	0.9	0.413	0.652	0.506	0.584	0.810	0.795	0.499

- 기준 → 재현율, F1, F2, Bal_Acc, G-Mean, MCC
- ROSE & Borderline SMOTE의 과적합 발생으로 인한 미포함 [상대적으로 덜한 성능 증가]
- 재현율의 약 2.5배 증가 및 기준 지표들의 월등한 상승
- 각 방법론에 따른 최적 샘플링 비율의 다양성 [부스팅 모형마다 다른 샘플링 기법 & 비율]

  → 2번 가설 기각

❖ 성능의 수치는 10번의 반복시행의 평균

[매번 생성된 합성데이터의 변동성 및 클래스 비율 고려]

2. 샘플링 비율에 따른 모형별 분류 성능 비교

	Sampling	Model	ratio	Precision	Recall	F1	F2	Bal_Acc	G-Mean	MCC
1	SMOTE	LightGBM	0.9	0.424	0.636	0.509	0.579	0.804	0.786	0.500
2	SMOTE	XGBoost	0.7	0.400	0.636	0.491	0.569	0.802	0.785	0.484
3	SMOTE	XGBoost	1.0	0.429	0.636	0.515	0.580	0.804	0.786	0.503
4	ADASYN	LightGBM	0.7	0.408	0.636	0.497	0.572	0.803	0.785	0.489
5	ADASYN	XGBoost	0.7	0.433	0.636	0.515	0.582	0.804	0.787	0.506
6	ADASYN	XGBoost	1.0	0.406	0.652	0.500	0.581	0.810	0.794	0.494
7	SVM-SMOTE	GBM	0.7	0.382	0.591	0.464	0.533	0.780	0.756	0.454
8	SMOTE-ENN	XGBoost	0.7	0.370	0.667	0.476	0.574	0.814	0.801	0.475
9	SMOTE-TomekLink	LightGBM	0.8	0.402	0.621	0.488	0.560	0.795	0.776	0.479
10	SMOTE-TomekLink	XGBoost	0.9	0.413	0.652	0.506	0.584	0.810	0.795	0.499

- 평가지표의 규칙 발견

  → F1(약0.5),Bal_Acc(약 0.8), G-Mean(약 0.8), MCC(0.5)일 경우 재현율이 약 0.6~0.65의 범위로 형성

  → 동시에 trade_off 고려 시 정밀도의 수치와의 어느 정도의 균형을 이룸

  → 약 0.4의 정밀도와 0.6의 재현율로 FN을 종합적 판단으로 낮춘 결과 출력

- GBM의 상대적 부진

  → 반복 실험의 평균 결과 정밀도 약 0.2 ~ 0.3, 재현율 약 0.7 ~ 0.8로 형성

  → F1, MCC의 현저히 떨어지는 수치 (약 0.4 이하)

  → 과적합 발생 및 노이즈로부터의 영향을 받은 것으로 판정

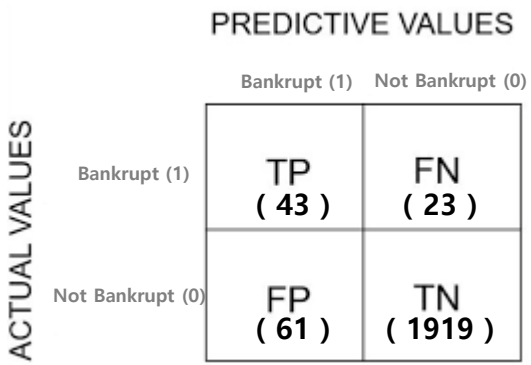
❖ 성능의 수치는 10번의 반복시행의 평균

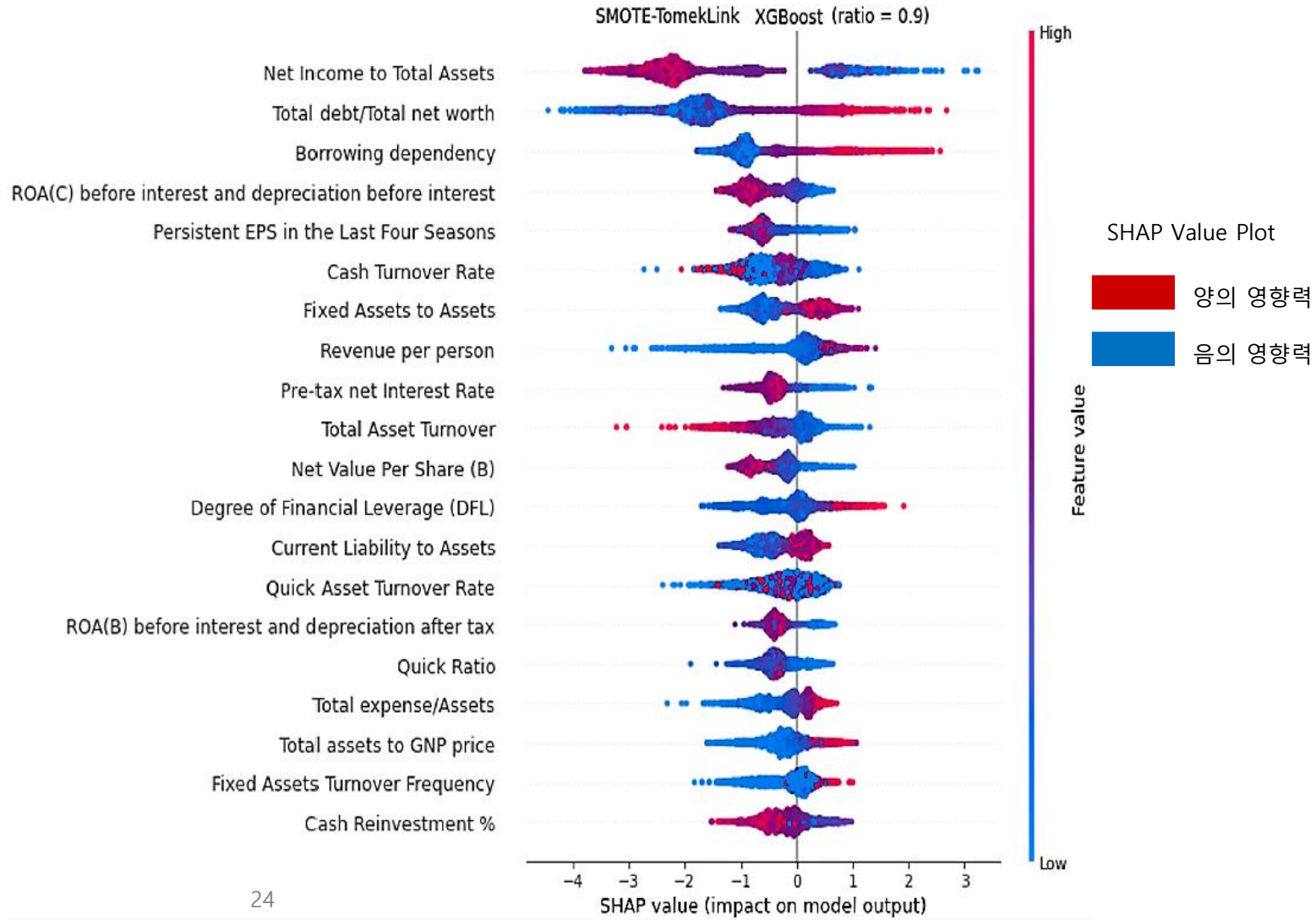
[매번 생성된 합성데이터의 변동성 및 클래스 비율 고려]

3. 설명변수의 중요도 분석

Ex.)

	Sampling	Model	ratio	Precision	Recall	F1	F2	Bal_Acc	G-Mean	MCC
10	SMOTE-TomekLink	XGBoost	0.9	0.413	0.652	0.506	0.584	0.810	0.795	0.499





3. 설명변수의 중요도 분석

- 상위권 중요도에 속한 변수들은 대체로 통계적으로 유의한 변수들로 구성

  → 모형의 성능 & 설명력 증가에 전반적인 역할 수행

  → 3번 가설 기각

- 종합 결과 기반 최고 중요 변수

  → Net Income to Total Assets(p-value < 0.05) & Total debt to Total net worth(p –value > 0.1)

- 의사결정 및 진단

  → 높은 당기순이익률 & 낮은 총부채비율로 기업의 재무건전성에 대한 1차 판단 가능
  [ex. 예의주시할 기업 사전에 파악 가능]
  → 하나 혹은 소수의 변수로 더 이상 부도 여부 판단 불가 [ex. 1968 Altman's Z-Score의 한계]

  → 우선시 고려될 변수(당기순이익률 & 총부채비율) 중심으로 나머지 변수들에 대한 종합적 판단 필요

요약

- 기업 부도 여부 예측 중심으로 클래스 불균형 문제 해결방안 제시
- 부스팅 알고리즘마다 최적의 샘플링 방법론 & 비율이 달라짐
- L1규제로 인한 정보 손실 & 성능 저하 X
- Balanced Accuracy, G-Mean, MCC, F1의 규칙 발견
- 통계적으로 유의한 변수들의 논리적 타당성 추론

한계

- 재무비율 변수만 해당된 데이터
- 다양한 모델 적용 결과의 부재
- 샘플링 방법론 & 부스팅 알고리즘의 파라미터 조정 다양성 부재

향후 연구방향

- 신용카드 부정거래, 보험 사기 탐지 등의 문제에 실험 진행
- GAN,VAE, 1D CNN 등의 딥러닝 기법 사용으로 본 연구 결과의 결과 및 해석의 확장

참고문헌

[1] An Investigation of Credit Card Default Prediction in the Imbalanced Datasets (2019)

[2] Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data(2017)

[3] A Machine Learning-based DSS for mid and long-term company crisis prediction(2021)

[4] The effect of green energy, global environmental indexes, and stock markets in predicting oil prices crashes: Evidence from explainable machine learning (2021)

[5] Application of Deep Learning for Credit Card Approval: A Comparison with Two Machine Learning Techniques(2021)

[6] Use of Data Mining in Banking(2012)

[7] Deep Learning Methods for Credit Card Fraud Detection(2021)

[8] Machine Learning-Based Detection of Credit Card Fraud : A Comparative Study(2019)

[9] Credit card fraud detection using artificial neural network(2021)

[10] Credit Card Fraud Detection using Imbalance Resampling Method with Feature Selection(2021)

[11] Deep Learning vs traditional Machine Learning algorithms used in Credit Card Fraud Detection(2017)

[12] Real-Time Deep Learning Based Credit Card Fraud Detection(2020)

[13] Comparing Different Supervised Machine Learning Accuracy on Analyzing COVID-19 Data Using ANOVA Test(2020)

[14] Default avoidance on credit card portfolios using accounting, demographical and exploratory factors: decision making based on machine learning(ML) techniques(2017)

[15] Application of machine learning and data visualization techniques for decision support in the insurance sector(2021)

[16] Predicting bank insolvencies using machine learning techniques(2020)

[17] Bankruptcy prediction for small-and-medium-sized companies using severely imbalanced datasets(2020)

[18] A survey on addressing high-class imbalance in big data(2018)

참고문헌

[19] A comparison of classification models for imbalanced datasets(2016)

[20] Bankruptcy prediction using machine learning and an application to the case of the COVID-19 recession(2021)

[21] A survey on Machine Learning and Statistical Techniques in Bankruptcy Prediction(2018)

[22] Bankruptcy Prediction Using Machine Learning(2017)

[23] Classification of Imbalance Data using TomekLink(T-Link) Combined with Random Under-sampling(RUS) as a Data Reduction Method(2016)

[24] Geometric Mean-based Optimization Boosting for Bankruptcy Prediction(2021)

[25] 기업부도와 기계학습(2019)

[26] 머신러닝과 금융: 머신러닝 기반 신용평가모형(2020)

[27] LightGBM 알고리즘을 활용한 고속도로 교통사고심각도 예측모델 구축(2020)

[28] 카드사 빅데이터와 딥러닝 신경망 분석 도입의 의미와 활용(2017)

[29] 딥러닝 신경망 이용한 신용카드 부도율 예측의 효용성 분석(2017)

[30] SVM과 meta-learning algorithm을 이용한 고지혈증 유병 예측모형 개발과 활용(2017)

[31] 효과적인 기업부도 예측모형을 위한 ROSE 표본추출기법의 적용(2018)

[32] 불균형 데이터 분류를 위한 오버샘플링 및 언더샘플링 조합 방법(2019)

[33] 최적 샘플링 비율 탐색을 통한 불균형 자료 문제 해결 방안(2020)

[34] 부트스트랩 표집 비율과 불균형 데이터 전처리 방법에 따른 배깅과 랜덤 포레스트 분류 모형 비교(2021)

[35] 부도 데이터의 불균형 문제 해결을 위한 적대적 생성 신경망(GAN) 기반 오버샘플링 기법(2020)

[36] 불균형 정형데이터 문제 해결을 위한 SMOTE와 CycleGAN 기반 하이브리드 오버샘플링 기법 개발 및 적용: 금융사기를 중심으로