방중세미나

2팀

진수정
염예빈
황정현
한유진

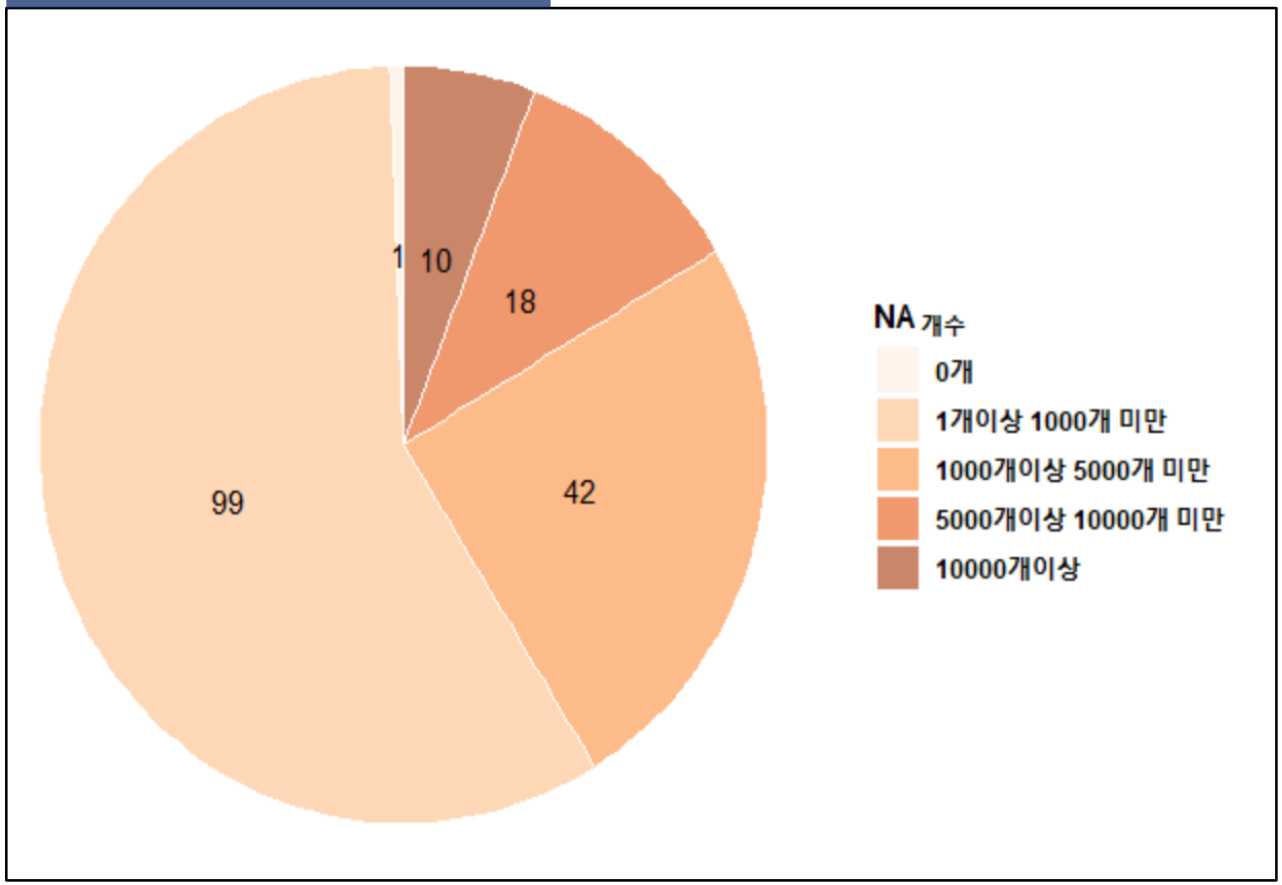
INDEX

1

EDA

EDA

col: NA 개수 시각화



각 변수들의
의미 파악 불가

+

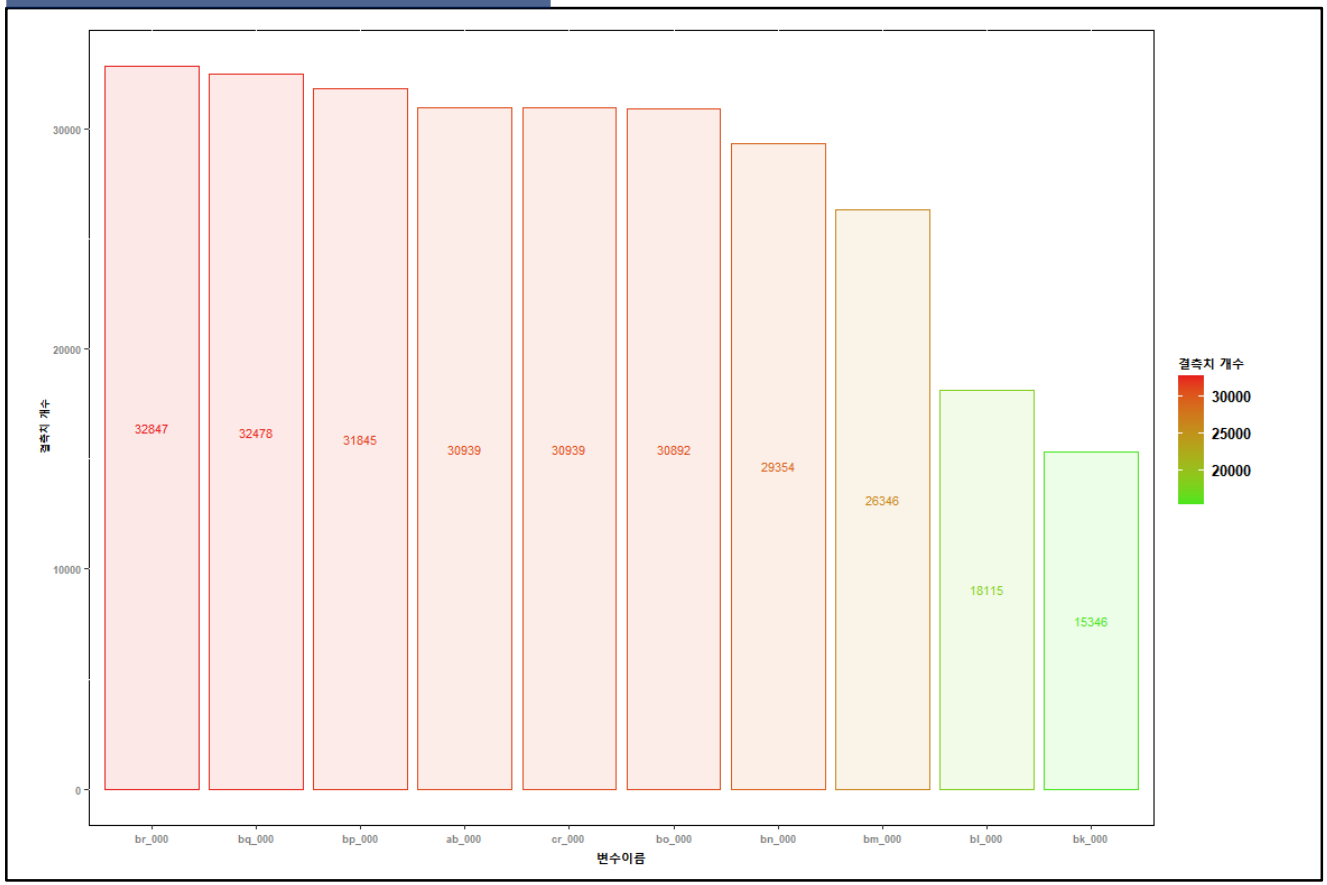
차원이 매우 커
Column들 간의
다중공선성 해결
어려움

+

NA가 없는
Column은 하나뿐

EDA

col: NA 개수 시각화



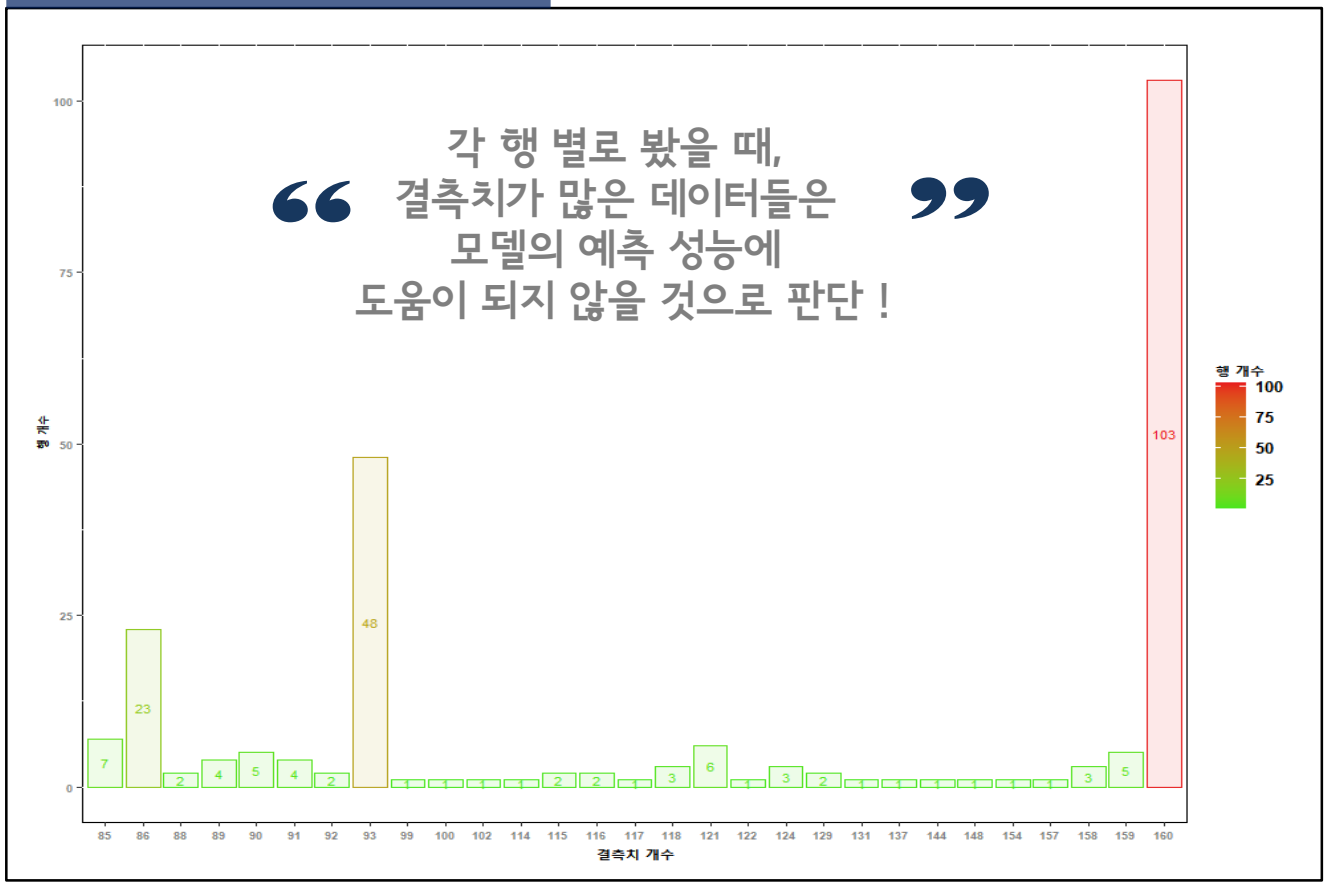
NA가 20000개↑
(50% 이상)
변수 제거

변수 163개로
차원 축소

EDA

NA 개수 클래스 불균형 0 개수

row: NA 개수 시각화

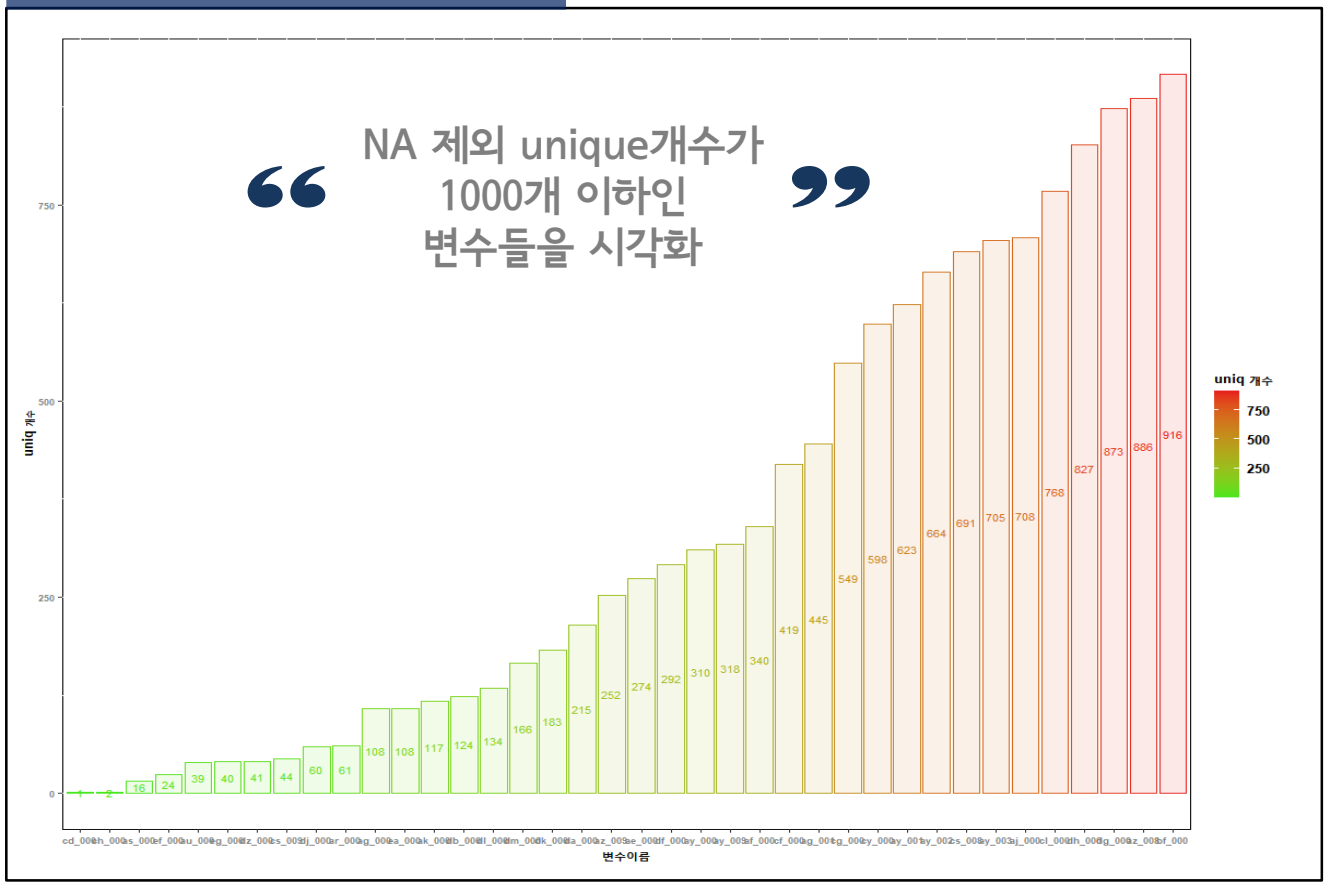


NA가 85개↑
(50% 이상)
행 제거

row 39764개로
차원 축소

EDA

col: unique 개수 시각화



추가적으로
각 열의
unique개수를
분석

unique 값이
2 이하인
변수 삭제

EDA

EDA를 통한 train data 수정

NA 문제가 많았던
변수와 데이터

40000 row
*
172 col

NA 비율이
높았던
열과 행 제거

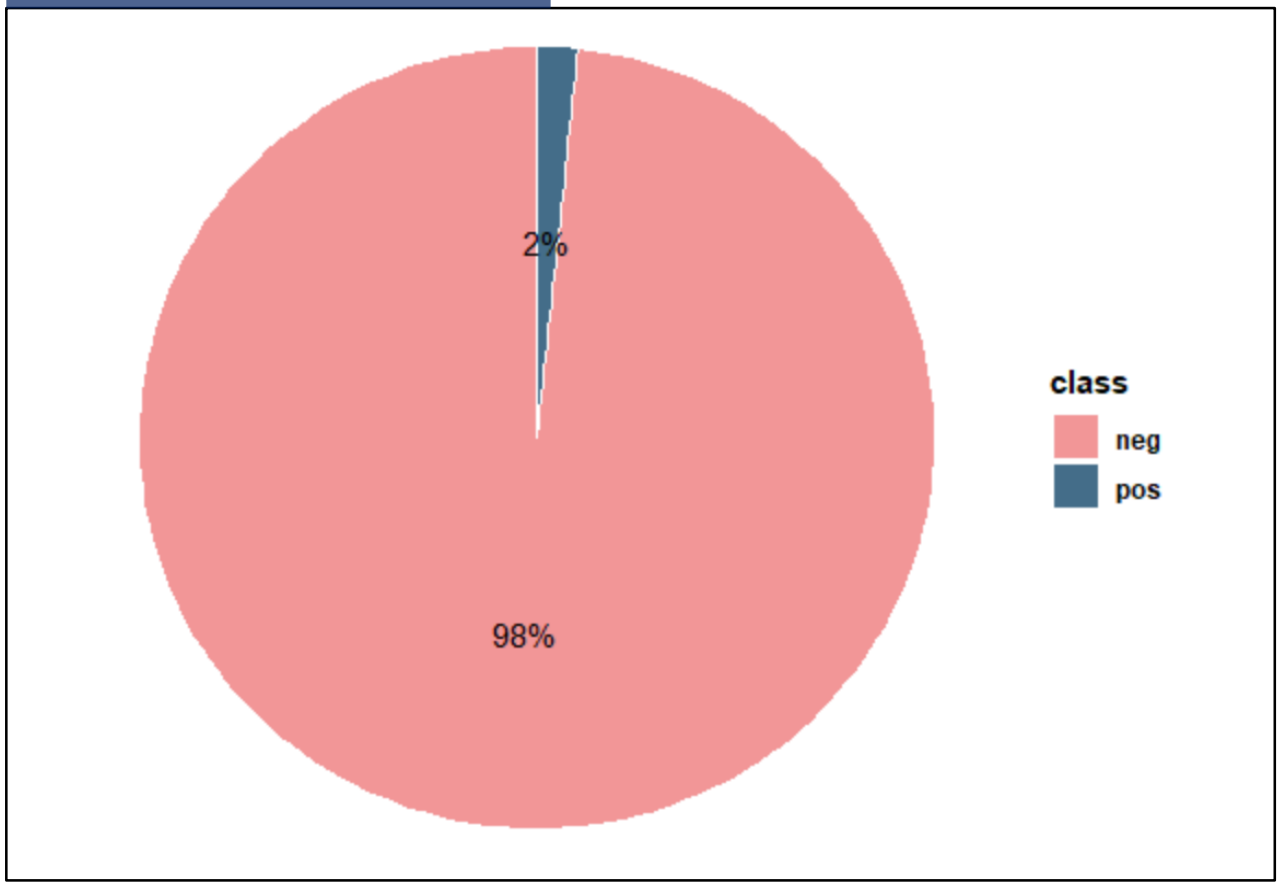
이후 모든 과정에서
train_delect 사용

39764 row
*
162 col

EDA

NA 개수 | **클래스 불균형** | 0 개수

클래스 불균형 시각화



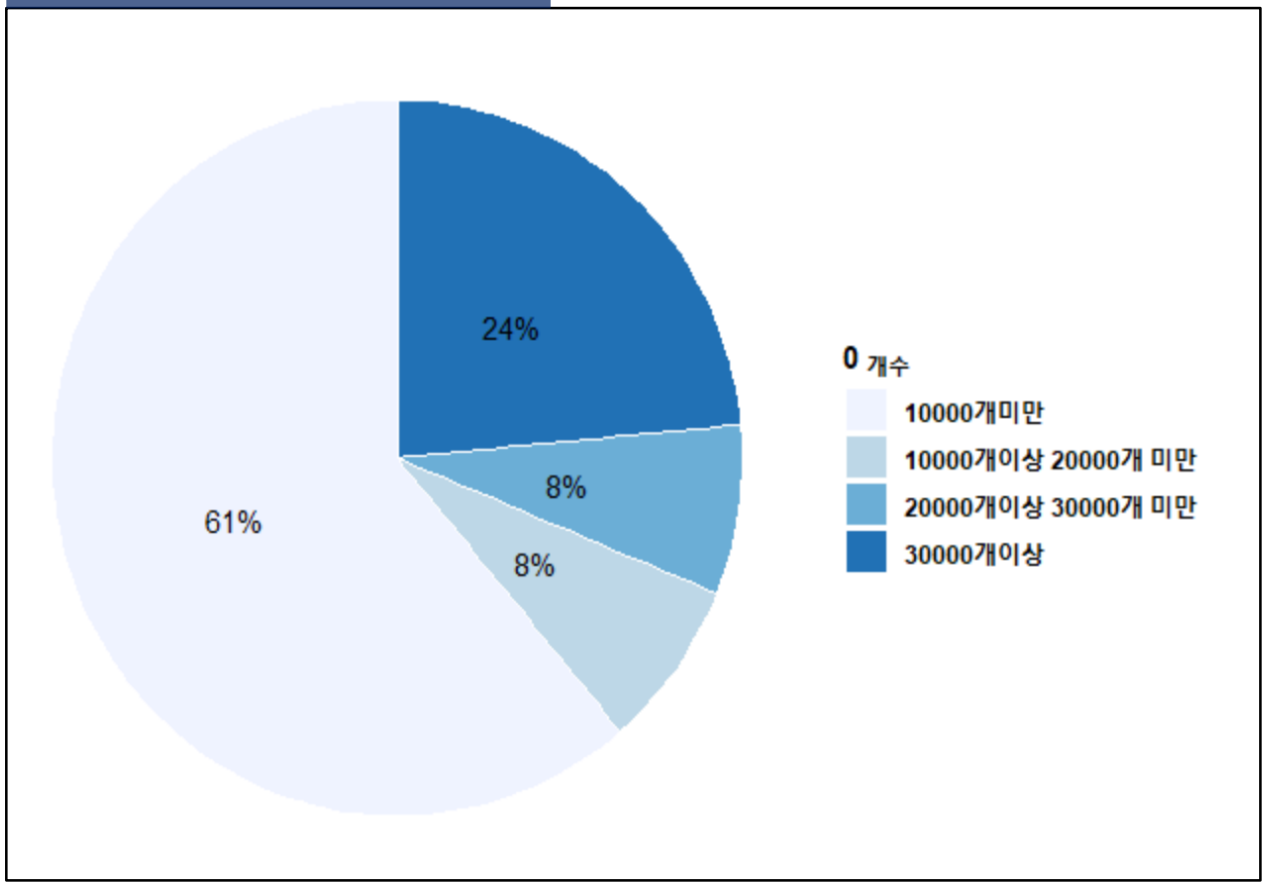
class 변수
neg : pos
매우 불균형한
비율

편향된 모델링을
피하기위해
클래스 불균형을
해결해야함

EDA

NA 개수 | 클래스 불균형 | **0 개수**

column 0 개수 시각화



0의 개수가
10000개 이상인
Column 들이
40%

Mode, median으로
결측치를 대체하기는
어려울 것으로 판단

2

데이터 전처리

결측치 대체

: 데이터에 많은 결측치가 존재할 경우 이를 다 제거하면 데이터의 손실이 있기에 NA를 대체하여 사용함

1. 단일대체법

: 1개의 대표할 수 있는 값으로 대체하는 방법

- Mean Imputation
- Mode Imputation

2. 다중대체법

: 각 결측치를 2개이상의 값으로 대체하는 방법

- KNN
- MICE

① Mean / mode Imputation

: 데이터의 요약 통계량(평균, 최빈값)등을 활용하여 NA를 대체

장점 직관적이며 간단하게 결측치를 대체할 수 있음

단점 모델의 편향을 높여 모델링 결과에 안 좋은 결과를 끼칠 수 있음

결측치 대체

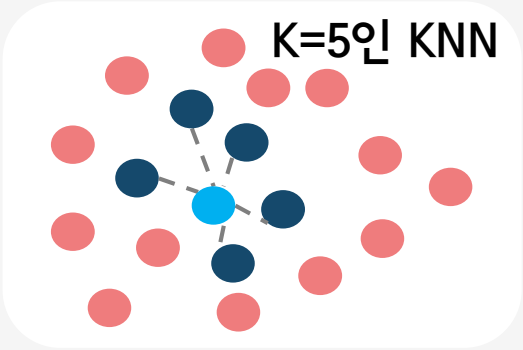
: 데이터에 많은 결측치가 존재할 경우 이를 다 제거하면 데이터의 손실이 있기에 NA를 대체하여 사용함

② KNN

: K개의 최근접 이웃을 구하여 이를 이용해 결측치 대체

장점 Mean, Mode에 비해 비교적 정확함

단점 연속형 데이터의 결측치를 처리할 때만 사용 가능



③ MICE : 연쇄 방정식을 사용한 다중대체방법

결측치를 단순히 대체하여 자료 생성➡ 결측치가 있는 변수들을 모델링한 후 예측값을 대입 ➡ NA를 계속 업데이트

장점 범주형 변수의 NA를 처리할 수 있음

단점 결측치가 랜덤으로 생겨났다고 가정 (MAR)

Imputation 데이터 변환

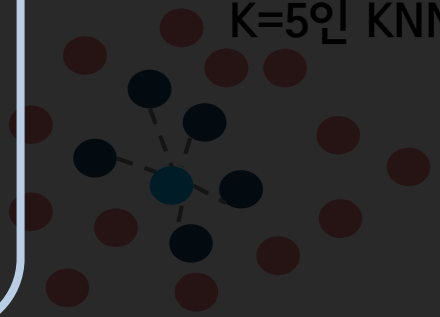
결측치 대체

: 데이터에 많은 결측치가 존재할 경우 이를 다 제거하면 데이터의 손실이 있기에 NA를 대체하여 사용함

Mean Imputation
KNN = 5
KNN = 10
MICE

② KNN

: K개의 최근접 이웃을 구하여 이를 이용해 결측치 대체



장점 Mean, Mode에 비해 비교적 정확한

단점 연속형 데이터의 결측치를 처리할 때만 사용 가능

③ MICE : 연쇄 방정식을 사용한 다중대체방법

결측치를 단순히 대체하여 자료 생성 → 결측치가 있는 변수들을 모델링한 후 예측값을 대입 → NA를 계속 업데이트

다양한 imputation 방법을 모두 시도!

장점 범주형 변수의 NA를 처리할 수 있음

단점 결측치가 랜덤으로 생겨났다고 가정 (MAR)

2 데이터 전처리

Imputation

데이터 변환

데이터	PCA여부	로지스틱 cost	로지스틱 F1 score
MEAN	O	105890	0.985194
KNN=5	O	100840	0.9858436
KNN=10	O	100830	0.9858868
MICE	X	37980	0.9947626
MICE	O	48790	0.9730249

모든 데이터를
train-val(7:3)으로 나눠
로지스틱 회귀 모델에
적용 후 결과 비교

MICE를 활용하여
결측치를 대체한 데이터를
최종 데이터로 선택!

Imputation

데이터 변환

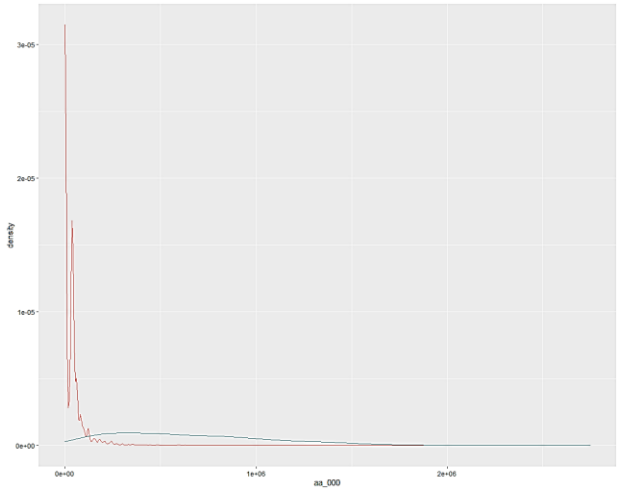
Yeo-Johnson Power Transformations

: 분산을 안정화 시키기 위한 방법의 일종으로 실수전체를 정규화 시키는 방법

데이터가 한쪽으로 쏠린 데이터가 많아 이를 해결하기 위해 Yeo-Johnson변환을 적용

$$\psi(\lambda, y) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

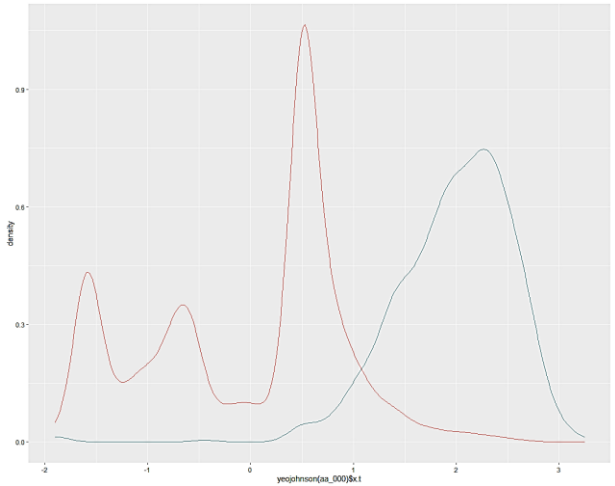
aa_000변수 Yeo-Johnson변환 적용 전



pos

neg

aa_000변수 Yeo-Johnson변환 적용 후



3

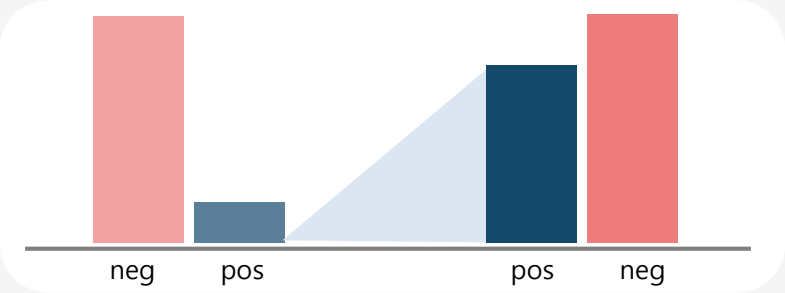
모델링

모델링

클래스 불균형 처리

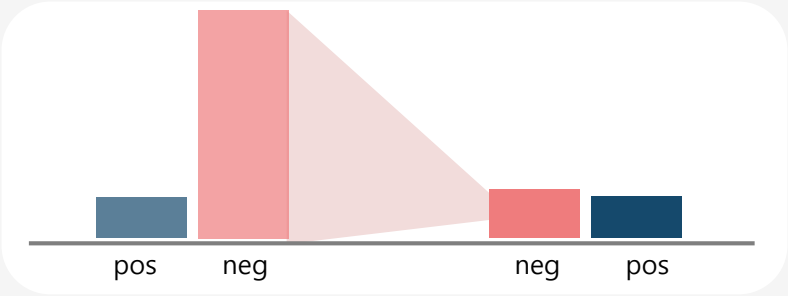
불균형 상태의 모델링은 다수에 속하는 neg쪽으로 편향된 학습을 하게 되므로 처리가 필요

과대표집
(Over-sampling)



- 소수 데이터를 다수 데이터 수준으로 크게 복제 샘플링하는 방법
- 정보의 손실이 없음
- 오버피팅의 위험

과소표집
(Under-sampling)



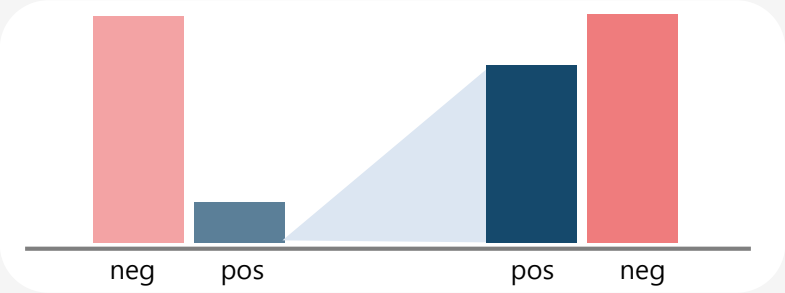
- 다수 데이터를 소수 데이터에 맞춰 적게 샘플링하는 방법
- 유의한 데이터만 남김
- 정보 유실의 문제

모델링

클래스 불균형 처리:

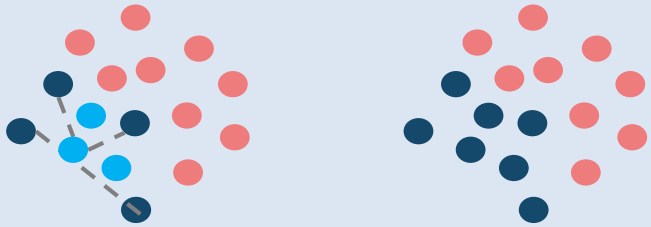
불균형 상태의 모델링은 다수에 속하는 neg쪽으로 편향된 학습을 하게 되므로 처리가 필요

과대표집
(Over-sampling)



- 소수 데이터를 다수 데이터 수준으로 크게 복제 샘플링하는 방법
- 정보의 손실이 없음
- 오버피팅의 위험

SMOTE

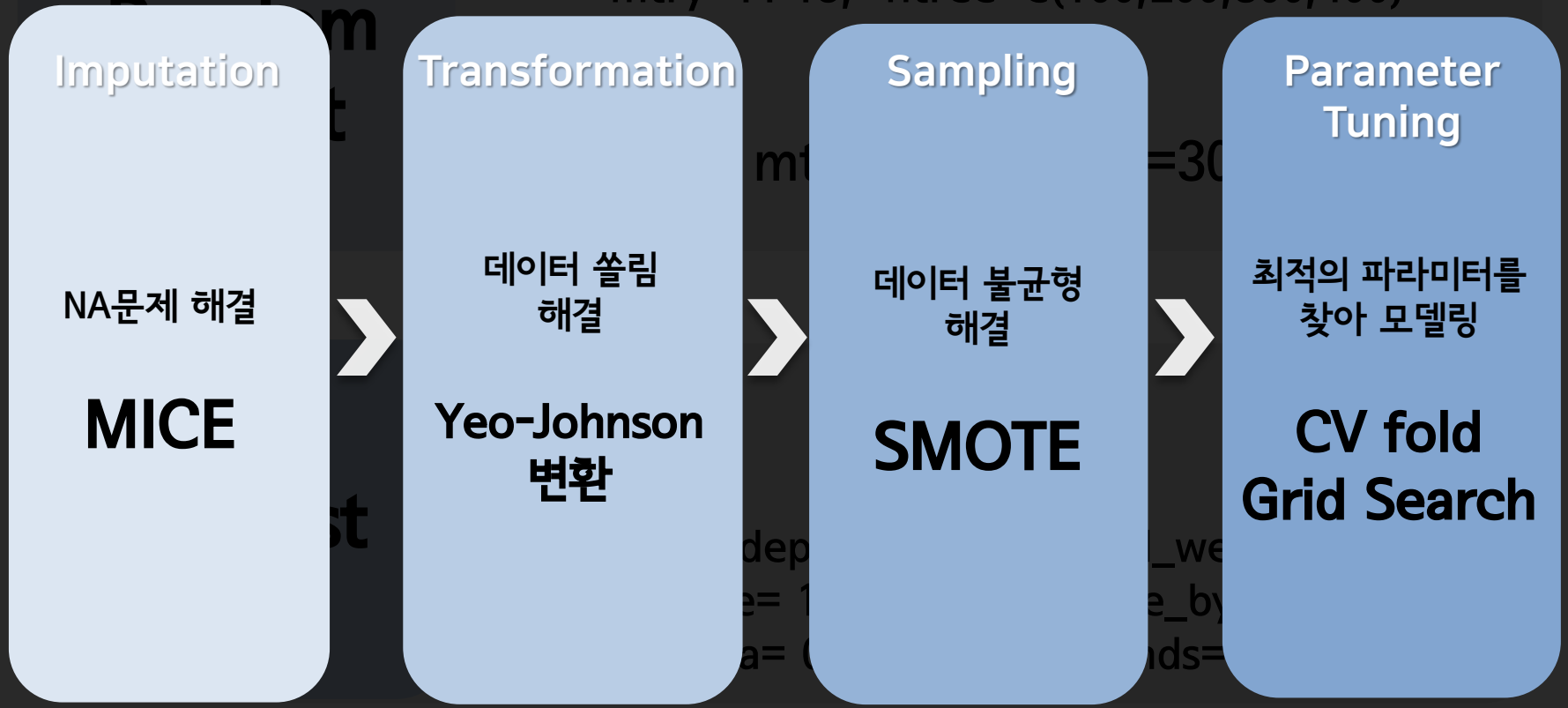


- 소수 데이터의 주변 이웃을 고려해 만든 점들을 샘플로 추가하는 방법
- 오버피팅의 위험이 줄어듦

grid search ⇒ 최적의 파라미터 탐색

<모델링 flow 복습과 예습>

mtry=11:15, ntree=c(100,200,300,400)



grid search ⇒ 최적의 파라미터 탐색

Random Forest

mtry=11:15, ntree=c(100,200,300,400)

↓

mtry=12, ntree=100

Xgboost

grid search

↓

max_depth=8 / min_child_weight=5,
subsample= 0.8413 / colsample_bytree=0.8651,
eta=0.06714255 / nrounds=1000

4개의 모델을 비교 ⇒ 최소비용을 가지는 모델 선택

<Logistic Regression>

Cost = 47534.29
F1 score = 0.968794

<Xgboost>

Cost = 6337.143
F1 score = 0.9857033

<Random Forest>

Cost = 4577.143
F1 score = 0.9857843

<LightBGM>

Cost = 5230
F1 score = 0.629291

4

결론 / 의의와 한계

4 결론 / 의의와 한계

최종 결론 의의와 한계

< test 예측하기 >

Train 데이터셋
Yeo-Johnson 변환
& SMOTE

RandomForest 모델링

Test 데이터셋에 적용

Kaggle 결과

F1 score = 0.97250

4 결론 / 의의와 한계

최종 결론

의의와 한계

데이터에 NA, 0이 많아 예측의 정확도를 높이는데 한계가 있었음

변수에 대한 정보가 없어 분석 및 결과 해석 과정에서 어려움이 있었음

EDA를 더 열심히 해서 발견해낸 패턴으로 Feature Engineering까지 진행했다면..?

지난학기 학회활동(클린업, 주분, 패키지)을 되돌아 볼 수 있는 시간이었다!!!

R, python을 망라한 다양한 모델링 기법을 사용했다!!!

Imputation 종류, 차원축소여부, 모델링기법 등 다양한 경우의 수를 고려해 최소비용을 예측했다!!!

THANK YOU