

34기 방학세미나

1팀

김수진
김재원
김준영
이나연
이채은

CONTENTS

0. INTRO

1. EDA 및 전처리

2. 모델링

3. 결과 및 해석

분석 목표 및 기준

TP(True Positive): 양성을 양성으로 예측한 경우
FN(False Negative): 양성임에도 음성으로 예측한 경우

Score Function

$$\begin{aligned} & -10 \times FN(\text{Class 4}) + 5 \times TP(\text{Class 3}) + 100 \times TP(\text{Class 2}) + \\ & 50 \times TP(\text{Class 1}) + 300 \times TP(\text{Class 0}) \end{aligned}$$

클래스 불균형으로 0의 수가 적은 것을 확인!



Score function을 최대화할 수 있도록
Class 0을 정확히 맞추는 데 초점

1

EDA 및 전처리

데이터셋 구조

Topic

주어진 데이터를 활용하여 성능이 좋은 다중 분류 모델 만들기
변수명은 마스킹되어 있고, 자료형은 수치형과 범주형 혼합

Training_set

6165 * 35

Int, float, object로 구성

Test_set

1542 * 34

Int, float, object로 구성

데이터셋 구조

변수명과 값의 성격 등을 고려해 34개의 변수를
22개의 **수치형 변수**와 12개의 **범주형 변수**로 분류

수치형 변수

X1, X5, X7, X8, X9, X11,
X12, X16, X18 ~ X31

범주형 변수

X2, X3, X4, X6, X10, X13,
X14, X15, X17, X32, X33, X34

X34는 소수점 값을 반올림하여 범주형 변수로 분류!

데이터셋 구조

범주형 변수

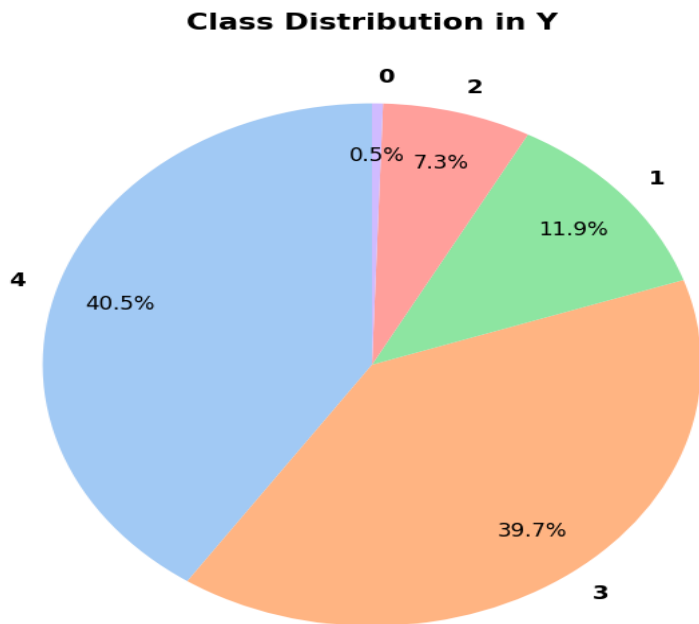
범주 간에 순서가 없는 **명목형 자료**와 순서가 있는 **순서형 자료**로 나뉨
자료형에 따라 적합한 통계 기법, 변수 처리 방법이 달라지고
잘못 처리할 경우 해석에 왜곡이 생길 수 있기 때문에 구분 필요

⋮

변수명과 고유값의 수 등을 통해 **명목형 변수**와 **순서형 변수**로 구분
주어진 정보만으로 **구분이 불가능한 경우** 따로 분류

Ex. X3 (disease)는 명목형, X6 (income)은 순서형, X33 (Surrogate)은 모호

클래스 불균형



클래스 0의 비율이 0.5%로

클래스 불균형이 존재



학습 과정에서 소수 클래스에 대한 학습이
잘 이루어지지 않을 가능성이 있음

클래스 불균형

자세한 내용은 범주형자료분석팀 3주차 클린업 참고!

클래스 불균형을 조정해야 하는 이유

불균형 데이터로 모델을 학습시켜 예측할 경우
모델의 성능을 정확히 파악하기 어렵기 때문



Y=1인 관측치가 95개, Y=0인 관측치가 5개인 데이터가 있을 때

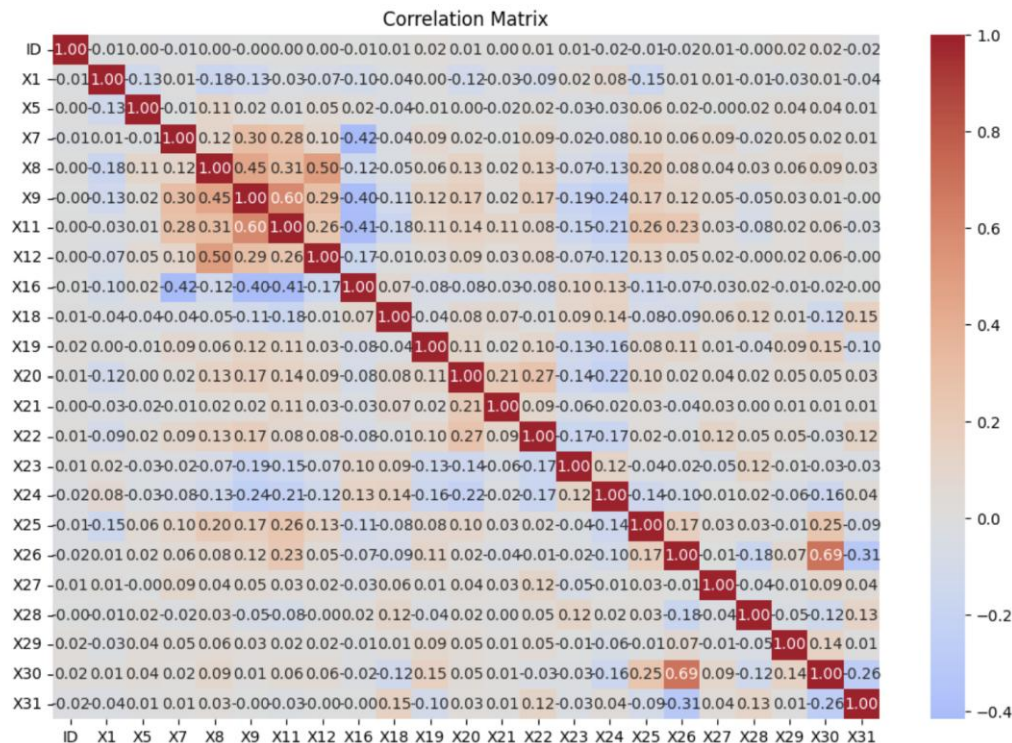
모델이 모두 $\hat{Y} = 1$ 로 예측할 경우

정확도는 95%지만 Y=0을 전혀 예측하지 못했으므로

성능이 우수하다고 말하기는 어려움

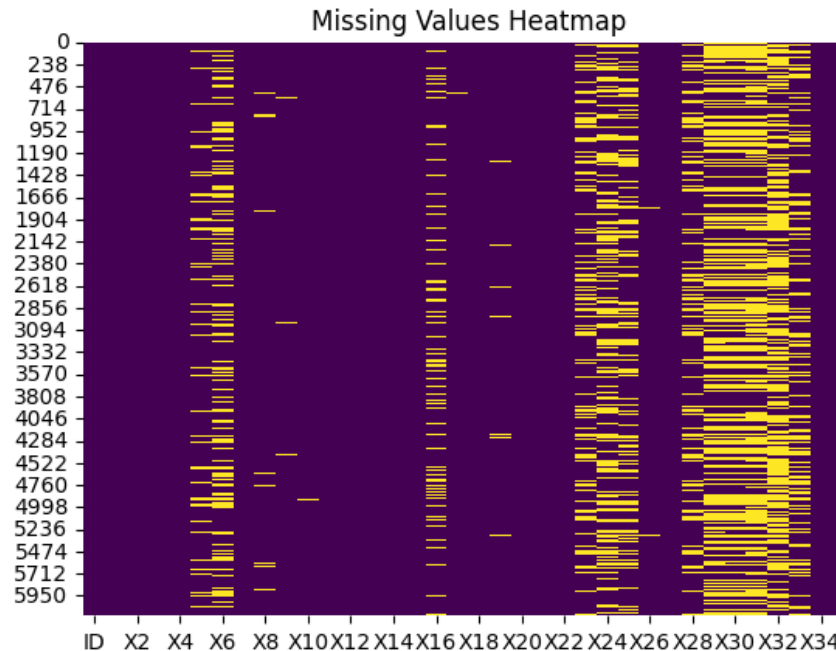
➡ **샘플링을 통한 클래스 불균형 해소가 필요!**

변수간 상관계수 시각화



가장 높은 상관계수 값이 0.69로
변수간 상관계수가 크지 않음을 알 수 있음

열별 결측치 확인



결측치의 비율이 20% 이상인 변수: X6, X23, X24, X25, X28~X33

Income(소득) 변수와 Measurement(ex. 생체지표) / Survey(설문조사)

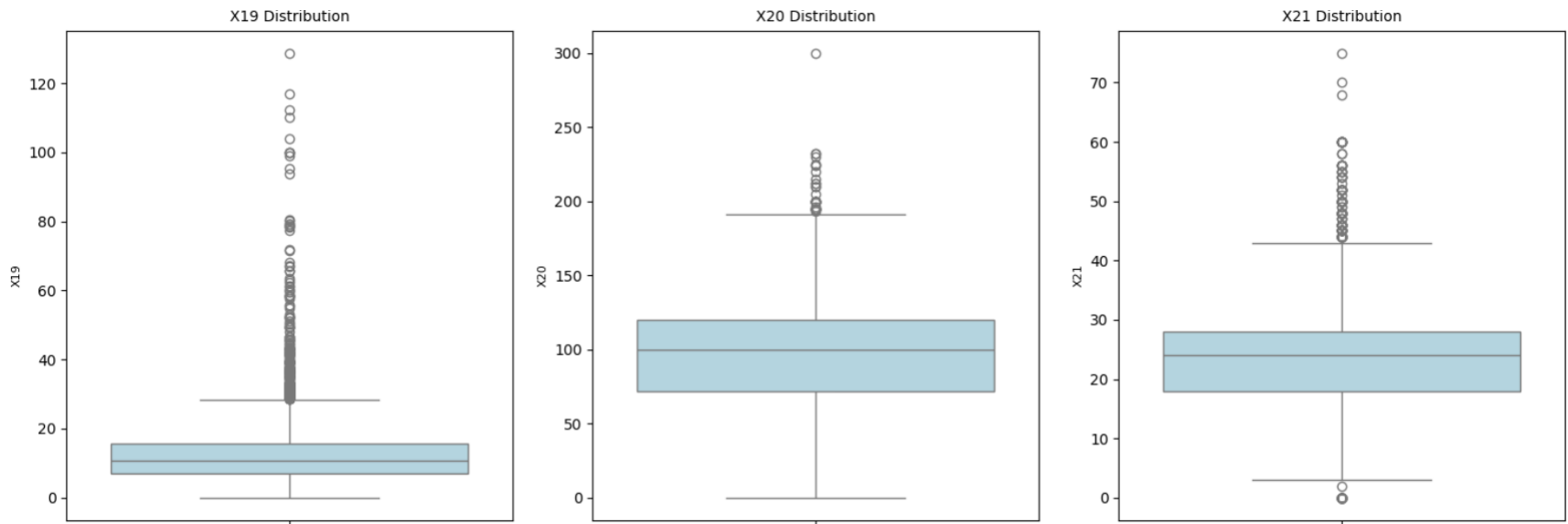
카테고리에서 결측치의 비율이 높은 것을 알 수 있음

열별 결측치 확인

Column	Missing Count	blue
X32	3750	60.84
X31	3275	53.13
X29	2993	48.56
X30	2914	48.27
X24	2226	
X6	1762	
X25	1721	
X33	1602	
X23	1528	
X28	1498	
...

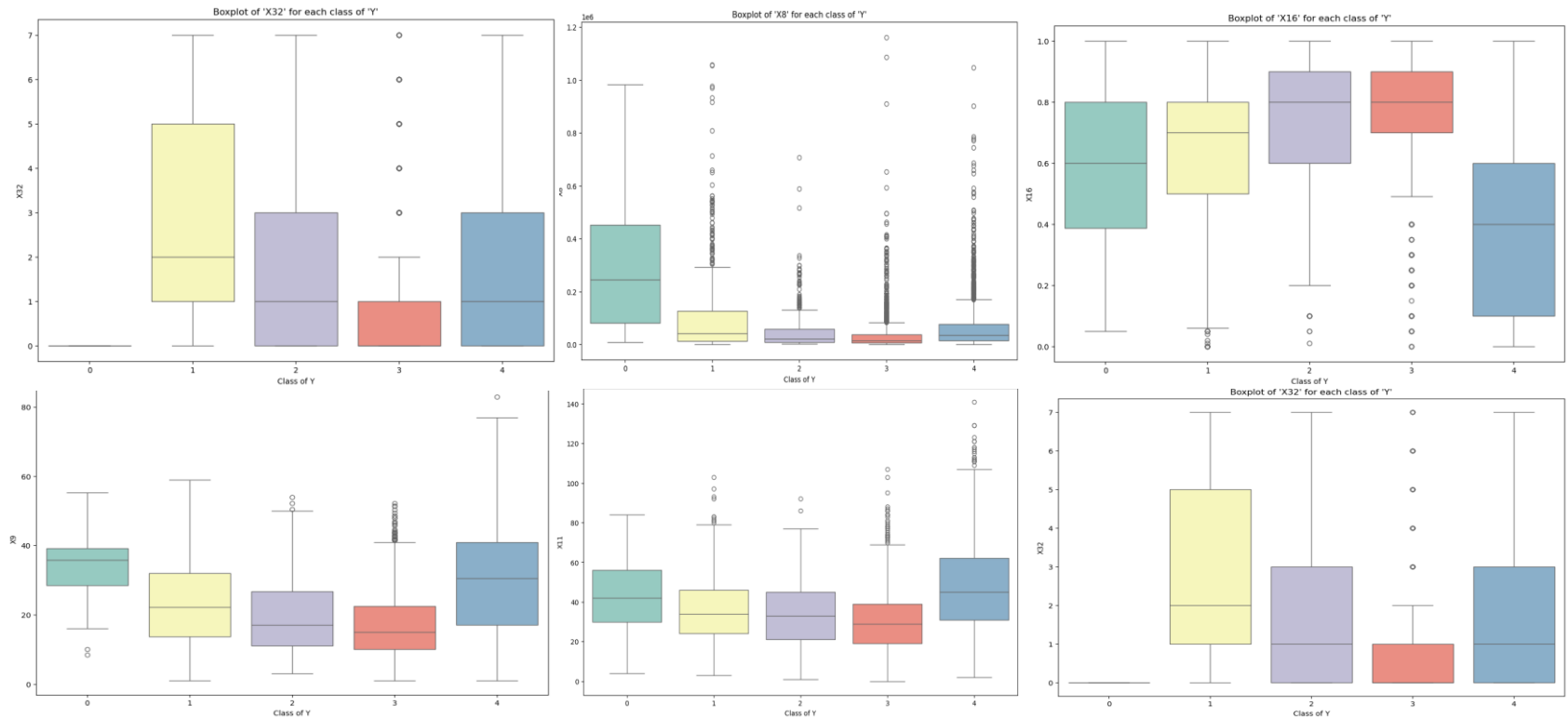
결측치 개수 상위 10개 변수

이상치 확인



✓ 박스 플롯을 통해 이상치가 존재하는 변수가 있음을 확인

이상치 확인



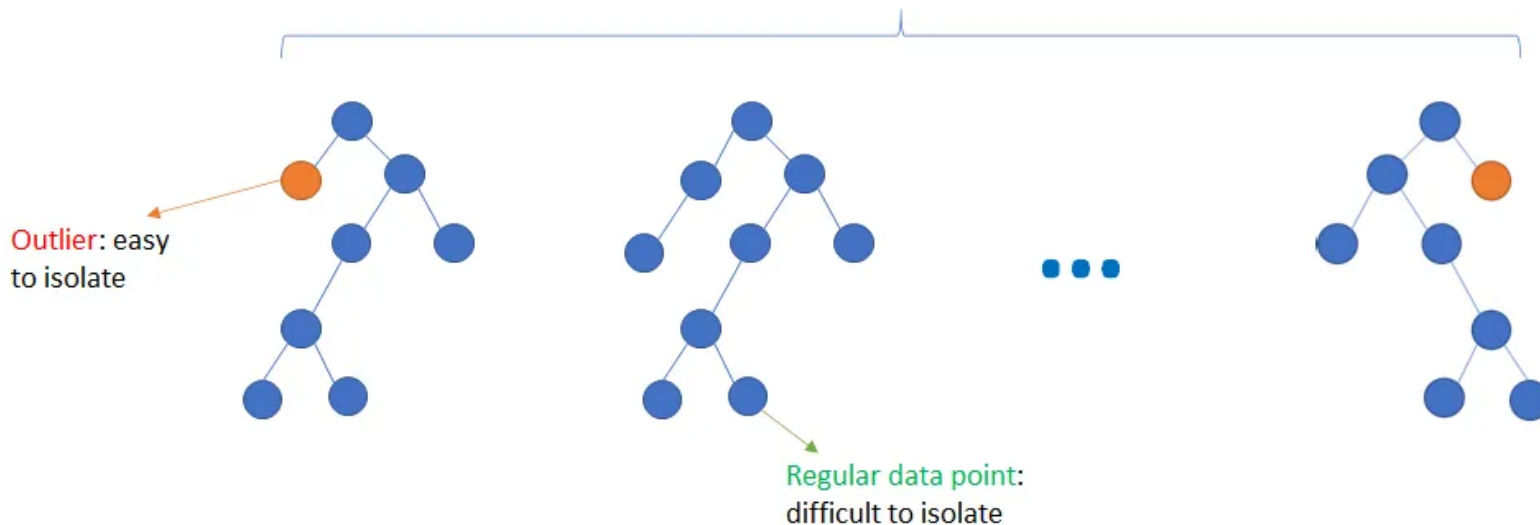
✓ 각 변수에 대해 'Y의 클래스별 평균 값 및 IQR 확인

이상치 처리

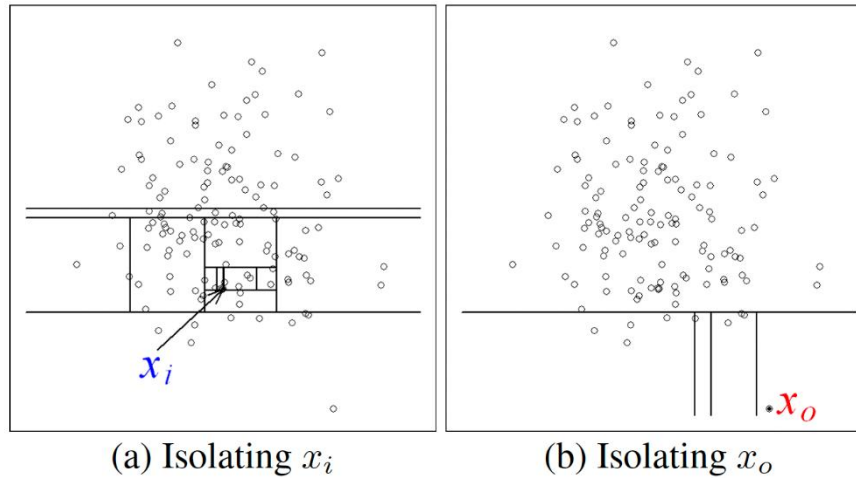
Isolation Forest

데이터 포인트를 고립시키는 과정을 통해 **이상치를 탐지하는 앙상블 기반 모델**

Isolation Forest



이상치 처리



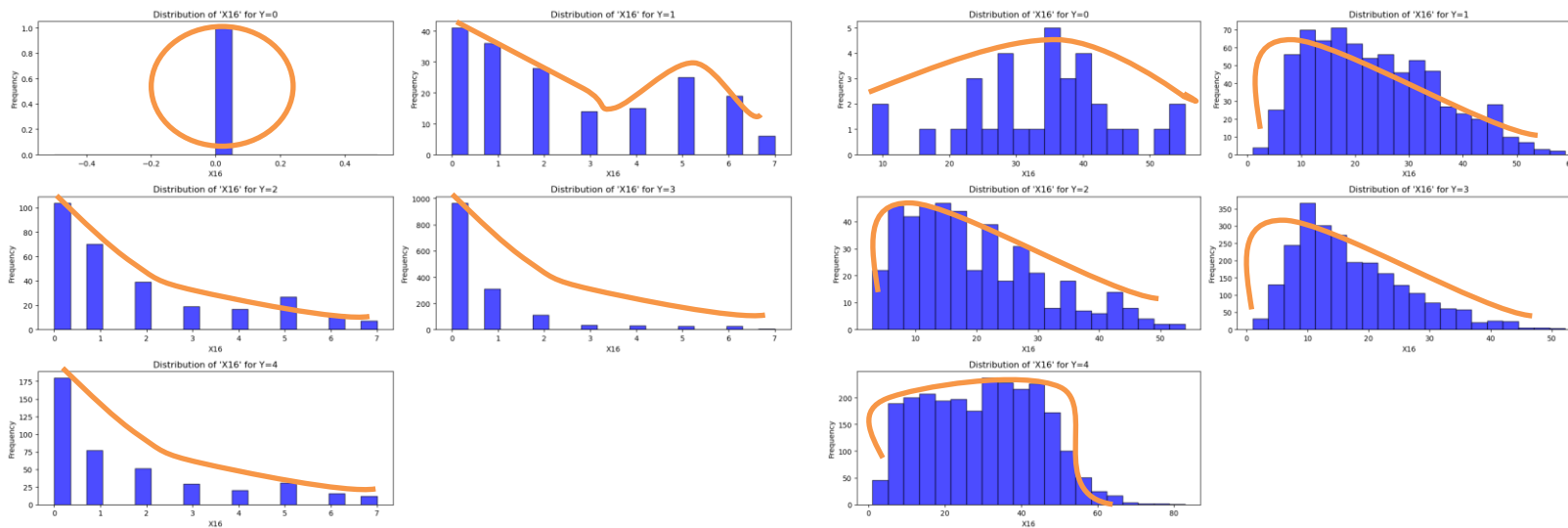
데이터 분포 가정이나 거리 계산 없이
이상치가 정상치보다 더 빠르게 **분리(Isolation)**될 것이라고 가정

이상치 처리

n_estimators	생성할 isolation tree의 개수
max_samples	학습할 샘플의 최대 개수
contamination	데이터셋에서 이상치로 간주할 비율
max_features	학습할 최대 특성의 개수

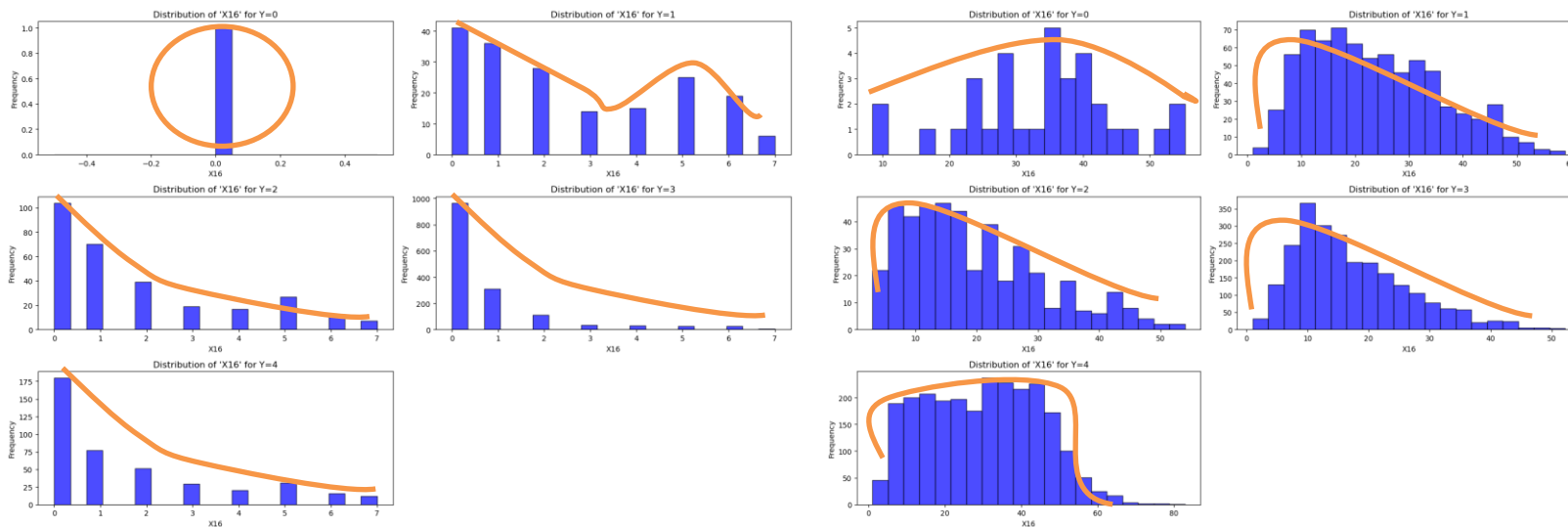
contamination 값을 0.03~0.1로 조정하며 최적의 파라미터 선택

분포 확인



✓ 타겟 변수인 'Y'의 클래스에 따른 변수별 분포에 대한 EDA 진행

분포 확인



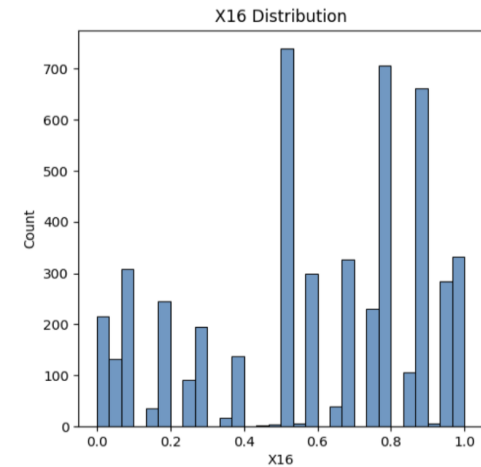
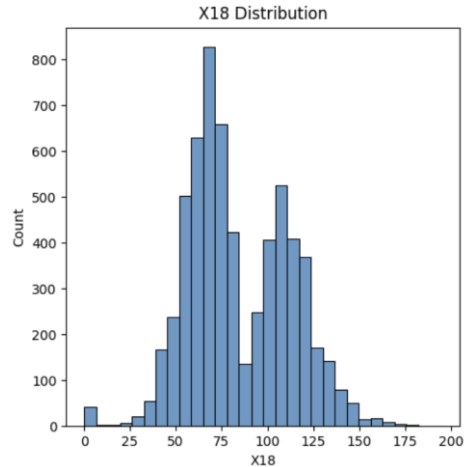
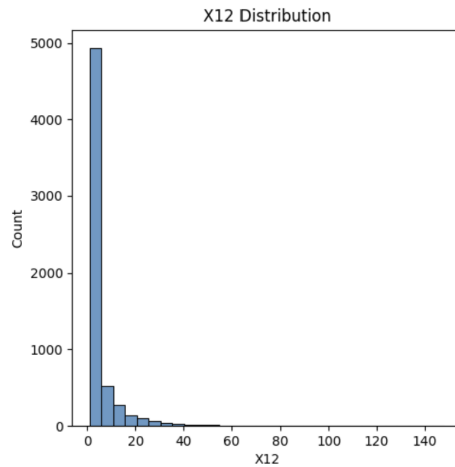
✓ 타겟

'X5', 'X8', 'X9', 'X32' 변수의 특정 클래스 분포가

진행

다른 클래스와 확연히 구분됨을 확인

분포 확인



양으로 치우친 분포, 다봉 분포, 넓게 펼쳐진 분포 등
각 변수 별로 분포가 다르게 나타남

변환/스케일링 | 변환

변환 *Transformation*

비대칭 분포(skewed distribution)를 보이는 데이터를
정규 분포에 가깝게 변환하는 데이터 전처리 과정

Log

✓ Positive Skewed

Box-Cox

✓ Slightly Positive Skewed

✓ Bimodal

Yeo-Johnson

✓ High Variance

분포에 따라 위 3가지 변환 방식을 사용

변환/스케일링 | 변환

변환 *Transformation*

비대칭 분포(skewed distribution)를 보이는 데이터를
정규 분포에 가깝게 변환하는 데이터 전처리 과정

Log

✓ Positive Skewed

Box-Cox

✓ Slightly Positive Skewed
✓ Bimodal

Yeo-Johnson

✓ High Variance

분포에 따라 위 3가지 변환 방식을 고려

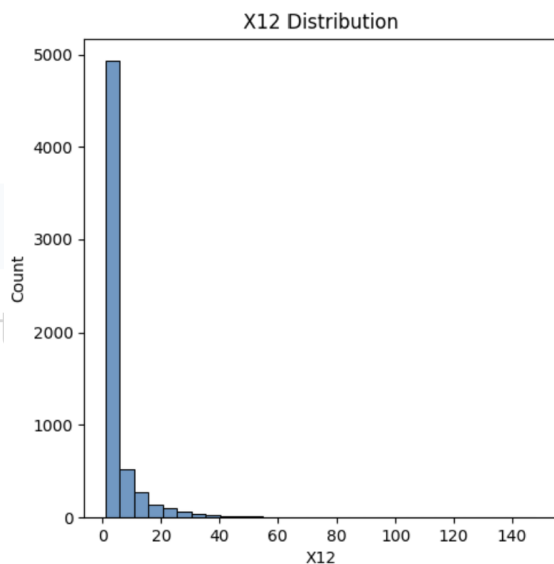
변환/스케일링 | 변환

변환 Transformation

왜도값 3을 기준으로 분류

비대칭 분포(skewed distribution)를 보이는 데이터를

정규 분포에 가깝게 변환하는 데이터 전처리 과정

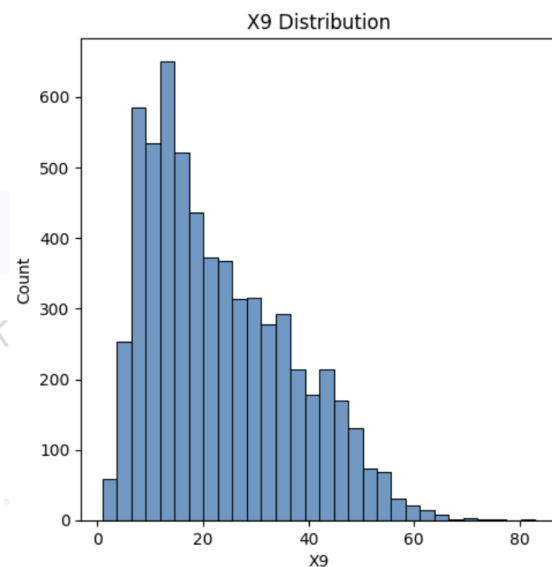


Positive Skewed

Box-Cox

Slightly Positive Sk

✓ Bimodal



Slightly Positive Skewed

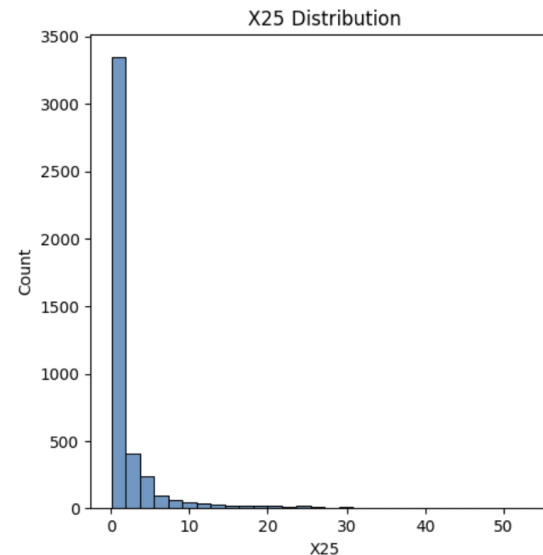
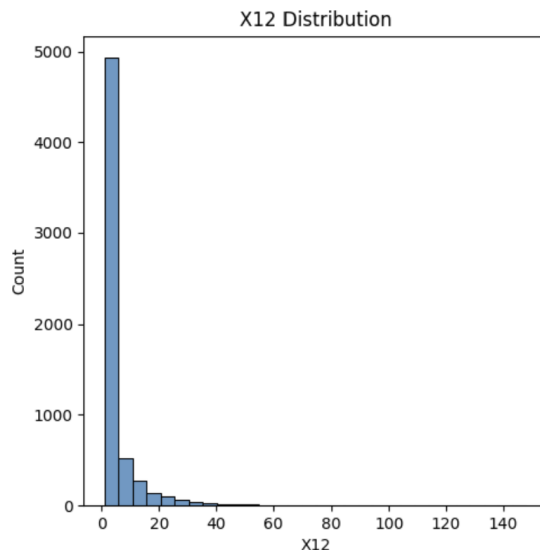
위 3가지 변환 방

변환/스케일링 | 변환

Log Transformation

왼쪽 극단으로 치우친 경우에 사용

데이터의 각 값에 **로그 함수를 적용**하여 변환



Positive Skewed Distribution

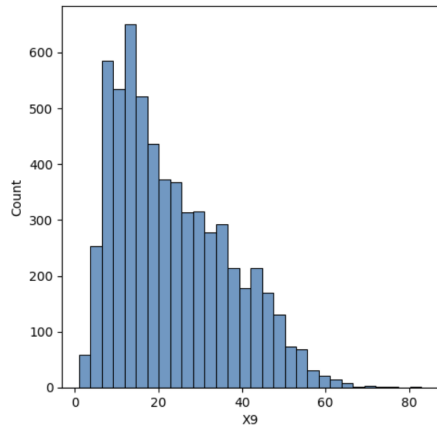
변환/스케일링 | 변환

Box-Cox Transformation

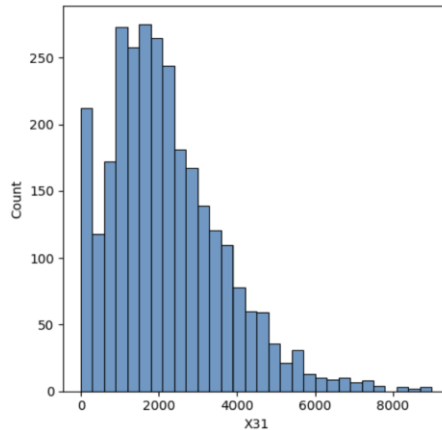
왼쪽으로 살짝 치우친 경우 혹은 다봉 분포의 경우에 사용

데이터에 **특정 매개변수(λ)**를 적용하여 변환

X9 Distribution

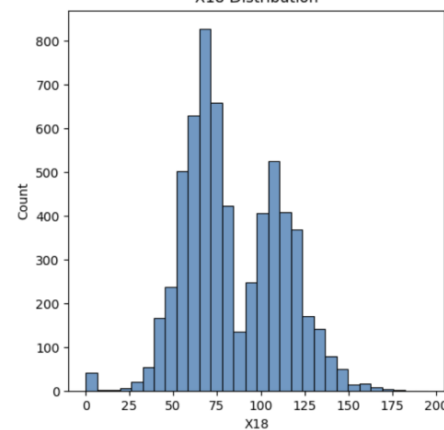


X31 Distribution

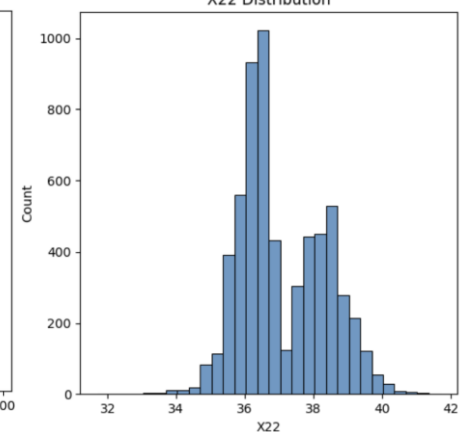


Slightly Positive Skewed

X18 Distribution



X22 Distribution



Bimodal

변환/스케일링 | 변환

Box-Cox Transformation

왼쪽으로 살짝 치우친 경우 혹은 다봉 분포의 경우에 사용

데이터에 **특정 매개변수(λ)**를 적용하여 변환

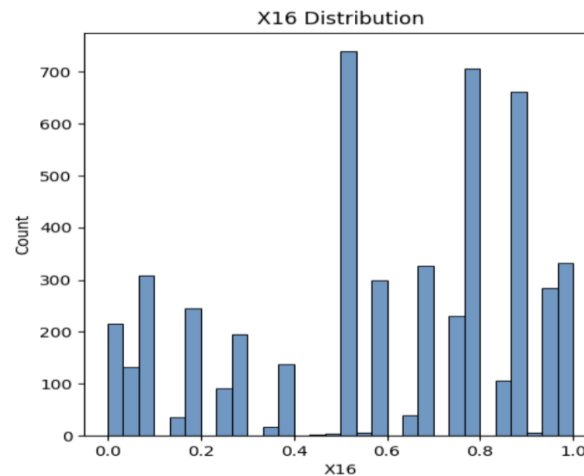


변환/스케일링 | 변환

Yeo-Johnson Transformation

분포가 넓게 퍼져 있는 경우에 사용

Box-Cox 변환의 확장판으로, 0과 음수에도 적용 가능



High Variance Distribution

변환/스케일링 | 스케일링

스케일링

특징(Feature)들의 값 범위를 조정하여 데이터의 크기 차이를 줄이는 과정

RobustScaler

중앙값과 IQR을 사용하여
특성들을 같은 스케일로 변환

MinMaxScaler

매우 다른 스케일의 범위를
0과 1사이로 변환

StandardScaler

각 특성의 평균을 0,
분산을 1로 변환

열 결측치 처리 | 수치형 변수

수치형 변수

복잡성과 데이터의 구조 및 변수 간 관계를
고려하는 정도에 따라 처리 방법 다양함

통계적인 방법

평균
최빈값
중앙값

알고리즘 기반

KNN Imputer
Iterative Imputer

모델 기반

랜덤 포레스트

열 결측치 처리 | 수치형 변수

KNN Imputer

KNN 알고리즘을 사용하여 결측치를 대체하는 방법

변수 간 상관관계나 데이터 분포를 고려하므로
단순 대체 방법보다 더 정교한 결측치 보간이 가능!



NA값의 **가장 가까운 주변 k개의 평균**을

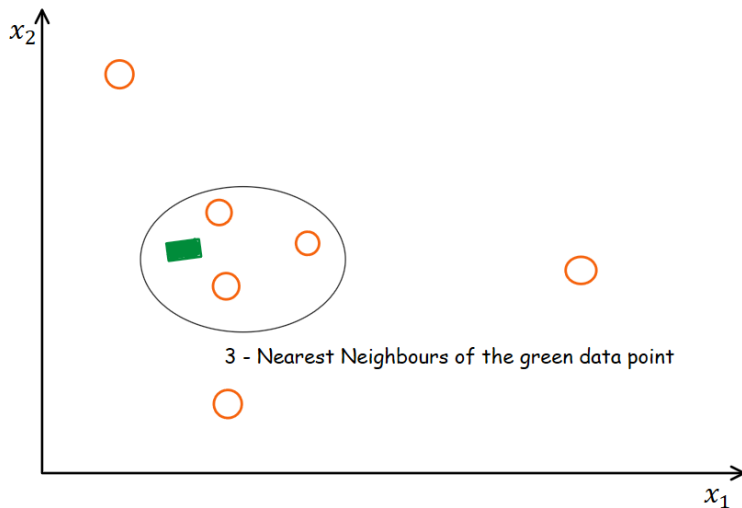
NA값으로 대체하는 알고리즘

열 결측치 처리 | 수치형 변수

KNN Imputer

KNN 알고리즘을 사용하여 결측치를 대체하는 방법

변수 간 상관관계나 데이터 분포를 고려하므로
단순 대체 방법보다 더 정교한 결측치 보간이 가능!



NA값의 **가장 가까운 주변 k개의 평균**을
NA값으로 대체하는 알고리즘

열 결측치 처리 | 수치형 변수

Iterative Imputer

결측치를 **반복적으로 예측**해 채우는 방법
변수 간의 관계를 최대한 활용해 결측치를 예측!



초기값으로 결측치를 채운 후,
각 변수를 타겟 변수로 설정하고,
나머지 변수를 독립 변수로 사용하여
회귀 모델 학습을 반복



열 결측치 처리 | 수치형 변수

Iterative Imputer

결측치를 **반복적으로 예측**해 채우는 방법
변수 간의 관계를 최대한 활용해 결측치를 예측!



초기값으로 결측치를 채운 후,
각 변수를 타겟 변수로 설정하고,
나머지 변수를 독립 변수로 사용하여
회귀 모델 학습을 반복



열 결측치 처리 | 수치형 변수

RF 기반 예측

랜덤 포레스트를 사용하여 결측값을 타겟 변수로 설정하고,
나머지 변수들을 독립 변수로 사용하여 모델을 학습하는 방식



수치형 변수 중 결측치가 없는 변수들 학습



결측값이 있는 **행에서 독립 변수만 추출**해
학습된 모델을 기반으로 결측값 예측

열 결측치 처리 | 수치형 변수

수치형 변수

평균, 최빈값, 중앙값 등으로 대체
혹은 알고리즘, 모델을 기반으로 대체



다수의 결측치가 존재하고 있기 때문에,
정밀한 보간이 가능하고 성능이 가장 좋았던

KNN Imputer 선택!



열 결측치 처리 | 수치형 변수

수치형 변수

평균, 최빈값, 중앙값 등으로 대체

혹은 알고리즘, 모델을 기반으로 대체

뒤에 나올 모델 선택까지 종합적으로 고려하여,

변환보다는 모든 수치형 변수를 같은 스케일로 맞춰주는 것이 더 필요하다고 판단

스케일러를 KNN Imputer 보간 방식에 적합한 **MinMaxScaler**로 결정!



다수의 결측치가 존재하고 있기 때문에,

정밀한 보간이 가능하고 성능이 가장 좋았던

KNN Imputer 선택!

(박 력)



열 결측치 처리 | 범주형 변수

범주형 변수

각 변수의 결측 비율에 따라 서로 다른 결측치 처리 방법을 사용

결측치가 매우 적은 변수

최빈값으로 대체
혹은 행 삭제

결측치가 50% 미만인 변수

'Unknown' 처리 후
라벨 인코딩 진행

결측치가 50%이상인 변수

열 삭제

열 결측치 처리 | 범주형 변수

범주형 변수

각 변수의 결측 비율에 따라 서로 다른 결측치 처리 방법을 사용

결측치가 매우 적은 변수

최빈값으로 대체

혹은 행 삭제

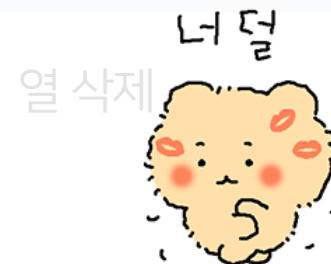
결측치가 1% 미만인 경우 최빈값,

그 외에는 unknown으로 지정하기로 결정!

UNKNOWN 처리 후

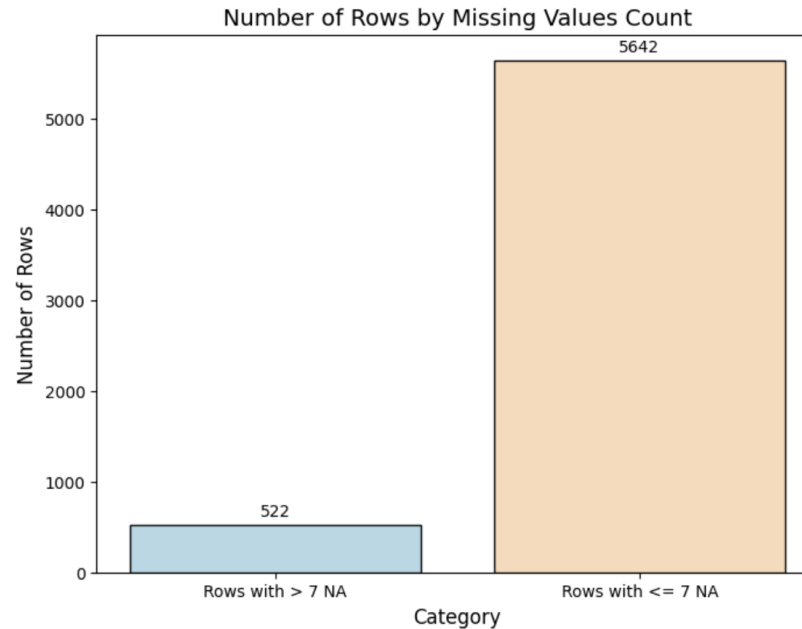
라벨 인코딩 진행

결측치가 50%이상인 변수



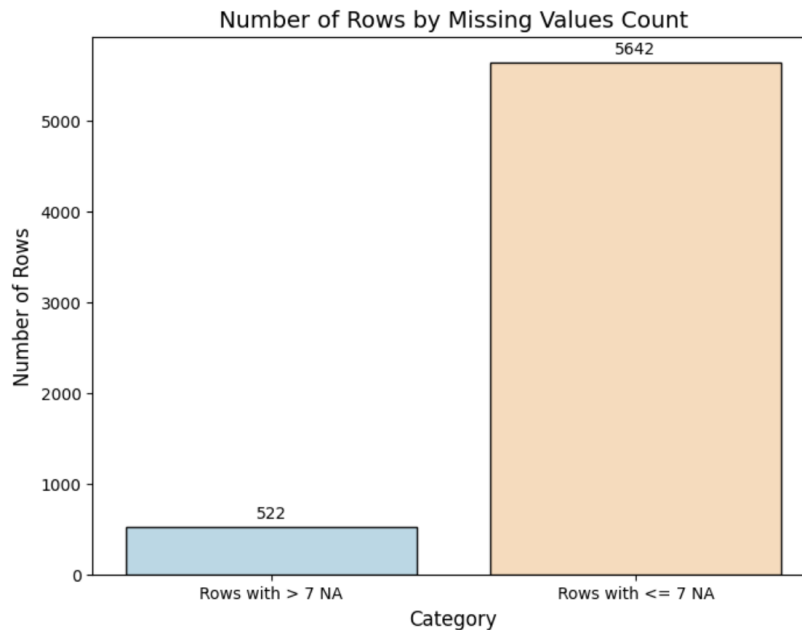
X32 -> 결측치 비율이 61%로 너무 커 열을 삭제하려 했으나 제거한 뒤 성능이 많이 떨어져 보간하는 것으로 결정...

행 결측치 처리



결측치 20%(7개) 초과인 관측치 522개의 행은
정보량이 부족하다고 판단하여 삭제를 고려

행 결측치 처리

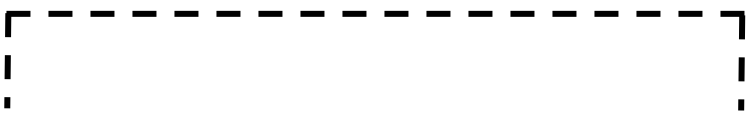


그러나 행 삭제를 한 뒤 성능의 변화가 미미하거나 떨어짐
행 삭제로 인한 **정보 손실**의 영향이 더 컸을 것이라고 추론해볼 수 있음

인코딩

인코딩 (Encoding)

범주형 데이터(문자열 또는 클래스 값)를
머신러닝 모델에 적용할 수 있는 **숫자 형태로 변환**하는 과정



라벨 인코딩

결정 트리, 랜덤 포레스트,
XGBoost 등에서 많이 사용

원핫 인코딩

선형 회귀, 로지스틱 회귀,
SVM, 신경망 등에서 많이 사용

인코딩

Label Encoding

각 범주를 **고유한 숫자**로 변환!
보통 **순서형** 자료 처리에 사용

One-Hot Encoding

각 범주를 **이진 값**(0 또는 1)으로
변환하여 **벡터** 형태로 표현!
보통 **명목형** 자료 처리에 사용

color
red
green
blue
red



color
red
green
blue
red

인코딩

Label Encoding

각 범주를 **고유한 숫자**로 변환!
보통 **순서형** 자료 처리에 사용

color
red
green
blue
red



One-Hot Encoding

각 범주를 **이진 값**(0 또는 1)으로
변환하여 **벡터** 형태로 표현!
보통 **명목형** 자료 처리에 사용

red	green	blue
1	0	0
0	1	0
0	0	1
1	0	0



인코딩

우리 팀의 선택 기준



명목형 자료에는 원핫인코딩, 나머지는 라벨인코딩을 진행했지만
성능이 좋지 못함

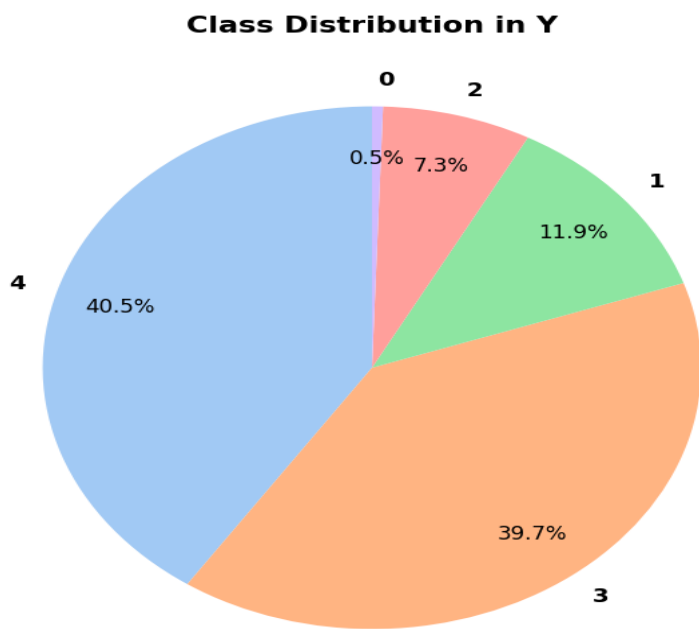


트리 기반 모델 사용 시 변수의 숫자 크기 또는 순서에 의존하지 않는
라벨 인코딩이 더 적합하다고 판단

color	red	green	blue
red	1	0	0
green	0	1	0
blue	0	0	1
red	0	0	0

전체 라벨 인코딩을 선택하기로 결정!

클래스 불균형 해결



- ✓ 학습 과정에서 소수 클래스에 대한 학습이 제대로 이루어지지 않을 수 있음
- ✓ 클래스 비율을 유지하여 분할 후
오버샘플링 및 언더샘플링을 통해
클래스 불균형 해결

'Y' 변수 값의 분포가 불균형을 확인

클래스 불균형 해결 | 오버 샘플링

오버샘플링 (OverSampling)

소수의 클래스를 다수의 클래스에 맞추어 **관측치를 증가**시키는 방법

SMOTE, ADASYN...

장점

정보의 손실이 발생하지 않아
일반적으로 언더 샘플링보다
성능이 좋음

단점

다수의 클래스에 맞도록
데이터가 커져 메모리 사용이나
처리 속도 측면에서 상대적으로 불리

클래스 불균형 해결 | 언더 샘플링

언더샘플링 (UnderSampling)

소수의 클래스는 변형하지 않고,
다수의 클래스를 소수의 클래스에 맞추어 **관측치를 감소**시키는 방법

ENN, Tomek Links ...

장점

데이터 사이즈가 줄어들어 메모리
사용이나 처리 속도에 있어 유리

단점

관측치 손실로
정보가 누락되는 문제 발생

클래스 불균형 해결 | 언더 샘플링

언더샘플링 (UnderSampling)

소수의 클래스는 변형하지 않고,

다수의 클래스를 소수의 클래스와 맞추어 **관측치를 감소**시키는 방법



다양한 기법 간의 조합을 찾아 가장 좋은 성능을 보이는 불균형 해결법 채택

장점

데이터 사이즈가 줄어들어 메모리
사용이나 처리 속도에 있어 유리

단점

관측치 손실로
정보가 누락되는 문제 발생

클래스 불균형 해결

결합 방법	Custom Score
Borderline-SMOTE	4550
Borderline-SMOTE + SVM	3065
SMOTE + ADAYSN	4250
SMOTENC	4505
SVM SMOTE	3580
Borderline-SMOTE + ENN	7995
Clustering + SMOTE + ENN	7960
SMOTE + ENN	8495

✓ 가장 성능이 좋은 **SMOTE + ENN** 조합을 사용하여 클래스 불균형 해결

클래스 불균형 해결 | SMOTE + ENN

SMOTE (Synthetic Minority Over-sampling Method)

소수 범주의 데이터를 가상으로 만들어내는 방법

ENN (Edited Nearest Neighbor)

다수 범주의 데이터를 줄이는 방법 (주로 다른 기법들과 결합하여 사용!)



SMOTEENN

SMOTE는 소수 클래스의 데이터를 증가시키고,
ENN은 다수 클래스의 데이터를 감소시킴으로써 **양방향 조정을 수행**




SMOTE + ENN

SMOTEENN을 사용하는 이유

SMOTE (Synthetic Minority Over-sampling Method)

소수 범주의 데이터를 가장으로 만들어내는 방법

ENN (Edited Nearest Neighbor)  SMOTE와 ENN을 결합함으로써다수 범주의 데이터를 줄이는 방법 (수도 다른 기법들과 결합하여 사용)
소수, 다수 클래스의 데이터 품질을 동시에 향상시키고,
노이즈 데이터를 제거하여 성능을 높임SMOTE  불균형 데이터에서 학습 알고리즘이

과대적합되는 문제를 완화하는데 유용

SMOTE는 소수 클래스의 데이터를 증가시키고, ENN은 다수 클래스의
데이터를 감소시킴으로써 양방향 조정을 수행

2

모델링

파생변수

데이터 정보가 부족하므로 다양한 방식으로 파생 변수 제작 진행

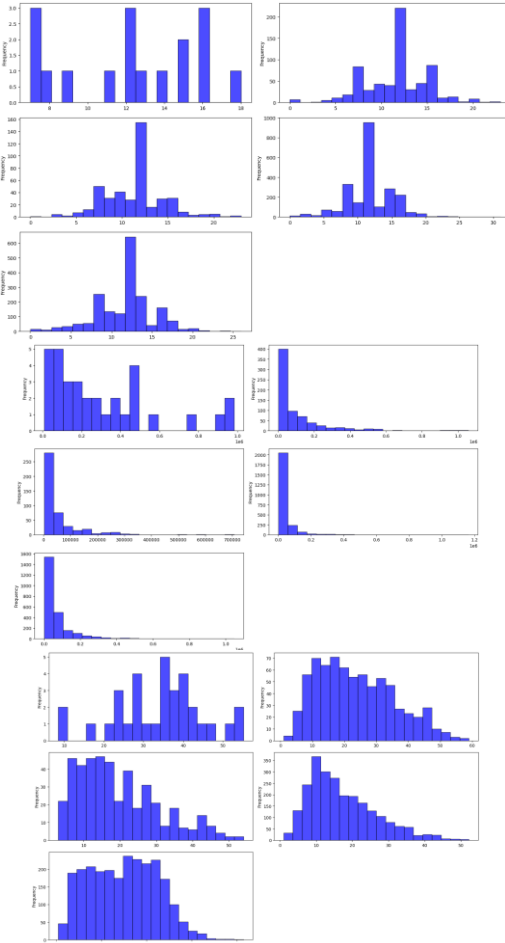
분포를 고려한 파생변수 제작

변수의 특성을 고려한 파생변수 제작



이후 PCA, Feature Selection 등에 활용 가능

파생변수



앞서 진행한 분포 EDA를 통해 'Y'의 클래스를
잘 구분해주는 변수들을 이용하여 파생변수 제작

$$\text{Total_charges} = \text{'X8'} + \text{'X9'}$$

$$\text{Edu_ratio} = \text{'X5'} / \text{'X5' max()}$$

파생변수

Hint로 제공된 변수에 대한 정보를
바탕으로 다양한 파생변수 제작

$$\text{사회경제적_요인} = X5 * X6 (\text{수입} * \text{교육})$$

$$\text{Disease_count} = X13 + X14 + X15$$

$$\text{환자상태 점수} = (X29 + X30 + X31) / X32$$

X12: 범주화

$$\text{의료진 문제 보고 비율} = X16 / X12$$

파생변수 | PCA

PCA (Principal Component Analysis)

고차원 데이터를 저차원으로 **축소**하면서도
데이터의 **정보를 최대한 보존**하려는 통계적 기법



PCA를 통해 주성분들을 구해
모델 적합에 사용하기 위한 **파생 변수**를 선택해보자!

파생변수 | PCA

PCA (Principal Component Analysis)

고차원 데이터를 저차원으로 **축소**하면서도
데이터의 **정보를 최대한 보존**하려는 통계적 기법

PCA는 변수간 다중공선성이
존재하는 경우
주성분 요소들로 변수들을 요약



해 주성분들을

하지만
유의미한 결과를 얻지 못함

모델 적합에 사용하기 위한 파생 변수를 선택해보자!

파생변수 | PCA

PCA (Principal Component Analysis)

고차원 데이터를 저차원으로 **축소**하면서도
데이터의 **정보를 최대한 보존**하려는 통계적 기법

PCA는 변수간 다중공선성이

PCA 대신 **Feature Selection**을 통해 변수를 선택해보자!

주성분 요소들로 변수들을 요약

모델 적합에 사용하기 위한 파생 변수를 선택해보자!



하지만
유의미한 결과를 얻지 못함

변수선택

변수 선택 (Feature Selection)

모델의 성능 향상과 효율성 개선을 위해
데이터의 변수 중 중요한 것만 선택하는 과정



시간 비용의 문제로 RFECV와 Wrapper의 방식 중
Backward Sequential Feature Selection을 통해 변수를 선택

변수선택

RFE (Recursive Feature Elimination)

모델의 성능에 **가장 중요한 변수를**
반복적으로 선택하여 최적의 변수 조합을 찾는 방법



하지만 우리가 선택한 전처리 방식에는
RFE가 효과적이지 못함

변수선택

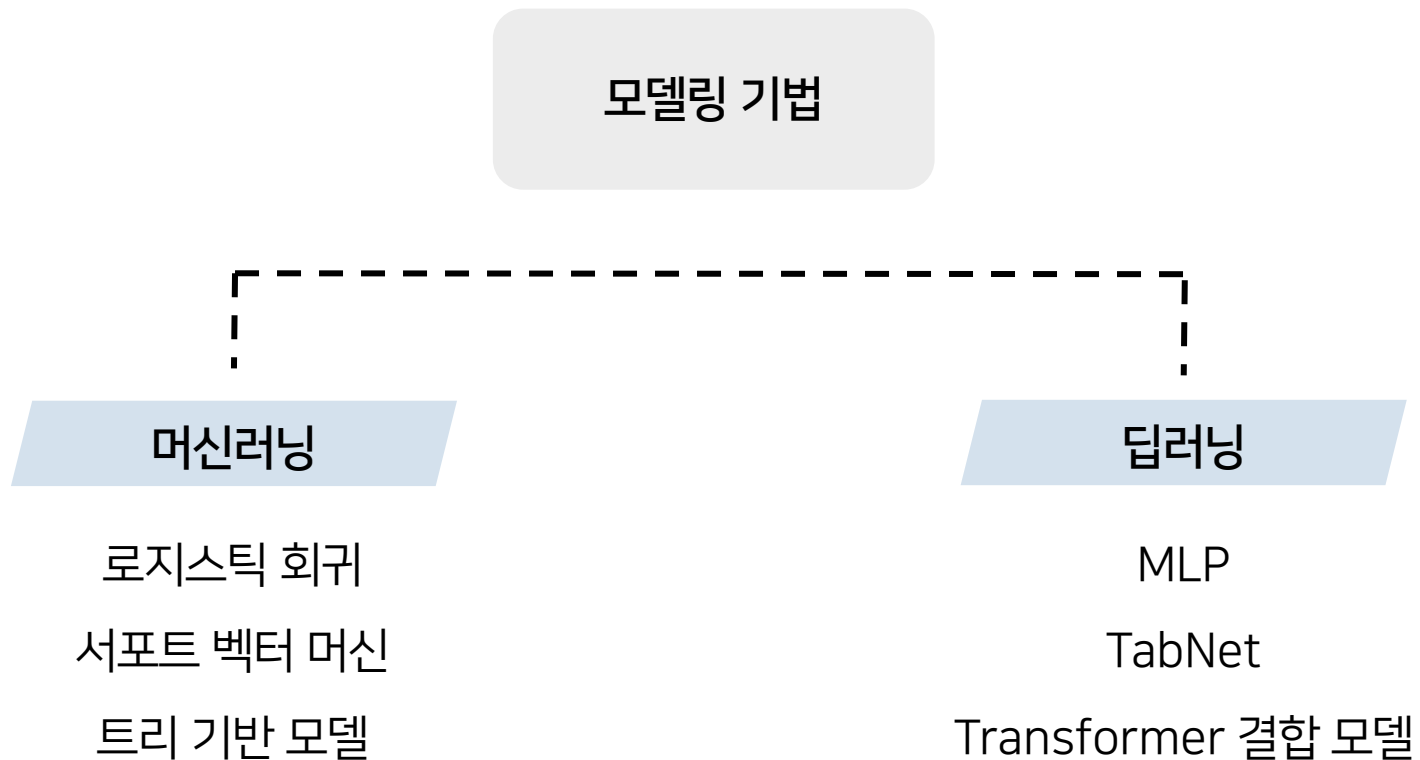
Backward Sequential Feature Selection

모든 변수 중 모델 성능에 **가장 낮은 기여**를 하는 변수를
반복적으로 제거하여 최적의 변수 집합을 찾는 방법



시간 비용의 문제로 최종 모델에 적용하지 못함...

모델 선택 | 머신러닝 vs 딥러닝





모델 선택 | 머신러닝 vs 딥러닝

딥러닝의 Tabular Data 적용

모델링

Tabular Data의 경우 **irregular function**을 학습하는 것이 중요하지만
딥러닝은 복잡하고 smooth한 함수에 쉽게 수렴하기 때문에
Tabular Data에 대해서는 좋은 성능을 보이지 못하고 있음

머신러닝

딥러닝

로지스틱 회귀

서포트 벡터 머신

트리 기반 모델



MLP

TabNet

머신러닝을 선택하기로 결정!

모델 선택 | Lazy Classifier

Lazy Classifier

여러 분류 모델을 자동으로 테스트하여 가장 적합한 모델을
빠르게 선택하고 성능을 비교하는 Python 라이브러리

선형 모델

로지스틱 회귀
Linear SVC

트리 기반 모델

결정 트리
랜덤 포레스트

기타 모델

KNN
SVM

모델 선택 | Lazy Classifier

Model	Score Function	Accuracy
CalibratedClassifierCV	7365	0.31
RidgeClassifierCV	7200	0.18
RidgeClassifier	7200	0.18
RandomForestClassifier	7195	0.39
LogisticRegression	7190	0.36
LinearSVC	7175	0.30
ExtraTreesClassifier	7090	0.41



Score Function 7000점 이상의 모델들을 조합하여 앙상블 성능 평가

모델 선택 | Lazy Classifier

Model	Score Function	Accuracy
CalibratedClassifierCV	7365	0.31
RidgeClassifierCV	7200	0.18
RidgeClassifier	7200	0.18



6000점 대에는 다수의 모델이 존재하였고
과적합 및 모델 간의 상관관계로 인한 앙상블의 역효과를 방지하기 위해
Score Function 7000점을 기준으로 앙상블 실험 제한!



Score Function 7000점 이상의 모델들을 조합하여 앙상블 성능 평가

모델 선택 | 앙상블

앙상블 기법

여러 개의 개별 모델을 결합하여

정확하고 안정적인 하나의 강력한 모델을 구성하는 기법

Hard Voting

가장 많이 나온 클래스를
최종 예측값으로 선택

Soft Voting

각 모델의 예측 확률 평균값이
가장 높은 클래스를
최종 예측값으로 선택

모델 선택 | 앙상블

앙상블 기법

여러 개의 개별 모델을 결합하여

정확하고 안정적인 하나의 강력한 모델을 구성하는 기법



hard voting과 soft voting 모두 적용할 수 있도록
상위 7개의 모델에 대하여 홀수 개의 모델 조합으로 실험!

최종 예측값으로 선택

가장 높은 클래스를

최종 예측값으로 선택

모델 선택 | 앙상블

Ensemble Models	Score Function (hard voting)
CalibratedClassifierCV RidgeClassifierCV RidgeClassifier LogisticRegression LinearSVC	7420
CalibratedClassifierCV RandomForestClassifier ExtraTreesClassifier	7415

soft voting이 불가능한 모델들이 포함되어 있으므로
soft voting이 가능한 2위 모델들을 활용하여 앙상블 성능 평가

모델 선택 | 앙상블

Ensemble Models	Score Function (hard voting)
CalibratedClassifierCV RidgeClassifierCV RidgeClassifier LogisticRegression LinearSVC	7420
CalibratedClassifierCV RandomForestClassifier ExtraTreesClassifier	7415



soft voting 수행 결과,
7780으로 향상 및 Kaggle 점수 5150 달성!

모델 선택 | 앙상블 최적 모델

CalibratedClassifierCV

로지스틱 회귀 등을 활용하여

분류기의 예측 확률을 보정하는 모델

RandomForestClassifier

배깅 기법을 사용하여 여러 개의 결정 트리를 학습시킨 후

예측을 합산하여 과적합을 방지하는 앙상블 모델

ExtraTreesClassifier

결정 트리 무작위 분할, 특성 랜덤 선택을 도입하여

학습 속도가 빠른 앙상블 모델

모델 선택 | 앙상블 최적 모델

CalibratedClassifierCV

로지스틱 회귀 등을 활용하여

분류기의 예측 확률을 보정하는 모델



R

Optuna를 활용하여

각 모델의 하이퍼파라미터 최적화를 수행

ExtraTreesClassifier

결정 트리 무작위 분할, 특성 랜덤 선택을 도입하여

학습 속도가 빠른 앙상블 모델

모델 선택 | 하이퍼파라미터 튜닝

Optuna

효율적인 탐색 알고리즘을 사용하는
하이퍼파라미터 최적화를 위한 자동화된 라이브러리

Model	Score Function
RandomForestClassifier	8645
ExtraTreesClassifier	8330
Top 2 hard voting	8350
Top 2 soft voting	7800

모델 선택 | 하이퍼파라미터 튜닝

Optuna

효율적인 탐색 알고리즘을 사용하는
하이퍼파라미터 최적화를 위한 자동화된 라이브러리

Model	Score Function
RandomForestClassifier	8645
ExtraTreesClassifier	8330
Top 2 hard voting	8350
Top 2 soft voting	7800

모델 선택 | 하이퍼파라미터 튜닝

Optuna

효율적인 탐색 알고리즘을 사용하는
하이퍼파라미터 최적화를 위한 자동화된 라이브러리

	Model	Score Function
Rand	하이퍼파라미터 튜닝 이후 대체로 성능 향상을 보였으나 Kaggle에서 성능 향상을 보이지 못함..	
Ext		
		하이잉
		/800
	soft voting	



3

결과 및 해석

결론 및 인사이트 | 왜 3개 모델을 조합했을 때 성능이 좋았을까?

트리 기반 모델이 많이 사용된 이유

트리 기반 모델은 이질적인 데이터 (범주형 + 수치형)에도 강건하게 작동
특히, 엑스트라 트리는 각 분할에서 무작위성을 극대화하여
모델이 데이터에 과도하게 적합하는 것을 방지

결론 및 인사이트 | 왜 3개 모델을 조합했을 때 성능이 좋았을까?

트리 기반 모델이 많이 사용된 이유

Why do tree-based models still outperform deep learning on typical tabular data? 논문에 따르면,

- ✓ 범주형/수치형 변수들은 신경망을 약화시키기 때문에 딥러닝에서 취약한 반면 트리 기반 모델들은 split을 통해 데이터를 효율적으로 처리
- ✓ 트리 기반 모델들은 딥러닝 모델들에 비해 과하게 smoothing되지 않기 때문에 특성이 다양한 데이터에 더 적합하여 성능이 우수

결론 및 인사이트 | 왜 3개 모델을 조합했을 때 성능이 좋았을까?

트리 기반 모델이 많이 사용된 이유

Calibrated Classifier 와의 결합

- ✓ Calibrated Classifier는 예측 확률 값을 보정하여
예측 확률이 실제 관측 확률과 더 잘 일치하도록 해줌
- ✓ 소수 클래스에 대한 확률 값을 보다 정확하게 보정하여
소수 클래스의 성능을 개선 가능

결론 및 인사이트 | 왜 3개 모델을 조합했을 때 성능이 좋았을까?

트리 기반 모델은 예측 확률이 편향되거나 보정되지 않을 수 있는데
CalibratedClassifierCV가 이를 보정하여 soft voting 과정에서
보다 정확하고 균형 잡힌 확률값을 제공



✓ 범주형/수치형 변수들은 신경망을 약화시키기 때문에 딥러닝에서

앙상블 기법을 통해 개별 모델의 약점을 보완하고 데이터의 다양한 특성을 반영하여
좋은 결과가 나왔을 것이라고 추론해볼 수 있음

않기 때문에 특성이 다양한 데이터에 더 적합하여 성능이 우수

결론 및 인사이트 | 의료 데이터와 앙상블 모델 간 효과

의료 데이터와 앙상블 모델의 효과

헬스케어 데이터는 클래스 불균형과 특성 간 복잡한 관계로 인해
단일모델만으로는 데이터 패턴을 충분히 학습하기 어려움



앙상블 모델이 모델의 다양성을 보장!



의의 | 전처리

결측치 처리의 중요성

모델마다 결측치 처리 방법에 따라 성능이 달라지기 때문에
통계적 방법, 알고리즘/모델 기반 방법 등 다양한 실험이 필요하며
범주형 변수 결측치 처리에 대한 인사이트가 필요

전처리 순서의 중요성

전처리 적용 순서에 따라 성능이 크게 달라지기 때문에
논리적 접근과 다양한 실험을 통해
적절한 전처리 순서를 결정하는 것이 중요

의의 | 모델링



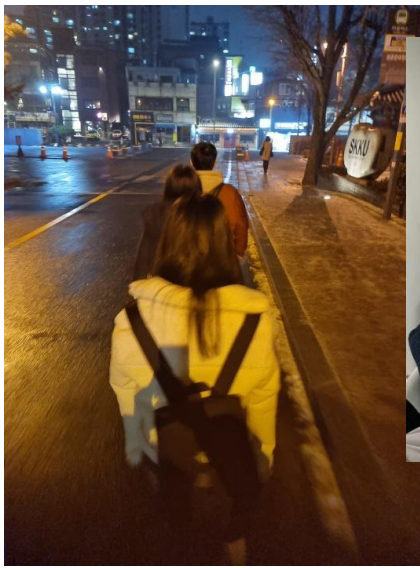
CalibratedClassifierCV와 같이 **예상치 못한 모델이**
앙상블을 통해 좋은 성능을 보일 수 있음



너무 많은 모델을 앙상블하면 오히려 역효과가 발생할 수 있으며
다양한 조합을 실험하며 **최적의 앙상블 모델**을 찾는 것이 중요



딥러닝이 모든 데이터에 대해 항상 우월한 것은 아니기 때문에
데이터셋에 맞는 모델을 사용하는 것이 중요

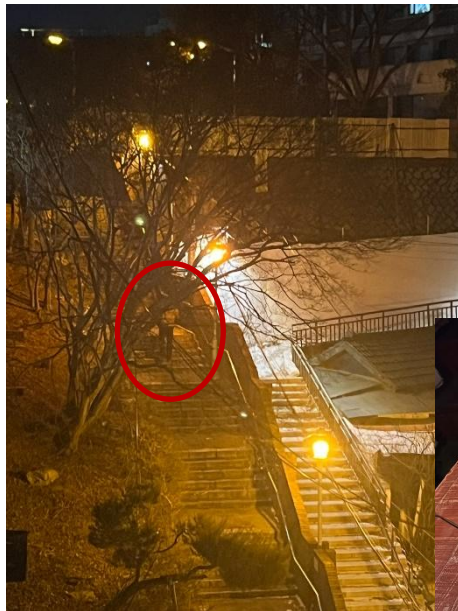


뭐가 다르게요~?



남남 맛있는 식사

미끄러워서 일렬로 내려가는 1팀..



혼자 걸어가겠다는 준영씨 보이지도 없어

소소한 회식...컴존 짱

서비스 주심!!!!!!!!!!!!!!

준영씨가 첫날 사준 일용할 양식ㅎㅎ -> 을 6일동안 먹음

감사합니다
