YonseiESC 2020 SPRING

FINAL PROJECT

1조 곽현지 서경덕 손동규 유은영 정규형

목차

1. 추가 EDA

1) 변수 추가 및 분류 - 21번 변수

- **Attribute 21: Sales(n) / Sales(n-1) => 매출액 증가율**

- 1차 EDA: NA가 많아서 삭제하기로 결정

- 하지만 NA값이 대부분 부도기업에서 발견되어 **"신생기업"을 나타내는 변수** 로 바꿀 수 있다고 판단
- 신생기업의 경우 n-1년도 매출 정보가 존재하지 않음

- Class(부도여부)와 correlation : 0.47

⇒ **신생기업: 1 , 기존기업:0**

	부도수	부도율
기존	33	0.0051
신생	79	0.2188

1. 추가 EDA

2) 변수 추가 및 분류 - 변수 추가

① X55: Working Capital -> X66: 순운전자본회전기일

: 여러 기업을 비교해야하는 경우 단일 항목은 의미 없음
: 기업의 활동성 지표인 **'순운전자본회전기일'** 로 교체

② X65: Ebit 대 매출액

: 다른 변수들간 correlation이 높아 사용하지 않음.

1. 추가 EDA

2) 변수 추가 및 분류 - 변수 분류

수익성

: 기업의 경영성과를 측정하는 비율.
: 자산 이용의 효율성, 이익창출능력 등에 대한 평가지표로 활용

- X1: 총자산 이익률
- X13, X39

안전성

: 자산 & 자본의 관계 비율.
: 단기채무지불능력 경기대응 능력을 측정하는 지표로 활용

- X4: 유동비율
- X6: 유보율
- X51: 단기부채비율
- X5, 15, 26

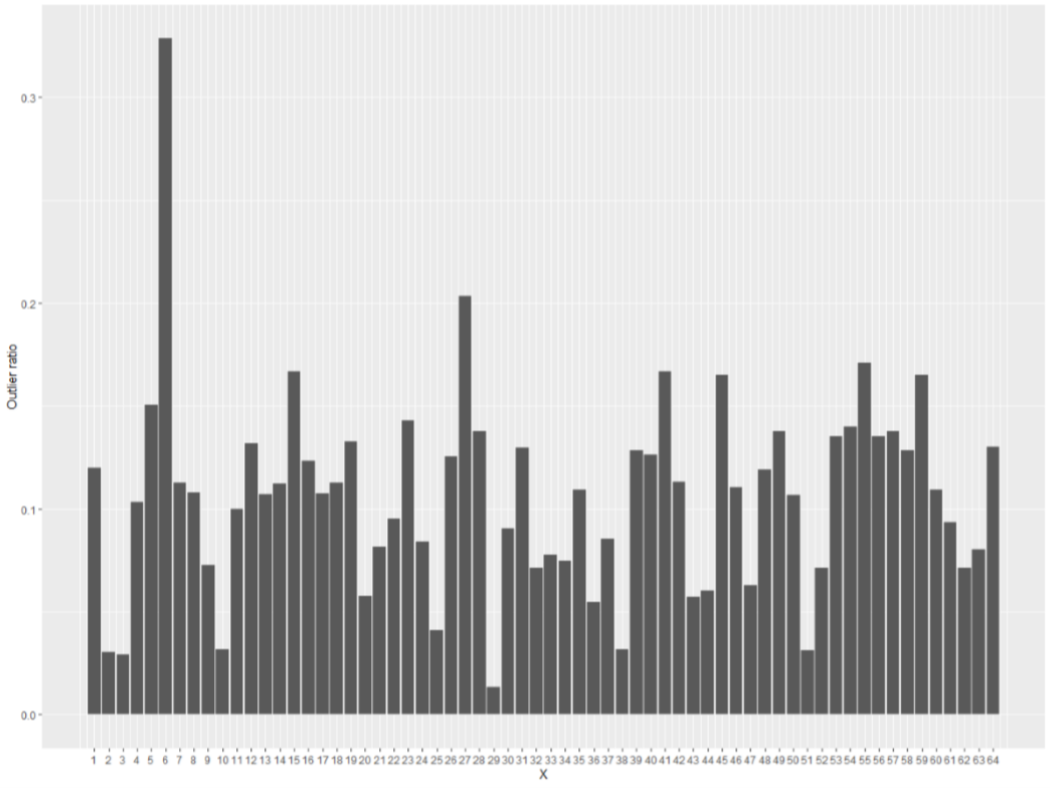
활동성

: 자산 & 자본의 회전율.
: 기업에 투하된 자본이 얼마나 활발하게 운용되었는가를 나타내는 비율.

- X9: 총자산회전율
- X44: 최권회전일수(역수)
- X47: 재고자산전환일(역수)
- X66: 순운전자본회전기일
- X20

1. 추가 EDA

3) Outlier handling



1) Multivariate outlier detection 방식

2) Local Outlier Factor(LOF)

3) Isolation Forest

1. 추가 EDA

4) NA imputation for ‘test set’

- Sales. Short-term liabilites=0인 경우
- Inventory =0인 경우 (409개) (X45, X60)
- Fixed asset, current asset=0인 경우 (160~200개) (X28, X53, X54, X64)

```
> score(bankrupt1)
[1] 0.2222222
> score(bankrupt)
[1] 0.3809524
```

- Liabilities 계열이 0인 경우 (X4, X12, X17)
- KNN method (X24, X27, X32, X40, X41, X46, X51, X61)

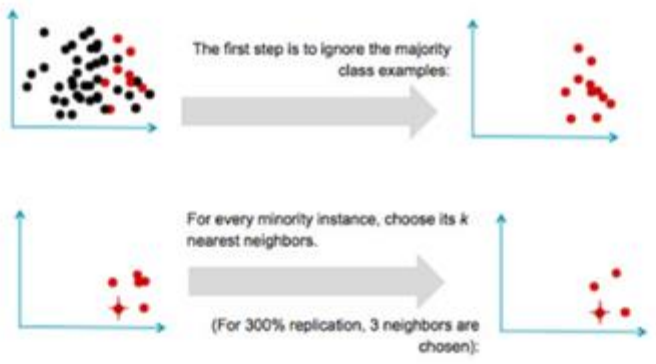
2. Modeling

1) Over Sampling

- Imbalanced data의 경우 **정확도가 높아도 재현율이 낮음.**
- F1 score가 낮은 문제 해결을 위한 Over sampling: '**SMOTE**'

부트스트래핑이나 KNN(최근접이웃) 모델 기법을 활용한다.

소수 데이터 중 특정 벡터 (샘플)와 가장 가까운 이웃 사이의 차이를 계산한다.
이 차이에 0과 1사이의 난수를 곱한다.
타겟 벡터에 추가한다.
두 개의 특정 기능 사이의 선분을 따라 임의의 점을 선택할 수 있다.



SMOTE ? ▼

SMOTE (Synthetic Minority Over-sampling Technique)

: SMOTE는 데이터의 개수가 적은 클래스의 표본을 가져온 뒤 임의의 값을 추가하여 새로운 샘플을 만들어 데이터에 추가하는 오버 샘플링 방식으로 동작

2. Modeling

1) Over Sampling

- Imbalanced data의 경우 **정확도가 높아도 재현율이 낮음.**
- F1 score가 낮은 문제 해결을 위한 Over sampling: **'SMOTE'**

```
lda = LDA().fit(X_train, y_train)
print('LDA Test :', f1_score(y_test, lda.predict(X_test), pos_label=1, average='binary'))
confusion_matrix(y_test, lda.predict(X_test))
```

```
LDA Test : 0.411214953271028

array([[1817,   10],
       [  53,   22]])
```

```
for i in range(5):
  sm = SMOTE(sampling_strategy=(i+1)/10, random_state=2020)
  X_sm, y_sm = sm.fit_resample(X_train, y_train)
  X_sm = pd.DataFrame(X_sm, columns = bank.columns[0:18])
  lda_sm = LDA().fit(X_sm, y_sm)
  print('LDA Test w/ sampling strategy', (i+1)/10, ":", f1_score(y_test, lda_sm.predict(X_test), pos_label=1, average='binary'))
  print(confusion_matrix(y_test, lda_sm.predict(X_test)))
```

```
LDA Test w/ sampling strategy 0.1 : 0.4036697247706422
[[1815   12]
 [  53   22]]
LDA Test w/ sampling strategy 0.2 : 0.39999999999999997
[[1814   13]
 [  53   22]]
LDA Test w/ sampling strategy 0.3 : 0.3893805309734513
[[1811   16]
 [  53   22]]
LDA Test w/ sampling strategy 0.4 : 0.41025641025641024
[[1809   18]
 [  51   24]]
LDA Test w/ sampling strategy 0.5 : 0.4033613445378151
[[1807   20]
 [  51   24]]
```

2. Modeling

2) 최종 모델 선정 - f1 score 비교

<NA deleted ver.>

```
score_logit = []
for train_index, test_index in cv.split(X_scaled, y):
  X_train, X_test = X_scaled.iloc[train_index, :], X_scaled.iloc[test_index, :]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]
  Logit = LR(solver='sag', max_iter=10000, multi_class='auto').fit(X_train, y_train)
  score_logit.append(f1_score(y_test, Logit.predict(X_test), pos_label=1, average='binary'))
print(np.mean(score_logit))
```

0.4093003441471685

```
score_lda = []
for train_index, test_index in cv.split(X, y):
  X_train, X_test = X.iloc[train_index, :], X.iloc[test_index, :]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]
  lda = LDA().fit(X_train, y_train)
  score_lda.append(f1_score(y_test, lda.predict(X_test), pos_label=1, average='binary'))
print(np.mean(score_lda))
```

0.41222446804576457

<NA imputed ver.>

```
score_logit = []
for train_index, test_index in cv.split(X_scaled, y): # StandarfScaling 적용
  X_train, X_test = X_scaled.iloc[train_index, :], X_scaled.iloc[test_index, :]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]
  Logit = LR(solver='sag', max_iter=10000, multi_class='auto').fit(X_train, y_train)
  score_logit.append(f1_score(y_test, Logit.predict(X_test), pos_label=1, average='binary'))
print(np.mean(score_logit))
```

0.32776221471027384

```
score_lda = []
for train_index, test_index in cv.split(X, y):
  X_train, X_test = X.iloc[train_index, :], X.iloc[test_index, :]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]
  lda = LDA().fit(X_train, y_train)
  score_lda.append(f1_score(y_test, lda.predict(X_test), pos_label=1, average='binary'))
print(np.mean(score_lda))
```

0.33021407495028343

	NA deleted	NA imputed
Logistic	0.409	0.238
LDA	0.412	0.330
Adaboost	0.326	0.301
Random Forest	0.393	0.312
Gradient Boost	0.406	0.151
Decision Tree	0.288	0.277

2. Modeling

2) 최종 모델 선정 - Test set 비율

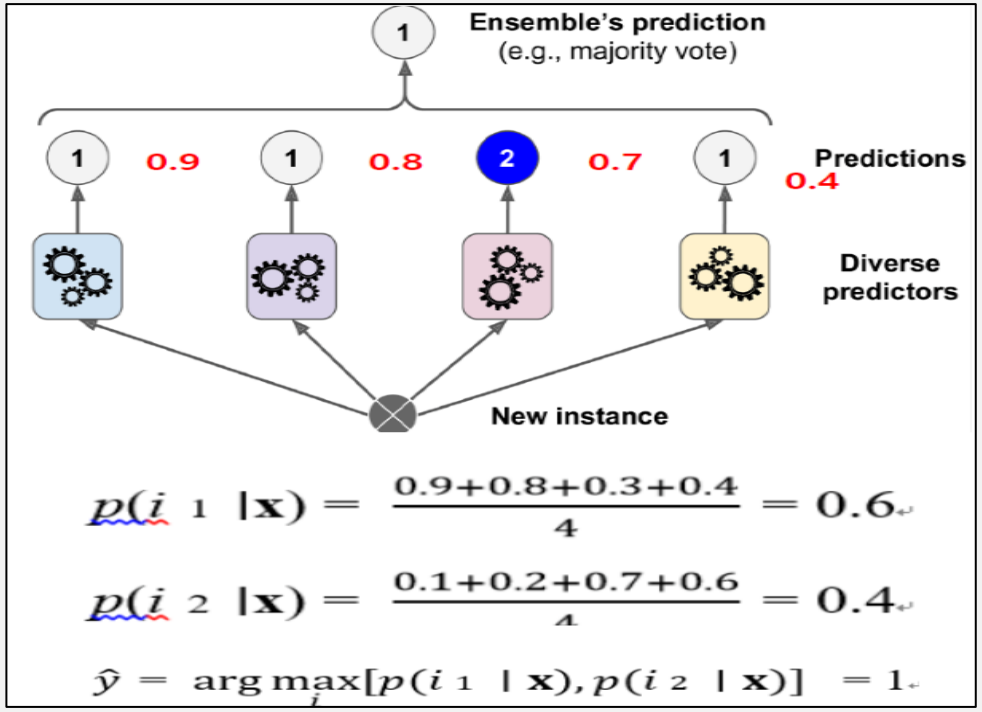
	정상	부도	비율
Logistic	2893	44	65.75 : 1
LDA	2888	49	58.94 : 1
Adaboost	2873	64	61.49 : 1
Random Forest	Adaboost와 동일		
Decision tree	2781	156	17.83 : 1

정상기업으로 분류하는 비율이
너무 높음!!

Train set 비율과 제일 비슷하게 분류
But,
f1 score가 너무 낮아 단독 사용 불가

2. Modeling

2) 최종 모델 선정 - Voting Classifier



Adaboost, Randomforest, Decision tree
=> Soft voting classifier 사용

	정상	부도	비율
Voting Classifier	2790	147	18.98 : 1

```
from sklearn.ensemble import VotingClassifier
vc = VotingClassifier(estimators=[('abc', abc), ('rf', rf), ('dc', dc)], voting='soft').fit(X, y)
vc_predict = vc.predict(bank_test)
np.sum(vc_predict==0)
```

```
2790
```

3. 결과 및 해석

자료의 한계1: 재무정보 중 성장성의 부재

[성장성]

기업의 경쟁력과 미래의 수익창출능력의 지표

총자산 증가율(Growth rate of total assets)

기업에 투하된 총자산이 얼마나 증가하였는가를 나타내는 비율로서 기업의 전체적인 성장성을 측정하는 지표이다.

■ 총자산 증가율 $= \frac{\text{당기말 총자산}}{\text{전기말 총자산}} \times 100 - 1$

유형자산 증가율(Growth rate of tangible assets)

토지, 건물, 기계장치 등 유형자산에 대한 투자가 어느정도 이루어졌는가를 보여주는 지표.

■ 유형자산 증가율 $\frac{\text{당기말 유형자산}}{\text{전기말 유형자산}} \times 100 - 1$

유동자산 증가율(Growth rate of current assets)

기업의 경상적인 영업활동을 위하여 소유하는 유동자산이 얼마만큼 늘어났는가를 나타내는 지표이다.

■ 유동자산 증가율 $\frac{\text{당기말 유동자산}}{\text{전기말 유동자산}} \times 100 - 1$

3. 결과 및 해석

2) 자료의 한계2: 기업의 비재무정보 부재

[기업 신용평가모형에서 활용되어 온 데이터 항목]

구분				주요 데이터 항목
정형 데이터	기업 내부 요인	재무정보		• 재무제표(재무상태표, 손익계산서, 현금흐름표) • 재무지표(성장성, 활동성, 수익성, 유동성 등의 지표)
		비재무 정보		• 평가자의 주관적인 판단에 의한 정성적 위험요소를 계량화(경영위험, 영업위험, 산업위험 등) • 기업 개황, 경영자 능력, 업력, 회사규모 등
	기업 외부 요인		금융시장 정보	• 주가, 거래량, 종합주가지수, 변동성지수, 시가총액, 주가 관련 비율 등 • 회사채 가격, 채권 관련 비율 등
			거시경제 지표	• GDP, 산업생산지수, 설비가동률, 생산능력지수 등 경기변동 및 기업 생산성 판단 지표 • 기업경기실사지수 등 경기전망에 대한 심리 지수 • 금리, 환율 등 거시경제 관련 정보
비정형 데이터		뉴스, SNS 등의 텍스트 정보		• 뉴스나 SNS 등에서 추출한 기업 관련 키워드 정보

- 실제 기업의 신용평가 모형에 사용되는 정보는 재무정보에 국한되지 않음
- 비재무정보 뿐만 아니라 미디어 관련 정보도 매우 중요!

Q&A