26기 겨울 방학세미나

1팀

김지민
문서영
심은주
황유나

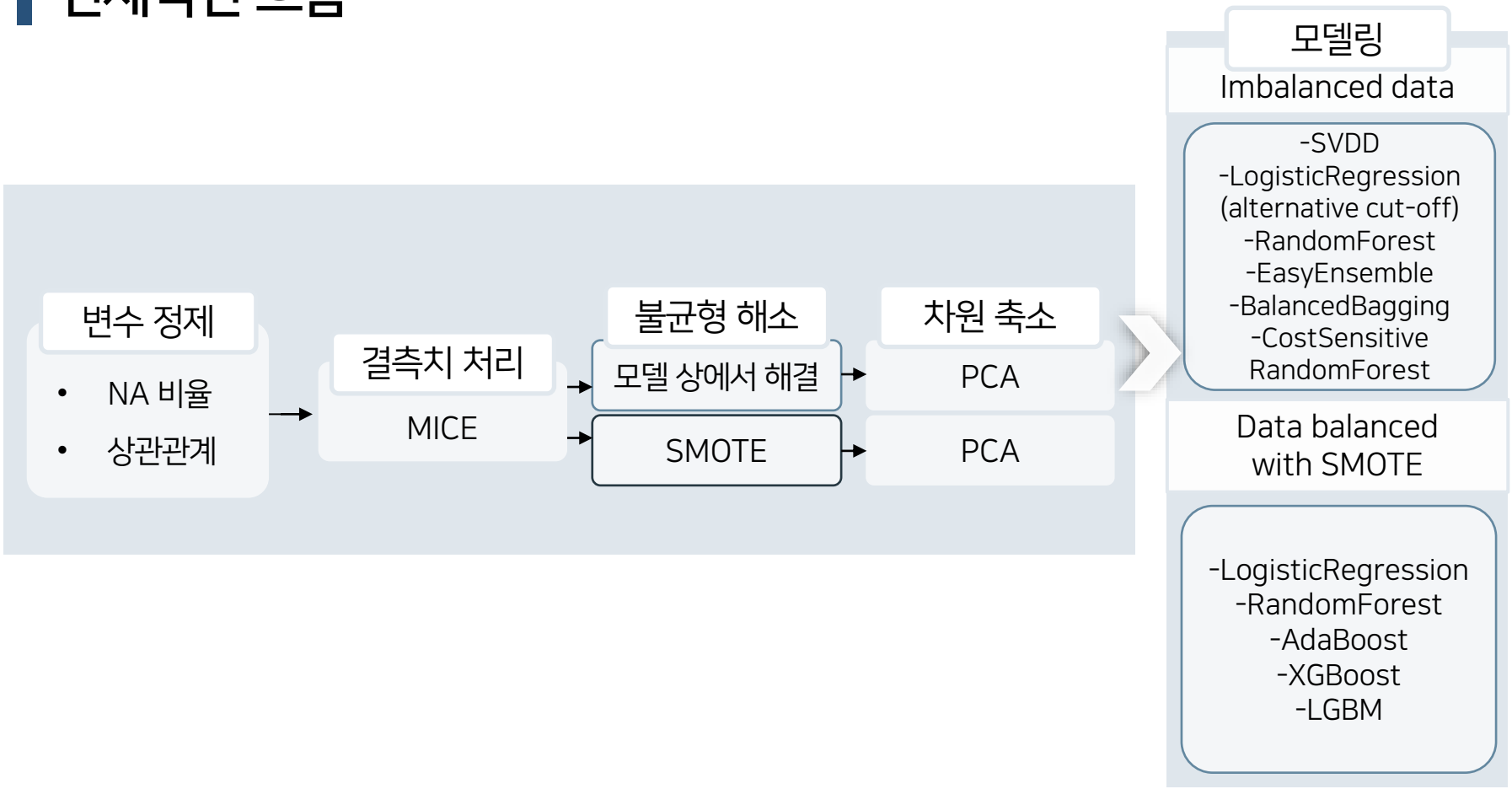
INDEX

1

데이터 전처리

데이터 전처리

전체적인 흐름



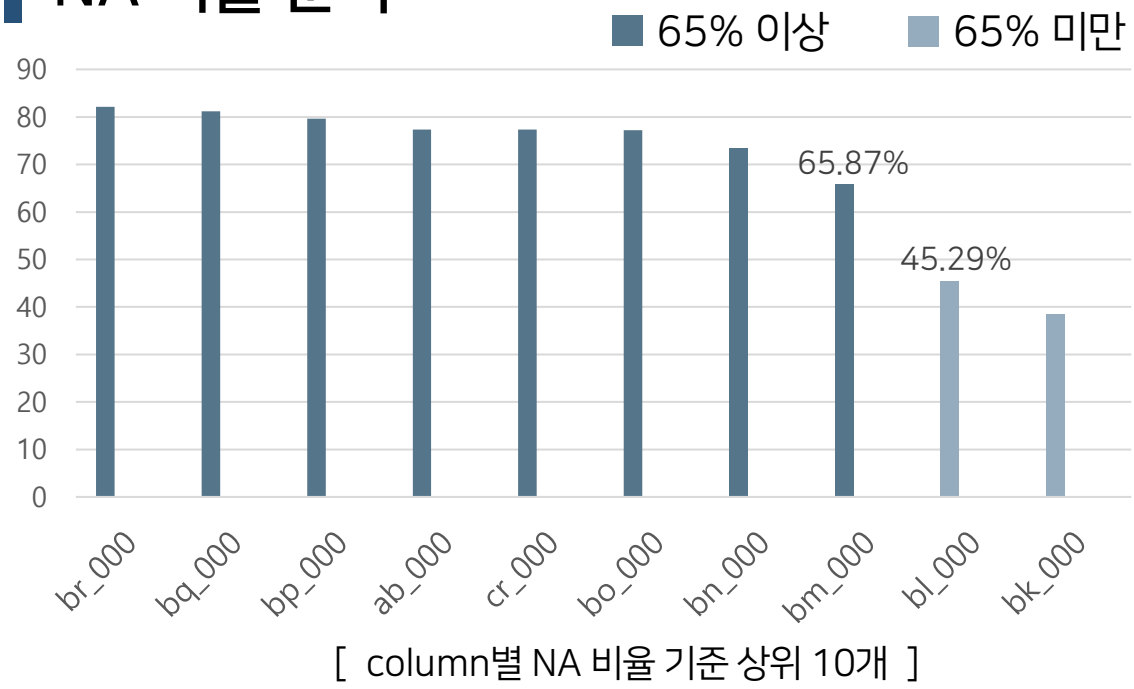
1 데이터 전처리

‘class’ 변수 형식 처리

pos → 1 neg → 0

class	aa_000	ab_000	ac_000
0	78696	NaN	0.000000e+00
1	153204	0.0	1.820000e+02
0	39196	NaN	2.040000e+02

NA 비율 분석



[column별 NA 비율 기준 상위 10개]

타겟변수 1개 | 독립변수 170개

변수 171개의 NA 비율을

내림차순으로 파악

8위(bm_000)와

9위(bl_000) 간

NA 비율 차가 큼

NA 비율을 기준으로 변수 처리

NA가 매우 많음 + 각 변수들의 정확한 의미를 알 수 없음

위의 두 가지를 모두 고려해 NA 비율을 근거로 하여 불필요한 변수를 처리하되,

그 기준을 65%로 높게 두어 주요 변수의 무차별적 제거 방지

8개의 변수 제거

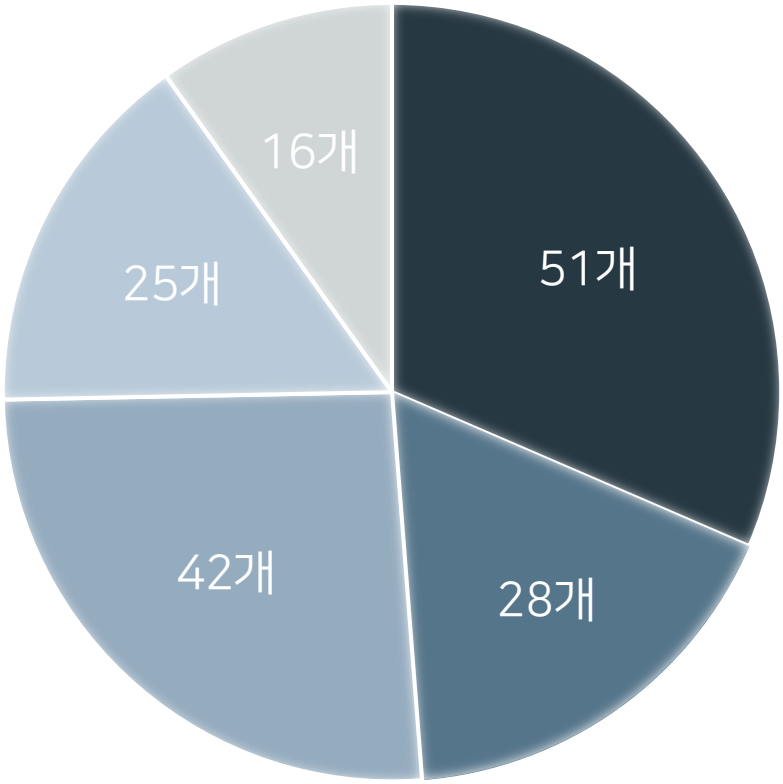
br_000, bq_000, bp_000, ab_000, cr_000, bo_000, bn_000, bm_000

class	aa_000	ac_000	ad_000	...	ef_000	eg_000
0	41040	2.280000e+02	100.0		0.0	0.0
0	12	7.000000e+01	66.0		4.0	32.0
0	60874	1.368000e+03	458.0		0.0	0.0

40000 X 163

데이터 전처리

상관관계 분석



타겟변수 1개 | 독립변수 162개

변수 163개 간의 상관관계를 파악하되,

상관행렬 plot은 변수명의 맨 앞

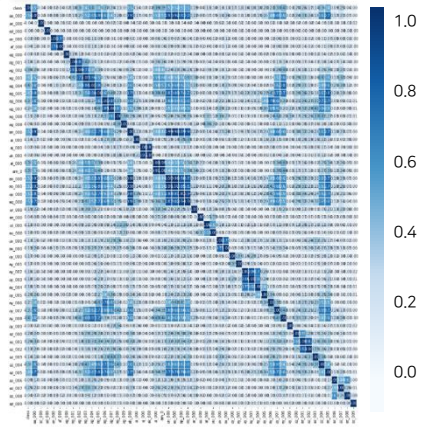
알파벳을 기준으로 분류해 도출

- ‘a’로 시작하는 변수
- ‘b’로 시작하는 변수
- ‘c’로 시작하는 변수
- ‘d’로 시작하는 변수
- ‘e’로 시작하는 변수

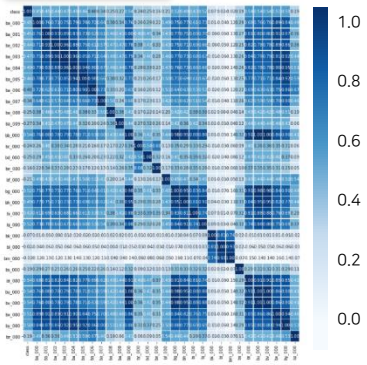
데이터 전처리

상관행렬 plot

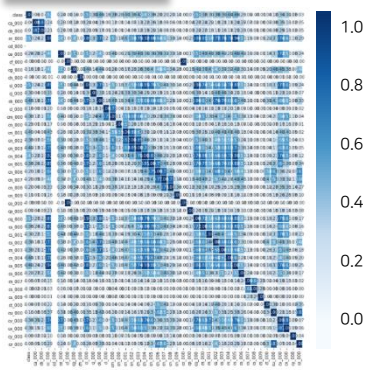
a -



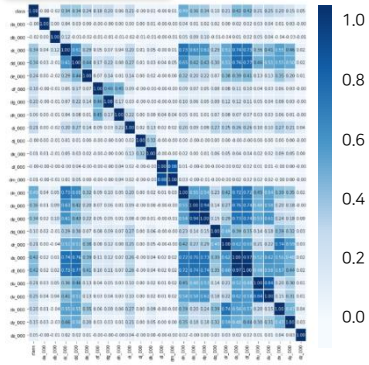
b -



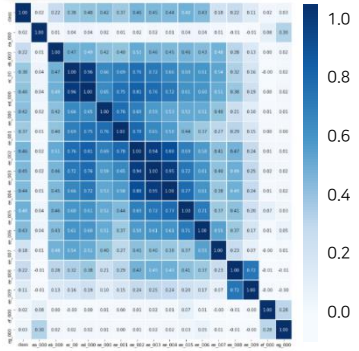
c -



d -



e -



상당히 많은 수의 변수들끼리

규칙적으로, 혹은 불규칙적으로

강한 상관관계를 보이는 모습

상관관계를 기준으로 변수 처리

강한 상관관계를 가진 변수들이 많아 정제 필요 + 각 변수들의 정확한 의미를 알 수 없음

위의 두 가지를 모두 고려해 **상관관계**를 근거로 하여 일차적 독립변수 처리를 진행하되,

그 기준을 0.9로 높게 두어 주요 변수의 무차별적 제거 방지

(-0.9의 상관계수를 가진 경우는 없었음)

a - 45개 (-6)　b - 15개 (-13)　c - 30개 (-12)　d - 21개 (-4)　e - 12개 (-4)

class	aa_000	ac_000	ad_000	...	ef_000	eg_000
0	41040	2.280000e+02	100.0		0.0	0.0
0	12	7.000000e+01	66.0		4.0	32.0
0	60874	1.368000e+03	458.0		0.0	0.0

40000 X 124

결측치 처리 : MICE

MICE 다중 대체법의 한 종류로,
결측값이 여러 변수에 걸쳐 존재할 경우 좋은 성능을 보이는 기법

Multivariate Imputation via Chained Equations

Scikit-learn의 IterativeImputer를 사용하여 구현

```
print(train.isnull().values.any())
print(test.isnull().values.any())
#현재 두 개 데이터프레임에 대해 모두 na가 있는 상황
```

```
True
True
```

```
imputed_train.isnull().values.any()
```

```
False
```

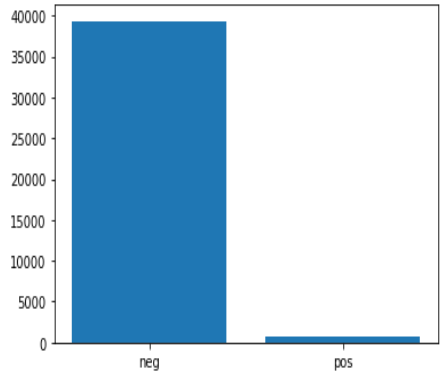
```
imputed_test.isnull().values.any()
```

```
False
```

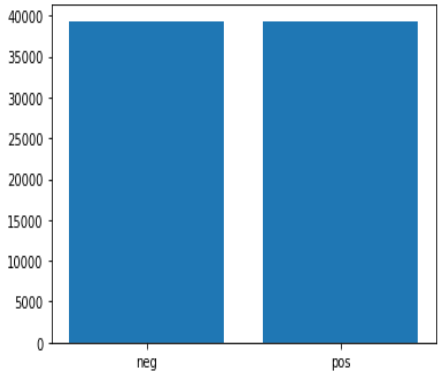
MICE를 진행한 결과 더 이상 NA 값을 지니지 않음을 확인

1 데이터 전처리

불균형 문제 해결 : SMOTE Synthetic Minority Over-sampling Technique

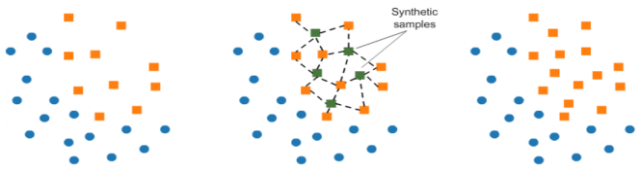


본 데이터는 약 98:2 비율의
imbalanced data



SMOTE 기법을 이용해
Over-sampling

SMOTE 알고리즘



1. 소수 클래스의 데이터 하나를 선택
2. 선택된 데이터와 가까운 소수 클래스 데이터에서 랜덤하게 k개 선택
3. 선택된 데이터와 k개의 데이터 사이의 가상의 직선 상에 소수 클래스 데이터 생성

분산이 0인 변수 제거

분산 = 0 = 모든 관찰값(truck air pressure system)이 같은 값을 가진다 > 변수 'cd_000' 제거

데이터 전처리

차원 축소 : PCA

: 고차원의 데이터 분포를 가장 잘 표현하는 성분들을 찾아 저차원의 데이터로 환원하는 기법

95% 분산 설명

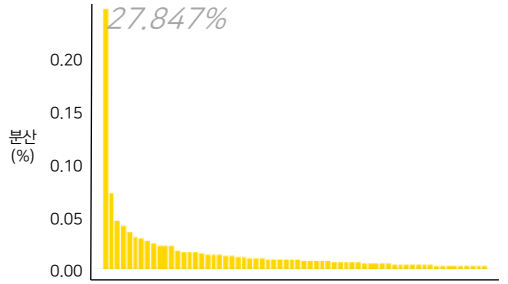
Imbalanced data

64개의 새 column으로 데이터의 변동성 설명

	PC1	PC2	PC3
class	0	1	2
0	-1.096905	-0.050946	-0.054705
0	-2.361722	0.335910	-0.181255
0	-0.028670	-0.325647	-0.134663

40000 X 65

[각 주성분으로 설명되는 분산]



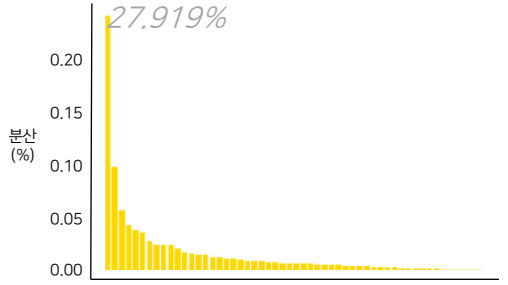
Data balanced by SMOTE

54개의 새 column으로 데이터의 변동성 설명

	PC1	PC2	PC3
class	0	1	2
0	-4.028291	-0.235768	-0.062710
0	-4.550281	-0.107800	-0.029269
0	-3.474246	-0.265006	-0.010141

78666 X 55

[각 주성분으로 설명되는 분산]



2

모델링

2 모델링

imbalanced data

Logistic Regression with Alternative cut-off

0.5가 아닌 최적의 cut-off를 탐색하여 새롭게 분류

One-Class SVM

최적의 서포트 벡터를 구하고 이 영역 밖의 데이터들은 outlier로 간주하는 비지도 학습 기법

SVDD

최소 최적의 초구체로 정상 데이터의 경계를 찾아냄

Easy Ensemble

random undersampling 수행, 여러 bootstrap sample들에 대해 Adaboost ensemble 적용

Balanced Bagging

random undersampling을 training단계에서 적용, base_estimator를 decision tree로 함

Cost Sensitive Random Forest

misclassification에 따른 cost를 고려함. random forest classifier를 base estimator로 둠

SMOTE data

Random Forest

AdaBoost

이전 분류기가 오분류한 샘플의 가중치를 유연하게 변경하여 약분류기들을 상호 보완하는 기법

XGBoost

Gradient Boosting 알고리즘 중 하나로, 병렬처리와 최적화를 장점으로 내세우는 알고리즘

LGBM

leaf-wise 트리분할을 사용해, 빠른 학습과 예측 수행 시간을 자랑하는 알고리즘

모델링

시도한 모델

공통적으로 사용한 evaluation method

5개의 confusion matrix를 합한
최종 행렬로 **총 비용 계산**

+

5개 F1 score의 평균값으로
최종 **F1 score** 도출

Cost function

```
def our_cost(fp,fn):
  cost=10*fp+500*fn
  return cost
```

모델링

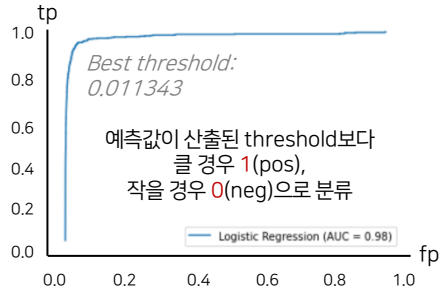
시도한 모델

imbalanced data

Logistic Regression with Alternative cut-off

5-fold cv

각 차례마다 최적의 threshold 선정



Validation dataset 40000개에 대해

F1-score:0.445
cost: 40860

One-Class SVM

5-fold cv

validation set으로 gamma를 튜닝.
nu의 경우, 데이터에서 'pos'의 비율이 약 2%이기 때문에 0.98로 설정.

F1-score: 0.39
cost: 159650

SVDD

5-fold cv

validation set으로 gamma를 튜닝.
nu의 경우, 데이터에서 'pos'의 비율이 약 2%이기 때문에 0.98로 설정.

F1-score: 0.13
cost: 340810

모델링

시도한 모델

Easy Ensemble

5-fold cv

AdaBoost learner 10개,
sampling_strategy = 'auto' (not minority) 적용

Validation dataset 40000개에 대해

F1-score:0.336
cost: 36710

Balanced Bagging

5-fold cv

decision tree learner 10개,
sampling_strategy = 'auto' (not minority) 적용

F1-score:0.379
cost: 43260

Cost Sensitive Random Forest

5-fold cv

combination = 'majority_voting',
max_features = sqrt(n_features) 적용

F1-score:0.334
cost: 44580

2 모델링

시도한 모델

SMOTE data

Random Forest

Validation dataset 78666개에 대해

5-fold cv

파라미터 튜닝을 시도했으나 시간이 너무 오래 걸려, 임의의 파라미터로 진행

F1-score:0.9894
cost: 41270

AdaBoost

5-fold cv

DecisionTreeClassifier를 반복적으로 학습하며 유연하게 가중치를 변경하여 오분류된 샘플에 더욱 집중

F1-score:0.9556
cost: 825930

XGBoost

5-fold cv

파라미터 튜닝을 시도했으나 시간이 너무 오래 걸려, 임의의 파라미터로 진행

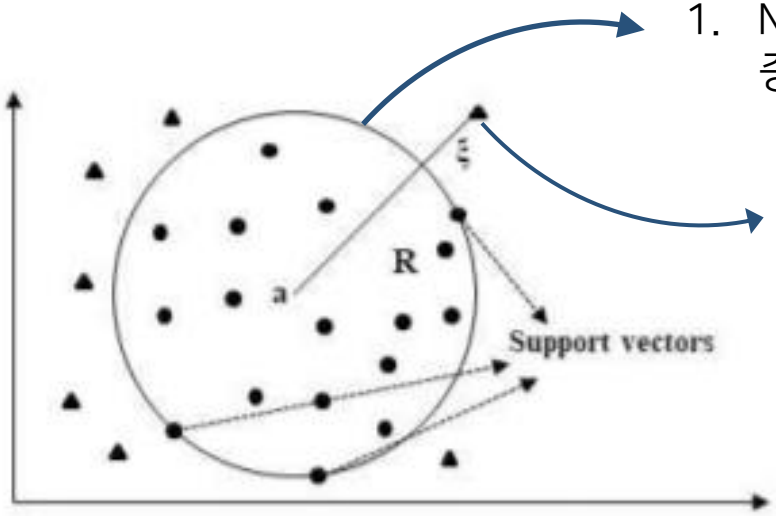
F1-score:0.9727
cost: 325010

시도한 모델

SVDD

Support Vector Data Description

One-class SVM의 일종으로, 데이터를 감싸는 최소 체적의 초구체(Hypersphere)를 찾아 데이터의 경계(boundary)를 설명하는 방법.



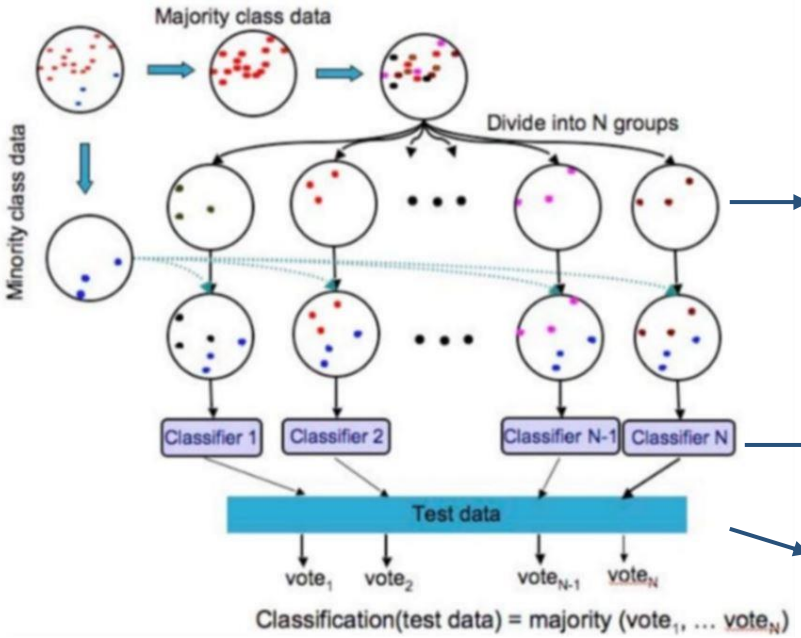
1. N차원 공간에 D개의 데이터에 대하여 중심이 a, 반지름이 R인 구를 형성
2. 구의 범위 밖에 존재하는 데이터는 slack variable로 간주 → negative
3. 구의 부피는 trade-off constant에 의해 조절

시도한 모델

Easy Ensemble

majority class의 sample들을 여러 개의 subset으로 구성한 뒤,

각 subset에 대해 train 시행. 최종 단계에서 이들 결과를 합산하는 방법.



1. majority class의 sample들을 n개의 subset으로 나누기
2. 이들 각각에 대해 classifier 적용, train
3. test data에 대해 classifier 적용, 결과 합산

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2009. Exploratory undersampling for class-imbalance learning. Trans. Sys. Man Cyber. Part B< 39, 2 (April 2009), 539–550. DOI:https://doi.org/10.1109/TSMCB.2008.2007853

시도한 모델

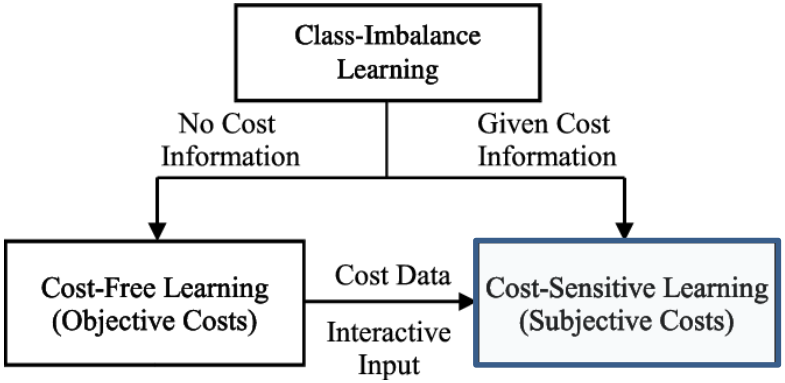
Cost-Sensitive Random Forest

Cost-Sensitive Learning

accuracy 최적화보다,
misclassification cost를 최적화하는 데 중점을 둔다.

분류 결과에 따른 cost 정보가 주어졌을 때 (class dependent cost),

이들 정보를 고려하여 효과적으로 분류를 수행하여 optimal solution을 산출하는 방법.



실데이터에서 false negative cost와
false positive cost가 다르게 정의된다는 점을 반영

dataset rebalancing을 통해 cost-sensitivity 확보

Algorithm 2 Cost Sensitive Reduction to Binary Case

```
procedure REDUCE(L, X, c)                          ▷ Learner L, Data X, cost matrix c
    for i ∈ {1, ..., κ − 1} do
        for j ∈ {i + 1, ..., κ} do
            X(i, j) ← {x(α) ∈ X | y(α) ∈ {i, j}}
            (c)(i, j) ← (i, j) − minor of c
            w(i), w(j) ← TwoClass(X(i, j), (c)(i, j))      ▷ from Algorithm 1
            M(i, j) ← L(X(i, j), w(i), w(j))               ▷ M(i, j)(x) ∈ {i, j}
        end for
    end for
    Define M by x ↦ Mode({M(i, j)(x) | i, j ∈ {1, ..., κ}, i < j})
    return M
end procedure
```

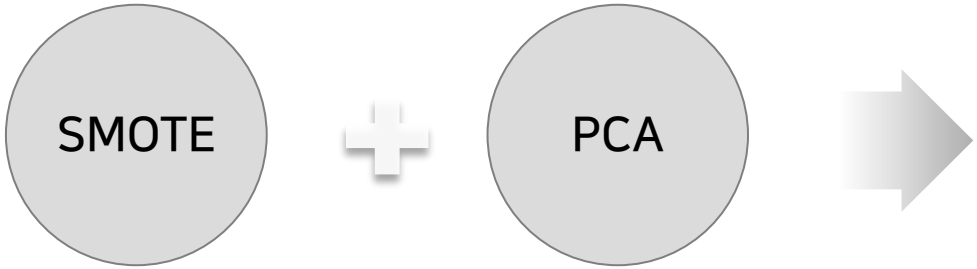
Charles Elkan. 2001. The foundations of cost-sensitive learning. In Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2(IJCAI'01). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 973–978.

3

최종 예측

3 최종 예측

최종 모델



	PC1	PC2	PC3
class	0	1	···
0	-4.028291	-0.235768	···
0	-4.550281	-0.107800	···
0	-3.474246	-0.265006	···

78666 X 55

SMOTE와 PCA를 이용한
78666개의 행 X 55개의 열로 이루어진 데이터

LGBM

Light Gradient Boosting Model

Parameter	value
learning_rate	0.3
max_depth	-1
metric	auc
num_leaves	31
feature_fraction	0.9
bagging_fraction	1.0
num_iterations	600

validation set(78666개)을 통해 얻은

F1-score : 0.9953

Cost : 9130

의의 및 한계

의의

- 지난 학기 PSAT에서 배운 내용들을 **엄청 많이 써먹었다**!
- 팀원 전원이 파이썬을 통한 데이터 전처리 및 모델링 전 과정 **워크플로우를 완전히 이해할 수 있게 되었다**!
- **다양한 상황에서의 모델링 기법**들을 섭렵했다! 균형/불균형 상황에 적합한 기법들을 적용해보고 test해보았다!

한계

- 차원축소 방법인 PPCA(Probabilistic PCA)의 fit, transform 과정을 완전히 익히지 못해 test 데이터에 적용해보지 못함
- 다소 시간이 부족하여 모델링 (랜덤포레스트, XGBoost, LGBM) 과정에서 파라미터 튜닝을 하는데 한계가 있었으며, 다양한 샘플링 방법을 구현해보지 못했다는 점에서 아쉬움이 남음

THANK YOU