

불균형 제조 데이터 분석 결과 보고서

1주차

LS 빅데이터 스쿨 3기
3! 4!

발표 순서

1 불균형 데이터란?

2 데이터 전처리 / 불균형 해소

 2-1. 변수 선별

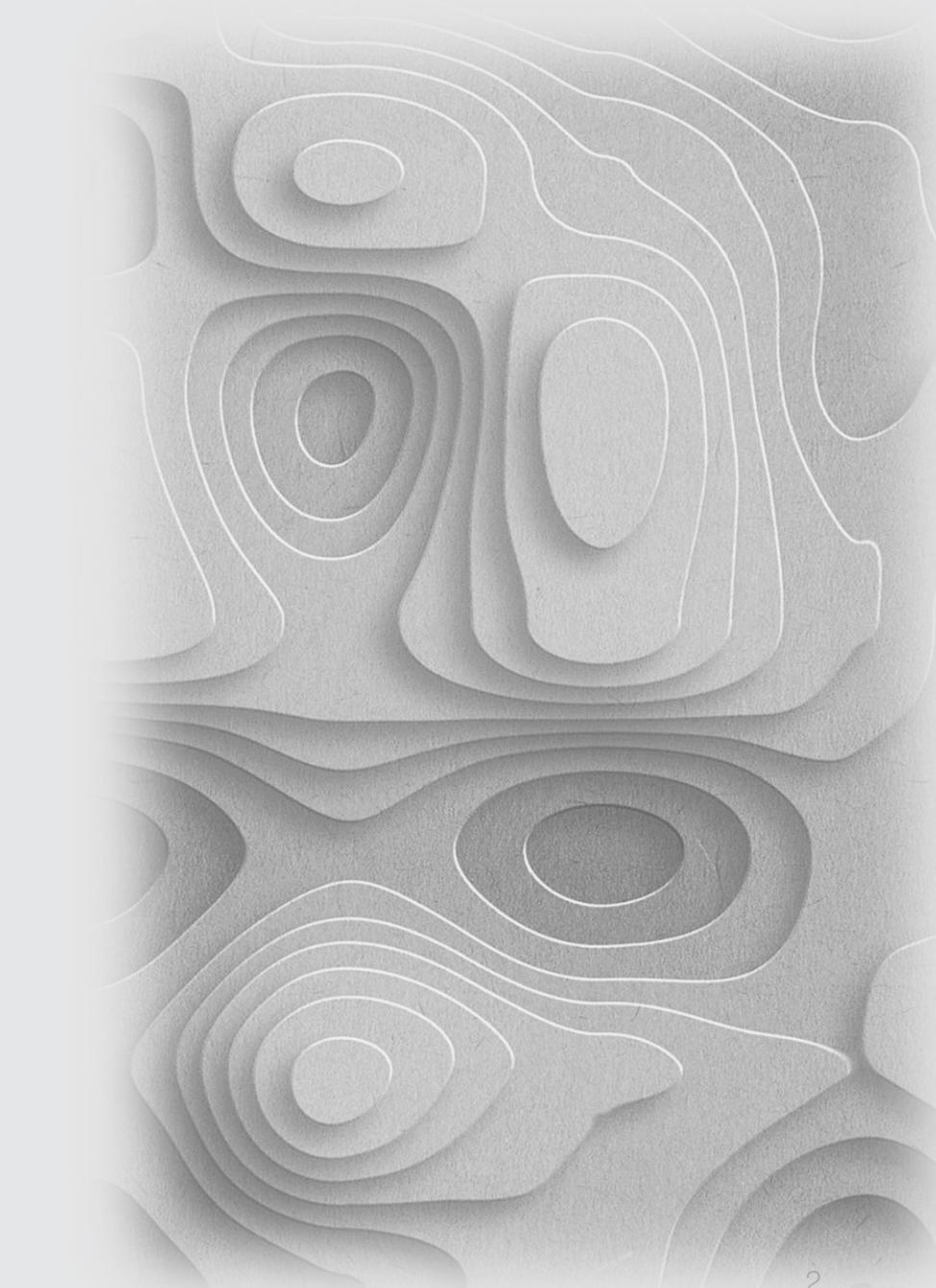
 2-2. 로그 / 제곱 변환

 2-3. 표준화

 2-4. 변수 조정

 2-5. 불균형 해소

 2-6. 모델 내장 매개변수



발표 순서

3

모델 생성 및 예측

3 - 1. 모델 및 평가 지표 선정

3 - 2. 성능 평가 및 최적 모델

3 - 3. 변수 중요도 분석

4

결론

4 - 1. 혼동 행렬 분석

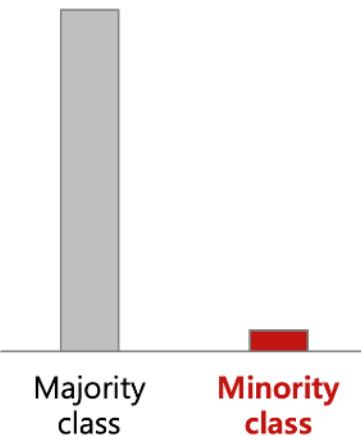
Part 1

불균형 데이터란 ?



불균형 데이터

두 개 이상의 클래스 중, 한 클래스의 데이터가 다른 클래스에 비해 상대적으로 많거나 적은 경우

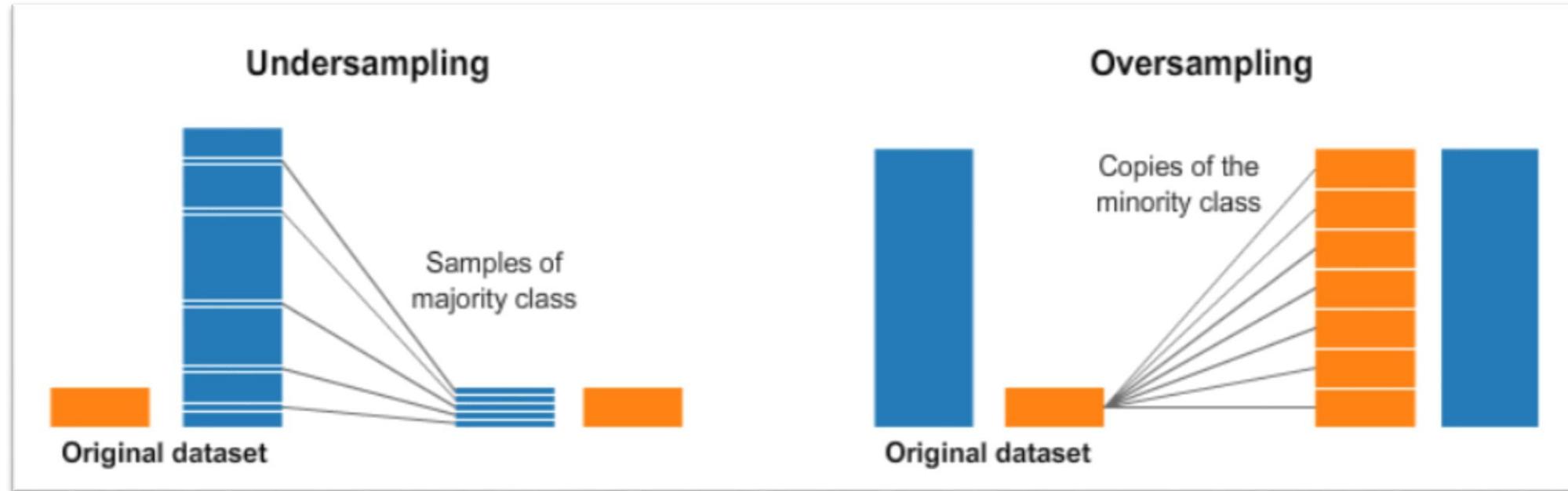
불균형 데이터 (imbalanced data)

분류 모델
(classification model)
훈련 어려움

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

모델 성능 저하

평가 지표(정확도) 왜곡



Resampling

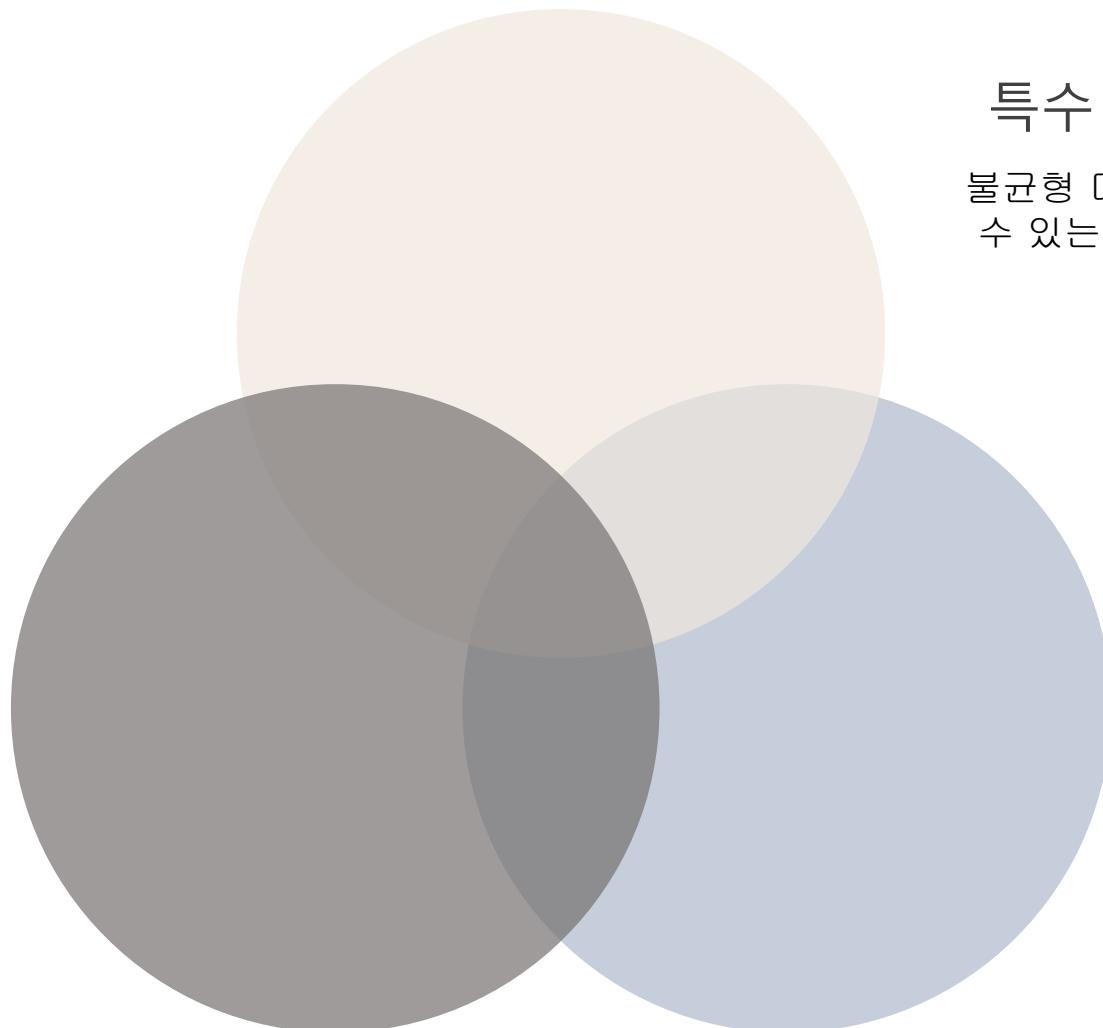
소수 클래스에 데이터를 추가하거나, 다수 클래스의 데이터를 줄이는 방식

적절한 평가 지표

F1-score, ROC-AUC, G-mean과 같은 지표로 성능 평가

Resampling

소수 클래스에 데이터를 추가하거나, 다수 클래스의 데이터를 줄이는 방식



특수 알고리즘

불균형 데이터를 처리할 수 있는 알고리즘 사용

적절한 평가 지표

F1-score, ROC-AUC, G-mean과 같은 지표로 성능 평가

Make Someth'ng,

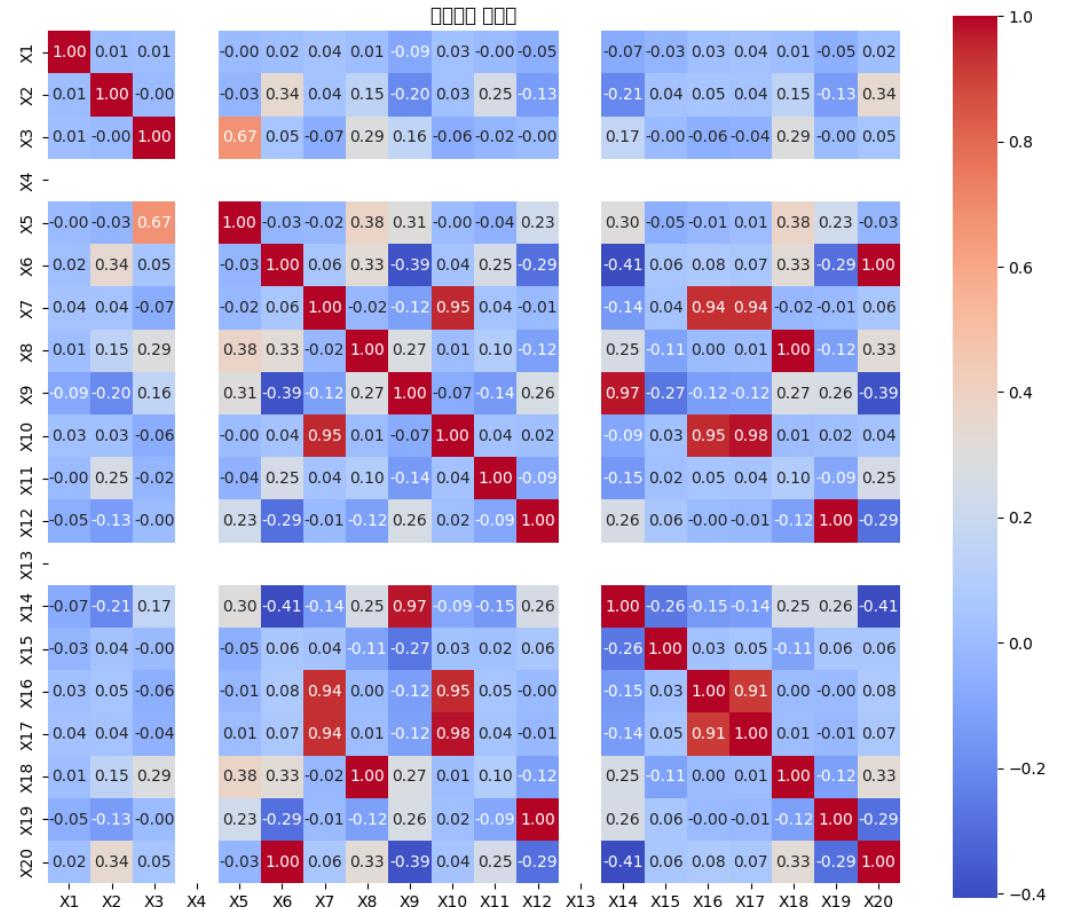
Someth'ng,

Part 2

데이터 전처리



히트맵 작성



변수 확인

X4

0	0.015348
1	0.015348
2	0.015348
3	0.015348
4	0.015348
	...
526995	0.015348
526996	0.015348
526997	0.015348
526998	0.015348
526999	0.015348

X13

0	0.249262
1	0.249262
2	0.249262
3	0.249262
4	0.249262
	...
526995	0.249262
526996	0.249262
526997	0.249262
526998	0.249262
526999	0.249262

X6 = X20

0	0.036360
1	0.067407
2	0.018944
3	0.031475
4	0.061888
	...
526995	0.015237
526996	0.021745
526997	0.068450
526998	0.020826
526999	0.028349

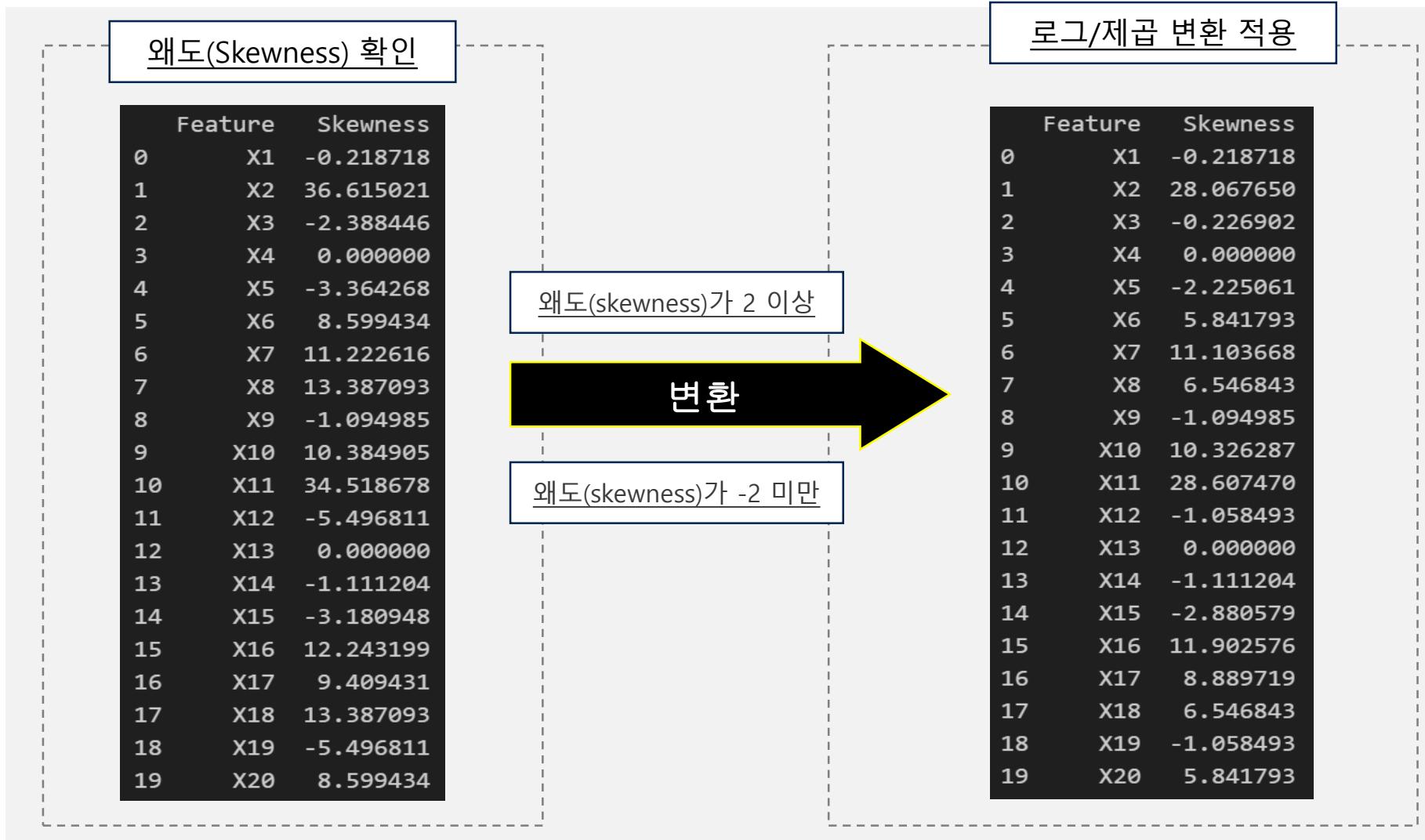
X8= X18

0	0.028087
1	0.028640
2	0.024502
3	0.025393
4	0.028450
	...
526995	0.020338
526996	0.024307
526997	0.028167
526998	0.024307
526999	0.025920

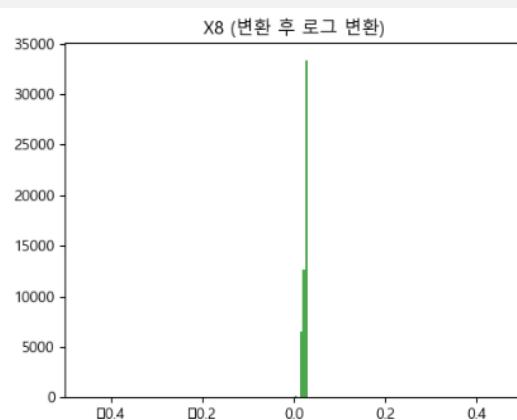
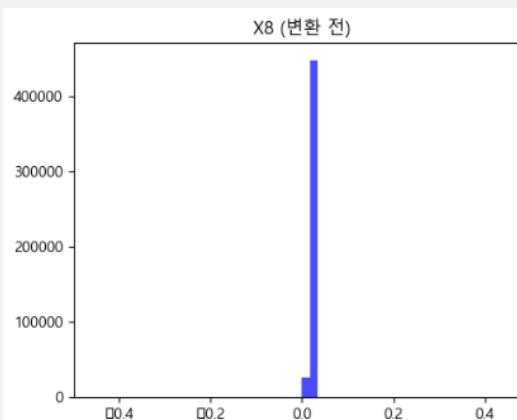
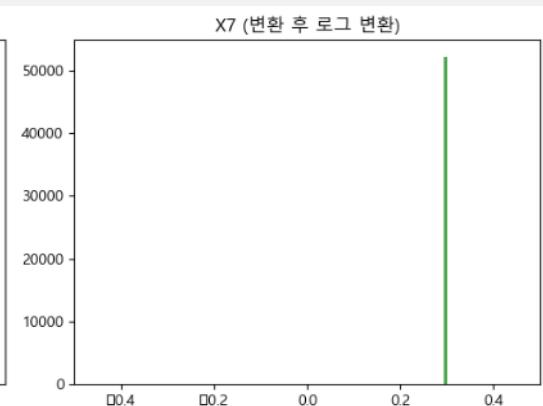
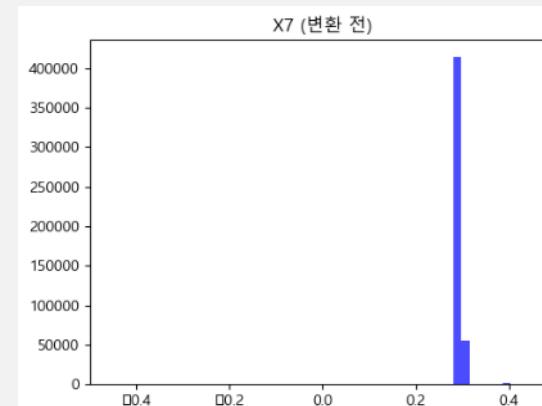
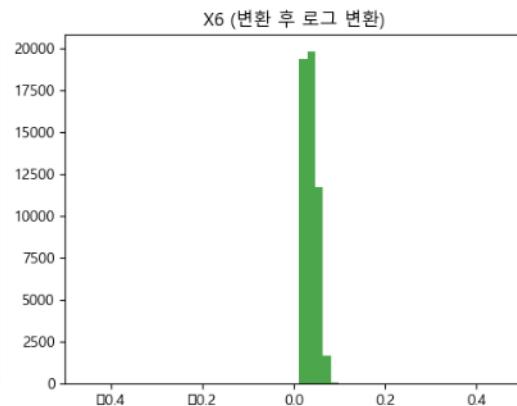
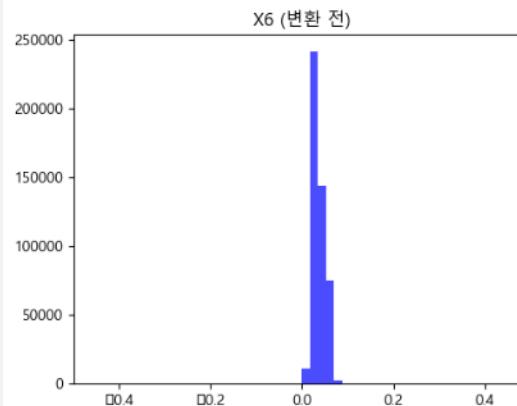
X12= X19

0	0.682731
1	0.680891
2	0.685525
3	0.677980
4	0.673286
	...
526995	0.679403
526996	0.687581
526997	0.680428
526998	0.689263
526999	0.683904

2 - 2 . 로그 / 제곱 변환



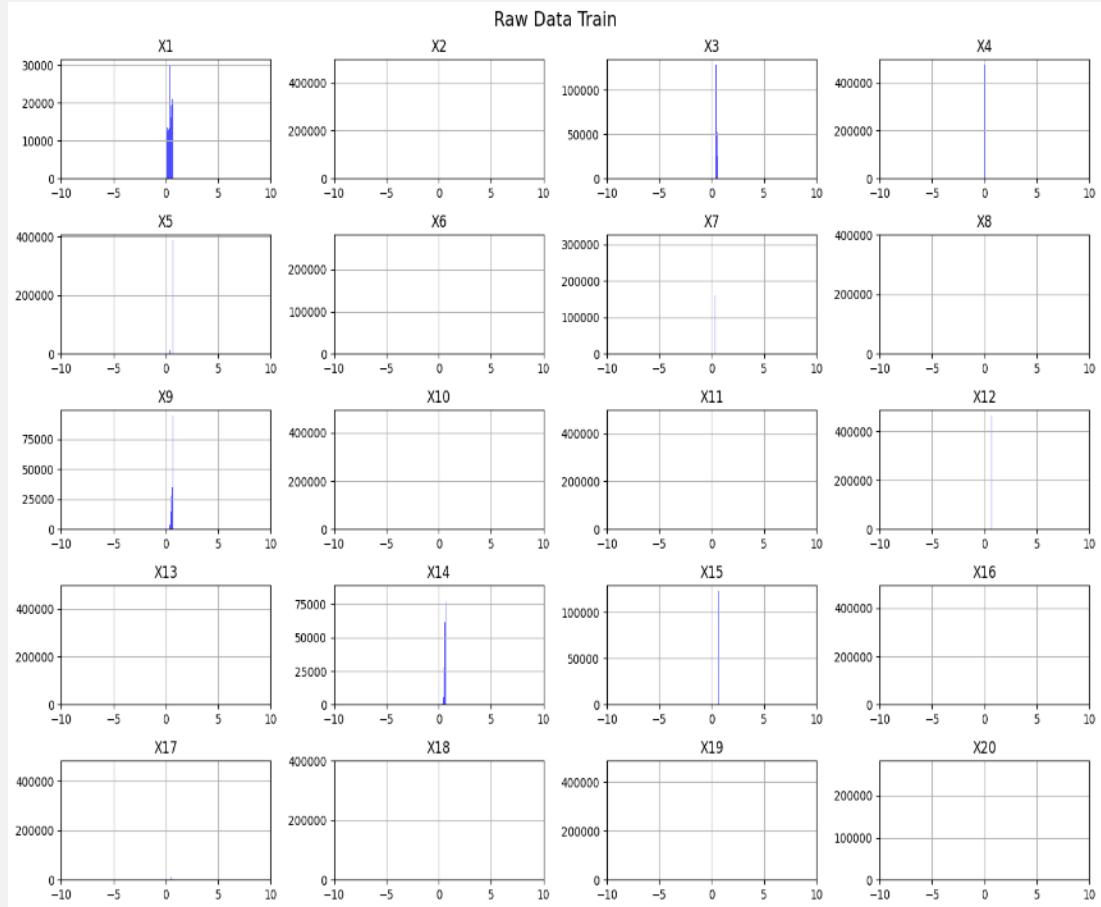
정규분포에 가까워졌을까?



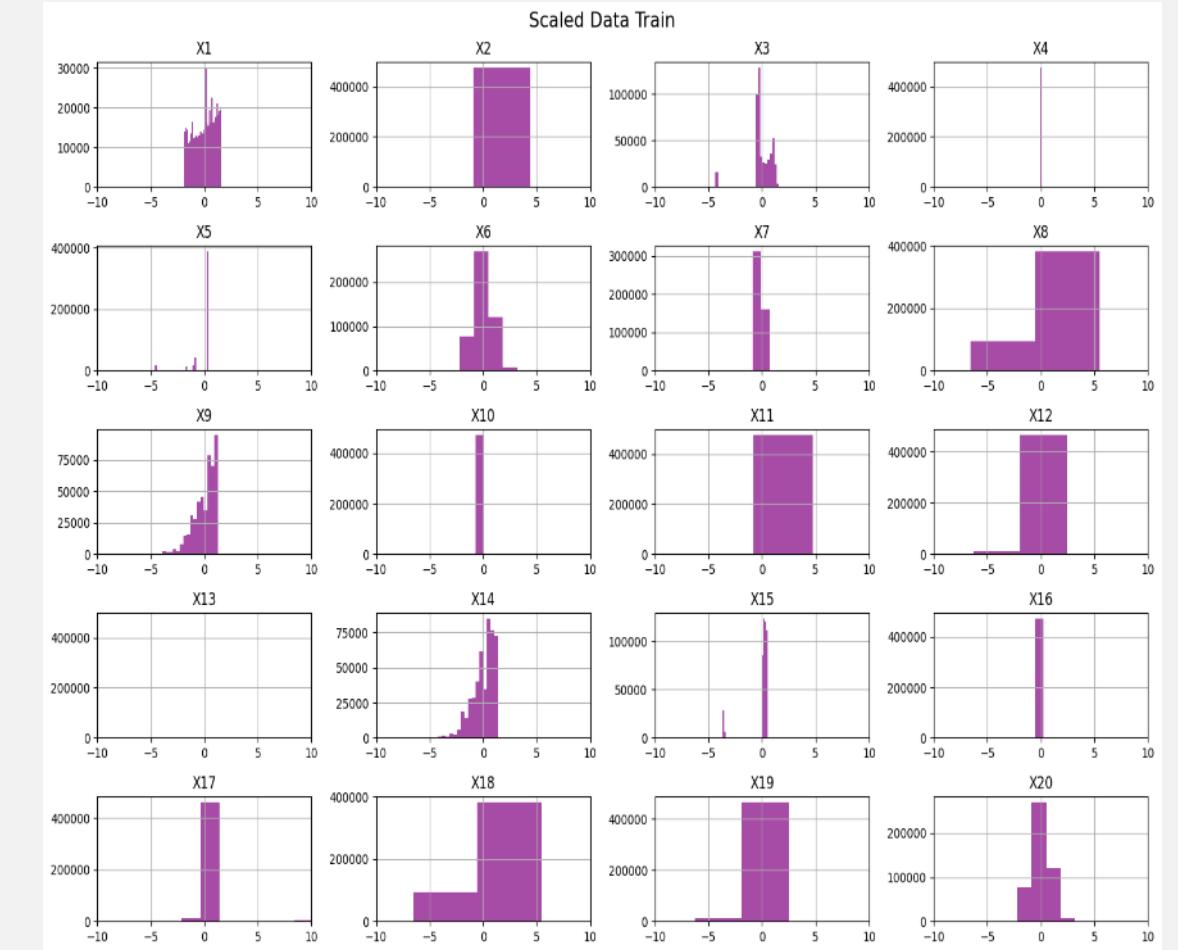
> 별로 변화가 없다
> 분류 모델의 경우 크게 영향이 없다

> 표준정규분포에 가까워짐
 > 분류 모델에 적용할 경우 성능에 영향이 거의 없음

적용 전



적용 후



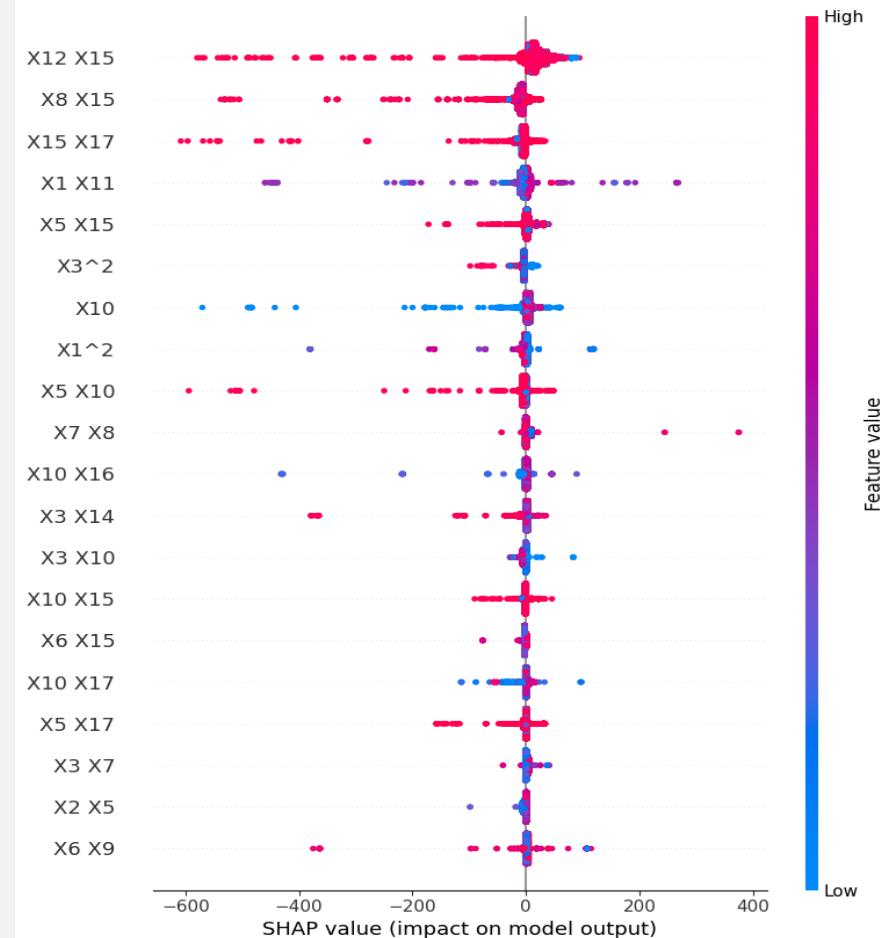
2 - 4 . 변수 조정 (Polynomial Features)

PolynomialFeatures 사용

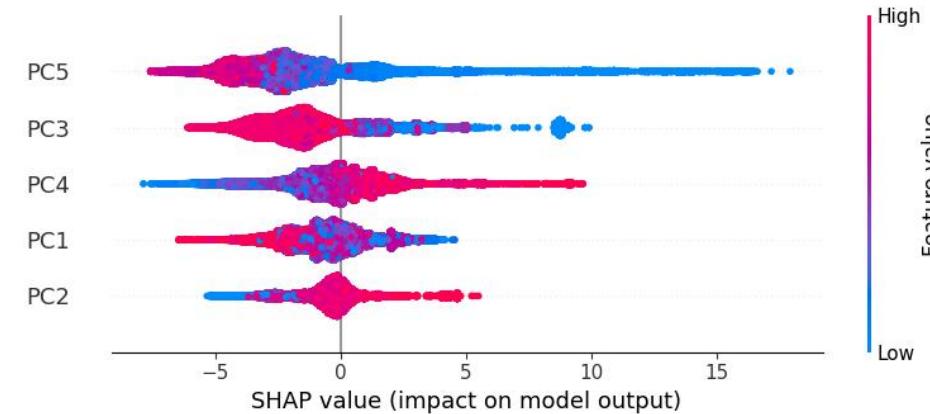
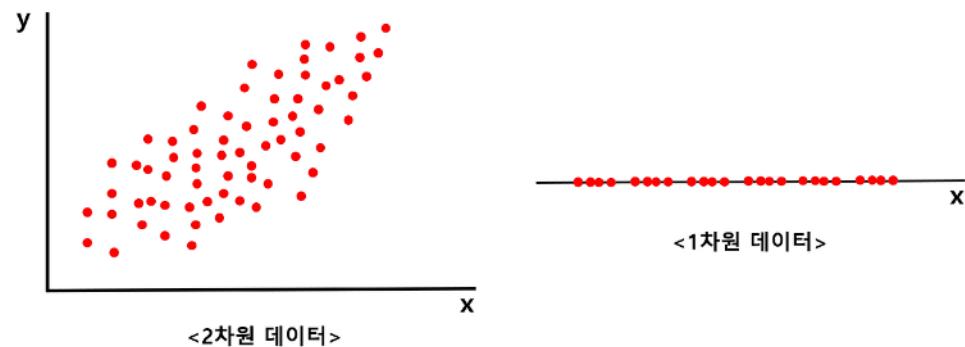
```
array(['X1', 'X3', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X12',
       'X14', 'X15', 'X16', 'X17', 'X1^2', 'X1 X3', 'X1 X5', 'X1 X6',
       'X1 X7', 'X1 X8', 'X1 X9', 'X1 X10', 'X1 X11', 'X1 X12', 'X1 X14',
       'X1 X15', 'X1 X16', 'X1 X17', 'X3^2', 'X3 X5', 'X3 X6', 'X3 X7',
       'X3 X8', 'X3 X9', 'X3 X10', 'X3 X11', 'X3 X12', 'X3 X14', 'X3 X15',
       'X3 X16', 'X3 X17', 'X5^2', 'X5 X6', 'X5 X7', 'X5 X8', 'X5 X9',
       'X5 X10', 'X5 X11', 'X5 X12', 'X5 X14', 'X5 X15', 'X5 X16',
       'X5 X17', 'X6^2', 'X6 X7', 'X6 X8', 'X6 X9', 'X6 X10', 'X6 X11',
       'X6 X12', 'X6 X14', 'X6 X15', 'X6 X16', 'X6 X17', 'X7^2', 'X7 X8',
       'X7 X9', 'X7 X10', 'X7 X11', 'X7 X12', 'X7 X14', 'X7 X15',
       'X7 X16', 'X7 X17', 'X8^2', 'X8 X9', 'X8 X10', 'X8 X11', 'X8 X12',
       'X8 X14', 'X8 X15', 'X8 X16', 'X8 X17', 'X9^2', 'X9 X10', 'X9 X11',
       'X9 X12', 'X9 X14', 'X9 X15', 'X9 X16', 'X9 X17', 'X10^2',
       'X10 X11', 'X10 X12', 'X10 X14', 'X10 X15', 'X10 X16', 'X10 X17',
       'X11^2', 'X11 X12', 'X11 X14', 'X11 X15', 'X11 X16', 'X11 X17',
       'X12^2', 'X12 X14', 'X12 X15', 'X12 X16', 'X12 X17', 'X14^2',
       'X14 X15', 'X14 X16', 'X14 X17', 'X15^2', 'X15 X16', 'X15 X17',
       'X16^2', 'X16 X17', 'X17^2'], dtype=object)
```

> 교호작용에 의해 파생된 변수들의 영향이 큰 것을 확인할 수 있음

변수 영향도



2 - 4 . 변 수 조 정 (PCA)



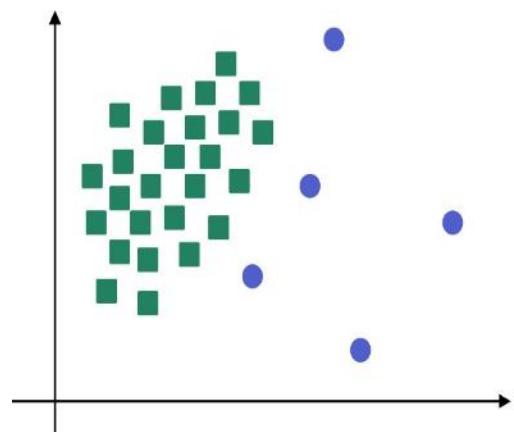
PCA를 사용하여 차원 축소

- ◉ 고차원 데이터를 저차원으로 변환하는 차원 축소 기법.
- ◉ 데이터의 분산을 가장 잘 설명하는 주성분(Principal Components)을 찾아, 중요한 정보는 유지하면서 데이터의 차원을 줄임.
- ◉ 데이터의 상관관계가 높은 피처를 줄이고, 잡음을 제거해 효율성을 높임.
- ◉ PCA는 기존 변수들을 조합하여 새로운 변수를 생성하기 때문에 모델을 설명하기 어렵다는 단점이 있음.

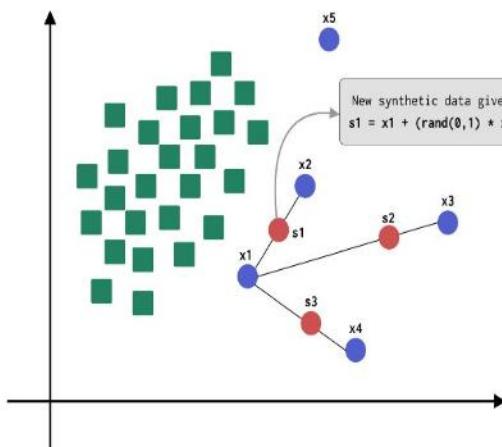
2 - 5 . 불균형 해소(오버샘플링)

SMOTE

새로운 데이터를 생성하여 소수 클래스의 데이터를 증가시키는 오버샘플링 기법

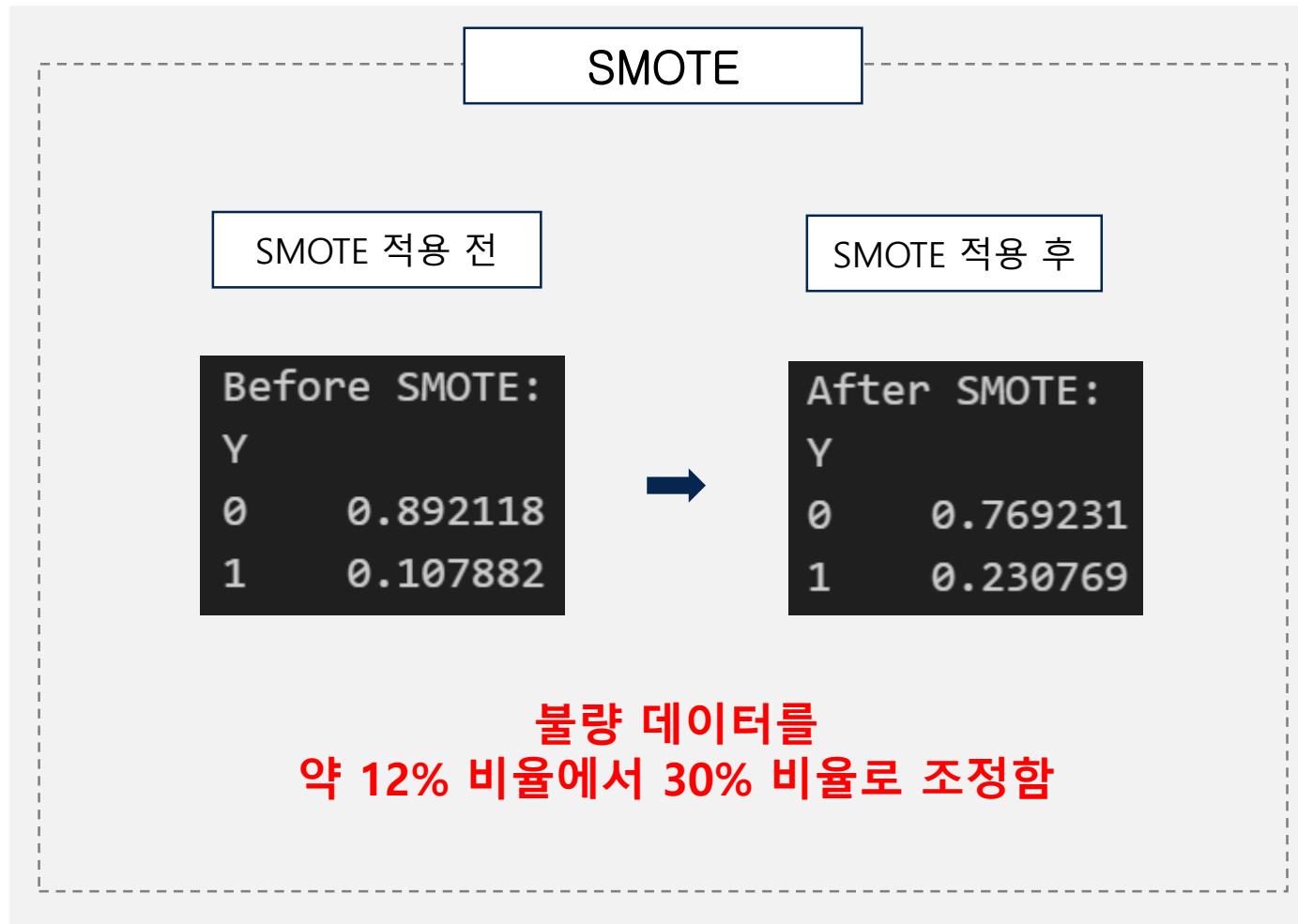


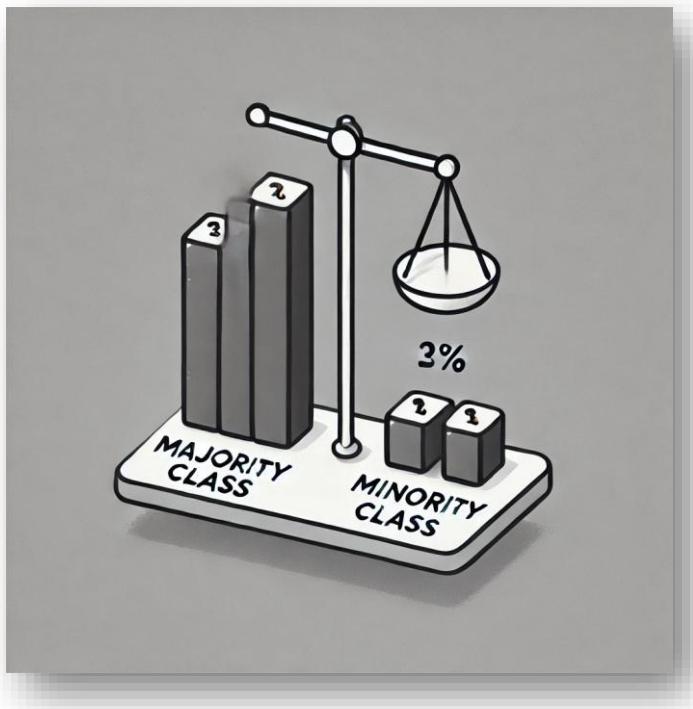
a) Class Imbalance



b) SMOTE

2 - 5 . 불균형 해소 (오버샘플링)





내장 매개변수

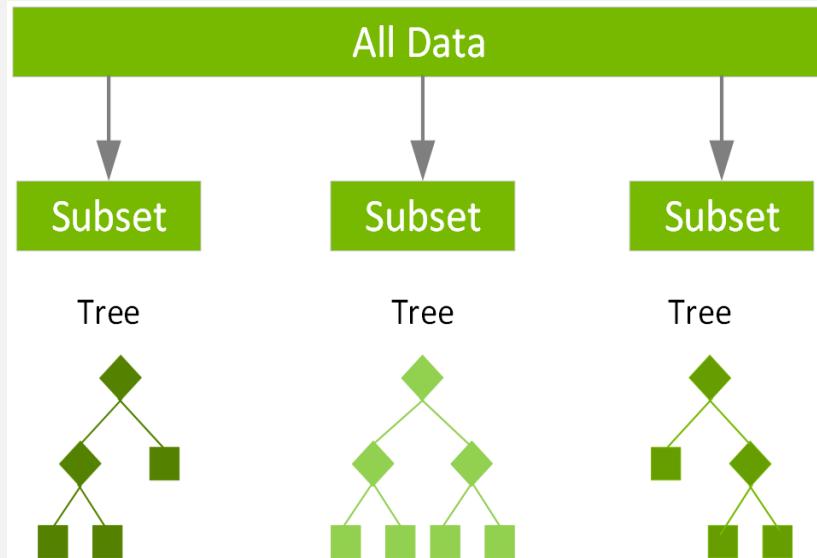
- xgboost – **scale_pos_weight**
- LogisticRegression – **class_weight**
- lightGBM – **scale_pos_weight**
- RandomForest – **class_weight**

모델이 손실 함수(loss function)를 계산할 때,
소수 클래스의 오류에 더 많은 가중치를 부여

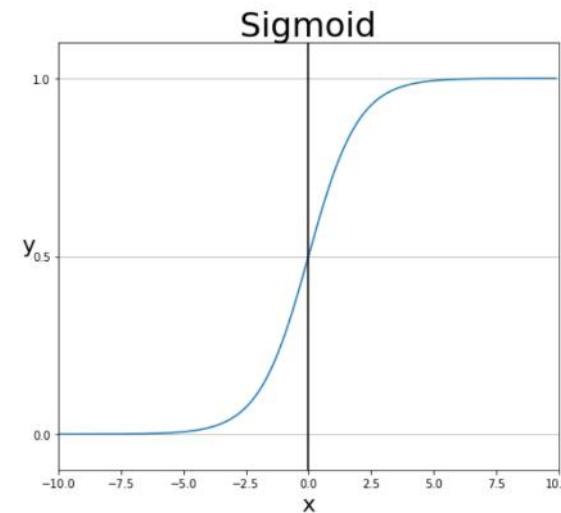
Part 3

모델 생성 및 예측



 XGBoost

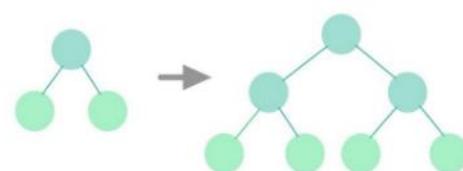
- 클래스의 가중치를 조정할 수 있어 불균형 데이터에 효과적으로 대응하며, 높은 예측 성능을 제공

 Logistic Regression

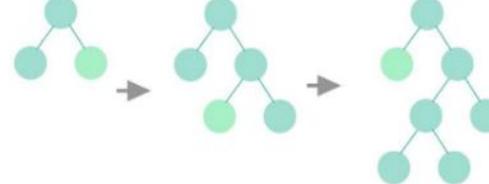
- 해석이 용이하고 빠른 속도로 모델을 구축할 수 있으며, 클래스 가중치를 설정하여 불균형 문제를 완화할 수 있음

☺ LightGBM

▣ 균형 트리 분할

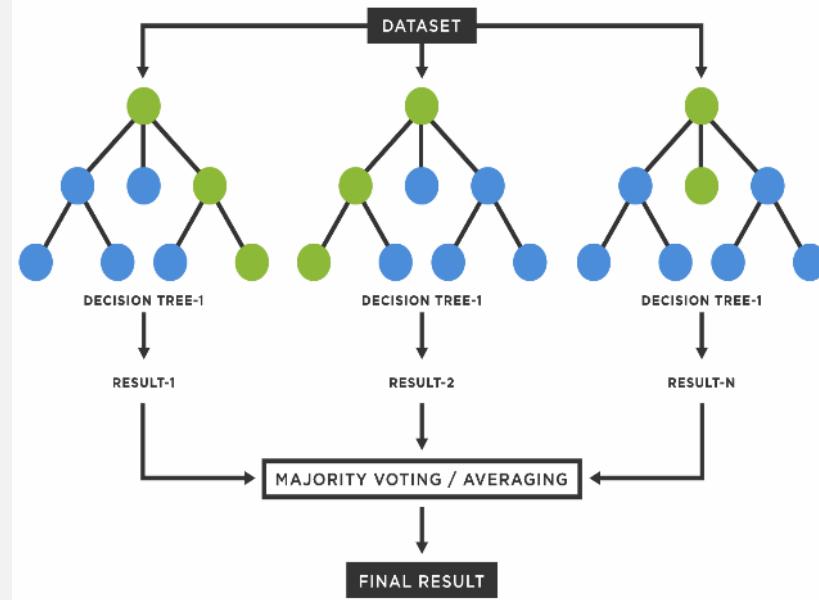


▣ 리프 중심 트리 분할



- 대규모 데이터셋을 처리할 수 있는 능력이 뛰어나고, 불균형 데이터에서 빠른 학습 속도를 제공

☺ Random Forest



- 랜덤 샘플링 기법을 활용하여 과적합을 방지하고, 불균형 데이터에서도 안정적인 성능을 유지

😊 F1-Score

- 정밀도와 재현율의 조화평균
- 불균형 데이터에서 전체적인 성능 표현
- 한쪽 클래스만 잘 맞추는 모델 방지

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

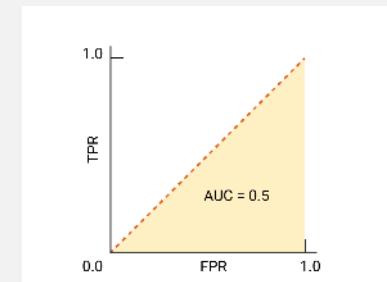
😊 G-mean

- 재현율과 정밀도의 기하평균
- 소수 클래스와 다수 클래스 모두에서 균형잡힌 성능 측정 가능
- 특정 클래스에 편향된 모델 방지

$$G\text{-Mean} = \sqrt{\text{Sensitivity (Recall)} \times \text{Specificity (TNR)}}$$

😊 AUC

- ROC 곡선 아래의 면적
- 양성 클래스와 음성 클래스를 얼마나 잘 구별하는지에 대한 수치적 지표



불균형 해소

SMOTE VS. Parameter



피쳐 조정

PCA(축소) VS.
Polynomial(확장)



모델 비교

XGB vs. LR vs. LGBM vs. RF

3 - 2 . 성 능 평 가 및 최 적 모 델 도 출

- 각 모델의 내장 파라미터와 SMOTE 적용 성능 비교

Table 1. Parameter VS. SMOTE

	model	xgboost	LR	lightGBM	RF
Evaluation Measures	Parameter (scale_pos_weight, class_weight)	F1	0.9997	0.9876	0.9997
		G-mean	0.9988	0.9939	0.9989
		AUC	0.9997	0.9983	0.9999
	Mean	0.9994	0.9933	0.9995	0.9990
SMOTE		F1	0.9997	0.9899	0.9997
		G-mean	0.9989	0.9915	0.9989
		AUC	0.9998	0.9979	0.9998
	Mean	0.9995	0.9931	0.9995	0.9992

3 - 2 . 성 능 평 가 및 최 적 모 델 도 출

- P C A 와 P o l y n o m i a l F e a t u r e s 적용 성 능 비 교

Table 2. PCA VS. Polynomial **SMOTE**

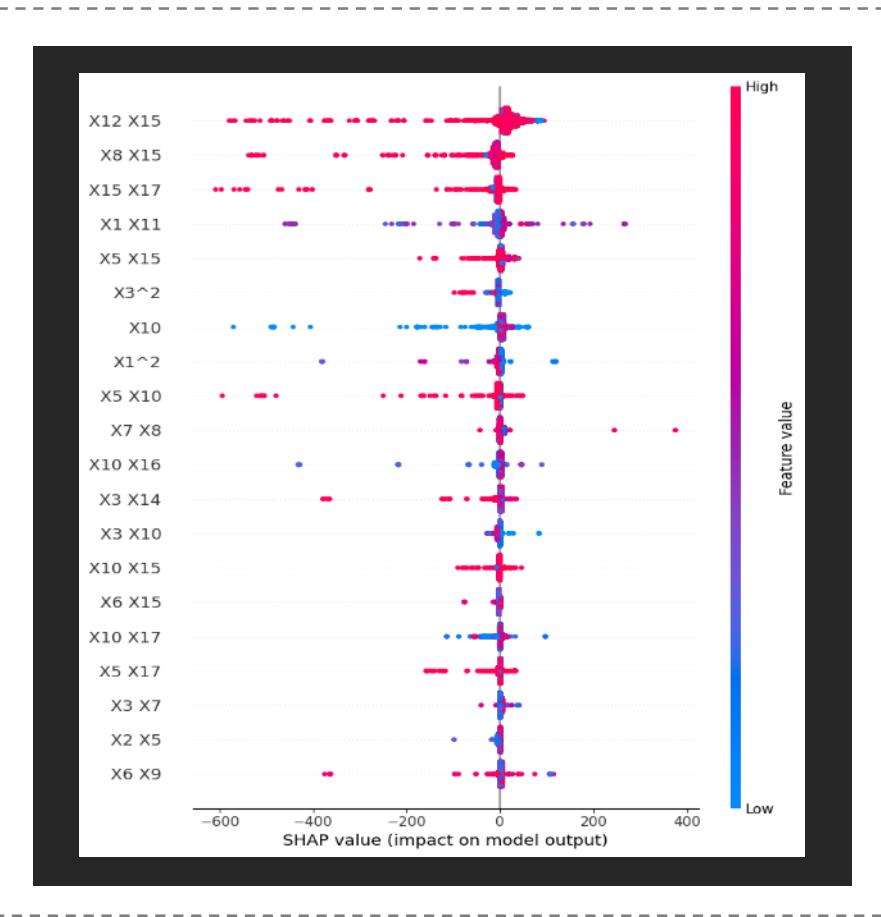
model		xgboost	LR	lightGBM	RF
Evaluation Measures	PCA	F1	0.9997	0.9981	0.9197
		G-mean	0.9992	0.9926	0.9302
		AUC	0.9998	0.9983	0.9303
		Mean	0.9995	0.9963	0.9267
Polynomial		F1	0.9997	0.9963	0.9997
		G-mean	0.9990	0.9835	0.9990
		AUC	0.9997	0.9978	0.9998
		Mean	0.9994	0.9925	0.9995



:: 모델 도출 근거

- ☺ 모델의 매개변수보다 **SMOTE**를 적용하였을 때의 성능이 높았음.
- ☺ PCA와 PolynomialFeatures 중 **PolynomialFeatures**를 사용하여 파생변수를 생성하였을 때 모델의 설명 측면에서 유리함
- ☺ XGBoost 모델과 LightGBM모델을 사용하였을 때 가장 성능이 높았고, 둘 중 분석시간이 더 짧은 **LightGBM**모델을 최종 모델로 선정

Shap 사용



어떤 항목이 불량률을
예측하는데 영향을 주는지?

☺ X12 X15와 X8 X15와 같은 파생변수
들이 모델의 성능에 중요한 기여를 함.

☺ 이는 단일 변수보다는 “변수 간 상호
작용”이 더 복잡한 패턴을 포착하여 모
델 예측 성능을 높였다는 것을 시사함.



결론





LightBGM + Polynomial Features + SMOTE

😊 음성 클래스는 모두 정확히 예측했으며, 양성 클래스에서도 매우 높은 정확도를 보였음.

😊 FN이 22개로 매우 적고, FP는 없었기 때문에 정밀도와 재현율이 모두 우수한 것으로 확인.

Q&A

