2021학년도 2학기

통계분석실습 기말프로젝트 보고서

- 회사채 신용등급 변화 예측모형 개발 -

2조

이은경(1810618), 정은정(1814216), 최윤서(1916129)

2021.12

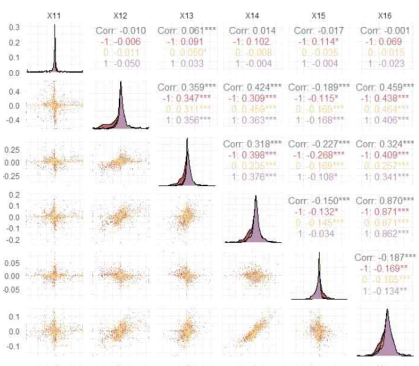
Ⅰ. 서론

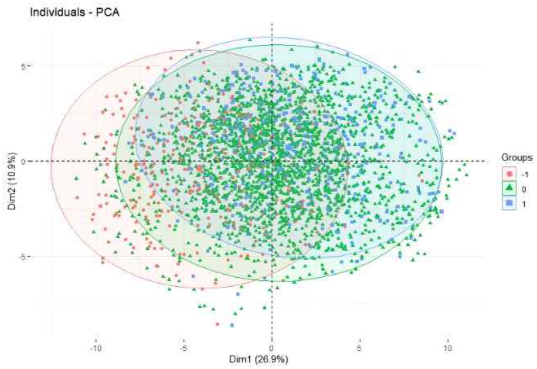
1. 프로젝트 개요

본 분석에서는 1998년부터 2020년까지 시간에 따라 관측된 여러 회사들의 데이터를 가지고 신용등급 변화(C)를 예측하는 프로젝트를 진행하였다. 종속변수는 신용등급 변화(C)이고, 설명변수는 현 시점과 과거 시점의 규모지표(5개), 비재무 지표(4개), 생산성 지표(2개), 수익성 지표(18개), 안정성 지표(17개), 현금흐름 지표(9개), 활동성 지표(4개)인 X1~60, BX1~60(X59, BX59는 제외)과 이전 시점의 신용등급 Y_b, 이전 시점의 신용등급이 측정된 해를 나타내는 yy_b 변수를 사용하였다. 종속변수 C의 경우 -1, 0, 1의 비율이 각각 16.98%, 72.36%, 10.67%로 클래스 불균형 데이터였다. 클래스 불균형 상황에서 단순히 Accuracy를 이용하여 모델을 평가하는 것은 모델의 성능을 과대평가할 가능성이 있으므로 precision과 recall을 동시에 고려하는 macro f1 score를 모델 평가지표로 사용하였다.

2. 설명변수 탐색

주어진 데이터, train set과 test set 자체에 결측치가 존재하지 않고, 이상치가 어느 정도 정제된 데이터였으므로 결측치 대체, 이상치 정제 과정은 수행하지 않아도 된다고 판단하였다. 그러나, 데이터를 전체적으로 살펴보는 것은 필요한 과정이므로 데이터 탐색 과정을 축약해 진행해보았다.





설명변수들의 히스토그램을 보면, 분포가 어느 정도 정규분포 모형을 따른다고 볼 수 있으며, 설명변수 간 높은 상관 계수를 갖는 경우가 있는 것으로 파악된다. 이 부분은 설명변수들의 정보력이 겹쳐 모형 적합 때 성능이 낮게 나오는 상황이 발생하지 않도록 주의를 기울여야 하는 부분이다. 설명변수들을 가지고 데이터 시각화를 목적으로 주성분 분석을 시행하고 주성분 2개를 축으로 하여 관측값들을 좌표축 상에 위치시킨 뒤, 반응변수 값을 기준으로 관측치들을

grouping 해보았다. 이때, 첫 번째 주성분은 수익성, 자산지표를 의미하고 두 번째 주성분은 규모 지표를 의미하는데 반응변수 값이 1인 관측치들의 경향이 수익성, 자산지표가 높은 것으로 보아 이 값이 높을수록, 반응변수 값이 1일 가능성이 커질 것으로 예측된다.

Ⅱ. 데이터 전처리 및 훈련, 검증데이터 분할

1. 데이터 전처리 (Data Transformation)

1) 설명변수 Y_b

기업의 신용등급을 나타내는 Y_b 변수를 문자형에서 연속형 변수로 변환시켜 주었다. train 데이터는 Y_b 변수가 총 21개의 범주로 구분되어 있었으나, model에 범주형 변수로 입력 시 one-hot encoding 같은 추가적인 작업이 필요하고, 이를 시행할 시, 설명변수의 개수가 많아지는 불편함을 해결하기 위해 신용등급이 낮은 것부터 오름차순으로 숫자를 대입하여 연속형 변수로 만들어 주었습니다. (즉, D부터 AAA까지의 신용등급을 0부터 20까지의 숫자로 차례대로 대응시켜줌.)

2) 종속변수 C

반응변수 C는 -1, 0, 1로 구성되어 있는데, 이는 순서가 있는 범주형 변수이므로 숫자형 변수에서 순서형 변수로 변환시켰다.

2. 훈련, 검증데이터 분할

본 분석을 진행하면서 분류 문제에 적용가능한 많은 모델을 고려하였으며 많은 모델 후보 중 가능성있는 후보를 추려내기 위해 검증데이터가 필요했다. 시계열 데이터의 특성상 훈련데이터보다 미래 시점의 데이터를 검증데이터로 사용하는 것이 적절하다고 판단하여 특정 시간을 기점으로 앞, 뒤로 훈련데이터와 검증데이터를 분할하였다. 최대한 많은 훈련데이터를 확보하기 위해 검증데이터를 400개 정보만 확보하고자 하였고 최종적으로 yy_b변수 기준으로 1998~2014년 데이터를 훈련데이터, 2015~2016년 데이터를 검증데이터, 2017~2020년 데이터를 테스트데이터로 사용하였다.

3. 설명변수 조합 선택

먼저 설명변수 조합을 선정하고, 각 모델에 적용하였다. 설명변수 조합에 따라 반응변수 예측 성능이 달라지기 때문이다. 고려한 설명변수 조합은 세 가지로, 첫 번째 조합은 X1~X60 - BX1 - BX60, Y_b, 두 번째 조합은 X1~X60, Y_b, 세 번째 조합은 X1~X60, BX1~BX60, Y_b 이다.

위 세 가지 조합을 선정한 이유는 다음과 같다. BX1~BX60은 이전 시점에 관측된 X1~X60이다. 우선 첫 번째 조합은 예측하고자 하는 C는 신용등급의 변화이기 때문에 지표의 변화가 종속변수 예측에 도움을 줄 것이라 생각하여 현재의 지표값인 X1~X60에서 BX1~BX60을 뺀 변수를 고려하였다. 두 번째 조합의 경우 Y_b가 BX1~BX60의 정보를 담고 있을 것이라 생각하여 전체 변수에서 BX1~60을 제외한 변수를 고려하였다. 마지막으로 모든 설명변수를 고려한 세 번째 조합을 고려하였다.

본격적으로 모델 적합 과정에 들어가기 전 다항 로지스틱 회귀 모형을 나이브 벤치마크 삼아 위의 세가지 설명변수 조합 중 하나의 조합을 선택하고자 했다. 설명변수 조합을 선정한 성능 기준은 검증데이터에서의 macro f1 score이다. 다항 로지스틱 회귀 모형 적합 결과, 검증데이터에서의 macro f1 score은 첫 번째 조합에서 0.4093, 두 번째 조합에서 0.5185, 세 번째 조합에서 0.5469였다. 따라서 가장 높은 성능을 보인 세 번째 조합(X1~X60, BX1~BX60, Y_b)을 최종 설명변수 조합으로 선정하였다.

4. SMOTE 처리

종속변수 C의 클래스 불균형 문제를 해결하기 위해 SMOTE 방법을 사용하였다. SMOTE의 경우 최종 클래스의 비율에 따라 모델 성능에 차이가 있을 수 있다고 생각하여 완벽하게 1:1:1의 비율을 가진 SMOTE데이터(이하 perfect_data.csv)와 기존데이터와 비율의 크기 차이(0 > -1 > 1)는 비슷하게 SMOTE처리를 한 데이터(이하 smote_data.csv) 두 가지를 준비하였다. 이 때 정확한 검증데이터 오류를 측정하기 위해서는 훈련데이터에 검증데이터의 정보가 포함되지 않아야 하기에 훈련데이터만을 SMOTE한 데이터와 최종 모델 훈련을 위한 전체데이터(훈련+검증데이터)에 대한 SMOTE데이터 (이하 final_perfect_data.csv)를 준비하였다.

Ⅲ. 모델 훈련

1. 나이브 베이즈

나이브 베이즈 분류는 공변량 사이의 독립성을 가정하는 베이즈 정리를 적용한 확률 분류기이다. 나이브 베이즈는 적은 훈련데이터로도 분류 성능이 크게 저하되지 않으며, 공변량이 많더라도 단순화 시켜서 쉽고 빠르게 판단한다는 장점이 있다. 나이브 베이즈 모델의 파라미터 추정은 최대우도방법을 사용한다. 나이브 베이즈는 왼쪽 식을 통하여 가장 가능성이 높은 클래스 k를 찾아낸다.

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k).$$

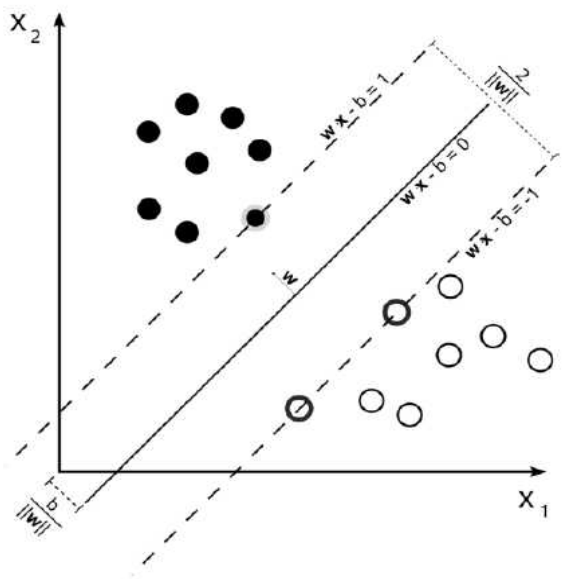
나이브 베이즈에서는 기존에 없는 새로운 값이 입력된다면 해당 값에 대한 확률이 0이 되기 때문에 모든 확률이 0이 되는 문제가 발생한다. 따라서 각 분자에 일정 값을 더하는 라플라스 스무딩 과정을 진행해야 한다. 따라서 하이퍼파라미터 laplace를 디폴트 값인 0부터 5까지 고려하여 성능을 비교하였다. 그 결과, 라플라스 스무딩을 진행하지 않는 laplace 0일 때의 성능이 가장 높았다. laplace를 0으로 설정하고 나이브 베이즈를 적합하였을 때 검증데이터에서의 macro f1 score은 0.4697이었다.

2. SVM

Support Vector machine(이하 SVM)은 분류를 위한 margine이 최대가 되는 최적의 결정경계를 추정하는 지도학습 모형이다. 주로 분류와 회귀분석을 위해 사용되며, 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만든다.

고려한 하이퍼파라미터는 cost, gamma, kernel이다. cost는 이상치를 얼마나 허용할 것인지를 결정하는 하이퍼파라미터이다. cost값이 클수록 오류허용을 하지 않고, cost값이 작을수록 오류허용을 한다는 의미이다. cost는 0부터 5까지를 고려하였다.

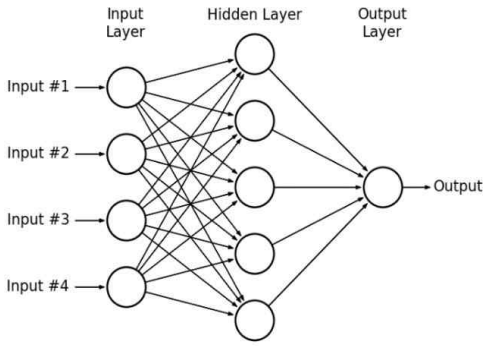
gamma는 결정경계를 얼마나 유연하게 그을지 결정하는 하이퍼파라미터이다. gamma값이 클수록 구불구불한 경계를 형성하여 오버피팅 가능성이 있으며, gamma값이 작을수록 직선에 가까운 경계를 형성하여 언더피팅 가능성이 있다. gamma는 0부터 0.5까지를 고려하였다. 또한 고차원 공간을 고려하여 분류를 수행하는 kernel 함수를 결정하였다. kernel 함수는 linear, radial, sigmoid를 고려하였다. kernel 함수가 radial일 때, 다른 kernel 함수보다 전반적인 성능이



좋았으므로 kernel을 radial로 고정하고 cost와 gamma를 튜닝하였다. cost가 2보다 크면 성능이 감소하는 경향을 보였고, gamma는 0.01보다 크면 성능이 감소하는 경향을 보였다. 따라서, 최적의 하이퍼파라미터 조합을 kernel이 radial일 때, cost가 2일 때, gamma가 0.01일 때로 선정하였다. 그 조합에서의 검증데이터 macro f1 score은 0.7112로 도출되었다.

3. Deep learning model

본 분석에서 사용한 딥러닝 모델은 다층 퍼셉트론이다. 다층 퍼셉트론(Multilayer perceptron, MLP)은 퍼셉트론을 여러 층 쌓은 순방향의 인공 신경망이다. 입력층(input layer)과 은닉층(hidden layer)과 출력층(output layer)으로 구성된다. 각 층에서는 활성함수(activation function)를 통해 입력을 처리한다. 모형을 시각화하자면 다음 그림과 같이 표현할 수 있다.



이 model의 hyperparameter라고 할 수 있는 option 값은 hidden layer의 개수, 각 layer node의 개수, activation function, overfitting을 방지하는 L2 regularization(weight space를 제약해 학습하는 option), Dropout rate(Output layer와 마지막 hidden layer 사이에 추가하는 layer로, Fully-connected layer로 출력되는 것이 아니라, 몇몇의 node를 dropout 함으로써 과적합을 방지할 수 있는 option)이다. 분석자가 사용한 option 값의 범위는 다음과 같다.

	고려한 값의 범위
hidden layer의 개수	2개, 3개, 4개, 5개, 6개
layer node의 개수	각 layer마다 512, 256, 128, 64, 32, 16, 10, 8개
activation function	Relu, Sigmoid function
L2 Regularization rate	0.001, 0.0015, 0.01, 0.15, 0.2
Dropout rate	0.1, 0.2, 0.3, 0.4, 0.5

각 조합마다 perfect_data.csv를 train set으로 model을 적합하고 validation set의 반응변수를 예측하고 macro f1-score을 계산한 결과, hidden layer의 개수가 2개일 때, 각 layer nodel의 개수가 16개, 10개 일 때, activation function은 Relu일 때, L2 regularization rate가 0.015일 때, Dropout rate가 0.1일 때 validation set의 macro f1-score가 0.6039로 가장 높게 나왔다.

4. Random forest

ensemble 기법 중 하나로, tree를 ensemble해 tree의 단점을 보완하는 model 중 하나이다. 이 model에서 고려한 hyperparameter는 base learner tree의 개수, 각 tree의 깊이를 지정할 수 있는 max_depth, 각 tree의 leaf(node) 안에 포함되는 최소한의 sample 개수를 의미하는 min_samples_leaf 이다. 분석자가 사용한 hyperparameter 값의 범위는 다음과 같다.

	고려한 값의 범위
tree의 개수	300개, 400개, 500개
max_depth	10, 20, 30
min_samples_leaf	2개, 3개, 4개

각 조합마다 perfect_data.csv을 이용하여 GridSearchCV를 시행한 결과, tree의 개수가 500개, max_depth가 30, min_samples_leaf가 2개가 최적의 parameter 조합으로 나왔다. 그리고 perfect_data.csv를 train set으로 하여 model을 적합하고 validation set의 반응변수를 예측하게 하였다. 이때, 반응변수 C값이 -1일 때와 1일 때 좀 더 집중적으로 학습하도록 가중치를 부여하였는데 각각의 가중치는 0.1, 0.126이다. 가중치 계산에 이용한 식은 (전체 데이터의 개수 / (3 * 각 클래스의 개수))이다. 그 결과, macro f1-score 값이 0.5803으로 가장 높게 나왔다.

5. Xgboost

Xgboost의 경우 Gradient boosting Model을 병렬처리 및 규제를 가하는 방식으로 속도와 성능면에서 향상시킨 알고리즘이다. 모델의 type에 따라 기본학습기와 학습 방식이 달라지기 때문에 각 모델 특성에 맞게 하이퍼파라미터 튜닝을 시도하였다.

1) xgbDART

xgbDART 모델의 경우 기본학습기는 tree이고 최종 모델에 학습된 기본학습기 중 일부만 사용한 모델이다. 모델의 학습률(eta)와 tree의 깊이(max_depth), 기본학습기의 개수(nrounds), 트리의 drop 비율과 확률(rate_drop, skip_drop), 기본학습기 모델 적합에 사용하는 데이터 샘플링 정도(subsample, colsample_bytree), 규제정도(gamma)를 나타내는 하이퍼파라미터를 아래와 같은 범위에서 고려하였다. 세밀한 학습을 위해 낮은 학습률을 고려하였으며, 높은 편향을 가진 기본학습기를 위해 낮은 depth를, 과적합 방지를 위해 데이터를 샘플링해주는 하이퍼파라미터를 고려하였다. dart모형의 경우 고려해야 할 하이퍼파라미터가 많아 eta, max_depth, nrounds와 같은 중요한 하이퍼파라미터를 1차적으로 튜닝한 뒤, 그 후 이 외의 파라미터를 튜닝하는 과정을 거쳤다. 그 결과 검증데이터 macro f1-score 0.5428을 얻었다.

	고려한 값의 범위
eta	0.05, 0.1
max_depth	1, 2, 3
nrounds	50, 100, 200
rate_drop	0, 0.25, 0.5
skip_drop	0, 0.25, 0.5
subsample	0.5, 0.7
gamma	0.1, 0.2, 0.3, 0.4
colsample_bytree	0.5, 0.7

2) xgbTree

xgbTree 모델의 경우 기본학습기가 tree이고, 기본학습기 훈련 시 규제를 가하는 모델이다. 아래의 표에 나타난 범위에서 모델 튜닝 과정을 거쳤으며, dart와 기본학습기의 drop 여부만 차이를 가졌기 때문에 동일한 사항을 고려하였다. 그 결과 검증데이터 macro f1-score 0.5242를 얻었다.

	고려한 값의 범위
eta	0.05, 0.1
max_depth	2, 3
subsample	0.3, 0.5, 0.7
nrounds	100, 200, 300, 400, 500
gamma	0.1, 0.2, 0.3, 0.4
colsample_bytree	0.3, 0.5, 0.7

3) xgbLinear

xgbLinear 모델의 경우 기본학습기는 규제가 더해진 logistic linear regression model이다. 모델의 학습률(eta), 기본학습기의 개수(nrounds), l1, l2규제 정도(lambda, alpha)를 나타내는 하이퍼파라미터를 위의 그림과 같이 고려하였다. xgbLinear 모델의 경우 교호작용항을 추가한 모델과 추가하지 않은 모델을 비교해보았으며 각각 검증데이터 macro f1-score 0.5106, 0.5129를 얻었다.

	고려한 값의 범위
eta	0.05, 0.1
nrounds	10, 20, 30
lambda	0, 0.1, 0.3
alpha	0, 0.1, 0.3

6. 최종모델 선정

검증데이터기준 macro f1-score가 나이브벤치마크의 0.5469보다 높거나 비슷하게 나온 Deep learning(0.6039), SVM(0.7112), Random Forest(0.5803), Xgboost(0.5428)을 최종 모델 후보로 선정하였다.

위의 모델 중 딥러닝 모델의 경우 80% 테스트데이터에서 최대 성능이 0.24974로 검증데이터에서의 성능과 큰 차이가 났다. 이와 같은 결과에서 데이터의 개수가 적어 딥러닝 모델이 feature들을 잘 학습하지 못했다고 판단하였고, 다층 퍼셉트론은 feature와 반응변수간의 선형관계를 가정하는데, 이 가정이 성립한다고 확신할 수 없는 상황이므로 새로운 예측 상황에서 모델의 성능이 좋지 않게 나왔다고 생각했다.

딥러닝 모델을 제외한 3개 모델들의 80% 테스트데이터에서의 성능은 SVM가 0.51108, Random Forest가 0.51774, Xgboost가 0.49728이였다. 각 모델들의 성능이 비슷한 수준으로 나타났지만, 각 모델들의 클래스 예측비율을 살펴보았을 때 SVM은 0.03:0.91:0.05, Random Forest는 0.12:0.85:0.03, Xgboost는 0.08:0.89:0.02로 다른 양상을 보였다.

따라서 각 모델들이 잘 예측하는 상황이 다르다고 판단하였고, 3개의 모델에서 나온 예측 결과로 class를 투표하는 방식으로 최종 예측을 진행하였다. 만약 -1, 0, 1과 같이 동점이 나온 경우 순서형의 특징을 고려하여 클래스를 0으로 분류하도록 코드를 작성하였다. 1, 0, 0 또는 -1, 0, 0과 같은 경우 1 또는 -1의 경향을 하나의 모델에서도 보였다는 점에 의미를 두어 각각 1과 -1로 분류하도록 코드를 작성하였다. 이러한 규칙을 바탕으로 각 모델의 예측 클래스를 합치는 방식으로 최종 예측을 진행한 결과 80% 테스트 데이터에서 macro f1-score 0.53480, 20% 테스트 데이터에서 macro f1-score 0.47287을 얻었다.

Ⅳ. 분석 프로젝트 자체 평가

1. 의의

본 분석의 경우 테스트데이터에서 좋은 성능을 내는 모델을 선택하기 위해 검증데이터를 적절히 분할하였다. 이 과정에서 검증데이터 오류를 정확히 측정하기 위해 훈련데이터와 전체 훈련데이터에 대해 SMOTE를 각각 적용하였고, 검증오류 측정 과정과 최종 모델 훈련과정에서 사용하는 데이터를 분할하였다. 그 결과 테스트데이터에서도 좋은 성능을 가지는 모델을 선택할 수 있었다는 점에서 의의가 있다. 또한, 분류 문제를 해결할 수 있는 모델을 최대한 많이 고려하였다는 점에서 선택할 수 있는 모델이 다양했고, 그 결과 모델의 클래스를 합쳐 최종 결과를 도출했을 때 더 안정적이고 좋은 성능이 나올 수 있었다고 생각한다.

2. SMOTE 외의 클래스 불균형 처리 방법 적용

본 분석에서 사용한 데이터의 경우, 반응변수의 클래스 불균형이 심한 데이터였다. 클래스 불균형을 해결하지 않은 채 모형을 적합할 시 낮은 성능을 보일 수 있어, 클래스 불균형 처리가 불가피하였다. 클래스 불균형을 해소할 수 있는 방법에는 Over-sampling, Under- sampling, SMOTE, ROSE가 있다. 그러나 이 분석에서는 SMOTE 방법만 사용하였다. 어떠한 방식으로 클래스 불균형을 처리하여 모형에 적합하느냐에 따라 예측 성능이 달라지기 때문에, 다양한 불균형 처리를 하고 성능을 비교하는 과정이 없었던 점이 아쉽다. 이후 분석에서는 불균형

처리를 다양하게 한 후 성능을 비교하는 과정을 추가할 것이다.

3. 각 모델 별로 잘못 예측된 결과를 분석하고 개선해보는 과정 필요

각 모델로 예측을 해본 후에 오분류율을 분석하는 과정이 부족하였다. 예측 이후의 confusion matrix를 통해 각 모델이 정확히 예측한 것과 잘못 예측한 것을 알 수 확인할 수 있다. 이를 통하여 각 모델의 장단점을 비교하거나, 보완이 필요한 부분을 튜닝하는 활동을 기대해볼 수 있었던 점이 아쉽다. 이후 분석에서는 예측 이후에 각 모델 별로 잘못 예측한 결과를 분석하고 개선하는 과정을 추가할 것이다.

4. 데이터 탐색 과정의 아쉬움

본 분석에서, 데이터 탐색 부분에서 데이터 시각화를 목적으로 PCA를 진행하고 반응변수의 class 별 관측치들의 경향을 살펴보았다. 그러나, PCA를 시행할 때 반응변수인 C 변수는 고려하지 않고 설명변수들만 고려하였으므로 반응변수 공간을 잘 설명하는 대표적인 주성분 2개가 수익성, 자산비율, 규모 지표라고 할 수 없다. 따라서 위에서 언급한 경향성은 실제로 사실이 아닐 가능성이 있으므로, PCA 대신 주어진 데이터로 가장 잘 판별할 수 있는 판별식을 만들어, 새로운 데이터를 분류하는 방법인 CDA(정준판별분석)을 사용해 class 별 관측치들의 경향성을 판단하는 것이 더 의미 있는 결과를 도출할 수 있지 않았을까 하는 아쉬움이 있다.