보험금 청구 건에 대한 자동지급, 심사, 조사분류 예측

미래에셋 4팀 김은지 신보람 조경민

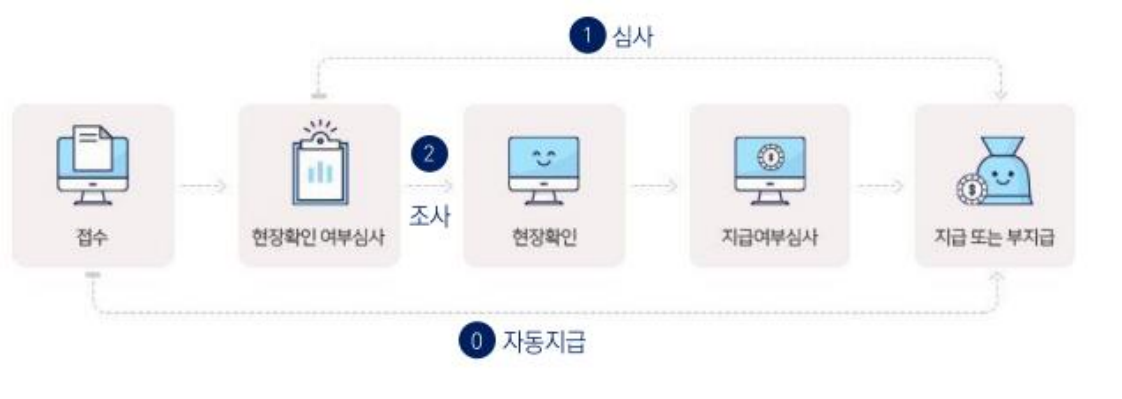
목 차

1. 데이터 소개

데이터 분석 목적

과제 소개

- 보험 고객은 질병 발생 시 보험금을 신청(청구)합니다.
- 고객이 보험금을 신청하면 위험도를 판별하여 자동 지급, 심사, 조사로 분류한 후, 보험금 지급 여부를 결정합니다.
- 보험금 청구 건에 대한 분류 결과를 Target으로 하여 다음 달의 청구 건에 대한 분류 결과를 예측하는 것이 과제입니다.



데이터 설명

- 데이터는 2019년 1월 부터 11월 까지의 월별 보험금 청구 데이터로 구성
- 청구 건 별로 고객, 상품, 판매자, 질병 정보가 나열되어 있으며 모든 정보는 접수일련번호에 대해 Unique하다.

데이터 분석 목적

파일의 구성

구분	학습용 데이터셋	테스트용 데이터셋 (문제파일)	제출용 파일 (답안 파일)
	train.csv	test.csv	sample.csv
구성 (행 개수와 변수 개수)	377,928개의 행 34개의 변수 (ID + 속성 + target)	22,072개의 행 33개의 변수 (ID + 속성)	22,072개의 행 2개의 변수 (ID + target)
기간	2019. 01 ~ 2019. 11	2019. 12	2019. 12

모델 생성 → target값 예측 → 최종 F1 Score

변수 설명

질병 정보 (21)

변수명	변수 설명	변수 종류
dsas_ltwt_gcd	질병 경중 등급 코드	categorical
kcd_gcd	KCD 등급 코드	categorical
dsas_acd_rst_dcd	질병 구분 코드	categorical
ar_rclss_cd	발생지역구분코드	categorical
blrs_cd	치료 행위 코드	categorical
mdct_inu_rclss_dcd	의료기관 구분 코드	categorical
nur_hosp_yn	요양병원 여부	categorical
optt_nbtm_s	접수 건 별 총 통원횟수	int
bilg_isamt_s	접수 건 별 청구보험금 총액	float
hspz_dys_s	접수 건 별 총 입원일수	int

변수 설명

질병 정보 (21)

변수명	변수 설명	변수 종류
hsp_avg_hspz_bilg_isamt_s	병원 별 평균 입원 청구 보험금	float
hsp_avg_optt_bilg_isamt_s	병원 별 평균 통원 청구 보험금	float
hsp_avg_surop_bilg_isamt_s	병원 별 평균 수술 청구 보험금	float
hsp_avg_diag_bilg_isamt_s	병원 별 평균 진단 청구 보험금	float
dsas_avg_hspz_bilg_isamt_s	질병 별 평균 입원 청구 보험금	float
dsas_avg_optt_bilg_isamt_s	질병 별 평균 통원 청구 보험금	float
dsas_avg_surop_bilg_isamt_s	질병 별 평균 수술 청구 보험금	float
dsas_avg_diag_bilg_isamt_s	질병 별 평균 진단 청구 보험금	float
hspz_blcnt_s	접수 건 별 입원 청구 건수	int
surop_blcnt_s	접수 건 별 수술 청구 건수	int
optt_blcnt_s	접수 건 별 통원 청구 건수	int

변수 설명

고객, 상품, 판매자 정보 (10)

변수명	변수 설명	변수 종류
isrd_age_dcd	고객 나이 구분 코드	categorical
fds_cust_yn	보험 사기 이력 고객 여부	categorical
smrtg_5y_passed_yn	부담보 5년 경과 여부	categorical
mtad_cntr_yn	중도 부가 계약 여부	categorical
heltp_pf_ntyn	건강인 우대 계약 여부	categorical
prm_nvcd	보험료 구간 코드	categorical
inamt_nvcd	가입 금액 구간 코드	categorical
ac_ctr_diff	청구일 계약일간 기간 구분 코드	categorical
ac_rst_diff	청구일 부활일간 기간 구분 코드	categorical
urlb_fc_yn	부실 판매자 계약 여부	categorical

변수 설명

ID, 시간, 타겟 변수 (3)

변수명	변수 설명	변수 종류
ID	접수 일련 번호	character
base_ym	접수년월	date
target	최종 배정 상태 / 자동지급(0), 심사(1), 조사(2)	categorical

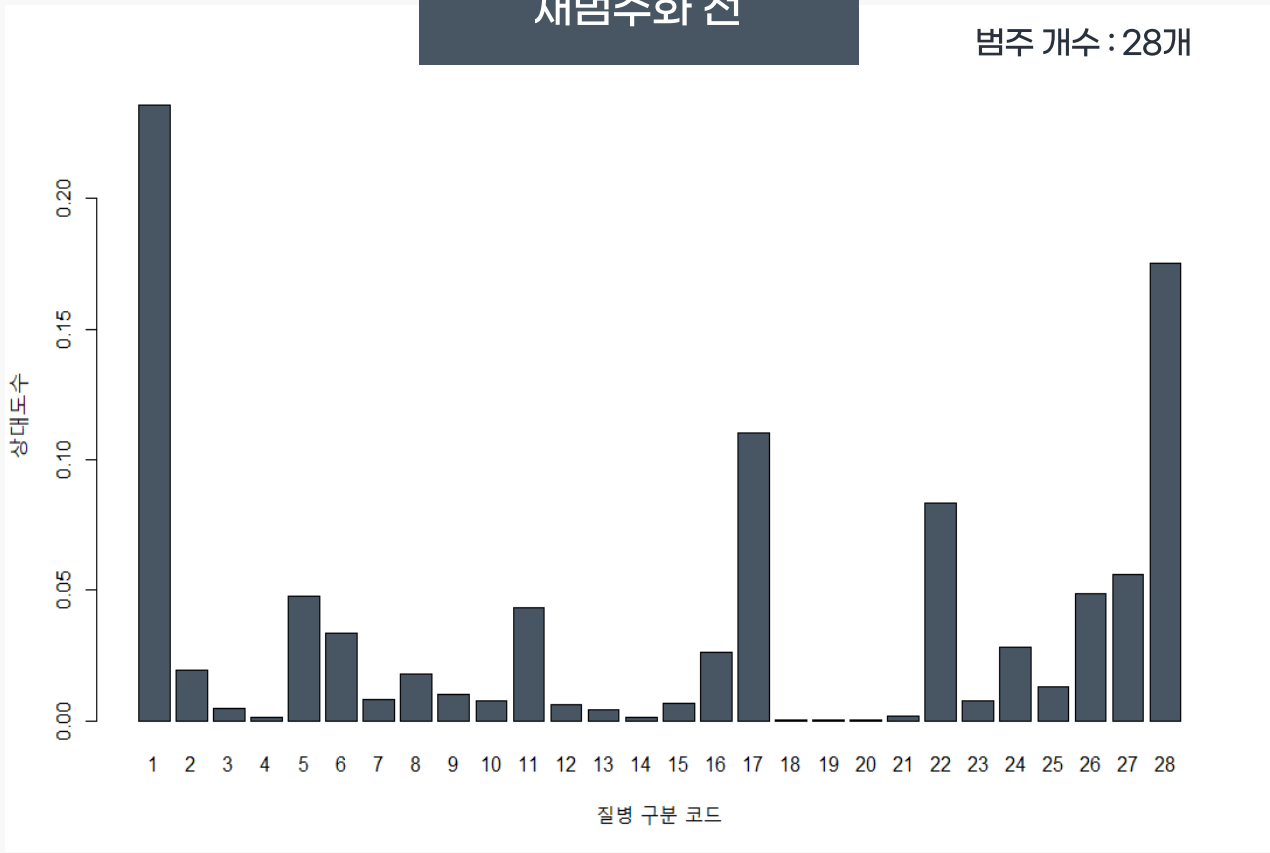
2. 데이터 탐색 및 전처리

범주형 변수 - 재범주화

질병 구분 코드 (dsas_acd_rst_dcd)

재범주화 전

범주 개수 : 28개



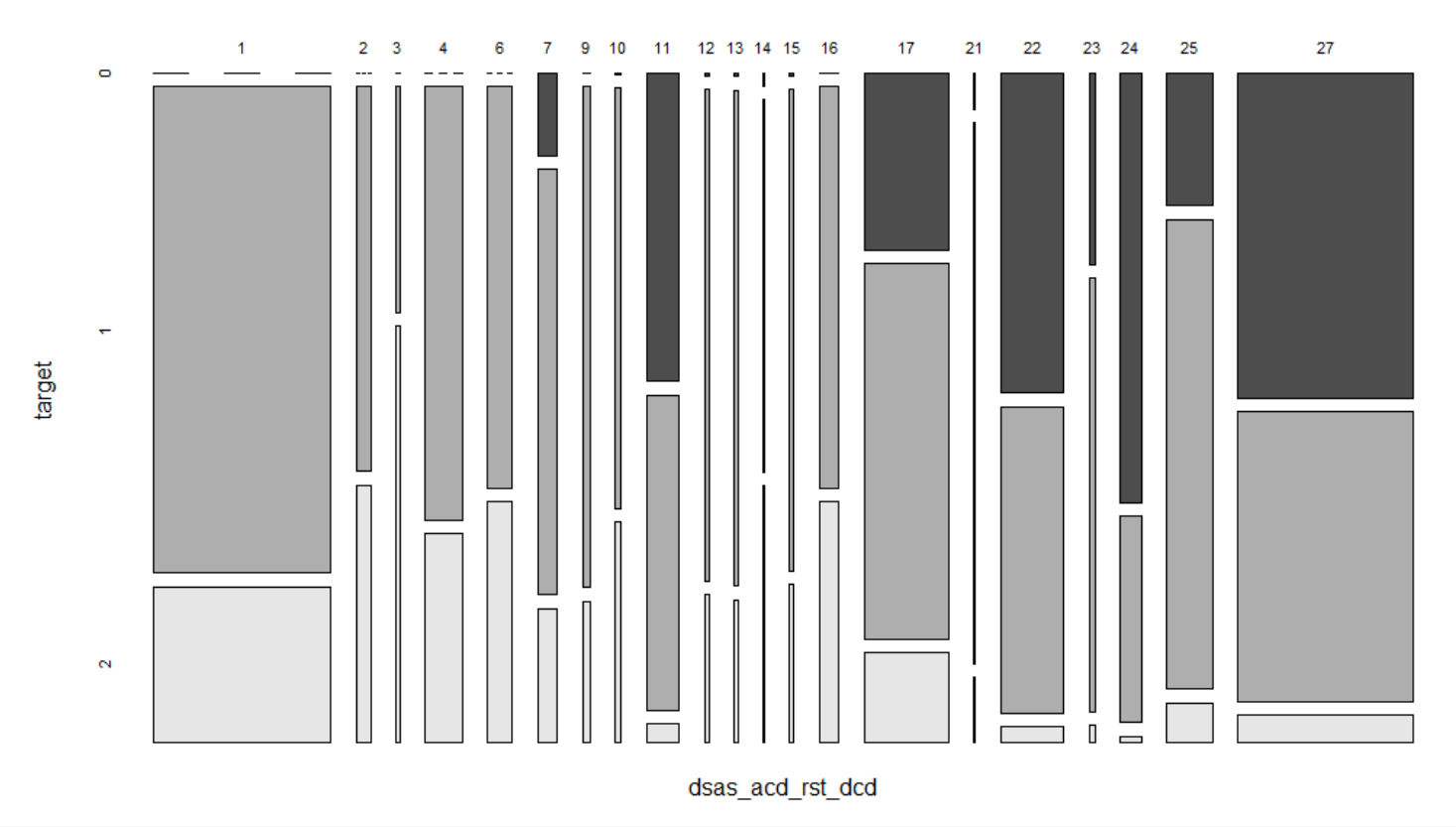
별첨 자료

질병구분코드	질병명	질병구분코드	질병명
1	암	15	고혈압
2	상피내암	16	당뇨병
3	경계성	17	관절염
4	심장질환	18	
5		19	
6	뇌혈관질환	20	
7	간질환	21	골다공증
8		22	백내장
9	신장질환	23	중이염
10	갑상선질환	24	충수염
11	폐렴	25	남성비뇨기계
12	천식	26	
13	위궤양	27	부인과
14	십이지장궤양	28	

질병명이 같은 것끼리 묶음

질병 구분 코드 (dsas_acd_rst_dcd)

Mosaic plot

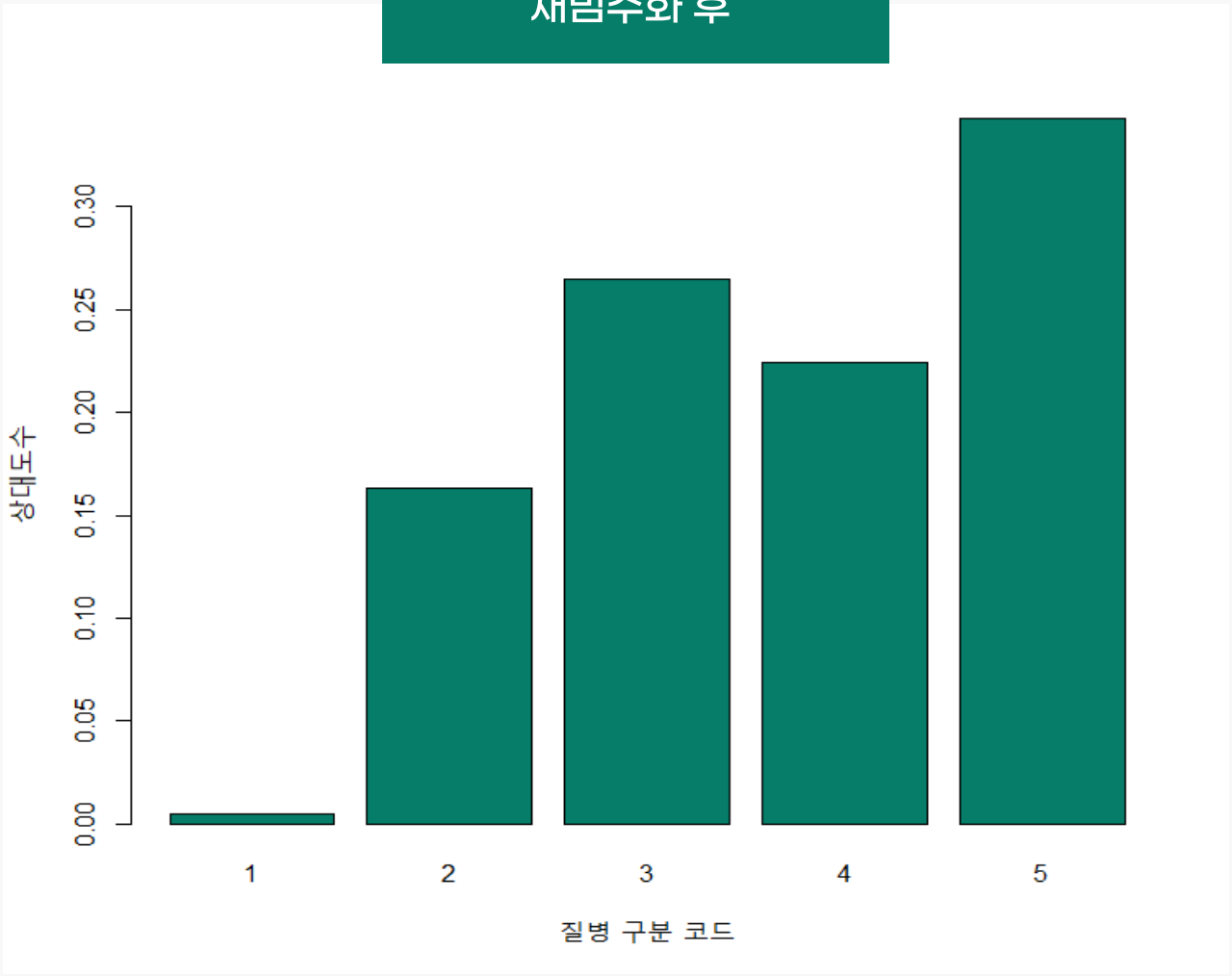


target 비율이 비슷한 범주끼리 묶음

Factor	0	1	2
1	0.00	0.76	0.24
2	0.00	0.60	0.40
3	0.00	0.35	0.65
4	0.00	0.68	0.32
6	0.00	0.63	0.37
7	0.13	0.66	0.21
9	0.00	0.78	0.22
10	0.00	0.66	0.34
11	0.48	0.49	0.03
12	0.00	0.77	0.23
13	0.01	0.77	0.22
14	0.02	0.58	0.40
15	0.01	0.75	0.24
16	0.00	0.62	0.38
17	0.28	0.58	0.14
21	0.06	0.84	0.10
22	0.50	0.48	0.02
23	0.30	0.67	0.03
24	0.67	0.32	0.01
25	0.21	0.73	0.06
27	0.51	0.45	0.04

질병 구분 코드 (dsas_acd_rst_dcd)

재범주화 후



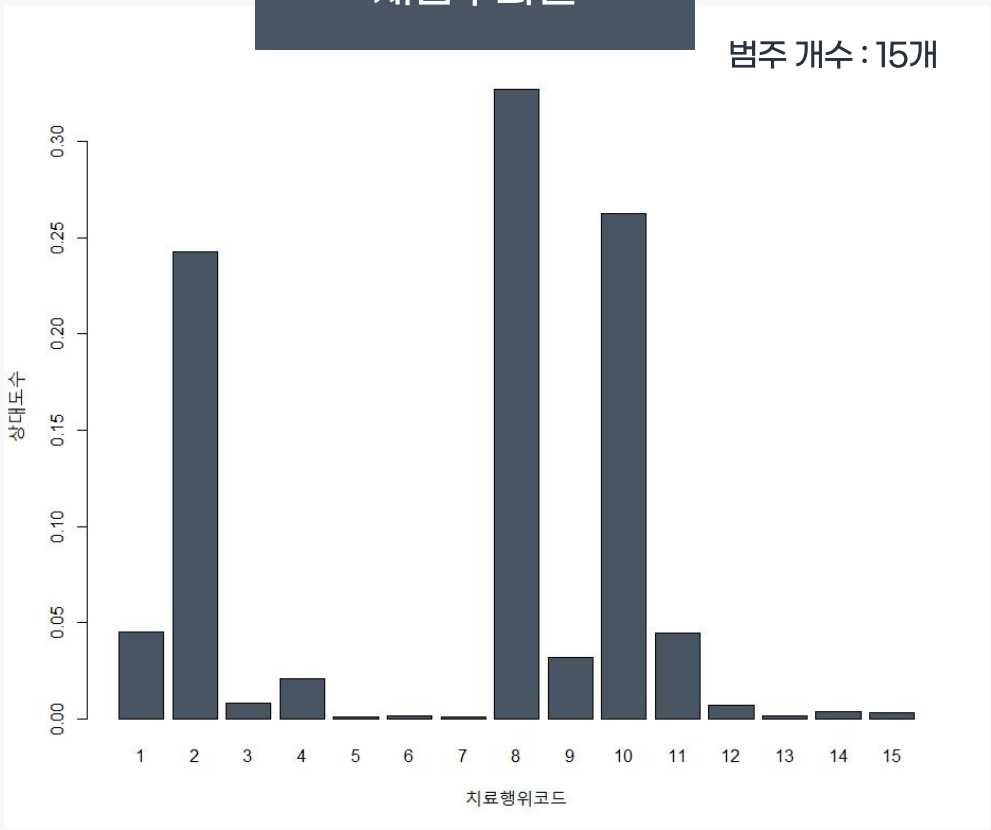
02 데이터 탐색 및 전처리

범주형 변수 - 재범주화

치료행위코드 (blrs_cd)

재범주화전

범주 개수 : 15개



4가지 치료행위의 유무로 분류!

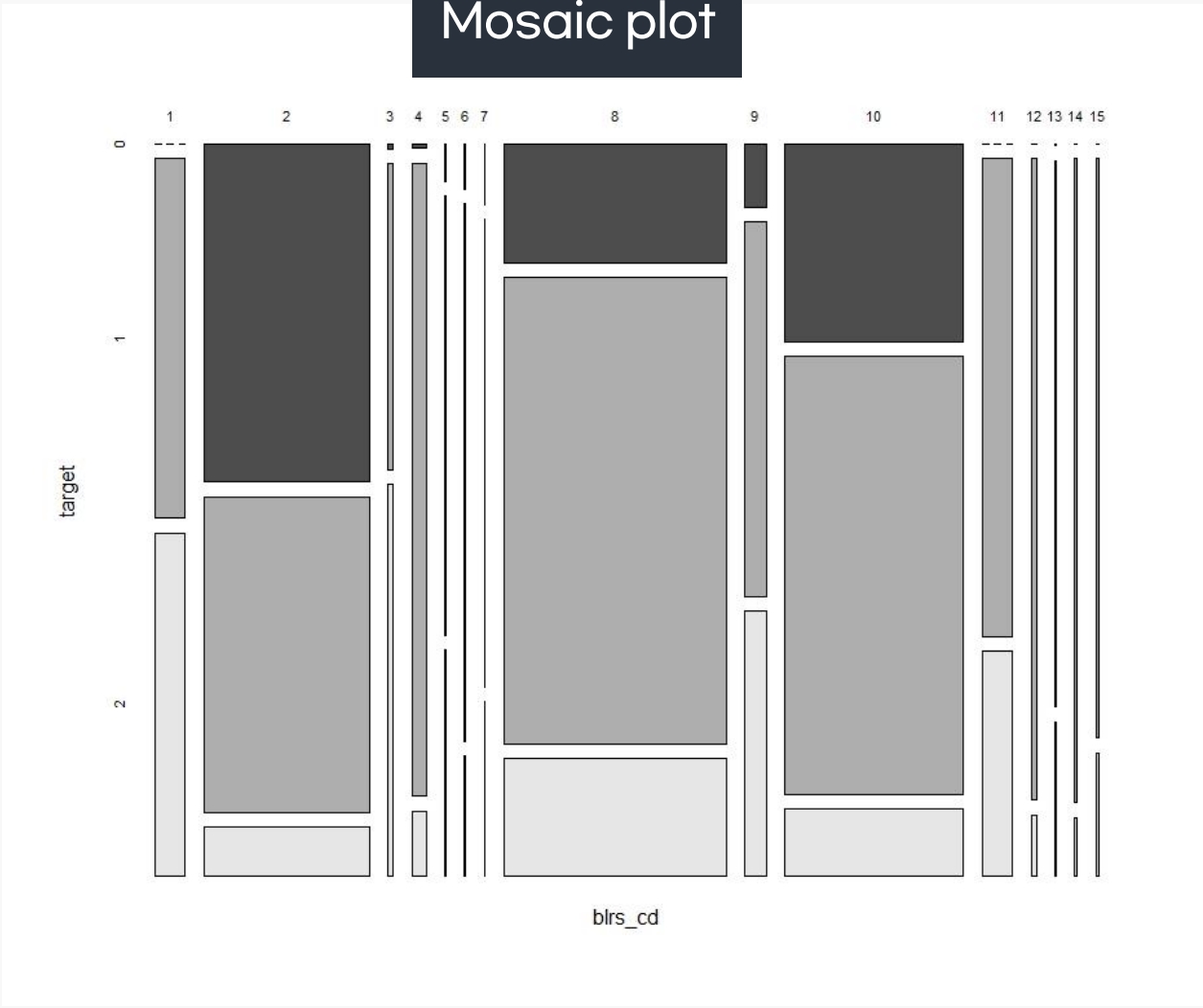
별첨 자료

코드	치료행위				설명
	입원	통원	수술	진단	
1				Y	질병 진단만 받음
2			Y		수술치료만 진행
3			Y	Y	질병 진단 받고 수술치료 진행
4		Y			통원치료만 진행
5		Y		Y	질병 진단 받고 통원치료 진행
6		Y	Y		수술치료 후 통원치료 진행
7		Y	Y	Y	진단 받고, 수술도 받고, 통원치료도 받음
8	Y				입원치료만 진행
9	Y			Y	질병 진단 받고 입원치료 진행
10	Y		Y		입원 및 수술치료 진행
11	Y		Y	Y	질병 진단 받고 입원 및 수술치료 진행
12	Y	Y			입원 및 통원치료 진행
13	Y	Y		Y	질병 진단 받고 입원 및 통원치료 진행
14	Y	Y	Y		입원, 수술, 통원치료 모두 진행
15	Y	Y	Y	Y	질병 진단 받고 입원, 수술, 통원치료 모두 진행

치료행위코드 (blrs_cd)

target 비율이 비슷한 범주끼리 묶음

Mosaic plot



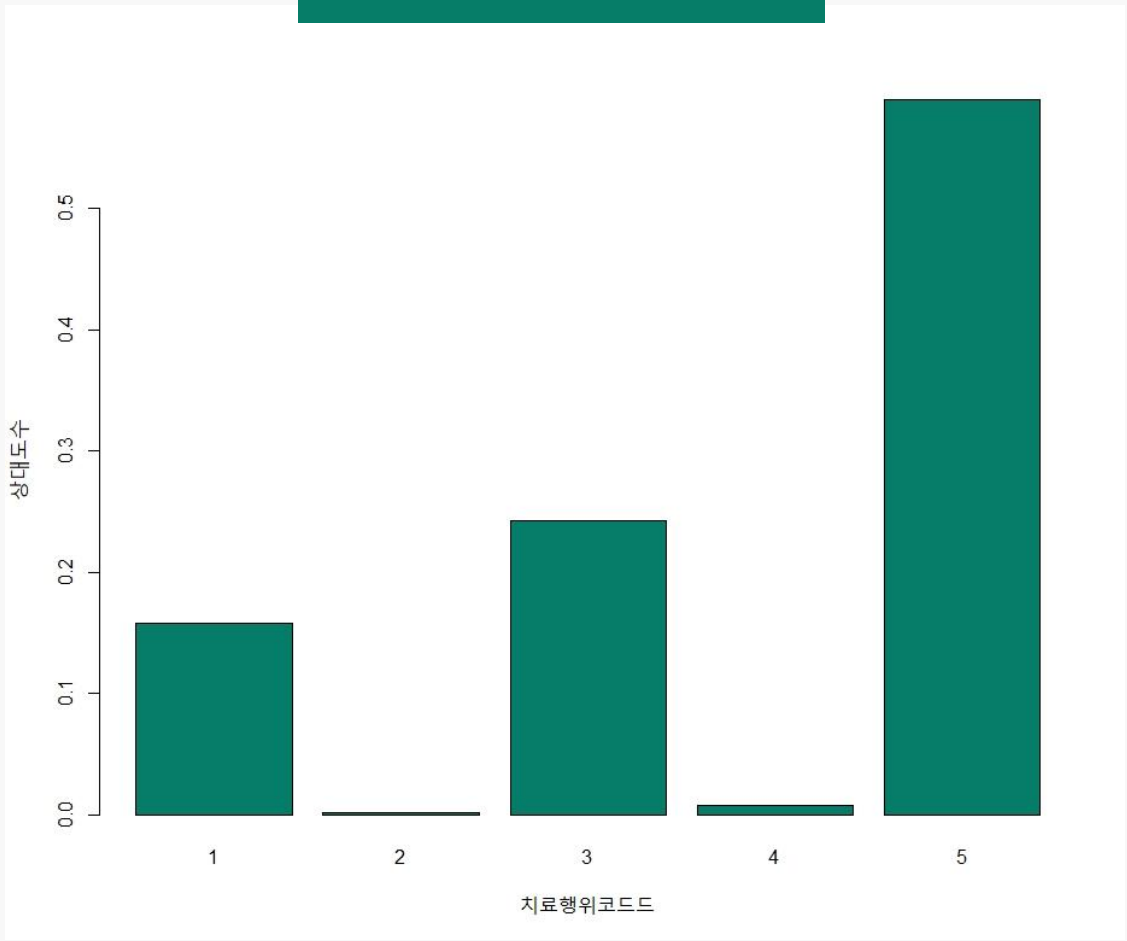
Target 비율이 큰 순서	재범주화
"1->2->0”-1	1
"1->2->0”-2	2
"0->1->2"	3
"2->1->0"	4
"1->0->2"	5

02 데이터 탐색 및 전처리

범주형 변수 - 재범주화

치료행위코드 (blrs_cd)

재범주화 후



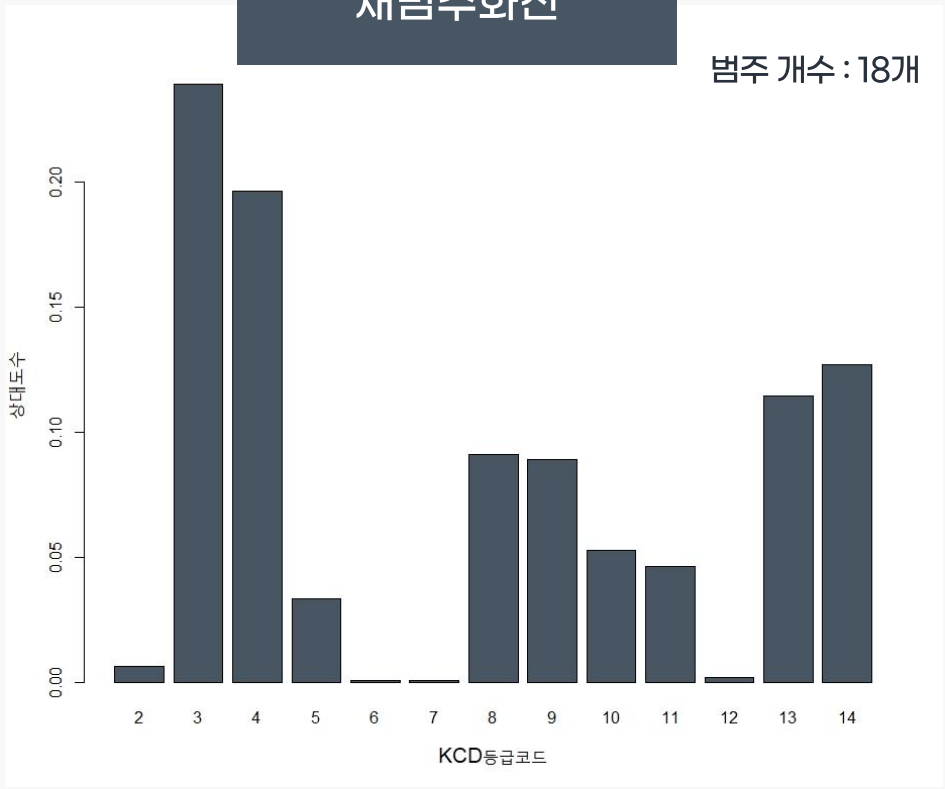
02 **데이터 탐색 및 전처리**

범주형 변수 - 재범주화

KCD등급코드 (kcd_gcd)

재범주화 후

재범주화전

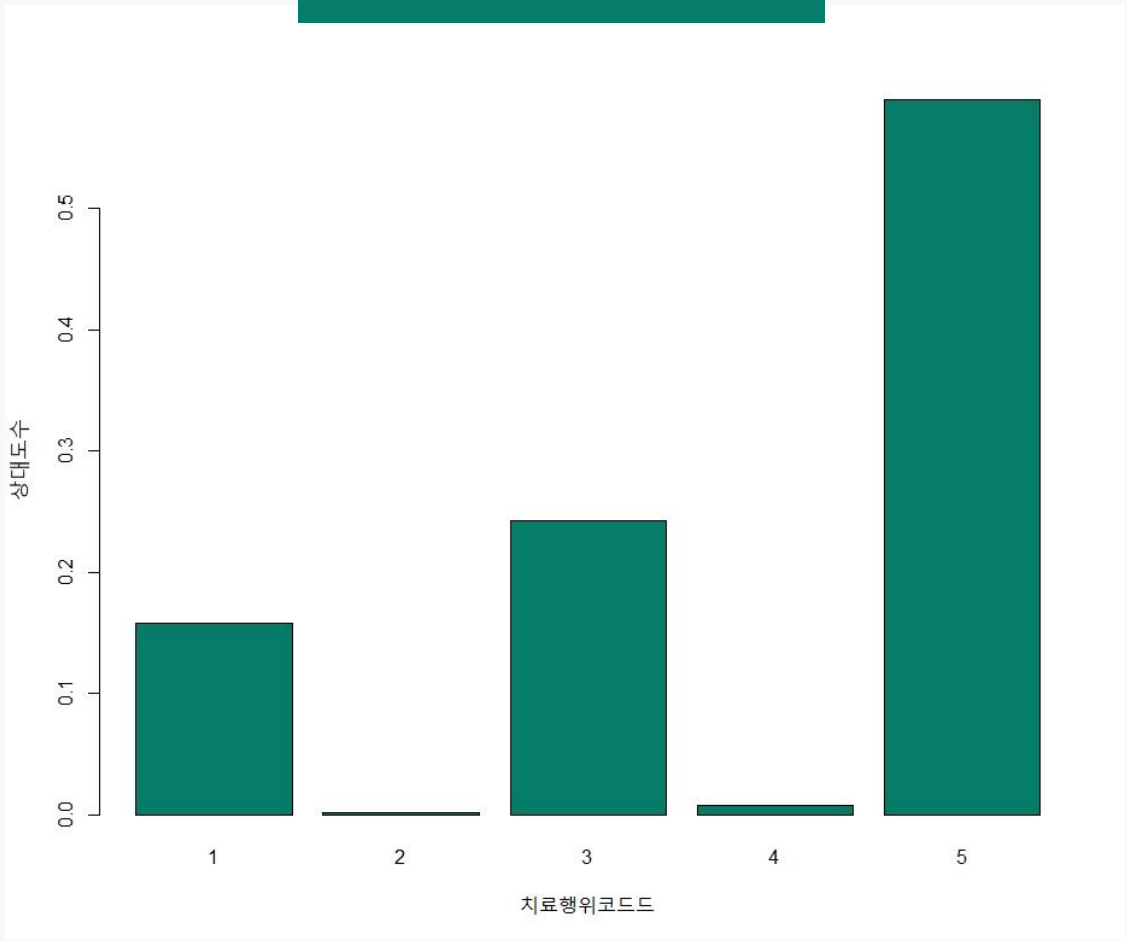


별첨 자료

KCD등급코드	질병기준	KCD등급코드명
1	A	감염성 및 기생충성 질환1
2	B	감염성 및 기생충성 질환2
3	C	신생물(암) 질환
4	D	신생물(기타)질환
5	E	내분비 질환
6	F	정신 질환
7	G	신경계통 질환
8	H	눈, 귀 질환
9	I	순환기 질환
10	J	호흡기 질환
11	K	소화기 질환
12	L	피부 질환
13	M	근골격계 질환
14	N	비뇨생식기 질환
15	O	임신, 출산 질환
16	P	주산기 질환
17	Q	선천 질환
18	R	달리 분류되지 않은 질환

KCD등급코드 (kcd_gcd)

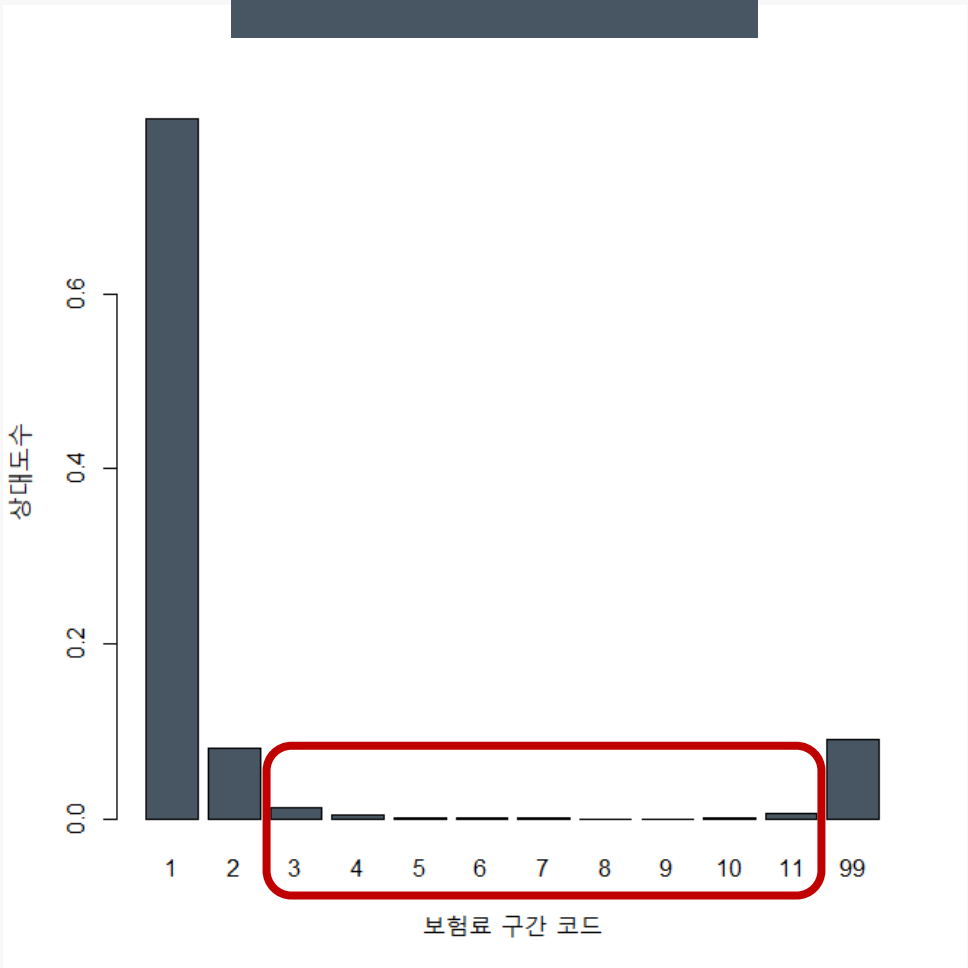
재범주화 후



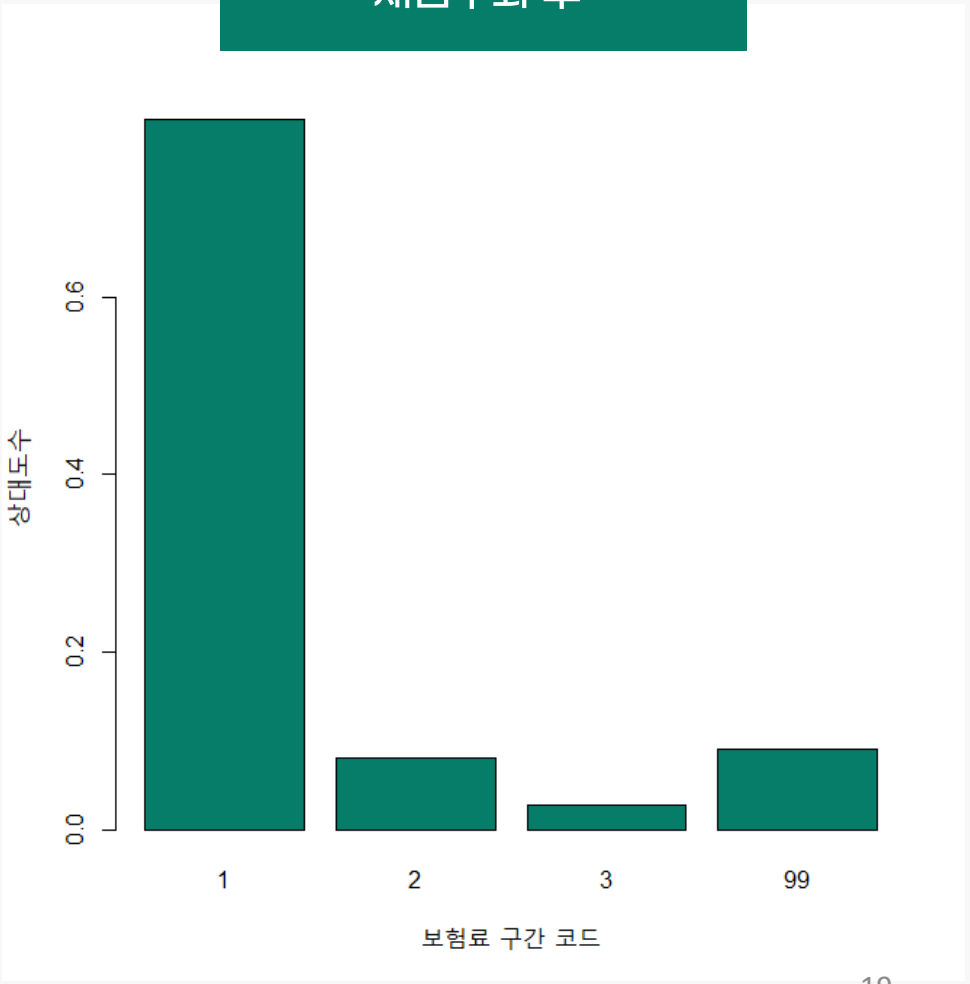
범주형 변수 - 재범주화

보험료 구간 코드 (prm_nvcd)

재범주화 전



재범주화 후

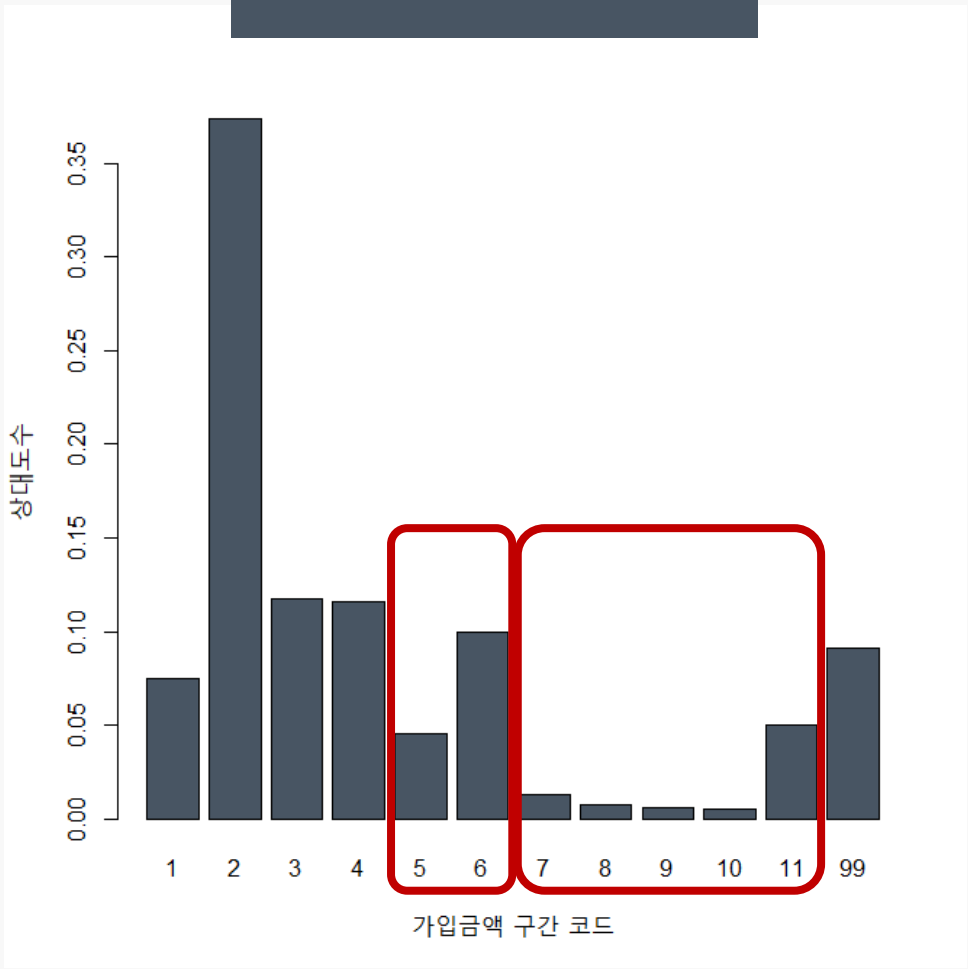


02 데이터 탐색 및 전처리

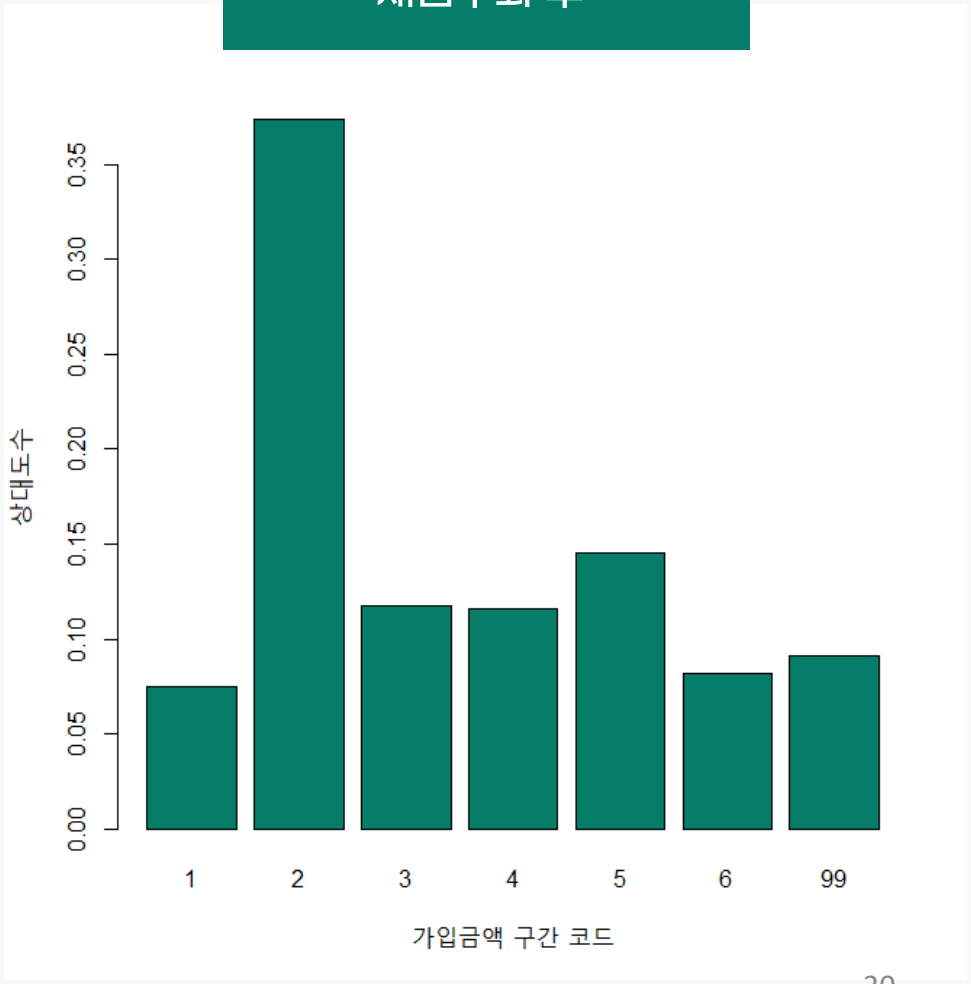
범주형 변수 - 재범주화

가입금액 구간 코드 (inamt_nvcd)

재범주화 전



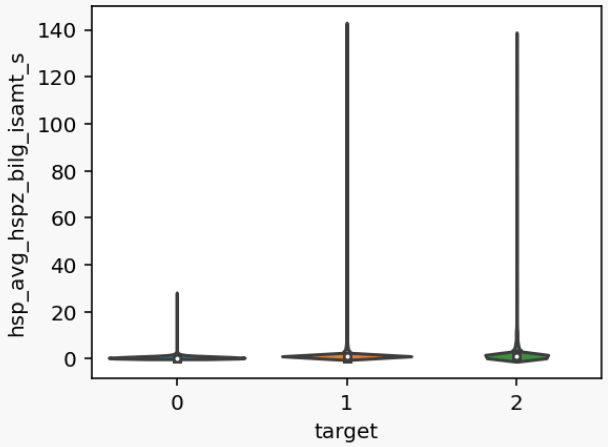
재범주화 후



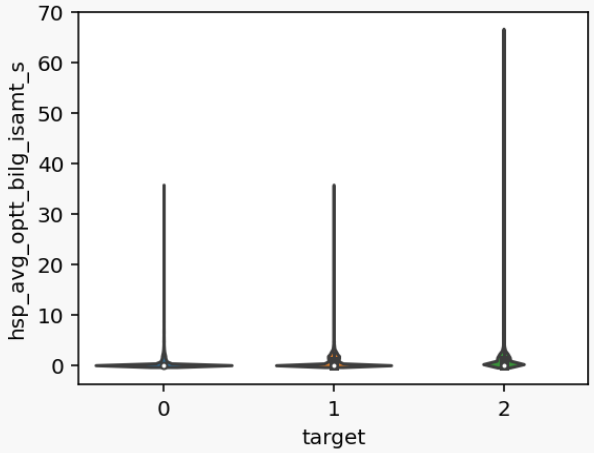
수치형 변수들 분포 확인

→ target 별 분포 확인

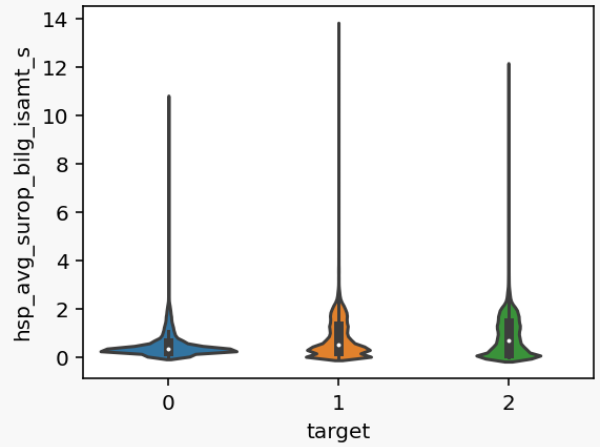
병원별평균입원청구보험금



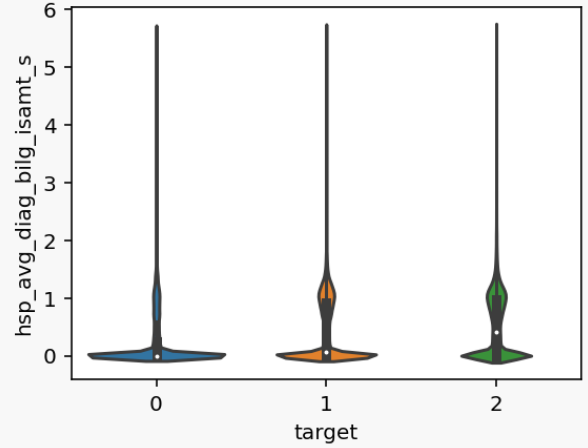
병원별평균통원청구보험금

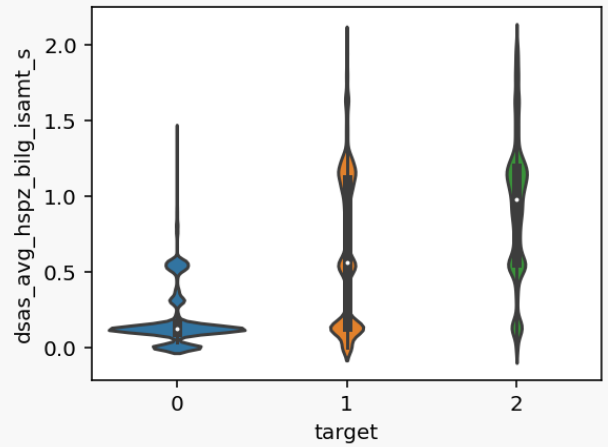


병원별평균수술청구보험금

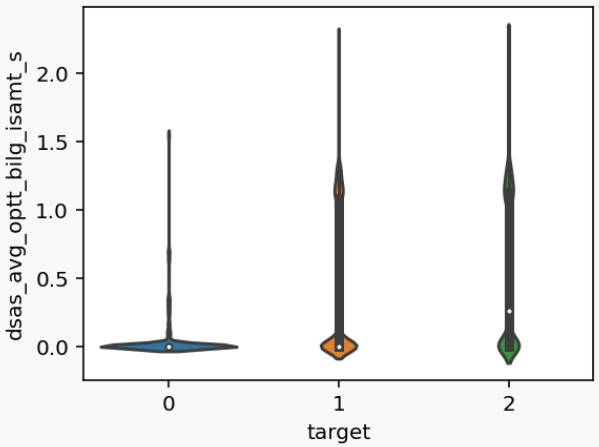


병원별평균진단청구보험금

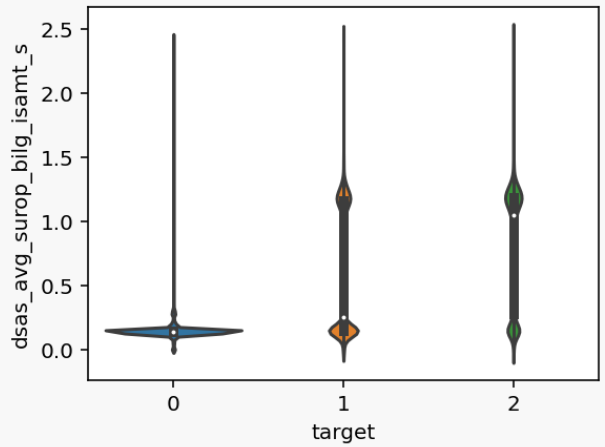




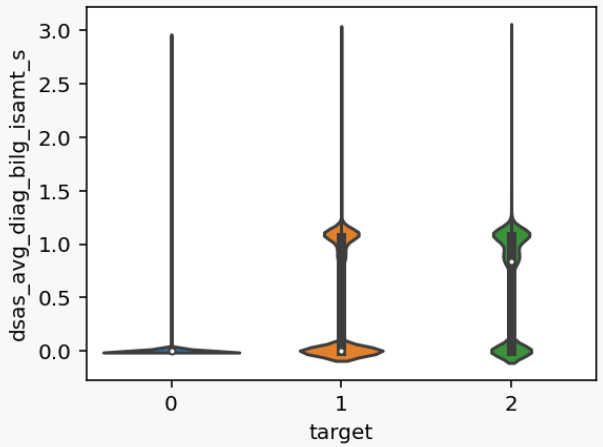
질병별평균입원청구보험금



질병별평균통원청구보험금



질병별평균수술청구보험금

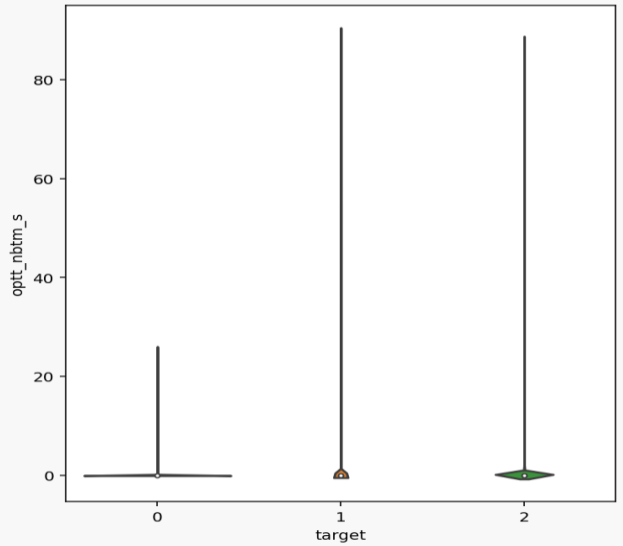


질병별평균진단청구보험금

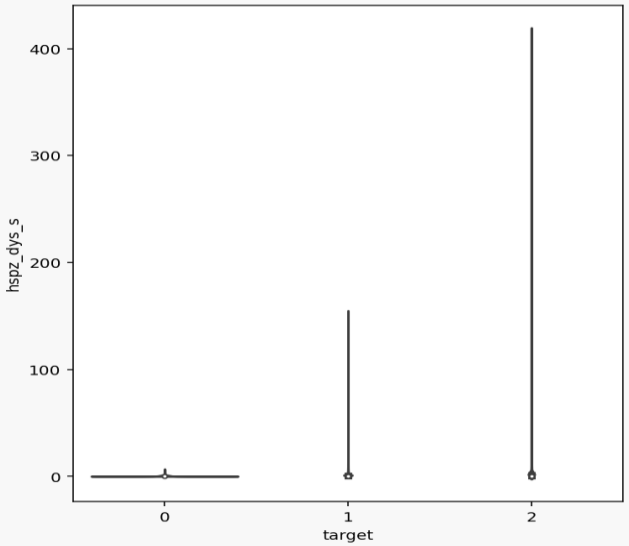
수치형 변수들 분포 확인

→ target 별 분포

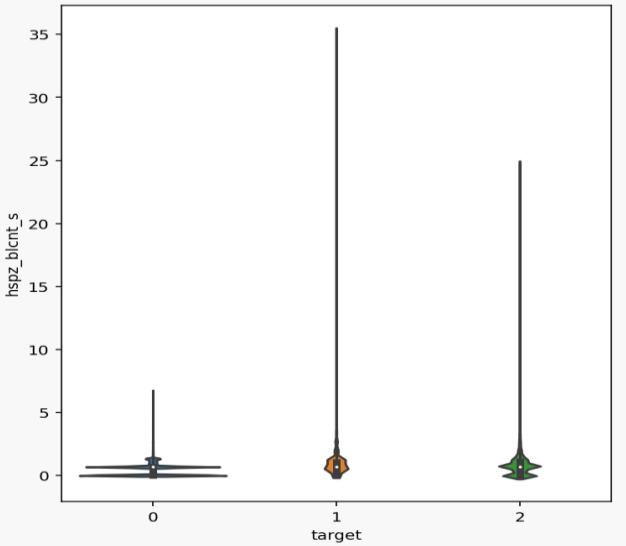
통원횟수



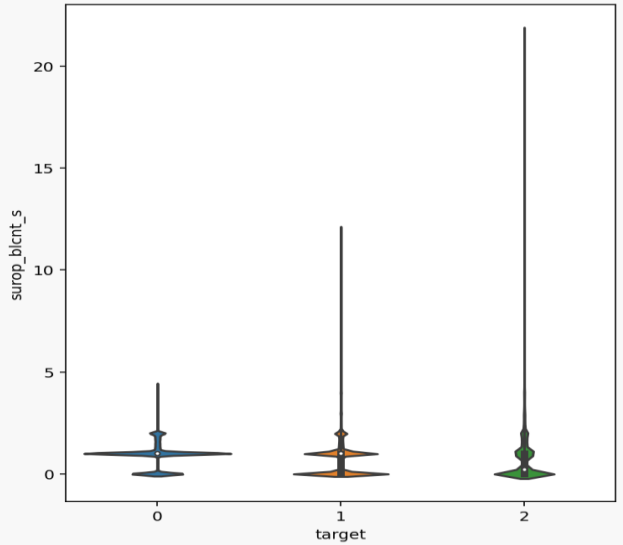
입원일수



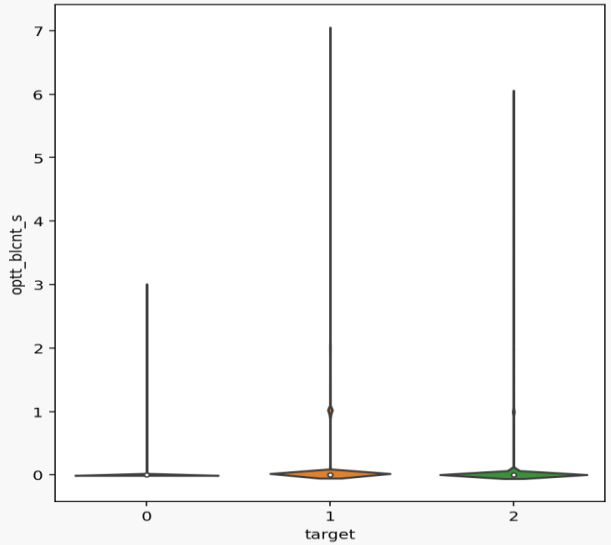
입원청구건수



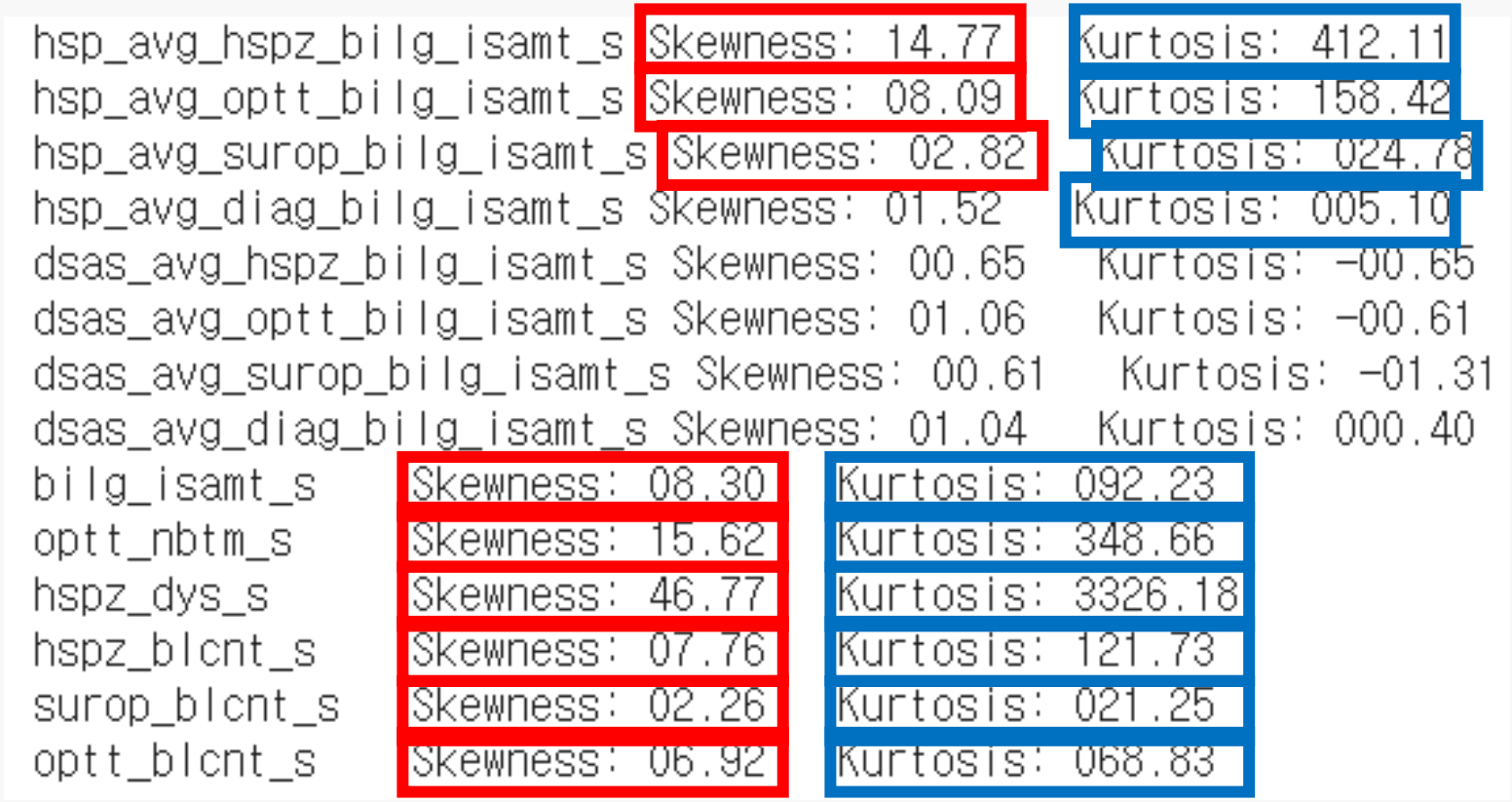
수술청구건수



통원청구건수분포 확인



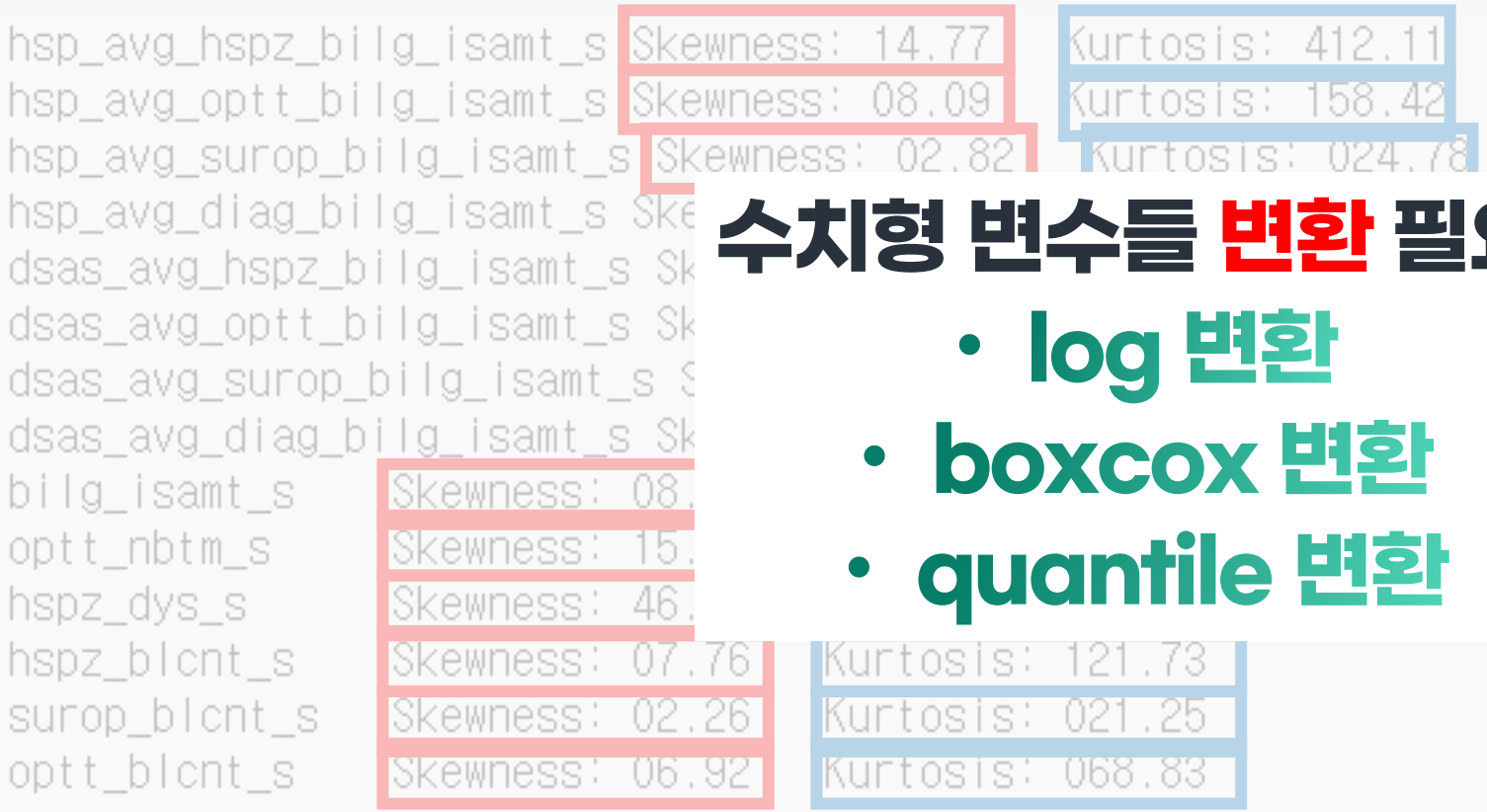
수치형 변수들 왜도 및 첨도 확인



대부분의 수치형 변수들의
왜도가 2 이상

대부분의 수치형 변수들의
첨도가 3 이상

수치형 변수들 왜도 및 첨도 확인



수치형 변수들 변환 필요!

- **log 변환**
- **boxcox 변환**
- **quantile 변환**

대부분의 수치형 변수들의
왜도가 2 이상

대부분의 수치형 변수들의
첨도가 3 이상

수치형 변수들 log, boxcox, quantile 변환 후 왜도 확인

log 변환

```
hsp_avg_hspz_bilg_isamt_s Skewness: -0.03
hsp_avg_optt_bilg_isamt_s Skewness: 00.37
hsp_avg_surop_bilg_isamt_s Skewness: -0.02
hsp_avg_diag_bilg_isamt_s Skewness: 00.28
dsas_avg_hspz_bilg_isamt_s Skewness: 00.65
dsas_avg_optt_bilg_isamt_s Skewness: 01.06
dsas_avg_surop_bilg_isamt_s Skewness: 00.61
dsas_avg_diag_bilg_isamt_s Skewness: 01.04
bilg_isamt_s    Skewness: 00.03
optt_nbtm_s     Skewness: 04.92
hspz_dys_s      Skewness: 00.15
hspz_blcnt_s    Skewness: -0.05
surop_blcnt_s   Skewness: -0.07
optt_blcnt_s    Skewness: 04.51
```

boxcox 변환

```
hsp_avg_hspz_bilg_isamt_s Skewness: 01.67
hsp_avg_optt_bilg_isamt_s Skewness: 01.63
hsp_avg_surop_bilg_isamt_s Skewness: 00.68
hsp_avg_diag_bilg_isamt_s Skewness: 00.75
dsas_avg_hspz_bilg_isamt_s Skewness: 00.65
dsas_avg_optt_bilg_isamt_s Skewness: 01.06
dsas_avg_surop_bilg_isamt_s Skewness: 00.61
dsas_avg_diag_bilg_isamt_s Skewness: 01.04
bilg_isamt_s    Skewness: 02.37
optt_nbtm_s     Skewness: 06.70
hspz_dys_s      Skewness: 02.20
hspz_blcnt_s    Skewness: 00.93
surop_blcnt_s   Skewness: 00.18
optt_blcnt_s    Skewness: 05.24
```

quantile 변환

```
hsp_avg_hspz_bilg_isamt_s Skewness: -0.13
hsp_avg_optt_bilg_isamt_s Skewness: 00.07
hsp_avg_surop_bilg_isamt_s Skewness: -0.09
hsp_avg_diag_bilg_isamt_s Skewness: 00.21
dsas_avg_hspz_bilg_isamt_s Skewness: 00.65
dsas_avg_optt_bilg_isamt_s Skewness: 01.06
dsas_avg_surop_bilg_isamt_s Skewness: 00.61
dsas_avg_diag_bilg_isamt_s Skewness: 01.04
bilg_isamt_s    Skewness: -0.04
optt_nbtm_s     Skewness: 04.92
hspz_dys_s      Skewness: -0.10
hspz_blcnt_s    Skewness: -0.18
surop_blcnt_s   Skewness: -0.11
optt_blcnt_s    Skewness: 04.51
```

변환 후, 왜도가 2이상인 수치형 변수들
왜도가 낮아짐

수치형 변수들 log, boxcox, quantile 변환 후 왜도 확인

log 변환

```
hsp_avg_hspz_bilg_isamt_s Skewness: -0.03
hsp_avg_optt_bilg_isamt_s Skewness: 00.37
hsp_avg_surop_bilg_isamt_s Skewness: -0.02
hsp_avg_diag_bilg_isamt_s Skewness: 00.28
dsas_avg_hspz_bilg_isamt_s Skewness: 00.65
dsas_avg_optt_bilg_isamt_s Skewness: 01.06
dsas_avg_surop_bilg_isamt_s Skewness: 00.61
dsas_avg_diag_bilg_isamt_s Skewness: 01.04
bilg_isamt_s    Skewness: 00.03
optt_nbtm_s     Skewness: 04.92
hspz_dys_s      Skewness: 00.15
hspz_blcnt_s    Skewness: -0.05
surop_blcnt_s   Skewness: -0.07
optt_blcnt_s    Skewness: 04.51
```

boxcox 변환

```
surop_blcnt_s   Skewness: 00.18
optt_blcnt_s    Skewness: 05.24
```

quantile 변환

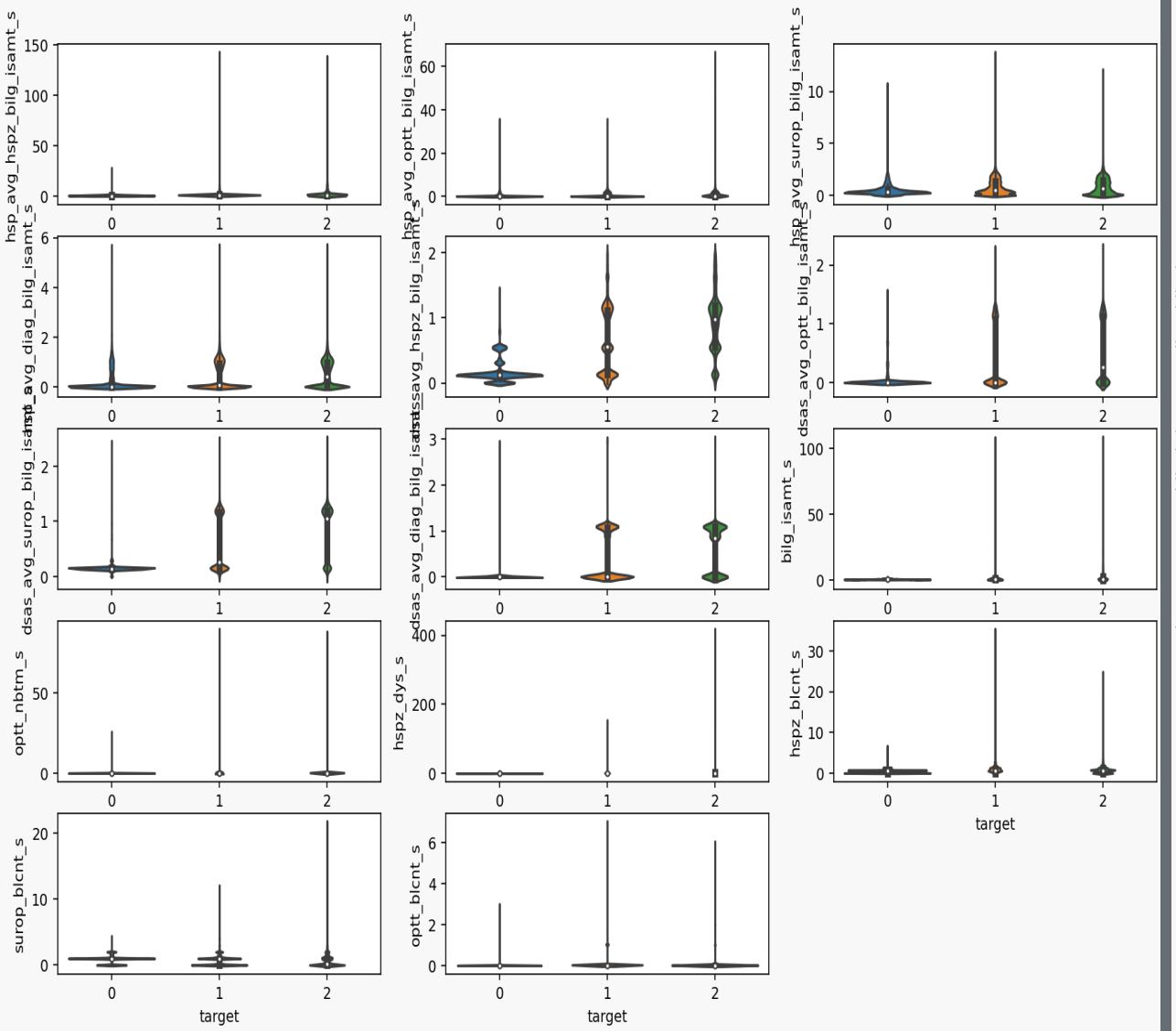
```
hsp_avg_hspz_bilg_isamt_s Skewness: -0.13
hsp_avg_optt_bilg_isamt_s Skewness: 00.07
hsp_avg_surop_bilg_isamt_s Skewness: -0.09
hsp_avg_diag_bilg_isamt_s Skewness: 00.21
dsas_avg_hspz_bilg_isamt_s Skewness: 00.65
dsas_avg_optt_bilg_isamt_s Skewness: 01.06
dsas_avg_surop_bilg_isamt_s Skewness: 00.61
dsas_avg_diag_bilg_isamt_s Skewness: 01.04
bilg_isamt_s    Skewness: -0.04
optt_nbtm_s     Skewness: 04.92
hspz_dys_s      Skewness: -0.10
hspz_blcnt_s    Skewness: -0.18
surop_blcnt_s   Skewness: -0.11
optt_blcnt_s    Skewness: 04.51
```

그 중 log 변환에서 왜도가 크게 감소함

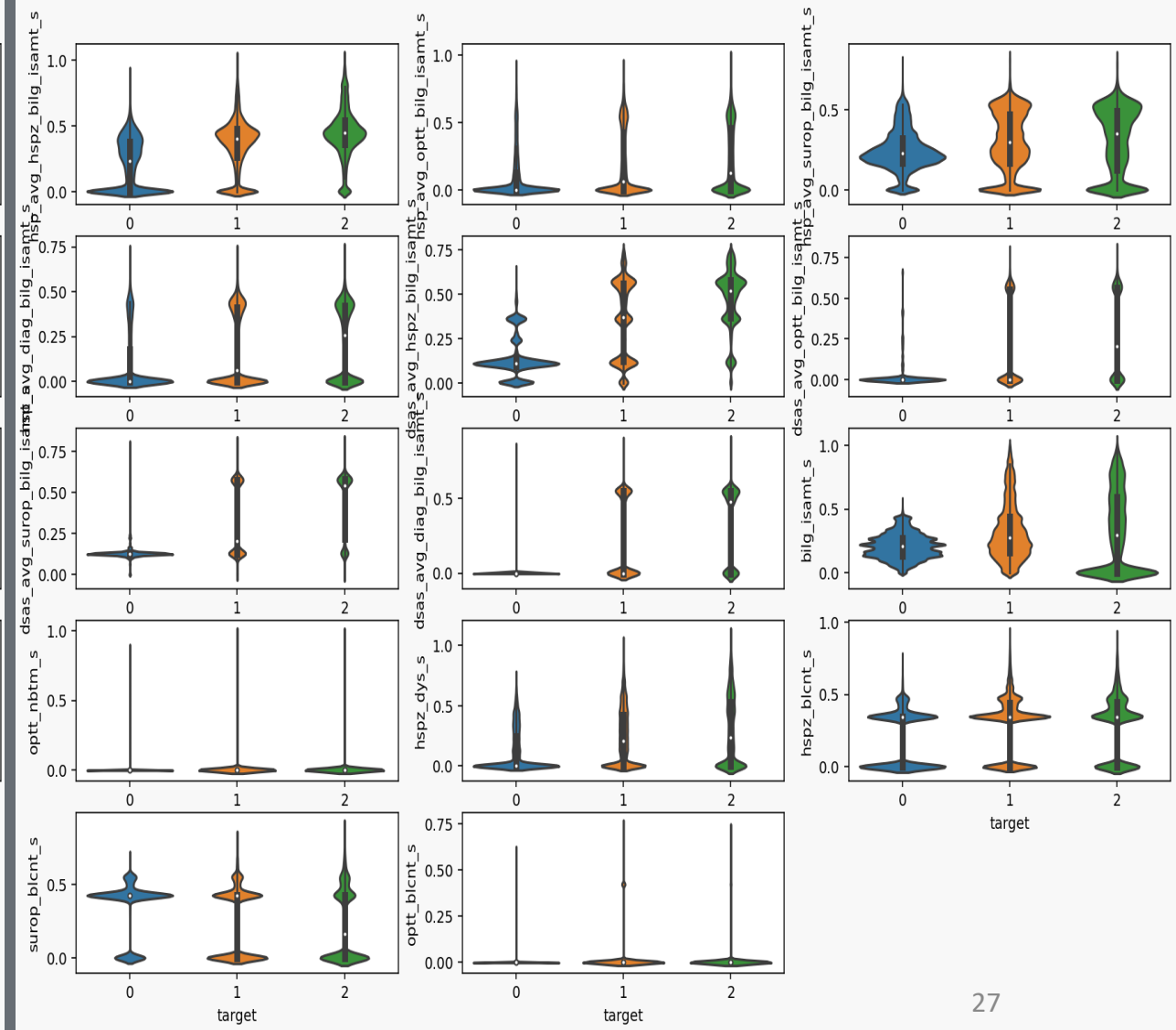
=> log 변환 이용

변환 후, 왜도가 2이상인 수치형 변수들
왜도가 낮아짐

수치형 변수 log 변환 전

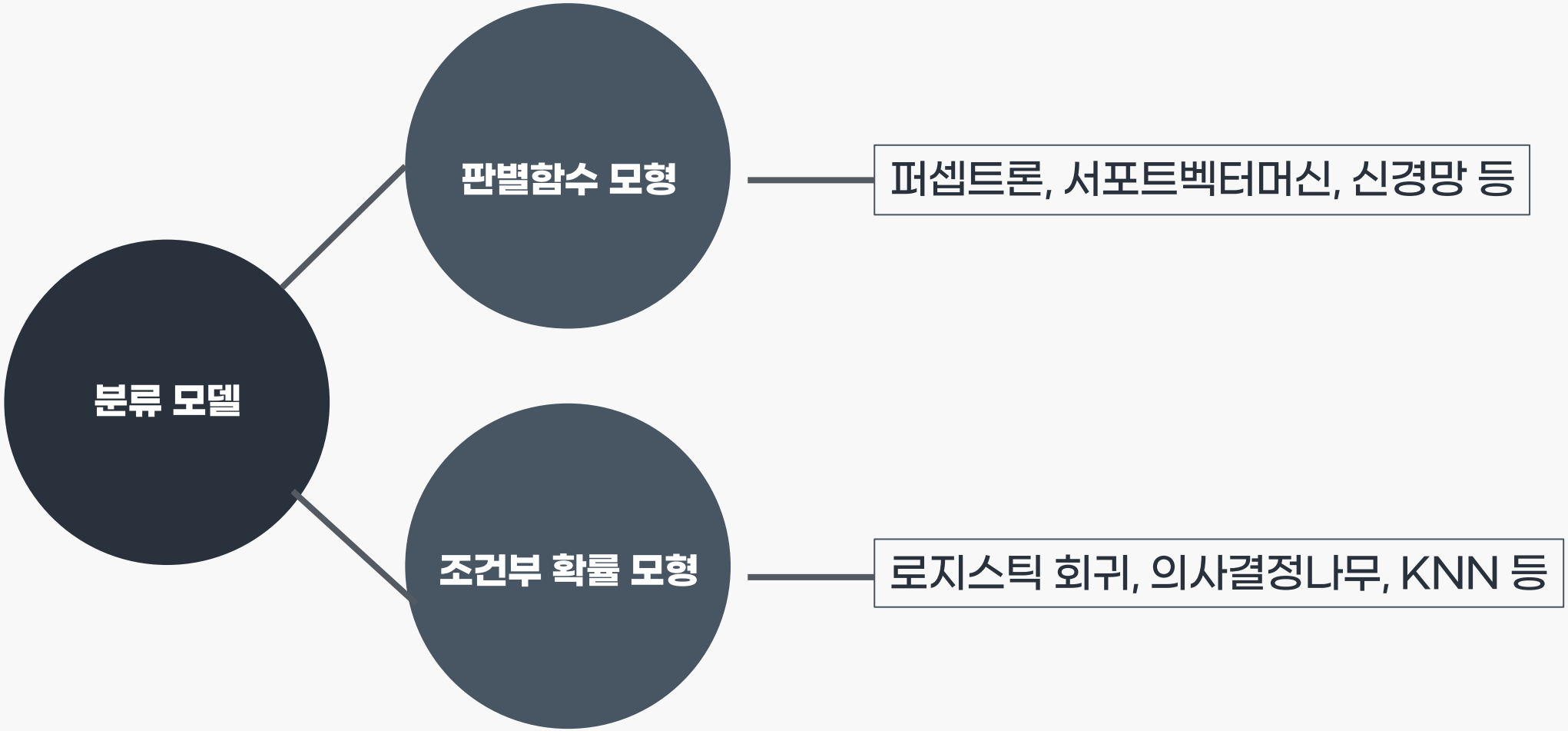


수치형 변수 log 변환 후



3. 모델링

target 변수가
범주형이므로 분류 모델 사용



target 변수가
category형

분류 모델

다중분류모형

(Multiclass Classfication Model)

- 로지스틱 회귀
- 랜덤포레스트
- LGBM
- Catboost

신, 신경망

결정나무, KNN

Catboost 모형

- Ensemble 이란? 여러 모델을 이용하여 데이터를 학습하고,
  모든 모델의 예측 결과를 평균하여 예측
  - error를 최소화하고,
    overfitting을 감소시키며,
    variance를 줄이기 위해 사용

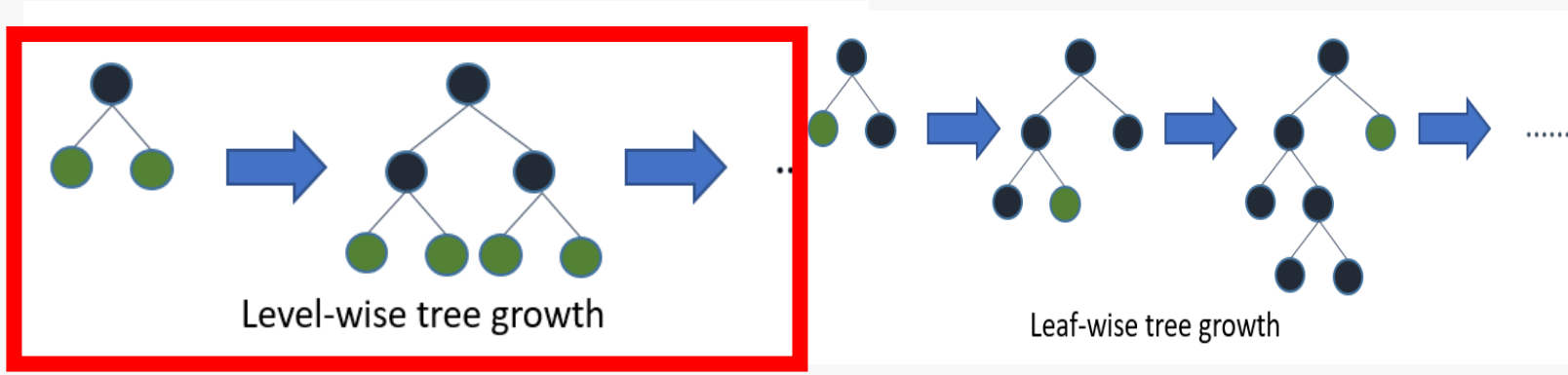
boosting 기법 : Ensemble 기법 중 하나

"More error, More weight"

방법)

1. 실제 값들의 평균과 실제 값의 차이인 잔차(Resicual)를 구한다.
2. 데이터로 이 잔차들을 학습하는 모형을 만든다.
3. 만든 모형으로 예측하여, 예측 값에 Learning_rate 을 곱해
   실제 예측 값(평균 + 잔차예측값 * learning_rate)을 업데이트 한다.
4. 1~3 반복

Catboost 모형



Catboost 모형의 특징

- XGBoost 와 더불어 Catboost는 Level-wise 로 트리를 만들어 나간다.
- 기존 부스팅 과정과 전체적인 양상은 비슷하되, 조금 다르다.
  일부만 가지고 잔차를 계산한 후, 이를 토대로 모델을 만들고,
  그 뒤에 데이터의 잔차는 이 모델로 예측한 값을 사용한다. = "Ordered Boosting"
- Orderd Boosting을 할 때, 데이터 순서를 섞어주지 않으면 매번 같은 순서대로 잔차를 예측하는 모델을 만들 가능성이 있다. 이 순서는 사실 임의로 정한 것이므로, 순서 역시 매번 섞어줘야 한다.
- Catboost는 이러한 것 역시 감안해서 데이터를 셔플링하여 추출한다.
  따라서 트리를 다각적으로 만들 수 있고, 오버피팅을 방지할 수 있다.
- Orderd Target Encoding : 범주형 변수를 수로 인코딩 하는 방법 중 하나.
  현재 데이터의 타겟 값을 사용하지 않고, 이전 데이터들의 타겟 값만 사용하여
  Data Leakage를 막을 수 있다.
  => 오버피팅도 막고 수치값의 다양성도 만들어 줌.
- Categorical Feature Combination, One-hot Encoding, Optimized Parameter tuning

Catboost 모형

1. 변수중요도 낮은 변수들 제거

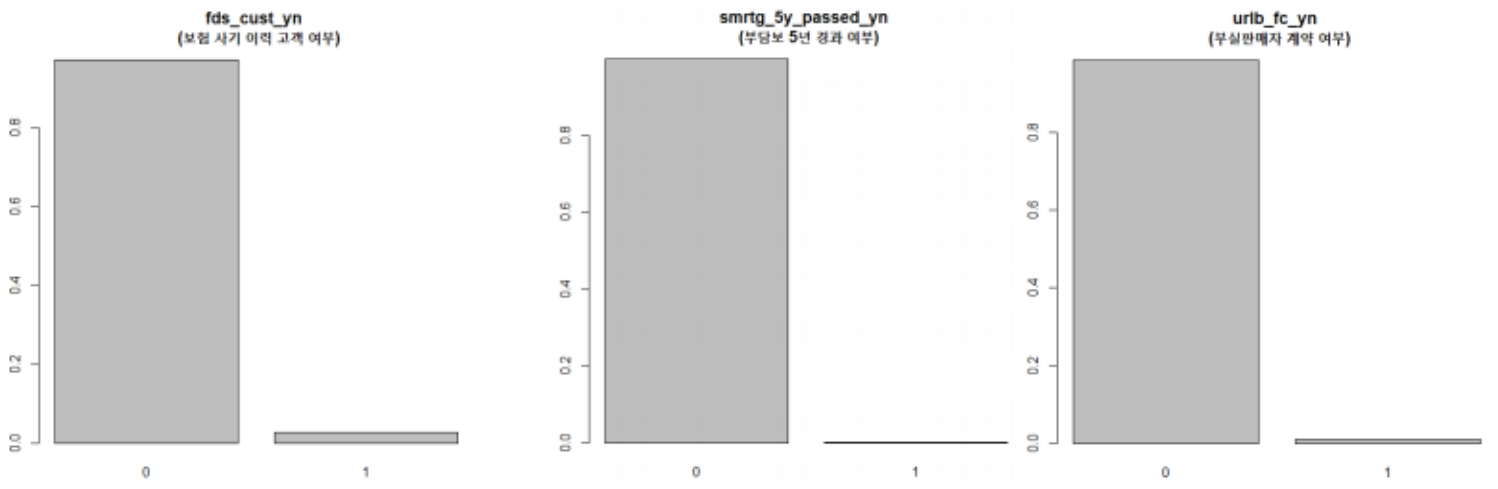
랜덤포레스트 모델을 돌려 변수중요도를 확인하였다.

변수중요도 **하위 5개** 변수들

변수명	변수중요도
heltp_pf_ntyn	4.694356e-03
smrtg_5y_passed_yn	1.659745e-02
urlb_fc_yn	8.269922e-02
mtad_cntr_yn	2.270465e-01
fds_cust_yn	1.487941e+00

Catboost 모형

1. 변수중요도 낮은 변수들 제거



변수중요도 **하위 5개** 변수들 plot을 확인했을 때,
범주가 **매우 불균형**한 것을 확인

따라서 이 변수들은 **제거**하기로 함

Catboost 모형

2. 수치형 변수들 scaling

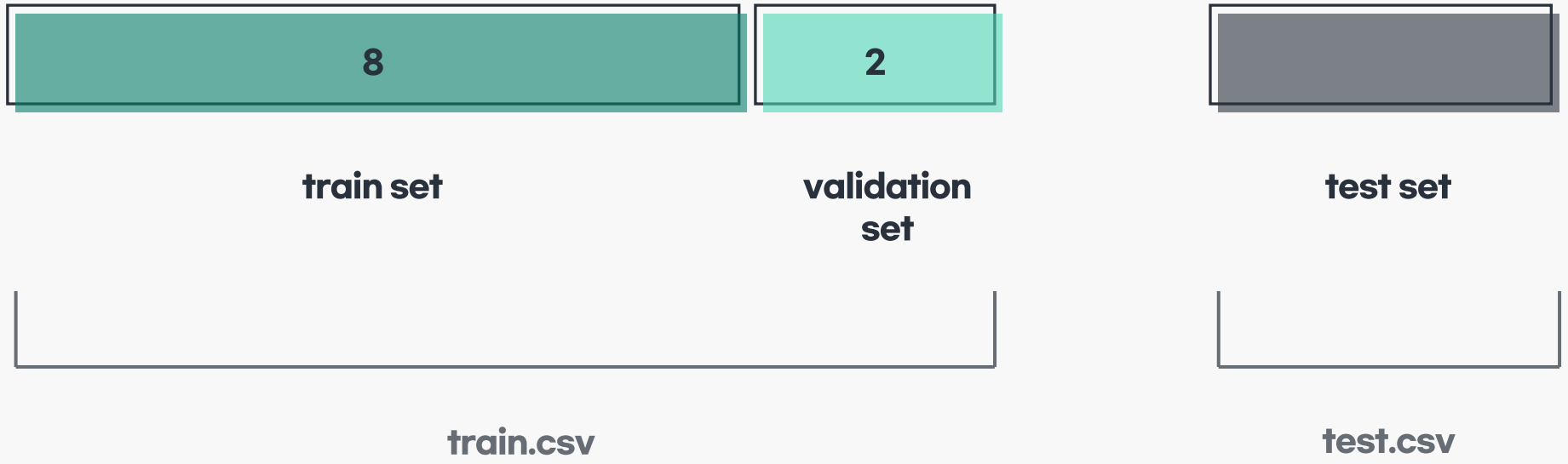
수치형 변수들의 범위가 제각각이라 모델이 잘 학습하지 못할 수 있으므로,
수치형 변수들에 대해 **standard scaling(z-score)** 을 했다.

```
# scaling - z score
num_f =['bilg_isamt_s','hspz_dys_s','hsp_avg_hspz_bilg_isamt_s',
          'hsp_avg_optt_bilg_isamt_s','hsp_avg_surop_bilg_isamt_s','hsp_avg_diag_bilg_isamt_s',
          'dsas_avg_hspz_bilg_isamt_s','dsas_avg_optt_bilg_isamt_s','dsas_avg_surop_bilg_isamt_s',
          'dsas_avg_diag_bilg_isamt_s','hspz_blcnt_s','surop_blcnt_s']

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X[num_f] = scaler.fit_transform(X[num_f])
```

Catboost 모형

3. Catboost 모델링

train set과 validation set을 8:2 로 split했다.



Catboost 모형

3. Catboost 모델링

```
import time
# train CatBoost
start = time.time()
cat_features = np.where(X.dtypes.astype("str").isin(["category","object"]))[0]
CatBoost = CatBoostClassifier(iterations=994,learning_rate=0.01,grow_policy = 'Depthwise',depth=11,
                              loss_function='MultiClass',eval_metric='TotalF1',random_seed=1234)

CatBoost.fit(X = X_train, y = y_train, cat_features = cat_features, eval_set=(X_valid, y_valid))

CatBoost_Runtime = time.time() - start
```

Catboost는 기본 파라미터가 기본적으로 최적화가 잘 되어있지만, **learning_rate, depth**를 수정해서 모델을 개선시킬 수 있었다.

```
980:    learn: 0.9964168        test: 0.9499022 best: 0.9499564 (979)   total: 3h 43m 21s       remaining: 2m 57s
981:    learn: 0.9964366        test: 0.9499014 best: 0.9499564 (979)   total: 3h 43m 33s       remaining: 2m 43s
982:    learn: 0.9964597        test: 0.9499833 best: 0.9499833 (982)   total: 3h 43m 53s       remaining: 2m 30s
983:    learn: 0.9964597        test: 0.9499422 best: 0.9499833 (982)   total: 3h 44m 14s       remaining: 2m 16s
984:    learn: 0.9964696        test: 0.9499965 best: 0.9499965 (984)   total: 3h 44m 37s       remaining: 2m 3s
985:    learn: 0.9964762        test: 0.9499675 best: 0.9499965 (984)   total: 3h 44m 59s       remaining: 1m 49s
986:    learn: 0.9965159        test: 0.9499935 best: 0.9499965 (984)   total: 3h 45m 19s       remaining: 1m 35s
987:    learn: 0.9965159        test: 0.9500472 best: 0.9500472 (987)   total: 3h 45m 34s       remaining: 1m 22s
988:    learn: 0.9965225        test: 0.9499809 best: 0.9500472 (987)   total: 3h 45m 56s       remaining: 1m 8s
989:    learn: 0.9965357        test: 0.9501133 best: 0.9501133 (989)   total: 3h 46m 16s       remaining: 54.9s
990:    learn: 0.9965589        test: 0.9500604 best: 0.9501133 (989)   total: 3h 46m 31s       remaining: 41.1s
991:    learn: 0.9965919        test: 0.9500744 best: 0.9501133 (989)   total: 3h 46m 51s       remaining: 27.4s
992:    learn: 0.9965853        test: 0.9500090 best: 0.9501133 (989)   total: 3h 47m 10s       remaining: 13.7s
993:    learn: 0.9965985        test: 0.9499827 best: 0.9501133 (989)   total: 3h 47m 32s       remaining: 0us

bestTest = 0.9501132957
bestIteration = 989

Shrink model to first 990 iterations.
```

최종 파라미터로 learning_rate = 0.01, depth = 11을 설정한 Catboost 모델 실행 결과 화면

Catboost 모형

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

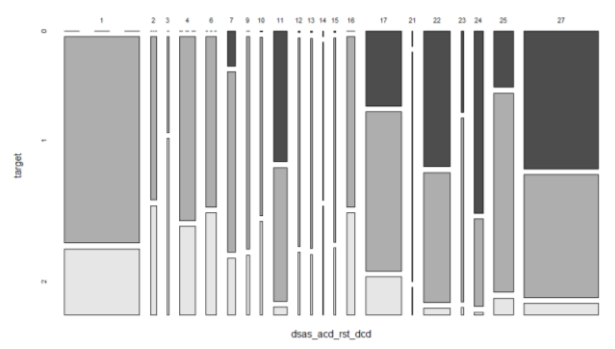
$$F1score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

F1 Score는 0.0 ~ 1.0 사이의 값을 가지며 높을수록 좋습니다.

4. 개별 모델링

모델의 파라미터를 수정해도 점수가 크게 오르지 않았다. 따라서 모델을 개선시키기 위해, 단일 모델만 사용할 것이 아니라, 개별 모델링을 진행해서 F1score를 개선시켰다.

질병구분코드 vs target



질병구분코드별로 target 분포가 다름

재범주화한 질병구분코드별로 개별 모델링

- 질병구분코드 = 0
- 질병구분코드 = 1
- 질병구분코드 = 5
- 질병구분코드 = 2
- 질병구분코드 = 3

질병구분코드 = 1 인 데이터만 넣고 돌린 모델에서 예측한 결과를 대체했을 때 점수가 가장 높게 나옴

결과 및 한계점

결과

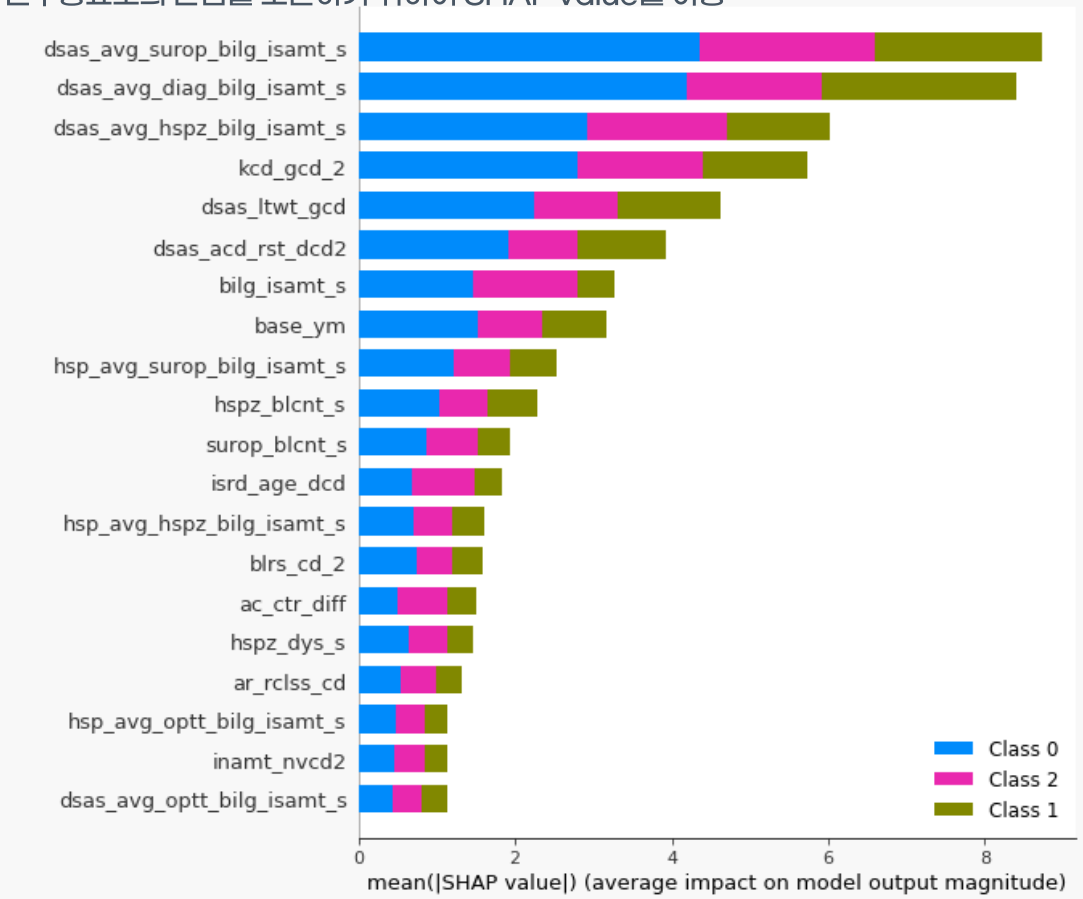
target 예측 결과

학습시킨 catboost 모델에
test.csv를 넣고 예측한 결과이다.

	Target
0	0
1	1
2	1
...	...
22069	0
22070	2
22071	0

SHAP value 확인

변수중요도의 단점을 보완하기 위하여 SHAP value를 이용



리더보드에서 실제 test의 target값과 비교해 나온 점수

최종 점수 : 80.0131

한계점

1. 전처리 과정에서 PCA, 불균형한 범주에 대해 upsampling,
   첨도가 30이상인 변수들은 이상치 제거 후 log 변환하는 방법들을 사용했으나 모두 결과가 더 안좋아져
   다양한 전처리를 하지 못했다.
2. 지역구분정보 코드나 연령구분코드 설명에 존재하지 않은 이상한 값들 (지역구분정보의 경우 6,
   연령구분코드의 경우 7,9)이 test.csv에도 존재했기에 제거할 수 없었고, 정보가 없어서 별도의 처리를 할 수 없었다.
3. 여러 모델(로지스틱 회귀, 랜덤포레스트, 인공신경망, LGBM) 을 돌려봤으나 파라미터를 수정해도 점수가 크게
   변하지 않아 개별 모델링을 진행했다. 질병구분코드별로 target의 분포가 달라서 각각을 설명하는 개별 모델을
   만들어서 단일 모델을 보완하고자 했다.
   하지만, 개별 모델링으로 대체해도 점수가 크게 오르지 않았다.
4. Catboost모델은 해석에 용이하지 않아 자세한 해석이 불가능했다.

Thank You :)

감사합니다!